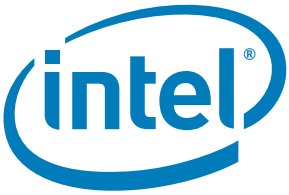


Intel® Ethernet Controller I350 Datasheet

Networking Division (ND)

Features

External Interfaces provided:

- PCIe v2.1 (2.5GT/s and 5GT/s) x4/x2/x1; called PCIe in this document.
- MDI (Copper) standard IEEE 802.3 Ethernet interface for 1000BASE-T, 100BASE-TX, and 10BASE-T applications (802.3, 802.3u, and 802.3ab)
- Serializer-Deserializer (SERDES) to support 1000BASE-SX/LX (optical fiber - IEEE802.3)
- Serializer-Deserializer (SERDES) to support 1000BASE-KX (802.3ap) and 1000BASE-BX (PICMIG 3.1) for Gigabit backplane applications
- SGMII (Serial-GMII Specification) interface for SFP (SFP MSA INF-8074i)/external PHY connections
- NC-SI (DMTF NC-SI) or SMBus for Manageability connection to BMC
- IEEE 1149.6 JTAG

Performance Enhancements:

- PCIe v2.1 TLP Process Hints (TPH)
- UDP, TCP and IP Checksum offload
- UDP and TCP Transmit Segmentation Offload (TSO)
- SCTP receive and transmit checksum offload

Virtualization ready:

- Next Generation VMDq support (8 VMs)
- Support of up to 8 VMs per port (1 queue allocated to each VM)
- PCI-SIG I/O SR-IOV support (Direct assignment)
- Queues per port: 8 TX and 8 RX queues

Power saving features:

- Advanced Configuration and Power Interface (ACPI) power management states and wake-up capability
- Advanced Power Management (APM) wake-up functionality
- Low power link-disconnect state
- PCIe v2.1 LTR
- DMA Coalescing for improved system power management
- EEE (IEEE802.3az) for reduced power consumption during low link utilization periods

IEEE802.1AS - Timing and Synchronization:

- IEEE 1588 Precision Time Protocol support
- Per-packet timestamp

Total Cost Of Ownership (TCO):

- IPMI BMC pass-thru; multi-drop NC-SI
- Internal BMC to OS and OS to BMC traffic support

Additional product details:

- 17x17 (256 Balls) or 25x25 (576 Balls) PBGA package
- Estimated power: 2.8W (max) in dual port mode and 4.2W (max) in quad port mode
- Memories have Parity or ECC protection

Order # 333171-002
Revision 2.4
January 2016

LEGAL

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors which may cause deviations from published specifications.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

* Other names and brands may be claimed as the property of others.

© 2016 Intel Corporation.

Revision History

Rev	Date		Comments
.3	1/8/2010		Initial public release.
.5	5/21/2010		Updated using latest internal specs.
1.0	1/7/2011		Updated using latest internal specs.
1.1	4/6/2011		Updated using latest internal specs.
1.9	4/14/2011		Updated with latest internal specs. Version number moved to 1.9 for PRQ.
1.91	5/6/2011		Added or updated: • Section 6.4.2, Port Identification LED blinking (Word 0x04) • Section 13.1, Thermal Sensor and Thermal Diode • Updated power numbers.
1.92	5/10/2011		Added (improves coverage of 2-port 17X17 package): • Section 2.2.13, 2-Port 17x17 PBGA Package Pin List (Alphabetical) • Section 2.2.14, 2-Port 17x17 PBGA Package No-Connect Pins
1.93	5/20/2011		Updated. • Section 1.6, I350 Packaging Options. Updated to cover both 17x17 options. • Section 11-5, Flash Timing Diagram. Removed meaningless line from diagram. • Section 11.7.1.1, 17x17 PBGA Package Schematics. Corrected display issue with diagram.
2.00	6/23/2011		SRA release. • RSVD_TX_TCLK was expressed as 1.25MHZ (clock speed). Corrected to 125MHz in two places. See Table 2-10, Analog Pins, Table 2-23, PHY Analog Pins. • Section 11.7.2.1, 25x25 PBGA Package Schematics. Diagram updated.
2.01	6/24/2011		• Section 8.5.5, Flow Control Receive Threshold Low - FCRTL0 (0x2160; R/W). Changed: "at least 1b (at least 16 bytes)" to "3b (at least 48 bytes) Diagram updated".
2.02	8/2/2011		• Figure 7-26, Figure 7-26 build issues corrected. • Section 10.6.3.16, Thermal Sensor Commands. Note added ("Thermal Sensor configuration can be done only through NC-SI channel 0.").
2.03	8/25/2011		• Section 6.2.22, Functions Control (Word 0x21), bit 9 note; Section 9.4.11.4, Base Address Register Fields, bit 9 description. Both contain the updated text: "This bit should be set only on systems that do not generate prefetchable cycles." • Section 8.26.1, Internal PHY Configuration - IPCNFG (0x0E38, RW) and Section 8.26.2, PHY Power Management - PHPM (0x0E14, RW); tables reformatted. • Table 10-49, Driver Info Host Command, Byte 1; description updated. • Table 11-6, Power Consumption 2 Ports, D0a - Active Link row, total power column has been corrected.
2.04	9/16/2011		• Section 5.1.1, PCI Device Power States. Section updated. See text starting with "The PCIe link state follows the power management state of the device..." • Section 6.3.11, NC-SI Configuration Module (Global MNG Offset 0x0A). Register descriptions for a number off offsets have been updated. These include: Offsets 0x01, 0x03, 0x05, and 0x07 • Table 8-10, Usable FLASH Size and CSR Mapping Window Size. Table added to Datasheet. • Table 10-30, Supported NC-SI Commands. "Set Ethernet Mac Address" corrected to "Set MAC Address". "Clear Ethernet MAC Address" removed from supported. This is an obsolete reference.

Rev	Date		Comments
2.05	12/20/2011		• Section 6.3.12.2, Traffic Type Data - Offset 0x1. Default values of 01 added for all traffic types. • Section 6.4.9, Reserved/3rd Party External Thermal Sensor – (Word 0x3E). New reserved section added. Section 8.16.28.1, Time Sync Interrupt Cause Register - TSICR (0xB66C; RC/W1C). Note in section updated. New text: "Once ICR.Time_Sync is set, TSICR should be read to determine the actual interrupt cause and to enable reception of an additional ICR.Time_Sync interrupt." • Figure 12-6: Updated to correct error. Section 12.5, Oscillator Support: Contains similar update in the section's first bullet.
2.06	4/10/2012		• Section 3.1.7.9, Completion with Completer Abort (CA). The discussion has been corrected. The updated paragraph is: "A DMA master transaction ending with a Completer Abort (CA) completion causes all PCIe master transactions to stop; the PICAUSE.ABR bit is set and an interrupt is generated if the appropriate mask bits are set. To enable PCIe master transactions following reception of a CA completion, software issues an FLR to the right function or a PCI reset to the device and re-initializes the function(s)." • Section 6.3.9.17, NC-SI over MCTP Configuration - 0ffset 0x10. Phrase in bit 7 description updated. New text: "If cleared, a payload type byte is expected in NC-SI over MCTP packets after the packet type..." • Section 6.4.3, EEPROM Image Revision (Word 0x05). Table updated; bit assignment descriptions changed. Changed to: 15:12 EEPROM major version; 11:8 are reserved; 7:0 EEPROM minor version. Example given in note. • Section 9.6.6.2, LTR Capabilities (0x1C4; RW). The reserved fields (bits 15:13 and 31:29) now indicate RO, not RW. • Figure 11-11 : Coupling cap data in figure corrected; changed 10pf to 1000pf. • Table 12-4, Crystal Manufacturers and Part Numbers. Footnote added to table for 7A25000165. Text states: "This part footprint compatible with X540 designs."

Rev	Date		Comments
2.1	3/22/2013		• Section 2.3.4, NC-SI Interface Pins. Notes added. They specify pull-ups/downs used when NC-SI is disconnected. • Section 7.8.2.2.5, Serial ID. New text provided: "The serial ID capability is not supported in VFs." • Section 8.8.10, Interrupt Cause Set Register - ICS (0x1504; WO). Time Sync (bit 19) exposed. • Table 11-6, Power Consumption 2 Ports. Some numbers updated. See bold copy. • Revised Table 2-15 - 2-Port 17x17 PBGA Package Pin List (Alphabetical); SDP2 and SDP3 connections. • Revised Section 2.3.8 (Power Supply and Ground Pins); removed C4. • Revised Section 2.3.9 (25x25 PBGA Package Pin List (Alphabetical); C4 signal name change. • Revised Section 3.7.6.3.1 (Setting Powerville to Internal PHY loopback Mode); added new bullet. • Revised Section 4.3.5 (Registers and Logic Reset Affects); step 10. • Revised Section 6.2.17 (PCIe Control 1 (Word 0x1B); bit 14 description. • Revised Section 6.3.9.17 (NC-SI over MCTP Configuration - 0ffset 0x10); bit 7 description. • Added Section 6.4.6.11 through Section 6.4.6.18 and (PXE VLAN Configuration Pointer (0x003C) bit descriptions. • Revised Table 8-6 - Register Summary); Management Flex UDP/TCP Ports address. • Revised Section 8.8.9 (Interrupt Cause Read Register - ICR (0x1500; RC/W1C); bit 20 description. • Revised Section 10.5.8.1 (Transmit Errors in Sequence Handling); note after table 10-10. • Revised Section 10.7.1.3 (Simplified MCTP Mode); removed payload type references. • Revised Section 10.7.4.1 (NC-SI Packets Format). • Added Section 10.7.4.1.1 (Control Packets). • Revised Section 10.7.4.1.2 (Command Packets); payload type and message type. • Revised Section 10.7.4.1.3 (Response Packets); payload type and message type. • Revised Section 11.3.1 (Power Supply Specification); added second footnote.
2.2	1/27/14		• Added Section 3.7.6.6, Line Loopback. • Section 6.2.24, Initialization Control 3 (LAN Base Address + Offset 0x24) — Updated description of Com_MDIO field. • Section 8.1.3, Register Summary — Corrected offset value for VFMPRC in Table 8-6. • Section 8.27.3, Register Set - CSR BAR — Corrected Virtual Address and Physical Address Base values for VFMPRC in associated table. • Section 8.28.43, Multicast Packets Received Count - VFMPRC (0x0F38; RO) — Corrected address value. • Section 10.6.2, Supported Features, Table 10-30 — Changed "Supported over MCTP" value from No to Yes for "Select Package" and "Deselect Package" commands. • Figure 11-5 — Changed Output Valid symbol from T_{Val} to T_V to match description in Table 11-15. • Figure 11-6 — Changed Output Valid symbol from T_{Val} to T_V to match description in Table 11-16. • Section 13.5.4, Package Thermal Characteristics — Revised text related to Flotherm* models.
2.3			• Updated Section 6.5.7.2. • Updated Section 11.3.1. • Updated Section 11.3.1.1. • Updated Section 13.5.4.
2.4	January 2016		• Updated note under Table 2-20 (NC-SI Interface Pins; changed pull-up to pull-down for pins NCSI_TXD[1:0]).

NOTE: ***This page intentionally left blank.***

Contents

1 Introduction 37
1.1 Scope 38
1.2 Terminology and Acronyms 38
1.2.1 External Specification and Documents 40
1.2.1.1 Network Interface Documents 40
1.2.1.2 Host Interface Documents 40
1.2.1.3 Networking Protocol Documents 40
1.2.1.4 Manageability Documents 41
1.3 Product Overview 41
1.4 External Interface 41
1.4.1 PCIe Interface 41
1.4.2 Network Interfaces 41
1.4.3 EEPROM Interface 41
1.4.4 Serial Flash Interface 42
1.4.5 SMBus Interface 42
1.4.6 NC-SI Interface 42
1.4.7 MDIO/I^{2}C 2-Wire Interfaces 42
1.4.8 Software-Definable Pins (SDP) Interface (General-Purpose I/O) 43
1.4.9 LEDs Interface 43
1.5 Features 43
1.6 I350 Packaging Options 47
1.7 Overview of Changes Compared to the 82580 48
1.7.1 Network Interface 48
1.7.1.1 Energy Efficient Ethernet (IEEE802.3AZ) 48
1.7.1.2 MDI Flip 48
1.7.2 Virtualization 48
1.7.2.1 PCI SR IOV 48
1.7.2.2 Promiscuous VLAN Filtering 49
1.7.2.3 Improvements to VMDq Switching 49
1.7.2.3.1 Promiscuous Modes 49
1.7.2.3.2 Microsoft NLB Mode Support 49
1.7.2.4 Number of Exact Match Filters 49
1.7.2.5 Support for 2K Header Buffer 49
1.7.2.6 Support for Port Based VLAN 49
1.7.2.7 Header Split on L2 Header 49
1.7.2.8 Updated Pool Decision Algorithm 49
1.7.2.9 Statistics 49
1.7.3 HOST Interface 50
1.7.3.1 MSI-X Support 50
1.7.3.2 ID-Based Ordering 50
1.7.3.3 Alternative Routing-ID Interpretation (ARI) 50
1.7.3.4 Link State Related Latency Tolerance Reporting (LTR) 50
1.7.3.5 Access Control Services (ACS) 50
1.7.3.6 ASPM Optionality Compliance Capability 50
1.7.4 Manageability 51
1.7.4.1 Auto-ARP Reply on SMBus 51
1.7.4.2 NC-SI Commands. 51
1.7.4.2.1 NC-SI Hardware Arbitration 51
1.7.4.3 OS to BMC Traffic 51
1.7.4.4 DMTF MCTP Protocol Over SMBus 51
1.7.4.5 Proxying 51
1.7.5 EEPROM Structures 52
1.7.6 Recovery from Memory Error 52
1.7.7 BOM Cost Reduction 52
1.7.7.1 On-Chip 1.8V LVR Control 52
1.7.7.2 On-Chip 1.0V SVR Control 52
1.7.7.3 Thermal Sensor 52
1.8 Device Data Flows 53
1.8.1 Transmit Data Flow 53
1.8.2 Receive Data Flow 54

2 Pin Interface 55
2.1 Signal Type Notation 55
2.2 17x17 PBGA Package Pin Assignment 55
2.2.1 PCIe 55
2.2.2 Flash and EEPROM Ports 56
2.2.3 System Management Bus (SMB) Interface 57
2.2.4 NC-SI Interface Pins 57
2.2.5 Miscellaneous Pins 58

2.2.6 SERDES/SGMII Pins 59
2.2.7 SFP Pins 61
2.2.8 PHY Pins 62
2.2.8.1 LED's 62
2.2.8.2 PHY Analog Pins 63
2.2.9 Voltage Regulator Pins 65
2.2.10 Testability Pins 65
2.2.11 Power Supply and Ground Pins 67
2.2.12 4-Port 17x17 PBGA Package Pin List (Alphabetical) 67
2.2.13 2-Port 17x17 PBGA Package Pin List (Alphabetical) 70
2.2.14 2-Port 17x17 PBGA Package No-Connect Pins 72
2.3 25x25 PBGA Package Pin Assignment 72
2.3.1 PCIe 72
2.3.2 Flash and EEPROM Ports 73
2.3.3 System Management Bus (SMB) Interface 73
2.3.4 NC-SI Interface Pins 74
2.3.5 Miscellaneous Pins 74
2.3.6 PHY Pins 75
2.3.6.1 LED's 75
2.3.6.2 PHY Analog Pins 76
2.3.7 Testability Pins 77
2.3.8 Power Supply and Ground Pins 78
2.3.9 25x25 PBGA Package Pin List (Alphabetical) 79
2.4 Pullups/Pulldowns 84
2.5 Strapping 87
2.6 Interface Diagram 89
2.7 17x17 PBGA Package Ball-Out 89
2.8 25x25 PBGA Package Ball-Out 91

3 Interconnects 93

3.1 PCIe 93
3.1.1 PCIe Overview 93
3.1.1.1 Architecture, Transaction and Link Layer Properties 94
3.1.1.2 Physical Interface Properties 95
3.1.1.3 Advanced Extensions 95
3.1.2 Functionality - General 95
3.1.2.1 Native/Legacy 95
3.1.2.2 Locked Transactions 95
3.1.3 Host Interface 96
3.1.3.1 Tag IDs 96
3.1.3.1.1 TAG ID Allocation for Read Transactions 96
3.1.3.1.2 TAG ID Allocation for Write Transactions 97
3.1.3.2 Completion Timeout Mechanism 98
3.1.3.2.1 Completion Timeout Period 99
3.1.4 Transaction Layer 99
3.1.4.1 Transaction Types Accepted by the I350 99
3.1.4.1.1 Configuration Request Retry Status 100
3.1.4.1.2 Partial Memory Read and Write Requests 100
3.1.4.2 Transaction Types Initiated by the I350 101
3.1.4.2.1 Data Alignment 101
3.1.4.2.2 Multiple Tx Data Read Requests (MULR) 101
3.1.4.3 Messages 102
3.1.4.3.1 Message Handling by the I350 (as a Receiver) 102
3.1.4.3.2 Message Handling by the I350 (as a Transmitter) 102
3.1.4.4 Ordering Rules 103
3.1.4.4.1 Out of Order Completion Handling 103
3.1.4.5 Transaction Definition and Attributes 103
3.1.4.5.1 Max Payload Size 103
3.1.4.5.2 Relaxed Ordering 104
3.1.4.5.3 Snoop Not Required 104
3.1.4.5.4 No Snoop and Relaxed Ordering for LAN Traffic 104
3.1.4.5.5 TLP processing Hint (TPH) 105
3.1.4.6 Flow Control 105
3.1.4.6.1 I350 Flow Control Rules 105
3.1.4.6.2 Upstream Flow Control Tracking 106
3.1.4.6.3 Flow Control Update Frequency 106
3.1.4.6.4 Flow Control Timeout Mechanism 106
3.1.4.7 Error Forwarding 106
3.1.5 Data Link Layer 107
3.1.5.1 ACK/NAK Scheme 107
3.1.5.2 Supported DLLPs 107
3.1.5.3 Transmit EDB Nullifying 108
3.1.6 Physical Layer 108
3.1.6.1 Link Speed 108
3.1.6.2 Link Width 108

3.1.6.3 Polarity Inversion 109
3.1.6.4 L0s Exit latency 109
3.1.6.5 Lane-to-Lane De-Skew 109
3.1.6.6 Lane Reversal 109
3.1.6.7 Reset 110
3.1.6.8 Scrambler Disable 110
3.1.7 Error Events and Error Reporting 111
3.1.7.1 Mechanism in General 111
3.1.7.2 Error Events 111
3.1.7.3 Error Forwarding (TLP poisoning) 113
3.1.7.4 ECRC 113
3.1.7.5 Partial Read and Write Requests 113
3.1.7.6 Error Pollution 114
3.1.7.7 Completion with Unsuccessful Completion Status 114
3.1.7.8 Error Reporting Changes 114
3.1.7.9 Completion with Completer Abort (CA) 115
3.1.8 PCIe Power Management 115
3.1.9 PCIe Programming Interface 115
3.2 Management Interfaces 115
3.2.1 SMBus 115
3.2.1.1 Channel Behavior 116
3.2.2 NC-SI 116
3.2.2.1 Electrical Characteristics 116
3.2.2.2 NC-SI Transactions 116
3.3 Flash / EEPROM 117
3.3.1 EEPROM Interface 117
3.3.1.1 General Overview 117
3.3.1.2 EEPROM Device 117
3.3.1.3 HW Initial Load Process. 118
3.3.1.4 Software Accesses 121
3.3.1.5 EEPROM Detection and Signature Field 122
3.3.1.5.1 EEPROM Detection 122
3.3.1.5.2 Detection of Valid EEPROM Image 122
3.3.1.6 Protected EEPROM Space 123
3.3.1.6.1 Initial EEPROM Programming 123
3.3.1.6.2 Activating the Protection Mechanism 123
3.3.1.6.3 Non Permitted Accessing to Protected Areas in the EEPROM 123
3.3.1.7 EEPROM Recovery 124
3.3.1.8 EEPROM-Less Support 125
3.3.1.8.1 Access to the EEPROM Controlled Feature 125
3.3.2 Shared EEPROM 126
3.3.2.1 EEPROM Deadlock Avoidance 126
3.3.2.2 EEPROM Map Shared Words 127
3.3.3 Vital Product Data (VPD) Support 127
3.3.4 Flash Interface 129
3.3.4.1 Flash Interface Operation 129
3.3.4.2 Flash Write Control 129
3.3.4.3 Flash Erase Control 130
3.3.5 Shared FLASH 130
3.3.5.1 Flash Access Contention 130
3.3.5.2 Flash Deadlock Avoidance 130
3.4 Configurable I/O Pins 131
3.4.1 General-Purpose I/O (Software-Definable Pins) 131
3.4.1.1 SDP usage for SFP connectivity 132
3.4.2 Software Watchdog 132
3.4.2.1 Watchdog Rearm 132
3.4.3 LEDs 133
3.5 Voltage Regulators 133
3.5.1 1.8V LVR Control 134
3.5.2 1.0V SVR Control 135
3.6 Thermal Sensor 135
3.6.1 Initializing Thermal Sensor 136
3.6.2 Firmware Based Thermal Management 137
3.6.3 Thermal Sensor Diagnostics 137
3.6.4 Thermal Sensor Characteristics 137
3.7 Network Interfaces 138
3.7.1 Overview 138
3.7.2 MAC Functionality 139
3.7.2.1 Internal GMII/MII Interface 139
3.7.2.2 MDIO/MDC PHY Management Interface 139
3.7.2.2.1 Detection of External I2C or MDIO Connection 140
3.7.2.2.2 MDIC and MDICNFG register usage 140
3.7.2.3 Duplex Operation with Copper PHY 141
3.7.2.3.1 Full Duplex 141
3.7.2.3.2 Half Duplex 141

3.7.3 SerDes/1000BASE-BX, SGMII and 1000BASE-KX Support 142
3.7.3.1 SerDes/1000BASE-BX, SGMII and 1000BASE-KX Analog Block 142
3.7.3.2 SerDes/1000BASE-BX, SGMII and 1000BASE-KX PCS Block 142
3.7.3.3 GbE Physical Coding Sub-Layer (PCS) 143
3.7.3.3.1 8B10B Encoding/Decoding 143
3.7.3.3.2 Code Groups and Ordered Sets 143
3.7.4 Auto-Negotiation and Link Setup Features 144
3.7.4.1 SerDes/1000BASE-BX Link Configuration 145
3.7.4.1.1 Signal Detect Indication 145
3.7.4.1.2 MAC Link Speed 145
3.7.4.1.3 SerDes/1000BASE-BX Mode Auto-Negotiation 145
3.7.4.1.4 Forcing Link-up in SerDes/1000BASE-BX Mode 146
3.7.4.1.5 HW Detection of Non-Auto-Negotiation Partner 146
3.7.4.2 1000BASE-KX Link Configuration 146
3.7.4.2.1 MAC Link Speed 147
3.7.4.2.2 1000BASE-KX Auto-Negotiation 147
3.7.4.2.3 Forcing Link-up in 1000BASE-KX Mode 147
3.7.4.2.4 1000BASE-KX HW Detection of Link Partner 147
3.7.4.3 SGMII Link Configuration 147
3.7.4.3.1 SGMII Auto-Negotiation 147
3.7.4.3.2 Forcing Link in SGMII mode 148
3.7.4.3.3 MAC Speed Resolution 148
3.7.4.4 Copper PHY Link Configuration 148
3.7.4.4.1 PHY Auto-Negotiation (Speed, Duplex, Flow Control) 148
3.7.4.4.2 MAC Speed Resolution 149
3.7.4.4.3 MAC Full-/Half- Duplex Resolution 150
3.7.4.4.4 Using PHY Registers 150
3.7.4.4.5 Comments Regarding Forcing Link 150
3.7.4.5 Loss of Signal/Link Status Indication 150
3.7.5 Ethernet Flow Control (FC) 151
3.7.5.1 MAC Control Frames and Receiving Flow Control Packets 151
3.7.5.1.1 Structure of 802.3X FC Packets 151
3.7.5.1.2 Operation and Rules 152
3.7.5.1.3 Timing Considerations 153
3.7.5.2 PAUSE and MAC Control Frames Forwarding 153
3.7.5.3 Transmission of PAUSE Frames 153
3.7.5.3.1 Operation and Rules 154
3.7.5.3.2 Software Initiated PAUSE Frame Transmission 155
3.7.5.4 IPG Control and Pacing 155
3.7.5.4.1 Fixed IPG Extension 155
3.7.6 Loopback Support 155
3.7.6.1 General 155
3.7.6.2 MAC Loopback 156
3.7.6.2.1 Setting the I350 to MAC Loopback Mode 156
3.7.6.3 Internal PHY Loopback 156
3.7.6.3.1 Setting the I350 to Internal PHY loopback Mode 156
3.7.6.4 SerDes, SGMII and 1000BASE-KX Loopback 157
3.7.6.4.1 Setting SerDes/1000BASE-BX, SGMII, 1000BASE-KX Loopback Mode 157
3.7.6.5 External PHY Loopback 158
3.7.6.5.1 Setting the I350 Internal PHY to External Loopback Mode 158
3.7.6.6 Line Loopback 158
3.7.7 Energy Efficient Ethernet (EEE) 159
3.7.7.1 Conditions to Enter EEE TX LPI 160
3.7.7.2 Exit of TX LPI to Active Link State 160
3.7.7.3 EEE Auto-Negotiation 161
3.7.7.4 EEE Link Level (LLDP) Capabilities Discovery 161
3.7.7.5 Programming the I350 for EEE Operation 162
3.7.7.6 EEE Statistics 163
3.7.8 Integrated Copper PHY Functionality 163
3.7.8.1 Determining Link State 163
3.7.8.1.1 False Link 164
3.7.8.1.2 Forced Operation 165
3.7.8.1.3 Auto Negotiation 165
3.7.8.1.4 Parallel Detection 165
3.7.8.1.5 Auto Cross-Over 166
3.7.8.1.6 10/100 MB/s Mismatch Resolution 166
3.7.8.1.7 Link Criteria 167
3.7.8.2 Link Enhancements 167
3.7.8.2.1 SmartSpeed 168
3.7.8.3 Flow Control 168
3.7.8.4 Management Data Interface 169
3.7.8.5 Internal PHY Low Power Operation and Power Management 169
3.7.8.5.1 Power Down via the PHY Register 169
3.7.8.5.2 Power Management State 169
3.7.8.5.3 Disable High Speed Power Saving Options 170

3.7.8.5.4 Low Power Link Up - Link Speed Control 170
3.7.8.5.5 Internal PHY Smart Power-Down (SPD) 172
3.7.8.5.6 Internal PHY Link Energy Detect 173
3.7.8.5.7 Internal PHY Power-Down State 173
3.7.8.6 Advanced Diagnostics 173
3.7.8.6.1 TDR - Time Domain Reflectometry 173
3.7.8.6.2 Channel Frequency Response 174
3.7.8.7 1000 Mb/s Operation 174
3.7.8.7.1 Introduction 174
3.7.8.7.2 Transmit Functions 175
3.7.8.7.3 Receive Functions 177
3.7.9 Media Auto Sense 180
3.7.9.1 Auto sense setup 180
3.7.9.1.1 SerDes/SGMII/1000BASE-KX Detect Mode (PHY is Active) 180
3.7.9.1.2 PHY Detect Mode (SerDes/SGMII/1000BASE-KX is active) 181
3.7.9.2 Switching between medias. 181
3.7.9.2.1 Transition to SerDes/1000BASE-KX/SGMII Modes 181
3.7.9.2.2 Transition to Internal PHY Mode 181

4 Initialization 183

4.1 Power Up 183
4.1.1 Power-Up Sequence 183
4.1.2 Power-Up Timing Diagram 184
4.2 Reset Operation 185
4.2.1 Reset Sources 185
4.2.1.1 LAN_PWR_GOOD 185
4.2.1.2 PE_RST_N 185
4.2.1.3 In-Band PCIe Reset 185
4.2.1.4 D3hot to D0 Transition 185
4.2.1.5 Function Level Reset (FLR) 186
4.2.1.5.1 PF (Physical Function) FLR or FLR in non-IOV Mode 186
4.2.1.5.2 VF (Virtual Function) FLR (Function Level Reset) 186
4.2.1.5.3 IOV (IO Virtualization) Disable 186
4.3 Software Reset 187
4.3.1 Full Port Software Reset (RST) 187
4.3.2 Physical Function (PF) Software Reset 187
4.3.3 VF Software Reset 187
4.3.4 Device Software Reset (DEV_RST) 188
4.3.4.1 BME (Bus Master Enable) 188
4.3.4.2 Force TCO 189
4.3.4.3 EEPROM Reset 189
4.3.4.4 PHY Reset 189
4.3.5 Registers and Logic Reset Affects 191
4.3.6 PHY Behavior During a Manageability Session 195
4.4 Function Disable 196
4.4.1 General 196
4.4.2 Overview 196
4.4.3 Disabling Both LAN Port and PCIe Function 197
4.4.4 Disabling PCIe Function Only 197
4.4.5 PCIe Functions to LAN Ports Mapping 197
4.4.6 Control Options 199
4.4.7 Event Flow for Enable/Disable Functions 200
4.4.7.1 Multi-Function Advertisement 201
4.4.7.2 Legacy Interrupts Utilization 201
4.4.7.3 Power Reporting 201
4.5 Device Disable 201
4.5.1 BIOS Handling of Device Disable 202
4.6 Software Initialization and Diagnostics 202
4.6.1 Introduction 202
4.6.2 Power Up State 203
4.6.3 Initialization Sequence 203
4.6.4 Interrupts During Initialization 203
4.6.5 Global Reset and General Configuration 203
4.6.6 Flow Control Setup 204
4.6.7 Link Setup Mechanisms and Control/Status Bit Summary 204
4.6.7.1 PHY Initialization 204
4.6.7.2 MAC/PHY Link Setup (CTRL_EXT.LINK_MODE = 00b) 204
4.6.7.2.1 MAC Settings Automatically Based on Duplex and Speed Resolved by PHY (CTRL.FRCDPLX = 0b, CTRL.FRCSPD = 0b) 204
4.6.7.2.2 MAC Duplex and Speed Settings Forced by Software Based on Resolution of PHY (CTRL.FRCDPLX = 1b, CTRL.FRCSPD = 1b) 205
4.6.7.2.3 MAC/PHY Duplex and Speed Settings Both Forced by Software (Fully-Forced Link Setup) (CTRL.FRCDPLX = 1b, CTRL.FRCSPD = 1b, CTRL.SLU = 1b) 205

4.6.7.3 MAC/SERDES Link Setup (CTRL_EXT.LINK_MODE = 11b) 206
4.6.7.3.1 Hardware Auto-Negotiation Enabled (PCS_LCTL. AN ENABLE = 1b; CTRL.FRCSPD = 0b; CTRL.FRCDPLX = 0) 206
4.6.7.3.2 Auto-Negotiation Skipped (PCS_LCTL. AN ENABLE = 0b; CTRL.FRCSPD = 1b; CTRL.FRCDPLX = 1) 206
4.6.7.4 MAC/SGMII Link Setup (CTRL_EXT.LINK_MODE = 10b) 207
4.6.7.4.1 Hardware Auto-Negotiation Enabled (PCS_LCTL. AN ENABLE = 1b, CTRL.FRCDPLX = 0b, CTRL.FRCSPD = 0b) 207
4.6.7.5 MAC/1000BASE-KX Link Setup (CTRL_EXT.LINK_MODE = 01b) 207
4.6.7.5.1 Auto-Negotiation Skipped (PCS_LCTL. AN ENABLE = 0b; CTRL.FRCSPD = 1b; CTRL.FRCDPLX = 1) 207
4.6.8 Initialization of Statistics 208
4.6.9 Receive Initialization 208
4.6.9.1 Initialize the Receive Control Register 209
4.6.9.2 Dynamic Enabling and Disabling of Receive Queues 209
4.6.10 Transmit Initialization 210
4.6.10.1 Dynamic Queue Enabling and Disabling 210
4.6.11 Virtualization Initialization Flow 210
4.6.11.1 VMDq Mode 210
4.6.11.1.1 Global Filtering and Offload Capabilities 210
4.6.11.1.2 Mirroring rules. 211
4.6.11.1.3 Per Pool Settings 211
4.6.11.1.4 Security Features 212
4.6.11.2 IOV Initialization 212
4.6.11.2.1 PF Driver Initialization 212
4.6.11.2.2 VF Driver Initialization 212
4.6.11.2.3 Full Reset Coordination 213
4.6.11.2.4 VFRE/VFTE 213
4.6.12 Alternate MAC Address Support 214
4.7 Access to Shared Resources 214
4.7.1 Acquiring Ownership Over a Shared Resource 215
4.7.2 Releasing Ownership Over a Shared Resource 215
4.7.3 Software to Software Mailbox 216

5 Power Management 217

5.1 General Power State Information 217
5.1.1 PCI Device Power States 217
5.1.2 PCIe Link Power States 218
5.2 Power States 218
5.2.1 D0 Uninitialized State (D0u) 219
5.2.1.1 Entry into D0u State 219
5.2.2 D0active State 220
5.2.2.1 Entry to D0a State 220
5.2.3 D3 State (PCI-PM D3hot) 220
5.2.3.1 Entry to D3 State 220
5.2.3.2 Exit from D3 State 221
5.2.3.3 Master Disable Via CTRL Register 222
5.2.4 Dr State (D3cold) 222
5.2.4.1 Dr Disable Mode 223
5.2.4.2 Entry to Dr State 223
5.2.4.3 Auxiliary Power Usage 224
5.2.5 Link Disconnect 224
5.2.6 Device Power-Down State 224
5.3 Power Limits by Certain Form Factors 225
5.4 Interconnects Power Management 225
5.4.1 PCIe Link Power Management 226
5.4.2 NC-SI Clock Control 228
5.4.3 Internal PHY Power-Management 228
5.5 Timing of Power-State Transitions 228
5.5.1 Power Up (Off to Dup to D0u to D0a 229
5.5.2 Transition from D0a to D3 and Back Without PE_RST_N 230
5.5.3 Transition From D0a to D3 and Back With PE_RST_N 231
5.5.4 Transition From D0a to Dr and Back Without Transition to D3 232
5.6 Wake Up 233
5.6.1 Advanced Power Management Wake Up 233
5.6.2 ACPI Power Management Wake Up 234
5.6.3 Wake-Up and Proxying Filters 235
5.6.3.1 Pre-Defined Filters 235
5.6.3.1.1 Directed Exact Packet 235
5.6.3.1.2 Directed Multicast Packet 236
5.6.3.1.3 Broadcast 236
5.6.3.1.4 Magic Packet 236
5.6.3.1.5 ARP/IPv4 Request Packet 237

5.6.3.1.6 Directed IPv4 Packet 238
5.6.3.1.7 Directed IPv6 Packet 239
5.6.3.1.8 NS and MLD IPv6 Packets 239
5.6.3.2 Flexible Filters 241
5.6.3.2.1 IPX Diagnostic Responder Request Packet 241
5.6.3.2.2 Directed IPX Packet 241
5.6.3.2.3 Utilizing Flex Wake-Up Filters In Normal Operation 242
5.6.3.3 Wake Up Packet Storage 242
5.6.4 Wake-up and Virtualization 242
5.7 Protocol Offload (Proxying) 242
5.7.1 Proxying and Virtualization 243
5.7.2 Protocol Offload Activation in D3 243
5.7.3 Protocol Offload Activation in D0 245
5.8 DMA Coalescing 246
5.8.1 DMA Coalescing Activation 246
5.8.2 Entering DMA Coalescing Operating Mode 248
5.8.2.1 Entering DMA Coalescing 248
5.8.3 Conditions to Exit DMA Coalescing 248
5.8.3.1 Exiting DMA Coalescing 248
5.9 Latency Tolerance Reporting (LTR) 249
5.9.1 Latency Tolerance Reporting Algorithm 250
5.9.2 Latency Tolerance Reporting Per Function 252
5.9.2.1 Conditions for Generating LTR Message with the Requirement Bits Cleared 252
5.9.2.2 Conditions for Generating LTR Message with Maximum LTR Value 253
5.9.2.3 Conditions for Generating LTR Message with Minimum LTR Value 253

6 Non-Volatile Memory Map - EEPROM 255

6.1 EEPROM General Map 255
6.2 Hardware Accessed Words 258
6.2.1 Ethernet Address (LAN Base Address + Offsets 0x00-0x02) 259
6.2.2 Initialization Control Word 1 (word 0x0A) 259
6.2.3 Subsystem ID (Word 0x0B) 260
6.2.4 Subsystem Vendor ID (Word 0x0C) 260
6.2.5 Device ID (LAN Base Address + Offset 0x0D) 260
6.2.6 Vendor ID (Word 0x0E) 260
6.2.7 Dummy Device ID (Word 0x1D) 261
6.2.8 Initialization Control Word 2 (LAN Base Address + Offset 0x0F) 261
6.2.9 EEPROM Sizing and Protected Fields (Word 0x12) 262
6.2.10 Initialization Control 4 (LAN Base Address + Offset 0x13) 263
6.2.11 PCIe L1 Exit latencies (Word 0x14) 264
6.2.12 PCIe Completion Timeout Configuration (Word 0x15) 265
6.2.13 MSI-X Configuration (LAN Base Address + Offset 0x16) 265
6.2.14 PCIe Init Configuration 1 (Word 0x18) 265
6.2.15 PCIe Init Configuration 2 Word (Word 0x19) 266
6.2.16 PCIe Init Configuration 3 Word (Word 0x1A) 266
6.2.17 PCIe Control 1 (Word 0x1B) 267
6.2.18 LED 1,3 Configuration Defaults (LAN Base Address + Offset 0x1C) 268
6.2.19 Device Rev ID (Word 0x1E) 268
6.2.20 LED 0,2 Configuration Defaults (LAN Base Address + Offset 0x1F) 269
6.2.21 Software Defined Pins Control (LAN Base Address + Offset 0x20) 270
6.2.22 Functions Control (Word 0x21) 271
6.2.23 LAN Power Consumption (Word 0x22) 272
6.2.24 Initialization Control 3 (LAN Base Address + Offset 0x24) 273
6.2.25 I/O Virtualization (IOV) Control (Word 0x25) 274
6.2.26 IOV Device ID (Word 0x26) 275
6.2.27 PCIe Control 2 (Word 0x28) 275
6.2.28 PCIe Control 3 (Word 0x29) 276
6.2.29 End of Read-Only (RO) Area (Word 0x2C) 276
6.2.30 Start of RO Area (Word 0x2D) 276
6.2.31 Watchdog Configuration (Word 0x2E) 277
6.2.32 VPD Pointer (Word 0x2F) 277
6.3 CSR Auto load Modules 277
6.3.1 Software Reset CSR Auto Configuration Pointer (LAN Base Address + Offset 0x17) 277
6.3.2 Software Reset CSR Configuration Section Length - Offset 0x0 278
6.3.2.1 Block CRC8 (Offset 0x1) 278
6.3.2.2 CSR Address - (Offset 3*n - 1; [n = 1... Section Length]) 278
6.3.2.3 CSR Data LSB - (Offset 3*n; [n = 1... Section Length]) 279
6.3.2.4 CSR Data MSB - (Offset 3*n + 1; [n = 1... Section Length]) 279
6.3.3 PCIe Reset CSR Auto Configuration Pointer (LAN Base Address + Offset 0x23) 279
6.3.3.1 PCIe Reset CSR Configuration Section Length - Offset 0x0 279
6.3.3.2 Block CRC8 (Offset 0x1) 280
6.3.3.3 CSR Address - (Offset 3*n - 1; [n = 1... Section Length]) 280
6.3.3.4 CSR Data LSB - (Offset 3*n; [n = 1... Section Length]) 280
6.3.3.5 CSR Data MSB - (Offset 3*n + 1; [n = 1... Section Length]) 280
6.3.4 CSR Auto Configuration Power-Up Pointer (LAN Base Address + Offset 0x27) 280

6.3.4.1 CSR Configuration Power-Up Section Length - Offset 0x0 281
6.3.4.2 Block CRC8 (Offset 0x1) 281
6.3.4.3 CSR Address - (Offset 3*n - 1; [n = 1... Section Length]) 281
6.3.4.4 CSR Data LSB - (Offset 3*n; [n = 1... Section Length]) 281
6.3.4.5 CSR Data MSB - (Offset 3*n + 1; [n = 1... Section Length]) 281
6.3.5 PCIe PHY Auto Configuration Pointer (Word 0x10) 282
6.3.5.1 PCIe PHY Configuration Section Length - Offset 0x0 282
6.3.5.2 Block CRC8 (Offset 0x1) 282
6.3.5.3 Register Address - (Offset 2*n; [n = 1... Number of Registers to be Written]) 282
6.3.5.4 Register Data - (Offset 2*n + 1; [n = 1... Number of Registers to be Written]) 282
6.3.5.5 Setting Default PCIe Link Width and Link Speed 283
6.3.5.5.1 PCIe Link Configuration Register Address - Offset 0x2 283
6.3.5.5.2 PCIe Link Configuration Register Data - Offset 0x3 283
6.3.5.5.3 PCIe Link Power Down Register Address - Offset 0x4 283
6.3.5.5.4 PCIe Link Power Down Register Data - Offset 0x5 283
6.3.6 Management Pass Through LAN Configuration Pointer (LAN Base Address + Offset 0x11) ... 284
6.3.6.1 PT LAN Configuration Structure 284
6.3.6.2 Management PT LAN Configuration Structure Section Length - Offset 0x0 284
6.3.6.3 Block CRC8 (Offset 0x1) 285
6.3.6.4 CSR Address - (Offset 2*n; [n = 1... Section Length]) 285
6.3.6.5 CSR Data LSB - (Offset 0x1 + 2*n; [n = 1... Section Length]) 285
6.3.6.6 CSR Data MSB - (Offset 0x2 + 2*n; [n = 1... Section Length]) 285
6.3.6.7 Manageability Filters 285
6.3.7 Common Firmware Parameters – (Global MNG Offset 0x3) 286
6.3.7.1 Section Header — Offset 0x0 286
6.3.7.2 Common Firmware Parameters 1 - Offset 0x1 286
6.3.7.3 Common Firmware Parameters 2 – Offset 0x2 287
6.3.8 Pass Through LAN 0...3 Configuration Modules (Global MNG Offsets 0x05, 0x08, 0x0D, 0X0E) 287
6.3.8.1 Section Header — Offset 0x0 288
6.3.8.2 LAN 0/1/2/3 IPv4 Address 0 LSB; (MIPAF12 LSB) — Offset 0x01 288
6.3.8.3 LAN 0/1/2/3 IPv4 Address 0 MSB; (MIPAF12 MSB) — Offset 0x02 288
6.3.8.4 LAN 0/1/2/3 IPv4 Address 1; (MIPAF13) — Offset 0x03-0x04 288
6.3.8.5 LAN 0/1/2/3 IPv4 Address 2; (MIPAF14) — Offset 0x05-0x06 288
6.3.8.6 LAN 0/1/2/3 IPv4 Address 3; (MIPAF15) — Offset 0x07-0x08 289
6.3.8.7 LAN 0/1/2/3 Ethernet MAC Address 0 LSB (MMAL0) — Offset 0x09 289
6.3.8.8 LAN 0/1/2/3 Ethernet MAC Address 0 MID; (MMAL0) — Offset 0x0A 289
6.3.8.9 LAN 0/1/2/3 Ethernet MAC Address 0 MSB; (MMAH0) — Offset 0x0B 289
6.3.8.10 LAN 0/1/2/3 Ethernet MAC Address 1; (MMAL/H1) — Offset 0x0C-0x0E 289
6.3.8.11 LAN 0/1/2/3 Ethernet MAC Address 2; (MMAL/H2) — Offset 0x0F-0x11 290
6.3.8.12 LAN 0/1/2/3 Ethernet MAC Address 3; (MMAL/H3) — Offset 0x12-0x14 290
6.3.8.13 LAN 0/1/2/3 UDP/TCP Flexible Filter Ports 0 — 7; (MFUTP Registers) — Offset 0x15 - 0x1C 290
6.3.8.14 Reserved EEPROM Words - Offset 0x1D - 0x24 290
6.3.8.15 LAN 0/1/2/3 VLAN Filter 0 - 7; (MAVTV Registers) — Offset 0x25 — 0x2C 290
6.3.8.16 Reserved EEPROM Words - Offset 0x2D to 0x2E 291
6.3.8.17 LAN 0/1/2/3 MANC value LSB; (LMANC LSB) — Offset 0x2F 291
6.3.8.18 LAN 0/1/2/3 MANC Value MSB; (LMANC MSB) — Offset 0x30 291
6.3.8.19 LAN 0/1/2/3 Receive Enable 1; (LRXEN1) — Offset 0x31 292
6.3.8.20 LAN 0/1/2/3 Receive Enable 2; (LRXEN2) — Offset 0x32 292
6.3.8.21 LAN 0/1/2/3 MNGONLY LSB; (LMNGONLY LSB) - Offset 0x33 293
6.3.8.22 LAN 0/1/2/3 MNGONLY MSB; (LMNGONLY MSB) - Offset 0x34 293
6.3.8.23 Manageability Decision Filters 0 LSB; (MDEF0 LSB) - Offset 0x35 293
6.3.8.24 Manageability Decision Filters 0 MSB; (MDEF0 MSB) - Offset 0x36 293
6.3.8.25 Manageability Decision Filters Extend 0 LSB; (MDEF_EXT0 LSB) - Offset 0x37 293
6.3.8.26 Manageability Decision Filters Extend 0 MSB; (MDEF_EXT0 MSB) - Offset 0x38 294
6.3.8.27 Manageability Decision Filters; (MDEF1-6 and MDEF_EXT1-6) - Offset 0x39-0x50 294
6.3.8.28 Manageability Ethertype Filter 0 LSB; (METF0 LSB) - Offset 0x51 294
6.3.8.29 Manageability Ethertype Filter 0 MSB; (METF0 MSB) - Offset 0x52 294
6.3.8.30 Manageability Ethertype Filter 1...3; (METF1...3) - Offset 0x53...0x58 294
6.3.8.31 ARP Response IPv4 Address 0 LSB; (ARP LSB) - Offset 0x59 295
6.3.8.32 ARP Response IPv4 Address 0 MSB; (ARP MSB) - Offset 0x5A 295
6.3.8.33 LAN0/1/2/3 IPv6 Address 0 LSB; (MIPAF0 LSB) - Offset 0x5B 295
6.3.8.34 LAN0/1/2/3 IPv6 Address 0 MSB; (MIPAF0 MSB) - Offset 0x5C 295
6.3.8.35 LAN0/1/2/3 IPv6 Address 0 LSB; (MIPAF1 LSB)- Offset 0x5D 295
6.3.8.36 LAN0/1/2/3 IPv6 Address 0 MSB; (MIPAF1 MSB) - Offset 0x5E 296

6.3.8.37 LAN0/1/2/3 IPv6 Address 0 LSB; (MIPAF2 LSB) - Offset 0x5F ... 296
6.3.8.38 LAN0/1/2/3 IPv6 Address 0 MSB; (MIPAF2 MSB) - Offset 0x60 ... 296
6.3.8.39 LAN0/1/2/3 IPv6 Address 0 LSB; (MIPAF3 LSB) - Offset 0x61 ... 296
6.3.8.40 LAN0/1/2/3 IPv6 Address 0 MSB; (MIPAF3 MSB) - Offset 0x62 ... 297
6.3.8.41 LAN0/1/2/3 IPv6 Address 1; MIPAF (Offset 0x63:0x6A) ... 297
6.3.8.42 LAN0/1/2/3 IPv6 Address 2; MIPAF (Offset 0x6B:0x72...)297
6.3.9 Sideband Configuration Module (Global MNG Offset 0x06) ... 297
6.3.9.1 Section Header — Offset 0x0 ... 297
6.3.9.2 SMBus Maximum Fragment Size — Offset 0x01 ... 298
6.3.9.3 SMBus Notification Timeout and Flags — Offset 0x02 ... 298
6.3.9.4 SMBus Slave Addresses 1 — Offset 0x03 ... 298
6.3.9.5 Reserved — Offset 0x04 ... 298
6.3.9.6 Reserved — Offset 0x05 ... 299
6.3.9.7 NC-SI Configuration - Offset 0x06 ... 299
6.3.9.8 NC-SI Configuration - Offset 0x07 ... 299
6.3.9.9 NC-SI Hardware Arbitration Configuration - Offset 0x08 ... 300
6.3.9.10 MCTP UUID - Time Low LSB (0ffset 0x09) ... 300
6.3.9.11 MCTP UUID - Time Low MSB (0ffset 0x0A) ... 300
6.3.9.12 MCTP UUID - Time MID (0ffset 0x0B) ... 300
6.3.9.13 MCTP UUID - Time High and Version (0ffset 0x0C) ... 300
6.3.9.14 MCTP UUID - Clock Seq (0ffset 0x0D) ... 300
6.3.9.15 SMBus Slave Addresses 2 - Offset 0x0E ... 301
6.3.9.16 Alternative IANA - 0ffset 0x0F ... 301
6.3.9.17 NC-SI over MCTP Configuration - 0ffset 0x10 ... 301
6.3.10 Flexible TCO Filter Configuration Module (Global MNG Offset 0x07) ... 301
6.3.10.1 Section Header — Offset 0x0 ... 301
6.3.10.2 Flexible Filter Length and Control — Offset 0x01 ... 301
6.3.10.3 Flexible Filter Enable Mask — Offset 0x02 – 0x09 ... 302
6.3.10.4 Flexible Filter Data — Offset 0x0A – Block Length ... 302
6.3.11 NC-SI Configuration Module (Global MNG Offset 0x0A) ... 302
6.3.11.1 Section Header — Offset 0x0 ... 303
6.3.11.2 Rx Mode Control1 (RR_CTRL[15:0]) - Offset 0x1 ... 303
6.3.11.3 Rx Mode Control2 (RR_CTRL[31:16]) - Offset 0x2 ... 303
6.3.11.4 Tx Mode Control1 (RT_CTRL[15:0]) - Offset 0x3 ... 303
6.3.11.5 Tx Mode Control2 (RT_CTRL[31:16]) - Offset 0x4 ... 303
6.3.11.6 MAC Tx Control Reg1 (TxCntrlReg1 (15:0]) - Offset 0x5 ... 303
6.3.11.7 MAC Tx Control Reg2 (TxCntrlReg1 (31:16]) - Offset 0x6 ... 303
6.3.11.8 MAC RX Buffer Size - Offset 0x7 ... 304
6.3.11.9 NCSI Flow Control XOFF - Offset 0x8 ... 304
6.3.11.10 NCSI Flow Control XON - Offset 0x9 ... 304
6.3.12 Traffic Type Parameters – (Global MNG Offset 0xB) ... 304
6.3.12.1 Section Header — Offset 0x0 ... 304
6.3.12.2 Traffic Type Data - Offset 0x1 ... 305
6.3.13 PHY Configuration Pointer – (Global MNG Offset 0xF) ... 306
6.3.13.1 PHY Configuration Structure ... 306
6.3.13.1.1 PHY Configuration Section Length - Offset 0x0 ... 306
6.3.13.1.2 Block CRC8 (Offset 0x1) ... 306
6.3.13.1.3 PHY Number and PHY Register Address - (Offset 2*n; [n = 1... Section Length]) ... 307
6.3.13.1.4 PHY data (Offset 2*n + 1; [n = 1... Section Length]) ... 307
6.3.14 Thermal Sensor Configuration Pointer – (Global MNG Offset 0x10) ... 307
6.3.14.1 Thermal Sensor Configuration Structure ... 307
6.3.14.1.1 Thermal Sensor Configuration Section Length - Offset 0x0 ... 308
6.3.14.1.2 Block CRC8 (Offset 0x1) ... 308
6.3.14.1.3 Thermal Sensor Register Address - (Offset 2*n; [n = 1... Section Length]) ... 308
6.3.14.1.4 Thermal Sensor data (Offset 2*n + 1; [n = 1... Section Length]) ... 308
6.4 Software Accessed Words ... 308
6.4.1 Compatibility (Word 0x03) ... 309
6.4.2 Port Identification LED blinking (Word 0x04) ... 309
6.4.3 EEPROM Image Revision (Word 0x05) ... 310
6.4.4 OEM Specific (Word 0x06, 0x07) ... 310
6.4.5 PBA Number/Pointer (Word 0x08, 0x09) ... 310
6.4.6 PXE Configuration Words (Word 0x30:3B) ... 311
6.4.6.1 Setup Options PCI Function 0 (Word 0x30) ... 311
6.4.6.2 Configuration Customization Options PCI Function 0 (Word 0x31) ... 312
6.4.6.3 PXE Version (Word 0x32) ... 313
6.4.6.4 Flash (Option ROM) Capabilities (Word 0x33) ... 314
6.4.6.5 Setup Options PCI Function 1 (Word 0x34) ... 314
6.4.6.6 Configuration Customization Options PCI Function 1 (Word 0x35) ... 314

6.4.6.7 Setup Options PCI Function 2 (Word 0x38) 314
6.4.6.8 Configuration Customization Options PCI Function 2 (Word 0x39) 314
6.4.6.9 Setup Options PCI Function 3 (Word 0x3A) 314
6.4.6.10 Configuration Customization Options PCI Function 3 (Word 0x3B) 315
6.4.6.11 PXE VLAN Configuration Pointer (0x003C) 315
6.4.6.12 PXE VLAN Configuration Section Summary Table 315
6.4.6.13 VLAN Block Signature - 0x0000 315
6.4.6.14 Version and Size - 0x0001 316
6.4.6.15 Port 0 VLAN Tag - 0x0002 316
6.4.6.16 Port 1 VLAN Tag - 0x0003 316
6.4.6.17 Port 2 VLAN Tag - 0x0004 316
6.4.6.18 Port 3 VLAN Tag - 0x0005 316
6.4.7 iSCSI Boot Words 317
6.4.7.1 iSCSI Option ROM Version (Word 0x36) 317
6.4.7.2 iSCSI boot Configuration Pointer (Word 0x3D) 317
6.4.7.3 iSCSI Module Structure 317
6.4.8 Alternate MAC address pointer (Word 0x37) 319
6.4.9 Reserved/3rd Party External Thermal Sensor – (Word 0x3E) 319
6.4.9.1 3rd Party External Thermal Sensor Configuration NVM Block 319
6.4.9.1.1 External Thermal Sensor Configuration Block 319
6.4.9.1.2 ETS General Configuration Word – Offset 0x0 319
6.4.9.1.3 Sensor Data – Offset 0x1 – 0xN (N is max of 4) 320
6.4.10 Checksum Word (Offset 0x3F) 320
6.4.11 Image Unique ID (Word 0x42, 0x43) 321

7 Inline Functions 323

7.1 Receive Functionality 323
7.1.1 L2 Packet Filtering 324
7.1.1.1 MAC Address Filtering 325
7.1.1.1.1 Unicast Filter 326
7.1.1.1.2 Multicast Filter (Inexact) 327
7.1.1.2 VLAN Filtering 327
7.1.1.3 Manageability Filtering 328
7.1.1.4 Size Filtering 329
7.1.2 Receive Queues Assignment 330
7.1.2.1 Queuing in a Non-Virtualized Environment 331
7.1.2.2 Receive Queuing in a Virtualized Environment 331
7.1.2.3 Queue Configuration Registers 332
7.1.2.4 L2 Ether-Type Filters 333
7.1.2.5 2-Tuple Filters 333
7.1.2.6 Flex Filters 334
7.1.2.7 SYN Packet Filters 335
7.1.2.8 Receive-Side Scaling (RSS) 336
7.1.2.8.1 RSS Hash Function 337
7.1.2.8.2 Indirection Table 340
7.1.2.8.3 RSS Verification Suite 340
7.1.2.8.4 Association Through MAC Address 341
7.1.3 Receive Data Storage 341
7.1.3.1 Host Buffers 341
7.1.3.2 On-Chip Receive Buffers 342
7.1.3.3 On-Chip Descriptor Buffers 342
7.1.4 Receive Descriptors 342
7.1.4.1 Legacy Receive Descriptor Format 342
7.1.4.2 Advanced Receive Descriptors 345
7.1.4.2.1 Advanced Receive Descriptors (RDESC) - Read Format 345
7.1.4.2.2 Advanced Receive Descriptors (RDESC) - Writeback Format 346
7.1.4.3 Receive Descriptor Fetching 351
7.1.4.4 Receive Descriptor Write-Back 351
7.1.4.5 Receive Descriptor Ring Structure 352
7.1.4.5.1 Low Receive Descriptors Threshold 353
7.1.5 Header Splitting and Replication 354
7.1.5.1 Purpose 354
7.1.5.2 Description 354
7.1.6 Receive Packet Timestamp in Buffer 356
7.1.7 Receive Packet Checksum and SCTP CRC Off Loading 357
7.1.7.1 Filters Details 358
7.1.7.1.1 MAC Address Filter 358
7.1.7.1.2 SNAP/VLAN Filter 358
7.1.7.1.3 IPv4 Filter 359
7.1.7.1.4 IPv6 Filter 359
7.1.7.1.5 IPv6 Extension Headers 359
7.1.7.1.6 UDP/TCP Filter 360
7.1.7.2 Packet Checksum 360
7.1.7.3 Receive UDP Fragmentation Checksum 361
7.1.7.4 SCTP Offload 361

7.2 Transmit Functionality 362
7.2.1 Packet Transmission 362
7.2.1.1 Transmit Data Storage 362
7.2.1.2 On-Chip Transmit Buffers 362
7.2.1.3 On-Chip Descriptor Buffers 363
7.2.1.4 Transmit Contexts 363
7.2.2 Transmit Descriptors 364
7.2.2.1 Legacy Transmit Descriptor Format 364
7.2.2.1.1 Buffer Address (64) 365
7.2.2.1.2 Length 365
7.2.2.1.3 Checksum Offset and Start - CSO and CSS 365
7.2.2.1.4 Command Byte - CMD 366
7.2.2.1.5 Status – STA 367
7.2.2.1.6 DD (Bit 0) - Descriptor Done Status 367
7.2.2.1.7 VLAN 367
7.2.2.2 Advanced Transmit Context Descriptor 368
7.2.2.2.1 IPLEN (9) 368
7.2.2.2.2 MACLEN (7) 368
7.2.2.2.3 TUCMD (11) 369
7.2.2.2.4 DTYP(4) 369
7.2.2.2.5 DEXT(1) 369
7.2.2.2.6 IDX (3) 369
7.2.2.2.7 L4LEN (8) 369
7.2.2.2.8 MSS (16) 369
7.2.2.3 Advanced Transmit Data Descriptor 371
7.2.2.3.1 Address (64) 371
7.2.2.3.2 DTALEN (16) 371
7.2.2.3.3 MAC (2) 371
7.2.2.3.4 DTYP (4) 371
7.2.2.3.5 DCMD (8) 372
7.2.2.3.6 STA (4) 372
7.2.2.3.7 IDX (3) 373
7.2.2.3.8 POPTS (6) 373
7.2.2.3.9 PAYLEN (18) 373
7.2.2.4 Transmit Descriptor Ring Structure 374
7.2.2.5 Transmit Descriptor Fetching 375
7.2.2.6 Transmit Descriptor Write-Back 376
7.2.3 Transmit Completions Head Write Back 377
7.2.3.1 Description 377
7.2.4 TCP/UDP Segmentation 378
7.2.4.1 Assumptions 378
7.2.4.2 Transmission Process 378
7.2.4.2.1 TCP Segmentation Data Fetch Control 379
7.2.4.2.2 TCP Segmentation Write-Back Modes 379
7.2.4.3 TCP Segmentation Performance 380
7.2.4.4 Packet Format 380
7.2.4.5 TCP/UDP Segmentation Indication 381
7.2.4.6 Transmit Checksum Offloading with TCP/UDP Segmentation 382
7.2.4.7 TCP/UDP/IP Headers Update 383
7.2.4.7.1 TCP/UDP/IP Headers for the First Frames 384
7.2.4.7.2 TCP/UDP/IP Headers for the Subsequent Frames 384
7.2.4.7.3 TCP/UDP/IP Headers for the Last Frame 385
7.2.4.8 Data Flow 386
7.2.5 Checksum Offloading in Non-Segmentation Mode 386
7.2.5.1 IP Checksum 387
7.2.5.2 TCP/UDP Checksum 387
7.2.5.3 SCTP CRC Offloading 388
7.2.5.4 Checksum Supported Per Packet Types 388
7.2.6 Multiple Transmit Queues 389
7.3 Interrupts 390
7.3.1 Interrupt Modes 390
7.3.1.1 MSI-X and Vectors 390
7.3.1.1.1 Usage of Spare MSI-X Vectors by Physical Function 390
7.3.2 Mapping of Interrupt Causes 390
7.3.2.1 Legacy and MSI Interrupt Modes 391
7.3.2.2 MSI-X Mode - Non-IOV Mode 392
7.3.2.3 MSI-X Interrupts in IOV Mode 393
7.3.3 Legacy Interrupt Registers 394
7.3.3.1 Interrupt Cause Register (ICR) 395
7.3.3.1.1 Legacy Mode 395
7.3.3.1.2 Advanced Mode 395
7.3.3.2 Interrupt Cause Set Register (ICS) 395
7.3.3.3 Interrupt Mask Set/Read Register (IMS) 395
7.3.3.4 Interrupt Mask Clear Register (IMC) 395
7.3.3.5 Interrupt Acknowledge Auto-mask register (IAM) 396

7.3.3.6 Extended Interrupt Cause Registers (EICR) 396
7.3.3.6.1 MSI/INT-A Mode (GPIE.Multiple_MSIX = 0) 396
7.3.3.6.2 MSI-X Mode (GPIE.Multiple_MSIX = 1) 396
7.3.3.7 Extended Interrupt Cause Set Register (EICS) 396
7.3.3.8 Extended Interrupt Mask Set and Read Register (EIMS) & Extended Interrupt Mask Clear Register (EIMC) 396
7.3.3.9 Extended Interrupt Auto Clear Enable Register (EIAC) 397
7.3.3.10 Extended Interrupt Auto Mask Enable Register (EIAM) 397
7.3.3.11 GPIE Register 397
7.3.4 Clearing Interrupt Causes 398
7.3.4.1 Auto-Clear 398
7.3.4.2 Write to Clear 399
7.3.4.3 Read to Clear 399
7.3.5 Interrupt Moderation 399
7.3.6 Rate Controlled Low Latency Interrupts (LLI) 401
7.3.6.1 Rate Control Mechanism 402
7.3.7 TCP Timer Interrupt 402
7.3.7.1 Introduction 402
7.3.7.2 Description 403
7.3.8 Setting of Interrupt Registers 403
7.4 802.1q VLAN Support 403
7.4.1 802.1q VLAN Packet Format 404
7.4.2 802.1q Tagged Frames 404
7.4.3 Transmitting and Receiving 802.1q Packets 405
7.4.3.1 Adding 802.1q Tags on Transmits 405
7.4.3.2 Stripping 802.1q Tags on Receives 405
7.4.4 802.1q VLAN Packet Filtering 405
7.4.5 Double VLAN Support 406
7.4.5.1 Transmit Behavior With External VLAN 407
7.4.5.2 Receive Behavior With External VLAN 407
7.5 Configurable LED Outputs 408
7.5.1 MODE Encoding for LED Outputs 408
7.6 Memory Error Correction and Detection 409
7.6.1 Software Recovery From Parity Error Event 410
7.6.1.1 Recovery from PCIe Parity Error Event 410
7.6.1.2 Recovery from DMA Parity Error Event 410
7.6.1.3 Recovery from LAN Port Parity Error Event 410
7.7 CPU affinity Features 411
7.7.1 Direct Cache Access (DCA) 411
7.7.1.1 DCA Description 411
7.7.1.2 Details of Implementation 412
7.7.1.2.1 PCIe Message Format for DCA 412
7.7.2 TLP Process Hints (TPH) 413
7.7.2.1 Steering Tag and Processing Hint Programming 414
7.8 Virtualization 414
7.8.1 Overview 414
7.8.1.1 Direct Assignment Model 415
7.8.1.1.1 Rationale 415
7.8.1.2 Virtualized System Overview 416
7.8.1.3 VMDq Supported Features 419
7.8.2 PCI Sig IOV Support 419
7.8.2.1 IOV Concepts 419
7.8.2.2 Configuration Space Replication 420
7.8.2.2.1 Legacy PCI Config Space 420
7.8.2.2.2 Memory BARs Assignment 420
7.8.2.2.3 Capability Structures 421
7.8.2.2.4 Extended Capability Structures 422
7.8.2.2.5 Serial ID 422
7.8.2.2.6 Error Reporting Capabilities (Advanced & Legacy) 422
7.8.2.3 Function Level Reset (FLR) Capability 422
7.8.2.4 Error Reporting 422
7.8.2.5 ARI & IOV Capability Structures 424
7.8.2.6 Requester ID Allocation 424
7.8.2.6.1 Bus-Device-Function Layout 424
7.8.2.7 Hardware Resources Assignment 425
7.8.2.7.1 Physical Function Resources 425
7.8.2.7.2 Assignment of Queues to VF 426
7.8.2.7.3 Assignment of MSI-X Vectors to VF 426
7.8.2.7.4 VF Resource Summary 426
7.8.2.8 CSR Organization 426
7.8.2.9 IOV Control 427
7.8.2.9.1 VF to PF Mailbox 427
7.8.2.10 Interrupt Handling 429
7.8.2.10.1 Low latency Interrupts 430
7.8.2.10.2 MSI-X 430

7.8.2.10.3 MSI 430
7.8.2.10.4 Legacy Interrupt (INT-x) 430
7.8.2.11 DMA 430
7.8.2.11.1 Requester ID 430
7.8.2.11.2 Sharing DMA Resources 430
7.8.2.11.3 DCA and TPH 430
7.8.2.12 Timers and Watchdog 431
7.8.2.12.1 TCP Timer 431
7.8.2.12.2 IEEE 1588 431
7.8.2.12.3 Watchdog. 431
7.8.2.12.4 Free Running Timer 431
7.8.2.13 Power Management and Wake-up 431
7.8.2.14 Link Control 431
7.8.2.14.1 Special Filtering Options 431
7.8.2.15 IOV Test Mode 432
7.8.2.15.1 Allocation of memory space for IOV functions 432
7.8.3 Packet Switching (VMDq) Model 432
7.8.3.1 VMDq Assumptions 432
7.8.3.2 VM/VF Selection 433
7.8.3.2.1 Filtering Capabilities 433
7.8.3.3 L2 Filtering 433
7.8.3.4 VMDq Receive Packets Switching 434
7.8.3.4.1 VMDq Replication Mode Enabled 434
7.8.3.4.2 VMDq Replication Mode Disabled 435
7.8.3.5 TX Packets Switching 436
7.8.3.5.1 Replication Mode Enabled 438
7.8.3.5.2 Replication Mode Disabled 439
7.8.3.6 Mirroring Support 440
7.8.3.7 Offloads 441
7.8.3.7.1 Split Header Offload 441
7.8.3.7.2 Local VM to VM Traffic Offload 441
7.8.3.7.3 Small Packets Padding 442
7.8.3.8 Security Features 442
7.8.3.8.1 Inbound Security 442
7.8.3.8.2 Outbound Security 442
7.8.3.8.3 Interrupt on Misbehavior of VM (Malicious Driver Detection) 443
7.8.3.8.4 Storm Control 445
7.8.3.9 External Switch Loopback Support 446
7.8.3.10 Switch Control 447
7.8.4 Virtualization of the Hardware 447
7.8.4.1 Per Pool Statistics 447
7.8.4.1.1 Byte Count Statistics 448
7.9 Time SYNC (IEEE1588 and IEEE 802.1AS) 449
7.9.1 Overview 449
7.9.2 Flow and Hardware/Software Responsibilities 450
7.9.2.1 TimeSync Indications in Receive and Transmit Packet Descriptors 451
7.9.3 Hardware Time Sync Elements 452
7.9.3.1 System Time Structure and Mode of Operation 452
7.9.3.2 Time Stamp Mechanism 452
7.9.3.3 Time Adjustment Mode of Operation 454
7.9.4 Time Sync Related Auxiliary Elements 454
7.9.4.1 Target Time 455
7.9.4.1.1 SYSTIM Synchronized Level Change Generation on SDP Pins 455
7.9.4.1.2 SYSTIM Synchronized Pulse Generation on SDP Pins 455
7.9.4.1.3 Start of Clock Generation on SDP Pins Synchronized to SYSTIM 456
7.9.4.2 Configurable Frequency Clock 456
7.9.4.3 Time Stamp Events 457
7.9.5 Time SYNC Interrupts 457
7.9.6 PTP Packet Structure 457
7.10 Statistic Counters 459
7.10.1 IEEE 802.3 clause 30 management 460
7.10.2 OID_GEN_STATISTICS 461
7.10.3 RMON 462
7.10.4 Linux* net_device_stats 462
7.10.5 Statistics Hierarchy. 463

8 Programming Interface 465

8.1 Introduction 465
8.1.1 Memory, I/O Address and Configuration Decoding 465
8.1.1.1 Memory-Mapped Access to Internal Registers and Memories 465
8.1.1.2 Memory-Mapped Access to Flash 466
8.1.1.3 Memory-Mapped Access to MSI-X Tables 466
8.1.1.4 Memory-Mapped Access to Expansion ROM 466
8.1.1.5 I/O-Mapped Access to Internal Registers and Memories 466
8.1.1.5.1 IOADDR (I/O offset 0x00) 466

8.1.1.5.2 IODATA (I/O offset 0x04) 467
8.1.1.5.3 Undefined I/O Offsets 467
8.1.1.6 Configuration Access to Internal Registers and Memories 467
8.1.2 Register Conventions 469
8.1.2.1 Registers Byte Ordering 470
8.1.3 Register Summary 471
8.1.3.1 Alias Addresses 482
8.1.4 MSI-X BAR Register Summary 483
8.2 General Register Descriptions 483
8.2.1 Device Control Register - CTRL (0x00000; R/W) 483
8.2.2 Device Status Register - STATUS (0x0008; RO) 486
8.2.3 Extended Device Control Register - CTRL_EXT (0x0018; R/W) 488
8.2.4 MDI Control Register - MDIC (0x0020; R/W) 490
8.2.5 MDC/MDIO Configuration Register – MDICNFG (0x0E04; R/W) 491
8.2.6 SERDES Control 0 - P1GCTRL0 (0x0E08; RW) 492
8.2.7 Copper/Fiber Switch Control - CONNSW (0x0034; R/W) 493
8.2.8 VLAN Ether Type - VET (0x0038; R/W) 494
8.2.9 LED Control - LEDCTL (0x0E00; RW) 494
8.3 Internal Packet Buffer Size Registers 495
8.3.1 Internal Receive Packet Buffer Size - IRPBS (0x2404; RO) 496
8.3.2 Internal Transmit Packet Buffer Size - ITPBS (0x3404; RO) 497
8.4 EEPROM/Flash Register Descriptions 497
8.4.1 EEPROM/Flash Control Register - EEC (0x0010; R/W) 497
8.4.2 EEPROM Read Register - EERD (0x0014; RW) 499
8.4.3 Flash Access - FLA (0x001C; R/W) 500
8.4.4 Flash Opcode - FLASHOP (0x103C; R/W) 501
8.4.5 EEPROM Auto Read Bus Control - EEARBC (0x1024; R/W) 501
8.4.6 Management-EEPROM CSR I/F 502
8.4.6.1 Management EEPROM Control Register - EEMNGCTL (0x1010; RW) 503
8.4.6.2 Management EEPROM Read/Write Data - EEMNGDATA (0x1014; RW) 504
8.5 Flow Control Register Descriptions 504
8.5.1 Flow Control Address Low - FCAL (0x0028; RO) 504
8.5.2 Flow Control Address High - FCAH (0x002C; RO) 504
8.5.3 Flow Control Type - FCT (0x0030; R/W) 505
8.5.4 Flow Control Transmit Timer Value - FCTTV (0x0170; R/W) 505
8.5.5 Flow Control Receive Threshold Low - FCRTL0 (0x2160; R/W) 505
8.5.6 Flow Control Receive Threshold High - FCRTH0 (0x2168; R/W) 506
8.5.7 Flow Control Refresh Threshold Value - FCRTV (0x2460; R/W) 507
8.5.8 Flow Control Status - FCSTS0 (0x2464; RO) 507
8.6 PCIe Register Descriptions 507
8.6.1 PCIe Control - GCR (0x5B00; RW) 507
8.6.2 PCIe Statistics Control #1 - GSCL_1 (0x5B10; RW) 508
8.6.3 PCIe Statistics Control #2 - GSCL_2 (0x5B14; RW) 508
8.6.4 PCIe Statistic Control Register #5...#8 - GSCL_5_8 (0x5B90 + 4*n[n=0...3]; RW) 509
8.6.5 PCIe Counter #0 - GSCN_0 (0x5B20; RC) 510
8.6.6 PCIe Counter #1 - GSCN_1 (0x5B24; RC) 510
8.6.7 PCIe Counter #2 - GSCN_2 (0x5B28; RC) 510
8.6.8 PCIe Counter #3 - GSCN_3 (0x5B2C; RC) 511
8.6.9 Function Active and Power State to MNG - FACTPS (0x5B30; RO) 511
8.6.10 Mirrored Revision ID - MREVID (0x5B64; R/W) 512
8.6.11 PCIe Control Extended Register - GCR_EXT (0x5B6C; RW) 513
8.6.12 PCIe BAR Control - BARCTRL (0x5BFC; R/W) Target 513
8.7 Semaphore Registers 514
8.7.1 Software Semaphore - SWSM (0x5B50; R/W) 515
8.7.2 Firmware Semaphore - FWSM (0x5B54; R/WS) 515
8.7.3 Software–Firmware Synchronization - SW_FW_SYNC (0x5B5C; RWS) 517
8.7.4 Software Mailbox Write - SWMBWR (0x5B04; R/W) 518
8.7.5 Software Mailbox 0 - SWMB0 (0x5B08; RO) 518
8.7.6 Software Mailbox 1 - SWMB1 (0x5B0C; RO) 519
8.7.7 Software Mailbox 2 - SWMB2 (0x5B18; RO) 519
8.7.8 Software Mailbox 3 - SWMB3 (0x5B1C; RO) 519
8.8 Interrupt Register Descriptions 519
8.8.1 PCIe Interrupt Cause - PICAUSE (0x5B88; RW1/C) 519
8.8.2 PCIe Interrupt Enable - PIENA (0x5B8C; R/W) 520
8.8.3 Extended Interrupt Cause - EICR (0x1580; RC/W1C) 520
8.8.4 Extended Interrupt Cause Set - EICS (0x1520; WO) 521
8.8.5 Extended Interrupt Mask Set/Read - EIMS (0x1524; RWS) 522
8.8.6 Extended Interrupt Mask Clear - EIMC (0x1528; WO) 522
8.8.7 Extended Interrupt Auto Clear - EIAC (0x152C; R/W) 523
8.8.8 Extended Interrupt Auto Mask Enable - EIAM (0x1530; R/W) 524
8.8.9 Interrupt Cause Read Register - ICR (0x1500; RC/W1C) 525
8.8.10 Interrupt Cause Set Register - ICS (0x1504; WO) 527
8.8.11 Interrupt Mask Set/Read Register - IMS (0x1508; R/W) 528
8.8.12 Interrupt Mask Clear Register - IMC (0x150C; WO) 529

8.8.13 Interrupt Acknowledge Auto Mask Register - IAM (0x1510; R/W) 530
8.8.14 Interrupt Throttle - EITR (0x1680 + 4*n [n = 0...24]; R/W) 530
8.8.15 Interrupt Vector Allocation Registers - IVAR (0x1700 + 4*n [n=0...3]; RW) 531
8.8.16 Interrupt Vector Allocation Registers - MISC IVAR_MISC (0x1740; RW) 533
8.8.17 General Purpose Interrupt Enable - GPIE (0x1514; RW) 533
8.9 MSI-X Table Register Descriptions 534
8.9.1 MSI–X Table Entry Lower Address - MSIXTADD (BAR3: 0x0000 + 0x10*n [n=0... 24]; R/W) 535
8.9.2 MSI–X Table Entry Upper Address - MSIXTUADD (BAR3: 0x0004 + 0x10*n [n=0...24]; R/W) 535
8.9.3 MSI–X Table Entry Message - MSIXTMSG (BAR3: 0x0008 + 0x10*n [n=0...24]; R/W) 535
8.9.4 MSI–X Table Entry Vector Control - MSIXTVCTRL (BAR3: 0x000C + 0x10*n [n=0...24]; R/W) 536
8.9.5 MSIXPBA Bit Description – MSIXPBA (BAR3: 0x2000; RO) 536
8.9.6 MSI-X PBA Clear – PBACL (0x5B68; R/W1C) 536
8.10 Receive Register Descriptions 536
8.10.1 Receive Control Register - RCTL (0x0100; R/W) 536
8.10.2 Split and Replication Receive Control - SRRCTL (0xC00C + 0x40*n [n=0...7]; R/W) 539
8.10.3 Packet Split Receive Type - PSRTYPE (0x5480 + 4*n [n=0...7]; R/W) 540
8.10.4 Replicated Packet Split Receive Type - RPLPSRTYPE (0x54C0; R/W) 541
8.10.5 Receive Descriptor Base Address Low - RDBAL (0xC000 + 0x40*n [n=0...7]; R/W) 542
8.10.6 Receive Descriptor Base Address High - RDBAH (0xC004 + 0x40*n [n=0...7]; R/W) 542
8.10.7 Receive Descriptor Ring Length - RDLEN (0xC008 + 0x40*n [n=0...7]; R/W) 542
8.10.8 Receive Descriptor Head - RDH (0xC010 + 0x40*n [n=0...7]; RO) 543
8.10.9 Receive Descriptor Tail - RDT (0xC018 + 0x40*n [n=0...7]; R/W) 543
8.10.10 Receive Descriptor Control - RXDCTL (0xC028 + 0x40*n [n=0...7]; R/W) 543
8.10.11 Receive Queue drop packet count - RQDPC (0xC030 + 0x40*n [n=0...7]; RC) 545
8.10.12 Receive Checksum Control - RXCSUM (0x5000; R/W) 545
8.10.13 Receive Long Packet Maximum Length - RLPML (0x5004; R/W) 547
8.10.14 Receive Filter Control Register - RFCTL (0x5008; R/W) 547
8.10.15 Multicast Table Array - MTA (0x5200 + 4*n [n=0...127]; R/W) 548
8.10.16 Receive Address Low - RAL (0x5400 + 8*n [n=0...15]; 0x54E0 + 8*n [n=0...15]; R/W) 548
8.10.17 Receive Address High - RAH (0x5404 + 8*n [n=0...15]; 0x54E4 + 8*n [n=0...15]; R/W) 549
8.10.18 VLAN Filter Table Array - VFTA (0x5600 + 4*n [n=0...127]; R/W) 550
8.10.19 Multiple Receive Queues Command Register - MRQC (0x5818; R/W) 550
8.10.20 RSS Random Key Register - RSSRK (0x5C80 + 4*n [n=0...9]; R/W) 551
8.10.21 Redirection Table - RETA (0x5C00 + 4*n [n=0...31]; R/W) 551
8.11 Filtering Register Descriptions 552
8.11.1 Immediate Interrupt RX - IMIR (0x5A80 + 4*n [n=0...7]; R/W) 552
8.11.2 Immediate Interrupt Rx Ext. - IMIREXT (0x5AA0 + 4*n [n=0...7]; R/W) 554
8.11.3 554
8.11.4 2-tuples Queue Filter - TTQF (0x59E0 + 4*n[n=0...7]; RW) 554
8.11.5 Immediate Interrupt Rx VLAN Priority - IMIRVP (0x5AC0; R/W) 555
8.11.6 SYN Packet Queue Filter - SYNQF (0x55FC; RW) 556
8.11.7 EType Queue Filter - ETQF (0x5CB0 + 4*n[n=0...7]; RW) 556
8.12 Transmit Register Descriptions 557
8.12.1 Transmit Control Register - TCTL (0x0400; R/W) 557
8.12.2 Transmit Control Extended - TCTL_EXT (0x0404; R/W) 558
8.12.3 Transmit IPG Register - TIPG (0x0410; R/W) 558
8.12.4 Retry Buffer Control – RETX_CTL (0x041C; RW) 559
8.12.5 DMA TX Control - DTXCTL (0x3590; R/W) 559
8.12.6 DMA TX TCP Flags Control Low - DTXTCPFLGL (0x359C; RW) 560
8.12.7 DMA TX TCP Flags Control High - DTXTCPFLGH (0x35A0; RW) 560
8.12.8 DMA TX Max Total Allow Size Requests - DTXMXSZRQ (0x3540; RW) 560
8.12.9 DMA TX Maximum Packet Size - DTXMXPKTSZ (0x355C; RW) 561
8.12.10 Transmit Descriptor Base Address Low - TDBAL (0xE000 + 0x40*n [n=0...7]; R/W) 561
8.12.11 Transmit Descriptor Base Address High - TDBAH (0xE004 + 0x40*n [n=0...7]; R/W) 562
8.12.12 Transmit Descriptor Ring Length - TDLEN (0xE008 + 0x40*n [n=0...7]; R/W) 562
8.12.13 Transmit Descriptor Head - TDH (0xE010 + 0x40*n [n=0...7]; RO) 562
8.12.14 Transmit Descriptor Tail - TDT (0xE018 + 0x40*n [n=0...7]; R/W) 563
8.12.15 Transmit Descriptor Control - TXDCTL (0xE028 + 0x40*n [n=0...7]; R/W) 563
8.12.16 Tx Descriptor Completion Write – Back Address Low - TDWBAL (0xE038 + 0x40*n [n=0...7]; R/W) 565
8.12.17 Tx Descriptor Completion Write–Back Address High - TDWBAH (0xE03C + 0x40*n [n=0...7];R/W) 565
8.12.18 Transmit Queue drop packet count - TQDPC (0xE030 + 0x40*n [n=0...7]; RW) 566
8.13 DCA and TPH Register Descriptions 566
8.13.1 Rx DCA Control Registers - RXCTL (0xC014 + 0x40*n [n=0...7]; R/W) 566
8.13.2 Tx DCA Control Registers - TXCTL (0xE014 + 0x40*n [n=0...7]; R/W) 568
8.13.3 DCA Requester ID Information - DCA_ID (0x5B70; RO) 569
8.13.4 DCA Control - DCA_CTRL (0x5B74; R/W) 569

8.14 Virtualization Register Descriptions 570
8.14.1 VMDq Control Register – VT_CTL (0x581C; R/W) 570
8.14.2 Physical Function Mailbox - PFMailbox (0x0C00 + 4*n[n=0...7]; RW) 570
8.14.3 Virtual Function Mailbox - VFMailbox (0x0C40 + 4*n [n=0...7]; RW) 571
8.14.4 Virtualization Mailbox Memory - VMBMEM (0x0800:0x083C + 0x40*n [n=0...7]; R/W) 571
8.14.5 Mailbox VF Interrupt Causes Register - MBVFICR (0x0C80; R/W1C) 572
8.14.6 Mailbox VF Interrupt Mask Register - MBVFIMR (0x0C84; RW) 572
8.14.7 FLR Events - VFLRE (0x0C88; R/W1C) 572
8.14.8 VF Receive Enable- VFRE (0x0C8C; RW) 572
8.14.9 VF Transmit Enable - VFTE (0x0C90; RW) 573
8.14.10 Wrong VM Behavior Register - WVBR (0x3554; RC) 573
8.14.11 Malicious Driver Free Block - MDFB (0x3558; RO) 573
8.14.12 VM Error Count Mask – VMECM (0x3510; RW) 573
8.14.13 Last VM Misbehavior Cause – LVMMC (0x3548; RC) 574
8.14.14 Queue Drop Enable Register - QDE (0x2408;RW) 574
8.14.15 TX Switch Control - TXSWC (0x5ACC; R/W) 575
8.14.16 VM VLAN Insert Register – VMVIR (0x3700 + 4 *n [n=0...7]; RW) 575
8.14.17 VM Offload Register - VMOLR (0x5AD0 + 4*n [n=0...7]; RW) 575
8.14.18 DMA VM Offload Register - DVMOLR (0xC038 + 0x40*n[n=0...7]; RW) 576
8.14.19 VLAN VM Filter - VLVF (0x5D00 + 4*n [n=0...31]; RW) 577
8.14.20 Unicast Table Array - UTA (0xA000 + 4*n [n=0...127]; R/W) 577
8.14.21 Storm Control - Control Register- SCCRL (0x5DB0; RW) 578
8.14.22 Storm Control status - SCSTS (0x5DB4;RO) 578
8.14.23 Broadcast Storm Control Threshold - BSCTRH (0x5DB8; RW) 578
8.14.24 Multicast Storm Control Threshold - MSCTRH (0x5DBC; RW) 579
8.14.25 Broadcast Storm Control Current Count - BSCCNT (0x5DC0; RO) 579
8.14.26 Multicast Storm Control Current Count - MSCCNT (0x5DC4; RO) 579
8.14.27 Storm Control Time Counter - SCTC (0x5DC8; RO) 579
8.14.28 Storm Control Basic Interval- SCBI (0x5DCC; RW) 580
8.14.29 Virtual Mirror Rule Control - VMRCTL (0x5D80 + 0x4*n [n= 0...3]; RW) 580
8.14.30 Virtual Mirror Rule VLAN - VMRVLAN (0x5D90 + 0x4*n [n= 0...3]; RW) 581
8.14.31 Virtual Mirror Rule VM - VMRVM (0x5DA0 + 0x4*n [n= 0...3]; RW) 581
8.15 Timer Registers Description 581
8.15.1 Watchdog Setup - WDSTP (0x1040; R/W) 581
8.15.2 Watchdog Software Device Status - WDSWSTS (0x1044; R/W) 582
8.15.3 Free Running Timer - FRTIMER (0x1048; RWS) 582
8.15.4 TCP Timer - TCPTIMER (0x104C; R/W) 582
8.16 Time Sync Register Descriptions 583
8.16.1 RX Time Sync Control Register - TSYNCRXCTL (0xB620;RW) 584
8.16.2 RX Timestamp Low - RXSTMPL (0xB624; RO) 584
8.16.3 RX Timestamp High - RXSTMPH (0xB628; RO) 585
8.16.4 RX Timestamp Attributes Low - RXSATRL(0xB62C; RO) 585
8.16.5 RX Timestamp Attributes High- RXSATRH (0xB630; RO) 585
8.16.6 TX Time Sync Control Register - TSYNCTXCTL (0xB614; RW) 585
8.16.7 TX Timestamp Value Low - TXSTMPL (0xB618; RO) 586
8.16.8 TX Timestamp Value High - TXSTMPH(0xB61C; RO) 586
8.16.9 System Time Register Residue - SYSTIMR (0xB6F8; RW) 586
8.16.10 System Time Register Low - SYSTIML (0xB600; RW) 586
8.16.11 System Time Register High - SYSTIMH (0xB604; RW) 587
8.16.12 Increment Attributes Register - TIMINCA (0xB608; RW) 587
8.16.13 Time Adjustment Offset Register Low - TIMADJL (0xB60C; RW) 587
8.16.14 Time Adjustment Offset Register High - TIMADJH (0xB610;RW) 587
8.16.15 TimeSync Auxiliary Control Register - TSAUXC (0xB640; RW) 588
8.16.16 Target Time Register 0 Low - TRGTTIML0 (0xB644; RW) 589
8.16.17 Target Time Register 0 High - TRGTTIMH0 (0xB648; RW) 589
8.16.18 Target Time Register 1 Low - TRGTTIML1 (0xB64C; RW) 590
8.16.19 Target Time Register 1 High - TRGTTIMH1 (0xB650; RW) 590
8.16.20 Frequency Out 0 Control Register FREQOUT0 (0xB654; RW) 590
8.16.21 Frequency Out 1 Control Register - FREQOUT1 (0xB658; RW) 591
8.16.22 Auxiliary Time Stamp 0 Register Low - AUXSTMPL0 (0xB65C; RO) 591
8.16.23 Auxiliary Time Stamp 0 Register High -AUXSTMPH0 (0xB660; RO) 591
8.16.24 Auxiliary Time Stamp 1 Register Low AUXSTMPL1 (0xB664; RO) 592
8.16.25 Auxiliary Time Stamp 1 Register High - AUXSTMPH1 (0xB668; RO) 592
8.16.26 Time Sync RX Configuration - TSYNCRXCFG (0x5F50; R/W) 592
8.16.27 Time Sync SDP Configuration Register - TSSDP (0x003C; R/W) 592
8.16.28 Time Sync Interrupt Registers 594
8.16.28.1 Time Sync Interrupt Cause Register - TSICR (0xB66C; RC/W1C) 594
8.16.28.2 Time Sync Interrupt Mask Register - TSIM (0xB674; RW) 594
8.16.28.3 Time Sync Interrupt Set Register - TSIS (0xB670; WO) 595
8.17 PCS Register Descriptions 596
8.17.1 PCS Configuration - PCS_CFG (0x4200; R/W) 596
8.17.2 PCS Link Control - PCS_LCTL (0x4208; RW) 596
8.17.3 PCS Link Status - PCS_LSTS (0x420C; RO) 598
8.17.4 AN Advertisement - PCS_ANADV (0x4218; R/W) 598
8.17.5 Link Partner Ability - PCS_LPAB (0x421C; RO) 599

8.17.6 Next Page Transmit - PCS_NPTX (0x4220; RW) 600
8.17.7 Link Partner Ability Next Page - PCS_LPABNP (0x4224; RO) 601
8.17.8 SFP I2C Command- I2CCMD (0x1028; R/W) 601
8.17.9 SFP I2C Parameters - I2CPARAMS (0x102C; R/W) 602
8.18 Statistics Register Descriptions 603
8.18.1 CRC Error Count - CRCERRS (0x4000; RC) 603
8.18.2 Alignment Error Count - ALGNERRC (0x4004; RC) 604
8.18.3 Symbol Error Count - SYMERRS (0x4008; RC) 604
8.18.4 RX Error Count - RXERRC (0x400C; RC) 604
8.18.5 Missed Packets Count - MPC (0x4010; RC) 605
8.18.6 Single Collision Count - SCC (0x4014; RC) 605
8.18.7 Excessive Collisions Count - ECOL (0x4018; RC) 605
8.18.8 Multiple Collision Count - MCC (0x401C; RC) 605
8.18.9 Late Collisions Count - LATECOL (0x4020; RC) 606
8.18.10 Collision Count - COLC (0x4028; RC) 606
8.18.11 Defer Count - DC (0x4030; RC) 606
8.18.12 Transmit with No CRS - TNCRS (0x4034; RC) 606
8.18.13 Host Transmit Discarded Packets by MAC Count - HTDPMC (0x403C; RC) 607
8.18.14 Receive Length Error Count - RLEC (0x4040; RC) 607
8.18.15 XON Received Count - XONRXC (0x4048; RC) 607
8.18.16 XON Transmitted Count - XONTXC (0x404C; RC) 607
8.18.17 XOFF Received Count - XOFFRXC (0x4050; RC) 608
8.18.18 XOFF Transmitted Count - XOFFTXC (0x4054; RC) 608
8.18.19 FC Received Unsupported Count - FCRUC (0x4058; RC) 608
8.18.20 Packets Received [64 Bytes] Count - PRC64 (0x405C; RC) 609
8.18.21 Packets Received [65—127 Bytes] Count - PRC127 (0x4060; RC) 609
8.18.22 Packets Received [128—255 Bytes] Count - PRC255 (0x4064; RC) 609
8.18.23 Packets Received [256—511 Bytes] Count - PRC511 (0x4068; RC) 610
8.18.24 Packets Received [512—1023 Bytes] Count - PRC1023 (0x406C; RC) 610
8.18.25 Packets Received [1024 to Max Bytes] Count - PRC1522 (0x4070; RC) 610
8.18.26 Good Packets Received Count - GPRC (0x4074; RC) 611
8.18.27 Broadcast Packets Received Count - BPRC (0x4078; RC) 611
8.18.28 Multicast Packets Received Count - MPRC (0x407C; RC) 611
8.18.29 Good Packets Transmitted Count - GPTC (0x4080; RC) 612
8.18.30 Good Octets Received Count - GORCL (0x4088; RC) 612
8.18.31 Good Octets Received Count - GORCH (0x408C; RC) 612
8.18.32 Good Octets Transmitted Count - GOTCL (0x4090; RC) 612
8.18.33 Good Octets Transmitted Count - GOTCH (0x4094; RC) 613
8.18.34 Receive No Buffers Count - RNBC (0x40A0; RC) 613
8.18.35 Receive Undersize Count - RUC (0x40A4; RC) 613
8.18.36 Receive Fragment Count - RFC (0x40A8; RC) 614
8.18.37 Receive Oversize Count - ROC (0x40AC; RC) 614
8.18.38 Receive Jabber Count - RJC (0x40B0; RC) 614
8.18.39 Management Packets Received Count - MNGPRC (0x40B4; RC) 615
8.18.40 Management Packets Dropped Count - MPDC (0x40B8; RC) 615
8.18.41 Management Packets Transmitted Count - MNGPTC (0x40BC; RC) 615
8.18.42 BMC2OS Packets Sent by BMC - B2OSPC (0x8FE0; RC) 615
8.18.43 BMC2OS Packets Received by host - B2OGPRC (0x4158; RC) 616
8.18.44 OS2BMC Packets Received by BMC - O2BGPTC (0x8FE4; RC) 616
8.18.45 OS2BMC Packets Transmitted by Host - O2BSPC (0x415C; RC) 616
8.18.46 Total Octets Received - TORL (0x40C0; RC) 616
8.18.47 Total Octets Received - TORH (0x40C4; RC) 617
8.18.48 Total Octets Transmitted - TOTL (0x40C8; RC) 617
8.18.49 Total Octets Transmitted - TOTH (0x40CC; RC) 617
8.18.50 Total Packets Received - TPR (0x40D0; RC) 618
8.18.51 Total Packets Transmitted - TPT (0x40D4; RC) 618
8.18.52 Packets Transmitted [64 Bytes] Count - PTC64 (0x40D8; RC) 618
8.18.53 Packets Transmitted [65—127 Bytes] Count - PTC127 (0x40DC; RC) 619
8.18.54 Packets Transmitted [128—255 Bytes] Count - PTC255 (0x40E0; RC) 619
8.18.55 Packets Transmitted [256—511 Bytes] Count - PTC511 (0x40E4; RC) 619
8.18.56 Packets Transmitted [512—1023 Bytes] Count - PTC1023 (0x40E8; RC) 620
8.18.57 Packets Transmitted [1024 Bytes or Greater] Count - PTC1522 (0x40EC; RC) 620
8.18.58 Multicast Packets Transmitted Count - MPTC (0x40F0; RC) 620
8.18.59 Broadcast Packets Transmitted Count - BPTC (0x40F4; RC) 620
8.18.60 TCP Segmentation Context Transmitted Count - TSCTC (0x40F8; RC) 621
8.18.61 Interrupt Assertion Count - IAC (0x4100; RC) 621
8.18.62 Rx Packets to Host Count - RPTHC (0x4104; RC) 621
8.18.63 EEE TX LPI Count - TLPIC (0x4148; RC) 621
8.18.64 EEE RX LPI Count - RLPIC (0x414C; RC) 622
8.18.65 Host Good Packets Transmitted Count-HGPTC (0x4118; RC) 622
8.18.66 Receive Descriptor Minimum Threshold Count-RXDMTC (0x4120; RC) 622
8.18.67 Host Good Octets Received Count - HGORCL (0x4128; RC) 622
8.18.68 Host Good Octets Received Count - HGORCH (0x412C; RC) 623
8.18.69 Host Good Octets Transmitted Count - HGOTCL (0x4130; RC) 623
8.18.70 Host Good Octets Transmitted Count - HGOTCH (0x4134; RC) 623

8.18.71 Length Error Count - LENERRS (0x4138; RC) 624
8.18.72 SerDes/SGMII/KX Code Violation Packet Count - SCVPC (0x4228; RW) 624
8.18.73 Switch Security Violation Packet Count - SSVPC (0x41A0; RC) 624
8.18.74 Switch Drop Packet Count - SDPC (0x41A4; RC/W) 624
8.18.75 Loopback Full Buffer Drop Packet Count - LPBKFBDPC (0x4150; RC/W) 625
8.18.76 Management Full Buffer Drop Packet Count - MNGFBDPC (0x4154; RC/W) 625
8.18.77 Statistical Counters Per Queue 625
8.18.77.1 Per Queue Good Packets Received Count - VFGPRC (0x10010 + n*0x100 [n=0...7]; RW) 625
8.18.77.2 Per Queue Good Packets Transmitted Count - VFGPTC (0x10014 + n*0x100 [n=0...7]; RW) 626
8.18.77.3 Per Queue Good Octets Received Count - VFGORC (0x10018 + n*0x100 [n=0...7]; RW) 626
8.18.77.4 Per Queue Good Octets Transmitted Count - VFGOTC (0x10034 + n*0x100 [n=0...7]; RW) 626
8.18.77.5 Per Queue Multicast Packets Received Count - VFMPRC (0x10038 + n*0x100 [n=0...7]; RO) 627
8.18.77.6 Good TX Octets loopback Count - VFGOTLBC (0x10050 + n*0x100 [n=0...7]; RW) 627
8.18.77.7 Good TX packets loopback Count - VFGPTLBC (0x10044+ n*0x100 [n=0...7]; RW) 628
8.18.77.8 Good RX Octets loopback Count - VFGORLBC (0x10048+ n*0x100 [n=0...7]; RW) 628
8.18.77.9 Good RX Packets loopback Count - VFGPRLBC (0x10040+ n*0x100 [n=0...7]; RW) 628
8.19 Manageability Statistics 628
8.19.1 BMC Management Receive Packets Dropped Count - BMRPDC (0x4140; RC) 629
8.19.2 BMC Management Transmit Packets Dropped Count - BMTPDC (0x8FDC; RC) 629
8.19.3 BMC Management Packets Transmitted Count - BMNGPTC (0x4144; RC) 629
8.19.4 BMC Management Packets Received Count - BMNGPRC (0x413C; RC) 629
8.19.5 BMC Total Unicast Packets Received - BUPRC (0x4400; RC) 630
8.19.6 BMC Total Multicast Packets Received - BMPRC (0x4404; RC) 630
8.19.7 BMC Total Broadcast Packets Received - BBPRC (0x4408; RC) 630
8.19.8 BMC Total Unicast Packets Transmitted - BUPTC (0x440C; RC) 630
8.19.9 BMC Total Multicast Packets Transmitted - BMPTC (0x4410; RC) 631
8.19.10 BMC Total Broadcast Packets Transmitted - BBPTC (0x4414; RC) 631
8.19.11 BMC FCS Receive Errors - BCRCERRS (0x4418; RC) 631
8.19.12 BMC Alignment Errors - BALGNERRC (0x441C; RC) 631
8.19.13 BMC Pause XON Frames Received - BXONRXC (0x4420; RC) 631
8.19.14 BMC Pause XOFF Frames Received - BXOFFRXC (0x4424; RC) 631
8.19.15 BMC Pause XON Frames Transmitted - BXONTXC (0x4428; RC) 632
8.19.16 BMC Pause XOFF Frames Transmitted - BXOFFTXC (0x442C; RC) 632
8.19.17 BMC Single Collision Transmit Frames- BSCC (0x4430; RC) 632
8.19.18 BMC Multiple Collision Transmit Frames - BMCC (0x4434; RC) 632
8.20 Wake Up Control Register Descriptions 632
8.20.1 Wakeup Control Register - WUC (0x5800; R/W) 632
8.20.2 Wakeup Filter Control Register - WUFC (0x5808; R/W) 633
8.20.3 Wakeup Status Register - WUS (0x5810; R/W1C) 635
8.20.4 Wakeup Packet Length - WUPL (0x5900; RO) 636
8.20.5 Wakeup Packet Memory - WUPM (0x5A00 + 4*n [n=0...31]; RO) 636
8.20.6 Proxying Filter Control Register - PROXYFC (0x5F60; R/W) 636
8.20.7 Proxying Status Register - PROXYS (0x5F64; R/W1C) 638
8.20.8 IP Address Valid - IPAV (0x5838; R/W) 639
8.20.9 IPv4 Address Table - IP4AT (0x5840 + 8*n [n=0...3]; R/W) 639
8.20.10 IPv6 Address Table - IP6AT (0x5880 + 4*n [n=0...3]; R/W) 640
8.20.11 Flexible Host Filter Table Registers - FHFT (0x9000 - 0x93FC; RW) 640
8.20.11.1 Flex Filter Queueing Field 641
8.20.11.2 Flex Filter 0 - Example 642
8.20.12 Flexible Host Filter Table Extended Registers - FHFT_EXT (0x9A00 - 0x9DFC; RW) 642
8.21 Management Register Descriptions 642
8.21.1 Management VLAN TAG Value - MAVTV (0x5010 +4*n [n=0...7]; RW) 642
8.21.2 Management Flex UDP/TCP Ports - MFUTP (0x5030 + 4*n [n=0...3]; RW) 643
8.21.3 Management Ethernet Type Filters- METF (0x5060 + 4*n [n=0...3]; RW) 643
8.21.4 Management Control Register - MANC (0x5820; RW) 644
8.21.5 Management Only Traffic Register - MNGONLY (0x5864; RW) 645
8.21.6 Manageability Decision Filters- MDEF (0x5890 + 4*n [n=0...7]; RW) 646
8.21.7 Manageability Decision Filters- MDEF_EXT (0x5930 + 4*n[n=0...7]; RW) 647
8.21.8 Manageability IP Address Filter - MIPAF (0x58B0 + 4*n [n=0...15]; RW) 648
8.21.9 Manageability MAC Address Low - MMAL (0x5910 + 8*n [n= 0...3]; RW) 650
8.21.10 Manageability MAC Address High - MMAH (0x5914 + 8*n [n=0...3]; RW) 651
8.21.11 Flexible TCO Filter Table registers - FTFT (0x9400 - 0x94FC; RW) 651
8.22 Management-Host Interface Register Descriptions 653
8.22.1 Host Slave Command Interface to MMS 653
8.22.1.1 Host Slave Command I/F flow 653
8.22.1.2 HOST Interface Control Register - HICR (0x8F00; RW) 654

8.22.2 General Manageability HOST CSR Registers 654
8.23 Memory Error Registers Description 654
8.23.1 Parity Error Indication- PEIND (0x1084; RC) 655
8.23.2 Parity and ECC Indication Mask – PEINDM (0x1088; RW) 655
8.23.3 DMA Transmit Parity and ECC Control - DTPARC (0x3F00; RW) 655
8.23.4 DMA Transmit Parity and ECC Status- DTPARS (0x3F10; R/W1C) 656
8.23.5 DMA Receive Parity and ECC Control - DRPARC (0x3F04; RW) 657
8.23.6 DMA Receive Parity and ECC Status - DRPARS (0x3F14; R/W1C) 658
8.23.7 Dhost ECC Control - DDECCC (0x3F08; RW) 658
8.23.8 Dhost ECC Status - DDECCS (0x3F18; R/W1C) 659
8.23.9 Rx Packet Buffer ECC Status - RPBECCSTS (0x245C; RW) 659
8.23.10 Tx Packet Buffer ECC Status - TPBECCSTS (0x345C; RW) 660
8.23.11 PCIe Parity Control Register - PCIEERRCTL (0x5BA0; RW) 660
8.23.12 PCIe Parity Status Register - PCIEERRSTS (0x5BA8; R/W1C) 661
8.23.13 PCIe ECC Control Register - PCIEECCCTL (0x5BA4; RW) 662
8.23.14 PCIe ECC Status Register - PCIEECCSTS (0x5BAC; R/W1C) 663
8.23.15 LAN Port Parity Error Control Register - LANPERRCTL (0x5F54; RW) 663
8.23.16 LAN Port Parity Error Status Register - LANPERRSTS (0x5F58; R/W1C) 664
8.24 Power Management Register Description 665
8.24.1 DMA Coalescing Control Register - DMACR (0x2508; R/W) 665
8.24.2 DMA Coalescing Transmit Threshold - DMCTXTH (0x3550;RW) 667
8.24.3 DMA Coalescing Time to Lx Request - DMCTLX (0x2514;RW) 667
8.24.4 DMA Coalescing Receive Packet Rate Threshold - DMCRTRH (0x5DD0;RW) 668
8.24.5 DMA Coalescing Current RX Count - DMCCNT (0x5DD4;RO) 668
8.24.6 Flow Control Receive Threshold Coalescing - FCRTC (0x2170; R/W) 669
8.24.7 Latency Tolerance Reporting (LTR) Minimum Values - LTRMINV (0x5BB0; R/W) 669
8.24.8 Latency Tolerance Reporting (LTR) Maximum Values - LTRMAXV (0x5BB4; R/W) 670
8.24.9 Latency Tolerance Reporting (LTR) Control - LTRC (0x01A0; R/W) 671
8.24.10 Energy Efficient Ethernet (EEE) Register - EEER (0x0E30; R/W) 672
8.25 Thermal Sensor Registers Description 673
8.25.1 Thermal Sensor Measured Junction Temperature - THMJT (0x8100; RO) 673
8.25.2 Thermal Sensor Low Threshold Control - THLOWTC (0x8104; RW) 674
8.25.3 Thermal Sensor Mid Threshold Control - THMIDTC (0x8108; RW) 676
8.25.4 Thermal Sensor High Threshold Control - THHIGHTC (0x810C; RW) 678
8.25.5 Thermal Sensor Status - THSTAT (0x8110; RO) 680
8.25.6 Thermal Sensor Auxiliary Configuration - THACNFG (0x8114; RW) 681
8.25.7 Rx Packet Buffer Wrap Around Counter - PBRWAC (0x24E8; RO) 683
8.26 PHY Software Interface 683
8.26.1 Internal PHY Configuration - IPCNFG (0x0E38, RW) 683
8.26.2 PHY Power Management - PHPM (0x0E14, RW) 684
8.26.3 Internal PHY Software Interface (PHYREG) 685
8.26.3.1 PHY Control Register - PCTRL (00d; R/W) 686
8.26.3.2 PHY Status Register - PSTATUS (01d; RO) 687
8.26.3.3 PHY Identifier Register 1 (LSB) - PHY ID 1 (02d; RO) 688
8.26.3.4 PHY Identifier Register 2 (MSB) - PHY ID 2 (03d; RO) 688
8.26.3.5 Auto–Negotiation Advertisement Register - ANA (04d; R/W) 688
8.26.3.6 Auto–Negotiation Link Partner Ability Register - ANLPA (05d; RO) 689
8.26.3.7 Auto–Negotiation Expansion Register - ANE (06d; RO) 690
8.26.3.8 Auto–Negotiation Next Page Transmit Register - NPT (07d; R/W) 690
8.26.3.9 Auto–Negotiation Next Page Link Partner Register - LPN (08d; RO) 691
8.26.3.10 1000BASE–T/100BASE–T2 Control Register - GCON (09d; R/W) 691
8.26.3.11 1000BASE–T/100BASE–TX Status Register - GSTATUS (10d; RO) 692
8.26.3.12 Extended Status Register - ESTATUS (15d; RO) 692
8.26.3.13 Extended Memory Indirect Address Register - EMIADD (16d; R/W) 693
8.26.3.14 Extended Memory Indirect Data Register - EMIDATA (17d; R/W) 693
8.26.3.15 EEE MMD Extended Register support 693
8.26.3.16 PHY Control Register 2 - PHCTRL2(18d; R/W) 694
8.26.3.17 Loopback Control Register - PHLBKC (19d; R/W) 695
8.26.3.17.1 Loopback Mode Setting 696
8.26.3.18 RX Error Counter Register - PHRERRC (20d; RO) 696
8.26.3.19 Management Interface (MI) Control Register - PHMIC (21d; R/W) 696
8.26.3.20 PHY Configuration Register - PHCNFG (22d; R/W) 697
8.26.3.21 PHY Control Register 1 - PHCTRL1 (23d; R/W) 698
8.26.3.22 Interrupt Mask Register - PHINTM (24d; R/W) 699
8.26.3.23 Interrupt Status Register - PHINT (25d; RC) 699
8.26.3.24 PHY Status Register - PHSTAT (26d; RO) 700
8.26.3.25 Diagnostics Control Register (Linking Disabled) - PHDIAG (30d; R/W) 701
8.26.3.26 Diagnostics Status Register (Linking Disabled) - PHDSTAT (31d; RO 702
8.26.3.27 Diagnostics Status Register (Linking Enabled) - PHDSTAT (31d; RO) 702
8.27 Virtual Function Device Registers 703
8.27.1 Queues Registers 703
8.27.2 Non-Queue Registers 703
8.27.2.1 EITR Registers 703
8.27.2.2 MSI-X Registers 704
8.27.3 Register Set - CSR BAR 704

8.27.4 Register Set - MSI-X BAR 705
8.28 Virtual Function Register Descriptions 706
8.28.1 VT Control Register - VTCTRL (0x0000; WO) 706
8.28.2 VF Status Register - STATUS (0x0008; RO) 706
8.28.3 VT Free Running Timer - VTFRTIMER (0x1048; RO) 706
8.28.4 VT Extended Interrupt Cause - VTEICR (0x1580; RC/W1C) 706
8.28.5 VT Extended Interrupt Cause Set - VTEICS (0x1520; WO) 707
8.28.6 VT Extended Interrupt Mask Set/Read - VTEIMS (0x1524; RWS) 707
8.28.7 VT Extended Interrupt Mask Clear - VTEIMC (0x1528; WO) 707
8.28.8 VT Extended Interrupt Auto Clear - VTEIAC (0x152C; RW) 707
8.28.9 VT Extended Interrupt Auto Mask Enable - VTEIAM (0x1530; RW) 708
8.28.10 VT Interrupt Throttle - VTEITR (0x01680 + 4*n[n = 0...2]; RW) 708
8.28.11 VT Interrupt Vector Allocation Registers - VTIVAR (0x1700; RW) 708
8.28.12 VT Interrupt Vector Allocation Registers - VTIVAR_MISC (0x1740; RW) 708
8.28.13 MSI—X Table Entry Lower Address - MSIXTADD (BAR3: 0x0000 + 16*n [n=0...2]; R/W) 709
8.28.14 MSI—X Table Entry Upper Address - SIXTUADD (BAR3: 0x0004 + 16*n [n=0...2]; R/W) 709
8.28.15 MSI-X Table Entry Message - MSIXTMSG (BAR3: 0x0008 + 16*n [n=0...2]; R/W) 709
8.28.16 MSI-X Table Entry Vector Control - MSIXTVCTRL (BAR3: 0x000C + 16*n [n=0...2]; R/W) 709
8.28.17 MSI-X Pending Bits - MSIXPBA (BAR3: 0x2000; RO) 710
8.28.18 MSI-X PBA Clear - PBACL (0x0F04; R/W1C) 710
8.28.19 Receive Descriptor Base Address Low - RDBAL (0x2800; RW) 710
8.28.20 Receive Descriptor Base Address High - RDBAH (0x2804; RW) 710
8.28.21 Receive Descriptor Ring Length - RDLEN (0x2808; RW) 710
8.28.22 Receive Descriptor Head - RDH (0x2810; RW) 710
8.28.23 Receive Descriptor Tail - RDT (0x2818; RW) 711
8.28.24 Receive Descriptor Control - RXDCTL (0x2828; RW) 711
8.28.25 Split and Replication Receive Control Register queue - SRRCTL(0x280C; RW) 711
8.28.26 Receive Queue drop packet count - RQDPC (0x2830; RO) 711
8.28.27 Replication Packet Split Receive Type - PSRTYPE (0x0F0C; RW) 711
8.28.28 Transmit Descriptor Base Address Low - TDBAL (0x3800; RW) 711
8.28.29 Transmit Descriptor Base Address High - TDBAH (0x3804; RW) 711
8.28.30 Transmit Descriptor Ring Length - TDLEN (0x3808; RW) 711
8.28.31 Transmit Descriptor Head - TDH (0x3810; RW) 712
8.28.32 Transmit Descriptor Tail - TDT (0x3818; RW) 712
8.28.33 Transmit Descriptor Control - TXDCTL (0x3828; RW) 712
8.28.34 Transmit Queue drop packet count - TQDPC (0xE030 + 0x40*n [n=0...7]; RO) 712
8.28.35 Tx Descriptor Completion Write–Back Address Low - TDWBAL (0x3838; RW) 712
8.28.36 Tx Descriptor Completion Write–Back Address High - TDWBAH (0x383C; RW) 712
8.28.37 Rx DCA Control Registers - RXCTL (0x2814; RW) 712
8.28.38 Tx DCA Control Registers - TXCTL (0x3814; RW) 712
8.28.39 Good Packets Received Count - VFGPRC (0x0F10; RO) 713
8.28.40 Good Packets Transmitted Count - VFGPTC (0x0F14; RO) 713
8.28.41 Good Octets Received Count - VFGORC (0x0F18; RO) 713
8.28.42 Good Octets Transmitted Count - VFGOTC (0x0F34; RO) 713
8.28.43 Multicast Packets Received Count - VFMPRC (0x0F38; RO) 713
8.28.44 Good TX Octets loopback Count - VFGOTLBC (0x0F50; RO) 713
8.28.45 Good TX packets loopback Count - VFGPTLBC (0x0F44; RO) 713
8.28.46 Good RX Octets loopback Count - VFGORLBC (0x0F48; RO) 714
8.28.47 Good RX Packets loopback Count - VFGPRLBC (0x0F40; RO) 714
8.28.48 Virtual Function Mailbox - VFMailbox (0x0C40; RW) 714
8.28.49 Virtualization Mailbox memory - VMBMEM (0x0800:0x083C; RW) 714

9 PCIe* Programming Interface 715

9.1 PCIe Compatibility 715
9.2 Configuration Sharing Among PCI Functions 716
9.3 PCIe Register Map 717
9.3.1 Register Attributes 717
9.3.2 PCIe Configuration Space Summary 718
9.4 Mandatory PCI Configuration Registers 722
9.4.1 Vendor ID (0x0; RO) 722
9.4.2 Device ID (0x2; RO) 722
9.4.3 Command Register (0x4; R/W) 723
9.4.4 Status Register (0x6; RO) 724
9.4.5 Revision (0x8; RO) 724
9.4.6 Class Code (0x9; RO) 724
9.4.7 Cache Line Size (0xC; R/W) 725
9.4.8 Latency Timer (0xD; RO) 725
9.4.9 Header Type (0xE; RO) 725
9.4.10 BIST (0xF; RO) 725
9.4.11 Base Address Registers (0x10...0x27; R/W) 725

9.4.11.1 32-bit LAN BARs Mode Mapping 726
9.4.11.2 64-bit LAN BARs Mode Mapping 726
9.4.11.3 Dummy Function BARs Mapping 727
9.4.11.4 Base Address Register Fields 727
9.4.12 CardBus CIS (0x28; RO) 727
9.4.13 Subsystem Vendor ID (0x2C; RO) 727
9.4.14 Subsystem ID (0x2E; RO) 728
9.4.15 Expansion ROM Base Address (0x30; RW) 728
9.4.16 Cap_Ptr (0x34; RO) 728
9.4.17 Interrupt Line (0x3C; RW) 728
9.4.18 Interrupt Pin (0x3D; RO) 728
9.4.19 Max_Lat/Min_Gnt (0x3E; RO) 729
9.5 PCI Capabilities 729
9.5.1 PCI Power Management Capability 729
9.5.1.1 Capability ID (0x40; RO) 730
9.5.1.2 Next Pointer (0x41; RO) 730
9.5.1.3 Power Management Capabilities - PMC (0x42; RO) 730
9.5.1.4 Power Management Control / Status Register - PMCSR (0x44; R/W) 731
9.5.1.5 Bridge Support Extensions - PMCSR_BSE (0x46; RO) 731
9.5.1.6 Data Register (0x47; RO) 732
9.5.2 MSI Configuration 732
9.5.2.1 Capability ID (0x50; RO) 732
9.5.2.2 Next Pointer (0x51; RO) 732
9.5.2.3 Message Control (0x52; R/W) 732
9.5.2.4 Message Address Low (0x54; R/W) 733
9.5.2.5 Message Address High (0x58; R/W) 733
9.5.2.6 Message Data (0x5C; R/W) 733
9.5.2.7 Mask Bits (0x60; R/W) 733
9.5.2.8 Pending Bits (0x64; R/W) 734
9.5.3 MSI-X Configuration 734
9.5.3.1 Capability ID (0x70; RO) 735
9.5.3.2 Next Pointer (0x71; RO) 735
9.5.3.3 Message Control (0x72; R/W) 735
9.5.3.4 MSI-X Table Offset (0x74; R/W) 736
9.5.3.5 MSI-X Pending Bit Array - PBA Offset (0x78; R/W) 736
9.5.4 CSR Access Via Configuration Address Space 736
9.5.4.1 IOADDR Register (0x98; R/W) 736
9.5.4.2 IODATA Register (0x9C; R/W) 737
9.5.5 Vital Product Data Registers 737
9.5.5.1 Capability ID (0xE0; RO) 737
9.5.5.2 Next Pointer (0xE1; RO) 737
9.5.5.3 VPD Address (0xE2; RW) 738
9.5.5.4 VPD Data (0xE4; RW) 738
9.5.6 PCIe Configuration Registers 738
9.5.6.1 Capability ID (0xA0; RO) 739
9.5.6.2 Next Pointer (0xA1; RO) 739
9.5.6.3 PCIe CAP (0xA2; RO) 739
9.5.6.4 Device Capabilities (0xA4; RO) 740
9.5.6.5 Device Control (0xA8; RW) 740
9.5.6.6 Device Status (0xAA; R/W1C) 741
9.5.6.7 Link Capabilities Register (0xAC; RO) 742
9.5.6.8 Link Control Register (0xB0; RO) 743
9.5.6.9 Link Status (0xB2; RO) 744
9.5.6.10 Reserved (0xB4-0xC0; RO) 745
9.5.6.11 Device Capabilities 2 (0xC4; RO) 745
9.5.6.12 Device Control 2 (0xC8; RW) 746
9.5.6.13 Link Control 2 (0xD0; RW) 748
9.5.6.14 Link Status 2 (0xD2; RW) 749
9.6 PCIe Extended Configuration Space 750
9.6.1 Advanced Error Reporting (AER) Capability 751
9.6.1.1 PCIe CAP ID (0x100; RO) 751
9.6.1.2 Uncorrectable Error Status (0x104; R/W1CS) 751
9.6.1.3 Uncorrectable Error Mask (0x108; RWS) 752
9.6.1.4 Uncorrectable Error Severity (0x10C; RWS) 753
9.6.1.5 Correctable Error Status (0x110; R/W1CS) 754
9.6.1.6 Correctable Error Mask (0x114; RWS) 754
9.6.1.7 Advanced Error Capabilities and Control Register (0x118; RWS) 755
9.6.1.8 Header Log (0x11C:0x128; RO) 755
9.6.2 Serial Number 756
9.6.2.1 Device Serial Number Enhanced Capability Header (0x140; RO) 756
9.6.2.2 Serial Number Register (0x144:0x148; RO) 757
9.6.3 ARI Capability Structure 758
9.6.3.1 PCIe ARI Header Register (0x150; RO) 758
9.6.3.2 PCIe ARI Capabilities & Control Register (0x154; RO) 759
9.6.4 SR-IOV Capability Structure 759

9.6.4.1 PCIe SR-IOV Header Register (0x160; RO) 760
9.6.4.2 PCIe SR-IOV Capabilities Register (0x164; RO) 760
9.6.4.3 PCIe SR-IOV Status and Control Register (0x168; RW) 761
9.6.4.4 PCIe SR-IOV Max/Total VFs Register (0x16C) 761
9.6.4.5 PCIe SR-IOV Num VFs Register (0x170; R/W) 762
9.6.4.6 PCIe SR-IOV VF RID Mapping Register (0x174; RO) 762
9.6.4.7 PCIe SR-IOV VF device ID (0x178; RO) 763
9.6.4.8 PCIe SR-IOV Supported Page Size Register (0x17C; RO) 763
9.6.4.9 PCIe SR-IOV System Page Size Register (0x180; R/W) 763
9.6.4.10 PCIe SR-IOV BAR 0 - Low Register (0x184; R/W) 763
9.6.4.11 PCIe SR-IOV BAR 0 - High Register (0x188; R/W) 764
9.6.4.12 PCIe SR-IOV BAR 2 (0x18C; RO) 765
9.6.4.13 PCIe SR-IOV BAR 3 - Low Register (0x190; R/W) 765
9.6.4.14 PCIe SR-IOV BAR 3 - High Register (0x194; R/W) 765
9.6.4.15 PCIe SR-IOV BAR 5 (0x198; RO) 765
9.6.4.16 PCIe SR-IOV VF Migration State Array Offset (0x19C; RO) 765
9.6.5 TLP Processing Hint Requester (TPH) Capability 766
9.6.5.1 TPH CAP ID (0x1A0; RO) 766
9.6.5.2 TPH Requester Capabilities (0x1A4; RO) 767
9.6.5.3 TPH Requester Control (0x1A8; R/W) 767
9.6.5.4 TPH Steering Table (0x1AC - 0x1B8; R/W) 768
9.6.6 Latency Tolerance Requirement Reporting (LTR) Capability 768
9.6.6.1 LTR CAP ID (0x1C0; RO) 768
9.6.6.2 LTR Capabilities (0x1C4; RW) 769
9.6.7 Access Control Services (ACS) Capability 769
9.6.7.1 ACS CAP ID (0x1D0; RO) 770
9.6.7.2 ACS Control and Capabilities (0x1D4; RO) 770
9.7 Virtual Functions (VF) Configuration Space 770
9.7.1 Legacy Header Details 772
9.7.2 Legacy Capabilities 773
9.7.2.1 MSI-X Capability 773
9.7.2.2 PCIe Capability Registers 774
9.7.3 Advanced Capabilities 775
9.7.3.1 Advanced Error Reporting Registers 775

10 System Manageability 779

10.1 Pass-Through (PT) Functionality 779
10.1.1 Pass Through Packet Routing 780
10.2 Components of the Sideband Interface 780
10.2.1 Physical Layer 780
10.2.1.1 SMBus 780
10.2.1.2 NC-SI 781
10.2.2 Logical Layer 781
10.2.2.1 Legacy SMBus 781
10.2.2.2 NC-SI 781
10.3 Packet Filtering 781
10.3.1 Manageability Receive Filtering 782
10.3.2 L2 Filters 783
10.3.2.1 MAC and VLAN Filters 783
10.3.2.2 EtherType Filters 783
10.3.3 L3/L4 Filtering 784
10.3.3.1 ARP Filtering 784
10.3.3.2 Neighbor Discovery Filtering 784
10.3.3.3 RMCP Filtering 784
10.3.3.4 Flexible Port Filtering 784
10.3.3.5 Flexible 128 Byte Filter 785
10.3.3.5.1 Flexible Filter Structure 785
10.3.3.5.2 TCO Filter Programming 785
10.3.3.6 IP Address Filtering 786
10.3.3.7 Checksum Filtering 786
10.3.4 Configuring Manageability Filters 786
10.3.4.1 Manageability Decision Filters (MDEF and MDEF_EXT) 786
10.3.4.2 Exclusive Traffic 789
10.3.5 Possible Configurations 790
10.3.5.1 Dedicated MAC Packet Filtering 790
10.3.5.2 Broadcast Packet Filtering 790
10.3.5.3 VLAN Packet Filtering 790
10.3.5.4 IPv6 Filtering 790
10.3.5.5 Receive Filtering with Shared IP 791
10.3.6 Determining Manageability MAC Address 791
10.4 OS to BMC Traffic 792
10.4.1 Overview 792
10.4.2 Filtering 793
10.4.2.1 OS2BMC Filtering 793
10.4.2.2 Handling of OS to BMC Packets 793

10.4.2.3 BMC to OS Filtering 794
10.4.2.4 Queuing of Packets Received from the BMC. 794
10.4.2.5 Offloads of Packets received from the BMC. 794
10.4.3 Blocking of Network to BMC Flow 795
10.4.4 Statistics 795
10.4.5 OS to BMC Enablement 796
10.5 SMBus Pass-Through Interface 796
10.5.1 General 796
10.5.2 Pass-Through Capabilities 796
10.5.3 Port to SMBus Mapping 796
10.5.4 Automatic Ethernet ARP Operation 797
10.5.4.1 ARP Packet Formats 797
10.5.5 SMBus Transactions 798
10.5.5.1 SMBus Addressing 799
10.5.5.2 SMBus ARP Functionality 799
10.5.5.3 SMBus ARP Flow 799
10.5.5.4 SMBus ARP UDID Content 802
10.5.5.5 SMBus ARP in Multi/Single Mode 803
10.5.5.6 Concurrent SMBus Transactions 803
10.5.6 SMBus Notification Methods 803
10.5.6.1 SMBus Alert and Alert Response Method 804
10.5.6.2 Asynchronous Notify Method 805
10.5.6.3 Direct Receive Method 805
10.5.7 Receive TCO Flow 805
10.5.8 Transmit TCO Flow 806
10.5.8.1 Transmit Errors in Sequence Handling 807
10.5.8.2 TCO Command Aborted Flow 807
10.5.9 SMBus ARP Transactions 807
10.5.9.1 Prepare to ARP 807
10.5.9.2 Reset Device (General) 808
10.5.9.3 Reset Device (Directed) 808
10.5.9.4 Assign Address 808
10.5.9.5 Get UDID (General and Directed) 809
10.5.10 SMBus Pass-Through Transactions 811
10.5.10.1 Write SMBus Transactions 811
10.5.10.1.1 Transmit Packet Command 811
10.5.10.1.2 Request Status Command 811
10.5.10.1.3 Receive Enable Command 812
10.5.10.1.4 Force TCO Command 814
10.5.10.1.5 Management Control 814
10.5.10.1.6 Update Management Receive Filter Parameters 815
10.5.10.1.7 Set Thermal Sensor Configuration 817
10.5.10.1.8 Perform Thermal Sensor Action 818
10.5.10.2 Read SMBus Transactions 818
10.5.10.2.1 Receive TCO LAN Packet Transaction 819
10.5.10.2.2 Read Status Command 821
10.5.10.2.3 Get System MAC Address Command 823
10.5.10.2.4 Read Management Parameters Command 824
10.5.10.2.5 Read Management Receive Filter Parameters Command 825
10.5.10.2.6 Read Receive Enable Configuration Command 826
10.5.10.2.7 Get Thermal Sensor Capabilities Command 827
10.5.10.2.8 Get Thermal Sensor Configuration Command 828
10.5.10.2.9 Get Thermal Sensor Status Command 829
10.5.11 Example Configuration Steps 829
10.5.11.1 Example 1 - Shared MAC, RMCP Only Ports 829
10.5.11.1.1 Example 1 Pseudo Code 829
10.5.11.2 Example 2 - Dedicated MAC, Auto ARP Response and RMCP Port Filtering 831
10.5.11.2.1 Example 2 - Pseudo Code 831
10.5.11.3 Example 3 - Dedicated MAC & IP Address 833
10.5.11.3.1 Example 3 - Pseudo Code 833
10.5.11.4 Example 4 - Dedicated MAC and VLAN Tag 835
10.5.11.4.1 Example 4 - Pseudo Code 835
10.5.12 SMBus Troubleshooting 837
10.5.12.1 TCO Alert Line Stays Asserted After a Power Cycle 837
10.5.12.2 When SMBus Commands Are Always NACK'd 838
10.5.12.3 SMBus Clock Speed Is 16.6666 KHz 838
10.5.12.4 A Network Based Host Application Is Not Receiving Any Network Packets 838
10.5.12.5 Unable to Transmit Packets from the BMC 838
10.5.12.6 SMBus Fragment Size 839
10.5.12.7 Losing Link 839
10.5.12.8 Enable XSum Filtering 840
10.5.12.9 Still Having Problems? 840
10.6 NC-SI Pass Through Interface 840

10.6.1 Overview 840
10.6.1.1 Terminology 840
10.6.1.2 System Topology 841
10.6.1.3 Data Transport 843
10.6.1.3.1 Control Frames 843
10.6.1.3.2 NC-SI Frames Receive Flow 843
10.6.2 Supported Features 844
10.6.2.1 Set Link Error Codes 846
10.6.3 NC-SI Mode — Intel Specific Commands 847
10.6.3.1 Overview 847
10.6.3.1.1 OEM Command (0x50) 847
10.6.3.1.2 OEM Response (0xD0) 848
10.6.3.2 Command Summary 848
10.6.3.3 Set Intel Filters Control — IP Filters Control Command (Intel Command 0x00, Filter Control Index 0x00) 850
10.6.3.3.1 Set Intel Filters Control — IP Filters Control Response (Intel Command 0x00, Filter Control Index 0x00) 850
10.6.3.4 Get Intel Filters Control Commands (Intel Command 0x01) 850
10.6.3.4.1 Get Intel Filters Control — IP Filters Control Command (Intel Command 0x01, Filter Control Index 0x00) 850
10.6.3.4.2 Get Intel Filters Control — IP Filters Control Response (Intel Command 0x01, Filter Control Index 0x00) 851
10.6.3.5 Set Intel Filters Formats 851
10.6.3.5.1 Set Intel Filters Command (Intel Command 0x02) 851
10.6.3.5.2 Set Intel Filters Response (Intel Command 0x02) 851
10.6.3.5.3 Set Intel Filters — Manageability Only Command (Intel Command 0x02, Filter Parameter 0x0F) 851
10.6.3.5.4 Set Intel Filters — Manageability Only Response (Intel Command 0x02, Filter Parameter 0x0F) 852
10.6.3.5.5 Set Intel Filters — Flex Filter Enable Mask and Length Command (Intel Command 0x02, Filter Parameter 0x10) 852
10.6.3.5.6 Set Intel Filters — Flex Filter Enable Mask and Length Response (Intel Command 0x02, Filter Parameter 0x10) 853
10.6.3.5.7 Set Intel Filters — Flex Filter Data Command (Intel Command 0x02, Filter Parameter 0x11) 853
10.6.3.5.8 Set Intel Filters — Flex Filter Data Response (Intel Command 0x02, Filter Parameter 0x11) 853
10.6.3.5.9 Set Intel Filters — Packet Addition Decision Filter Command (Intel Command 0x02, Filter Parameter 0x61) 854
10.6.3.5.10 Set Intel Filters — Packet Addition Decision Filter Response (Intel Command 0x02, Filter Parameter 0x61) 855
10.6.3.5.11 Set Intel Filters — Flex TCP/UDP Port Filter Command (Intel Command 0x02, Filter Parameter 0x63) 855
10.6.3.5.12 Set Intel Filters — Flex TCP/UDP Port Filter Response (Intel Command 0x02, Filter Parameter 0x63) 855
10.6.3.5.13 Set Intel Filters — IPv4 Filter Command (Intel Command 0x02, Filter Parameter 0x64) 855
10.6.3.5.14 Set Intel Filters — IPv4 Filter Response (Intel Command 0x02, Filter Parameter 0x64) 856
10.6.3.5.15 Set Intel Filters — IPv6 Filter Command (Intel Command 0x02, Filter Parameter 0x65) 856
10.6.3.5.16 Set Intel Filters — IPv6 Filter Response (Intel Command 0x02, Filter Parameter 0x65) 856
10.6.3.5.17 Set Intel Filters - EtherType Filter Command (Intel Command 0x02, Filter parameter 0x67) 857
10.6.3.5.18 Set Intel Filters - EtherType Filter Response (Intel Command 0x02, Filter parameter 0x67) 857
10.6.3.5.19 Set Intel Filters - Packet Addition Extended Decision Filter Command (Intel Command 0x02, Filter parameter 0x68) 857
10.6.3.5.20 Set Intel Filters – Packet Addition Extended Decision Filter Response (Intel Command 0x02, Filter parameter 0x68) 859
10.6.3.6 Get Intel Filters Formats 859
10.6.3.6.1 Get Intel Filters Command (Intel Command 0x03) 859
10.6.3.6.2 Get Intel Filters Response (Intel Command 0x03) 859
10.6.3.6.3 Get Intel Filters — Manageability Only Command (Intel Command 0x03, Filter Parameter 0x0F) 860
10.6.3.6.4 Get Intel Filters — Manageability Only Response (Intel Command 0x03, Filter Parameter 0x0F) 860
10.6.3.6.5 Get Intel Filters — Flex Filter 0 Enable Mask and Length Command (Intel Command 0x03, Filter Parameter 0x10) 860
10.6.3.6.6 Get Intel Filters — Flex Filter 0 Enable Mask and Length Response (Intel Command 0x03, Filter Parameter 0x10) 861
10.6.3.6.7 Get Intel Filters — Flex Filter 0 Data Command (Intel Command 0x03, Filter Parameter 0x11) 861

10.6.3.6.8 Get Intel Filters — Flex Filter 0 Data Response (Intel Command 0x03, Filter Parameter 0x11) 861
10.6.3.6.9 Get Intel Filters — Packet Addition Decision Filter Command (Intel Command 0x03, Filter Parameter 0x61) 862
10.6.3.6.10 Get Intel Filters — Packet Addition Decision Filter Response (Intel Command 0x03, Filter Parameter 0x61) 862
10.6.3.6.11 Get Intel Filters — Flex TCP/UDP Port Filter Command (Intel Command 0x03, Filter Parameter 0x63) 862
10.6.3.6.12 Get Intel Filters — Flex TCP/UDP Port Filter Response (Intel Command 0x03, Filter Parameter 0x63) 863
10.6.3.6.13 Get Intel Filters — IPv4 Filter Command (Intel Command 0x03, Filter Parameter 0x64) 863
10.6.3.6.14 Get Intel Filters — IPv4 Filter Response (Intel Command 0x03, Filter Parameter 0x64) 863
10.6.3.6.15 Get Intel Filters — IPv6 Filter Command (Intel Command 0x03, Filter Parameter 0x65) 864
10.6.3.6.16 Get Intel Filters — IPv6 Filter Response (Intel Command 0x03, Filter parameter 0x65) 864
10.6.3.6.17 Get Intel Filters - EtherType Filter Command (Intel Command 0x03, Filter parameter 0x67) 864
10.6.3.6.18 Get Intel Filters - EtherType Filter Response (Intel Command 0x03, Filter parameter 0x67) 865
10.6.3.6.19 Get Intel Filters – Packet Addition Extended Decision Filter Command (Intel Command 0x03, Filter parameter 0x68) 865
10.6.3.6.20 Get Intel Filters – Packet Addition Extended Decision Filter Response (Intel Command 0x03, Filter parameter 0x68) 865
10.6.3.7 Set Intel Packet Reduction Filters Formats 866
10.6.3.7.1 Set Intel Packet Reduction Filters Command (Intel Command 0x04) 866
10.6.3.7.2 Set Intel Packet Reduction Filters Response (Intel Command 0x04) 866
10.6.3.7.3 Set Unicast Packet Reduction Command (Intel Command 0x04, Reduction Filter Index 0x00) 867
10.6.3.7.4 Set Unicast Packet Reduction Response (Intel Command 0x04, Reduction Filter Index 0x00) 867
10.6.3.7.5 Set Multicast Packet Reduction Command (Intel Command 0x04, Reduction Filter Index 0x01) 868
10.6.3.7.6 Set Multicast Packet Reduction Response (Intel Command 0x04, Reduction Filter Index 0x01) 868
10.6.3.7.7 Set Broadcast Packet Reduction Command (Intel Command 0x04, Reduction Filter Index 0x02) 868
10.6.3.7.8 Set Broadcast Packet Reduction Response (Intel Command 0x08) 869
10.6.3.7.9 Set Unicast Extended Packet Reduction Command (Intel Command 0x04, Reduction Filter Index 0x10) 869
10.6.3.7.10 Set Unicast Extended Packet Reduction Response (Intel Command 0x04, Reduction Filter Index 0x10) 870
10.6.3.7.11 Set Multicast Extended Packet Reduction Command (Intel Command 0x04, Reduction Filter Index 0x11) 870
10.6.3.7.12 Set Multicast Extended Packet Reduction Response (Intel Command 0x04, Reduction Filter Index 0x11) 870
10.6.3.7.13 Set Broadcast Extended Packet Reduction Command (Intel Command 0x04, Reduction Filter Index 0x12) 871
10.6.3.7.14 Set Broadcast Extended Packet Reduction Response (Intel Command 0x04, Reduction Filter Index 0x12) 871
10.6.3.8 Get Intel Packet Reduction Filters Formats 871
10.6.3.8.1 Get Intel Packet Reduction Filters Command (Intel Command 0x05) 872
10.6.3.8.2 Get Intel Packet Reduction Filters Response (Intel Command 0x05) 872
10.6.3.8.3 Get Unicast Packet Reduction Command (Intel Command 0x05, Reduction Filter Index 0x00) 872
10.6.3.8.4 Get Unicast Packet Reduction Response (Intel Command 0x05, Reduction Filter Index 0x00) 872
10.6.3.8.5 Get Multicast Packet Reduction Command (Intel Command 0x05, Reduction Filter Index 0x01) 873
10.6.3.8.6 Get Multicast Packet Reduction Response (Intel Command 0x05, Reduction Filter Index 0x01) 873
10.6.3.8.7 Get Broadcast Packet Reduction Command (Intel Command 0x05, Reduction Filter Index 0x02) 873
10.6.3.8.8 Get Broadcast Packet Reduction Response (Intel Command 0x05, Reduction Filter Index 0x02) 873
10.6.3.8.9 Get Unicast Extended Packet Reduction Command (Intel Command 0x05, Reduction Filter Index 0x10) 874

10.6.3.8.10 Get Unicast Extended Packet Reduction Response (Intel Command 0x05, Reduction Filter Index 0x10) 874
10.6.3.8.11 Get Multicast Extended Packet Reduction Command (Intel Command 0x05, Reduction Filter Index 0x11) 874
10.6.3.8.12 Get Multicast Extended Packet Reduction Response (Intel Command 0x05, Reduction Filter Index 0x11) 875
10.6.3.8.13 Get Broadcast Extended Packet Reduction Command (Intel Command 0x05, Reduction Filter Index 0x12) 875
10.6.3.8.14 Get Broadcast Extended Packet Reduction Response (Intel Command 0x05, Reduction Filter Index 0x12) 875
10.6.3.9 System MAC Address 875
10.6.3.9.1 Get System MAC Address Command (Intel Command 0x06) 875
10.6.3.9.2 Get System MAC Address Response (Intel Command 0x06) 876
10.6.3.10 Set Intel Management Control Formats 876
10.6.3.10.1 Set Intel Management Control Command (Intel Command 0x20) 876
10.6.3.10.2 Set Intel Management Control Response (Intel Command 0x20) 877
10.6.3.11 Get Intel Management Control Formats 877
10.6.3.11.1 Get Intel Management Control Command (Intel Command 0x21) 877
10.6.3.11.2 Get Intel Management Control Response (Intel Command 0x21) 877
10.6.3.12 TCO Reset 878
10.6.3.12.1 Perform Intel TCO Reset Command (Intel Command 0x22) 878
10.6.3.12.2 Perform Intel TCO Reset Response (Intel Command 0x22) 879
10.6.3.13 Checksum Offloading 879
10.6.3.13.1 Enable Checksum Offloading Command (Intel Command 0x23) 880
10.6.3.13.2 Enable Checksum Offloading Response (Intel Command 0x23) 880
10.6.3.13.3 Disable Checksum Offloading Command (Intel Command 0x24) 880
10.6.3.13.4 Disable Checksum Offloading Response (Intel Command 0x24) 880
10.6.3.14 OS 2 BMC Configuration 881
10.6.3.14.1 EnableOS2BMC Flow Command (Intel Command 0x40, Index 0x1) 881
10.6.3.14.2 EnableOS2BMC Flow Response (Intel Command 0x40, Index 0x1) 881
10.6.3.14.3 Enable Network to BMC Flow Command (Intel Command 0x40, Index 0x2) 881
10.6.3.14.4 Enable Network to BMC Flow Response (Intel Command 0x40, Index 0x2) 881
10.6.3.14.5 Enable both Host and Network to BMC flows Command (Intel Command 0x40, Index 0x3) 882
10.6.3.14.6 Enable both Host and Network to BMC Flows Response (Intel Command 0x40, Index 0x3) 882
10.6.3.14.7 Get OS2BMC parameters Command (Intel Command 0x41) 882
10.6.3.14.8 Get OS2BMC parameters Response (Intel Command 0x41) 882
10.6.3.15 Inventory and Update System Parameters Commands 883
10.6.3.15.1 Get Controller Information Command (Intel Command 0x48, Index 0x1) 883
10.6.3.15.2 Get Controller information Response (Intel Command 0x48, Index 0x1) 883
10.6.3.16 Thermal Sensor Commands 884
10.6.3.16.1 Get Thermal Sensor Commands (Intel Command 0x4C) 884
10.6.3.16.2 Get Thermal Sensor Capabilities Command (Intel Command 0x4C, Index 0x0) 886
10.6.3.16.3 Get Thermal Sensor Capabilities Response (Intel Command 0x4c, Index 0x0) 886
10.6.3.16.4 Get Thermal Sensor Configuration Command (Intel Command 0x4C, Index 0x1) 887
10.6.3.16.5 Get Thermal Sensor Configuration Response (Intel Command 0x4c, Index 0x1) 888
10.6.3.16.6 Get Thermal Sensor Status Command (Intel Command 0x4C, Index 0x2) 888

10.6.3.16.7 Get Thermal Sensor Status Response (Intel Command 0x4c, Index 0x2) 888
10.6.3.16.8 Set Thermal Sensor Commands (Intel Command, Index 0x4D) 889
10.6.3.16.9 Thermal Sensor AEN (Intel AEN 0x81) 891
10.6.4 Basic NC-SI Workflows 891
10.6.4.1 Package States 891
10.6.4.2 Channel States 892
10.6.4.3 Discovery 892
10.6.4.4 Configurations 892
10.6.4.4.1 NC Capabilities Advertisement 892
10.6.4.4.2 Receive Filtering 893
10.6.4.4.3 VLAN 893
10.6.4.5 Pass-Through Traffic States 894
10.6.4.6 Channel Enable 894
10.6.4.7 Network Transmit Enable 894
10.6.5 Asynchronous Event Notifications 895
10.6.6 Querying Active Parameters 895
10.6.7 Resets 895
10.6.8 Advanced Workflows 896
10.6.8.1 Multi-NC Arbitration 896
10.6.8.2 Package Selection Sequence Example 897
10.6.8.3 Multiple Channels (Fail-Over) 897
10.6.8.3.1 Fail-Over Algorithm Example 897
10.6.8.4 Statistics 898
10.6.9 External Link Control 898
10.6.9.1 Set Link While LAN PCIe Functionality is Disabled 899
10.7 MCTP 899
10.7.1 MCTP Overview 899
10.7.1.1 NC-SI Over MCTP 899
10.7.1.2 MCTP Usage Model 900
10.7.1.3 Simplified MCTP Mode 900
10.7.2 NC-SI to MCTP Mappaing 901
10.7.3 MCTP Over SMBus 901
10.7.3.1 SMBus Discovery Process 902
10.7.3.2 SMBus over MCTP implementation notes 902
10.7.4 NC-SI Over MCTP 902
10.7.4.1 NC-SI Packets Format 902
10.7.4.1.1 Control Packets 903
10.7.4.1.2 Command Packets 903
10.7.4.1.3 Response Packets 904
10.7.4.1.4 AEN Packets 905
10.7.5 MCTP Programming 905
10.7.5.1 MCTP Commands Support 906
10.7.5.1.1 Get Endpoint ID 906
10.7.5.1.2 Get Endpoint UUID 907
10.7.5.1.3 Get MCTP Version Support 907
10.7.5.1.4 Get Message Type Support Command 907
10.7.5.1.5 Set Endpoint ID Command 907
10.8 Manageability Host Interface 908
10.8.1 HOST CSR Interface (All Functions) 908
10.8.2 Host Slave Command Interface to Manageability 908
10.8.2.1 Host Slave Command Interface Low Level Flow 908
10.8.2.2 Host Slave Command Registers 909
10.8.2.2.1 Host Interface Control Register (CSR Address 0x8F00) 909
10.8.2.3 Host Interface Structures 909
10.8.2.3.1 Host Interface Command Structure 909
10.8.2.3.2 Host Interface Status Structure 910
10.8.2.3.3 Checksum Calculation Algorithm 910
10.8.2.4 Host Interface Commands 910
10.8.2.4.1 Driver Info Host Command 910
10.8.2.4.2 Host Proxying Commands 911
10.8.3 Host Isolate Support 918

11 Electrical/Mechanical Specification 919

11.1 Introduction 919
11.2 Operating Conditions 919
11.2.1 Recommended Operating Conditions 920
11.3 Power Delivery 920
11.3.1 Power Supply Specification 920
11.3.1.1 Power On/Off Sequence 921
11.3.1.2 Power-On Reset Thresholds 922
11.4 Ball Summary 923
11.5 Current Consumption 923

11.6 DC/AC Specification 927
11.6.1 DC Specifications 927
11.6.1.1 Digital I/O 927
11.6.1.2 LEDs I/O 928
11.6.1.3 Open Drain I/Os 928
11.6.1.4 NC-SI Input and Output Pads 929
11.6.2 Digital I/F AC Specifications 929
11.6.2.1 Reset Signals 929
11.6.2.1.1 LAN_PWR_GOOD 929
11.6.2.2 SMBus 929
11.6.2.3 I^2C AC Specification 931
11.6.2.4 FLASH AC Specification 931
11.6.2.5 EEPROM AC Specification 932
11.6.2.6 NC-SI AC Specification 934
11.6.2.7 JTAG AC Specification 935
11.6.2.8 MDIO AC Specification 935
11.6.2.9 SFP 2 Wires I/F AC Specification 936
11.6.2.10 PCIe/SerDes DC/AC Specification 936
11.6.2.11 PCIe Specification - Receiver 937
11.6.2.12 PCIe Specification - Transmitter 937
11.6.2.13 PCIe Specification - Input Clock 937
11.6.3 Serdes DC/AC Specification 937
11.6.4 PHY Specification 937
11.6.5 XTAL/Clock Specification 937
11.6.5.1 Crystal Specification 938
11.6.5.2 External Clock Oscillator Specification 938
11.6.6 GbE PHY GE_REXT Bias Connection 939
11.6.7 SerDes SE_RSET Bias Connection 940
11.6.8 PCIe PE_TRIM Bias Connection 941
11.6.9 Voltage Regulator Electrical Specifications 942
11.6.9.1 1.0V SVR Electrical Specifications 942
11.6.9.2 SVR Efficiency 943
11.6.9.3 1.8V LVR Electrical Specifications 943
11.7 Package 944
11.7.1 Mechanical Specification for the 17x17 PBGA Package. 944
11.7.1.1 17x17 PBGA Package Schematics 945
11.7.2 Mechanical Specification for the 25x25 PBGA Package 946
11.7.2.1 25x25 PBGA Package Schematics 947
11.8 EEPROM Flash Devices 950
11.8.1 Flash 950
11.8.2 EEPROM 950
11.9 Voltage Regulator External Components 950
11.9.1 1.0V SVR External Components 951
11.9.2 1.8V LVR External Components 952

12 Design Guidelines 955

12.1 Ethernet Interface 955
12.1.1 Magnetics for 1000 BASE-T 955
12.1.2 Magnetics Module Qualification Steps 955
12.1.3 Discrete/Integrated Magnetics Specifications 955
12.1.4 Third-Party Magnetics Manufacturers 957
12.1.5 Layout Considerations for the Ethernet Interface 957
12.1.5.1 Guidelines for Component Placement 957
12.1.5.2 Layout Guidelines for Use with Integrated and Discrete Magnetics 958
12.1.5.3 Board Stack-Up Recommendations 959
12.1.5.4 Differential Pair Trace Routing for 10/100/1000 Designs 960
12.1.5.5 Maximum Trace Lengths Based on Trace Geometry 961
12.1.5.6 Signal Termination and Coupling 962
12.1.5.7 Signal Trace Geometry for 1000 BASE-T Designs 962
12.1.5.8 Trace Length and Symmetry for 1000 BASE-T Designs 963
12.1.5.8.1 Signal Detect 963
12.1.5.9 Impedance Discontinuities 963
12.1.5.10 Reducing Circuit Inductance 963
12.1.5.11 Signal Isolation 964
12.1.5.12 Traces for Decoupling Capacitors 964
12.1.5.13 Light Emitting Diodes for Designs Based on the I350 964
12.1.6 Physical Layer Conformance Testing 964
12.1.6.1 Conformance Tests for 10/100/1000 Mb/s Designs 965
12.1.7 Troubleshooting Common Physical Layout Issues 965
12.2 PCIe 966
12.2.1 Link Width Configuration 966
12.2.2 Polarity Inversion and Lane Reversal 966
12.2.3 PCIe Reference Clock 966
12.3 Clock Source 966
12.3.1 Frequency Control Device Design Considerations 967

12.3.2 Frequency Control Component Types 967
12.3.2.1 Quartz Crystal 967
12.3.2.2 Fixed Crystal Oscillator 967
12.3.2.3 Programmable Crystal Oscillators 968
12.3.2.4 Ceramic Resonator 968
12.4 Crystal Support 968
12.4.1 Crystal Selection Parameters 968
12.4.1.1 Vibrational Mode 969
12.4.1.2 Nominal Frequency 969
12.4.1.3 Frequency Tolerance 969
12.4.1.4 Temperature Stability and Environmental Requirements 969
12.4.1.5 Calibration Mode 969
12.4.1.6 Load Capacitance 970
12.4.1.7 Shunt Capacitance 970
12.4.1.8 Equivalent Series Resistance 970
12.4.1.9 Drive Level 970
12.4.1.10 Aging 971
12.4.1.11 Reference Crystal 971
12.4.1.11.1 Reference Crystal Selection 971
12.4.1.11.2 Circuit Board 971
12.4.1.11.3 Temperature Changes 972
12.4.2 Crystal Placement and Layout Recommendations 972
12.5 Oscillator Support 974
12.5.1 Oscillator Placement and Layout Recommendations 975
12.6 Device Disable 975
12.6.1 BIOS Handling of Device Disable 975
12.7 SMBus and NC-SI 976
12.8 NC-SI 977
12.8.1 Design Requirements 977
12.8.1.1 Network Controller 977
12.8.1.2 External Management Controller (MC) 977
12.8.1.3 Reference Schematic 977
12.8.2 Layout Requirements 980
12.8.2.1 Board Impedance 980
12.8.2.2 Trace Length Restrictions 980

13 Thermal Management 985

13.1 Thermal Sensor and Thermal Diode 985
13.2 Thermal Design Considerations 986
13.3 Terminology 987
13.4 Thermal Specifications 987
13.4.1 Case Temperature 988
13.5 Thermal Attributes 989
13.5.1 Designing for Thermal Performance 989
13.5.2 Typical System Definition 989
13.5.3 Package Mechanical Attributes 989
13.5.4 Package Thermal Characteristics 989
13.6 Thermal Enhancements 993
13.6.1 Clearances 993
13.6.2 Default Enhanced Thermal Solution 993
13.6.3 Extruded Heatsinks 993
13.6.4 Attaching the Extruded Heatsink 997
13.6.4.1 Clips 997
13.6.4.2 Thermal Interface (PCM45 Series) 997
13.6.4.3 Maximum Static Normal Load 998
13.6.5 Reliability 998
13.7 Thermal Interface Management for Heat-Sink Solutions 999
13.7.1 Bond Line Management 999
13.7.2 Interface Material Performance 999
13.7.2.1 Thermal Resistance of the Material 999
13.7.2.2 Wetting/Filling Characteristics of the Material 999
13.8 Measurements for Thermal Specifications 1000
13.8.1 Case Temperature Measurements 1000
13.8.1.1 Attaching the Thermocouple (No Heatsink) 1000
13.8.1.2 Attaching the Thermocouple (Heatsink) 1001
13.9 Thermal Diode 1002
13.10 Heatsink and Attach Suppliers 1002
13.11 PCB Guidelines 1002

14 Diagnostics 1003

14.1 JTAG Test Mode Description 1003

Appendix AChanges from 82580 1005

NOTE: ***This page intentionally left blank.***

1 Introduction

The Intel® Ethernet Controller I350 is a single, compact, low power component that supports quad port and dual port gigabit Ethernet designs. The device offers four fully-integrated gigabit Ethernet media access control (MAC), physical layer (PHY) ports and four SGMII/SerDes ports that can be connected to an external PHY. The I350 supports PCI Express* (PCIe v2.1 (2.5GT/s and 5GT/s)).

The device enables two-port or four port 1000BASE-T implementations using integrated PHY's. It can be used for server system configurations such as rack mounted or pedestal servers, in an add-on NIC or LAN on Motherboard (LOM) design. Another possible system configuration is for blade servers. Here, the I350 can support up to 4 SerDes ports as LOM or mezzanine card. It can also be used in embedded applications such as switch add-on cards and network appliances.

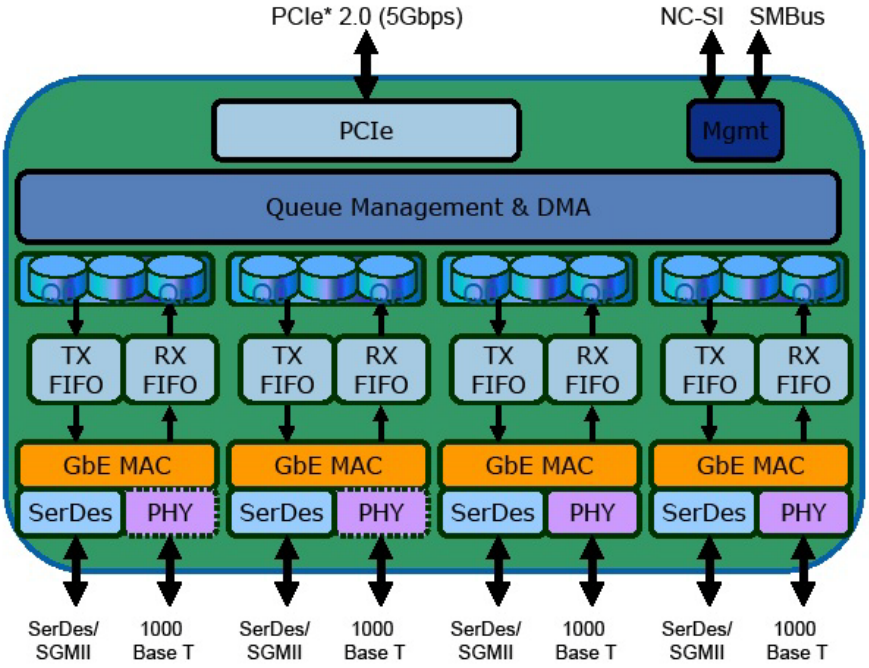


Figure 1-1 Intel® Ethernet Controller I350

1.1 Scope

This document provides the external architecture (including device operation, pin descriptions, register definitions, etc.) for the I350.

This document is a reference for software device driver developers, board designers, test engineers, and others who may need specific technical or programming information.

1.2 Terminology and Acronyms

Table 1-1 Glossary

Definition	Meaning
1000BASE-BX	1000BASE-BX is the PICMG 3.1 electrical specification for transmission of 1 Gb/s Ethernet or 1 Gb/s fibre channel encoded data over the backplane.
1000BASE-KX	1000BASE-KX is the IEEE802.3ap electrical specification for transmission of 1 Gb/s Ethernet over the backplane.
1000BASE-CX	1000BASE-X over specialty shielded 150 Ω balanced copper jumper cable assemblies as specified in IEEE 802.3 Clause 39.
1000BASE-T	1000BASE-T is the specification for 1 Gb/s Ethernet over category 5e twisted pair cables as defined in IEEE 802.3 clause 40.
AEN	Asynchronous Event Notification
b/w	Bandwidth.
BIOS	Basic Input/Output System.
BMC	Baseboard Management Controller (often used interchangeably with MC).
BT	Bit Time.
CRC	Cyclic redundancy check
DCA	Direct Cache Access.
DFT	Design for Testability.
DQ	Descriptor Queue.
DMTF	Distributed Management Task Force standard body.
DW	Double word (4 bytes).
EEE	Energy Efficient Ethernet - IEEE802.3az standard
EEPROM	Electrically Erasable Programmable Memory. A non-volatile memory located on the LAN controller that is directly accessible from the host.
EOP	End of Packet.
FC	Flow Control.
FCS	Frame Check Sequence.
Firmware (FW)	Embedded code on the LAN controller that is responsible for the implementation of the NC-SI protocol and pass through functionality.
Host Interface	RAM on the LAN controller that is shared between the firmware and the host. RAM is used to pass commands from the host to firmware and responses from the firmware to the host.
HPC	High - Performance Computing.
IPC	Inter Processor Communication.
IPG	Inter Packet Gap.
IPMI	Intelligent Platform Management Interface specification

Table 1-1 Glossary (Continued)

Definition	Meaning
LAN (auxiliary Power-Up)	The event of connecting the LAN controller to a power source (occurs even before system power-up).
LLDP	Link Layer Discovery Protocol defined in IEEE802.1AB used by IEEE802.3az (EEE) for system wake time negotiation.
LOM	LAN on Motherboard.
LPI	Low Power Idle - Low power state of Ethernet link as defined in IEEE802.3az.
LSO	Large Send Offload.
LTR	Latency Tolerance Reporting (PCIe protocol)
LVR	Linear Voltage Regulator
MAC	Media Access Control.
MC	Management Controller
MCTP	DMTF Management Component Transport Protocol (MCTP) specification. A transport protocol to allow communication between a management controller and controlled device over various transports.
MDIO	Management Data Input/Output Interface over MDC/MDIO lines.
MIFS/MIPG	Minimum Inter Frame Spacing/Minimum Inter Packet Gap.
MMW	Maximum Memory Window.
MSS	Maximum Segment Size. Largest amount of data, in a packet (without headers) that can be transmitted. Specified in Bytes.
MPS	Maximum Payload Size in PCIe specification.
MTU	Maximum Transmit Unit. Largest packet size (headers and data) that can be transmitted. Specified in Bytes.
NC	Network Controller.
NC-SI	Network Controller Sideband Interface DMTF Specification
NIC	Network Interface Controller.
TPH	TLP Process Hints (PCIe protocol).
PCS	Physical Coding Sub layer.
PHY	Physical Layer Device.
PMA	Physical Medium Attachment.
PMD	Physical Medium Dependent.
RMII	Reduced Media Independent Interface (Reduced MII).
SA	Source Address.
SDP	Software Defined Pins.
SerDes	serializer/deserializer. A transceiver that converts parallel data to serial data and vice-versa.
SFD	Start Frame Delimiter.
SGMII	Serialized Gigabit Media Independent Interface.
SMBus	System Management Bus. A bus that carries various manageability components, including the LAN controller, BIOS, sensors and remote-control devices.
SVR	Switched Voltage Regulator
TCO	Total Cost of Ownership (TCO) System Management.
TLP	Transaction Layer Packet in the PCI Express specification.
TSO	Transmit Segmentation offload - A mode in which a large TCP/UDP I/O is handled to the device and the device segments it to L2 packets according to the requested MSS.
VLAN	Virtual LAN
VPD	Vital Product Data (PCI protocol).

1.2.1 External Specification and Documents

The I350 implements features from the following specifications.

1.2.1.1 Network Interface Documents

1. IEEE standard 802.3, 2006 Edition (Ethernet). Incorporates various IEEE Standards previously published separately. Institute of Electrical and Electronic Engineers (IEEE).
2. IEEE standard 1149.1, 2001 Edition (JTAG). Institute of Electrical and Electronics Engineers (IEEE)
3. IEEE Std 1149.6-2003, IEEE Standard for Boundary-Scan Testing of Advanced Digital Networks, IEEE, 2003.
4. IEEE standard 802.1Q for VLAN
5. PICMG3.1 Ethernet/Fibre Channel Over PICMG 3.0 Draft Specification, January 14, 2003, Version D1.0
6. Serial-GMII Specification, Cisco Systems document ENG-46158, Revision 1.7
7. INF-8074i Specification for SFP (Small Form factor Pluggable) Transceiver (ftp://ftp.seagate.com/sff)
8. IEEE Std 802.3ap-2007
9. IEEE 1588™ Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, November 8 2002
10. IEEE 802.1AS Timing and Synchronization for Time- Sensitive Applications in Bridged Local Area Networks Draft 2.0, February 22, 2008
11. IEEE 802.3az Energy Efficient Ethernet Draft 1.4, May 2009

1.2.1.2 Host Interface Documents

1. PCI-Express 2.1 Base specification
2. PCI Specification, version 3.0
3. PCI Bus Power Management Interface Specification, Rev. 1.2, March 2004
4. Advanced Configuration and Power Interface Specification, Rev 2.0b, October 2002
5. Single Root I/O Virtualization and Sharing Specification Revision 1.1 Draft, September 11, 2007

1.2.1.3 Networking Protocol Documents

1. IPv4 specification (RFC 791)
2. IPv6 specification (RFC 2460)
3. TCP/UDP specification (RFC 793/768)
4. SCTP specification (RFC 2960)
5. ARP specification (RFC 826)
6. Neighbor Discovery for IPv6 (RFC 4861)
7. Multicast Listener Discovery (MLD) for IPv6 (RFC 2710)
8. Multicast Listener Discovery Version 2 (MLDv2) for IPv6 (RFC 3810)
9. EUI-64 specification, http://standards.ieee.org/regauth/oui/tutorials/EUI64.html.

1.2.1.4 Manageability Documents

1. DMTF Network Controller Sideband Interface (NC-SI) Specification rev 1.0.0, May 2009
2. System Management Bus (SMBus) Specification, SBS Implementers Forum, Ver. 2.0, August 2000

1.3 Product Overview

The I350 supports 4 SerDes or SGMII ports for MAC to MAC blade server connections or MAC to external PHY connections. Alternatively, four internal 1000BASE-T PHYs can be used to implement a quad port NIC or LOM design.

1.4 External Interface

1.4.1 PCIe Interface

The PCIe v2.1 (5GT/s) Interface is used by the I350 as a host interface. The interface supports both PCIe v2.1 (2.5GT/s) and PCIe v2.1 (5GT/s) rates and can be configured to x4, x2 and x1. The maximum aggregated raw bandwidth for a typical x4 PCIe v2.1 (5GT/s) configuration is 16 Gb/s in each direction. Refer to Section 2.3.1 for a full pin description. The timing characteristics of this interface are defined in PCI Express Card Electromechanical Specification rev 2.0 and in the PCIe v2.1 (2.5GT/s and 5GT/s) specification.

1.4.2 Network Interfaces

Four independent interfaces are used to connect the four I350 ports to external devices. The following protocols are supported:

- MDI (Copper) support for standard IEEE 802.3 Ethernet interface for 1000BASE-T, 100BASE-TX, and 10BASE-T applications (802.3, 802.3u, and 802.3ab)
- SerDes interface to connect over a backplane to another SerDes compliant device or to an Optical module. The I350 supports both 1000BASE-BX and 1000BASE-KX (Without IEEE802.3ap Backplane Auto-Negotiation)
- SGMII interface to attach to an external PHY, either on board or via an SFP module. The SGMII interface shares the same pins as the SerDes

Refer to Section 2.3.6.2 and Section 2.3.6 for full pin description; Section 11.6.3 and Section for timing characteristics of this interface.

1.4.3 EEPROM Interface

The I350 uses an EEPROM device for storing product configuration information. Several words of the EEPROM are accessed automatically by the I350 after reset in order to provide pre-boot configuration data that must be available to the I350 before it is accessed by host software. The remainder of the stored information is accessed by various software modules used to report product configuration, serial number, etc.

The I350 is intended for use with a SPI (4-wire) serial EEPROM device such as an AT25040AN or compatible EEPROM device refer to Section 11.8.2 for full list of supported EEPROM devices. Refer to Section 2.3.2 for full pin description and Section 11.6.2.5 for timing characteristics of this interface.

The I350 also supports an EEPROM-less mode, where all the setup is done by software.

1.4.4 Serial Flash Interface

The I350 provides an external SPI serial interface to a Flash or Boot ROM device such as the Atmel* AT25F1024 or compatible Flash device. Refer to Section 11.8.1 for full list of supported Flash devices. The I350 supports serial Flash devices with up to 64 Mbit (8 MByte) of memory. The size of the Flash used by the I350 can be configured by the EEPROM. Refer to Section 2.3.2 for full pin description and Section 11.6.2.4 for timing characteristics of this interface.

Note: Though the I350 supports devices with up to 8 MB of memory, bigger devices can also be used. Accesses to memory beyond the Flash device size results in access wrapping as only the lower address bits are used by the Flash device.

1.4.5 SMBus Interface

SMBus is an optional interface for pass-through and/or configuration traffic between a BMC and the I350.

The I350's SMBus interface can be configured to support both slow and fast timing modes. Refer to Section 2.3.3 for full pin description and Section 11.6.2.2 for timing characteristics of this interface.

1.4.6 NC-SI Interface

NC-SI and SMBus interfaces are optional for pass-through and/or configuration traffic between a BMC and the I350. The NC-SI interface meets the DMTF NC-SI Specification, Rev. 1.0.0 as an integrated Network Controller (NC) device.

See Chapter 2.3.4 for full pin description and Chapter 11.6.2.6 for timing characteristics of this interface.

1.4.7 MDIO/I^2C 2-Wire Interfaces

The I350 implements four management Interfaces for control of an optional external PHY. Each interface can be either a 2-wire Standard-mode I^2C interface used to control an SFP module or an MII Management Interface (also known as the Management Data Input/Output or MDIO Interface) for control plane connection between the MAC and PHY devices (master side). This interface provides the MAC and software with the ability to monitor and control the state of the external PHY. The I350 supports the data formats defined in IEEE 802.3 clause 22.

The I350 supports shared MDIO operation and separate MDIO connection. When configured via the *MDICNFG* register to separate MDIO operation each MDIO interface should be connected to the relevant PHY. When configured via the *MDICNFG* register to shared MDIO operation the MDC/MDIO interface of

LAN port 0 can be shared by all ports to support connection to a multi-port PHY with a single MDC/ MDIO interface. Refer to Section 2.3.6 for a full pin description, Section 11.6.2.8 for MDIO timing characteristics and Section 11.6.2.3 for I^2C timing characteristics of this interface.

1.4.8 Software-Definable Pins (SDP) Interface (General-Purpose I/O)

The I350 has four software-defined pins (SDP pins) per port that can be used for IEEE1588 auxiliary device connections, control of the SFP optical module interface, passing thermal sensor limit indication and other miscellaneous hardware or software-control purposes. These pins can be individually configurable to act as either standard inputs, general-purpose interrupt (GPI) inputs or output pins. The default direction of each pin is configurable via the EEPROM (refer to Section 6.2.21, Section 8.2.1 and Section 8.2.3), as well as the default value of all pins configured as outputs. Information on SDP usage can be found in Section 3.4 and Section 7.9.4. Refer to Section 2.3.5 for pin description of this interface.

1.4.9 LEDs Interface

The I350 implements four output drivers per port intended for driving external LED circuits. Each of the four LED outputs can be individually configured to select the particular event, state, or activity, which is indicated on that output. In addition, each LED can be individually configured for output polarity as well as for blinking versus non-blinking (steady-state) indication.

The configuration for LED outputs is specified via the LEDCTL register. Furthermore, the hardware-default configuration for all LED outputs can be specified via EEPROM fields (refer to Section 6.2.18 and Section 6.2.20), thereby supporting LED displays configurable to a particular OEM preference.

Refer to Section 2.3.6.1 for full pin description of this interface.

Refer to Section 7.5 for more detailed description of LED behavior.

1.5 Features

Table 1-2 to Table 1-7 list the I350's features and compares them to other LAD products .

Table 1-2 I350 Network Features

Feature	I350	82580	82599	82576
Half duplex at 10/100 Mb/s operation and full duplex operation at all supported speeds	Y	Y	100 Mb/s full duplex	Y
10/100/1000 Copper PHY integrated on-chip	4 ports	4 ports	N	2 ports
Jumbo frames supported	Y	Y	Y	Y
Size of jumbo frames supported	9.5 KB	9.5 KB	16 KB	9.5 KB
Flow control support: send/receive PAUSE frames and receive FIFO thresholds	Y	Y	Y	Y
Statistics for management and RMON	Y	Y	Y	Y
802.1q VLAN support	Y	Y	Y	Y

Table 1-2 I350 Network Features (Continued)

Feature	I350	82580	82599	82576
802.3az EEE support	Y	N	N	N
MDI Flip	Y	N	N/A	Y
SerDes interface for external PHY connection or system interconnect	4 ports	4 ports	2 ports	2 ports
1000BASE-KX interface for Blade Server Backplane connections	Y	Y	Y	N
802.3ap Backplane Auto-negotiation	N	N	Y	N
SGMII interface for external 1000BASE-T PHY connection	4 ports	4 ports	2 ports	2 ports
Fiber/copper auto-sense	4 ports	4 ports	N/A	2 ports
SerDes support of non-Auto-Negotiation partner	Y	Y	Y	Y
SerDes signal detect	Y	Y	N	Y
External PHY control I/F MDC/MDIO 2 wire I/F	Shared or per function Per function	Shared or per function Per function	Per function Per function	Per function Per function

Table 1-3 I350 Host Interface Features

Feature	I350	82580	82599	82576
PCIe revision	2.1 (5 Gbps or 2.5 Gbps)	2.0 (5 Gbps or 2.5 Gbps)	2.0 (5 Gbps or 2.5 Gbps)	2.0 (2.5 Gbps)
PCIe physical layer	Gen 2	Gen 2	Gen 2	Gen 1
Bus width	x1, x2, x4	x1, x2, x4	x1, x4, x8	x1, x2, x4
64-bit address support for systems using more than 4 GB of physical memory	Y	Y	Y	Y
Outstanding requests for Tx buffers per port	24 Per port and for all ports	24 Per port and for all ports	16	4
Outstanding requests for Tx descriptors per port	4 Per port and for all ports	4 Per port and for all ports	8	1
Outstanding requests for Rx descriptors per port	4 Per port and for all ports	4 Per port and for all ports	4	1
Credits for posted writes	4	4	8	2
Max payload size supported	512 B	512 B	512 B	512 B
Max request size supported	2 KB	2 KB	2 KB	512 B
Link layer retry buffer size	3.2 KB	3.2 KB	3.2 KB	2 KB
Vital Product Data (VPD)	Y	Y	Y	Y
End to End CRC (ECRC)	Y	Y	Y	N
LTR (Latency Tolerance Reporting)	Y	Y	N	N
TPH	Y	Y	N	N
CSR access via Configuration space	Y	Y	N	N
ACS (Access Control Services)	Y	N	N	N

Table 1-4 I350 LAN Functions Features

Feature	I350	82580	82599	82576
Programmable host memory receive buffers	Y	Y	Y	Y
Descriptor ring management hardware for transmit and receive	Y	Y	Y	Y
Software controlled global reset bit (resets everything except the configuration registers)	Y	Y	Y	Y
Software Definable Pins (SDP) - per port	4	4	8	4
Four SDP pins can be configured as general purpose interrupts	Y	Y	Y	Y
Wake up	Y	Y	Y	Y
Flexible wake-up filters	8	8	6	6
Flexible filters for queue assignment in normal operation	8	8	N	N
IPv6 wake-up filters	Y	Y	Y	Y
Default configuration by the EEPROM for all LEDs for pre-driver functionality	4 LEDs	4 LEDs	4 LEDs	4 LEDs
LAN function disable capability	Y	Y	Y	Y
Programmable memory transmit buffers	Y	Y	Y	Y
Double VLAN	Y	Y	Y	Y
IEEE 1588	Y	Y	Y	Y
Per-Packet Timestamp	Y	Y	N	N
TX rate limiting per queue	N	N	Y	Y

Table 1-5 I350 LAN Performance Features

Feature	I350	82580	82599	82576
TCP segmentation offload Up to 256 KB	Y	Y	Y	Y
iSCSI TCP segmentation offload (CRC)	N	N	Y	N
IPv6 support for IP/TCP and IP/UDP receive checksum offload	Y	Y	Y	Y
Fragmented UDP checksum offload for packet reassembly	Y	Y	Y	Y
Message Signaled Interrupts (MSI)	Y	Y	Y	Y
Message Signaled Interrupts (MSI-X) number of vectors	25	10	256	25
Packet interrupt coalescing timers (packet timers) and absolute-delay interrupt timers for both transmit and receive operation	Y	Y	Y	Y
Interrupt throttling control to limit maximum interrupt rate and improve CPU utilization	Y	Y	Y	Y
Rx packet split header	Y	Y	Y	Y
Receive Side Scaling (RSS) number of queues per port	Up to 8	Up to 8	Up to 16	Up to 16
Total number of Rx queues per port	8	8	128	16
Total number of TX queues per port	8	8	128	16
RX header replication Low latency interrupt DCA support TCP timer interrupts No snoop Relax ordering	Yes to all	Yes to all	Yes to all	Yes to all
TSO interleaving for reduced latency	Y	Y	Y	Y

Table 1-5 I350 LAN Performance Features (Continued)

Feature	I350	82580	82599	82576
Receive side coalescing	N	N	Y	N
SCTP receive and transmit checksum offload	Y	Y	Y	Y
UDP TSO	Y	Y	Y	Y
IPSec offload	N	N	Y	Y

Table 1-6 I350 Virtualization Features

Feature	I350	82580	82599	82576
Support for Virtual Machines Device queues (VMDq) per port	8 pools (single queue)	8 pools (single queue)	16/32/64 pools	8 pools
L2 MAC address filters (unicast and multicast)	32	24	128	24
L2 VLAN filters	Per pool	Per pool	64	Per pool
PCI-SIG SR-IOV	8 VF	N	16/32/64 VF	8 VF
Multicast/Broadcast Packet replication	Y	Y on Receive	Y	Y
VM to VM Packet forwarding (Packet Loopback)	Y	N	Y	Y
RSS replication	N	N	Y	N
Traffic shaping	N	N	Y	Y
MAC and VLAN anti-spoofing	Y	N	Y	Y
Malicious driver detection	Y	Y	N	N
Per-pool statistics	Y	Y	Y	Y
Per-pool off loads	Y	Y	Y	Y
Per-pool jumbo support	Y	Y	Y	Y
Mirroring rules	4	4	4	4
External switch VEPA support	Y	Y	N	Y
External switch NIV (VNTAG) support	N	N	N	N
Promiscuous modes	VLAN, unicast multicast	Multicast	Multicast	Multicast

Table 1-7 I350 Manageability Features

Feature	I350	82580	82599	Kawela
Advanced pass-through-compatible management packet transmit/ receive support	Y	Y	Y	Y
Managed ports on SMBus interface to external BMC	4	4	2	2
Fail-over support over SMBus	N	N	Y	Y
Auto-ARP reply over SMBus	Y	N	Y	Y
NC-SI Interface to an External BMC	Y	Y	Y	Y
Standard DMTF NC-SI protocol support	Y	Y	Y	Y
DMTF MCTP protocol over SMBus	Y	N	N	N
NC-SI HW arbitration	Y	N	N	Y
OS to BMC traffic	Y	N	N	N
L2 address filters	2	2	4	2
VLAN L2 filters	8	8	8	8

Table 1-7 I350 Manageability Features (Continued)

Feature	I350	82580	82599	Kawela
EtherType filters	4	4	4	4
Flex L4 port filters	8	8	16	16
Flex TCO filters	1	1	4	4
L3 address filters (IPv4)	4	4	4	4
L3 address filters (IPv6)	4	4	4	4
Proxying	1 ARP Offload per PF 2 NS Offloads per PF 2 MLD Offloads per PF	N	N	N

Table 1-8 I350 power management Features

Feature	I350	82580	82599	Kawela
Magic packet wake-up enable with unique MAC address	Y	Y	Y	Y
ACPI register set and power down functionality supporting D0 and D3 states	Y	Y	Y	Y
Full wake-up support (APM and ACPI 2.0)	Y	Y	Y	Y
Smart power down at S0 no link and Sx no link	Y	Y	Y	Y
LAN disable functionality	Y	Y	Y	Y
EEE	Y	N	N	N
DMA coalescing	Y	N	N	N

1.6 I350 Packaging Options

The I350 is available in multiple packaging options:

1. 17x17 PBGA package (2 ports and 4 ports).
2. 25x25 PBGA package

Table 1-9 lists the differences between features supported by the 17x17 and 25x25 packages.

Table 1-9 I350 17x17 and 25x25 Package Feature

Feature	17x17 Package	25x25 Package
Number of SerDes ports	2 Ports and 4 Ports	SerDes and SGMII not supported.
Number of Copper ports	2 Ports and 4 Ports	2 Ports (port 0 and 1).
Integrated SVR and LVR control	Supported	Not supported

1.7 Overview of Changes Compared to the 82580

The following section describes modifications done in I350 compared to the 82580.

1.7.1 Network Interface

1.7.1.1 Energy Efficient Ethernet (IEEE802.3AZ)

The I350 supports negotiation and link transition to low power Idle (LPI) state as defined in the IEEE802.3az (EEE) standard. EEE is supported for the following technologies:

- 1000BASE-T
- 100BASE-TX

Energy Efficient Ethernet enables reduction of I350 power consumption as a function of link utilization. In addition the I350 enables overall system power reduction as a function of link utilization by reporting increased latency tolerance values via PCIe LTR messages when link is in Low Power Idle state. For more information, refer to Section 3.7.7.

1.7.1.2 MDI Flip

To simplify on-board routing of MDI signals it is sometimes beneficial to be able to swap between MDI Lanes:

- A <-> D.
- B <-> C.

The I350 supports the MDI Flip option in the internal 1000BASE-T PHY. For more information. refer to Section 8.26.1.

1.7.2 Virtualization

The virtualization feature set implemented in the I350 is equivalent to the virtualization feature set supported by the 82576 with the following changes.

1.7.2.1 PCI SR IOV

The I350 supports the PCI-SIG Single-Root I/O Virtualization and Sharing specification (SR-IOV) Rev 1.1.

- Support for up to 8 virtual functions (VFs).
- Partial replication of PCI configuration space.

For information, refer to Section 7.8.2.

1.7.2.2 Promiscuous VLAN Filtering

The I350 supports promiscuous VLAN filtering per queue. For information, refer to Section 8.14.17.

1.7.2.3 Improvements to VMDq Switching

1.7.2.3.1 Promiscuous Modes

The I350 adds support for VLAN and unicast promiscuous modes. Refer to Section 7.8.3.4 for details.

1.7.2.3.2 Microsoft NLB Mode Support

NLB is a mode defined for Microsoft* Windows Server Operating System where a unicast address behaves as a multicast address in that it is used by multiple machines. In order to support this mode, it should be possible to forward part of the unicast MAC addresses to the network the same way we do for multicast addresses. In order to support this mode, the *RAH.TRMCST* bit is added. This bit is used to decide if packets are forwarded to the network even if they are forwarded to local addresses. Refer to Section 7.8.3.4 for details.

1.7.2.4 Number of Exact Match Filters

The number of *RAH*/*RAL* registers was expanded to 32.

1.7.2.5 Support for 2K Header Buffer

The I350 supports Rx header buffers of up to 2Kbytes as opposed to 960 bytes in previous products. Refer to Section 7.1.3.1 for details.

1.7.2.6 Support for Port Based VLAN

Previous products support port based VLAN by enabling enforcement of the VLAN insertion policy via hardware. To complete this capability, the I350 improves removal of the VLAN tag from received packet, so that the receiving VM is not aware of the VLAN network it belongs to. Refer to Section 7.8.3.8.1 for details.

1.7.2.7 Header Split on L2 Header

Added support for Header Split on L2 header using bit *PSRTYPE.PSR_type0*.

1.7.2.8 Updated Pool Decision Algorithm

The I350 includes an improved Pool decision queuing algorithm. Refer to Section 7.8.3 for details.

1.7.2.9 Statistics

A counter to count dropped packet per Tx queue (*TQDPC*) was added.

1.7.3 HOST Interface

1.7.3.1 MSI-X Support

The number of MSI-X vectors supported per function by the I350 changed to 25. When not working in an SR-IOV environment, the number of MSI-X vectors allocated to the PF (Physical function) is 10. When working in a SR-IOV environment, the number of MSI-X vectors per port allocated to the VFs (Virtual Functions) is 24 (3 vectors per VF) and an additional MSI-X vector is allocated to the PF. For further information, refer to Section 7.3.

1.7.3.2 ID-Based Ordering

ID-Based Ordering provides opportunity-independent-request-streams to bypass another congested stream, yielding a performance improvement. The new ordering attribute relaxes ordering requirements between unrelated traffic by comparing the Requester/Completer IDs of the associated TLPs. The I350 supports the new ordering ID-Based Ordering (IDO) attribute bit in the TLP header and the relevant configuration bits. For information, refer to Section 9.5.6.12.

1.7.3.3 Alternative Routing-ID Interpretation (ARI)

To allow more than eight functions per end point without requesting an internal switch, as usually needed in virtualization scenarios, the I350 supports the PCI-SIG defined ARI capability structure.This capability enables interpretation of the *Device* and *Function* fields as a single identification of a function within the bus. In addition, a new structure used to support the IOV capabilities reporting and control is defined. For further information, refer to Section 9.6.3.

Note: Since the OS will sometimes decide on ARI and other PCIe feature support based on the functionality reported in function 0, ARI support and IOV capabilities are reported also when a function is disabled and replaced by a dummy function.

1.7.3.4 Link State Related Latency Tolerance Reporting (LTR)

The I350 supports report of increased Latency Tolerance values as a function of Link state. When link is in EEE Low Power Idle state, the I350 will send an updated PCIe LTR message with increased latency tolerance values. For information, refer to Section 5.9 and Section 9.6.6.

1.7.3.5 Access Control Services (ACS)

The I350 supports ACS Extended Capability structures on all functions. The I350 reports no support for the various ACS capabilities in the ACS Extended Capability structure. For information, refer to Section 9.6.7.

1.7.3.6 ASPM Optionality Compliance Capability

A new capability bit, ASPM (Active State Power Management) Optionality Compliance bit has been added to the I350. Software is permitted to use the bit to help determine whether to enable ASPM or whether to run ASPM compliance tests. New bit indicates that the I350 can optionally support entry to L0s. For information, refer to Section 9.5.6.7.

1.7.4 Manageability

1.7.4.1 Auto-ARP Reply on SMBus

The I350 can be programmed for auto-ARP reply on reception of ARP request packets and supports sending of gratuitous ARP packets to reduce the traffic over the SMBus BMC interconnect. For information, refer to Section 10.5.4.

1.7.4.2 NC-SI Commands.

Support for the NC-SI Get Controller Packet Statistics command and additional OS to BMC and BMC to OS OEM commands were added in the I350.

Support for filtering related NC-SI commands and NC-SI flow control were also added. Refer to Section 10.6.2 for supported NC-SI commands.

1.7.4.2.1 NC-SI Hardware Arbitration

The I350 supports NC-SI HW arbitration between different Network Controller packages.

1.7.4.3 OS to BMC Traffic

In previous network controller chips traffic from OS to BMC or BMC to OS needed to pass through an external switch if a dedicated port was allocated for manageability or through a dedicated interface such as an IPMI KCS interface. The I350 supports transmission and reception of traffic internally via the regular pass-through interface used to communicate between the OS and local BMC, without need to utilize a dedicated interface or pass the traffic through an external switch. For information, refer to Section 10.4.

1.7.4.4 DMTF MCTP Protocol Over SMBus

The I350 enables reporting and controlling all information exposed in a LOM device via NC-SI using the MCTP protocol over SMBus. The MCTP interface will be used by the BMC to only control the NIC and not for pass through traffic. All network ports are mapped to a single MCTP endpoint on SMBus. For information, refer to Section 10.7.

1.7.4.5 Proxying

When system is in low power S3 or S4 state and the I350 is in D3 low power state, the I350 supports Host Protocol Offload as required for Win7 compliance of the following protocols:

1. IPv4 ARP - Single IPv4 address.
2. IPv6 Neighbor Solicitation (NS) - Four IPv6 addresses.
3. When NS protocol offload is enabled IPv6 Multicast Listener Discovery (MLD) - Two MLD Multicast-Address-Specific Queries and also supports response to MLD General Queries.

For further information, refer to Section 5.7.

1.7.5 EEPROM Structures

Management related EEPROM structures and other EEPROM words were updated. For further information see Chapter 6.

1.7.6 Recovery from Memory Error

The I350 supports recovery from memory error using per port software reset and does not require initiation of a full device reset to recover from a memory error condition. The I350 includes ECC protection on memories to verify data integrity. For further information see Section 7.6.

1.7.7 BOM Cost Reduction

1.7.7.1 On-Chip 1.8V LVR Control

The I350 includes an on-chip Linear Voltage regulator (LVR) control circuit. Together with an external low cost BJT transistor, circuit can be used to generate a 1.8V power supply without need for a higher cost on-board 1.8V voltage regulator (refer to Section 3.5).

1.7.7.2 On-Chip 1.0V SVR Control

The I350 includes an on-chip Switched Voltage Regulator (SVR) control circuit. Together with external matched P/N MOS power transistors and a LC filter the SVR can be used to generate a 1.0V power supply without need for a higher cost on-board 1.0V voltage regulator (refer to Section 3.5).

1.7.7.3 Thermal Sensor

The I350 implements autonomous on-die thermal management to monitor on-die temperature and react when the temperature exceeds a pre-defined threshold. Thermal management policies and thresholds are loaded from the EEPROM for flexibility. The I350 provides an interface to external devices to read its status through the management sideband interfaces.

Using the on die Thermal Sensor the I350 can be programmed to indicate that device temperature has passed one of 3 thermal trip points by:

- Asserting a SDP pin.
- Sending an interrupt.
- Issuing an Alert to the external BMC.

In addition the I350 can be programmed to reduce link speed if one of the Thermal Trip points has been passed. For further information, refer to Chapter 3.

1.8 Device Data Flows

1.8.1 Transmit Data Flow

Table 1-10 provides a high level description of all data/control transformation steps needed for sending Ethernet packets to the line.

Table 1-10 Transmit Data Flow

Step	Description
1	The host creates a descriptor ring and configures one of I350's transmit queues with the address location, length, head and tail pointers of the ring (one of 8 available Tx queues).
2	The host is requested by the TCP/IP stack to transmit a packet, it gets the packet data within one or more data buffers.
3	The host initializes descriptor(s) that point to the data buffer(s) and have additional control parameters that describe the needed hardware functionality. The host places that descriptor in the correct location at the appropriate Tx ring.
4	The host updates the appropriate queue tail pointer (TDT)
5	The I350's DMA senses a change of a specific TDT and as a result sends a PCIe request to fetch the descriptor(s) from host memory.
6	The descriptor(s) content is received in a PCIe read completion and is written to the appropriate location in the descriptor queue internal cache.
7	The DMA fetches the next descriptor from the internal cache and processes its content. As a result, the DMA sends PCIe requests to fetch the packet data from system memory.
8	The packet data is received from PCIe completions and passes through the transmit DMA that performs all programmed data manipulations (various CPU off loading tasks as checksum off load, TSO off load, etc.) on the packet data on the fly.
9	While the packet is passing through the DMA, it is stored into the transmit FIFO. After the entire packet is stored in the transmit FIFO, it is forwarded to the transmit switch module.
10	If the packet destination is also local, it is sent also to the local switch memory and join the receive path.
11	The transmit switch arbitrates between host and management packets and eventually forwards the packet to the MAC.
12	The MAC appends the L2 CRC to the packet and sends the packet to the line using a pre-configured interface.
13	When all the PCIe completions for a given packet are done, the DMA updates the appropriate descriptor(s).
14	After enough descriptors are gathered for write back or the interrupt moderation timer expires, the descriptors are written back to host memory using PCIe posted writes. Alternatively, the head pointer can only be written back.
15	After the interrupt moderation timer expires, an interrupt is generated to notify the host device driver that the specific packet has been read to the I350 and the driver can release the buffers.

1.8.2 Receive Data Flow

Table 1-11 provides a high level description of all data/control transformation steps needed for receiving Ethernet packets.

Table 1-11 Receive Data Flow

Step	Description
1	The host creates a descriptor ring and configures one of the I350's receive queues with the address location, length, head, and tail pointers of the ring (one of 8 available Rx queues).
2	The host initializes descriptors that point to empty data buffers. The host places these descriptors in the correct location at the appropriate Rx ring.
3	The host updates the appropriate queue tail pointer (RDT).
4	The I350's DMA senses a change of a specific RDT and as a result sends a PCIe request to fetch the descriptors from host memory.
5	The descriptors content is received in a PCIe read completion and is written to the appropriate location in the descriptor queue internal cache.
6	A packet enters the Rx MAC. The RX MAC checks the CRC of the packet.
7	The MAC forwards the packet to an Rx filter
8	If the packet matches the pre-programmed criteria of the Rx filtering, it is forwarded to the Rx FIFO. VLAN and CRC are optionally stripped from the packet and L3/L4 checksum are checked and the destination queue is fixed.
9	The receive DMA fetches the next descriptor from the internal cache of the appropriate queue to be used for the next received packet.
10	After the entire packet is placed into the Rx FIFO, the receive DMA posts the packet data to the location indicated by the descriptor through the PCIe interface. If the packet size is greater than the buffer size, more descriptors are fetched and their buffers are used for the received packet.
11	When the packet is placed into host memory, the receive DMA updates all the descriptor(s) that were used by packet data.
12	After enough descriptors are gathered for write back or the interrupt moderation timer expires or the packet requires immediate forwarding, the receive DMA writes back the descriptor content along with status bits that indicate the packet information including what off loads were done on that packet.
13	After the interrupt moderation timer completes or an immediate packet is received, the I350 initiates an interrupt to the host to indicate that a new received packet is already in host memory.
14	Host reads the packet data and sends it to the TCP/IP stack for further processing. The host releases the associated buffers and descriptors once they are no longer in use.

§ §

2 Pin Interface

2.1 Signal Type Notation

Table 2-1 defines I350 signal types.

Table 2-1 Signal Type Definition

Type	Description	DC specification
In	LVTTL input-only signal.	See section 11.6.1.1
Out	LVTTL Output active driver.	See section 11.6.1.1 and Section 11.6.1.2
T/S	LVTTL bi-directional, tri-state input/output pin.	See section 11.6.1.1
O/D	Open Drain allows multiple devices to share line using wired-OR configuration.	See section 11.6.1.3
NC-SI-in	NC-SI compliant input signal	See section 11.6.1.4
NC-SI-out	NC-SI compliant output signal	See section 11.6.1.4
A	Analog signals	See section 11.6.3 and Section 11.6.4
A-in	Analog input signals	See section 11.6.3 and Section 11.6.4
A-out	Analog output signals	See section 11.6.3 and Section 11.6.4
B	Input bias	See section 11.6.6 and Section 11.6.7
PS	Power Supply	

2.2 17x17 PBGA Package Pin Assignment

The I350 is packaged in a 17x17 PBGA package with 1.0 mm ball pitch.

2.2.1 PCIe

The AC specification for these pins is described in Section 11.6.2.10.

Table 2-2 PCIe Pins

	Symbol	Ball #	Type	Name and Function
	PE_CLK_p PE_CLK_n	A16 A15	A-in	PCIe Differential Reference Clock in: A 100MHz differential clock input. This clock is used as the reference clock for the PCIe Tx/Rx circuitry and by the PCIe core PLL to generate clocks for the PCIe core logic.

Table 2-2 PCIe Pins (Continued)

	Symbol	Ball #	Type	Name and Function
	PET_0_p PET_0_n	A12 B12	A-out	PCIe Serial Data output Lane 0: A serial differential output pair running at a bit rate of 2.5Gb/s or 5Gb/s.
	PET_1_p PET_1_n	A11 B11	A-out	PCIe Serial Data output Lane 1: A serial differential output pair running at a bit rate of 2.5Gb/s or 5Gb/s.
	PET_2_p PET_2_n	A6 B6	A-out	PCIe Serial Data output Lane 2: A serial differential output pair running at a bit rate of 2.5 Gb/s or 5Gb/s.
	PET_3_p PET_3_n	A5 B5	A-out	PCIe Serial Data output Lane 3: A serial differential output pair running at a bit rate of 2.5Gb/s or 5Gb/s.
	PER_0_p PER_0_n	A14 B14	A-in	PCIe Serial Data input Lane 0: A Serial differential input pair running at a bit rate of 2.5Gb/s or 5Gb/s.
	PER_1_p PER_1_n	A9 B9	A-in	PCIe Serial Data input Lane 1: A Serial differential input pair running at a bit rate of 2.5Gb/s or 5Gb/s.
	PER_2_p PER_2_n	A8 B8	A-in	PCIe Serial Data input Lane 2: A Serial differential input pair running at a bit rate of 2.5Gb/s or 5Gb/s.
	PER_3_p PER_3_n	A3 B3	A-in	PCIe Serial Data input Lane 3: A Serial differential input pair running at a bit rate of 2.5Gb/s or 5Gb/s.
	PE_WAKE_N	D16	O/D	WAKE#: Active low signal pulled to '0' to indicate that a Power Management Event (PME) is pending and the PCI Express link should be restored. Defined in the PCI Express CEM specification.
	PE_RST_N	B1	In	PERST#: Active low PCI Express fundamental reset input. When pulled to '0' resets chip and when de-asserted (set to '1') indicates that power and PCI Express reference clock are within specified values. Defined in the PCI Express specification. On exit from reset all registers and state machines are set to their initialization values.
	PE_TXVTERM1 PE_TXVTERM3 PE_TXVTERM4	C6 C9 C11	A-in	Should be connected to 1.8V power supply for termination
	PE_TRIM1 PE_TRIM2	A2 A1	A	PCIe Trimming A 1.5KΩ 1% resistor connected between these pins.

2.2.2 Flash and EEPROM Ports

The AC specification for these pins is described in Section 11.6.2.4 to Section 11.6.2.5.

Table 2-3 Flash and EEPROM Ports Pins

	Symbol	Ball #	Type	Name and Function
	FLSH_SI	B15	T/S	Serial Data output to the Flash
	FLSH_SO	C15	In	Serial Data input from the Flash
	FLSH_SCK	B16	T/S	Flash serial clock Operates at 15.625MHz.
	FLSH_CE_N	C16	T/S	Flash chip select Output
	EE_DI	E15	T/S	Data output to EEPROM
	EE_DO	F15	In	Data input from EEPROM
	EE_SK	E16	T/S	EEPROM serial clock output Operates at ~2 MHz.
	EE_CS_N	F16	T/S	EEPROM chip select Output

2.2.3 System Management Bus (SMB) Interface

The AC specification for these pins is described in Section 11.6.2.2.

Table 2-4 SMB Interface Pins

	Symbol	Ball #	Type	Name and Function
	SMBD	G3	T/S, O/D	SMB Data. Stable during the high period of the clock (unless it is a start or stop condition).
	SMBCLK	E5	T/S, O/D	SMB Clock. One clock pulse is generated for each data bit transferred.
	SMBALRT_N	G4	T/S, O/D	SMB Alert: acts as an Interrupt pin of a slave device on the SMB

2.2.4 NC-SI Interface Pins

The AC specification for these pins is described in Section 11.6.2.6.

Table 2-5 NC-SI Interface Pins

	Symbol	Ball #	Type	Name and Function
	NCSI_CLK_IN	H1	NC-SI-In	NC-SI Reference Clock Input – Synchronous clock reference for receive, transmit and control interface. It is a 50MHz clock +/- 100 ppm. Note: When the I350 drives the NC-SI clock NCSI_CLK_IN should be connected to NCSI_CLK_OUT pin on-board.
	NCSI_CLK_OUT	H2	NC-SI-Out	NC-SI Reference Clock Output – Synchronous clock reference for receive, transmit and control interface. It is a 50MHz clock +/- 100 ppm. Serves as a clock source to the BMC and the I350 (when configured so).
	NCSI_CRS_DV	H3	NC-SI-Out	CRS/DV – Carrier Sense / Receive Data Valid.
	NCSI_RXD_1 NCSI_RXD_0	J1 H4	NC-SI-Out	Receive data signals from the I350 to BMC.
	NCSI_TX_EN	J2	NC-SI-In	Transmit Enable.
	NCSI_TXD_1 NCSI_TXD_0	J4 J3	NC-SI-In	Transmit data signals from BMC to the I350.
	NCSI_ARB_IN	C13	NC-SI-In	NC-SI HW arbitration token output pin
	NCSI_ARB_OUT	D12	NC-SI-Out	NC-SI HW arbitration token input pin

2.2.5 Miscellaneous Pins

The AC specification for the XTAL pins is described in Section 11.6.5.

Table 2-6 Miscellaneous Pins

	Symbol	Ball #	Type	Name and Function
	SDP0_0 SDP0_1 SDP0_2 SDP0_3	K1 K2 K3 K4	T/S	SW Defined Pins for port 0: These pins are reserved pins that are software programmable write/read input/output capability. These default to inputs upon power up, but may have their direction and output values defined in the EEPROM. The SDP bits may be mapped to the General Purpose Interrupt bits when configured as inputs. 1. The SDP0_0 pin can be used as a watchdog output indication. 2. The SDP0_0 and SDP1_0 pins can be used to define the NC-SI Package ID (refer to Section 10.2.2.2). 3. All the SDP pins can be used as SFP sideband signals (TxDisable, present and TxFault). The I350 does not use these signals; it is available for SW control over SFP. 4. The SDP0_1 pin can be used as a strapping option to disable PCIe Function 0. In this case it is latched at the rising edge of PE_RST# or In-Band PCIe Reset (refer to Section 4.4.4).
	SDP1_0 SDP1_1 SDP1_2 SDP1_3	L1 L2 L3 L4	T/S	SW Defined Pins for port 1: Reserved pins that are software programmable write/read input/output capability. These default to inputs upon power up, but may have their direction and output values defined in the EEPROM. The SDP bits may be mapped to the General Purpose Interrupt bits when configured as inputs. 1. The SDP1_0 pin can be used as a watchdog output indication. 2. The SDP0_0 and SDP1_0 pins can be used to define the NC-SI Package ID (refer to Section 10.2.2.2). 3. All the SDP pins can be used as SFP sideband signals (TxDisable, present and TxFault). The I350 does not use these signals; it is available for SW control over SFP. 4. The SDP1_1 pin can be used as a strapping option to disable PCIe Function 1. In this case it is latched at the rising edge of PE_RST# or In-Band PCIe Reset (refer to Section 4.4.4).
	SDP2_0 SDP2_1 SDP2_2 SDP2_3	M1 M2 M3 M4	T/S	SW Defined Pins for port 2: These pins are reserved pins that are software programmable write/read input/output capability. These default to inputs upon power up, but may have their direction and output values defined in the EEPROM. The SDP bits may be mapped to the General Purpose Interrupt bits when configured as inputs. 1. The SDP2_0 pin can be used as a watchdog output indication. 2. All the SDP pins can be used as SFP sideband signals (TxDisable, present and TxFault). The I350 does not use these signals; it is available for SW control over SFP. 3. The SDP2_1 pin can be used as a strapping option to disable PCIe Function 2. In this case it is latched at the rising edge of PE_RST# or In-Band PCIe Reset (refer to Section 4.4.4).

Table 2-6 Miscellaneous Pins (Continued)

	Symbol	Ball #	Type	Name and Function
	SDP3_0 SDP3_1 SDP3_2 SDP3_3	N1 N2 N3 N4	T/S	SW Defined Pins for port 3: These pins are reserved pins that are software programmable write/read input/output capability. These default to inputs upon power up, but may have their direction and output values defined in the EEPROM. The SDP bits may be mapped to the General Purpose Interrupt bits when configured as inputs. 1. The SDP3_0 pin can be used as a watchdog output indication. 2. All the SDP pins can be used as SFP sideband signals (TxDisable, present and TxFault). The I350 does not use these signals; it is available for SW control over SFP. 3. The SDP3_1 pin can be used as a strapping option to disable PCIe Function 3. In this case it is latched at the rising edge of PE_RST# or In-Band PCIe Reset (refer to Section 4.4.4).
	LAN_PWR_GOOD	D4	In	LAN Power Good: A 3.3v input signal. A transition from low to high initializes the device into operation. If the internal Power-on-Reset circuit is used to trigger device power-up, this signal should be connected to VCC3P3.
	MAIN_PWR_OK	B2	In	Main Power OK – Indicates that platform main power is up. Must be connected externally to main core 3.3V power.
	DEV_OFF_N	C4	In	Device Off: Assertion of DEV_OFF_N puts the device in Device Disable mode. This pin is asynchronous and is sampled once the EEPROM is ready to be read following power-up. The DEV_OFF_N pin should always be connected to VCC3P3 to enable device operation.
	XTAL1 XTAL2	P1 P2	A-In A-out	Reference Clock / XTAL: These pins may be driven by an external 25MHz crystal or driven by a single ended external CMOS compliant 25MHz oscillator.
	TSENSP	R2	A-out	Thermal Diode output; Can be used to measure the I350 on-die temperature.
	TSENSZ	T1	GND	Thermal Diode Ground.

2.2.6 SERDES/SGMII Pins

The AC specification for these pins is described in Section 11.6.3.

Table 2-7 SERDES/SGMII Pins

	Symbol	Ball #	Type	Name and Function
	SER0_p SER0_n	P16 P15	A-in	SERDES/SGMII Serial Data input Port 0: Differential SERDES Receive interface. A Serial differential input pair running at 1.25Gb/s. An embedded clock present in this input is recovered along with the data.
	SET0_p SET0_n	R16 R15	A-out	SERDES/SGMII Serial Data output Port 0: Differential SERDES Transmit interface. A serial differential output pair running at 1.25Gb/s. This output carries both data and an embedded 1.25GHz clock that is recovered along with data at the receiving end.

Table 2-7 SERDES/SGMII Pins (Continued)

	Symbol	Ball #	Type	Name and Function
	SRDS_0_SIG_DET	N13	In	Port 0 Signal Detect: Indicates that signal (light) is detected from the Fiber. High for signal detect, Low otherwise. Polarity of Signal Detect pin is controlled by *CTRL.ILOS* bit. For non-fiber serdes applications link indication is internal, *CONNSW.ENRGSRC* bit should be 0b and pin should be connected to a pull-up resistor.
	SER1_p SER1_n	M16 M15	A-in	SERDES/SGMII Serial Data input Port 1: Differential fiber SERDES Receive interface. A Serial differential input pair running at 1.25Gb/s. An embedded clock present in this input is recovered along with the data.
	SET1_p SET1_n	N16 N15	A-out	SERDES/SGMII Serial Data output Port 1: Differential fiber SERDES Transmit interface. A serial differential output pair running at 1.25Gb/s. This output carries both data and an embedded 1.25GHz clock that is recovered along with data at the receiving end.
	SRDS_1_SIG_DET	P14	In	Port 1 Signal Detect: Indicates that signal (light) is detected from the fiber. High for signal detect, Low otherwise. Polarity of Signal Detect pin is controlled by *CTRL.ILOS* bit. For non-fiber serdes applications link indication is internal, *CONNSW.ENRGSRC* bit should be 0b and pin should be connected to a pull-up resistor.
	SER2_p SER2_n	K16 K15	A-in	SERDES/SGMII Serial Data input Port 2: Differential SERDES Receive interface. A Serial differential input pair running at 1.25Gb/s. An embedded clock present in this input is recovered along with the data.
	SET2_p SET2_n	L16 L15	A-out	SERDES/SGMII Serial Data output Port 2: Differential SERDES Transmit interface. A serial differential output pair running at 1.25Gb/s. This output carries both data and an embedded 1.25GHz clock that is recovered along with data at the receiving end.
	SRDS_2_SIG_DET	T15	In	Port 2 Signal Detect: Indicates that signal (light) is detected from the Fiber. High for signal detect, Low otherwise. Polarity of Signal Detect pin is controlled by *CTRL.ILOS* bit. For non-fiber serdes applications link indication is internal, *CONNSW.ENRGSRC* bit should be 0b and pin should be connected to a pull-up resistor.
	SER3_p SER3_n	H16 H15	A-in	SERDES/SGMII Serial Data input Port 3: Differential SERDES Receive interface. A Serial differential input pair running at 1.25Gb/s. An embedded clock present in this input is recovered along with the data.
	SET3_p SET3_n	J16 J15	A-out	SERDES/SGMII Serial Data output Port 3: Differential SERDES Transmit interface. A serial differential output pair running at 1.25Gb/s. This output carries both data and an embedded 1.25GHz clock that is recovered along with data at the receiving end.

Table 2-7 SERDES/SGMII Pins (Continued)

	Symbol	Ball #	Type	Name and Function
	SRDS_3_SIG_DET	T16	In	Port 3 Signal Detect: Indicates that signal (light) is detected from the Fiber. High for signal detect, Low otherwise. Polarity of Signal Detect pin is controlled by *CTRL.ILOS* bit. For non-fiber serdes applications link indication is internal, *CONNSW.ENRGSRC* bit should be 0b and pin should be connected to a pull-up resistor.
	SE_RSET	K13	B	SerDes Bias Connect 2.37KΩ 1% resistor between pin and ground.

2.2.7 SFP Pins

The AC specification for these pins is described in Section 11.6.2.9.

Table 2-8 SFP Pins

	Symbol	Ball #	Type	Name and Function
	SFP0_I2C_CLK	M13	Out, O/D	Port 0 SFP 2 wire interface clock – connects to Mod-Def1 input of SFP (O/D). Can also be used as MDC pin (Out).
	SFP0_I2C_DATA	J13	T/S, O/D	Port 0 SFP 2 wire interface data – connects to Mod-Def2 pin of SFP (O/D). Can also be used as MDIO pin (T/S).
	SFP1_I2C_CLK	M14	Out, O/D	Port 1 SFP 2 wire interface clock – connects to Mod-Def1 input of SFP (O/D). Can also be used as MDC pin (Out).
	SFP1_I2C_DATA	N14	T/S, O/D	Port 1 SFP 2 wire interface data – connects to Mod-Def2 pin of SFP (O/D). Can also be used as MDIO pin (T/S).
	SFP2_I2C_CLK	J14	Out, O/D	Port 2 SFP 2 wire interface clock – connects to Mod-Def1 input of SFP (O/D). Can also be used as MDC pin (Out).
	SFP2_I2C_DATA	H13	T/S, O/D	Port 2 SFP 2 wire interface data – connects to Mod-Def2 pin of SFP (O/D). Can also be used as MDIO pin (T/S).
	SFP3_I2C_CLK	H14	Out, O/D	Port 3 SFP 2 wire interface clock – connects to Mod-Def1 input of SFP (O/D). Can also be used as MDC pin (Out).
	SFP3_I2C_DATA	G16	T/S, O/D	Port 3 SFP 2 wire interface data – connects to Mod-Def2 pin of SFP (O/D). Can also be used as MDIO pin (T/S).

2.2.8 PHY Pins

2.2.8.1 LED's

The table below describes the functionality of the LED output pins. Default activity of the LED may be modified in the EEPROM word offsets 1Ch and 1Fh from start of relevant LAN Port section. The LED functionality is reflected and can be further modified in the configuration registers LEDCTL.

Table 2-9 LED Output Pins

	Symbol	Ball #	Type	Name and Function
	LED0_0	C1	Out	Port 0 LED0. Programmable LED which indicates by default Link Up. ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.20) or *LEDCTL* register (refer to Section 8.2.9).
	LED0_1	C2	Out	Port 0 LED1. Programmable LED which indicates by default activity (when packets are transmitted or received that match MAC filtering). ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.18) or *LEDCTL* register (refer to Section 8.2.9).
	LED0_2	C3	Out	Port 0 LED2. Programmable LED which indicates by default a 100Mbps Link. ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.20) or *LEDCTL* register (refer to Section 8.2.9).
	LED0_3	E4	Out	Port 0 LED3. Programmable LED which indicates by default a 1000Mbps Link. ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.18) or *LEDCTL* register (refer to Section 8.2.9).
	LED1_0	D1	Out	Port 1 LED0. Programmable LED which indicates by default Link up. ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.20) or *LEDCTL* register (refer to Section 8.2.9).
	LED1_1	D2	Out	Port 1 LED1. Programmable LED which indicates by default activity (when packets are transmitted or received that match MAC filtering). ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.18) or *LEDCTL* register (refer to Section 8.2.9).
	LED1_2	D3	Out	Port 1 LED2. Programmable LED which indicates by default a 100Mbps Link. ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.20) or *LEDCTL* register (refer to Section 8.2.9).
	LED1_3	F4	Out	Port 1 LED3. Programmable LED which indicates by default a 1000Mbps Link. ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.18) or *LEDCTL* register (refer to Section 8.2.9).
	LED2_0	E1	Out	Port 2 LED0. Programmable LED which indicates by default Link up. ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.20) or *LEDCTL* register (refer to Section 8.2.9).

Table 2-9 LED Output Pins (Continued)

	Symbol	Ball #	Type	Name and Function
	LED2_1	E2	Out	Port 2 LED1. Programmable LED which indicates by default activity (when packets are transmitted or received that match MAC filtering). ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.18) or *LEDCTL* register (refer to Section 8.2.9).
	LED2_2	E3	Out	Port 2 LED2. Programmable LED which indicates by default a 100Mbps Link. ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.20) or *LEDCTL* register (refer to Section 8.2.9).
	LED2_3	G2	Out	Port 2 LED3. Programmable LED which indicates by default a 1000Mbps Link. ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.18) or *LEDCTL* register (refer to Section 8.2.9).
	LED3_0	F1	Out	Port 3 LED0. Programmable LED which indicates by default Link up. ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.20) or *LEDCTL* register (refer to Section 8.2.9).
	LED3_1	F2	Out	Port 3 LED1. Programmable LED which indicates by default activity (when packets are transmitted or received that match MAC filtering). ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.18) or *LEDCTL* register (refer to Section 8.2.9).
	LED3_2	F3	Out	Port 3 LED2. Programmable LED which indicates by default a 100Mbps Link. ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.20) or *LEDCTL* register (refer to Section 8.2.9).
	LED3_3	G1	Out	Port 3 LED3. Programmable LED which indicates by default a 1000Mbps Link. ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.18) or *LEDCTL* register (refer to Section 8.2.9).

2.2.8.2 PHY Analog Pins

The AC specification for these pins is described in Section 11.6.4.

Table 2-10 Analog Pins

	Symbol	Ball #	Type	Name and Function
	MDI0_0_p MDI0_0_n MDI1_0_p MDI1_0_n MDI2_0_p MDI2_0_n MDI3_0_p MDI3_0_n	T3 R3 T6 R6 T9 R9 T12 R12	A	Media Dependent Interface[0] for port 0, port 1, Port 2 and port 3 accordingly: 1000BASE-T: In MDI configuration, MDI[0]+/- corresponds to BI_DA+/- and in MDIX configuration MDI[0]+/- corresponds to BI_DB+/-. 100BASE-TX: In MDI configuration, MDI[0]+/- is used for the transmit pair and in MDIX configuration MDI[0]+/- is used for the receive pair. 10BASE-T: In MDI configuration, MDI[0]+/- is used for the transmit pair and in MDIX configuration MDI[0]+/- is used for the receive pair. ***Note:*** When *IPCNFG.MDI_Flip* register bit is set to 1b MDI[0]+/- and MDI[3]+/- are swapped.

Table 2-10 Analog Pins (Continued)

	Symbol	Ball #	Type	Name and Function
	MDI0_1_p MDI0_1_n MDI1_1_p MDI1_1_n MDI2_1_p MDI2_1_n MDI3_1_p MDI3_1_n	P4 P5 P6 P7 T10 R10 T13 R13	A	Media Dependent Interface[1] for port 0, port 1, port 2 and port 3 accordingly: 1000BASE-T: In MDI configuration, MDI[1]+/- corresponds to BI_DB+/- and in MDIX configuration MDI[1]+/- corresponds to BI_DA+/-. 100BASE-TX: In MDI configuration, MDI[1]+/- is used for the receive pair and in MDIX configuration MDI[1]+/- is used for the transmit pair. 10BASE-T: In MDI configuration, MDI[1]+/- is used for the receive pair and in MDIX configuration MDI[1]+/- is used for the transmit pair. ***Note:*** When *IPCNFG.MDI_Flip* register bit is set to 1b MDI[1]+/- and MDI[2]+/- are swapped.
	MDI0_2_p MDI0_2_n MDI1_2_p MDI1_2_n MDI2_2_p MDI2_2_n MDI3_2_p MDI3_2_n	T4 R4 T7 R7 T11 R11 T14 R14	A	Media Dependent Interface[2] for port 0, port 1 port 2 and port 3: 1000BASE-T: In MDI configuration, MDI[2]+/- corresponds to BI_DC+/- and in MDIX configuration MDI[2]+/- corresponds to BI_DD+/-. 100BASE-TX: Unused. 10BASE-T: Unused. ***Note:*** When *IPCNFG.MDI_Flip* register bit is set to 1b MDI[1]+/- and MDI[2]+/- are swapped.
	MDI0_3_p MDI0_3_n MDI1_3_p MDI1_3_n MDI2_3_p MDI2_3_n MDI3_3_p MDI3_3_n	T5 R5 T8 R8 P9 P10 P12 P13	A	Media Dependent Interface[3] for port 0, port 1, port 2 and port 3: 1000BASE-T: In MDI configuration, MDI[3]+/- corresponds to BI_DD+/- and in MDIX configuration MDI[3]+/- corresponds to BI_DC+/-. 100BASE-TX: Unused. 10BASE-T: Unused. ***Note:*** When *IPCNFG.MDI_Flip* register bit is set to 1b MDI[0]+/- and MDI[3]+/- are swapped.
	GE_REXT3K	T2	B	PHY Bias Connect 3.01KΩ 1% resistor between this pin and ground.
	RSVD_TX_TCLK	R1	Out	Transmit 125MHz clock for IEEE testing. Shared for the 4 ports. Not connected in normal operation

2.2.9 Voltage Regulator Pins

The electrical specifications for the SVR and LVR is described in Section 11.6.9.

Table 2-11 Voltage Regulator Pins

	Symbol	Ball #	Type	Name and Function
	VR_EN	G14	T/S	LVR 1.8V and SVR 1.0V enable In case pin is driven low or left floating it indicates that Internal 1.8V LVR Control circuit and internal 1.0V SVR Control circuit are disabled and the 1.0V and 1.8V power supplies are driven externally.
	SVR_HDRV	C5	A-out	Internal 1.0V SVR PFET gate drive Driver output for high-side switch Connected to external PFET power transistor. ***Note:*** When 1.0V SVR is disabled and the I350 is placed in a 82580 socket, ball is unconnected.
	SVR_LDRV	D5	A-out	Internal 1.0V SVR NFET gate drive Driver output for low-side switch Connected to external NFET power transistor. ***Note:*** When 1.0V SVR is disabled and the I350 is placed in a 82580 socket, ball is unconnected.
	SVR_SW	G7	A-In	Internal 1.0V SVR Control Voltage switch sense input. ***Note:*** When 1.0V SVR is disabled and the I350 is placed in a 82580 socket, ball is connected to VSS,
	SVR_FB	G5	A-In	Internal 1.0V SVR Control Feedback input. ***Note:*** When 1.0V SVR is disabled and the I350 is placed in a 82580 socket, ball is connected to VSS.
	SVR_COMP	F6	A-out	Internal 1.0V SVR Control Compensation output. ***Note:*** When 1.0V SVR is disabled and the I350 is placed in a 82580 socket, ball is unconnected.
	LVR_1P8_CTRL	C8	A-out	Internal 1.8V LVR Control output, connected to external BJT transistor. ***Note:*** When 1.8V LVR is disabled and the I350 is placed in a 82580 socket, ball is connected to PE_TXVTERM2 (1.8V power supply) or is unconnected,

2.2.10 Testability Pins

Table 2-12 Testability Pins

	Symbol	Ball #	Type	Name and Function
	RSVD_TE_VSS	C12	In	Enables test mode. When high test pins are multiplexed on functional signals. In functional mode, must be connected to ground.
	JTCK	F13	In	JTAG Clock Input
	JTDI	E12	In	JTAG TDI Input
	JTDO	D13	T/S, O/D	JTAG TDO Output
	JTMS	G13	In	JTAG TMS Input
	RSRVD_JRST_3P3	E13	In	JTAG Reset Input

Table 2-12 Testability Pins (Continued)

	Symbol	Ball #	Type	Name and Function
	AUX_PWR	D15	T/S	Auxiliary Power Available: This pin is a strapping option pin, latched at the rising edge of PE_RST# or In-Band PCIe Reset. This pin has an internal weak pull-up resistor. In case this pin is driven high during init time it indicates that Auxiliary Power is available and the device should support D3cold power state if enabled to do so. This pin is also used for testing and scan.
	LAN0_DIS_N	F14	T/S	This pin is a strapping option pin, latched at the rising edge of PE_RST# or In-Band PCIe Reset. In case this pin is asserted during init time, LAN 0 function is disabled. This pin is also used for testing and scan. Refer to Section 4.4.3 and Section 4.4.4 for additional information.
	LAN1_DIS_N	E14	T/S	This pin is a strapping option pin, latched at the rising edge of PE_RST# or In-Band PCIe Reset. In case this pin is asserted during init time, LAN 1 function is disabled. This pin is also used for testing and scan. Refer to Section 4.4.3 and Section 4.4.4 for additional information.
	LAN2_DIS_N	D14	T/S	This pin is a strapping option pin, latched at the rising edge of PE_RST# or In-Band PCIe Reset. In case this pin is asserted during init time, LAN 2 function is disabled. Refer to Section 4.4.3 and Section 4.4.4 for additional information.
	LAN3_DIS_N	C14	T/S	This pin is a strapping option pin, latched at the rising edge of PE_RST# or In-Band PCIe Reset. In case this pin is asserted during init time, LAN 3 function is disabled. This pin is also used for testing and scan. Refer to Section 4.4.3 and Section 4.4.4 for additional information.
	RSVD_JTP8	G15	In	Test pin for production testing. In functional mode should not be connected.

2.2.11 Power Supply and Ground Pins

Table 2-13 Power Supply Pins

	Symbol	Ball #	Type	Name and Function
	VCC3P3	K5	3.3V	3.3V Periphery power supply
	VCC3P3	F5, H5	3.3V	3.3V Periphery power supply
	VCC3P3	F12, H12, K12, L12	3.3V	3.3V Periphery power supply
	VCC1P0	E6,G6, H6, J6,E11, G11, H11, J11, K11, M11, N12	1.0V	1.0V digital power supply
	VCC1P0	K6	1.0V	1.0V digital power supply
	VCC1P0_APE	D6, D8, D9, D11	1.0V	1.0V PCIe Analog Power Supply
	VCC1P0_ASE	L13, K14, L14	1.0V	1.0V SerDes Analog power supply
	VCC1P0_AGE	L7, L8, L9, L10	1.0V	1.0V PHY analog power supply
	VCC3P3_A	M6, M7, M8, M9, M10, P8, P11	3.3V	3.3V PHY analog power supply
	VCC3P3_AGE	L5	3.3V	3.3V PHY analog power supply
	VCC1P8_PE_1	C7	1.8V	PCIe VCO Analog power supply connected to 1.8V.
	VCC1P8_PE_2	C10	1.8V	PCIe VCO Analog power supply connected to 1.8V.

Signal	Pin
VSS	A4, A7, A10, A13, B4, B7, B10, B13, D7, D10, E7, E8, E9, E10, F7, F8, F9, F10, F11, G8, G9, G10, G12, H7, H8, H9, H10, J5, J7, J8, J9, J10, J12, K7, K8, K9, K10, L6, L11, M5, M12, N5, N6, N7, N8, N9, N10, N11, P3

2.2.12 4-Port 17x17 PBGA Package Pin List (Alphabetical)

Table 2-14 lists the pins and signals in ball alphabetical order.

Table 2-14 17x17 PBGA Package Pin List in Alphabetical Order

Signal	Ball	Signal	Ball	Signal	Ball
PE_TRIM2	A1	SVR_HDRV	C5	VSS	E9
PE_TRIM1	A2	PE_TXVTERM1	C6	VSS	E10
PER_3_p	A3	VCC1P8_PE_1	C7	VCC1P0	E11
VSS	A4	LVR_1P8_CTRL	C8	JTDI	E12
PET_3_p	A5	PE_TXVTERM3	C9	RSVD_JRST_3P3	E13
PET_2_p	A6	VCC1P8_PE_2	C10	LAN1_DIS_N	E14
VSS	A7	PE_TXVTERM4	C11	EE_DI	E15

Table 2-14 17x17 PBGA Package Pin List in Alphabetical Order (Continued)

Signal	Ball	Signal	Ball	Signal	Ball
PER_2_p	A8	RSVD_TE_VSS	C12	EE_SK	E16
PER_1_p	A9	NCSI_ARB_IN	C13	LED3_0	F1
VSS	A10	LAN3_DIS_N	C14	LED3_1	F2
PET_1_p	A11	FLSH_SO	C15	LED3_2	F3
PET_0_p	A12	FLSH_CE_N	C16	LED1_3	F4
VSS	A13	LED1_0	D1	VCC3P3	F5
PER_0_p	A14	LED1_1	D2	SVR_COMP	F6
PE_CLK_n	A15	LED1_2	D3	VSS	F7
PE_CLK_p	A16	LAN_PWR_GOOD	D4	VSS	F8
PE_RST_N	B1	SVR_LDRV	D5	VSS	F9
MAIN_PWR_OK	B2	VCC1P0_APE	D6	VSS	F10
PER_3_n	B3	VSS	D7	VSS	F11
VSS	B4	VCC1P0_APE	D8	VCC3P3	F12
PET_3_n	B5	VCC1P0_APE	D9	JTCK	F13
PET_2_n	B6	VSS	D10	LAN0_DIS_N	F14
VSS	B7	VCC1P0_APE	D11	EE_DO	F15
PER_2_n	B8	NCSI_ARB_OUT	D12	EE_CS_N	F16
PER_1_n	B9	JTDO	D13	LED3_3	G1
VSS	B10	LAN2_DIS_N	D14	LED2_3	G2
PET_1_n	B11	AUX_PWR	D15	SMBD	G3
PET_0_n	B12	PE_WAKE_N	D16	SMBALRT_N	G4
VSS	B13	LED2_0	E1	SVR_FB	G5
PER_0_n	B14	LED2_1	E2	VCC1P0	G6
FLSH_SI	B15	LED2_2	E3	SVR_SW	G7
FLSH_SCK	B16	LED0_3	E4	VSS	G8
LED0_0	C1	SMBCLK	E5	VSS	G9
LED0_1	C2	VCC1P0	E6	VSS	G10
LED0_2	C3	VSS	E7	VCC1P0	G11
DEV_OFF_N	C4	VSS	E8	VSS	G12
JTMS	G13	VCC1P0	K6	SER1_n	M15
VR_EN	G14	VSS	K7	SER1_p	M16
RSVD_JTP8	G15	VSS	K8	SDP3_0	N1
SFP3_I2C_DATA	G16	VSS	K9	SDP3_1	N2
NCSI_CLK_IN	H1	VSS	K10	SDP3_2	N3
NCSI_CLK_OUT	H2	VCC1P0	K11	SDP3_3	N4
NCSI_CRS_DV	H3	VCC3P3	K12	VSS	N5
NCSI_RXD_0	H4	SE_RSET	K13	VSS	N6
VCC3P3	H5	VCC1P0_ASE	K14	VSS	N7
VCC1P0	H6	SER2_n	K15	VSS	N8
VSS	H7	SER2_p	K16	VSS	N9
VSS	H8	SDP1_0	L1	VSS	N10

Table 2-14 17x17 PBGA Package Pin List in Alphabetical Order (Continued)

Signal	Ball	Signal	Ball	Signal	Ball
VSS	H9	SDP1_1	L2	VSS	N11
VSS	H10	SDP1_2	L3	VCC1P0	N12
VCC1P0	H11	SDP1_3	L4	SRDS_0_SIG_DET	N13
VCC3P3	H12	VCC3P3_AGE	L5	SFP1_I2C_DATA	N14
SFP2_I2C_DATA	H13	VSS	L6	SET1_n	N15
SFP3_I2C_CLK	H14	VCC1P0_AGE	L7	SET1_p	N16
SER3_n	H15	VCC1P0_AGE	L8	XTAL_CLK_I	P1
SER3_p	H16	VCC1P0_AGE	L9	XTAL_CLK_O	P2
NCSI_RXD_1	J1	VCC1P0_AGE	L10	VSS	P3
NCSI_TX_EN	J2	VSS	L11	MDI0_1_p	P4
NCSI_TXD_0	J3	VCC3P3	L12	MDI0_1_n	P5
NCSI_TXD_1	J4	VCC1P0_ASE	L13	MDI1_1_p	P6
VSS	J5	VCC1P0_ASE	L14	MDI1_1_n	P7
VCC1P0	J6	SET2_n	L15	VCC3P3_A	P8
VSS	J7	SET2_p	L16	MDI2_3_p	P9
VSS	J8	SDP2_0	M1	MDI2_3_n	P10
VSS	J9	SDP2_1	M2	VCC3P3_A	P11
VSS	J10	SDP2_2	M3	MDI3_3_p	P12
VCC1P0	J11	SDP2_3	M4	MDI3_3_n	P13
VSS	J12	VSS	M5	SRDS_1_SIG_DET	P14
SFP0_I2C_DATA	J13	VCC3P3_A	M6	SER0_n	P15
SFP2_I2C_CLK	J14	VCC3P3_A	M7	SER0_p	P16
SET3_n	J15	VCC3P3_A	M8	RSVD_TX_TCLK	R1
SET3_p	J16	VCC3P3_A	M9	TSENSP	R2
SDP0_0	K1	VCC3P3_A	M10	MDI0_0_n	R3
SDP0_1	K2	VCC1P0	M11	MDI0_2_n	R4
SDP0_2	K3	VSS	M12	MDI0_3_n	R5
SDP0_3	K4	SFP0_I2C_CLK	M13	MDI1_0_n	R6
VCC3P3	K5	SFP1_I2C_CLK	M14	MDI1_2_n	R7
MDI1_3_n	R8	TSENSZ	T1	MDI2_1_p	T10
MDI2_0_n	R9	GE_REXT3K	T2	MDI2_2_p	T11
MDI2_1_n	R10	MDI0_0_p	T3	MDI3_0_p	T12
MDI2_2_n	R11	MDI0_2_p	T4	MDI3_1_p	T13
MDI3_0_n	R12	MDI0_3_p	T5	MDI3_2_p	T14
MDI3_1_n	R13	MDI1_0_p	T6	SRDS_2_SIG_DET	T15
MDI3_2_n	R14	MDI1_2_p	T7	SRDS_3_SIG_DET	T16
SET0_n	R15	MDI1_3_p	T8		
SET0_p	R16	MDI2_0_p	T9		

2.2.13 2-Port 17x17 PBGA Package Pin List (Alphabetical)

Note: 2-port and 4-port 17x17 PBGA packages share the same NVM_EEPROM images. In the case of the 2 port device the additional EEPROM bits are ignored although the checksums must still be valid.

Table 2-15 2-Port 17x17 PBGA Package Pin List (Alphabetical)

Signal	Ball	Signal	Ball	Signal	Ball
PE_TRIM2	A1	SVR_HDRV	C5	VSS	E9
PE_TRIM1	A2	PE_TXVTERM1	C6	VSS	E10
PER_3_p	A3	VCC1P8_PE_1	C7	VCC1P0	E11
VSS	A4	LVR_1P8_CTRL	C8	JTDI	E12
PET_3_p	A5	PE_TXVTERM3	C9	RSVD_JRST_3P3	E13
PET_2_p	A6	VCC1P8_PE_2	C10	LAN1_DIS_N	E14
VSS	A7	PE_TXVTERM4	C11	EE_DI	E15
PER_2_p	A8	RSVD_TE_VSS	C12	EE_SK	E16
PER_1_p	A9	NCSI_ARB_IN	C13	N/C	F1
VSS	A10	N/C	C14	N/C	F2
PET_1_p	A11	FLSH_SO	C15	N/C	F3
PET_0_p	A12	FLSH_CE_N	C16	LED1_3	F4
VSS	A13	LED1_0	D1	VCC3P3	F5
PER_0_p	A14	LED1_1	D2	SVR_COMP	F6
PE_CLK_n	A15	LED1_2	D3	VSS	F7
PE_CLK_p	A16	LAN_PWR_GOOD	D4	VSS	F8
PE_RST_N	B1	SVR_LDRV	D5	VSS	F9
MAIN_PWR_OK	B2	VCC1P0_APE	D6	VSS	F10
PER_3_n	B3	VSS	D7	VSS	F11
VSS	B4	VCC1P0_APE	D8	VCC3P3	F12
PET_3_n	B5	VCC1P0_APE	D9	JTCK	F13
PET_2_n	B6	VSS	D10	LAN0_DIS_N	F14
VSS	B7	VCC1P0_APE	D11	EE_DO	F15
PER_2_n	B8	NCSI_ARB_OUT	D12	EE_CS_N	F16
PER_1_n	B9	JTDO	D13	N/C	G1
VSS	B10	N/C	D14	N/C	G2
PET_1_n	B11	AUX_PWR	D15	SMBD	G3
PET_0_n	B12	PE_WAKE_N	D16	SMBALRT_N	G4
VSS	B13	N/C	E1	SVR_FB	G5
PER_0_n	B14	N/C	E2	VCC1P0	G6
FLSH_SI	B15	N/C	E3	SVR_SW	G7
FLSH_SCK	B16	LED0_3	E4	VSS	G8
LED0_0	C1	SMBCLK	E5	VSS	G9
LED0_1	C2	VCC1P0	E6	VSS	G10
LED0_2	C3	VSS	E7	VCC1P0	G11

Table 2-15 2-Port 17x17 PBGA Package Pin List (Alphabetical)

DEV_OFF_N	C4	VSS	E8	VSS	G12
JTMS	G13	VCC1P0	K6	SER1_n	M15
VR_EN	G14	VSS	K7	SER1_p	M16
RSVD_JTP8	G15	VSS	K8	SDP3_0	N/C
N/C	G16	VSS	K9	SDP3_1	N/C
NCSI_CLK_IN	H1	VSS	K10	SDP3_2	N/C
NCSI_CLK_OUT	H2	VCC1P0	K11	SDP3_3	N/C
NCSI_CRS_DV	H3	VCC3P3	K12	VSS	N5
NCSI_RXD_0	H4	SE_RSET	K13	VSS	N6
VCC3P3	H5	VCC1P0_ASE	K14	VSS	N7
VCC1P0	H6	N/C	K15	VSS	N8
VSS	H7	N/C	K16	VSS	N9
VSS	H8	SDP1_0	L1	VSS	N10
VSS	H9	SDP1_1	L2	VSS	N11
VSS	H10	SDP1_2	L3	VCC1P0	N12
VCC1P0	H11	SDP1_3	L4	SRDS_0_SIG_DET	N13
VCC3P3	H12	VCC3P3_AGE	L5	SFP1_I2C_DATA	N14
N/C	H13	VSS	L6	SET1_n	N15
N/C	H14	VCC1P0_AGE	L7	SET1_p	N16
N/C	H15	VCC1P0_AGE	L8	XTAL_CLK_I	P1
N/C	H16	VCC1P0_AGE	L9	XTAL_CLK_O	P2
NCSI_RXD_1	J1	VCC1P0_AGE	L10	VSS	P3
NCSI_TX_EN	J2	VSS	L11	MDI0_1_p	P4
NCSI_TXD_0	J3	VCC3P3	L12	MDI0_1_n	P5
NCSI_TXD_1	J4	VCC1P0_ASE	L13	MDI1_1_p	P6
VSS	J5	VCC1P0_ASE	L14	MDI1_1_n	P7
VCC1P0	J6	N/C	L15	VCC3P3_A	P8
VSS	J7	N/C	L16	N/C	P9
VSS	J8	SDP2_0	N/C	N/C	P10
VSS	J9	SDP2_1	N/C	VCC3P3_A	P11
VSS	J10	SDP2_2	N/C	N/C	P12
VCC1P0	J11	SDP2_3	N/C	N/C	P13
VSS	J12	VSS	M5	SRDS_1_SIG_DET	P14
SFP0_I2C_DATA	J13	VCC3P3_A	M6	SER0_n	P15
N/C	J14	VCC3P3_A	M7	SER0_p	P16
N/C	J15	VCC3P3_A	M8	RSVD_TX_TCLK	R1
N/C	J16	VCC3P3_A	M9	TSENSP	R2
SDP0_0	K1	VCC3P3_A	M10	MDI0_0_n	R3
SDP0_1	K2	VCC1P0	M11	MDI0_2_n	R4
SDP0_2	K3	VSS	M12	MDI0_3_n	R5
SDP0_3	K4	SFP0_I2C_CLK	M13	MDI1_0_n	R6
VCC3P3	K5	SFP1_I2C_CLK	M14	MDI1_2_n	R7
MDI1_3_n	R8	TSENSZ	T1	N/C	T10
N/C	R9	GE_REXT3K	T2	N/C	T11
N/C	R10	MDI0_0_p	T3	N/C	T12

Table 2-15 2-Port 17x17 PBGA Package Pin List (Alphabetical)

N/C	R11	MDI0_2_p	T4	N/C	T13
N/C	R12	MDI0_3_p	T5	N/C	T14
N/C	R13	MDI1_0_p	T6	N/C	T15
N/C	R14	MDI1_2_p	T7	N/C	T16
SET0_n	R15	MDI1_3_p	T8		
SET0_p	R16	N/C	T9		

2.2.14 2-Port 17x17 PBGA Package No-Connect Pins

Table 2-16 2-Port No-Connect Pins

	Ball
No-Connect	C14, D14, E1, E2, E3, F1, F2, F3, G1, G2, G16, H13, H14, H15, H16, J14, J15, J16, K15, K16, L15, L16, P9, P10, P12, P13, R9, R10, R11, R12, R13, R14, T9, T10, T11, T12, T13, T14, T15, T16

2.3 25x25 PBGA Package Pin Assignment

The I350 is packaged in a 25x25 PBGA package with 1.0 mm ball pitch.

Following tables describe functionality of the various balls.

2.3.1 PCIe

The AC specification for these pins is described in Section 11.6.2.10.

Table 2-17 PCIe Pins

	Symbol	Ball #	Type	Name and Function
	PE_CLK_p PE_CLK_n	Y2 Y1	A-in	PCIe Differential Reference Clock in: A 100MHz differential clock input. This clock is used as the reference clock for the PCIe Tx/Rx circuitry and by the PCIe core PLL to generate clocks for the PCIe core logic.
	PET_0_p PET_0_n	AC3 AD3	A-out	PCIe Serial Data output Lane 0: A serial differential output pair running at a bit rate of 2.5Gb/s or 5Gb/s.
	PET_1_p PET_1_n	AC4 AD4	A-out	PCIe Serial Data output Lane 1: A serial differential output pair running at a bit rate of 2.5Gb/s or 5Gb/s.
	PET_2_p PET_2_n	AC9 AD9	A-out	PCIe Serial Data output Lane 2: A serial differential output pair running at a bit rate of 2.5 Gb/s or 5Gb/s.
	PET_3_p PET_3_n	AC10 AD10	A-out	PCIe Serial Data output Lane 3: A serial differential output pair running at a bit rate of 2.5Gb/s or 5Gb/s.
	PER_0_p PER_0_n	AB2 AB1	A-in	PCIe Serial Data input Lane 0: A Serial differential input pair running at a bit rate of 2.5Gb/s or 5Gb/s.
	PER_1_p PER_1_n	AD6 AC6	A-in	PCIe Serial Data input Lane 1: A Serial differential input pair running at a bit rate of 2.5Gb/s or 5Gb/s.
	PER_2_p PER_2_n	AD7 AC7	A-in	PCIe Serial Data input Lane 2: A Serial differential input pair running at a bit rate of 2.5Gb/s or 5Gb/s.

Table 2-17 PCIe Pins (Continued)

	Symbol	Ball #	Type	Name and Function
	PER_3_p PER_3_n	AD12 AC12	A-in	PCIe Serial Data input Lane 3: A Serial differential input pair running at a bit rate of 2.5Gb/s or 5Gb/s.
	PE_WAKE_N	W1	O/D	WAKE#: Active low signal pulled to '0' to indicate that a Power Management Event (PME) is pending and the PCI Express link should be restored. Defined in the PCI Express CEM specification.
	PE_RST_N	W2	In	PERST#: Active low PCI Express fundamental reset input. When pulled to '0' resets chip and when de-asserted (set to '1') indicates that power and PCI Express reference clock are within specified values. Defined in the PCI Express specification. On exit from reset all registers and state machines are set to their initialization values.
	PE_TRIM1 PE_TRIM2	V1 V2	A	PCIe Trimming A 1.5KΩ 1% resistor connected between these pins.

2.3.2 Flash and EEPROM Ports

The AC specification for these pins is described in Section 11.6.2.4 to Section 11.6.2.5.

Table 2-18 Flash and EEPROM Ports Pins

	Symbol	Ball #	Type	Name and Function
	FLSH_SI	K2	T/S	Serial Data output to the Flash
	FLSH_SO	K1	In	Serial Data input from the Flash
	FLSH_SCK	J1	T/S	Flash serial clock Operates at 15.625MHz.
	FLSH_CE_N	J2	T/S	Flash chip select Output
	EE_DI	R1	T/S	Data output to EEPROM
	EE_DO	T1	In	Data input from EEPROM
	EE_SK	N2	T/S	EEPROM serial clock output Operates at ~2 MHz.
	EE_CS_N	N3	T/S	EEPROM chip select Output

2.3.3 System Management Bus (SMB) Interface

The AC specification for these pins is described in Section 11.6.2.2.

Table 2-19 SMB Interface Pins

	Symbol	Ball #	Type	Name and Function
	SMBD	L1	O/D	SMB Data. Stable during the high period of the clock (unless it is a start or stop condition).
	SMBCLK	L2	O/D	SMB Clock. One clock pulse is generated for each data bit transferred.
	SMBALRT_N	M2	O/D	SMB Alert: acts as an Interrupt pin of a slave device on the SMB

2.3.4 NC-SI Interface Pins

The AC specification for these pins is described in Section 11.6.2.6.

Table 2-20 NC-SI Interface Pins

	Symbol	Ball #	Type	Name and Function
	NCSI_CLK_IN	G2	NC-SI-In	NC-SI Reference Clock Input – Synchronous clock reference for receive, transmit and control interface. It is a 50MHz clock /- 50 ppm. Note: When the I350 drives the NC-SI clock NCSI_CLK_IN should be connected to NCSI_CLK_OUT pin on-board.
	NCSI_CLK_OUT	L3	NC-SI-Out	NC-SI Reference Clock Output – Synchronous clock reference for receive, transmit and control interface. It is a 50MHz clock /- 50 ppm. Serves as a clock source to the BMC and the I350 (when configured so).
	NCSI_CRS_DV	H1	NC-SI-Out	CRS/DV – Carrier Sense / Receive Data Valid.
	NCSI_RXD_1 NCSI_RXD_0	G1 H3	NC-SI-Out	Receive data signals from the I350 to BMC.
	NCSI_TX_EN	G4	NC-SI-In	Transmit Enable.
	NCSI_TXD_1 NCSI_TXD_0	G3 H2	NC-SI-In	Transmit data signals from BMC to the I350.
	NCSI_ARB_OUT	F2	NC-SI-Out	NC-SI HW arbitration token output pin.
	NCSI_ARB_IN	F1	NC-SI-In	NC-SI HW arbitration token input pin.

Note: If NC-SI is disconnected: (1) an external pull-down should be used for the NCSI_CLK_IN and NCSI_TX_EN pins; a pull-down (10 K Ω) should be used for NCSI_TXD[1:0].

2.3.5 Miscellaneous Pins

The AC specification for the XTAL pins is described in Section 11.6.5.

Table 2-21 Miscellaneous Pins

	Symbol	Ball #	Type	Name and Function
	SDP0_0 SDP0_1 SDP0_2 SDP0_3	R4 P3 T4 R3	T/S	SW Defined Pins for port 0: These pins are reserved pins that are software programmable write/read input/output capability. These default to inputs upon power up, but may have their direction and output values defined in the EEPROM. The SDP bits may be mapped to the General Purpose Interrupt bits when configured as inputs. 1. The SDP0_0 pin can be used as a watchdog output indication. 2. All the SDP pins can be used as SFP sideband signals (TxDisable, present and TxFault). The I350 does not use these signals; it is available for SW control over SFP. 3. The SDP0_1 pin can be used as a strapping option to disable PCIe Function 0. In this case it is latched at the rising edge of PE_RST# or In-Band PCIe Reset (refer to Section 4.4.4). ***Note:***

Table 2-21 Miscellaneous Pins (Continued)

	Symbol	Ball #	Type	Name and Function
	SDP1_0 SDP1_1 SDP1_2 SDP1_3	T21 T22 U21 U22	T/S	SW Defined Pins for port 1: Reserved pins that are software programmable write/read input/output capability. These default to inputs upon power up, but may have their direction and output values defined in the EEPROM. 1. The SDP1_0 pin can be used as a watchdog output indication. 2. All the SDP pins can be used as SFP sideband signals (TxDisable, present and TxFault). The I350 does not use these signals; it is available for SW control over SFP. 3. The SDP1_1 pin can be used as a strapping option to disable PCIe Function 1. In this case it is latched at the rising edge of PE_RST# or In-Band PCIe Reset (refer to Section 4.4.4).
	TSENSP	F4	A-out	Thermal Diode output; Can be used to measure the I350 on-die temperature.
	TSENSZ	F3	GND	Thermal Diode Ground.
	LAN_PWR_GOOD	L24	In	LAN Power Good: A 3.3v input signal. A transition from low to high initializes the device into operation. If the internal Power-on-Reset circuit is used to trigger device power-up, this signal should be connected to VCC3P3.
	MAIN_PWR_OK	R2	In	Main Power OK – Indicates that platform main power is up. Must be connected externally to main core 3.3V power.
	DEV_OFF_N	V24	In	Device Off: Assertion of DEV_OFF_N puts the device in Device Disable mode. This pin is asynchronous and is sampled once the EEPROM is ready to be read following power-up. The DEV_OFF_N pin should always be connected to VCC3P3 to enable device operation.
	XTAL1 XTAL2	D23 D24	A-In A-out	Reference Clock / XTAL: These pins may be driven by an external 25MHz crystal or driven by a single ended external CMOS compliant 25MHz oscillator.

2.3.6 PHY Pins

2.3.6.1 LED's

The table below describes the functionality of the LED output pins. Default activity of the LED may be modified in the EEPROM word offsets 1Ch and 1Fh from start of relevant LAN Port section. The LED functionality is reflected and can be further modified in the configuration registers LEDCTL.

Table 2-22 LED Output Pins

	Symbol	Ball #	Type	Name and Function
	LED0_0	H4	Out	Port 0 LED0. Programmable LED which indicates by default Link Up. ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.20) or *LEDCTL* register (refer to Section 8.2.9).

Table 2-22 LED Output Pins (Continued)

	Symbol	Ball #	Type	Name and Function
	LED0_1	J3	Out	Port 0 LED1. Programmable LED which indicates by default activity (when packets are transmitted or received that match MAC filtering). ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.18) or *LEDCTL* register (refer to Section 8.2.9).
	LED0_2	J4	Out	Port 0 LED2. Programmable LED which indicates by default a 100Mbps Link. ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.20) or *LEDCTL* register (refer to Section 8.2.9).
	LED0_3	K4	Out	Port 0 LED3. Programmable LED which indicates by default a 1000Mbps Link. ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.18) or *LEDCTL* register (refer to Section 8.2.9).
	LED1_0	J21	Out	Port 1 LED0. Programmable LED which indicates by default Link up. ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.20) or *LEDCTL* register (refer to Section 8.2.9).
	LED1_1	J22	Out	Port 1 LED1. Programmable LED which indicates by default activity (when packets are transmitted or received that match MAC filtering). ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.18) or *LEDCTL* register (refer to Section 8.2.9).
	LED1_2	K21	Out	Port 1 LED2. Programmable LED which indicates by default a 100Mbps Link. ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.20) or *LEDCTL* register (refer to Section 8.2.9).
	LED1_3	K22	Out	Port 1 LED3. Programmable LED which indicates by default a 1000Mbps Link. ***Note:*** Pin is active low by default, can be programmed via EEPROM (refer to Section 6.2.18) or *LEDCTL* register (refer to Section 8.2.9).

2.3.6.2 PHY Analog Pins

The AC specification for these pins is described in sections Section .

Table 2-23 PHY Analog Pins

	Symbol	Ball #	Type	Name and Function
	MDI0_0_p MDI0_0_n MDI1_0_p MDI1_0_n	A3 B3 A22 B22	A	Media Dependent Interface[0] for port 0 and port 1 accordingly: 1000BASE-T: In MDI configuration, MDI[0]+/- corresponds to BI_DA+/- and in MDIX configuration MDI[0]+/- corresponds to BI_DB+/-. 100BASE-TX: In MDI configuration, MDI[0]+/- is used for the transmit pair and in MDIX configuration MDI[0]+/- is used for the receive pair. 10BASE-T: In MDI configuration, MDI[0]+/- is used for the transmit pair and in MDIX configuration MDI[0]+/- is used for the receive pair. ***Note:*** When *IPCNFG.MDI_Flip* register bit is set to 1b MDI[0]+/- and MDI[3]+/- are swapped.

Table 2-23 PHY Analog Pins (Continued)

	Symbol	Ball #	Type	Name and Function
	MDI0_1_p MDI0_1_n MDI1_1_p MDI1_1_n	A5 B5 A20 B20	A	Media Dependent Interface[1] for port 0 and port 1 accordingly: 1000BASE-T: In MDI configuration, MDI[1]+/- corresponds to BI_DB+/- and in MDIX configuration MDI[1]+/- corresponds to BI_DA+/-. 100BASE-TX: In MDI configuration, MDI[1]+/- is used for the receive pair and in MDIX configuration MDI[1]+/- is used for the transmit pair. 10BASE-T: In MDI configuration, MDI[1]+/- is used for the receive pair and in MDIX configuration MDI[1]+/- is used for the transmit pair. ***Note:*** When *IPCNFG.MDI_Flip* register bit is set to 1b MDI[1]+/- and MDI[2]+/- are swapped.
	MDI0_2_p MDI0_2_n MDI1_2_p MDI1_2_n	A7 B7 A18 B18	A	Media Dependent Interface[2] for port 0 and port 1: 1000BASE-T: In MDI configuration, MDI[2]+/- corresponds to BI_DC+/- and in MDIX configuration MDI[2]+/- corresponds to BI_DD+/-. 100BASE-TX: Unused. 10BASE-T: Unused. ***Note:*** When *IPCNFG.MDI_Flip* register bit is set to 1b MDI[1]+/- and MDI[2]+/- are swapped.
	MDI0_3_p MDI0_3_n MDI1_3_p MDI1_3_n	A9 B9 A16 B16	A	Media Dependent Interface[3] for port 0 and port 1: 1000BASE-T: In MDI configuration, MDI[3]+/- corresponds to BI_DD+/- and in MDIX configuration MDI[3]+/- corresponds to BI_DC+/-. 100BASE-TX: Unused. 10BASE-T: Unused. ***Note:*** When *IPCNFG.MDI_Flip* register bit is set to 1b MDI[0]+/- and MDI[3]+/- are swapped.
	GE_REXT3K	D12	B	PHY Bias Connect 3.01KΩ 1% resistor between this pin and ground.
	RSVD_TX_TCLK	L23	Out	Transmit 125MHz clock for IEEE testing. Shared for the 4 ports. Not connected in normal operation
	RSVD_ATST_P RSVD_ATST_N	G21 G22	A-out	Analog differential test pins. Shared for the 4 ports. Not connected in normal operation.

2.3.7 Testability Pins

Table 2-24 Testability Pins

	Symbol	Ball #	Type	Name and Function
	RSVD_TE_VSS	Y4	In	Enables test mode. When high test pins are multiplexed on functional signals. In functional mode, must be connected to ground.
	RSVD_JTP8	V3	In	Test pin for production testing. In functional mode should not be connected.
	JTCK	Y22	In	JTAG Clock Input
	JTDI	W22	In	JTAG TDI Input
	JTDO	V22	T/S, O/D	JTAG TDO Output
	JTMS	W21	In	JTAG TMS Input
	RSRVD_JRST_3P3	W23	In	JTAG Reset Input

Table 2-24 Testability Pins (Continued)

	Symbol	Ball #	Type	Name and Function
	AUX_PWR	P2	T/S	Auxiliary Power Available: This pin is a strapping option pin, latched at the rising edge of PE_RST# or In-Band PCIe Reset. This pin has an internal weak pull-up resistor. In case this pin is driven high during init time it indicates that Auxiliary Power is available and the device should support D3COLD power state if enabled to do so. This pin is also used for testing and scan.
	LAN0_DIS_N	K23	T/S	This pin is a strapping option pin, latched at the rising edge of PE_RST# or In-Band PCIe Reset. In case this pin is asserted during init time, LAN 0 function is disabled. This pin is also used for testing and scan. Refer to Section 4.4.3 and Section 4.4.4 for additional information.
	LAN1_DIS_N	K24	T/S	This pin is a strapping option pin, latched at the rising edge of PE_RST# or In-Band PCIe Reset. In case this pin is asserted during init time, LAN 1 function is disabled. This pin is also used for testing and scan. Refer to Section 4.4.3 and Section 4.4.4 for additional information.
	RSVD_TP_3	U23	T/S	Test pin for production testing. ***Note:*** In functional mode should not be connected.
	RSVD_TP_4	U24	T/S	Test pin for production testing. ***Note:*** In functional mode should not be connected.
	RSVDP22_NC	P22	T/S	Production test pin should not be connected.
	RSVDR23_NC	R23	T/S	Production test pin should not be connected.
	RSVDR24_NC	R24	T/S	Production test pin should not be connected.
	RSVDM22_NC	M22	T/S	Production test pin should not be connected.
	RSVDU1_NC	U1	T/S	Production test pin should not be connected.
	RSVDU2_NC	U2	T/S	Production test pin should not be connected.

2.3.8 Power Supply and Ground Pins

Table 2-25 Power Supply and Ground Pins

	Symbol	Ball #	Type	Name and Function
	VCC3P3	A2, A4, A6, A8, A10, A12, A13, A15, A17, A19, A21, A23, AB3, AB5, AB8, AB11, AB14, AB17, AB20, AB22, C2, C6, C8, C10, C15, C17, C19, C21, C23, C24, L5, M6, M20, N5, N19, P6, P20,R5, R19, T6, T20, U5, U19, V6, V20, W5, W19	3.3V	3.3V power supply

Table 2-25 Power Supply and Ground Pins (Continued)

	Symbol	Ball #	Type	Name and Function
	VCC1P0	G5, G7, G9, G13, G15, G17, G19, H6, H18, H20, J5, J7, J9, J11, J13, J15, J19, K6, K10, K12, K14, K16, K18, K20, L7, L9, L11, L13, L15, L19, M10, M14, M16, M18, N7, N9, N11, N15, P10, P12, P14, P16, P18, R7, R9, R11, R13, R15, T10, T12, T14, T16, T18, U7, V8, V16, V18, W7, W9, W11, W13, W15, W17	1.0V	1.0V power supply
	VCC1P8	D10, D16, D18, D4, D6, D8, D20, E3, E5, E7, E9, E11, E13, E15, E17, E19, E21, Y6, Y8, Y10, Y12, Y14, Y16, Y18	1.8V	1.8V power supply

Table 2-26 VSS

Signal	Pin
VSS	AA1, AA2, AA3, AA4, AA5, AA7, AA9, AA11, AA13, AA15, AA17, AA19, AA20, AA21, AA22, AA23, AB4, AB6, AB7, AB9, AB10, AB12, AB13, AB15, AB16, AB18, AB19, AB21, AC1, AC5, AC8, AC11, AC14, AC17, AC2, AC20, AC23, AC24, AD2, AD5, AD8, AD11, AD14, AD17, AD20, AD23, B1, B2, B4, B6, B8, B10, B12, B13, B15, B17, B19, B21, B23, B24, C3, C5, C7, C9, C14, C16, C18, C20, C22, D2, D3, D5, D7, D9, D15, D17, D19, D21, D22, E2, E4, E6, E8, E10, E12, E14, E16, E18, E20, E22, F5, F6, F7, F8, F9, F10, F11, F12, F13, F14, F15, F16, F17, F18, F19, F20, F21, F22, F23, G6, G8, G10, G18, G20, H5, H7, H19, J6, J10, J12, J14, J16, J18, J20, K5, K7, K9, K11, K13, K15, K19, L6, L10, L12, L14, L16, L18, L20, M5, M7, M9, M11, M13, M15, M19, N6, N10, N12, N14, N16, N18, N20, P5, P7, P9, P11, P13, P15, P19, R6, R10, R12, R14, R16, R18, R20, T5, T7, T9, T11, T13, T15, T19, U6, U18, U20, V5, V7, V17, V19, W3, W6, W8, W10, W12, W14, W16, W18, W20, Y3, Y7, Y9, Y11, Y13, Y15, Y17, Y19, Y23, Y24

Table 2-27 Reserved Pins[1]

Signal	Pin
Reserved_N	A1, A11, A14, AA6, AA8, AA10, AA14, AA16, AA18, AB23, AB24, AC13, AC15, AC16, AC18, AC19, AC21, AC22, AD13, AD15, AD16, AD18,AD19, AD21, AD22, B11, B14, C1, C11, C12, C13, D1, D11, D13, E1, E23, E24, F24, G11, L4, L21, L22, M3, M4, M12, M21, N1, N4, N13, N21, N22, P1, P4, P21, R21, R22, T2, T3, U3, U4, V4, V23, Y5, Y20
Reserved_G	A24, AA12, AA24, AD1, AD24, D14, G23, G24, H21, H22, H23, H24, J23, J24, K3, M1, M23, M24, N23, N24, P23, P24, V21, T23, T24, W4, W24, Y21
NB (No Ball - Depopulated Balls)	G12, G14, G16, H9, H11, H13, H15, H17, J8, L8, N8, R8, U8, U10, U12, U14, U16, V9, V11, V13, V15, K17, M17, P17, T17, H8, H10, H12, H14, H16, K8, M8, P8, T8, U9, U11, U13, U15, U17, V10, V12, V14, J17, L17, N17, R17

1. Reserved Pins can be left unconnected.

2.3.9 25x25 PBGA Package Pin List (Alphabetical)

Table 2-28 lists the pins and signals in ball alphabetical order.

Table 2-28 25x25 PBGA Package Pin List in Alphabetical Order

Signal	Ball	Signal	Ball	Signal	Ball
Reserved_N	A1	VSS	B17	VSS	D9
VCC3P3	A2	MDI1_2_n	B18	VCC1P8	D10
MDI0_0_p	A3	VSS	B19	Reserved_N	D11
VCC3P3	A4	MDI1_1_n	B20	GE_REXT3K	D12
MDI0_1_p	A5	VSS	B21	Reserved_N	D13

Table 2-28 25x25 PBGA Package Pin List in Alphabetical Order

Signal	Ball	Signal	Ball	Signal	Ball
VCC3P3	A6	MDI1_0_n	B22	Reserved_G	D14
MDI0_2_p	A7	VSS	B23	VSS	D15
VCC3P3	A8	VSS	B24	VCC1P8	D16
MDI0_3_p	A9	Reserved_N	C1	VSS	D17
VCC3P3	A10	VCC3P3	C2	VCC1P8	D18
Reserved_N	A11	VSS	C3	VSS	D19
VCC3P3	A12	DEV_OFF_N	C4	VCC1P8	D20
VCC3P3	A13	VSS	C5	VSS	D21
Reserved_N	A14	VCC3P3	C6	VSS	D22
VCC3P3	A15	VSS	C7	XTAL1	D23
MDI1_3_p	A16	VCC3P3	C8	XTAL2	D24
VCC3P3	A17	VSS	C9	Reserved_N	E1
MDI1_2_p	A18	VCC3P3	C10	VSS	E2
VCC3P3	A19	Reserved_N	C11	VCC1P8	E3
MDI1_1_p	A20	Reserved_N	C12	VSS	E4
VCC3P3	A21	Reserved_N	C13	VCC1P8	E5
MDI1_0_p	A22	VSS	C14	VSS	E6
VCC3P3	A23	VCC3P3	C15	VCC1P8	E7
Reserved_G	A24	VSS	C16	VSS	E8
VSS	B1	VCC3P3	C17	VCC1P8	E9
VSS	B2	VSS	C18	VSS	E10
MDI0_0_n	B3	VCC3P3	C19	VCC1P8	E11
VSS	B4	VSS	C20	VSS	E12
MDI0_1_n	B5	VCC3P3	C21	VCC1P8	E13
VSS	B6	VSS	C22	VSS	E14
MDI0_2_n	B7	VCC3P3	C23	VCC1P8	E15
VSS	B8	VCC3P3	C24	VSS	E16
MDI0_3_n	B9	Reserved_N	D1	VCC1P8	E17
VSS	B10	VSS	D2	VSS	E18
Reserved_N	B11	VSS	D3	VCC1P8	E19
VSS	B12	VCC1P8	D4	VSS	E20
VSS	B13	VSS	D5	VCC1P8	E21
Reserved_N	B14	VCC1P8	D6	VSS	E22
VSS	B15	VSS	D7	Reserved_N	E23
MDI1_3_n	B16	VCC1P8	D8	Reserved_N	E24
NCSI_ARB_IN	F1	VSS	G20	VCC1P0	J15
NCSI_ARB_OUT	F2	RSVD_ATST_P	G21	VSS	J16
TSENSZ	F3	RSVD_ATST_N	G22	NB	J17
TSENSP	F4	Reserved_G	G23	VSS	J18
VSS	F5	Reserved_G	G24	VCC1P0	J19
VSS	F6	NCSI_CRS_DV	H1	VSS	J20
VSS	F7	NCSI_TXD_0	H2	LED1_0	J21
VSS	F8	NCSI_RXD_0	H3	LED1_1	J22

Table 2-28 25x25 PBGA Package Pin List in Alphabetical Order

Signal	Ball	Signal	Ball	Signal	Ball
VSS	F9	LED0_0	H4	Reserved_G	J23
VSS	F10	VSS	H5	Reserved_G	J24
VSS	F11	VCC1P0	H6	FLSH_SO	K1
VSS	F12	VSS	H7	FLSH_SI	K2
VSS	F13	NB	H8	Reserved_G	K3
VSS	F14	NB	H9	LED0_3	K4
VSS	F15	NB	H10	VSS	K5
VSS	F16	NB	H11	VCC1P0	K6
VSS	F17	NB	H12	VSS	K7
VSS	F18	NB	H13	NB	K8
VSS	F19	NB	H14	VSS	K9
VSS	F20	NB	H15	VCC1P0	K10
VSS	F21	NB	H16	VSS	K11
VSS	F22	NB	H17	VCC1P0	K12
VSS	F23	VCC1P0	H18	VSS	K13
Reserved_N	F24	VSS	H19	VCC1P0	K14
NCSI_RXD_1	G1	VCC1P0	H20	VSS	K15
NCSI_CLK_IN	G2	Reserved_G	H21	VCC1P0	K16
NCSI_TXD_1	G3	Reserved_G	H22	NB	K17
NCSI_TX_EN	G4	Reserved_G	H23	VCC1P0	K18
VCC1P0	G5	Reserved_G	H24	VSS	K19
VSS	G6	FLSH_SCK	J1	VCC1P0	K20
VCC1P0	G7	FLSH_CE_N	J2	LED1_2	K21
VSS	G8	LED0_1	J3	LED1_3	K22
VCC1P0	G9	LED0_2	J4	LAN0_DIS_N	K23
VSS	G10	VCC1P0	J5	LAN1_DIS_N	K24
Reserved_N	G11	VSS	J6	SMBD	L1
NB	G12	VCC1P0	J7	SMBCLK	L2
VCC1P0	G13	NB	J8	NCSI_CLK_OUT	L3
NB	G14	VCC1P0	J9	Reserved_N	L4
VCC1P0	G15	VSS	J10	VCC3P3	L5
NB	G16	VCC1P0	J11	VSS	L6
VCC1P0	G17	VSS	J12	VCC1P0	L7
VSS	G18	VCC1P0	J13	NB	L8
VCC1P0	G19	VSS	J14	VCC1P0	L9
VSS	L10	VCC3P3	N5	Reserved_G	P24
VCC1P0	L11	VSS	N6	EE_DI	R1
VSS	L12	VCC1P0	N7	MAIN_PWR_OK	R2
VCC1P0	L13	NB	N8	SDP0_3	R3
VSS	L14	VCC1P0	N9	SDP0_0	R4
VCC1P0	L15	VSS	N10	VCC3P3	R5
VSS	L16	VCC1P0	N11	VSS	R6
NB	L17	VSS	N12	VCC1P0	R7

Table 2-28 25x25 PBGA Package Pin List in Alphabetical Order

Signal	Ball	Signal	Ball	Signal	Ball
VSS	L18	Reserved_N	N13	NB	R8
VCC1P0	L19	VSS	N14	VCC1P0	R9
VSS	L20	VCC1P0	N15	VSS	R10
Reserved_N	L21	VSS	N16	VCC1P0	R11
Reserved_N	L22	NB	N17	VSS	R12
RSVD_TX_TCLK	L23	VSS	N18	VCC1P0	R13
LAN_PWR_GOOD	L24	VCC3P3	N19	VSS	R14
Reserved_G	M1	VSS	N20	VCC1P0	R15
SMBALRT_N	M2	Reserved_N	N21	VSS	R16
Reserved_N	M3	Reserved_N	N22	NB	R17
Reserved_N	M4	Reserved_G	N23	VSS	R18
VSS	M5	Reserved_G	N24	VCC3P3	R19
VCC3P3	M6	Reserved_N	P1	VSS	R20
VSS	M7	AUX_PWR	P2	Reserved_N	R21
NB	M8	SDP0_1	P3	Reserved_N	R22
VSS	M9	Reserved_N	P4	RSVDR23_NC	R23
VCC1P0	M10	VSS	P5	RSVDR24_NC	R24
VSS	M11	VCC3P3	P6	EE_DO	T1
Reserved_N	M12	VSS	P7	Reserved_N	T2
VSS	M13	NB	P8	Reserved_N	T3
VCC1P0	M14	VSS	P9	SDP0_2	T4
VSS	M15	VCC1P0	P10	VSS	T5
VCC1P0	M16	VSS	P11	VCC3P3	T6
NB	M17	VCC1P0	P12	VSS	T7
VCC1P0	M18	VSS	P13	NB	T8
VSS	M19	VCC1P0	P14	VSS	T9
VCC3P3	M20	VSS	P15	VCC1P0	T10
Reserved_N	M21	VCC1P0	P16	VSS	T11
RSVDM22_NC	M22	NB	P17	VCC1P0	T12
Reserved_G	M23	VCC1P0	P18	VSS	T13
Reserved_G	M24	VSS	P19	VCC1P0	T14
Reserved_N	N1	VCC3P3	P20	VSS	T15
EE_SK	N2	Reserved_N	P21	VCC1P0	T16
EE_CS_N	N3	RSVDP22_NC	P22	NB	T17
Reserved_N	N4	Reserved_G	P23	VCC1P0	T18
VSS	T19	NB	V14	VSS	Y9
VCC3P3	T20	NB	V15	VCC1P8	Y10
SDP1_0	T21	VCC1P0	V16	VSS	Y11
SDP1_1	T22	VSS	V17	VCC1P8	Y12
Reserved_G	T23	VCC1P0	V18	VSS	Y13
Reserved_G	T24	VSS	V19	VCC1P8	Y14
RSVDU1_NC	U1	VCC3P3	V20	VSS	Y15
RSVDU2_NC	U2	Reserved_G	V21	VCC1P8	Y16

Table 2-28 25x25 PBGA Package Pin List in Alphabetical Order

Signal	Ball	Signal	Ball	Signal	Ball
Reserved_N	U3	JTDO	V22	VSS	Y17
Reserved_N	U4	Reserved_N	V23	VCC1P8	Y18
VCC3P3	U5	DEV_OFF_N	V24	VSS	Y19
VSS	U6	PE_WAKE_N	W1	Reserved_N	Y20
VCC1P0	U7	PE_RST_N	W2	Reserved_G	Y21
NB	U8	VSS	W3	JTCK	Y22
NB	U9	Reserved_G	W4	VSS	Y23
NB	U10	VCC3P3	W5	VSS	Y24
NB	U11	VSS	W6	VSS	AA1
NB	U12	VCC1P0	W7	VSS	AA2
NB	U13	VSS	W8	VSS	AA3
NB	U14	VCC1P0	W9	VSS	AA4
NB	U15	VSS	W10	VSS	AA5
NB	U16	VCC1P0	W11	Reserved_N	AA6
NB	U17	VSS	W12	VSS	AA7
VSS	U18	VCC1P0	W13	Reserved_N	AA8
VCC3P3	U19	VSS	W14	VSS	AA9
VSS	U20	VCC1P0	W15	Reserved_N	AA10
SDP1_2	U21	VSS	W16	VSS	AA11
SDP1_3	U22	VCC1P0	W17	Reserved_G	AA12
RSVD_TP_3	U23	VSS	W18	VSS	AA13
RSVD_TP_4	U24	VCC3P3	W19	Reserved_N	AA14
PE_TRIM1	V1	VSS	W20	VSS	AA15
PE_TRIM2	V2	JTMS	W21	Reserved_N	AA16
RSVD_JTP8	V3	JTDI	W22	VSS	AA17
Reserved_N	V4	RSRVD_JRST_3P3	W23	Reserved_N	AA18
VSS	V5	Reserved_G	W24	VSS	AA19
VCC3P3	V6	PE_CLK_n	Y1	VSS	AA20
VSS	V7	PE_CLK_p	Y2	VSS	AA21
VCC1P0	V8	VSS	Y3	VSS	AA22
NB	V9	RSVD_TE_VSS	Y4	VSS	AA23
NB	V10	Reserved_N	Y5	Reserved_G	AA24
NB	V11	VCC1P8	Y6	PER_0_n	AB1
NB	V12	VSS	Y7	PER_0_p	AB2
NB	V13	VCC1P8	Y8	VCC3P3	AB3
VSS	AB4	PET_0_p	AC3	VSS	AD2
VCC3P3	AB5	PET_1_p	AC4	PET_0_n	AD3
VSS	AB6	VSS	AC5	PET_1_n	AD4
VSS	AB7	PER_1_n	AC6	VSS	AD5
VCC3P3	AB8	PER_2_n	AC7	PER_1_p	AD6
VSS	AB9	VSS	AC8	PER_2_p	AD7
VSS	AB10	PET_2_p	AC9	VSS	AD8
VCC3P3	AB11	PET_3_p	AC10	PET_2_n	AD9

Table 2-28 25x25 PBGA Package Pin List in Alphabetical Order

Signal	Ball	Signal	Ball	Signal	Ball
VSS	AB12	VSS	AC11	PET_3_n	AD10
VSS	AB13	PER_3_n	AC12	VSS	AD11
VCC3P3	AB14	Reserved_N	AC13	PER_3_p	AD12
VSS	AB15	VSS	AC14	Reserved_N	AD13
VSS	AB16	Reserved_N	AC15	VSS	AD14
VCC3P3	AB17	Reserved_N	AC16	Reserved_N	AD15
VSS	AB18	VSS	AC17	Reserved_N	AD16
VSS	AB19	Reserved_N	AC18	VSS	AD17
VCC3P3	AB20	Reserved_N	AC19	Reserved_N	AD18
VSS	AB21	VSS	AC20	Reserved_N	AD19
VCC3P3	AB22	Reserved_N	AC21	VSS	AD20
Reserved_N	AB23	Reserved_N	AC22	Reserved_N	AD21
Reserved_N	AB24	VSS	AC23	Reserved_N	AD22
VSS	AC1	VSS	AC24	VSS	AD23
VSS	AC2	Reserved_G	AD1	Reserved_G	AD24

2.4 Pullups/Pulldowns

The table below lists internal & external pull-up resistors and their functionality in different device states. Each internal PUP has a nominal value of 100KΩ, ranging from 50KΩ to 150KΩ.

The device states are defined as follows:

- Power-up = while 3.3V is stable, yet 1.0V isn't
- Active = normal mode (not power up or disable)
- Disable = device disable (a.k.a. dynamic IDDQ – refer to Section 4.5)

Table 2-29 Pull-Up Resistors

Signal Name	Power up[5]		Active		Disable[6]		External
	PUP	Comments	PUP	Comments	PUP	Comments	
LAN_PWR_GOOD	N		N		N		PU
PE_WAKE_N	N		N		N		Y
PE_RST_N	N		N		N		N
FLSH_SI	Y		N		Y		N
FLSH_SO	Y		Y		Y		N
FLSH_SCK	Y		N		Y		N
FLSH_CE_N	Y		N		Y		N
EE_DI	Y		N		Y		N
EE_DO	Y		Y		Y		N
EE_SK	Y		N		Y		N

Table 2-29 Pull-Up Resistors (Continued)

Signal Name	Power up[5]		Active		Disable[6]		External
	PUP	Comments	PUP	Comments	PUP	Comments	
EE_CS_N	Y		N		Y		N
SMBD	N		N		N		Y
SMBCLK	N		N		N		Y
SMBALRT_N	N		N		N		Y
NCSI_CLK_IN	N	HiZ	N		N		PD (Note 1.)
NCSI_CLK_OUT	N		N		N		N
NCSI_CRS_DV	N	HiZ	N		N		Y (Note 2.)
NCSI_RXD[1:0]	N	HiZ	N		N		Y (Note 2.)
NCSI_TX_EN	N	HiZ	N		N		PD (Note 1.)
NCSI_TXD[1:0]	N	HiZ	N		N		PD (Note 1.)
NCSI_ARB_OUT	N		N		N	Stable High output	N
NCSI_ARB_IN	N	HiZ	N	(Note 7.)	N	(Note 7.)	
SDP0[3:0]	Y		Y	Until EEPROM done	Y	May keep state by EEPROM control	N
SDP1[3:0]	Y		Y	Until EEPROM done	Y	May keep state by EEPROM control	N
SDP2[3:0]	Y		Y	Until EEPROM done.	Y	May keep state by EEPROM control	N
SDP3[3:0]	Y		Y	Until EEPROM done.	Y	May keep state by EEPROM control	N
DEV_OFF_N	Y		N		N		Must be connected on board
MAIN_PWR_OK	Y		Y		N		Must be connected on board
SRDS_0_SIG_DET	Y		N		N		Must be connected externally
SRDS_1_SIG_DET	Y		N		N		Must be connected externally
SRDS_2_SIG_DET	Y		N		N		Must be connected externally
SRDS_3_SIG_DET	Y		N		N		Must be connected externally
SFP0_I2C_CLK	Y		Y	Until EEPROM done or if *I2C disable* set in EEPROM	Y		Y if I2C

Table 2-29 Pull-Up Resistors (Continued)

Signal Name	Power up[5]		Active		Disable[6]		External
	PUP	Comments	PUP	Comments	PUP	Comments	
SFP0_I2C_DATA	Y		Y	Until EEPROM done or if *I2C disable* set in EEPROM	Y		Y
SFP1_I2C_CLK	Y		Y	Until EEPROM done or if *I2C disable* set in EEPROM	Y		Y if I2C
SFP1_I2C_DATA	Y		Y	Until EEPROM done or if *I2C disable* set in EEPROM	Y		Y
SFP2_I2C_CLK	Y		Y	Until EEPROM done or if *I2C disable* set in EEPROM.	Y		Y if I2C
SFP2_I2C_DATA	Y		Y	Until EEPROM done or if *I2C disable* set in EEPROM	Y		Y
SFP3_I2C_CLK	Y		Y	Until EEPROM done or if *I2C disable* set in EEPROM.	Y		Y if I2C
SFP3_I2C_DATA	Y		Y	Until EEPROM done or if *I2C disable* set in EEPROM.	Y		Y
LED0_0	Y		N		N	HiZ	
LED0_1	Y		N		N	HiZ	
LED0_2	Y		N		N	HiZ	
LED0_3	Y		N		N	HiZ	
LED1_0	Y		N		N	HiZ	
LED1_1	Y		N		N	HiZ	
LED1_2	Y		N		N	HiZ	
LED1_3	Y		N		N	HiZ	
LED2_0	Y		N		N	HiZ	
LED2_1	Y		N		N	HiZ	
LED2_2	Y		N		N	HiZ	
LED2_3	Y		N		N	HiZ	
LED3_0	Y		N		N	HiZ	
LED3_1	Y		N		N	HiZ	
LED3_2	Y		N		N	HiZ	
LED3_3	Y		N		N	HiZ	
RSVD_TE_VSS	N		N		N		Connect to ground
RSVD_JTP8	Y		Y When input		Y		

Table 2-29 Pull-Up Resistors (Continued)

Signal Name	Power up[5]		Active		Disable[6]		External
	PUP	Comments	PUP	Comments	PUP	Comments	
JTCK	N		N		N		Y- Connect PD
JTDI	N		N		N		Y
JTDO	N		N		N		Y
JTMS	N		N		N		Y
RSRVD_JRST_3P3	N		N		N		Y- Connect PU
AUX_PWR	Y		N		N		PU or PD (note 3.)
LAN0_DIS_N	Y		Y when input		Y		PU or PD (note 4.)
LAN1_DIS_N	Y		Y when input		Y		PU or PD (note 4.)
LAN2_DIS_N	Y		Y when input		Y		PU or PD (note 4.)
LAN3_DIS_N	Y		Y when input		Y		PU or PD (note 4.)
VR_EN	N		N		N		PU or PD (note 8.)
RSVD_TP_3 (Note 10.)	Y		Y when input		Y		
RSVD_TP_4 (Note 10.)	Y		Y when input		Y		

Notes:

1. Should be pulled down if NC-SI interface is disabled.
2. Only if NC-SI is unused or set to multi drop configuration.
3. If Aux power is connected, should be pulled up, else should be pulled down.
4. If the specific function is disabled, should be pulled down, else should be pulled up.
5. Power up - LAN_PWR_GOOD = 0
6. Refer to Section 5.2.6 for description of Disable state.
7. If NC-SI Hardware arbitration is disabled via the *NC-SI ARB Enable* EEPROM bit (refer to Section 6.2.22), NCSI_ARB_IN pin is pulled-up internally.
8. If SVR and LVR internal control circuitry is enabled should be pulled up, else should be pulled down or not connected
9.
10. Signal exists only in 25x25 package.

2.5 Strapping

The following signals are used for static configuration. Unless otherwise stated, strapping options are latched on the rising edge of LAN_PWR_GOOD, at power up, at in-band PCI Express reset and at PE_RST_N assertion. At other times, they revert to their standard usage.

Table 2-30 Strapping Options

Purpose	Pin	Behavior	Pull-up / Pull-down
LAN0 Disable	LAN0_DIS_N	When asserted LAN0 is disabled (refer to Section 4.4.3 and Section 4.4.4).	Internal pull-up
LAN1 Disable	LAN1_DIS_N	When asserted LAN1 is disabled (refer to Section 4.4.3 and Section 4.4.4).	Internal pull-up

Table 2-30 Strapping Options (Continued)

Purpose	Pin	Behavior	Pull-up / Pull-down
LAN2 Disable	LAN2_DIS_N[1]	When asserted LAN2 is disabled (refer to Section 4.4.3 and Section 4.4.4).	Internal pull-up
LAN3 Disable	LAN3_DIS_N[1]	When asserted LAN0 is disabled (refer to Section 4.4.3 and Section 4.4.4).	Internal pull-up
AUX_PWR	AUX_PWR	0b – AUX power is not available 1b – AUX power is available	None
PCIe Function 0 Disable	SDP0_1	If the *en_pin_pcie_func_dis* EEPROM bit is set to 1b, when pin is asserted PCIe Function 0 is disabled (refer to Section 4.4.4).	None
PCIe Function 1 Disable	SDP1_1	If the *en_pin_pcie_func_dis* EEPROM bit is set to 1b, when pin is asserted PCIe Function 1 is disabled (refer to Section 4.4.4).	None
PCIe Function 2 Disable	SDP2_1[1]	If the *en_pin_pcie_func_dis* EEPROM bit is set to 1b, when pin is asserted PCIe Function 2 is disabled (refer to Section 4.4.4).	None
PCIe Function 3 Disable	SDP3_1[1]	If the *en_pin_pcie_func_dis* EEPROM bit is set to 1b, when pin is asserted PCIe Function 3 is disabled (refer to Section 4.4.4).	None

1. Only in 17x17 package.

2.6 Interface Diagram

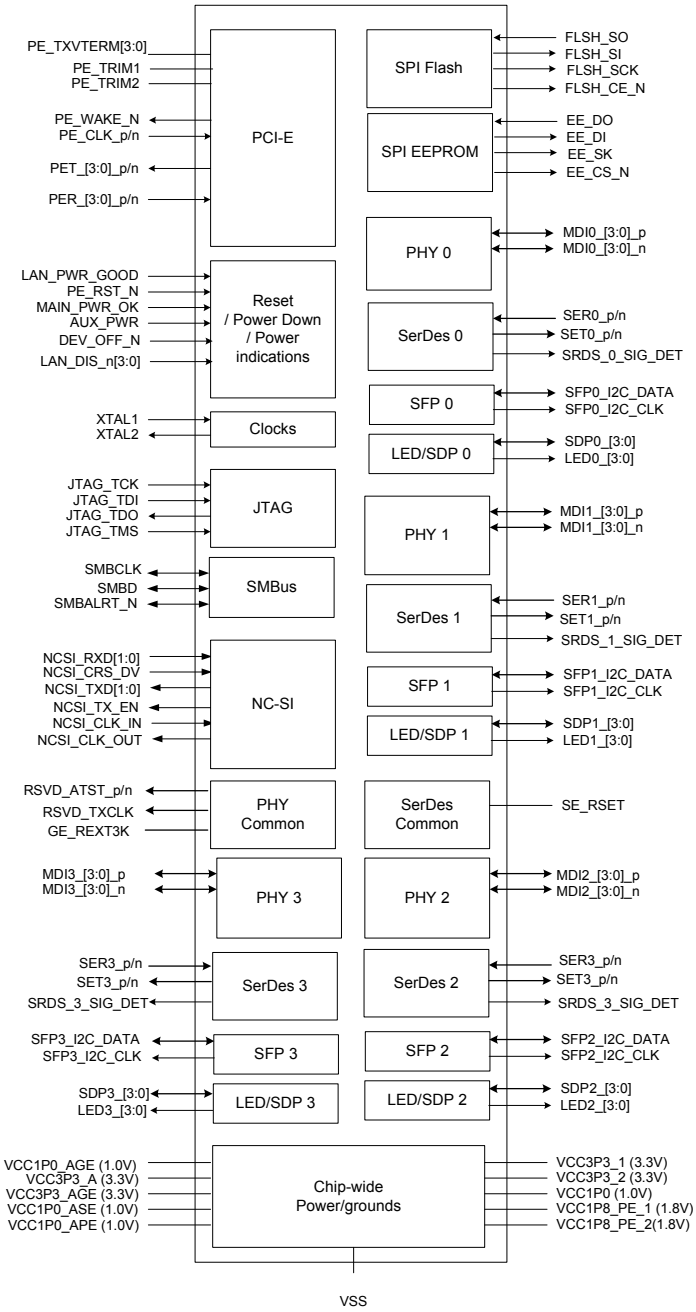


Figure 2-1 I350 Interface Diagram

2.7 17x17 PBGA Package Ball-Out

Figure 2-2 depicts a top view ball map of the I350, in a 17x17 PBGA package. Refer to Section 11.7.2.1 for locating A1 corner ball on package.

*	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
A	PE_TRIM2	PE_TRIM1	PER3_p	VSS	PET_3_p	PET_2_p	VSS	PER_2_p	PER_1_p	VSS	PET_1_p	PET_0_p	VSS	PER0_p	PE_CLK_n	PE_CLK_p
B	PE_RST_N	MAIN_PWR_OK	PER3_n	VSS	PET_3_n	PET_2_n	VSS	PER_2_n	PER_1_n	VSS	PET_1_n	PET_0_n	VSS	PER0_n	FLSH_SI	FLSH_SCK
C	LED0_0	LED0_1	LED0_2	DEVICE_OFF_N	SVR_HDRV	PE_TXVTERM1	VCC1P8_PE_1	LVR_1P8_CTRL	PE_TXVTERM3	VCC1P8_PE_2	PE_TXVTERM4	RSVD_TE_VSS	NCSI_ARB_IN	LAN3_DIS_N	FLSH_SO	FLSH_CE_N
D	LED1_0	LED1_1	LED1_2	LAN_PWR_GOOD	SVR_LDRV	VCC1P0_APE	VSS	VCC1P0_APE	VCC1P0_APE	VSS	VCC1P0_APE	NCSI_ARB_OUT	JTDO	LAN2_DIS_N	AUX_PWR	PE_WAKE_N
E	LED2_0	LED2_1	LED2_2	LED0_3	SMBCLK	VCC1P0	VSS	VSS	VSS	VSS	VCC1P0	JTDI	RSVD_JRST_3P3	LAN1_DIS_N	EE_DI	EE_SK
F	LED3_0	LED3_1	LED3_2	LED1_3	VCC3P3	SVR_COMP	VSS	VSS	VSS	VSS	VSS	VCC3P3	JTCK	LAN0_DIS_N	EE_DO	EE_CS_N
G	LED3_3	LED2_3	SMBD	SMBALRT_N	SVR_FB	VCC1P0	SVR_SW	VSS	VSS	VSS	VCC1P0	VSS	JTMS	VR_EN	RSVD_JTP8	SFP3_I2C_DATA
H	NCSI_CLK_IN	NCSI_CLK_OUT	NCSI_CRS_DV	NCSI_RXD_0	VCC3P3	VCC1P0	VSS	VSS	VSS	VSS	VCC1P0	VCC3P3	SFP2_I2C_DATA	SFP3_I2C_CLK	SER3_n	SER3_p
J	NCSI_RXD_1	NCSI_TX_EN	NCSI_TXD_0	NCSI_TXD_1	VSS	VCC1P0	VSS	VSS	VSS	VSS	VCC1P0	VSS	SFP0_I2C_DATA	SFP2_I2C_CLK	SETn_3	SET3_p
K	SDP0_0	SDP0_1	SDP0_2	SDP0_3	VCC3P3	VCC1P0	VSS	VSS	VSS	VSS	VCC1P0	VCC3P3	SE_RSET	VCC1P0_ASE	SER2_n	SER2_p
L	SDP1_0	SDP1_1	SDP1_2	SDP1_3	VCC3P3_AGE	VSS	VCC1P0_AGE	VCC1P0_AGE	VCC1P0_AGE	VCC1P0_AGE	VSS	VCC3P3	VCC1P0_ASE	VCC1P0_ASE	SET2_n	SET2_p
M	SDP2_0	SDP2_1	SDP2_2	SDP2_3	VSS	VCC3P3_A	VCC3P3_A	VCC3P3_A	VCC3P3_A	VCC3P3_A	VCC1P0	VSS	SFP0_I2C_CLK	SFP1_I2C_CLK	SER1_n	SER1_p
N	SDP3_0	SDP3_1	SDP3_2	SDP3_3	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VCC1P0	SRDS_0_SIG_DET	SFP1_I2C_DATA	SET1_n	SET1_p
P	XTAL_CLK_I	XTAL_CLK_O	VSS	MDI0_1_p	MDI0_1_n	MDI1_1_p	MDI1_1_n	VCC3P3_A	MDI2_3_p	MDI2_3_n	VCC3P3_A	MDI3_3_p	MDI3_3_n	SRDS_1_SIG_DET	SER0_n	SER0_p
R	RSVD_TX_TCLK	TSENSP	MDI0_0_n	MDI0_2_n	MDI0_3_n	MDI1_0_n	MDI1_2_n	MDI1_3_n	MDI2_0_n	MDI2_1_n	MDI2_2_n	MDI3_0_n	MDI3_1_n	MDI3_2_n	SET0_n	SET0_p
T	TSENSZ	GE_REXT3K	MDI0_0_p	MDI0_2_p	MDI0_3_p	MDI1_0_p	MDI1_2_p	MDI1_3_p	MDI2_0_p	MDI2_1_p	MDI2_2_p	MDI3_0_p	MDI3_1_p	MDI3_2_p	SRDS_2_SIG_DET	SRDS_3_SIG_DET

Figure 2-2 17x17 PBGA Package Ball-out

2.8 25x25 PBGA Package Ball-Out

Figure 2-3 depicts a top view ball map of the I350, in a 25x25 PBGA package. Refer to Section 11.7.2.1 for locating A1 corner ball on package.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
A	NC	VCC3P3	MDI0_p_0	VCC3P3	MDI0_p_1	VCC3P3	MDI0_p_2	VCC3P3	MDI0_p_3	VCC3P3	NC	VCC3P3	VCC3P3	NC	VCC3P3	MDI1_p_3	VCC3P3	MDI1_p_2	VCC3P3	MDI1_p_1	VCC3P3	MDI1_p_0	VCC3P3	RSVDA24_VSS	A
B	VSS	VSS	MDI0_n_0	VSS	MDI0_n_1	VSS	MDI0_n_2	VSS	MDI0_n_3	VSS	NC	VSS	VSS	NC	VSS	MDI1_n_3	VSS	MDI1_n_2	VSS	MDI1_n_1	VSS	MDI1_n_0	VSS	VSS	B
C	NC	VCC3P3	VSS	VCC3P3	VSS	VCC3P3	VSS	VCC3P3	VSS	VCC3P3	NC	NC	NC	VSS	VCC3P3	VSS	VCC3P3	VSS	VCC3P3	VSS	VCC3P3	VSS	VCC3P3	VCC3P3	C
D	NC	VSS	VSS	VCC1P8	VSS	VCC1P8	VSS	VCC1P8	VSS	VCC1P8	NC	GE_REXT3K	NC	RSVDD14_VSS	VSS	VCC1P8	VSS	VCC1P8	VSS	VCC1P8	VSS	VSS	XTAL_I	XTAL_O	D
E	NC	VSS	VCC1P8	VSS	VCC1P8	VSS	VCC1P8	VSS	VCC1P8	VSS	VCC1P8	VSS	VCC1P8	VSS	VCC1P8	VSS	VCC1P8	VSS	VCC1P8	VSS	VCC1P8	VSS	NC	NC	E
F	NCSI_ARB_IN	NCSI_ARB_OUT	TSENSZ	TSENSP	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	VSS	NC	F
G	NCSI_RXD1	NCSI_CLK_IN	NCSI_TXD1	NCSI_TX_EN	VCC1P0	VSS	VCC1P0	VSS	VCC1P0	VSS	NC		VCC1P0		VCC1P0		VCC1P0	VSS	VCC1P0	VSS	RSVD_ATST_P	RSVD_ATST_N	RSVDG23_VSS	RSVDG24_VSS	G
H	NCSI_RX_DV	NCSI_TXD0	NCSI_RXD0	LED0_0	VSS	VCC1P0	VSS											VCC1P0	VSS	VCC1P0	RSVDH21_VSS	RSVDH22_VSS	RSVDH23_VSS	RSVDH24_VSS	H
J	FLSH_SCK	FLSH_CE_N	LED0_1	LED0_2	VCC1P0	VSS	VCC1P0		VCC1P0	VSS	VCC1P0	VSS	VCC1P0	VSS	VCC1P0	VSS		VSS	VCC1P0	VSS	LED1_0	LED1_1	RSVDAJ23_VSS	RSVDJ24_VSS	J
K	FLSH_SO	FLSH_SI	RSVDK3_VSS	LED0_3	VSS	VCC1P0	VSS		VSS	VCC1P0	VSS	VCC1P0	VSS	VCC1P0	VSS	VCC1P0		VCC1P0	VSS	VCC1P0	LED1_2	LED1_3	LAN0_DIS_N	LAN1_DIS_N	K
L	SMBD	SMBCLK	NCSI_CLK_OUT	NC	VCC3P3	VSS	VCC1P0		VCC1P0	VSS	VCC1P0	VSS	VCC1P0	VSS	VCC1P0	VSS		VSS	VCC1P0	VSS	NC	NC	RSVD_TX_TCLK	LAN_PWR_GOOD	L
M	RSVDM1_VSS	SMBALRT_N	NC	NC	VSS	VCC3P3	VSS		VSS	VCC1P0	VSS	NC	VSS	VCC1P0	VSS	VCC1P0		VCC1P0	VSS	VCC3P3	NC	NC	RSVDM23_VSS	RSVDM24_VSS	M
N	NC	EE_SK	EE_CS	NC	VCC3P3	VSS	VCC1P0		VCC1P0	VSS	VCC1P0	VSS	NC	VSS	VCC1P0	VSS		VSS	VCC3P3	VSS	NC	NC	RSVDN23_VSS	RSVDN24_VSS	N
P	NC	AUX_PWR	SDP0_1	NC	VSS	VCC3P3	VSS		VSS	VCC1P0	VSS	VCC1P0	VSS	VCC1P0	VSS	VCC1P0		VCC1P0	VSS	VCC3P3	NC	NC	RSVDP23_VSS	RSVDP24_VSS	P
R	EE_DI	MAIN_PWR_OK	SDP0_3	SDP0_0	VCC3P3	VSS	VCC1P0		VCC1P0	VSS	VCC1P0	VSS	VCC1P0	VSS	VCC1P0	VSS		VSS	VCC3P3	VSS	NC	NC	NC	NC	R
T	EE_DO	NC	NC	SDP0_2	VSS	VCC3P3	VSS		VSS	VCC1P0	VSS	VCC1P0	VSS	VCC1P0	VSS	VCC1P0		VCC1P0	VSS	VCC3P3	SDP1_0	SDP1_1	RSVDT23_VSS	RSVDT24_VSS	T
U	NC	NC	NC	NC	VCC3P3	VSS	VCC1P0											VSS	VCC3P3	VSS	SDP1_2	SDP1_3	LAN2_DIS_N/RSVD_TP	LAN3_DIS_N/RSVD_TP	U
V	PE_TRIM1	PE_TRIM2	RSVD_JTP8	NC	VSS	VCC3P3	VSS	VCC1P0								VCC1P0	VSS	VCC1P0	VSS	VCC3P3	RSVDV21_VSS	TDO	NC	DEV_OFF_N	V
W	PE_WAKE_N	PE_RST_N	VSS	RSVDW4_VSS	VCC3P3	VSS	VCC1P0	VSS	VCC1P0	VSS	VCC1P0	VSS	VCC1P0	VSS	VCC1P0	VSS	VCC1P0	VSS	VCC3P3	VSS	TMS	TDI	RSRVD_JRST_3P3	RSVDW24_VSS	W
Y	PE_CLK_n	PE_CLK_p	VSS	RSVD_TE_VSS	NC	VCC1P8	VSS	VCC1P8	VSS	VCC1P8	VSS	VCC1P8	VSS	VCC1P8	VSS	VCC1P8	VSS	VCC1P8	VSS	NC	RSVDY21_VSS	TCK	VSS	VSS	Y
AA	VSS	VSS	VSS	VSS	VSS	NC	VSS	NC	VSS	NC	VSS	RSVDAA12_VSS	VSS	NC	VSS	NC	VSS	NC	VSS	VSS	VSS	VSS	VSS	RSVDAA24_VSS	AA
AB	PER_0_n	PER_0_p	VCC3P3	VSS	VCC3P3	VSS	VSS	VCC3P3	VSS	VSS	VCC3P3	VSS	VSS	VCC3P3	VSS	VSS	VCC3P3	VSS	VSS	VCC3P3	VSS	VCC3P3	NC	NC	AB
AC	VSS	VSS	PET_0_p	PET_1_p	VSS	PER_1_n	PER_2_n	VSS	PET_2_p	PET_3_p	VSS	PER_3_n	NC	VSS	NC	NC	VSS	NC	NC	VSS	NC	NC	VSS	VSS	AC
AD	RSVDAD1_VSS	VSS	PET_0_n	PET_1_n	VSS	PER_1_p	PER_2_p	VSS	PET_2_n	PET_3_n	VSS	PER_3_p	NC	VSS	NC	NC	VSS	NC	NC	VSS	NC	NC	VSS	RSVDAD24_VSS	AD
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	

Figure 2-3 25x25 PBGA Package Ball-out

NOTE: ***This page intentionally left blank.***

§ §

3 Interconnects

3.1 PCIe

3.1.1 PCIe Overview

PCIe is a third generation I/O architecture that enables cost competitive next generation I/O solutions providing industry leading price/performance and features. It is an industry-driven specification.

PCIe defines a basic set of requirements that encases the majority of the targeted application classes. Higher-end applications' requirements, such as enterprise class servers and high-end communication platforms, are encased by a set of advanced extensions that compliment the baseline requirements.

To guarantee headroom for future applications of PCIe, a software-managed mechanism for introducing new, enhanced, capabilities in the platform is provided. Figure 3-1 shows PCIe architecture.

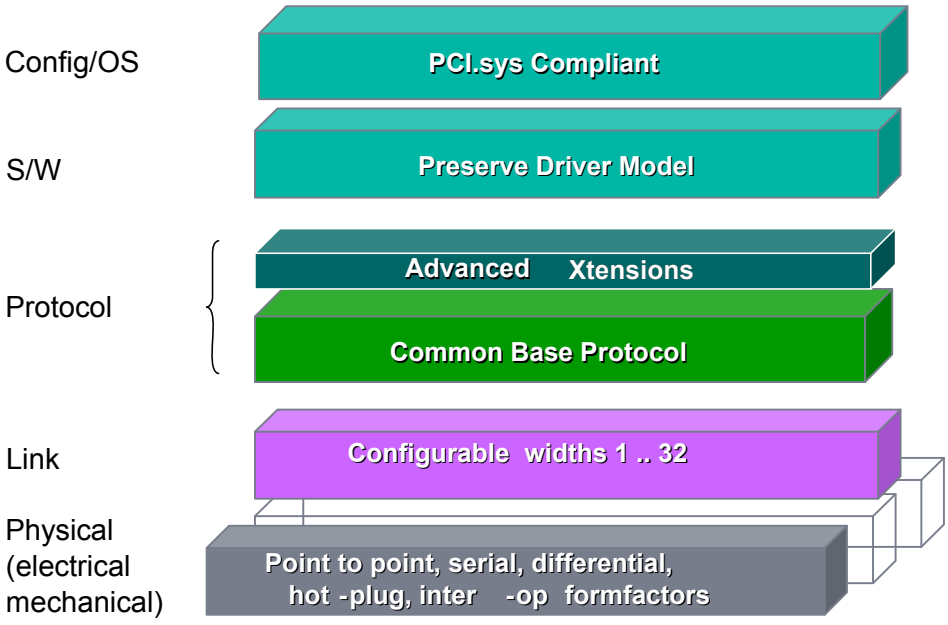


Figure 3-1 PCIe Stack Structure

PCIe's physical layer consists of a differential transmit pair and a differential receive pair. Full-duplex data on these two point-to-point connections is self-clocked such that no dedicated clock signals are required. The bandwidth of this interface increases linearly with frequency.

The packet is the fundamental unit of information exchange and the protocol includes a message space to replace the various side-band signals found on many buses today. This movement of hard-wired signals from the physical layer to messages within the transaction layer enables easy and linear physical layer width expansion for increased bandwidth.

The common base protocol uses split transactions and several mechanisms are included to eliminate wait states and to optimize the reordering of transactions to further improve system performance.

3.1.1.1 Architecture, Transaction and Link Layer Properties

- Split transaction, packet-based protocol
- Common flat address space for load/store access (such as PCI addressing model)
  - Memory address space of 32-bits to allow compact packet header (must be used to access addresses below 4 GB)
  - Memory address space of 64-bit using extended packet header
- Transaction layer mechanisms:
  - PCI-X style relaxed ordering
  - Optimizations for no-snoop transactions
- Credit-based flow control
- Packet sizes/formats:
  - Maximum upstream (write) payload size of 512 Bytes
  - Maximum downstream (read) payload size of 2 KBytes
- Reset/initialization:
  - Frequency/width/profile negotiation performed by hardware
- Data integrity support
  - Using CRC-32 for transaction layer packets
- Link layer retry for recovery following error detection
  - Using CRC-16 for link layer messages
- No retry following error detection
  - 8b/10b encoding with running disparity
- Software configuration mechanism:
  - Uses PCI configuration and bus enumeration model
  - PCIe-specific configuration registers mapped via PCI extended capability mechanism
- Baseline messaging:
  - In-band messaging of formerly side-band legacy signals (such as interrupts, etc.)
  - System-level power management supported via messages
- Power management:
  - Full support for PCI-PM
  - Wake capability from D3cold state
  - Compliant with ACPI, PCI-PM software model

  - — Active state power management
- Support for PCIe rev 2.0
  - — Support for completion time out
  - — Support for additional registers in the PCIe capability structure.

3.1.1.2 Physical Interface Properties

- Point to point interconnect
  - — Full-duplex; no arbitration
- Signaling technology:
  - — Low Voltage Differential (LVD)
  - — Embedded clock signaling using 8b/10b encoding scheme
- Serial frequency of operation: 5 Gbps (Gen2) or 2.5Gbps (Gen1).
- Interface width of x4, x2, or x1.
- DFT and DFM support for high volume manufacturing

3.1.1.3 Advanced Extensions

PCIe defines a set of optional features to enhance platform capabilities for specific usage modes. The I350 supports the following optional features:

- Extended error reporting - messaging support to communicate multiple types/severity of errors.
- Device serial number.
- Completion timeout control.
- TLP Processing Hints (TPH) - provides hints on a per transaction basis to facilitate optimized processing of transactions that target Memory Space.
- Latency Tolerance Reporting (LTR) - messaging support to communicate service latency requirements for Memory Reads and Writes to the Root Complex.

3.1.2 Functionality - General

3.1.2.1 Native/Legacy

All I350 PCI functions are native PCIe functions.

3.1.2.2 Locked Transactions

The I350 does not support locked requests as target or master.

3.1.3 Host Interface

3.1.3.1 Tag IDs

PCIe device numbers identify logical devices within the physical device (the I350 is a physical device). The I350 implements a single logical device with up to four separate PCI functions: LAN 0, LAN 1, LAN2 and LAN3. The device number is captured from each type 0 configuration write transaction.

Each of the PCIe functions interfaces with the PCIe unit through one or more clients. A client ID identifies the client and is included in the *Tag* field of the PCIe packet header. Completions always carry the tag value included in the request to enable routing of the completion to the appropriate client.

Tag IDs are allocated differently for read and write. Messages are sent with a tag of 0x0.

3.1.3.1.1 TAG ID Allocation for Read Transactions

Table 3-1 lists the Tag ID allocation for read accesses. The tag ID is interpreted by hardware in order to forward the read data to the required device.

Table 3-1 IDs in Read Transactions

Tag ID	Description	Comment
0x0	Data request 0	
0x1	Data request 1	
0x2	Data request 2	
0x3	Data request 3	
0x4	Data request 4	
0x5	Data request 5	
0x6	Data request 6	
0x7	Data request 7	
0x8	Data request 8	
0x9	Data request 9	
0xA	Data request 10	
0xB	Data request 11	
0xC	Data request 12	
0xD	Data request 13	
0xE	Data request 14	
0xF	Data request 15	
0x10	Data request 16	
0x11	Data request 17	
0x12	Data request 18	
0x13	Data request 19	
0x14	Data request 20	
0x15	Data request 21	
0x16	Data request 22	
0x17	Data request 23	
0x18	Descriptor Tx 0	
0x19	Descriptor Tx 1	
0x1A	Descriptor Tx 2	

Table 3-1 IDs in Read Transactions (Continued)

0x1B	Descriptor Tx 3	
0x1C	Descriptor Rx 0	
0x1D	Descriptor Rx 1	
0x1E	Descriptor Rx 2	
0x1F	Descriptor Rx 3	

3.1.3.1.2 TAG ID Allocation for Write Transactions

Request tag allocation depends on these system parameters:

- DCA supported/not supported in the system (*DCA_CTRL.DCA_DIS* - refer to Section 8.13.4 for details)
- TPH enabled in the system.
- DCA enabled/disabled for each type of traffic (*TXCTL.TX Descriptor DCA EN*, *RXCTL.RX Descriptor DCA EN*, *RXCTL.RX Header DCA EN*, *RXCTL.Rx Payload DCA EN*).
- TPH enabled or disabled for the specific type of traffic carried by the TLP (*TXCTL.TX Descriptor TPH EN*, *RXCTL.RX Descriptor TPH EN*, *RXCTL.RX Header TPH EN*, *RXCTL.Rx Payload TPH EN*).
- System type: Legacy DCA vs. DCA 1.0 (*DCA_CTRL.DCA_MODE* - refer to Section 8.13.4 for details).
- CPU ID (*RXCTL.CPUID* or *TXCTL.CPUID*).

3.1.3.1.2.1 Case 1 - DCA Disabled in the System:

Table 3-2 describes the write requests tags. Unlike read, the values are for debug only, allowing tracing of requests through the system.

Table 3-2 IDs in Write Transactions, DCA Disabled Mode

Tag ID	Description
0x0 - 0x1	Reserved
0x2	Tx descriptors write-back / Tx Head write-back
0x3	Reserved
0x4	Rx descriptors write-back
0x5	Reserved
0x6	Write data
0x7 - 0x1D	Reserved
0x1E	MSI and MSI-X
0x1F	Reserved

3.1.3.1.2.2 Case 2 - DCA Enabled in the System, but Disabled for the Request:

- Legacy DCA platforms - If DCA is disabled for the request, the tags allocation is identical to the case where DCA is disabled in the system. Refer to Table 3-2 above.
- DCA 1.0 platforms - All write requests have a tag value of 0x00.

Note: When in DCA 1.0 mode, messages and MSI/MSI-x write requests are sent with the no-hint tag.

3.1.3.1.2.3 Case 3 - DCA Enabled in the System, DCA Enabled for the Request:

- Legacy DCA Platforms: the request tag is constructed as follows:
  - Bit[0] – DCA Enable
  - Bits[3:1] - The CPU ID field taken from the CPUID[2:0] bits of the RXCTL or TXCTL registers
  - Bits[7:4] - Reserved
- DCA 1.0 Platforms: the request tag (all 8 bits) is taken from the CPUID field of the RXCTL or TXCTL registers

3.1.3.1.2.4 Case 4 - TPH Enabled in the System, TPH Enabled for the Request:

- The request tag (all 8 bits) is taken from the CPUID field of the adequate register or context as described in Table 7-60.

3.1.3.2 Completion Timeout Mechanism

In any split transaction protocol, there is a risk associated with the failure of a requester to receive an expected completion. To enable requesters to attempt recovery from this situation in a standard manner, the completion timeout mechanism is defined.

The completion timeout mechanism is activated for each request that requires one or more completions when the request is transmitted. The I350 provides a programmable range for the completion timeout, as well as the ability to disable the completion timeout altogether. The completion timeout is programmed through an extension of the PCIe capability structure (refer to Section 9.5.6.12).

The I350's reaction in case of a completion timeout is defined in Table 3-12.

The I350 controls the following aspects of completion timeout:

- Disabling or enabling completion timeout.
- Disabling or enabling re-send of a request on completion timeout.
- A programmable range of re-sends on completion timeout, if re-send enabled.
- A programmable range of timeout values.
- Programming the behavior of completion timeout is summarized in Table 3-12. System software may configure completion timeout independently per each LAN function.

Table 3-3 Completion Timeout Programming

Capability	Programming capability
Completion Timeout Enabling	Controlled through *PCI Device Control 2* configuration register.
Resend Request Enable	Loaded from the EEPROM into the *GCR* register.
Number of re-sends on timeout	Controlled through *GCR* register.
Completion Timeout Period	Controlled through *PCI Device Control 2* configuration register.

Completion Timeout Enable - Programmed through the *PCI Device Control 2* configuration register. The default is: Completion Timeout Enabled.

Resend Request Enable - The *Completion Timeout Resend* EEPROM bit (loaded to the *Completion_Timeout_Resend* bit in the PCIe Control register (GCR) enables resending the request (applies only when completion timeout is enabled). The default is to resend a request that timed out.

Number of re-sends on timeout - Programmed through the *Number of resends* field in the GCR register. The default value of resends is 3.

3.1.3.2.1 Completion Timeout Period

Programmed through the *PCI Device Control 2* configuration register (refer to Section 9.5.6.12). The I350 supports all ranges defined by PCIe v2.1 (2.5GT/s and 5GT/s).

A memory read request for which there are multiple completions are considered completed only when all completions have been received by the requester. If some, but not all, requested data is returned before the completion timeout timer expires, the requestor is permitted to keep or to discard the data that was returned prior to timer expiration.

Note: The Completion Timeout Value must be programmed correctly in PCIe configuration space (in Device Control 2 Register); the value must be set above the expected maximum latency for completions in the system in which the I350 is installed. This will ensure that the I350 receives the completions for the requests it sends out, avoiding a completion timeout scenario. It is expected that the system BIOS will set this value appropriately for the system.

3.1.4 Transaction Layer

The upper layer of the PCIe architecture is the transaction Layer. The transaction layer connects to the I350 core using an implementation specific protocol. Through this core-to-transaction-layer protocol, the application-specific parts of the I350 interact with the PCIe subsystem and transmit and receive requests to or from the remote PCIe agent, respectively.

3.1.4.1 Transaction Types Accepted by the I350

Table 3-4 Transaction Types Accepted by the Transaction Layer

Transaction Type	FC Type	Tx Later Reaction	Hardware Should Keep Data From Original Packet	For Client
Configuration Read Request	NPH	CPLH + CPLD	Requester ID, TAG, Attribute	Configuration space
Configuration Write Request	NPH + NPD	CPLH	Requester ID, TAG, Attribute	Configuration space
Memory Read Request	NPH	CPLH + CPLD	Requester ID, TAG, Attribute	CSR
Memory Write Request	PH + PD	-	-	CSR
IO Read Request	NPH	CPLH + CPLD	Requester ID, TAG, Attribute	CSR
IO Write Request	NPH + NPD	CPLH	Requester ID, TAG, Attribute	CSR
Read completions	CPLH + CPLD	-	-	DMA
Message	PH	-	-	Message Unit / PM

Flow control types:

- PH - Posted request headers
- PD - Posted request data payload
- NPH - Non-posted request headers
- NPD - Non-posted request data payload
- CPLH - Completion headers
- CPLD - Completion data payload

3.1.4.1.1 Configuration Request Retry Status

PCIe supports devices requiring a lengthy self-initialization sequence to complete before they are able to service configuration requests. This is the case for the I350 where initialization is long due to the EEPROM read operation following reset.

If the read of the PCIe section in the EEPROM was not completed and the I350 receives a configuration request, the I350 responds with a configuration request retry completion status to terminate the request, and thus effectively stall the configuration request until the subsystem has completed local initialization and is ready to communicate with the host.

3.1.4.1.2 Partial Memory Read and Write Requests

The I350 has limited support of read and write requests when only part of the byte enable bits are set as described later in this section.

Partial writes to the MSI-X table are supported. All other partial writes are ignored and silently dropped.

Zero-length writes have no internal impact (nothing written, no effect such as clear-by-write). The transaction is treated as a successful operation (no error event).

Partial reads with at least one byte enabled are answered as a full read. Any side effect of the full read (such as clear by read) is applicable to partial reads also.

Zero-length reads generate a completion, but the register is not accessed and undefined data is returned.

3.1.4.2 Transaction Types Initiated by the I350

Table 3-5 Transaction Types Initiated by the Transaction Layer

Transaction type	Payload Size	FC Type	From Client
Configuration Read Request Completion	Dword	CPLH + CPLD	Configuration space
Configuration Write Request Completion	-	CPLH	Configuration space
I/O Read Request Completion	Dword	CPLH + CPLD	CSR
I/O Write Request Completion	-	CPLH	CSR
Read Request Completion	Dword/Qword	CPLH + CPLD	CSR
Memory Read Request	-	NPH	DMA
Memory Write Request	<= MAX_PAYLOAD_SIZE	PH + PD	DMA
Message	-	PH	INT / PM / Error Unit / LTR

Note: MAX_PAYLOAD_SIZE supported is loaded from EEPROM (128 bytes, 256 bytes or 512 bytes). IF ARI capability is not exposed, the effective MAX_PAYLOAD_SIZE is defined for each PCI function according to configuration space register of this function. If ARI capability is exposed, effective MAX_PAYLOAD_SIZE is defined for all PCI functions according to configuration space register of function zero

3.1.4.2.1 Data Alignment

Requests must never specify an address/length combination that causes a memory space access to cross a 4KB boundary. The I350 breaks requests into 4KB-aligned requests (if needed). This does not pose any requirement on software. However, if software allocates a buffer across a 4KB boundary, hardware issues multiple requests for the buffer. Software should consider limiting buffer sizes and base addresses to comply with a 4KB boundary in cases where it improves performance.

The general rules for packet alignment are as follows:

1. The length of a single request should not exceed the PCIe limit of MAX_PAYLOAD_SIZE for write and MAX_READ_REQ for read.
2. The length of a single request does not exceed the I350's internal limitation.
3. A single request should not span across different memory pages as noted by the 4 KB boundary previously mentioned.

Note: The rules apply to all I350 requests (read/write, snoop and no snoop).

If a request can be sent as a single PCIe packet and still meet rules 1-3, then it is not broken at a cache-line boundary (as defined in the PCIe Cache line size configuration word), but rather, sent as a single packet (motivation is that the chipset might break the request along cache-line boundaries, but the I350 should still benefit from better PCIe utilization). However, if rules 1-3 require that the request is broken into two or more packets, then the request is broken at a cache-line boundary.

3.1.4.2.2 Multiple Tx Data Read Requests (MULR)

The I350 supports 24 pipelined requests for transmit data on all ports. In general, the 24 requests might belong to the same packet or to consecutive packets to be transmitted on a single LAN port or on multiple LAN ports. However, the following restriction applies:

- All requests for a packet are issued before a request is issued for a consecutive packet

Read requests can be issued from any of the supported queues, as long as the restriction is met. Pipelined requests might belong to the same queue or to separate queues. However, as previously noted, all requests for a certain packet are issued (from same queue) before a request is issued for a different packet (potentially from a different queue or LAN port).

The PCIe specification does not ensure that completions for separate requests return in-order. Read completions for concurrent requests are not required to return in the order issued. The I350 handles completions that arrive in any order. Once all completions arrive for a given request, the I350 might issue the next pending read data request.

- The I350 incorporates a re-order buffer to support re-ordering of completions for all requests. Each request/completion can be up to 2 KBytes long. The maximum size of a read request is defined as the minimum {2KB, Max_Read_Request_Size}.

In addition to the 24 pipeline requests for transmit data, the I350 can issue up to 4 read requests for all ports (either for a single port or for multiple LAN ports) to fetch transmit descriptors and 4 read requests for all ports (either for a single LAN port or for multiple LAN ports) to fetch receive descriptors. The requests for transmit data, transmit descriptors, and receive descriptors are independently issued. Each descriptor read request can fetch up to 16 descriptors for reception and 24 descriptors for transmission.

3.1.4.3 Messages

3.1.4.3.1 Message Handling by the I350 (as a Receiver)

Message packets are special packets that carry a message code.

The upstream device transmits special messages to the I350 by using this mechanism.

The transaction layer decodes the message code and responds to the message accordingly.

Table 3-6 Supported Messages in the I350 (as a Receiver)

Message code [7:0]	Routing r2r1r0	Message	Device later response
0x14	100	PM_Active_State_NAK	Internal signal set
0x19	011	PME_Turn_Off	Internal signal set
0x50	100	Slot power limit support (has one Dword data)	Silently drop
0x7E	010,011,100	Vendor_defined type 0 no data	Unsupported request [1]
0x7E	010,011,100	Vendor_defined type 0 data	Unsupported request [1]
0x7F	010,011,100	Vendor_defined type 1 no data	Silently drop
0x7F	010,011,100	Vendor_defined type 1 data	Silently drop
0x00	011	Unlock	Silently drop

1. No Completion is expected for this type of packets

3.1.4.3.2 Message Handling by the I350 (as a Transmitter)

The transaction layer is also responsible for transmitting specific messages to report internal/external events (such as interrupts and PMEs).

Table 3-7 Supported Message in the I350 (as a Transmitter)

Message code [7:0]	Routing r2r1r0	Message
0x20	100	Assert INT A
0x21	100	Assert INT B
0x22	100	Assert INT C

Table 3-7 Supported Message in the I350 (as a Transmitter) (Continued)

0x23	100	Assert INT D
0x24	100	De-assert INT A
0x25	100	De-assert INT B
0x26	100	De-assert INT C
0x27	100	De-Assert INT D
0x30	000	ERR_COR
0x31	000	ERR_NONFATAL
0x33	000	ERR_FATAL
0x18	000	PM_PME
0x1B	101	PME_TO_ACK
0x10	100	Latency Tolerance Reporting (LTR)

3.1.4.4 Ordering Rules

The I350 meets the PCIe ordering rules (PCI-X rules) by following the PCI simple device model:

- Deadlock avoidance - Master and target accesses are independent. The response to a target access does not depend on the status of a master request to the bus. If master requests are blocked, such as due to no credits, target completions might still proceed (if credits are available).
- Descriptor/data ordering - The I350 does not proceed with some internal actions until respective data writes have ended on the PCIe link:
  - The I350 does not update an internal header pointer until the descriptors that the header pointer relates to are written to the PCIe link.
  - The I350 does not issue a descriptor write until the data that the descriptor relates to is written to the PCIe link.

The I350 might issue the following master read request from each of the following clients:

- Rx Descriptor Read (up to 4 for all LAN ports)
- Tx Descriptor Read (up to 4 for all LAN ports)
- Tx Data Read (up to 24 for all LAN ports)

Completion of separate read requests are not guaranteed to return in order. Completions for a single read request are guaranteed to return in address order.

3.1.4.4.1 Out of Order Completion Handling

In a split transaction protocol, when using multiple read requests in a multi processor environment, there is a risk that completions arrive from the host memory out of order and interleaved. In this case, the I350 sorts the request completions and transfers them to the Ethernet in the correct order.

3.1.4.5 Transaction Definition and Attributes

3.1.4.5.1 Max Payload Size

The I350 policy to determine Max Payload Size (MPS) is as follows:

- Master requests initiated by the I350 (including completions) limits MPS to the value defined for the function issuing the request.

- Target write accesses to the I350 are accepted only with a size of one Dword or two Dwords. Write accesses in the range of (three Dwords, MPS, etc.) are flagged as UR. Write accesses above MPS are flagged as malformed.

Refer to Section 2.2.2 - TLPs with Data Payloads - Rules of the PCIe base specification.

3.1.4.5.2 Relaxed Ordering

The I350 takes advantage of the relaxed ordering rules in PCIe. By setting the relaxed ordering bit in the packet header, the I350 enables the system to optimize performance in the following cases:

- Relaxed ordering for descriptor and data reads: When the I350 emits a read transaction, its split completion has no ordering relationship with the writes from the CPUs (same direction). It should be allowed to bypass the writes from the CPUs.
- Relaxed ordering for receiving data writes: When the I350 issues receive DMA data writes, it also enables them to bypass each other in the path to system memory because software does not process this data until their associated descriptor writes complete.
- The I350 cannot relax ordering for descriptor writes, MSI/MSI-X writes or PCIe messages.

Relaxed ordering can be used in conjunction with the no-snoop attribute to enable the memory controller to advance non-snoop writes ahead of earlier snooped writes.

Relaxed ordering is enabled in the I350 by clearing the *RO_DIS* bit in the CTRL_EXT register. Actual setting of relaxed ordering is done for LAN traffic by the host through the DCA registers.

3.1.4.5.3 Snoop Not Required

The I350 sets the *Snoop Not Required* attribute bit for master data writes. System logic might provide a separate path into system memory for non-coherent traffic. The non-coherent path to system memory provides higher, more uniform, bandwidth for write requests.

Note: The *Snoop Not Required* attribute does not alter transaction ordering. Therefore, to achieve maximum benefit from *Snoop Not Required* transactions, it is advisable to set the relaxed ordering attribute as well (assuming that system logic supports both attributes). In fact, some chipsets require that relaxed ordering is set for no-snoop to take effect.

Global no-snoop support is enabled in the I350 by clearing the *NS_DIS* bit in the *CTRL_EXT* register. Actual setting of no snoop is done for LAN traffic by the host through the DCA registers.

3.1.4.5.4 No Snoop and Relaxed Ordering for LAN Traffic

Software might configure non-snoop and relax order attributes for each queue and each type of transaction by setting the respective bits in the *RXCTRL* and *TXCTRL* registers.

Table 3-8 lists Software configuration for the *No-Snoop* and *Relaxed Ordering* bits for LAN traffic when I/OAT 2 is enabled.

Table 3-8 LAN Traffic Attributes

Transaction	No-Snoop	Relaxed Ordering	Comments
Rx Descriptor Read	N	Y	
Rx Descriptor Write-Back	N	N	Relaxed ordering must never be used for this traffic.
Rx Data Write	Y	Y	Refer to Note 1 below and Section 3.1.4.5.4.1
Rx Replicated Header	N	Y	

Table 3-8 LAN Traffic Attributes (Continued)

Tx Descriptor Read	N	Y	
Tx Descriptor Write-Back	N	Y	
Tx TSO Header Read	N	Y	
Tx Data Read	N	Y	

Note:
1. Rx payload no-snoop is also conditioned by the *NSE* bit in the receive descriptor. Refer to Section 3.1.4.5.4.1.

3.1.4.5.4.1 No-Snoop Option for Payload

Under certain conditions, which occur when I/OAT is enabled, software knows that it is safe to transfer (DMA) a new packet into a certain buffer without snooping on the front-side bus. This scenario typically occurs when software is posting a receive buffer to hardware that the CPU has not accessed since the last time it was owned by hardware. This might happen if the data was transferred to an application buffer by the I/OAT DMA engine.

In this case, software should be able to set a bit in the receive descriptor indicating that the I350 should perform a no-snoop DMA transfer when it eventually writes a packet to this buffer.

When a non-snoop transaction is activated, the TLP header has a non-snoop attribute in the *Transaction Descriptor* field.

This is triggered by the *NSE* bit in the receive descriptor. Refer to Section 7.1.4.2.

3.1.4.5.5 TLP processing Hint (TPH)

The TPH bit can be set to provide information to the root complex about the cache in which the data should be stored or from which the data should be read as described in Section 7.7.2.

TPH is enabled via the TPH Requester Enable field in the TPH control register of the configuration space (refer to Section 9.6.5.3). Setting of the TPH bit for different type of traffic is described in Table 7-60.

3.1.4.6 Flow Control

3.1.4.6.1 I350 Flow Control Rules

The I350 implements only the default Virtual Channel (VC0). A single set of credits is maintained for VC0.

Table 3-9 Allocation of FC Credits

Credit Type	Operations	Number Of Credits
Posted Request Header (PH)	Target Write (one unit) Message (one unit)	16 credit units to support tail write at wire speed.
Posted Request Data (PD)	Target Write (Length/16 bytes=1) Message (one unit)	MAX_PAYLOAD_SIZE/16
Non-Posted Request Header (NPH)	Target Read (one unit) Configuration Read (one unit) Configuration Write (one unit)	Four units (to enable concurrent target accesses to all LAN ports).
Non-Posted Request Data (NPD)	Configuration Write (one unit)	Four units.
Completion Header (CPLH)	Read Completion (N/A)	Infinite (accepted immediately).
Completion Data (CPLD)	Read Completion (N/A)	Infinite (accepted immediately).

Rules for FC updates:

- The I350 maintains four credits for NPD at any given time. It increments the credit by one after the credit is consumed and sends an UpdateFC packet as soon as possible. UpdateFC packets are scheduled immediately after a resource is available.
- The I350 provides four credits for PH (such as for four concurrent target writes) and four credits for NPH (such as for four concurrent target reads). UpdateFC packets are scheduled immediately after a resource becomes available.
- The I350 follows the PCIe recommendations for frequency of UpdateFC FCPs.

3.1.4.6.2 Upstream Flow Control Tracking

The I350 issues a master transaction only when the required FC credits are available. Credits are tracked for posted, non-posted, and completions (the later to operate with a switch).

3.1.4.6.3 Flow Control Update Frequency

In any case, UpdateFC packets are scheduled immediately after a resource becomes available.

When the link is in the L0 or L0s link state, Update FCPs for each enabled type of non-infinite FC credit must be scheduled for transmission at least once every 30 µs (-0%/+50%), except when the *Extended Sync* bit of the Control Link register is set, in which case the limit is 120 µs (-0%/+50%).

3.1.4.6.4 Flow Control Timeout Mechanism

The I350 implements the optional FC update timeout mechanism.

The mechanism is activated when the Link is in L0 or L0s Link state. It uses a timer with a limit of 200µs (-0%/+50%), where the timer is reset by the receipt of any Init or Update FCP. Alternately, the timer may be reset by the receipt of any DLLP.

After timer expiration, the mechanism instructs the PHY to re-establish the link (via the LTSSM recovery state).

3.1.4.7 Error Forwarding

If a TLP is received with an error-forwarding trailer (Poisoned TLP received), the transaction may either be resent or dropped and not delivered to its destination, depending on the *GCR.Completion Timeout resend enable* bit and the *GCR.Number of resends* field. If the re-sends were unsuccessful or if re-send is disabled, the I350 does not initiate any additional master requests for that PCI function until it detects an internal reset or a software reset for the associated LAN. Software is able to access device registers after such a fault.

System logic is expected to trigger a system-level interrupt to inform the operating system of the problem. The operating system can then stop the process associated with the transaction, re-allocate memory instead of the faulty area, etc.

3.1.5 Data Link Layer

3.1.5.1 ACK/NAK Scheme

The I350 will send ACK/NAK immediately in the following cases:

1. NAK needs to be sent.
2. ACK for duplicate packet
3. ACK/NAK before low power state entry

In all other cases, the I350 will schedule ACK transmission according to time-outs specified in the PCIe specification (depends on link speed, link width, and max_payload_size).

3.1.5.2 Supported DLLPs

The following DLLPs are supported by the I350 as a receiver:

Table 3-10 DLLPs Received by the I350

DLLP type	Remarks
Ack	
Nak	
PM_Request_Ack	
InitFC1-P	Virtual Channel 0 only
InitFC1-NP	Virtual Channel 0 only
InitFC1-Cpl	Virtual Channel 0 only
InitFC2-P	Virtual Channel 0 only
InitFC2-NP	Virtual Channel 0 only
InitFC2-Cpl	Virtual Channel 0 only
UpdateFC-P	Virtual Channel 0 only
UpdateFC-NP	Virtual Channel 0 only
UpdateFC-Cpl	Virtual Channel 0 only

The following DLLPs are supported by the I350 as a transmitter:

Table 3-11 DLLPs Initiated by the I350

DLLP type	Remarks
Ack	
Nak	
PM_Enter_L1	
PM_Enter_L23	
PM_Active_State_Request_L1	
InitFC1-P	Virtual Channel 0 only
InitFC1-NP	Virtual Channel 0 only
InitFC1-Cpl	Virtual Channel 0 only
InitFC2-P	Virtual Channel 0 only
InitFC2-NP	Virtual Channel 0 only

Table 3-11 DLLPs Initiated by the I350 (Continued)

DLLP type	Remarks
InitFC2-Cpl	Virtual Channel 0 only
UpdateFC-P	Virtual Channel 0 only
UpdateFC-NP	Virtual Channel 0 only

Note: UpdateFC-Cpl is not sent because of the infinite FC-Cpl allocation.

3.1.5.3 Transmit EDB Nullifying

In case of a necessity to re-train, there is a need to guarantee that no abrupt termination of the Tx packet happens. For this reason, early termination of the transmitted packet is possible. This is done by appending an EDB (EnD Bad symbol) to the packet.

3.1.6 Physical Layer

3.1.6.1 Link Speed

The I350 supports 2.5GT/s and 5GT/s link speeds. The following PCIe configuration bits define the link speed:

- *Max Link Speed* bit in the *Link CAP* register — Indicates the link speed supported by the I350 as determined by the *Disable PCIe Gen 2* bit in the *PCIe PHY Auto Configuration* EEPROM section.
- *Link Speed* bit in the *Link Status register* — Indicates the negotiated Link speed.
- *Target Link Speed* bit in the *Link Control 2* register — used to set the target compliance mode speed when software is using the Enter Compliance bit to force a link into compliance mode. The default value is determined by the *Disable PCIe Gen 2* bit in the *PCIe PHY Auto Configuration* EEPROM section.

The I350 does not initiate a hardware autonomous speed change and as a result the *Hardware Autonomous Speed Disable* bit in the *PCIe Link Control 2* register is hardwired to 0b.

The I350 supports entering compliance mode at the speed indicated in the *Target Link Speed* field in the *PCIe Link Control 2* register. Compliance mode functionality is controlled via the *Enter Compliance* bit in the *PCIe Link Control 2* register.

3.1.6.2 Link Width

The I350 supports a maximum link width of x4, x2, or x1 as determined by the *Disable Lane* bits in the *PCIe PHY Auto Configuration* EEPROM section.

The max link width is loaded into the *Maximum Link Width* field of the *PCIe Capability* register (*LCAP[11:6]*). The hardware default is x4 link.

During link configuration, the platform and the I350 negotiate on a common link width. The link width must be one of the supported PCIe link widths (x1, x2, x4), such that:

- If *Maximum Link Width* = x4, then the I350 negotiates to either x4, x2 or x1.[1]

1. See restriction in Section 3.1.6.6.

- If *Maximum Link Width* = x2, then the I350 negotiates to either x2 or x1.
- If *Maximum Link Width* = x1, then the I350 only negotiates to x1.

3.1.6.3 Polarity Inversion

If polarity inversion is detected, the receiver must invert the received data.

During the training sequence, the receiver looks at Symbols 6-15 of TS1 and TS2 as the indicator of lane polarity inversion (D+ and D- are swapped). If lane polarity inversion occurs, the TS1 Symbols 6-15 received are D21.5 as opposed to the expected D10.2. Similarly, if lane polarity inversion occurs, Symbols 6-15 of the TS2 ordered set are D26.5 as opposed to the expected D5.2. This provides clear indication of lane polarity inversion.

3.1.6.4 L0s Exit latency

The number of FTS sequences (N_FTS) sent during L1 exit, can be loaded from the EEPROM.

3.1.6.5 Lane-to-Lane De-Skew

A multi-lane link might have many sources of lane to lane skew. Although symbols are transmitted simultaneously on all lanes, they cannot be expected to arrive at the receiver without lane-to-lane skew. The lane-to-lane skew can include components, which are less than a bit time, bit time units (400/200 ps for 2.5/5 Gbps), or full symbol time units (4 ns) of skew caused by the re-timing repeaters' insert/delete operations. Receivers use TS1 or TS2 or Skip Ordered Sets (SOS) to perform link de-skew functions.

The I350 supports de-skew of up to 12 symbol times (48 ns for 2.5 GbpS link rate and 24 ns for 5Gbps link rate).

3.1.6.6 Lane Reversal

The following lane reversal modes are supported (see Figure 3-2):

- Lane configuration of x4, x2, and x1.
- Lane reversal in x4, x2 and in x1.
- Degraded mode (downshift) from x4 to x2 to x1 and from x2 to x1, with one restriction - if lane reversal is executed in x4, then downshift is only to x1 and not to x2.

Note: The restriction requires that a x2 interface to the I350 must connect to lanes 0 and 1 on the I350. The PCIe Card Electromechanical specification does not allow to route a x2 link to a wider connector. Therefore, a system designer is not allowed to connect a x2 link to lanes 2

and 3 of a PCIe connector. It is also recommended that when used in x2 mode on a NIC, the I350 is connected to lanes 0 and 1 of the NIC.

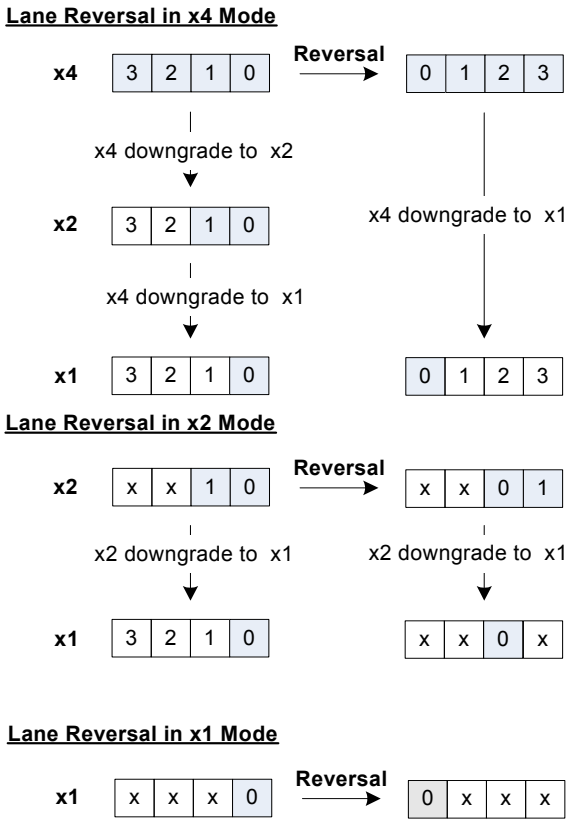


Figure 3-2 Lane Reversal Supported Modes

3.1.6.7 Reset

The PCIe PHY can supply core reset to the I350. The reset can be caused by two sources:

1. Upstream move to hot reset - Inband Mechanism (LTSSM).
2. Recovery failure (LTSSM returns to detect).
3. Upstream component moves to Disable.

3.1.6.8 Scrambler Disable

The scrambler/de-scrambler functionality in the I350 can be eliminated by the two Upstream according to the PCIe specification.

3.1.7 Error Events and Error Reporting

3.1.7.1 Mechanism in General

PCIe defines two error reporting paradigms: the baseline capability and the Advanced Error Reporting (AER) capability. The baseline error reporting capabilities are required of all PCIe devices and define the minimum error reporting requirements. The AER capability is defined for more robust error reporting and is implemented with a specific PCIe capability structure.

Both mechanisms are supported by the I350.

Also the *SERR# Enable* and the *Parity Error* bits from the legacy Command register take part in the error reporting and logging mechanism.

In a multi-Function device, PCI Express errors that are not related to any specific Function within the device, are logged in the corresponding status and logging registers of all functions in that device. These include the following cases of Unsupported Request (UR):

- A memory or I/O access that does not match any BAR for any function
- Messages.
- Configuration accesses to a non-existent function.

3.1.7.2 Error Events

Table 3-12 lists the error events identified by the I350 and the response in terms of logging, reporting, and actions taken. Consult the PCIe specification for the effect on the PCI Status register.

Table 3-12 Response and Reporting of PCIe Error Events

Error Name	Error Events	Default Severity	Action
PHY errors			
Receiver error	8b/10b decode errors Packet framing error	Correctable. Send ERR_CORR	TLP to initiate NAK and drop data. DLLP to drop.
Data link errors			
Bad TLP	• Bad CRC • Not legal EDB • Wrong sequence number	Correctable. Send ERR_CORR	TLP to initiate NAK and drop data.
Bad DLLP	• Bad CRC	Correctable. Send ERR_CORR	DLLP to drop.
Replay timer timeout	• REPLAY_TIMER expiration	Correctable. Send ERR_CORR	Follow LL rules.
REPLAY NUM rollover	• REPLAY NUM rollover	Correctable. Send ERR_CORR	Follow LL rules.
Data link layer protocol error	• Violations of Flow Control Initialization Protocol • Reception of NACK/ACK with no corresponding TLP	Uncorrectable. Send ERR_FATAL	Follow LL rules.
TLP errors			
Poisoned TLP received	• TLP with error forwarding	Uncorrectable. ERR_NONFATAL Log header	A poisoned completion is ignored and the request can be retried after timeout. If enabled, the error is reported.

Table 3-12 Response and Reporting of PCIe Error Events (Continued)

Error Name	Error Events	Default Severity	Action
Unsupported Request (UR)	• Wrong config access • MRdLk • Configuration request type 1 • Unsupported vendor Defined type 0 message • Not valid MSG code • Not supported TLP type • Wrong function number • Received TLP outside address range	Uncorrectable. ERR_NONFATAL Log header	Send completion with UR.
Completion timeout	• Completion timeout timer expired	Uncorrectable. ERR_NONFATAL	Error is non-fatal (default case) • Send error message if advisory • Retry the request once and send advisory error message on each failure • If fails, send uncorrectable error message Error is defined as fatal • Send uncorrectable error message
Completer abort	• Received target access with data size > 64-bit	Uncorrectable. ERR_NONFATAL Log header	Send completion with CA.
Unexpected completion	• Received completion without a request for it (tag, ID, etc.)	Uncorrectable. ERR_NONFATAL Log header	Discard TLP.
Receiver overflow	• Received TLP beyond allocated credits	Uncorrectable. ERR_FATAL	Receiver behavior is undefined.
Flow control protocol error	• Minimum initial flow control advertisements • Flow control update for infinite credit advertisement	Uncorrectable. ERR_FATAL	Receiver behavior is undefined. The I350 doesn’t report violations of Flow Control initialization protocol
Malformed TLP (MP)	• Data payload exceed Max_Payload_Size • Received TLP data size does not match length field • TD field value does not correspond with the observed size • Power management messages that doesn’t use TC0. • Usage of unsupported VC.	Uncorrectable. ERR_FATAL Log header	Drop the packet and free FC credits.
Completion with unsuccessful completion status		No action (already done by originator of completion).	Free FC credits.
Byte count integrity in completion process.	When byte count isn’t compatible with the length field and the actual expected completion length. For example, length field is 10 (in Dword), actual length is 40, but the byte count field that indicates how many bytes are still expected is smaller than 40, which is not reasonable.	No action	The I350 doesn't check for this error and accepts these packets. This may cause a completion timeout condition.

3.1.7.3 Error Forwarding (TLP poisoning)

If a TLP is received with an error-forwarding trailer, the transaction can be re-sent a number of times as programmed in the GCR register. If transaction still fails the packet is dropped and is not delivered to its destination. The I350 then reacts as described in Table 3-12.

The I350 does not initiate any additional master requests for that PCI function until it detects an internal software reset for associated LAN port. Software is able to access device registers after such a fault.

System logic is expected to trigger a system-level interrupt to inform the operating system of the problem. Operating systems can then stop the process associated with the transaction, re-allocate memory instead of the faulty area, etc.

3.1.7.4 ECRC

The I350 supports End to End CRC (ECRC) as defined in the PCIe spec. The following functionality is provided:

- Insertion of ECRC in all transmitted TLPs:
  - The I350 indicates support for insertion of ECRC in the ECRC Generation Capable bit of the PCIe configuration registers. This bit is loaded from the ECRC Generation EEPROM bit.
  - Insertion of ECRC is enabled by the ECRC Generation Enable bit of the PCIe configuration registers.
- ECRC is checked on all incoming TLPs. A packet received with an ECRC error is dropped. Note that for completions, a completion timeout will occur later (if enabled), which would result in re-issuing the request.
  - The I350 indicates support for ECRC checking in the ECRC Check Capable bit of the PCIe configuration registers. This bit is loaded from the ECRC Check EEPROM bit.
  - Checking of ECRC is enabled by the ECRC Check Enable bit of the PCIe configuration registers.
- ECRC errors are reported.
- System SW may configure ECRC independently per each LAN function.

3.1.7.5 Partial Read and Write Requests

Partial Memory Accesses

The I350 has limited support of reads and writes requests with only part of the byte enable bits set:

- Partial writes with at least one byte enabled are silently dropped.
- Zero-length writes has no internal impact (nothing written, no effect such as clear-by-write). The transaction is treated as a successful operation (no error event).
- Partial reads with at least one byte enabled are handled as a full read. Any side effect of the full read (such as clear by read) is also applicable to partial reads.
- Zero-length reads generate a completion, but the register is not accessed and undefined data is returned.

The I350 does not generate an error indication in response to any of the above events.

Partial I/O Accesses

- Partial access on address

  - — A write access is discarded
  - — A read access returns 0xFFFF
- Partial access on data, where the address access was correct
  - — A write access is discarded
  - — A read access performs the read

3.1.7.6 Error Pollution

Error pollution can occur if error conditions for a given transaction are not isolated on the error's first occurrence. If the Physical layer detects and reports a receiver error, to avoid having this error propagate and cause subsequent errors at upper layers the same packet is not signaled at the data link or transaction layers.

Similarly, when the data link layer detects an error, subsequent errors that occur for the same packet are not signaled at the transaction layer.

3.1.7.7 Completion with Unsuccessful Completion Status

A completion with unsuccessful completion status is dropped and not delivered to its destination. An interrupt is generated to indicate unsuccessful completion.

3.1.7.8 Error Reporting Changes

The Rev. 1.1 specification defines two changes to advanced error reporting. A new *Role-Based Error Reporting* bit in the *Device Capabilities* register is set to 1b to indicate that these changes are supported by the I350. These changes are:

1. Setting the *SERR# Enable* bit in the PCI Command register also enables UR reporting (in the same manner that the *SERR# Enable* bit enables reporting of correctable and uncorrectable errors). In other words, the *SERR# Enable* bit overrides the UR *Error Reporting Enable* bit in the PCIe Device Control register.
2. Changes in the response to some uncorrectable non-fatal errors, detected in non-posted requests to the I350. These are called advisory Non-fatal error cases. For each of the errors that follow, the following behavior is defined:
   a. The *Advisory Non-Fatal Error Status* bit is set in the Correctable Error Status register to indicate the occurrence of the advisory error and the *Advisory Non-Fatal Error Mask* corresponding bit in the Correctable Error Mask register is checked to determine whether to proceed further with logging and signaling.
   b. If the *Advisory Non-Fatal Error Mask* bit is clear, logging proceeds by setting the corresponding bit in the Uncorrectable Error Status register, based upon the specific uncorrectable error that's being reported as an advisory error. If the corresponding uncorrectable error bit in the Uncorrectable Error Mask register is clear, the First Error Pointer and Header Log registers are updated to log the error, assuming they are not still occupied by a previously unserviced error.
   c. An ERR_COR message is sent if the *Correctable Error Reporting Enable* bit is set in the Device Control register. An ERROR_NONFATAL message is not sent for this error.

The following uncorrectable non-fatal errors are considered as advisory non-fatal Errors:

- A completion with an Unsupported Request or Completer Abort (UR/CA) status that signals an uncorrectable error for a non-posted request. If the severity of the UR/CA error is non-fatal, the completer must handle this case as an advisory non-fatal error.

- When the requester of a non-posted request times out while waiting for the associated completion, the requester is permitted to attempt to recover from the error by issuing a separate subsequent request, or to signal the error without attempting recovery. The requester is permitted to attempt recovery zero, one, or multiple (finite) times, but must signal the error (if enabled) with an uncorrectable error message if no further recovery attempts are made. If the severity of the completion timeout is non-fatal and the requester elects to attempt recovery by issuing a new request, the requester must first handle the current error case as an advisory non-fatal error.
- Reception of a poisoned TLP. Refer to Section 3.1.7.3.
- When a receiver receives an unexpected completion and the severity of the unexpected completion error is non-fatal, the receiver must handle this case as an advisory non-fatal error.

3.1.7.9 Completion with Completer Abort (CA)

A DMA master transaction ending with a Completer Abort (CA) completion causes all PCIe master transactions to stop; the *PICAUSE.ABR* bit is set and an interrupt is generated if the appropriate mask bits are set. To enable PCIe master transactions following reception of a CA completion, software issues an FLR to the right function or a PCI reset to the device and re-initializes the function(s).

Note: Asserting *CTRL.DEV_RST* Flushes any pending transactions on the PCIe and reset's all ports.

3.1.8 PCIe Power Management

Described in Section 5.4.1 - Power Management.

3.1.9 PCIe Programming Interface

Described in Section 9 - PCIe Programming Interface

3.2 Management Interfaces

The I350 contains two possible interfaces to an external BMC.

- SMBus
- NC-SI

3.2.1 SMBus

SMBus is an optional interface for pass-through and/or configuration traffic between an external BMC and the I350. The SMBus channel behavior and the commands used to configure or read status from the I350 are described in Section 10.5.

The I350 also enables reporting and controlling the device using the MCTP protocol over SMBus. The MCTP interface will be used by the BMC to only control the NIC and not for pass through traffic. All network ports are mapped to a single MCTP endpoint on SMBus. For information, refer to Section 10.7.

3.2.1.1 Channel Behavior

The SMBus specification defines a maximum frequency of 100 KHz. However, the SMBus interface can be activated up to 400 KHz without violating any hold and setup time.

SMBus connection speed bits define the SMBus mode. Also, SMBus frequency support can be defined only from the EEPROM.

3.2.2 NC-SI

The NC-SI interface in the I350 is a connection to an external BMC defined by the DMTF NC-SI protocol. It operates as a single interface with an external BMC, where all traffic between the I350 and the BMC flows through the interface.

The I350 supports the standard DMTF NC-SI protocol for both pass-through and control traffic as defined in Section 10.6.

3.2.2.1 Electrical Characteristics

The I350 complies with the electrical characteristics defined in the NC-SI specification.

The I350 NC-SI behavior is configured on power-up in the following manner:

- The I350 provides an NC-SI clock output if defined by the *NC-SI Output Clock Disable* EEPROM bit (Section 6.2.22). The default value is to use an external clock source as defined in the NC-SI specification.
- The output driver strength for the NC-SI_CLK_OUT pad is configured by the *NC-SI Clock Pad Drive Strength* bit (default = 0b) in the *Functions Control* EEPROM word (refer to Section 6.2.22).
- The output driver strength for the NC-SI output signals (NC-SI_DV & NC-SI_RX) is configured by the EEPROM *NC-SI Data Pad Drive Strength* bit (default = 0b; refer to Section 6.2.22).
- The *Multi-Drop NC-SI* EEPROM bit (refer to Section 6.3.7.3) defines the NC-SI topology (point-to-point or multi-drop; the default is multi-drop).

The I350 can provide an NC-SI clock output as previously mentioned. The NC-SI clock input (NC-SI_CLK_IN) serves as an NC-SI input clock in either case. That is, if the I350 provides an NC-SI output clock, the platform is required to route it back through the NC-SI clock input with the correct latency. Refer to Chapter 11, "Electrical/Mechanical Specification" for details.

The I350 dynamically drives its NC-SI output signals (NC-SI_DV and NC-SI_RX) as required by the sideband protocol:

- On power-up, the I350 floats the NC-SI outputs except for NCSI_CLK_OUT.
- If the I350 operates in point-to-point mode, then the I350 starts driving the NC-SI outputs some time following power-up.
- If the I350 operates in a multi-drop mode, the I350 drives the NC-SI outputs as configured by the BMC.

3.2.2.2 NC-SI Transactions

The NC-SI link supports both pass-through traffic between the BMC and the I350 LAN functions, as well as configuration traffic between the BMC and the I350 internal units as defined in the NC-SI protocol. Refer to Section 10.6.2for information.

3.3 Flash / EEPROM

3.3.1 EEPROM Interface

3.3.1.1 General Overview

The I350 uses an EEPROM device for storing product configuration information. The EEPROM is divided into three general regions:

- Hardware accessed - Loaded by the I350 after power-up, PCI reset de-assertion, D3 ->D0 transition, or a software-commanded EEPROM read (*CTRL_EXT.EE_RST*).
- Manageability firmware accessed - Loaded by the I350 in pass-through mode after power-up or firmware reset.
- Software accessed - Used only by software. The meaning of these registers, as listed here, is a convention for software only and is ignored by the I350.

Table 3-13 lists the structure of the EEPROM image in the I350.

Table 3-13 EEPROM Structure

Address	Content
0x0 – 0x9	LAN 0 MAC address and software area
0xA – 0x2F	LAN 0 and Common hardware area
0x30 – 0x3E	PXE area
0x3F	Software Checksum, for Words 0x00 - 0x3E
0x40 – 0x4F	Software area
0x50 – 0x7F	FW pointers
0x80 -0xBF	LAN 1 hardware area (with SW checksum in 0xBF)
0xC0 - 0xFF	LAN 2 hardware area (with SW checksum in 0xFF)
0x100 - 0x13F	LAN 3 hardware area (with SW checksum in 0x13F)
…	
	Firmware structures
…	
	VPD area
…	
	CSR and Analog configuration (PCIe/PHY/PLL/SerDes structures)

The EEPROM mapping is described in Chapter 6.

3.3.1.2 EEPROM Device

The EEPROM interface supports an SPI interface and expects the EEPROM to be capable of 2 MHz operation.

The I350 is compatible with various sizes of 4-wire serial EEPROM devices. If pass-through mode functionality is desired, up to 256 Kbits serial SPI compatible EEPROM can be used. If no manageability mode is desired, a 128 Kbit (16 Kbyte) serial SPI compatible EEPROM can be used. All EEPROM's are accessed in 16-bit words although the EEPROM interface is designed to also accept 8-bit data accesses.

The I350 automatically determines the address size to be used with the SPI EEPROM it is connected to and sets the EEPROM address size field of the *EEPROM/FLASH Control and Data* register (*EEC.EE_ADDR_SIZE*) field appropriately. Software can use this size to determine how to access the EEPROM. The exact size of the EEPROM is determined within one of the EEPROM words.

Note: The different EEPROM sizes have two differing numbers of address bits (8 bits or 16 bits), and therefore must be accessed with a slightly different serial protocol. Software must be aware of this if it accesses the EEPROM using direct access.

3.3.1.3 HW Initial Load Process.

Upon power on reset or PCIe reset, the I350 reads the global device parameters from the EEPROM including all the parameters impacting the content of the PCIe configuration space. Upon a software reset to one of the ports (*CTRL.RST* set to 1), a partial load is done of the parameters relevant to the port where the software reset occurred. Upon a software reset to all ports (*CTRL.DEV_RST* = 1) a partial load is done of the parameters relevant to all ports. Table 3-14 lists the words read in each EEPROM auto-read sequence. During full load after power-on all hardware related EEPROM words are loaded. Following a software reset only a subset of the hardware related EEPROM words are loaded. For details of the content of each word - refer to Chapter 6.

Notes:

1. LANx_start parameter in Table 3-14 relates to start of LAN related EEPROM section where:
   — LAN0_start = 0x0
   — LAN1_start = 0x80
   — LAN2_start = 0xC0
   — LAN3_start = 0x100
2. In the Dual port SKU and 25x25 package SKU EEPROM words related to ports 2 and 3 are not read during the Auto-load sequence.

Table 3-14 EEPROM Auto-Load Sequence

EEPROM Word	EEPROM Word Address	Full Load (Power-up)	Full Load No MGMT (PCI RST)	SW[1] reset port 0 Load	SW[1] reset port 1 Load	SW[1] reset port 2 Load	SW[1] reset port 3 Load
EEPROM sizing and protected fields	0x12[4]	Y	Y	Y	Y	Y	Y
PCIe PHY Auto Configuration Pointer and PCIe PHY Auto Configuration structures.	0x10	Y					
CSR Auto Configuration Power-Up LAN0	0x027	Y					
CSR Auto Configuration Power-Up LAN1	LAN1_start + 0x27	Y					
CSR Auto Configuration Power-Up LAN2	LAN2_start + 0x27	Y					
CSR Auto Configuration Power-Up LAN3	LAN3_start + 0x27	Y					
Init Control 1	0x0A	Y	Y				
PCIe init configuration 1	0x18	Y	Y				
PCIe init configuration 2	0x19	Y	Y				
PCIe init configuration 3	0x1A	Y	Y				

Table 3-14 EEPROM Auto-Load Sequence (Continued)

EEPROM Word	EEPROM Word Address	Full Load (Power-up)	Full Load No MGMT (PCI RST)	SW[1] reset port 0 Load	SW[1] reset port 1 Load	SW[1] reset port 2 Load	SW[1] reset port 3 Load
PCIe control 1	0x1B	Y	Y				
PCIe control 2	0x28	Y	Y				
PCIe control 3	0x29	Y	Y				
Functions control	0x21	Y	Y				
Device Rev ID	0x1E	Y	Y				
PCIe L1 Exit latencies	0x14	Y	Y				
PCIe completion timeout configuration	0x15	Y	Y				
Subsystem ID[2]	0x0B	Y	Y				
Subsystem Vendor ID[2]	0x0C	Y	Y				
Device ID - LAN 0[3]	0x0D	Y	Y				
Device ID - LAN 1[3]	LAN1_start + 0x0D	Y	Y				
Device ID - LAN 2[3]	LAN2_start + 0x0D	Y	Y				
Device ID - LAN 3[3]	LAN3_start + 0x0D	Y	Y				
Vendor ID - LAN 0[3]	0x0E	Y	Y				
Dummy function device ID[3]	0x1D	Y	Y				
MSI-X configuration LAN 0	0x16	Y	Y				
MSI-X configuration LAN 1	LAN1_start + 0x16	Y	Y				
MSI-X configuration LAN 2	LAN2_start + 0x16	Y	Y				
MSI-X configuration LAN 3	LAN3_start + 0x16	Y	Y				
LAN power consumption	0x22	Y	Y				
VPD Pointer to table	0x2F	Y	Y				
VPD table entry ID TAG	ID STRING	Y	Y				
VPD read or write area TAG	VPD TAG 1	Y	Y				
VPD read or write area length	VPD TAG 1 LENGTH	Y	Y				
VPD read or write area TAG	VPD TAG 2	Y	Y				
VPD read or write area length	VPD TAG 2 LENGTH	Y	Y				
VPD end TAG	VPD END	Y	Y				
Init Control 3 LAN 0	0x24	Y	Y				
Init Control 3 LAN 1	LAN1_start + 0x24	Y	Y				
Init Control 3 LAN 2	LAN2_start + 0x24	Y	Y				
Init Control 3 LAN 3	LAN3_start + 0x24	Y	Y				
LEDCTL 1 default LAN 0	0x1C	Y	Y				
LEDCTL 0 default LAN 0	0x1F	Y	Y				
LEDCTL 1 default LAN 1	LAN1_start + 0x1C	Y	Y				
LEDCTL 0 default LAN 1	LAN1_start + 0x1F	Y	Y				
LEDCTL 1 3 default LAN 2	LAN2_start + 0x1C	Y	Y				
LEDCTL 0 2 default LAN 2	LAN2_start + 0x1F	Y	Y				
LEDCTL 1 3 default LAN 3	LAN3_start + 0x1C	Y	Y				
LEDCTL 0 2 default LAN 3	LAN3_start + 0x1F	Y	Y				
End of read only area	0x2C	Y	Y				
Start of read only area	0x2D	Y	Y				

Table 3-14 EEPROM Auto-Load Sequence (Continued)

EEPROM Word	EEPROM Word Address	Full Load (Power-up)	Full Load No MGMT (PCI RST)	SW[1] reset port 0 Load	SW[1] reset port 1 Load	SW[1] reset port 2 Load	SW[1] reset port 3 Load
I/O Virtualization (IOV) Control	0x25	Y	Y				
IOV Device ID	0x26	Y	Y				
PCIe reset Configuration Pointer and PCIe reset CSR Auto Configuration structures - LAN 0	0x23	Y	Y				
PCIe reset Configuration Pointer and PCIe reset CSR Auto Configuration structures - LAN 1	LAN1_start + 0x23	Y	Y				
PCIe reset Configuration Pointer and PCIe reset CSR Auto Configuration structures - LAN 2	LAN2_start + 0x23	Y	Y				
PCIe reset Configuration Pointer and PCIe reset CSR Auto Configuration structures - LAN 3	LAN3_start + 0x23	Y	Y				
Port 0 Only Load							
Init Control 4 LAN 0	0x13	Y	Y	Y			
Init Control 2 LAN 0	0x0F	Y	Y	Y			
Ethernet address byte 2-1 - LAN 0	0x00	Y	Y	Y			
Ethernet address byte 4-3 - LAN 0	0x01	Y	Y	Y			
Ethernet address byte 6-5 - LAN 0	0x02	Y	Y	Y			
Software defined pins control - LAN0	0x20	Y	Y	Y			
SW Reset CSR Auto Configuration Pointer and SW Reset CSR Auto Configuration structures - LAN0	0x17	Y	Y	Y			
Watchdog Configuration	0x2E	Y	Y	Y			
Port 1 Only Load							
Init Control 4 LAN 1	LAN1_start + 0x13	Y	Y		Y		
Init Control 2 LAN 1	LAN1_start + 0x0F	Y	Y		Y		
Ethernet address byte 2-1 - LAN 1	LAN1_start + 0x00	Y	Y		Y		
Ethernet address byte 4-3 - LAN 1	LAN1_start + 0x01	Y	Y		Y		
Ethernet address byte 6-5 - LAN 1	LAN1_start + 0x02	Y	Y		Y		
Software defined pins control - LAN1	LAN1_start + 0x20	Y	Y		Y		
SW Reset CSR Auto Configuration Pointer and SW Reset CSR Auto Configuration structures - LAN1	LAN1_start + 0x17	Y	Y		Y		
Watchdog Configuration	0x2E	Y	Y		Y		
Port 2 Only Load							
Init Control 4 LAN 2	LAN2_start + 0x13	Y	Y			Y	
Init Control 2 LAN 2	LAN2_start + 0x0F	Y	Y			Y	
Ethernet address byte 2-1 - LAN 2	LAN2_start + 0x00	Y	Y			Y	
Ethernet address byte 4-3 - LAN 2	LAN2_start + 0x01	Y	Y			Y	
Ethernet address byte 6-5 - LAN 2	LAN2_start + 0x02	Y	Y			Y	
Software defined pins control - LAN2	LAN2_start + 0x20	Y	Y			Y	

Table 3-14 EEPROM Auto-Load Sequence (Continued)

EEPROM Word	EEPROM Word Address	Full Load (Power-up)	Full Load No MGMT (PCI RST)	SW[1] reset port 0 Load	SW[1] reset port 1 Load	SW[1] reset port 2 Load	SW[1] reset port 3 Load
SW Reset CSR Auto Configuration Pointer and SW Reset CSR Auto Configuration structures - LAN2	LAN2_start + 0x17	Y	Y			Y	
Watchdog Configuration	0x2E	Y	Y			Y	
Port 3 Only Load							
Init Control 4 LAN 3	LAN3_start + 0x13	Y	Y				Y
Init Control 2 LAN 3	LAN3_start + 0x0F	Y	Y				Y
Ethernet address byte 2-1 - LAN 3	LAN3_start + 0x00	Y	Y				Y
Ethernet address byte 4-3 - LAN 3	LAN3_start + 0x01	Y	Y				Y
Ethernet address byte 6-5 - LAN 3	LAN3_start + 0x02	Y	Y				Y
Software defined pins control - LAN3	LAN3_start + 0x20	Y	Y				Y
SW Reset CSR Auto Configuration Pointer and SW Reset CSR Auto Configuration structures - LAN3	LAN3_start + 0x17	Y	Y				Y
Watchdog Configuration	0x2E	Y	Y				Y
Management Section[4]							
Management Pass Through LAN Configuration Pointer - LAN0	0x11	Y					
Management Pass Through LAN Configuration Pointer - LAN1	LAN1_start + 0x11	Y					
Management Pass Through LAN Configuration Pointer - LAN2	LAN2_start + 0x11	Y					
Management Pass Through LAN Configuration Pointer - LAN3	LAN3_start + 0x11	Y					

1. Upon assertion of *CTRL.DEV_RST* by software partial load of parameters relevant to all ports is done. Assertion of *CTRL_EXT.EE_RST* causes load of per port parameters similar to *CTRL.RST*.
2. Loaded only if load subsystem ID bit is set
3. Loaded only if load device ID bit is set
4. EEPROM words listed under Management Section are also loaded following Firmware Reset in addition to word 0x12 (read by HW).

3.3.1.4 Software Accesses

The I350 provides two different methods for software access to the EEPROM. It can either use the built-in controller to read the EEPROM or access the EEPROM directly using the EEPROM's 4-wire interface.

In addition, the VPD area of the EEPROM can be accessed via the VPD capability structure of the PCIe.

Software can use the EEPROM Read (*EERD*) register to cause the I350 to read a word from the EEPROM that the software can then use. To do this, software writes the address to read to the *Read Address* (*EERD.ADDR*) field and simultaneously writes a 1b to the *Start Read* bit (*EERD.START*). The I350 reads the word from the EEPROM, sets the *Read Done* bit (*EERD.DONE*), and places the data in the *Read Data* field (*EERD.DATA*). Software can poll the EEPROM Read register until it sees the *Read Done* bit set and then uses the data from the *Read Data* field. Any words read this way are not written to the I350's internal registers.

Software can also directly access the EEPROM's 4-wire interface through the EEPROM/Flash Control (*EEC*) register. It can use this for reads, writes, or other EEPROM operations.

To directly access the EEPROM, software should follow these steps:

1. Take ownership of the EEPROM Semaphore bit as described in Section 4.7.1.
2. Write a 1b to the *EEPROM Reques*t bit (*EEC.EE_REQ*).
3. Poll the *EEPROM Grant* bit (*EEC.EE_GNT*) until it becomes 1b. It remains 0b as long as the hardware is accessing the EEPROM.
4. Write or read the EEPROM using the direct access to the 4-wire interface as defined in the EEPROM/Flash Control and Data (*EEC*) register. The exact protocol used depends on the EEPROM placed on the board and can be found in the appropriate datasheet.
5. Write a 0b to the *EEPROM Request* bit (*EEC.EE_REQ*) to enable EEPROM access by other drivers.

Notes: If direct access via the EEPROM's 4-wire interface to a read protected area is attempted, the I350 blocks the access and sets the *EEC.EE_BLOCKED* bit. To clear the block condition and enable further access to the EEPROM software should write 1b to the *EEC.EE_CLR_ERR* bit.

Following execution of an EEPROM write operation using direct access software should verify that the write operation has completed and EEPROM status is ready before clearing the *EEC.EE_REQ* and *EEC.EE_GNT* bits.

Finally, software can cause the I350 to re-read the per-function hardware accessed fields of the EEPROM (setting the I350's internal registers appropriately similar to software reset) by writing a 1b to the *EEPROM Reset* bit of the Extended Device Control register (*CTRL_EXT.EE_RST*).

Note: If the EEPROM does not contain a valid signature (refer to Section 3.3.1.5), the I350 assumes 16-bit addressing. In order to access an EEPROM that requires 8-bit addressing, software must use the direct access mode.

3.3.1.5 EEPROM Detection and Signature Field

The I350 supports detection of EEPROM existence following power-up and detection of a valid EEPROM image via the EEPROM signature field in the *Sizing and Protected Fields* EEPROM word (refer to Section 6.2.9).

3.3.1.5.1 EEPROM Detection

The I350 will check if an EEPROM is connected following power-up by sending a get status command to the EEPROM. If the EEPROM response is correct, the I350 will set the *EEC.EE_DET* bit to 1b. If EEPROM status received is incorrect, the I350 assumes that there is no EEPROM connected, clears the *EEC.EE_DET* bit to 0b and works in EEPROM-less mode. The I350 will not attempt any further EEPROM auto-load operations as defined in Section 3.3.1.3 if the *EEC.EE_DET* bit is cleared to 0b after attempting an auto-load operation following power-up. Even if an EEPROM with a valid image is later connected, the I350 will not attempt an auto-read until a full power-up cycle is performed.

Note: When an incorrect EEPROM status is read after power-up the EEPROM is still accessible to software via the *EERD* register or by issuing bit banging operations using the *EEC* register, however EEPROM auto-load following reset as defined in Section 3.3.1.3 is not executed.

3.3.1.5.2 Detection of Valid EEPROM Image

Following the various resets before executing an EEPROM auto-load operation as defined in Section 3.3.1.3 the I350 determines if a valid EEPROM image is present by first reading the EEPROM signature in the *Sizing and Protected Fields* EEPROM word at address 0x12. It checks the signature value in bits 15 and 14 of the EEPROM word. If bit 15 is 0b and bit 14 is 1b, it considers the EEPROM image valid, the *EEC.EE_PRES* bit is set to 1 to indicate that a valid signature was detected and

additional EEPROM words are read to program its internal registers as defined in Section 3.3.1.3. Otherwise, it ignores the value read from the *EEPROM Sizing and Protected Fields* word at address 0x12, clears the *EEC.EE_PRES* bit to 0 and does not read any other words as part of the auto-load process.

Note: Following the various resets the I350 executes an EEPROM auto-load operation as defined in Section 3.3.1.3 if bit 15 in the *EEPROM Sizing and Protected Fields* word is read as 1b or bit 14 is read as 0b, the I350 assumes that the EEPROM image is not valid and does not continue the auto-load process. If a valid image is later programmed, the I350 will attempt to do an auto-load operation following the various reset assertions and set the *EEC.EE_PRES* bit is set to 1 if a valid signature is read.

3.3.1.6 Protected EEPROM Space

The I350 provides a mechanism for a hidden area in the EEPROM to the host. The hidden area cannot be accessed (read or written to) via the EEPROM registers in the CSR space. It can be accessed only by the manageability subsystem. This area is located at the end of the EEPROM memory. its size is defined by the HEPSize field in EEPROM word 0x12.

Note: Current I350 manageability firmware does not use any hidden area protected by the HEPSize mechanism.

A mechanism to protect part of the EEPROM from host writes and the VPD area from host writes is also provided. This mechanism is controlled by words 0x2D and 0x2C that define the start and the end of the read-only area and bit 4 (enable protection) of EEPROM word 0x12 that enables the mechanism.

3.3.1.6.1 Initial EEPROM Programming

In most applications, initial EEPROM programming is done directly on the EEPROM pins. Nevertheless, it is desired to enable existing software utilities (accessing the EEPROM via the host interface) to initially program the entire EEPROM without breaking the protection mechanism. Following a power-up sequence, the I350 reads the hardware initialization words in the EEPROM. If the signature in word 0x12 does not equal 01b, the EEPROM is assumed as non-programmed. There are two outcomes of a non-valid signature:

- The I350 does not read any further EEPROM data and sets the relevant registers to default.
- The I350 enables access to any location in the EEPROM via the EEPROM *EERD* and *ECC* registers.

3.3.1.6.2 Activating the Protection Mechanism

Following initialization, the I350 reads the EEPROM and turns on the protection mechanism if word 0x12 contains a valid signature (equals 01b) and bit 4 (enable protection) of word 0x12 is set. Once the protection mechanism is turned on, words 0x12, 0x2C, 0x2D and 0x2F (VPD pointer) become write-protected, the area that is defined by word 0x12 becomes hidden (read/write protected) and the area defined by words 0x2C and 0x2D and the VPD area becomes write protected.

- No matter what is designated as the read only protected area, words 0x30:0x3F (used by PXE driver) are writable, unless the area is defined as hidden or is part of the VPD area.

3.3.1.6.3 Non Permitted Accessing to Protected Areas in the EEPROM

This paragraph refers to EEPROM accesses via the EEC (bit banging) or EERD (parallel read access) registers. Following a write access to the protected areas in the EEPROM, hardware responds properly on the PCIe interface but does not initiate any access to the EEPROM. Following a read access to the hidden area in the EEPROM (as defined by word 0x12), hardware does not access the EEPROM and returns meaningless data to the host.

Note: Using bit banging, the SPI EEPROM can be accessed in a burst mode. For example, providing op-code, address, and then read or write data for multiple bytes. Hardware inhibits any attempt to access the protected EEPROM locations even in burst accesses.

Software should not access the EEPROM in a burst-write mode starting in a non-protected area and continue to a protected one. In such a case it is not guaranteed that the write access to any area ever takes place.

3.3.1.7 EEPROM Recovery

The EEPROM contains fields that if programmed incorrectly might affect the functionality of the I350. The impact can range from an incorrect setting of some function (such as LED programming), via disabling of entire features (such as no manageability) and link disconnection, to the inability to access the I350 via the regular PCIe interface.

The I350 implements a mechanism that enables recovery from a faulty EEPROM no matter what the impact is, using an SMBus message that instructs firmware to invalidate the EEPROM.

This mechanism uses a SMBus message that the firmware is able to receive in all modes when a EEPROM with a valid signature is detected, no matter what the content of the EEPROM is (even in diagnostic mode). After receiving this kind of message, firmware clears EEPROM in word 0x12 together with the signature (bits 15/14 to 00b). Afterwards, the BIOS/operating system initiates a reset to force an EEPROM auto-load process that fails in order to enable access to the I350. At this stage software can now re-program the EEPROM using the *EEC* register (refer to Section 3.3.1.4).

Firmware is programmed to receive such a command only from a PCIe reset until one of the functions changes its status from D0u to D0a. Once one of the functions moves to D0a, it can be safely assumed that the I350 is accessible to the host and there is no further need for this function. This reduces the possibility of malicious software using this command as a back door and limits the time firmware must be active in non-manageability mode.

The command is sent on a fixed SMBus address of 0xC8. The format of the command is the SMBus Block write as follows:

Table 3-15 Command Format

Function	Command	Byte Count	Data Byte
Release EEPROM	0xC7	0x01	0xAA

Notes:

1. This solution requires a controllable SMBus connection to the I350.
2. If more than one I350 part is in a state to accept this command, all of the devices in this state will respond with an ACK to this command and accept it. A device with one of its ports in D0a should not respond with an ACK to this command if not in D0u state.
3. The I350 is guaranteed to accept the command on the SMBus interface and on address 0xC8. If one of the functions is not in D0u state, the I350 will not accept the command and will return a NACK. If the firmware has a station address, it may answer on this address also. If the SMBus address is

different from 0xC8 or the station address, then the I350 disregards the command but an ACK is returned.

4. When one of the functions is not in D0u state the NACK is sent at end of the command. A ACK is always returned after the address.
5. After receiving a release EEPROM command, firmware should keep its current state. It is the responsibility of the programmer that is updating the EEPROM to send a firmware reset (if required) after the full EEPROM update process completes.

3.3.1.8 EEPROM-Less Support

The I350 supports EEPROM-less operation with the following limitations:

- Non-manageability mode only.
- No support for legacy Wake on LAN (magic packets).
- No support for Flash storage of option ROM code (such as PXE) on a NIC.
- No legacy option ROM support (PXE, iSCSI Boot, etc.) is available on NIC or LOM.
- No support for serial ID PCIe capability.
- The EEPROM images released by Intel contains a set of configuration defaults overrides. All initialization values usually taken from the EEPROM should be done by the host driver.

3.3.1.8.1 Access to the EEPROM Controlled Feature

The *EEARBC* register (refer to Section 8.4.5) enables access to registers that are not accessible via regular CSR access (such as PCIe configuration read-only registers) by emulating the auto-read process. *EEARBC* contains six strobe fields that emulate the internal strobes of the internal auto-read process. This register is common to all functions and should be accessed only after verifying that it's not being accessed by other functions.

Writing to one of the EEPROM words is executed in two steps:

1. Write 0x0 to the *EEARBC* register.
2. Program the *EEARBC* register with the EEPROM address (*EEARBC.ADDR*), Data to be written (*EEARBC.DATA*) and set the relevant Strobe bits (*EEARBC.VALID_**) as defined in Table 3-16.

Table 3-16 lists the strobe to be used when emulating a read of a specific word of the EEPROM auto-read feature.

Table 3-16 Strobes for EEARBC Auto-Read Emulation

EEPROM Word Emulated (In Hex)	Content	Port 0 Strobe	Port 1 Strobe	Port 2 Strobe	Port 3 Strobe
0:2	MAC address	VALID_CORE0	N/A	N/A	N/A
LAN1_start + 0:2		N/A	VALID_CORE1	N/A	N/A
LAN2_start + 0:2		N/A	N/A	VALID_CORE2	N/A
LAN3_start + 0:2		N/A	N/A	N/A	VALID_CORE3
0A	Init control 1	VALID_CORE0	VALID_CORE1	VALID_CORE2	VALID_CORE3
0B/0C/0E[1]	Sub-system device and vendor	VALID_COMMON	VALID_COMMON	VALID_COMMON	VALID_COMMON
1E/1D[2]	Dummy device ID, Rev ID	VALID_COMMON	VALID_COMMON	VALID_COMMON	VALID_COMMON
21	Function control	VALID_COMMON	VALID_COMMON	VALID_COMMON	VALID_COMMON

Table 3-16 Strobes for EEARBC Auto-Read Emulation (Continued)

EEPROM Word Emulated (In Hex)	Content	Port 0 Strobe	Port 1 Strobe	Port 2 Strobe	Port 3 Strobe
0D[2]	Device ID port 0	VALID_CORE0	N/A	N/A	N/A
LAN1_start + 0D[2]	Device ID port 1	N/A	VALID_CORE1	N/A	N/A
LAN2_start + 0D[2]	Device ID port 2	N/A	N/A	VALID_CORE2	N/A
LAN3_start + 0D[2]	Device ID port 3	N/A	N/A	N/A	VALID_CORE3
20	SDP control LAN 0	VALID_CORE0	N/A	N/A	N/A
LAN1_start + 20	SDP control LAN 1	N/A	VALID_CORE1	N/A	N/A
LAN2_start + 20	SDP control LAN 2	N/A	N/A	VALID_CORE2	N/A
LAN3_start + 20	SDP control LAN 3	N/A	N/A	N/A	VALID_CORE3
0F/24/13	Init control 2/3/4	VALID_CORE0	N/A	N/A	N/A
LAN1_start + 0F/24/13	Init control 3/4	N/A	VALID_CORE1	N/A	N/A
LAN2_start + 0F/24/13	Init control 3/4	N/A	N/A	VALID_CORE2	N/A
LAN3_start + 0F/24/13	Init control 3/4	N/A	N/A	N/A	VALID_CORE3
14/15[3]/16/18/19/1A/ 1B/22/25/2628/29/2A/ 2B	PCIe and NC-SI configuration	VALID_COMMON	VALID_COMMON	VALID_COMMON	VALID_COMMON
1C/1F[4]	LED control port 0	VALID_CORE0	N/A	N/A	N/A
LAN1_start + 1C/1F[4]	LED control port 1	N/A	VALID_CORE1	N/A	N/A
LAN2_start + 1C/1F[4]	LED control port 2	N/A	N/A	VALID_CORE2	N/A
LAN3_start + 1C/1F[4]	LED control port 3	N/A	N/A	N/A	VALID_CORE3
2E[4]	Watchdog configuration	VALID_CORE0	VALID_CORE1	VALID_CORE2	VALID_CORE3
2F[5]	VPD area	N/A	N/A	N/A	N/A

1. If word 0xA was accessed before the subsystem or subvendor ID are set, care must be taken that the load Subsystem IDs bit in word 0xA is set.
2. If word 0xA was accessed before one of the device IDs is set, care must be taken that the load Device IDs bit in word 0xA is set.
3. For the write of EEPROM word 0x15 to take effect a software reset needs to be issued following the write.
4. Part of the parameters that can be configured through the *EEARBC* register can be directly set through regular registers and thus usage of this mechanism is not needed for them. Specifically, words 0x1C, 0x1F and 0x2E control parameters that can be set through regular registers.
5. In EEPROM-less mode VPD is not supported.

3.3.2 Shared EEPROM

The I350 uses a single EEPROM device to configure hardware default parameters for all LAN devices, including Ethernet Individual Addresses (IA), LED behaviors, receive packet filters for manageability, wake-up capability, etc. Certain EEPROM words are used to specify hardware parameters that are LAN device-independent (such as those that affect circuit behavior). Other EEPROM words are associated with a specific LAN device. All LAN devices access the EEPROM to obtain their respective configuration settings.

3.3.2.1 EEPROM Deadlock Avoidance

The EEPROM is a shared resource between the following clients:

- Hardware auto-read.
- Port 0 LAN driver accesses.
- Port 1 LAN driver accesses.
- Port 2 LAN driver accesses.

- Port 3 LAN driver accesses.
- Firmware accesses.

All clients can access the EEPROM using parallel access, where hardware implements the actual access to the EEPROM. Hardware can schedule these accesses so that all clients get served without starvation.

However, software and hardware clients can access the EEPROM using bit banging. In this case, there is a request/grant mechanism that locks the EEPROM to the exclusive usage of one client. If this client is stuck (without releasing the lock), the other clients are not able to access the EEPROM. In order to avoid this, the I350 implements a timeout mechanism, which releases ownership of a client that didn't toggle the EEPROM bit-bang interface for more than two seconds. When the timeout mechanism is activated the *EEC.EE_REQ* and *EEC.EE_GNT* bits of the offending port are cleared. To initiate a new bit banging access SW will need to re-assert the *EEC.EE_REQ* bit. The EEPROM deadlock avoidance mechanism is enabled when the *Deadlock Timeout Enable* bit in the *Initialization Control Word 1* EEPROM word is set to 1.

Note: If an agent that was granted access to the EEPROM for bit-bang access didn't toggle the bit bang interface for 500 ms, it should check if it still owns the interface and is not blocked before continuing the bit-banging.

When Hardware EEPROM bit-bang access is aborted due to Deadlock avoidance or management reset the *EEC.EE_ABORT* bit is set. To clear the block condition and enable further access to the EEPROM, software should write 1b to the *EEC.EE_CLR_ERR* bit.

3.3.2.2 EEPROM Map Shared Words

The EEPROM map in Section 6.1 identifies those words configuring either LAN devices or the entire I350 component as “all”. Those words configuring a specific LAN device parameter are identified by their LAN number.

The following EEPROM words warrant additional notes specifically related to quad-LAN support:

Table 3-17 Notes on EEPROM Words

Initialization Control 1, (shared between LANs)	This EEPROM word specifies hardware-default values for parameters that apply a single value to all LAN devices, such as link configuration parameters required for auto-negotiation, wake-up settings, PCIe bus advertised capabilities, etc.
Initialization Control 2 Initialization Control 3, Initialization Control 4 (unique to each LAN)	These EEPROM words configure default values associated with each LAN device’s hardware connections, including which link mode (internal PHY, SGMII, SerDes, 1000BASE-BX, 1000BASE-KX) is used with this LAN device. Because a separate EEPROM word configures the defaults for each LAN, extra care must be taken to ensure that the EEPROM image does not specify a resource conflict.

3.3.3 Vital Product Data (VPD) Support

The EEPROM image might contain an area for VPD. This area is managed by the OEM vendor and doesn’t influence the behavior of hardware. Word 0x2F of the EEPROM image contains a pointer to the VPD area in the EEPROM. A value of 0xFFFF means VPD is not supported and the VPD capability doesn’t appear in the configuration space.

The maximum area size is 256 bytes but can be smaller. The VPD block is built from a list of resources. A resource can be either large or small. The structure of these resources is listed in the following tables.

Table 3-18 Small Resource Structure

Offset	0	1 - n
Content	Tag = 0xxx, xyyyb (Type = Small(0), Item Name = xxxx, length = yyy bytes)	Data

Table 3-19 Large Resource Structure

Offset	0	1 - 2	3 - n
Content	Tag = 1xxx, xxxxb (Type = Large(1), Item Name = xxxxxxx)	Length	Data

The I350 parses the VPD structure during the auto-load process (power up and PCIe reset or warm reset) in order to detect the read-only and read/write area boundaries. The I350 assumes the following VPD structure:

Table 3-20 VPD Structure

Tag	Structure Type	Length (Bytes)	Data	Resource Description
0x82	Large	Length of identifier string	Identifier	Identifier string.
0x90	Large	Length of RO area	RO data	VPD-R list containing one or more VPD keywords This part is optional and might not appear.
0x91	Large	Length of R/W area	RW data	VPD-W list containing one or more VPD keywords. This part is optional and might not appear.
0x78	Small	N/A	N/A	End tag.

Note: The VPD-R and VPD-W structures can be in any order.

If the I350 doesn't detect a value of 0x82 in the first byte of the VPD area, or the structure doesn't follow the description listed in Table 3-20, it assumes the area is not programmed and the entire 256 bytes area is read only. If a VPD-W tag is found after the VPD-R tag, the area defined by its size is writable via the VPD structure. Refer to the PCI 3.0 specification (Appendix I) for details of the different tags.

In any case, the VPD area is accessible for read and write via the regular EEPROM mechanisms pending the EEPROM protection capabilities enabled. For example, if VPD is in the protected area, the VPD area is not accessible to the software device driver (parallel or serial), but accessible through the VPD mechanism. If the VPD area is not in the protected area, then the software device driver can access all of it for read and write.

The VPD area can be accessed through the PCIe configuration space VPD capability structure described in Section 9.5.5. Write accesses to a read-only area or any access outside of the VPD area via this structure are ignored.

Note: Write access to Dwords, which are only partially in the read/write area, are ignored. It is responsibility of VPD software to make the right alignment to enable a write to the entire area.

3.3.4 Flash Interface

3.3.4.1 Flash Interface Operation

The I350 provides two different methods for software access to the Flash.

Using the legacy Flash transactions, the Flash is read from or written to each time the host CPU performs a read or a write operation to a memory location that is within the Flash address mapping or after a re-boot via accesses in the space indicated by the Expansion ROM Base Address register. All accesses to the Flash require the appropriate command sequence for the device used. Refer to the specific Flash data sheet for more details on reading from or writing to Flash. Accesses to the Flash are based on a direct decode of CPU accesses to a memory window defined in either:

1. The I350's Flash Base Address register (PCIe Control register at offset 0x10 and 0x14. Refer to Section 9.4.11).
2. The Expansion ROM Base Address register (PCIe Control register at offset 0x30. Refer to Section 9.4.15).

The I350 controls accesses to the Flash when it decodes a valid access.

Notes:

1. Flash read accesses are assembled by the I350 each time the read access is greater than a byte-wide access.
2. Flash read access times is in the order of 2 µs (Depending on Flash specification).
3. Flash write access times can be in the order of 2 µs to 200 µs (Depending on Flash specification). Following a write access to the Flash Software should avoid initiating any read or write access to the device until the Flash write access completed.
4. The I350 supports only byte writes to the Flash.

Another way for software to access the Flash is directly using the Flash's 4-wire interface through the Flash Access (*FLA*) register. It can use this for reads, writes, or other Flash operations (accessing the Flash status register, erase, etc.).

To directly access the Flash, software should follow these steps:

1. Take ownership of the Flash Semaphore bit as described in Section 4.7.1.
2. Write a 1b to the *Flash Request* bit (*FLA.FL_REQ*).
3. Read the *Flash Grant* bit (*FLA.FL_GNT*) until it becomes 1b. It remains 0b as long as there are other accesses to the Flash.
4. Write or read the Flash using the direct access to the 4-wire interface as defined in the *FLA* register. The exact protocol used depends on the Flash placed on the board and can be found in the appropriate datasheet.
5. Write a 0b to the *Flash Request* bit (*FLA.FL_REQ*).

Note: When Hardware Flash bit-bang access is aborted due to Deadlock avoidance the *FLA.FLA_ABORT* bit is set. To clear the block condition and enable further access to the Flash, software should write 1b to the *FLA.FLA_CLR_ERR* bit.

3.3.4.2 Flash Write Control

The Flash is write controlled by the *FWE* bits in the *EEPROM/FLASH Control and Data* (*EEC*) register.

After sending one byte write to the Flash, software should wait 2 µs to 200 µs (Depending on Flash specification) for the write access to complete before initiating the next Flash byte write access or before accessing any other register.

3.3.4.3 Flash Erase Control

When software needs to erase the Flash, it should set bit *FLA.FL_ER* in the *FLA* register to 1b (Flash erase) and then set bits *EEC.FWE* in the *EEPROM/Flash Control* register to 0b.

Hardware gets this command and sends the Erase command to the Flash. The erase process finishes by itself. Software should wait for the end of the erase process before any further access to the Flash. This can be checked by using the Flash write control mechanism previously described in Section 3.3.4.2.

The op-code used for erase operation is defined in the *FLASHOP* register.

Note: Sector erase by software is not supported. In order to delete a sector, the serial (bit bang) interface should be used.

3.3.5 Shared FLASH

The I350 provides an interface to an external serial Flash/ROM memory device, as described in Section 2.3.2. This Flash/ROM device can be mapped into memory and/or IO address space for each LAN device through the use of Base Address Registers (BARs).

Clearing the *Flash Size* and *CSR_Size* fields in *PCIe Control 2* EEPROM word (Word 0x28) to 0, disables Flash mapping to PCI space of all LAN ports via the *Flash Base Address* register. Setting the *LAN Boot Disable* bit in the per LAN port *Initialization Control 3* EEPROM word, disables Flash mapping to PCI space for LAN 0, LAN1, LAN2 and LAN 3 respectively, via the *Expansion ROM Base Address* register.

3.3.5.1 Flash Access Contention

The I350 implements internal arbitration between Flash accesses initiated from the LAN 0, LAN 1, LAN 2 and LAN 3 devices. If accesses from these LAN devices are initiated during the same window, The first one is served first and only then the following devices are served in a Round Robin fashion.

Note: The I350 does not synchronize between the entities accessing the Flash. Contentions caused by one entity reading and the other modifying the same location is possible.

To avoid this contention, accesses from the LAN devices should be synchronized using external software synchronization of the memory or I/O transactions responsible for the access. It might be possible to ensure contention-avoidance by the nature of the software sequence.

3.3.5.2 Flash Deadlock Avoidance

The Flash is a shared resource between the following clients:

- Port 0 LAN driver accesses.
- Port 1 LAN driver accesses.
- Port 2 LAN driver accesses.
- Port 3 LAN driver accesses.

- BIOS parallel access via expansion ROM mechanism.
- Firmware accesses.

All clients can access the flash using parallel access, where hardware implements the actual access to the Flash. Hardware can schedule these accesses so that all the clients get served without starvation.

However, the driver and firmware clients can access the serial Flash using bit banging. In this case, there is a request/grant mechanism that locks the serial Flash to the exclusive usage of one client. If this client is stuck without releasing the lock, the other clients are unable to access the Flash. In order to avoid this, the I350 implements a time-out mechanism that releases the grant from a client that doesn't toggle the Flash bit-bang interface for more than two seconds.

Note: If an agent that was granted access to the Flash for bit-bang access doesn't toggle the bit-bang interface for 5 Seconds, it should check that it still owns the interface and is not blocked before continuing the bit banging.

Note: When Hardware Flash bit-bang access is aborted due to Deadlock avoidance or management reset the *FLA.FLA_ABORT* bit is set.

This mode is enabled by bit five in word 0xA of the EEPROM.

3.4 Configurable I/O Pins

3.4.1 General-Purpose I/O (Software-Definable Pins)

The I350 has four software-defined pins (SDP pins) per port that can be used for miscellaneous hardware or software-controllable purposes. These pins and their function are bound to a specific LAN device. For example, eight SDP pins cannot be associated with a single LAN device. These pins can each be individually configurable to act as either input or output pins. The default direction of each of the four pins is configurable via the EEPROM as well as the default value of any pins configured as outputs. To avoid signal contention, all four pins are set as input pins until after the EEPROM configuration has been loaded.

In addition to all four pins being individually configurable as inputs or outputs, they can be configured for use as General-Purpose Interrupt (GPI) inputs. To act as GPI pins, the desired pins must be configured as inputs. A separate GPI interrupt-detection enable is then used to enable rising-edge detection of the input pin (rising-edge detection occurs by comparing values sampled at the internal clock rate as opposed to an edge-detection circuit). When detected, a corresponding GPI interrupt is indicated in the Interrupt Cause register.

The use, direction, and values of SDP pins are controlled and accessed using fields in the Device Control (*CTRL*) register and Extended Device Control (*CTRL_EXT*) register.

The SDPs can be used for special purpose mechanisms such as watch dog indication (refer to Section 3.4.2 for details), IEEE 1588 support (refer to Section 7.9.4 for details).

3.4.1.1 SDP usage for SFP connectivity

When an SFP module is connected to the SerDes interface, some of the SFP signals may be connected to SDPs. The following table describes a possible connection as used in Intel's Customer Reference boards.

Table 3-21 SDP connection for SFP boards.

SDP #	Usage	Direction	Default
0	Module Detect - MOD_ABS (SFP pin #6)	Input	
1	Tx Disable (SFP pin #3)	Output	Tx Enable
2	Tx Fault (SFP pin #2)	Input	
3	Power on	Output	Power off

3.4.2 Software Watchdog

In some situations it might be useful to give an indication to manageability firmware or to external devices that the I350 hardware or the software device driver is not functional. For example, in a pass-through NIC, the I350 might be bypassed if it is not functional. In order to provide this functionality, a watchdog mechanism is used. This mechanism can be enabled by default, according to EEPROM configuration.

Once the host driver is up and it determines that hardware is functional, it might reset the watchdog timer to indicate that the I350 is functional. The software device driver should then re-arm the timer periodically. If the timer is not re-armed after pre-programmed timeout, an interrupt is sent to firmware and a pre-programmed SDPx_0 pin (either SDP0_0, SDP1_0, SDP2_0 or SDP3_0) is asserted. Note that the SDP indication is shared between the ports. Additionally the *ICR.Software WD* bit can be set to give an interrupt to the driver when the timeout is reached.

The SDPx_0 pin on which the watchdog timeout is indicated, is defined via the *CTRL.SDP0_WDE bit* on the relevant port. In this mode, the *CTRL.SDP0_IODIR* should be set to output. The *CTRL.SDP0_DATA* bit indicates the polarity of the indication. Setting the *CTRL.SDP0_WDE* bit in one of the ports causes the watchdog timeout indication of all ports to be routed to this SDPx_0 pin.

The register controlling the watchdog timeout feature is the *WDSTP* register. This register enables defining a time-out period and the activation of this mode. Default watchdog timeout activation and timeout period can be set in the EEPROM.

The timer is re-armed by setting the *WDSWSTS.Dev_functional* bit.

If software needs to trigger the watchdog immediately because it suspects hardware is stuck, it can set the *WDSWSTS.Force_WD* bit. It can also supply firmware the cause for the watchdog, by placing additional information in the *WDSWSTS.Stuck Reason* field.

Note: The watchdog circuitry has no logic to detect if hardware is not functional. If the hardware is not functional, the watchdog may expire due to software not being able to access the hardware, thus indicating there is potential hardware problem.

3.4.2.1 Watchdog Rearm

After a watchdog indication was received, in order to rearm the mechanism the following flow should be used:

1. Clear *WD_enable* bit in the *WDSTP* register.
2. Clear *SDP0_WDE* bit in *CTRL* register.
3. Set *SDP0_WDE* bit in *CTRL* register.
4. Set *WD_enable* bit in the *WDSTP* register.

3.4.3 LEDs

The I350 provides four LEDs per port that can be used to indicate different statuses of the traffic. The default setup of the LEDs is done via EEPROM word offsets 0x1C and 0x1F from start of relevant LAN port section (LAN port 0, 1, 2 and 3). This setup is reflected in the *LEDCTL* register of each port. Each software device driver can change its setup individually. For each of the LEDs, the following parameters can be defined:

- Mode: Defines which information is reflected by this LED. The encoding is described in the *LEDCTL* register.
- Polarity: Defines the polarity of the LED.
- Blink mode: Determines whether or not the LED should blink or be stable.

In addition, the blink rate of all LEDs can be defined. The possible rates are 200 ms or 83 ms for each phase. There is one rate for all the LEDs of a port.

3.5 Voltage Regulators

The I350 operates from 3 power rails 1.0V, 1.8V and 3.3V. By using the internal 1.8V LVR (Linear Voltage Regulator) control circuit and the internal 1.0V SVR (Switched Voltage Regulator) control circuit with external low cost power transistors and LC circuitry, the I350 can be configured to operate from a single 3.3V power rail.

Note: To activate the 1.0V SVR control circuit and 1.8V LVR control circuitry, the VR_EN pin should be driven high (refer to Section 2.2.9).

3.5.1 1.8V LVR Control

The I350 includes an on-chip Linear Voltage regulator (LVR) control circuit. Together with an external low cost BJT transistor, the circuit can be used to generate a 1.8V power supply without need for a higher cost on-board 1.8V voltage regulator.See Figure 3-3.

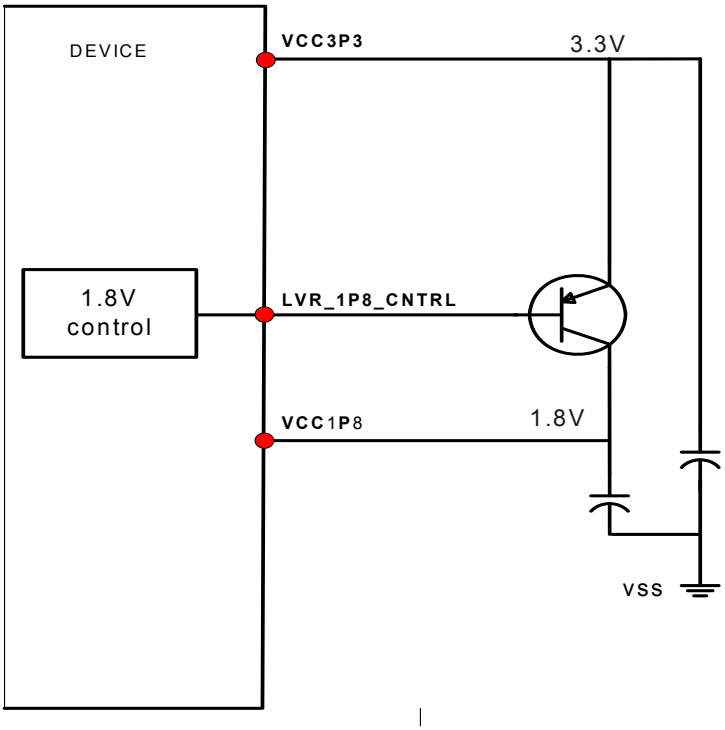


Figure 3-3 1.8V LVR Connection

3.5.2 1.0V SVR Control

The I350 includes an on-chip Switched Voltage Regulator (SVR) control circuit. Together with external matched P/N MOS power transistors, resistors and a LC filter the SVR can be used to generate a 1.0V power supply without need for a higher cost on-board 1.0V Switched Voltage Regulator. See Figure 3-4.

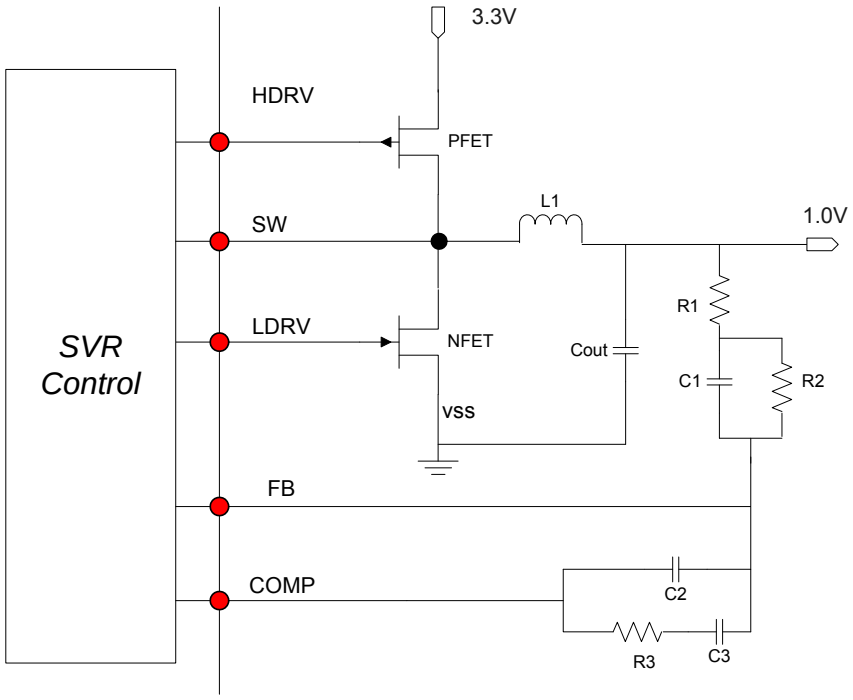


Figure 3-4 1.0V SVR Connection

3.6 Thermal Sensor

Using the on die Thermal Sensor, the I350 can be programmed to execute certain actions to mitigate a thermal event once device temperature passed one of 3 thermal trip points:

- Assert a SDP pin.
- Send an interrupt to the Host.
- Issue an Alert to the external BMC.
- Reduce link speed (thermal throttling).
- Power down device.

In addition, the I350 can be programmed to accept an input from a SDP pin as an indication to execute one of the above actions.

3.6.1 Initializing Thermal Sensor

The BMC or the EEPROM can program up to 3 thermal trip points and define thermal policies or thermal correction actions to be executed once on-die temperature exceeds pre-defined thresholds, via the *THLOWTC*, *THMIDTC* and *THHIGHTC* registers.

In addition, a hysteresis value can be defined in each of the registers to allow the thermal correction action to continue once trip point is triggered until on-die temperature is below the Threshold minus the Hysteresis value. Hysteresis provides a small amount of positive feedback to the thermal sensor circuit to prevent a trip point from flipping back and forth rapidly when the temperature is right at the trip point.

Once on-die temperature exceeds the value programmed in the *Threshold* field of the *THLOWTC*, *THMIDTC* or *THHIGHTC* Thermal Control registers; the following thermal correction actions can be taken:

- A SDP pin is asserted if the *THSDP_OUT* bit is set to 1b in the relevant Thermal Control register. The SDP pin set and its polarity is defined in the *THACNFG* register.
- An interrupt is sent to the Host by asserting the *ICR.THS* bit if the *HINTR Thres* bit is set in the relevant Thermal Control register.
- An interrupt is sent to the Host by asserting the *ICR.THS* bit if the on-die temperature passed the trip point value defined in the *Threshold* field and then as a result of a thermal correction action went below the *Threshold - Hysteresis* value. This functionality is enabled by setting the *HINTR Hyst* bit in the relevant Thermal Control register.
- If device is in D3, a wake-up event is generated when relevant Thermal Control register generates a Thermal Sensor interrupt and the *Wake_TH* bit is set in the register. Wake-up is initiated only if the *WUFC.THS_WK* bit is set.
- An alert is sent to the BMC if the *BMCAL Thres* bit is set in the relevant Thermal Control register.
- An alert is sent to the BMC if the on-die temperature passed the trip point value defined in the *Threshold* field and then as a result of a thermal correction action went below the *Threshold - Hysteresis* value. This functionality is enabled by setting the *BMCAL Hyst* bit in the relevant Thermal Control register.
- Link rate of all active ports configured to internal copper PHY (*CTRL_EXT.Link_Mode* = 00b) can be reduced according to the link rate defined in the *TTHROTLE* field in the relevant Thermal Control register.
- Device can be placed in thermal power down state if the *PWR_DN* bit of the relevant Thermal Control register is set.
  - This action should be executed at a temperature which the chip must be shut down immediately. It therefore must correspond to a temperature guaranteed to be functional. Such a trip point is the last measure to prevent permanent damage to the device. On triggering the trip point, the I350 enters thermal power-down.

    In thermal power down state:
    The following functionality is kept:
    - The PCIe interface is kept on.
    - Host can access the thermal sensor registers to identify the thermal state.

    The BMC can access the thermal sensor registers to identify the thermal state.
    The following functionality is disabled:
    - Network interfaces are down in low power state.

A thermal correction event can also be triggered by an external circuit by asserting a SDP pin if the *THSDP_IN* bit in the *THLOWTC*, *THMIDTC* or *THHIGHTC* Thermal Control registers is set. The SDP pin used for triggering the event and its polarity can be defined in the *THACNFG* register.

The BMC or Host can read the on-die temperature and status of the Thermal Sensor circuitry by reading the *THMJT* register and the *THSTAT* register respectively.

A thermal trip point should be reached infrequently. If appropriate thermal action correction is taken, device power can be decreased before the maximum junction temperature reaches Tj (max).

Setting the trip point should take into account inaccuracies in measuring the maximum junction temperature:

- Since the thermal sensor is not necessarily located at the hottest position on-die, the trip point value is reduced by the difference in temperature between the location of the thermal sensor and the hottest place on the die.
- Since temperature measurement carries some error, the trip point value is reduced by the size of this error.

Note: The Thermal sensor registers are common to all functions. Before accessing any of the registers defined in this section, firmware or software should take ownership of thermal sensor semaphore bits (SW_FW_SYNC.SW_PWRTS_SM for software and SW_FW_SYNC.FW_PWRTS_SM for firmware) according to the flow defined in Section 4.7.1 and release ownership of the Thermal sensor semaphore bits according to the flow defined in Section 4.7.2.

3.6.2 Firmware Based Thermal Management

The I350 can be programmed via the BMC on the NC-SI or SMBus interfaces to initiate Thermal actions and report thermal occurrences.

3.6.3 Thermal Sensor Diagnostics

To enable testing thermal sensor related logic and code without need to vary the temperature, the I350 enables forcing the Thermal Sensor temperature output by setting the *THDIAG.TS Bypass* bit to 1b and specifying the required temperature indication in the *THDIAG.Tj Force field*. The forced temperature can be read in the *THMJT.Tj* field similar to the behavior in non-forced mode.

3.6.4 Thermal Sensor Characteristics

Table 3-22 summarizes the Thermal Sensor Characteristics.

Table 3-22 Thermal Sensor Characteristics

Parameter	Min.	Nom.	Max.	Units	Comments/Conditions
DC power supply Voltage	0.95	1.0	1.05	V	+/-5%
Data Conversion Time		13.1		mS	
Resolution		0.25		°C	Minimum temperature that can be resolved.
Absolute accuracy		+/-5 °C		°C	Quadratic equation.
Power Supply rejection		3 °C/V		°C/V	

3.7 Network Interfaces

3.7.1 Overview

The I350 MAC provides a complete CSMA/CD function supporting IEEE 802.3 (10 Mb/s), 802.3u (100 Mb/s), 802.3z and 802.3ab (1000 Mb/s) implementations. The I350 performs all of the functions required for transmission, reception, and collision handling called out in the standards.

Each I350 MAC can be configured to use a different media interface. The I350 supports the following potential configurations:

- Internal copper PHY.
- External SerDes device such as an optical SerDes (SFP or on board) or backplane (1000BASE-BX or 1000BASE-KX) connections.
- External SGMII device. This mode is used for connections to external 10/100/1000 BASE-T PHYs that support the SGMII MAC/PHY interface.

Selection between the various configurations is programmable via each MAC's Extended Device Control register (*CTRL_EXT.LINK_MODE* bits) and default is set via EEPROM settings. Table 3-23 lists the encoding on the *LINK_MODE* field for each of the modes.

Table 3-23 Link Mode Encoding

Link Mode	I350 Mode
00b	Internal PHY
01b	1000BASE-KX
10b	SGMII
11b	SerDes/1000BASE-BX

The GMII/MII interface, used to communicate between the MAC and the internal PHY or the SGMII PCS, supports 10/100/1000 Mb/s operation, with both half- and full-duplex operation at 10/100 Mb/s, and only full-duplex operation at 1000 Mb/s.

The SerDes function can be used to implement a fiber-optics-based solution or backplane connection without requiring an external TBI mode transceiver/SerDes.

The SerDes interface can be used to connect to SFP modules. As such, this SerDes interface has the following limitations:

- No Tx clock
- AC coupling only

The internal copper PHY supports 10/100/1000BASE-T signaling and is capable of performing intelligent power-management based on both the system power-state and LAN energy-detection (detection of unplugged cables). Power management includes the ability to shut-down to an extremely low (powered-down) state when not needed, as well as the ability to auto-negotiate to lower-speed (and less power-hungry) 10/100 Mb/s operation when the system is in low power-states.

3.7.2 MAC Functionality

3.7.2.1 Internal GMII/MII Interface

The I350's MAC and PHY/PCS communicate through an internal GMII/MII interface that can be configured for either 1000 Mb/s operation (GMII) or 10/100 Mb/s (MII) mode of operation. For proper network operation, both the MAC and PHY must be properly configured (either explicitly via software or via hardware auto-negotiation) to identical speed and duplex settings.

All MAC configuration is performed using Device Control registers mapped into system memory or I/O space; an internal MDIO/MDC interface, accessible via software, is used to configure the Internal PHY. In addition an external MDIO/MDC interface is available to configure external PHY's that are connected to the I350 via the SGMII interface.

3.7.2.2 MDIO/MDC PHY Management Interface

The I350 implements an IEEE 802.3 MII Management Interface (also known as the Management Data Input/Output or MDIO Interface) between the MAC and a PHY. This interface provides the MAC and software the ability to monitor and control the state of the PHY. The MDIO interface defines a physical connection, a special protocol that runs across the connection, and an internal set of addressable registers. The interface consists of a data line (MDIO) and clock line (MDC), which are accessible by software via the MAC register space.

- MDC (management data clock): This signal is used by the PHY as a clock timing reference for information transfer on the MDIO signal. The MDC is not required to be a continuous signal and can be frozen when no management data is transferred. The MDC signal has a maximum operating frequency of 2.5 MHz.
- MDIO (management data I/O): This bi-directional signal between the MAC and PHY is used to transfer control and status information to and from the PHY (to read and write the PHY management registers).

Software can use MDIO accesses to read or write registers of the internal PHY or an external SGMII PHY, by accessing the I350's *MDIC* register (refer to Section 8.2.4). MDIO configuration setup (Internal/ External PHY, PHY Address and Shared MDIO) is defined in the *MDICNFG* register (refer to Section 8.2.5).

When working in SGMII/SerDes mode, the external PHY (if it exists) can be accessed either through MDC/MDIO as previously described, or via a two wire I^2C interface bus using the *I2CCMD* register (refer to Section 8.17.8). The two wire I^2C interface bus or the MDC/MDIO bus are connected via the same pins, and thus are mutually exclusive. In order to be able to control an external device, either by I^2C or MDC/MDIO, the 2-wires *SFP Enable* bit in *Initialization Control 3* EEPROM word, that's loaded into the *CTRL_EXT.I2C Enabled* register bit, should be set.

As the MDC/MDIO command can be targeted either to the internal PHY or to an external bus, the *MDICNFG.destination* bit is used to define the target of the transaction. Following reset, the value of the *MDICNFG.destination* bit is loaded from the *External MDIO* bit in the *Initialization Control 3* EEPROM word. When the *MDICNFG.destination* is clear, the MDIO access is always to the internal PHY and the PHY address is ignored.

Each port has its own MDC/MDIO or two wire interface bus. However, the MDC/MDIO bus of LAN port 0 may be shared by all ports configured to external PHY operation (*MDICNFG.destination* set to 1), to allow control of a multi PHY chip with a single MDC/MDIO bus.

MDIO operation using a shared bus or a separate bus is controlled by the *MDICNFG.Com_MDIO* bit This bit is loaded from *Initialization Control 3* EEPROM word following reset. The external port PHY Address is written in the *MDICNFG.PHYADD* register field, which is loaded from the *Initialization Control 4* EEPROM word following reset.

3.7.2.2.1 Detection of External I2C or MDIO Connection

When the *CTRL_EXT.I2C Enabled* bit is set to 1, software can recognize type of external PHY control bus (MDIO or I2C) connection according to the values loaded from the EEPROM to the *MDICNFG.Destination* bit and the *CTRL_EXT.LINK_MODE* field in the following manner:

- External I2C operating mode - *MDICNFG.Destination* equals 0 and *CTRL_EXT.LINK_MODE* is not equal to 0.
- External MDIO Operating mode - *MDICNFG.Destination* equals 1 and *CTRL_EXT.LINK_MODE* is not equal to 0.

3.7.2.2.2 MDIC and MDICNFG register usage

For a MDIO read cycle, the sequence of events is as follows:

1. If default *MDICNFG* register values loaded from EEPROM need to be updated. The processor performs a PCIe write access to the *MDICNFG* register to define the:
   — PHYADD = Address of external PHY.
   — Destination = Internal or external PHY.
   — Com_MDIO = Shared or separate MDIO external PHY connection.
2. The processor performs a PCIe write cycle to the *MDIC* register with:
   — Ready = 0b
   — Interrupt Enable set to 1b or 0b
   — Opcode = 10b (read)
   — REGADD = Register address of the specific register to be accessed (0 through 31).
3. The MAC applies the following sequence on the MDIO signal to the PHY:

   <PREAMBLE><01><10><PHYADD><REGADD><Z> where Z stands for the MAC tri-stating the MDIO signal.
4. The PHY returns the following sequence on the MDIO signal<0><DATA><IDLE>.
5. The MAC discards the leading bit and places the following 16 data bits in the MII register.
6. The I350 asserts an interrupt indicating MDIO “Done” if the *Interrupt Enable* bit was set.
7. The I350 sets the *Ready* bit in the *MDIC* register indicating the Read is complete.
8. The processor might read the data from the *MDIC* register and issue a new MDIO command.

For a MDIO write cycle, the sequence of events is as follows:

1. If default *MDICNFG* register values loaded from EEPROM need to be updated. The processor performs a PCIe write cycle to the *MDICNFG* register to define the:
   — PHYADD = Address of external PHY.
   — Destination = Internal or external PHY.
   — Com_MDIO = Shared or separate MDIO external PHY connection.
2. The processor performs a PCIe write cycle to the *MDIC* register with:
   — Ready = 0b.

— Interrupt Enable set to 1b or 0b.
— Opcode = 01b (write).
— REGADD = Register address of the specific register to be accessed (0 through 31).
— Data = Specific data for desired control of the PHY.

3. The MAC applies the following sequence on the MDIO signal to the PHY:

   <PREAMBLE><01><01><PHYADD><REGADD><10><DATA><IDLE>

4. The I350 asserts an interrupt indicating MDIO "Done" if the *Interrupt Enable* bit was set.
5. The I350 sets the *Ready* bit in the *MDIC* register to indicate that the write operation completed.
6. The CPU might issue a new MDIO command.

Note: A MDIO read or write might take as long as 64 μs from the processor write to the *Ready* bit assertion. When a shared MDC/MDIO bus is used, each transaction can take up to 256 μs to complete if other ports are using the bus concurrently.

If an invalid opcode is written by software, the MAC does not execute any accesses to the PHY registers.

If the PHY does not generate a 0b as the second bit of the turn-around cycle for reads, the MAC aborts the access, sets the *E* (error) bit, writes 0xFFFF to the data field to indicate an error condition, and sets the *Ready* bit.

Note: After a PHY reset, access through the *MDIC* register should not be attempted for 300 μsec.

3.7.2.3 Duplex Operation with Copper PHY

The I350 supports half-duplex and full-duplex 10/100 Mb/s MII mode either through the internal copper PHY or SGMII interface. However, only full-duplex mode is supported when SerDes/1000BASE-BX or 1000BASE-KX modes are used or in any 1000 Mb/s connection.

Configuration of the duplex operation of the I350 can either be forced or determined via the auto-negotiation process. Refer to Section 3.7.4.4 for details on link configuration setup and resolution.

3.7.2.3.1 Full Duplex

All aspects of the IEEE 802.3, 802.3u, 802.3z, and 802.3ab specifications are supported in full-duplex operation. Full-duplex operation is enabled by several mechanisms, depending on the speed configuration of the I350 and the specific capabilities of the link partner used in the application. During full-duplex operation, the I350 can transmit and receive packets simultaneously across the link interface.

In full-duplex, transmission and reception are delineated independently by the GMII/MII control signals. Transmission starts TX_EN is asserted, which indicates there is valid data on the TX_DATA bus driven from the MAC to the PHY/PCS. Reception is signaled by the PHY/PCS by the asserting the RX_DV signal, which indicates valid receive data on the RX_DATA lines to the MAC.

3.7.2.3.2 Half Duplex

In half-duplex operation, the MAC attempts to avoid contention with other traffic on the link by monitoring the CRS signal provided by the PHY and deferring to passing traffic. When the CRS signal is de-asserted or after a sufficient Inter-Packet Gap (IPG) has elapsed after a transmission, frame transmission begins. The MAC signals the PHY/PCS with TX_EN at the start of transmission.

In the case of a collision, the PHY/SGMII detects the collision and asserts the COL signal to the MAC. Frame transmission stops within four link clock times and then the I350 sends a JAM sequence onto the link. After the end of a collided transmission, the I350 backs off and attempts to re-transmit per the standard CSMA/CD method.

Note: The re-transmissions are done from the data stored internally in I350 MAC transmit packet buffer (no re-access to the data in host memory is performed).

The MAC behavior is different if a regular collision or a late collision is detected. If a regular collision is detected, the MAC always tries to re-transmit until the number of excessive collisions is reached. In case of late collision, the MAC retransmission is configurable. In addition, statistics are gathered on late collisions.

In the case of a successful transmission, I350 is ready to transmit any other frame(s) queued in the MAC's transmit FIFO, after the minimum inter-frame spacing (IFS) of the link has elapsed.

During transmit, the PHY is expected to signal a carrier-sense (assert the CRS signal) back to the MAC before one slot time has elapsed. The transmission completes successfully even if the PHY fails to indicate CRS within the slot time window. If this situation occurs, the PHY can either be configured incorrectly or be in a link down situation. Such an event is counted in the Transmit without CRS statistic register (refer to Section 8.18.12).

3.7.3 SerDes/1000BASE-BX, SGMII and 1000BASE-KX Support

The I350 can be configured to follow either SGMII, SerDes/1000BASE-BX or 1000BASE-KX standards. When in SGMII mode, the I350 can be configured to operate in 1 Gb/s, 100 Mb/s or 10 Mb/s speeds. When in the 10/100 Mb/s speed, the I350 can be configured to half-duplex mode of operation. When configured for SerDes/1000BASE-BX or 1000BASE-KX operation, the port supports only 1 Gb/s, full-duplex operation. Since the serial interfaces are defined as differential signals, internally the hardware has analog and digital blocks. Following is the initialization/configuration sequence for the analog and digital blocks.

3.7.3.1 SerDes/1000BASE-BX, SGMII and 1000BASE-KX Analog Block

The analog block may require some changes to its configuration registers in order to work properly. There is no special requirement for designers to do these changes as the hardware internally updates the configuration using a default sequence or a sequence loaded from the EEPROM.

3.7.3.2 SerDes/1000BASE-BX, SGMII and 1000BASE-KX PCS Block

The link setup for SerDes/1000BASE-BX, 1000BASE-KX and SGMII are described in sections 3.7.4.1, 3.7.4.2 and 3.7.4.3 respectively.

3.7.3.3 GbE Physical Coding Sub-Layer (PCS)

The I350 integrates the 802.3z PCS function on-chip. The on-chip PCS circuitry is used when the link interface is configured for SerDes/1000BASE-BX, 1000BASE-KX or SGMII operation and is bypassed for internal PHY mode.

The packet encapsulation is based on the Fiber Channel (FC0/FC1) physical layer and uses the same coding scheme to maintain transition density and DC balance. The physical layer device is the SerDes and is used for 1000BASE-SX, -L-, or -CX configurations.

3.7.3.3.1 8B10B Encoding/Decoding

The GbE PCS circuitry uses the same transmission-coding scheme used in the fiber channel physical layer specification. The 8B10B-coding scheme was chosen by the standards committee in order to provide a balanced, continuous stream with sufficient transition density to allow for clock recovery at the receiving station. There is a 25% overhead for this transmission code, which accounts for the data-signaling rate of 1250 Mb/s with 1000 Mb/s of actual data.

3.7.3.3.2 Code Groups and Ordered Sets

Code group and ordered set definitions are defined in clause 36 of the IEEE 802.3z standard. These represent special symbols used in the encapsulation of GbE packets. The following table contains a brief description of defined ordered sets and included for informational purposes only. Refer to clause 36 of the IEEE 802.3z specification for more details.

Table 3-24 Brief Description of Defined Ordered Sets

Code	Ordered_Set	# of Code Groups	Usage
/C/	Configuration	4	General reference to configuration ordered sets, either /C1/ or /C2/, which is used during auto-negotiation to advertise and negotiate link operation information between link partners. Last 2 code groups contain configuration base and next page registers.
/C1/	Configuration 1	4	See /C/. Differs from /C2/ in 2nd code group for maintaining proper signaling disparity[1].
/C2/	Configuration 2	4	See /C/. Differs from /C1/ in 2nd code group for maintaining proper signaling disparity[1].
/I/	IDLE	2	General reference to idle ordered sets. Idle characters are continually transmitted by the end stations and are replaced by encapsulated packet data. The transitions in the idle stream enable the SerDes to maintain clock and symbol synchronization between link partners.
/I1/	IDLE 1	2	See /I/. Differs from /I2/ in 2nd code group for maintaining proper signaling disparity[1].
/I2/	IDLE 2	2	See /I/. Differs from /I1/ in 2nd code group for maintaining proper signaling disparity[1].
/R/	Carrier_Extend	1	This ordered set is used to indicate carrier extension to the receiving PCS. It is also used as part of the end_of_packet encapsulation delimiter as well as IPG for packets in a burst of packets.

Table 3-24 Brief Description of Defined Ordered Sets (Continued)

Code	Ordered_Set	# of Code Groups	Usage
/S/	Start_of_Packet	1	The SPD (start_of_packet delimiter) ordered set is used to indicate the starting boundary of a packet transmission. This symbol replaces the last byte of the preamble received from the MAC layer.
/T/	End_of_Packet	1	The EPD (end_of_packet delimiter) is comprised of three ordered sets. The /T/ symbol is always the first of these and indicates the ending boundary of a packet.
/V/	Error_Propagation	1	The /V/ ordered set is used by the PCS to indicate error propagation between stations. This is normally intended to be used by repeaters to indicate collisions.

1. The concept of running disparity is defined in the standard. In summary, this refers to the 1-0 and 0-1 transitions within 8B10B code groups.

3.7.4 Auto-Negotiation and Link Setup Features

The method for configuring the link between two link partners is highly dependent on the mode of operation as well as the functionality provided by the specific physical layer device (PHY or SerDes). In SerDes/1000BASE-BX mode, the I350 provides the complete PCS and Auto-negotiation functionality as defined in IEEE802.3 clause 36 and clause 37. In internal PHY mode, the PCS and IEEE802.3 clause 28 and clause 40 auto-negotiation functions are maintained within the PHY. In SGMII mode, the I350 supports the SGMII link auto-negotiation process, whereas the link auto-negotiation, as defined in IEEE802.3 clause 28 and clause 40, is done by the external PHY. In 1000BASE-KX mode, the I350 supports only parallel detect of 1000BASE-KX signaling and does not support the full Auto-Negotiation for Backplane Ethernet protocol as defined in IEEE802.3ap clause 73.

Configuring the link can be accomplished by several methods ranging from software forcing link settings, software-controlled negotiation, MAC-controlled auto-negotiation, to auto-negotiation initiated by a PHY. The following sections describe processes of bringing the link up including configuration of the I350 and the transceiver, as well as the various methods of determining duplex and speed configuration.

The process of determining link configuration differs slightly based on the specific link mode (internal PHY, SerDes/1000BASE-BX, SGMII or 1000BASE-KX) being used.

When operating in a SerDes/1000BASE-BX mode, the PCS layer performs auto-negotiation per clause 37 of the 802.3z standard. The transceiver used in this mode does not participate in the auto-negotiation process as all aspects of auto-negotiation are controlled by the I350.

When operating in internal PHY mode, the PHY performs auto-negotiation per 802.3ab clause 40 and extensions to clause 28. Link resolution is obtained by the MAC from the PHY after the link has been established. The MAC accomplishes this via the MDIO interface, via specific signals from the internal PHY to the MAC, or by MAC auto-detection functions.

When operating in SGMII mode, the PCS layer performs SGMII auto-negotiation per the SGMII specification. The external PHY is responsible for the Ethernet auto-negotiation process.

When operating in 1000BASE-KX mode the I350 performs parallel detect of 1000BASE-KX operation but does not implement the full Auto-Negotiation for Backplane Ethernet sequence as defined in IEEE802.3ap clause 73.

3.7.4.1 SerDes/1000BASE-BX Link Configuration

When using SerDes/1000BASE-BX link mode, link mode configuration can be performed using the PCS function in the I350. The hardware supports both hardware and software auto-negotiation methods for determining the link configuration, as well as allowing for a manual configuration to force the link. Hardware auto-negotiation is the preferred method.

3.7.4.1.1 Signal Detect Indication

When the *CONNSW.ENRGSRC* bit is set to 1, the SRDS_0/1/2/3_SIG_DET pins can be connected to a Signal Detect or loss-of-signal (LOS) output of the optical module that indicates when no laser light is being received when the I350 is used in a 1000BASE-SX or -LX implementation (SerDes operation). It prevents false carrier cases occurring when transmission by a non connected port couples in to the input. No standard polarity for the Signal Detect or loss-of-signal driven from different manufacturer optical modules exists. The *CTRL.ILOS* bit provides the capability to invert the signal from different external optical module vendors, and should be set when the external optical module provides a negative-true loss-of-signal.

Note: In internal PHY, SGMII, 1000BASE-BX and 1000BASE-KX connections energy detect source is always internal and value of *CONNSW.ENRGSRC* bit should be 0. The *CTRL.ILOS* bit also inverts the internal Link-up input that provides link status indication and thus should be set to 0 for proper operation.

3.7.4.1.2 MAC Link Speed

SerDes/1000BASE-BX operation is only defined for 1000 Mb/s operation. Other link speeds are not supported. When configured for the SerDes interface, the MAC speed-determination function is disabled and the Device Status register bits (*STATUS.SPEED*) indicate a value of 10b for 1000 Mb/s.

3.7.4.1.3 SerDes/1000BASE-BX Mode Auto-Negotiation

In SerDes/1000BASE-BX mode, after power up or I350 reset via PE_RST_N, the I350 initiates IEEE802.3 clause 37 auto-negotiation based on the default settings in the device control and transmit configuration or PCS Link Control Word registers, as well as settings read from the EEPROM. If enabled in the EEPROM, the I350 immediately performs auto-negotiation.

TBI mode auto-negotiation, as defined in clause 37 of the IEEE 802.3z standard, provides a protocol for two devices to advertise and negotiate a common operational mode across a GbE link. The I350 fully supports the IEEE 802.3z auto-negotiation function when using the on-chip PCS and internal SerDes.

TBI mode auto-negotiation is used to determine the following information:

- Duplex resolution (even though the I350 MAC only supports full-duplex in SerDes/1000BASE-BX mode).
- Flow control configuration.

Notes: Since speed for SerDes/1000BASE-BX modes is fixed at 1000 Mb/s, speed settings in the Device Control register are unaffected by the auto-negotiation process.

Auto-negotiation can be initiated at power up or by asserting PE_RST_N and enabling specific bits in the EEPROM.

The auto-negotiation process is accomplished by the exchange of /C/ ordered sets that contain the capabilities defined in the *PCS_ANADV* register in the 3rd and 4th symbols of the ordered sets. Next page are supported using the *PCS_NPTX_AN* register.

Bits *FD* and *LU* in the Device Status (*STATUS*) register, and bits in the *PCS_LSTS* register provide status information regarding the negotiated link.

Auto-negotiation can be initiated by the following:

- *PCS_LCMD.AN_ENABLE* transition from 0b to 1b
- Receipt of /C/ ordered set during normal operation
- Receipt of a different value of the /C/ ordered set during the negotiation process
- Transition from loss of synchronization to synchronized state (if *AN_ENABLE* is set).
- *PCS_LCMD.AN_RESTART* transition from 0b to 1b

Resolution of the negotiated link determines device operation with respect to flow control capability and duplex settings. These negotiated capabilities override advertised and software-controlled device configuration.

Software must configure the *PCS_ANADV* fields to the desired advertised base page. The bits in the Device Control register are not mapped to the *txConfigWord* field in hardware until after auto-negotiation completes. Table 3-25 lists the mapping of the *PCS_ANADV* fields to the Config_reg Base Page encoding per clause 37 of the standard.

Table 3-25 802.3z Advertised Base Page Mapping

15	14	13:12	11:9	8:7	6	5	4:0
Nextp	Ack	RFLT	rsv	ASM	Hd	Fd	rsv

The partner advertisement can be seen in the *PCS_ LPAB* and *PCS_ LPABNP* registers.

3.7.4.1.4 Forcing Link-up in SerDes/1000BASE-BX Mode

Forcing link can be accomplished by software by writing a 1b to *CTRL.SLU*, which forces the MAC PCS logic into a link-up state (enables listening to incoming characters when SRDS_[n]_SIG_DET is asserted by the external optical module or an equivalent signal is asserted by the internal PHY).

Note: The *PCS_LCMD.AN_ENABLE* bit must be set to a logic zero to enable forcing link.

When link is forced via the *CTRL.SLU* bit, the link does not come up unless the SRDS_[n]_SIG_DET signal is asserted or an internal energy indication is received from the SerDes receiver, implying that there is a valid signal being received by the optical module or SerDes circuitry.

The source of the signal detect is defined by the *ENRGSRC* bit in the *CONNSW* register.

3.7.4.1.5 HW Detection of Non-Auto-Negotiation Partner

Hardware can detect a SerDes link partner that sends idle code groups continuously, but does not initiate or answer an auto-negotiation process. In this case, hardware initiates an auto-negotiation process, and if it fails after some timeout, a link up is assumed. To enable this functionality the *PCS_LCTL.AN_TIMEOUT_EN* bit should be set. This mode can be used instead of the force link mode as a way to support a partner that do not support auto-negotiation.

3.7.4.2 1000BASE-KX Link Configuration

When using 1000BASE-KX link mode, link mode configuration is forced manually by software since the I350 does not support IEEE802.3 clause 73 backplane auto-negotiation.

3.7.4.2.1 MAC Link Speed

1000BASE-KX operation is only defined for 1000 Mb/s operation. Other link speeds are not supported. When configured for the 1000BASE-KX interface, the MAC speed-determination function is disabled and the Device Status register bits (*STATUS.SPEED*) indicate a value of 10b for 1000 Mb/s.

3.7.4.2.2 1000BASE-KX Auto-Negotiation

The I350 only supports parallel detection of the 1000BASE-KX link and does not support the full IEEE802.3ap clause 73 backplane auto-negotiation protocol.

3.7.4.2.3 Forcing Link-up in 1000BASE-KX Mode

In 1000BASE-KX mode (*EXT_CTRL.LINK_MODE* = 01b) the I350 should always operates in force link mode (*CTRL.SLU* bit is set to 1). The MAC PCS logic is placed in a link-up state once energy indication is received, implying that a valid signal is being received by the 1000BASE-KX circuitry. When in the link-up state PCS logic can lock on incoming characters.

Note: In 1000BASE-KX mode energy detect source is internal and value of *CONNSW.ENRGSRC* bit should be 0. Clause 37 auto-negotiation should be disabled and the value of the *PCS_LCMD.AN_ENABLE* bit and *PCS_LCMD.AN TIMEOUT EN* bit should be 0.

3.7.4.2.4 1000BASE-KX HW Detection of Link Partner

In 1000BASE-KX mode, hardware detects a 1000BASE-KX link partner that sends idle or none idle code groups continuously. In 1000BASE-KX operation force link-up mode is used.

3.7.4.3 SGMII Link Configuration

When working in SGMII mode, the actual link setting is done by the external PHY and is dependent on the settings of this PHY. The SGMII auto-negotiation process described in the sections that follow is only used to establish the MAC/PHY connection.

3.7.4.3.1 SGMII Auto-Negotiation

This auto-negotiation process is not dependent on the SRDS_[n]_SIG_DET signal and the *CONNSW.ENRGSRC* bit should be 0, as this signal indicates optical module signal detection and is not relevant in SGMII mode.

The outcome of this auto-negotiation process includes the following information:

- Link status
- Speed
- Duplex

This information is used by hardware to configure the MAC, when operating in SGMII mode.

Bits *FD* and *LU* of the Device Status (*STATUS*) register and bits in the *PCS_LSTS* register provide status information regarding the negotiated link.

Auto-negotiation can be initiated by the following:

- *PCS_LCMD.AN_ENABLE* transition from 0b to 1b.

- Receipt of /C/ ordered set during normal operation.
- Receipt of different value of the /C/ ordered set during the negotiation process.
- Transition from loss of synchronization to a synchronized state (if *AN_ENABLE* is set).
- *PCS_LCMD.AN_RESTART* transition from 0b to 1b.

Auto-negotiation determines I350 operation with respect to speed and duplex settings. These negotiated capabilities override advertised and software controlled device configuration.

When working in SGMII mode, there is no need to set the *PCAS_ANADV* register, as the MAC advertisement word is fixed. In SGMII mode the *PCS_LCMD.AN TIMEOUT EN* bit should be 0, since Auto-negotiation outcome is required for correct operation. The result of the SGMII level auto-negotiation can be read from the *PCS_LPAB* register.

3.7.4.3.2 Forcing Link in SGMII mode

In SGMII, forcing of the link cannot be done at the PCS level, only in the external PHY. The forced speed and duplex settings are reflected by the SGMII auto-negotiation process; the MAC settings are automatically done according to this functionality.

3.7.4.3.3 MAC Speed Resolution

The MAC speed and duplex settings are always set according to the SGMII auto-negotiation process.

3.7.4.4 Copper PHY Link Configuration

When operating with the internal PHY, link configuration is generally determined by PHY auto-negotiation. The software device driver must intervene in cases where a successful link is not negotiated or the designer desires to manually configure the link. The following sections discuss the methods of link configuration for copper PHY operation.

3.7.4.4.1 PHY Auto-Negotiation (Speed, Duplex, Flow Control)

When using a copper PHY, the PHY performs the auto-negotiation function. The actual operational details of this operation are described in the IEEE P802.3ab draft standard and are not included here.

Auto-negotiation provides a method for two link partners to exchange information in a systematic manner in order to establish a link configuration providing the highest common level of functionality supported by both partners. Once configured, the link partners exchange configuration information to resolve link settings such as:

- Speed: - 10/100/1000 Mb/s
- Duplex: - Full or half
- Flow control operation

PHY specific information required for establishing the link is also exchanged.

Note: If flow control is enabled in the I350, the settings for the desired flow control behavior must be set by software in the PHY registers and auto-negotiation restarted. After auto-negotiation completes, the software device driver must read the PHY registers to determine the resolved flow control behavior of the link and reflect these in the MAC register settings (*CTRL.TFCE* and *CTRL.RFCE*).

Once PHY auto-negotiation completes, the PHY asserts a link indication (LINK) to the MAC. Software must have set the *Set Link Up* bit in the Device Control register (*CTRL.SLU*) before the MAC recognizes the LINK indication from the PHY and can consider the link to be up.

3.7.4.4.2 MAC Speed Resolution

For proper link operation, both the MAC and PHY must be configured for the same speed of link operation. The speed of the link can be determined and set by several methods with the I350. These include:

- Software-forced configuration of the MAC speed setting based on PHY indications, which might be determined as follows:
  - Software reads of PHY registers directly to determine the PHY's auto-negotiated speed
  - Software reads the PHY's internal PHY-to-MAC speed indication (SPD_IND) using the MAC *STATUS.SPEED* register
- Software asks the MAC to attempt to auto-detect the PHY speed from the PHY-to-MAC RX_CLK, then programs the MAC speed accordingly
- MAC automatically detects and sets the link speed of the MAC based on PHY indications by using the PHY's internal PHY-to-MAC speed indication (SPD_IND)

Aspects of these methods are discussed in the sections that follow.

3.7.4.4.2.1 Forcing MAC Speed

There might be circumstances when the software device driver must forcibly set the link speed of the MAC. This can occur when the link is manually configured. To force the MAC speed, the software device driver must set the *CTRL.FRCSPD* (force-speed) bit to 1b and then write the speed bits in the Device Control register (*CTRL.SPEED*) to the desired speed setting. Refer to Section 8.2.1 for details.

Note: Forcing the MAC speed using *CTRL.FRCSPD* overrides all other mechanisms for configuring the MAC speed and can yield non-functional links if the MAC and PHY are not operating at the same speed/configuration.

When forcing the I350 to a specific speed configuration, the software device driver must also ensure the PHY is configured to a speed setting consistent with MAC speed settings. This implies that software must access the PHY registers to either force the PHY speed or to read the PHY status register bits that indicate link speed of the PHY.

Note: Forcing speed settings by *CTRL.SPEED* can also be accomplished by setting the *CTRL_EXT.SPD_BYPS* bit. This bit bypasses the MAC's internal clock switching logic and enables the software device driver complete control of when the speed setting takes place. The *CTRL.FRCSPD* bit uses the MAC's internal clock switching logic, which does delay the effect of the speed change.

3.7.4.4.2.2 Using Internal PHY Direct Link-Speed Indication

The I350's internal PHY provides a direct internal indication of its speed to the MAC (SPD_IND). When using the internal PHY, the most direct method for determining the PHY link speed and either manually or automatically configuring the MAC speed is based on these direct speed indications.

For MAC speed to be set/determined from these direct internal indications from the PHY, the MAC must be configured such that *CTRL.ASDE* and *CTRL.FRCSPD* are both 0b (both auto-speed detection and forced-speed override disabled). After configuring the Device Control register, MAC speed is re-configured automatically each time the PHY indicates a new link-up event to the MAC.

When MAC speed is neither forced nor auto-sensed by the MAC, the current MAC speed setting and the speed indicated by the PHY is reflected in the Device Status register bits *STATUS.SPEED*.

3.7.4.4.3 MAC Full-/Half- Duplex Resolution

The duplex configuration of the link is also resolved by the PHY during the auto-negotiation process. The I350's internal PHY provides an internal indication to the MAC of the resolved duplex configuration using an internal full-duplex indication (FDX).

When using the internal PHY, this internal duplex indication is normally sampled by the MAC each time the PHY indicates the establishment of a good link (LINK indication). The PHY's indicated duplex configuration is applied in the MAC and reflected in the MAC Device Status register (*STATUS.FD*).

Software can override the duplex setting of the MAC via the *CTRL.FD* bit when the *CTRL.FRCDPLX* (force duplex) bit is set. If *CTRL.FRCDPLX* is 0b, the *CTRL.FD* bit is ignored and the PHY's internal duplex indication is applied.

3.7.4.4.4 Using PHY Registers

The software device driver might be required under some circumstances to read from, or write to, the MII management registers in the PHY. These accesses are performed via the *MDIC* register (refer to Section 8.2.4). The MII registers enable the software device driver to have direct control over the PHY's operation, which can include:

- Resetting the PHY
- Setting preferred link configuration for advertisement during the auto-negotiation process
- Restarting the auto-negotiation process
- Reading auto-negotiation status from the PHY
- Forcing the PHY to a specific link configuration

The set of PHY management registers required for all PHY devices can be found in the IEEE P802.3ab standard. The registers for the I350 PHY are described in Section 3.7.9.

3.7.4.4.5 Comments Regarding Forcing Link

Forcing link in GMII/MII mode (internal PHY) requires the software device driver to configure both the MAC and PHY in a consistent manner with respect to each other as well as the link partner. After initialization, the software device driver configures the desired modes in the MAC, then accesses the PHY registers to set the PHY to the same configuration.

Before enabling the link, the speed and duplex settings of the MAC can be forced by software using the *CTRL.FRCSPD*, *CTRL.FRCDPX*, *CTRL.SPEED*, and *CTRL.FD* bits. After the PHY and MAC have both been configured, the software device driver should write a 1b to the *CTRL.SLU* bit.

3.7.4.5 Loss of Signal/Link Status Indication

For all modes of operation, an LOS/LINK signal provides an indication of physical link status to the MAC. When the MAC is configured for optical SerDes mode, the input reflects loss-of-signal connection from the optics. In backplane mode, where there is no LOS external indication, an internal indication from the SerDes receiver can be used. In SFP systems the LOS indication from the SFP can be used. In internal PHY mode, this signal from the PHY indicates whether the link is up or down; typically indicated

after successful auto-negotiation. Assuming that the MAC has been configured with *CTRL.SLU*=1b, the MAC status bit *STATUS.LU*, when read, generally reflects whether the PHY or SerDes has link (except under forced-link setup where even the PHY link indication might have been forced).

When the link indication from the PHY is de-asserted or the loss-of-signal asserted from the SerDes, the MAC considers this to be a transition to a link-down situation (such as cable unplugged, loss of link partner, etc.). If the Link Status Change (*LSC*) interrupt is enabled, the MAC generates an interrupt to be serviced by the software device driver.

3.7.5 Ethernet Flow Control (FC)

The I350 supports flow control as defined in 802.3x as well as the specific operation of asymmetrical flow control defined by 802.3z.

Flow control is implemented as a means of reducing the possibility of receive buffer overflows, which result in the dropping of received packets, and allows for local controlling of network congestion levels. This can be accomplished by sending an indication to a transmitting station of a nearly full receive buffer condition at a receiving station.

The implementation of asymmetric flow control allows for one link partner to send flow control packets while being allowed to ignore their reception. For example, not required to respond to PAUSE frames.

The following registers are defined for the implementation of flow control:

- *CTRL.RFCE* field is used to enable reception of legacy flow control packets and reaction to them.
- *CTRL.TFCE* field is used to enable transmission of legacy flow control packets.
- Flow Control Address Low, High (*FCAL/H*) - 6-byte flow control multicast address
- Flow Control Type (*FCT*) 16-bit field to indicate flow control type
- Flow Control bits in Device Control (*CTRL*) register - Enables flow control modes.
- Discard PAUSE Frames (*DPF*) and Pass MAC Control Frames (*PMCF*) in *RCTL* - controls the forwarding of control packets to the host.
- Flow Control Receive Threshold High (*FCRTH0*) - A 13-bit high watermark indicating receive buffer fullness. A single watermark is used in link FC mode.
- DMA Coalescing Receive Threshold High (*FCRTC*) - A 13-bit high watermark indicating receive buffer fullness when in DMA coalescing and TX buffer is empty. Value in this register can be higher than value placed in the *FCRTH0* register since watermark needs to be set to allow for only reception of a maximum sized RX packet before XOFF flow control takes effect and reception is stopped (refer to Table 3-29 for information on flow control threshold calculation).
- Flow Control Receive Threshold Low (*FCRTL0*) - A 13-bit low watermark indicating receive buffer emptiness. A single watermark is used in link FC mode.
- Flow Control Transmit Timer Value (*FCTTV*) - a set of 16-bit timer values to include in transmitted PAUSE frame. A single timer is used in Link FC mode.
- Flow Control Refresh Threshold Value (*FCRTV*) - 16-bit PAUSE refresh threshold value

3.7.5.1 MAC Control Frames and Receiving Flow Control Packets

3.7.5.1.1 Structure of 802.3X FC Packets

Three comparisons are used to determine the validity of a flow control frame:

1. A match on the 6-byte multicast address for MAC control frames or to the station address of the I350 (Receive Address Register 0).
2. A match on the type field.
3. A comparison of the MAC *Control Op-Code* field.

The 802.3x standard defines the MAC control frame multicast address as 01-80-C2-00-00-01.

The *Type* field in the FC packet is compared against an IEEE reserved value of 0x8808.

The final check for a valid PAUSE frame is the MAC control op-code. At this time only the PAUSE control frame op-code is defined. It has a value of 0x0001.

Frame-based flow control differentiates XOFF from XON based on the value of the *PAUSE* timer field. Non-zero values constitute XOFF frames while a value of zero constitutes an XON frame. Values in the *Timer* field are in units of pause quantum (slot time). A pause quantum lasts 64 byte times, which is converted in absolute time duration according to the line speed.

Note: XON frame signals the cancellation of the pause from initiated by an XOFF frame - pause for zero pause quantum.

Table 3-26 lists the structure of a 802.3X FC packet.

Table 3-26 802.3X Packet Format

DA	01_80_C2_00_00_01 (6 bytes)
SA	Port MAC address (6 bytes)
Type	0x8808 (2 bytes)
Op-code	0x0001 (2 bytes)
Time	XXXX (2 bytes)
Pad	42 bytes
CRC	4 bytes

3.7.5.1.2 Operation and Rules

The I350 operates in Link FC.

- Link FC is enabled by the *RFCE* bit in the *CTRL* Register.

Note: Link flow control capability is negotiated between link partners via the auto negotiation process. It is the software device driver responsibility to reconfigure the link flow control configuration after the capabilities to be used where negotiated as it might modify the value of these bits based on the resolved capability between the local device and the link partner.

Once the receiver has validated receiving an XOFF, or PAUSE frame, the I350 performs the following:

- Increments the appropriate statistics register(s)
- Sets the *Flow_Control State* bit in the *FCSTS0* register.
- Initializes the pause timer based on the packet's *PAUSE* timer field (overwriting any current timer's value)
- Disables packet transmission or schedules the disabling of transmission after the current packet completes.

Resumption of transmission might occur under the following conditions:

- Expiration of the PAUSE timer

- Reception of an XON frame (a frame with its PAUSE timer set to 0b)

Both conditions clear the relevant *Flow_Control State* bit in the relevant *FCSTS0* register and transmission can resume. Hardware records the number of received XON frames.

3.7.5.1.3 Timing Considerations

When operating at 1 Gb/s line speed, the I350 must not begin to transmit a (new) frame more than two pause-quantum-bit times after receiving a valid link XOFF frame, as measured at the wires. A pause quantum is 512-bit times.

When operating in full duplex at 100 Mb/s or 1 Gb/s line speeds, the I350 must not begin to transmit a (new) frame more than 576-bit times after receiving a valid link XOFF frame, as measured at the wire.

3.7.5.2 PAUSE and MAC Control Frames Forwarding

Two bits in the Receive Control register, control forwarding of PAUSE and MAC control frames to the host. These bits are *Discard PAUSE Frames* (*DPF*) and *Pass MAC Control Frames* (*PMCF*):

- The *DPF* bit controls forwarding of PAUSE packets to the host.
- The *PMCF* bit controls forwarding of non-PAUSE packets to the host.

Note: When flow control reception is disabled (*CTRL.RFCE* = 0), legacy flow control packets are not recognized and are parsed as regular packets.

Table 3-27 lists the behavior of the *DPF* bit.

Table 3-27 Forwarding of PAUSE Packet to Host (*DPF* Bit)

RFCE	DPF	Are FC Packets Forwarded to Host?
0	X	Yes. Packets needs to pass the L2 filters (refer to Section 7.1.1.1).[1]
1	0	Yes. Packets needs to pass the L2 filters (refer to Section 7.1.1.1).
1	1	No if unicast, Yes, if multicast.

1. The flow control multicast address is not part of the L2 filtering unless explicitly required.

Table 3-28 defines the behavior of the *PMCF* bit.

Table 3-28 Transfer of Non-PAUSE Control Packets to Host (*PMCF* Bit)

RFCE	PMCF	Are Non-FC MAC Control Packets Forwarded to Host?
0	X	Yes. Packets needs to pass the L2 filters (refer to Section 7.1.1.1).
X	1	Yes. Packets needs to pass the L2 filters (refer to Section 7.1.1.1).
X	0	No

3.7.5.3 Transmission of PAUSE Frames

The I350 generates PAUSE packets to ensure there is enough space in its receive packet buffers to avoid packet drop. The I350 monitors the fullness of its receive packet buffers and compares it with the contents of a programmable threshold. When the threshold is reached, the I350 sends a PAUSE frame. The I350 also supports the sending of link Flow Control (FC).

Note: Similar to receiving link flow control packets previously mentioned, link XOFF packets can be transmitted only if this configuration has been negotiated between the link partners via the auto-negotiation process or some higher level protocol. The setting of this bit by the software device driver indicates the desired configuration.

The transmission of flow control frames should only be enabled in full-duplex mode per the IEEE 802.3 standard. Software should ensure that the transmission of flow control packets is disabled when the I350 is operating in half-duplex mode.

3.7.5.3.1 Operation and Rules

Transmission of link PAUSE frames is enabled by software writing a 1b to the *TFCE* bit in the Device Control register.

The I350 sends a PAUSE frame when Rx packet buffer is full above the high threshold defined in the Flow Control Receive Threshold High (*FCRT0.RTH*) register field. When the threshold is reached, the I350 sends a PAUSE frame with its pause time field equal to *FCTTV*. The threshold should be large enough to overcome the worst case latency from the time that crossing the threshold is sensed till packets are not received from the link partner. The Flow Control Receive Threshold High value should be calculated as follows:

```
Flow Control Receive Threshold High =Internal RX Buffer Size - (Threshold
Cross to XOFF Transmission + Round-trip Latency + XOFF Reception to Link
Partner response)
```

Parameter values to be used for calculating the *FCRT0.RTH* value can be found in Table 3-29.

Table 3-29 Flow Control Receive Threshold High (FCRTH0.RTH) Value Calculation

Latency Parameter	Affected by	Parameter Value
Threshold Cross to XOFF Transmission	Max packet size	Max packet Size * 1.25
XOFF Reception to Link Partner response	Max packet size	Max packet size
Round trip latency	The latencies on the wire and the LAN devices at both sides of the wire	320 Byte (for 1000Base-T operation).

Note: When DMA Coalescing is enabled (*DMACR.DMAC_EN* = 1) value placed in the *FCRTC.RTH_Coal* field should be equal or lower than:

```
FCRTC.RTH_Coal = FCRTH0.RTH + Max packet Size * 1.25
```

The *FCRTC.RTH_Coal* is used as the high watermark to generate XOFF flow control packets when the internal TX buffer is empty and the I350 is executing DMA coalescing. In this case, no delay to transmission of flow control packet exists so its possible to increase level of watermark before issuing a XOFF flow control frame.

After transmitting a PAUSE frame, the I350 activates an internal shadow counter that reflects the link partner pause timeout counter. When the counter reaches the value indicated in the *FCRTV* register, then, if the PAUSE condition is still valid (meaning that the buffer fullness is still above the high watermark), a XOFF message is sent again.

Once the receive buffer fullness reaches the low water mark, the I350 sends a XON message (a PAUSE frame with a timer value of zero). Software enables this capability with the *XONE* field of the *FCRTL*.

The I350 sends an additional PAUSE frame if it has previously sent one and the packet buffer overflows. This is intended to minimize the amount of packets dropped if the first PAUSE frame did not reach its target.

3.7.5.3.2 Software Initiated PAUSE Frame Transmission

The I350 has the added capability to transmit an XOFF frame via software. This is accomplished by software writing a 1b to the *SWXOFF* bit of the Transmit Control register. Once this bit is set, hardware initiates the transmission of a PAUSE frame in a manner similar to that automatically generated by hardware.

The *SWXOFF* bit is self-clearing after the PAUSE frame has been transmitted.

Note: The Flow Control Refresh Threshold mechanism does not work in the case of software-initiated flow control. Therefore, it is the software's responsibility to re-generate PAUSE frames before expiration of the pause counter at the other partner's end.

The state of the *CTRL.TFCE* bit or the negotiated flow control configuration does not affect software generated PAUSE frame transmission.

Note: Software sends an XON frame by programming a 0b in the PAUSE timer field of the *FCTTV* register. Software generation of XON packet is not allowed while the hardware flow control mechanism is active, as both use the *FCTTV* registers for different purposes.

XOFF transmission is not supported in 802.3x for half-duplex links. Software should not initiate an XOFF or XON transmission if the I350 is configured for half-duplex operation.

When flow control is disabled, pause packets (XON, XOFF, and other FC) are not detected as flow control packets and can be counted in a variety of counters (such as multicast).

3.7.5.4 IPG Control and Pacing

The I350 supports the following modes of controlling IPG duration:

- Fixed IPG - the IPG is extended by a fixed duration

3.7.5.4.1 Fixed IPG Extension

The I350 allows controlling of the IPG duration. The IPGT configuration field enables an extension of IPG in 4-byte increments. One possible use of this capability is to allow the insertion of bytes into the transmit packet after it has been transmitted by the I350 without violating the minimum IPG requirements. For example, a security device connected in series to the I350 might add security headers to transmit packets before the packets are transmitted on the network.

3.7.6 Loopback Support

3.7.6.1 General

The I350 supports the following types of internal loopback in the LAN interfaces:

- MAC Loopback (Point 1) - see Section 3.7.6.2.
- PHY Loopback (Point 2) - see Section 3.7.6.3.
- SerDes, SGMII or 1000BASE-KX Loopback (Point 3) - see Section 3.7.6.4.

- External PHY Loopback (Point 4) - see Section 3.7.6.5.

By setting the device to loopback mode, packets that are transmitted towards the line will be looped back to the host. The I350 is fully functional in these modes, just not transmitting data over the lines. Figure 3-5 shows the points of loopback.

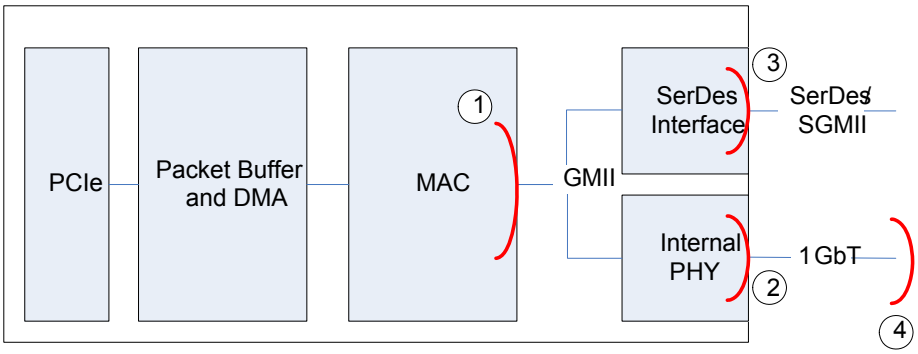


Figure 3-5 I350 Loopback Modes

3.7.6.2 MAC Loopback

In MAC loopback, the PHY and SerDes blocks are not functional and data is looped back before these blocks.

3.7.6.2.1 Setting the I350 to MAC Loopback Mode

The following procedure should be used to put the I350 in MAC loopback mode:

- Set *RCTL.LBM* to 2'b01 (bits 7:6)
- Set *CTRL.SLU* (bit 6, should be set by default)
- Set *CTRL.FRCSPD* and *FRCDPLX* (bits 11 and 12)
- Set the *CTRL.FD* bit and program the *CTRL.SPEED* field to 10b (1G).
- Set *EEER.EEE_FRC_AN* to 1b to enable checking EEE operation in MAC loopback mode.

Filter configuration and other TX/RX processes are the same as in normal mode.

3.7.6.3 Internal PHY Loopback

In PHY loopback, the SerDes block is not functional and data is looped back at the end of the PHY functionality. This means all the design, that is functional in copper mode, is involved in the loopback

3.7.6.3.1 Setting the I350 to Internal PHY loopback Mode

The following procedure should be used to place the I350 in PHY loopback mode on any LAN port:

- Set Link mode to Internal PHY: *CTRL_EXT.LINK_MODE* = 00b.
- Clear *PHPM.SPD_EN*.
- In PHY control register (*PCTRL* - Address 0 in the PHY):

  - Set Duplex mode (bit 8)
  - Set Loopback bit (Bit 14)
  - Clear Auto Neg enable bit (Bit 12)
  - Set speed using bits 6 and 13 as described in EAS.
  - Register value should be:
    For 10 Mbps 0x4100
    For 100 Mbps 0x6100
    For 1000 Mbps 0x4140.
- Determine the exact type of loopback using the loopback control register (*PHLBKC* - address 19d, refer to Section 8.26.3.17).

Note: While in MII loopback mode *PHLBKC.Force Link Status* should be set to 1 to receive valid link state and be able to Transmit and Receive normally.

Make sure a Configure command is re-issued (loopback bits set to 00b) to cancel the loopback mode.

3.7.6.4 SerDes, SGMII and 1000BASE-KX Loopback

In SerDes, SGMII or 1000BASE-KX loopback, the PHY block is not functional and data is looped back at the end of the relevant functionality. This means all the design that is functional in SerDes/SGMII or 1000BASE-KX mode, is involved in the loopback.

Note: SerDes loopback is functional only if the SerDes link is up.

3.7.6.4.1 Setting SerDes/1000BASE-BX, SGMII, 1000BASE-KX Loopback Mode

The following procedure should be used to place the I350 in SerDes loopback mode:

- Set Link mode to either SerDes, SGMII or 1000BASE-KX by:
  - 1000BASE-KX: *CTRL_EXT.LINK_MODE* = 01b
  - SGMII: *CTRL_EXT.LINK_MODE* = 10b
  - SerDes/1000BASE-BX: *CTRL_EXT.LINK_MODE* = 11b
- Configure SERDES to loopback: *RCTL.LBM* = 11b
- Move to Force mode by setting the following bits:
  - *CTRL.FD* (CSR 0x0 bit 0) = 1
  - *CTRL.SLU* (CSR 0x0 bit 6) = 1
  - *CTRL.RFCE* (CSR 0x0 bit 27) = 0
  - *CTRL.TFCE* (CSR 0x0 bit 28) = 0
  - *PCS_LCTL.FORCE_LINK* (CSR 0X4208 bit 5) = 1
  - *PCS_LCTL.FSD* (CSR 0X4208 bit 4) = 1
  - *PCS_LCTL.FDV* (CSR 0X4208 bit 3) = 1
  - *PCS_LCTL.FLV* (CSR 0X4208 bit 0) = 1
  - *PCS_LCTL.AN_ENABLE* (CSR 0X4208 bit 16) = 0

3.7.6.5 External PHY Loopback

In External PHY loopback, the SerDes block is not functional and data is sent through the MDI interface and looped back using an external loopback plug. This means all the design, that is functional in copper mode, is involved in the loopback. If connected at 10/100 Mbps, the loopback will work without any special setup. For 1000 Mbps operation, the following flow should be used:

3.7.6.5.1 Setting the I350 Internal PHY to External Loopback Mode

The following procedure should be used to put the I350 internal PHY into external loopback mode:

- Connect the external loopback cable to the port
- Set Link mode to PHY: *CTRL_EXT.LINK_MODE* = 00b
- In PHY Loopback Control Register - PHLBKC (Address 19d in the PHY):
  - Set External Cable mode (bit 7)
- In PHY Control Register (Address 0 in the PHY):
  - Restart auto-negotiation (Set bit 9)
- Wait for auto-negotiation to complete, then transmit and receive normally.

3.7.6.6 Line Loopback

In line loopback (Figure 3-6), MAC and SerDes interfaces are not functional, and the data is sent from a link partner to the PHY to test transmit and receive data paths. Frames that originate from a link partner are looped back from the PHY and sent out on the wire before reaching the MAC interface pins.

The following should be confirmed before enabling the line loopback feature:

- The PHY must first establish a full-duplex link with another PHY link partner, either through autonegotiation or through forcing the same link speed.

To enable line loopback mode once the link is established, set bit 9 to 1b in the Loopback Control Register - PHLBKC (19d; R/W) (see Section 8.26.3.17).

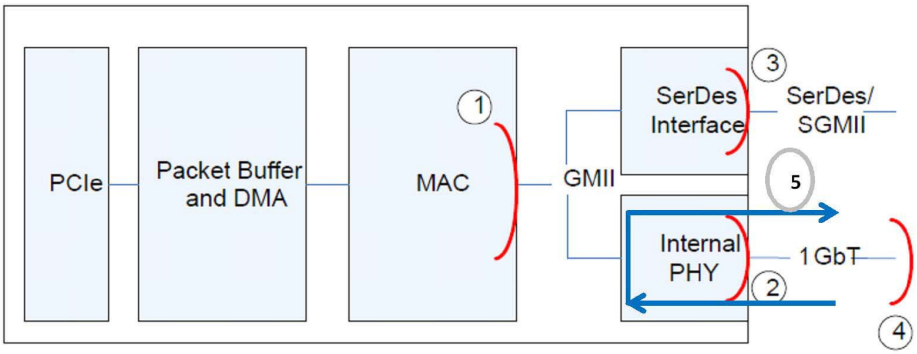


Figure 3-6 Line Loopback

3.7.7 Energy Efficient Ethernet (EEE)

EEE (Energy Efficient Ethernet) Low Power Idle (LPI) mode defined in IEEE802.3az optionally allows power saving by switching off part of the I350 functionality when no data needs to be transmitted or/ and received. Decision on whether the I350 transmit path should enter Low Power Idle mode or exit Low Power Idle mode is done according to need to transmit. Information on whether Link Partner has entered Low Power Idle mode is detected by the I350 and utilized for power saving in the receive circuitry.

When no data needs to be transmitted, a request to enter transmit Low Power Idle is issued on the internal xxMII TX interface causing the PHY to transmit sleep symbols for a predefined period of time followed by a quite period. During LPI, the PHY periodically transmits refresh symbols that are used by the link partner to update adaptive filters and timing circuits in order to maintain link integrity. This quiet-refresh cycle continues until transmission of 'normal inter-frame' encoding on the internal xxMII interface. The PHY communicates to the link partner the move to active link state by sending Wake symbols for a predefined period of time. The PHY then enters normal operating state where data or idle symbols are transmitted.

In the receive direction, entering Low Power Idle mode is triggered by the reception of sleep symbols from the link partner. This signals that the link partner is about to enter Low Power Idle mode. After sending the sleep symbols, the link partner ceases transmission. When Link partner enters LPI the PHY indicates "assert low power idle" on the internal xxMII RX interface and the the I350 receiver disables certain functionality to reduce power consumption.

Figure 3-7 and Table 3-30 illustrate general principles of EEE LPI operation on the Ethernet Link.

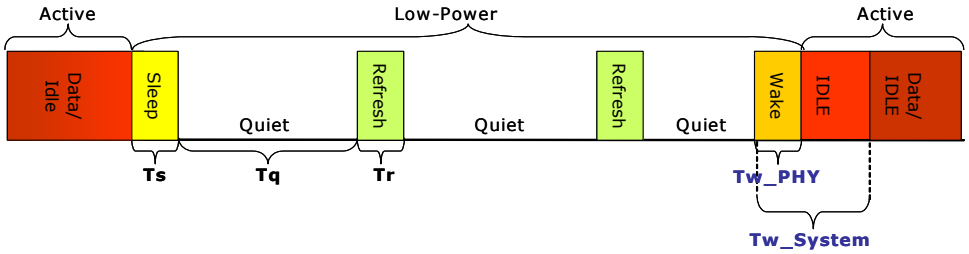


Figure 3-7 Energy Efficient Ethernet Operation

Table 3-30 Energy Efficient Ethernet Parameters

Parameter	Description
Sleep Time (Ts)	Duration PHY sends Sleep symbols before going Quiet.
Quiet Duration (Tq)	Duration PHY remains Quiet before it must wake for Refresh period.
Refresh Duration (Tr)	Duration PHY sends Refresh symbols for timing recovery and coefficient synchronization.
PHY Wake Time (Tw_PHY)	Minimum duration PHY takes to resume to Active state after decision to Wake.
Receive System Wake Time (Tw_System_rx)	Wait period where no data is expected to be received to give the local receiving system time to wake up.
Transmit System Wake Time (Tw_System_tx)	Wait period where no data is transmitted to give the remote receiving system time to wake up.

3.7.7.1 Conditions to Enter EEE TX LPI

In the transmit direction when network interface is internal copper PHY (*CTRL_EXT.LINK_MODE* = 00b), entry into to EEE Low Power Idle (LPI) mode of operation is triggered when one of the following conditions exist:

1. No transmission is pending, Management does not need to transmit and internal Transmit buffer is empty and *EEER.TX_LPI_EN* is set to 1.
2. If the *EEER.TX_LPI_EN* and *EEER.LPI_FC* bits are set to 1 and a XOFF flow control packet is received from the Link partner the I350 will move the link into TX LPI state for the Pause duration even if transmission is pending.
3. When *EEER.Force_TLPI* is set (even if *EEER.TX_LPI_EN* is cleared).
   — If *EEER.Force_TLPI* is set in mid-packet the I350 will complete packet transmission and then move TX to LPI.
   — Setting the *EEER.Force_TLPI* bit to 1 only stops transmission of packets from the Host. The I350 will move link out of TX LPI to transmit packets from the Management even when *EEER.Force_TLPI* is set to 1.
4. When function enters D3 state and there's no Management TX traffic, internal transmit buffers are empty and *EEER.TX_LPI_EN* is set to 1.

When one of the above conditions to enter TX LPI state is detected "assert low power idle" is transmitted on the internal xxMII interface and the I350 PHY transmits Sleep symbols on the network interface to communicate to the link partner entry into TX Low Power Idle link state. After Sleep symbols transmission, behavior of PHY differs according to link speed (100BASE-TX or 1000BASE-T):

1. In 100BASE-TX PHY enters low power operation in an asymmetric manner. After Sleep symbols transmission, the PHY immediately enters a low power quiet state.
2. In 1000BASE-T PHY entry into quiet state is symmetric. Only after PHY transmits sleep symbols and receives sleep symbols from the remote PHY does the PHY enter the quiet state.

After entry into quiet link state the PHY periodically transitions between quiet link state, where link is idle, to sending refresh symbols until a request to transition link back to normal (active) mode is transmitted on the internal xxMII TX interface (see Figure 3-7).

Note: MAC entry into TX LPI state is always asymmetric (in both 100BASE-TX and 1000BASE-T PHY operating modes).

3.7.7.2 Exit of TX LPI to Active Link State

The I350 will exit TX LPI link state and transition link into active link state when none of the conditions defined in Section 3.7.7.1 exist. To transition into active link state the I350 transmits:

1. Normal 'inter-frame' encoding on the internal xxMII TX interface for a pre-defined link rate dependant period time of Tw_sys_tx-min. As a result PHY will transmit Wake symbols for a Tw_phy duration followed by Idle symbols.
2. If the Tw_System_tx duration defined in the *EEER.Tw_system* field is longer than Tw_sys_tx-min the I350 will continue transmitting the 'inter-frame' encoding on the internal xxMII interface until the time defined in the *EEER.Tw_system* field has expired, before transmitting the actual data. During this period the PHY will continue transmitting Idle symbols.

Note: When moving out of TX LPI to transmit a 802.3x flow control frame the I350 will wait only the Tw_sys_tx-min duration before transmitting the flow control frame. It should be noted that even in this scenario, actual data will be transmitted only after the Tw_System_tx time defined in the *EEER.Tw_system* field has expired.

3.7.7.3 EEE Auto-Negotiation

Auto-Negotiation provides the capability to negotiate Energy Efficient Ethernet capabilities with the Link partner using the Next page mechanism defined in IEEE802.3 Annex 28C. IEEE802.3 Auto-Negotiation is performed at power up, on command from SW, upon detection of a PHY error or following link re-connection.

During the link establishment process, both link partners indicate their EEE capabilities via the IEEE802.3 Auto-negotiation process. If EEE is supported by both link partners for the negotiated PHY type then the EEE function may be used independently in either direction.

When operating in Internal PHY mode (*CTRL_EXT.LINK_MODE = 00b*), the I350 supports EEE auto-negotiation. EEE capabilities advertised during Auto-negotiation can be modified via the *EEE advertisement* field in the internal PHY (refer to Section 8.26.3.15) or via the *EEER.EEE_1G_AN* and *EEER.EEE_100M_AN* bits.

3.7.7.4 EEE Link Level (LLDP) Capabilities Discovery

When operating in Internal PHY mode (*CTRL_EXT.LINK_MODE* = 00b), the I350 supports LLDP negotiation via software, using the EEE IEEE802.1AB LLDP (Link Layer Discovery Protocol) Type, Length, Value (TLV) fields defined in IEEE802.3az clause 78 and clause 79. LLDP negotiation enables negotiation of increased System wake time (Transmit T_w and Receive T_w) to enable improving system energy efficiency.

After software negotiates a new System Wake Time Via EEE LLDP negotiation, software should update the:

1. EEER.Tw_system field with the negotiated Transmit T_w time value, to increase the duration where idle symbols are transmitted following move out of EEE TX LPI state before actual data can be transmitted.
   — Value placed in *EEER.Tw_system* field does not affect transmission of flow control packets. Depending on the technology (100BASE-TX or 1000BASE-T) flow control packet transmission is delayed following move out of EEE TX LPI state only by the minimum Tw_sys_tx time as defined in IEEE802.3az clause 78.5.
2. The *LTRMAXV* register with a value:

   ```
   LTRMINV =< LTRMAXV <= LTRMINV + negotiated Receive Tw Time.
   ```

3. Set *LTRC.EEEMS_EN* bit to 1 (if bit was cleared), so that on detection of EEE RX LPI on the network an updated LTR message with the value programmed in the *LTRMAXV* register will be sent on the PCIe interface.
4. Set *EEER.TX_LPI_EN* bit to 1 (if bit was cleared), to enable entry into EEE LPI on TX path.

Set *EEER.RX_LPI_EN* bit to 1 (if bit was cleared), to enable detection of link partner entering EEE LPI state on RX path.Once the *LTRC.EEEMS_EN* bit is set and a port detects link partner entry into the EEE LPI state on the internal xxMII RX interface, the port will increase its reported latency tolerance to the value programed in the *LTRMAXV* register. If all ports have increased latency tolerance, then on detection of the RX EEE LPI state an updated LTR message will be sent on the PCIe interface.

When Wake symbols are detected on the Ethernet Link, due to link partner moving out of EEE RX LPI state, the port will report a reduced latency tolerance that equals the value placed in the *LTRMINV* register and the I350 will send on the PCIe interface a new LTR message with a reduced latency tolerance value of *LTRMINV.*

Note: If link is disconnected or Auto-negotiation is re-initiated, then the *LTRC.EEEMS_EN* bit is cleared by HW. Bit should be set to 1b by software following re-execution of an EEE LLDP negotiation.

Figure 3-8 shows the format of the EEE TLV, meaning of the various TLV parameters can be found in IEEE802.3az clause 78 and clause 79.

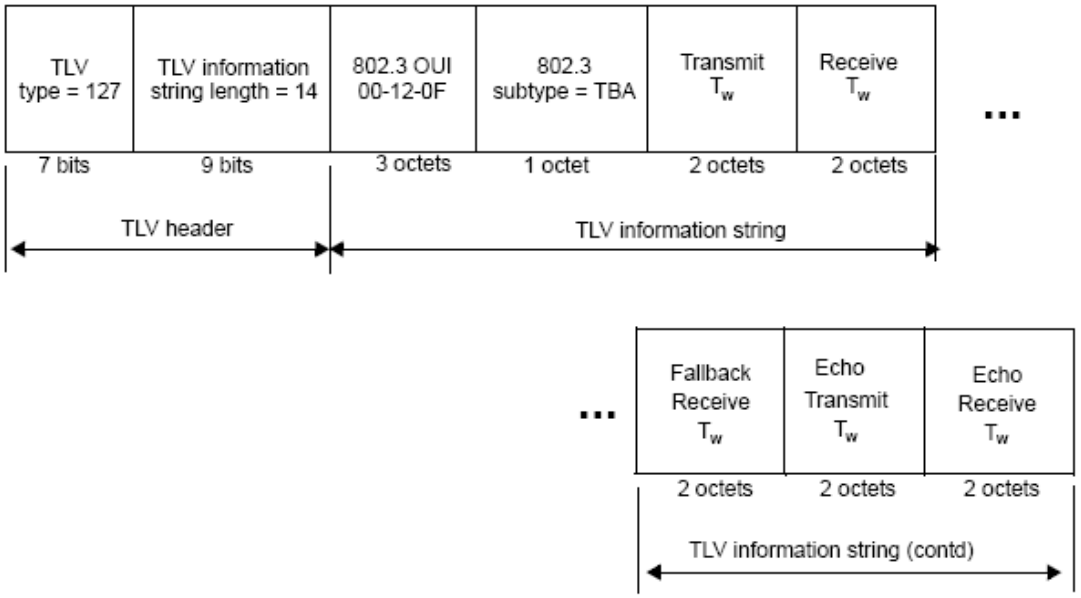


Figure 3-8 EEE LLDP TLV

3.7.7.5 Programming the I350 for EEE Operation

To activate EEE support when operating in Internal PHY mode (*CTRL_EXT.LINK_MODE* = 00b), software should program the following fields to enable EEE on a LAN port:

1. *IPCNFG* register (refer to Section 8.26.1) if default EEE advertised Auto-negotiation values need to be modified.
2. Set the *EEER.TX_LPI_EN and EEER.RX_LPI_EN* bits (refer to Section 8.24.10) to 1 to enable EEE LPI support on TX and RX paths respectively, if result of Auto-negotiation at the specified link speed enables entry to LPI.
3. Set the *EEER.LPI_FC* bit (refer to Section 8.24.10) if required to enable move into EEE TX LPI state for the Pause duration when link partner sends a XOFF Flow control packet even if internal Transmit buffer is not empty and transmit descriptors are available.
4. Update *EEER.Tw_system* field (refer to Section 8.24.10) with the new negotiated Transmit T_w time after completion of EEE LLDP negotiation.
5. Following EEE LLDP negotiation program the *LTRMAXV* register (refer to Section 8.24.8) with a value of:

   LTRMINV =< *LTRMAXV* <= *LTRMINV* + negotiated Receive T_w Time.

6. Set the *LTRC.EEEMS_EN* bit to 1b, to enable sending an updated PCIe Latency Tolerance Report (LTR) message when detecting link partner entry into EEE RX LPI state.

Notes:

1. The *LTRC.EEEMS_EN* bit is cleared following Link disconnect or Auto-negotiation and should be set to 1 by software following EEE LLDP re-negotiation.
2. The I350 waits for at least 1 second following Auto-negotiation (due to reset, Link disconnect or link speed change) and link-up indication (*STATUS.LU* set to 1, refer to Section 8.2.2) before enabling link entry into EEE TX LPI state to comply with the IEEE802.3az specification.

3.7.7.6 EEE Statistics

The I350 supports reporting number of EEE LPI TX and RX events via the *RLPIC* and *TLPIC* registers.

3.7.8 Integrated Copper PHY Functionality

The register set used to control the PHY functionality (PHYREG) is described in Section 8.26. the registers can be programmed using the *MDIC* register (refer to Section 8.2.4).

3.7.8.1 Determining Link State

The PHY and its link partner determine the type of link established through one of three methods:

- Auto-negotiation
- Parallel detection
- Forced operation

Auto-negotiation is the only method allowed by the 802.3ab standard for establishing a 1000BASE-T link, although forced operation could be used for test purposes. For 10/100 links, any of the three methods can be used. The following sections discuss each in greater detail.

Figure 3-9 provides an overview of link establishment. First the PHY checks if auto-negotiation is enabled. By default, the PHY supports auto-negotiation, see PHY Register 0, bit 12. If not, the PHY forces operation as directed. If auto-negotiation is enabled, the PHY begins transmitting Fast Link Pulses (FLPs) and receiving FLPs from its link partner. If FLPs are received by the PHY, auto-negotiation proceeds. It also can receive 100BASE-TX MLT3 and 10BASE-T Normal Link Pulses (NLPs). If either MLT3 or NLPs are received, it aborts FLP transmission and immediately brings up the corresponding half-duplex link.

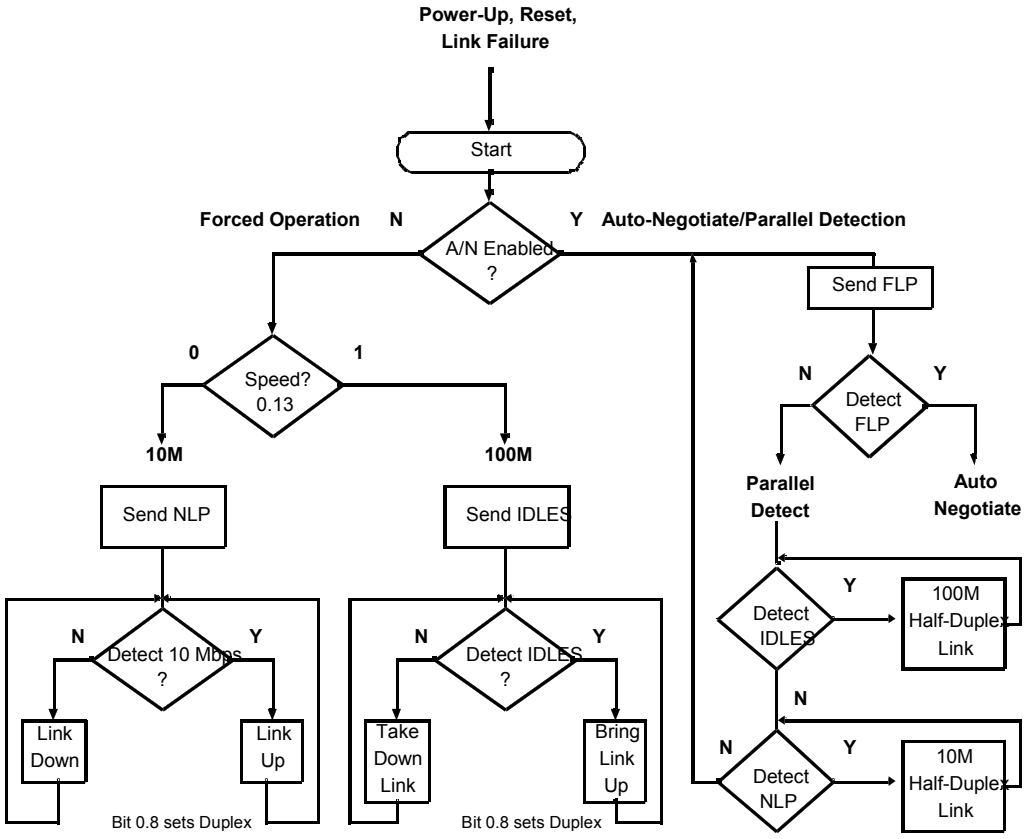


Figure 3-9 Overview of Link Establishment

3.7.8.1.1 False Link

The PHY does not falsely establish link with a partner operating at a different speed. For example, the PHY does not establish a 1 Gb/s or 10 Mb/s link with a 100 MB/s link partner.

When the PHY is first powered on, reset, or encounters a link down state; it must determine the line speed and operating conditions to use for the network link.

The PHY first checks the MDIO registers (initialized via the hardware control interface or written by software) for operating instructions. Using these mechanisms, designers can command the PHY to do one of the following:

- Force twisted-pair link operation to:
  - 1000T, full duplex
  - 1000T, half duplex
  - 100TX, full duplex
  - 100TX, half duplex
  - 10BASE-T, full duplex

  — 10BASE-T, half duplex
- Allow auto-negotiation/parallel-detection.

In the first six cases (forced operation), the PHY immediately begins operating the network interface as commanded. In the last case, the PHY begins the auto-negotiation/parallel-detection process.

3.7.8.1.2 Forced Operation

Forced operation can be used to establish 10 Mb/s and 100 Mb/s links, and 1000 Mb/s links for test purposes. In this method, auto-negotiation is disabled completely and the link state of the PHY is determined by MII Register 0.

Note: When speed is forced, the auto cross-over feature is not functional.

In forced operation, the designer sets the link speed (10, 100, or 1000 MB/s) and duplex state (full or half). For Gigabit (1000 MB/s) links, designers must explicitly designate one side as the master and the other as the slave.

Note: The paradox (per the standard): If one side of the link is forced to full-duplex operation and the other side has auto-negotiation enabled, the auto-negotiating partner parallel-detects to a half-duplex link while the forced side operates as directed in full-duplex mode. The result is spurious, unexpected collisions on the side configured to auto-negotiate.

Table 3-31 lists link establishment procedures.

Table 3-31 Determining Duplex State Via Parallel Detection

Configuration	Result
Both sides set for auto-negotiate	Link is established via auto-negotiation.
Both sides set for forced operation	No problem as long as duplex settings match.
One side set for auto-negotiation and the other for forced, half-duplex	Link is established via parallel detect.
One side set for auto-negotiation and the other for forced full-duplex	Link is established; however, sides disagree, resulting in transmission problems (Forced side is full-duplex, auto-negotiation side is half-duplex.).

3.7.8.1.3 Auto Negotiation

The PHY supports the IEEE 802.3u auto-negotiation scheme with next page capability. Next page exchange uses Register 7 to send information and Register 8 to receive them. Next page exchange can only occur if both ends of the link advertise their ability to exchange next pages.

3.7.8.1.4 Parallel Detection

Parallel detection can only be used to establish 10 and 100 Mb/s links. It occurs when the PHY tries to negotiate (transmit FLPs to its link partner), but instead of sensing FLPs from the link partner, it senses 100BASE-TX MLT3 code or 10BASE-T Normal Link Pulses (NLPs) instead. In this case, the PHY immediately stops auto-negotiation (terminates transmission of FLPs) and immediately brings up whatever link corresponds to what it has sensed (MLT3 or NLPs). If the PHY senses both technologies, the parallel detection fault is detected and the PHY continues sending FLPs.

With parallel detection, it is impossible to determine the true duplex state of the link partner and the IEEE standard requires the PHY to assume a half-duplex link. Parallel detection also does not allow exchange of flow-control ability (PAUSE and ASM_DIR) or the master/slave relationship required by 1000BASE-T. This is why parallel detection cannot be used to establish GbE links.

3.7.8.1.5 Auto Cross-Over

Twisted pair Ethernet PHY's must be correctly configured for MDI or MDI-X operation to inter operate. This has historically been accomplished using special patch cables, magnetics pinouts or Printed Circuit Board (PCB) wiring. The PHY supports the automatic MDI/MDI-X configuration originally developed for 1000Base-T and standardized in IEEE 802.3u section 40. Manual (non-automatic) configuration is still possible.

For 1000BASE-T links, pair identification is determined automatically in accordance with the standard.

For 10/100/1000 Mb/s links and during auto-negotiation, pair usage is determined by bits 9 and 10 in the *PHCTRL2* register (PHYREG18). The PHY activates an automatic cross-over detection function. If bit *PHCTRL2.Automatic MDI/MDI-X (*18.10) = 1b, the PHY automatically detects which application is being used and configures itself accordingly.

The automatic MDI/MDI-X state machine facilitates switching the MDI_PLUS[0] and MDI_MINUS[0] signals with the MDI_PLUS[1] and MDI_MINUS[1] signals, respectively, prior to the auto-negotiation mode of operation so that FLPs can be transmitted and received in compliance with Clause 28 auto-negotiation specifications. An algorithm that controls the switching function determines the correct polarization of the cross-over circuit. This algorithm uses an 11-Bit Linear Feedback Shift Register (LFSR) to create a pseudo-random sequence that each end of the link uses to determine its proposed configuration. After making the selection to either MDI or MDI-X, the node waits for a specified amount of time while evaluating its receive channel to determine whether the other end of the link is sending link pulses or PHY-dependent data. If link pulses or PHY-dependent data are detected, it remains in that configuration. If link pulses or PHY-dependent data are not detected, it increments its LFSR and makes a decision to switch based on the value of the next bit. The state machine does not move from one state to another while link pulses are being transmitted.

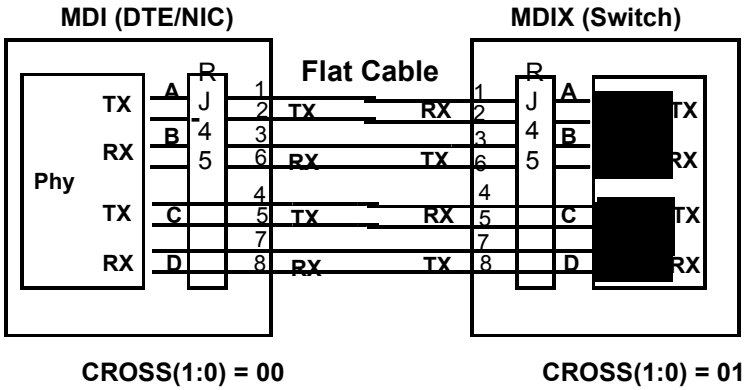


Figure 3-10 Cross-Over Function

3.7.8.1.6 10/100 MB/s Mismatch Resolution

It is a common occurrence that a link partner (such as a switch) is configured for forced full-duplex (FDX) 10/100 Mb/s operation. The normal auto-negotiation sequence would result in the other end settling for half-duplex (HDX) 10/100 Mb/s operation. The mechanism described in this section resolves the mismatch automatically and transitions the I350 into FDX mode, enabling it to operate with a partner configured for FDX operation.

The I350 enables the system software device driver to detect the mismatch event previously described and sets its duplex mode to the appropriate value without a need to go through another auto-negotiation sequence or breaking link. Once software detects a possible mismatch, it might instruct the I350 to change its duplex setting to either HDX or FDX mode. Software sets the *Duplex_manual_set* bit to indicate that duplex setting should be changed to the value indicated by the *Duplex Mode* bit in PHY Register 0. Any change in the value of the *Duplex Mode* bit in PHY Register 0 while the *Duplex_manual_set* bit is set to 1b would also cause a change in the device duplex setting.

The *Duplex_manual_set* bit is cleared on all PHY resets, following auto-negotiation, and when the link goes down. Software might track the change in duplex through the PHY *Duplex Mode* bit in Register 17 or a MAC indication.

3.7.8.1.7 Link Criteria

Once the link state is determined-via auto-negotiation, parallel detection or forced operation, the PHY and its link partner bring up the link.

3.7.8.1.7.1 1000BASE-T

For 1000BASE-T links, the PHY and its link partner enter a training phase. They exchange idle symbols and use the information gained to set their adaptive filter coefficients. These coefficients are used to equalize the incoming signal, as well as eliminate signal impairments such as echo and cross talk.

Either side indicates completion of the training phase to its link partner by changing the encoding of the idle symbols it transmits. When both sides so indicate, the link is up. Each side continues sending idle symbols each time it has no data to transmit. The link is maintained as long as valid idle, data, or carrier extension symbols are received.

3.7.8.1.7.2 100BASE-TX

For 100BASE-TX links, the PHY and its link partner immediately begin transmitting idle symbols. Each side continues sending idle symbols each time it has no data to transmit. The link is maintained as long as valid idle symbols or data is received.

In 100 Mb/s mode, the PHY establishes a link each time the scrambler becomes locked and remains locked for approximately 50 ms. Link remains up unless the descrambler receives less than 12 consecutive idle symbols in any 2 ms period. This provides for a very robust operation, essentially filtering out any small noise hits that might otherwise disrupt the link.

3.7.8.1.7.3 10BASE-T

For 10BASE-T links, the PHY and its link partner begin exchanging Normal Link Pulses (NLPs). The PHY transmits an NLP every 16 ms and expects to receive one every 10 to 20 ms. The link is maintained as long as normal link pulses are received.

In 10 Mb/s mode, the PHY establishes link based on the link state machine found in 802.3, clause 14.

Note: 100 Mb/s idle patterns do not bring up a 10 Mb/s link.

3.7.8.2 Link Enhancements

The PHY offers two enhanced link functions, each of which are discussed in the sections that follow:

- SmartSpeed

- Flow control

3.7.8.2.1 SmartSpeed

SmartSpeed is an enhancement to auto-negotiation that enables the PHY to react intelligently to network conditions that prohibit establishment of a 1000BASE-T link, such as cable problems. Such problems might allow auto-negotiation to complete, but then inhibit completion of the training phase. Normally, if a 1000BASE-T link fails, the PHY returns to the auto-negotiation state with the same speed settings indefinitely. With SmartSpeed enabled by programming the *PHCNFG.Automatic Speed Downshift Mode* field (refer to Section 8.26.3.20), after a configurable number of failed attempts, as configured in the *PHCTRL1* register (bits 12:10 - refer to Section 8.26.3.21) the PHY automatically downgrades the highest ability it advertises to the next lower speed: from 1000 to 100 to 10 Mb/s. Once a link is established, and if it is later broken, the PHY automatically upgrades the capabilities advertised to the original setting. This enables the PHY to automatically recover once the cable plant is repaired.

3.7.8.2.1.1 Using SmartSpeed

SmartSpeed is enabled by programming the *PHCNFG.Automatic Speed Downshift Mode* field (refer to Section 8.26.3.20). When SmartSpeed downgrades the PHY advertised capabilities, it sets bit *PHINT.Automatic Speed Downshift* (*PHYREG.25.1* - refer to Section 8.26.3.23). When link is established, its speed is indicated in the *PHSTAT.Speed Status* field (PHYREG.26.9:8 - refer to Section 8.26.3.24). SmartSpeed automatically resets the highest-level auto-negotiation abilities advertised, if link is established and then lost.

The number of failed attempts allowed is configured in the *PHCTRL1* register (bits 12:10 - refer to Section 8.26.3.21).

Note: SmartSpeed and M/S fault - When SmartSpeed is enabled, the M/S (Master-Slave) number of Attempts Before Downshift is programmed to be less than 7, resolution is not given seven attempts to try to resolve M/S status (see IEEE 802.3 clause 40.5.2).

Time To Link with Smart Speed - in most cases, any attempt duration is approximately 2.5 seconds, in other cases it could take more than 2.5 seconds depending on configuration and other factors.

3.7.8.3 Flow Control

Flow control is a function that is described in Clause 31 of the IEEE 802.3 standard. It allows congested nodes to pause traffic. Flow control is essentially a MAC-to-MAC function. MACs indicate their ability to implement flow control during auto-negotiation. This ability is communicated through two bits in the auto-negotiation registers (*PHYREG.4.10* and *PHYREG.4.11*).

The PHY transparently supports MAC-to-MAC advertisement of flow control through its auto-negotiation process. Prior to auto-negotiation, the MAC indicates its flow control capabilities via *PHYREG.4.10* (Pause) and *PHYREG.4.11* (*ASM_DIR*). After auto-negotiation, the link partner's flow control capabilities are indicated in *PHYREG.5.10* and *PHYREG.5.11*.

There are two forms of flow control that can be established via auto-negotiation: symmetric and asymmetric. Symmetric flow control is for point-to-point links; asymmetric for hub-to-end-node connections. Symmetric flow control enables either node to flow-control the other. Asymmetric flow-control enables a repeater or switch to flow-control a DTE, but not vice versa.

Table 3-32 lists the intended operation for the various settings of *ASM_DIR* and *PAUSE*. This information is provided for reference only; it is the responsibility of the MAC to implement the correct function. The PHY merely enables the two MACs to communicate their abilities to each other.

Table 3-32 Pause And Asymmetric Pause Settings

ASM_DIR settings Local (PHYREG.4.10) and Remote (PHYREG.5.10)	Pause Setting - Local (PHYREG.4.9)	Pause Setting - Remote (PHYREG.5.9)	Result
Both ASM_DIR = 1b	1	1	Symmetric - Either side can flow control the other
	1	0	Asymmetric - Remote can flow control local only
	0	1	Asymmetric - Local can flow control remote
	0	0	No flow control
Either or both ASM_DIR = 0b	1	1	Symmetric - Either side can flow control the other
	Either or both = 0		No flow control

3.7.8.4 Management Data Interface

The PHY supports the IEEE 802.3 MII Management Interface also known as the Management Data Input/Output (MDIO) Interface. This interface enables upper-layer devices to monitor and control the state of the PHY. The MDIO interface consists of a physical connection, a specific protocol that runs across the connection, and an internal set of addressable registers.

The PHY supports the core 16-bit MDIO registers. Registers 0-10 and 15 are required and their functions are specified by the IEEE 802.3 specification. Additional registers are included for expanded functionality. Specific bits in the registers are referenced using an PHY REG X.Y notation, where X is the register number (0-31) and Y is the bit number (0-15). Refer to Section 8.26, "PHY Software Interface" on page 683.

3.7.8.5 Internal PHY Low Power Operation and Power Management

The Internal PHY incorporates numerous features to maintain the lowest power possible.

The PHY can be entered into a low-power state according to MAC control (Power Management controls) or via PHY Register 0. In either power down mode, the PHY is not capable of receiving or transmitting packets.

3.7.8.5.1 Power Down via the PHY Register

The PHY can be powered down using the control bit found in *PHYREG.0.11*. This bit powers down a significant portion of the port but clocks to the register section remain active. This enables the PHY management interface to remain active during register power down. The power down bit is active high. When the PHY exits software power-down (*PHYREG.0.11* = 0b), it re-initializes all analog functions, but retains its previous configuration settings.

3.7.8.5.2 Power Management State

The internal PHY is aware of the power management state. If the PHY is not in a power down state, then PHY behavior regarding several features are different depending on the power state, refer to Section 3.7.8.5.4.

3.7.8.5.3 Disable High Speed Power Saving Options

The I350 supports disabling 1000 Mb/s or both 1000 Mb/s and 100 Mb/s advertisement by the internal PHY regardless of the values programmed in the PHY *ANA* Register (address - 4d) and the PHY *GCON* Register (address - 9d).

This is for cases where the system doesn't support working in 1000 Mb/s or 100 Mb/s due to power limitations.

This option is enabled in the following *PHPM* register bits:

- *PHPM.Disable 1000 in non-D0a* - disable 1000 Mb/s when in non-D0a states only.
- *PHPM.Disable 100 in non-D0a* - disable 1000 Mb/s and 100 Mb/s when in non-D0a states only.
- *PHPM.Disable 1000* - disable 1000 Mb/s always.

Note: When Value of *PHPM.Disable 1000* bit is changed, PHY initiates Auto-negotiation without direct driver command.

3.7.8.5.4 Low Power Link Up - Link Speed Control

Normal Internal PHY speed negotiation drives to establish a link at the highest possible speed. The I350 supports an additional mode of operation, where the PHY drives to establish a link at a low speed. The link-up process enables a link to come up at the lowest possible speed in cases where power is more important than performance. Different behavior is defined for the D0 state and the other non-D0 states.

Table 3-33 lists link speed as function of power management state, link speed control, and GbE speed enabling:

Table 3-33 Link Speed vs. Power State

Power Management State	Low Power Link Up (*PHPM.1*, *PHPM.2*)	GbE Disable Bits		100M Disable Bit	PHY Speed Negotiation
		Disable 1000 (*PHPM.6*)	Disable 1000 in non-D0a (*PHPM.3*)	Disable 100 in non-D0a (*PHPM.9*)	
D0a	0, Xb	0b	X	X	PHY negotiates to highest speed advertised (normal operation).
		1b	X	X	PHY negotiates to highest speed advertised (normal operation), excluding 1000 Mb/s.
	1, Xb	0b	X	X	PHY goes through Low Power Link Up (LPLU) procedure, starting with advertised values.
		1b	X	X	PHY goes through LPLU procedure, starting with advertised values. Does not advertise 1000 Mb/s.
Non-D0a	X, 0b	0b	0b	0b	PHY negotiates to highest speed advertised.
		0b	1b	0b	PHY negotiates to highest speed advertised, excluding 1000 Mb/s.
		1b	X	0b	
		X	X	1b	PHY negotiates and advertises only 10 Mb/s
	X, 1b	0b	0b	0b	PHY goes through LPLU procedure, starting at 10 Mb/s.
		0b	1b	0b	PHY goes through LPLU procedure, starting at 10 Mb/s. Does not advertise 1000 Mb/s.
		X	X	1b	PHY negotiates and advertises only 10 Mb/s

The Internal PHY initiates auto-negotiation without a direct driver command in the following cases:

- When the *PHPM.Disable 1000 in non-D0a* bit is set and 1000 Mb/s is disabled on D3 or Dr entry (but not in D0a), the PHY auto-negotiates on entry.
- When the *PHPM.Disable 100 in non-D0a* is set and 1000 Mb/s and 100 Mb/s are disabled on D3 or Dr entry (but not in D0a), the PHY auto-negotiates on entry.
- When *PHPM.LPLU* changes state with a change in a power management state. For example, on transition from D0a without *PHPM.LPLU* to D3 with *PHPM.LPLU*. Or, on transition from D3 with *PHPM.LPLU* to D0 without *LPLU*.
- On a transition from D0a state to a non-D0a state, or from a non-D0a state to D0a state, and *PHPM.LPLU* is set.

Notes:

- The Low-Power Link-Up (LPLU) feature previously described should be disabled (in both D0a state and non-D0a states) when the intended advertisement is anything other than 10 Mb/s only, 10/100 Mb/s only, or 10/100/1000 Mb/s. This is to avoid reaching (through the LPLU procedure) a link speed that is not advertised by the user.
- When the LAN PCIe Function is disabled via the *LAN_PCI_DIS* bit in the *Software Defined Pins Control* EEPROM word the relevant Function is in a Non-D0a state. As a result Management might operate with reduced link speed if the *LPLU*, *Disable 1000 in Non-D0a* or *Disable 100 in Non-D0a* EEPROM bits are set and the *MANC.Keep_PHY_Link_Up* bit (also known as "Veto bit") is cleared.
- When the *Keep_PHY_Link_Up* bit (also known as "veto bit") in the *MANC* Register is set, The PHY does not change its link speed as a result of a change in the device power state (e.g. move to D3).

3.7.8.5.4.1 D0a State

A power-managed link speed control lowers link speed (and power) when highest link performance is not required. When enabled (D0 Low Power Link Up mode), any link negotiation tries to establish a low-link speed, starting with an initial advertisement defined by software.

The D0LPLU configuration bit enables *D0 Low Power Link Up*. Before enabling this feature, software must advertise to one of the following speed combinations: 10 Mb/s only, 10/100 Mb/s only, or 10/100/1000 Mb/s.

When speed negotiation starts, the PHY tries to negotiate at a speed based on the currently advertised values. If link establishment fails, the PHY tries to negotiate with different speeds; it enables all speeds up to the lowest speed supported by the partner. For example, PHY advertises 10 Mb/s only, and the partner supports 1000 Mb/s only. After the first try fails, the PHY enables 10/100/1000 Mb/s and tries again. The PHY continues to try and establish a link until it succeeds or until it is instructed otherwise. In the second step (adjusting to partner speed), the PHY also enables parallel detect, if needed. Automatic MDI/MDI-X resolution is done during the first auto-negotiation stage.

3.7.8.5.4.2 Non-D0a State

The PHY might negotiate to a low speed while in non-D0a states (Dr, D0u, D3). This applies only when the link is required by one of the following: Manageability, APM Wake, or PME. Otherwise, the PHY is disabled during the non-D0 state.

The *Low Power on Link-Up* (Register *PHPM.LPLU*, is also loaded from EEPROM) bit enables reduction in link speed:

- At power-up entry to Dr state, the PHY advertises supports for 10 Mb/s only and goes through the link up process.

- At any entry to a non-D0a state (Dr, D0u, D3), the PHY advertises support for 10 Mb/s only and goes through the link up process.
- While in a non-D0 state, if auto-negotiation is required, the PHY advertises support for 10 Mb/s only and goes through the link up process.

Link negotiation begins with the PHY trying to negotiate at 10 Mb/s speed only regardless of user auto-negotiation advertisement. If link establishment fails, the PHY tries to negotiate at additional speeds; it enables all speeds up to the lowest speed supported by the partner. For example, the PHY advertises 10 Mb/s only and the partner supports 1000 Mb/s only. After the first try fails, PHY enables 10/100/1000 Mb/s and tries again. The PHY continues to try and establish a link until it succeeds or until it is instructed otherwise. In the second step (adjusting to partner speed), the PHY also enables parallel detect, if needed. Automatic MDI/MDI-X resolution is done during the first auto-negotiation stage.

3.7.8.5.5 Internal PHY Smart Power-Down (SPD)

Smart power-down is a link-disconnect capability applicable to all power management states. Smart Power-down combines a power saving mechanism with the fact that the link might disappear and resume.

Smart power-down is enabled by *PHPM.SPD_EN* or by *SPD Enable* bit in the EEPROM if the following conditions are met:

1. Auto-negotiation is enabled.
2. PHY detects link loss.

While in the smart power-down state, the PHY powers down circuits and clocks that are not required for detection of link activity. The PHY is still be able to detect link pulses (including parallel detect) and wake-up to engage in link negotiation. The PHY does not send link pulses (NLP) while in SPD state; however, register accesses are still possible.

When the Internal PHY is in smart power-down mode and detects link activity, it re-negotiates link speed based on the power state and the *Low Power Link Up* bits as defined by the *PHPM.D0LPLU* and *PHPM.LPLU* bits.

Note: The PHY does not enter the SPD state unless auto-negotiation is enabled.

While in the SPD state, the PHY powers down all circuits not required for detection of link activity. The PHY must still be able to detect link pulses (including parallel detect) and wake up to engage in link negotiation. The PHY does not send link pulses (NLP) while in SPD state.

Notes: While in the link-disconnect state, the PHY must allow software access to its registers.
The link-disconnect state applies to all power management states (Dr, D0u, D0a, D3).
The link might change status, that is go up or go down, while in any of these states.

3.7.8.5.5.1 Internal PHY Back-to-Back Smart Power-Down

While in link disconnect, the I350 monitors the link for link pulses to identify when a link is re-connected. The I350 also periodically transmits pulses (every 100 ms) to resolve the case of two I350 devices (or devices with I350-like behavior) connected to each other across the link. Otherwise, two such devices might be locked in Smart power-down mode, not capable of identifying that a link was re-connected.

Back-to-back smart power-down is enabled by the *SPD_B2B_EN* bit in the *PHPM* register. The default value is enabled. The *Enable* bit applies to smart power-down mode.

Note: This bit should not be altered by software once the I350 was set in smart power-down mode. If software requires changing the back-to-back status, it first needs to transition the PHY out of smart power-down mode and only then change the back-to-back bit to the required state.

3.7.8.5.6 Internal PHY Link Energy Detect

The I350 asserts the *Link Energy Detect* bit (*PHPM.Link Energy Detect*) each time energy is not detected on the link. This bit provides an indication of a cable becoming plugged or unplugged.

This bit is valid only if *PHPM.Go Link disconnect* is set to 1b.

In order to correctly deduce that there is no energy, the bit must read 0b for three consecutive reads each second.

3.7.8.5.7 Internal PHY Power-Down State

The I350 ports 0 to 3 enter a power-down state when none of the port's clients are enabled and therefore the internal PHY has no need to maintain a link. This can happen in one of the following cases, if the Internal PHY power-down functionality is enabled through the EEPROM PHY *Power Down Enable* bit.

1. D3/Dr state: Each Internal PHY enters a low-power state if the following conditions are met:
   a. The LAN function associated with this PHY is in a non-D0 state
   b. APM WOL is inactive
   c. Manageability doesn't use this port.
   d. ACPI PME is disabled for this port.
   e. The PHY *Power Down Enable* EEPROM bit is set (*Initialization Control Word 2*, word 0xF, bit 6).
2. SerDes mode: Each Internal PHY is disabled when its LAN function is configured to SerDes mode.
3. LAN disable: Each Internal PHY can be disabled if its LAN function's LAN Disable input indicates that the relevant function should be disabled. Since the PHY is shared between the LAN function and manageability, it might not be desirable to power down the PHY in LAN Disable. The *PHY_in_LAN_Disable* EEPROM bit determines whether the PHY (and MAC) are powered down when the LAN Disable pin is asserted. The default is not to power down.

A LAN port can also be disabled through EEPROM settings. If the *LAN_DIS* EEPROM bit is set, the Internal PHY enters power down.

Note: Setting the EEPROM *LAN_PCI_DIS* bit does not move the internal PHY into power down. However if the *LPLU, Disable 1000 in Non-D0a* or the *Disable 100 in Non-D0a* EEPROM bits are set and the *MANC.Keep_PHY_Link_Up* bit is cleared Management may operate with reduced link speed since the Function is in a Non-D0a (uninitialized) state.

3.7.8.6 Advanced Diagnostics

The I350 Integrated PHY incorporates hardware support for advanced diagnostics.

The hardware support enables output of internal PHY data to host memory for post processing by the software device driver.

The current diagnostics supported are:

3.7.8.6.1 TDR - Time Domain Reflectometry

By sending a pulse onto the twisted pair and observing the retuned signal, the following can be deduced:

1. Is there a short?
2. Is there an open?
3. Is there an impedance mismatch?
4. What is the length to any of these faults?

3.7.8.6.2 Channel Frequency Response

By doing analysis on the Tx and Rx data, it can be established that a channel's frequency response (also known as insertion loss) can determine if the channel is within specification limits. (Clause 40.7.2.1 in IEEE 802.3).

3.7.8.7 1000 Mb/s Operation

3.7.8.7.1 Introduction

Figure 3-11 shows an overview of 1000BASE-T functions, followed by discussion and review of the internal functional blocks.

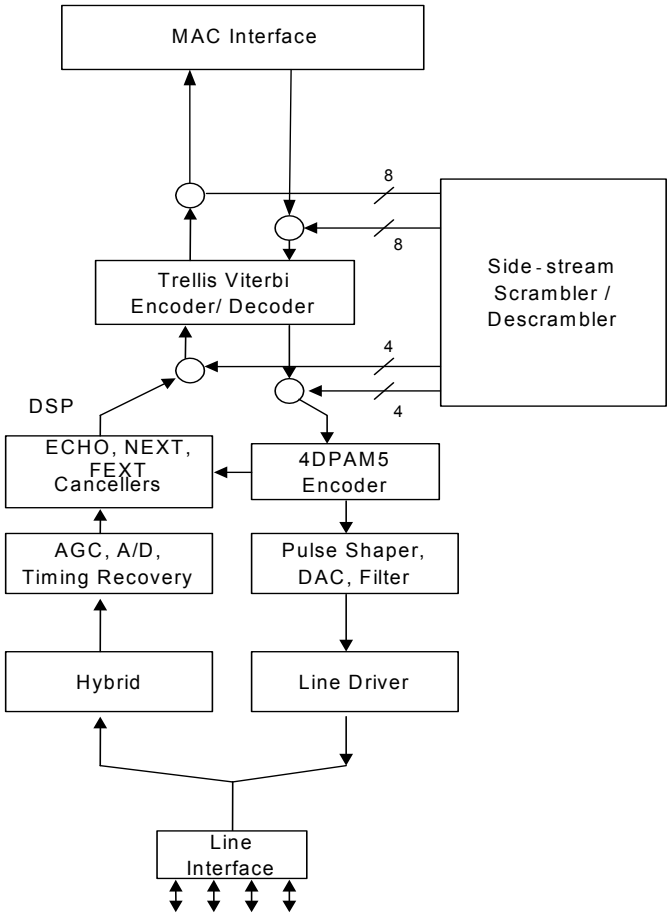


Figure 3-11 1000BASE-T Functions Overview

3.7.8.7.2 Transmit Functions

This section describes functions used when the Media Access Controller (MAC) transmits data through the PHY and out onto the twisted-pair connection (see Figure 3-11).

3.7.8.7.2.1 Scrambler

The scrambler randomizes the transmitted data. The purpose of scrambling is twofold:

1. Scrambling eliminates repeating data patterns (also known as spectral lines) from the 4DPAM5 waveform in order to reduce EMI.
2. Each channel (A, B, C, D) has a unique signature that the receiver uses for identification.

The scrambler is driven by a 33-bit Linear Feedback Shift Register (LFSR), which is randomly loaded at power up. The LFSR function used by the master differs from that used by the slave, giving each direction its own unique signature. The LFSR, in turn, generates twelve mutually uncorrelated outputs. Eight of these are used to randomize the inputs to the 4DPAM5 and Trellis encoders. The remaining four outputs randomize the sign of the 4DPAM5 outputs.

3.7.8.7.2.2 Transmit FIFO

The transmit FIFO re-synchronizes data transmitted by the MAC to the transmit reference used by the PHY. The FIFO is large enough to support a frequency differential of up to +/- 1000 ppm over a packet size of 10 KB (jumbo frame).

3.7.8.7.2.3 Transmit Phase-Locked Loop PLL

This function generates the 125 MHz timing reference used by the PHY to transmit 4DPAM5 symbols. When the PHY is the master side of the link, the XI input is the reference for the transmit PLL. When the PHY is the slave side of the link, the recovered receive clock is the reference for the transmit PLL.

3.7.8.7.2.4 Trellis Encoder

The Trellis encoder uses the two high-order bits of data and its previous output to generate a ninth bit, which determines if the next 4DPAM5 pattern should be even or odd.

For data, this function is:

```
Trellisn = Data7n-1 XOR Data6n-2 XOR Trellisn-3
```

This provides forward error correction and enhances the Signal-To-Noise (SNR) ratio by a factor of 6 dB.

3.7.8.7.2.5 4DPAM5 Encoder

The 4DPAM5 encoder translates 8-byte codes transmitted by the MAC into 4DPAM5 symbols. The encoder operates at 125 MHz, which is both the frequency of the MAC interface and the baud rate used by 1000BASE-T.

Each 8-byte code represents one of 28 or 256 data patterns. Each 4DPAM5 symbol consists of one of five signal levels (-2,-1,0,1,2) on each of the four twisted pair (A,B,C,D) representing 54 or 625 possible patterns per baud period. Of these, 113 patterns are reserved for control codes, leaving 512 patterns for data. These data patterns are divided into two groups of 256 even and 256 odd data patterns. Thus, each 8-byte octet has two possible 4DPAM5 representations: one even and one odd pattern.

3.7.8.7.2.6 Spectral Shaper

This function causes the 4DPAM5 waveform to have a spectral signature that is very close to that of the MLT3 waveform used by 100BASE-TX. This enables 1000BASE-T to take advantage of infrastructure (cables, magnetics) designed for 100BASE-TX.

The shaper works by transmitting 75% of a 4DPAM5 code in the current baud period, and adding the remaining 25% into the next baud period.

3.7.8.7.2.7 Low-Pass Filter

To aid with EMI, this filter attenuates signal components more than 180 MHz. In 1000BASE-T, the fundamental symbol rate is 125 MHz.

3.7.8.7.2.8 Line Driver

The line driver drives the 4DPAM5 waveforms onto the four twisted-pair channels (A, B, C, D), adding them onto the waveforms that are simultaneously being received from the link partner.

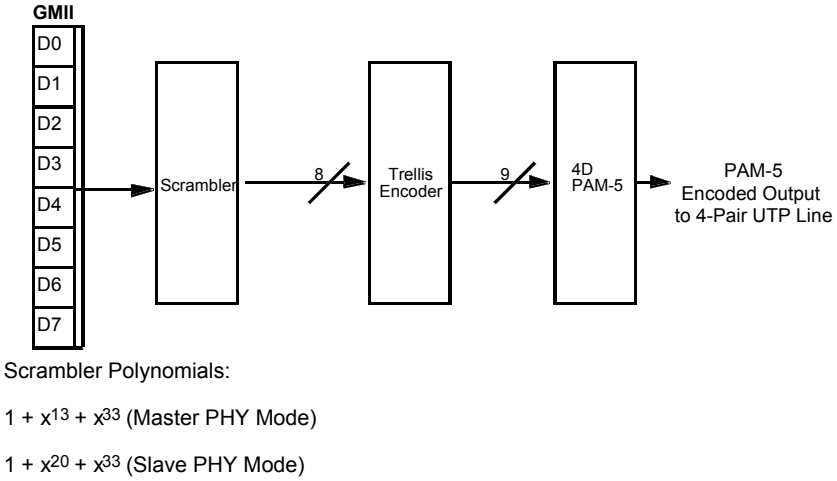


Figure 3-12 1000BASE-T Transmit Flow And Line Coding Scheme

Figure 3-13 Transmit/Receive Flow

3.7.8.7.3 Receive Functions

This section describes function blocks that are used when the PHY receives data from the twisted pair interface and passes it back to the MAC (see Figure 3-13).

3.7.8.7.3.1 Hybrid

The hybrid subtracts the transmitted signal from the input signal, enabling the use of simple 100BASE-TX compatible magnetics.

3.7.8.7.3.2 Automatic Gain Control (AGC)

AGC normalizes the amplitude of the received signal, adjusting for the attenuation produced by the cable.

3.7.8.7.3.3 Timing Recovery

This function re-generates a receive clock from the incoming data stream which is used to sample the data. On the slave side of the link, this clock is also used to drive the transmitter.

3.7.8.7.3.4 Analog-to-Digital Converter (ADC)

The ADC function converts the incoming data stream from an analog waveform to digitized samples for processing by the DSP core.

3.7.8.7.3.5 Digital Signal Processor (DSP)

DSP provides per-channel adaptive filtering, which eliminates various signal impairments including:

- Inter-symbol interference (equalization)
- Echo caused by impedance mismatch of the cable
- Near-end crosstalk (NEXT) between adjacent channels (A, B, C, D)
- Far-end crosstalk (FEXT)
- Propagation delay variations between channels of up to 120 ns
- Extraneous tones that have been coupled into the receive path

The adaptive filter coefficients are initially set during the training phase. They are continuously adjusted (adaptive equalization) during operation through the decision-feedback loop.

3.7.8.7.3.6 Descrambler

The descrambler identifies each channel by its characteristic signature, removing the signature and re-routing the channel internally. In this way, the receiver can correct for channel swaps and polarity reversals. The descrambler uses the same base 33-bit LFSR used by the transmitter on the other side of the link.

The descrambler automatically loads the seed value from the incoming stream of scrambled idle symbols. The descrambler requires approximately 15 μs to lock, normally accomplished during the training phase.

3.7.8.7.3.7 Viterbi Decoder/Decision Feedback Equalizer (DFE)

The Viterbi decoder generates clean 4DPAM5 symbols from the output of the DSP. The decoder includes a Trellis encoder identical to the one used by the transmitter. The Viterbi decoder simultaneously looks at the received data over several baud periods. For each baud period, it predicts whether the symbol received should be even or odd, and compares that to the actual symbol received. The 4DPAM5 code is organized in such a way that a single level error on any channel changes an even code to an odd one

and vice versa. In this way, the Viterbi decoder can detect single-level coding errors, effectively improving the signal-to-noise (SNR) ratio by a factor of 6 dB. When an error occurs, this information is quickly fed back into the equalizer to prevent future errors.

3.7.8.7.3.8 4DPAM5 Decoder

The 4DPAM5 decoder generates 8-byte data from the output of the Viterbi decoder.

3.7.8.7.3.9 100 Mb/s Operation

The MAC passes data to the PHY over the MII. The PHY encodes and scrambles the data, then transmits it using MLT-3 for 100TX over copper. The PHY de-scrambles and decodes MLT-3 data received from the network. When the MAC is not actively transmitting data, the PHY sends out idle symbols on the line.

3.7.8.7.3.10 10 Mb/s Operation

The PHY operates as a standard 10 Mb/s transceiver. Data transmitted by the MAC as 4-bit nibbles is serialized, Manchester-encoded, and transmitted on the MDI[0]+/- outputs. Received data is decoded, de-serialized into 4-bit nibbles and passed to the MAC across the internal MII. The PHY supports all the standard 10 Mb/s functions.

3.7.8.7.3.11 Link Test

In 10 Mb/s mode, the PHY always transmits link pulses. If link test function is enabled, it monitors the connection for link pulses. Once it detects two to seven link pulses, data transmission are enabled and remain enabled as long as the link pulses or data reception continues. If the link pulses stop, the data transmission is disabled.

If the link test function is disabled, the PHY might transmit packets regardless of detected link pulses. Setting the Port Configuration register bit (PHYREG.16.14) can disable the link test function.

3.7.8.7.3.12 10Base-T Link Failure Criteria and Override

Link failure occurs if link test is enabled and link pulses stop being received. If this condition occurs, the PHY returns to the auto-negotiation phase, if auto-negotiation is enabled. Setting the Port Configuration register bit (PHYREG.16.14) disables the link integrity test function, then the PHY transmits packets, regardless of link status.

3.7.8.7.3.13 Jabber

If the MAC begins a transmission that exceeds the jabber timer, the PHY disables the transmit and loopback functions and asserts collision indication to the MAC. The PHY automatically exits jabber mode after 250-750 ms. This function can be disabled by setting bit PHYREG.16.10 = 1b.

3.7.8.7.3.14 Polarity Correction

The PHY automatically detects and corrects for the condition where the receive signal (MDI_PLUS[0]/ MDI_MINUS[0]) is inverted. Reversed polarity is detected if eight inverted link pulses or four inverted end-of-frame markers are received consecutively. If link pulses or data are not received for 96-130 ms, the polarity state is reset to a non-inverted state.

Automatic polarity correction can be disabled by setting bit PHYREG.27.5.

3.7.8.7.3.15 Dribble Bits

The PHY handles dribble bits for all of its modes. If between one and four dribble bits are received, the nibble is passed across the interface. The data passed across is padded with 1's if necessary. If between five and seven dribble bits are received, the second nibble is not sent onto the internal MII bus to the MAC. This ensures that dribble bits between 1-7 do not cause the MAC to discard the frame due to a CRC error.

3.7.8.7.3.16 PHY Address

The External PHY MDIO Address is defined in the *MDICNFG.PHYADD* field and is loaded at power-up from EEPROM. If the *MDICNFG.Destination* bit is cleared (Internal PHY), MDIO access is always to the internal PHY.

3.7.9 Media Auto Sense

The I350 provides a significant amount of flexibility in pairing a LAN device with a particular type of media (such as copper or fiber-optic) as well as the specific transceiver/interface used to communicate with the media. Each MAC, representing a distinct LAN device, can be coupled with an internal copper PHY or SerDes/SGMII/1000BASE-KX interface independently. The link configuration specified for each LAN device can be specified in the *LINK_MODE* field of the Extended Device Control (*CTRL_EXT*) register and initialized from the EEPROM Initialization Control Word 3 associated with each LAN Port.

In some applications, software might need to be aware of the presence of a link on the media not currently active. In order to supply such an indication, any of the I350 ports can set the *AUTOSENSE_EN* bit in the CONNSW register (address 0x0034) in order to enable sensing of the non active media activity.

Note: When in SerDes/SGMII/1000BASE-KX detect mode, software should define which indication is used to detect the energy change on the SerDes/SGMII/1000BASE-KX media. It can be either the external signal detect pin or the internal signal detect. This is done using the *CONNSW.ENRGSRC* bit. The signal detect pin is normally used when connecting in SerDes mode to optical media where the receive LED provide such an indication.

Software can then enable the *OMED* interrupt in *ICR* in order to get an indication on any detection of energy in the non active media.

Note: The auto-sense capability can be used in either port independent of the usage of the other port.

The following sections describes the procedures that should be followed in order to enable the auto-sense mode

3.7.9.1 Auto sense setup

3.7.9.1.1 SerDes/SGMII/1000BASE-KX Detect Mode (PHY is Active)

1. Set *CONNSW.ENRGSRC* to determine the sources for the signal detect indication (1b = external SIG_DET, 0b = internal SerDes electrical idle). The default of this bit is set by EEPROM.
2. Set *CONNSW.AUTOSENSE_EN*.

3. When link is detected on the SerDes/SGMII/1000BASE-KX media, the I350 sets the interrupt bit *OMED* in ICR and if enabled, issues an interrupt. The *CONNSW.AUTOSENSE_EN* bit is cleared.

3.7.9.1.2 PHY Detect Mode (SerDes/SGMII/1000BASE-KX is active)

1. Set *CONNSW.AUTOSENSE_CONF* = 1b.
2. Reset the PHY as described in Section 4.3.4.4.
3. Enable PHY to move to low power mode when cable is disconnected by setting the *PHPM.SPD_EN* bit.
4. Set *CONNSW.AUTOSENSE_EN* = 1b and then clear *CONNSW.AUTOSENSE_CONF*.
5. When signal is detected on the PHY media, the I350 sets the *ICR.OMED* interrupt bit and issues an interrupt if enabled.
6. The I350 puts the PHY in power down mode.

According to the result of the interrupt, software can then decide to switch to the other media.

Note: Assertion of *ICR.OMED* PHY is a one time event. To re-enable Auto-detect after cable is unplugged Software should clear the *CONNSW.AUTOSENSE_EN* bit and the procedure defined above should be executed again.

3.7.9.2 Switching between medias.

The I350's link mode is controlled by the *CTRL_EXT.LINK_MODE* field. The default value for the *LINK_MODE* setting is directly mapped from the EEPROM's *initialization Control Word 3 Link Mode* field. Software can modify the *LINK_MODE* indication by writing the corresponding value into this register.

Notes: Before dynamically switching between medias, the software should ensure that the current mode of operation is not in the process of transmitting or receiving data. This is achieved by disabling the transmitter and receiver, waiting until the I350 is in an idle state, and then beginning the process for changing the link mode.

The mode switch in this method is only valid until the next hardware reset of the I350. After a hardware reset, the link mode is restored to the default setting by the EEPROM. To get a permanent change of the link mode, the default in the EEPROM should be changed.

The following procedures need to be followed to actually switch between the two modes.

3.7.9.2.1 Transition to SerDes/1000BASE-KX/SGMII Modes

1. Disable the receiver by clearing *RCTL.RXEN*.
2. Disable the transmitter by clearing *TCTL.EN*.
3. Verify the I350 has stopped processing outstanding cycles and is idle.
4. Modify LINK mode to SerDes, 1000BASE-KX or SGMII by setting *CTRL_EXT.LINK_MODE* to 11b, 01b or 10b respectively.
5. Set up the link as described in Section 4.6.7.3,Section 4.6.7.4 or Section 4.6.7.5.
6. Set up Tx and Rx queues and enable Tx and Rx processes.

3.7.9.2.2 Transition to Internal PHY Mode

1. Disable the receiver by clearing *RCTL.RXEN*.
2. Disable the transmitter by clearing *TCTL.EN*.

3. Verify the I350 has stopped processing outstanding cycles and is idle.
4. Modify LINK mode to PHY mode by setting *CTRL_EXT.LINK_MODE* to 00b.
5. Set link-up indication by setting *CTRL.SLU*.
6. Reset the PHY as described in Section 4.3.4.4.
7. Set up the link as described in Section 4.6.7.2.
8. Set up the Tx and Rx queues and enable the Tx and Rx processes.

§ §

4 Initialization

4.1 Power Up

4.1.1 Power-Up Sequence

Figure 4-1 shows the power-up sequence from power ramp up and to when the I350 is ready to accept host commands.

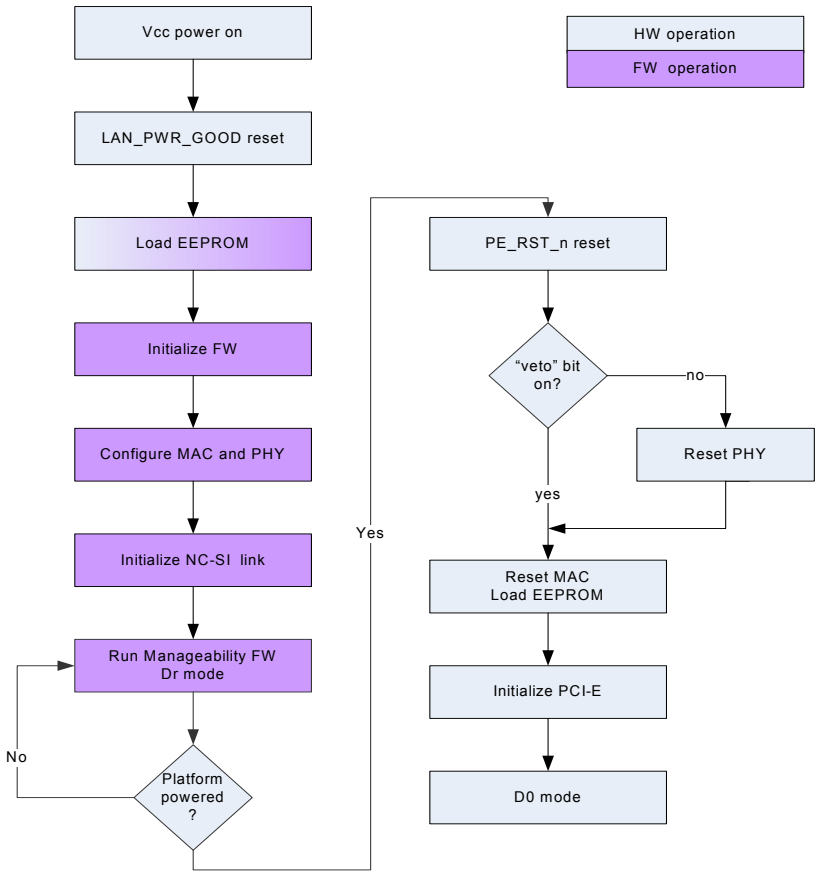


Figure 4-1 Power-Up - General Flow

Note: The Keep_PHY_Link_Up bit (*Veto* bit) is set by firmware when the BMC is running IDER or SoL. Its purpose is to prevent interruption of these processes when power is being turned on.

4.1.2 Power-Up Timing Diagram

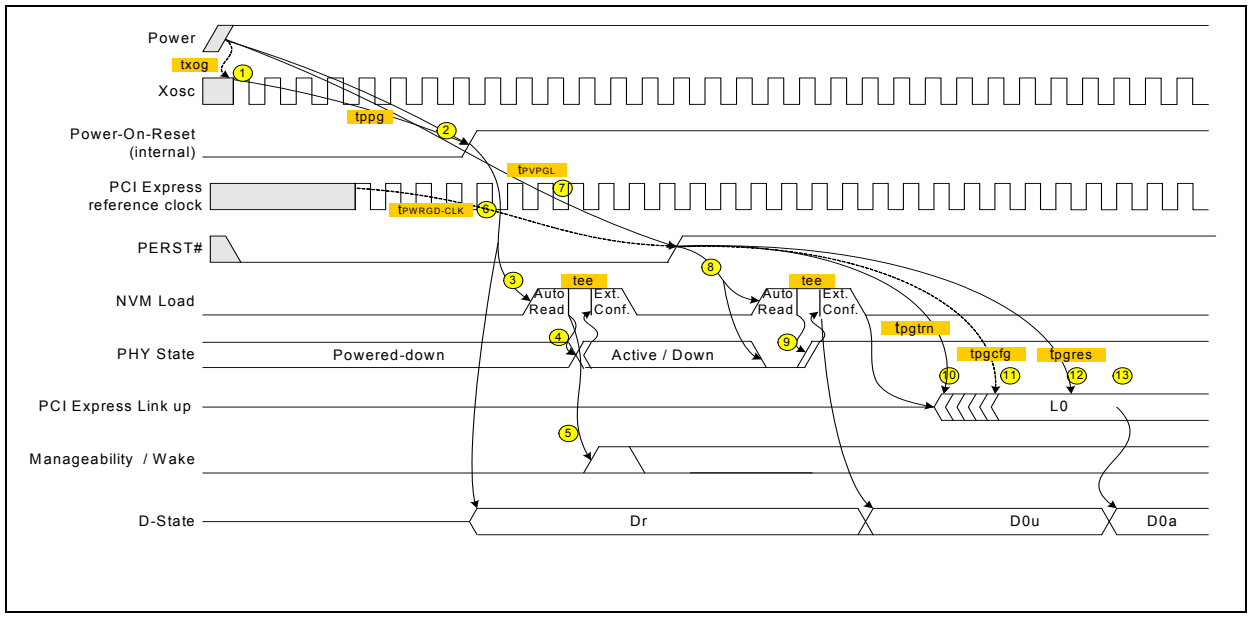


Figure 4-2 Power-Up Timing Diagram

Table 4-1 Notes to Power-Up Timing Diagram

Note	
1	Xosc is stable t_{xog} after the Power is stable
2	Internal Reset is released after all power supplies are good and t_{ppg} after Xosc is stable.
3	An EEPROM read starts on the rising edge of the internal Reset or LAN_PWR_GOOD.
4	After reading the EEPROM, PHY might exit power down mode.
5	APM Wakeup and/or manageability might be enabled based on EEPROM contents.
6	The PCIe reference clock is valid $t_{PE_RST-CLK}$ before the de-assertion of PE_RST# (according to PCIe spec).
7	PE_RST# is de-asserted t_{PVPGL} after power is stable (according to PCIe spec).
8	De-assertion of PE_RST# causes the EEPROM to be re-read, asserts PHY power-down (except if veto bit also known as Keep_PHY_Link_Up bit is set), and disables Wake Up.
9	After reading the EEPROM, PHY exits power-down mode.
10	Link training starts after t_{pgtrn} from PE_RST# de-assertion.
11	A first PCIe configuration access might arrive after t_{pgcfg} from PE_RST# de-assertion.
12	A first PCI configuration response can be sent after tpgres from PE_RST# de-assertion
13	Writing a 1 to the *Memory Access Enable* bit in the PCI *Command Register* transitions the device from D0u to D0 state.

4.2 Reset Operation

The I350 has a number of reset sources described below. Following the reset Software driver should verify that the *EEMNGCTL.CFG_DONE* bit (refer to Section 8.4.6.1) is set to 1b and no errors were reported in the *FWSM.Ext_Err_Ind* (refer to Section 8.7.2) field.

4.2.1 Reset Sources

The I350 reset sources are described below:

4.2.1.1 LAN_PWR_GOOD

The I350 has an internal mechanism for sensing the power pins. Once the power is up and stable, the I350 creates an internal reset. This reset acts as a master reset of the entire chip. It is level sensitive, and while it is zero holds all of the registers in reset. LAN_PWR_GOOD is interpreted to be an indication that device power supplies are all stable. LAN_PWR_GOOD changes state during system power-up.

4.2.1.2 PE_RST_N

The de-assertion of PE_RST_N indicates that both the power and the PCIe clock sources are stable. This pin asserts an internal reset also after a D3cold exit. Most units are reset on the rising edge of PE_RST_N. The only exception is the PCIe unit, which is kept in reset while PE_RST_N is asserted (level).

4.2.1.3 In-Band PCIe Reset

The I350 generates an internal reset in response to a Physical layer message from the PCIe or when the PCIe link goes down (entry to Polling or Detect state). This reset is equivalent to PCI reset in previous (PCI) gigabit LAN controllers.

4.2.1.4 D3hot to D0 Transition

This is also known as ACPI Reset. the I350 generates an internal reset on the transition from D3hot power state to D0 (caused after configuration writes from D3 to D0 power state). Note that this reset is per function and resets only the function that transitions from D3hot to D0 (refer to Section 5.2.3 for additional information).

When the *PMCSR.No_Soft_Reset* bit in the configuration space is set on transition from D3hot to D0 the I350 will reset internal CSRs (similar to *CTRL.RST* assertion) but will not reset registers in the PCIe configuration space. If the *PMCSR.No_Soft_Reset* bit is cleared the I350 will reset all per function registers except for registers defined as sticky in the configuration space.

Note: Regardless of the value of the *PMCSR.No_Soft_Reset* bit the function will be reset (including bits that are not defined as sticky in PCIe configuration space) if the Link state transitions to the L2/L3 Ready state, on transition from D3cold to D0, if FLR is asserted or if transition D3hot to D0 is caused by assertion of PCIe reset (PE_RST pin).

Note: Software drivers should implement the handshake mechanism defined in Section 5.2.3.3 to verify that all pending PCIe completions are done, before moving the I350 to D3.

4.2.1.5 Function Level Reset (FLR)

The FLR bit is required for the PF and per VF (Virtual Function). Setting of this bit for a VF resets only the part of the logic dedicated to the specific VF and does not influence the shared part of the port. Setting the PF FLR bit resets the entire function.

4.2.1.5.1 PF (Physical Function) FLR or FLR in non-IOV Mode

A FLR reset to a function is issued, by setting bit *15* in the *Device Control* configuration register (refer to Section 9.5.6.5), is equivalent to a D0 ⇨ D3 ⇨ D0 transition. The only difference is that this reset does not require driver intervention in order to stop the master transactions of this function. This reset is per function and resets only the function without affecting activity of other functions or LAN ports. In an IOV enabled system, this reset resets all the VFs attached to the PF.

The EEPROM is partially reloaded after an FLR reset. The words read from EEPROM at FLR are the same as read following a full software reset. A list of these words can be found in Section 3.3.1.3.

A FLR reset to a function resets all the queues, interrupts, and statistics registers attached to the function. It also resets PCIe R/W configuration bits as well as disables transmit and receive flows for the queues allocated to the function. All pending read requests are dropped and PCIe read completions to the function might be completed as unexpected completions and silently discarded (following update of flow control credits) without logging or signaling as an error.

Note: If software initiates a FLR when the *Transactions Pending* bit in the *Device Status* configuration register is set to 1b (refer to Section 9.5.6.6), then software must not initialize the function until allowing time for any associated completions to arrive. The *Transactions Pending* bit is cleared upon completion of the FLR.

4.2.1.5.2 VF (Virtual Function) FLR (Function Level Reset)

An FLR reset to a VF function resets all the queues, interrupts, and statistics registers attached to this VF. It also resets the PCIe R/W configuration bits allocated to this function. It also disables Tx and Rx flow for the queues allocated to this VF. All pending read requests are dropped and PCIe read completions to this function might be completed as unsupported requests.

4.2.1.5.3 IOV (IO Virtualization) Disable

Clearing of the IOV enable bit in the IOV structure is equivalent to a VFLR to all the active VFs in the PF.

4.3 Software Reset

4.3.1 Full Port Software Reset (RST)

Software can reset a port in the I350 by setting the Port Software Reset (*CTRL.RST)* in the Device Control Register. The port software reset (*CTRL.RST*) is per function and resets only the function that received the software reset. Following the reset the PCI configuration space (configuration and mapping) of the device is unaffected. Prior to issuing software reset the driver needs to operate the master disable algorithm as defined in Section 5.2.3.3.

The *CTRL.RST* bit is provided primarily to recover from an indeterminate or suspected Port hung hardware state. Most registers (receive, transmit, interrupt, statistics, etc.) and state machines in the port are set to their power-on reset values, approximating the state following a power-on or PCIe reset (refer to Table 4-3 for further information on affects of Software reset). However, PCIe configuration registers and logic common to all ports is not reset, leaving the device mapped into system memory space and accessible by a driver.

Note: To ensure that software reset has fully completed and that the I350 responds correctly to subsequent accesses after setting the *CTRL.RST* bit, the driver should wait at least 3 milliseconds before accessing any register and then verify that *EEC.Auto_RD* is set to 1b and that the *STATUS.PF_RST_DONE* bit is set to 1b.

When asserting the *CTRL.RST* software reset bit, only some EEPROM bits related to the specific function are re-read (refer to Section 3.3.1.3). Bits re-read from EEPROM are reset to default values.

4.3.2 Physical Function (PF) Software Reset

A software reset by the PF in IOV mode has the same consequences as a port software reset in a non-IOV mode (refer to Table 4-3 for further information on affects of PF Software reset). The procedure for PF software reset is as follows:

- The PF driver disables master accesses by the device through the Master Disable mechanism (refer to Section 5.2.3.3). Master Disable affects all VFs traffic.
- Execute the procedure described in Section 4.6.11.2.3 to synchronize between the PF and VFs.

VFs are expected to timeout and check on the *VFMailbox.RSTD* bit in order to identify a PF software reset event. The *VFMailbox.RSTD* bits are cleared on read.

4.3.3 VF Software Reset

A software reset applied to a VF is equivalent to a FLR reset to this VF with the exception that the PCIe configuration bits allocated to this function are not reset. This can be activated by setting the *VFCTRL.RST* bit (refer to Table 4-4 for further information on affects of Software reset).

4.3.4 Device Software Reset (DEV_RST)

Software can reset all I350 ports by setting the Device Reset bit (*CTRL.DEV_RST*) in the Device Control Register. The Device Reset (*CTRL.DEV_RST)* resets all functions and common logic (refer to Table 4-2 for further information on affects of Device Software reset). PCI configuration space (configuration and mapping) of the device is unaffected.

Device Reset (*CTRL.DEV_RST*) can be used to globally reset the entire component, if the *DEV_RST_EN* bit in *Initialization Control 4* EEPROM word is set. This bit is provided as a last-ditch software mechanism to recover from an indeterminate or suspected hardware hung state that could not be resolved by setting the *CTRL.RST* bit. When setting *CTRL.DEV_RST*, most registers (receive, transmit, interrupt, statistics, etc.) and state machines on ports are set to their default values similar to the state following a Function Level reset on all functions. PCIe configuration registers are not reset, leaving the device mapped into system memory space and accessible by the drivers.

When *CTRL.DEV_RST* is asserted by software on a LAN port, all LAN ports (including LAN ports that didn't initiate the reset) are placed in a reset state. To notify software device drivers on all ports that *CTRL.DEV_RST* has been asserted, an interrupt is generated and the *ICR.DRSTA* bit is set on all ports that didn't initiate the Device reset. In addition the *STATUS.DEV_RST_SET* bit is set on all ports to indicate that Device reset was asserted.

Prior to issuing Device reset the driver needs to:

1. Get ownership of the Device reset functionality by sending message via the mailbox mechanism described in section Section 4.7.3 and receiving acknowledge message from other drivers.
2. Initiate the master disable algorithm as defined in Section 5.2.3.3.

Note: To ensure that Device reset has fully completed and that the I350 responds correctly to subsequent accesses, wait at least 3 milliseconds after setting *CTRL.DEV_RST* before attempting to access any other register.

Following Device Reset assertion or reception of Device Reset interrupt (*ICR.DRSTA*) software should initiate the following steps to re-initialize the port:

1. Wait for the *GCR.DEV_RST In progress* bit to be cleared.
2. Read *STATUS.DEV_RST_SET* bit and clear bit by write 1.
3. Verify that *EEC.Auto_RD* is set to 1b and EEPROM Auto load completed, since setting the Device reset bit (*CTRL.DEV_RST*) causes EEPROM bits related to all ports to be re-read (refer to Section 3.3.1.3).
4. Check that the *STATUS.PF_RST_DONE* bit is set to 1b to verify that internal reset has completed.
5. Re-initialize the port.
6. Check *STATUS.DEV_RST_SET* bit and verify that bit is still 0. If bit is set, return to 1. and re-start initialization process.
7. Driver that initiated the Device reset should release ownership of Device Reset and Mailbox using the flow described in Section 4.7.3.

4.3.4.1 BME (Bus Master Enable)

Disabling Bus Master activity of a function by clearing the Configuration *Command register.BME* bit to 0, resets all DMA activities and MSI/MSIx operations related to the port. The Master disable is per function and resets only the DMA activities related to this function without affecting activity of other functions or LAN ports. Configuration accesses and target accesses to the function are still enabled and BMC can still transmit and receive packets on the port.

A Master Disable to a function resets all the queues and DMA related interrupts attached to this function. It also disables the transmit and receive flows for the queues allocated to this function. All pending read requests are dropped and PCIe read completions to this function might be completed as unexpected completions and silently discarded (following update of flow control credits) without logging or signaling it as an error.

Note: Prior to issuing a master disable the Driver needs to implement the master disable algorithm as defined in Section 5.2.3.3. After Master Enable is set back to 1 driver should re-initialize the transmit and receive queues.

4.3.4.2 Force TCO

This reset is generated when manageability logic is enabled and BMC detects that the I350 does not receive or transmit data correctly. Force TCO reset is enabled if the *Reset on Force TCO* bit in the *Management Control* EEPROM word is set 1. Table 4-3 describes affects of TCO reset on the I350 functionality.

Force TCO reset is generated in pass through mode when BMC issues a Force TCO command with bit 1 set and the above conditions exist.

4.3.4.3 EEPROM Reset

Writing a 1 to the EEPROM Reset bit of the Extended Device Control Register (*CTRL_EXT.EE_RST*) causes the I350 to re-read the per-function configuration from the EEPROM, setting the appropriate bits in the registers loaded by the EEPROM.

4.3.4.4 PHY Reset

Software can write a 1 to the PHY Reset bit of the Device Control Register (*CTRL.PHY_RST*) to reset the internal PHY. The PHY is internally configured after a PHY reset.

Note: The internal PHY should not be reset using PHYREG 0 bit 15 (*PCTRL.Reset*), since in this case the internal PHY configuration process is bypassed and there is no guarantee the PHY will operate correctly.

As the PHY may be accessed by the internal firmware and the driver software, the driver software should coordinate any PHY reset with the firmware using the following procedure:

1. Check that *MANC.BLK_Phy_Rst_On_IDE* (offset 0x5820 bit 18) is cleared. If it is set, the BMC requires a stable link and thus the PHY should not be reset at this stage. The driver may skip the PHY reset if not mandatory or wait for *MANC.BLK_Phy_Rst_On_IDE* to clear. Refer to Section 4.3.6 for more details.
2. Take ownership of the relevant PHY (on port 0,1,2 or 3) using the following flow:
   a. Get ownership of the software/software semaphore *SWSM.SMBI* bit (offset 0x5B50 bit 0).
      - Read the *SWSM* register.
      - If *SWSM.SMBI* is read as zero, the semaphore was taken.
      - Otherwise, go back to step a.
      - This step assure that other software will not access the shared resources register (*SW_FW_SYNC*).
   b. Get ownership of the software/firmware semaphore *SWSM.SWESMBI* bit (offset 0x5B50 bit 1):
      - Set the *SWSM.SWESMBI* bit.

- Read *SWSM*.
- If *SWSM.SWESMBI* was successfully set - the semaphore was acquired - otherwise, go back to step a.
- This step assure that the internal firmware will not access the shared resources register (*SW_FW_SYNC*).

c. Software reads the Software-Firmware Synchronization Register (*SW_FW_SYNC)* and checks both bits in the pair of bits that control the PHY it wishes to own.

- If both bits are cleared (both firmware and other software does not own the PHY), software sets the software bit in the pair of bits that control the resource it wishes to own.
- If one of the bits is set (firmware or other software owns the PHY), software tries again later.

d. Release ownership of the software/firmware semaphore by clearing the *SWSM.SWESMBI* bit.

3. Drive PHY reset bit in CTRL bit 31.
4. Wait 100 µs.
5. Release PHY reset in CTRL bit 31.
6. Release ownership of the relevant PHY to the FW using the following flow:

a. Get ownership of the software/firmware semaphore *SWSM.SWESMBI* (offset 0x5B50 bit 1):

- Set the *SWSM.SWESMBI* bit.
- Read SWSM.
- If *SWSM.SWESMBI* was successfully set - the semaphore was acquired - otherwise, go back to step a.
- Clear the bit in *SW_FW_SYNC* that control the software ownership of the resource to indicate this resource is free.
- Release ownership of the software/firmware semaphore by clearing the *SWSM.SWESMBI* bit.

7. Wait for the relevant *CFG_DONE* bit (*EEMNGCTL.CFG_DONE0*, *EEMNGCTL.CFG_DONE1*, *EEMNGCTL.CFG_DONE2* or *EEMNGCTL.CFG_DONE3*).
8. Take ownership of the relevant PHY using the following flow:

a. Get ownership of the software/firmware semaphore *SWSM.SWESMBI* (offset 0x5B50 bit 1):

- Set the *SWSM.SWESMBI* bit.
- Read SWSM.
- If *SWSM.SWESMBI* was successfully set - the semaphore was acquired - otherwise, go back to step a.
- This step assure that the internal firmware will not access the shared resources register (*SW_FW_SYNC*).

b. Software reads the Software-Firmware Synchronization Register (*SW_FW_SYNC*) and checks both bits in the pair of bits that control the PHY it wishes to own.

- If both bits are cleared (both firmware and other software does not own the PHY), software sets the software bit in the pair of bits that control the resource it wishes to own.
- If one of the bits is set (firmware or other software owns the PHY), software tries again later.

c. Release ownership of the software/software semaphore and the software/firmware semaphore by clearing *SWSM.SMBI* and *SWSM.SWESMBI* bits.

9. Configure the PHY.
10. Release ownership of the relevant PHY using the flow described in Section 4.7.2.

Note: Software PHY ownership should not exceed 100 mS. If Software takes PHY ownership for a longer duration, Firmware may implement a timeout mechanism and take ownership of the PHY.

4.3.5 Registers and Logic Reset Affects

The resets affect the following registers and logic:

Table 4-2 I350 Reset Affects - Common Resets

Reset Activation	LAN_PWR_GOOD	PE_RST_N	SW DEV_RST	In-Band PCIe Reset	FW Reset	Notes
LTSSM (PCIe back to detect/ polling)	X	X		X		
PCIe Link data path	X	X		X		
Read EEPROM (Per Function)			X			19.
Read EEPROM (Complete Load)	X	X		X		
PCI Configuration Registers- non sticky	X	X		X		3.
PCI Configuration Registers - sticky	X	X		X		4.
PCIe local registers	X	X		X		5.
Data path	X	X	X	X		
On-die memories	X	X	X	X		16.
MAC, PCS, Auto Negotiation and other port related logic	X	X	X	X		
Virtual function queue enable	X	X	X	X		
Virtual function interrupt and statistics registers	X	X	X	X		14.
DMA	X	X	X	X		
Functions queue enable	X	X	X	X		
Function interrupt & statistics registers	X	X	X	X		
Wake Up (PM) Context	X	1.				7.
Wake Up Control Register	X					8.
Wake Up Status Registers	X					9.
Manageability Control Registers	X					10.
MMS Unit	X				X	
Wake-Up Management Registers	X	X	X	X		3.,11.
Memory Configuration Registers	X	X	X	X		3.
EEPROM and flash request	X		X			17.
PHY/SERDES PHY	X	X		X		2.
Strapping Pins	X	X		X		

Table 4-3 I350 Reset Affects - Per Function Resets

Reset Activation	D3hot –> D0	FLR	Port SW Reset (CTRL.RST)	Force TCO	EE Reset	PHY Reset	Notes
Read EEPROM (Per Function)	X	X	X	X	X		
PCI Configuration Registers RO							3.
PCI Configuration Registers MSI-X	X	X					6.
PCI Configuration Registers RW							
PCIe local registers							5.
Data path	X	X	X	X			
On-die memories	X	X	X	X			16.
MAC, PCS, Auto Negotiation and other port related logic	X	X	X	X			
DMA	X	X					18.
Wake Up (PM) Context							7.
Wake Up Control Register							8.
Wake Up Status Registers							9.
Manageability Control Registers							10.
Virtual function queue enable	X	X	X	X			
Virtual function interrupt & statistics registers	X	X	X				14.
Function queue enable	X	X	X	X			
Function interrupt & statistics registers	X	X	X				
Wake-Up Management Registers	X	X	X	X			3.,11.
Memory Configuration Registers	X	X	X	X			3.
EEPROM and flash request	X	X					17.
PHY/SERDES PHY	X	X		X		X	2.
Strapping Pins							

Table 4-4 I350 Reset Affects -Virtual Function Resets

Reset Activation	VFLR[20.]	Software Reset	Notes
Interrupt registers	X	X	14.
Queue disable	X	X	
VF specific PCIe configuration space	X		13.
Data path			

Notes:

1. If AUX_POWER = 0b the Wakeup Context is reset (*PME_Status* and *PME_En* bits should be 0b at reset if the I350 does not support PME from D3cold).
2. The MMS unit must configure the PHY after any PHY reset.
3. The following register fields do not follow the general rules above:
   a. "*CTRL.SDP0_IODIR*, *CTRL.SDP1_IODIR*, *CTRL_EXT.SDP2_IODIR*, *CTRL_EXT.SDP3_IODIR*, *CONNSW.ENRGSRC field*, *CTRL_EXT.SFP_Enable*, *CTRL_EXT.LINK_MODE*, *CTRL_EXT.EXT_VLAN* and LED configuration registers are reset on LAN_PWR_GOOD only. Any EEPROM read resets these fields to the values in the EEPROM.
   b. The Aux Power Detected bit in the PCIe Device Status register is reset on LAN_PWR_GOOD and PE_RST_N (PCIe reset) assertion only.
   c. FLA - reset on LAN_PWR_GOODInternal Power only.
   d. The bits mentioned in the next note.
4. The following registers are part of this group:
   a. VPD registers
   b. Max payload size field in PCIe Capability Control register (offset 0xA8).
   c. Active State Link PM Control field, Common Clock Configuration field and Extended Synch field in PCIe Capability Link Control register (Offset 0xB0).
   d. *ARI enable* bit in *IOV capability Command* register (offset 0x168).
   e. Read Completion Boundary in the PCIe Link Control register (Offset 0xB0).
5. The following registers are part of this group:
   a. SWSM
   b. *GCR* (only part of the bits - see register description for details)
   c. FUNCTAG
   d. GSCL_1/2/3/4
   e. GSCN_0/1/2/3
   f. *SW_FW_SYNC* - only part of the bits - see register description for details.
6. The following registers are part of this group:
   a. *MSIX control* register, *MSIX PBA* and *MSIX per vector mask*.
7. The Wake Up Context is defined in the PCI Bus Power Management Interface Specification (Sticky bits). It includes:
   — *PME_En* bit of the Power Management Control/Status Register (*PMCSR*).
   — *PME_Status* bit of the Power Management Control/Status Register (*PMCSR*).
   — *Aux_En* in the PCIe registers

— The device Requester ID (since it is required for the PM_PME TLP).

The shadow copies of these bits in the Wakeup Control Register are treated identically.

8. Refers to bits in the Wake Up Control Register that are not part of the Wake-Up Context (the *PME_En* and *PME_Status* bits).
9. The Wake Up Status Registers include the following:
   a. Wake Up Status Register
   b. Wake Up Packet Length.
   c. Wake Up Packet Memory.
10. The manageability control registers refer to the following registers:
   a. *MANC* 0x5820
   b. *MFUTP0-3* 0x5030 - 0x503C
   c. *MNGONLY* 0x5864
   d. *MAVTV0-7* 0x5010 - 0x502C
   e. *MDEF0-7* 0x5890 - 0x58AC
   f. *MDEF_EXT* 0x5930 - 0x594C
   g. *METF0-3* 0x5060 - 0x506C
   h. *MIPAF0-15* 0x58B0 - 0x58EC
   i. *MMAH/MMAL0*-1 0x5910 - 0x591C
   j. FWSM
11. The Wake-up Management Registers include the following:
   a. Wake Up Filter Control
   b. IP Address Valid
   c. IPv4 Address Table
   d. IPv6 Address Table
   e. Flexible Filter Length Table
   f. Flexible Filter Mask Table
12. The Other Configuration Registers include:
   a. General Registers
   b. Interrupt Registers
   c. Receive Registers
   d. Transmit Registers
   e. Statistics Registers
   f. Diagnostic Registers

Of these registers, *MTA[n]*, *VFTA[n]*, *WUPM[n]*, *FTFT[n]*, *FHFT[n]*, *FHFT_EXT[n]*, *TDBAH/TDBAL*, and *RDBAH/RDVAL* registers have no default value. If the functions associated with the registers are enabled they must be programmed by software. Once programmed, their value is preserved through all resets as long as power is applied to the I350.

Note: In situations where the device is reset using the software reset *CTRL.RST* or *CTRL.DEV_RST* the transmit data lines are forced to all zeros. This causes a substantial number of symbol errors to be detected by the link partner. In TBI mode, if the duration is long enough, the link partner might restart the Auto-Negotiation process by sending “break-link” (/C/ codes with the configuration register value set to all zeros).

13. These registers include:
    a. MSI/MSI-X enable bits
    b. BME
    c. Error indications
14. These registers include:
    a. VTEICS
    b. VTEIMS
    c. VTEIAC
    d. VTEIAM
    e. VTEITR 0-2
    f. VTIVAR0
    g. VTIVAR_MISC
    h. PBACL
    i. VFMailbox
15. These registers include:
    a. RXDCTL.Enable
    b. Adequate bit in *VFTE* and *VFRE* registers.
16. The contents of the following memories are cleared to support the requirements of PCIe FLR:
    a. The Tx packet buffers
    b. The Rx packet buffers
17. Includes *EEC.REQ*, *EEC.GNT*, *FLA.REQ* and *FLA.GNT* fields.
18. The following DMA Registers are cleared only by LAN_PWR_GOOD, PCIe Reset or *CTRL.DEV_RST*: *DMCTLX*, *DTPARS*, *DRPARS* and *DDPARS*.
19. *CTRL.DEV_RST* assertion causes read of function related sections for all ports
20. A VFLR does not reset the configuration of the VF, only disables the interrupts and the queues.

4.3.6 PHY Behavior During a Manageability Session

During some manageability sessions (e.g. an IDER or SoL session as initiated by an external BMC), the platform is reset so that it boots from a remote media. This reset must not cause the Ethernet link to drop since the manageability session is lost. Also, the Ethernet link should be kept on continuously during the session for the same reasons. The I350 therefore limits the cases in which the internal PHY would restart the link, by masking two types of events from the internal PHY:

- PE_RST# and PCIe resets (in-band and link drop) do not reset the PHY during such a manageability session
- The PHY does not change link speed as a result of a change in power management state, to avoid link loss. For example, the transition to D3hot state is not propagated to the PHY.
  — Note however that if main power is removed, the PHY is allowed to react to the change in power state (i.e., the PHY might respond in link speed change). The motivation for this exception is to reduce power when operating on auxiliary power by reducing link speed.

The capability described in this section is disabled by default on LAN Power Good reset. The *Keep_PHY_Link_Up_En* bit in the EEPROM must be set to '1' to enable it. Once enabled, the feature is enabled until the next LAN Power Good (i.e., I350 does not revert to the hardware default value on PE_RST#, PCIe reset or any other reset but LAN Power Good).

When the Keep_PHY_Link_Up bit (also known as “veto bit”) in the MANC Register is set, the following behaviors are disabled:

- The PHY is not reset on PE_RST# and PCIe resets (in-band and link drop). Other reset events are not affected - LAN Power Good reset, Device Disable, Force TCO, and PHY reset by software.
- The PHY does not change its power state. As a result link speed does not change.
- The I350 does not initiate configuration of the PHY to avoid losing link.

The keep_PHY_link_up bit is set by the BMC through the Management Control command (refer to Section 10.5.10.1.5 for SMBus command and Section 10.6.3.10 for NC-SI command) on the sideband interface. It is cleared by the external BMC (again, through a command on the sideband interface) when the manageability session ends. Once the keep_PHY_link_up bit is cleared, the PHY updates its Dx state and acts accordingly (e.g. negotiates its speed).

The Keep_PHY_Link_Up bit is also cleared on de-assertion of the MAIN_PWR_OK input pin. MAIN_PWR_OK must be de-asserted at least 1 msec before power drops below its 90% value. This allows enough time to respond before auxiliary power takes over.

The Keep_PHY_Link_Up bit is a R/W bit and can be accessed by host software, but software is not expected to clear the bit. The bit is cleared in the following cases:

- On LAN Power Good
- When the BMC resets or initializes it
- On de-assertion of the MAIN_PWR_OK input pin. The BMC should set the bit again if it wishes to maintain speed on exit from Dr state.

4.4 Function Disable

4.4.1 General

For a LOM (Lan on Motherboard) design, it might be desirable for the system to provide BIOS-setup capability for selectively enabling or disabling LAN functions. It allows the end-user more control over system resource-management and avoid conflicts with add-in NIC solutions. The I350 provides support for selectively enabling or disabling one or more LAN device(s) in the system.

4.4.2 Overview

Device presence (or non-presence) must be established early during BIOS execution, in order to ensure that BIOS resource-allocation (of interrupts, of memory or IO regions) is done according to devices that are present only. This is frequently accomplished using a BIOS CVDR (Configuration Values Driven on Reset) mechanism. The I350 LAN-disable mechanism is implemented in order to be compatible with such a solution.

4.4.3 Disabling Both LAN Port and PCIe Function

The I350 provides two mechanisms to disable its LAN ports and the PCIe functions they reside on:

1. Four pins (LANx_DIS_N, one per LAN port) are sampled on PCIe reset to determine the LAN-enable configuration. If the relevant LANx_DIS_N is asserted and the *PHY_in_LAN_Disable* EEPROM bit (refer to Section 6.2.21) associated with it is set to 1b, both the PCIe function and the LAN port associated with it are disabled.
   — The LANx_DIS_N pins are sampled on power-up, de-assertion of PE_RST_N or In-band PCIe reset.
   — Active polarity (active high or active low) of the LANx_DIS_N pins can be set via the *LAN_DIS_POL* EEPROM bit (refer to Section 6.2.28).
2. All LAN ports except the first (LAN Port 0) might be disabled via the *LAN_DIS* EEPROM bit associated with the LAN port (refer to Section 6.2.21). LAN port 0 can not be disabled via the EEPROM to avoid cases where all LAN ports are disabled in the EEPROM resulting in a EEPROM that can't be accessed.

4.4.4 Disabling PCIe Function Only

The I350 supports also disabling just the PCIe function but keeping the LAN port that resides on it fully active (for manageability purposes and BMC pass-through traffic). This functionality can be achieved in three ways:

1. By setting the *LAN_PCI_DIS* EEPROM bit (refer to Section 6.2.21) associated with the PCIe function.
2. By asserting the relevant LANx_DIS_N pin and clearing the *PHY_in_LAN_Disable* EEPROM bit (refer to Section 6.2.21).
   — Active polarity (active high or active low) of the LANx_DIS_N pins can be set via the *LAN_DIS_POL* EEPROM bit (refer to Section 6.2.28).
   — The LANx_DIS_N pins are sampled on de-assertion of PE_RST_N or In-band PCIe reset.
   — The LANx_DIS_N pins have an internal weak pull-up resistor. If the *LAN_DIS_POL* EEPROM bit (refer to Section 6.2.28) is cleared to 0b and a pin is not connected or driven high during init time, LAN 0 is enabled.
3. By asserting the relevant SDPx_1 pin (SDP0_1, SDP1_1, SDP2_1 or SDP3_1 pins to disable PCIe Functions 0, 1, 2, and 3 respectively) when the *en_pin_pcie_func_dis* EEPROM bit (refer to Section 6.2.28) is set to 1b.
   — The SDPx_1 pins are sampled on de-assertion of PE_RST_N or In-band PCIe reset.
   — Active polarity (active high or active low) of the SDPx_1 pins can be set via the *PCI_DIS_POL* EEPROM bit (refer to Section 6.2.28).

Note: In this case when only the PCIe function is disabled, if the *PHPM.LPLU* register bit is set to 1b, the internal copper PHY associated with the disabled PCIe function will attempt to create a link with its link partner at the lowest common link speed via Auto-negotiation.

4.4.5 PCIe Functions to LAN Ports Mapping

When PCIe function 0 is disabled (Could be either associated with LAN0 or LAN3 as a function of the *FACTPS.LAN Function Sel* bit), two different behaviors are possible. The behavior is controlled by the *Dummy Function Enable* EEPROM bit (refer to Section 6.2.17):

1. Dummy Function mode — In some systems, it is required to keep all the functions at their respective location, even when other functions are disabled. In Dummy Function mode, if PCIe function 0 (either associated with LAN0 or LAN3) is disabled, then it does not disappear from the PCIe configuration space. Rather, the function presents itself as a dummy function. The device ID and class code of this function changes to other values (dummy function Device ID 0x10A6, Class Code 0xFF0000). In addition, the function does not require any memory or I/O space, and does not require an interrupt line.
2. Legacy mode - when PCIe function 0 is disabled, then the LAN port residing on the next existing PCIe function moves to reside on function 0. All other LAN ports keeps their respective locations.

Note: In some systems, the dummy function is not recognized by the enumeration process as a valid PCIe function. In these systems, all ports will not be enumerated and it is recommended to work in legacy mode.

PCIe function mapping to LAN ports as a function of disabled LAN ports and PCIe functions and the *FACTPS.LAN Function Sel* bit is summarized in the following tables.

Notes: PCIe Functions Mapping of the I350 dual port SKU behaves like the quad port SKU with Port 2 and Port 3 disabled.

In the Following tables a port is considered enabled if both the PCIe function and the LAN port is enabled.

Table 4-5 PCI Functions Mapping (Legacy Mode)

Port 0 enabled	Port 1 enabled	Port 2 enabled	Port 3 enabled	FACTPS.LAN Function Sel	Function 0	Function 1	Function 2	Function 3
Yes	Yes	Yes	Yes	0	Port 0	Port 1	Port 2	Port 3
Yes	X	X	X	0	Port 0	As above if enabled		
No	Yes	X	X	0	Port 1	Disabled	As above if enabled	
No	No	Yes	X	0	Port 2	Disabled	Disabled	Port 3 if enabled
No	No	No	Yes	0	Port 3	Disabled	Disabled	Disabled
Yes	Yes	Yes	Yes	1	Port 3	Port 2	Port 1	Port 0
X	X	X	Yes	1	Port 3	As above if enabled		
X	X	Yes	No	1	Port 2	Disabled	As above if enabled	
X	Yes	No	No	1	Port 1	Disabled	Disabled	Port 0 if enabled
Yes	No	No	No	1	Port 0	Disabled	Disabled	Disabled
No	No	No	No	X	All PCI functions are disabled. Device is in low power mode unless used by manageability			

The following EEPROM bits control Function Disable:

- PCIe functions 1 to 3 can be enabled or disabled according to the *LAN_PCI_DIS* EEPROM bit (refer to Section 6.2.21).
  - PCIe function 0 can not be disabled via EEPROM.
- The *LAN_DIS* EEPROM bit (refer to Section 6.2.21), indicates which function and LAN port are disabled. When this bit is set port is not available also to the manageability channel.
  - PCIe function 0 and the LAN port associated with it can not be disabled via EEPROM.
- The *LAN Function Sel* EEPROM bit (refer to Section 6.2.22), defines the correspondence between LAN Port and PCIe function.

- The *Dummy Function Enable* EEPROM bit (refer to Section 6.2.17) enables the Dummy Function mode for function 0. Default value is disabled.
- The *PHY_in_LAN_disable* EEPROM bit (refer to Section 6.2.21), controls the availability of the disabled function to the manageability channel.
- The *en_pin_pcie_func_dis* EEPROM bit (refer to Section 6.2.28), controls whether asserting SDPx_1 pins (SDP0_1, SDP1_1, SDP2_1 or SDP3_1 pins for PCIe Functions 0, 1, 2, and 3 respectively) during PCIe reset disables the relevant PCIe function.
- Polarity (Active low or Active high) of the SDPx_1 pins can be set via the *PCI_DIS_POL* EEPROM bit (refer to Section 6.2.28).
- Polarity (Active Low or Active High) of the LANx_DIS_N pins can be set by the EEPROM via the *LAN_DIS_POL* EEPROM bit (refer to Section 6.2.28).

When a particular LAN is fully disabled, all internal clocks to that LAN are disabled, the device is held in reset, and the internal PHY for that LAN is powered-down. In both modes, the device does not respond to PCI configuration cycles. Effectively, the LAN device becomes invisible to the system from both a configuration and power-consumption standpoint.

4.4.6 Control Options

The functions have a separate enabling Mechanism. Any function that is not enabled does not function and does not expose its PCI configuration registers.

Table 4-6 Strapping for Control Options

Function	Control Options
LAN 0	Strapping Option + EEPROM bits *PHY_in_LAN_disable* (full/PCI only disable in case of LANx_DIS_N strap), *en_pin_pcie_func_dis* (Enable SDPx_1 strap), *LAN_DIS_POL* (LANx_DIS_N active polarity) and *PCI_DIS_POL* (SDPx_1 pin active polarity)
LAN 1	Strapping Option + EEPROM bits *PHY_in_LAN_disable* (full/PCI only disable in case of LANx_DIS_N strap), *LAN_DIS* (full disable), *LAN_PCI_DIS* (PCIe Function only disable), *en_pin_pcie_func_dis* (Enable SDPx_1 strap), *LAN_DIS_POL* (LANx_DIS_N active polarity) and *PCI_DIS_POL* (SDPx_1 pin active polarity)
LAN 2	Strapping Option + EEPROM bits *PHY_in_LAN_disable* (full/PCI only disable in case of LANx_DIS_N strap), *LAN_DIS* (full disable), *LAN_PCI_DIS* (PCIe Function only disable), *en_pin_pcie_func_dis* (Enable SDPx_1 strap), *LAN_DIS_POL* (LANx_DIS_N active polarity) and *PCI_DIS_POL* (SDPx_1 pin active polarity)
LAN 3	Strapping Option + EEPROM bits *PHY_in_LAN_disable* (full/PCI only disable in case of LANx_DIS_N strap), *LAN_DIS* (full disable), *LAN_PCI_DIS* (PCIe Function only disable), *en_pin_pcie_func_dis* (Enable SDPx_1 strap), *LAN_DIS_POL* (LANx_DIS_N active polarity) and *PCI_DIS_POL* (SDPx_1 pin active polarity)

The I350 strapping option for LAN Disable feature are:

Table 4-7 Strapping for LAN Disable

Symbol	Ball #	Name and Function
LAN0_DIS_N		This pin is a strapping option pin always active. In case this pin is asserted during init time, LAN 0 is disabled. This pin is also used for testing and scan. When used for testing or scan, the LAN disable functionality is not active. Refer to Section 4.4.3 and Section 4.4.4 for additional information.
LAN1_DIS_N		This pin is a strapping option pin always active. In case this pin is asserted during init time, LAN 1 function is disabled. This pin is also used for testing and scan. When used for testing or scan, the LAN disable functionality is not active. Refer to Section 4.4.3 and Section 4.4.4 for additional information.
LAN2_DIS_N		This pin is a strapping option pin always active. In case this pin is asserted during init time, LAN 2 is disabled. This pin is also used for testing and scan. When used for testing or scan, the LAN disable functionality is not active. Refer to Section 4.4.3 and Section 4.4.4 for additional information.
LAN3_DIS_N		This pin is a strapping option pin always active. In case this pin is asserted during init time, LAN 3 is disabled. This pin is also used for testing and scan. When used for testing or scan, the LAN disable functionality is not active. Refer to Section 4.4.3 and Section 4.4.4 for additional information.
SDP0_1		This pin is a strapping option if the *en_pin_pcie_func_dis* EEPROM bit is set to 1b. In case this pin is asserted during init time, PCIe function 0 is disabled. This pin is also used for testing and scan. When used for testing or scan or when the *en_pin_pcie_func_dis* EEPROM bit is 0b, the PCIe function disable functionality is not active. Refer to Section 4.4.4 for additional information.
SDP1_1		This pin is a strapping option if the *en_pin_pcie_func_dis* EEPROM bit is set to 1b. In case this pin is asserted during init time, PCIe function 1 is disabled. This pin is also used for testing and scan. When used for testing or scan or when the *en_pin_pcie_func_dis* EEPROM bit is 0b, the PCIe function disable functionality is not active. Refer to Section 4.4.4 for additional information.
SDP2_1		This pin is a strapping option if the *en_pin_pcie_func_dis* EEPROM bit is set to 1b. In case this pin is asserted during init time, PCIe function 2 is disabled. This pin is also used for testing and scan. When used for testing or scan or when the *en_pin_pcie_func_dis* EEPROM bit is 0b, the PCIe function disable functionality is not active. Refer to Section 4.4.4 for additional information.
SDP3_1		This pin is a strapping option if the *en_pin_pcie_func_dis* EEPROM bit is set to 1b. In case this pin is asserted during init time, PCIe function 3 is disabled. This pin is also used for testing and scan. When used for testing or scan or when the *en_pin_pcie_func_dis* EEPROM bit is 0b, the PCIe function disable functionality is not active. Refer to Section 4.4.4 for additional information.

4.4.7 Event Flow for Enable/Disable Functions

This section describes the driving levels and event sequence for device functionality. Following a Power on Reset / LAN_PWR_GOOD / PE_RST_N/ In-Band reset the LANx_DIS_N pins and the SDPx_1 pins (when the *en_pin_pcie_func_dis* EEPROM bit is set) should be de-asserted for nominal operation. If any of the LAN functions are not required statically its associated Disable strapping pin can be tied statically to low.

Case A - BIOS Disables the LAN Function at boot time by using strapping:

1. Assume that following power up the LANx_DIS_N pins and the SDPx_1 pins are de-asserted.
2. The PCIe link is established following the PE_RST_N.
3. BIOS recognizes that a LAN function in the I350 should be disabled.
4. The BIOS asserts the LANx_DIS_N pin or SDPx_1 pin.
5. The BIOS should assert the PCIe reset, either in-band or via PE_RST_N.

6. As a result, the I350 samples the LANx_DIS_N and SDPx_1 pins, disables the LAN function and issues an internal reset to the disabled function.
7. BIOS might start with the Device enumeration procedure (the disabled LAN function is invisible or changed to dummy function).
8. Proceed with Nominal operation.
9. Re-enable of the function could be done by de-asserting the LANx_DIS_N pin or SDPx_1 pin and then request the user to issue a warm boot that causes bus enumeration.

4.4.7.1 Multi-Function Advertisement

If all but one of the LAN devices are disabled, the I350 is no longer a multi-function device. The I350 normally reports a 0x80 in the PCI Configuration Header field Header Type, indicating multi-function capability. However, if only a single LAN is enabled, the I350 reports a 0x0 in this field to signify single-function capability.

4.4.7.2 Legacy Interrupts Utilization

When more than one LAN device is enabled, the I350 can utilize the INTA# to INTC# interrupts for interrupt reporting. The EEPROM *Initialization Control Word 3* (bits 12:11) associated with each LAN device controls which of these interrupts are used for each LAN device. The specific interrupt pin utilized is reported in the PCI Configuration Header Interrupt Pin field associated with each LAN device.

However, if only one LAN device is enabled, then the INTA# must be used for this LAN device, regardless of the EEPROM configuration. Under these circumstances, the Interrupt Pin field of the PCI Header always reports a value of 0x1, indicating INTA# usage.

4.4.7.3 Power Reporting

When more than one LAN function is enabled, the PCI Power Management Register Block has the capability of reporting a "Common Power" value. The Common Power value is reflected in the Data field of the PCI Power Management registers. The value reported as Common Power is specified via the *LAN Power Consumption* EEPROM word (word 0x22), and is reflected in the Data field whenever the Data_Select field has a value of 0x8 (0x8 = Common Power Value Select).

When only one LAN is enabled, the I350 appears as a single-function device, the Common Power value, if selected, reports 0x0 (undefined value), as Common Power is undefined for a single-function device.

4.5 Device Disable

For a LOM design, it might be desirable for the system to provide BIOS-setup capability for selectively enabling or disabling LOM devices. This might allow the end-user more control over system resource-management; avoid conflicts with add-in NIC solutions, etc. The I350 provides support for selectively enabling or disabling it.

Note: If the I350 is configured to provide a 50MHz NC-SI clock (via the NC-SI Output Clock EEPROM bit), then the device should not be disabled.

Device Disable is initiated by assertion of the asynchronous DEV_OFF_N pin. The DEV_OFF_N pin should always be connected to enable correct device operation.

The EEPROM *Power Down Enable* bit (refer to Section 6.2.19) enables device disable mode (Hardware default is that this mode is disabled) and the EEPROM bit *Deep DEV_OFF* (refer to Section 6.2.22) defines amount of power saving when DEV_OFF_N is asserted.

While in device disable mode, the PCIe link is in L3 state. The PHY is in power down mode. Output buffers are tri-stated.

Note: Behavior of SDP pins in device disable mode is controlled by the *SDP_IDDQ_EN* EEPROM bit (refer to Section 6.2.2).

Assertion or deassertion of PCIe PE_RST_N does not have any affect while the device is in device disable mode (i.e., the device stays in the respective mode as long as DEV_OFF_N is asserted). However, the device might momentarily exit the device disable mode from the time PCIe PE_RST_N is de-asserted again and until the EEPROM is read.

During power-up, the DEV_OFF_N pin is ignored until the EEPROM is read. From that point, the device might enter Device Disable if DEV_OFF_N is asserted.

Note: De-assertion of the DEV_OFF_N pin causes a fundamental reset to the I350.

Note to system designer: The DEV_OFF_N pin should maintain its state during system reset and system sleep states. It should also insure the proper default value on system power-up. For example, one could use a GPIO pin that defaults to '1' (enable) and is on system suspend power (i.e., it maintains state in S0-S5 ACPI states).

4.5.1 BIOS Handling of Device Disable

1. Assume that following power up sequence the DEV_OFF_N signal is driven high (else it is already disabled).
2. The PCIe is established following the PE_RST_N.
3. BIOS recognize that the whole Device should be disabled.
4. The BIOS drive the DEV_OFF_N signal to the low level.
5. As a result, the I350 samples the DEV_OFF_N signal and enters the device disable mode.
6. The BIOS places the Link in the Electrical IDLE state (at the other end of the PCIe link) by clearing the LINK Disable bit in the Link Control Register.
7. BIOS might start with the Device enumeration procedure (all of the Device functions are invisible).
8. Proceed with Nominal operation.
9. Re-enable could be done by driving the DEV_OFF_N signal high followed later by bus enumeration.

4.6 Software Initialization and Diagnostics

4.6.1 Introduction

This chapter discusses general software notes for the I350, especially initialization steps. This includes general hardware, power-up state, basic device configuration, initialization of transmit and receive operation, link configuration, software reset capability, statistics, and diagnostic hints.

4.6.2 Power Up State

When the I350 powers up it reads the EEPROM. The EEPROM contains sufficient information to bring the link up and configure the I350 for manageability and/or APM wakeup. However, software initialization is required for normal operation.

The power-up sequence, as well as transitions between power states, are described in Section 4.1.1. The detailed timing is given in Section 5.5. The next section gives more details on configuration requirements.

4.6.3 Initialization Sequence

The following sequence of commands is typically issued to the device by the software device driver in order to initialize the I350 to normal operation. The major initialization steps are:

- Disable Interrupts - see Interrupts during initialization.
- Issue Global Reset and perform General Configuration - see Global Reset and General Configuration.
- Setup the PHY and the link - see Link Setup Mechanisms and Control/Status Bit Summary.
- Initialize all statistical counters - refer to Section 4.6.8.
- Initialize Receive - refer to Section 4.6.9.
- Initialize Transmit - refer to Section 4.6.10.
- Enable Interrupts - refer to Section 4.6.4.

4.6.4 Interrupts During Initialization

- Most drivers disable interrupts during initialization to prevent re-entering the interrupt routine. Interrupts are disabled by writing to the *EIMC* (Extended Interrupt Mask Clear) register. Note that the interrupts need to be disabled also after issuing a global reset, so a typical driver initialization flow is:
- Disable interrupts
- Issue a Global Reset
- Disable interrupts (again)
- …

After the initialization is done, a typical driver enables the desired interrupts by writing to the *EIMS* (Extended Interrupt Mask Set) register.

4.6.5 Global Reset and General Configuration

Device initialization typically starts with a global reset that places the device into a known state and enables the device driver to continue the initialization sequence.

Several values in the Device Control Register (*CTRL*) need to be set, upon power up, or after a device reset for normal operation.

- *FD* bit should be set per interface negotiation (if done in software), or is set by the hardware if the interface is Auto-Negotiating. This is reflected in the *Device Status* Register in the Auto-Negotiation case.
- Speed is determined via Auto-Negotiation by the PHY, Auto-Negotiation by the PCS layer in SGMII/SerDes mode, or forced by software if the link is forced. Status information for speed is also readable in the *STATUS* register.
- *ILOS* bit should normally be set to 0.

4.6.6 Flow Control Setup

If flow control is enabled, program the *FCRTL0*, *FCRTH0*, *FCTTV* and *FCRTV* registers. In order to avoid packet losses, *FCRTH* should be set to a value equal to at least two max size packet below the receive buffer size. E.g. Assuming a packet buffer size of 36 KB and expected max size packet of 9.5K, the *FCRTH0* value should be set to 36 - 2 * 9.5 = 17KB i.e. *FCRTH0.RTH* should be set to 0x440.

If DMA Coalescing is enabled, to avoid packet loss, the *FCRTC.RTH_Coal* field should also be programmed to a value equal to at least a single max packet size below the receive buffer size (i.e. a value equal or less than *FCRTH0.RTH* + max size packet).

4.6.7 Link Setup Mechanisms and Control/Status Bit Summary

The *CTRL_EXT.LINK_MODE* value should be set to the desired mode prior to the setting of the other fields in the link setup procedures.

4.6.7.1 PHY Initialization

Refer to the PHY documentation for the initialization and link setup steps. The device driver uses the *MDIC* register to initialize the PHY and setup the link. Section 3.7.4.4 describes the link setup for the internal copper PHY. Section 3.7.2.2 Section describes the usage of the *MDIC* register.

4.6.7.2 MAC/PHY Link Setup (CTRL_EXT.LINK_MODE = 00b)

This section summarizes the various means of establishing proper MAC/PHY link setups, differences in MAC CTRL register settings for each mechanism, and the relevant MAC status bits. The methods are ordered in terms of preference (the first mechanism being the most preferred).

4.6.7.2.1 MAC Settings Automatically Based on Duplex and Speed Resolved by PHY (CTRL.FRCDPLX = 0b, CTRL.FRCSPD = 0b)

CTRL.FD	Don't care; duplex setting is established from PHY's internal indication to the MAC (FDX) after PHY has auto-negotiated a successful link-up.
CTRL.SLU	Must be set to 1 by software to enable communications between MAC and PHY.
CTRL.RFCE	Must be programmed by software after reading capabilities from PHY registers and resolving the desired flow control setting.
CTRL.TFCE	Must be programmed by software after reading capabilities from PHY registers and resolving the desired flow control setting.

CTRL.SPEED	Don't care; speed setting is established from PHY's internal indication to the MAC (*SPD_IND*) after PHY has auto-negotiated a successful link-up.
STATUS.FD	Reflects the actual duplex setting (FDX) negotiated by the PHY and indicated to MAC.
STATUS.LU	Reflects link indication (LINK) from PHY qualified with *CTRL.SLU* (set to 1).
STATUS.SPEED	Reflects actual speed setting negotiated by the PHY and indicated to the MAC (*SPD_IND*).

4.6.7.2.2 MAC Duplex and Speed Settings Forced by Software Based on Resolution of PHY (CTRL.FRCDPLX = 1b, CTRL.FRCSPD = 1b)

CTRL.FD	Set by software based on reading PHY status register after PHY has auto-negotiated a successful link-up.
CTRL.SLU	Must be set to 1 by software to enable communications between MAC and PHY.
CTRL.RFCE	Must be programmed by software after reading capabilities from PHY registers and resolving the desired flow control setting.
CTRL.TFCE	Must be programmed by software after reading capabilities from PHY registers and resolving the desired flow control setting.
CTRL.SPEED	Set by software based on reading PHY status register after PHY has auto-negotiated a successful link-up.
STATUS.FD	Reflects the MAC forced duplex setting written to *CTRL.FD*.
STATUS.LU	Reflects link indication (LINK) from PHY qualified with *CTRL.SLU* (set to 1).
STATUS.SPEED	Reflects MAC forced speed setting written in *CTRL.SPEED*.

4.6.7.2.3 MAC/PHY Duplex and Speed Settings Both Forced by Software (Fully-Forced Link Setup) (CTRL.FRCDPLX = 1b, CTRL.FRCSPD = 1b, CTRL.SLU = 1b)

CTRL.FD	Set by software to desired full/half duplex operation (must match duplex setting of PHY).
CTRL.SLU	Must be set to 1 by software to enable communications between MAC and PHY. PHY must also be forced/configured to indicate positive link indication (LINK) to the MAC.
CTRL.RFCE	Must be programmed by software to desired flow-control operation (must match flow-control settings of PHY).
CTRL.TFCE	Must be programmed by software to desired flow-control operation (must match flow-control settings of PHY).
CTRL.SPEED	Set by software to desired link speed (must match speed setting of PHY).
STATUS.FD	Reflects the MAC duplex setting written by software to *CTRL.FD*.
STATUS.LU	Reflects 1 (positive link indication LINK from PHY qualified with *CTRL.SLU*). Note that since both *CTRL.SLU* and the PHY link indication LINK are forced, this bit set does not guarantee that operation of the link has been truly established.
STATUS.SPEED	Reflects MAC forced speed setting written in *CTRL.SPEED*.

4.6.7.3 MAC/SERDES Link Setup (CTRL_EXT.LINK_MODE = 11b)

Link setup procedures using an external SERDES interface mode:

4.6.7.3.1 Hardware Auto-Negotiation Enabled (PCS_LCTL. AN ENABLE = 1b; CTRL.FRCSPD = 0b; CTRL.FRCDPLX = 0)

CTRL.FD	Ignored; duplex is set by priority resolution of *PCS_ANDV* and *PCS_LPAB*.
CTRL.SLU	Must be set to 1 by software to enable communications to the SerDes.
CTRL.RFCE	Set by Hardware according to auto negotiation resolution[1].
CTRL.TFCE	Set by Hardware according to auto negotiation resolution[1].
CTRL.SPEED	Ignored; speed always 1000Mb/s when using SerDes mode communications.
STATUS.FD	Reflects hardware-negotiated priority resolution.
STATUS.LU	Reflects *PCS_LSTS.AN COMPLETE* (Auto-Negotiation complete) and link is up.
STATUS.SPEED	Reflects 1000Mb/s speed, reporting fixed value of (10)b.
PCS_LCTL.FSD	Must be zero.
PCS_LCTL.Force Flow Control	Must be zero[1].
PCS_LCTL.FSV	Must be set to 10b. Only 1000 Mb/s is supported in SerDes mode.
PCS_LCTL.FDV	Ignored; duplex is set by priority resolution of *PCS_ANDV* and *PCS_LPAB*.
PCS_LCTL.AN TIMEOUT EN	Must be 1b to enable Auto-negotiation time-out.
CONNSW.ENRGSRC	Must be 0b on 1000BASE-BX backplane, when source of the signal detect indication is internal. When connected to an optical module and SRDS_[n]_SIG_DET pin is connected to the module, should be 1b.
CTRL.ILOS	If SRDS_[n]_SIG_DET pin connected to optical module, should be set according to optical module polarity.

4.6.7.3.2 Auto-Negotiation Skipped (PCS_LCTL. AN ENABLE = 0b; CTRL.FRCSPD = 1b; CTRL.FRCDPLX = 1)

CTRL.FD	Must be set to 1b. - only full duplex is supported in SerDes mode.
CTRL.SLU	Must be set to 1b by software to enable communications to the SerDes.
CTRL.RFCE	Must be 0b (No Auto-negotiation).
CTRL.TFCE	Must be 0b (No Auto-negotiation).
CTRL.SPEED	Must be set to 10b. Only 1000 Mb/s is supported in SerDes mode.
STATUS.FD	Reflects the value written by software to *CTRL.FD*.
STATUS.LU	Reflects whether the PCS is synchronized, qualified with *CTRL.SLU* (set to 1).
STATUS.SPEED	Reflects 1000Mb/s speed, reporting fixed value of (10b).
PCS_LCTL.FSD	Must be set to 1b by software to enable communications to the SerDes.

1. If *PCS_LCTL.Force Flow Control* is set, the auto negotiation result is not reflected in the *CTRL.RFCE* and *CTRL.TFCE* registers. In This case, the software must set these fields after reading flow control resolution from PCS registers.

PCS_LCTL.Force Flow Control
Must be set to 1b.

PCS_LCTL.FSV Must be set to 10b. Only 1000 Mb/s is supported in SerDes mode.

PCS_LCTL.FDV Must be set to 1b - only full duplex is supported in SerDes mode.

PCS_LCTL.AN TIMEOUT EN
Must be 0b when Auto-negotiation is disabled.

CONNSW.ENRGSRC Must be 0b on 1000BASE-BX backplane, when source of the signal detect indication is internal. When connected to an optical module and SRDS_[n]_SIG_DET pin is connected to the module, should be 1b.

CTRL.ILOS If SRDS_[n]_SIG_DET pin connected to optical module, should be set according to optical module polarity.

4.6.7.4 MAC/SGMII Link Setup (CTRL_EXT.LINK_MODE = 10b)

Link setup procedures using an external SGMII interface mode:

4.6.7.4.1 Hardware Auto-Negotiation Enabled (PCS_LCTL. AN ENABLE = 1b, CTRL.FRCDPLX = 0b, CTRL.FRCSPD = 0b)

CTRL.FD Ignored; duplex is set by priority resolution of PCS_ANDV and PCS_LPAB.

CTRL.SLU Must be set to 1 by software to enable communications to the SerDes.

CTRL.RFCE Must be programmed by software after reading capabilities from external PHY registers and resolving the desired setting.

CTRL.TFCE Must be programmed by software after reading capabilities from external PHY registers and resolving the desired setting.

CTRL.SPEED Ignored; speed setting is established from SGMII's internal indication to the MAC after SGMII PHY has auto-negotiated a successful link-up.

STATUS.FD Reflects hardware-negotiated priority resolution.

STATUS.LU Reflects *PCS_LSTS.Link OK*

STATUS.SPEED Reflects actual speed setting negotiated by the SGMII and indicated to the MAC.

PCS_LCTL.Force Flow Control
Ignored.

PCS_LCTL.FSD Should be set to zero.

PCS_LCTL.FSV Ignored; speed is set by priority resolution of *PCS_ANDV* and *PCS_LPAB*.

PCS_LCTL.FDV Ignored; duplex is set by priority resolution of *PCS_ANDV* and *PCS_LPAB*.

PCS_LCTL.AN TIMEOUT EN
Must be 0b. Auto-negotiation time-out should be disabled in SGMII mode.

CONNSW.ENRGSRC Must be 0b. In SGMII mode source of the signal detect indication is internal.

4.6.7.5 MAC/1000BASE-KX Link Setup (CTRL_EXT.LINK_MODE = 01b)

4.6.7.5.1 Auto-Negotiation Skipped (PCS_LCTL. AN ENABLE = 0b; CTRL.FRCSPD = 1b; CTRL.FRCDPLX = 1)

Link setup procedures using an external 1000BASE-KX Server Backplane interface mode:

CTRL.FD	Must be set to 1b. 1000BASE-KX always in full duplex mode.
CTRL.SLU	Must be set to 1b by software to enable communications to the SerDes.
CTRL.RFCE	Must be 0b (no Auto-negotiation).
CTRL.TFCE	Must be 0b (no Auto-negotiation).
CTRL.SPEED	Must be set to 10b. Only 1000 Mb/s is supported in 1000BASE-KX mode.
STATUS.FD	Reflects the value written by software to *CTRL.FD*.
STATUS.LU	Reflects whether the PCS is synchronized, qualified with *CTRL.SLU* (set to 1b).
STATUS.SPEED	Reflects 1000Mb/s speed, reporting fixed value of (10b).
PCS_LCTL.FSD	Must be set to 1b by software to enable communications to the 1000BASE-KX SerDes.
PCS_LCTL.Force Flow Control	Must be set to 1b.
PCS_LCTL.FSV	Must be set to 10b. Only 1000 Mb/s is supported in 1000BASE-KX mode.
PCS_LCTL.FDV	Must be set to 1b - only full duplex is supported in 1000BASE-KX mode.
PCS_LCTL.AN TIMEOUT EN	Must be 0b. Auto-negotiation not supported in 1000BASE-KX mode.
CONNSW.ENRGSRC	Must be 0b. In 1000BASE-KX mode source of the signal detect indication is internal.

4.6.8 Initialization of Statistics

Statistics registers are hardware-initialized to values as detailed in each particular register's description. The initialization of these registers begins upon transition to D0active power state (when internal registers become accessible, as enabled by setting the Memory Access Enable of the PCIe Command register), and is guaranteed to be completed within 1 μsec of this transition. Access to statistics registers prior to this interval might return indeterminate values.

All of the statistical counters are cleared on read and a typical device driver reads them (thus making them zero) as a part of the initialization sequence.

4.6.9 Receive Initialization

Program the Receive address register(s) per the station address. This can come from the EEPROM or by any other means (for example, on some machines, this comes from the system PROM not the EEPROM on the adapter card).

Set up the *MTA* (Multicast Table Array) by software. This means zeroing all entries initially and adding in entries as requested.

Program *RCTL* with appropriate values. If initializing it at this stage, it is best to leave the receive logic disabled (*RCTL.RXEN* = 0b) until after the receive descriptor rings have been initialized. If VLANs are not used, software should clear *VFE*. Then there is no need to initialize the *VFTA*. Select the receive descriptor type.

The following should be done once per receive queue needed:

1. Allocate a region of memory for the receive descriptor list.
2. Receive buffers of appropriate size should be allocated and pointers to these buffers should be stored in the descriptor ring.
3. Program the descriptor base address with the address of the region.
4. Set the length register to the size of the descriptor ring.
5. Program *SRRCTL* of the queue according to the size of the buffers, the required header handling and the drop policy.
6. If header split or header replication is required for this queue, program the *PSRTYPE* register according to the required headers.
7. Enable the queue by setting *RXDCTL.ENABLE*. In the case of queue zero, the enable bit is set by default - so the ring parameters should be set before *RCTL.RXEN* is set.
8. Poll the *RXDCTL* register until the *ENABLE* bit is set. The tail should not be bumped before this bit was read as one.
9. Program the direction of packets to this queue according to the mode selected in the *MRQC* register. Packets directed to a disabled queue are dropped.

Note: The tail register of the queue (*RDT[n]*) should not be bumped until the queue is enabled.

4.6.9.1 Initialize the Receive Control Register

To properly receive packets the receiver should be enabled by setting *RCTL.RXEN*. This should be done only after all other setup is accomplished. If software uses the Receive Descriptor Minimum Threshold Interrupt, that value should be set.

4.6.9.2 Dynamic Enabling and Disabling of Receive Queues

Receive queues can be dynamically enabled or disabled given the following procedure is followed:

Enabling a queue:

- Follow the per queue initialization sequence described in Section 4.6.9.

Note: If there are still packets in the packet buffer assigned to this queue according to previous settings, they are received after the queue is re-enabled. In order to avoid this condition, the software might poll the *PBRWAC* register. Once a an empty condition of the relevant packet buffer is detected or 2 wrap around occurrences are detected the queue can be re-enabled.

Disabling a queue:

1. Disable assignment of packets to this queue.
2. Clear the *VFRE* bit allocated to the queue in the *VFRE* register.
3. Poll the *PBRWAC* register until an empty condition of the relevant packet buffer is detected or 2 wrap around occurrences are detected.
4. Disable the queue by clearing *RXDCTL.ENABLE*. The I350 stops fetching and writing back descriptors from this queue immediately. The I350 eventually completes the storage of one buffer allocated to this queue. Any further packet directed to this queue is dropped. If the currently processed packet is spread over more than one buffer, all subsequent buffers are not written.
5. The I350 clears *RXDCTL.ENABLE* only after all pending memory accesses to the descriptor ring or to the buffers are done. The driver should poll this bit before releasing the memory allocated to this queue.
6. Set the *VFRE* bit allocated to the queue so that the queue can be re-enabled.

Note: The RX path can be disabled only after all Rx queues are disabled.

4.6.10 Transmit Initialization

- Program the *TCTL* register according to the MAC behavior needed.

If work in half duplex mode is expected, program the *TCTL_EXT.COLD* field. For internal PHY mode the default value of 0x42 is OK. For SGMII mode, a value reflecting the I350 and the PHY SGMII delays should be used. A suggested value for a typical PHY is 0x46 for 10 Mbps and 0x4C for 100 Mbps.

The following should be done once per transmit queue:

- Allocate a region of memory for the transmit descriptor list.
- Program the descriptor base address with the address of the region.
- Set the length register to the size of the descriptor ring.
- Program the *TXDCTL* register with the desired TX descriptor write back policy. Suggested values are:
  - *WTHRESH* = 1b
  - All other fields 0b.
- If needed, set the *TDWBAL/TWDBAH* to enable head write back
- Enable the queue using *TXDCTL.ENABLE* (queue zero is enabled by default).
- Poll the *TXDCT*L register until the *ENABLE* bit is set.

Note: The tail register of the queue (*TDT[n]*) should not be bumped until the queue is enabled.

Enable transmit path by setting *TCTL.EN*. This should be done only after all other settings are done.

4.6.10.1 Dynamic Queue Enabling and Disabling

Transmit queues can be dynamically enabled or disabled given the following procedure is followed:

Enabling:

- Follow the per queue initialization described in the previous section.

Disabling:

- Stop storing packets for transmission in this queue.
- Wait until the head of the queue (*TDH)* is equal to the tail (*TDT*), i.e. the queue is empty.
- Disable the queue by clearing *TXDCTL.ENABLE*.

The Tx path might be disabled only after all Tx queues are disabled.

4.6.11 Virtualization Initialization Flow

4.6.11.1 VMDq Mode

4.6.11.1.1 Global Filtering and Offload Capabilities

- Select the VMDQ pooling method - MAC/VLAN filtering for pool selection. *MRQC.Multiple Receive Queues Enable* = 011b.
- Set the *RPLPSRTYPE* registers to define the behavior of replicated packets.
- Configure *VT_CTL.DEF_PL* to define the default pool. If packets with no pools should be dropped, set *VT_CTL.Dis_def_Pool* field.
- If needed, enable padding of small packets via the *RCTL.PSP*

4.6.11.1.2 Mirroring rules.

For each mirroring rule to be activated:

a. Set the type of traffic to be mirrored in the *VMRCTL[n]* register.

b. Set the mirror pool in the *VMRCTL[n].MP*

c. For pool mirroring, set the *VMRVM[n]* register with the pools to be mirrored.

d. For VLAN mirroring, set the *VMVRLAN[n]* with the indexes from the *VLVF* registers of the VLANs to be mirrored.

4.6.11.1.3 Per Pool Settings

As soon as a pool of queues is associated to a VM the software should set the following parameters:

1. Address filtering:
   a. The unicast MAC address of the VM by enabling the pool in the *RAH/RAL* registers.
   b. If all the MAC addresses are used, the unicast hash table (*UTA*) can be used. Pools servicing VMs whose address is in the hash table should be declared as so by setting the *VMOLR.ROPE*. Packets received according to this method didn't pass perfect filtering and are indicated as such.
   c. Enable the pool in all the *RAH/RA*L registers representing the multicast MAC addresses this VM belongs to.
   d. If all the MAC addresses are used, the multicast hash table (*MTA*) can be used. Pools servicing VMs using multicast addresses in the hash table should be declared as so by setting the *VMOLR.ROMPE*. Packets received according to this method didn't pass perfect filtering and are indicated as such.
   e. Define whether this VM should get all multicast/broadcast packets in the same VLAN via the *VMOLR.MPE* and *VMOLR.BAM fields*
   f. Enable the pool in each *VLVF* register representing a VLAN this VM belongs to.
   g. Define whether the pool belongs to the default VLAN and should accept untagged packets via the *VMOLR.AUPE* field
2. Offloads
   a. Define whether VLAN header should be stripped from the packet (defined by D*VMOLR.strvlan*).
   b. Set which header split is required via the *PSRTYPE* register.
   c. Set whether larger than standard packet are allowed by the VM and what is the largest packet allowed (jumbo packets support) via the *VMOLR.*RLPML and *VMOLR.RLE* fields.
3. Queues
   a. Enable Rx and Tx queues as described in Section 4.6.9 and Section 4.6.10
   b. For each Rx queue a drop/no drop flag can be set in *SRRCTL.DROP_EN* or via the QDE register, controlling the behavior in cases no receive buffers are available in the queue to receive packets. The usual behavior is to allow drops in order to avoid head of line blocking, unless a no-drop behavior is needed for some type of traffic (e.g. storage).

4.6.11.1.4 Security Features

4.6.11.1.4.1 Storm control

The driver may set limits to the broadcast or multicast traffic it can receive.

1. It should set the how many 64 byte chunks of Broadcast and Multicast traffic are acceptable per interval via the *BSCTRH* and *MSCTRH* respectively.
2. It should then set the interval to be used via the *SCCRL.Interval* field and which action to take when the broadcast or multicast traffic crosses the programmed threshold via the *SCCRL.BDIPW*, *SCCRL.BDICW*, *SCCRL.MDIPW*, and *SCCRL.MDICW* fields.
3. The driver may be notified of storm control events through the *ICR.SCE* interrupt cause.

4.6.11.2 IOV Initialization

The initialization flow used to enable an IOV function can be found in chapter 2 of the PCI-Express Single Root I/O Virtualization and Sharing Specification.

4.6.11.2.1 PF Driver Initialization

The PF driver is responsible for the link setup and handling of all the filtering & off load capabilities for all the VFs as described in Section 4.6.11.1.1 and the security features as described in Section 4.6.11.1.4. It should also set the bandwidth allocation per transmit queue for each VF as described in Section 4.6.10.

After all the common parameters are set, the PF driver should set all the *VFMailbox.RSTD* bit by setting the *CTRL.PFRSTD*.

The PF might disable all the active VFs traffic via the *VFTE* & *VFRE* registers until the parameters of a VF are set as defined in Section 4.6.11.1.3, the VF can be enabled using the same registers.

4.6.11.2.1.1 VF Specific Reset Coordination

After the PF driver receives an indication of a VF FLR via the *VFLRE* register, it should enable the receive and transmit for the VF only once the device is programmed with the right parameters as defined in Section 4.6.11.1.3. The receive filtering is enabled using the *VFRE* register and the transmit filtering is enabled via the VFTE register.

Note: The filtering and offloads setup might be based on a central IT settings or on requests from the VF drivers.

The PF driver should assert the VF reset via the *VTCTRL* register before configuration of the VF parameters.

4.6.11.2.2 VF Driver Initialization

Upon init, after the PF indicated that the global init was done via the *VFMailbox.RSTD* bit, the VF driver should communicate with the PF, either via the mailbox or other software mechanisms to assure that the right parameters of the VF are programmed as described in Section 4.6.11.1.3.

The mailbox mechanism is described in Section 7.8.2.9.1.

The PF should also setup the security measures as described in Section 4.6.11.1.4. In addition, the PF may also program whether the VF is allowed to control VLAN insertion or whether VLAN insertion is controlled by the PF via the relevant *VMVIR* register.

The PF should initialize all the statistical counters described in Section 8.18.77 to zero.

The PF driver might then send an acknowledge message with the actual setup done according to the VF request and the IT policy.

The VF driver should then setup the interrupts and the queues as described in Section 4.6.9 & Section 4.6.10.

4.6.11.2.3 Full Reset Coordination

A mechanism is provided to synchronize reset procedures between the Physical Function and the VFs. It is provided specifically for PF software reset but can be used in other reset cases as described below.

The procedure is as follows:

When one of the following reset cases occurs:

- LAN_PWR_GOOD
- PCIe Reset (PE_RST_N or in-band PCIe reset)
- D3hot --> D0
- FLR
- Software reset by the PF (*CTRL.RST*)
- Device Reset by PF (*CTRL.DEV_RST*)

The I350 sets the *RSTI* bits in all the *VFMailbox* registers. Once the reset completes, each VF might read its *VFMailbox* register to identify a reset in progress.

Once the PF completed configuring the device, it sets the *CTRL_EXT.PFRSTD* bit. As a result, the I350 clears the *RSTI* bits in all the *VFMailbox* registers and sets the *RSTD* (Reset Done) bits in all the *VFMailbox* registers.

Until a *VFMailbox.RSTD* condition is detected, the VFs should access only the *VFMailbox* register and should not attempt to activate the interrupt mechanism or the transmit and receive process.

Note: Before issuing a software reset (*CTRL.RST* or *CTRL.DEV_RST*) the PF driver should send a mailbox message to the various VFs via the mailbox mechanism described in Section 7.8.2.9.1 to indicate that a reset is going to occur, following acknowledgment of the message by all the VFs the PF driver can issue the software reset.

4.6.11.2.4 VFRE/VFTE

This mechanism insures that a VF cannot transmit or receive before the Tx and Rx path have been initialized by the PF. It is required for VFLR reset and must also be used in case of VF software reset. It is optional for PF software reset as described above. The VFRE register contains a bit per VF. When the bit is cleared assignment of Rx packet for the VF's pool is disabled. When set, assignment of Rx packet for the VF's pool is enabled.

The *VFTE* register contains a bit per VF. When the bit is cleared, fetching of data for the VF's pool is disabled. When set, fetching of data for the VF's pool is enabled. Fetching of descriptors for the VF pool is maintained, up to the limit of the internal descriptor queues - regardless to *VFTE* settings.

The *VFRE* and *VFTE* registers apply in all device modes (not just IOV). The default values for both registers are therefore '1', enabling transmission and reception in non-IOV modes.

4.6.12 Alternate MAC Address Support

In some systems, the MAC address used by a port needs to be replaced with a temporary MAC address in a way that is transparent to the software layer. One possible usage is in blade systems, to allow a standby blade to use the MAC address of another blade that failed, so that the network image of the entire blade system does not change.

In order to allow this mode, a management console might change the MAC address in the EEPROM image. It is important in this case to be able to keep the original MAC address of the device as programmed at the factory.

In order to support this mode, the I350 provides the *Alternate Ethernet MAC Address* EEPROM structure to store the original MAC addresses. This structure is described in Section 6.4.8. When the MAC address is changed the port Factory MAC address should be written to the *Alternate Ethernet MAC Address* structure before writing the new Ethernet MAC address to the ports *Ethernet Address* EEPROM words (refer to Section 6.2.1).

In some systems, it might be advantageous to restore the original MAC address at power on reset, to avoid conflicts where two network controllers would have the same MAC address. The I350 restores the LAN MAC addresses stored in the *Alternate Ethernet MAC Address* EEPROM structure to the regular *Ethernet MAC address* EEPROM words (refer to Section 6.2.1) if the following conditions are met:

1. The *restore MAC address* bit in the *Common Firmware Parameters* EEPROM word is set (Section 6.3.7.2).
2. The value in word 0x37 (Section 6.4.8) is not 0xFFFF.
3. The MAC address set in the regular *Ethernet MAC Address* EEPROM words is different than the address stored in the *Alternate Ethernet MAC* Address EEPROM structure.
4. The address stored in the Alternate Ethernet MAC Address structure is valid (not all zeros or all ones).

If the Factory MAC address was restored by the internal firmware, the *FWSM.Factory MAC address restored* bit is set. This bit is common to all ports.If the value at word 0x37 is valid, but the MAC addresses in the Alternate MAC structure are not valid (0xFFFFFFFF), the regular MAC address is backed up in the Alternate MAC structure.

The I350 supports replacement of the MAC address with via a BIOS CLP interface as described in TBR.

4.7 Access to Shared Resources

Part of the resources in the I350 are shared between several software entities - namely the drivers of the four ports and the internal firmware. In order to avoid contentions, a driver that needs to access one of these resources should use the flow described in Section 4.7.1 in order to acquire ownership of this resource and use the flow described in Section 4.7.2 in order to relinquish ownership of this resource.

The shared resources are:

1. EEPROM.
2. All PHYs or SerDes ports.

3. CSRs accessed by the internal firmware after the initialization process. Currently there are no such CSRs.
4. The flash.
5. Software to Software Mailbox.
6. Thermal Sensor registers.
7. Management Host Interface

Note: Any other software tool that accesses the register set directly should also follow the flow described below.

4.7.1 Acquiring Ownership Over a Shared Resource

The following flow should be used to acquire a shared resource:

1. Get ownership of the software/software semaphore *SWSM.SMBI* (offset 0x5B50 bit 0).
   a. Read the *SWSM* register.
   b. If *SWSM.SMBI* is read as zero, the semaphore was taken.
   c. Otherwise, go back to step a.

   This step assure that other software will not access the shared resources register (*SW_FW_SYNC*).
2. Get ownership of the software/firmware semaphore *SWSM.SWESMBI* (offset 0x5B50 bit 1):
   a. Set the *SWSM.SWESMBI* bit.
   b. Read *SWSM*.
   c. If *SWSM.SWESMBI* was successfully set - the semaphore was acquired - otherwise, go back to step a.

   This step assure that the internal firmware will not access the shared resources register (*SW_FW_SYNC*).
3. Software reads the Software-Firmware Synchronization Register (*SW_FW_SYNC*) and checks both bits in the pair of bits that control the resource it wishes to own.
   a. If both bits are cleared (both firmware and other software does not own the resource), software sets the software bit in the pair of bits that control the resource it wishes to own.
   b. If one of the bits is set (firmware or other software owns the resource), software tries again later.
4. Release ownership of the software/software semaphore and the software/firmware semaphore by clearing *SWSM.SMBI* and *SWSM.SWESMBI* bits.
5. At this stage, the shared resources is owned by the driver and it may access it. The *SWSM* and *SW_FW_SYNC* registers can now be used to take ownership of another shared resources.

Notes: Software ownership of *SWSM.SWESMBI* bit should not exceed 100 mS. If Software takes ownership for a longer duration, Firmware may implement a timeout mechanism and take ownership of the *SWSM.SWESMBI* bit.

Software ownership of bits in *SW_FW_SYNC* register should not exceed 1 Second. If Software takes ownership for a longer duration, Firmware may implement a timeout mechanism and take ownership of the relevant *SW_FW_SYNC* bits.

4.7.2 Releasing Ownership Over a Shared Resource

The following flow should be used to release a shared resource:

1. Get ownership of the software/software semaphore SWSM.SMBI (offset 0x5B50 bit 0).
   a. Read the *SWSM* register.
   b. If *SWSM.SMBI* is read as zero, the semaphore was taken.
   c. Otherwise, go back to step a.

   This step assures that other software will not access the shared resources register (*SW_FW_SYNC*).
2. Get ownership of the software/firmware semaphore *SWSM.SWESMBI* (offset 0x5B50 bit 1):
   a. Set the *SWSM.SWESMBI* bit.
   b. Read *SWSM*.
   c. If *SWSM.SWESMBI* was successfully set - the semaphore was acquired - otherwise, go back to step a.

   This step assure that the internal firmware will not access the shared resources register (*SW_FW_SYNC*).
3. Clear the bit in *SW_FW_SYNC* that controls the software ownership of the resource to indicate this resource is free.
4. Release ownership of the software/software semaphore and the software/firmware semaphore by clearing *SWSM.SMBI* and *SWSM.SWESMBI* bits.
5. At this stage, the shared resource are released by the driver and it may not access it. The *SWSM* and *SW_FW_SYNC* registers can now be used to take ownership of another shared resource.

4.7.3 Software to Software Mailbox

In order to allow different I350 drivers to coordinate activities, a simple mailbox mechanism is defined. This mechanism allows each driver to send a broadcast message to all the other drivers on the same device.

In order to send a message the following flow should be used:

1. The Driver that wants to send the message should acquire the mailbox semaphore (*SW_FW_SYNC.SW_MB_SM*) using the flow described in Section 4.7.1.
2. The Driver should then write the message in the *SWMBWR* register.
3. All the drivers will then receive an interrupt (except for the driver that initiated the message) via the *SWMB* cause in the *ICR* registers.
4. All the drivers will read the *SWMB0, SWMB1, SWMB2 and SWMB3* registers to understand which driver sent a message.

Note: The mapping of *SWMB0, SWMB1, SWMB2 and SWMB3* registers is according to the physical ports. A function can detect which physical port it is mapped to, by reading the *STATUS.LAN ID* field.

5. If the message requires an acknowledgment from the other drivers, each driver may write an acknowledge message through their *SWMBWR* register.
6. The driver that sent the original message can then poll the *SWMB0, SWMB1, SWMB2 and SWMB3 registers* for acknowledge messages.
7. After the message was acknowledged, the driver that sent the original message should release the Software Mailbox semaphore (*SW_FW_SYNC.SW_MB_SM*) using the flow described in Section 4.7.2.
   — Together with clearing the *SWSM.SMBI* and *SWSM.SWESMBI* bits when releasing the Software Mailbox semaphore, Software driver should also clear the *SWMBWR, SWMB0, SWMB1, SWMB2 and SWMB3* registers by setting the *SWSM.SWMB_CLR* bit to avoid confusion when future messages are sent.

5 Power Management

This section describes how power management is implemented in the I350. The I350 supports the Advanced Configuration and Power Interface (ACPI) specification as well as Advanced Power Management (APM).

Power management can be disabled via the *power management* bit in the *Initialization Control Word 1* EERPROM word (see Section 6.2.2), which is loaded during power-up reset. Even when disabled, the power management register set is still present. Power management support is required by the PCIe specification.

5.1 General Power State Information

5.1.1 PCI Device Power States

The PCIe Specification defines function power states (D-states) that enable the platform to establish and control power states for the I350 ranging from fully on to fully off (drawing no power) and various in-between levels of power-saving states, annotated as D0-D3. Similarly, PCIe defines a series of link power states (L-states) that work specifically within the link layer between the I350 and its upstream PCIe port (typically in the host chipset).

The PCIe link state follows the power management state of the device. Since the I350 incorporates multiple PCI functions, the device power management state is defined as the power management state of the most awake function:

- If any function is in D0a state in ARI mode or either D0a or D0u in non-ARI mode, the PCIe link assumes the device is in D0 state. Else,
- If in ARI mode, at least one of the functions is in D3 state and the other functions are not in D0a state, or if in non-ARI mode all of the functions are in the D3 state, the PCIe link assumes the device is in D3 state. Else,
- The device is in Dr state (PE_RST_N is asserted to all functions).

For a given device D-state, only certain L-states are possible as follows.

- D0 (fully on): The I350 is completely active and responsive during this D-state. The link can be in either L0 or a low-latency idle state referred to as L0s. Minimizing L0s exit latency is paramount for enabling frequent entry into L0s while facilitating performance needs via a fast exit. A deeper link power state, L1 state, is supported as well.
- D1 and D2: These modes are not supported by the I350.
- D3 (off): Two sub-states of D3 are supported:
  - D3hot, where primary power is maintained.
  - D3cold, where primary power is removed.

Link states are mapped into device states as follows:

— D3hot maps to L1 to support clock removal on mobile platforms

— D3cold maps to L2 if auxiliary power is supported on the I350 with wake-capable logic, or to L3 if no power is delivered to the I350. A sideband PE_WAKE_N mechanism is supported to interface wake-enabled logic on mobile platforms during the L2 state.

5.1.2 PCIe Link Power States

Configuring the I350 into a D-state automatically causes the PCIe link to transition to the appropriate L-state.

- L2/L3 Ready: This link state prepares the PCIe link for the removal of power and clock. The I350 is in the D3hot state and is preparing to enter D3cold. The power-saving opportunities for this state include, but are not limited to, clock gating of all PCIe architecture logic, shutdown of the PLL, and shutdown of all transceiver circuitry.
- L2: This link state is intended to comprehend D3cold with auxiliary power support. Note that sideband PE_WAKE_N signaling exists to cause wake-capable devices to exit this state. The power-saving opportunities for this state include, but are not limited to, shutdown of all transceiver circuitry except detection circuitry to support exit, clock gating of all PCIe logic, and shutdown of the PLL as well as appropriate platform voltage and clock generators.
- L3 (link off): Power and clock are removed in this link state, and there is no auxiliary power available. To bring the I350 and its link back up, the platform must go through a boot sequence where power, clock, and reset are reapplied appropriately.

5.2 Power States

The I350 supports the D0 and D3 architectural power states as described earlier. Internally, the I350 supports the following power states:

- D0u (D0 un-initialized) - an architectural sub-state of D0
- D0a (D0 active) - an architectural sub-state of D0
- D3 - architecture state D3hot
- Dr - internal state that contains the architecture D3cold state. Dr state is entered when PE_RST_N is asserted or a PCIe in-band reset is received

Figure 5-1 shows the power states and transitions between them.

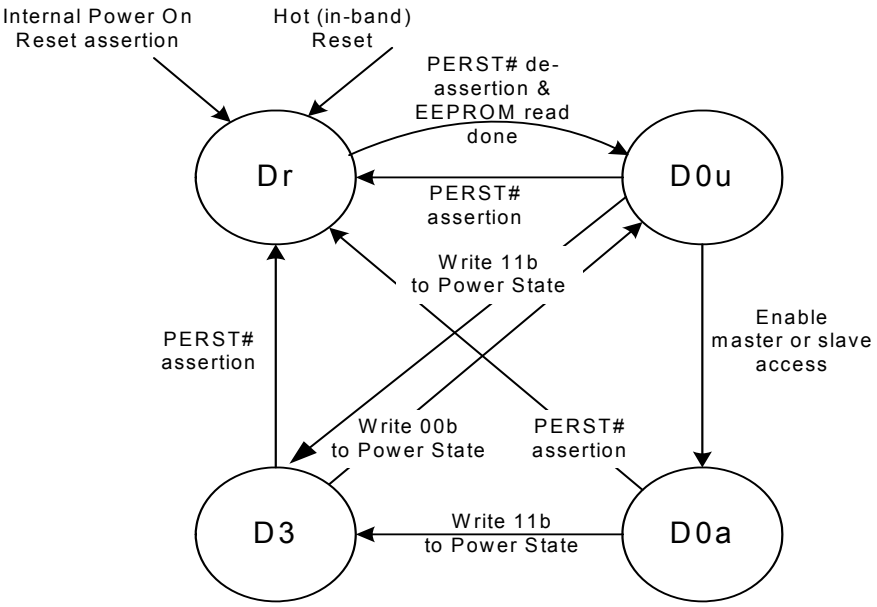


Figure 5-1 Power Management State Diagram

5.2.1 D0 Uninitialized State (D0u)

The D0u state is an architectural low-power state.

When entering D0u, the I350:

- Asserts a reset to the PHY while the EEPROM is being read.
- Disables wake up. However APM wake up is enabled (See additional information in Section 5.6.1), if all of the following register bits are set:
  - The *WUC.APME* bit is set to 1b.
  - The *WUC.APMPME* bit or the *PMCSR.PME_en* bits are set to 1b.
  - The *WUC.EN_APM_D0* bit is set to 1b.

5.2.1.1 Entry into D0u State

D0u is reached from either the Dr state (on de-assertion of *PE_RST_N*) or the D3hot state (by configuration software writing a value of 00b to the *Power State* field of the PCI PM registers).

De-asserting *PE_RST_N* means that the entire state of the I350 is cleared, other than sticky bits. State is loaded from the EEPROM, followed by establishment of the PCIe link. Once this is done, configuration software can access the I350.

On a transition from D3hot state to D0u state, the I350 PCI configuration space is not reset (since the *No_Soft_Reset* bit in the *PMCSR* register is set to 1b). However following move to D0a state, the I350 requires that the software driver perform a full re-initialization of the function.

5.2.2 D0active State

Once memory space is enabled, the I350 enters the D0 active state. It can transmit and receive packets if properly configured by the software device driver. The PHY is enabled or re-enabled by the software device driver to operate/auto-negotiate to full line speed/power if not already operating at full capability.

Notes:

1. In the I350 if the WUC.*EN_APM_D0* is cleared to 0 an APM wake event due to reception of a Magic packet is not generated when the function is not in D3 (or Dr) state. Any APM wake up previously active remains active when moving from D3 to D0.
2. If APM wake is required in D3 software driver should not disable APM wake-up via the *WUC.APME* bit on D0 entry. Otherwise APM wake following a system crash and entry into S3, S4 or S5 system power management state will not be enabled.
3. Following entry into D0 software device driver can activate other wake-up filters by writing to the Wake Up Filter Control (*WUFC*) register.

5.2.2.1 Entry to D0a State

D0a is entered from the D0u state by writing a 1b to the *Memory Access Enable* or the *I/O Access Enable* bit of the PCI Command register (See Section 9.4.3). The DMA, MAC, and PHY of the appropriate LAN function are also enabled.

5.2.3 D3 State (PCI-PM D3hot)

The I350 transitions to D3 when the system writes a 11b to the Power State field of the *Power Management Control/Status Register (PMCSR)*. Any wake-up filter settings that were enabled before entering this state are maintained. If the *PMCSR.No_Soft_reset* bit is cleared upon completion or during the transition to D3 state, the I350 clears the *Memory Access Enable* and *I/O Access Enable* bits of the PCI Command register, which disables memory access decode. If the *PMCSR.No_Soft_reset* bit is set the I350 doesn't clear any bit in the PCIe configuration space. While in D3, the I350 does not generate master cycles.

Configuration and message requests are the only TLPs accepted by a function in the D3hot state. All other received requests must be handled as unsupported requests, and all received completions are handled as unexpected completions. If an error caused by a received TLP (such as an unsupported request) is detected while in D3hot, and reporting is enabled, the link must be returned to L0 if it is not already in L0 and an error message must be sent. See section 5.3.1.4.1 in The PCIe Base Specification.

5.2.3.1 Entry to D3 State

Transition to D3 state is through a configuration write to the *Power State* field of the *PMCSR* PCIe configuration register.

Prior to transition from D0 to the D3 state, the software device driver disables scheduling of further tasks to the I350; it masks all interrupts and does not write to the Transmit Descriptor Tail (*TDT*) register or to the Receive Descriptor Tail (*RDT*) register and operates the master disable algorithm as defined in Section 5.2.3.3.

If wake up capability is needed, system should enable wake capability by setting to 1b the *PME_En* bit in the *PMCSR* PCIe configuration register. After Wake capability has been enabled Software device driver should set up the appropriate wake up registers (*WUC*, *WUFC* and associated filters) prior to the D3 transition.

Note: Software driver can override the *PMCSR.PME_En* bit setting via the *WUC.APMPME* bit.

If Protocol offload (Proxying) capability is required and the *MANC.MPROXYE* bit is set to 1b Software device driver should:

1. Send to the management Firmware the relevant protocol offload information (Type of protocol offloads required, MAC and IPv4/6 addresses information for protocol offload) via the shared RAM Firmware/Software Host interface as defined in Section 8.22.1, Section 10.8 and Section 10.8.2.4.2.
2. Program the *PROXYFC* register and associated filters according to the protocol offload required.
3. Program t*he WUC.PPROXYE* bit to 1b.

Note: If operation during $D3_{cold}$ is required, even when Wake capability is not required (e.g. for manageability operation), system should also set the *Auxiliary (AUX) Power PM Enable* bit in the PCIe *Device Control register*.

As a response to being programmed into D3 state, the I350 transitions its PCIe link into the L1 link state. As part of the transition into L1 state, the I350 suspends scheduling of new TLPs and waits for the completion of all previous TLPs it has sent. If the *PMCSR.No_Soft_reset* bit is cleared the I350 clears the *Memory Access Enable* and *I/O Access Enable* bits of the PCI Command register, which disables memory access decode. Any receive packets that have not been transferred into system memory are kept in the I350 (and discarded later on D3 exit). Any transmit packets that have not be sent can still be transmitted (assuming the Ethernet link is up).

In order to reduce power consumption, if the link is still needed for manageability, wake-up or proxying functionality, the PHY can auto-negotiate to a lower link speed on D3 entry (See Section 3.7.8.5.4). In addition in preparation for a possible transition to D3cold state, the device driver might disable some of the LAN ports to further reduce power consumption.

5.2.3.2 Exit from D3 State

A D3 state is followed by either a D0u state (in preparation for a D0a state) or by a transition to Dr state (PCI-PM D3cold state). To transition back to D0u, the system writes a 00b to the *Power State* field of the *Power Management Control/Status* Register (*PMCSR*). Transition to Dr state is through *PE_RST_N* assertion.

The *No_Soft_Reset* bit in the *PCIe Power Management Control / Status* (*PMCSR*) register in the I350 is set to 1b, to indicate that the I350 does not performs an internal reset on transition from D3hot to D0 so that transition will not disrupt the proper operation of other active Functions. In this case, software is not required to re-initialize the function's configuration space after a transition from D3hot to D0 (the Function will be in the $D0_{initialized}$ state), however the Software driver needs to re-initialize internal registers since transition from D3hot to D0 causes an internal port reset (similar to asserting the *CTRL.RST* bit).

The I350 can be configured via EEPROM to clear the *No_Soft_Reset* bit in the *PMCSR* register (See Section 6.2.17). In this case an internal reset is generated when transition from D3hot to D0 occurs and functional context is not maintained also in PCIe configuration bits (except for bits defined as sticky). As a result, in this case software is required to fully re-initialize the Function after a transition to D0 as the Function will be in the $D0_{uninitialized}$ state.

Note: The Function will be reset if the Link state transitions to the L2/L3 Ready state, on transition from D3cold to D0, if FLR is asserted or if transition D3hot to D0 is caused by assertion of PCIe reset (PE_RST pin) regardless of the value of the *No_Soft_Reset* bit.

5.2.3.3 Master Disable Via CTRL Register

System software can disable master accesses on the PCIe link by either clearing the PCI *Bus Master* bit or by bringing the function into a D3 state. From that time on, the I350 must not issue master accesses for this function. Due to the full-duplex nature of PCIe, and the pipelined design in the I350, it might happen that multiple requests from several functions are pending when the master disable request arrives. The protocol described in this section insures that a function does not issue master requests to the PCIe link after its *Master Enable* bit is cleared (or after entry to D3 state).

Two configuration bits are provided for the handshake between the I350 function and its software device driver:

- *GIO Master Disable* bit in the Device Control (*CTRL*) register - When the *GIO Master Disable* bit is set, the I350 blocks new master requests by this function. The I350 then proceeds to issue any pending requests by this function. This bit is cleared on master reset (LAN_PWR_GOOD, PCIe reset and software reset) to enable master accesses.
- *GIO Master Enable Status* bit in the Device Status (*STATUS*) register - Cleared by the I350 when the *GIO Master Disable* bit is set and no master requests are pending by the relevant function and is set otherwise. Indicates that no master requests are issued by this function as long as the *GIO Master Disable* bit is set. The following activities must end before the I350 clears the *GIO Master Enable Status* bit:
  - Master requests by the transmit and receive engines (for both data and MSI/MSIx interrupts).
  - All pending completions to the I350 are received.

In the event of a PCIe Master disable (Configuration *Command register.BME* set to 0) on a certain function or LAN port or if the function is moved into D3 state during a DMA access, the I350 generates an internal reset to the function and stops all port DMA accesses and interrupts related to the function. Following move to normal operating mode software driver should re-initialize the receive and transmit queues of the relevant port.

Notes: The software device driver sets the *GIO Master Disable* bit when notified of a pending master disable (or D3 entry). The I350 then blocks new requests and proceeds to issue any pending requests by this function. The software device driver then polls the *GIO Master Enable Status* bit. Once the bit is cleared, it is guaranteed that no requests are pending from this function. The software device driver might time out if the *GIO Master Enable Status* bit is not cleared within a given time.

The *GIO Master Disable* bit must be cleared to enable a master request to the PCIe link. This can be done either through reset or by the software device driver.

5.2.4 Dr State (D3cold)

Transition to Dr state is initiated on several occasions:

- On system power up - Dr state begins with the assertion of the internal power detection circuit and ends with de-assertion of *PE_RST_N*.
- On transition from a D0a state - During operation the system might assert *PE_RST_N* at any time. In an ACPI system, a system transition to the G2/S5 state causes a transition from D0a to Dr state.
- On transition from a D3 state - The system transitions the I350 into the Dr state by asserting PCIe *PE_RST_N*.

Any wake-up filter settings or proxying filter settings that were enabled before entering this reset state are maintained.

The system might maintain *PE_RST_N* asserted for an arbitrary time. The de-assertion (rising edge) of *PE_RST_N* causes a transition to D0u state.

While in Dr state, the I350 might enter one of several modes with different levels of functionality and power consumption. The lower-power modes are achieved when the I350 is not required to maintain any functionality (see Section 5.2.4.1).

Note: If the I350 is configured to provide a 50 MHz NC-SI clock (via the *NC-SI Output Clock* EEPROM bit), then the NC-SI clock must be provided in Dr state as well.

5.2.4.1 Dr Disable Mode

The I350 enters a Dr disable mode on transition to D3cold state when it does not need to maintain any functionality. The conditions to enter either state are:

- The I350 (all PCI functions) is in Dr state
- APM WoL (Wake-on-LAN) is inactive for all LAN functions
- Proxying is not required for all LAN functions (*WUC.PPROXYE* is cleared to 0b).
- Pass-through manageability is disabled
- ACPI PME is disabled for all PCI functions
- The I350 *Power Down Enable* EEPROM bit (word 0x1E, bit 15) is set (default hardware value is disabled).
- The *PHY Power Down Enable* EEPROM bit is set (word 0xF, bit 6).

Entering Dr disable mode is usually done by asserting PCIe *PE_RST_N*. It might also be possible to enter Dr disable mode by reading the EEPROM while already in Dr state. The usage model for this later case is on system power up, assuming that manageability, wake up and proxying are not required. Once the I350 enters Dr state on power-up, the EEPROM is read. If the EEPROM contents determine that the conditions to enter Dr disable mode are met, the I350 then enters this mode (assuming that PCIe *PE_RST_N* is still asserted).

The I350 exits Dr disable mode when Dr state is exited (See Figure 5-1 for conditions to exit Dr state).

5.2.4.2 Entry to Dr State

Dr entry on platform power-up begins with the assertion of the internal power detection circuit. The EEPROM is read and determines the I350 configuration. If the *APM Enable* bit in the EEPROM's *Initialization Control Word 3* is set, then APM wake up is enabled. PHY and MAC states are redetermined by the state of manageability and APM wake. To reduce power consumption, if manageability or APM wake is enabled, the PHY auto-negotiates to a lower link speed on Dr entry (See Section 3.7.8.5.4). The PCIe link is not enabled in Dr state following system power up (since *PE_RST_N* is asserted).

Entering Dr state from D0a state is done by asserting *PE_RST_N*. An ACPI transition to the G2/S5 state is reflected in the I350 transition from D0a to Dr state. The transition can be orderly (such as user selecting the shut down option), in which case the software device driver might have a chance to intervene. Or, it might be an emergency transition (such as power button override), in which case, the software device driver is not notified.

To reduce power consumption, if any of manageability, APM wake or PCI-PM PME[1] is enabled, the PHY auto-negotiates to a lower link speed on D0a to Dr transition (see Section 3.7.8.5.4).

Transition from D3 (hot) state to Dr state is done by asserting *PE_RST_N*. Prior to that, the system initiates a transition of the PCIe link from L1 state to either the L2 or L3 state (assuming all functions were already in D3 state). The link enters L2 state if PCI-PM PME is enabled.

5.2.4.3 Auxiliary Power Usage

The EEPROM *D3COLD_WAKEUP_ADVEN* bit and the *AUX_PWR strapping pin determine when D3cold PME is supported:*

- D3COLD_WAKEUP_ADVEN denotes that PME wake should be supported
- AUX_PWR strapping pin indicates that auxiliary power is provided

D3cold PME is supported as follows:

- If the *D3COLD_WAKEUP_ADVEN is set to '1' and* the AUX_PWR strapping is set to '1', then *D3cold PME is supported*
- Else *D3cold PME is not supported*

The amount of power required for the function (including the entire NIC) is advertised in the *Power Management Data* register, which is loaded from the EEPROM.

If D3cold is supported, the *PME_En* and *PME_Status* bits of the *Power Management Control/Status* Register (PMCSR), as well as their shadow bits in the Wake Up Control (WUC) register are reset only by the power up reset (detection of power rising).

5.2.5 Link Disconnect

In any of D0u, D0a, D3, or Dr power states, the I350 enters a link-disconnect state if it detects a link-disconnect condition on the Ethernet link. Note that the link-disconnect state in the internal PHY is invisible to software (other than the *PHPM.Link Energy Detect* bit state). In particular, while in D0 state, software might be able to access any of the I350 registers as in a link-connect state.

5.2.6 Device Power-Down State

The I350 enters a global power-down state if all of the following conditions are met:

- The I350 *Power Down Enable* EEPROM bit (word 0x1E bit 15) was set (default hardware value is disabled).
- The I350 is in Dr state.

1. ACPI 2.0 specifies that "OSPM will not disable wake events before setting the SLP_EN bit when entering the S5 sleeping state. This provides support for remote management initiatives by enabling Remote Power On (RPO) capability. This is a change from ACPI 1.0 behavior."

- The link connections of all ports (PHY or SerDes) are in power down mode.

The I350 also enters a power-down state when the DEV_OFF_N pin is asserted and the relevant EEPROM bits were configured as previously described (see Section 4.5 for more details on DEV_OFF_N functionality).

5.3 Power Limits by Certain Form Factors

Table 5-1 lists power limitation introduced by different form factors.

Table 5-1 Power Limits by Form-Factor

	Form Factor	
	LOM	PCIe add-in card (10 W slot)
Main	N/A	3 A @ 3.3 V
Auxiliary (aux enabled)	375 mA @ 3.3 V	375 mA @ 3.3 V
Auxiliary (aux disabled)	20 mA @ 3.3 V	20 mA @ 3.3 V

Note: This auxiliary current limit only applies when the primary 3.3 V voltage source is not available (the card is in a low power D3 state).

The I350 exceeds the allocated auxiliary power in some configurations (such as all ports running at 1000 Mb/s speed). The I350 must therefore be configured to meet the previously mentioned certain requirements. To do so, the I350 implements three EEPROM bits to disable operation in certain cases:

1. The *PHPM.Disable_1000* PHY register bit disables 1000 Mb/s operation under all conditions.
2. The *PHPM.Disable 1000 in non-D0a* PHY CSR bit disables 1000 Mb/s operation in non-D0a states[1]. If *PHPM.Disable 1000 in non-D0a* is set, and the I350 is at 1000 Mb/s speed on entry to a non-D0a state, then the I350 removes advertisement for 1000 Mb/s and auto-negotiates.
3. The *PHPM.Disable 100 in non-D0a* PHY CSR bit disables 1000 Mb/s and 100 Mb/s operation in non-D0a states. If *PHPM.Disable 100 in non-D0a* is set, and the I350 is at 1000 Mb/s or 100 Mb/s speeds on entry to a non-D0a state, then the I350 removes advertisement for 1000 Mb/s and 100 Mb/s and auto-negotiates.

Note that the I350 restarts link auto-negotiation each time it transitions from a state where 1000 Mb/s or 100 Mb/s speed is enabled to a state where 1000 Mb/s or 100 Mb/s speed is disabled, or vice versa. For example, if *PHPM.Disable 1000 in non-D0a* is set but *PHPM.Disable_1000* is cleared, the I350 restarts link auto-negotiation on transition from D0 state to D3 or Dr states.

5.4 Interconnects Power Management

This section describes the power reduction techniques employed by the I350 main interconnects.

1. The restriction is defined for all non-D0a states to have compatible behavior with previous products.

5.4.1 PCIe Link Power Management

The PCIe link state follows the power management state of the I350. Since the I350 incorporates multiple PCI functions, its power management state is defined as the power management state of the most awake function (see Figure 5-2):

- If any function is in D0 state (either D0a or D0u), the PCIe link assumes the I350 is in D0 state. Else,
- If the functions are in D3 state, the PCIe link assumes the I350 is in D3 state. Else,
- The I350 is in Dr state (PE_RST_N is asserted to all functions).

The I350 supports all PCIe power management link states:

- L0 state is used in D0u and D0a states.
- The L0s state is used in D0a and D0u states each time link conditions apply.
- The L1 state is also used in D0a and D0u states when idle conditions apply for a longer period of time. The L1 state is also used in the D3 state.
- The L2 state is used in the Dr state following a transition from a D3 state if *PCI-PM PME* is enabled.
- The L3 state is used in the Dr state following power up, on transition from D0a, and if *PME* is not enabled in other Dr transitions.

The I350 support for active state link power management is reported via the PCIe *Active State Link PM Support* register and is loaded from the EEPROM.

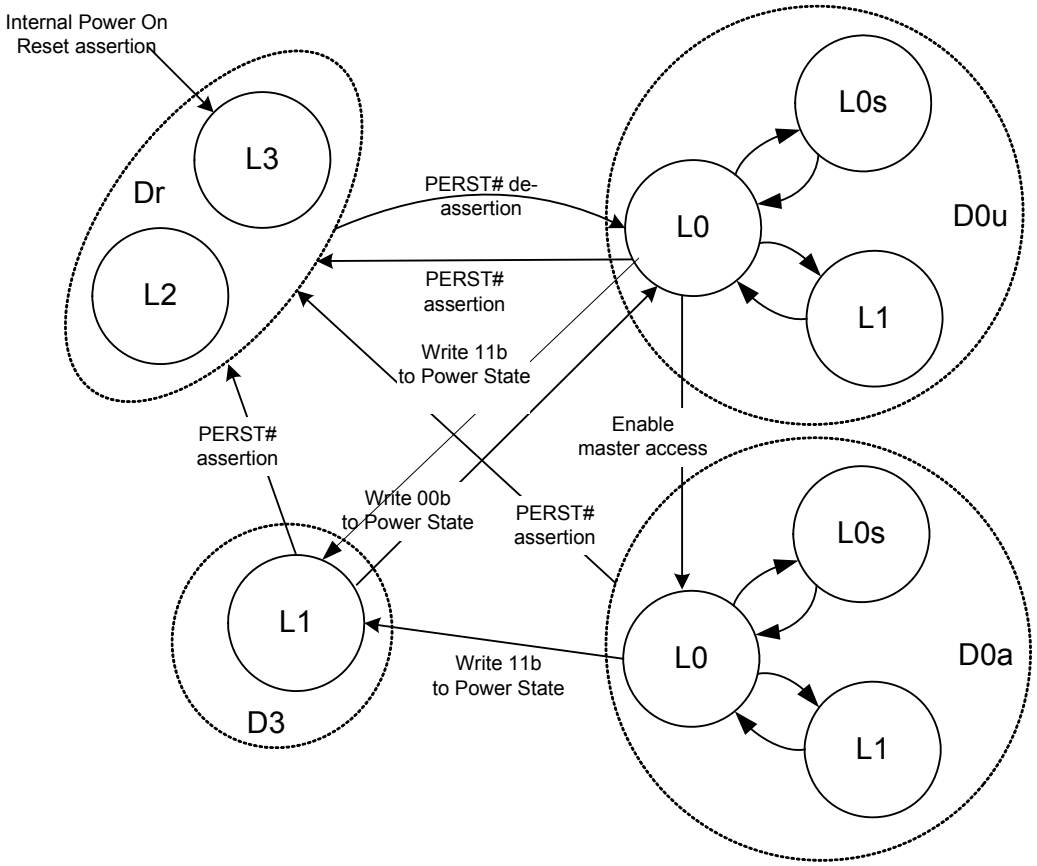


Figure 5-2 Link Power Management State Diagram

While in L0 state, the I350 transitions the transmit lane(s) into L0s state once the idle conditions are met for a period of time as follows:

L0s configuration fields are:

- L0s enable - The default value of the *Active State Link PM Control* field in the PCIe *Link Control* Register is set to 00b (both L0s and L1 disabled). System software may later write a different value into the Link Control Register. The default value is loaded on any reset of the PCI configuration registers.
- L0s exit latency (as published in the L0s Exit Latency field of the Link Capabilities Register) is loaded from EEPROM. Separate values are loaded when the I350 shares the same reference PCIe clock with its partner across the link, and when the I350 uses a different reference clock than its partner across the link. The I350 reports whether it uses the slot clock configuration through the PCIe Slot Clock Configuration bit loaded from the *Slot_Clock_Cfg* bit in the *PCIe Init Configuration 3* EEPROM Word.
- L0s Acceptable Latency (as published in the Endpoint L0s Acceptable Latency field of the Device Capabilities Register) is loaded from EEPROM.

The I350 The I350transitions the PCIe link into low power state as defined in the *DMACR.DMAC_Lx* field once it detects no PCIe activity for a duration defined in the *DMCTLX.TTLX* field.

Note: To comply with the PCIe specification if the link idle time exceeds the Latency_To_Enter_L0s value defined in the EEPROM then the I350 will enter L0s even if the PCIe idle time defined in the *DMCTLX.TTLX* field has not expired. Once the PCIe idle time defined in the *DMCTLX.TTLX* has expired the I350 enters L1 according to the programming of the *DMACR.DMAC_Lx* field.

The following EEPROM fields control L1 behavior:

- *Act_Stat_PM_Sup* - Indicates support for ASPM L1 in the PCIe configuration space (loaded into the Active State Link PM Support field)
- *L1_Act_Ext_Latency* - Defines L1 active exit latency
- *L1_Act_Acc_Latency* - Defines L1 active acceptable exit latency
- *Latency_To_Enter_L1* - Defines the period (in the L0s state) before the transition into L1 state

5.4.2 NC-SI Clock Control

The I350 can be configured to provide a 50 MHz output clock to its NC-SI interface and other platform devices. When enabled through the *NC-SI Output Clock Disable* EEPROM bit (See Section 6.2.22), the NC-SI clock is provided in all power states without exception.

5.4.3 Internal PHY Power-Management

The PHY power management features are described in Section 3.7.8.5.

5.5 Timing of Power-State Transitions

The following sections give detailed timing for the state transitions. In the diagrams the dotted connecting lines represent I350 requirements, while the solid connecting lines represent the I350 guarantees.

The timing diagrams are not to scale. The clocks edges are shown to indicate running clocks only and are not to be used to indicate the actual number of cycles for any operation.

5.5.1 Power Up (Off to Dup to D0u to D0a)

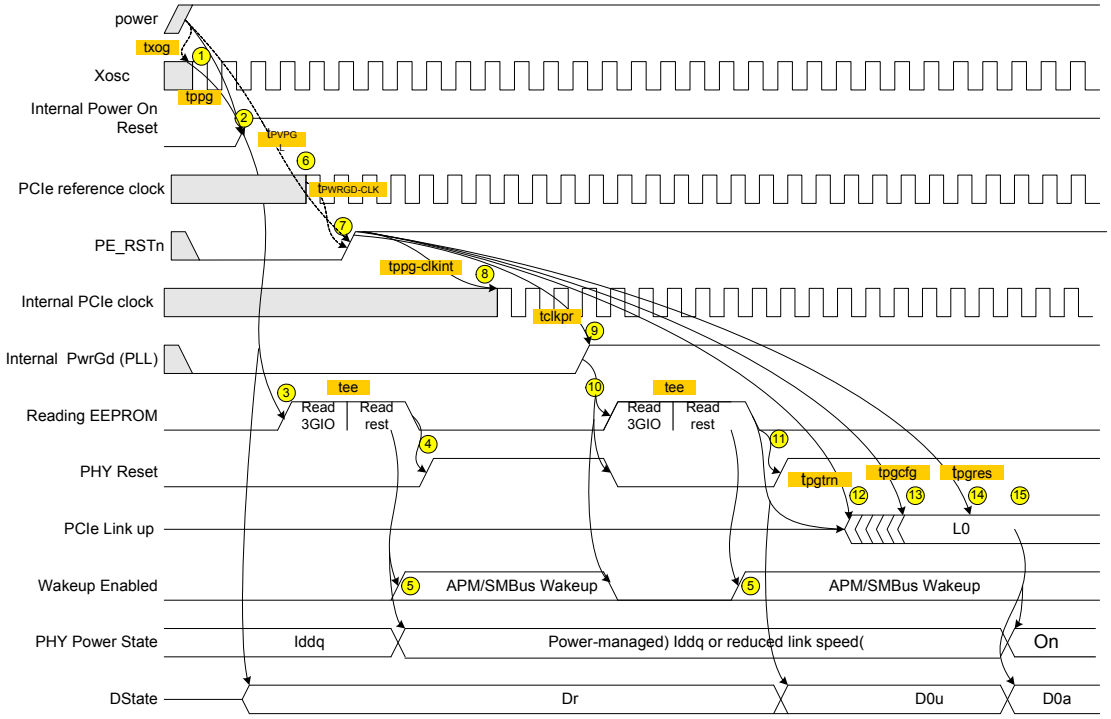


Figure 5-3 Power Up (Off to Dup to D0u to D0a)

Table 5-2 Power Up (Off to Dup to D0u to D0a)

Note	Description
1	Xosc is stable t_{xog} after power is stable.
2	LAN_PWR_GOOD is asserted after all power supplies are good and t_{ppg} after Xosc is stable.
3	An EEPROM read starts on the rising edge of LAN_PWR_GOOD.
4	After reading the EEPROM, PHY reset is de-asserted.
5	APM wake-up mode can be enabled based on what is read from the EEPROM.
6	The PCIe reference clock is valid $t_{PE_RST-CLK}$ before de-asserting PE_RST_N (according to PCIe specification).
7	PE_RST_N is de-asserted t_{PVPGL} after power is stable (according to PCIe specification).
8	The internal PCIe clock is valid and stable $t_{ppg-clkint}$ from PE_RST_N de-assertion.
9	The PCIe internal PWRGD signal is asserted t_{clkpr} after the external PE_RST_N signal.
10	Asserting internal PCIe PWRGD causes the EEPROM to be re-read, asserts PHY reset, and disables wake up.
11	After reading the EEPROM, PHY reset is de-asserted.
12	Link training starts after t_{pgtrn} from PE_RST_N de-assertion.
13	A first PCIe configuration access might arrive after t_{pgcfg} from PE_RST_N de-assertion.
14	A first PCI configuration response can be sent after t_{pgres} from PE_RST_N de-assertion.
15	Writing a 1b to the *Memory Access Enable* bit in the PCI Command Register transitions the I350 from D0u to D0. state.

5.5.2 Transition from D0a to D3 and Back Without PE_RST_N

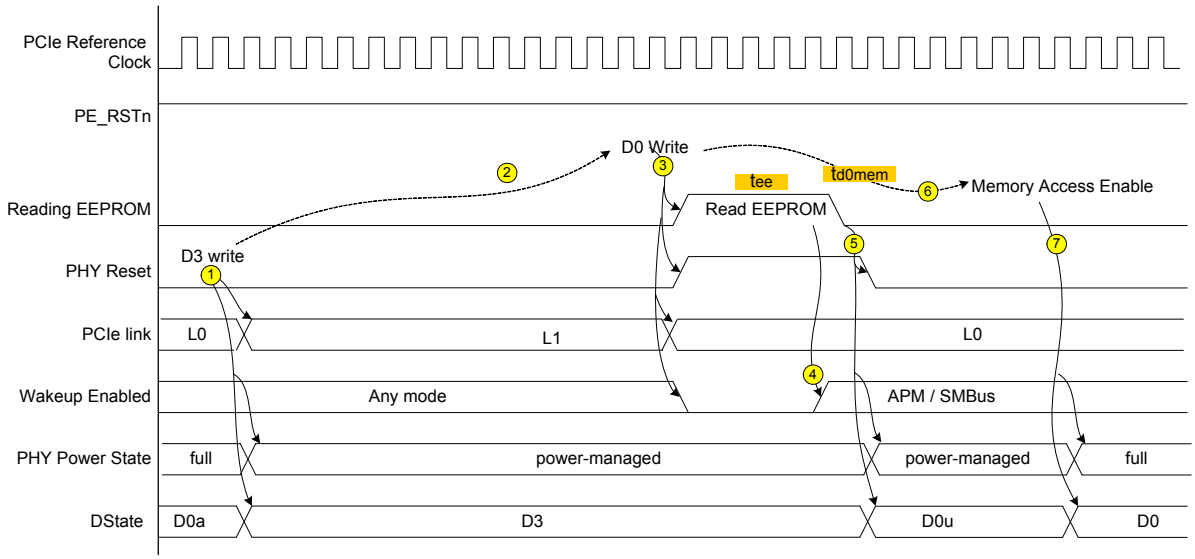


Figure 5-4 Transition from D0a to D3 and Back Without PE_RST_N

Table 5-3 Transition from D0a to D3 and Back Without PE_RST_N

Note	Description
1	Writing 11b to the *Power State* field of the Power Management Control/Status Register (PMCSR) transitions the I350 to D3.
2	The system can keep the I350 in D3 state for an arbitrary amount of time.
3	To exit D3 state, the system writes 00b to the *Power State* field of the PMCSR.
4	APM wake-up or SMBus mode might be enabled based on what is read in the EEPROM.
5	After reading the EEPROM, reset to the PHY is de-asserted. The PHY operates at reduced-speed if APM wake up or SMBus is enabled, else powered-down.
6	The system can delay an arbitrary time before enabling memory access.
7	Writing a 1b to the *Memory Access Enable* bit or to the *I/O Access Enable* bit in the PCI Command Register transitions the I350 from D0u to D0 state and returns the PHY to full-power/speed operation.

5.5.3 Transition From D0a to D3 and Back With PE_RST_N

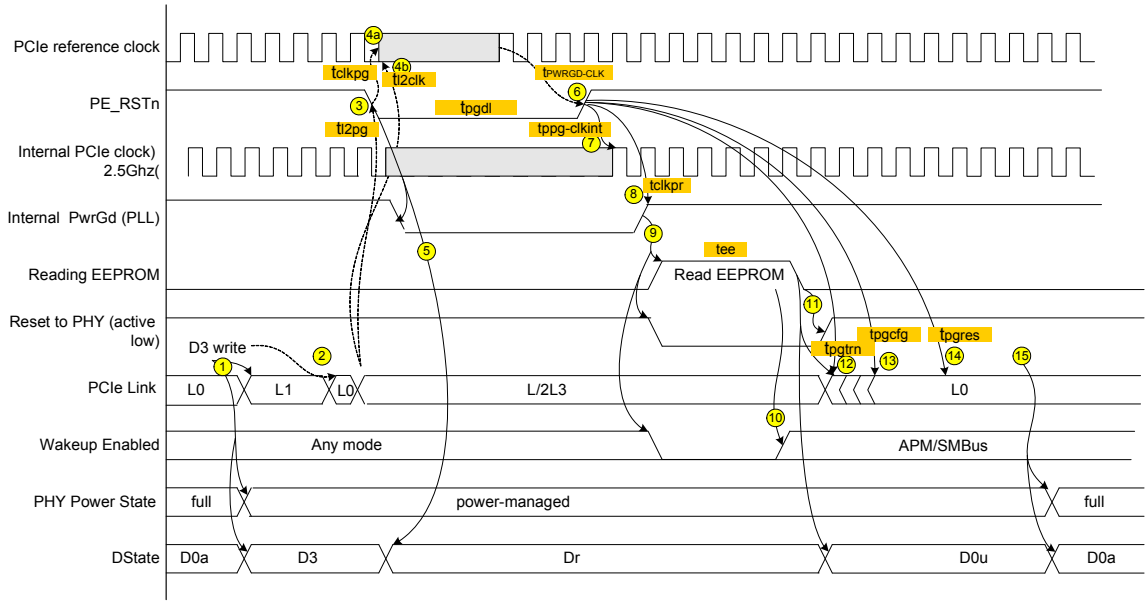


Figure 5-5 Transition From D0a to D3 and Back With PE_RST_N

Table 5-4 Transition From D0a to D3 and Back With PE_RST_N

Note	Description
1	Writing 11b to the *Power State* field of the PMCSR transitions the I350 to D3. PCIe link transitions to L1 state.
2	The system can delay an arbitrary amount of time between setting D3 mode and moving the link to a L2 or L3 state.
3	Following link transition, PE_RST_N is asserted.
4	The system must assert PE_RST_N before stopping the PCIe reference clock. It must also wait t_{l2clk} after link transition to L2/L3 before stopping the reference clock.
5	On assertion of PE_RST_N, the I350 transitions to Dr state.
6	The system starts the PCIe reference clock $t_{PE_RST-CLK}$ before de-assertion PE_RST_N.
7	The internal PCIe clock is valid and stable $t_{ppg-clkint}$ from PE_RST_N de-assertion.
8	The PCIe internal PWRGD signal is asserted t_{clkpr} after the external PE_RST_N signal.
9	Asserting internal PCIe PWRGD causes the EEPROM to be re-read, asserts PHY reset, and disables wake up.
10	APM wake-up mode might be enabled based on what is read from the EEPROM.
11	After reading the EEPROM, PHY reset is de-asserted.
12	Link training starts after t_{pgtrn} from PE_RST_N de-assertion.
13	A first PCIe configuration access might arrive after t_{pgcfg} from PE_RST_N de-assertion.
14	A first PCI configuration response can be sent after t_{pgres} from PE_RST_N de-assertion.
15	Writing a 1b to the *Memory Access Enable* bit in the PCI Command Register transitions the I350 from D0u to D0 state.

5.5.4 Transition From D0a to Dr and Back Without Transition to D3

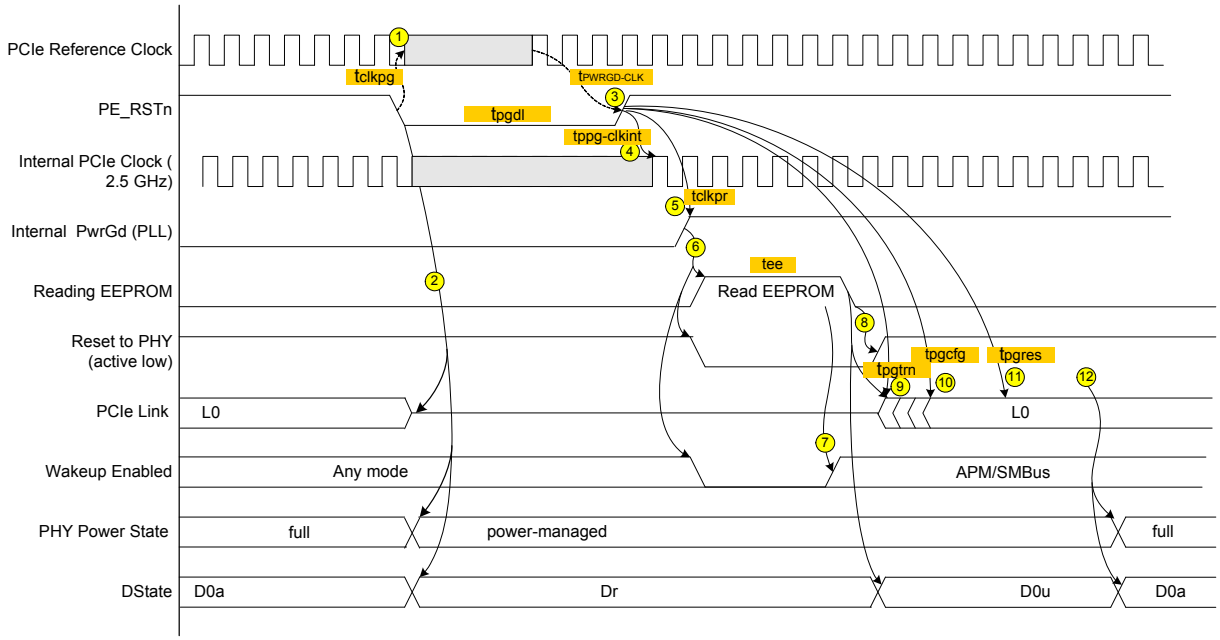


Figure 5-6 Transition From D0a to Dr and Back Without Transition to D3

Table 5-5 Transition From D0a to Dr and Back Without Transition to D3

Note	Description
1	The system must assert PE_RST_N before stopping the PCIe reference clock. It must also wait t_{l2clk} after link transition to L2/L3 before stopping the reference clock.
2	On assertion of PE_RST_N, the I350 transitions to Dr state and the PCIe link transition to electrical idle.
3	The system starts the PCIe reference clock $t_{PE_RST-CLK}$ before de-assertion PE_RST_N.
4	The internal PCIe clock is valid and stable $t_{ppg-clkint}$ from PE_RST_N de-assertion.
5	The PCIe internal PWRGD signal is asserted t_{clkpr} after the external PE_RST_N signal.
6	Asserting internal PCIe PWRGD causes the EEPROM to be re-read, asserts PHY reset, and disables wake up.
7	APM wake-up mode might be enabled based on what is read from the EEPROM.
8	After reading the EEPROM, PHY reset is de-asserted.
9	Link training starts after t_{pgtrn} from PE_RST_N de-assertion.
10	A first PCIe configuration access might arrive after t_{pgcfg} from PE_RST_N de-assertion.
11	A first PCI configuration response can be sent after t_{pgres} from PE_RST_N de-assertion.
12	Writing a 1b to the *Memory Access Enable* bit in the PCI Command Register transitions the I350 from D0u to D0 state.

5.6 Wake Up

The I350 supports two modes of wake-up management:

1. Advanced Power Management (APM) wake up
2. ACPI/PCIe defined wake up

The usual model is to activate one mode at a time but not both modes together. If both modes are activated, the I350 might wake up the system on unexpected events. For example, if APM is enabled together with the ACPI/PCIe Magic packet in the *WUFC* register, a magic packet might wake up the system even if APM is disabled (*WUC.APME* = 0b). Alternatively, if APM is enabled together with some of the ACPI/PCIe filters (enabled in the *WUFC* register), packets matching these filters might wake up the system even if PCIe PME is disabled.

5.6.1 Advanced Power Management Wake Up

Advanced Power Management Wake Up or APM Wakeup (also known as Wake on LAN) is a feature that existed in earlier 10/100 Mb/s NICs. This functionality was designed to receive a broadcast or unicast packet with an explicit data pattern, and then assert a subsequent signal to wake up the system. This was accomplished by using a special signal that ran across a cable to a defined connector on the motherboard. The NIC would assert the signal for approximately 50 ms to signal a wake up. The I350 now uses (if configured) an in-band PM_PME message for this functionality.

On power up, the I350 reads the *APM Enable* bits from the EEPROM *Initialization Control Word 3* into the *APM Enable (APME)* bits of the *Wakeup Control (WUC)* register. These bits control enabling of APM wake up.

When APM wake up is enabled, the I350 checks all incoming packets for Magic Packets. See Section 5.6.3.1.4 for a definition of Magic Packets.

Once the I350 receives a matching Magic packet, and if the *WUC.APMPME* bit or the *PMCSR.PME_En* bits are set to 1b and the *WUC.APME* bit is set to 1b it:

- Sets the *PME_Status* bit in the *PMCSR* register and issues a PM_PME message (in some cases, this might require asserting the PE_WAKE_N signal first to resume power and clock to the PCIe interface).
- Stores the first 128 bytes of the packet in the Wake Up Packet Memory (*WUPM*) register.
- Sets the *Magic Packet Received* bit in the Wake Up Status (*WUS*) register.
- Sets the packet length in the Wake Up Packet Length (*WUPL*) register.

The I350 maintains the first Magic Packet received in the *Wake Up Packet Memory* (*WUPM*) register until the software device driver writes a 1b to the *WUS.MAG* bit.

If the *WUC.EN_APM_D0* bit is set to 1b APM wake up is supported in all power states and only disabled if a subsequent EEPROM read results in the *WUC.APME* bit being cleared or software explicitly writes a 0b to the *WUC.APME* bit. If the *WUC.EN_APM_D0* bit is cleared APM wake-up is supported only in the D3 or Dr power states.

Notes:

1. When the *WUC.APMPME* bit is set a wake event is issued (PE_WAKE_N pin is asserted and a PM_PME PCIe message is issued) even if the *PMCSR.PME_En* bit in configuration space is cleared.

To enable disabling of system Wake-up when *PMCSR.PME_En* is cleared, Software driver should clear the *WUC.APMPME* bit after power-up or PCIe reset.

2. So long as APM is enabled and the I350 is programmed to issue a wake event on the PCIe, each time a Magic packet is received. A wake event is generated on the PCIe interface even if the *WUS.MAG* bit is set as a result of reception of a previous Magic packet. Consecutive magic packets will generate consecutive Wake events.

5.6.2 ACPI Power Management Wake Up

The I350 supports PCIe power management based wake-up. It can generate system wake-up events from a number of sources:

- Reception of a Magic Packet.
- Reception of a network wakeup packet.
- Detection of change in network link state (cable connected or disconnected).
- Detection of a thermal event due to crossing of a thermal trip point.
- Wake-up by Manageability on reception of an unsupported packets for Proxying.

Activating PCIe power management wake up requires the following:

- System software writes at configuration time a 1b to the PCI *PMCSR.PME_En* bit.
- Software device driver clears all pending wake-up status bits in the Wake Up Status (*WUS*) register.
- The software device driver programs the Wake Up Filter Control (*WUFC*) register to indicate the packets that should initiate system wake up and programs the necessary data to the IPv4/v6 Address Table (*IP4AT, IP6AT*) and the Flexible Host Filter Table (*FHFT*). It can also set the *WUFC.LNKC* bit and/or the *WUFC.TS* bit to cause wake up on link status change or thermal event.
- Once the I350 wakes the system, the driver needs to clear the *WUS* and *WUFC* registers until the next time the system moves to a low power state with wake up enabled.

Normally, after enabling wake up, system software moves the device to D3 low power state by writing a 11b to the PCI *PMCSR.Power State* field.

Once wake up is enabled, the I350 monitors incoming packets, first filtering them according to its standard address filtering method, then filtering them with all of the enabled wakeup filters. If a packet passes both the standard address filtering and at least one of the enabled wakeup filters, the I350:

- Sets the *PME_Status* bit in the *PMCSR*.
- Asserts PE_WAKE_N (if the *PME_En* bit in the *PMCSR* configuration register is set).
- Stores the first 128 bytes of the packet in the *Wakeup Packet Memory (WUPM)* register.
- Sets one or more bits in the *Wake Up Status (WUS)* register. Note that the I350 sets more than one bit if a packet matches more than one filter.
- Sets the packet length in the *Wake Up Packet Length (WUPL)* register.

Note: If enabled, a link state change wake-up or a thermal sensor event wake-up also causes similar results. Sets the *PMCSR.PME_Status* bit, asserts the *PE_WAKE_N* signal and sets the relevant bit in the *WUS* register.

The PE_WAKE_N remains asserted until the operating system either writes a 1b to the *PMCSR.PME_Status* bit or writes a 0b to the *PMCSR.PME_En* bit.

After receiving a wake-up packet, the I350 ignores any subsequent wake-up packets until the software device driver clears all of the received bits in the Wake Up Status (WUS) register. It also ignores link change events until the software device driver clears the *Link Status Changed* (*LNKC*) bit in the *Wake Up Status* (*WUS*) register.

Note: A wake on link change is not supported when configured to SerDes or 1000BASE-KX mode.

5.6.3 Wake-Up and Proxying Filters

The I350 supports issuing wake-up to Host when device is in D3 or protocol offload (Proxying) of packets using two types of filters:

- Pre-defined filters
- Flexible filters

Each of these filters are enabled if the corresponding bit in the *Wake Up Filter Control* (*WUFC*) register or *Proxying Filter Control* (*PROXYFC*) register is set to 1b.

Note: When VLAN filtering is enabled, packets that passed any of the receive wake-up filters should only cause a wake-up event if they also passed the VLAN filtering.

5.6.3.1 Pre-Defined Filters

The following packets are supported by the I350's pre-defined filters:

- Directed packet (including exact, multicast indexed, and broadcast)
- Magic Packet
- ARP/IPv4 request packet
- Directed IPv4 packet
- Directed IPv6 packet
- ICMPv6 packet like:
  - — IPv6 Neighbor Solicitation (NS) packet
  - — IPV6 Multicast Listener Discovery (MLD) packet

Each of these filters are enabled if the corresponding bit in the *Wakeup Filter Control (WUFC)* register or *Proxying Filter Control* (*PROXYFC*) register is set to 1b.

Following sections include a description of each filter and a table describing which bytes at which offsets need to be compared, to determine if the packet passes the filter.

Note: Both VLAN fields and LLC/SNAP fields can increase the given offsets if they are present.

5.6.3.1.1 Directed Exact Packet

The I350 generates a wake-up event after receiving any packet whose destination address matches one of the 32 valid programmed receive destination MAC addresses (defined in the *RAL* and *RAH* registers), if the *Directed Exact Wake Up Enable* bit is set in the *Wake Up Filter Control* (*WUFC.EX*) register.

The I350 forwards a packet to Management for Proxying after receiving any packet whose MAC destination address matches one of the 32 valid programmed receive destination MAC addresses (defined in the *RAL* and *RAH* registers), if the *Directed Exact Proxying Enable* bit is set in the *Proxying Filter Control* (*PROXYFC*) register.

Offset	# of bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	Match any pre-programmed address

5.6.3.1.2 Directed Multicast Packet

For multicast packets, the upper bits of the incoming packet's destination address index a bit vector, the Multicast Table Array (*MTA*) that indicates whether to accept the packet.

If the *Directed Multicast Wake Up Enable* bit set in the *Wake Up Filter Control* (*WUFC.MC*) register and the indexed bit in the vector is one, then the I350 generates a wake-up event. If the *Directed Multicast Proxying Enable* bit is set in the *Proxying Filter Control* (*PROXYFC*) register and the indexed bit in the vector is one, then the I350 forwards packet to Management for Proxying.

Note: The exact bits used in the comparison are programmed by software in the *Multicast Offset* field of the Receive Control (*RCTL.MO*) register.

5.6.3.1.3 Broadcast

If the Broadcast Wake Up Enable bit in the Wake Up Filter Control (WUFC.BC) register is set, the I350 generates a wake-up event when it receives a broadcast packet.If the Broadcast Proxying Enable bit in the Proxying Filter Control (PROXYFC) register is set, the I350 forwards packet to Management for Proxying when receiving a broadcast packet.

Offset	# of bytes	Field	Value	Action	Comment
0	6	Destination Address	FF*6	Compare	

5.6.3.1.4 Magic Packet

Magic packets are defined in:

http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/20213.pdf as:

"Once the LAN controller has been put into the Magic Packet mode, it scans all incoming frames addressed to the node for a specific data sequence. This sequence indicates to the controller that this is a Magic Packet frame. A Magic Packet frame must also meet the basic requirements for the LAN technology chosen, such as SOURCE ADDRESS, DESTINATION ADDRESS (which may be the receiving station's IEEE address or a MULTICAST address which includes the BROADCAST address), and CRC. The specific data sequence consists of 16 repetitions of the IEEE address of this node, with no breaks or interruptions. This sequence can be located anywhere within the packet, but must be preceded by a synchronization stream. The synchronization stream allows the scanning state machine to be much simpler. The synchronization stream is defined as 6 bytes of 0xFF. The device will also accept a BROADCAST frame, as long as the 16 repetitions of the IEEE address match the address of the machine to be awakened."

The I350 expects the destination address to either:

- Be the broadcast address (FF.FF.FF.FF.FF.FF)
- Match the value in Receive Address 0 (*RAH0*, *RAL0*) register. This is initially loaded from the EEPROM but can be changed by the software device driver.
- Match any other address filtering (*RAH*[n], *RAL*[n]) enabled by the software device driver.

The I350 searches for the contents of Receive Address 0 (*RAH0*, *RAL0*) register as the embedded IEEE address. It considers any non-0xFF byte after a series of at least 6 0xFFs to be the start of the IEEE address for comparison purposes. For example, it catches the case of 7 0xFFs followed by the IEEE address). As soon as one of the first 96 bytes after a string of 0xFFs don't match, it continues to search for another set of at least 6 0xFFs followed by the 16 copies of the IEEE address later in the packet. Note that this definition precludes the first byte of the destination address from being FF.

A Magic Packet's destination address must match the address filtering enabled in the configuration registers with the exception that broadcast packets are considered to match even if the *Broadcast Accept* bit of the *Receive Control (RCTL.BAM)* register is 0b. If APM wake up (wake up by a Magic Packet) is enabled in the EEPROM, the I350 starts up with the Receive Address 0 (*RAH0*, *RAL0*) register loaded from the EEPROM. This enables the I350 to accept packets with the matching IEEE address before the software device driver loads.

Table 5-6 Magic Packet Structure

Offset	# of bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC header – processed by main address filter.
6	6	Source Address		Skip	
12	S=(0/4)	Possible VLAN Tag		Skip	
12 + S	D=(0/8)	Possible Length + LLC/SNAP Header		Skip	
12 + S + D	2	Type		Skip	
Any	6	Synchronizing Stream	FF*6+	Compare	
any+6	96	16 copies of Node Address	A*16	Compare	Compared to Receive Address 0 (RAH0, RAL0) register.

5.6.3.1.5 ARP/IPv4 Request Packet

The I350 supports receiving ARP request packets for wake up if the *Directed ARP* bit or the *ARP* bit is set in the *Wake Up Filter Control* (*WUFC*) register and Proxying if the *Directed ARP* bit or the *ARP* bit is set in the *Proxying Filter Control* (*PROXYFC*) register.

- If the *Directed ARP* bit is set, a successfully matched packet must contain a broadcast MAC address, match VLAN tag if programmed, a Ethernet type of 0x0806, an ARP op-code of 0x01 and the Target IP address matches one of the four IPv4 addresses programmed in the *IPv4 Address Table* (*IP4AT*).
- If the *ARP* bit is set, a successfully matched packet must contain a broadcast MAC address, match VLAN tag if programmed, a Ethernet type of 0x0806 and an ARP op-code of 0x01.

The I350 also handles ARP request packets that have VLAN tagging on both Ethernet II and Ethernet SNAP types.

Table 5-7 ARP Packet Structure and Processing

Offset	# of bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC header – processed by main address filter.
6	6	Source Address		Skip	
12	S=(0/4)	Possible VLAN Tag		Compare	Processed by main address filter.
12 + S	D=(0/8)	Possible Length + LLC/SNAP Header		Skip	
12 + S + D	2	Ethernet Type	0x0806	Compare	ARP
14 + S + D	2	HW Type	0x0001	Compare	
16 + S + D	2	Protocol Type	0x0800	Compare	
18 + S + D	1	Hardware Size	0x06	Compare	
19 + S + D	1	Protocol Address Length	0x04	Compare	
20 + S + D	2	Operation	0x0001	Compare	
22 + S + D	6	Sender HW Address	-	Ignore	
28 + S + D	4	Sender IP Address	-	Ignore	
32 + S + D	6	Target HW Address	-	Ignore	
38 + S + D	4	Target IP Address	IP4AT	Compare	Compare if the *Directed ARP* bit is set to 1b. May match any of four values in *IP4AT*.

5.6.3.1.6 Directed IPv4 Packet

The I350 supports receiving directed IPv4 packets for wake up if the *IPV4* bit is set in the Wake Up Filter Control (*WUFC*) register and Proxying if the *IPV4* bit is set in the *Proxying Filter Control* (*PROXYFC*) register.

Four IPv4 addresses are supported, which are programmed in the IPv4 Address Table (*IP4AT*). A successfully matched packet must contain the station's MAC address, match VLAN tag if programmed, a Ethernet type of 0x0800, and one of the four programmed IPv4 addresses. The I350 also handles directed IPv4 packets that have VLAN tagging on both Ethernet II and Ethernet SNAP types.

Table 5-8 IPv4 Packet Structure and Processing

Offset	# of bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC header – processed by main address filter.
6	6	Source Address		Skip	
12	S=(0/4)	Possible VLAN Tag		Compare	Processed by main address filter.
12 + S	D=(0/8)	Possible Length + LLC/SNAP Header		Skip	
12 + S + D	2	Ethernet Type	0x0800	Compare	IPv4
14 + S + D	1	Version/ HDR length	0x4X	Compare	Check IPv4
15 + S + D	1	Type of Service	-	Ignore	
16 + S + D	2	Packet Length	-	Ignore	
18 + S + D	2	Identification	-	Ignore	
20 + S + D	2	Fragment Info	-	Ignore	
22 + S + D	1	Time to live	-	Ignore	

Table 5-8 IPv4 Packet Structure and Processing (Continued)

Offset	# of bytes	Field	Value	Action	Comment
23 + S + D	1	Protocol	-	Ignore	
24 + S + D	2	Header Checksum	-	Ignore	
26 + S + D	4	Source IP Address	-	Ignore	
30 + S + D	4	Destination IP Address	IP4AT	Compare	May match any of four values in *IP4AT*.

5.6.3.1.7 Directed IPv6 Packet

The I350 supports receiving directed IPv6 packets for wake up if the *IPV6* bit is set in the *Wake Up Filter Control (WUFC)* register and Proxying if the *IPV6* bit is set in the *Proxying Filter Control* (*PROXYFC*) register.

One IPv6 address is supported and is programmed in the *IPv6 Address Table (IP6AT)*. A successfully matched packet must contain the station's MAC address, match VLAN tag if programmed, a Ethernet type of 0x86DD, and the programmed IPv6 address. In addition, the *IPAV.V60* bit should be set. The I350 also handles directed IPv6 packets that have VLAN tagging on both Ethernet II and Ethernet SNAP types.

Table 5-9 IPv6 Packet Structure and Processing

Offset	# of bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC header – processed by main address filter.
6	6	Source Address		Skip	
12	S=(0/4)	Possible VLAN Tag		Compare	Processed by main address filter.
12+ S	D=(0/8)	Possible Length + LLC/SNAP Header		Skip	
12 + S + D	2	Ethernet Type	0x86DD	Compare	IPv6
14 + S + D	1	Version/ Priority	0x6X	Compare	Check IPv6
15 + S + D	3	Flow Label	-	Ignore	
18 + S + D	2	Payload Length	-	Ignore	
20 + S + D	1	Next Header	-	Ignore	
21 + S + D	1	Hop Limit	-	Ignore	
22 + S + D	16	Source IP Address	-	Ignore	
38 + S + D	16	Destination IP Address	IP6AT	Compare	Match value in *IP6AT*.

5.6.3.1.8 NS and MLD IPv6 Packets

The I350 supports receiving:

1. IPv6 Neighbor Solicitation (NS) packets sent by a node to determine the link-layer address of a neighbor, or to verify that a neighbor is still reachable for wake up or Proxying.
2. IPV6 Multicast Listener Discovery (MLD) packets sent by an IPv6 router to discover the presence of multicast listeners (that is, nodes wishing to receive multicast packets) on its directly attached links, and to discover specifically which multicast addresses are of interest to those neighboring nodes.
3. Other ICMPv6 packets.

If the *NS* or *NS Directed* bits are set in the *Wake Up Filter Control (WUFC)* register, Wake up is executed on reception of the relevant ICMPv6 packets. Else if the *NS* or *NS Directed* bits are set in the *Proxying Filter Control* (*PROXYFC*) register the relevant ICMPv6 packets are sent to Firmware for Protocol offload.

- If the *NS directed* bit is set a successfully matched packet must contain the station's MAC address (Unicast or Multicast), match VLAN tag if programmed, a Ethernet type of 0x86DD, a IPv6 Header Type of ICMPv6 (0x3A), correct ICMPv6 Checksum and the single programmed IPv6 address in the *IPv6 Address Table* (*IP6AT*) must match the Target IPv6 Address in a NS packet or the Multicast Address field in a MLD packet. In addition, the *IPAV.V60* bit should be set.
- If the *NS* bit is set a successfully matched packet must contain the station's MAC address (Unicast or Multicast), match VLAN tag if programmed, a protocol type of 0x86DD, a IPv6 Header Type of ICMPv6 (0x3A) and a correct ICMPv6 Checksum. In this case all ICMPv6 packets are forwarded to Firmware.

The I350 also handles Neighbor Solicitation (NS) and Multicast Listener Discovery (MLD) IPv6 packets that have VLAN tagging on both Ethernet II and Ethernet SNAP types.

Table 5-10 Neighbor Solicitation (NS) and Multicast Listener Discovery (MLD) Packet Structure and Processing

Offset	# of bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC header – processed by main address filter.
6	6	Source Address		Skip	
12	S=(0/4)	Possible VLAN Tag		Compare	Processed by main address filter.
12+ S	D=(0/8)	Possible Length + LLC/SNAP Header		Skip	
12 + S + D	2	Ethernet Type	0x86DD	Compare	IPv6
14 + S + D	1	Version/ Priority	0x6X	Compare	Check IPv6
15 + S + D	3	Flow Label	-	Ignore	
18 + S + D	2	Payload Length	-	Ignore	
20 + S + D	1	Next Header	IPv6 next header types or 0x3A	Compare	58 decimal (0x3A) - ICMPv6 header type
21 + S + D	1	Hop Limit	-	Ignore	
22 + S + D	16	Source IP Address	-	Ignore	
38 + S + D	16	Destination IP Address	-	Ignore	
54+D+S	N	Possible IPv6 Next Headers	-	Ignore	
ICMPv6 header					
54+D+S+N	1	Type	-	Ignore	
55+D+S+N	1	Code	0x0	Ignore	
56+D+S+N	2	Checksum		Check	
58+D+S+N	4	Reserved	0x0	Ignore	
62+D+S+N	16	Target IP Address/ Multicast Address	IP6AT	Compare	Match value in *IP6AT* for *NS directed* Match. ***Note:*** Relevant for NS and MLD packets.
78+D+S+N	F	ICMPv6 Message Body		Ignore	

5.6.3.2 Flexible Filters

The I350 supports a total of 8 flexible filters. Each filter can be configured to recognize any arbitrary pattern within the first 128 bytes of the packet. To configure the flexible filters, software programs the mask values (required values and the minimum packet length), into the Flexible Host Filter Table (*FHFT* and *FHFT_EXT*). These 8 flexible filters contain separate values for each filter.

To enable Wake on LAN operation based on the Flex filters Software must also enable the filters in the *Wake Up Filter Control* (*WUFC*) register, and enable the overall wake up functionality. The overall wake up functionality must be enabled by setting *PME_En* bit in the *PMCSR* configuration register or the *PME_En* bit in the *Wake Up Control* (*WUC*) register.

To enable Proxying operation based on the Flex filters Software must also enable the filters in the *Proxying Filter Control* (*PROXYFC*) register, and enable the overall Proxying functionality by setting to 1b the *WUC.PPROXYE* bit.

Once enabled, the flexible filters scan incoming packets for a match. If the filter encounters any byte in the packet where the mask bit is one and the byte doesn't match the value programmed in the Flexible Host Filter Table (*FHFT* or *FHFT_EXT*), then the filter fails that packet. If the filter reaches the required length without failing the packet, it passes the packet and generates a wake-up event. It ignores any mask bits set to one beyond the required length.

Note: The flex filters are temporarily disabled when read from or written to by the host. Any packet received during a read or write operation is dropped. Filter operation resumes once the read or write access completes.

The following packets are listed for reference purposes only. The flexible filter could be used to filter these packets.

5.6.3.2.1 IPX Diagnostic Responder Request Packet

An IPX diagnostic responder request packet must contain a valid MAC address, a Ethernet type of 0x8137, and an IPX diagnostic socket of 0x0456. It might include LLC/SNAP headers and VLAN tags. Since filtering this packet relies on the flexible filters, which use offsets specified by the operating system directly, the operating system must account for the extra offset LLC/SNAP headers and VLAN tags.

Table 5-11 IPX Diagnostic Responder Request Packet Structure and Processing

Offset	# of bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	
6	6	Source Address		Skip	
12	S=(0/4)	Possible VLAN Tag		Skip	
12+ S	D=(0/8)	Possible Length + LLC/SNAP Header		Skip	
12 + S + D	2	Ethernet Type	0x8137	Compare	IPX
14 + S + D	16	Some IPX Stuff	-	Ignore	
30 + S + D	2	IPX Diagnostic Socket	0x0456	Compare	

5.6.3.2.2 Directed IPX Packet

A valid directed IPX packet contains the station's MAC address, a Ethernet type of 0x8137, and an IPX node address that is equal to the station's MAC address. It might include LLC/SNAP headers and VLAN tags. Since filtering this packet relies on the flexible filters, which use offsets specified by the operating system directly, the operating system must account for the extra offset LLC/SNAP headers and VLAN tags.

Table 5-12 IPX Packet Structure and Processing

Offset	# of bytes	Field	Value	Action	Comment
0	6	Destination Address		Compare	MAC header – processed by main address filter.
6	6	Source Address		Skip	
12	S=(0/4)	Possible VLAN Tag		Skip	
12+ S	D=(0/8)	Possible Length + LLC/SNAP Header		Skip	
12 + S + D	2	Ethernet Type	0x8137	Compare	IPX
14 + S + D	10	Some IPX Info	-	Ignore	
24 + S + D	6	IPX Node Address	Receive Address 0	Compare	Must match receive address 0.

5.6.3.2.3 Utilizing Flex Wake-Up Filters In Normal Operation

The I350 enables utilizing the WoL Flex filters in normal operation, when in D0 power management state, for queuing decisions. Further information can be found in Section 7.1.2.6.

5.6.3.3 Wake Up Packet Storage

The I350 saves the first 128 bytes of the wake-up packet in its internal buffer, which can be read through the *Wake Up Packet Memory (WUPM)* register after the system wakes up.

5.6.4 Wake-up and Virtualization

When operating in a virtualized environment, all wake-up capabilities are managed by a single entity (such as the VMM or an IOVM). In an IOV architecture, the physical (PF) driver controls wake-up and none of the Virtual Machines (VMs) has direct access to the wake-up registers. The wake-up registers are not replicated per VF.

5.7 Protocol Offload (Proxying)

In order to avoid spurious wake up events and reduce system power consumption when the device is in D3 low power state and system is in S3 or S4 low power states the I350 supports protocol offload (Proxying) of:

1. A single IPv4 ARP (Address Resolution Protocol) request per function.
   - Responds to IPv4 address resolution request with the Host MAC (L2) address (as defined in RFC 826).
2. Two IPv6 NS (Neighbor Solicitation) requests per function, where each NS protocol offload request includes 2 IPv6 addresses, for a total of 4 possible IPv6 addresses per function.

  - — Responds to IPv6 Neighbor Solicitation requests with the Host MAC (L2) address (as defined in RFC 4861).
3. When NS (Neighbor Solicitation) Protocol offload is enabled the I350 supports up to two IPv6 Multicast-Address-Specific MLD (Multicast Listener Discovery) Queries (either MLDv1 or MLDv2) per function. In addition the I350 will also respond to General MLD Queries, used to learn which IPv6 multicast addresses have listeners on an attached link.
   - — MLD protocol offload is supported when NS protocol offload is enabled so that IPv6 routers will discover the presence of multicast listeners (that is, nodes wishing to receive multicast packets), for packets with the IPv6 NS "solicited-node multicast address" and continue forwarding these Neighbor Solicitation (NS) requests on the link.
   - — MLD Protocol offload is supported for either "MLD Multicast Listener Query packets" or "MLD Multicast Address and Source Specific Query" packets that check for IPv6 Multicast Listeners with the "solicited-node multicast address" placed in the IPv6 destination address field of the IPv6 Neighbor Solicitation (NS) packets that are off-loaded bythe I350.
   - — Responds to the IPv6 MLD (Multicast Listener Discovery) Queries, with the "solicited-node multicast address" placed in the IPv6 destination address field of the IPv6 Neighbor Solicitation (NS) packets that are off-loaded bythe I350 (as defined in RFC 2710 and RFC 3810).

In addition when the *PROXYFC.D0_PROXY* bit is set to 1b, the I350 can support protocol offload when the device is not in the D3 low power state. Enabling this feature will allow system to be in low power S0ix state for longer durations and increase system power saving.

5.7.1 Proxying and Virtualization

When operating in a virtualized environment, all proxying capabilities are managed by a single entity (such as the VMM or an IOVM). In an IOV architecture, the physical (PF) driver controls Proxying and none of the Virtual Machines (VMs) has direct access to the Proxying registers. The Proxying registers are not replicated per VF.

5.7.2 Protocol Offload Activation in D3

To enable Protocol Offload software device driver should implement the following steps before D3 entry:

1. Read *MANC.MPROXYE* bit to verify that proxying is supported by management.
2. Clear all pending proxy status bits in the Proxying Status (*PROXYS*) register.
3. Program the Proxying Filter Control (*PROXYFC*) register to indicate type of packets that should be forwarded to manageability for proxying and program the necessary data to the IPv4/v6 Address Table (*IP4AT, IP6AT*) and the Flexible Host Filter Table (*FHFT*) registers.
4. Set the *WUFC.FW_RST_WK* bit to 1b to initiate a wake if firmware reset was issued when in D3 state and proxying information was lost.
5. Take ownership of the Management Host interface semaphore (*SW_FW_SYNC.SW_MNG_SM* register bit) using the flow defined in Section 4.7.1 to send Protocol Offload information to Firmware.
6. Read and clear the *FWSTS.FWRI* firmware reset indication bit.
   - — If a firmware reset was issued as reported in the *FWSTS.FWRI* bit software device driver should clear the bit and then re-initialize the Protocol Offload list even if previously Firmware was configured to keep protocol Offload list on move from D3 to D0 (See *Set Firmware Proxying Configuration Command* in Section 10.8.2.4.2.2).

7. Verify that the *HICR.En* bit (See Section 8.22.1.2) is set 1b which indicates that the Shared RAM interface is available.
8. Write proxying information in the shared RAM interface located in addresses 0x8800-0x8EFF using the format defined in Section 10.8.2.4.2. All addresses should be placed in Networking order.
9. Once information is written into the shared RAM software should set the *HICR.C* bit to 1b.
10. Poll the *HICR.C* bit until bit is cleared by Firmware indicating that command was processed and verify that command completed successfully by checking that *HICR.SV* bit was set.
11. Read Firmware response from the Shared RAM to verify that data was received correctly.
12. Return to 8. if additional commands need to be sent to Firmware.
13. Release management Host interface semaphore (*SW_FW_SYNC.SW_MNG_SM* register bit) using the flow defined in Section 4.7.2.
14. Verify that a Firmware reset was not initiated during the Proxying configuration process by reading the *FWSTS.FWRI* firmware reset indication bit. If a firmware reset was initiated Return to 1.
15. Set *WUC.PPROXYE* bit to 1b and enable entry into D3 low power state.
16. Once the I350 moves back into D0 state, the Software Device driver needs to clear the *WUC.PPROXYE* bit, *PROXYS* and *PROXYFC* registers until the next time the system moves to a low power state with proxying enabled.
    — On transition from D3 to D0 Firmware may either delete all proxying requirements or not depending on the configuration defined by Software driver via the *Set Firmware Proxying Configuration Command* using the Shared RAM interface (See Section 10.8.2.4.2.2).

Normally, after enabling wake-up or proxying, system software moves the device to D3 low power state by writing a 11b to the PCI *PMCSR.Power State* field.

Once proxying is enabled by setting the *WUC.PPROXYE* bit to 1b and device is placed in the D3 low power state, the I350 monitors incoming packets, first filtering them according to its standard address filtering method, then filtering them with all of the proxying filters enabled in the *PROXYFC* register. If a packet passes both the standard address filtering and at least one of the enabled proxying filters and does not pass any of the enabled wake-up filters, the I350 will:

1. Execute the relevant Protocol offload for the packet and not forward the packet to the Host.
2. Set one or more bits in the *Proxying Status (PROXYS)* register according to the Proxying filters matched.
    — Note that the I350 sets more than one bit in the *PROXYS* register if a packet matches more than one filter.
3. Will wake the system and forward a packet that matches the proxying filters but can't be supported to the Host for further processing if configured to do so by Software driver via the *Set Firmware Proxying Configuration Command* using the Shared RAM interface (See Section 10.8.2.4.2.2).

Notes:

1. When device is in D3 a packet that matches both one of the enabled Proxying filters as defined in the *PROXYFC* register and one of the enabled Wake-up filters as defined in the *WUFC* register will only wake-up the system and protocol offload (Proxying) will not occur.
2. Protocol Offload is not executed for illegal packets with CRC errors or Checksum errors and the packets are silently discarded.
3. Once a packet that meets the criteria for Proxying is received the I350 should respond to the request after less than 60 Seconds.

5.7.3 Protocol Offload Activation in D0

To enable Protocol Offload in D0 software device driver should implement the following steps:

1. Read *MANC.MPROXYE* bit to verify that proxying is supported by management.
2. Clear all pending proxy status bits in the Proxying Status (*PROXYS*) register.
3. Program the Proxying Filter Control (*PROXYFC*) register to indicate type of packets that should be forwarded to manageability for proxying and program the necessary data to the IPv4/v6 Address Table (*IP4AT*, *IP6AT*) and the Flexible Host Filter Table (*FHFT*) registers.
4. Take ownership of the Management Host interface semaphore (*SW_FW_SYNC.SW_MNG_SM* register bit) using the flow defined in Section 4.7.1 to send Protocol Offload information to Firmware.
5. Verify that the *HICR.En* bit (See Section 8.22.1.2) is set 1b which indicates that the Shared RAM interface is available.
6. Read and clear the *FWSTS.FWRI* firmware reset indication bit.
   — If a firmware reset was issued as reported in the *FWSTS.FWRI* bit software device driver should clear the bit and then re-initialize the Protocol Offload list even if previously Firmware was configured to keep protocol Offload list on move from D3 to D0 (See *Set Firmware Proxying Configuration Command* in Section 10.8.2.4.2.2).
7. Write proxying information in shared RAM interface located in addresses 0x8800-0x8EFF using the format defined in Section 10.8.2.4.2. All addresses should be placed in Networking order.
8. Once information is written into the shared RAM software should set the *HICR.C* bit to 1b.
9. Poll the *HICR.C* bit until bit is cleared by Firmware indicating that command was processed and verify that command completed successfully by checking that *HICR.SV* bit was set.
10. Read Firmware response from the Shared RAM to verify that data was received correctly.
11. Return to 9. if additional commands need to be sent to Firmware.
12. Release management Host interface semaphore (*SW_FW_SYNC.SW_MNG_SM* register bit) using the flow defined in Section 4.7.2.
13. Verify that a Firmware reset was not initiated during the Proxying configuration process by reading the *FWSTS.FWRI* firmware reset indication bit. If a firmware reset was initiated Return to 1.
14. Set the *PROXYFC.D0_PROXY* bit to 1b.
15. Set *WUC.PPROXYE* bit to 1b to enable protocol offload.

Once proxying is enabled in D0 by setting both the *WUC.PPROXYE* bit to 1b and the *PROXYFC.D0_PROXY* bit to 1b, the I350 monitors incoming packets, first filtering them according to the standard address filtering method and then filtering them according to the proxying filters enabled in the *PROXYFC* register. If a packet passes both the standard address filtering and at least one of the enabled proxying filters then the I350 will:

1. Execute the relevant Protocol offload for the packet and not forward the packet to the Host.
2. Set one or more bits in the *Proxying Status (PROXYS)* register according to the Proxying filter that detected a match.
   — Note that the I350 sets more than one bit in the *PROXYS* register if a packet matches more than one filter.
3. Discard silently illegal packets with CRC errors or Checksum errors without implementing the Protocol offload.
4. Forward a packet that matches the proxying filters but can't be supported by Firmware to the Host for further processing, if configured to do so by Software driver via the *Set Firmware Proxying Configuration Command* using the Shared RAM interface (See Section 10.8.2.4.2.2).

5.8 DMA Coalescing

The I350 supports DMA Coalescing to enable synchronizing port activity and optimize power management of memory, CPU and RC internal circuitry. When conditions to enter DMA coalescing operating mode as defined in Section 5.8.2 exist, the I350 will:

- Stop initiation of any activity on the PCIe link.
- Data received from the Ethernet Link is buffered in internal Receive buffer.
- When executing DMA coalescing, once internal TX buffer is empty, the internal RX buffer watermark for transmission of XOFF flow control packets on the network is defined by the *FCRTC.RTH_Coal* threshold field.

The I350 will exit DMA coalescing once the conditions defined in Section 5.8.3, to exit DMA coalescing, exist.

5.8.1 DMA Coalescing Activation

To activate DMA coalescing functionality software driver should program the following fields:

1. *DMACR.DMACTHR* field to set the receive threshold that causes move out of DMA Coalescing operating mode. Receive watermark programmed should take into account latency tolerance reported (See Section 5.9) and L1 to L0 latency to avoid Receive Buffer overflow when DMA Coalescing is enabled.
2. *DMCTXTH.DMCTTHR* field to set transmit threshold that causes move out of DMA Coalescing operating mode. Transmit watermark programmed should take into account latency tolerance reported (See Section 5.9) and L1 to L0 latency to allow transmission of back to back packets when DMA Coalescing is enabled.
3. *DMACR.DMACWT* field that defines a maximum timeout value for:
   a. A receive packet to be stored in the internal receive buffer before the I350 moves packet to Host memory.
   b. Fetch the transmit descriptors and transmit data from Host memory once on-chip transmit tail pointer was updated.
   c. Time to delay an interrupt that is not defined as an immediate interrupt in the *IMIR[n]*, *IMIREXT[n]* or *IMIRVP* registers, when other conditions specified in Section 5.8.3 to exit DMA coalescing do not exist.

   — Each time the I350 enters DMA coalescing, internal DMA coalescing watchdog timer is re-armed with the value placed in the *DMACR.DMACWT* field. When in DMA coalescing the internal watchdog timer starts to count when one of the following conditions occurs:
   - A RX packet is received in the Internal buffer.
   - An Interrupt is pending.
   - A descriptor write-back is pending.
   - On-chip transmit tail pointer was updated.

   Once interval defined in the *DMACR.DMACWT* field has passed, the I350 exits DMA coalescing and internal buffers, pending interrupts and pending descriptor write-backs are flushed.

   — The *DMACR.DC_LPBKW_EN* and *DMACR.DC_BMC2OSW_EN* bits define if a VM to VM loopback traffic or a BMC to OS traffic is delayed by the time defined in the *DMACR.DMACWT* field when the I350 is in DMA coalescing state or if the traffic causes immediate exit out of DMA coalescing.
4. *DMCTLX.*TTLX timer field to define:

a. The time between detection of DMA idle status to actual move into low power link state (L0s or L1). Value programmed in this register reduces amount of entries into low power PCIe link state when traffic rate is high.
   - Even when DMA coalescing is disabled (*DMACR.DMAC_EN* = 0) entry into low power link state when no DMA activity is expected, will be delayed according to value placed in *DMCTLX.TTLX* field.

b. The time between detection of DMA idle condition and entry into DMA coalescing state. To limit entry into DMA coalescing state when packet rate is high.

5. *DMCTLX.DCFLUSH_DIS* to define if pending descriptor write-back flush and pending interrupt flush should occur before entry into DMA coalescing state.
   — When *DMCTLX.DCFLUSH_DIS* is set to 1b any pending interrupts or descriptor write-back operations do not cause the I350 to move out of DMA coalescing state.
6. *DMCTLX.DC_FLUSH* to define when pending descriptor write-back flush, pending interrupt flush, entry into DMA coalescing state and moving PCIe link into low power Lx state occurs relative to the *DMCTLX.TTLX* timer activation and expiration (See Section 8.24.1 for further information).
7. *DMCRTRH.UTRESH* low rate threshold field to define a minimum RX data rate. Below this data rate the I350 does not enter DMA coalescing operating mode if value of field is greater than 0.
   a. This field prevents moving into DMA coalescing operating mode in current time window when traffic in the previous time window is sparse and effectiveness of DMA coalescing on system power saving is limited. After reception is enabled, if value of this field is greater than 0, the I350 does not enter DMA coalescing operating mode until duration of at least one time window has passed, to allow for collection of enough statistical data on receive data rate.
   b. The time window to measure data rate is defined according to the port link rate (10Mbps, 100Mbps or 1Gbps), *SCCRL.INTERVAL* field and the *SCBI.BI* field. The Port link rate and the *SCBI.BI* value define a basic interval that's multiplied by the *SCCRL.INTERVAL* to define the time window.
      For example to disable entry into DMA coalescing operating mode so long as data rate is below 1 Mbyte/s when port speed is 1Gbps the following values can be programmed:
      - *SCBI.BI* = 0xC35000 (defines basic interval of 2.048 msec at 1Gbps link rate).
      - *SCCRL.INTERVAL* = 0x3E8 (The *SCCRL.INTERVAL* decimal value of 1,000 multiplied by the 2.048 msec basic interval defined in the *SCBI.BI* field equals a time window of 2.048 seconds).
      - *DMCRTRH.UTRESH* = 0x7D00 (The *DMCRTRH.UTRESH d*ecimal value of 32,000 multiplied by 64 Byte chunks equals 2,048,000 Bytes of data. This defines the minimum amount of data to be received in the 2.048 second time window before DMA coalescing is activated in the next time window. As a result at least 1 Mbyte per second data rate needs to be received before entry into DMA coalescing operating mode is enabled).
   c. The time window is also used for Storm Control, see Section 7.8.3.8.4.2 for further information on how to set the time window.
8. *FCRTC.RTH_Coal* field that defines a flow control receive high watermark for sending flow control packets. The I350 uses the *FCRTC.RTH_Coal* threshold when:
   — Flow control is enabled by setting the *CTRL.TFCE* bit.
   — The I350 is in DMA Coalescing mode.
   — Internal transmit buffer is empty.
9. *SRRCTL[n].DMACQ_Dis* bit to define high priority queues. When a received packet is forwarded to a queue with the *SRRCTL[n].DMACQ_Dis* bit set, the I350 moves immediately out of DMA Coalescing mode and executes a DMA operation to store the packet in host memory.
10. *DMACR.DMAC_EN* bit should be set to 1 to enable activation of DMA Coalescing operating mode.
11. *DMACR.DMAC_Lx* to 10b or 11b to define that low power L1 PCIe ASPM link state is entered when in DMA Coalescing operating mode.

Notes:

1. The values of *DMACR.DMACTHR* and *FCRTC.RTH_Coal* should be set so that XOFF packet generation is avoided. In DMA Coalescing mode, when transmit buffer is empty, the XOFF flow control threshold (*FCRTC.RTH_Coal*) value can be increased by maximum jumbo frame size compared to normal operation, where high threshold is set by the *FCRTH0* register.
2. When entering DMA coalescing mode, the value written in the *FCRTH0* register is used to generate XOFF flow control frames until the internal transmit buffer is empty. Once the internal transmit buffer is empty the value written in the *FCRTC.RTH_Coal* field is used as a watermark for generation of XOFF flow control frames.
3. When *PCIEMISC.Lx_decision* bit is set to 1, the I350 transitions the PCIe link into low power state as defined in the *DMACR.DMAC_Lx* field once it detects no PCIe activity for a duration defined in the *DMCTLX.TTLX* field. However, when entry to the L0s state is enabled in the PCIe configuration *Link Control Register* (See Section 9.5.6.8), the I350 will always enter the L0s state after the duration defined in the *Latency_To_Enter_L0s* field (if not already in L1, L2 or L3 low power link state) to comply with the PCIe specification.

5.8.2 Entering DMA Coalescing Operating Mode

Enabling DMA Coalescing operation by setting the *DMACR.DMAC_EN* bit to 1, enables alignment of bus master traffic and interrupts from all ports. Power saving is achieved since synchronizing PCIe accesses between ports increases the occurrence of idle intervals on the PCIe bus and also increases the duration of these idle intervals. Power Management Unit on platform can utilize these Idle intervals to reduce system power.

5.8.2.1 Entering DMA Coalescing

The I350 will enter DMA coalescing when all of the following conditions exist:

1. DMA Coalescing is enabled (*DMACR.DMAC_EN* = 1).
2. Internal Receive buffers are empty.
3. There are no pending DMA operations.
4. None of the conditions defined in Section 5.8.3.1 to move out of DMA Coalescing exist.

Before entering the DMA coalescing power saving mode, if the *DMCTLX.DCFLUSH_DIS* bit is programmed to 0, the I350 will:

- Flush all pending interrupts that were delayed due to the Interrupt Throttling (ITR) mechanism.
- The I350 will flush all pending Receive descriptor and Transmit descriptor write backs and pre-fetch available Receive descriptors and Transmit descriptors to internal cache.

Note: Timing of pending interrupts and Pending descriptor write-back flush operation relative to the expiration of the *DMCTLX.TTLX* timer is defined by the *DMCTLX.DC_FLUSH* bit (See Section 8.24.3).

5.8.3 Conditions to Exit DMA Coalescing

5.8.3.1 Exiting DMA Coalescing

When the I350 is in DMA Coalescing operating mode, DMA Coalescing mode is exited when one of the following events occurs:

1. Amount of data in internal receive buffer passed the *DMACR.DMACTHR* threshold.
2. Empty space in internal transmit buffer is above the value defined in *DMCTXTH.DMCTTHR* field and available transmit descriptors exist.
3. A high priority packet was received (See Section 7.3.6 for definition of high priority packets). A high priority packet is a packet that generates an immediate interrupt, as defined in the *IMIR[n]*, *IMIREXT[n]* or *IMIRVP* registers.
4. A received packet destined to a high priority queue (*SRRCTL[n].DMACQ_Dis* =1b) was detected.
5. DMA coalescing Watchdog timer defined in the *DMACR.DMACWT* field expires as a result of the following occurrences not being serviced for the duration defined in the *DMACR.DMACWT* field:
   — A RX packet was received in the Internal buffer.
   — An Interrupt is pending.
   — A descriptor write-back is pending.
   — On-chip transmit tail pointer was updated.
6. Received data rate detected is lower than defined in the *DMCRTRH.UTRESH* field.
7. DMA Coalescing is disabled (*DMACR.DMAC_EN* = 0).
8. Another the I350 function initiated a transaction on the PCIe bus.
9. Updated PCIe LTR message with reduced latency tolerance value needs to be sent (See Section 5.9 for additional information on LTR).
   — The I350 does not exit DMA coalescing to send a PCIe LTR message with increased latency tolerance.
10. Software initiates move out of DMA Coalescing by writing 1b to the *DMACR.EXIT_DC* self clearing bit.
11. VM to VM loopback traffic if the *DMACR.DC_LPBKW_EN* bit is programmed to 0b.
12. BMC to OS traffic if the *DMACR.DC_BMC2OSW_EN* bit is programmed to 0b.

Notes:

1. Even when conditions for DMA Coalescing do not exist, the I350 will continue to be in low power PCIe link state (L0s or L1) if there is no requirement for PCIe access.
2. If PCIe PME wake message needs to be sent, PCIe link will move from L1 low power state to L0 to send the message but DMA will remain in DMA coalescing state.
3. Pending interrupts or pending descriptor write-back operations do not cause the I350 to move out of DMA coalescing state.

5.9 Latency Tolerance Reporting (LTR)

The I350 generates PCIe LTR messages to report service latency requirements for memory reads and writes to the Root Complex for system power management.

The I350 will report either minimum latency tolerance, maximum latency tolerance or no latency tolerance requirements as a function of link, LAN port and function status. Minimum and maximum latency tolerance values are programmed in the *LTRMINV* and *LTRMAXV* registers respectively per PF by the software driver to optimize power consumption without incurring packet loss due to receive buffer overflow.

5.9.1 Latency Tolerance Reporting Algorithm

The I350 sends LTR messages according to the following algorithm when the capability is enabled in the Latency Tolerance Reporting (LTR) Capability structure of Function 0 located in PCIe configuration space:

1. When Links on all ports are disconnected or all LAN ports are disabled (transmit and receive activity not enabled) and the *LTRC.LNKDLS_EN* and *LTRC.PDLS_EN* bits are set respectively, the I350 will send a LTR PCIe message with LTR Requirement bits cleared, to indicate that no Latency tolerance requirements exists.
2. If the I350 reported following PCIe link-up latency tolerance requirements with any requirement bit set in the PCIe LTR message and all enabled functions were placed in D3 low power state via the *PMCSR* register, the I350 will send a new LTR Message with all the Requirement bits clear.
3. If the I350 reported following PCIe link-up latency tolerance requirements with any requirement bit set and the LTR Mechanism Enable bit in the PCIe configuration space is cleared, the I350 will send a new LTR Message with all the Requirement bits clear.
4. The I350 will send a LTR message with the value placed in the *LTRMAXV* register when either one of following conditions exist:
   a. Software set the *LTRC.LTR_MAX* register bit.
   b. RX EEE LPI state is detected on the Ethernet Link and *LTRC.EEEMS_EN* is set (See Section 3.7.7.4 for additional information).
5. Otherwise, the I350 will send a LTR message with a minimum value.

Note: In all cases maximum LTR Value sent by the I350 does not exceed the maximum latency values in the *Max No-Snoop Latency* and *Max Snoop Latency* Registers in the Latency Tolerance Reporting (LTR) Capability structure of Function 0.

Figure 5-7 describes the I350 LTR message generation flow.

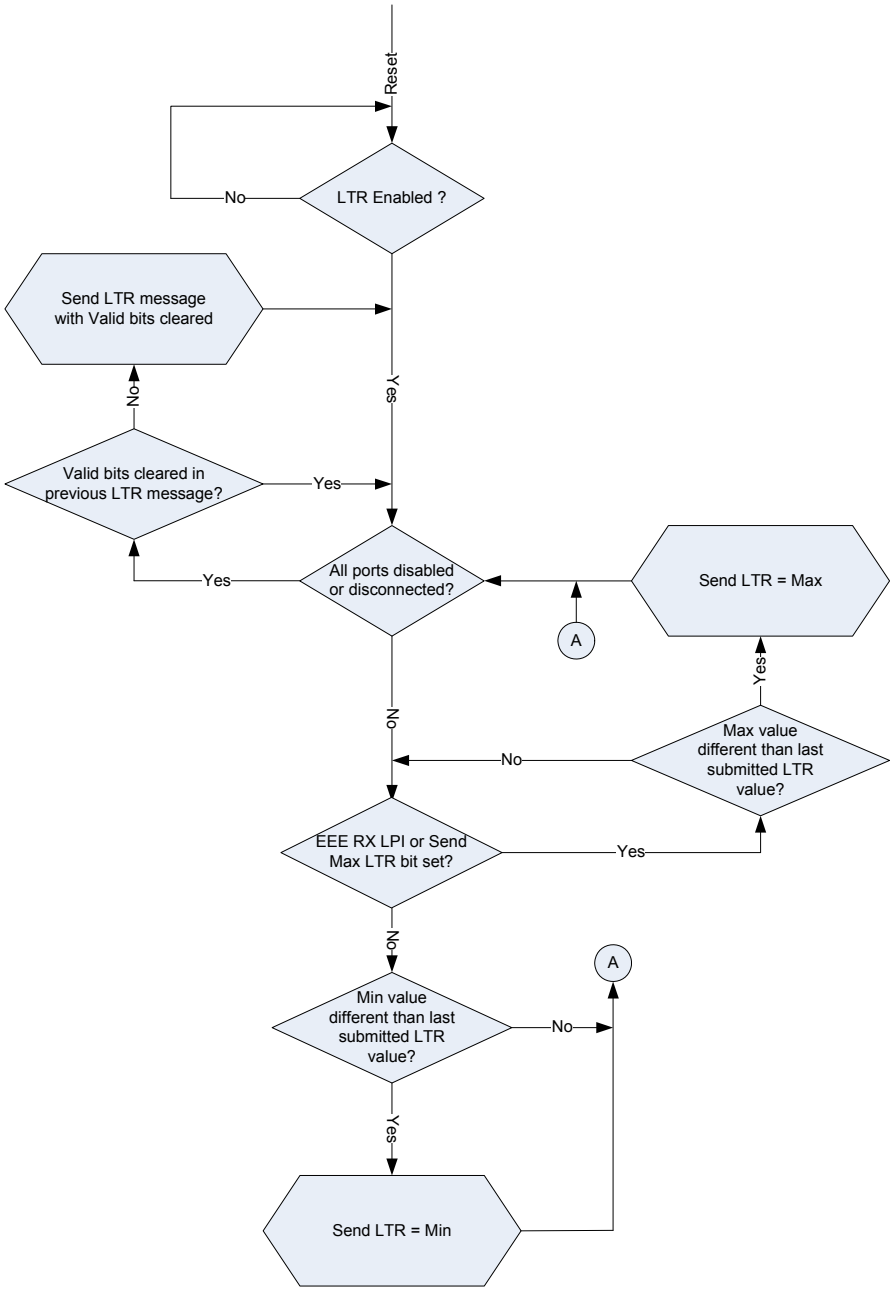


Figure 5-7 PCIe LTR Message Generation Flow per Function

5.9.2 Latency Tolerance Reporting Per Function

Internal to the I350 each function can request to generate a Minimum value LTR, a Maximum value LTR and a LTR message with the Requirement bits cleared. The I350 conglomerates latency requirements from the functions that have LTR messaging enabled and sends a single LTR message in the following manner:

- The acceptable latency values for the message sent upstream by the I350 must reflect the lowest latency tolerance values associated with any function.
  - It is permitted that the Snooped and Non-Snooped values reported in the conglomerated message are associated with different functions.
  - If none of the functions have a Latency requirement for a certain type of traffic (Snoop/Non-snoop), the message sent by the I350 will not have the Requirement bit corresponding to that type of traffic set.
- The I350 transmits a new LTR message upstream when the capability is enabled and when any function changes the values it has reported internally in such a way as to change the conglomerated value reported previously by the I350.

Each function in the I350 reports support of LTR messaging in the configuration space by:

- Setting the *LTR Mechanism Supported* bit in the PCIe *Device Capabilities 2* configuration register (Support defined by *LTR_EN* bit in *Initialization Control Word 1* EEPROM word, that controls enabling of the LTR structures).
- Supporting the *Latency Tolerance Requirement Reporting (LTR) Capability* structure in the PCIe configuration space.

To enable generation of LTR messages by a function the *LTR Mechanism Enable* bit in the *Device Control 2* configuration register of Function 0 should be set.

Note: A function that does not have LTR messaging enabled is considered a function that does not have any Latency Tolerance requirements.

5.9.2.1 Conditions for Generating LTR Message with the Requirement Bits Cleared

When LTR messaging is enabled the I350 functions will send a LTR message with the Requirement bits cleared in the following cases:

1. Following PE_RST_N assertion (PCIe reset) after LTR capability is enabled.
2. LAN port is disabled (both *RCTL.RXEN* and *TCTL.EN* are cleared), receive buffer is empty and *LTRC.PDLS_EN* is set.
3. LAN port is disconnected, BMC to Host traffic is disabled (*MANC.EN_BMC2HOST* = 0) and *LTRC.LNKDLS_EN* is set.
4. Function is not in D0a state.
5. When the LSNP and LNSNP bits are cleared in the *LTRMINV* register and minimum LTR value needs to be sent.
6. When the LSNP and LNSNP bits are cleared in the *LTRMAXV* register and maximum LTR value needs to be sent.
7. When the *LTR Mechanism Enable* bit in the *Device Control 2* configuration register of Function 0 was cleared and the I350 sent previously a LTR message with Requirement bits set.

When one of the above conditions exist in all functions that are enabled, the I350 will send a LTR message with the requirement bits cleared.

Note: A disabled function does not generate Latency Tolerance requirements.

5.9.2.2 Conditions for Generating LTR Message with Maximum LTR Value

When LTR messaging is enabled and conditions to send LTR Message with Valid bits cleared do not exist, the I350 functions will send a maximum value LTR message, with the values programmed in the *LTRMAXV* register in the following cases:

1. Following a software write 1 operation to *LTRC.LTR_MAX* bit and the last PCIe LTR message sent had a latency tolerance value different then the value specified in the *LTRMAX* register.
2. RX EEE LPI state is detected on the Ethernet Link and *LTRC.EEEMS_EN* is set.
3. When updated data was written to the *LTRMAXV* register and conditions defined in 1. or 2. to send a LTR message with a maximum value exists.

When one of the above conditions exist in all functions that are enabled and at least in one enabled function conditions to send a LTR message with requirement bits cleared (See Section 5.9.2.1) doesn't exist, the I350 will send a LTR message with the values programmed in the *LTRMAXV* register.

Note: When the *LTRC.LTR_MAX* bit is cleared, the I350 will send a LTR message with the value placed in the *LTRMINV* register, if the value is smaller than the value placed in the *LTRMAXV* register.

5.9.2.3 Conditions for Generating LTR Message with Minimum LTR Value

When LTR messaging is enabled, the I350 functions will send a minimum value LTR message, with the values programmed in the *LTRMINV* register in the following cases:

1. Following a software write 1 operation to the *LTRC.LTR_MIN* bit and the last PCIe LTR message sent had a latency tolerance value different then the value specified in the *LTRMINV* register.
2. When updated data was written to the *LTRMINV* register and conditions to send LTR message with the Requirement bits cleared (See Section 5.9.2.1) or Maximum value LTR (See Section 5.9.2.2) do not exist.

Note: If a LTR message that indicates that best possible service is requested needs to be sent, the *Latency Tolerance value* in the *LTRMINV* and *LTRMAXV* registers should be programmed to 0 with the appropriate *requirement* bits set. In this case the I350 will send a LTR message with both the value and scale fields cleared to 0's.

NOTE: ***This page intentionally left blank.***

6 Non-Volatile Memory Map - EEPROM

6.1 EEPROM General Map

The I350 EEPROM is partitioned into 4 main blocks followed by Firmware and PXE structures. First 128 word section is allocated to words common to all LAN ports, Firmware, Software, PXE and LAN port 0. Following 64 word sections are allocated to Lan port 1, Lan port 2 and Lan port 3 as shown in Table 6-1.

A detailed list of EEPROM words loaded by Hardware following Power-up, Hardware reset or software generated resets (*CTRL.RST*, *CTRL_EXT.EE_RST* or *CTRL.DEV_RST*) can be found in the auto load sequence table in Section 3.3.1.3.

Table 6-1 EEPROM Top Level Partitioning

EEPROM Word Offsets	Partition
0x00 to 0x7F	Common Words (PCIe, PXE, SW and FW) and LAN port 0 words - see Table 6-2
0x80 to 0xBF	LAN Port 1 words - see Table 6-3
0xC0 to 0xFF	LAN Port 2 words - see Table 6-3
0x100 to 0x13F	LAN Port 3 words - see Table 6-3
0x140...	Option ROM and Firmware Structures

Table 6-2 lists the I350 EEPROM word map for Common words and Lan Port 0.

Table 6-2 Common and LAN Port 0 EEPROM Map

EEPROM Word Offsets	Used By/In	High Byte	Low Byte	Which LAN
0x00:0x02	HW	Section 6.2.1, Ethernet Address (LAN Base Address + Offsets 0x00-0x02)		LAN 0
0x03	SW	Compatibility Word - Section 6.4.1		All
0x04	SW	Port Identification LED blinking - Section 6.4.2		All
0x05	SW	Section 6.4.3, EEPROM Image Revision (Word 0x05)		All
0x06	SW	OEM specific - Section 6.4.4		All
0x07	SW			All
0x08	SW	Section 6.4.5, PBA Number/Pointer (Word 0x08, 0x09)		All
0x09	SW			All
0x0A	HW	Section 6.2.2, Initialization Control Word 1 (word 0x0A)		All
0x0B	HW	Section 6.2.3, Subsystem ID (Word 0x0B)		All
0x0C	HW	Section 6.2.4, Subsystem Vendor ID (Word 0x0C)		All
0x0D	HW	Section 6.2.5, Device ID (LAN Base Address + Offset 0x0D)		LAN 0
0x0E	HW	Section 6.2.6, Vendor ID (Word 0x0E)		All
0x0F	HW	Section 6.2.8, Initialization Control Word 2 (LAN Base Address + Offset 0x0F)		LAN0

Table 6-2 Common and LAN Port 0 EEPROM Map (Continued)

EEPROM Word Offsets	Used By/In	High Byte	Low Byte	Which LAN
0x10	HW	Section 6.3.5, PCIe PHY Auto Configuration Pointer (Word 0x10)		All
0x11	HW (MNG HW)	Section 6.3.6, Management Pass Through LAN Configuration Pointer (LAN Base Address + Offset 0x11)		LAN 0 (MNG HW)
0x12	HW	Section 6.2.9, EEPROM Sizing and Protected Fields (Word 0x12)		All
0x13	HW	Section 6.2.10, Initialization Control 4 (LAN Base Address + Offset 0x13)		LAN 0
0x14	HW	Section 6.2.11, PCIe L1 Exit latencies (Word 0x14)		All
0x15	HW	Section 6.2.12, PCIe Completion Timeout Configuration (Word 0x15)		All
0x16	HW	Section 6.2.13, MSI-X Configuration (LAN Base Address + Offset 0x16)		LAN 0
0x17	HW	Section 6.3.1, Software Reset CSR Auto Configuration Pointer (LAN Base Address + Offset 0x17)		LAN 0
0x18	HW	Section 6.2.14, PCIe Init Configuration 1 (Word 0x18)		All
0x1B	HW	Section 6.2.17, PCIe Control 1 (Word 0x1B)		All
0x1C	HW	Section 6.2.18, LED 1,3 Configuration Defaults (LAN Base Address + Offset 0x1C)		LAN 0
0x1D	HW	Section 6.2.7, Dummy Device ID (Word 0x1D)		All
0x1E	HW	Section 6.2.19, Device Rev ID (Word 0x1E)		All
0x1F	HW	Section 6.2.20, LED 0,2 Configuration Defaults (LAN Base Address + Offset 0x1F)		LAN 0
0x20	HW	Section 6.2.21, Software Defined Pins Control (LAN Base Address + Offset 0x20)		LAN 0
0x21	HW	Section 6.2.22, Functions Control (Word 0x21)		All
0x22	HW	Section 6.2.23, LAN Power Consumption (Word 0x22)		All
0x23	HW	Section 6.3.3, PCIe Reset CSR Auto Configuration Pointer (LAN Base Address + Offset 0x23)		LAN 0
0x24	HW	Section 6.2.24, Initialization Control 3 (LAN Base Address + Offset 0x24)		LAN 0
0x25	HW	Section 6.2.25, I/O Virtualization (IOV) Control (Word 0x25)		All
0x26	HW	Section 6.2.26, IOV Device ID (Word 0x26)		All
0x27	HW	Section 6.3.4, CSR Auto Configuration Power-Up Pointer (LAN Base Address + Offset 0x27)		LAN0
0x28	HW	Section 6.2.27, PCIe Control 2 (Word 0x28)		All
0x29	HW	Section 6.2.28, PCIe Control 3 (Word 0x29)		All
0x2A	HW	Reserved		All
0x2B	HW	Reserved		All
0x2C	HW	Reserved		All
0x2D	HW	Section 6.2.30, Start of RO Area (Word 0x2D)		All
0x2E	HW	Section 6.2.31, Watchdog Configuration (Word 0x2E)		All
0x2F	OEM	Section 6.2.32, VPD Pointer (Word 0x2F)		
0x30	PXE	Section 6.4.6.1, Setup Options PCI Function 0 (Word 0x30)		
0x31	PXE	Section 6.4.6.2, Configuration Customization Options PCI Function 0 (Word 0x31)		
0x32	PXE	Section 6.4.6.3, PXE Version (Word 0x32)		
0x33	PXE	Section 6.4.6.4, Flash (Option ROM) Capabilities (Word 0x33)		
0x34	PXE	Section 6.4.6.5, Setup Options PCI Function 1 (Word 0x34)		
0x35	PXE	Section 6.4.6.6, Configuration Customization Options PCI Function 1 (Word 0x35)		
0x36	PXE	Section 6.4.7.1, iSCSI Option ROM Version (Word 0x36)		
0x37	PXE	Section 6.4.8, Alternate MAC address pointer (Word 0x37)		

Table 6-2 Common and LAN Port 0 EEPROM Map (Continued)

EEPROM Word Offsets	Used By/In	High Byte	Low Byte	Which LAN
0x38	PXE	Section 6.4.6.7, Setup Options PCI Function 2 (Word 0x38)		
0x39	PXE	Section 6.4.6.8, Configuration Customization Options PCI Function 2 (Word 0x39)		
0x3A	PXE	Section 6.4.6.9, Setup Options PCI Function 3 (Word 0x3A)		
0x3B	PXE	Section 6.4.6.10, Configuration Customization Options PCI Function 3 (Word 0x3B)		
0x3C	PXE	Reserved		
0x3D	PXE	Section 6.4.7.2, iSCSI boot Configuration Pointer (Word 0x3D)		
0x3E	SW	Section 6.4.9, Reserved/3rd Party External Thermal Sensor – (Word 0x3E)		
0x3F	SW	Section 6.4.10, Checksum Word (Offset 0x3F)		
0x40:0x41	SW	Reserved		
0x42	SW	Section 6.4.11, Image Unique ID (Word 0x42, 0x43)		
0x43	SW	Section 6.4.11, Image Unique ID (Word 0x42, 0x43)		
0x44:0x4F	SW	Reserved		
0x50	FW	Reserved		MNG
0x51	FW	Reserved		MNG
0x52:0x7F	FW	Reserved		MNG
0x80:0xBF	LAN Port 1 words - see Table 6-3			
0xC0:0xFF	LAN Port 2 words - see Table 6-3			
0x100:0x13F	LAN Port 3 words - see Table 6-3			
0x140...	Option ROM and Firmware Structures			

Table 6-3 maps the I350 EEPROM words that can hold different content for LAN Ports 0, 1, 2 and 3. Addresses listed in the table are an offset from the LAN Base address of the relevant EEPROM LAN section. EEPROM LAN Base addresses of the LAN ports are as follows:

- LAN Port 0 EEPROM section Base Address - 0x0
- LAN Port 1 EEPROM section Base Address – 0x80
- LAN Port 2 EEPROM section Base Address – 0xC0
- LAN Port 3 EEPROM section Base Address – 0x100

Table 6-3 LAN Ports 1, 2 and 3 EEPROM Map

EEPROM Word Offsets	Used By/In	High Byte	Low Byte
0x00:0x02	HW	Section 6.2.1, Ethernet Address (LAN Base Address + Offsets 0x00-0x02)	
0x03:0x09	SW	Reserved	
0x10:0x0C	HW	Reserved	
0x0D	HW	Section 6.2.5, Device ID (LAN Base Address + Offset 0x0D)	
0x0E	HW	Reserved	
0x0F	HW	Section 6.2.8, Initialization Control Word 2 (LAN Base Address + Offset 0x0F)	
0x10	HW	Reserved	
0x11	HW (MNG HW)	Section 6.3.6, Management Pass Through LAN Configuration Pointer (LAN Base Address + Offset 0x11)	
0x12	HW	Reserved	
0x13	HW	Section 6.2.10, Initialization Control 4 (LAN Base Address + Offset 0x13)	
0x14:0x15	HW	Reserved	
0x16	HW	Section 6.2.13, MSI-X Configuration (LAN Base Address + Offset 0x16)	
0x17	HW	Section 6.3.1, Software Reset CSR Auto Configuration Pointer (LAN Base Address + Offset 0x17)	
0x18:0x1B	HW	Reserved	
0x1C	HW	Section 6.2.18, LED 1,3 Configuration Defaults (LAN Base Address + Offset 0x1C)	
0x1D:0x1E	HW	Reserved	
0x1F	HW	Section 6.2.20, LED 0,2 Configuration Defaults (LAN Base Address + Offset 0x1F)	
0x20	HW	Section 6.2.21, Software Defined Pins Control (LAN Base Address + Offset 0x20)	
0x21:0x22	HW	Reserved	
0x23	HW	Section 6.3.3, PCIe Reset CSR Auto Configuration Pointer (LAN Base Address + Offset 0x23)	
0x24	HW	Section 6.2.24, Initialization Control 3 (LAN Base Address + Offset 0x24)	
0x25:0x26	HW	Reserved	
0x27	HW	Section 6.3.4, CSR Auto Configuration Power-Up Pointer (LAN Base Address + Offset 0x27)	
0x28:0x3E	HW	Reserved	
0x3F	SW	Section 6.4.10, Checksum Word (Offset 0x3F)	

6.2 Hardware Accessed Words

This section describes the EEPROM words that are loaded by the I350 hardware. Most of these bits are located in configuration registers. The words are only read and used if the signature field in the *EEPROM Sizing and Protected Fields* EEPROM word (word 0x12) is valid.

Note: When **Word** is mentioned before an EEPROM address, address is the absolute address in the EEPROM. When **Offset** is mentioned before an EEPROM address, the address is relative to the start of the relevant EEPROM section.

6.2.1 Ethernet Address (LAN Base Address + Offsets 0x00-0x02)

The Ethernet Individual Address (IA) is a 6-byte field that must be unique for each NIC, and thus unique for each copy of the EEPROM image. The first three bytes are vendor specific. The value from this field is loaded into the Receive Address Register 0 (RAL0/RAH0).

The Ethernet address is loaded for LAN0 from Addresses 0x0 to 0x02 and for LAN 1, 2 and 3 from offsets 0x0 to 0x2 at the start of the relevant sections.

Following table depicts mapping of the Ethernet MAC addresses to the EEPROM words.

LAN Port	MAC Address	LAN Base Address + 0x00	LAN Base Address + 0x01	LAN Base Address + 0x02
0	00-A0-C9-00-00-00	0xA000	0x00C9	0x0000
1	00-A0-C9-00-00-01	0xA000	0x00C9	0x0100
2	00-A0-C9-00-00-02	0xA000	0x00C9	0x0200
3	00-A0-C9-00-00-03	0xA000	0x00C9	0x0300

6.2.2 Initialization Control Word 1 (word 0x0A)

The *Initialization Control Word 1* in the Common section contains initialization values that:

- Set defaults for some internal registers
- Enable/disable specific features
- Determine which PCI configuration space values are loaded from the EEPROM

Bit	Name	Default in EEPROM less mode	Description
15	Reserved		Reserved
14	GPAR_EN	0b	Global parity Enable Enables parity checking of all the I350 memories. 0b - Disable parity check 1b - Enable Parity Check according to the per RAM parity enable bits. Loaded to *PCIEERRCTL* register (refer to Section 8.23.11).
13	LTR_EN	1b	LTR capabilities reporting enable. 0 - Do not report LTR support in the PCIe configuration Device Capabilities 2 register. 1 - Report LTR support in the PCIe configuration Device Capabilities 2 register. Defines default setting of LTR capabilities reporting (refer to Section 9.5.6.11).
12:7	Reserved	0x0	Reserved
6	SDP_IDDQ_EN	0b	When set, SDP IOs keep their value and direction when the I350 enters dynamic IDDQ mode either due to PCIe entering Dr state or DEV_OFF_N pin being asserted. Otherwise, SDP IOs moves to HighZ + pull-up mode in dynamic IDDQ mode.

Bit	Name	Default in EEPROM less mode	Description
5	Deadlock Timeout Enable	1b	If set, a driver granted access to the EEPROM or Flash that does not toggle the EEPROM interface for more than 2 seconds or the FLASH interface for more than 8 seconds will have the grant revoked. Refer to Section 3.3.2.1. This bit also enables *EERD* and *EEMNGCTL* timeout if EEPROM is not responding to status read,
4	LAN PLL Shutdown Enable	0b	When set, enables shutting down the PHY PLL in low-power states when the Internal PHY is powered down (such as link disconnect). When cleared, the PHY PLL is not shut down in a low-power state.
3	Power Management	1b	0b = Power Management registers set to read only. In this mode, the I350 does not execute a hardware transition to D3.Reserved 1b = Full support for power management (For normal operation, this bit must be set to 1b). See section 9.5.1 .
2	DMA clock gating Disabled	1b	When set Disables DMA clock gating power saving mode.
1	Load Subsystem IDs	1b	When this bit is set to 1b the I350 loads its PCIe Subsystem ID and Subsystem Vendor ID from the EEPROM (*Subsystem ID* and *Subsystem Vendor ID* EEPROM words).
0	Load Vendor/Device IDs	1b	When set to 1b the I350 loads its PCIe Device IDs from the EEPROM (*Device ID* or *Dummy Device ID* EEPROM words) and the PCIe Vendor ID from the EEPROM.

6.2.3 Subsystem ID (Word 0x0B)

If the *Load Subsystem IDs* in *Initialization Control Word 1* EEPROM word is set, the *Subsystem ID* word in the Common section is read in to initialize the PCIe Subsystem ID. Default value is 0x0 (refer to Section 9.4.14).

6.2.4 Subsystem Vendor ID (Word 0x0C)

If the *Load Subsystem IDs* bit in *Initialization Control Word 1* EEPROM word is set, the Subsystem Vendor ID word in the Common section is read in to initialize the PCIe Subsystem Vendor ID. The default value is 0x8086 (refer to Section 9.4.13).

6.2.5 Device ID (LAN Base Address + Offset 0x0D)

If the *Load Vendor/Device IDs* bit in *Initialization Control Word 1* is set, the *Device ID* EEPROM word is read in from the Common, LAN 1, LAN 2 and LAN 3 sections to initialize the Device ID of LAN0, LAN1, LAN2 and LAN3 functions, respectively. The default value in EEPROM-less operation is 0x151F (refer to Section 9.4.2).

6.2.6 Vendor ID (Word 0x0E)

If the *Load Vendor/Device IDs* bit in *Initialization Control Word 1* EEPROM word is set, this word is read in to initialize the PCIe Vendor ID. The default value is 0x8086 (refer to Section 9.4.1).

Note: If a value of 0xFFFF is placed in the *Vendor ID* EEPROM word, the value in the PCIe *Vendor ID* register will return to the default 0x8086 value. This functionality is implemented to avoid a system hang situation.

6.2.7 Dummy Device ID (Word 0x1D)

If the *Load Vendor/Device IDs* bit in *Initialization Control Word 1* EEPROM word is set, this word is read in to initialize the Device ID of dummy devices. The default value is 0x10A6 (refer to Section 9.4.2).

6.2.8 Initialization Control Word 2 (LAN Base Address + Offset 0x0F)

The *Initialization Control Word 2* read by the I350, contains additional initialization values that:

- Set defaults for some internal registers
- Enable/disable specific features

Bit	Name	Default in EEPROM less mode	Description
15	APM PME# Enable	0b	Initial value of the *Assert PME On APM Wakeup* bit in the Wake Up Control (*WUC.APMPME*) register. Refer to Section 8.20.1.
14	PCS parallel detect	1b	Enables PCS parallel detect. Mapped to *PCS_LCTL.AN TIMEOUT EN* bit. Refer to Section 8.17.2. Note: Bit should be 0b only when port operates in SGMII mode (*CTRL_EXT.LINK_MODE* = 10b).
13:12	Pause Capability	11b	Desired pause capability for advertised configuration base page. Mapped to *PCS_ANADV.ASM*. Refer to Section 8.17.4.
11	ANE	0b	Auto-Negotiation Enable Mapped to *PCS_LCTL.AN_ENABLE*. Refer to Section 8.17.2. Note: Bit should be 0b when port operates in internal copper PHY mode and 1000BASE-KX modes.
10	FRCSPD	0b	Force Speed Default setting for the *Force Speed* bit in the Device Control register (CTRL[11]). Refer to Section 8.2.1
9	FD	1b	Full-Duplex Default setting for duplex setting. Mapped to *CTRL*[0]. Refer to Section 8.2.1
8	TX_LPI_EN	0b	Enable entry into EEE LPI on TX path. Refer to Section 8.24.10. 0b - Disable entry into EEE LPI on TX path. 1b - Enable entry into EEE LPI on TX path.
7	MAC clock gating enable	1b	Enables MAC clock gating power saving mode. Mapped to *STATUS[31]*. This bit is relevant only if the *Enable Dynamic MAC Clock Gating* bit is set. Refer to Section 8.2.2.
6	Reserved	1b	Must be set to zero

Bit	Name	Default in EEPROM less mode	Description
5	10BASE-TE	0b	Enable low amplitude 10BASE-T operation Setting this bit enables the I350 to operate in IEEE802.3az 10BASE-Te low power operation. Bit is loaded to *IPCNFG.10BASE-TE* register bit (refer to Section 8.26.1). 0b - 10BASE-Te operation disabled. 1b - 10BASE-Te operation enabled ***Note:*** When operating in 10BASE-T mode and bit is set supported cable length is reduced.
4	Reserved	0b	Reserved
3	Enable Dynamic MAC Clock Gating	0b	When set, enables dynamic MAC clock gating mechanism. Refer to Section 8.2.3.
2	Reserved	0b	Must be Zero
1	EEE_1G_AN	1b	Report EEE 1G capability in Auto-negotiation. Refer to Section 8.26.1. 0b - Do not report EEE 1G capability in Auto-negotiation. 1b - Report EEE 1G capability in Auto-negotiation.
0	EEE_100M_AN	1b	Report EEE 100M capability in Auto-negotiation. Refer to Section 8.26.1. 0b - Do not report EEE 100M capability in Auto-negotiation. 1b - Report EEE 100M capability in Auto-negotiation.

6.2.9 EEPROM Sizing and Protected Fields (Word 0x12)

Provides indication on EEPROM size and protection.

Note: If the Enable Protection Bit in this word is set and the signature is valid, the software device driver has read but no write access to this word via the *EEC* and *EERD* registers; In this case, write access is possible only via an authenticated firmware interface.

Bit	Name	Default in EEPROM less mode	Description
15:14	Signature		The *Signature* field indicates to the I350 that there is a valid EEPROM present. If the signature field is 01b, EEPROM read is performed, otherwise the other bits in this word are ignored, no further EEPROM read is performed, and default values are used for the configuration space IDs.
13:10	EEPROM Size	0111b	These bits indicate the EEPROM's actual size. Mapped to *EEC.EE_SIZE* See (Section 8.4.1). Field Value EEPROM Size EEPROM Address Size 0000b - 0110b Reserved 0111b 16 Kbytes 2 bytes 1000b 32 Kbytes 2 bytes 1001b 64 Kbytes 2 bytes ***Note:*** 1001b - 1111b Reserved1001b - 1111b Reserved ***Note:*** The EERD register can access only the first 32 KB of the NVM.

9	EEPROM burst limit	0b	When bit is set EEPROM write burst or read burst when doing a Bit Banging operation is limited to a single word, any transaction longer than 1 word will be blocked. ***Note:*** Feature can be enabled only when EEPROM protection is enabled.
8	Disable Roll Over	0b	When bit is set and EEPROM protection is enabled EEPROM Roll over when reaching maximum address (according to EEPROM Size) when doing a write burst or read burst is blocked (either via Bit Banging or *EERD* register). ***Note:*** Feature can be enabled only when EEPROM protection is enabled.
7:5	EEPROM Write Page Size	100b	This field indicates EEPROM write page size. 000b: 2 bit (4B) write page 001b: 3 bit (8B) write page 010b: 4 bit (16B) write page 011b: 5 bit (32B) write page 100b: 6 bit (64B) write page 101b: 7 bit (128B) write page 110b: 8 bit (256B) write page 111b: 9 bit (512B) write page ***Note:*** Value used only when EEPROM protection is enabled.
4	Enable EEPROM Protection	0b	If set, all EEPROM protection schemes are enabled.
3:0	HEPSize	0x0	Hidden EEPROM Block Size This field defines the EEPROM area accessible only by manageability firmware. It can be used to store secured data and other manageability functions. The size in bytes of the secured area equals: 0 bytes if HEPSize equals zero 2^ HEPSize bytes else (for example, 2 B, 4 B, …32 KB)

6.2.10 Initialization Control 4 (LAN Base Address + Offset 0x13)

These words control general initialization values of LAN 0, LAN 1, LAN 2 and LAN 3 ports.

Bit	Name	Default in EEPROM less mode	Description
15:12	TXPbsize	0x0	Transmit internal buffer size: 0x0 - 20 KB 0x1 - 40 KB 0x2 - 80 KB 0x3 - 1 KB 0x4 - 2 KB 0x5 - 4 KB 0x6 - 8 KB 0x7 - 16 KB 0x8 - 19 KB 0x9 - 38 KB 0xA - 76 KB 0xB:0XF reserved. Notes: 1. When 4 ports are enabled maximum buffer size is 20KB. When 2 ports are enabled maximum buffer size is 40KB. When only a single port is enabled maximum buffer size is 80KB. 2. When OS to BMC traffic is enabled available buffer size for all ports is reduced by 4 KB. Sets value of ITPBS.TXPbsize. Refer to Section 8.3.

Bit	Name	Default in EEPROM less mode	Description
11:8	RXPbsize	0x0	Receive internal buffer size: 0x0 - 36 KB 0x1 - 72 KB 0x2 - 144 KB 0x3 - 1 KB 0x4 - 2 KB 0x5 - 4 KB 0x6 - 8 KB 0x7 - 16 KB 0x8 - 35 KB 0x9 - 70 KB 0xA - 140 KB Notes: 1. When 4 ports are enabled maximum buffer size is 36 KB. When 2 ports are enabled maximum buffer size is 72 KB. When only a single port is enabled maximum buffer size is 144 KB. 2. When BMC to OS traffic is enabled available buffer size for all ports is reduced by 4 KB. Sets value of *IRPBS.RXPbsize*. Refer to Section 8.3.
7	SPD Enable	1b	Smart Power Down – When set, enables Internal PHY Smart Power Down mode (refer to Section 3.7.8.5.5).
6	LPLU	1b	Low Power Link Up Enables a decrease in link speed in non-D0a states when power policy and power management states dictate it (refer to Section 3.7.8.5.4).
5:1	PHY_ADD	0x00 0x01 0x02 0x03	PHY address. Value loaded to *MDICNFG.PHYADD* field. Refer to Section 8.2.5.
0	DEV_RST_EN	1b	Enable software reset (*CTRL.DEV_RST*) generation to all ports (refer to Section 4.3).

6.2.11 PCIe L1 Exit latencies (Word 0x14)

Bits	Name	Default in EEPROM less mode	Description
15	Reserved	1b	Reserved
14:12	L1_Act_Acc_Latency	110b	Loaded to the "Endpoint L1 Acceptable Latency" field in the "Device Capabilities" in the "PCIe configuration registers" at power up.
11:9	L1 G2 Sep exit latency	101	L1 exit latency G2S. Loaded to "Link Capabilities" -> "L1 Exit Latency" at PCIe v2.1 (5GT/s) system in Separate clock setting.
8:6	L1 G2 Com exit latency	011b	L1 exit latency G2C. Loaded to "Link Capabilities" -> "L1 Exit Latency" at PCIe v2.1 (5GT/s) system in Common clock setting. ***Note:*** Should be set to 101b (16 to 32 µs) to reflect an exit time of 17 µs
5:3	L1 G1 Sep exit latency	100b	L1 exit latency G1S. Loaded to "Link Capabilities" -> "L1 Exit Latency" at PCIe v2.1 (2.5GT/s) system in Separate clock setting. ***Note:*** Should be set to 101b (16 to 32 µs) to reflect an exit time of 17 µs
2:0	L1 G1 Com exit latency	010b	L1 exit latency G1C. Loaded to "Link Capabilities" -> "L1 Exit Latency" at PCIe v2.1 (2.5GT/s) system in Common clock setting. ***Note:*** Should be set to 101b (16 to 32 µs) to reflect an exit time of 17 µs

6.2.12 PCIe Completion Timeout Configuration (Word 0x15)

Bit	Name	Default in EEPROM less mode	Description
15:5	Reserved		Reserved
4	Completion Timeout Re-send	1b	When set, enables to re-send a request once the completion timeout expired 0b = Do not re-send request on completion timeout. 1b = Re-send request on completion timeout. Refer to Section 8.6.1.
3:0	Reserved	0x0	Reserved

6.2.13 MSI-X Configuration (LAN Base Address + Offset 0x16)

These words configure MSI-X functionality for LAN 0, LAN 1, LAN 2 and LAN 3.

Bit	Name	Default in EEPROM less mode	Description
15:11	MSI_X_N	0x9	This field specifies the number of entries in MSI-X tables of the relevant LAN. The range is 0-24. MSI_X_N is equal to the number of entries minus one. Refer to Section 9.5.3.3.
10	MSI Mask	1b	MSI per-vector masking setting. This bit is loaded to the masking bit (bit 8) in the *Message Control* word of the *MSI Configuration Capability structure*.
9:0	Reserved	0x0	Reserved

6.2.14 PCIe Init Configuration 1 (Word 0x18)

This word is used to define L0s exit latencies.

Bits	Name	Default in EEPROM less mode	Description
15	Reserved	0b	Reserved.
14:12	L0s acceptable latency	011b	Loaded to the "Endpoint L0s Acceptable Latency" field in the "Device Capabilities" in the "PCIe configuration registers" at power up.
11:9	L0s G2 Sep exit latency	111b	L0s exit latency G2S. Loaded to L0s Exit Latency field in the Link Capabilities register in the PCIe configuration registers in PCIe v2.1 (5GT/s) system at Separate clock setting.
8:6	L0s G2 Com exit latency	100b	L0s exit latency G2C. Loaded to L0s Exit Latency field in the Link Capabilities register in the PCIe configuration registers in PCIe v2.1 (5GT/s) system at Common clock setting.
5:3	L0s G1 Sep exit latency	111b	L0s exit latency G1S. Loaded to L0s Exit Latency field in the Link Capabilities register in the PCIe configuration registers in PCIe v2.1 (2.5GT/s) system at Separate clock setting.

Bits	Name	Default in EEPROM less mode	Description
2:0	L0s G1 Com exit latency	011b	L0s exit latency G1C. Loaded to L0s Exit Latency field in the Link Capabilities register in the PCIe configuration registers in PCIe v2.1 (2.5GT/s) system at Common clock setting.

6.2.15 PCIe Init Configuration 2 Word (Word 0x19)

This word is used to set defaults for some internal PCIe configuration registers.

Bit	Name	Default in EEPROM less mode	Description
15	Reserved		Reserved
14	IO_Sup	1b	I/O Support (affects I/O BAR request) When set to 1b, I/O is supported. When cleared the "I/O Access Enable" bit in the "Command Reg" in the "Mandatory PCI Configuration" area is RO with a value of 0. For additional information on CSR access via IO address space see Section 8.1.1.5.
13	CSR_conf_en	1b	Enable CSR access via configuration space. When set enables CSR access via the configuration registers located at configuration address space 0x98 and 0x9C. For additional information on CSR access via configuration address space see Section 8.1.1.6.
12	Serial Number enable	0b	"Serial number capability" enable. Should be set to one.
11:0	Reserved		Reserved

6.2.16 PCIe Init Configuration 3 Word (Word 0x1A)

This word is used to set defaults for some internal PCIe registers.

Bit	Name	Default in EEPROM less mode	Description
15:13	Reserved		Reserved
12	Cache_Lsize	1b	Cache Line Size 0b = 64 bytes. 1b = 128 bytes. This bit defines the Cache line size reported in the PCIe mandatory configuration register area. Refer to Section 9.4.7.
11:10	GIO_Cap	10b	PCIe Capability Version The value of this field is reflected in the two LSBs of the capability version in the PCIe CAP register (config space – offset 0xA2). This field must be set to 10b to use extended configuration capability. Note that this is not the PCIe version. It is the PCIe capability version. This version is a field in the PCIe capability structure and is not the same as the PCIe version. It changes only when the content of the capability structure changes. For example, PCIe 1.0, 1.0a, and 1.1 all have a capability version of one. PCIe 2.0 has a version of two because it added registers to the capabilities structures. Refer to Section 9.5.6.3.

Bit	Name	Default in EEPROM less mode	Description
9:8	Max Payload Size	10b	Default packet size 00b = 128 bytes. 01b = 256 bytes. 10b = 512 bytes. 11b = Reserved. Loaded to 2 lsb bits of the *Max Payload Size Supported* field in the *Device Capabilities* register (refer to Section 9.5.6.4).
7:4	Reserved		Reserved
3:2	Act_Stat_PM_Sup	11b	Determines support for active state link power management Loaded into the PCIe Active State Link PM Support register. Refer to Section 9.5.6.7.
1	Slot_Clock_Cfg	1b	When set, the I350 uses the PCIe reference clock supplied on the connector (for add-in solutions).
0	Reserved		Reserved

6.2.17 PCIe Control 1 (Word 0x1B)

This word is used to configure initial settings for PCIe default functionality.

Bit	Name	Default in EEPROM less mode	Description
15	Reserved	0b	Reserved
14	Dummy Function Enable	0b	Controls the behavior of function 0 when disabled. Refer to Section 4.4.5. 0b - Legacy Mode 1b - Dummy Function Mode If the value of this bit is changed, a full power cycle should be performed so the change takes effect.
13:11	Reserved	010b	Reserved
10	No_Soft_Reset	1b	No_Soft_Reset Bit defines behavior of the I350 when a transition from the D3hot to D0 Power State occurs. When bit is set no internal reset is issued on transition from D3hot to D0. Value is loaded to the *No_Soft_Reset* bit in the *PMCSR* register (refer to Section 9.5.4.1).
9:7	Reserved	0x0	Reserved
6:0	Reserved	0100110b	Reserved

6.2.18 LED 1,3 Configuration Defaults (LAN Base Address + Offset 0x1C)

These EEPROM words specify the hardware defaults for the *LEDCTL* register fields controlling the LED1 (ACTIVITY indication) and LED3 (LINK_1000 indication) output behavior. Words control LEDs behavior of LAN 0, LAN 1, LAN 2 and LAN 3 ports.

Bit	Name	Default in EEPROM less mode	Description
15	LED3 Blink	0b	Initial value of *LED3_BLINK* field. 0b = Non-blinking. See Section 8.2.9 and Section 7.5.
14	LED3 Invert	0b	Initial value of *LED3_IVRT* field. 0b = Active-low output. See Section 8.2.9 and Section 7.5.
13:12	Reserved		Reserved
11:8	LED3 Mode	0111b	Initial value of the *LED3_MODE* field specifying what event/state/pattern is displayed on LED3 (LINK_1000) output. A value of 0111b (0x7) indicates 1000 Mb/s operation. See Section 8.2.9 and Section 7.5.
7	LED1 Blink	1b	Initial value of *LED1_BLINK* field. 0b = Non-blinking. See Section 8.2.9 and Section 7.5.
6	LED1 Invert	0b	Initial value of *LED1_IVRT* field. 0b = Active-low output. See Section 8.2.9 and Section 7.5.
5:4	Reserved_1	11b	Value should be 11b.
3:0	LED1 Mode	0011b	Initial value of the *LED1_MODE* field specifying what event/state/pattern is displayed on LED1 (ACTIVITY) output. A value of 0011b (0x3) indicates the ACTIVITY state. See Section 8.2.9 and Section 7.5.

6.2.19 Device Rev ID (Word 0x1E)

Bit	Name	Default in EEPROM less mode	Description
15	Power Down Enable	0b	Enable Power down when DEV_OFF_N pin is asserted or PCIe in Dr state. Refer to Section 5.2.4.1 for details.
14	LAN 3 iSCSI enable	0b	When set, LAN 3 class code is set to 0x010000 (SCSI) When reset, LAN 3 class code is set to 0x020000 (LAN) Refer to Section 9.4.6.
13	LAN 2 iSCSI enable	0b	When set, LAN 2 class code is set to 0x010000 (SCSI) When reset, LAN 2 class code is set to 0x020000 (LAN) Refer to Section 9.4.6.
12	LAN 1 iSCSI enable	0b	When set, LAN 1 class code is set to 0x010000 (SCSI) When reset, LAN 1 class code is set to 0x020000 (LAN) Refer to Section 9.4.6.

Bit	Name	Default in EEPROM less mode	Description
11	LAN 0 iSCSI enable	0b	When set, LAN 0 class code is set to 0x010000 (SCSI) When reset, LAN 0 class code is set to 0x020000 (LAN) Refer to Section 9.4.6.
10:8	AER Capability Version	0x2	AER Capability Version Number PCIe AER extended capability version number. Refer to Section 9.6.1.1. ***Note:*** Only the 3 LSB bits of the field are configured.
7:0	DEVREVID	0x0	Device Revision ID This field is loaded to *MREVID.EEPROM_RevID* (Section 8.6.10) The actual device revision ID is the EEPROM value XORed with the hardware default of Rev ID. For I350 A1, the default value is one. Refer to Section 9.4.5.

6.2.20 LED 0,2 Configuration Defaults (LAN Base Address + Offset 0x1F)

These EEPROM words specify the hardware defaults for the LEDCTL register fields controlling the LED0 (LINK_UP) and LED2 (LINK_100) output behaviors. Words control LEDs behavior of LAN 0, LAN 1, LAN 2 and LAN 3 ports.

Bit	Name	Default in EEPROM less mode	Description
15	LED2 Blink	0b	Initial value of *LED2_BLINK* field. 0b = Non-blinking. Refer to Section 8.2.9 and Section 7.5.
14	LED2 Invert	0b	Initial value of *LED2_IVRT* field. 0b = Active-low output. Refer to Section 8.2.9 and Section 7.5.
13:12	Reserved	0x0	Reserved
11:8	LED2 Mode	0110b	Initial value of the *LED2_MODE* field specifying what event/state/pattern is displayed on LED2 (LINK_100) output. A value of 0110b (0x6) indicates 100 Mb/s operation. Refer to Section 8.2.9 and Section 7.5.
7	LED0 Blink	0b	Initial value of *LED0_BLINK* field. 0b = Non-blinking. Refer to Section 8.2.9 and Section 7.5.
6	LED0 Invert	0b	Initial value of *LED0_IVRT* field. 0b = Active-low output. Refer to Section 8.2.9 and Section 7.5.
5	Global Blink Mode	0b	Global Blink Mode 0b = Blink at 200 ms on and 200ms off. 1b = Blink at 83 ms on and 83 ms off. Refer to Section 8.2.9 and Section 7.5.
4	Reserved	0b	Reserved. Set to 0b.
3:0	LED0 Mode	0010b	Initial value of the *LED0_MODE* field specifying what event/state/pattern is displayed on LED0 (LINK_UP) output. A value of 0010b (0x2) indicates the LINK_UP state. Refer to Section 8.2.9 and Section 7.5.

6.2.21 Software Defined Pins Control (LAN Base Address + Offset 0x20)

These words at offset 0x20 from start of relevant EEPROM section are used to configure initial settings of software defined pins (SDPs) for LAN 0, LAN 1, LAN 2 and LAN 3.

Bit	Name	Default in EEPROM less mode	Description
15	SDPDIR[3]	0b	SDP3 Pin – Initial Direction This bit configures the initial hardware value of the *SDP3_IODIR* bit in the Extended Device Control (CTRL_EXT) register following power up. Refer to Section 8.2.3.
14	SDPDIR[2]	0b	SDP2 Pin – Initial Direction This bit configures the initial hardware value of the *SDP2_IODIR* bit in the Extended Device Control (CTRL_EXT) register following power up. See section 8.2.3 .
13	PHY_in_LAN_disable	1b	Determines the behavior of the MAC and PHY when a LAN port is disabled through an external pin. 0b = MAC and PHY are kept functional in LAN disable (to support manageability). 1b = MAC and PHY are powered down in LAN disable (manageability cannot access the network through this port). ***Note:***
12	Disable 100 in non-D0a	0b	Disables 1000Mb/s and 100 Mb/s operation in non-D0a states (refer to Section 3.7.8.5.4). Sets default value of PHPM.Disable 100 in non-D0a bit.
11	LAN_DIS	0b	LAN Disable In LAN ports 1,2 and 3, when set to 1b, the appropriate LAN is disabled (both PCIe function and LAN access for manageability are disabled). Note: LAN port 0 can not be disabled via EEPROM to avoid case where all ports are disabled and EEPROM can not be modified. ***Note:***
10	LAN_PCI_DIS	0b	LAN PCIe Function Disable In LAN ports 1,2 and 3, when set to 1b, the appropriate LAN PCI function is disabled. For example, in the case where the LAN is functional for manageability operation but is not connected to the host through the PCIe interface. Note: LAN port 0 can not be disabled via EEPROM to avoid case where all ports are disabled and EEPROM can not be modified. ***Notes:*** 1. Bit has no effect on LAN port 0 and is Reserved with a value of 0b. 2. When bit is set, Function is in a Non-D0a (uninitialized) state. As a result if the *LPLU*, *Disable 1000 in Non-D0a* or *Disable 100 in Non-D0a* EEPROM bits are set and the *MANC.Keep_PHY_Link_Up* bit is cleared Management might operate with reduced link speed.
9	SDPDIR[1]	0b	SDP1 Pin – Initial Direction This bit configures the initial hardware value of the *SDP1_IODIR* bit in the Device Control (CTRL) register following power up. See section 8.2.1 .
8	SDPDIR[0]	0b	SDP0 Pin – Initial Direction This bit configures the initial hardware value of the *SDP0_IODIR* bit in the Device Control (CTRL) register following power up. See section 8.2.1 .
7	SDPVAL[3]	0b	SDP3 Pin – Initial Output Value This bit configures the initial power-on value output on SDP3 (when configured as an output) by configuring the initial hardware value of the *SDP3_DATA* bit in the Extended Device Control (CTRL_EXT) register after power up. See section 8.2.3 .
6	SDPVAL[2]	0b	SDP2 Pin – Initial Output Value This bit configures the initial power-on value output on SDP2 (when configured as an output) by configuring the initial hardware value of the *SDP2_DATA* bit in the Extended Device Control (CTRL_EXT) register after power up. See section 8.2.3 .

Bit	Name	Default in EEPROM less mode	Description
5	WD_SDP0	0b	When set, SDP[0] is used as a watchdog timeout indication. When reset, it is used as an SDP (as defined in bits 8 and 0). See section 8.2.1 .
4	Giga Disable	0b	When set, GbE operation is disabled. A usage example for this bit is to disable GbE operation if system power limits are exceeded (refer to Section 3.7.8.5.4).
3	Disable 1000 in non-D0a	1b	Disables 1000 Mb/s operation in non-D0a states (refer to Section 3.7.8.5.4).
2	D3COLD_WAKEUP_ADVEN	1b	Controls reporting of D3 Cold wake-up support in the *Power Management Capabilities* (*PMC*) configuration register (refer to Section 9.5.1.3). In addition bit is loaded to *CTRL.ADVD3WUC* (refer to Section 8.2.1*)*. When set, D3Cold wake up capability is advertised based on whether AUX_PWR pin is connected to 3.3V to advertise presence of auxiliary power (yes if AUX_PWR is indicated, no otherwise). When 0b, however, D3Cold wake up capability is not advertised even if AUX_PWR presence is indicated. If full 1Gb/sec. operation in D3 state is desired but the system's power requirements in this mode would exceed the D3Cold Wake up-Enabled specification limit (375mA at 3.3V), this bit can be used to prevent the capability from being advertised to the system.
1	SDPVAL[1]	0b	SDP1 Pin – Initial Output Value This bit configures the initial power-on value output on SDP1 (when configured as an output) by configuring the initial hardware value of the *SDP1_DATA* bit in the Device Control (*CTRL*) register after power up. See section 8.2.1 .
0	SDPVAL[0]	0b	SDP0 Pin – Initial Output Value This bit configures the initial power-on value output on SDP0 (when configured as an output) by configuring the initial hardware value of the *SDP0_DATA* bit in the Device Control (*CTRL*) register after power up. See section 8.2.1 .

6.2.22 Functions Control (Word 0x21)

Bit	Name	Default in EEPROM less mode	Description
15	NC-SI Clock Pad Drive Strength	0b	Defines the drive strength of the NCSI_CLK_OUT pad. If set, the driving strength is stronger. Refer to Section 11.6.1.4 for details.
14	NC-SI Data Pad Drive Strength	0b	Defines the drive strength of the NCSI_CRS_DV, NCSI_RXD and NCSI_ARB_OUT pads. If set, the driving strength is stronger. Refer to Section 11.6.1.4 for details.
13	NC-SI Output Clock Disable	0b	If set, the clock source is external. In this case, the NCSI_CLK_OUT pad is kept stable at zero and the NCSI_CLK_IN pad is used as an input source of the clock. If cleared, the I350 outputs the NC-SI clock through the NCSI_CLK_OUT pad. The NCSI_CLK_IN pad is still used as a NC-SI clock input. If NC-SI is not used, then this bit should be set. If this bit is cleared, the Device Power Down Enable bit in word 0x1E (bit 15) should not be set.
12	LAN Function Sel	0b	LAN Function Select When all LAN ports are enabled and *LAN Function Sel* = 0b, LAN 0 is routed to PCI Function 0, LAN 1 is routed to PCI Function 1, etc. If LAN Function Sel = 1b, LAN 0 is routed to PCI Function 3, LAN 1 is routed to PCI Function 2, etc. This bit is Mapped to *FACTPS[30]* bit (refer to Section 8.6.9). Refer to Section 4.4.2 for a detailed function to port mapping as a function of the *LAN Function Select* bit and the LAN Disable signals.
11	NC-SI ARB Enable	0b	NCSI Hardware Arbitration enable 0b - NCSI_ARB_IN and NCSI_ARB_OUT pads are not used. NCSI_ARB_IN is pulled up internally to provide stable input. 1b - NCSI_ARB_IN and NCSI_ARB_OUT pads are used.

Bit	Name	Default in EEPROM less mode	Description
10	BAR32	1b	Bit (loaded to the *BARCTRL* register) preserves the legacy 32 bit BAR mode when BAR32 is set. When cleared to 0b 64 bit BAR addressing mode is selected. Note: If *PREFBAR* is set the *BAR32* bit should always be 0 (64 bit BAR addressing mode). Refer to Section 9.4.11.
9	PREFBAR	0b	0b - BARs are marked as non prefetchable 1b - BARs are marked as prefetchable Refer to Section 9.4.11. ***Notes:*** 1. This bit should be set only on systems that do not generate prefetchable cycles. This bit is loaded from the PREFBAR bit in the EEPROM. 2. If PREBAR bit is set then the BAR32 bit should be 0b.
8	drop_os2bmc	1b	0b - Do not drop OS2BMC packets when management buffer is not available. 1b - Drop OS2BMC packets when management buffer is not available. ***Note:*** Clearing bit will avoid loss of OS2BMC traffic but may cause head of line blocking on traffic to network.
7:3	Reserved	01000b	Reserved
2	drop_bmc2os	1b	0b - Do not drop BMC2OS packets when no RX descriptors are available. 1b - Drop BMC2OS packets when no RX descriptors are available.
1	Deep DEV_OFF	0b	Deep DEV_OFF_N power saving mode When bit is set to 1b power saving when DEV_OFF_N is asserted is increased. ***Note:*** For bit to take effect the *Power Down Enable* bit in the Section 6.2.19, Device Rev ID (Word 0x1E) EEPROM word should be set.
0	NC-SI Slew Rate	1b	Defines the slew rate of the NCSI_CLK_OUT, NCSI_CRS_DV, NCSI_RXD and NCSI_ARB_OUT pads. If set, the slew rate is high.

6.2.23 LAN Power Consumption (Word 0x22)

Bit	Name	Default in EEPROM less mode	Description
15:8	LAN D0 Power	0x0	The value in this field is reflected in the PCI Power Management Data Register of the LAN functions for D0 power consumption and dissipation (*Data_Select* = 0 or 4). Power is defined in 100mW units. The power includes also the external logic required for the LAN function. Refer to Section 9.5.1.4.
7:5	Function 0 Common Power	0x0	The value in this field is reflected in the PCI Power Management Data register of function 0 when the *Data_Select* field is set to 8 (common function). The MSBs in the data register that reflects the power values are padded with zeros. Refer to Section 9.5.1.4.
4:0	LAN D3 Power	0x0	The value in this field is reflected in the PCI Power Management Data register of the LAN functions for D3 power consumption and dissipation (*Data_Select* = 3 or 7). Power is defined in 100 mW units. The power also includes the external logic required for the LAN function. The MSBs in the data register that reflects the power values are padded with zeros. Refer to Section 9.5.1.4.

6.2.24 Initialization Control 3 (LAN Base Address + Offset 0x24)

These words control general initialization values of LAN 0, LAN 1, LAN 2 and LAN 3 ports.

Bit	Name	Default in EEPROM less mode	Description
15	SerDes Energy Source	0b	SerDes Energy Source Detection When set to 0b, source is internal SerDes Rx circuitry for electrical idle or link-up indication. When set to 1b, source is external SRDS_[n]_SIG_DET signal for electrical idle or Link-up indication. This bit also indicates the source of the signal detect while establishing a link in SerDes mode. This bit sets the default value of the *CONNSW.ENRGSRC* bit. Refer to Section 8.2.7.
14	2 wires SFP Enable	0b	2 wires SFP interface *enable* - bit is used to enable interfacing an external PHY either VIA the MDIO or I^2C interface 0b = Disabled. When disabled, the 2 wires I/F pads are isolated. 1b = Enabled. Used to set the default value of *CTRL_EXT.I2C Enabled*. Refer to Section 8.2.3.
13	ILOS	0b	Invert Loss-of-Signal (LOS/LINK) Signal Default setting for the loss-of-signal polarity bit (CTRL[7]). Refer to Section 8.2.1.
12:11	Interrupt Pin	00b LAN 0 01b LAN 1 10b LAN 2 11b LAN3	Controls the value advertised in the *Interrupt Pin* field of the PCI Configuration header for this device/function. The encoding of this field is as follow: Value — INT Line — *Interrupt Pin* Field Value 00b — INTA — 1 01b — INTB — 2 10b — INTC — 3 11b — INTD — 4 If only a single device/function of the I350 component is enabled, this value is ignored and the *Interrupt Pin* field of the enabled device reports INTA# usage. Refer to Section 9.4.18.
10	APM Enable	0b	Initial value of *Advanced Power Management Wake Up Enable* bit in the Wake Up Control (*WUC.APME*) register. Mapped to *CTRL[6]* and to *WUC[0]*. Refer to Section 8.2.1 and Section 8.20.1. ***Note:*** On disabled port the has the *PHY_in_LAN_disable* EEPROM bit (refer to Section 6.2.21) set to 1b, the *APM enable* EEPROM bit should be 0b.
9	MDI_Flip	0b	MDI Flip When set MDI Channel D is exchanged with MDI Channel A and MDI Channel C is exchanged with MDI Channel B. Refer to Section 8.26.1.
8	ACBYP	0b	Bypass on-chip AC coupling in RX input buffers ACBYP = 0 -Normal mode; on-chip AC coupling present. ACBYP = 1 - On-chip AC coupling bypassed. Used to set default value of *P1GCTRL0.ACBYP* (refer to Section 8.2.6).
7	LAN Boot Disable	1b	A value of 1b disables the expansion ROM BAR in the PCI configuration space.
6	EN_APM_D0	0b	Enable APM wake on D0 0b - Enable APM wake only when function is in D3 and *WUC.APME* is set to 1b. 1b - Always enable APM wake when *WUC.APME* is set to 1b. Loaded to the *WUC.EN_APM_D0* bit (refer to Section 8.20.1).

Bit	Name	Default in EEPROM less mode	Description
5:4	Link Mode	00b	Initial value of *Link Mode* bits of the Extended Device Control (*CTRL_EXT.LINK_MODE*) register, specifying which link interface and protocol is used by the MAC. 00b = MAC operates with internal copper PHY (10/100/1000Base-T). 01b = MAC and SerDes I/F operate in 1000BASE-KX mode. 10b = MAC and SerDes operate in SGMII mode. 11b = MAC and SerDes I/F operate in SerDes (1000BASE-BX) mode. Section 8.2.3.
3	Com_MDIO	0b	When interfacing an external SGMII PHY or SerDes, bit defines if MDIO access is routed to a shared MDIO port on LAN 0, to support multi port external PHYs or to the dedicated per function MDIO port. 0b - MDIO access routed to the LAN port's MDIO interface. 1b - MDIO accesses on this LAN port routed to LAN port 0 MDIO interface. Used to set the default value of *MDICNFG.Com_MDIO* bit (refer to Section 8.2.5).
2	External MDIO	0b	When set PHY management interface is via external MDIO interface. Loaded to MDICNFG.Destination (refer to Section 8.2.5).
1	EXT_VLAN	0b	Sets the default for CTRL_EXT[26] bit. Indicates that additional VLAN is expected in this system (refer to Section 8.2.3).
0	Keep_PHY_Link_Up_En	0b	Enables *No PHY Reset* when the Baseboard Management Controller (BMC) indicates that the PHY should be kept on. When asserted, this bit prevents the PHY reset signal and the power changes reflected to the PHY according to the *MANC.Keep_PHY_Link_Up* value. Note: This EEPROM bit should be set to the same value for all LAN ports.

6.2.25 I/O Virtualization (IOV) Control (Word 0x25)

This word controls IOV functionality.

Bit	Name	Default in EEPROM less mode	Description
15	drop_vm_lpbk	0b	0b - Do not drop VM to VM loopback packets when loopback buffer is not available. 1b - Drop VM to VM loopback packets when loopback buffer is not available. ***Notes:*** 1. Clearing bit will avoid loss of VM to VM traffic but may cause head of line blocking on traffic to network. 2. When RX descriptor is not available for loop backed packet, drop operation always occurs.
14:9	Reserved	0x0	Reserved - must be zero
8	ARI Enabled	1b	0b = ARI capability structure not exposed as part of the capabilities link list. 1b = ARI capability structure exposed as part of the capabilities link list.
7:5	Max VFs	0x7	Defines the value of MaxVF exposed in the IOV structure. Valid values are 0-7. The value exposed in the PCIe configuration space is the value of this field + one.
4:3	MSI-X table	0x2	Defines the size of the VF function MSI-X table to request. Valid values are 0-2 (refer to Section 9.7.2.1).
2	64-bit Advertisement	1b	0b = VF BARs advertise 32-bit size. 1b = VF BARs advertise 64-bit size. ***Note:*** The *Prefetchable* bit should be set in the *I/O Virtualization (IOV) Control* EEPROM word when the bit is set to 1b.

Bit	Name	Default in EEPROM less mode	Description
1	Prefetchable	0b	0b = IOV memory BARS (0 and 3) are declared as non prefetchable. 1b = IOV memory BARS (0 and 3) are declared as prefetchable. ***Note:*** The *64-bit Advertisement* bit should be set in the *I/O Virtualization (IOV) Control* EEPROM word when the bit is set to 1b.
0	IOV Enabled	1b	0b = IOV capability structure not exposed as part of the capabilities link list. 1b = IOV capability structure exposed as part of the capabilities link list.

6.2.26 IOV Device ID (Word 0x26)

Bit	Name	Default in EEPROM less mode	Description
15:0	VDev ID	0x1520	Virtual function device ID. Loaded to PCIe SR-IOV VF device ID configuration register (refer to Section 9.6.4.7).

6.2.27 PCIe Control 2 (Word 0x28)

This word is used to configure initial settings for the PCIe default functionality.

Bits	Name	Default in EEPROM less mode	Description
15:13	Reserved		Reserved
12	ECRC Check	1b	Loaded into the ECRC Check Capable bit of the PCIe configuration registers 0b - Function is not capable of checking ECRC 1b - Function is capable of checking ECRC
11	ECRC Generation	1b	Loaded into the ECRC Generation Capable bit of the PCIe configuration registers. 0b - Function is not capable of generating ECRC 1b - Function is capable of generating ECRC
10	FLR capability enable	1b	FLR capability Enable bit is loaded to "PCIe configuration registers" -> "Device Capabilities".
9:6	FLR delay	0x1	Delay in microseconds from D0 to D3 move till reset assertion.
5	FLR delay disable	0b	FLR delay disable. 0 - Add delay to FLR assertion. 1 - Do not add delay to FLR assertion.
4	Reserved		Reserved
3:1	Flash Size	000b	Indicates Flash size according to the following equation: Size = 64 KB * 2**(*Flash Size* field). From 64 KB up to 8 MB in powers of 2. The Flash size impacts the requested memory space for the Flash and expansion ROM BARs in PCIe configuration space. Note: When CSR_Size and Flash_size fields in the EEPROM are set to 0, Flash BAR in the PCI configuration space is disabled.
0	CSR_Size	0b	The *CSR_Size* and *FLASH_Size* fields define the usable FLASH size and CSR mapping window size as shown in *BARCTRL* register description. Note: When *CSR_Size* and *Flash Size* fields in the EEPROM are set to 0, Flash BAR in the PCI configuration space is disabled.

6.2.28 PCIe Control 3 (Word 0x29)

This word is used for programming PCIe functionality and function disable control.

Bits	Name	Default in EEPROM less mode	Description
15	en_pin_pcie_func_dis	0b	When set to 1b enables disabling the relevant PCIe function by driving the relevant SDPx_1 pin (SDP0_1, SDP1_1, SDP2_1 or SDP3_1) to "0" (refer to Section 4.4.4). ***Note:*** The SDPx_1 pins on all 4 ports are sampled on Power-up and during PCIe reset.
14	LAN_DIS_POL	0b	Defines active polarity (active high or active low) of the LANx_DIS_N pins. 0b - LANx_DIS_N pins are active low (LAN port disabled when 0 driven during PCIe reset). 1b - LANx_DIS_N pins are active high (LAN port disabled when 1 driven during PCIe reset). Refer to Section 4.4.3 and Section 4.4.4 for additional information.
13	PCI_DIS_POL	0b	Defines active polarity (active high or active low) of the SDPx_1 pins when the en_pin_pcie_func_dis EEPROM bit is set to 1b. 0b - SDPx_1 pins are active low (PCIe function disabled when 0 is driven during PCIe reset). 1b - SDPx_1 pins are active high (PCIe function disabled when 1 is driven during PCIe reset). Refer to Section 4.4.4 for additional information.
12:7	Reserved	0x0	Reserved
6	Reserved	0b	Reserved
5	Wake_pin_enable	0b	Enables the use of the WAKE# pin for a PME event in all non LTSSM L2 power states. When bit is set to 1b WAKE# pin will be asserted even when device is not in D3cold state, if a wake event is detected.
4	DIS Clock Gating in DISABLE	1b	Disable clock gating when LTSSM is at DISABLE state.
3	DIS Clock Gating in L2	1b	Disable clock gating when LTSSM is at L2 state
2	DIS Clock Gating in L1	1b	Disable clock gating when LTSSM is at L1 state
1:0	Reserved		Reserved

6.2.29 End of Read-Only (RO) Area (Word 0x2C)

Defines the end of the area in the EEPROM that is RO.

Bit	Name	Description
15	Reserved	Reserved
14:0	EORO_area	Defines the end of the area in the EEPROM that is RO. The resolution is one word and can be up to byte address 0xFFFF (0x7FFF words). A value of zero indicates no RO area.

6.2.30 Start of RO Area (Word 0x2D)

Defines the start of the area in the EEPROM that is RO.

Bit	Name	Description
15	Reserved	Reserved
14:0	SORO_area	Defines the start of the area in the EEPROM that is RO. The resolution is one word and can be up to byte address 0xFFFF (0x7FFF words).

6.2.31 Watchdog Configuration (Word 0x2E)

Bit	Name	Default in EEPROM less mode	Description
15	Watchdog Enable	0b	Enable watchdog interrupt. Refer to Section 8.15.1. ***Note:*** If bit is set to 1b value of EEPROM *Watchdog Timeout* field should be 2 or higher to avoid immediate generation of a watchdog interrupt.
14:11	Watchdog Timeout	0x2	Watchdog timeout period (in seconds). Refer to Section 8.15.1. Note: Loaded to 4 LSB bits of *WDSTP.WD_Timeout* field.
10:0	Reserved		Reserved

6.2.32 VPD Pointer (Word 0x2F)

This word points to the Vital Product Data (VPD) structure. This structure is available for the NIC vendor to store its own data. A value of 0xFFFF indicates that the structure is not available.

Bit	Name	Description
15:0	VPD offset	Offset to VPD structure in words. ***Notes:*** 1. A value of 0xFFFF indicates that the structure is not available. 2. Value of bit 15 is ignored.

6.3 CSR Auto load Modules

The structures in this section are used to auto load CSRs in some reset events.

Note: PHY CSRs can not be programmed using these modules, only through the PHY configuration module (Section 6.3.13).

6.3.1 Software Reset CSR Auto Configuration Pointer (LAN Base Address + Offset 0x17)

Word points to the software Reset CSR auto configuration structures of LAN 0, LAN 1, LAN 2 and LAN3. Sections are loaded during HW auto-load as described in Section 3.3.1.3. If no CSR autoload is required for the specific LAN port, the word must be set to 0xFFFF.

The software Reset CSR Auto Configuration structure format is listed in the following tables.

Table 6-4 SW Reset CSR Auto Configuration Structure Format

Offset	High Byte[15:8]	Low Byte[7:0]	Section
0x0	Section Length = 3*n (n – number of CSRs to configure)		Section 6.3.2
0x1	Block CRC8		Section 6.3.2.1
0x2	CSR Address		Section 6.3.2.2
0x3	Data LSB		Section 6.3.2.3
0x4	Data MSB		Section 6.3.2.4
	…		
3*n - 1	CSR Address		Section 6.3.2.2
3*n	Data LSB		Section 6.3.2.3
3*n + 1	Data MSB		Section 6.3.2.4

6.3.2 Software Reset CSR Configuration Section Length - Offset 0x0

The section length word contains the length of the section in words. Note that section length count does not include the section length word and Block CRC8 word.

Bits	Name	Default	Description
15	Reserved		
14:0	Section_length		Section length in words (3 * number of CSRs to be configured).

6.3.2.1 Block CRC8 (Offset 0x1)

Bit	Name	Description
15:8	Reserved	
7:0	CRC8	

6.3.2.2 CSR Address - (Offset 3*n - 1; [n = 1... Section Length])

Bits	Name	Default	Description
15	Reserved		
14:0	CSR_ADDR		CSR Address in Double Words (4 bytes)

6.3.2.3 CSR Data LSB - (Offset 3*n; [n = 1... Section Length])

Bits	Name	Default	Description
15:0	CSR_Data_LSB		CSR Data LSB

6.3.2.4 CSR Data MSB - (Offset 3*n + 1; [n = 1... Section Length])

Bits	Name	Default	Description
15:0	CSR_Data_MSB		CSR Data MSB

6.3.3 PCIe Reset CSR Auto Configuration Pointer (LAN Base Address + Offset 0x23)

This word points to the *PCIe Reset CSR auto configuration* structures of LAN 0, LAN 1, LAN 2 and LAN3 that are read only following power-up or PCIe reset. Sections are loaded during HW auto-load as described in Section 3.3.1.3. If no CSR autoload is required for the specific LAN port, the word must be set to 0xFFFF.

The *PCIe Reset CSR Auto Configuration* structure format is listed in the following tables.

Table 6-5 PCIe Reset CSR Auto Configuration Structure Format

Offset	High Byte[15:8]	Low Byte[7:0]	Section
0x0	Section Length = 3*n (n – number of CSRs to configure)		Section 6.3.3.1
0x1	Block CRC8		Section 6.3.3.2
0x2	CSR Address		Section 6.3.3.3
0x3	Data LSB		Section 6.3.3.4
0x4	Data MSB		Section 6.3.3.5
	…		
3*n - 1	CSR Address		Section 6.3.4.3
3*n	Data LSB		Section 6.3.4.4
3*n + 1	Data MSB		Section 6.3.4.5

6.3.3.1 PCIe Reset CSR Configuration Section Length - Offset 0x0

The section length word contains the length of the section in words. Note that section length count does not include the section length word and Block CRC8 word.

Bits	Name	Default	Description
15	Reserved		
14:0	Section_length		Section length in words (3 * number of CSRs to be configured).

6.3.3.2 Block CRC8 (Offset 0x1)

Bit	Name	Description
15:8	Reserved	
7:0	CRC8	

6.3.3.3 CSR Address - (Offset 3*n - 1; [n = 1... Section Length])

Bits	Name	Default	Description
15	Reserved		
14:0	CSR_ADDR		CSR Address in Double Words (4 bytes)

6.3.3.4 CSR Data LSB - (Offset 3*n; [n = 1... Section Length])

Bits	Name	Default	Description
15:0	CSR_Data_LSB		CSR Data LSB

6.3.3.5 CSR Data MSB - (Offset 3*n + 1; [n = 1... Section Length])

Bits	Name	Default	Description
15:0	CSR_Data_MSB		CSR Data MSB

6.3.4 CSR Auto Configuration Power-Up Pointer (LAN Base Address + Offset 0x27)

This word points to the CSR auto configuration Power-Up structures of LAN 0, LAN 1, LAN 2 and LAN3 that are read only following power-up. Sections are loaded during HW auto-load as described in Section 3.3.1.3. If no CSR autoload is required for the specific LAN port, the word must be set to 0xFFFF.

The CSR Auto Configuration Power-Up structure format is listed in the following tables.

Table 6-6 CSR Auto Configuration Power-Up Structure Format

Offset	High Byte[15:8]	Low Byte[7:0]	Section
0x0	Section Length = 3*n (n – number of CSRs to configure)		Section 6.3.4.1
0x1	Block CRC8		Section 6.3.4.2
0x2	CSR Address		Section 6.3.4.3
0x3	Data LSB		Section 6.3.4.4
0x4	Data MSB		Section 6.3.4.5
	...		

Table 6-6 CSR Auto Configuration Power-Up Structure Format

Offset	High Byte[15:8]	Low Byte[7:0]	Section
3*n - 1	CSR Address		Section 6.3.4.3
3*n	Data LSB		Section 6.3.4.4
3*n + 1	Data MSB		Section 6.3.4.5

6.3.4.1 CSR Configuration Power-Up Section Length - Offset 0x0

The section length word contains the length of the section in words. Note that section length count does not include the section length word and Block CRC8 word.

Bits	Name	Default	Description
15	Reserved		
14:0	Section_length		Section length in words (3 * number of CSRs to be configured).

6.3.4.2 Block CRC8 (Offset 0x1)

Bit	Name	Description
15:8	Reserved	
7:0	CRC8	

6.3.4.3 CSR Address - (Offset 3*n - 1; [n = 1... Section Length])

Bits	Name	Default	Description
15	Reserved		
14:0	CSR_ADDR		CSR Address in Double Words (4 bytes)

6.3.4.4 CSR Data LSB - (Offset 3*n; [n = 1... Section Length])

Bits	Name	Default	Description
15:0	CSR_Data_LSB		CSR Data LSB

6.3.4.5 CSR Data MSB - (Offset 3*n + 1; [n = 1... Section Length])

Bits	Name	Default	Description
15:0	CSR_Data_MSB		CSR Data MSB

6.3.5 PCIe PHY Auto Configuration Pointer (Word 0x10)

This word points to the PCIe PHY auto configuration structure used to configure PCIe PHY related circuitry. Sections are loaded during HW auto-load as described in Section 3.3.1.3.

The PCIe PHY Auto Configuration structure format is listed in the following table.

Table 6-7 PCIe PHY Auto Configuration Structure format

Offset	High Byte[15:8]	Low Byte[7:0]	Section
0x0	Section Length = 2*n (n – number of registers to configure)		Section 6.3.5.1
0x1	Block CRC8		Section 6.3.5.3
0x2	Register Address		Section 6.3.5.3
0x3	Data		Section 6.3.5.4
	…		
2*n	Register Address		Section 6.3.5.3
2*n + 1	Data		Section 6.3.5.4

6.3.5.1 PCIe PHY Configuration Section Length - Offset 0x0

The section length word contains the length of the section in words. Note that section length count does not include the section length word and Block CRC8 word.

Bits	Name	Default	Description
15	Reserved		
14:0	Section_length		Length in words of register write section (2 * registers to be written).

6.3.5.2 Block CRC8 (Offset 0x1)

Bit	Name	Description
15:8	Reserved	
7:0	CRC8	

6.3.5.3 Register Address - (Offset 2*n; [n = 1... Number of Registers to be Written])

Bits	Name	Default	Description
15:0	Reg_ADDR		PCIe PHY Register Address in Words (2 bytes)

6.3.5.4 Register Data - (Offset 2*n + 1; [n = 1... Number of Registers to be Written])

Bits	Name	Default	Description
15:0	Reg_Data		CSR Data

6.3.5.5 Setting Default PCIe Link Width and Link Speed

By default, the I350 PCIe interface is configured to a maximum Link speed of 5.0 Gb/s and Link Width of 4 lanes. Default link width and speed can be modified by programming the Internal *PCIe Link Configuration Register* using the PCIe PHY Configuration data EEPROM Section as described below:

6.3.5.5.1 PCIe Link Configuration Register Address - Offset 0x2

Bits	Field	Address	Description
15:0	Reg_ADDR	0x94	PCIe Link Configuration Register Address in Words (2 bytes)

6.3.5.5.2 PCIe Link Configuration Register Data - Offset 0x3

Placing a value of 0x0 in the *PCIe Link Configuration Register Data* EEPROM word configures the I350 PCIe interface to PCIe Gen 2 link rates and a link width of 4.

Bits	Field	Default in EEPROMless mode	Description
15:9	Reserved	0x0	Reserved. Value should be 0.
8	Disable PCIe Gen 2	0b	1b - 2.5 Gb/s Link speed supported 0b - 5.0 Gb/s and 2.5 GT/s Link speeds supported
7:4	Reserved	0x0	Reserved. Value should b 0.
3	Disable Lane 3	0b	0b - Lane 3 enabled 1b - Lane 3 disabled
2	Disable Lane 2	0b	0b - Lane 2 enabled 1b - Lane 2 disabled
1	Disable Lane 1	0b	0b - Lane 1 enabled 1b - Lane 1 disabled
0	Disable Lane 0	0b	0b - Lane 0 enabled 1b - Lane 0 disabled

6.3.5.5.3 PCIe Link Power Down Register Address - Offset 0x4

Bits	Field	Address	Description
15:0	Reg_ADDR	0x96	PCIe Link Configuration Register Address in Words (2 bytes)

6.3.5.5.4 PCIe Link Power Down Register Data - Offset 0x5

Bits	Field	Default in EEPROMless mode	Description
15:12	Reserved	0x0	Reserved. Value should be 0.
11:8	Latency_To_Enter_L0s	0x3	Sets Latency in µs between PCIe Idle detection and entry to L0s ***Note:*** PCIe specification defines that the value should not exceed 7 µs.
7:6	Reserved	0x0	Reserved. Value should be 0.

Bits	Field	Default in EEPROMless mode	Description
5:0	Latency_To_Enter_L1	0x1F	Sets Latency in µs between entry to L0s and entry to L1.

6.3.6 Management Pass Through LAN Configuration Pointer (LAN Base Address + Offset 0x11)

The Management Pass Through LAN Configuration Pointer (LAN Base Address + Offset 0x11) points to the start (offset 0x0) of this type of structure, to configure manageability filters. If pointer is 0xFFFF then no structure exists. Structure is loaded during HW EEPROM auto-load as described in Section 3.3.1.3.

Bit	Name	Description
15:0	Pointer	Pointer to the PT LAN Configuration Structure. Refer to for details of the structure. ***Note:*** Placing a value of 0xFFFF for a pointer indicates that the structure is not present in the EEPROM.

6.3.6.1 PT LAN Configuration Structure

The Management PT LAN Configuration Structure format is listed in the following tables.

Table 6-8 Management PT LAN Configuration Structure Format

Offset	High Byte[15:8]	Low Byte[7:0]	Section
0x0	Section Length = 3*n (n – number of CSRs to configure)		Section 6.3.6.2
0x1	Block CRC8		Section 6.3.6.3
0x2	CSR Address		Section 6.3.6.4
0x3	Data LSB		Section 6.3.6.5
0x4	Data MSB		Section 6.3.6.6
	…		
3*n - 1	CSR Address		Section 6.3.6.4
3*n	Data LSB		Section 6.3.6.5
3*n + 1	Data MSB		Section 6.3.6.6

6.3.6.2 Management PT LAN Configuration Structure Section Length - Offset 0x0

The section length word contains the length of the section in words. Note that section length count does not include the section length word and Block CRC8 word.

Bits	Name	Default	Description
15	Reserved		
14:0	Section_length		Section length in words.

6.3.6.3 Block CRC8 (Offset 0x1)

Bit	Name	Description
15:8	Reserved	
7:0	CRC8	

6.3.6.4 CSR Address - (Offset 2*n; [n = 1... Section Length])

Bits	Name	Default	Description
15	Reserved		
14:0	CSR_ADDR		CSR Address in Double Words (4 bytes)

6.3.6.5 CSR Data LSB - (Offset 0x1 + 2*n; [n = 1... Section Length])

Bits	Name	Default	Description
15:0	CSR_Data_LSB		CSR Data LSB

6.3.6.6 CSR Data MSB - (Offset 0x2 + 2*n; [n = 1... Section Length])

Bits	Name	Default	Description
15:0	CSR_Data_MSB		CSR Data MSB

6.3.6.7 Manageability Filters

Following table lists registers that can be programmed via the Management PT LAN Configuration structure.

Name	Description	Section
MAVTV	Management VLAN TAG Value	Section 8.21.1
MFUTP	Management Flex UDP/TCP Ports	Section 8.21.2
METF	Management Ethernet Type Filters	Section 8.21.3
MNGONLY	Management Only Traffic Register	Section 8.21.5
MDEF	Manageability Decision Filters	Section 8.21.6
MDEF_EXT	Manageability Decision Filters Extension	Section 8.21.7
MIPAF	Manageability IP Address Filter registers (IPv4 or IPV6) Note: Can be used to filter IPV4 Address of ARP packets.	Section 8.21.8
MMAL	Manageability MAC Address Low Registers	Section 8.21.9
MMAH	Manageability MAC Address High Registers	Section 8.21.10
FTFT	Flexible TCO Filter Table registers	Section 8.21.11

6.3.7 Common Firmware Parameters – (Global MNG Offset 0x3)

6.3.7.1 Section Header — Offset 0x0

Bits	Name	Default	Description	Reserved
15:8	Block CRC8			
7:0	Block Length	0x3	Block length in words.	

6.3.7.2 Common Firmware Parameters 1 - Offset 0x1

Bits	Name	Default	Description	Reserved
15	Enable Firmware Reset		0b = Firmware reset via HICR is disabled. 1b = Firmware reset via HICR is enabled.	
14:13	Redirection Sideband Interface		00b = SMBus. 01b = NC-SI. 10b = MCTP. 11b = Reserved.	
12	Restore MAC Address		0b = Do not restore MAC address at power on. 1b = Restore MAC address at power on.	
11	Reserved	1b	Reserved	
10:8	Manageability Mode		0x0 = None. 0x1 = Reserved. 0x2 = Pass Through (PT) mode. 0x7:0x3 = Reserved.	
7	Serdes Power down - port 3	1b	When set, enables the SerDes of port 3 to enter a low power state when the function is in Dr state. Refer to Chapter 5 and Section 8.2.3.	
6	Serdes Power down - port 2	1b	When set, enables the SerDes of port 2 to enter a low power state when the function is in Dr state. Refer to Chapter 5 and Section 8.2.3.	
5	LAN1 Force TCO Reset Disable	1b	0b = Enable Force TCO reset on LAN1. 1b = Disable Force TCO reset on LAN1.	
4	LAN0 Force TCO Reset Disable	1b	0b = Enable Force TCO reset on LAN0. 1b = Disable Force TCO reset on LAN0.	
3	Proxying Capable	1b	0b = Disable Protocol Offload. 1b = Enable Protocol Offload.	
2	OS2BMC Capable		0b = Disable. 1b = Enable.	
1	Serdes Power down - port 1	1b	When set, enables the SerDes of port 1 to enter a low power state when the function is in Dr state. Refer to Chapter 5 and Section 8.2.3.	
0	Serdes Power down - port 0	1b	When set, enables the SerDes of port 0 to enter a low power state when the function is in Dr state. Refer to Chapter 5 and Section 8.2.3.	

6.3.7.3 Common Firmware Parameters 2 – Offset 0x2

Bits	Name	Default	Description	Reserved
15	Reserved	1b	Reserved.	
14	Reserved	1b	Reserved.	
13	Reserved	1b	Reserved.	
12	Reserved	1b	Reserved.	
11	Multi-Drop NC-SI	1b	Multi-Drop NC-SI topology. 0b - Point-to-point 1b - Multi-drop (default) When bit is set the NCSI_CRS_DV and NCSI_RXD[1:0] pins are High-Z following power-up, otherwise the pins are driven.	
10	PARITY_ERR_RST_EN	1b	When set enables reset of Management logic and generation of internal Firmware reset as a result of Parity Error detected in Management memories.	
9	Enable All Phys in D3	0b	0b - Only ports activated for BMC activity will stay active in D3: In NCSI mode – according to enable channel command to the specific port. In SMBUS mode – according to EEPROM port enable bit. 1b - All PHYs will stay active in D3.	
8	Restore KEEP_PHY_LINK_UP Disable	1b	0b = Restore the KEEP_PHY_LINK_UP bit in the MANC CSR according to the value maintained in firmware. 1b = Legacy behavior (KEEP_PHY_LINK_UP is cleared by MAIN_PWR_OK and is not restored by firmware.)	
7:6	Reserved	11b	Reserved	
5	LAN3 Force TCO Reset Disable	1b	0b = Enable Force TCO reset on LAN3. 1b = Disable Force TCO reset on LAN3.	
4	LAN2 Force TCO Reset Disable	1b	0b = Enable Force TCO reset on LAN2. 1b = Disable Force TCO reset on LAN2.	
3	LAN3_FTCO_ISOL_DIS	1b	LAN3 force TCO Isolate disable (1b disable; 0b enable).	
2	LAN2_FTCO_ISOL_DIS	1b	LAN2 force TCO Isolate disable (1b disable; 0b enable).	
1	LAN1_FTCO_ISOL_DIS	1b	LAN1 force TCO Isolate disable (1b disable; 0b enable).	
0	LAN0_FTCO_ISOL_DIS	1b	LAN0 force TCO Isolate disable (1b disable; 0b enable).	

6.3.8 Pass Through LAN 0...3 Configuration Modules (Global MNG Offsets 0x05, 0x08, 0x0D, 0X0E)

The following sections describe pointers and structures dedicated to pass-through mode for LAN0, LAN1, LAN2 and LAN3.

- LAN0 structure is pointed by the Firmware Module pointer at offset 0x5.
- LAN1 structure is pointed by the Firmware Module pointer at offset 0x8.
- LAN2 structure is pointed by the Firmware Module pointer at offset 0xD.
- LAN3 structure is pointed by the Firmware Module pointer at offset 0xE.

Note: If there's a conflict between the definition in this EEPROM structure and the EEPROM setting in the *PT LAN Configuration* Structure (refer to Section 6.3.6.1) then the values defined in this structure take precedence.

6.3.8.1 Section Header — Offset 0x0

Bits	Name	Default	Description	Reserved
15:8	Block CRC8			
7:0	Block Length		Block length in words.	

6.3.8.2 LAN 0/1/2/3 IPv4 Address 0 LSB; (MIPAF12 LSB) — Offset 0x01

This value will be stored in the *MIPAF[12]* register (0x58E0). Refer to Section 8.21.8 for a description of this register.

Bits	Name	Default	Description	Reserved
15:8	LAN IPv4 Address 0 Byte 1		LAN 0/1/2/3 IPv4 Address 0, Byte 1	
7:0	LAN IPv4 Address 0 Byte 0		LAN 0/1/2/3 IPv4 Address 0, Byte 0	

6.3.8.3 LAN 0/1/2/3 IPv4 Address 0 MSB; (MIPAF12 MSB) — Offset 0x02

This value will be stored in the *MIPAF[12]* register (0x58E0). Refer to Section 8.21.8 for a description of this register.

Bits	Name	Default	Description	Reserved
15:8	LAN IPv4 Address 0 Byte 3		LAN 0/1/2/3 IPv4 Address 0, Byte 3	
7:0	LAN IPv4 Address 0 Byte 2		LAN 0/1/2/3 IPv4 Address 0, Byte 2	

6.3.8.4 LAN 0/1/2/3 IPv4 Address 1; (MIPAF13) — Offset 0x03-0x04

Same structure as LAN0/1/2/3 IPv4 Address 0.

These values will be stored in the *MIPAF[13]* register (0x58E4). Refer to Section 8.21.8 for a description of this register.

6.3.8.5 LAN 0/1/2/3 IPv4 Address 2; (MIPAF14) — Offset 0x05-0x06

Same structure as LAN0/1/2/3 IPv4 Address 0.

These values will be stored in the *MIPAF[14]* register (0x58E8). Refer to Section 8.21.8 for a description of this register.

6.3.8.6 LAN 0/1/2/3 IPv4 Address 3; (MIPAF15) — Offset 0x07-0x08

Same structure as LAN0/1/2/3 IPv4 Address 0.

These values will be stored in the *MIPAF[15]* register (0x58EC). Refer to Section 8.21.8 for a description of this register.

6.3.8.7 LAN 0/1/2/3 Ethernet MAC Address 0 LSB (MMAL0) — Offset 0x09

This word is loaded by firmware to the 16 LS bits of the *MMAL[0]* register. Refer to Section 8.21.9 for a description of the register.

Bits	Name	Default	Description	Reserved
15:8			LAN 0/1/2/3 Ethernet MAC Address 0, Byte 1	
7:0			LAN 0/1/2/3 Ethernet MAC Address 0, Byte 0	

6.3.8.8 LAN 0/1/2/3 Ethernet MAC Address 0 MID; (MMAL0) — Offset 0x0A

This word is loaded by firmware to the 16 MS bits of the *MMAL[0]* register.

Bits	Name	Default	Description	Reserved
15:8			LAN 0/1/2/3 Ethernet MAC Address 0, Byte 3	
7:0			LAN 0/1/2/3 Ethernet MAC Address 0, Byte 2	

6.3.8.9 LAN 0/1/2/3 Ethernet MAC Address 0 MSB; (MMAH0) — Offset 0x0B

This word is loaded by firmware to the *MMAH[0]* register.

Bits	Name	Default	Description	Reserved
15:8			LAN 0/1/2/3 Ethernet MAC Address 0, Byte 5	
7:0			LAN 0/1/2/3 Ethernet MAC Address 0, Byte 4	

6.3.8.10 LAN 0/1/2/3 Ethernet MAC Address 1; (MMAL/H1) — Offset 0x0C-0x0E

Same structure as LAN0 Ethernet MAC Address 0. Loaded to *MMAL[1]* and *MMAH[1]* registers.

6.3.8.11 LAN 0/1/2/3 Ethernet MAC Address 2; (MMAL/H2) — Offset 0x0F-0x11

Same structure as LAN0 Ethernet MAC Address 0. Loaded to *MMAL[2]* and *MMAH[2]* registers.

6.3.8.12 LAN 0/1/2/3 Ethernet MAC Address 3; (MMAL/H3) — Offset 0x12-0x14

Same structure as LAN0 Ethernet MAC Address 0. Loaded to *MMAL[3]* and *MMAH[3]* registers.

6.3.8.13 LAN 0/1/2/3 UDP/TCP Flexible Filter Ports 0 — 7; (MFUTP Registers) — Offset 0x15 - 0x1C

The words depicted in Table 6-9 are loaded by Firmware to the *MFUTP* registers. Refer to Section 8.21.2 for a description of the register.

Table 6-9 MFUTP EEPROM Words

Offset	Bits	Description	Reserved
0x15	15:0	LAN UDP/TCP Flexible Filter Value Port0	
0x16	15:0	LAN UDP/TCP Flexible Filter Value Port1	
0x17	15:0	LAN UDP/TCP Flexible Filter Value Port2	
0x18	15:0	LAN UDP/TCP Flexible Filter Value Port3	
0x19	15:0	LAN UDP/TCP Flexible Filter Value Port4	
0x1A	15:0	LAN UDP/TCP Flexible Filter Value Port5	
0x1B	15:0	LAN UDP/TCP Flexible Filter Value Port6	
0x1C	15:0	LAN UDP/TCP Flexible Filter Value Port7	

6.3.8.14 Reserved EEPROM Words - Offset 0x1D - 0x24

EEPROM words at offsets 0x1D to 0x24 are reserved.

6.3.8.15 LAN 0/1/2/3 VLAN Filter 0 - 7; (MAVTV Registers) — Offset 0x25 — 0x2C

The words depicted in Table 6-9 are loaded by Firmware to the *MAVTV* registers. Refer to Section 8.21.1 for a description of the register.

Table 6-10 MAVTV EEPROM Words

Offset	Bits	Description	Reserved
0x25	15:12	Reserved	
0x25	11:0	LAN 0/1/2/3 VLAN Filter 0 Value	
0x26	15:12	Reserved	
0x26	11:0	LAN 0/1/2/3 VLAN Filter 1 Value	
0x27	15:12	Reserved	

Offset	Bits	Description	Reserved
0x27	11:0	LAN 0/1/2/3 VLAN Filter 2 Value	
0x28	15:12	Reserved	
0x28	11:0	LAN 0/1/2/3 VLAN Filter 3 Value	
0x29	15:12	Reserved	
0x29	11:0	LAN 0/1/2/3 VLAN Filter 4 Value	
0x2A	15:12	Reserved	
0x2A	11:0	LAN 0/1/2/3 VLAN Filter 5 Value	
0x2B	15:12	Reserved	
0x2B	11:0	LAN 0/1/2/3 VLAN Filter 6 Value	
0x2C	15:12	Reserved	
0x2C	11:0	LAN 0/1/2/3 VLAN Filter 7 Value	

6.3.8.16 Reserved EEPROM Words - Offset 0x2D to 0x2E

EEPROM words at offsets 0x2D to 0x2E are reserved.

6.3.8.17 LAN 0/1/2/3 MANC value LSB; (LMANC LSB) — Offset 0x2F

The value in this EEPROM word will be stored in the LSB word of the *MANC* register. Refer to Section 8.21.4 for a description of this register.

Bits	Name	Default	Description	Reserved
15:0	Reserved	0x0	Reserved	

6.3.8.18 LAN 0/1/2/3 MANC Value MSB; (LMANC MSB) — Offset 0x30

The value in this EEPROM word will be stored in the MSB word of the *MANC* register. Refer to Section 8.21.4 for a description of this register.

Bits	Name	Default	Description	Reserved
15	Reserved		Reserved.	
14	MPROXYE	0b	Management Proxying Enable When set to 1b Proxying of packets is enabled when device is in D3 low power state. Bit loaded to *MANC.MPROXYE* bit (refer to Section 8.21.4).	
13:11	Reserved		Reserved.	
10	NET_TYPE	0b	NET TYPE: 0b = pass only un-tagged packets. 1b = pass only VLAN tagged packets. Valid only if *FIXED_NET_TYPE* is set.	
9	FIXED_NET_TYPE	0b	Fixed net type If set, only packets matching the net type defined by the NET_TYPE field passes to manageability. Otherwise, both tagged and un-tagged packets can be forwarded to the manageability engine.	

Bits	Name	Default	Description	Reserved
8	EN_IPv4_FILTER	0b	Enable IPv4 address Filters When set, the last 128 bits of the *MIPAF* register are used to store 4 IPv4 addresses for IPv4 filtering. When cleared, these bits store a single IPv6 filter.	
7	EN_XSUM_FILTER	0b	Enable checksum filtering to MNG When this bit is set, only packets that pass L3, L4 checksum are sent to the Management Controller.	
6:0	Reserved		Reserved. Value should be 0.	
3	Reserved	0b	Reserved. Value should be 0.	

6.3.8.19 LAN 0/1/2/3 Receive Enable 1; (LRXEN1) — Offset 0x31

Bits	Name	Default	Description	Reserved
15:8	Receive Enable byte 12		BMC SMBus slave address.	
7	Enable BMC Dedicated MAC		Enable BMC Dedicated MAC 0b - Disable BMC Dedicated MAC 1b - Enable BMC Dedicated MAC	
6	Reserved		Reserved Must be set to 1b.	
5:4	Notification method		00b = SMBus alert. 01b = Asynchronous notify. 10b = Direct receive. 11b = Reserved.	
3	Enable ARP Response		Enable ARP Response 0b - Disable ARP Response 1b - Enable ARP Response	
2	Enable Status Reporting		Enable Status Reporting 0b - Disable Status Reporting 1b - Enable Status Reporting	
1	Enable Receive All		Enable Receive All 0b - Disable Receive All 1b - Enable Receive All	
0	Enable Receive TCO		Enable Receive TCO 0b - Disable Receive TCO 1b - Enable Receive TCO	

6.3.8.20 LAN 0/1/2/3 Receive Enable 2; (LRXEN2) — Offset 0x32

Bits	Name	Default	Description	Reserved
15:8	Receive Enable byte 14	0x0	Alert Value	
7:0	Receive Enable byte 13	0x0	Interface Data	

6.3.8.21 LAN 0/1/2/3 MNGONLY LSB; (LMNGONLY LSB) - Offset 0x33

The value in this EEPROM word will be stored in the LSB word of the *MNGONLY* register. Refer to Section 8.21.5 for a description of this register.

Bits	Name	Default	Description	Reserved
15:8	Reserved			
7:0	Exclusive to MNG		Exclusive to MNG – when set, indicates that packets forwarded by the manageability filters to manageability are not sent to the host. Bits 0...7 correspond to decision rules defined in registers *MDEF*[0...7] and *MDEF_EXT*[0...7].	

6.3.8.22 LAN 0/1/2/3 MNGONLY MSB; (LMNGONLY MSB) - Offset 0x34

The value in this EEPROM word will be stored in the MSB word of the *MNGONLY* register. Refer to Section 8.21.5 for a description of this register.

Bits	Name	Default	Description	Reserved
15:0	Reserved	0x0	Reserved	

6.3.8.23 Manageability Decision Filters 0 LSB; (MDEF0 LSB) - Offset 0x35

The value in this EEPROM word will be stored in the LSB word of the *MDEF[0]* register. Refer to Section 8.21.6 for a description of this register.

Bits	Name	Default	Description	Reserved
15:0	MDEF0_L		Loaded to 16 LS bits of *MDEF[0]* register.	

6.3.8.24 Manageability Decision Filters 0 MSB; (MDEF0 MSB) - Offset 0x36

The value in this EEPROM word will be stored in the MSB word of the *MDEF[0]* register. Refer to Section 8.21.6 for a description of this register.

Bits	Name	Default	Description	Reserved
15:0	MDEF0_M		Loaded to 16 MS bits of *MDEF[0]* register.	

6.3.8.25 Manageability Decision Filters Extend 0 LSB; (MDEF_EXT0 LSB) - Offset 0x37

The value in this EEPROM word will be stored in the LSB word of the *MDEF_EXT[0]* register. Refer to Section 8.21.7 for a description of this register.

Bits	Name	Default	Description	Reserved
15:0	MDEFEXT0_L		Loaded to 16 LS bits of *MDEF_EXT[0]* register.	

6.3.8.26 Manageability Decision Filters Extend 0 MSB; (MDEF_EXT0 MSB) - Offset 0x38

The value in this EEPROM word will be stored in the MSB word of the *MDEF_EXT[0]* register. Refer to Section 8.21.7 for a description of this register.

Bits	Name	Default	Description	Reserved
15:0	MDEF0EXT_M		Loaded to 16 MS bits of *MDEF_EXT[0]* register.	

6.3.8.27 Manageability Decision Filters; (MDEF1-6 and MDEF_EXT1-6) - Offset 0x39-0x50

Same as words 0x035...0x38 for MDEF[1] and MDEF_EXT[1]...MDEF[6] and MDEF_EXT[6]

6.3.8.28 Manageability Ethertype Filter 0 LSB; (METF0 LSB) - Offset 0x51

The value in this EEPROM word will be stored in the LSB word of the *METF[0]* register. Refer to Section 8.21.3 for a description of this register.

Bits	Name	Default	Description	Reserved
15:0	METF0_L		Loaded to 16 LS bits of *METF*[0] register.	

6.3.8.29 Manageability Ethertype Filter 0 MSB; (METF0 MSB) - Offset 0x52

The value in this EEPROM word will be stored in the MSB word of the *METF[0]* register. Refer to Section 8.21.3 for a description of this register.

Bits	Name	Default	Description	Reserved
15:0	METF0_M		Loaded to 16 MS bits of *METF[0]* register (reserved bits in the METF registers should be set in the EEPROM to the register's default values).	

6.3.8.30 Manageability Ethertype Filter 1...3; (METF1...3) - Offset 0x53...0x58

Same as words 0x51 and 0x52 for *METF[1]*...*METF[3]* registers.

6.3.8.31 ARP Response IPv4 Address 0 LSB; (ARP LSB) - Offset 0x59

Bits	Name	Default	Description	Reserved
15:0	ARP Response IPv4 Address 0, Byte 1		ARP Response IPv4 Address 0, Byte 1 (firmware use).	
7:0	ARP Response IPv4 Address 0, Byte 0		ARP Response IPv4 Address 0, Byte 0 (firmware use).	

6.3.8.32 ARP Response IPv4 Address 0 MSB; (ARP MSB) - Offset 0x5A

Bits	Name	Default	Description	Reserved
15:8	ARP Response IPv4 Address 0, Byte 3		ARP Response IPv4 Address 0, Byte 3 (firmware use).	
7:0	ARP Response IPv4 Address 0, Byte 2		ARP Response IPv4 Address 0, Byte 2 (firmware use).	

6.3.8.33 LAN0/1/2/3 IPv6 Address 0 LSB; (MIPAF0 LSB) - Offset 0x5B

This value will be stored in the *MIPAF[0]* register (0x58B0). Refer to Section 8.21.8 for a description of this register.

Bits	Name	Default	Description	Reserved
15:8	LAN IPv6 Address 0 Byte 1		LAN IPv6 Address 0 Byte 1	
7:0	LAN IPv6 Address 0 Byte 0		LAN IPv6 Address 0 Byte 0	

6.3.8.34 LAN0/1/2/3 IPv6 Address 0 MSB; (MIPAF0 MSB) - Offset 0x5C

This value will be stored in the *MIPAF[0]* register (0x58B0). Refer to Section 8.21.8 for a description of this register.

Bits	Name	Default	Description	Reserved
15:8	LAN IPv6 Address 0 Byte 3		LAN IPv6 Address 0 Byte 3	
7:0	LAN IPv6 Address 0 Byte 2		LAN IPv6 Address 0 Byte 2	

6.3.8.35 LAN0/1/2/3 IPv6 Address 0 LSB; (MIPAF1 LSB)- Offset 0x5D

This value will be stored in the *MIPAF[1]* register (0x58B4). Refer to Section 8.21.8 for a description of this register.

Bits	Name	Default	Description	Reserved
15:8	LAN IPv6 Address 0 Byte 5		LAN IPv6 Address 0 Byte 5	
7:0	LAN IPv6 Address 0 Byte 4		LAN IPv6 Address 0 Byte 4	

6.3.8.36 LAN0/1/2/3 IPv6 Address 0 MSB; (MIPAF1 MSB) - Offset 0x5E

This value will be stored in the *MIPAF[1]* register (0x58B4). Refer to Section 8.21.8 for a description of this register.

Bits	Name	Default	Description	Reserved
15:8	LAN IPv6 Address 0 Byte 7		LAN IPv6 Address 0 Byte 7	
7:0	LAN IPv6 Address 0 Byte 6		LAN IPv6 Address 0 Byte 6	

6.3.8.37 LAN0/1/2/3 IPv6 Address 0 LSB; (MIPAF2 LSB) - Offset 0x5F

This value will be stored in the *MIPAF[2]* register (0x58B8). See Section 8.21.8 for a description of this register.

Bits	Name	Default	Description	Reserved
15:8	LAN IPv6 Address 0 Byte 9		LAN IPv6 Address 0 Byte 9	
7:0	LAN IPv6 Address 0 Byte 8		LAN IPv6 Address 0 Byte 8	

6.3.8.38 LAN0/1/2/3 IPv6 Address 0 MSB; (MIPAF2 MSB) - Offset 0x60

This value will be stored in the *MIPAF[2]* register (0x58B8). Refer to Section 8.21.8 for a description of this register.

Bits	Name	Default	Description	Reserved
15:8	LAN IPv6 Address 0 Byte 11		LAN IPv6 Address 0 Byte 11	
7:0	LAN IPv6 Address 0 Byte 10		LAN IPv6 Address 0 Byte 10	

6.3.8.39 LAN0/1/2/3 IPv6 Address 0 LSB; (MIPAF3 LSB) - Offset 0x61

This value will be stored in the *MIPAF[3]* register (0x58BC). Refer to Section 8.21.8 for a description of this register.

Bits	Name	Default	Description	Reserved
15:8	LAN IPv6 Address 0 Byte 13		LAN IPv6 Address 0 Byte 13	
7:0	LAN IPv6 Address 0 Byte 12		LAN IPv6 Address 0 Byte 12	

6.3.8.40 LAN0/1/2/3 IPv6 Address 0 MSB; (MIPAF3 MSB) - Offset 0x62

This value will be stored in the *MIPAF[3]* register (0x58BC). Refer to Section 8.21.8 for a description of this register.

Bits	Name	Default	Description	Reserved
15:8	LAN IPv6 Address 0 Byte 15		LAN IPv6 Address 0 Byte 15	
7:0	LAN IPv6 Address 0 Byte 14		LAN IPv6 Address 0 Byte 14	

6.3.8.41 LAN0/1/2/3 IPv6 Address 1; MIPAF (Offset 0x63:0x6A)

Same structure as LAN0/1/2/3 IPv6 Address 0.

These value are stored in the *MIPAF[7:4]* registers (0x58C0 - 0x58CC).

6.3.8.42 LAN0/1/2/3 IPv6 Address 2; MIPAF (Offset 0x6B:0x72)

Same structure as LAN01/2/3 IPv6 Address 0.

These value are stored in the *MIPAF[11:8]* registers (0x58D0 - 0x58DC).

6.3.9 Sideband Configuration Module (Global MNG Offset 0x06)

This module is pointed to by global offset 0x06 of the manageability control table.

6.3.9.1 Section Header — Offset 0x0

Bits	Name	Default	Description	Reserved
15:8	Block CRC8			
7:0	Block Length	0x0E	Section length in words.	

6.3.9.2 SMBus Maximum Fragment Size — Offset 0x01

Bits	Name	Default	Description	Reserved
15:0	Max Fragment Size	0x20	SMBus Maximum Fragment Size (bytes). Note: Value should be in the 32 to 240 Byte range. In MCTP mode, this value should be set to 0x45 (64 bytes payload + 5 bytes of MCTP header)	

6.3.9.3 SMBus Notification Timeout and Flags — Offset 0x02

Bits	Name	Default	Description	Reserved
15:8	SMBus Notification Timeout (ms)	0xFF	SMBus Notification Timeout Timeout value in milliseconds from notification to completion of packet read by the external BMC. When completion of read exceeds the specified time packet is discarded. ***Note:*** A value 0, no discard.	
7:6	SMBus Connection Speed	00b	Defines the clock speed when accessing the bus as a master. 00b = Standard SMBus connection (100 KHz). 01b = Reserved. 10b = Reserved. 11b = Reserved	
5	SMBus Block Read Command	0b	0b = Block read command is 0xC0. 1b = Block read command is 0xD0.	
4	SMBus Addressing Mode	1b	0b = Single address mode. 1b = Multi address mode.	
3	Enable fairness arbitration	1b	0b = Disable fairness arbitration. 1b = Enable fairness arbitration.	
2	Disable SMBus ARP Functionality	1b	Disable SMBus ARP Functionality 0b = Enable SMBus ARP Functionality. 1b = Disable SMBus ARP Functionality.	
1	SMBus ARP PEC	1b	SMBus ARP PEC 0b = Disable SMBus ARP PEC. 1b = Enable SMBus ARP PEC.	
0	Reserved		Reserved	

6.3.9.4 SMBus Slave Addresses 1 — Offset 0x03

Bits	Name	Default	Description	Reserved
15:9	SMBus 1 Slave Address	0x4A	SMBus 1 Slave Address ***Note:*** Multi address mode only.	
8	Reserved	0b	Reserved	
7:1	SMBus 0 Slave Address	0x49	SMBus 0 Slave Address	
0	Reserved	0b	Reserved	

6.3.9.5 Reserved — Offset 0x04

6.3.9.6 Reserved — Offset 0x05

Bits	Name	Default	Description	Reserved
15:0	Reserved	0x0	Reserved	

Bits	Name	Default	Description	Reserved
15:0	Reserved	0x0	Reserved	

6.3.9.7 NC-SI Configuration - Offset 0x06

Bits	Name	Default	Description	Reserved
15:12	Reserved		Reserved.	
11	Legacy Statistics implementation	0b	0 = Implement statistics as defined in NC-SI 1.0.0 spec 1 = Implement statistics as defined in the 82576 ***Note:***	
10	Flow control	0b	0b = NC-SI flow Control Disable 1b = NC-SI flow control Enable.	
9	NC-SI HW arbitration support	0b	NC-SI HW arbitration support 0b = NC-SI HW arbitration not supported. 1b = NC-SI HW arbitration supported. ***Note:*** if the *NC-SI ARB Enable* bit (bit 11) in the *Functions Control* EEPROM word is set to 1b and NC-SI HW arbitration is not supported in the Firmware EEPROM bit then NC-SI Hardware arbitration logic operates in Bypass mode and the I350 is allowed to transmit through the NC-SI interface anytime.	
8	NC-SI HW-based packet copy enable	1b	NC-SI HW-based packet copy enable 0b = Disable. 1b = Enable.	
7:5	Package ID	0b	Package ID	
4:0	Reserved	0x0	Must be 0	

6.3.9.8 NC-SI Configuration - Offset 0x07

Bits	Name	Default	Description	Reserved
15	Read NCSI Package ID from SDP	0b	0 - Read from NVM 1 - Read from SDP	yes
14:8	Reserved		Reserved.	
7:4	EEPROM Semaphore Interval Timer	1b	Number of 10 ms ticks that firmware must wait before taking the NVM semaphore ownership again since it has sent an NC-SI command response related to an NVM update process.	Yes
3:0	Max XOFF renewal	0x0	NC-SI Flow Control MAX XOFF Renewal (# of XOFF renewals allowed). 0x0 – Disabled. Unlimited number of XOFF frames may be sent. 0x1 – Up to 2 consecutive XOFFs frames may be sent by the I350. 0x2 – Up to 3 consecutive XOFFs frames may be sent by the I350. … 0xF - Up to 0x10 consecutive XOFFs frames may be sent by the I350.	

6.3.9.9 NC-SI Hardware Arbitration Configuration - Offset 0x08

Bits	Name	Default	Description	
15:0	Token Timeout	0XFFFF	NC-SI HW-Arbitration TOKEN Timeout (in 16 ns cycles). In order to get the value if NC-SI REF_CLK cycles, this field should be multiplied by 5/4. ***Notes:*** 1. Setting value to 0 disables the timeout mechanism. 2. The timeout shall be no fewer than 32,000 REF_CLK cycles (i.e. value of field should be greater or equal to 0x9C40).	

6.3.9.10 MCTP UUID - Time Low LSB (0ffset 0x09)

The value stored in the MCTP UUID register should indicate the creation date of the image or an earlier arbitrary date.

Bits	Name	Default	Description	Reserved
15:0	time low LSB		Byte 0 & 1 of UUID as defined in DSP0236	

6.3.9.11 MCTP UUID - Time Low MSB (0ffset 0x0A)

Bits	Name	Default	Description	Reserved
15:0	time low MSB		Byte 2 & 3 of UUID as defined in DSP0236	

6.3.9.12 MCTP UUID - Time MID (0ffset 0x0B)

Bits	Name	Default	Description	Reserved
15:0	time mid		Byte 4 & 5 of UUID as defined in DSP0236	

6.3.9.13 MCTP UUID - Time High and Version (0ffset 0x0C)

Bits	Name	Default	Description	Reserved
15:0	time high and version		Byte 7 & 8 of UUID as defined in DSP0236	

6.3.9.14 MCTP UUID - Clock Seq (0ffset 0x0D)

Bits	Name	Default	Description	Reserved
15:0	Clock seq and reserved		Byte 9 & 10 of UUID as defined in DSP0236	

6.3.9.15 SMBus Slave Addresses 2 - Offset 0x0E

Bits	Name	Default	Description	Reserved
15:9	SMBus 3 Slave Address	0x4C	SMBus 3 Slave Address ***Note:*** Multi address mode only.	
8	Reserved	0b	Reserved	
7:1	SMBus 2 Slave Address	0x4B	SMBus 2 Slave Address ***Note:*** Multi address mode only.	
0	Reserved	0b	Reserved	

6.3.9.16 Alternative IANA - 0ffset 0x0F

Bits	Name	Default	Description	
15:0	Alternative IANA number	0x0	If not zero and not 0x157, the I350 will accept NC-SI OEM commands with this IANA number.	

6.3.9.17 NC-SI over MCTP Configuration - 0ffset 0x10

Bits	Name	Default	Description	
15:8	NC-SI packet type	0x2	Defines the MCTP packet type used to identify NC-SI packets	No
7	Simplified MCTP	0x0	If set, only SOM and EOM bits are used for the re-assembly process. Relevant only in SMBus mode.	No
6:0	Reserved	0x0	Reserved	No

6.3.10 Flexible TCO Filter Configuration Module (Global MNG Offset 0x07)

This module is pointed to by global offset 0x07 of the manageability control section.

6.3.10.1 Section Header — Offset 0x0

Bits	Name	Default	Description	Reserved
15:8	Block CRC8			
7:0	Block Length		Section length in words.	

6.3.10.2 Flexible Filter Length and Control — Offset 0x01

Bits	Name	Default	Description	Reserved
15:8	Flexible Filter Length (Bytes)		Flexible Filter Length in Bytes.	
7	Reserved		Reserved	
6	Apply Filter to LAN 3		Apply Filter to LAN 3 0b - Do not apply Flex Filter. 1b - Apply Flex Filter.	

Bits	Name	Default	Description	Reserved
5	Apply Filter to LAN 2		Apply Filter to LAN 2 0b - Do not apply Flex Filter. 1b - Apply Flex Filter.	
4	Last Filter	1b	Last Filter	
3:2	Filter Index (0)	0x0	Filter Index	
1	Apply Filter to LAN 1		Apply Filter to LAN 1 0b - Do not apply Flex Filter. 1b - Apply Flex Filter.	
0	Apply Filter to LAN 0		Apply Filter to LAN 0 0b - Do not apply Flex Filter. 1b - Apply Flex Filter.	

6.3.10.3 Flexible Filter Enable Mask — Offset 0x02 – 0x09

Bits	Name	Default	Description	Reserved
15:0	Flexible Filter Enable Mask		Flexible Filter Enable Mask Up to 128 Flex filter mask bits for Bytes defined in the Flexible Filter Data	

6.3.10.4 Flexible Filter Data — Offset 0x0A – Block Length

Bits	Name	Default	Description	Reserved
15:0	Flexible Filter Data		Flexible Filter Data Up to 128 Bytes of data starting at offset 0x0A.	

Note: This section loads all of the flexible filters, The control + mask + filter data are repeatable as the number of filters. Section length in offset 0 is for all filters.

6.3.11 NC-SI Configuration Module (Global MNG Offset 0x0A)

This module is pointed to by global offset 0x0A of the manageability control table.

6.3.11.1 Section Header — Offset 0x0

Bits	Name	Default	Description	Reserved
15:8	Block CRC8			
7:0	Block Length	0x9	Section length in words.	

6.3.11.2 Rx Mode Control1 (RR_CTRL[15:0]) - Offset 0x1

Bits	Name	Default	Description	Reserved
15:1	Reserved		Set to 0x0.	
0	NC-SI Loopback Enable		When set, enables NC-SI TX to RX loop. All data that is transmitted from NC-SI is returned to it. No data is actually transmitted from NC-SI.	

6.3.11.3 Rx Mode Control2 (RR_CTRL[31:16]) - Offset 0x2

Bits	Name	Default	Description	Reserved
15:0	Reserved	0x0		

6.3.11.4 Tx Mode Control1 (RT_CTRL[15:0]) - Offset 0x3

Bits	Name	Default	Description	Reserved
15:0	Reserved		Set to 0x0.	

6.3.11.5 Tx Mode Control2 (RT_CTRL[31:16]) - Offset 0x4

Bits	Name	Default	Description	Reserved
15:0	Reserved	0x0	Set to 0x0.	

6.3.11.6 MAC Tx Control Reg1 (TxCntrlReg1 (15:0]) - Offset 0x5

Bits	Name	Default	Description	Reserved
15:5	Reserved	0x0	Set to 0x0.	
4	Append_fcs		When set, computes and appends the FCS on Tx frames.	
3:0	Reserved		Reserved	

6.3.11.7 MAC Tx Control Reg2 (TxCntrlReg1 (31:16]) - Offset 0x6

Bits	Name	Default	Description	Reserved
15:0	Reserved		Reserved Should be set to 0b.	

6.3.11.8 MAC RX Buffer Size - Offset 0x7

Bits	Name	Default	Description	Reserved
15:0	RX Buffer size		Per port Buffer size allocated to Data received from the BMC in KBytes. ***Notes:*** 1. In SMBus and MCTP operating mode value must be 0x4. 2. In NCSI operating mode value should be at least 0x6 and not exceed 0x8.	

6.3.11.9 NCSI Flow Control XOFF - Offset 0x8

Bits	Name	Default	Description	Reserved
15:0	XOFF Threshold		TX Buffer watermark for sending a XOFF NC-SI flow control packet in Bytes. The *XOFF Threshold* value refers to the occupied space in the buffer. ***Notes:*** 1. Field relevant for NCSI operation mode only. 2. To support maximum packet size of 1.5 KBytes, Value programmed should be: *TX Buffer size* (refer to Section 6.3.11.8) - 3,400 Bytes a. When *TX Buffer size* is 6 KBytes value of field should be 0xAB8 (2,744 Bytes)	

6.3.11.10 NCSI Flow Control XON - Offset 0x9

Bits	Name	Default	Description	Reserved
15:0	XON Threshold		TX Buffer water mark for sending a XON NC-SI flow control packet in Bytes. The *XON Threshold* value refers to the available space in the TX buffer ***Notes:*** 1. Field relevant for NCSI operation mode only. 2. To support maximum packet size of 1.5 KBytes, Value programmed should be a positive value that equals: *TX Buffer size* (refer to Section 6.3.11.8) - *XOFF Threshold* (refer to Section 6.3.11.9) + 1536 Bytes. a. When the *TX Buffer size* is 6 KBytes and the *XOFF Threshold* is 2,744 Bytes value of field should be 0x1348 (4,936 Bytes).	

6.3.12 Traffic Type Parameters – (Global MNG Offset 0xB)

6.3.12.1 Section Header — Offset 0x0

Bits	Name	Default	Description	Reserved
15:8	Block CRC8			
7:0	Block Length	0x1	Section length in words.	

6.3.12.2 Traffic Type Data - Offset 0x1

Bits	Name	Default	Description	
15:14	Reserved		Reserved	
13:12	Port 3 traffic types	01	00b = Reserved. 01b = Network to BMC traffic only allowed through port 3. 10b = OS2BMC traffic only allowed through port 3. 11b = Both Network to BMC traffic and OS2BMC traffic allowed through port 3. ***Notes:*** 1. The traffic types defined by this field are enabled by the Manageability Mode field and the OS2BMC Capable bit in the Common Firmware Parameters 1 EEPROM word (refer to Section 6.3.7.2). 2. Field loaded to MANC.EN_BMC2NET bit and MANC.EN_BMC2OS bit of port 3 (refer to Section 8.21.4).	
11:10	Reserved		Reserved	
9:8	Port 2 traffic types	01	00b = Reserved. 01b = Network to BMC traffic only allowed through port 2. 10b = OS2BMC traffic only allowed through port 2. 11b = Both Network to BMC traffic and OS2BMC traffic allowed through port 2. ***Notes:*** 1. The traffic types defined by this field are enabled by the Manageability Mode field and the OS2BMC Capable bit in the Common Firmware Parameters 1 EEPROM word (refer to Section 6.3.7.2). 2. Field loaded to MANC.EN_BMC2NET bit and MANC.EN_BMC2OS bit of port 3 (refer to Section 8.21.4).	
7:6	Reserved		Reserved	
5:4	Port 1 traffic types	01	00b = Reserved. 01b = Network to BMC traffic only allowed through port 1. 10b = OS2BMC traffic only allowed through port 1. 11b = Both Network to BMC traffic and OS2BMC traffic allowed through port 1. ***Notes:*** 1. The traffic types defined by this field are enabled by the Manageability Mode field and the OS2BMC Capable bit in the Common Firmware Parameters 1 EEPROM word (refer to Section 6.3.7.2). 2. Field loaded to MANC.EN_BMC2NET bit and MANC.EN_BMC2OS bit of port 3 (refer to Section 8.21.4).	
3:2	Reserved		Reserved	
1:0	Port 0 traffic types	01	00b = Reserved. 01b = Network to BMC traffic only allowed through port 0. 10b = OS2BMC traffic only allowed through port 0. 11b = Both Network to BMC traffic and OS2BMC traffic allowed through port 0. ***Notes:*** 1. The traffic types defined by this field are enabled by the Manageability Mode field and the OS2BMC Capable bit in the Common Firmware Parameters 1 EEPROM word (refer to Section 6.3.7.2). 2. Field loaded to MANC.EN_BMC2NET bit and MANC.EN_BMC2OS bit of port 3 (refer to Section 8.21.4).	

6.3.13 PHY Configuration Pointer – (Global MNG Offset 0xF)

Bit	Name	Description
15:0	Pointer	Pointer to PHY configuration structure. Refer to Section 6.3.13.1 for details of the structure. A value of 0xFFFF means the pointer is invalid.

6.3.13.1 PHY Configuration Structure

This section describes the PHY auto configuration structure used to configure PHY related circuitry. The programming in this section is applied after each PHY reset.

The PHY Configuration Pointer Global MNG Offset 0xF) points to the start (offset 0x0) of this type of structure, to configure PHY registers (Internal and External PHYs). If pointer is 0xFFFF then no structure exists.

Table 6-11 PHY Auto Configuration Structure Format

Offset	High Byte[15:8]	Low Byte[7:0]	Section
0x0	Section Length = 2*n (n – number of registers to configure)		Section 6.3.13.1.1
0x1	Block CRC8		Section 6.3.13.1.2
0x2	PHY number and PHY register address		Section 6.3.13.1.3
0x3	PHY data (MDIC[15:0] or I2CCMD[15:0])		Section 6.3.13.1.4
	…		
2*n	PHY number and PHY register address		Section 6.3.13.1.3
2*n + 1	PHY data (MDIC[15:0] or I2CCMD[15:0])		Section 6.3.13.1.4

6.3.13.1.1 PHY Configuration Section Length - Offset 0x0

The section length word contains the length of the section in words. Note that section length count does not include the section length word and Block CRC8 word.

Bits	Name	Default	Description
15	Reserved		
14:0	Section_length		Section length in words.

6.3.13.1.2 Block CRC8 (Offset 0x1)

Bit	Name	Description
15:8	Reserved	
7:0	CRC8	

6.3.13.1.3 PHY Number and PHY Register Address - (Offset 2*n; [n = 1... Section Length])

Bits	Name	Default	Description
15:12	Reserved	0x0	Reserved
11	Apply to port 3		If set, apply to programming when the PHY of port three is reset.
10	Apply to port 2		If set, apply to programming when the PHY of port two is reset.
9	Apply to port 1		If set, apply to programming when the PHY of port one is reset.
8	Apply to port 0		If set, apply to programming when the PHY of port zero is reset.
7:0	PHY register address		PHY register address to which the data is written. See Section 8.2.4 and Section 8.17.8 for information on the *MDIC* and *I2CCMD* registers respectively. ***Note:*** 5 LSB bits define register address when access is via the *MDIC* register.

6.3.13.1.4 PHY data (Offset 2*n + 1; [n = 1... Section Length])

Bits	Name	Default	Description
15:0	Reg_Data		*MDIC[15:0]/I2CCMD[15:0]* value (Data). See Section 8.2.4 and Section 8.17.8 for information *MDIC* and *I2CCMD* registers respectively.

6.3.14 Thermal Sensor Configuration Pointer – (Global MNG Offset 0x10)

Bit	Name	Description
15:0	Pointer	Pointer to Thermal Sensor configuration structure. Refer to Section 6.3.14.1 for details of the structure. A value of 0xFFFF means the pointer is invalid.

6.3.14.1 Thermal Sensor Configuration Structure

This section describes the PHY auto configuration structure used to configure Thermal Sensor related circuitry. The programming in this section is applied after power-up or Thermal Sensor reset.

The Thermal Sensor Configuration Pointer – (Global MNG Offset 0x10) points to the start (offset 0x0) of this type of structure, to configure Thermal Sensor registers. If pointer is 0xFFFF then no structure exists.

Table 6-12 Thermal Sensor Auto Configuration Structure Format

Offset	High Byte[15:8]	Low Byte[7:0]	Section
0x0	Section Length = 2*n (n – number of registers to configure)		Section 6.3.14.1.1
0x1	Block CRC8		Section 6.3.14.1.2
0x2	Thermal Sensor register address		Section 6.3.14.1.3
0x3	Thermal Sensor data		Section 6.3.14.1.4

Table 6-12 Thermal Sensor Auto Configuration Structure Format

Offset	High Byte[15:8]	Low Byte[7:0]	Section
	…		
2*n	Thermal Sensor register address		Section 6.3.14.1.3
2*n + 1	Thermal Sensor data		Section 6.3.14.1.4

6.3.14.1.1 Thermal Sensor Configuration Section Length - Offset 0x0

The section length word contains the length of the section in words. Note that section length count does not include the section length word and Block CRC8 word.

Bits	Name	Default	Description
15	Reserved		
14:0	Section_length		Section length in words.

6.3.14.1.2 Block CRC8 (Offset 0x1)

Bit	Name	Description
15:8	Reserved	
7:0	CRC8	

6.3.14.1.3 Thermal Sensor Register Address - (Offset 2*n; [n = 1… Section Length])

Bits	Name	Default	Description
15:5	Reserved	0x0	Reserved
4:0	Thermal Sensor register address		Thermal Sensor register address to which the data is written.

6.3.14.1.4 Thermal Sensor data (Offset 2*n + 1; [n = 1… Section Length])

Bits	Name	Default	Description
15:0	Reg_Data		Thermal Sensor register data.

6.4 Software Accessed Words

Words 0x03 to 0x07 in the EEPROM image are reserved for compatibility information. New bits within these fields will be defined as the need arises for determining software compatibility between various hardware revisions.

Words 0x8 and 0x09 are used to indicate the Printed Board Assembly (PBA) number and words 0x42 and 0x43 identifies the EEPROM image.

Words 0x30 to 0x3E have been reserved for configuration and version values to be used by PXE code. The only exceptions are words 0x36 and 0x3D which are used for the iSCSI boot configuration and word 0x37 which is used as a pointer to the Alternate MAC address.

6.4.1 Compatibility (Word 0x03)

Bit	Description
15:12	Reserved (set to 0000b).
11	LOM/Not a LOM. 0b = NIC. 1b = LOM.
10	Reserved (set to 0b)
9	Client/Not a Client NIC. 0b = Server. 1b = Client.
8:0	Reserved (set to 0x0)

6.4.2 Port Identification LED blinking (Word 0x04)

Driver software provides a method to identify an external port on a system through a command that causes the LEDs to blink. Based on the setting in word 0x4, the LED drivers should blink between STATE1 and STATE2 when a port identification command is issued.

When word 0x4 is equal to 0xFFFF or 0x0000, the blinking behavior reverts to a default.

Bit	Description
15:12	Control for LED 3 0000b or 1111b: Default LED Blinking operation is used. 0001b = Default in STATE1 + Default in STATE2. 0010b = Default in STATE1 + LED is ON in STATE2. 0011b = Default in STATE1 + LED is OFF in STATE2. 0100b = LED is ON in STATE1 + Default in STATE2. 0101b = LED is ON in STATE1 + LED is ON in STATE2. 0110b = LED is ON in STATE1 + LED is OFF in STATE2. 0111b = LED is OFF in STATE1 + Default in STATE2. 1000b = LED is OFF in STATE1 + LED is ON in STATE2. 1001b = LED is OFF in STATE1 + LED is OFF in STATE2. All other values are Reserved.
11:8	Control for LED 2 – same encoding as for LED 3.
7:4	Control for LED 1 – same encoding as for LED 3.
3:0	Control for LED 0 – same encoding as for LED 3.

6.4.3 EEPROM Image Revision (Word 0x05)

This word is valid only for device starter images and indicates the ID and version of the EEPROM image.

Bit	Description
15:12	EEPROM major version.
11:8	Reserved
7:0	EEPROM minor version.

Note: The value shown is the decimal representation of the actual version. For example: 0x1044 = v1.44

6.4.4 OEM Specific (Word 0x06, 0x07)

These words are available for OEM use.

6.4.5 PBA Number/Pointer (Word 0x08, 0x09)

The nine-digit Printed Board Assembly (PBA) number used for Intel manufactured Network Interface Cards (NICs).

Current PBA numbers have exceeded the length that can be stored as hex values in these two words. For these PBA numbers the high word is a flag (0xFAFA) indicating that the PBA is stored in a separate PBA Block. The low word is a pointer to a PBA block.

PBA Number	Word 0x8	Word 0x9
G23456-003	FAFA	Pointer to PBA Block

The PBA Block is pointed to by word 0x9:

Word Offset	Description	End User Reserved
0x0	Length in words of the PBA Block (default is 0x6)	No
0x1.. 0x5	PBA Number stored in hexadecimal ASCII values.	No

The PBA block contains the complete PBA number including the dash and the first digit of the 3-digit suffix.

Example:

PBA Number	Word Offset 0	Word Offset 1	Word Offset 2	Word Offset 3	Word Offset 4	Word Offset 5
G23456-003	0006	4732	3334	3536	2D30	3033

For older products, the PBA number is stored directly in words 0x8/0x9. In this case, the PBA is stored in a four-byte field. The dash itself is not stored nor is the first digit of the 3-digit suffix, as it is always zero for the affected products.

Example:

Product	PWA Number	Byte 1	Byte 2	Byte 3	Byte 4
Example	123456-003	12	34	56	03

Note that through the course of hardware ECOs, the suffix field (byte 4/6) increments. The purpose of this information is to allow customer support (or any user) to identify the exact revision level of a product. Network driver software should not rely on this field to identify the product or its capabilities.

Note: This PBA number is not related to the MSI-X Pending Bit Array (PBA).

6.4.6 PXE Configuration Words (Word 0x30:3B)

PXE configuration is controlled by the following words.

6.4.6.1 Setup Options PCI Function 0 (Word 0x30)

The main setup options are stored in word 0x30. These options are those that can be changed by the user via the Control-S setup menu. Word 0x30 has the following format:

Bit(s)	Name	Function
15:13	RFU	Reserved. Must be 0.
12:10	FSD	Bits 12-10 control forcing speed and duplex during driver operation. 000b – Auto-negotiate 001b – 10Mbps Half Duplex 010b – 100Mbps Half Duplex 011b – Not valid (treated as 000b) 100b – Not valid (treated as 000b) 101b – 10Mbps Full Duplex 110b – 100Mbps Full Duplex 111b – 1000Mbps Full Duplex Default value is 000b.
9	RSV	Reserved. Set to 0.
9	RFU	Reserved. Must be 0.
8	DSM	Display Setup Message. If the bit is set to 1, the "Press Control-S" message is displayed after the title message. Default value is 1.
7:6	PT	Prompt Time. These bits control how long the CTRL-S setup prompt message is displayed, if enabled by DIM. 00 = 2 seconds (default) 01 = 3 seconds 10 = 5 seconds 11 = 0 seconds Note: CTRL-S message is not displayed if 0 seconds prompt time is selected.

Bit(s)	Name	Function
5	DEP	Deprecated. Must be 0.
4:3	DBS	Default Boot Selection. These bits select which device is the default boot device. These bits are only used if the agent detects that the BIOS does not support boot order selection or if the MODE field of word 0x31 is set to MODE_LEGACY. 00 = Network boot, then local boot (default) 01 = Local boot, then network boot 10 = Network boot only 11 = Local boot only
2:0	PS	Protocol Select. See Table 6-13 for details of this field.

Table 6-13 Protocol Select Field Description

Value	Port Status	CLP (Combo) Executes	iSCSI Boot Option ROM CTRL-D Menu	FCoE Boot Option ROM CTRL-D Menu[1]
0	PXE	PXE	Displays port as PXE. Allows changing to Boot Disabled, iSCSI Primary or Secondary	Displays port as PXE. Allows changing to Boot Disabled, FCoE enabled
1	Boot Disabled	NONE	Displays port as Disabled. Allows changing to iSCSI Primary/ Secondary	Displays port as Disabled. Allows changing to FCoE enabled
2	iSCSI Primary	iSCSI	Displays port as iSCSI Primary. Allows changing to Boot Disabled, iSCSI Secondary	Displays port as iSCSI. Allows changing to Boot Disabled, FCoE enabled
3	iSCSI Secondary	iSCSI	Displays port as iSCSI Secondary. Allows changing to Boot Disabled, iSCSI Primary	Displays port as iSCSI. Allows changing to Boot Disabled, FCoE enabled
4	FCoE	FCOE	Displays port as FCoE.Allows changing to port to Boot Disabled, iSCSI Primary or Secondary	Displays port as FCoE.Allows changing to Boot Disabled
5-7	Reserved.	Same as Disabled	Same as Disabled	Same as Disabled

1. FcoE Boot ROM is N/A for the I350.

6.4.6.2 Configuration Customization Options PCI Function 0 (Word 0x31)

Word 0x31 of the EEPROM contains settings that can be programmed by an OEM or network administrator to customize the operation of the software. These settings cannot be changed from within the Control-S setup menu. The lower byte contains settings that would typically be configured by a network administrator using an external utility; these settings generally control which setup menu options are changeable. The upper byte is generally settings that would be used by an OEM to control the operation of the agent in a LOM environment, although there is nothing in the agent to prevent their use on a NIC implementation. The default value for this word is 4000h.

Bit(s)	Name	Function
15:14	SIG	Signature. Must be set to 01 to indicate that this word has been programmed by the agent or other configuration software.
13	RFU	Reserved. Must be 0.
12	RFU	Reserved. Must be 0.

Bit(s)	Name	Function
11	RETRY	Selects Continuous Retry operation. If this bit is set, IBA will NOT transfer control back to the BIOS if it fails to boot due to a network error (such as failure to receive DHCP replies). Instead, it will restart the PXE boot process again. If this bit is set, the only way to cancel PXE boot is for the user to press ESC on the keyboard. Retry will not be attempted due to hardware conditions such as an invalid EEPROM checksum or failing to establish link. Default value is 0.
10:8	MODE	Selects the agent's boot order setup mode. This field changes the agent's default behavior in order to make it compatible with systems that do not completely support the BBS and PnP Expansion ROM standards. Valid values and their meanings are: 000b Normal behavior. The agent will attempt to detect BBS and PnP Expansion ROM support as it normally does. 001b Force Legacy mode. The agent will not attempt to detect BBS or PnP Expansion ROM supports in the BIOS and will assume the BIOS is not compliant. The user can change the BIOS boot order in the Setup Menu. 010b Force BBS mode. The agent will assume the BIOS is BBS-compliant, even though it may not be detected as such by the agent's detection code. The user can NOT change the BIOS boot order in the Setup Menu. 011b Force PnP Int18 mode. The agent will assume the BIOS allows boot order setup for PnP Expansion ROMs and will hook interrupt 18h (to inform the BIOS that the agent is a bootable device) in addition to registering as a BBS IPL device. The user can NOT change the BIOS boot order in the Setup Menu. 100b Force PnP Int19 mode. The agent will assume the BIOS allows boot order setup for PnP Expansion ROMs and will hook interrupt 19h (to inform the BIOS that the agent is a bootable device) in addition to registering as a BBS IPL device. The user can NOT change the BIOS boot order in the Setup Menu. 101b Reserved for future use. If specified, is treated as a value of 000b. 110b Reserved for future use. If specified, is treated as a value of 000b. 111b Reserved for future use. If specified, is treated as a value of 000b.
7	RFU	Reserved. Must be 0.
6	RFU	Reserved. Must be 0.
5	DFU	Disable Flash Update. If this bit is set to 1, the user is not allowed to update the flash image using PROSet. Default value is 0.
4	DLWS	Disable Legacy Wakeup Support. If this bit is set to 1, the user is not allowed to change the Legacy OS Wakeup Support menu option. Default value is 0.
3	DBS	Disable Boot Selection. If this bit is set to 1, the user is not allowed to change the boot order menu option. Default value is 0.
2	DPS	Disable Protocol Select. If set to 1, the user is not allowed to change the boot protocol. Default value is 0.
1	DTM	Disable Title Message. If this bit is set to 1, the title message displaying the version of the Boot Agent is suppressed; the Control-S message is also suppressed. This is for OEMs who do not wish the boot agent to display any messages at system boot. Default value is 0.
0	DSM	Disable Setup Menu. If this bit is set to 1, the user is not allowed to invoke the setup menu by pressing Control-S. In this case, the EEPROM may only be changed via an external program. Default value is 0.

6.4.6.3 PXE Version (Word 0x32)

Word 0x32 of the EEPROM is used to store the version of the boot agent that is stored in the flash image. When the Boot Agent loads, it can check this value to determine if any first-time configuration needs to be performed. The agent then updates this word with its version. Some diagnostic tools also read this word to report the version of the PXE Boot Agent in the flash.

The format of this word is:

Bit(s)	Name	Function
15 - 12	MAJ	PXE Boot Agent Major Version.
11 – 8	MIN	PXE Boot Agent Minor Version.
7 – 0	BLD	PXE Boot Agent Build Number.

6.4.6.4 Flash (Option ROM) Capabilities (Word 0x33)

Word 0x33 of the EEPROM is used to enumerate the boot technologies that have been programmed into the flash. This is updated by flash configuration tools and is not updated or read by IBA.

Bit(s)	Name	Function
15 - 14	SIG	Signature. Must be set to 01 to indicate that this word has been programmed by the agent or other configuration software.
13 – 5	RFU	Reserved. Must be 0.
4	ISCSI	iSCSI Boot is present in flash if set to 1.
3	UEFI	UEFI UNDI driver is present in flash if set to 1.
2	RPL	Reserved. Must be 0.
1	UNDI	PXE UNDI driver is present in flash if set to 1.
0	BC	PXE Base Code is present in flash if set to 1.

6.4.6.5 Setup Options PCI Function 1 (Word 0x34)

This word is the same as word 0x30, but for function 1 of the device.

6.4.6.6 Configuration Customization Options PCI Function 1 (Word 0x35)

This word is the same as word 0x31, but for function 1 of the device.

6.4.6.7 Setup Options PCI Function 2 (Word 0x38)

This word is the same as word 0x30, but for function 2 of the device.

6.4.6.8 Configuration Customization Options PCI Function 2 (Word 0x39)

This word is the same as word 0x31, but for function 2 of the device.

6.4.6.9 Setup Options PCI Function 3 (Word 0x3A)

This word is the same as word 0x30, but for function 3 of the device.

6.4.6.10 Configuration Customization Options PCI Function 3 (Word 0x3B)

This word is the same as word 0x31, but for function 3 of the device.

6.4.6.11 PXE VLAN Configuration Pointer (0x003C)

Bits	Name	Default	Description
15:0	PXE VLAN Configu-ration Pointer	0x0	The pointer contains offset of the first Flash word of the PXE VLAN config block.

6.4.6.12 PXE VLAN Configuration Section Summary Table

Word Offset	Word Name	Description
0x0000	VLAN Block Signature	ASCII 'V', 'L'.
0x0001	Version and Size	Contains version and size of structure.
0x0002	Port 0 VLAN Tag	VLAN tag value for the first port of the I350. Contains PCP, CFI and VID fields. A value of 0 means no VLAN is configured for this port.
0x0003	Port 1 VLAN Tag	VLAN tag value for the second port of the I350. Contains PCP, CFI and VID fields. A value of 0 means no VLAN is configured for this port.
0x0004	Port 2 VLAN Tag	VLAN tag value for the third port of the I350. Contains PCP, CFI and VID fields. A value of 0 means no VLAN is configured for this port.
0x0005	Port 3 VLAN Tag	VLAN tag value for the fourth port of the I350. Contains PCP, CFI and VID fields. A value of 0 means no VLAN is configured for this port.

6.4.6.13 VLAN Block Signature - 0x0000

Bits	Field Name	Default	Description
15:0	VLAN Block Signature	0x4C56	ASCII 'V', 'L'.

6.4.6.14 Version and Size - 0x0001

Bits	Field Name	Default	Description
15:8	Size		Total size in bytes of section.
7:0	Version	0x01	Version of this structure. Should be set to 0x1.

6.4.6.15 Port 0 VLAN Tag - 0x0002

Bits	Field Name	Default	Description
15:13	Priority (0-7)	0x0	Priority 0-7.
12	Reserved	0x0	Always 0.
11:0	VLAN ID (1- 4095)	0x0	VLAN ID (1-4095).

6.4.6.16 Port 1 VLAN Tag - 0x0003

Bits	Field Name	Default	Description
15:13	Priority (0-7)	0x0	Priority 0-7.
12	Reserved	0x0	Always 0.
11:0	VLAN ID (1- 4095)	0x0	VLAN ID (1-4095).

6.4.6.17 Port 2 VLAN Tag - 0x0004

Bits	Field Name	Default	Description
15:13	Priority (0-7)	0x0	Priority 0-7.
12	Reserved	0x0	Always 0.
11:0	VLAN ID (1- 4095)	0x0	VLAN ID (1-4095).

6.4.6.18 Port 3 VLAN Tag - 0x0005

Bits	Field Name	Default	Description
15:13	Priority (0-7)	0x0	Priority 0-7.
12	Reserved	0x0	Always 0.
11:0	VLAN ID (1- 4095)	0x0	VLAN ID (1-4095).

6.4.7 iSCSI Boot Words

6.4.7.1 iSCSI Option ROM Version (Word 0x36)

Word 0x36 is used to store the version of the iSCSI Boot option ROM if present. Values below 0x2000 are reserved and should not be used. This word may be modified by flash update utilities.

6.4.7.2 iSCSI boot Configuration Pointer (Word 0x3D)

Bit	Name	Description
15:0	iSCSI Address	iSCSI Configuration Block EEPROM Offset Offset of iSCSI configuration block from the start of the EEPROM. If set to 0000h or FFFFh there is no EEPROM configuration data available for the iSCSI adapter. In this case configuration data must be provided by the BIOS through the SM CLP interface.

6.4.7.3 iSCSI Module Structure

The table below defines the layout of the iSCSI boot configuration block stored in EEPROM. EEPROM word 0x3D described above stores the offset within the EEPROM of the configuration block. Software must first read word 0x3D to determine the offset of the configuration table before attempting to read or write the configuration block.

The strings defined below are stored in UTF-8 encoding and NULL terminated. All data words are stored in little-endian (Intel) byte order.

Configuration Item	Offset (Bytes)	Size in Bytes	Comments
Boot Signature	0x1:0x0	2	0x5369 ('i', 'S')
Block Size	0x3:0x2	2	The structure size is stored in this field and is set depending on the amount of free EEPROM space available. The total size of this structure, including variable length fields, must fit within this space. 0x0384 - single port 0x05E0 - dual port 0x0A98 - quad port
Structure Version	0x4	1	Version of this structure. Should be set to one.
Reserved	0x5	1	Reserved for future use, should be set to zero.
iSCSI Initiator Name	0x105:0x6	255 + 1	iSCSI Initiator Name - This field is optional and can also be built by DHCP.
iSCSI Configuration Block	0x107:0x106	2	Bits 15:8 (Major) - Combo image major version. Bits 7:0 (Build) - Combo image build number (15:8).
	0x109:0x108	2	Bits 15:8 (Build) - Combo image build number (7:0). Bits 7:0 (Minor) - Combo image minor version.
Reserved	0x127:0x10A	30	Reserved for future use, should be set to zero.

Configuration Item	Offset (Bytes)	Size in Bytes	Comments
Below fields are per port			
iSCSI Flags	0x129:0x128	2	Bit 0 ⇨ Enable DHCP 0 - Use static configurations from this structure 1 - Overrides configurations retrieved from DHCP. Bit 01h ⇨ Enable DHCP for getting iSCSI target information. 0 - Use static target configuration 1 - Use DHCP to get target information. Bit 02h - 03h ⇨ Authentication Type 00 - none 01 - one way chap 02 - mutual chap Bit 04h - 05h ⇨ Ctrl-D setup menu 00 - enabled 03 - disabled Bit 06h - 07h ⇨ Reserved Bit 08h - 09h ⇨ ARP Retries Retry value Bit 0Ah - 0Fh ⇨ ARP Timeout Timeout value for each retry
iSCSI Initiator IP	0x12D:0x12A	4	DHCP flag not set ⇨ This field should contain the configured IP address. DHCP flag set ⇨ If DHCP bit is set this field is ignored.
Initiator Subnet Mask	0x131:0x12E	4	DHCP flag not set ⇨ This field should contain the configured subnet mask. DHCP flag set ⇨ If DHCP bit is set this field is ignored.
Initiator Gateway IP	0x135:0x132	4	DHCP flag not set ⇨ This field should contain the configured gateway DHCP flag set ⇨ If DHCP bit is set this field is ignored.
iSCSI Boot LUN	0x137:0x136	2	DHCP flag not set ⇨ Target LUN that Initiator will be attached to. DHCP flag set ⇨ If DHCP bit is set this field is ignored.
iSCSI Target IP	0x13B:0x138	4	DHCP flag not set ⇨ IP address of iSCSI target. DHCP flag set ⇨ If DHCP bit is set this field is ignored.
iSCSI Target Port	0x13D:0x13C	2	DHCP flag not set ⇨ IP port of iSCSI target. Default is 3260. DHCP flag set ⇨ If DHCP bit is set this field is ignored
iSCSI Target Name	0x23D:0x13E	255 + 1	DHCP flag set ⇨ If DHCP bit is set this field is ignored
CHAP Password	0x24F:0x23E	16 + 2	The minimum CHAP secret must be 12 octets and maximum CHAP secret size is 16. 1 byte is reserved for alignment padding and 1 byte for null.
CHAP User Name	0x2CF:0x250	127 + 1	The user name must be non-null value and maximum size of user name allowed is 127 characters.
Vlan ID	0x2D1:0x2D0	2	Vlan Id to be used for iSCSI boot traffic. a valid Vlan ID is between 1 and 4094
Mutual CHAP Password	0x2E3:0x2D2	16 + 2	The minimum mutual CHAP secret must be 12 octets and maximum CHAP secret size is 16. 1 byte is reserved for alignment padding and 1 byte for null.
Reserved	0x323:0x2E4	64	Reserved for FCoE - not relevant in the I350 - should be set to zero.
Reserved	0x383:0x324	96	Reserved for future use, should be set to zero.
Port 1 Configuration			
Port 1 Configuration	0x5DF:0x384		Same configuration as port 0. Add to each offset in port 0, 0x25C
Port 2 Configuration			
Port 2 Configuration	0x83B:0x560		Same configuration as port 0. Add to each offset in port 0, 0x4B8
Port 3 Configuration			
Port 3 Configuration	0xA97:0x83C		Same configuration as port 0. Add to each offset in port 0, 0x714

6.4.8 Alternate MAC address pointer (Word 0x37)

This word may point to a location in the EEPROM containing additional MAC addresses used by system management functions. If the additional MAC addresses are not supported, the word must be set to 0xFFFF. The structure of the alternate MAC address block can be found in Table 6-14.

Table 6-14 Alternate MAC Address Block

Word Offset	Description[1]
0x0...0x2	Alternate MAC Address for LAN port assigned to PCI function 0
0x3...0x5	Alternate MAC Address for LAN port assigned to PCI function 1
0x6...0x8	Alternate MAC Address for LAN port assigned to PCI function 2
0x9...0xB	Alternate MAC Address for LAN port assigned to PCI function 3

1. An alternate MAC Address value of 0xFFFF-FFFF-FFFF means that no alternate MAC address is present for the port.

6.4.9 Reserved/3rd Party External Thermal Sensor – (Word 0x3E)

Bits	Name	Default	Description
15:0	External Thermal Sensor	0xFFFF	Pointer to 3rd Party External Thermal Sensor Configuration block. 0x0000 and 0xFFFF indicates an invalid pointer.

6.4.9.1 3rd Party External Thermal Sensor Configuration NVM Block

6.4.9.1.1 External Thermal Sensor Configuration Block

Word Offset	Description
0x0	ETS General configuration word
0x1	Sensor Data – Sensor 1
…	(all sensor data)
N	Sensor data – Sensor N (N is max of 4)

6.4.9.1.2 ETS General Configuration Word – Offset 0x0

This word contains general information about the external thermal sensor.

Bits	Name	Description
15:10	Reserved	Reserved (default 0x0)

10:6	Low Threshold Delta	The delta from the sensors High Threshold in °C. Used to calculate the Low Threshold value for the thermal sensors. Low Threshold = High Threshold – Low Threshold Delta
5:3	Sensor type	000b = EMC1413 – I2C 001b = Reserved
2:0	Num Sensors	The number of **supported** external thermal sensors. This is **not** necessarily all the physical thermal sensors on the device. This must match the number of Sensor Data words that follow.

6.4.9.1.3 Sensor Data – Offset 0x1 – 0xN (N is max of 4)

The Sensor Data blocks contain sensor specific data for each external thermal sensor.

Bits	Name	Description
15:14	Reserved	Reserved (default 0x0)
13:10	Location	Thermal sensor location to be reported to end user. 0000b = Not Applicable 0001b = Reserved 0010b = Hot Spot (near MAC) 0011b = On board near PCIe connector (NIC only) 0100b = On board near bulkhead connector (NIC only) 0101b = On board other 0110b = Reserved 0111b = Inlet ambient on blade server add-in/mezz card. 1000b -1111b = Reserved **Note:** 0000b indicates that the sensor requires it's thresholds to be initialized but the sensor should not be reported to the end user.
9:8	Sensor Index	Thermal Sensor Index 00b = Internal Sensor 01b = External Diode 1 10b = External Diode 2 11b = External Diode 3
7:0	High Threshold	8-bit critical temperature limit for current sensor in °C. Value between 0°C - 250°C.

6.4.10 Checksum Word (Offset 0x3F)

The checksum words (Offset 0x3F from start of Common, LAN 1, LAN 2 and LAN 3 sections) are used to ensure that the base EEPROM image is a valid image. The value of this word should be calculated such that after adding all the words (0x00:0x3E), including the checksum word itself, the sum should be 0xBABA. The initial value in the 16-bit summing register should be 0x0000 and the carry bit should be ignored after each addition.

Note: Hardware does not calculate the checksum word during EEPROM write; it must be calculated by software independently and included in the EEPROM write data. Hardware does not compute a checksum over words 0x00:0x3F during EEPROM reads in order to determine validity of the EEPROM image; this field is provided strictly for software verification of EEPROM validity. All hardware configurations based on word 0x00:0x3F content is based on the validity of the *Signature* field of the *EEPROM Sizing & Protected Fields* EEPROM word (*Signature* must be 01b).

6.4.11 Image Unique ID (Word 0x42, 0x43)

These words contain a unique 32-bit ID for each image generated by Intel to enable tracking of images and comparison to the original image if testing a customer EEPROM image.

NOTE: ***This page intentionally left blank.***

§ §

7 Inline Functions

7.1 Receive Functionality

Typically, packet reception consists of recognizing the presence of a packet on the wire, performing address filtering, storing the packet in the receive data FIFO, transferring the data to one of the 8 receive queues in host memory, and updating the state of a receive descriptor.

A received packet goes through three stages of filtering as shown in Figure 7-1. Figure 7-1 describes a switch-like structure that is used in virtualization mode to route packets between the network port (top of drawing) and one or more virtual ports (bottom of figure), where each virtual port can be associated with a virtual machine, a VMM or any other software entity.

The first step in queue assignment is to verify that the packet is destined to the port. This is done by a set of L2 filters as described in Section 7.1.3.

The second stage is specific to virtualization environments and defines the virtual ports (called pools) that are the targets for the Rx packet. A packet can be associated with any number of ports/pools using a selection process described in Section 7.1.2.2.

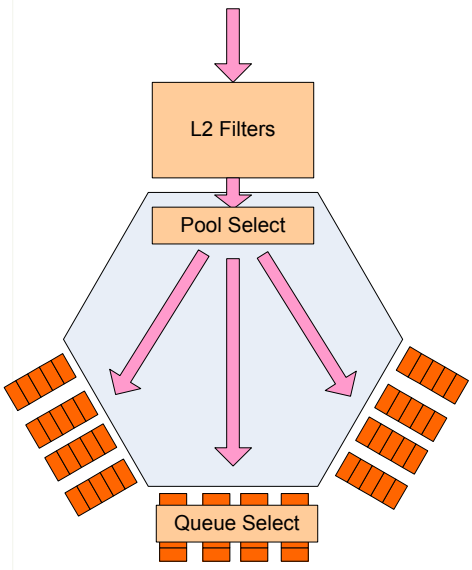


Figure 7-1 Stages in Packet Filtering

In the third stage, relevant only to non virtualized cases, a received packet that successfully passed the Rx filters is associated with one or more receive descriptor queues as described in Section 7.1.1. In the virtualized case, the queue is fixed only according to the pool.

7.1.1 L2 Packet Filtering

The receive packet filtering role is to determine which of the incoming packets are allowed to pass to the local system and which of the incoming packets should be dropped since they are not targeted to the local system. Received packets can be destined to the host, to a manageability controller (BMC), or to both. This section describes how host filtering is done, and the interaction with management filtering.

As shown in Figure 7-2, host filtering has three stages:

1. Packets are filtered by L2 filters (MAC address, unicast/multicast/broadcast). See Section 7.1.1.1 for details.
2. Packets are then filtered by VLAN if a VLAN tag is present. See Section 7.1.1.2 for details.
3. Packets are filtered by the manageability filters (port, IP, flex, other). See Section 10.3.3 for details.

A packet is not forwarded to the host if any of the following takes place:

1. The packet does not pass MAC address filters as described later in this section.
2. The packet does not pass VLAN filtering as described later in this section.
3. The packet passes manageability filtering and then the manageability filters determine that the packet should be sent only to the BMC (see Section 10.3 and the *MNGONLY* register).

A packet that passes receive filtering as previously described might still be dropped due to other reasons. Normally, only good packets are received. These are defined as those packets with no Under Size Error, Over Size Error (see Section 7.1.1.4), Packet Error, Length Error and CRC Error are detected. However, if the *store-bad-packet* bit is set (*RCTL.SBP*), then bad packets that pass the filter function are stored in host memory. Packet errors are indicated by error bits in the receive descriptor (*RDESC.ERRORS*). It is possible to receive all packets, regardless of whether they are bad, by setting the promiscuous enabled (Unicast and Multicast) and the *store-bad-packet* bits in the *RCTL* register.

If there is insufficient space in the receive FIFO, hardware drops the packet and indicates the missed packet in the appropriate statistics registers.

When the packet is routed to a queue with the *SRRCTL.Drop_En* bit set to 1, receive packets are dropped when insufficient receive descriptors exist to write the packet into system memory.

Note: CRC errors before the SFD are ignored. Any packet must have a valid SFD in order to be recognized by the I350 (even bad packets).

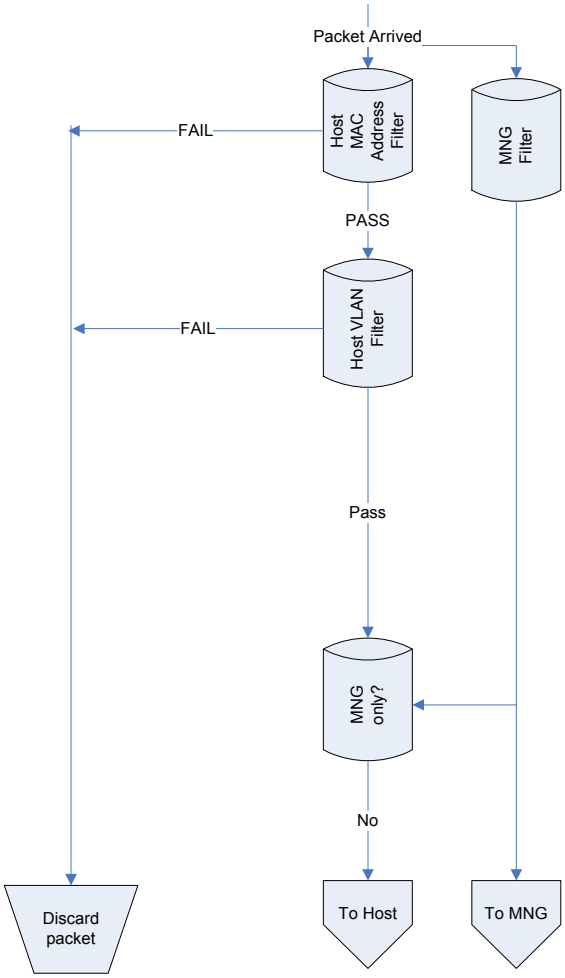


Figure 7-2 Receive Filtering Flow Chart

7.1.1.1 MAC Address Filtering

Figure 7-3 shows the MAC address filtering. A packet passes successfully through the MAC address filtering if any of the following conditions are met:

1. It is a unicast packet and promiscuous unicast filtering is enabled.
2. It is a multicast packet and promiscuous multicast filtering is enabled.
3. It is a unicast packet and it matches one of the unicast MAC filters.
4. It is a multicast packet and it matches one of the multicast filters.
5. It is a broadcast packet and Broadcast Accept Mode (*RCTL.BAM*) is enabled.

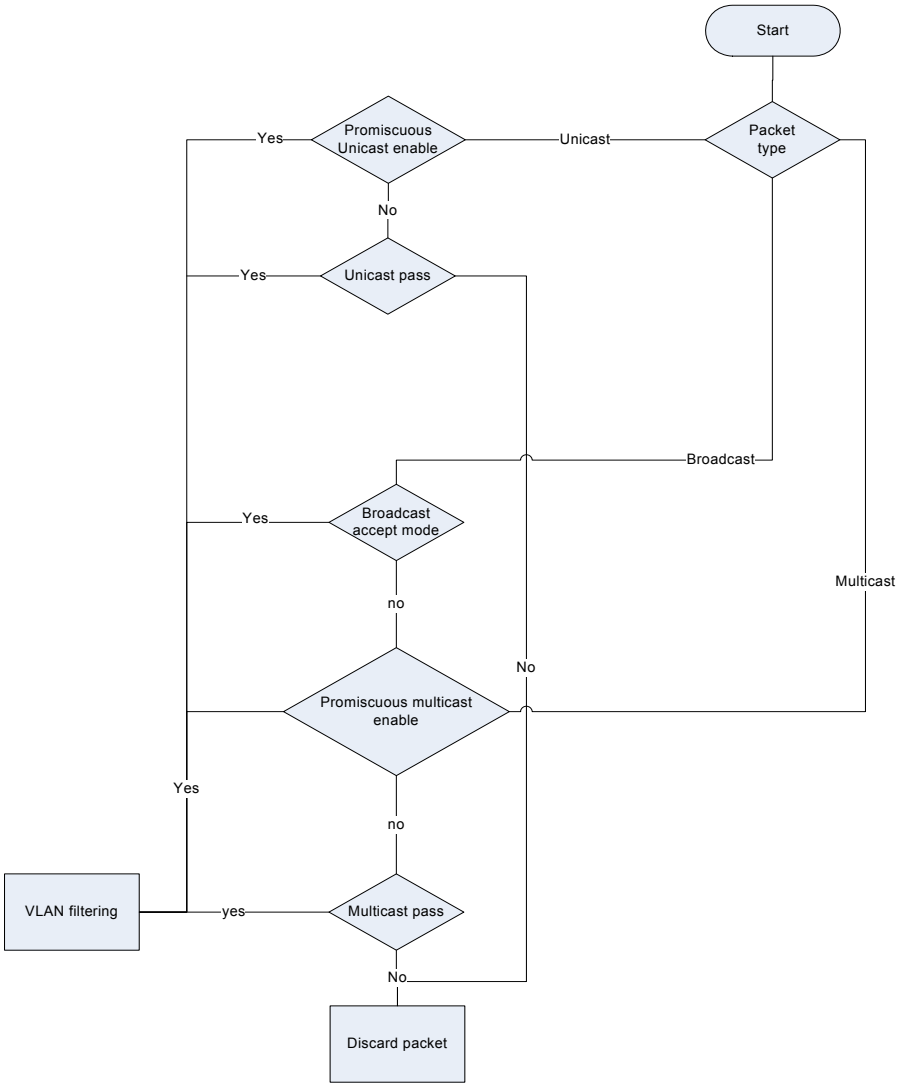


Figure 7-3 Host MAC Address Receive Filtering Flow Chart

7.1.1.1.1 Unicast Filter

The entire MAC address is checked against the 32 host unicast addresses. The 32 host unicast addresses are controlled by the host interface (the BMC must not change them). The other 4 addresses are dedicated to management functions and are only accessed by the BMC. The destination address of incoming packet must exactly match one of the pre-configured host address filters. These addresses can be unicast or multicast. Those filters are configured through *RAL*, and *RAH* registers.

Promiscuous Unicast — Receive all unicasts. Promiscuous unicast mode in the *RCTL* register can be set/cleared only through the host interface (not by the BMC). This mode is usually used when I350 is used as a sniffer.

Unicast Hash Table — Destination address matching the Unicast Hash Table (*UTA*). In this case if the packet only matches to the Unicast Hash Table (*UTA*) the *PIF* bit in the receive descriptor is set to 1b (See Section 7.1.4.1) and Software needs to examine the packet to verify that it's destined to the station.

7.1.1.1.2 Multicast Filter (Inexact)

A 12-bit portion of incoming packet multicast address must exactly match Multicast Filter Address (*MFA*) in order to pass multicast filtering. Which 12 bits out of 48 bits of the destination address are used can be selected by the *MO* field of *RCTL* (Section 8.10.1). The 12 bits extracted from the Multicast Destination address are used as an address for a bit in the Multicast Table Array (MTA). If the value of the bit selected in the MTA table is 1b, the packet is sent to the Host (See Section 8.10.15). These entries can be configured only by the host interface and cannot be controlled by the BMC. Packets received according to this mode have the *PIF* bit in the descriptor set to indicate imperfect filtering that should be validated by the software device driver.

Promiscuous Multicast — Receive all multicast. Promiscuous multicast mode can be set/cleared in the *RCTL* register only through the host interface (not by the BMC) and it is usually used when the I350 is used as a sniffer.

Note: When the promiscuous bit is set and a multicast packet is received, the *PIF* bit of the packet status is not set.

7.1.1.2 VLAN Filtering

A receive packet that successfully passed MAC address filtering is then subjected to VLAN header filtering.

1. If the packet does not have a VLAN header, it passes to the next filtering stage.

Note: If external VLAN is enabled (*CTRL_EXT.EXT_VLAN* is set), it is assumed that the first VLAN tag is an external VLAN and it is skipped. All next stages refer to the second VLAN.

2. If VLAN filtering is disabled (*RCTL.VFE* bit is cleared), the packet is forwarded to the next filtering stage.
3. If the packet has a VLAN header, and it matches an enabled host VLAN filter (relevant bit in *VFTA* table is set), the packet is forwarded to the next filtering stage.
4. Otherwise, the packet is dropped.

Figure 7-4 shows the VLAN filtering flow.

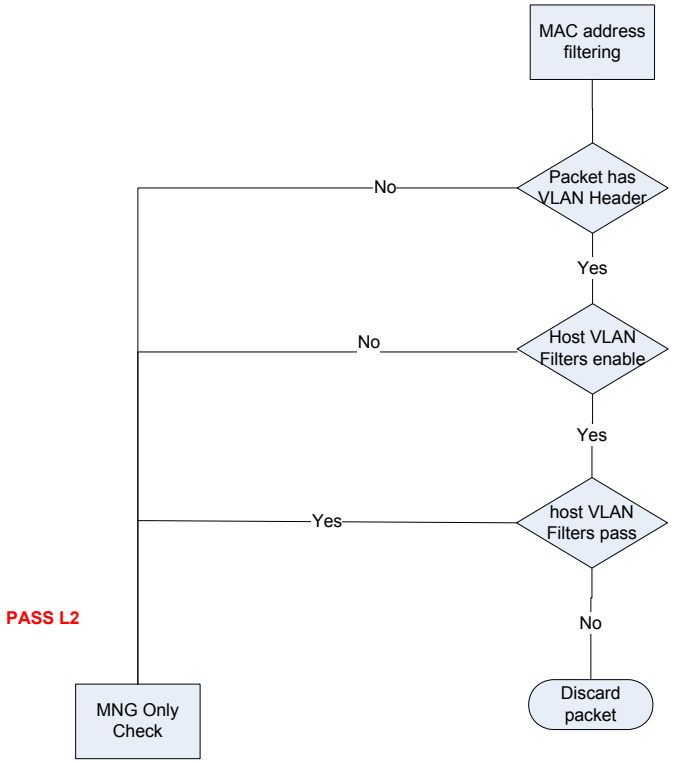


Figure 7-4 VLAN Filtering

7.1.1.3 Manageability Filtering

Manageability filtering is described in Section 10.3.

Figure 7-5 shows the manageability portion of the packet filtering and it is brought here to make the receive packet filtering functionality description complete.

Note: The manageability engine might decide to block part of the received packets from also being sent to the Host, according to the external BMC instructions and the EEPROM settings.

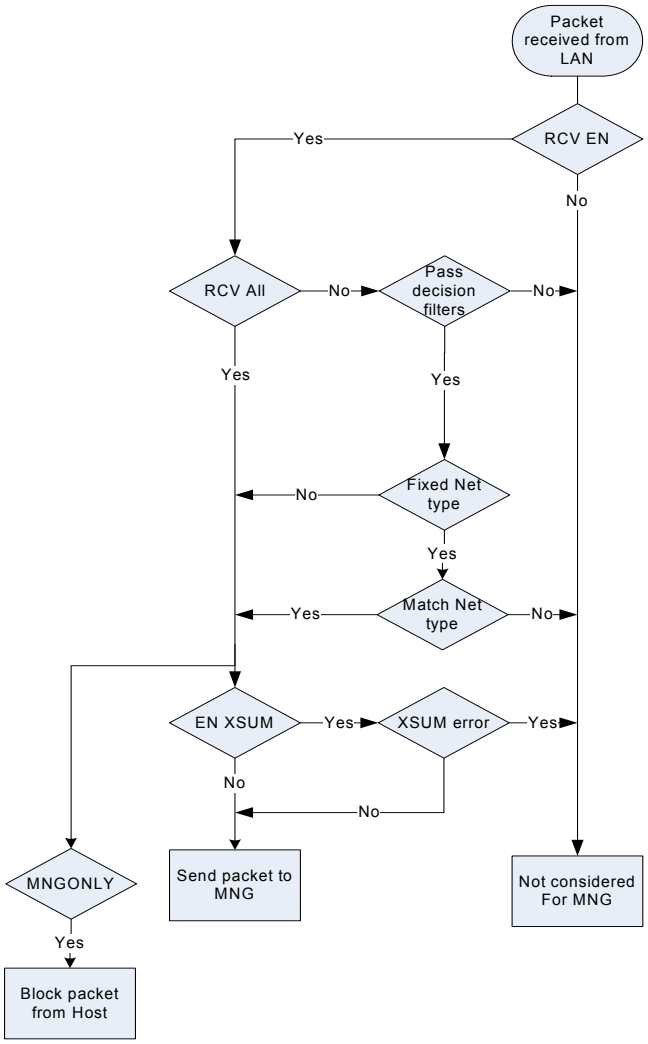


Figure 7-5 Manageability Filtering

7.1.1.4 Size Filtering

A packet is defined as undersize if it is smaller than 64 bytes.

A packet is defined as oversize in the following conditions:

- The *RCTL.LPE* bit cleared and one of the following conditions is met:
  - The packet is bigger than 1518 bytes and there are no VLAN tags in the packet.
  - The packet is bigger than 1522 bytes and there is one VLAN tag in the packet.
  - The packet is bigger than 1526 bytes and there are two VLAN tags in the packet.
- The *RCTL.LPE* bit is set to 1b and the packet is bigger than *RLPML.RLPML* bytes.

Note: Maximum supported received-packet size is 9.5 KB (9728 bytes).

Note: In VMDq mode, a packet is defined as oversized only if it is bigger than the *VOMLR.RLPML* value for all the VM/VFs that were supposed to receive the packet.

7.1.2 Receive Queues Assignment

The following filter mechanisms determines the destination of a receive packet. These are described briefly in this section and in full details in separate sections:

- Virtualization — In a virtualized environment, DMA resources are shared between more than one software entity (operating system and/or software device driver). This is done by allocating receive descriptor queues/pools to virtual partitions (VMM or VMs). Virtualization assigns to each received packet one or more pool indices. Packets are routed to a pool based on their pool index and other considerations. See Section 7.1.2.2 for details on routing for virtualization.
- RSS — Receive Side Scaling distributes packet processing between several processor cores by assigning packets into different descriptor queues. RSS assigns to each received packet an RSS index. Packets are routed to a queue out of a set of Rx queues based on their RSS index and other considerations. See Section 7.1.2.8 for details on RSS.
- L2 Ethertype filters — These filters identify packets by their L2 Ether type and assign them to receive queues. Examples of possible uses are LLDP packets and 802.1X packets. See Section 7.1.2.4 for mode details. The I350 incorporates 8 Ether-type filters per port.
- 2-tuple filters — These filters identify packets with specific TCP/UDP destination port and/or L4 protocol. Each filter consists of a 2-tuple (protocol and destination TCP/UDP port) and routes packets into one of the Rx queues. The I350 has 8 such filters per port. See Section 7.1.2.5 for details.
- TCP SYN filters — The I350 might route TCP packets with their *SYN* flag set into a separate queue. *SYN* packets are often used in *SYN* attacks to load the system with numerous requests for new connections. By filtering such packets to a separate queue, security software can monitor and act on *SYN* attacks. The I350 has one such filter per port. See Section 7.1.2.7 for more details.
- Flex Filters - These filters can be either used as WoL filters when the I350 is in D3 state or for queueing in normal operating mode (D0 state). Filters enable queueing according to a match of any 128 Byte sequence at the beginning of a packet. Each one of the 128 bytes can be either compared or masked using a dedicated mask field. The I350 has 8 such filters per port. See Section 7.1.2.6 for details.

A received packet is allocated to a queue as described in the following sections.

The tables below describe allocation of queues in each of the modes.

Table 7-1 Queue Allocation[1]

Virtualization	RSS	Queue allocation
Disabled	Disabled	One default queue (MRQC.DEF_Q)
	Enabled	Up to 8 queues by RSS.
Enabled	Disabled	One queue per VM (queues 0-7 for VM 0-7).
	Enabled	Two queues per VM (queues 0, 8; 1, 9; 2, 10; 3, 11; 4, 12; 5, 13; 6, 14; 7; 15 for VM 0-7, respectively). Spread between the queues by RSS.

Table 7-1 Queue Allocation[1]

Virtualization	RSS	Queue allocation
Disabled	Disabled	One queue per TC (Queue 0 and 8).
	Enabled	Eight queues per TC (queues 0-7 for TC0 and queues 8-15 for TC1). Spread between the queues by RSS.
Enabled	Disabled	One queue per TC per VM (Queues 0, 8; 1, 9; 2, 10; 3, 11; 4, 12; 5, 13; 6, 14; 7; 15 for VM 0-7/TC 0,1 respectively).
	Enabled	Not available

1. On top of this allocation, the special filters can override the queueing decision.

7.1.2.1 Queuing in a Non-Virtualized Environment

When the *MRQC.Multiple Receive Queues Enable* field equals 010b (Multiple receive queues as defined by filters and RSS for 8 queues) or 000b (Multiple receive queues as defined by filters (2-tuple filters, L2 Ether-type filters, SYN filter and Flex Filters), the received packet is assigned to a queue in the following manner (Each filter identifies one of 8 receive queues):

1. Queue by L2 Ether-type filters (if a match)
2. If RFCTL.SYNQFP is 0b (2-tuple filter and Flex filter have priority), then:
   a. Queue by Flex filter (if a match)
   b. Queue by 2-tuple filter
   c. Queue by SYN filter (if a match)
3. If RFCTL.SYNQFP is 1b (SYN filter has priority), then:
   a. Queue by SYN filter (if a match)
   b. Queue by Flex filter (if a match)
   c. Queue by 2-tuple filter (if a match)
4. Queue by RSS (if RSS enabled) - Identifies one of 1 x 8 queues through the RSS index. The following modes are supported:
   — No RSS — The default queue as defined in *MRQC.DEF_Q* is used for packets that do not meet any of the previous conditions.
   — RSS only — A set of 8 queues is allocated for RSS. The queue is identified through the RSS index. Note that it is possible to use a subset of the 8 queues.

Note: No RSS here mean either that RSS is disabled (*MRQC.Multiple Receive Queues Enable* field equals 000b) or that the packet did not match any of the RSS filters.

Figure 7-6 describes the non virtualized receive queue assignment flow.

7.1.2.2 Receive Queuing in a Virtualized Environment

In VMDq mode, system software allocates the pools to the VMM, an IOVM, or to VMs. When the *MRQC.Multiple Receive Queues Enable* field equals 011b (Multiple receive queues as defined by VMDq), the received packets are allocated to the 8 receive queues/pools in the following manner:

Incoming packets are associated with pools/queues based on their L2 characteristics as described in Section 7.8.3.

Figure 7-6 describes the generic virtualized receive queue assignment flow.

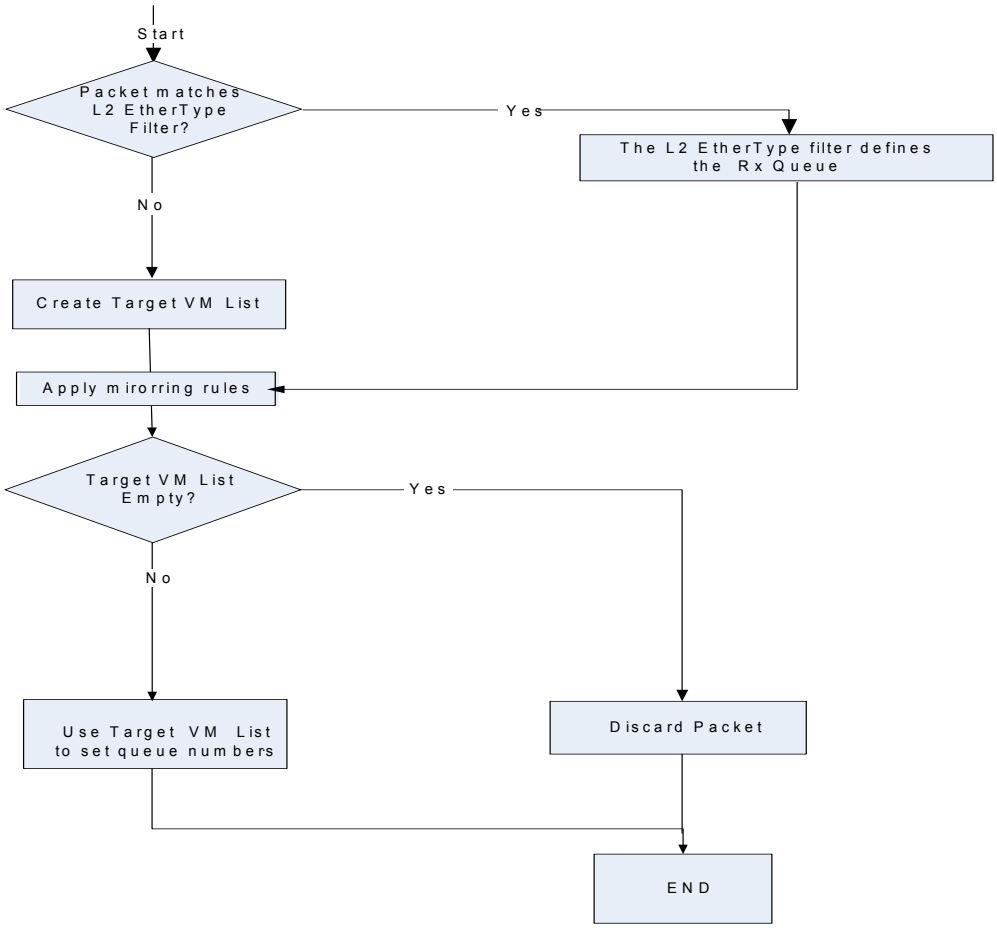


Figure 7-6 Receive Queuing Flow (Virtualization)

7.1.2.3 Queue Configuration Registers

Configuration registers (CSRs) that control queue operation are replicated per queue (total of 8 copies of each register per port). Each of the replicated registers correspond to a queue such that the queue index equals the serial number of the register (such as register 0 corresponds to queue 0, etc.). Registers included in this category are:

- *RDBAL* and *RDBAH* — Rx Descriptor Base
- *RDLEN* — RX Descriptor Length
- *RDH* — RX Descriptor Head
- *RDT* — RX Descriptor Tail
- *RXDCTL* — Receive Descriptor Control
- *RXCTL* — Rx DCA Control
- *SRRCTL* — Split and Replication Receive Control

CSRs that define the functionality of descriptor queues are replicated per VM pool to allow for a separate configuration in a virtualized environment (total of 8 copies of each register). Each of the replicated registers correspond to a set of queues in the same VM pool. Registers included in this category are:

- *PSRTYPE* — Packet Split Receive type

7.1.2.4 L2 Ether-Type Filters

These filters identify packets by L2 Ether-type and assign them to a receive queue. The following usages have been identified:

- IEEE 802.1X packets — Extensible Authentication Protocol over LAN (EAPOL).
- Time sync packets (such as IEEE 1588) — Identifies Sync or Delay_Req packets
- IEEE802.1AB LLDP (Link Layer Discovery Protocol) packets.

The I350 incorporates 8 Ether-type filters.

The *Packet Type* field in the Rx descriptor captures the filter number that matched the L2 Ether-type. See Section 7.1.4.2 for decoding of the *Packet Type* field.

The Ether-type filters are configured via the ETQF register as follows:

- The *EType* field contains the 16-bit Ether-type compared against all L2 type fields in the Rx packet.
- The *Filter Enable* bit enables identification of Rx packets by Ether-type according to this filter. If this bit is cleared, the filter is ignored for all purposes.
- The *Rx Queue* field contains the absolute destination queue for the packet.
- The *1588 Time Stamp* field indicates that the packet should be time stamped according to the IEEE 1588 specification.
- The *Queue Enable* field enables forwarding Rx packets based on the Ether-type defined in this register.

Note: Software should not assign the same Ether-type value to different ETQF filters with different *Rx Queue* assignments.

Special considerations for Virtualization modes:

- Packets that match an Ether-type filter are diverted from their original pool (the pool identified by the L2 filters) to the pool used as the pool to which the queue in the *Queue* field belongs. In other words, The L2 filters are ignored in determining the pool for such packets.
- The same applies for multi-cast packets. A single copy is posted to the pool defined by the filter.
- Mirroring rules:
  - If a pool is being mirrored, the pool to which the queue in the *Queue* field belongs to is used to determine if a packet that matches the filter should be mirrored.
  - The queue inside the pool (indicated by the *Queue* field) is used for both the original pool and the mirroring pool.

7.1.2.5 2-Tuple Filters

These filters identify specific packets destined to a certain TCP/UDP port and implement a specific protocol. Each filter consists of a 2-tuple (protocol and destination TCP/UDP port) and forwards packets into one of the receive queues.

The I350 incorporates 8 such filters.

The 2-tuple filters are configured via the *TTQF* (See Section 8.11.4), *IMIR* (See Section 8.11.1) and *IMIR_EXT* (See Section 8.11.2) registers as follows (per filter):

- Protocol — Identifies the IP protocol, part of the 2-tuple queue filters. Enabled by a bit in the *TTQF.Mask* field.
- Destination port — Identifies the TCP/UDP destination port, part of the 2-tuple queue filters. Enabled by the *IMIR.PORT_BP* bit.
- Size threshold (*IMIREXT.Size_Thresh*) — Identifies the length of the packet that should trigger the filter. This is the length as received by the host, not including any part of the packet removed by hardware. Enabled by the *IMIREXT.Size_BP* field.
- Control Bits — Identify TCP flags that might be part of the filtering process. Enabled by the *IMIREXT.CtrlBit_BP* field.

Note: When using the Control Bits filters, the Protocol filter must be enabled and set to TCP.

- Rx queue — Determines the Rx queue for packets that match this filter:
  - In a non-virtualized configuration, the *TTQF.Rx Queue* field contains the queue serial number.
  - In the virtualized configuration, the *FTQF.Rx Queue* field contains the queue serial number within the set of queues of the VF associated (via the *FTQF.VF* field) with this filter. In this case, the packet is sent to all VFs in the VF index list (see Section 7.1.2.2 for details) in the queue defined in the filter.
- Queue enable — Enables forwarding a packet that uses this filter to the queue defined in the *TTQF.Rx Queue* field.
- VF — Identifies the VF associated with this filter by its VF index (virtualization modes only). A packet must match the VF filters (such as MAC address) and the 5-tuple filter for this filter to apply.

Note: The above field should not be set to match a mirror port (such as a port that receives promiscuous traffic), as it influences the queuing of packets sent to mirrored port.

- VF Mask — Determines if the *VF* field participates in the 5-tuple match or is ignored:
  - Must be set to 1b in non-virtualized case
  - In a virtualized configuration:
    - When set to 0b, only unicast packets that match the *VF* field are candidates for this filter.
    - When set to 1b, unicast, multicast, and broadcast packets might all match with the 5-tuple filter. VF association is not checked. The *Rx Queue* field defines a queue for each VF.
- Mask — A 1-bit field that masks the L4 protocol check. The filter is a logical AND of the non-masked 2-tuple fields. If all 2-tuple fields are masked, the filter is not used for queue forwarding.

Notes:

- If more than one 2-tuple filter with the same priority is matched by the packet, the first filter (lowest ordinal number) is used in order to define the queue destination of this packet.
- The immediate interrupt and 1588 actions are defined by the OR of all the matching filters.

7.1.2.6 Flex Filters

The I350 supports a total of 8 flexible filters. Each filter can be configured to recognize any arbitrary pattern within the first 128 bytes of the packet. To configure the flexible filters, software programs the mask values (required values and the minimum packet length), into the Flexible Host Filter Table (*FHFT*

and *FHFT_EXT*, See Section 8.20.11 and Section 8.20.12). These 8 flexible filters can be used as for wake-up or proxying when in D3 state or for queueing when in D0 state. Software must enable the filters in the *Wake Up Filter* Control (*WUFC See* Section 8.20.2) register or Proxying Filter Control (*PROXYFC* see Section 8.20.6) for operation in D3 low power mode or in the *WUFC* register in D0 mode. In D0 mode these filters enable forwarding of packets that match up to 128 Bytes defined in the filter to one of the receive queues. In D3 mode these filters can be used for Wake-on-Lan as described in Section 5.6.3.1.8 or proxying as described in Section 5.7.

Once enabled, the flexible filters scan incoming packets for a match. If the filter encounters any byte in the packet where the mask bit is one and the byte doesn't match the value programmed in the Flexible Host Filter Table (*FHFT* or *FHFT_EXT*), then the filter fails that packet. If the filter reaches the required length without failing the packet, it forwards the packet to the appropriate receive queue. It ignores any mask bits set to one beyond the required length (defined in the Length field in the *FHFT* or *FHFT_EXT* registers).

Note: The flex filters are temporarily disabled when read from or written to by the host. Any packet received during a read or write operation is dropped. Filter operation resumes once the read or write access completes.

The flex fitters are configured in D0 state via the *WUFC*, *FHFT* and *FHFT_EXT* registers as follows (per filter):

- Byte Sequence to be compared - Program 128 Byte sequence, mask bits and *Length* field in FHFT and *FHFT_EXT* registers.
- Filter Priority - Program filter priority in queueing field in *FHFT* and *FHFT_EXT* registers.
- Receive queue - Program receive queue to forward packet in queueing field in *FHFT* and *FHFT_EXT* registers.
- Filter actions - Program immediate interrupt requirement in queueing field in *FHFT* and *FHFT_EXT* registers.
- Filter enable - Set *WUFC.FLEX_HQ* bit to 1 to enable flex filter operation in D0 state. Set appropriate *WUFC.FLX[n]* bit to 1 to enable specific flex filter.

Before entering D3 state software device driver programs the *FHFT* and *FHFT_EXT* filters for appropriate wake events and enables relevant filters by setting the *WUFC.FLX[n]* bit to 1 or the *PROXYFC.FLX[n]* bit to 1. Following move to D0 state the software device driver programs the FHFT and FHFT_EXT filters for appropriate queueing decisions and enables the relevant filters by setting the *WUFC.FLX[n]* bit to 1 and the *WUFC.FLEX_HQ* bit to 1.

Notes: If more than one flex filter with the same priority is matched by the packet, the first filter (lowest address) is used in order to define the queue destination of this packet.

The immediate interrupt action is defined by the OR of all the matching filters.

These filters are not available for VM to VM traffic forwarding.

7.1.2.7 SYN Packet Filters

The I350 might forward TCP packets whose *SYN* flag is set into a separate queue. SYN packets are often used in SYN attacks to load the system with numerous requests for new connections. By filtering such packets to a separate queue, security software can monitor and act on SYN attacks.

SYN filters are configured via the SYNQF registers as follows:

- Queue En — Enables forwarding of SYN packets to a specific queue.
- Rx Queue field — Contains the destination queue for the packet.

This filter is not to be used in a virtualized environment.

7.1.2.8 Receive-Side Scaling (RSS)

RSS is a mechanism to distribute received packets into several descriptor queues. Software then assigns each queue to a different processor, sharing the load of packet processing among several processors.

The I350 uses RSS as one ingredient in its packet assignment policy (the others are the various filters and virtualization). The RSS output is a RSS index. The I350's global assignment uses these bits (or only some of the LSB bits) as part of the queue number.

RSS is enabled in the *MRQC* register. The RSS *Status* field in the descriptor write-back is enabled when the *RXCSUM.PCSD* bit is set (fragment checksum is disabled). RSS is therefore mutually exclusive with UDP fragmentation. Also, support for RSS is not provided when legacy receive descriptor format is used.

When RSS is enabled, the I350 provides software with the following information as required by Microsoft* RSS specification or for device driver assistance:

- A Dword result of the Microsoft* RSS hash function, to be used by the stack for flow classification, is written into the receive packet descriptor (required by Microsoft* RSS).
- A 4-bit RSS *Type* field conveys the hash function used for the specific packet (required by Microsoft* RSS).

Figure 7-7 shows the process of computing an RSS output:

1. The receive packet is parsed into the header fields used by the hash operation (such as IP addresses, TCP port, etc.).
2. A hash calculation is performed. The I350 supports a single hash function, as defined by Microsoft* RSS. The I350 does not indicate to the software device driver which hash function is used. The 32-bit result is fed into the packet receive descriptor.
3. The seven LSB bits of the hash result are used as an index into a 128-entry indirection table. Each entry provides a 3-bit RSS output index.

When RSS is disabled, packets are assigned an RSS output index = zero. System software might enable or disable RSS at any time. While disabled, system software might update the contents of any of the RSS-related registers.

When multiple requests queues are enabled in RSS mode, un-decodable packets are assigned an RSS output index = zero. The 32-bit tag (normally a result of the hash function) equals zero.

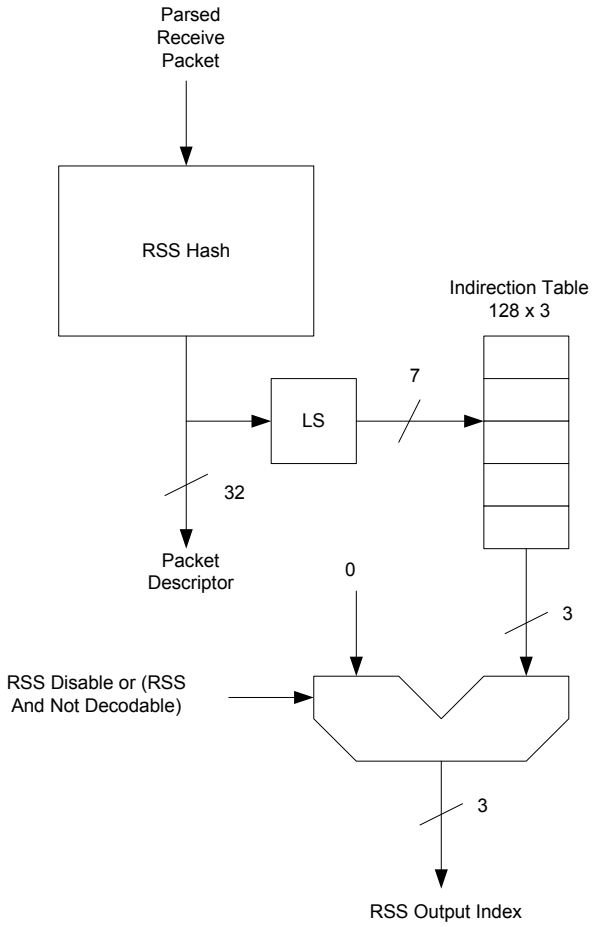


Figure 7-7 RSS Block Diagram

7.1.2.8.1 RSS Hash Function

Section 7.1.2.8.1 provides a verification suite used to validate that the hash function is computed according to Microsoft* nomenclature.

The I350 hash function follows Microsoft* definition. A single hash function is defined with several variations for the following cases:

- TcpIPv4 — The I350 parses the packet to identify an IPv4 packet containing a TCP segment per the criteria described later in this section. If the packet is not an IPv4 packet containing a TCP segment, RSS is not done for the packet.
- IPv4 — The I350 parses the packet to identify an IPv4 packet. If the packet is not an IPv4 packet, RSS is not done for the packet.
- TcpIPv6 — The I350 parses the packet to identify an IPv6 packet containing a TCP segment per the criteria described later in this section. If the packet is not an IPv6 packet containing a TCP segment, RSS is not done for the packet.

- TcpIPv6Ex — The I350 parses the packet to identify an IPv6 packet containing a TCP segment with extensions per the criteria described later in this section. If the packet is not an IPv6 packet containing a TCP segment, RSS is not done for the packet. Extension headers should be parsed for a *Home-Address-Option* field (for source address) or the *Routing-Header-Type-2* field (for destination address).
- IPv6Ex — The I350 parses the packet to identify an IPv6 packet. Extension headers should be parsed for a *Home-Address-Option* field (for source address) or the *Routing-Header-Type-2* field (for destination address). Note that the packet is not required to contain any of these extension headers to be hashed by this function. In this case, the IPv6 hash is used. If the packet is not an IPv6 packet, RSS is not done for the packet.
- IPv6 — The I350 parses the packet to identify an IPv6 packet. If the packet is not an IPv6 packet, receive-side-scaling is not done for the packet.

The following additional cases are not part of the Microsoft* RSS specification:

- UdpIPV4 — The I350 parses the packet to identify a packet with UDP over IPv4.
- UdpIPV6 — The I350 parses the packet to identify a packet with UDP over IPv6.
- UdpIPV6Ex — The I350 parses the packet to identify a packet with UDP over IPv6 with extensions.

A packet is identified as containing a TCP segment if all of the following conditions are met:

- The transport layer protocol is TCP (not UDP, ICMP, IGMP, etc.).
- The TCP segment can be parsed (such as IP options can be parsed, packet not encrypted).
- The packet is not fragmented (even if the fragment contains a complete TCP header).

Bits[31:16] of the Multiple Receive Queues Command (*MRQC)* register enable each of the above hash function variations (several can be set at a given time). If several functions are enabled at the same time, priority is defined as follows (skip functions that are not enabled):

IPv4 packet:

1. Try using the TcpIPv4 function.
2. Try using IPV4_UDP function.
3. Try using the IPv4 function.

IPv6 packet:

1. If TcpIPv6Ex is enabled, try using the TcpIPv6Ex function; else if TcpIPv6 is enabled try using the TcpIPv6 function.
2. If UdpIPv6Ex is enabled, try using UdpIPv6Ex function; else if UpdIPv6 is enabled try using UdpIPv6 function.
3. If IPv6Ex is enabled, try using the IPv6Ex function, else if IPv6 is enabled, try using the IPv6 function.

The following combinations are currently supported:

- Any combination of IPv4, TcpIPv4, and UdpIPv4.
- And/or.
- Any combination of either IPv6, TcpIPv6, and UdpIPv6 or IPv6Ex, TcpIPv6Ex, and UdpIPv6Ex.

When a packet cannot be parsed by the previously mentioned rules, it is assigned an RSS output index = zero. The 32-bit tag (normally a result of the hash function) equals zero.

The 32-bit result of the hash computation is written into the packet descriptor and also provides an index into the indirection table.

The following notation is used to describe the hash functions:

- Ordering is little endian in both bytes and bits. For example, the IP address 161.142.100.80 translates into 0xa18e6450 in the signature.
- A "^ "denotes bit-wise XOR operation of same-width vectors.
- @x-y denotes bytes x through y (including both of them) of the incoming packet, where byte 0 is the first byte of the IP header. In other words, it is considered that all byte-offsets as offsets into a packet where the framing layer header has been stripped out. Therefore, the source IPv4 address is referred to as @12-15, while the destination v4 address is referred to as @16-19.
- @x-y, @v-w denotes concatenation of bytes x-y, followed by bytes v-w, preserving the order in which they occurred in the packet.

All hash function variations (IPv4 and IPv6) follow the same general structure. Specific details for each variation are described in the following section. The hash uses a random secret key length of 320 bits (40 bytes); the key is typically supplied through the RSS Random Key Register (RSSRK).

The algorithm works by examining each bit of the hash input from left to right. Intel's nomenclature defines left and right for a byte-array as follows: Given an array K with k bytes, Intel's nomenclature assumes that the array is laid out as shown:

```
K[0] K[1] K[2] … K[k-1]
```

K[0] is the left-most byte, and the MSB of K[0] is the left-most bit. K[k-1] is the right-most byte, and the LSB of K[k-1] is the right-most bit.

```
ComputeHash(input[], N)
For hash-input input[] of length N bytes (8N bits) and a random secret key K of 320 bits
Result = 0;
For each bit b in input[] {
if (b == 1) then Result ^= (left-most 32 bits of K);
shift K left 1 bit position;
}
return Result;
```

The following four pseudo-code examples are intended to help clarify exactly how the hash is to be performed in four cases, IPv4 with and without ability to parse the TCP header and IPv6 with an without a TCP header.

7.1.2.8.1.1 Hash for IPv4 with TCP

Concatenate SourceAddress, DestinationAddress, SourcePort, DestinationPort into one single byte-array, preserving the order in which they occurred in the packet:

```
Input[12] = @12-15, @16-19, @20-21, @22-23.
Result = ComputeHash(Input, 12);
```

7.1.2.8.1.2 Hash for IPv4 with UDP

Concatenate SourceAddress, DestinationAddress, SourcePort, DestinationPort into one single byte-array, preserving the order in which they occurred in the packet:

```
Input[12] = @12-15, @16-19, @20-21, @22-23.
Result = ComputeHash(Input, 12);
```

7.1.2.8.1.3 Hash for IPv4 without TCP

Concatenate SourceAddress and DestinationAddress into one single byte-array

```
Input[8] = @12-15, @16-19
```

```
Result = ComputeHash(Input, 8)
```

7.1.2.8.1.4 Hash for IPv6 with TCP

Similar to above:

```
Input[36] = @8-23, @24-39, @40-41, @42-43
Result = ComputeHash(Input, 36)
```

7.1.2.8.1.5 Hash for IPv6 with UDP

Similar to above:

```
Input[36] = @8-23, @24-39, @40-41, @42-43
Result = ComputeHash(Input, 36)
```

7.1.2.8.1.6 Hash for IPv6 without TCP

```
Input[32] = @8-23, @24-39
Result = ComputeHash(Input, 32)
```

7.1.2.8.2 Indirection Table

The *RETA* indirection table is a 128-entry structure, indexed by the seven LSB bits of the hash function output. Each entry of the table contains the following:

- Bits [2:0] - RSS index

Note: In RSS only mode, all 3 bits are used. In VMDq mode RSS is not supported.

System software might update the indirection table during run time. Such updates of the table are not synchronized with the arrival time of received packets. Therefore, it is not guaranteed that a table update takes effect on a specific packet boundary.

7.1.2.8.3 RSS Verification Suite

Assume that the random key byte-stream is:

```
0x6d, 0x5a, 0x56, 0xda, 0x25, 0x5b, 0x0e, 0xc2,
0x41, 0x67, 0x25, 0x3d, 0x43, 0xa3, 0x8f, 0xb0,
0xd0, 0xca, 0x2b, 0xcb, 0xae, 0x7b, 0x30, 0xb4,
0x77, 0xcb, 0x2d, 0xa3, 0x80, 0x30, 0xf2, 0x0c,
0x6a, 0x42, 0xb7, 0x3b, 0xbe, 0xac, 0x01, 0xfa
```

7.1.2.8.3.1 IPv4

Table 7-2 IPv4

Destination Address/Port	Source Address/Port	IPv4 Only	IPv4 With TCP
161.142.100.80:1766	66.9.149.187:2794	0x323e8fc2	0x51ccc178
65.69.140.83:4739	199.92.111.2:14230	0xd718262a	0xc626b0ea
12.22.207.184:38024	24.19.198.95:12898	0xd2d0a5de	0x5c2b394a
209.142.163.6:2217	38.27.205.30:48228	0x82989176	0xafc7327f
202.188.127.2:1303	153.39.163.191:44251	0x5d1809c5	0x10e828a2

7.1.2.8.3.2 IPv6

The IPv6 address tuples are only for verification purposes and might not make sense as a tuple.

Table 7-3 IPv6

Destination Address/Port	Source Address/Port	IPv6 Only	IPv6 With TCP
3ffe:2501:200:3::1 (1766)	3ffe:2501:200:1fff::7 (2794)	0x2cc18cd5	0x40207d3d
ff02::1 (4739)	3ffe:501:8::260:97ff:fe40:efab (14230)	0x0f0c461c	0xdde51bbf
fe80::200:f8ff:fe21:67cf (38024)	3ffe:1900:4545:3:200:f8ff:fe21:67cf (44251)	0x4b61e985	0x02d1feef

7.1.2.8.4 Association Through MAC Address

Each of the 32 MAC address filters can be associated with a VF/VM. The *POOLSEL* field in the Receive Address High (*RAH*) register determines the target VM. Packets that do not match any of the MAC filters (such as promiscuous) are assigned with the default VM as defined in the *VT_CTL.DEF_PL* field.

Software can program different values to the MAC filters (any bits in *RAH* or *RAL*) at any time. The I350 would respond to the change on a packet boundary but does not guarantee the change to take place at some precise time.

7.1.3 Receive Data Storage

7.1.3.1 Host Buffers

Each descriptor points to a one or more memory buffers that are designated by the software device driver to store packet data.

The size of the buffer can be set using either the generic *RCTL.BSIZE* field, or the per queue *SRRCTL[n].BSIZEPACKET* field.

If *SRRCTL[n].BSIZEPACKET* is set to zero for any queue, the buffer size defined by *RCTL.BSIZE* is used. Otherwise, the buffer size defined by *SRRCTL[n].BSIZEPACKET* is used.

If the receive buffer size is selected by bit settings in the Receive Control (*RCTL.BSIZE*) buffer sizes of 256, 512, 1024, and 2048 bytes are supported.

If the receive buffer size is selected by *SRRCTL[n].BSIZEPACKET*, buffer sizes of 1Kbytes to 127 KBytes are supported with a resolution of 1KByte.

In addition, for advanced descriptor usage the *SRRCTL.BSIZEHEADER* field is used to define the size of the buffers allocated to headers. Header Buffer sizes of 64 bytes to 2048 bytes with a resolution of 64 bytes are supported.

The I350 places no alignment restrictions on receive memory buffer addresses. This is desirable in situations where the receive buffer was allocated by higher layers in the networking software stack, as these higher layers might have no knowledge of a specific device's buffer alignment requirements.

Note: When the *No-Snoop Enable* bit is used in advanced descriptors, the buffer address is 16-bit (2-byte) aligned.

7.1.3.2 On-Chip Receive Buffers

The I350 allocates by default a 36 KB on-chip packet buffer per port. The buffer can be used to store packets until they are forwarded to the host. The I350 utilizes a single common ram structure for the on-chip receive buffers allocated to the various ports. If a port is disabled, so that it can't be accessed by host and management, by either:

1. Pin assertion (LAN0_DIS_N, LAN1_DIS_N, LAN2_DIS_N, LAN3_DIS_N for port 0 to 3 respectively) and setting the EEPROM bit *PHY_in_LAN_disable* in the "*Software Defined Pins Control*" word to 1 for the relevant port.
2. Setting EEPROM bit *LAN_DIS* or the LAN_PCI_DIS in the "*Software Defined Pins Control*" word for the relevant port to 1.

The freed buffer space can be allocated to the active ports via the "*Initialization Control 4*" EEPROM word. Actual on-chip receive buffer allocated to the port can be read in the *IRPBS register*.

7.1.3.3 On-Chip Descriptor Buffers

The I350 contains a 16 descriptor cache for each receive queue used to reduce the latency of packet processing and to optimize the usage of PCIe bandwidth by fetching and writing back descriptors in bursts. The fetch and writeback algorithm are described in Section 7.1.4.3 and Section 7.1.4.4.

7.1.4 Receive Descriptors

7.1.4.1 Legacy Receive Descriptor Format

A receive descriptor is a data structure that contains the receive data buffer address and fields for hardware to store packet information. If *SRRCTL[n].DESCTYPE* = 000b, the I350 uses the legacy Receive descriptor as shown in Table 7-4. The shaded areas indicate fields that are modified by hardware upon packet reception (so-called descriptor write-back).

Note: Legacy descriptors should not be used when advanced features such as Virtualization are activated.

Table 7-4 Legacy Receive Descriptor (RDESC) Layout

	63 48	47 40	39 32	31 16	15 0
0	Buffer Address [63:0]				
8	VLAN Tag	Errors	Status	Fragment Checksum	Length

After receiving a packet for the I350, hardware stores the packet data into the indicated buffer and writes the length, packet checksum, status, errors, and status fields.

Packet Buffer Address (64) - Physical address of the packet buffer.

Length Field (16)

Length covers the data written to a receive buffer including CRC bytes (if any). Software must read multiple descriptors to determine the complete length for a packet that spans multiple receive buffers.

Fragment Checksum (16)

This field is used to provide the fragment checksum value. This field equals to the unadjusted 16-bit ones complement of the packet. Checksum calculation starts at the L4 layer (after the IP header) until the end of the packet excluding the CRC bytes. In order to use the fragment checksum assist to offload L4 checksum verification, software might need to back out some of the bytes in the packet. For more details see Section 7.1.7.3

Status Field (8)

Status information indicates whether the descriptor has been used and whether the referenced buffer is the last one for the packet. See Table 7-5 for the layout of the *Status* field. Error status information is shown in Figure 7-9.

Table 7-5 Receive Status (RDESC.STATUS) Layout

7	6	5	4	3	2	1	0
PIF	IPCS	L4CS	UDPCS	VP	Rsv	EOP	DD

- PIF (bit 7) - Passed imperfect filter only
- IPCS (bit 6) - IPv4 checksum calculated on packet
- L4CS (bit 5) - L4 (UDP or TCP) checksum calculated on packet
- UDPCS (bit 4) - UDP checksum or IP payload checksum calculated on packet.
- VP (bit 3) - Packet is 802.1q; indicates strip VLAN in 802.1q packet
- RSV (bit 2) - Reserved
- EOP (bit 1) - End of packet
- DD (bit 0) - Descriptor done

EOP and DD

The following table lists the meaning of these bits:

Table 7-6 Receive Status Bits

DD	EOP	Description
0b	0b	Software setting of the descriptor when it hands it off to the hardware.
0b	1b	Reserved (invalid option).
1b	0b	A completion status indication for a non-last descriptor of a packet that spans across multiple descriptors. In a single packet case, DD indicates that the hardware is done with the descriptor and its buffers. Only the *Length* fields are valid on this descriptor.
1b	1b	A completion status indication of the entire packet. Note that software Might take ownership of its descriptors. All fields in the descriptor are valid (reported by the hardware).

VP Field

The *VP* field indicates whether the incoming packet's type matches the VLAN Ethernet Type programmed in the *VET* Register. For example, if the packet is a VLAN (802.1q) type, it is set if the packet type matches *VET* and *CTRL.VME* is set (VLAN mode enabled). It also indicates that VLAN has been stripped from the 802.1q packet. For more details, see Section 7.4.

IPCS (IPv4 Checksum), L4CS (L4 Checksum), and UDPCS (UDP Checksum)

The meaning of these bits is shown in the table below:

Table 7-7 IPCS, L4CS, and UDPCS

L4CS	UDPCS	IPCS	Functionality
0b	0b	0b	Hardware does not provide checksum offload. Special case: Hardware does not provide UDP checksum offload for IPV4 packet with UDP checksum = 0b
1b	0b	1b / 0b	Hardware provides IPv4 checksum offload if IPCS is active and TCP checksum is offload. A pass/fail indication is provided in the *Error* field – IPE and L4E.
0b	1b	1b / 0b	Hardware provides IPv4 checksum offload if IPCS is active and UDP checksum is offload. A pass/fail indication is provided in the *Error* field – IPE and L4E.

Refer to Table 7-19 for a description of supported packet types for receive checksum offloading. Unsupported packet types do not have the *IPCS* or *L4CS* bits set. IPv6 packets do not have the *IPCS* bit set, but might have the *L4CS* bit set if the I350 recognized the TCP or UDP packet.

PIF

Hardware supplies the *PIF* field to expedite software processing of packets. Software must examine any packet with *PIF* bit set to determine whether to accept the packet. If the *PIF* bit is clear, then the packet is known to be destined to this station, so software does not need to look at the packet contents. Multicast packets passing only the Multicast Vector (MTA) or unicast packets passing only the Unicast Hash Table (UTA) but not any of the MAC address exact filters (RAH, RAL) set the *PIF* bit. In addition, the following condition causes *PIF* to be cleared:

- The DA of the packet is a multicast address and promiscuous multicast is set (*RCTL.MPE* = 1b).
- The DA of the packet is a broadcast address and accept broadcast mode is set (*RCTL.BAM* = 1b)

A MAC control frame forwarded to the host (*RCTL.PMCF* = 0b) that does not match any of the exact filters, has the *PIF* bit set.

Error Field (8)

Most error information appears only when the *store-bad-packet* bit (*RCTL.SBP*) is set and a bad packet is received. See Table 7-8 for a definition of the possible errors and their bit positions.

Table 7-8 RXE, IPE and L4E

7	6	5	4	3	2	1	0
RXE	IPE	L4E	Reserved				

- RXE (bit 7) - RX Data Error
- IPE (bit 6) - IPv4 Checksum Error
- L4E (bit 5) - TCP/UDP Checksum Error
- Reserved (bit 4:0)

IPE/L4E

The IP and TCP/UDP checksum error bits from Table 7-8 are valid only when the IPv4 or TCP/UDP checksum(s) is performed on the received packet as indicated via IPCS and L4CS. These, along with the other error bits, are valid only when the *EOP* and *DD* bits are set in the descriptor.

Note: Receive checksum errors have no effect on packet filtering.

If receive checksum offloading is disabled (*RXCSUM.IPOFLD* and *RXCSUM.TUOFLD*), the *IPE* and *L4E* bits are 0b.

RXE

The RXE error bit is asserted in the following case:

1. CRC error is detected. CRC can be a result of reception of /V/ symbol on the TBI interface (see section 3.7.3.3.2) or assertion of RxERR on the MII/GMII interface or bad EOP or lose of sync during packet reception. Packets with a CRC error are posted to host memory only when *store-bad-packet* bit (*RCTL.SBP*) is set.

VLAN Tag Field (16)

Hardware stores additional information in the receive descriptor for 802.1q packets. If the packet type is 802.1q (determined when a packet matches *VET* and *CTRL.VME* = 1b), then the *VLAN Tag* field records the VLAN information and the four-byte VLAN information is stripped from the packet data storage. Otherwise, the *VLAN Tag* field contains 0x0000. The rule for *VLAN tag* is to use network ordering (also called big endian). It appears in the following manner in the descriptor:

Table 7-9 VLAN Tag Field Layout (for 802.1q Packet)

15 13	12	11 0
PRI	CFI	VLAN

7.1.4.2 Advanced Receive Descriptors

7.1.4.2.1 Advanced Receive Descriptors (RDESC) - Read Format

Table 7-10 shows the receive descriptor. This is the format that software writes to the descriptor queue and hardware reads from the descriptor queue in host memory. Hardware writes back the descriptor in a different format, shown in Table 7-11.

Table 7-10 RDESC Descriptor Read Format

	63 1	0
0	Packet Buffer Address [63:1]	A0/NSE
8	Header Buffer Address [63:1]	DD

Packet Buffer Address (64) - Physical address of the packet buffer. The lowest bit is either A0 (LSB of address) or NSE (No-Snoop Enable), depending on bit *RXCTL.RXdataWriteNSEn* of the relevant queue. See Section 8.13.1.

Header Buffer Address (64) - Physical address of the header buffer. The lowest bit is DD.

Note: The I350 does not support null descriptors (a descriptor with a packet or header address that is always equal to zero).

When software sets the *NSE* bit in the receive descriptor, the I350 places the received packet associated with this descriptor in memory at the packet buffer address with *NSE* set in the PCIe attribute fields. *NSE* does not affect the data written to the header buffer address.

When a packet spans more than one descriptor, the header buffer address is not used for the second, third, etc. descriptors; only the packet buffer address is used in this case.

NSE is enabled for packet buffers that the software device driver knows have not been touched by the processor since the last time they were used, so the data cannot be in the processor cache and snoop is always a miss. Avoiding these snoop misses improves system performance. No-snoop is particularly useful when the DMA engine is moving the data from the packet buffer into application buffers, and the software device driver is using the information in the header buffer for its work with the packet.

Note: When No-Snoop Enable is used, relaxed ordering should also be enabled with *CTRL_EXT.RO_DIS*.

7.1.4.2.2 Advanced Receive Descriptors (RDESC) - Writeback Format

When the I350 writes back the descriptors, it uses the descriptor format shown in Table 7-11.

Note: *SRRCTL[n]. DESCTYPE* must be set to a value other than 000b for the I350 to write back the special descriptors.

Table 7-11 RDESC Descriptor Write-Back Format

	63 48	47 35	34 32	31	30 21	20	19	18 17	16 4	3 0
0	RSS Hash Value/ {Fragment Checksum, IP identification}			SPH	HDR_LEN[9:0]	HDR_LEN[11:10]		RSV	Packet Type	RSS Type
8	VLAN Tag	PKT_LEN		Extended Error			Extended Status			

RSS Type (4)

Table 7-12 RSS Type

Packet Type	Description
0x0	No hash computation done for this packet.
0x1	HASH_TCP_IPV4
0x2	HASH_IPV4
0x3	HASH_TCP_IPV6
0x4	HASH_IPV6_EX
0x5	HASH_IPV6
0x6	HASH_TCP_IPV6_EX
0x7	HASH_UDP_IPV4

Table 7-12 RSS Type

Packet Type	Description
0x8	HASH_UDP_IPV6
0x9	HASH_UDP_IPV6_EX
0xA:0xF	Reserved

The I350 must identify the packet type and then choose the appropriate RSS hash function to be used on the packet. The RSS type reports the packet type that was used for the RSS hash function.

Packet Type (13)

- VPKT (bit 12) - VLAN Packet indication

The 12 LSB bits of the packet type reports the packet type identified by the hardware as follows:

Table 7-13 Packet Type LSB Bits (11:10)

Bit Index	Bit 11 = 0b	Bit 11 = 1b (L2 packet[1])
0	IPV4 - Indicates IPv4 header present[2]	EtherType - ETQF register index that matches the packet. Special types might be defined for 1588, 802.1X, LLDP or any other requested type.
1	IPV4E - Indicates IPv4 Header includes IP options[2]	
2	IPV6 - Indicates IPv6 header present[2 3 4]	
3	IPV6E - Indicates IPv6 Header includes extensions[2 3 4]	Reserved
4	TCP - Indicates TCP header present[2 4 5]	
5	UDP - Indicates UDP header present[2 4 5]	Reserved
6	SCTP - Indicates SCTP header present[2 4 5]	Reserved
7	NFS - Indicates NFS header present[2 4 5]	Reserved
10:8	Reserved	Reserved

1. L2 packet (not L3 or L4 packet) with an EtherType that matches the *EType* field of one of the *ETQF[n]* registers that has the *ETQF[n].Filter enable* bit set to 1b.
2. On unsupported tunneled frames only packet types of external IP header will be set if detected.
3. When a packet is fragmented then the internal packet type bits on a supported tunneled packet (IPv6 tunneled in IPv4 only) won't be set.
4. On supported tunneled frames (IPv6 tunneled in IPv4 only) then all the internal Packet types are set if detected (IPV6, IPV6E, TCP, UDP, SCTP and NFS)
5. When a packet is fragmented the TCP, UDP, SCTP and NFS bits won't be set.

RSV(5):

Reserved.

HDR_LEN (10) - The length (bytes) of the header as parsed by the I350. In split mode when HBO (Header Buffer Overflow) is set in the Extended error field, the *HDR_LEN* can be greater then zero though nothing is written to the header buffer. In header replication mode, the *HDR_LEN* field does not reflect the size of the data actually stored in the header buffer because the I350 fills the buffer up to the size configured by *SRRCTL[n].BSIZEHEADER*, which might be larger than the header size reported here. This field is only valid in the first descriptor of a packet and should be ignored in all subsequent descriptors.

Note: When the packet is time stamped and the time stamp is placed at the beginning of the buffer the *RDESC.HDR_LEN* field is updated with the additional time stamp bytes (16 bytes). For further information see Section 7.1.6.

Packet types supported by the header split and header replication are listed in Appendix B.1. Other packet types are posted sequentially in the host packet buffer. Each line in the following table has an enable bit in the PSRTYPE register. When one of the bits is set, the corresponding packet type is split. If the bit is not set, a packet matching the header layout is not split.

Header split and replication is described in Section 7.1.5 while the packet types for this functionality are enabled by the *PSRTYPE[n]* registers (Section 8.10.3).

Note: The header of a fragmented IPv6 packet is defined before the fragmented extension header.

SPH (1) - Split Header - When set, indicates that the *HDR_LEN* field reflects the length of the header found by hardware. If cleared, the *HDR_LEN* field should be ignored.In the case where *SRRCTL[n].DESCTYPE* is set to *Header replication mode*, *SPH* bit is set but the *HDR_LEN* field does not reflect the size of the data actually stored in the header buffer, because the I350 fills the buffer up to the size configured by *SRRCTL[n].BSIZEHEADER*.

RSS Hash / {Fragment Checksum, IP identification} (32)

This field has multiplexed functionality according to the received packet type (reported on the *Packet Type* field in this descriptor) and device setting.

Fragment Checksum (16-Bit; 63:48)

The fragment checksum word contains the unadjusted one's complement checksum of the IP payload and is used to offload checksum verification for fragmented UDP packets as described in Section 7.1.7.3. This field is mutually exclusive with the RSS hash. It is enabled when the *RXCSUM.PCSD* bit is cleared and the *RXCSUM.IPPCSE* bit is set.

IP identification (16-Bit; 47:32)

The IP identification word identifies the IP packet to whom this fragment belongs and is used to offload checksum verification for fragmented UDP packets as described in Section 7.1.7.3. This field is mutually exclusive with the RSS hash. It is enabled when the *RXCSUM.PCSD* bit is cleared and the *RXCSUM.IPPCSE* bit is set.

RSS Hash Value (32)

The RSS hash value is required for RSS functionality as described in Section 7.1.2.8. This bit is mutually exclusive with the fragment checksum. It is enabled when the *RXCSUM.PCSD* bit is set.

Extended Status (20)

Status information indicates whether the descriptor has been used and whether the referenced buffer is the last one for the packet. Table 7-14 lists the extended status word in the last descriptor of a packet (*EOP* is set). Table 7-15 lists the extended status word in any descriptor but the last one of a packet (*EOP* is cleared).

Table 7-14 Receive Status (RDESC.STATUS) Layout of the Last Descriptor

19	18	17	16	15	14	13	12	11	10
BMC	LB	Rsv	TS	TSIP	Reserved		Strip CRC	LLINT	UDPV

Table 7-14 Receive Status (RDESC.STATUS) Layout of the Last Descriptor

19	18	17	16	15	14	13	12	11	10
VEXT	Rsv	PIF	IPCS	L4I	UDPCS	VP	Rsv	EOP	DD
9	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Table 7-15 Receive Status (RDESC.STATUS) Layout of Non-Last Descriptor

19 2	1	0
Reserved	EOP = 0b	DD

BMC (19) - Packet received from BMC. The BMC bit is set to indicate the packet was sent by the local BMC. Bit is cleared if packet arrives from the network. For more details see Section 10.4.

LB (18) - This bit provides a loopback status indication meaning that this packet is sent by a local virtual machine (VM to VM switch indication). For additional details see Section 7.8.3.

TS (16) - Time Stamped Packet (Time Sync). The Time Stamp bit is set to indicate that the device recognized a Time Sync packet and time stamped it in the RXSTMPL/H time stamp registers (See Section 7.9.3.2 and Section 7.9.2.1).

TSIP (15) - Timestamp in packet. The Timestamp In Packet bit is set to indicate that the received packet arrival time was captured by the hardware and the timestamp was placed in the receive buffer. For further details see Section 7.1.6.

Reserved (2, 8, 14:13, 17) - Reserved at zero.

PIF (7), IPCS(6), UDPCS(4), VP(3), EOP (1), DD (0) - These bits are described in the legacy descriptor format in Section 7.1.4.

L4I (5) - This bit indicates that an L4 integrity check was done on the packet, either TCP checksum, UDP checksum or SCTP CRC checksum. This bit is valid only for the last descriptor of the packet. An error in the integrity check is indicated by the L4E bit in the error field. The type of check done can be induced from the packet type bits 4, 5 and 6. If bit 4 is set, a TCP checksum was done. If bit 5 is set a UDP checksum was done, and if bit 6 is set, a SCTP CRC checksum was done.

VEXT (9) - First VLAN is found on a double VLAN packet. This bit is valid only when *CTRL_EXT.EXT_VLAN* is set. For more details see Section 7.4.5.

UDPV (10) - This bit indicates that the incoming packet contains a valid (non-zero value) checksum field in an incoming first fragment UDP IPv4 packet. This means that the Fragment Checksum field in the receive descriptor contains the IP payload checksum as described in Section 7.1.7.2. When this field is cleared in the first fragment that contains the UDP header, means that the packet does not contain a valid UDP checksum and the fragment checksum field in the Rx descriptor should be ignored. This field is always cleared in incoming fragments that do not contain the UDP header or in non fragmented packet.

LLINT (11) - This bit indicates that the packet caused an immediate interrupt via the low latency interrupt mechanism.

Strip CRC (12) - This bit indicates that Ethernet CRC has been stripped from incoming packet. Strip CRC operation is defined by the *RCTL.SECRC* bit for non virtualized mode and *DVMOLR.STRCRC* in virtualized mode.

Note: The non used field (*RCTL.SECRC* bit for virtualized mode and *DVMOLR.STRCRC* in non virtualized mode) should be cleared.

Extended Error (12)

Table 7-16 and the text that follows describes the possible errors reported by hardware.

Table 7-16 Receive Errors (RDESC.ERRORS) Layout

11	10	9	8 7	6 4	3	2 0
RXE	IPE	L4E	Reserved	Reserved	HBO	Reserved

RXE (bit 11)

RXE is described in the legacy descriptor format in Section 7.1.4.

IPE (bit 10)

The IPE error indication is described in the legacy descriptor format in Section 7.1.4.

L4E (bit 9)

L4 error indication - When set, indicates that hardware attempted to do an L4 integrity check as described in the *L4I* bit, but the check failed.

Reserved (bits 8:7)

Reserved (bits 6:4)

HBO (bit 3) - Header Buffer Overflow

Note: The *HBO* bit is relevant only if *SPH* is set.

1. In both header replication modes, *HBO* is set if the header size (as calculated by hardware) is bigger than the allocated buffer size (*SRRCTL.BSIZEHEADER*) but the replication still takes place up to the header buffer size. Hardware sets this bit in order to indicate to software that it needs to allocate bigger buffers for the headers.
2. In header split mode, when *SRRCTL[n] BSIZEHEADER* is smaller than *HDR_LEN*, then *HBO* is set to 1b, In this case, the header is not split. Instead, the header resides within the host packet buffer. The *HDR_LEN* field is still valid and equal to the calculated size of the header. However, the header is not copied into the header buffer.

Note: Most error information appears only when the *store–bad–packet* bit (*RCTL.SBP*) is set and a bad packet is received.

Reserved (bits 2:0) - Reserved

PKT_LEN (16)

Number of bytes existing in the host packet buffer

The length covers the data written to a receive buffer including CRC bytes (if any). Software must read multiple descriptors to determine the complete length for packets that span multiple receive buffers. If *SRRCTL.DESC_TYPE* = 4 (advanced descriptor header replication large packet only) and the total packet length is smaller than the size of the header buffer (no replication is done), this field continues to reflect the size of the packet, although no data is written to the packet buffer. Otherwise, if the buffer is not split because the header is bigger than the allocated header buffer, this field reflects the size of the data written to the first packet buffer (header and data).

Note: When the packet is time stamped and the time stamp is placed at the beginning of the buffer, the *RDESC.PKT_LEN* field is updated with the additional time stamp bytes (16 bytes). For further information see Section 7.1.6.

VLAN Tag (16)

These bits are described in the legacy descriptor format in Section 7.1.4.

7.1.4.3 Receive Descriptor Fetching

The fetching algorithm attempts to make the best use of PCIe bandwidth by fetching a cache-line (or more) descriptor with each burst. The following paragraphs briefly describe the descriptor fetch algorithm and the software control provided.

When the *RXDCTL[n].ENABLE* bit is set and the on-chip descriptor cache is empty, a fetch happens as soon as any descriptors are made available (Host increments the *RDT[n]* tail pointer). When the on-chip buffer is nearly empty (defined by *RXDCTL.PTHRESH*), a prefetch is performed each time enough valid descriptors (defined by *RXDCTL.HTHRESH*) are available in host memory.

When the number of descriptors in host memory is greater than the available on-chip descriptor cache, the I350 might elect to perform a fetch that is not a multiple of cache-line size. Hardware performs this non-aligned fetch if doing so results in the next descriptor fetch being aligned on a cache-line boundary. This enables the descriptor fetch mechanism to be more efficient in the cases where it has fallen behind software.

All fetch decisions are based on the number of descriptors available and do not take into account any split of the transaction due to bus access limitations.

Note: The I350 NEVER fetches descriptors beyond the descriptor tail pointer.

7.1.4.4 Receive Descriptor Write-Back

Processors have cache-line sizes that are larger than the receive descriptor size (16 bytes). Consequently, writing back descriptor information for each received packet would cause expensive partial cache-line updates. A receive descriptor packing mechanism minimizes the occurrence of partial line write-backs.

To maximize memory efficiency, receive descriptors are packed together and written as a cache-line whenever possible. Descriptors write-backs accumulate and are opportunistically written out in cache line-oriented chunks, under the following scenarios:

- *RXDCTL[n].WTHRESH* descriptors have been used (the specified maximum threshold of unwritten used descriptors has been reached).
- The receive timer expires (*EITR*) - in this case all descriptors are flushed ignoring any cache-line boundaries.
- Explicit software flush (*RXDCTL.SWFLS*).
- Dynamic packets - if at least one of the descriptors that are waiting for write-back are classified as packets requiring immediate notification the entire queue is flushed out.

When the number of descriptors specified by *RXDCTL[n].WTHRESH* have been used, they are written back regardless of cache-line alignment. It is therefore recommended that *RXDCTL[n].WTHRESH* be a multiple of cache-line size. When the receive timer (*EITR*) expires, all used descriptors are forced to be written back prior to initiating the interrupt, for consistency. Software might explicitly flush accumulated descriptors by writing the *RXDCTL[n]* register with the *SWFLS* bit set.

When the I350 does a partial cache-line write-back, it attempts to recover to cache-line alignment on the next write-back.

For applications where the latency of received packets is more important than the bus efficiency and the CPU utilization, an *EITR* value of zero may be used. In this case, each receive descriptor will be written to the host immediately. If *RXDCTL[n].WTHRESH* equals zero, then each descriptor will be written back separately, otherwise, write back of descriptors may be coalesced if descriptor accumulates in the internal descriptor ring due to bandwidth constrains.

All write-back decisions are based on the number of descriptors available and do not take into account any split of the transaction due to bus access limitations.

7.1.4.5 Receive Descriptor Ring Structure

Figure 7-8 shows the structure of each of the 8 receive descriptor rings. Hardware maintains 8 circular queues of descriptors and writes back used descriptors just prior to advancing the head pointer(s). Head and tail pointers wrap back to base when size descriptors have been processed.

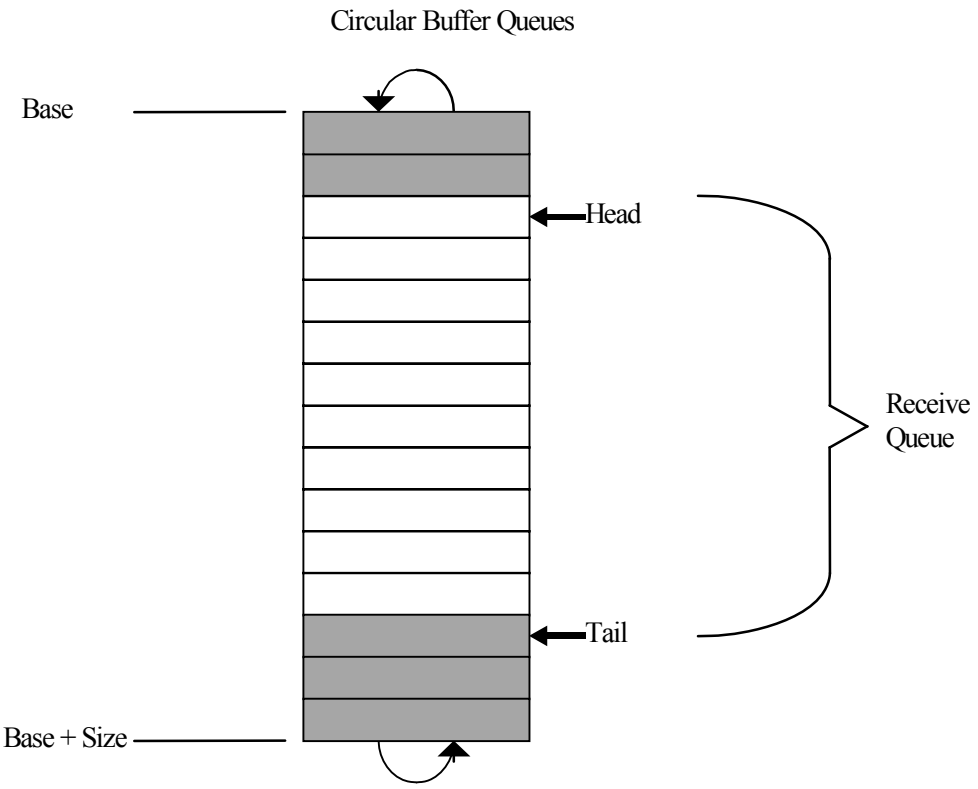


Figure 7-8 Receive Descriptor Ring Structure

Software inserts receive descriptors by advancing the tail pointer(s) to refer to the address of the entry just beyond the last valid descriptor. This is accomplished by writing the descriptor tail register(s) with the offset of the entry beyond the last valid descriptor. The hardware adjusts its internal tail pointer(s) accordingly. As packets arrive, they are stored in memory and the head pointer(s) is incremented by hardware. When the head pointer(s) is equal to the tail pointer(s), the queue(s) is empty. Hardware stops storing packets in system memory until software advances the tail pointer(s), making more receive buffers available.

The receive descriptor head and tail pointers reference to 16-byte blocks of memory. Shaded boxes in Figure 7-8 represent descriptors that have stored incoming packets but have not yet been recognized by software. Software can determine if a receive buffer is valid by reading the descriptors in memory. Any descriptor with a non-zero DD value has been processed by the hardware and is ready to be handled by the software.

Note: The head pointer points to the next descriptor that is written back. After the descriptor write-back operation completes, this pointer is incremented by the number of descriptors written back. Hardware owns all descriptors between [head... tail]. Any descriptor not in this range is owned by software.

The receive descriptor rings are described by the following registers:

- Receive Descriptor Base Address (*RDBA7* to *RDBA0*) register:
  This register indicates the start of the descriptor ring buffer. This 64-bit address is aligned on a 16-byte boundary and is stored in two consecutive 32-bit registers. Note that hardware ignores the lower 4 bits.
- Receive Descriptor Length (*RDLEN7* to *RDLEN0*) registers:
  This register determines the number of bytes allocated to the circular buffer. This value must be a multiple of 128 (the maximum cache-line size). Since each descriptor is 16 bytes in length, the total number of receive descriptors is always a multiple of eight.
- Receive Descriptor Head (*RDH7* to *RDH0*) registers:
  This register holds a value that is an offset from the base and indicates the in-progress descriptor. There can be up to 64 KB, 8 KB descriptors in the circular buffer. Hardware maintains a shadow copy that includes those descriptors completed but not yet stored in memory.
- Receive Descriptor Tail (*RDT7* to *RDT0*) registers:
  This register holds a value that is an offset from the base and identifies the location beyond the last descriptor hardware can process. This is the location where software writes the first new descriptor.

If software statically allocates buffers, uses legacy receive descriptors, and uses memory read to check for completed descriptors, it has to zero the status byte in the descriptor before bumping the tail pointer to make it ready for reuse by hardware. Zeroing the status byte is not a hardware requirement but is necessary for performing an in-memory scan.

All the registers controlling the descriptor rings behavior should be set before receive is enabled, apart from the tail registers that are used during the regular flow of data.

7.1.4.5.1 Low Receive Descriptors Threshold

As described above, the size of the receive queues is measured by the number of receive descriptors. During run time the software processes completed descriptors and then increments the Receive Descriptor Tail registers (*RDT*). At the same time, the hardware may post new packets received from the LAN that increments the Receive Descriptor Head registers (*RDH*) for each used descriptor.

The number of usable (free) descriptors for the hardware is the distance between Tail and Head registers. When the Tail reaches the Head, there are no free descriptors and further packets may be either dropped or block the receive FIFO. In order to avoid this behavior, the I350 may generate a low latency interrupt (associated with the relevant receive queue) once the amount of free descriptors is less or equal than the threshold. The threshold is defined in 16 descriptors granularity per queue in the *SRRCTL[n].RDMTS* field.

7.1.5 Header Splitting and Replication

7.1.5.1 Purpose

This feature consists of splitting or replicating packet's header to a different memory space. This helps the host to fetch headers only for processing: headers are replicated through a regular snoop transaction in order to be processed by the host CPU. It is recommended to perform this transaction with the DCA feature enabled (see Section 8.13) or in conjunction with a software-prefetch.

The packet (header and payload) is stored in memory through a (optionally) non-snoop transaction. Later, a data movement engine transaction moves the payload from the software device driver buffer to application memory or it is moved using a normal memory copy operation.

The I350 supports header splitting in several modes:

- Legacy mode: legacy descriptors are used; headers and payloads are not split.
- Advanced mode, no split: advanced descriptors are in use; header and payload are not split.
- Advanced mode, split: advanced descriptors are in use; header and payload are split to different buffers. If the packet cannot be split, only the packet buffer is used.
- Advanced mode, replication: advanced descriptors are in use; header is replicated in a separate buffer and also in a payload buffer.
- Advanced mode, replication, conditioned by packet size: advanced descriptors are in use; replication is performed only if the packet is larger than the header buffer size.

7.1.5.2 Description

In Figure 7-9 and Figure 7-10, the header splitting and header replication modes are shown.

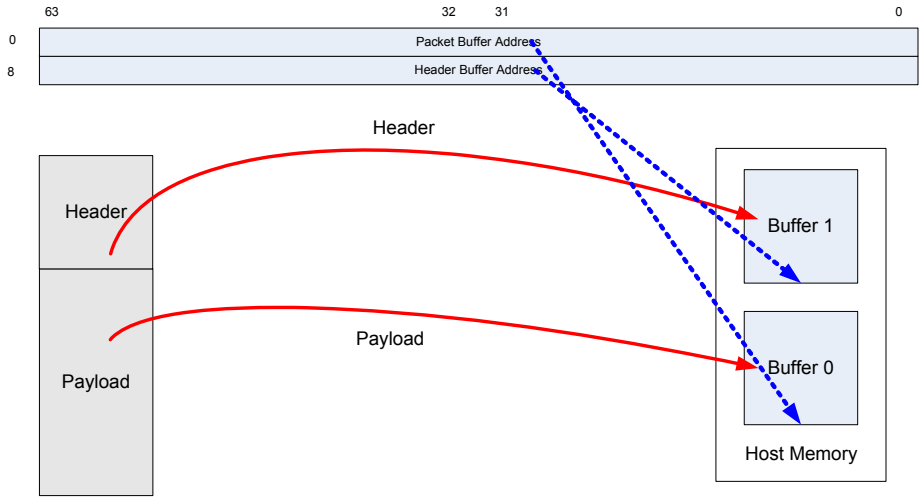


Figure 7-9 Header Splitting

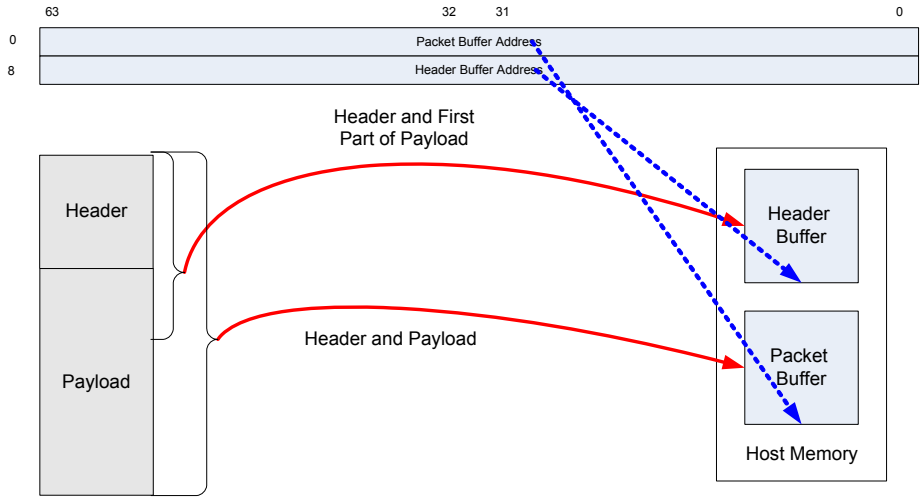


Figure 7-10 Header Replication

The physical address of each buffer is written in the *Buffer Addresses* fields. The sizes of these buffers are statically defined by *BSIZEPACKET* and *BSIZEHEADER* fields in the *SRRCTL[n]* registers.

The packet buffer address includes the address of the buffer assigned to the replicated packet, including header and data payload portions of the received packet. In the case of a split header, only the payload is included.

The header buffer address includes the address of the buffer that contains the header information. The receive DMA module stores the header portion of the received packets into this buffer.

The I350 uses the packet replication or splitting feature when the *SRRCTL[n].DESCTYPE* is larger than one. The software device driver must also program the buffer sizes in the *SRRCTL[n]* registers.

When header split is selected, the packet is split only on selected types of packets. A bit exists for each option in *PSRTYPE[n]* registers so several options can be used in conjunction with them. If one or more bits are set, the splitting is performed for the corresponding packet type. See Appendix B.1 for details on the possible headers type supported).

The following table lists the behavior of the I350 in the different modes.

Table 7-17 I350 Split/Replicated Header Behavior

DESCTYPE	Condition	SPH	HBO	PKT_LEN	HDR_LEN	Header and Payload DMA
Split	1. Header can't be decoded	0b	0b	Min(Packet length, *BSIZEPACKET*)	N/A	Header + Payload ⇨ Packet buffer
	2. Header <= *BSIZEHEADER*	1b	0b	Min(Payload length, *BSIZEPACKET*)[1]	Header size	Header ⇨ Header buffer Payload ⇨ Packet buffer
	3. Header > *BSIZEHEADER*	1b	1b	Min(Packet length, *BSIZEPACKET*)	Header size[2]	Header + Payload ⇨ Packet buffer

Table 7-17 I350 Split/Replicated Header Behavior

DESCTYPE	Condition	SPH	HBO	PKT_LEN	HDR_LEN	Header and Payload DMA
Replicate	1. Header can't be decoded	0b[3]	0b	Min(Packet length, *BSIZEPACKET*)	N/A	(Header + Payload) (partial[5]) ⇨ Header buffer Header + Payload ⇨ Packet buffer
	2. Packet length <= *BSIZEHEADER*	1b[3]	0b	Min(Packet length, *BSIZEPACKET*)	Header size	Header + Payload ⇨ Header buffer Header + Payload ⇨ Packet buffer
	3. Packet length > *BSIZEHEADER*	1b[3]	0b/1b[4]	Min(Packet length, *BSIZEPACKET*)	Header size	Header + Payload (partial[5]) ⇨ Header buffer Header + Payload ⇨ Packet buffer
Replicate Large Packet only	1. Header can't be decoded	0b[3]	0b	Min(Packet length, *BSIZEPACKET*)	N/A	(Header + Payload) (partial[5]) ⇨ Header buffer Header + Payload ⇨ Packet buffer
	2. Packet length <= *BSIZEHEADER*	1b[3]	0b	Packet length	Header size	Header + Payload ⇨ Header buffer
	2. Packet length > *BSIZEHEADER*	1b[3]	0b/1b[5]	Min(Packet length, *BSIZEPACKET*)	Header size	(Header + Payload) (partial[5]) ⇨ Header buffer Header + Payload ⇨ Packet buffer

1. In a header only packet (such as TCP ACK packet), the PKT_LEN is zero.
2. The HDR_LEN doesn't reflect the actual data size stored in the Header buffer. It reflects the header size determined by the parser. When timestamp in packet is enabled header size reflects the additional 16 bytes of the timestamp.
3. In replicate mode if *SPH* = 0b due to no match to any of the headers selected in the *PSRTYPE[n]* register, then the header size is not relevant. In any case, even if *SPH* = 1b due to match to one of the headers selected in the *PSRTYPE[n]* register, the *HDR_LEN* doesn't reflect the actual data size stored in the header buffer.
4. HBO is 1b if the header size is bigger than *BSIZEHEADER* and zero otherwise.
5. Partial means up to *BSIZEHEADER*

Software Notes:

- If *SRRCTL[n].NSE* is set, all buffers' addresses in a packet descriptor must be word aligned.
- Packet header can't span across buffers, therefore, the size of the header buffer must be larger than any expected header size. Otherwise, only the part of the header fitting the header buffer is replicated. In the case of header split mode (*SRRCTL[n].DESCTYPE* = 010b), a packet with a header larger than the header buffer is not split.
- Section B.1 describes the details of the split/replicate conditions for different types of headers according to the settings of the *PSRTYPE* register values.

7.1.6 Receive Packet Timestamp in Buffer

The I350 supports adding an optional tailored header before the MAC header of the packet in the receive buffer. The 64 MSB bits of the 128 bit tailored header include a timestamp composed of the packet reception time measured in the *SYSTIML* (Low DW) and *SYSTIMH* (High DW) registers (See Section 7.9.3.1 for further information on *SYSTIML/H* operation). The 64 LSB bits of the tailored header are reserved.

The timestamp information is placed in Networking order (Big Endian) format as depicted in Table 7-18.

Table 7-18 Timestamp Layout in Buffer

0 3	4 7	8 11	12 15	16...
Reserved (0x0)	Reserved (0x0)	SYSTIMH	SYSTIML	Received Packet

When the *TSAUXC.Disable systime* bit is cleared and the *SRRCTL[n].Timestamp* bit is set to 1, packets received to the queue will be time stamped if they meet one of the following conditions:

— Meet the criteria defined in the *TSYNCRXCTL.Type* field (See Section 8.16.1 and Section 8.16.26).

— Match the value defined in one of the *ETQF* registers with the *1588 time stamp* bit set (See Section 7.1.2.4) if *TSYNCRXCTL.Type* field defines time stamping of L2 packets.

— Match a 2-tuple filter with the *TTQF.1588 time stamp* set (See Section 7.1.2.5) if *TSYNCRXCTL.Type* field defines time stamping of L4 packets.

When detecting a receive packet that should be time stamped, the I350 will:

- Place a 64 bit timestamp, indicating the time a packet was received by the MAC, at the beginning of the receive buffer before the received packet.
- Set the *TSIP* bit in the *RDESC.STATUS* field of the last receive descriptor.
- Update the *RDESC.Packet Type* field in the last receive descriptor. Value in this field enables identifying that this is a PTP (Precision Time Protocol) packet (this indication is only relevant for L2 packets).
- Update the *RDESC.HDR_LEN* and *RDESC.PKT_LEN* values to include size of timestamp.

Software driver should take into account the additional size of the timestamp when preparing the receive descriptors for the relevant queue.

7.1.7 Receive Packet Checksum and SCTP CRC Off Loading

The I350 supports the off loading of four receive checksum calculations: packet checksum, fragment payload checksum, the IPv4 header checksum, and the TCP/UDP checksum. In addition, SCTP CRC32 calculation is supported as described in Section 7.1.7.4

The packet checksum and the fragment payload checksum shares the same location as the RSS field and is reported in the receive descriptor when the *RXCSUM.PCSD* bit is cleared. If the *RXCSUM.IPPCSE* is set, the Packet checksum is aimed to accelerate checksum calculation of fragmented UDP packets. Please refer to Section 7.1.7.3 for a detailed explanation. If *RXCSUM.IPPCSE* is cleared (the default value), the checksum calculation that is reported in the Rx Packet checksum field is the unadjusted 16-bit one's complement of the packet as described in Section 7.1.7.2

For supported packet/frame types, the entire checksum calculation can be off loaded to the I350. If *RXCSUM.IPOFLD* is set to 1b, the I350 calculates the IPv4 checksum and indicates a pass/fail indication to software via the IPv4 *Checksum Error* bit (*RDESC.IPE*) in the *Error* field of the receive descriptor. Similarly, if *RXCSUM.TUOFLD* is set to 1b, the I350 calculates the TCP or UDP checksum and indicates a pass/fail condition to software via the TCP/UDP *Checksum Error* bit (*RDESC.L4E*). These error bits are valid when the respective status bits indicate the checksum was calculated for the packet (*RDESC.IPCS* and *RDESC.L4CS*, respectively).

If neither *RXCSUM.IPOFLD* nor *RXCSUM.TUOFLD* are set, the *Checksum Error* bits (*IPE* and *L4E*) are 0b for all packets.

Supported frame types:

- Ethernet II
- Ethernet SNAP

Table 7-19 Supported Receive Checksum Capabilities

Packet Type	Hardware IP Checksum Calculation	Hardware TCP/ UDP Checksum Calculation	Hardware SCTP CRC calculation
IPv4 packets.	Yes	Yes	Yes
IPv6 packets.	No (n/a)	Yes	Yes
IPv6 packet with next header options: • Hop-by-hop options • Destinations options (without Home option) • Destinations options (with Home option) • Routing (with Segments Left zero) • Routing (with Segments Left > zero) • Fragment	 No (n/a) No (n/a) No (n/a) No (n/a) No (n/a) No (n/a)	 Yes Yes No Yes No No	Yes
IPv4 tunnels: • IPv4 packet in an IPv4 tunnel. • IPv6 packet in an IPv4 tunnel.	Yes (External - as if L3 only) Yes (IPv4)	No Yes[1]	No Yes
IPv6 tunnels: • IPv4 packet in an IPv6 tunnel. • IPv6 packet in an IPv6 tunnel.	 No No	 No No	 No No
Packet is an IPv4 fragment.	Yes	No[2]	No
Packet is greater than 1518, 1522 or 1526 bytes (LPE=1b)[3].	Yes	Yes	Yes
Packet has 802.3ac tag.	Yes	Yes	Yes
IPv4 packet has IP options (IP header is longer than 20 bytes).	Yes	Yes	Yes
Packet has TCP or UDP options.	Yes	Yes	Yes
IP header's protocol field contains a protocol number other than TCP or UDP or SCTP.	Yes	No	No

1. The IPv6 header portion can include supported extension headers as described in the "IPv6 packet with next header options" row.
2. UDP checksum of first fragment is supported.
3. Depends on number of VLAN tags.

7.1.7.1 Filters Details

The previous table lists general details about what packets are processed. In more detail, the packets are passed through a series of filters to determine if a receive checksum is calculated:

7.1.7.1.1 MAC Address Filter

This filter checks the MAC destination address to be sure it is valid (such as IA match, broadcast, multicast, etc.). The receive configuration settings determine which MAC addresses are accepted. See the various receive control configuration registers such as *RCTL* (*RCTL.UPE*, *RCTL.MPE*, *RCTL.BAM*), MTA, RAL, and RAH.

7.1.7.1.2 SNAP/VLAN Filter

This filter checks the next headers looking for an IP header. It is capable of decoding Ethernet II, Ethernet SNAP, and IEEE 802.3ac headers. It skips past any of these intermediate headers and looks for the IP header. The receive configuration settings determine which next headers are accepted. See the various receive control configuration registers such as *RCTL* (*RCTL.VFE*), *VET*, and *VFTA*.

7.1.7.1.3 IPv4 Filter

This filter checks for valid IPv4 headers. The version field is checked for a correct value (4).

IPv4 headers are accepted if they are any size greater than or equal to five (Dwords). If the IPv4 header is properly decoded, the IP checksum is checked for validity. The *RXCSUM.IPOFLD* bit must be set for this filter to pass.

7.1.7.1.4 IPv6 Filter

This filter checks for valid IPpv6 headers, which are a fixed size and have no checksum. The IPv6 extension headers accepted are: hop-by-hop, destination options, and routing. The maximum size next header accepted is 16 Dwords (64 bytes).

7.1.7.1.5 IPv6 Extension Headers

IPv4 and TCP provide header lengths, which enable hardware to easily navigate through these headers on packet reception for calculating checksum and CRCs, etc. For receiving IPv6 packets; however, there is no IP header length to help hardware find the packet's ULP (such as TCP or UDP) header. One or more IPv6 extension headers might exist in a packet between the basic IPv6 header and the ULP header. The hardware must skip over these extension headers to calculate the TCP or UDP checksum for received packets.

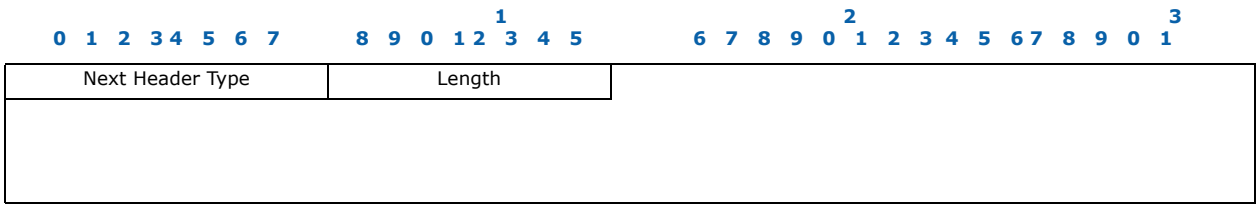
The IPv6 header length without extensions is 40 bytes. The IPv6 field *Next Header Type* indicates what type of header follows the IPv6 header at offset 40. It might be an upper layer protocol header such as TCP or UDP (*Next Header Type* of 6 or 17, respectively), or it might indicate that an extension header follows. The final extension header indicates with its *Next Header Type* field the type of ULP header for the packet.

IPv6 extension headers have a specified order. However, destinations must be able to process these headers in any order. Also, IPv6 (or IPv4) might be tunneled using IPv6, and thus another IPv6 (or IPv4) header and potentially its extension headers might be found after the extension headers.

The IPv4 *Next Header Type* is at byte offset nine. In IPv6, the first *Next Header Type* is at byte offset six.

All IPv6 extension headers have the *Next Header Type* in their first eight bits. Most have the length in the second eight bits (Offset Byte[1]) as shown:

Table 7-20 Typical IPv6 Extended Header Format (Traditional Representation)



The following table lists the encoding of the *Next Header Type* field and information on determining each header type's length. The IPv6 extension headers are not otherwise processed by the I350 so their details are not covered here.

Table 7-21 Header Type Encoding and Lengths

Header	Next Header Type	Header Length (Units are Bytes Unless Otherwise Specified)
IPv6	6	Always 40 bytes
IPv4	4	Offset Bits[7:4] Unit = 4 bytes
TCP	6	Offset Byte[12].Bits[7:4] Unit = 4 bytes
UDP	17	Always 8 bytes
Hop by Hop Options	0 (Note 1)	8+Offset Byte[1]
Destination Options	60	8+Offset Byte[1]
Routing	43	8+Offset Byte[1]
Fragment	44	Always 8 bytes
Authentication	51	8+4*(Offset Byte[1])
Encapsulating Security Payload	50	Note 3
No Next Header	59	Note 2

Notes:
1. Hop-by-hop options header is only found in the first *Next Header Type* of an IPv6 header.
2. When a *No Next Header* type is encountered, the rest of the packet should not be processed.
3. Encapsulated security payload - the I350 cannot offload packets with this header type.

Note that the I350 hardware acceleration does not support all IPv6 extension header types (refer to Table 7-19).

7.1.7.1.6 UDP/TCP Filter

This filter checks for a valid UDP or TCP header. The prototype next header values are 0x11 and 0x06, respectively. The *RXCSUM.TUOFLD* bit must be set for this filter to pass.

7.1.7.2 Packet Checksum

This feature allows raw checksum of part of the packet, independent of the protocol identified. This feature can not be used together with the receive UDP fragment checksum described in the next section.

The packet checksum is the 16-bit one's complement of the received packet, starting from the byte indicated by *RXCSUM.PCSS* (zero corresponds to the first byte of the packet).

For packets with a VLAN header, the packet checksum includes the header if VLAN striping is not enabled by the *CTRL.VME*. If a VLAN header strip is enabled using *CTRL.VME*, the packet checksum and the starting offset of the packet checksum exclude the VLAN header due to masking of VLAN header.

For example, for an Ethernet II frame encapsulated as an 802.3ac VLAN packet and *CTRL.VME* is set and with *RXCSUM.PCSS* set to 14, the packet checksum would include the entire encapsulated frame, excluding the 14-byte Ethernet header (DA, SA, type/length) and the 4-byte q-tag.

Note: If VLAN strip is enabled via the per queue *DVMOLR.STRVLAN* field, the packet checksum includes the VLAN header.

The packet checksum does not include the Ethernet CRC if the *RCTL.SECRC* bit is set.

Software must make the required offsetting computation (to remove the bytes that should not have been included and to include the pseudo-header) prior to comparing the packet checksum against the TCP checksum stored in the packet.

Note: The *RXCSUM.PCSS* value should point to a field that is before or equal to the IP header start. Otherwise the IP header checksum or TCP/UDP checksum is not calculated correctly.

7.1.7.3 Receive UDP Fragmentation Checksum

The I350 might provide receive fragmented UDP checksum offload. The I350 should be configured in the following manner to enable this mode:

The *RXCSUM.PCSD* bit should be cleared. The *Fragment Checksu*m and *IP Identification* fields are mutually exclusive with the RSS hash. When the RXCSUM.*PCSD* bit is cleared, *Fragment Checksum* and *IP Identification* are active instead of RSS hash.

The *RXCSUM.IPPCSE* bit should be set. This field enables the IP payload checksum enable that is designed for the fragmented UDP checksum.

The *RXCSUM.PCSS* field must be zero. The packet checksum start should be zero to enable auto-start of the checksum calculation. The following table lists the exact description of the checksum calculation.

The following table also lists the outcome descriptor fields for the following incoming packets types:

Table 7-22 Descriptor Fields

Incoming Packet Type	Fragment Checksum (if RXCSUM.PCSD is cleared)	UDPV	UDPCS / L4CS /L4I
Non IP Packet	Packet checksum	0b	0b / 0b /0b
IPv6 Packet	Packet checksum	0b	Depends on transport header.
Non fragmented IPv4 packet	Packet checksum	0b	Depends on transport header.
Fragmented IPv4, when not first fragment	The unadjusted one's complement checksum of the IP payload.	0b	1b / 0b / 0b
Fragmented IPv4, for the first fragment	Same as above	1 if the UDP header checksum is valid (not zero)	1b / 0b / 0b

Note: When the software device driver computes the 16-bit ones complement, the sum on the incoming packets of the UDP fragments, it should expect a value of 0xFFFF. Refer to Section 7.1.7 for supported packet formats.

7.1.7.4 SCTP Offload

If a receive packet is identified as SCTP, the I350 checks the CRC32 checksum of this packet if the *RXCSUM.CRCOFL* bit is set to 1b and identifies this packet as SCTP. Software is notified on the execution of the CRC check via the *L4I* bit in the *Extended Status* field of the Rx descriptor and is notified on detection of a CRC error via the *L4E* bit in the *Extended Error* field of the RX descriptor. The detection of a SCTP packet is indicated via the *SCTP* bit in the *packet Type* field of the Rx descriptor. The following SCTP packet format is expected to enable support of the SCTP CRC check:

Table 7-23 SCTP Header

0 1 2 3 4 5 6 7	1 8 9 0 1 2 3 4 5	2 6 7 8 9 0 1 2 3	3 4 5 6 7 8 9 0 1
Source Port		Destination Port	
Verification Tag			
Checksum			
Chunks 1...n			

7.2 Transmit Functionality

7.2.1 Packet Transmission

Output packets to be transmitted are created using pointer-length pairs constituting a descriptor chain (descriptor based transmission). Software forms transmit packets by assembling the list of pointer-length pairs, storing this information in one of the transmit descriptor rings, and then updating the adequate on-chip transmit tail pointer. The transmit descriptors and buffers are stored in host memory. Hardware typically transmits the packet only after it has completely fetched all the packet data from host memory and stored it into the on-chip transmit FIFO (store and forward architecture). This permits TCP or UDP checksum computation and avoids problems with PCIe under-runs. Another transmit feature of the I350 is TCP/UDP segmentation. The hardware has the capability to perform packet segmentation on large data buffers offloaded from the Network Stack. This feature is discussed in detail in Section 7.2.4.

In addition, the I350 supports SCTP offloading for transmit requests. See section Section 7.2.5.3 for details about SCTP.

Table 1-10 provides a high level description of all data/control transformation steps needed for sending Ethernet packets to the line.

7.2.1.1 Transmit Data Storage

Data is stored in buffers pointed to by the descriptors. The data can be aligned to arbitrary byte boundary with the maximum size per descriptor limited only to the maximum allowed packet size (9728 bytes). A packet typically consists of two (or more) buffers, one (or more) for the header and one for the actual data. Each buffer is referenced by a different descriptor. Some software implementations may copy the header(s) and packet data into one buffer and use only one descriptor per transmitted packet.

7.2.1.2 On-Chip Transmit Buffers

The I350 allocates by default a 20KB on-chip packet buffer per port. The buffers are used to store packets until they are transmitted on the line. The I350 utilizes a common memory structure for the on-chip transmit buffers allocated to the various ports. If a port is disabled, so that it can't be accessed by host and management, the freed buffer space can be allocated to the active ports via the "*Initialization Control 4*" EEPROM word. A port can be disabled by either:

1. Pin assertion (LAN0_DIS_N, LAN1_DIS_N, LAN2_DIS_N, LAN3_DIS_N for port 0 to 3 respectively) and setting the EEPROM bit *PHY_in_LAN_disable* in the "*Software Defined Pins Control*" word to 1 for the relevant port.
2. Setting EEPROM bit *LAN_DIS* in the "*Software Defined Pins Control*" word for the relevant port to 1.

Actual on-chip transmit buffer allocated to the port can be read in the *ITPBS register.*

7.2.1.3 On-Chip Descriptor Buffers

The I350 contains a 24 descriptor cache for each transmit queue used to reduce the latency of packet processing and to optimize the usage of the PCIe bandwidth by fetching and writing back descriptors in bursts. The fetch and writeback algorithm are described in Section 7.2.2.5 and Section 7.2.2.6.

7.2.1.4 Transmit Contexts

The I350 provides hardware checksum offload and TCP/UDP segmentation facilities. These features enable TCP and UDP packet types to be handled more efficiently by performing additional work in hardware, thus reducing the software overhead associated with preparing these packets for transmission. Part of the parameters used by these features is handled though context descriptors.

A context descriptor refers to a set of device registers loaded or accessed as a group to provide a particular function. The I350 supports 2x8 context descriptor sets (two per queue) per port on-chip. The transmit queues can contain transmit data descriptors (similar to the receive queue) as well as transmit context descriptors.

The contexts are queue specific and one context cannot be reused from one queue to another. This differs from the method used in previous devices that supported a pool of contexts to be shared between queues.

A transmit context descriptor differs from a data descriptor as it does not point to packet data. Instead, this descriptor provides the ability to write to the on-chip context register sets that support the transmit checksum offloading and the segmentation features of the I350.

The I350 supports one type of transmit context. This on-chip context is written with a transmit context descriptor DTYP=2 and is always used as context for transmit data descriptor DTYP=3.

The *IDX* field contains an index to one of the two queue contexts. Software must track what context is stored in each *IDX* location.

Each advanced data descriptor that uses any of the advanced offloading features must refer to a context.

Contexts can be initialized with a transmit context descriptor and then used for a series of related transmit data descriptors. The context, for example, defines the checksum and offload capabilities for a given type of TCP/IP flow. All packets of this type can be sent using this context.

Software is responsible for ensuring that a context is only overwritten when it is no longer needed. Hardware does not include any logic to manage the on-chip contexts; it is completely up to software to populate and then use the on-chip context table.

Note: Software should not queue more than 2 context descriptors in sequence without an intervening data descriptor, to achieve adequate performance.

Each context defines information about the packet sent including the total size of the MAC header (*TDESC.MACLEN*), the maximum amount of payload data that should be included in each packet (*TDESC.MSS*), UDP or TCP header length (*TDESC.L4LEN*), IP header length (*TDESC.IPLEN*), and information about what type of protocol (TCP, IP, etc.) is used. Other than TCP, IP (*TDESC.TUCMD*), most information is specific to the segmentation capability.

Because there are dedicated on-chip resources for contexts, they remain constant until they are modified by another context descriptor. This means that a context can be used for multiple packets (or multiple segmentation blocks) unless a new context is loaded prior to each new packet. Depending on the environment, it might be unnecessary to load a new context for each packet. For example, if most traffic generated from a given node is standard TCP frames, this context could be setup once and used for many frames. Only when some other frame type is required would a new context need to be loaded by software. This new context could use a different index or the same index.

This same logic can also be applied to the TCP/UDP segmentation scenario, though the environment is a more restrictive one. In this scenario, the host is commonly asked to send messages of the same type, TCP/IP for instance, and these messages also have the same Maximum Segment Size (MSS). In this instance, the same context could be used for multiple TCP messages that require hardware segmentation.

7.2.2 Transmit Descriptors

The I350 supports legacy descriptors and I350 advanced descriptors.

Legacy descriptors are intended to support legacy drivers to enable fast platform power up and to facilitate debug.

Note: These descriptors should not be used when advanced features such as virtualization are used. If legacy descriptors are used when virtualization is enabled such as when *TXSWC.Loopback* enable or *STATUS.VFE* or one of the *TXSWC.MACAS* bits or one of the *TXSWC.VLANAS* bits are set, the packets are ignored and not sent.

The Legacy descriptors are recognized as such based on the *DEXT* bit as discussed later in this section.

In addition, the I350 supports two types of advanced transmit descriptors:

1. Advanced Transmit Context Descriptor, *DTYP* = 0010b.
2. Advanced Transmit Data Descriptor, *DTYP* = 0011b.

Note: *DTYP* values 0000b and 0001b are reserved.

The transmit data descriptor (both legacy and advanced) points to a block of packet data to be transmitted. The advanced transmit context descriptor does not point to packet data. It contains control/context information that is loaded into on-chip registers that affect the processing of packets for transmission. The following sections describe the descriptor formats.

7.2.2.1 Legacy Transmit Descriptor Format

Legacy descriptors are identified by having bit 29 of the descriptor (*TDESC.DEXT*) set to 0b. In this case, the descriptor format is defined as shown in Table 7-24. Note that the address and length must be supplied by software. Also note that bits in the command byte are optional, as are the CSO, and CSS fields.

Table 7-24 Transmit Descriptor (TDESC) Fetch Layout - Legacy Mode

	63 48	47 40	39 36	35 32	31 24	23 16	15 0
0	Buffer Address [63:0]						
8	VLAN	CSS	Reserved	STA	CMD	CSO	Length

Table 7-25 Transmit Descriptor (TDESC) Write-Back Layout - Legacy Mode

	63 48	47 40	39 36	35 32	31 24	23 16	15 0
0	Reserved				Reserved		
8	VLAN	CSS	Reserved	STA	CMD	CSO	Length

Note: For frames that span multiple descriptors, the *VLAN*, *CSS*, *CSO*, *CMD.VLE*, *CMD.IC*, and *CMD.IFCS* are valid only in the first descriptors and are ignored in the subsequent ones.

7.2.2.1.1 Buffer Address (64)

Physical address of a data buffer in host memory that contains a portion of a transmit packet.

7.2.2.1.2 Length

Length (*TDESC.LENGTH*) specifies the length in bytes to be fetched from the buffer address provided.

The maximum length associated with any single legacy descriptor is 9728 bytes.

Descriptor length(s) might be limited by the size of the transmit FIFO. All buffers comprising a single packet must be able to be stored simultaneously in the transmit FIFO. For any individual packet, the sum of the individual descriptors' lengths must be below 9728 bytes.

Note: The maximum allowable packet size for transmits can change, based on the value written to the *DMA TX Max Allowable packet size* (*DTXMXPKTSZ*) register.

Descriptors with zero length (null descriptors) transfer no data. Null descriptors can only appear between packets and must have their *EOP* bits set.

If the *TCTL.PSP* bit is set, the total length of the packet transmitted, not including FCS should be at least 17 bytes. If bit is cleared the total length of the packet transmitted, not including FCS should be at least 60 bytes.

7.2.2.1.3 Checksum Offset and Start - CSO and CSS

A *Checksum Offset* (*TDESC.CSO*) field indicates where, relative to the start of the packet, to insert a TCP checksum if this mode is enabled. A *Checksum Start* (*TDESC.CSS*) field indicates where to begin computing the checksum.

Both CSO and CSS are in units of bytes and must be in the range of data provided to the I350 in the descriptors. For short packets that are not padded by software, CSS and CSO must be in the range of the unpadded data length, not the eventual padded length (64 bytes).

CSO must be set to the location of TCP checksum in the packet. CSS must be set to the beginning of the IP header or the L4 (TCP) header. Checksum calculation is not done if CSO or CSS are out of range. This occurs if (CSS > length) OR (CSO > length - 1).

In the case of an 802.1Q header, the offset values depend on the VLAN insertion enable (*VLE*) bit. If it is not set (VLAN tagging included in the packet buffers), the offset values should include the VLAN tagging. If this bit is set (VLAN tagging is taken from the packet descriptor), the offset values should exclude the VLAN tagging.

Note: Assuming CSS points to the beginning of the IP header, software must compute an offsetting entry to back out the bytes of the header that are not part of the IP pseudo header and should not be included in the TCP checksum and store it in the position where the hardware computed checksum is to be inserted. Hardware does not add the 802.1Q Ethertype or the VLAN field following the 802.1Q Ethertype to the checksum. So for VLAN packets, software can compute the values to back out only the encapsulated IP header packet and not the added fields.

UDP checksum calculation is not supported by the legacy descriptors. When using legacy descriptors the I350 is not aware of the L4 type of the packet and thus, does not support the translation of a checksum result of 0x0000 to 0xFFFF needed to differentiate between an UDP packet with a checksum of zero and an UDP packet without checksum.

Because the *CSO* field is eight bits wide, it puts a limit on the location of the checksum to 255 bytes from the beginning of the packet.

Hardware adds the checksum to the field at the offset indicated by the *CSO* field. Checksum calculations are for the entire packet starting at the byte indicated by the *CSS* field. A value of zero corresponds to the first byte in the packet.

CSS must be set in the first descriptor of the packet.

Table 7-26 Transmit Command (TDESC.CMD) Layout

7	6	5	4	3	2	1	0
RSV	VLE	DEXT	Rsv	RS	IC	IFCS	EOP

7.2.2.1.4 Command Byte - CMD

The CMD byte stores the applicable command and has the fields shown in Figure 7-26.

- RSV (bit 7) - Reserved
- VLE (bit 6) - VLAN Insertion Enable (See Table 7-27).
- DEXT (bit 5) - Descriptor Extension (0 for legacy mode)
- Reserved (bit 4) - Reserved
- RS (bit 3) - Report Status
- IC (bit 2) - Insert Checksum
- IFCS (bit 1) - Insert FCS
- EOP (bit 0) - End of Packet

VLE: Indicates that the packet is a VLAN packet. For example, hardware should add the VLAN Ethertype and an 802.1q VLAN tag to the packet.

Table 7-27 VLAN Tag Insertion Decision Table

VLE	Action
0b	Send generic Ethernet packet.
1b	Send 802.1Q packet; The VLAN data comes from the *VLAN* field of the TX descriptor.

RS: Signals the hardware to report the status information. This is used by software that does in-memory checks of the transmit descriptors to determine which ones are done. For example, if software queues up 10 packets to transmit, it can set the *RS* bit in the last descriptor of the last packet. If software maintains a list of descriptors with the *RS* bit set, it can look at them to determine if all packets up to (and including) the one with the *RS* bit set have been buffered in the output FIFO. Looking at the status byte and checking the *Descriptor Done* (*DD*) bit enables this operation. If *DD* is set, the descriptor has been processed. Refer to Table 7-28 for the layout of the status field.

IC: If set, requests hardware to add the checksum of the data from *CSS* to the end of the packet at the offset indicated by the *CSO* field.

IFCS: When set, hardware appends the MAC FCS at the end of the packet. When cleared, software should calculate the FCS for proper CRC check. There are several cases in which software must set IFCS:

- Transmitting a short packet while padding is enabled by the *TCTL.PSP* bit.
- Checksum offload is enabled by the *IC* bit in the TDESC.CMD.
- VLAN header insertion enabled by the *VLE* bit in the TDESC.CMD or by the *VMVIR[n]* registers.

EOP: When set, indicates this is the last descriptor making up the packet. Note that more than one descriptor can be used to form a packet.

Note: the 8257, 1VLE, IFCS, CSO, and IC must be set correctly only in the first descriptor of each packet. In previous silicon generations, some of these bits were required to be set in the last descriptor of a packet.

7.2.2.1.5 Status – STA

Table 7-28 Transmit Status (TDESC.STA) Layout

3	2	1	0
Reserved			DD

7.2.2.1.6 DD (Bit 0) - Descriptor Done Status

The DD bit provides the transmit status, when *RS* is set in the command: *DD* indicates that the descriptor is done and is written back after the descriptor has been processed.

Note: When head write back is enabled (*TDWBAL[n].Head_WB_En* = 1), there is no write-back of the *DD* bit to the descriptor. When using legacy Tx descriptors, Head writeback should not be enabled (*TDWBAL[n].Head_WB_En* = 0).

7.2.2.1.7 VLAN

The *VLAN* field is used to provide the 802.1q/802.1ac tagging information. The *VLAN* field is valid only on the first descriptor of each packet when the *VLE* bit is set and the *VMVIR[n].VLANA* register field is 0. The rule for VLAN tag is to use network ordering. The VLAN field is placed in the transmit descriptor in the following manner:

Table 7-29 VLAN Field (TDESC.VLAN) Layout

15 13	12	11 0
PRI	CFI	VLAN ID

- VLAN ID - the 12-bit tag indicating the VLAN group of the packet.
- Canonical Form Indication (CFI) - Set to zero for Ethernet packets.
- PRI - indicates the priority of the packet.

Note: The VLAN tag is sent in network order (also called big endian).

7.2.2.2 Advanced Transmit Context Descriptor

Table 7-30 Transmit Context Descriptor (TDESC) Layout - (Type = 0010b)

	63 40	39 32	31 16	15 9	8 0
0	Reserved	Reserved	VLAN	MACLEN	IPLEN

	63 48	47 40	39	38 36	35 30	29	28 24	23 20	19 9	8 0
8	MSS	L4LEN	RSV[1]	IDX	Reserved	DEXT	RSV[1]	DTYP	TUCMD	Reserved

1. RSV - Reserved

7.2.2.2.1 IPLEN (9)

IP header length. If an offload is requested, *IPLEN* must be greater than or equal to 20 and less than or equal to 511.

7.2.2.2.2 MACLEN (7)

This field indicates the length of the MAC header. When an offload is requested (either *TSE* or *IXSM* or *TXSM* is set), *MACLEN* must be larger than or equal to 14 and less than or equal to 127. This field should include only the part of the L2 header supplied by the software device driver and not the parts added by hardware. The following table lists the value of *MACLEN* in the different cases.

Table 7-31 MACLEN Values

SNAP	Regular VLAN	External VLAN	MACLEN
No	By hardware or no VLAN	No	14
No	By hardware or no VLAN	Yes	18
No	By software	No	18
No	By software	Yes	22
Yes	By hardware or no VLAN	No	22
Yes	By hardware or no VLAN	Yes	26

Table 7-31 MACLEN Values (Continued)

SNAP	Regular VLAN	External VLAN	MACLEN
Yes	By software	No	26
Yes	By software	Yes	30

VLAN (16) - 802.1Q VLAN tag to be inserted in the packet during transmission. This VLAN tag is inserted and needed only when a packet using this context has its *DCMD.VLE* bit set and the *VMVIR[n].VLANA* register field is 0. This field should include the entire 16-bit *VLAN* field including the *CFI* and *Priority* fields as shown in Table 7-29.

Note: The VLAN tag is sent in network order.

7.2.2.2.3 TUCMD (11)

Table 7-32 Transmit Command (TDESC.TUCMD) Layout

10 6	5	4	3 2	1	0
Reserved	Reserved	Reserved	L4T	IPV4	SNAP

- RSV (bit 10:6) - Reserved
- RSV (bit 5:4) - Reserved
- L4T (bit 3:2) - L4 Packet TYPE (00b: UDP; 01b: TCP; 10b: SCTP; 11b: Reserved)
- IPV4 (bit 1) - IP Packet Type: When 1b, IPv4; when 0b, IPv6
- SNAP (bit 0) - SNAP indication

7.2.2.2.4 DTYP(4)

Always 0010b for this type of descriptor.

7.2.2.2.5 DEXT(1)

Descriptor Extension (1b for advanced mode).

7.2.2.2.6 IDX (3)

Index into the hardware context table where this context is stored. In the I350 the 2 available register context sets per queue are accessed using the LSB bit and the two MSB bits are reserved and should always be 0.

7.2.2.2.7 L4LEN (8)

Layer 4 header length. If *TSE* is set in the data descriptor pointing to this context, this field must be greater than or equal to 12 and less than or equal to 255. Otherwise, this field is ignored.

7.2.2.2.8 MSS (16)

Controls the Maximum Segment Size (MSS). This specifies the maximum TCP payload segment sent per frame, not including any header or trailer. The total length of each frame (or section) sent by the TCP/UDP segmentation mechanism (excluding Ethernet CRC) as follows:

Total length is equal to:

```
MACLEN + 4(if VLAN is inserted) + IPLEN + L4LEN + MSS
```

A VLAN is inserted if VLE set and the VMVIR[n].VLANA register field is 00b or if VMVIR[n].VLANA register field is 01b.

The one exception is the last packet of a TCP/UDP segmentation, which is typically shorter.

MSS is ignored when *DCMD.TSE* is not set.

Note: The headers lengths must meet the following:

```
MACLEN + IPLEN + L4LEN <= 512
```

Note: The MSS value should be larger than 0 and the maximum MSS value should not exceed 9216 bytes (9KB) length.

The context descriptor requires valid data only in the fields used by the specific offload options. The following table lists the required valid fields according to the different offload options.

Table 7-33 Valid Field in Context vs. Required Offload

Required Offload			Valid Fields in Context						
TSE	TXSM	IXSM	VLAN[1]	L4LEN	IPLEN	MACLEN	MSS	L4T	IPV4
1b[2]	1b	X[3]	VLE	Yes	Yes	Yes	Yes	Yes	Yes
0b	1b	X[2]	VLE	No	Yes	Yes	No	Yes	Yes
0b	0b	1b	VLE	No	Yes	Yes	No	No	Yes
0b	0b	0b	No context required unless VLE is set.						

1. VLAN field is required only if VLE bit in TX Descriptor is set and the VMVIR[n].VLANA register field is 0.
2. If TSE is set, *TXSM* must be set to 1.
3. X - don’t care

7.2.2.3 Advanced Transmit Data Descriptor

Table 7-34 Advanced Transmit Data Descriptor (TDESD) Layout - (Type = 0011b)

0	Address[63:0]									
8	PAYLEN	POPTS	RSV[1]	IDX	STA	DCMD	DTYP	MAC	RSV[1]	DTALEN
	63 46	45 40	39	38 36	35 32	31 24	23 20	19 18	17 16	15 0

1. RSV - Reserved

Table 7-35 Advanced Tx descriptor write-back format

0	RSV[1]		
8	Reserved	STA	Reserved
	63 36	35 32	31 0

1. RSV - Reserved

Note: For frames that span multiple descriptors, all fields **apart** from *DCMD.EOP*, *DCMD.RS*, *DCMD.DEXT*, *DTALEN*, Address and DTYP are valid only in the first descriptor and are ignored in the subsequent ones.

7.2.2.3.1 Address (64)

Physical address of a data buffer in host memory that contains a portion of a transmit packet.

7.2.2.3.2 DTALEN (16)

Length in bytes of data buffer at the address pointed to by this specific descriptor.

Note: If the *TCTL.PSP* bit is set, the total length of the packet transmitted, not including FCS, should be at least 17 bytes. If bit is cleared the total length of the packet transmitted, not including FCS should be at least 60 bytes.

The maximum allowable packet size for transmits is based on the value written to the *DMA TX Max Allowable packet size* (*DTXMXPKTSZ*) register. Default value is 9,728 bytes.

7.2.2.3.3 MAC (2)

Table 7-36 Transmit Data (TDESD.MAC) Layout

1	0
1588	Reserved

- 1588 (bit 1) - IEEE1588 Timestamp packet.

7.2.2.3.4 DTYP (4)

0011b is the value for this descriptor type.

7.2.2.3.5 DCMD (8)

Table 7-37 Transmit Data (TDESD.DCMD) Layout

7	6	5	4	3	2	1	0
TSE	VLE	DEXT	Reserved	RS	Reserved	IFCS	EOP

- TSE (bit 7) - TCP/UDP Segmentation Enable
- VLE (bit 6) - VLAN Packet Enable
- DEXT (bit 5) - Descriptor Extension (1b for advanced mode)
- Reserved (bit 4)
- RS (bit 3) - Report Status
- Reserved (bit 2)
- IFCS (bit 1) - Insert FCS
- EOP (bit 0) - End Of Packet

TSE indicates a TCP/UDP segmentation request. When *TSE* is set in the first descriptor of a TCP packet, hardware must use the corresponding context descriptor in order to perform TCP segmentation. The type of segmentation applied is defined according to the *TUCMD.L4T* field in the context descriptor.

Note: It is recommended that *TCTL.PSP* be enabled when *TSE* is used since the last frame can be shorter than 60 bytes - resulting in a bad frame if TCTL.*PSP* is disabled.

VLE indicates that the packet is a VLAN packet and hardware must add the VLAN Ethertype and an 802.1q VLAN tag to the packet if the *VMVIR[n].VLANA* register field is 0.

Note: If *VLE* is set when the *VMVIR[n].VLANA* register field is not 0 the packet will be dropped.

DEXT must be 1b to indicate advanced descriptor format (as opposed to legacy).

RS signals hardware to report the status information. This is used by software that does in-memory checks of the transmit descriptors to determine which ones are done. For example, if software queues up 10 packets to transmit, it can set the *RS* bit in the last descriptor of the last packet. If software maintains a list of descriptors with the *RS* bit set, it can look at them to determine if all packets up to (and including) the one with the *RS* bit set have been buffered in the output FIFO. Looking at the status byte and checking the *DD* bit do this. If *DD* is set, the descriptor has been processed. Refer to the next section for the layout of the status field.

Note: Descriptors with zero length transfer no data.

IFCS, when set, hardware appends the MAC FCS at the end of the packet. When cleared, software should calculate the FCS for proper CRC check. There are several cases in which the hardware changes the packet, and thus the software must set *IFCS*:

- Transmitting a short packet while padding is enabled by the *TCTL.PSP* bit.
- Checksum offload is enabled by the either the *TXSM* or *IXSM* bits in the *TDESD.POPTS* field.
- VLAN header insertion enabled by the *VLE* bit in the *TDESD.DCMD* descriptor field when the *VMVIR[n].VLANA* register field is 0.
- TCP/UDP segmentation offload enabled by *TSE* bit in the *TDESD.DCMD*.

EOP indicates whether this is the last buffer for an incoming packet.

7.2.2.3.6 STA (4)

- Rsv (bits 1-3) - Reserved
- DD (bit 0) - Descriptor Done

7.2.2.3.7 IDX (3)

Index into the hardware context table to indicate which context should be used for this request. If no offload is required, this field is not relevant and no context needs to be initiated before the packet is sent. See Table 7-33 for details on type of transmit packet offloads that require a context reference.

7.2.2.3.8 POPTS (6)

Table 7-38 Transmit Data (TDESD.POPTS) Layout

5 3	2	1	0
Reserved	Reserved	TXSM	IXSM

- Reserved (bits 5:3)
- Reserved (bit 2)
- TXSM (bit 1) - Insert L4 Checksum
- IXSM (bit 0) - Insert IP Checksum

TXSM, when set to 1b, L4 checksum must be inserted. In this case, TUCMD.L4T in the context descriptor indicates whether the checksum is TCP, UDP, or SCTP.

When *DCMD.TSE* in TDESD is set, *TXSM* must be set to 1b.

If this bit is set, the packet should at least contain a TCP header.

IXSM, when set to 1b, indicates that IP checksum must be inserted. For IPv6 packets this bit must be cleared.

If the *DCMD.TSE* bit is set in data descriptor, and *TUCMD.IPV4* is set in context descriptor, *POPTS.IXSM* must be set to 1b as well.

If this bit is set, the packet should at least contain an IP header.

7.2.2.3.9 PAYLEN (18)

PAYLEN indicates the size (in byte units) of the data buffer(s) in host memory for transmission. In a single send packet, *PAYLEN* defines the entire packet size fetched from host memory. It does not include the fields that hardware adds such as: optional VLAN tagging, Ethernet CRC or Ethernet padding. When TCP or UDP segmentation offload is enabled (*DCMD.TSE* is set), *PAYLEN* defines the TCP/UDP payload size fetched from host memory.

Note: When a packet spreads over multiple descriptors, all the descriptor fields are only valid in the first descriptor of the packet, except for *RS*, which is always checked, *DTALEN* that reflects the size of the buffer in the current descriptor and *EOP*, which is always set at last descriptor of the series.

7.2.2.4 Transmit Descriptor Ring Structure

The transmit descriptor ring structure is shown in Figure 7-11. A set of hardware registers maintains each transmit descriptor ring in the host memory. New descriptors are added to the queue by software by writing descriptors into the circular buffer memory region and moving the tail pointer associated with that queue. The tail pointer points to one entry beyond the last hardware owned descriptor. Transmission continues up to the descriptor where head equals tail at which point the queue is empty.

Descriptors passed to hardware should not be manipulated by software until the head pointer has advanced past them.

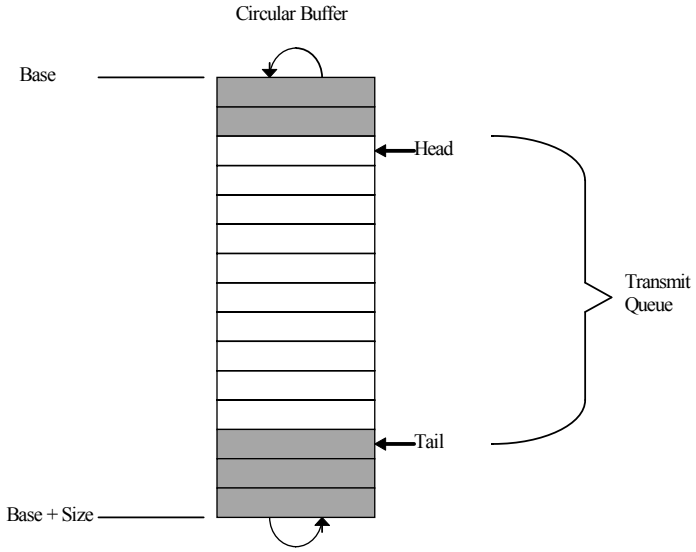


Figure 7-11 Transmit Descriptor Ring Structure

The shaded boxes in the figure represent descriptors that are not currently owned by hardware that software can modify.

The transmit descriptor ring is described by the following registers:

- Transmit Descriptor Base Address register (*TDBA 0-7*):
  This register indicates the start address of the descriptor ring buffer in the host memory; this 64-bit address is aligned on a 16-byte boundary and is stored in two consecutive 32-bit registers. Hardware ignores the lower four bits.
- Transmit Descriptor Length register (*TDLEN 0-7*):
  This register determines the number of bytes allocated to the circular buffer. This value must be zero modulo 128.
- Transmit Descriptor Head register (*TDH 0-7*):
  This register holds a value that is an offset from the base and indicates the in-progress descriptor. There can be up to 64 KB descriptors in the circular buffer. Reading this register returns the value of head corresponding to descriptors already loaded in the output FIFO. This register reflects the internal head of the hardware write-back process including the descriptor in the posted write pipe and might point further ahead than the last descriptor actually written back to the memory.
- Transmit Descriptor Tail register (*TDT 0-7*):

This register holds a value, which is an offset from the base, and indicates the location beyond the last descriptor hardware can process. This is the location where software writes the first new descriptor.

The driver should not handle to the I350 descriptors that describe a partial packet. Consequently, the number of descriptors used to describe a packet can not be larger than the ring size.

- Tx Descriptor Completion Write–Back Address High/Low Registers (*TDWBAH*/*TDWBAL* 0-7):

  These registers hold a value that can be used to enable operation of head writeback operation. When *TDWBAL.Head_WB_En* is set and the RS bit is set in the Tx descriptor, following corresponding data upload into packet buffer, the I350 writes the Transmit Descriptor Head value for this queue to the 64 bit address specified by the *TDWBAH* and *TDWBAL* registers. The Descriptor Head value is an offset from the base, and indicates the descriptor location hardware processed and software can utilize for new Transmit packets. See Section 7.2.3 for additional information.

The base register indicates the start of the circular descriptor queue and the length register indicates the maximum size of the descriptor ring. The lower seven bits of length are hard wired to 0b. Byte addresses within the descriptor buffer are computed as follows: address = base + (ptr * 16), where ptr is the value in the hardware head or tail register.

The size chosen for the head and tail registers permit a maximum of 65536 (64 KB) descriptors, or approximately 16 KB packets for the transmit queue given an average of four descriptors per packet.

Once activated, hardware fetches the descriptor indicated by the hardware head register. The hardware tail register points one descriptor beyond the last valid descriptor. Software can read and detect which packets have already been processed by hardware as follows:

- Read the head register to determine which packets (those logically before the head) have been transferred to the on-chip FIFO or transmitted. Note that this method is not recommended as races between the internal update of the head register and the actual write-back of descriptors might occur.
- Read the value of the head as stored at the address pointed by the *TDWBAH*/*TDWBAL* pair.
- Track the *DD* bits in the descriptor ring.

All the registers controlling the descriptor rings behavior should be set before transmit is enabled, apart from the tail registers which are used during the regular flow of data.

Note: Software can determine if a packet has been sent by either of three methods: setting the *RS* bit in the transmit descriptor command field or by performing a PIO read of the transmit head register, or by reading the head value written by the I350 to the address pointed by the *TDWBAL* and *TDWBAH* registers (see Section 7.2.3 for details). Checking the transmit descriptor *DD* bit or head value in memory eliminates a potential race condition. All descriptor data is written to the I/O bus prior to incrementing the head register, but a read of the head register could pass the data write in systems performing I/O write buffering. Updates to transmit descriptors use the same I/O write path and follow all data writes. Consequently, they are not subject to the race.

In general, hardware prefetches packet data prior to transmission. Hardware typically updates the value of the head pointer after storing data in the transmit FIFO.

7.2.2.5 Transmit Descriptor Fetching

When the *TXDCTL[n].ENABLE* bit is set and the on-chip descriptor cache is empty, a fetch happens as soon as any descriptors are made available (Host increments the *TDT[n]* tail pointer). The descriptor processing strategy for transmit descriptors is essentially the same as for receive descriptors except that a different set of thresholds are used. The number of on-chip transmit descriptors per queue is 24. When there is an on-chip descriptor buffer empty, a descriptor fetch happens as soon as any

descriptors are made available (host writes to the tail pointer). If several on-chip transmit descriptor queues need to fetch descriptors, descriptors from queues that are more starved are fetched. If a number of queues have a similar starvation level, highest indexed queue is served first and so forth, down to the lowest indexed queue.

Note: The starvation level of a queue corresponds to the number of descriptors above the prefetch threshold (*TXDCTL[n].PTHRESH)* that are already in the internal queue. The queue is more starved if there are less descriptors in the internal transmit descriptor cache. Comparing starvation level might be done roughly, not at the single descriptor level of resolution.

A queue is considered empty for the transmit descriptor fetch algorithm as long as:

- There is still no complete packet (single or large send) in its corresponding internal queue.
- There is no descriptor already in its way from system memory to the internal cache.
- The internal corresponding internal descriptor cache is not full.

Each time a descriptor fetch request is sent for an empty queue, the maximum available number of descriptor is requested, regardless of cache alignment issues.

When the on-chip buffer is nearly empty (below *TXDCTL[n].PTHRESH*), a prefetch is performed each time enough valid descriptors (*TXDCTL[n].HTHRESH*) are available in host memory and no other DMA activity of greater priority is pending (descriptor fetches and write-backs or packet data transfers).

When the number of descriptors in host memory is greater than the available on-chip descriptor storage, the I350 might elect to perform a fetch that is not a multiple of cache-line size. Hardware performs this non-aligned fetch if doing so results in the next descriptor fetch being aligned on a cache-line boundary. This enables the descriptor fetch mechanism to be more efficient in the cases where it has fallen behind software.

Note: The I350 NEVER fetches descriptors beyond the descriptor tail pointer.

7.2.2.6 Transmit Descriptor Write-Back

The descriptor write-back policy for transmit descriptors is similar to that of the receive descriptors when the *TXDCTL[n].WTHRESH* value is not 0x0. In this case, all descriptors are written back regardless of the value of their *RS* bit.

When the *TXDCTL[n].WTHRESH* value is 0x0, since transmit descriptor write-backs do not happen for every descriptor, only transmit descriptors that have the *RS* bit set are written back.

Any descriptor write-back includes the full 16 bytes of the descriptor.

Since the benefit of delaying and then bursting transmit descriptor write-backs is small at best, it is likely that the threshold is left at the default value (0x0) to force immediate write-back of transmit descriptors with their *RS* bit set and to preserve backward compatibility.

Descriptors are written back in one of three cases:

- *TXDCTL[n].WTHRESH* = 0x0 and a descriptor which has *RS* set is ready to be written back.
- The corresponding *EITR* counter has reached zero.
- *TXDCTL[n].WTHRESH* > 0x0 and *TXDCTL[n].WTHRESH* descriptors have accumulated.

For the first condition, write-backs are immediate. This is the default operation and is backward compatible with previous device implementations.

The other two conditions are only valid if descriptor bursting is enabled (Section 8.12.15). In the second condition, the *EITR* counter is used to force timely write-back of descriptors. The first packet after timer initialization starts the timer. Timer expiration flushes any accumulated descriptors and sets an interrupt event (*TXDW*).

For the last condition, if *TXDCTL[n].WTHRESH* descriptors are ready for write-back, the write-back is performed.

An additional mode in which transmit descriptors are not written back at all and the head pointer of the descriptor ring is written instead as described in Section 7.2.3.

Note: When transmit ring is smaller than internal cache size (24 descriptors) then at least one full packet should be placed in the ring and *TXDCTL[n].WTHRESH* value should be less than ring size. If *TXDCTL[n].WTHRESH* is 0x0 (transmit *RS* mode) then at least one descriptor should have the *RS* bit set inside the ring.

7.2.3 Transmit Completions Head Write Back

In legacy hardware, transmit requests are completed by writing the *DD* bit to the transmit descriptor ring. This causes cache thrash since both the software device driver and hardware are writing to the descriptor ring in host memory. Instead of writing the *DD* bits to signal that a transmit request completed, hardware can write the contents of the descriptor queue head to host memory. The software device driver reads that memory location to determine which transmit requests are complete. In order to improve the performance of this feature, the software device driver may program DCA registers to configure which CPU is processing each TX queue to allow pre-fetching of the head write back value from the right cache.

7.2.3.1 Description

The head counter is reflected in a memory location that is allocated by software, for each queue.

Head write back occurs if *TDWBAL[n].Head_WB_En* is set for this queue and the *RS* bit is set in the Tx descriptor, following corresponding data upload into packet buffer. If the head write-back feature is enabled, the I350 ignores *TXDCTL[n].WTHRESH* and takes in account only descriptors with the *RS* bit set (as if the *TXDCTL[n].WTHRESH* field was set to 0x0). In addition, the head write-back occurs upon *EITR* expiration for queues where the *WB_on_EITR* bit in *TDWBAL[n]* is set.

Software can also enable coalescing of the head write-back operations to reduce traffic on the PCIe bus, by programming the *TXDCTL.HWBTHRESH* field to a value greater than 0. In this case head write-back operation will occur only after the internal pending write-back count is greater than the *TXDCTL[n].HWBTHRESH* value.

The software device driver has control on this feature through Tx queue 0-7 head write-back address, low (*TDWBAL[n]*) and high *(TDWBAH[n]*) registers thus supporting 64-bit address access. See registers description in Section 8.12.16 and Section 8.12.17.

The 2 low register's LSB bits of the *TDWBAL[n]* register hold the control bits.

1. The *Head_WB_En* bit enables activation of the head write back feature. When *TDWBAL[n].Head_WB_En is set to 1* no TX descriptor write-back is executed for this queue.
2. *The WB_on_EITR* bit enables head write upon *EITR* expiration. When Head write back operation is enabled (*TDWBAL[n].Head_WB_En* = 1) setting the *TDWBAL[n].WB_on_EITR* bit to 1 enables placing an upper limit on delay of head write-back operation.

The 30 upper bits of the *TDWBAL[n]* register hold the lowest 32 bits of the head write-back address, assuming that the two last bits are zero. The *TDWBAH[n]* register holds the high part of the 64-bit address.

Note: Hardware writes a full Dword when writing this value, so software should reserve enough space for each head value.

If software enables Head Write-Back, it must also disable PCI Express Relaxed Ordering on the write-back transactions. This is done by disabling bit 11 in the *TXCTL* register for each active transmit queue. See Section 8.13.2.

The I350 might update the Head with values that are larger then the last Head pointer which holds a descriptor with RS bit set, but still the value will always point to a free descriptor (descriptor that is not owned by the I350 anymore).

Note: Software should program *TDWBAL[n]*, *TDWBAH[n]* registers when queue is disabled (*TXDCTL[n].Enable* = 0).

7.2.4 TCP/UDP Segmentation

Hardware TCP segmentation is one of the offloading options supported by the Windows* and Linux* TCP/IP stack. This is often referred to as TCP Segmentation Offloading or TSO. This feature enables the TCP/IP stack to pass to the network device driver a message to be transmitted that is bigger than the Maximum Transmission Unit (MTU) of medium. It is then the responsibility of the software device driver and hardware to divide the TCP message into MTU size frames that have appropriate layer 2 (Ethernet), 3 (IP), and 4 (TCP) headers. These headers must include sequence number, checksum fields, options and flag values as required. Note that some of these values (such as the checksum values) are unique for each packet of the TCP message and other fields such as the source IP address are constant for all packets associated with the TCP message.

The I350 supports also UDP segmentation for embedded applications, although this offload is not supported by the regular Windows* and Linux* stacks. Any reference in this section to TCP segmentation, should be considered as referring to both TCP and UDP segmentation.

Padding (TCTL.PSP) must be enabled in TCP segmentation mode, since the last frame might be shorter than 60 bytes, resulting in a bad frame if *PSP* is disabled.

The offloading of these mechanisms from the software device driver to the I350 saves significant CPU cycles. Note that the software device driver shares the additional tasks to support these options.

7.2.4.1 Assumptions

The following assumptions apply to the TCP segmentation implementation in the I350:

- The *RS* bit operation is not changed.
- Interrupts are set after data in buffers pointed to by individual descriptors is transferred (DMA'd) to hardware.

7.2.4.2 Transmission Process

The transmission process for regular (non-TCP segmentation packets) involves:

- The protocol stack receives from an application a block of data that is to be transmitted.

- The protocol stack calculates the number of packets required to transmit this block based on the MTU size of the media and required packet headers.

For each packet of the data block:

- Ethernet, IP and TCP/UDP headers are prepared by the stack.
- The stack interfaces with the software device driver and commands it to send the individual packet.
- The software device driver gets the frame and interfaces with the hardware.
- The hardware reads the packet from host memory (via DMA transfers).
- The software device driver returns ownership of the packet to the Network Operating System (NOS) when hardware has completed the DMA transfer of the frame (indicated by an interrupt).

The transmission process for the I350 TCP segmentation offload implementation involves:

- The protocol stack receives from an application a block of data that is to be transmitted.
- The stack interfaces to the software device driver and passes the block down with the appropriate header information.
- The software device driver sets up the interface to the hardware (via descriptors) for the TCP segmentation context.

Hardware DMA's (transfers) the packet data and performs the Ethernet packet segmentation and transmission based on offset and payload length parameters in the TCP/IP context descriptor including:

- Packet encapsulation
- Header generation and field updates including IPv4, IPV6, and TCP/UDP checksum generation
- The software device driver returns ownership of the block of data to the NOS when hardware has completed the DMA transfer of the entire data block (indicated by an interrupt).

7.2.4.2.1 TCP Segmentation Data Fetch Control

To perform TCP Segmentation in the I350, the DMA must be able to fit at least one packet of the segmented payload into available space in the on-chip Packet Buffer. The DMA does various comparisons between the remaining payload and the Packet Buffer available space, fetching additional payload and sending additional packets as space permits.

To support interleaving between descriptor queues at Ethernet frame resolution inside TSO requests, the frame header pointed to by the so called header descriptors are reread from system memory by hardware for every LSO segment. The I350 stores in an internal cache only the header's descriptors instead of the header's content.

To limit the internal cache size software should not spread the L3/L4 header (TCP, UDP, IPV4 or IPV6) on more than 4 descriptors. In the last header buffer it's allowed to mix header and data. This limitation stands for up to Layer4 header included, and for IPv4 or IPv6 indifferently.

7.2.4.2.2 TCP Segmentation Write-Back Modes

Since the TCP segmentation mode uses the buffers that contains the L3/L4 header multiple times, there are some limitations on the usage of different combinations of writeback and buffer release methods in order to guarantee the header buffer's availability until the entire packet is processed. These limitations are described in Table 7-39 below.

Table 7-39 Write Back Options For Large Send

WTHRESH	RS	HEAD Write Back Enable	Hardware Behavior	Software Expected Behavior for TSO packets.
0	Set in EOP descriptors only	Disable	Hardware writes back descriptors with RS bit set one at a time.	Software can retake ownership of all descriptors up to last descriptor with DD bit set.
0	Set in any descriptors	Disable	Hardware writes back descriptors with RS bit set one at a time.	Software can retake ownership of entire packets (EOP bit set) up to last descriptor with DD bit set.
0	Not set at all	Disable	Hardware does not write back any descriptor (since *RS* bit is not set)	Software should poll the TDH register. The TDH register reflects the last descriptor that software can take ownership of.[1]
0	Not set at all	Enable	Hardware writes back the head pointer only at EITR expire event reflecting the last descriptor that software can take ownership of.	Software may poll the TDH register or use the head value written back at EITR expire event. The TDH register reflects the last descriptor that software can take ownership of.
>0	Don't care	Disable	Hardware writes back all the descriptors in bursts and set all the *DD* bits.	Software can retake ownership of entire packets up to last descriptor with both DD and EOP bits set. Note: The TDH register reflects the last descriptor that software can take ownership of[1].
Don't care	Set in EOP descriptors only	Enable	Hardware writes back the Head pointer per each descriptor with *RS* bit set.[2]	Software can retake ownership of all descriptors up to the descriptor pointed by the head pointer read from system memory (by interrupt or polling).
Don't care	Set in any descriptors	Enable	Hardware writes back the Head pointer per each descriptor with *RS* bit set.	This mode is illegal since software won't access the descriptor, it cannot tell when the pointer passed the EOP descriptor.

1. Note that polling of the TDH register is a valid method only when the RS bit is never set, otherwise race conditions between software and hardware accesses to the descriptor ring can occur.
2. At EITR expire event, the Hardware writes back the head pointer reflecting the last descriptor that software can take ownership of.

7.2.4.3 TCP Segmentation Performance

Performance improvements for a hardware implementation of TCP Segmentation off-load include:

- The stack does not need to partition the block to fit the MTU size, saving CPU cycles.
- The stack only computes one Ethernet, IP, and TCP header per segment, saving CPU cycles.
- The Stack interfaces with the device driver only once per block transfer, instead of once per frame.
- Larger PCIe bursts are used which improves bus efficiency (such as lowering transaction overhead).
- Interrupts are easily reduced to one per TCP message instead of one per packet.
- Fewer I/O accesses are required to command the hardware.

7.2.4.4 Packet Format

Typical TCP/IP transmit window size is 8760 bytes (about 6 full size frames). Today the average size on corporate Intranets is 12-14KB, and normally the maximum window size allowed is 64KB (unless Windows Scaling - RFC 1323 is used). A TCP message can be as large as 256 KB and is generally fragmented across multiple pages in host memory. The I350 partitions the data packet into standard Ethernet frames prior to transmission according to the requested MSS. The I350 supports calculating the Ethernet, IP, TCP, and UDP headers, including checksum, on a frame-by-frame basis.

Table 7-40 TCP/IP or UDP/IP Packet Format Sent by Host

L2/L3/L4 Header			Data
Ethernet	IPv4/IPv6	TCP/UDP	DATA (full TCP message)

Table 7-41 TCP/IP or UDP/IP Packet Format Sent by the I350

L2/L3/L4 Header (updated)	Data (first MSS)	FCS	...	L2/L3/L4 Header (updated)	Data (Next MSS)	FCS	...

Frame formats supported by the I350 include:

- Ethernet 802.3
- IEEE 802.1Q VLAN (Ethernet 802.3ac)
- Ethernet Type 2
- Ethernet SNAP
- IPv4 headers with options
- IPv6 headers with extensions
- TCP with options
- UDP with options.

VLAN tag insertion might be handled by hardware

Note: UDP (unlike TCP) is not a “reliable protocol”, and fragmentation is not supported at the UDP level. UDP messages that are larger than the MTU size of the given network medium are normally fragmented at the IP layer. This is different from TCP, where large TCP messages can be fragmented at either the IP or TCP layers depending on the software implementation. The I350 has the ability to segment UDP traffic (in addition to TCP traffic), however, because UDP packets are generally fragmented at the IP layer, the I350's “TCP Segmentation” feature is not normally useful to handle UDP traffic.

7.2.4.5 TCP/UDP Segmentation Indication

Software indicates a TCP/UDP Segmentation transmission context to the hardware by setting up a TCP/IP Context Transmit Descriptor (see Section 7.2.2). The purpose of this descriptor is to provide information to the hardware to be used during the TCP segmentation off-load process.

Setting the *TSE* bit in the *TDESD.DCMD* field to 1b indicates that this descriptor refers to the TCP Segmentation context (as opposed to the normal checksum off loading context). This causes the checksum off loading, packet length, header length, and maximum segment size parameters to be loaded from the Context descriptor into the device.

The TCP Segmentation prototype header is taken from the packet data itself. Software must identity the type of packet that is being sent (IPv4/IPv6, TCP/UDP, other), calculate appropriate checksum off loading values for the desired checksum, and calculate the length of the header which is pre-appended. The header might be up to 240 bytes in length.

Once the TCP Segmentation context has been set, the next descriptor provides the initial data to transfer. This first descriptor(s) must point to a packet of the type indicated. Furthermore, the data it points to might need to be modified by software as it serves as the prototype header for all packets

within the TCP Segmentation context. The following sections describe the supported packet types and the various updates which are performed by hardware. This should be used as a guide to determine what must be modified in the original packet header to make it a suitable prototype header.

The following summarizes the fields considered by the driver for modification in constructing the prototype header.

IP Header

For IPv4 headers:

- *Identification* Field should be set as appropriate for first packet to be sent
- Header Checksum should be zeroed out unless some adjustment is needed by the driver

TCP Header

- Sequence Number should be set as appropriate for first packet of send (if not already)
- PSH, and FIN flags should be set as appropriate for LAST packet of send
- TCP Checksum should be set to the partial pseudo-header sum as follows (there is a more detailed discussion of this is Section 7.2.4.6):

Table 7-42 TCP Partial Pseudo-Header Sum for IPv4

IP Source Address		
IP Destination Address		
Zero	Layer 4 Protocol ID	Zero

Table 7-43 TCP Partial Pseudo-Header Sum for IPv6

IPv6 Source Address	
IPv6 Final Destination Address	
Zero	
Zero	Next Header

UDP Header

- Checksum should be set as in TCP header, above

The following sections describe the updating process performed by the hardware for each frame sent using the TCP Segmentation capability.

7.2.4.6 Transmit Checksum Offloading with TCP/UDP Segmentation

The I350 supports checksum off-loading as a component of the TCP Segmentation off-load feature and as a standalone capability. Section 7.2.5 describes the interface for controlling the checksum off-loading feature. This section describes the feature as it relates to TCP Segmentation.

The I350 supports IP and TCP header options in the checksum computation for packets that are derived from the TCP Segmentation feature.

Note: The I350 is capable of computing one level of IP header checksum and one TCP/UDP header and payload checksum. In case of multiple IP headers, the driver needs to compute all but

one IP header checksum. The I350 calculates check sums on the fly on a frame-by-frame basis and inserts the result in the IP/TCP/UDP headers of each frame. TCP and UDP checksum are a result of performing the checksum on all bytes of the payload and the pseudo header.

Two specific types of checksum are supported by the hardware in the context of the TCP Segmentation off-load feature:

- IPv4 checksum
- TCP checksum
- See Section 7.2.5 for description of checksum off loading of a single-send packet.

Each packet that is sent via the TCP segmentation off-load feature optionally includes the IPv4 checksum and either the TCP checksum.

All checksum calculations use a 16-bit wide one's complement checksum. The checksum word is calculated on the outgoing data.

Table 7-44 Supported Transmit Checksum Capabilities

Packet Type	Hardware IP Checksum Calculation	Hardware TCP/UDP Checksum Calculation
IP v4 packets	Yes	Yes
IP v6 packets (no IP checksum in IPv6)	NA	Yes
Packet is greater than 1518, 1522 or 1526 bytes; (LPE=1b)[1]	Yes	Yes
Packet has 802.3ac tag	Yes	Yes
Packet has IP options (IP header is longer than 20 bytes)	Yes	Yes
Packet has TCP or UDP options	Yes	Yes
IP header's protocol field contains a protocol # other than TCP or UDP.	Yes	No

1. Depends on number of VLAN tags.

The table below summarizes the conditions of when checksum off loading can/should be calculated.

Table 7-45 Conditions for Checksum Off Loading

Packet Type	IPv4	TCP/UDP	Reason
Non TSO	Yes	No	IP Raw packet (non TCP/UDP protocol)
	Yes	Yes	TCP segment or UDP datagram with checksum off-load
	No	No	Non-IP packet or checksum not offloaded
TSO	Yes	Yes	For TSO, checksum off-load must be done

7.2.4.7 TCP/UDP/IP Headers Update

IP/TCP or IP/UDP header is updated for each outgoing frame based on the IP/TCP header prototype which hardware DMA's from the first descriptor(s). The checksum fields and other header information are later updated on a frame-by-frame basis. The updating process is performed concurrently with the packet data fetch.

The following sections define what fields are modified by hardware during the TCP Segmentation process by the I350.

Note: Placing incorrect values in the Context descriptors may cause failure of Large Send. The indication of Large Send failure can be checked in the *TSCTC* statistics register.

7.2.4.7.1 TCP/UDP/IP Headers for the First Frames

The hardware makes the following changes to the headers of the first packet that is derived from each TCP segmentation context.

MAC Header (for SNAP)

- Type/Len field = *MSS* + *MACLEN* + *IPLEN* + *L4LEN* - 14 - 4 (if VLAN added by Software)

IPv4 Header

- IP Identification: Value in the IPv4 header of the prototype header in the packet data itself
- IP Total Length = *MSS* + *L4LEN* + *IPLEN*
- IP Checksum

IPv6 Header

- Payload Length = *MSS* + *L4LEN* + IPV6_HDR_extension[1]

TCP Header

- Sequence Number: The value is the Sequence Number of the first TCP byte in this frame.
- The flag values of the first frame are set by ANDing the flag word in the pseudo header with the *DTXTCPFLGL.TCP_flg_first_seg* register field. The default value of the *DTXTCPFLGL.TCP_flg_first_seg* are set so that the FIN flag and the PSH flag are cleared in the first frame.
- TCP Checksum

UDP Header

- UDP Length = MSS + L4LEN
- UDP Checksum

7.2.4.7.2 TCP/UDP/IP Headers for the Subsequent Frames

The hardware makes the following changes to the headers for subsequent packets that are derived as part of a TCP segmentation context:

Number of bytes left for transmission = *PAYLEN* - (N * MSS). Where N is the number of frames that have been transmitted.

MAC Header (for SNAP Packets)

Type/Len field = *MSS* + *MACLEN* + *IPLEN* + *L4LEN* - 14 - 4 (if VLAN added by Software)

IPv4 Header

- IP Identification: incremented from last value (wrap around based on 16 bit-width)
- IP Total Length = *MSS* + *L4LEN* + *IPLEN*

1. IPV6_HDR_extension is calculated as IPLEN - 40 bytes.

- IP Checksum

IPv6 Header

- Payload Length = *MSS* + *L4LEN* + IPV6_HDR_extension[1]

TCP Header

- Sequence Number update: Add previous TCP payload size to the previous sequence number value. This is equivalent to adding the *MSS* to the previous sequence number.
- The flag values of the subsequent frames are set by ANDing the flag word in the pseudo header with the *DTXTCPFLGL.TCP_Flg_mid_seg* register field. The default value of the *DTXTCPFLGL.TCP_Flg_mid_seg* are set so that if the FIN flag and the PSH flag are cleared in these frames.
- TCP Checksum

UDP Header

- UDP Length = MSS + L4LEN
- UDP Checksum

7.2.4.7.3 TCP/UDP/IP Headers for the Last Frame

The hardware makes the following changes to the headers for the last frame of a TCP segmentation context:

Last frame payload bytes = *PAYLEN* - (N * *MSS*)

MAC Header (for SNAP Packets)

- Type/Len field = Last frame payload bytes + *MACLEN* + *IPLEN* + *L4LEN* - 14 - 4 (if VLAN added by Software)

IPv4 Header

- IP Total Length = last frame payload bytes + *L4LEN* + *IPLEN*
- IP Identification: incremented from last value (wrap around based on 16 bit-width)
- IP Checksum

IPv6 Header

- Payload Length = last frame payload bytes + *L4LEN* + IPV6_HDR_extension[1]

TCP Header

- Sequence Number update: Add previous TCP payload size to the previous sequence number value. This is equivalent to adding the MSS to the previous sequence number.
- The flag values of the last frames are set by ANDing the flag word in the pseudo header with the DTXTCPFLGH.TCP_Flg_lst_seg register field. The default value of the *DTXTCPFLGH.TCP_Flg_lst_seg* are set so that if the FIN flag and the PSH flag are set in the last frame.
- TCP Checksum

UDP Header

- UDP Length = Last frame payload bytes + L4LEN
- UDP Checksum

1. IPV6_HDR_extension is calculated as IPLEN - 40 bytes.

7.2.4.8 Data Flow

The flow used by the I350 to do TCP segmentation is as follows:

1. Get a descriptor with a request for a TSO off-load of a TCP packet.
2. First Segment processing:
   a. Fetch all the buffers containing the header as calculated by the *MACLEN*, *IPLEN* and *L4LEN* fields. Save the addresses and lengths of the buffers containing the header (up to 4 buffers). The header content is not saved.
   b. Fetch data up to the MSS from subsequent buffers & calculate the adequate checksum(s).
   c. Update the Header accordingly and update internal state of the packet (next data to fetch and TCP SN).
   d. Send the packet to the network.
   e. If total packet was sent, go to step 4. else continue.
3. Next segments:
   a. Wait for next arbitration of this queue.
   b. Fetch all the buffers containing the header from the saved addresses. Subsequent reads of the header might be done with a no snoop attribute.
   c. Fetch data up to the MSS or end of packet from subsequent buffers & calculate the adequate checksum(s).
   d. Update the Header accordingly and update internal state of the packet (next data to fetch and TCP SN).
   e. If total packet was sent, request is done, else restart from step 3.
4. Release all buffers (update head pointer).

Note: Descriptors are fetched in a parallel process according to the consumption of the buffers.

7.2.5 Checksum Offloading in Non-Segmentation Mode

The previous section on TCP Segmentation off-load describes the IP/TCP/UDP checksum off loading mechanism used in conjunction with TCP Segmentation. The same underlying mechanism can also be applied as a standalone feature. The main difference in normal packet mode (non-TCP Segmentation) is that only the checksum fields in the IP/TCP/UDP headers need to be updated.

Before taking advantage of the I350's enhanced checksum off-load capability, a checksum context must be initialized. For the normal transmit checksum off-load feature this is performed by providing the device with a Descriptor with *TSE* = 0b in the *TDESD.DCMD* field and setting either the *TXSM* or *IXSM* bits in the *TDESD.POPTS* field. Setting *TSE* = 0b indicates that the normal checksum context is being set, as opposed to the segmentation context. For additional details on contexts, refer to Section 7.2.2.4.

Note: Enabling the checksum off loading capability without first initializing the appropriate checksum context leads to unpredictable results. CRC appending (*TDESC.CMD.IFCS*) must be enabled in TCP/IP checksum mode, since CRC must be inserted by hardware after the checksum has been calculated.

As mentioned in Section 7.2.2, it is not necessary to set a new context for each new packet. In many cases, the same checksum context can be used for a majority of the packet stream. In this case, some performance can be gained by only changing the context on an as needed basis or electing to use the off-load feature only for a particular traffic type, thereby avoiding the need to read all context descriptors except for the initial one.

Each checksum operates independently. Insertion of the IP and TCP checksum for each packet are enabled through the Transmit Data Descriptor *POPTS.TSXM* and *POPTS.IXSM* fields, respectively.

7.2.5.1 IP Checksum

Three fields in the Transmit Context Descriptor (*TDESC*) set the context of the IP checksum off loading feature:

- TUCMD.IPv4
- IPLEN
- MACLEN

TUCMD.IPv4 = 1b specifies that the packet type for this context is IPv4, and that the IP header checksum should be inserted. *TUCMD.IPv4* = 0b indicates that the packet type is IPv6 (or some other protocol) and that the IP header checksum should not be inserted.

MACLEN specifies the byte offset from the start of the DMA'd data to the first byte to be included in the checksum, the start of the IP header. The minimal allowed value for this field is 12. Note that the maximum value for this field is 127. This is adequate for typical applications.

Note: The *MACLEN* + *IPLEN* value needs to be less than the total DMA length for a packet. If this is not the case, the results are unpredictable.

IPLEN specifies the IP header length. Maximum allowed value for this field is 511 Bytes.

MACLEN + *IPLEN* specify where the IP checksum should stop. This is limited to the first 127 + 511 bytes of the packet and must be less than or equal to the total length of a given packet. If this is not the case, the result is unpredictable.

The 16-bit IPv4 Header Checksum is placed at the two bytes starting at *MACLEN* + 10.

As mentioned in Section 7.2.2.2, Transmit Contexts, it is not necessary to set a new context for each new packet. In many cases, the same checksum context can be used for a majority of the packet stream. In this case, some performance can be gained by only changing the context on an as needed basis or electing to use the off-load feature only for a particular traffic type, thereby avoiding all context descriptor reads except for the initial one.

7.2.5.2 TCP/UDP Checksum

Three fields in the Transmit Context Descriptor (*TDESC*) set the context of the TCP/UDP checksum off loading feature:

- MACLEN
- IPLEN
- TUCMD.L4T

TUCMD.L4T = 01b specifies that the packet type is TCP, and that the 16-bit TCP header checksum should be inserted at byte offset *MACLEN* + *IPLEN* +16. *TUCMD.L4T* = 00b indicates that the packet is UDP and that the 16-bit checksum should be inserted starting at byte offset *MACLEN* + *IPLEN* + 6.

IPLEN + *MACLEN* specifies the byte offset from the start of the DMA'd data to the first byte to be included in the checksum, the start of the TCP header. The minimal allowed value for this sum is 32/42 for UDP or TCP respectively.

Note: The *IPLEN* + *MACLEN* + *L4LEN* value needs to be less than the total DMA length for a packet. If this is not the case, the results are unpredictable.

The TCP/UDP checksum always continues to the last byte of the DMA data.

Note: For non-TSO, software still needs to calculate a full checksum for the TCP/UDP pseudo-header. This checksum of the pseudo-header should be placed in the packet data buffer at the appropriate offset for the checksum calculation.

7.2.5.3 SCTP CRC Offloading

For SCTP packets, a CRC32 checksum offload is provided.

Three fields in the Transmit Context Descriptor (*TDESC*) set the context of the STCP checksum off loading feature:

- MACLEN
- IPLEN
- TUCMD.L4T

TUCMD.L4T = 10b specifies that the packet type is SCTP, and that the 32-bit STCP CRC should be inserted at byte offset *MACLEN* + *IPLEN* + 8.

IPLEN + *MACLEN* specifies the byte offset from the start of the DMA'd data to the first byte to be included in the checksum, the start of the STCP header. The minimal allowed value for this sum is 26.

The SCTP CRC calculation always continues to the last byte of the DMA data.

The SCTP total L3 payload size (*TDESCD.PAYLEN* - *IPLEN* - *MACLEN*) should be a multiple of 4 bytes (SCTP padding not supported).

Notes:

1. TSO is not available for SCTP packets.
2. The CRC field of the SCTP header must be set to zero prior to requesting a CRC calculation offload.

7.2.5.4 Checksum Supported Per Packet Types

The following table summarizes which checksum is supported per packet type.

Note: TSO is not supported for packet types for which IP checksum & TCP checksum can not be calculated.

Table 7-46 Checksum Per Packet Type

Packet Type	Hardware IP Checksum Calculation	Hardware TCP/UDP/SCTP Checksum Calculation
IPv4 packets	Yes	Yes
IPv6 packets	No (n/a)	Yes
IPv6 packet with next header options: • Hop-by-Hop options • Destinations options • Routing (w len 0b) • Routing (w len >0b) • Fragment • Home option	 No (n/a) No (n/a) No (n/a) No (n/a) No (n/a) No (n/a)	 Yes Yes Yes No No No
IPv4 tunnels: • IPv4 packet in an IPv4 tunnel • IPv6 packet in an IPv4 tunnel	 Either IP or TCP/SCTP[1] Either IP or TCP/SCTP[1]	 Either IP or TCP/SCTP [1] Either IP or TCP/SCTP[1]
IPv6 tunnels: • IPv4 packet in an IPv6 tunnel • IPv6 packet in an IPv6 tunnel	 No No	 Yes Yes
Packet is an IPv4 fragment	Yes	No
Packet is greater than 1518, 1522 or 1526 bytes; (LPE=1b)[2]	Yes	Yes
Packet has 802.3ac tag	Yes	Yes
Packet has TCP or UDP options	Yes	Yes
IP header's protocol field contains protocol # other than TCP or UDP.	Yes	No

1. For the tunneled case, the driver might do only the TCP checksum or IPv4 checksum. If TCP checksum is desired, the driver should define the IP header length as the combined length of both IP headers in the packet. If an IPv4 checksum is required, the IP header length should be set to the IPv4 header length.
2. Depends on number of VLAN tags.

7.2.6 Multiple Transmit Queues

The number of transmit queues is 8, to match the expected number of CPU cores on server processors and to support virtualization mode.

If there are more CPUs cores than queues, then one queue might be used to service more than one CPU.

For transmission process, each thread might place a queue in the host memory of the CPU it is tied to.

The I350 supports assigning either high or low priority to each transmit queue. High priority is assigned to by setting the *TXDCTL[n].priority* bit to 1. When high priority is assigned to a specific transmit queue, the I350 will always prioritize transmit data fetch DMA accesses, before servicing transmit data fetch of lower priority transmit queues on a specific Physical Function.

Note: Throughput of low priority transmit queues can be significantly impacted if high priority queues utilize the DMA resources fully.

7.3 Interrupts

7.3.1 Interrupt Modes

The I350 supports the following interrupt modes:

- PCI legacy interrupts or MSI - selected when *GPIE.Multiple_MSIX* is 0b
- MSI-X in non-IOV mode - selected when *GPIE.Multiple_MSIX* is 1b and the *VFE* bit in PCIe SR-IOV control register is cleared.
- MSI-X in IOV mode - selected when *GPIE.Multiple_MSIX* is 1b and the *VFE* bit in PCIe SR-IOV control register is set.

7.3.1.1 MSI-X and Vectors

MSI-X defines a separate optional extension to basic MSI functionality. Compared to MSI, MSI-X supports a larger maximum number of vectors per function, the ability for software to control aliasing when fewer vectors are allocated than requested, plus the ability for each vector to use an independent address and data value, specified by a table that resides in Memory Space. However, most of the other characteristics of MSI-X are identical to those of MSI. For more information on MSI-X, refer to the PCI Local Bus Specification, Revision 3.0.

MSI-X maps each of the I350 interrupt causes into an interrupt vector that is conveyed by the I350 as a posted-write PCIe transaction. Mapping of an interrupt cause into an MSI-X vector is determined by system software (a device driver) through a translation table stored in the MSI-X Allocation registers. Each entry of the allocation registers defines the vector for a single interrupt cause.

7.3.1.1.1 Usage of Spare MSI-X Vectors by Physical Function

In an IOV mode, it is possible that not all the Virtual functions are activated. In this case, part of the MSI-X vectors are not used. The PF might then claim part of these for its own use in addition to the single MSI-X vector allocated to it. The PF function requests MSI-X vectors according to the following formula:

```
PF vector requested = MIN(PF vectors in EEPROM, 25 - 3*NumVFs).
```

7.3.2 Mapping of Interrupt Causes

There are 18 extended interrupt causes that exist in the I350:

1. 16 traffic causes — 8 Tx, 8 Rx.
2. TCP timer
3. Other causes — Summarizes legacy interrupts into one extended cause.

The way the I350 exposes causes to the software is determined by the interrupt mode as described below.

Mapping of interrupts causes is different in each of the interrupt modes and is described in the following sections of this chapter.

Note: If only one MSI-X vector is allocated by the operating system, then the driver might use the non MSI-X mapping method even in MSI-X mode.

7.3.2.1 Legacy and MSI Interrupt Modes

In legacy and MSI modes, an interrupt cause is reflected by setting a bit in the *EICR* register. This section describes the mapping of interrupt causes, like a specific Rx queue event or a Link Status Change event, to bits in the *EICR* register.

Mapping of queue-related causes is accomplished through the *IVAR* register. Each possible queue interrupt cause (each RX or TX queue) is allocated an entry in the *IVAR*, and each entry in the *IVAR* identifies one bit in the *EICR* register among the bits allocated to queue interrupt causes. It is possible to map multiple interrupt causes into the same *EICR* bit.

In this mode, different queue related interrupt causes can be mapped to the first 8 bits of the *EICR* register.

Interrupt causes related to non-queue causes are mapped into the *ICR* legacy register; each cause is allocated a separate bit. The sum of all causes is reflected in the *Other Cause* bit in *EICR*. Figure 7-12 below describes the allocation process.

The following configuration and parameters are involved:

- The *IVAR[3:0]* entries map 8 Tx queues and 8 Rx queues into the *EICR[7:0]* bits.
- The *IVAR_MISC* that maps non-queue causes is not used.
- The *EICR[30]* bit is allocated to the TCP timer interrupt cause.
- The *EICR[31]* bit is allocated to the other interrupt causes summarized in the ICR register.
- A single interrupt vector is provided.

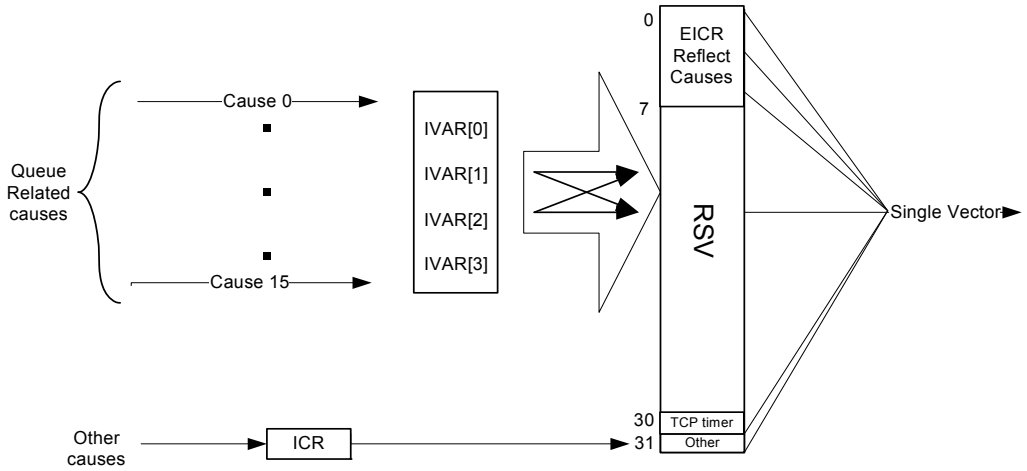


Figure 7-12 Cause Mapping in Legacy Mode

The Table below maps the different interrupt causes into the *IVAR* registers.

Table 7-47 Cause Allocation in the IVAR Registers - MSI and Legacy Mode

Interrupt	Entry	Description
Rx_i	i*2 (i= 0...7)	Receive queues i - Associates an interrupt occurring in the RX queue I with a corresponding bit in the EICR register.
Tx_i	i*2+1 (i= 0...7)	Transmit queues i- Associates an interrupt occurring in the TX queue I with a corresponding bit in the EICR register.

7.3.2.2 MSI-X Mode - Non-IOV Mode

In MSI-X mode, in a non Single Root - IOV setup (SR-IOV capability is not exposed in the PCIe config space), the I350 can request up to 25 Vectors.

In MSI-X mode, an interrupt cause is mapped into an MSI-X vector. This section describes the mapping of interrupt causes, like a specific RX queue event or a Link Status Change event, to MSI-X vectors.

Mapping is accomplished through the *IVAR* register. Each possible cause for an interrupt is allocated an entry in the *IVAR*, and each entry in the *IVAR* identifies one MSI-X vector. It is possible to map multiple interrupt causes into the same MSI-X vector.

The *EICR* also reflects interrupt vectors. The *EICR* bits allocated for queue causes reflect the MSI-X vector (bit 2 is set when MSI-X vector 2 is used). Interrupt causes related to non-queue causes are mapped into the *ICR* (as in the legacy case). The MSI-X vector for all such causes is reflected in the *EICR*.

The following configuration and parameters are involved:

- The *IVAR[3:0]* registers map 8 Tx queues and 8 Rx queues events to up to 23 interrupt vectors
- The *IVAR_MISC* register maps a TCP timer and other events to 2 MSI-X vectors

Figure 7-13 describes the allocation process.

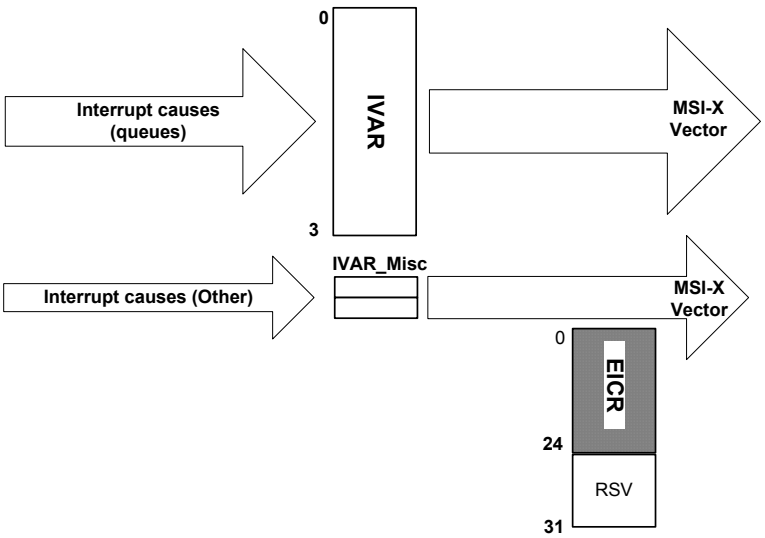


Figure 7-13 Cause Mapping in MSI-X Mode

Table 7-48 below defines which interrupt cause is represented by each entry in the MSI-X Allocation registers. In non IOV mode, the The software has access to 18 mapping entries to map each cause to one of the 25 MSI-x vectors.

Table 7-48 Cause Allocation in the IVAR Registers - Non-IOV Mode

Interrupt	Entry	Description
Rx_i	i*2 (i= 0...7)	Receive queues i - Associates an interrupt occurring in the RX queue I with a corresponding entry in the MSI-X Allocation registers.
Tx_i	i*2+1 (i= 0...7)	Transmit queues i- Associates an interrupt occurring in the TX queues I with a corresponding entry in the MSI-X Allocation registers.
TCP timer	16	TCP Timer - Associates an interrupt issued by the TCP timer with a corresponding entry in the MSI-X Allocation registers
Other cause	17	Other causes - Associates an interrupt issued by the "other causes" with a corresponding entry in the MSI-X Allocation registers

7.3.2.3 MSI-X Interrupts in IOV Mode

Each of the VF functions in PCI-SIG IOV mode is allocated 3 MSI-X vectors. The PF is allocated all the unused vectors and a minimum of 10 vectors.

Interrupt allocation for the physical function (PF) is done as in the MSI-X non-IOV case. However, the PF should not assign interrupt vectors to queues not assigned to it. The *IVAR_MISC* register allocates non-queue interrupts as in the non-IOV case with a single change - the entry assigned to "other" causes also handles interrupt on the mailbox.

Each of the VFs in IOV mode is allocated separate *IVAR* registers (called *VTIVAR*), translating its queue-related interrupt causes into MSI-X vectors for this virtual function. The *IVAR* register has one entry per Tx or Rx queue. A *VTIVAR_MISC* register is provided to map the mailbox interrupt into an MSI-X vector.

The PF can allocate interrupt causes not used by the VFs to one of its own vectors.

The *EICR* of each VF or of the PF reflects the status of the MSI-X vectors allocated to this function.

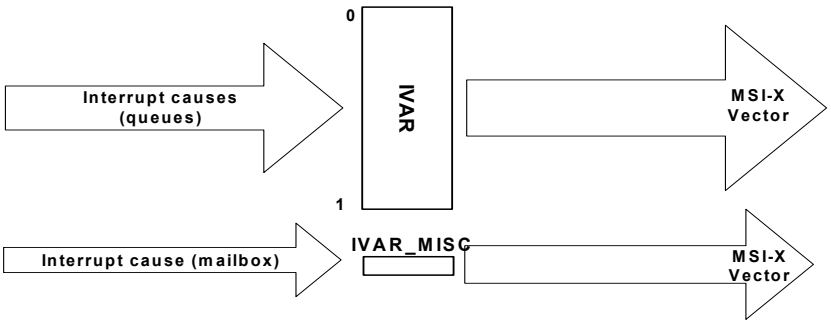


Figure 7-14 Cause Mapping of a VF in MSI-X Mode (IOV)

Table 7-49 below, defines for a given VM (not PF) which interrupt cause is represented by each entry in the MSI-X Allocation registers.

In the IOV mode Software has access to 3 mapping entries to map each cause to one of the 3 MSI-x vectors.

The 3 VM vectors (per each VM) can be allocated to one or more causes (1 queue RX traffic interrupt, 1 queue TX traffic interrupt and Mail Box interrupt).

Please refer to Section 7.8.2.9.1 for details of the Mail Box mechanism.

Table 7-49 Cause Allocation for a VF in the *VTIVAR* Registers - IOV Mode

Interrupt	Entry	Description
Rx Queue	0	Associates an interrupt occurring in the Rx queue with a corresponding entry in the MSI-X Allocation registers.
Tx Queue	1	Associates an interrupt occurring in the Tx queue with a corresponding entry in the MSI-X Allocation registers.

7.3.3 Legacy Interrupt Registers

The interrupt logic consists of the registers listed in the tables below, plus the registers associated with MSI/MSI-X signaling. The first table describes the use of the registers in legacy mode and the second one the use of the registers when using the extended interrupts functionality

Table 7-50 Interrupt Registers - Legacy Mode

Register	Acronym	Function
Interrupt Cause	ICR	Records interrupt conditions.
Interrupt Cause Set	ICS	Allows software to set bits in the ICR.
Interrupt Mask Set/Read	IMS	Sets or reads bits in the interrupt mask.
Interrupt Mask Clear	IMC	Clears bits in the interrupt mask.
Interrupt Acknowledge auto-mask	IAM	Under some conditions, the content of this register is copied to the mask register following read or write of ICR.

Table 7-51 Interrupt Registers - Extended Mode

Register	Acronym	Function
Extended Interrupt Cause	EICR	Records interrupt causes from receive and transmit queues. An interrupt is signaled when unmasked bits in this register are set.
Extended Interrupt Cause Set	EICS	Allows software to set bits in the Interrupt Cause register.
Extended Interrupt Mask Set/ Read	EIMS	Sets or read bits in the interrupt mask.
Extended Interrupt Mask Clear	EIMC	Clears bits in the interrupt mask.
Extended Interrupt Auto Clear	EIAC	Allows bits in the EICR to be cleared automatically following an MSI-X interrupt without a read or write of the EICR.
Extended Interrupt Acknowledge auto-mask	EIAM	This register is used to decide which masks are cleared in the extended mask register following read or write of EICR or which masks are set following a write to EICS. In MSI-X mode, this register also controls which bits in EIMC are cleared automatically following an MSI-X interrupt.
Interrupt Cause	ICR	Records interrupt conditions for special conditions - a single interrupt from all the conditions of ICR is reflected in the “other” field of the EICR.
Interrupt Cause Set	ICS	Allows software to set bits in the ICR.
Interrupt Mask Set/Read	IMS	Sets or reads bits in the other interrupt mask.
Interrupt Mask Clear	IMC	Clears bits in the Other interrupt mask.
Interrupt Acknowledge auto-mask	IAM	Under some conditions, the content of this register is copied to the mask register following read or write of ICR.
General Purpose Interrupt Enable	GPIE	Controls different behaviors of the interrupt mechanism.

7.3.3.1 Interrupt Cause Register (ICR)

7.3.3.1.1 Legacy Mode

In Legacy mode, *ICR* is used as the sole interrupt cause register. Upon reception of an interrupt, the interrupt handling routine can read this register in order to find out what are the causes of this interrupt.

7.3.3.1.2 Advanced Mode

In advanced mode, this register captures the interrupt causes not directly captured by the *EICR*. These are infrequent management interrupts and error conditions.

Note that when *EICR* is used in advanced mode, the RX /TX related bits in *ICR* should be masked.

ICR bits are cleared on register read. If *GPIE.NSICR* = 0b, then the clear on read occurs only if no bit is set in the *IMS* register or at least one bit is set in the *IMS* register and there is a true interrupt as reflected in the *ICR.INTA* bit.

7.3.3.2 Interrupt Cause Set Register (ICS)

This register allows software to set bits in the *ICR* register. Writing a 1b in an *ICS* bit causes the corresponding bit in the *ICS* register to be set. Used usually to re-arm interrupts the software device driver didn't have time to handle in the current interrupt routine.

7.3.3.3 Interrupt Mask Set/Read Register (IMS)

An interrupt is enabled if its corresponding mask bit in this register is set to 1b, and disabled if its corresponding mask bit is set to 0b. A PCIe interrupt is generated whenever one of the bits in this register is set, and the corresponding interrupt condition occurs. The occurrence of an interrupt condition is reflected by having a bit set in the *Interrupt Cause Register* (*ICR*).

Reading this register returns which bits have an interrupt mask set.

A particular interrupt might be enabled by writing a 1b to the corresponding mask bit in this register. Any bits written with a 0b are unchanged. Thus, if software desires to disable a particular interrupt condition that had been previously enabled, it must write to the Interrupt Mask Clear (*IMC*) Register, rather than writing a 0b to a bit in this register.

7.3.3.4 Interrupt Mask Clear Register (IMC)

Software blocks interrupts by clearing the corresponding mask bit. This is accomplished by writing a 1b to the corresponding bit in this register. Bits written with 0b are unchanged (their mask status does not change).

7.3.3.5 Interrupt Acknowledge Auto-mask register (IAM)

An *ICR* read or write has the side effect of writing the contents of this register to the *IMC* register to auto-mask additional interrupts from the *ICR* bits in the locations where the *IAM* bits are set. If *GPIE.NSICR* = 0b, then the copy of this register to the *IMC* register occurs only if at least one bit is set in the *IMS* register and there is a true interrupt as reflected in the *ICR.INT*A bit.

7.3.3.6 Extended Interrupt Cause Registers (EICR)

7.3.3.6.1 MSI/INT-A Mode (GPIE.Multiple_MSIX = 0)

This register records the interrupts causes, to provide Software with information on the interrupt source.

The interrupt causes include:

1. The Receive and Transmit queues — Each queue (either Tx or Rx) can be mapped to one of the 8 interrupt causes bits (RxTxQ) available in this register according to the mapping in the *IVAR* registers
2. Indication for the TCP timer interrupt.
3. Legacy and other indications — When any interrupt in the Interrupt Cause register is active.

Writing a 1b clears the corresponding bit in this register. Reading this register auto-clears all bits.

7.3.3.6.2 MSI-X Mode (GPIE.Multiple_MSIX = 1)

This register records the interrupt vectors currently emitted. In this mode only the first 25 bits are valid.

For all the subsequent registers, in MSI-X mode, each bit controls the behavior of one vector.

Bits in this register can be configured to auto-clear when the MSI-X interrupt message is sent, in order to minimize driver overhead when using MSI-X interrupt signaling.

Writing a 1b clears the corresponding bit in this register. Reading this register does not clear any bits.

7.3.3.7 Extended Interrupt Cause Set Register (EICS)

This register enables the software device driver to set *EICR* bits. Writing a 1b in a *EICS* bit causes the corresponding bit in the *EICR* register to be set. Used usually to re-arm interrupts that the software didn't have time to handle in the current interrupt routine.

7.3.3.8 Extended Interrupt Mask Set and Read Register (EIMS) & Extended Interrupt Mask Clear Register (EIMC)

Interrupts appear on PCIe only if the interrupt cause bit is a one and the corresponding interrupt mask bit is a one. Software blocks assertion of an interrupt by clearing the corresponding bit in the mask register. The cause bit stores the interrupt event regardless of the state of the mask bit. Different Clear (*EIMC*) and set (*EIMS*) registers make this register more “thread safe” by avoiding a read-modify-write

operation on the mask register. The mask bit is set for each bit written as a one in the set register (*EIMS*) and cleared for each bit written as a one in the clear register (*EIMC*). Reading the set register (*EIMS*) returns the current mask register value.

7.3.3.9 Extended Interrupt Auto Clear Enable Register (EIAC)

Each bit in this register enables clearing of the corresponding bit in *EICR* following interrupt generation. When a bit is set, the corresponding bit in the *EICR* register is automatically cleared following an interrupt. This feature should only be used in MSI-X mode.

When used in conjunction with MSI-X interrupt vector, this feature allows interrupt cause recognition, and selective interrupt cause, without requiring software to read or write the *EICR* register; therefore, the penalty related to a PCIe read or write transaction is avoided.

See section 7.3.4 for additional information on the interrupt cause reset process.

7.3.3.10 Extended Interrupt Auto Mask Enable Register (EIAM)

Each bit set in this register enables clearing of the corresponding bit in the extended mask register following read or write-to-clear to *EICR*. It also enables setting of the corresponding bit in the extended mask register following a write-to-set to *EICS*.

This mode is provided in case MSI-X is not used, and therefore auto-clear through *EIAC* register is not available.

In MSI-X mode, the driver software might set the bits of this register to select mask bits that must be reset during interrupt processing. In this mode, each bit in this register enables clearing of the corresponding bit in *EIMC* following interrupt generation.

7.3.3.11 GPIE Register

There are a few bits in the *GPIE* register that define the behavior of the interrupt mechanism. The setting of these bits is different in each mode of operation. The following table describes the recommended setting of these bits in the different modes:

Table 7-52 Settings for Different Interrupt Modes

Field	Bit(s)	Initial Value	Description	INT-x/ MSI + Legacy	INT-x/ MSI + Extend	MSI-X Multi vector	MSI-X Single vector
NSICR	0	0b	**Non Selective Interrupt clear on read**: When set, every read of the *ICR* register clears the *ICR* register. When this bit is cleared, an *ICR* register read causes the *ICR* register to be cleared only if an actual interrupt was asserted or *IMS* = 0x0.	0b[1]	1b	1b	1b
Multiple_ MSIX	4	0b	**Multiple_MSIX - multiple vectors:** 0b = non-MSIX or MSI-X with 1 vector *IVAR* maps Rx/Tx causes to 8 *EICR* bits, but MSIX[0] is asserted for all. 1b = MSIX mode, IVAR maps Rx/Tx causes to 25 *EICR* bits. When set, the *EICR* register is not clear on read.	0b	0b	1b	0b
EIAME	30	0b	**EIAME:** When set, upon firing of an MSI-X message, mask bits set in EIAM associated with this message are cleared. Otherwise, EIAM is used only upon read or write of EICR/EICS registers.	0b	0b	1b	1b
PBA_ support	31	0b	**PBA support:** When set, setting one of the extended interrupts masks via EIMS causes the PBA bit of the associated MSI-X vector to be cleared. Otherwise, the I350 behaves in a way that supports legacy INT-x interrupts. Should be cleared when working in INT-x or MSI mode and set in MSI-X mode.	0b	0b	1b	1b

1. In systems where interrupt sharing is not expected, the *NSICR* bit can be set by legacy drivers also.

As this register affects the way the hardware interprets write operations to other interrupt control registers, it should be set to the correct mode before accessing other interrupt control registers.

7.3.4 Clearing Interrupt Causes

The I350 has three methods available to clear *EICR* bits: Autoclear, clear-on-write, and clear-on-read. *ICR* bits might only be cleared with clear-on-write or clear-on-read.

7.3.4.1 Auto-Clear

In systems that support MSI-X, the interrupt vector allows the interrupt service routine to know the interrupt cause without reading the *EICR*. With interrupt moderation active, software load from spurious interrupts is minimized. In this case, the software overhead of a I/O read or write can be avoided by setting appropriate *EICR* bits to autoclear mode by setting the corresponding bits in the Extended Interrupt Auto-clear Enable Register (*EIAC*).

When auto-clear is enabled for an interrupt cause, the *EICR* bit is set when a cause event mapped to this vector occurs. When the *EITR* Counter reaches zero, the MSI-X message is sent on PCIe. Then the *EICR* bit is cleared and enabled to be set by a new cause event. The vector in the MSI-X message signals software the cause of the interrupt to be serviced.

It is possible that in the time after the *EICR* bit is cleared and the interrupt service routine services the cause, for example checking the transmit and receive queues, that another cause event occurs that is then serviced by this ISR call, yet the *EICR* bit remains set. This results in a "spurious interrupt". Software can detect this case, for example if there are no entries that require service in the transmit and receive queues, and exit knowing that the interrupt has been automatically cleared. The use of interrupt moderations through the *EITR* register limits the extra software overhead that can be caused by these spurious interrupts.

7.3.4.2 Write to Clear

In the case where the driver wishes to configure itself in MSI-X mode to not use the "auto-clear" feature, it might clear the *EICR* bits by writing to the *EICR* register. Any bits written with a 1b is cleared. Any bits written with a 0b remain unchanged.

7.3.4.3 Read to Clear

The *EICR* and *ICR* registers are cleared on a read.

Note: The driver should never do a read-to-clear of the *EICR* when in MSI-X mode, since this might clear interrupt cause events which are processed by a different interrupt handler (assuming multiple vectors).

7.3.5 Interrupt Moderation

An interrupt is generated upon receiving of incoming packets, as throttled by the *EITR* registers (see Section 8.8.14). There is an *EITR* register per MSI-X vector.

In MSI-X mode, each active bit in *EICR* can trigger the interrupt vector it is allocated to. Following the allocation, the *EITR* corresponding to the MSI-X vector is tied to one or more bits in *EICR*.

When multi vector MSI-X is not activated, the interrupt moderation is controlled by register *EITR[0]*.

Software can use *EITR* to limit the rate of delivery of interrupts to the host CPU. This register provides a guaranteed inter-interrupt delay between interrupts asserted by the network controller, regardless of network traffic conditions.

The following formula converts the inter-interrupt interval value to the common 'interrupts/sec.' performance metric:

$$\text{interrupts/sec} = (1 * 10^{-6}\text{sec} \times \text{interval})^{-1}$$

Note: In the I350 the interval granularity is 1 μsec so some of the LSB bits of the interval are used for the low latency interrupt moderation.

For example, if the interval is programmed to 125d, the network controller guarantees the CPU is not interrupted by the network controller for at least 125 μs from the last interrupt. In this case, the maximum observable interrupt rate from the adapter should not exceed 8000 interrupts/sec.

Inversely, inter-interrupt interval value can be calculated as:

$$\text{inter-interrupt interval} = (1 * 10^{-6}\ \text{sec} \times \text{interrupt/sec})^{-1}$$

The optimal performance setting for this register is system and configuration specific.

The Extended Interrupt Throttle Register should default to zero upon initialization and reset. It loads in the value programmed by the software after software initializes the device.

When software wants to force an immediate interrupt, for example after setting a bit in the *EICR* with the Extended Interrupt Cause Set register, a value of 0 can be written to the Counter to generate an interrupt immediately. This write should include re-writing the *Interval* field with the desired constant, as it is used to reload the Counter immediately for the next throttling interval.

The I350 implements interrupt moderation to reduce the number of interrupts software processes. The moderation scheme is based on the *EITR* (Interrupt Throttle Register). Whenever an interrupt event happens, the corresponding bit in the *EICR* is activated. However, an interrupt message is not sent out on the PCIe interface until the *EITR* counter assigned to that *EICR* bit has counted down to zero. As soon as the interrupt is issued, the *EITR* counter is reloaded with its initial value and the process repeats again. The interrupt flow should follow the diagram below:

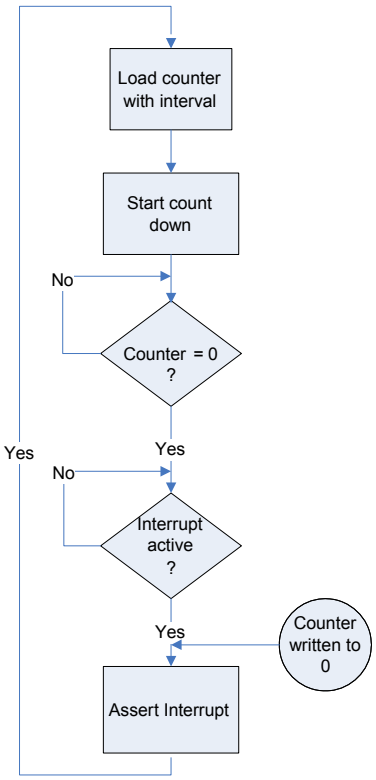


Figure 7-15 Interrupt Throttle Flow Diagram

EITR is designed to guarantee the total number of interrupts per second so for cases where the I350 is connected to a network with low traffic load, if the *EITR* counter counted down to zero and no interrupt event has happened, then the *EITR* counter is not re-armed but stays at zero. Thus, the next interrupt event triggers an interrupt immediately. That scenario is illustrated as "Case B" below.

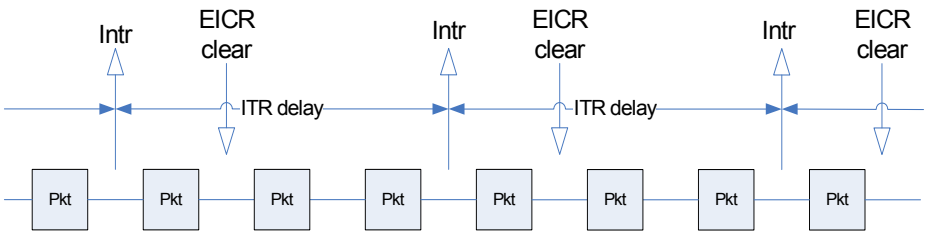


Figure 7-16 Case A: Heavy Load, Interrupts Moderated

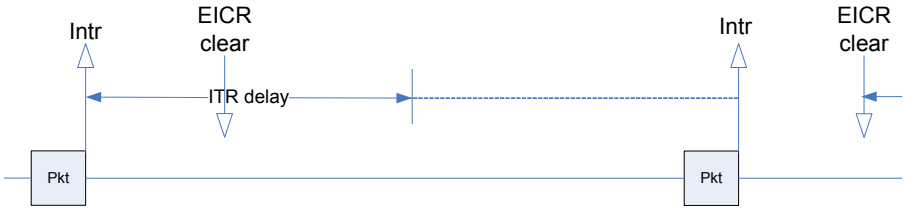


Figure 7-17 Light load, Interrupts Immediately on Packet Receive

7.3.6 Rate Controlled Low Latency Interrupts (LLI)

There are some types of network traffic for which latency is a critical issue. For these types of traffic, interrupt moderation hurts performance by increasing latency between the time a packet is received by hardware and the time it is handled to the host operating system. This traffic can be identified by the 2-tuple value, in conjunction with Control Bits and specific size. In addition packets with specific Ethernet types, TCP flag or specific VLAN priority might generate an immediate interrupt.

Low latency interrupts shares the filters used by the queueing mechanism described in Section 7.1.1. Each of these filters, in addition to the queueing action might also indicate matching packets might generate immediate interrupt.

If a received packet matches one of these filters, hardware should interrupt immediately, overriding the interrupt moderation by the *EITR* counter.

Each time a Low Latency Interrupt is fired, the *EITR* interval is loaded and down-counting starts again.

The logic of the low latency interrupt mechanism is as follows:

- There are 8 2-tuple filters. The content of each filter is described in Section 7.1.2.5. The immediate interrupt action of each filter can be enabled or disabled. If one of the filters detects an adequate packet, an immediate interrupt is issued.

- There are 8 flex filters. The content of each filter is described in Section 7.1.2.6. The immediate interrupt action of each filter can be enabled or disabled. If one of the filters detects an adequate packet, an immediate interrupt is issued.
- When VLAN priority filtering is enabled, VLAN packets must trigger an immediate interrupt when the VLAN Priority is equal to or above the VLAN priority threshold. This is regardless of the status of the 2-tuple or Flex filters.
- The SYN packets filter defined in Section 7.1.2.7 and the ethernet type filters defined in section Section 7.1.2.4 might also be used to indicate low latency interrupt conditions.

Note: Immediate interrupts are available only when using advanced receive descriptors and not for legacy descriptors.

Packets that are dropped or have errors do not cause a Low Latency Interrupt.

7.3.6.1 Rate Control Mechanism

In a network with lots of latency sensitive traffics the Low Latency Interrupt can eliminate the Interrupt throttling capability by flooding the Host with too many interrupts (more than the Host can handle).

In order to mitigate the above, the I350 supports a credit base mechanism to control the rate of the Low Latency Interrupts.

Rules:

- The default value of each counter is 0b (no moderation). This also preserves backward compatibility.
- The counter increments at a configurable rate, and saturates at the maximum value (31d).
  - The configurable rate granularity is 4 μs (250K interrupt/sec. down to 250K/32 ~ 8K interrupts per sec.).
- A LLI might be issued as long as the counter value is strictly positive (> zero).
  - The credit counter allows bursts of low latency interrupts but the interrupt average are not more than the configured rate.
- Each time a Low Latency Interrupt is fired the credit counter decrements by one.
- Once the counter reaches zero, a low latency interrupt cannot be fired
  - Must wait for the next ITR expired or for the next incrementing of this counter (if the EITR expired happened first the counter does not decrement).

The *EITR* and *GPIE* registers manage rate control of *LLI*:

- The *LL Interval* field in the GPIE register controls the rate of credits
- The 5-bit *LL Counter* field in the *EITR* register contains the credits

7.3.7 TCP Timer Interrupt

7.3.7.1 Introduction

The TCP Timer interrupt provides an accurate and efficient way for a periodic timer to be implemented using hardware. The driver would program a timeout value (usual value of 10 ms), and each time the timer expires, hardware sets a specific bit in the *EICR*. When an interrupt occurs (due to normal interrupt moderation schemes), software reads the *EICR* and discovers that it needs to process timer events during that DPC.

The timeout should be programmable by the driver, and the driver should be able to disable the timer interrupt if it is not needed.

7.3.7.2 Description

A stand-alone down-counter is implemented. An interrupt is issued each time the value of the counter is zero.

The software is responsible for setting initial value for the timer in the *TCPTIMER.Duration* field. Kick-starting is done by writing a 1b to the *TCPTIMER.KickStart* bit.

Following the kick-start, an internal counter is set to the value defined by the *TCPTIMER.Duration* field. Then during the count operation, the counter is decreased by one each millisecond. When the counter reaches zero, an interrupt is issued (see EICR register Section 8.8.3). The counter re-starts counting from its initial value if the *TCPTIMER.Loop* field is set.

7.3.8 Setting of Interrupt Registers

In each mode, the registers controlling the interrupts should be set in a different way to assure the right behavior.

Table 7-53 Registers Settings for Different Interrupt Modes

Field	Description	INT-x/MSI + Legacy	INT-x/ MSI + Extend	MSI-X Multi vector	MSI-X Single vector
IMS	Legacy Masks	Set[1]	Set[2]	Set[2]	Set[2]
IAM	Legacy Auto Mask Register	May be Set	0x0	0x0	0x0
EIMS	Extended Masks	Set Other Cause only.	Set[1]	Set[1]	Set[1]
EIAC	Extended Auto Clear register	0x0	0x0	At least one [3]	0x0
EIAM	Extended Auto Mask Register	0x0	Set[1]		Set[1]
EITR[0]	Interrupt Moderation register	May be Enabled	May be Enabled	Enable[4]	Enable
EITR[1...n]	Extended Interrupt Moderation register	Disable	Disable	Enable[4]	Disable
GPIE	Interrupts configuration	See Table 7-52 for details			

1. According to the requested causes
2. Only non traffic causes.
3. EIAC or EIAM or both should be set for each cause.
4. EITR must be enabled if Auto Mask is disabled. If Auto Mask is enabled, moderation may be disabled for the specific vector.

7.4 802.1q VLAN Support

The I350 provides several specific mechanisms to support 802.1q VLANs:

- Optional adding (for transmits) and stripping (for receives) of IEEE 802.1q VLAN tags.
- Optional ability to filter packets belonging to certain 802.1q VLANs.
- Double VLAN Support.

7.4.1 802.1q VLAN Packet Format

The following diagram compares an untagged 802.3 Ethernet packet with an 802.1q VLAN tagged packet:

Table 7-54 Comparing Packets

802.3 Packet	#Octets		802.1q VLAN Packet	#Octets
DA	6		DA	6
SA	6		SA	6
Type/Length	2		802.1q Tag	4
Data	46-1500		Type/Length	2
CRC	4		Data	46-1500
			CRC*	4

Note: The CRC for the 802.1q tagged frame is re-computed, so that it covers the entire tagged frame including the 802.1q tag header. Also, max frame size for an 802.1q VLAN packet is 1522 octets as opposed to 1518 octets for a normal 802.3z Ethernet packet.

7.4.2 802.1q Tagged Frames

For 802.1q, the *Tag Header* field consists of four octets comprised of the Tag Protocol Identifier (TPID) and Tag Control Information (TCI); each taking 2 octets. The first 16 bits of the tag header makes up the TPID. It contains the “protocol type” which identifies the packet as a valid 802.1q tagged packet.

The two octets making up the TCI contain three fields:

- User Priority (UP)
- Canonical Form Indicator (CFI). Should be 0b for transmits. For receives, the device has the capability to filter out packets that have this bit set. See the *CFIEN* and *CFI* bits in the *RCTL* described in Section 8.10.1.
- VLAN Identifier (VID)

The bit ordering is shown below:

Table 7-55 TCI Bit Ordering

Octet 1								Octet 2							
UP			CFI	VID											

7.4.3 Transmitting and Receiving 802.1q Packets

7.4.3.1 Adding 802.1q Tags on Transmits

Software might command the I350 to insert an 802.1q VLAN tag on a per packet or per flow basis. If the *VLE* bit in the transmit descriptor is set to 1b, then the I350 inserts a VLAN tag into the packet that it transmits over the wire. The *Tag Protocol Identifier (TPID)* field of the 802.1q tag comes from the VET register. 8021.Q tag insertion is done in different ways for legacy and advanced Tx descriptors:

- Legacy Transmit Descriptors:, The Tag Control Information (TCI) of the 802.1q tag comes from the *VLAN* field (see Figure 7-9) of the descriptor. Refer to Table 7-27, for more information regarding hardware insertion of tags for transmits.
- Advanced Transmit Descriptor: The Tag Control Information (TCI) of the 802.1q tag comes from the VLAN *Tag* field (see Table 7.2.2.2.1) of the advanced context descriptor. The *IDX* field of the advanced Tx descriptor should be set to the adequate context.

7.4.3.2 Stripping 802.1q Tags on Receives

Software might instruct the I350 to strip 802.1q VLAN tags from received packets. If VLAN stripping is enabled and the incoming packet is an 802.1q VLAN packet (its *Ethernet Type* field matched the VET), then the I350 strips the 4 byte VLAN tag from the packet, and stores the TCI in the VLAN *Tag* field (see Figure 7-5 and See "Packet Checksum) of the receive descriptor.

The I350 also sets the *VP* bit in the receive descriptor to indicate that the packet had a VLAN tag that was stripped. If the *CTRL.VME* bit is not set, the 802.1Q packets can still be received if they pass the receive filter, but the VLAN tag is not stripped and the *VP* bit is not set. Refer Figure 7-18 for more information regarding receive packet filtering.

VLAN stripping can be enabled using two different modes:

1. By setting the *DVMOLR.STRVLAN* for the relevant queue.

By setting the *CTRL.VME* bit.

7.4.4 802.1q VLAN Packet Filtering

VLAN filtering is enabled by setting the *RCTL.VFE* bit to 1b. If enabled, hardware compares the type field of the incoming packet to a 16-bit field in the VLAN Ether Type (VET) register. If the VLAN type field in the incoming packet matches the VET register, the packet is then compared against the VLAN Filter Table Array (VFTA[127:0]) for acceptance.

The I350 provides exact VLAN filtering for VLAN tags for host traffic and VLAN tags for manageability traffic.

Host VLAN filtering:

The *Virtual LAN ID* field indexes a 4096 bit vector. If the indexed bit in the vector is one; there is a Virtual LAN match. Software might set the entire bit vector to ones if the node does not implement 802.1q filtering. The register description of the VLAN Filter Table Array is described in detail in Section 8.10.18.

In summary, the 4096-bit vector is comprised of 128, 32-bit registers. The *VLAN Identifier (VID)* field consists of 12 bits. The upper 7 bits of this field are decoded to determine the 32-bit register in the VLAN Filter Table Array to address and the lower 5 bits determine which of the 32 bits in the register to evaluate for matching.

Manageability VLAN filtering:

The BMC configures the I350 with eight different manageability VIDs via the Management VLAN TAG Value [7:0] - MAVTV[7:0] registers and enables each filter in the MDEF register.

Two other bits in the Receive Control register (see Section 8.10.1), CFIEN and CFI, are also used in conjunction with 802.1q VLAN filtering operations. CFIEN enables the comparison of the value of the *CFI* bit in the 802.1q packet to the Receive Control register *CFI* bit as acceptance criteria for the packet.

Note: The *VFE* bit does not affect whether the VLAN tag is stripped. It only affects whether the VLAN packet passes the receive filter.

The following table lists reception actions per control bit settings.

Figure 7-18 Packet Reception Decision Table

Is packet 802.1q?	CTRL. VME	RCTL. VFE	Action
No	X[1]	X[1]	Normal packet reception
Yes	0b	0b	Receive a VLAN packet if it passes the standard MAC address filters (only). Leave the packet as received in the data buffer. *VP* bit in receive descriptor is cleared.
Yes	0b	1b	Receive a VLAN packet if it passes the standard filters and the VLAN filter table. Leave the packet as received in the data buffer (the VLAN tag would not be stripped). *VP* bit in receive descriptor is cleared.
Yes	1b	0b	Receive a VLAN packet if it passes the standard filters (only). Strip off the VLAN information (four bytes) from the incoming packet and store in the descriptor. Sets *VP* bit in receive descriptor.
Yes	1b	1b	Receive a VLAN packet if it passes the standard filters and the VLAN filter table. Strip off the VLAN information (four bytes) from the incoming packet and store in the descriptor. Sets *VP* bit in receive descriptor.

1. X - Don't care

Note: A packet is defined as a VLAN/802.1q packet if its type field matches the VET.

7.4.5 Double VLAN Support

The I350 supports a mode where most of the received and sent packet have at least one VLAN tag in addition to the regular tagging which might optionally be added. This mode is used for systems where the switches add an additional tag containing switching information.

Note: The only packets that may not have the additional VLAN are local packets that will not have any VLAN tag.

This mode is activated by setting *CTRL_EXT.EXT_VLAN* bit. The default value of this bit is set according to the EXT_VLAN (bit 1) in the *Initialization Control 3* EEPROM word for ports 0 to 3. See Section 6.2.26 for more information.

The type of the VLAN tag used for the additional VLAN is defined in the *VET.VET_EXT* field.

7.4.5.1 Transmit Behavior With External VLAN

It is expected that the driver include the external VLAN header as part of the transmit data structure. The software may post the internal VLAN header as part of the transmit data structure or embedded in the transmit descriptor (see Section 7.2.2 for details). The I350 does not relate to the external VLAN header other than the capability of "skipping" it for parsing of inner fields.

Notes:

- If the *CTRL_EXT.EXT_VLAN* bit is set the VLAN header in a packet that carries a single VLAN header is treated as the external VLAN.
- If the *CTRL_EXT.EXT_VLAN* bit is set the I350 expects that any transmitted packet to have at least the external VLAN added by the software. For those packets where an external VLAN is not present, any offload that relates to inner fields to the EtherType may not be provided.
- If the regular VLAN is inserted using the switch based VLAN insertion mechanism (see Section 7.8.3.8.2.2) or from the descriptor (see Section 7.4.3.1), and the packet does not contain an external VLAN, the packet will be dropped, and if configured, the queue from which the packet was sent will be disabled.

7.4.5.2 Receive Behavior With External VLAN

When a port of the I350 is working in this mode, the I350 assumes that all packets received by this port have at least one VLAN, including packet received or sent on the manageability interface.

One exception to this rule are flow control PAUSE packets which are not expected to have any VLAN. Other packets may contain no VLAN, however a received packet that does not contain the first VLAN is forwarded to the host but filtering and offloads are not applied to this packet.

See Table 7-56 for the supported receive processing functions when the device is set to "Double VLAN" mode.

Stripping of VLAN is done on the second VLAN if it exists. All the filtering functions of the I350 ignore the first VLAN in this mode.

The presence of a first VLAN tag is indicated it in the *RDESC.STATUS.VEXT* bit.

Queue assignment of the Rx packets is not affected by the external VLAN header. It may depend on the internal VLAN, MAC address or any upper layer content as described in Section 7.1.1.

Table 7-56 Receive Processing in Double VLAN Mode

VLAN Headers	Status.VEXT	Status.VP	Packet Parsing	Rx offload functions
External and internal	1	1	+	+
Internal Only	Not supported			
V-Ext.	1	0	+	+
None[1]	0	0	+ (flow control only)	-

1. A few examples for packets that may not carry any VLAN header may be: Flow control and Priority Flow Control; LACP; LLDP; GMRP; 802.1x packets

7.5 Configurable LED Outputs

The I350 implements 4 output drivers intended for driving external LED circuits per port. Each LAN device provides an independent set of LED outputs - these pins and their function are bound to a specific LAN device. Each of the four LED outputs can be individually configured to select the particular event, state, or activity, which is indicated on that output. In addition, each LED can be individually configured for output polarity as well as for blinking versus non-blinking (steady-state) indication.

The configuration for LED outputs is specified via the *LEDCTL* register. Furthermore, the hardware-default configuration for all the LED outputs, can be specified via EEPROM fields, thereby supporting LED displays configurable to a particular OEM preference.

Each of the 4 LED's might be configured to use one of a variety of sources for output indication. The MODE bits control the LED source as described in Table 7-57.

The IVRT bits allow the LED source to be inverted before being output or observed by the blink-control logic. LED outputs are assumed to normally be connected to the negative side (cathode) of an external LED.

The BLINK bits control whether the LED should be blinked (on for 200ms, then off for 200ms) while the LED source is asserted. The blink control might be especially useful for ensuring that certain events, such as ACTIVITY indication, cause LED transitions, which are sufficiently visible by a human eye.

Note: When LED Blink mode is enabled the appropriate LED Invert bit should be set to 0b.

The LINK/ACTIVITY source functions slightly different from the others when BLINK is enabled. The LED is off if there is no LINK, on if there is LINK and no ACTIVITY, and blinking if there is LINK and ACTIVITY.

The dynamic LED modes (FILTER_ACTIVITY, LINK/ACTIVITY, COLLISION, ACTIVITY, PAUSED) should be used with LED Blink mode enabled.

7.5.1 MODE Encoding for LED Outputs

Table 7-57 lists the MODE encoding for LED outputs used to select the desired LED signal source for each LED output.

Table 7-57 Mode Encoding for LED Outputs

Mode	Selected Mode	Source Indication
0000b	LINK_10/1000	Asserted when either 10 or 1000 Mb/s link is established and maintained.
0001b	LINK_100/1000	Asserted when either 100 or 1000 Mb/s link is established and maintained.
0010b	LINK_UP	Asserted when any speed link is established and maintained.
0011b	FILTER_ACTIVITY	Asserted when link is established and packets are being transmitted or received that passed MAC filtering.
0100b	LINK/ACTIVITY	Asserted when link is established and when there is no transmit or receive activity.
0101b	LINK_10	Asserted when a 10 Mb/s link is established and maintained.
0110b	LINK_100	Asserted when a 100 Mb/s link is established and maintained.
0111b	LINK_1000	Asserted when a 1000 Mb/s link is established and maintained.

Table 7-57 Mode Encoding for LED Outputs (Continued)

Mode	Selected Mode	Source Indication
1000b	SDP_MODE	LED activation is a reflection of the SDP signal. SDP0, SDP1, SDP2, SDP3 are reflected to LED0, LED1, LED2, LED3 respectively.
1001b	FULL_DUPLEX	Asserted when the link is configured for full duplex operation (de-asserted in half-duplex).
1010b	COLLISION	Asserted when a collision is observed.
1011b	ACTIVITY	Asserted when link is established and packets are being transmitted or received.
1100b	BUS_SIZE	Asserted when the I350 detects a 4-lane PCIe connection.
1101b	PAUSED	Asserted when the I350's transmitter is flow controlled.
1110b	LED_ON	Always high (Asserted)
1111b	LED_OFF	Always low (De-asserted)

7.6 Memory Error Correction and Detection

The I350 main internal memories are protected by error correcting code or parity bits. Large memories or critical memories are protected by an error correcting code (ECC). Smaller memories are protected either with an error correcting code (ECC for critical memories) or by parity.

The I350 reports parity errors in the *PEIND* register according to the region in which the parity error occurred (PCIe, DMA, LAN Port or Management). An interrupt is issued via the *ICR.FER* bit on occurrence of a parity error. Parity error interrupt generation per region can be masked via the *PEINDM* register.

Additional per region granularity in parity or ECC enablement and reporting of parity error or ECC parity correction occurrence is supported in the following registers:

1. PCIe region:
   a. The *PCIEERRCTL* and *PCIEECCCTL* registers enable parity checks and ECC parity correction respectively in the various rams in the PCIe region.
   b. The *PCIEERRSTS* and *PCIEECCSTS* registers report parity error and ECC parity correction occurrence in the various rams in the PCIe region. Only parity errors that were not corrected by the ECC circuitry are reported by asserting the *PEIND.pcie_parity_fatal_ind* bit and the *ICR.FER* bit. Parity errors that were corrected by the internal ECC circuit do not generate an interrupt but are logged in the *PCIEECCSTS* register.
2. DMA region:
   a. The *DTPARC* and *DRPARC* registers enable parity checks while the *DDECCC*, *RPBECCSTS* and TPBECCSTS registers enable ECC parity correction in the various rams in the DMA region.
   b. The *DTPARS* and *DRPARS* registers report detection of parity errors while the DDECCS, *RPBECCSTS* and TPBECCSTS registers report occurrence of ECC parity correction events in the various rams in the DMA region. Only parity errors that were not corrected are reported by setting the *PEIND.dma_parity_fatal_ind* bit and the *ICR.FER* bit. Parity errors that were corrected by the internal ECC circuitry don't generate an interrupt but are logged in the DDECCS, *RPBECCSTS* and TPBECCSTS registers.
3. LAN Port region:
   a. The *LANPERRCTL* register enables parity checks in the various rams in the LAN Port region.
   b. The *LANPERRSTS* register reports detection of parity errors. The parity errors that were not corrected are reported via the *PEIND.lanport_parity_fatal_ind* bit and the *ICR.FER* bit.
   c.

Notes:

1. An interrupt to the Host is generated on occurrence of a fatal memory error if the appropriate mask bits in the *PEINDM* register are set and the *IMS.FER* Mask bit is set.
2. All Parity error checking can be disabled via the *GPAR_EN* bit in the *Initialization Control Word 1* EEPROM word (See Section 6.2.2) or by clearing the *PCIEERRCTL.GPAR_EN* bit (See Section 8.23.11).

7.6.1 Software Recovery From Parity Error Event

If a parity error was detected in one of the internal control memories of the DMA, PCIe or LAN port clusters, the consistency of the receive/transmit flow can not be guaranteed any more. In this case the traffic on the PCIe interface is stopped, since this is considered a fatal error.

If a parity error is detected on the Manageability cluster, PCIe traffic is not affected but an internal reset to the Manageability cluster is generated.

Note: An internal Firmware reset is issued, if enabled by the *PARITY_ERR_RST_EN* bit in the *Common Firmware Parameters 2* EEPROM word, following detection of a parity error in the Management cluster.

To recover from a parity error event Software should initiate the following actions depending on the region in which the parity error occurred.

7.6.1.1 Recovery from PCIe Parity Error Event

To recover from a parity error condition in the PCIe region Software driver should:

1. Issue a Device Reset by asserting the *CTRL.RST* bit.
2. wait at least 3 milliseconds after setting *CTRL.RST* bit before attempting to check if the bit was cleared or before attempting to access any other register.
3. Initiate the master disable algorithm as defined in Section 5.2.3.3.
4. Clear the PCIe parity error status bits that were set in the *PCIEERRSTS* register.
5. Re-initialize the port.

7.6.1.2 Recovery from DMA Parity Error Event

To recover from a parity error condition in the DMA region Software driver should issue a SW reset by asserting the *CTRL.RST* bit as specified in Section 4.3.1 and re-initializing the port.

7.6.1.3 Recovery from LAN Port Parity Error Event

To recover from a parity error condition in the LAN port region Software driver should take the actions depicted in Section 8.23.16 (*LANPERRSTS* register) according to the ram that failed.

7.7 CPU affinity Features

7.7.1 Direct Cache Access (DCA)

7.7.1.1 DCA Description

Direct Cache Access (DCA) is a method to improve network I/O performance by placing some posted inbound writes indirectly within CPU cache. DCA requires that memory writes go to host memory and then the processor prefetch the cache lines specified by the memory write. Through research and experiments, DCA has been shown to reduce CPU Cache miss rates significantly.

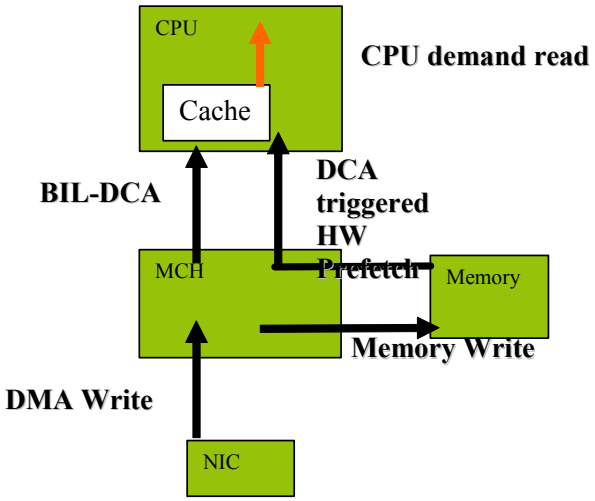


Figure 7-19 Diagram of DCA Implementation on FSB System

As shown in Figure 7-19, DCA provides a mechanism where the posted write data from an I/O device, such as an Ethernet NIC, can be placed into CPU cache with a hardware pre-fetch. This mechanism is initialized upon a power good reset. A software device driver for the I/O device configures the I/O device for DCA and sets up the appropriate DCA target ID for the device to send data. The device will then encapsulate that information in PCIe TLP headers, in the *TAG* field, to trigger a hardware pre-fetch by the MCH /IOH to the CPU cache.

DCA implementation is controlled by separated registers (*RXCTL* and *TXCTL*) for each receive and transmit queue. In addition, a *DCA Enable* bit can be found in the *DCA_CTRL* register, and a *DCA_ID* register can be found for each port, in order to make visible the function, device, and bus numbers to the driver.

The *RXCTL* and *TXCTL* registers can be written by software on the fly and can be changed at any time. When software changes the register contents, hardware applies changes only after all the previous packets in progress for DCA have been completed.

However, in order to implement DCA, the I350 has to be aware of the Crystal Beach version used. Software driver must initialize the I350 to be aware of the Crystal Beach version. A register named *DCA_CTRL* is used in order to properly define the system configuration.

There are 2 modes for DCA implementation:

1. Legacy DCA: The DCA target ID is derived from CPU ID.
2. DCA: The DCA target ID is derived from APIC ID.

The software driver selects one of these modes through the DCA_mode register.

The details of both modes are described below.

7.7.1.2 Details of Implementation

7.7.1.2.1 PCIe Message Format for DCA

Figure 7-20 shows the format of the PCIe message for DCA.

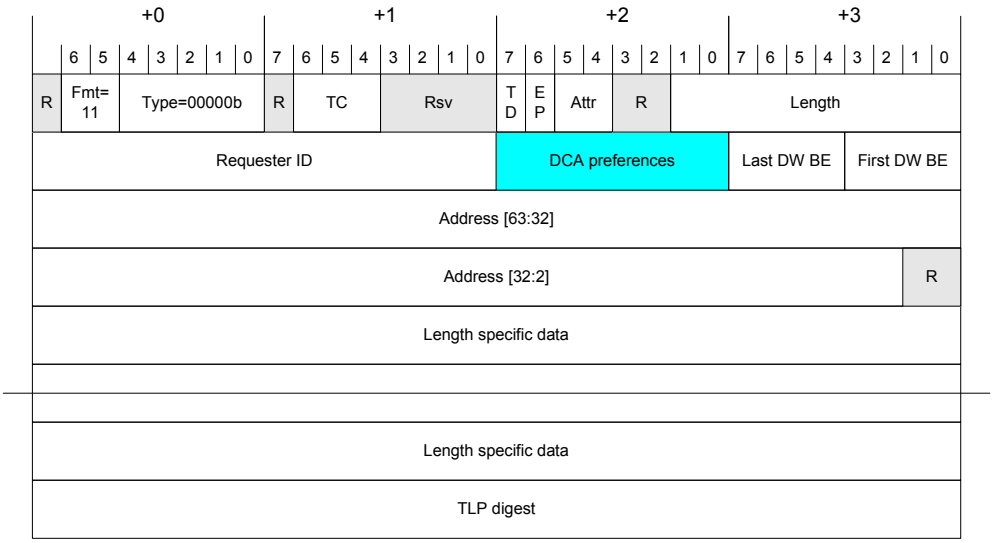


Figure 7-20 PCIe Message Format for DCA

The DCA preferences field has the following formats.

Table 7-58 Legacy DCA Systems

Bits	Name	Description
0	DCA indication	0b: DCA disabled 1b: DCA enabled
4:1	DCA Target ID	The DCA Target ID specifies the target cache for the data.
7:5	Reserved	Reserved

Table 7-59 DCA Systems

Bits	Name	Description
7:0	DCA target ID	0000.0000b: DCA is disabled

Note: All functions within the I350 have to adhere to the "tag encoding" rules for DCA writes. Even if a given function is not capable of DCA, but other functions are capable of DCA, memory writes from the non-DCA function must set the *Tag* field to "00000000".

7.7.2 TLP Process Hints (TPH)

The I350 supports the TPH capability defined in the PCI Express specification (See Section 9.6.3). It does not support Extended TPH requests.

On the PCIe link existence of a TLP Process Hint (TPH) is indicated by setting the TH bit in the TLP header. Using the PCIe TLP Steering Tag (ST) and Processing Hints (PH) fields, the I350 can provide hints to the root complex about the destination (socket ID) and about data access patterns (locality in Cache), when executing DMA memory writes or read operations. Supply of TLP Processing Hints facilitates optimized processing of transactions that target Memory Space.

The I350 supports a steering table with 8 entries in the PCIe TPH capability structure (See Section 9.6.5.4). The PCIe Steering table can be used by Software to provide Steering Tag information to the Device via the *TXCTL.CPUID* and *RXCTL.CPUID* fields.

To enable TPH usage:

1. For a given function, the *TPH Requester Enable* bit in the PCIe configuration *TPH Requester Control Register* should be set.
2. Appropriate TPH Enable bits in *RXCTL* or *TXCTL* registers should be set.

Note: In SR-IOV mode, the *TPH Requester Enable* bit of the VFs does not gate the emission of hints. Thus in VFs, TPH Enable bits in *RXCTL* or *TXCTL* registers should be set only if the *TPH Requester Enable* bit in the config space of the VF is set.

3. Processing hints should be programmed in the *DCA_CTRL.Desc_PH* and *DCA_CTRL.Data_PH* Processing hints (PH) fields.
4. Steering information should be programed in the CPUID fields in the *RXCTL* and *TXCTL* registers.

The Processing hints (PH) and Steering Tags (ST) are set according to the characteristics of the traffic as described in Table 7-60.

Note: In order to enable TPH usage, all the memory reads are done without setting any of the byte enable bits.

Per queue, the DCA and TPH features are exclusive. Software can enable either the DCA feature or the TPH feature for a given queue.

7.7.2.1 Steering Tag and Processing Hint Programming

The following table describes how the Steering tag (socket ID) and Processing hints are generated and how TPH operation is enabled for different types of DMA traffic.

Table 7-60 Steering tag and Processing hint programming

Traffic type	ST Programming	PH value	Enable
Transmit descriptor write back or head write back	TXCTL.CPUID[1]	*DCA_CTRL.Desc_PH*[2]	*Tx Descriptor Writeback TPH EN* field in *TXCTL*.
Receive data buffers write	RXCTL.CPUID[1]	*DCA_CTRL.Data_PH*[3]	*RX Header TPH EN or Rx Payload TPH EN* fields in *RXCTL*.
Receive descriptor writeback	RXCTL.CPUID[1]	*DCA_CTRL.Desc_PH*[2]	*RX Descriptor Writeback TPH EN* field in *RXCTL*.
Transmit descriptor fetch	TXCTL.CPUID[4]	*DCA_CTRL.Desc_PH*[2]	*Tx Descriptor Fetch TPH EN* field in *TXCTL*.
Receive descriptor fetch	RXCTL.CPUID[2]	*DCA_CTRL.Desc_PH*[2]	*Rx Descriptor fetch TPH EN* field in *RXCTL*.
Transmit packet read	TXCTL.CPUID[2]	*DCA_CTRL.Data_PH*[3]	*Tx Packet TPH EN field in TXCTL*.

1. the driver should always set bits [7:3] to zero and place Socket ID in bits [2:0].
2. Default is 00b (Bidirectional data structure).
3. Default is 10b (Target).
4. the hints are always zero.

7.8 Virtualization

7.8.1 Overview

I/O virtualization is a mechanism to share I/O resources among several consumers. For example, in a virtual system, multiple operating systems are loaded and each executes as though the whole system's resources were at its disposal. However, for the limited number of I/O devices, this presents a problem because each operating system might be in a separate memory domain and all the data movement and device management has to be done by a VMM (Virtual Machine Monitor). VMM access adds latency and delay to I/O accesses and degrades I/O performance. Virtualized devices are designed to reduce the burden of VMM by making certain functions of an I/O device shared and thus can be accessed directly from each guest operating system or Virtual Machine (VM).

Two modes to support operation in a Virtualized environment were implemented in previous products:

1. Direct assignment of part of the port resources to different guest OSes using the PCI sig SR-IOV standard. Also known as “Native mode” or pass through mode. This mode is referenced as IOV mode through this chapter
2. Central management of the networking resources by an IOVM or by the VMM. Also known as software switch acceleration mode. This mode is referenced as Next Generation VMDq mode.

The I350 supports fully Next Generation VMDq mode and SR-IOV.

In a virtualized environment, the I350 serves up to 8 virtual machines (VMs) per port. The I350's 8 queues can be accessed by 8 different VMs if configured properly. When the I350 is enabled for multiple queue direct access for VMs, it becomes a VMDq device.

Note: Most configuration and resources are shared across queues. System software must resolve any conflicts in configuration between the VMs.

The virtualization offloads capabilities provided by the I350 apart from the replication of functions defined in the PCI-sig IOV spec are also part of Next Generation VMDq.

An hybrid model, where part of the virtual machines are assigned a dedicated share of the port and the other ones are serviced by an IOVM should also be supported. However, in this case the offloads provided to the software switch might be more limited. This model can be used when parts of the VMs runs OSes for which VF drivers are available and thus can benefit from IOV and others runs older OSes for which VF drivers are not available and are serviced by an intermediary. In this case, the IOVM or VMM is assigned one VF and receives all the packets with MAC addresses of the VMs behind it.

The following section describes the support the I350 provides for virtualization. This chapter assumes a single root implementation of IOV and no support for multi root.

7.8.1.1 Direct Assignment Model

The direct assignment support in the I350 is built according to the software model defined by the SR-IOV spec:

It is assumed that one of the software drivers sharing the port hardware behaves as a master driver (Physical Function or PF driver). This driver is responsible for the initialization and the handling of the common resources of the port. All the other drivers might read part of the status of the common parts but can not change it. The PF driver might run either in the VMM or in some service operating system. It might be part of an IOVM or part of a dedicated service operating system.

In addition, part of the non time critical tasks are also handled by the PF driver. For example, access to CSR through the I/O space or access to the configuration space are available only through the master interface. Time critical CSR space like control of the Tx and Rx queue or interrupt handling is replicated per VF, and directly accessible by the VF driver.

Note: In some systems with a Thick Hypervisor, the Service operating system might be an integral part of the VMM - for these systems, each reference to the service operating system in the document below refers to the VMM.

7.8.1.1.1 Rationale

Direct assignment purpose is to enable each of the virtual machines to receive and transmit packets with minimum of overhead. The non time critical operations such as init and error handling can be done via the PF driver. In addition, it is important that the VMs can operate independently with minimal disturbance. It is also preferable that the VM interface to the hardware should be as close as possible to the native interface in non virtualized systems in order to minimize the software development effort.

The main time critical operations that require direct handling by the VM are:

1. Maintenance of the data buffers and descriptor rings in host memory. In order to support this, the DMA accesses of the queues associated to a VM should be identified as such on the PCIe bus using a different requester ID.
2. Handling of the hardware ring (tail bump and head updates)
3. Interrupts handling.

The capabilities needed to provide independence between VMs are:

1. Per VM reset and enable capabilities.

2. TX Rate control
3. Allocation of separate CSR space per VM. This CSR space is organized as close as possible to the regular CSR space to allow sharing of the base driver code.

Note: The rate control and VF enable capabilities are controlled by the PF.

7.8.1.2 Virtualized System Overview

The following drawings describe the various elements involved in the I/O process in a virtualized system. Figure 7-21 describes the flow in software Next Generation VMDq operation mode and Figure 7-22 the flow in IOV mode.

This document assumes that in IOV mode, the driver on the guest operating system is aware that it works in a virtual system (para-virtualized) and there is a channel between each of the virtual machine drivers and the PF driver allowing message passing such as configuration request or interrupt messages. This channel might use the mailbox system implemented in the I350 or any other method provided by the VMM vendor.

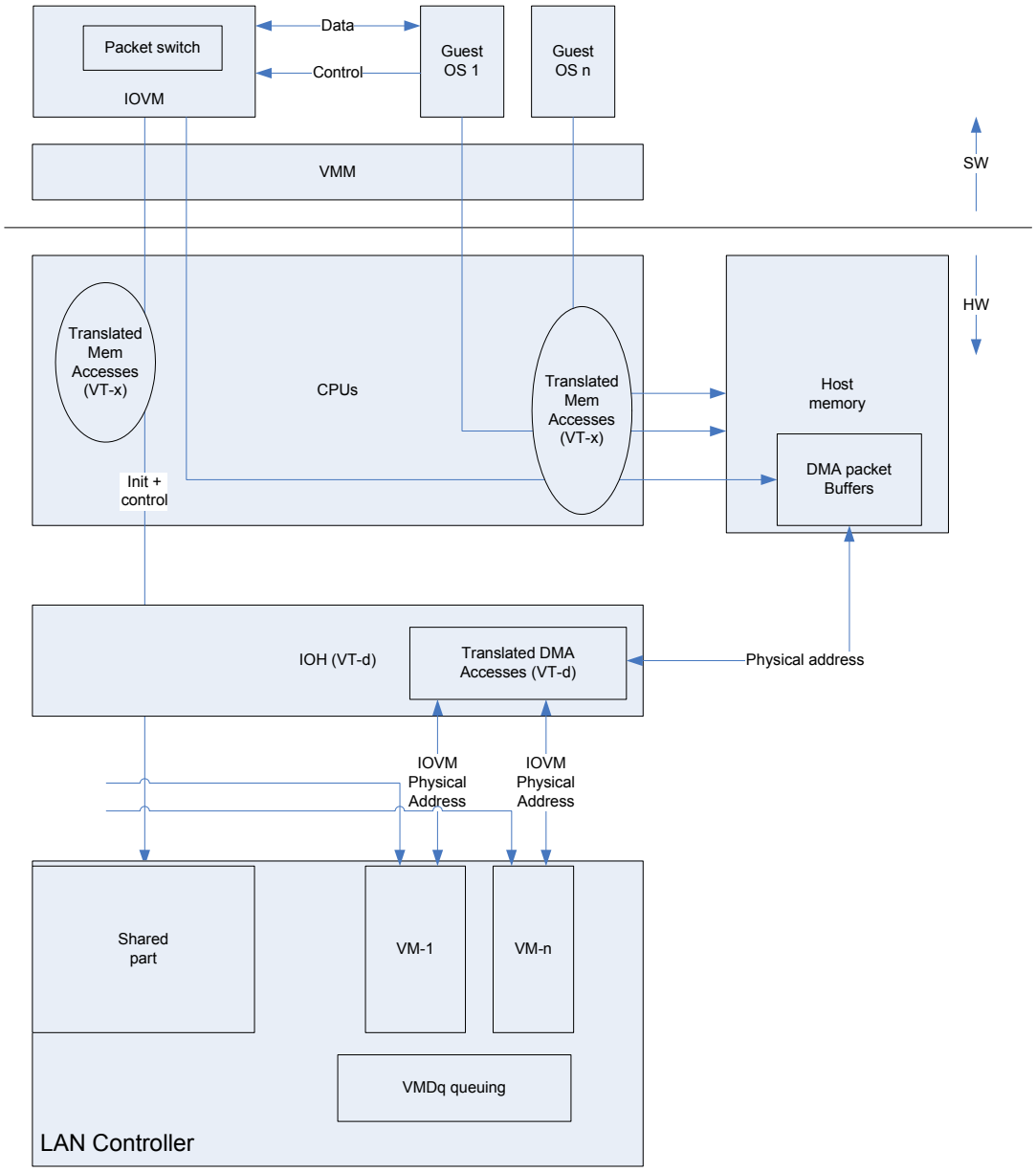


Figure 7-21 IOVM (VMDq) System

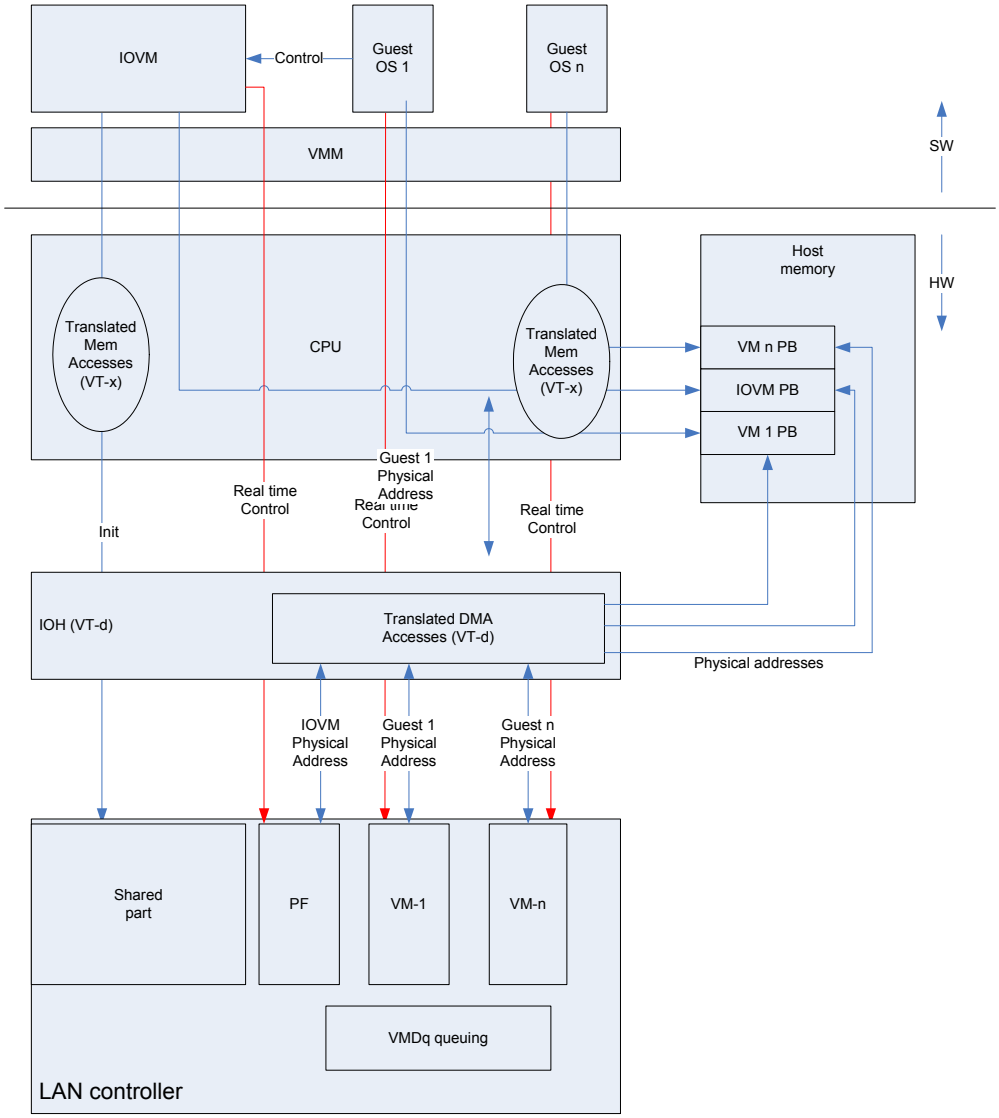


Figure 7-22 SR-IOV Based System

7.8.1.3 VMDq Supported Features

The I350 supports a superset of the VMDq virtualization features supported in the 82576 and in the 82580. The following table compares the virtualization features in the three products.

Table 7-61 I350 Versus the 82576 versus 82580 VMDq Support

Feature	I350 VMDq Support	82576 VMDq Support	82580 VMDq Support
Queues	8	16	8
Pools	8 (single queue)	8	8 (single queue)
MAC addresses	32	24	24
Queuing to pool method	SA or VLAN or (SA and VLAN)	SA or VLAN or (SA and VLAN)	SA or VLAN or (SA and VLAN)
RSS in pool	RSS not supported in VMDq	Common redirection table - enable per pool.	RSS not supported in VMDq
VM to VM Switching	Yes	Yes	No
Broadcast and multicast Replication	Yes	Yes	Yes (Receive only)
MAC and VLAN Anti spoof protection	Yes	Yes	No
VLAN filtering	Global and per pool	Global and per pool	Global and per pool
Drop if no pool	Yes	Yes	Yes
per pool statistics	Yes	Yes	Yes
Per pool offloads	Yes	Yes	Yes
Mirroring	Yes	Yes	Yes (Receive only)
Long packet filtering	Global and per pool	Global and per pool	Global and per pool
Storm Control	Yes	Yes	Yes
Promiscuous modes per VM	VLAN, Multicast, Unicast	Multicast	Multicast

7.8.2 PCI Sig IOV Support

7.8.2.1 IOV Concepts

IOV defines the following entities in relation to I/O virtualization:

1. Virtual Image (VI): A virtual machine to which a part of the I/O resources is assigned. Also known as a VM.
2. I/O Virtual Intermediary (IOVI) or I/O Virtual Machine (IOVM): A special virtual machine that owns the physical device and is responsible for the configuration of the physical device.
3. End point (EP): The physical device that might contain a few physical functions - in our case, the I350.
4. Physical function (PF): A function representing a Physical instance - in our case, one port. The PF driver is responsible for the configuration and management of the shared resources in the function.
5. Virtual function (VF): A part of a PF assigned to a VI.

7.8.2.2 Configuration Space Replication

The IOV working group defines the configuration space of the Virtual functions as a mirror of the Physical function configuration space with the exception of some fields which are implemented per VF. the I350 complies with the SR-IOV specification. This section describes how the I350 implements the configuration space for virtual functions as defined in the specification.

Details of the configuration space for virtual functions can be found in Section 9.7.

7.8.2.2.1 Legacy PCI Config Space

The legacy configuration space is allocated to the PF only and emulated for the VFs. A separate set of BARs and one Bus master enable bit is allocated to the whole set of VFs.

All the legacy error reporting bits are emulated for the VF. See Section 7.8.2.4 for details.

7.8.2.2.2 Memory BARs Assignment

The IOV spec defines a fixed stride for all the VF BARs, so that each VF can be allocated part of the memory BARs at a fixed stride from the a basic set of BARs. In this method only two decoders per replicated BAR per PF are required and the BARs reflected to the VF are emulated by the VMM

The only BARs that are useful for the VFs are BAR0 & BAR3, thus only those are replicated.

The following table describes the existing BARs and the stride used for the VFs:

Table 7-62 VF BARs in the I350

BAR	Type	Usage	Requested Size Per VF
0	Mem	CSR space	min(16K, page size)
1	Mem	High word of CSR space address	N/A
2	N/A	Not used	N/A
3	Mem	MSI-X	min(16K, page size)
4	Mem	High word of MSI-X space address	N/A
5	N/A	Not used	N/A

BAR0 of the VFs are a sparse version of the original PF BAR and includes only the register relevant to the VF. For more details see Section 8.27 and Section 8.28.

The following figure describes the different BARs in an IOV enabled system:

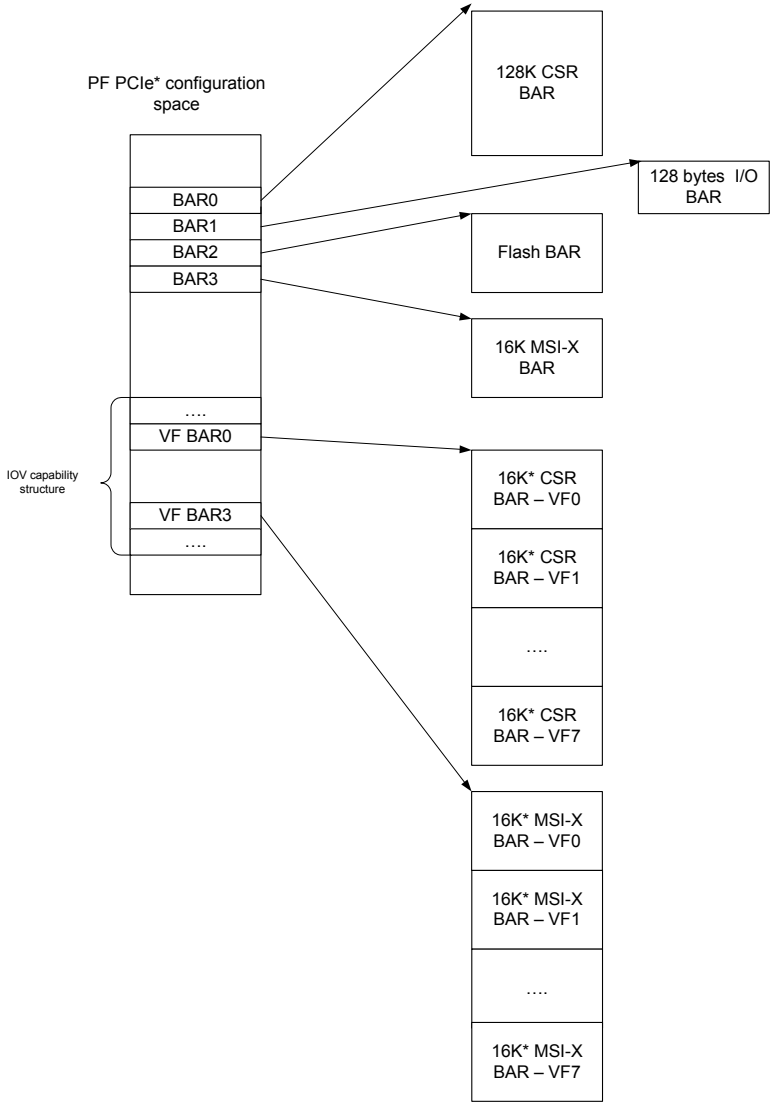


Figure 7-23 Memory BAR Allocation to VFs

7.8.2.2.3 Capability Structures

7.8.2.2.3.1 PCIe capability structure

The PCIe capability structures are shared between the PF & the VFs. The only replicated bits are:

1. Transaction pending
2. Enable No Snoop
3. Enable Relaxed Ordering

4. Initiate FLR
5. Error reporting bits. See Section 7.8.2.4 for details.

7.8.2.2.3.2 MSI and MSI-X Capabilities

Both MSI & MSI-X are implemented in the PF in the I350. MSI-X vectors can be assigned per VF. MSI is not supported for the VFs.

See Section 9.7 for more details of the MSI-X and PBA tables implementation.

7.8.2.2.3.3 VPD Capability

VPD is implemented only once and is accessible only from the PF.

7.8.2.2.3.4 Power Management Capability

PCI SIG SR-IOV specification makes VF power management optional. The I350 does not support power management in VFs.

7.8.2.2.4 Extended Capability Structures

7.8.2.2.5 Serial ID

The serial ID capability is not supported in VFs.

7.8.2.2.6 Error Reporting Capabilities (Advanced & Legacy)

All the bits in this capability structure is implemented only for the PF. The VMs see an emulated version of this. See Section 7.8.2.4 for details.

7.8.2.3 Function Level Reset (FLR) Capability

The *FLR* bit is required per VF. Setting of this bit resets only the part of the logic dedicated to the specific VF and do not influence the shared part of the port. This reset should disable the queues, disable interrupts and stop receive and transmit process per VF.

Setting the *PF FLR* bit resets the entire function.

7.8.2.4 Error Reporting

Error reporting includes legacy error reporting and AER (advanced error reporting or role based) capability.

The legacy error management includes the following functions:

1. Error capabilities enablement. These are set by the PF for all the VFs. Narrower error reporting for a given VM can be achieved by filtering of the errors by the VMM. This includes:
   a. SERR# Enable
   b. Parity Error Response
   c. Correctable Reporting Enable

d. Non-Fatal Reporting Enable
e. Fatal Reporting Enable
f. UR Reporting Enable

2. Error status in the config space. These should be set separately for each VF. This includes:
   a. Master Data Parity Error
   b. Signaled Target Abort
   c. Received Target Abort
   d. Master Abort
   e. SERR# Asserted
   f. Detected Parity Error
   g. Correctable Error Detected
   h. Non-Fatal Error Detected
   i. Unsupported Request Detected

AER capability includes the following functions:

1. Error capabilities enablement. The Error Mask, and Severity bits are set by the PF for all the VFs. Narrower error reporting for a given VM can be achieved by filtering of the errors by the VMM. These includes:
   a. Uncorrectable Error Mask Register
   b. Uncorrectable Error Severity Register
   c. Correctable Error Mask Register
   d. ECRC Generation Enable
   e. ECRC Check Enable
2. Non-Function Specific Errors Status in the config space.
   a. Non-Function Specific Errors are logged in the PF
   b. Error logged in one register only
   c. VI avoids touching all VFs to clear device level errors
   d. The following errors are not function specific
   — All Physical Layer errors
   — All Link Layer errors
   — ECRC Fail
   — UR, when caused by no function claiming a TLP
   — Receiver Overflow
   — Flow Control Protocol Error
   — Malformed TLP
   — Unexpected Completion
3. Function Specific Errors Status in the config space.
   a. Allows Per VF error detection and logging
   b. Help with fault isolation
   c. The following errors are function specific
   — Poisoned TLP received
   — Completion Timeout

— Completer Abort
— UR, when caused by a function that claims a TLP
— ACS Violation

4. Error logging. Each VF has its own header log.
5. Error messages. In order to ease the detection of the source of the error, the error messages should be emitted using the requester ID of the VF in which the error occurred.

7.8.2.5 ARI & IOV Capability Structures

In order to allow more than 8 functions per end point without requesting an internal switch, as usually needed in virtualization scenarios, the PCI sig defines the Alternative Routing ID (ARI) capability structure. This is a capability that allows an interpretation of the Device & Function fields as a single identification of a function within the bus. In addition a new structure used to support the IOV capabilities reporting and control is defined. Both structures are described in sections 9.6.3 & 9.6.4. See next section for details on the RID allocation to VFs.

7.8.2.6 Requester ID Allocation

The requester ID allocation of the VF is done using the *Offset* field in the IOV structure. This field should be replicated per VF and is used to do the enumeration of the VFs.

Each PF includes an offset to the first associated VF. This pointer is a relative offset to the BDF of the first VF. The *Offset* field is added to PF's requester ID to determine requester ID of the next VF. An additional field in the IOV capability structure describes the distance between two consecutive VF's requester ID.

7.8.2.6.1 Bus-Device-Function Layout

7.8.2.6.1.1 ARI Mode

The ARI mode allows interpretation of the device ID part of the RID as part of the function ID inside a device. Thus a single device can span up to 256 functions. In order to ease the decoding, the 2 least significant bits of the function number points to the physical port number. The next bits indicate the VF number. The following table describes the VF requester IDs.

Table 7-63 RID Per VF - ARI Mode

Port	VF#	B,D,F	Binary	Notes
0	PF	B,0,0	B,00000,000	PF #0
1	PF	B,0,1	B,00000,001	PF #1
2	PF	B,0,0	B,00000,010	PF #2
3	PF	B,0,1	B,00000,011	PF #3
0	0	B,16,0	B,10000,000	Offset to first VF from PF is 128.
1	0	B,16,1	B,10000,001	
2	0	B,16,2	B,10000,010	
3	0	B,16,3	B,10000,011	
0	1	B,16,4	B,10000,100	
1	1	B,16,5	B,10000,101	
2	1	B,16,6	B,10000,110	

Table 7-63 RID Per VF - ARI Mode (Continued)

Port	VF#	B,D,F	Binary	Notes
3	1	B,16,7	B,10000,111	
...				
0	7	B,19,4	B,10011,100	
1	7	B,19,5	B,10011,101	
2	7	B,19,6	B,10011,110	
3	7	B,19,7	B,10011,111	Last

7.8.2.6.1.2 Non ARI Mode

When ARI is disabled, non zero devices in the first bus can not be used, thus a second bus is needed to provide enough requester IDs. In this mode, the RID layout is as follow:

Table 7-64 RID Per VF - Non ARI Mode

Port	VF#	B,D,F	Binary	Notes
0	PF	B,0,0	B,00000,000	PF #0
1	PF	B,0,1	B,00000,001	PF #1
2	PF	B,0,2	B,00000,010	PF #2
3	PF	B,0,3	B,00000,011	PF #3
0	0	B+1,16,0	B+1,10000,000	Offset to first VF from PF is 384.
1	0	B+1,16,1	B+1,10000,001	
2	0	B+1,16,2	B+1,10000,010	
3	0	B+1,16,3	B+1,10000,011	
0	1	B+1,16,4	B+1,10000,100	
1	1	B+1,16,5	B+1,10000,101	
2	1	B+1,16,6	B+1,10000,110	
3	1	B+1,16,7	B+1,10000,111	
...				
0	7	B+1,19,4	B+1,10011,100	
1	7	B+1,19,5	B+1,10011,101	
2	7	B+1,19,6	B+1,10011,110	
3	7	B+1,19,7	B+1,10011,111	Last

Note: When the device ID of a physical function changes (because of LAN disable or *FACTPS.LAN Function Sel* settings), the VF device IDs changes accordingly.

7.8.2.7 Hardware Resources Assignment

The main resources to allocate per VM are queues and interrupts. The assignment is a static one. If a VM requires more resources, it might be allocated more than one VF. In this case, each VF gets a specific MAC address/VLAN tag in order to allow forwarding of incoming traffic. The two VFs are then teamed in software.

7.8.2.7.1 Physical Function Resources

A possible use of the Physical function is for configuration setting without transmit and receive capabilities. In this case it is not allocated any queues and is allocated one MSI-X vector.

Physical function have access to all the resources of all the virtual machines but it is not expected to make use of resources allocated to active Virtual Functions.

7.8.2.7.2 Assignment of Queues to VF

Each VF is assigned one queue. Queue N is assigned to VF N.

7.8.2.7.3 Assignment of MSI-X Vectors to VF.

MSI-X vectors are used for three purposes:

1. Differentiation of interrupt causes that avoids the need to read an interrupt cause register.
2. Assignment of different interrupt handling to different CPUs.
3. The implementation of interrupts in the I350 adds another use of allowing different interrupt moderation rates.

The I350 supports 3 MSI-X vectors per PF. The interrupt causes mapped to these MSI-X vectors via the *VTIVAR* and *VTIVAR_MISC* registers are RX traffic interrupt, TX traffic interrupt and Mail Box interrupt. The VF can vary the interrupt rates on these queues using the *VTEITR0*, *VTEITR1* and *VTEITR2* registers.

7.8.2.7.4 VF Resource Summary

The I350 supports 8 VFs per port, each VF can utilize 1 queue pair (Tx/Rx) per VF and 3 MSI-X vectors per VM. If the amount of VFs supported is less than 8, the available resources can be used by the PF.

7.8.2.8 CSR Organization

The CSR of the NIC can be divided to three types:

1. Global Configuration registers that should be accessible only to the PF (such as link control, LED control, etc.). This type of registers includes also all the debug features such as the mapping of the packet buffers and is responsible for most of the CSR area requested by the NIC. This includes per VF configuration parameters that can be set by the PF without performance impact.
2. Per queue parameters that should be replicated per queue (head, tail, Rx buffer size, and DCA tag). These parameters are used both by a VF in an IOV system and by the PF in a non IOV mode.
3. Per VF parameters (per VF reset) interrupt enable. Multiple instances of these parameters are used only in an IOV system and only one instance is needed for non IOV systems.

In order to support IOV without distributing the current drivers operation in legacy mode, the following method is used:

1. The PF instance of BAR0 continues to contain the legacy and control registers. It is accessible only to the PF. The BAR allows access to all the resources including the VF queues and other VF parameters. However it is expected that the PF driver does not access these queues in IOV mode.
2. The VF instances of BAR0 provide control of the VF specific registers. These registers have the same relative mapping to BAR0 of the VF as the original BAR0 of the PF with the following exceptions:
   a. Fields related to the shared resources are reserved.

b. The VF queue related registers are mapped at the same relative location to BAR0 as the queue registers of the first queue (Queue 0) of the PF.

3. To supply backward compatibility for the IOV drivers, the PF/VF parameters block contains a partial register set as described in Section 8.27 and Section 8.28.

7.8.2.9 IOV Control

In order to control the IOV operation, the physical driver is provided with a set of registers. These includes:

1. The mailbox mechanism described below.
2. The switch and filtering control registers described in Section 7.8.3.10.
3. VFLRE: register indicating that a VFLR reset occurred in one of the VFs (bitmap).
4. VFTE: Enables Tx traffic per VF. A VF Tx is disabled by an FLR to this VF until the PF enables it again. This allows the PF to block transmit process until the configurations for this VM are done.
5. VFRE: Enables Rx filtering per VF. A VF Rx is disabled by an FLR to this VF until the PF enables it again. This allows the PF to block the receive process until the configurations for this VM are done.

7.8.2.9.1 VF to PF Mailbox

The VF drivers and the PF driver requires some mean of communication between them. This channel can be used for the PF driver to send status updates to the VFs (link change, memory parity error, etc.) or for the VF to send requests to the PF (add to VLAN).

Such a channel can be implemented in software, but it requires enablement by the VMM vendors. In order to avoid the need for such an enablement, the I350 provides such a channel that allows direct communication between the two drivers.

The channel consists of a mailbox similar to the host interface currently defined between the software and the manageability. Each driver can then receive and indication (either poll or interrupt) when the other side wrote a message.

Assuming a max message size of 64 bytes (one cache line), RAM of 64 bytes x 8 VMs= 0.5 Kbyte is provided. Table 7-65 shows how RAM is organized.

Table 7-65 Mailbox Memory

RAM Address	Function	PF BAR 0 Mapping[1]	VF BAR 0 Mapping[2]
0 - 63	VF0 ↔ PF	0 - 63	VF0 + MBO
64 - 127	VF1 ↔ PF	64 - 127	VF1 + MBO
....			
448 - 512	VF7 ↔ PF	448 - 512	VF7 + MBO

1. Relative to mailbox offset
2. MBO = mailbox offset in VF CSR space

In addition for each VF, the *VFMailbox* & *PFMailbox* registers are defined in order to coordinate the transmission of the messages. These registers contains a semaphore mechanism to allow coordination of the mailbox usage.

The PF driver can decide which VFs are allowed to interrupt the PF to indicate a mailbox message using the *MBVFIMR* mask register.

The following flows describes the usage of the mailbox:

Table 7-66 PF to VF Messaging Flow

Step	PF Driver	Hardware	VF #n Driver
1	Set *PFMailbox[n].PFU*		
2		Set *PFU* bit if *PFMailbox[n].VFU* is cleared	
3	Read *PFMailbox[n]* and check that *PFU* bit was set. Otherwise wait and go to step 1		
4	Read *MBVFICR* register and verify that *VFREQ* bit of VF[n] is 0, otherwise clear PFU bit in *PFMailbox[n]* and respond to the VF message.		
5	Write message to relevant location in *VMBMEM*		
6	Clear *PFMailbox[n].PFU* and set the *PFMailbox[n].STS* bit and wait for *ACK*[1].		
7		Indicate an interrupt to VF #n	
8			Set *VFMailbox.VFU*
9		Set *VFU* bit if *VFMailbox[n].PFU* is cleared	
10			Read *VFMailbox[n]* and check that *VFU* bit was set. Otherwise wait and go to step 8
11			Read the message from *VMBMEM*
12			Set the *VFMailbox.ACK* bit
13		Indicate an interrupt to PF	
14	Read *MBVFICR* to check that *VFACK* bit of VF[n] was set. Otherwise wait and recheck.		

1. The PF might implement a timeout mechanism to detect non responsive VFs.

Table 7-67 VF to PF Messaging Flow

Step	PF Driver	Hardware	VF #n Driver
1			Set *VFMailbox.VFU*
2		Set *VFU* bit if *VFMailbox[n].PFU* is cleared	
3			Read *VFMailbox[n]* and check that *VFU* bit was set and that *PFSTS* bit is clear. Otherwise: 1. If *PFSTS* bit was set clear *VFU* bit and respond to PF message. 2. Else if *VFU* bit was clear wait and go to step 1.
4			Write message to relevant location in *VMBMEM*
5			Clear *VFMailbox.VFU* bit and set the *VFMailbox.REQ* bit
6		Indicate an interrupt to PF via *ICR.SWMB*	
7	Read *MBVFICR* to detect which VF caused the interrupt		

Table 7-67 VF to PF Messaging Flow (Continued)

Step	PF Driver	Hardware	VF #n Driver
8	Set *PFMailbox[n].PFU* bit		
9		Set *PFU* bit if *PFMailbox[n].VFU* bit is cleared	
10	Read *PFMailbox[n]* and check that *PFU* bit was set. Otherwise wait and go to step 8		
11	Read the adequate message from *VMBMEM*		
12	Clear *PFMailbox[n].PFU* bit and Set the *PFMailbox.ACK* bit		
13		Indicate an interrupt to VF #n	
14			Read *VFMailbox[n]* and check that the *ACK* bit was set. Otherwise wait and recheck

The content of the message is hardware independent and can be fixed by the software.

The messages currently assumed by this specification are:

1. Registration to VLAN/Multicast packet/Broadcast packets - A VF can request to be part of a given VLAN or to get some multicast/broadcast traffic.
2. Reception of large packet - Each VF should notify the PF driver what is the largest packet size allowed in receive.
3. Get global statistics - A VF can request information from the PF driver on the global statistics.
4. Filter allocation request - A VF can request allocation of a filter for queuing/immediate interrupt support.
5. Global interrupt indication.
6. Indication of errors.

7.8.2.10 Interrupt Handling

Interrupts can be separated into two types:

1. Interrupts relevant to the behavior of each VM, including Rx & Tx packet sent indications, mailbox message and device status indications.
2. Interrupts relevant only to the handling of the shared resources. These are mainly error indications - such as packet buffer full and parity errors.

The first type of interrupts should be provided directly to the VM driver and the second type can be handled by the PF driver.

Interrupt control in the VF uses the same mechanism as the in the non virtualized case. The cause bits are independent and each VF can clear its own cause bits independently. The following registers are added per VF:

1. *VTEICR*, *VTEICS*, *VTEIMS*, *VTEIMC*, *VTEIAC*, *VTEIAM* with the following fields:
   a. RTxQ[1:0]
   b. Mailbox
2. *VTEITR0,1,2*.

3. *VTIVAR & VTIVAR_MISC* for Mailbox.

7.8.2.10.1 Low latency Interrupts

Low Latency Interrupts (LLI) are described in Section 7.3.6. Several packet types generate LLI:

- A packet matching a 5-tuple filter assigned to a VF - Each VF can require from the PF driver an LLI for one of its flows.
- A packet matching a L2 EtherType filter - an LLI is generated to specific VFs (based on the queue assignment) that handle control traffic.
- A packet matching a certain VLAN priority - an LLI is generated to the target VF based on the queue assignment for the Rx packet

An AND condition on the VM number is added to the immediate interrupt decision in order to prevent a VM from requiring immediate interrupts for flows not owned by it and in order to allow a filter to apply only to a given VM. For example, assume a given VM would require immediate interrupts on packets with PSH flag set. The VM number filtering prevents other VM from receiving immediate interrupts on such packets.

7.8.2.10.2 MSI-X

MSI-X tables are in BAR3. The MSI-X vectors might be used either as one big set of vectors in non IOV mode or as small sets allocated to VFs. In order to support both modes and save the need for duplication of the logic the first IOV vectors should be mapped as non IOV vectors also. The mapping of the vectors in IOV mode is described in Section 8.28.49

The PBA vector is replicated for the IOV case, as the saving in area is low and different per bit encoding is complicated.

7.8.2.10.3 MSI

MSI implementation is optional in the IOV spec. The I350 doesn’t support MSI in Virtual functions.

7.8.2.10.4 Legacy Interrupt (INT-x)

Legacy interrupts are not supported in IOV mode.

7.8.2.11 DMA

7.8.2.11.1 Requester ID

Each VF is allocated a requester ID. Each DMA request should use the RID of the VF that requested it. See Section 7.8.2.6 for details.

7.8.2.11.2 Sharing DMA Resources

The outstanding requests an completion credits are shared between all the VFs. The tags attached to read requests are assigned the same way they are today, although in VF systems tags can be re-used for different requester IDs.

7.8.2.11.3 DCA and TPH

The DCA enable is common to all the devices (all PFs & VFs). Given a DCA enabled device, each VM might decide for each queue, on which type of traffic (data, headers, Tx descriptors, Rx descriptors) DCA should be asserted and what is the CPU ID assigned to this queue.

The TPH enable bit is set per VF. Each TPH enabled VM might decide for each queue, on which type of traffic (data, headers, Tx descriptors, Rx descriptors) DCA should be asserted and what is the CPU ID assigned to this queue.

Note: There are no plans to virtualize DCA or TPH in the root Complex. Thus the physical CPU ID should be used in the programming of the *CPUID* field.

7.8.2.12 Timers and Watchdog

7.8.2.12.1 TCP Timer

The TCP timer is available only to the PF. It might indicate an interrupt to the VFs via the mailbox mechanism.

7.8.2.12.2 IEEE 1588

IEEE 1588 is a per link function and thus is controlled by the PF driver. The VMs have access to the real time clock register.

7.8.2.12.3 Watchdog.

The watchdog was originally developed for pass-through NICs where virtualization is not an interesting use case. Thus, this functionality is used only by the PF.

7.8.2.12.4 Free Running Timer

The free running timer is a PF driver resource the VMs can access. This register is read only to all VF. It is reset only by the PCI reset.

7.8.2.13 Power Management and Wake-up

Power management, Wake-up and proxying is a PF resource and is not supported per VF.

7.8.2.14 Link Control

The link is a shared resource and as such is controllable only by the PF. This include PHY settings, speed and duplex settings, flow control settings, etc. The flow control packets are sent with the station MAC address stored in the EEPROM. The watermarks of the flow control process and the time-out value are also controllable by the PF only.

Double VLAN is a network setting and as such should be common to all VFs.

7.8.2.14.1 Special Filtering Options

Pass Bad packets is a debug feature. As such pass bad packet is available only to the PF. Bad packets is passed according to the same filtering rules of the regular packets. As it might cause guest operating system s to get unexpected packets, it should be used only for debug purposes of the whole system.

Reception of long packet is controlled separately per VM. As this impact the flow control thresholds, the PF should be made aware of the decision of all the VMs. Because of this, the setup of the large send packets is centralized by the PF and each VF might request this setting.

7.8.2.15 IOV Test Mode

In order to support testing of IOV features in non IOV enabled systems, such as OEM test-benches and early validation platform, a few features are added to the I350. The first one, is a possibility to allow VFs to use the PF Requester ID in their DMA transactions - this is done by setting the *GCR.ignore RID* bit. A second mode, allows some of the VF config bits to behave as copies of the *PF Config* bit - specifically the MSI-X interrupt enable, MSI-X interrupt mask and MBE bits. This is done by setting the GCR.IOV test mode bit.

An additional debug feature, is the possibility to access the config space of the PF and all the VFs via the regular CSR space. This mode allows the PF driver to access the IOV settings and the specific VFs config space. In addition, it allows access to the VFs config spaces for configuration even if the operating system and the BIOS are not aware of these functions existence. The different config spaces are accessible through the CIAA & CIAD register. The CIAA register contains the VF number and the offset of the register in the config space to access. It also contains a bit indicating if the VF configuration space are accessed through this mechanism or through regular configuration transactions. A read of the CIAD register will return the value of the config register pointed by the CIAA register. A value written to the CIAD register will be written to the config register pointed by the CIAA register.

7.8.2.15.1 Allocation of memory space for IOV functions

If the BIOS didn't allocate memory for the IOV functions, the following flow may be used to allocate memory to the I350 virtual function:

1. In the EEPROM request some space for the serial flash BAR. This space should be large enough to cover the IOV VF memory space needs. For example, assuming the memory page size is 4K and 8 VFs are enabled, then 256 KBytes of RAM should be requested (16 K for the CSR BAR, 16 K for the MSI-X BAR by 8 functions).
2. Before enabling IOV, zero the flash BAR and program the IOV BARs to use the old flash BAR. The VFs CSR BAR may use the first half of the original flash memory and the MSI-X BAR may use the second half.

7.8.3 Packet Switching (VMDq) Model

7.8.3.1 VMDq Assumptions

The following assumption are made:

1. The required bandwidth for the VM to VM loopback traffic is low. For example, the PCIe BW is not congested by the combination of the VM to VM and the regular incoming traffic. This case is handled but not optimized for. Unless specified otherwise, Tx and Rx packets should not be dropped or lost due to congestion caused by loopback traffic.
2. Most of the offloads provided on Rx traffic are not provided for the VM to VM loopback traffic.

3. If the buffer allocated for the VM to VM loopback traffic is full, it is OK to back pressure the transmit traffic of the same traffic class. This mean that the outgoing traffic might be blocked if the loopback traffic is congested.
4. The decision on VM to VM loopback traffic is done only according to the Ethernet DA address and the VLAN tag. There is no filtering according to other parameters (IP, L4, etc.). This switch have no learning capabilities.
5. The forwarding decisions are based on the receive filtering programming.
6. When the link is down, the Tx flow is stopped, and thus the local switching traffic is stopped also.

7.8.3.2 VM/VF Selection

The VM/VF selection is done by MAC address and VLAN tag. Broadcast and Multicast packets are forwarded according to the individual setting of each VM and might be replicated to multiple VMs.

7.8.3.2.1 Filtering Capabilities

The following capabilities exists in to decide what is the final destination of each packet in addition to the regular L2 filtering capabilities:

- 32 MAC addresses filters (*RAH*/*RAL* registers) for both unicast and multicast filtering. These are shared with L2 filtering. For example, the same MAC addresses are used to determine if a packet is received by the switch and to determine the forwarding destination.
- 32 Shared VLAN filters (*VLVF* registers) - each VM can be made member of each VLAN.
- Multicast exact filtering using the existing remaining *RAH*/*RAL* registers otherwise an imperfect multicast table is shared between VMs.
- 256 hash filtering of multicast addresses shared between the VMs (*MTA* table).
- Promiscuous unicast, multicast & enable broadcast per VM.
- Promiscuous VLAN per VM.

Note: Packets for which no queueing decision was done and still accepted by the L2 filtering, is directed to the queue pool of the default VM/VF or dropped.

7.8.3.3 L2 Filtering

L2 filtering is the 1st stage of 3 stages that determine the destination of a received packet. The 3 stages are defined in Section 7.1.1.

All received packets pass the same filtering as in the non virtualized case; regular VLAN filtering using the global VLAN table (*VFTA*) of the PF and filtering according to the *RAH*/*RAL* registers and according to the various promiscuous bits.

Note: Every VLAN tag set in the *VLVF* registers should be asserted also in the *VFTA* table.

Note: The *RCTL.UPE* bit (Promiscuous unicast) is not available per VM and might be modified by the IOVM or VMM and might be modified only by the PF driver. This bit should be set if VMOLR.UPE is set for one of the VF drivers.

7.8.3.4 VMDq Receive Packets Switching

Receive packet switching is the 2nd stage of 3 stages that determine the destination of a received packet. The 3 stages are defined in Section 7.1.1.

As far as switching is concerned, it doesn't matter whether our virtual environment operates in IOV mode or in Next Generation VMDq mode. In this stage the The VM/VF is identified by the "pool list" as described in Section 7.8.3.4.1 and Section 7.8.3.4.2.

When working in a virtualized environment, the 3rd stage of definition of a queue is not relevant.

When working in replication mode, broadcast and multicast packets can be forwarded to more than one VM, and can be replicated to more than one receive queue. Replication is enabled by the *RplI_En* bit in the VT_CTL register.

In virtualization mode, the pool list is a list of one or more VMs to which the packet should be forwarded. The pool list is used in choosing the target queue list except for cases in which high priority filters take precedence. There is a difference in the way the pool list is generated when replication mode is enabled or disabled.

7.8.3.4.1 VMDq Replication Mode Enabled

When replication mode is enabled (*VT_CTL.RplI_En* = 1), each broadcast/multicast packet can go to more than one pool. Finding the pool list should be done according to the following steps:

1. Exact unicast or multicast match - If there is a match in one of the exact filters (*RAL*/*RAH*), for unicast or multicast packets, take the *RAH.POOLSEL[7:0]* field as a candidate for the pool list.
2. VFRE — If any bit in the VFRE register is cleared, clear the respective bit in the pool list.
3. Broadcast - If the packet is a broadcast packet, add pools for which their *VMOLR.BAM* bit (Broadcast Accept Mode) is set.
4. Unicast hash - If the packet is a unicast packet, and the prior steps yielded no pools, check it against the Unicast Table Array hash (*UTA*). If there is a match, add pools for which their *VMOLR.ROPE* bit (Receive Overflow packet enable) is set.
5. Multicast hash - If the packet is a multicast packet and the prior steps yielded no pools, check it against the multicast Table Array hash (*MTA*). If there is a match, add pools for which their *VMOLR.ROMPE* bit (Receive Multicast packet enable) is set.
6. Multicast Promiscuous - If the packet is a multicast packet, take the candidate list from prior steps and add pools for which their *VMOLR.MPE* bit (Multicast Promiscuous Enable) is set.
7. Unicast Promiscuous - If the packet is a unicast packet, take the candidate list from prior steps and add pools for which their *VMOLR.UPE* bit (Unicast Promiscuous Enable) is set.
8. Ignore MAC (VLAN only filtering) - If *VT_CTL.IGMAC* bit is set, then the previous steps are ignored and a full pool list is assumed for the next step.
9. VLAN groups - This step is relevant only if the *RCTL.VFE* bit is set, otherwise it is skipped. Packets should be sent only to VMs that belong to the packet's VLAN group. Thus the following rules are applied only to entries set in the pool list by previous steps:
    a. Tagged packets: enable only pools in the packet's VLAN group as defined by the VLAN filters - *VLVF[n].VLAN_id* and their pool list - *VLVF[n].POOLSEL[7:0]* or pools for which the *VMOLR.VPE* (VLAN Promiscuous Enable) is set.
    b. Untagged packets: enable only pools with their *VMOLR.AUPE* bit set
    c. If there is no match, the pool list should be empty.
    d. If the *VLVF.LVLAN* is set, then the packet is not received from the network even if one of the previous conditions is met and the packet is dropped.

Note: In a VLAN network, untagged packets are not expected. Such packets received by the switch should be dropped, unless their destination is a virtual port set to receive these packets. The setting is done through the *VMOLR.AUPE* bit. It is assumed that VMs for which this bit is set are members of a default VLAN and thus only MAC queuing is done on these packets.

10. VFRE — If any bit in the *VFRE* register is cleared, clear the respective bit in the pool list.
11. Default Pool - If the pool list is empty at this stage and the *VT_CTL.Dis_Def_Pool* bit is not set, then set the default pool bit in the target pool list (from *VT_CTL.Def_PL*).
12. Ethertype filters - If the one of the Ethertype filters (*ETQF*) matches the packet and queuing action is requested, the VM list is set only to the pool pointed by the filter.
13. Filter Local Packets: If the *VT_CTL.FLP* bit is set, and the packet SA matches one of the RAH/RAL, remove from the VM list all the VMs set in the *RAH.POOLSEL[7:0].*
14. VFRE — If any bit in the VFRE register is cleared, clear the respective bit in the pool list.
15. Length Limit: If the packet is longer than a legal Ethernet packet, remove from the pool list all the pools for which the *VMOLR.LPE* bit is not set or for which the packet length is larger than the value in the *VMOLR.RLPML* field.
16. Mirroring - If the pool list is not empty, for each of the 4 mirroring rules add the destination (mirroring) pool (*VMRCTL.MP*) to the pool list according to the following rules:
    a. Pool mirroring - if *VMRCTL.VPME* is set and one of the bits in the pool list matches one of the bits in the *VMRVM* register.
    b. VLAN port mirroring - if *VMRCTL.VLME* is set and the index of the VLAN of the packet in the *VLVF* table matches one of the bits in the *VMRVLAN* register.
    c. Uplink port mirroring - if *VMRCTL.UPME* is set and the packet came from the LAN.
    d. *VFRE* — If any bit in the *VFRE* register is cleared, clear the respective bit in the pool list.
17. Length Limit: If the packet is longer than a legal Ethernet packet, remove from the pool list all the pools for which the *VMOLR.LPE* bit is not set or for which the packet length is larger than the value in the *VMOLR.RLPML* field.

7.8.3.4.2 VMDq Replication Mode Disabled

When replication mode is disabled (*VT_CTL.Rpl_En* = 0), the software should take care of multicast and broadcast packets and check which of the VMs should get them. In this mode the pool list always contains one pool only according to the following steps:

1. Exact unicast or multicast match - If the packet DA matches one of the exact filters (*RAL/RAH*), take the *RAH.POOLSEL[7:0]* field as a candidate for the pool list.
2. VFRE — If any bit in the *VFRE* register is cleared, clear the respective bit in the pool list
3. Unicast hash - If the packet is a unicast packet, and the prior steps yielded no pools, check it against the Unicast Table Array hash (*UTA*). If there is a match, add the pool for which the *VMOLR.ROPE* bit (Receive Overflow packet enable) is set. (See software limitation no 3. below).
4. Ignore MAC (VLAN only filtering) - If *VT_CTL.IGMAC* bit is set, then the previous step is ignored and a full pool list is assumed for the next step.
5. VLAN groups - This step is relevant only if the *RCTL.VFE* bit is set, otherwise it is skipped. Packets should be sent only to VMs that belong to the packet's VLAN group. Thus the following rules are applied only to entries set in the pool list by previous steps:
    a. Tagged packets: enable only pools in the packet's VLAN group as defined by the VLAN filters - *VLVF[n].VLAN_id* and their pool list - *VLVF[n].POOLSEL[7:0]* or pools for which the *VMOLR.VPE* (VLAN Promiscuous Enable) is set.
    b. Untagged packets: enable only pools with their *VMOLR.AUPE* bit set
    c. If there is no match, the pool list should be empty.

   d. If the *VLVF.LVLAN* is set, then the packet is not received from the network even if one of the previous conditions is met and the packet is dropped.
6. VFRE — If any bit in the *VFRE* register is cleared, clear the respective bit in the pool list
7. Default pool - If the packet is a unicast packet or *VT_CTL.IGMAC* is set and no pool was chosen and the *VT_CTL.Dis_Def_Pool* bit is not set, then set the default pool bit in the pool list (from *VT_CTL.Def_PL*).
8. Broadcast or Multicast - If the packet is a Multicast or Broadcast packet and V*T_CTL.IGMAC* is not set and no pool was chosen, set the default pool bit in the pool list (from *VT_CTL.Def_PL*).
9. Ethertype filters - If the one of the Ethertype filters (*ETQF*) matches the packet and queuing action is requested, the VM list is set only to the pool pointed by the filter.
10. Filter Local Packets: If the *VT_CTL.FLP* bit is set, and the packet SA matches one of the RAH/RAL, remove from the VM list the VM set in the *RAH.POOLSEL[7:0].*
11. Length Limit: If the packet is longer than a legal Ethernet packet, remove from the pool list all the pools for which the *VMOLR.LPE* bit is not set or for which the packet length is larger than the value in the *VMOLR.RLPML* field.
12. VFRE — If any bit in the *VFRE* register is cleared, clear the respective bit in the pool list.

The following limitations applies when replication is disabled:

1. It is the software responsibility to not set more than one bit in the bitmaps of the exact filters. Note that multiple bits might be set in an RAH register as long as it is guaranteed that the packet is sent to only one queue by other means (VLAN)
2. The software must not set per-VM promiscuous bits (unicast, multicast or broadcast) in the *VMOLR* register.
3. The software must not set the *ROPE* bit in more than one *VMOLR* register.
4. If *VT_CTL.IGMAC* bit is set, the software must not set the *VMOLR.AUPE* in more than one *VMOLR* register and must not set more than one bit in each of the *VLVF.POOLSEL* bitmaps and must not set the *VMOLR.VPE* in any pool.
5. The software must not activate mirroring.

7.8.3.5 TX Packets Switching

TX switching is used only in a virtualized environment to serve VM to VM traffic. Packets that are destined to one or more local VMs, are loop backed to the RX path through a separate packet buffer. Enabling TX switching is done by setting the *TXSWC.Loopback_en* bit.

TX switching is very similar to RX switching in a virtualized environment, with the following exceptions and rules:

- The high priority filters (EType/SYN/5-Tuple) are not applied to the Tx traffic.
- If a target pool is not found, the default pool is not used, and the packet will only go to the external LAN.
- A unicast packet that is destined to one of the VMs by an exact filter is not sent to the LAN unless the *RAH.TRMCST* (Treat as Multicast) bit is set.
- Broadcast and multicast packets are always sent to the external LAN too, unless member of a local VLAN.
- If an outgoing packets VLAN matches a VLVF entry with the *LVLAN* bit set, this packet is not sent to the external LAN. This rule overrides previous rules.
- A packet might not be sent back to the originating VM (even if the destination address is equal to the source address). However, In order to off-load a software switch allowing Multiple VMs sharing the same pool or for VF loopback diagnostics, the I350 provides the capability to loopback packets

inside a pool. In the normal case, a packet whose source and destination are the same is dropped (usually occurs with multicast packets). If the Local Loopback bit mode (*LLE*) in *TXSWC* is set for this pool, packets originating from a given pool can be sent to the same pool.

The detailed flow for pool selection is described below.

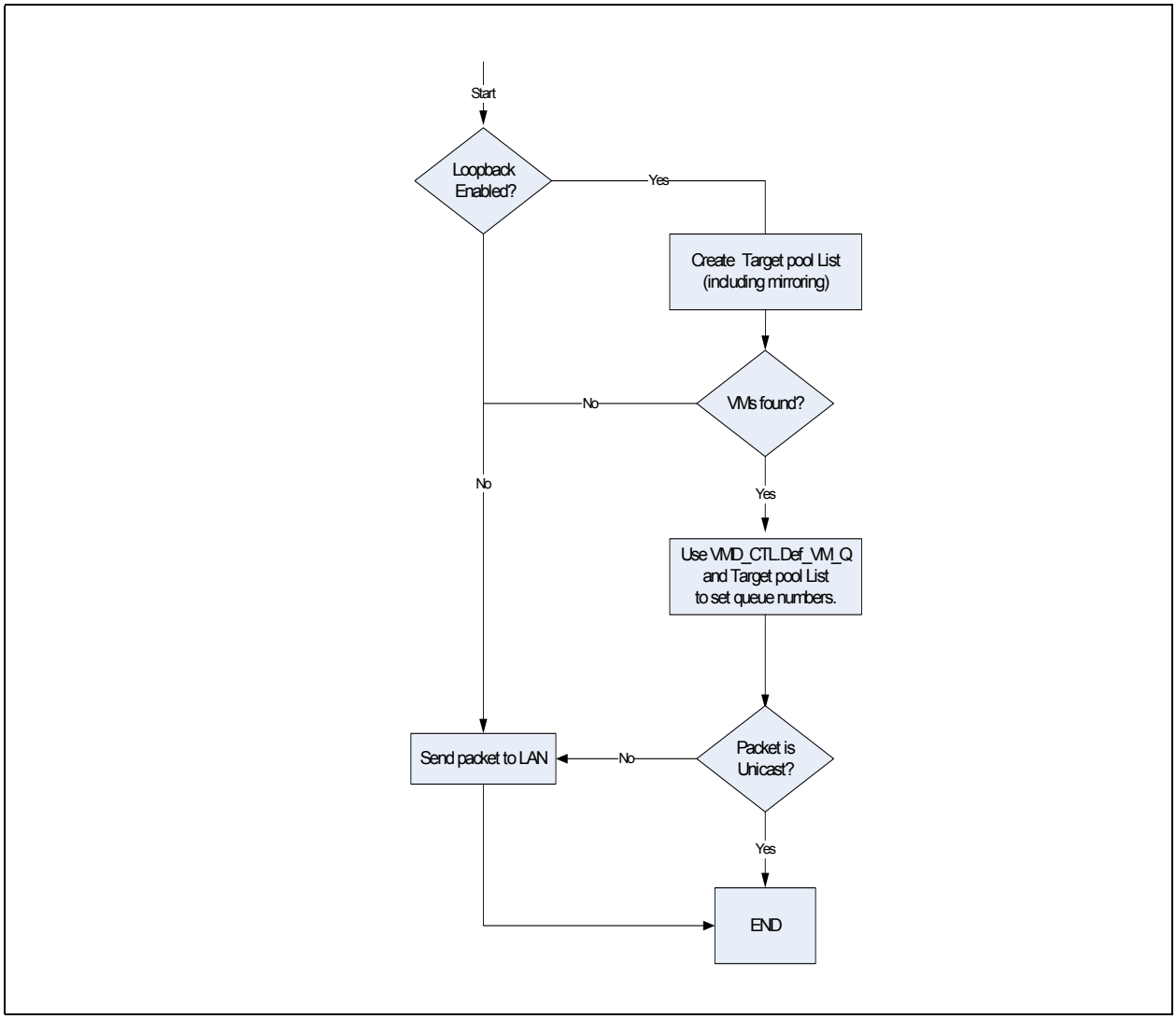


Figure 7-24 Tx Filtering

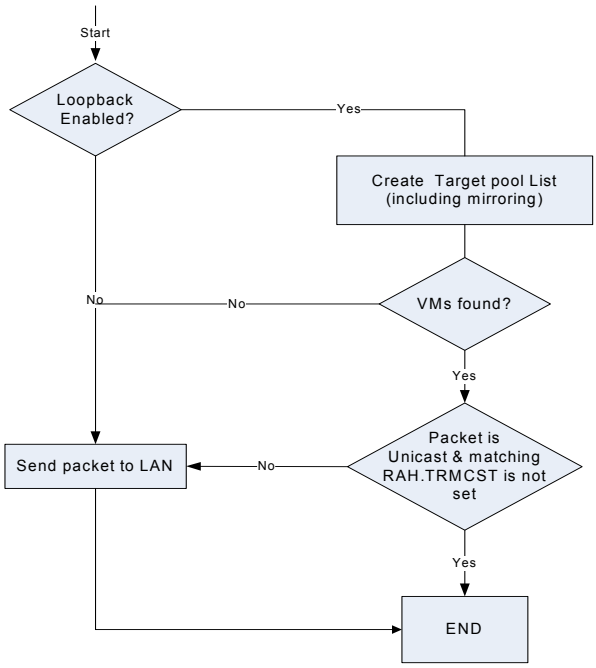


Figure 7-25 Tx Filtering

The following rules apply to loopback traffic:

- Loopback is disabled when the network link is disconnected. It is expected (but not required) that system software (including virtual machines) does not post packets for transmission when the link is disconnected.
- Loopback is disabled when the *Receive Enable (RXEN)* bit is cleared.
- Loopback packets are identified by the *LB* bit in the receive descriptor.

Note: NLB is a mode of Microsoft where a unicast address behaves as a multicast address in that it is used by multiple machines. In order to support this mode, it should be possible to forward part of the unicast MAC addresses to the network the same way we do for multicast addresses. In order to support this mode, the RAH.TRMCST bit is added. This bit is used to decide if packets are forwarded to the network even if they are forwarded to local addresses.

7.8.3.5.1 Replication Mode Enabled

When replication mode is enabled, the pool list for Tx packets is determined according to the following steps:

1. Exact unicast or multicast match - If there is a match in one of the exact filters (*RAL/RAH*), for unicast or multicast packets, take the *RAH.PLSEL* field as a candidate for the pool list.
2. VFRE — If any bit in the *VFRE* register is cleared, clear the respective bit in the pool list.
3. Broadcast - If the packet is a broadcast packet, add pools for which their *VMOLR.BAM* bit (Broadcast Accept Mode) is set.

4. Unicast hash - If the packet is a unicast packet, and the prior steps yielded no pools, check it against the Unicast Table Array hash (*UTA*). If there is a match, add pools for which their *VMOLR.ROPE* bit (Receive Overflow packet enable) is set.
5. Multicast hash - If the packet is a multicast packet and the prior steps yielded no pools, check it against the Multicast Table Array hash (*MTA*). If there is a match, add pools for which their *VMOLR.ROMPE* bit (Receive Multicast packet enable) is set.
6. Multicast Promiscuous - If the packet is a multicast packet, take the candidate list from prior steps and add pools for which their *VMOLR.MPE* bit (Multicast Promiscuous Enable) is set.
7. Unicast Promiscuous - If the packet is a unicast packet, take the candidate list from prior steps and add pools for which their *VMOLR.UPE* bit (Unicast Promiscuous Enable) is set.
8. Ignore MAC (VLAN only filtering) - If *VT_CTL.IGMAC* bit is set, then the previous step is ignored and a full pool list is assumed for the next step.
9. VLAN groups - This step is relevant only if the *RCTL.VFE* bit is set, otherwise it is skipped. Packets should be sent only to VMs that belong to the packet's VLAN group. Thus the following rules are applied only to entries set in the pool list by previous steps:
   a. Tagged packets: enable only pools in the packet's VLAN group as defined by the VLAN filters - *VLVF[n].VLAN_id* and their pool list - *VLVF[n].POOLSEL[7:0]* or pools for which the *VMOLR.VPE* (VLAN Promiscuous Enable) is set.
   b. Untagged packets: enable only pools with their VMOLR.AUPE bit set
   c. If there is no match, the pool list should be empty.
10. Forwarding to the Network:
    a. All broadcast and multicast packets are sent to the network also.
    b. Unicast packets: If *VT_CTL.IGMAC* bit is cleared and after step 2. the pool list is empty or the matching *RAH.TRMCST* bit is set, the packet is sent to the network also. If *VT_CTL.IGMAC* bit is set and the pool list at this stage is empty the packet is sent to the network also.
    c. If the *VLVF.LVLAN* is set, then the packet is not sent to the network even if one of the previous conditions is met.
11. VFRE — If any bit in the *VFRE* register is cleared, clear the respective bit in the pool list.
12. Length Limit: If the packet is longer than a legal Ethernet packet, remove from the pool list all the pools for which the *VMOLR.LPE* bit is not set or for which the packet length is larger than the value in the *VMOLR.RLPML* field.
13. Filter source port - The pool from which the packet was sent is removed from the pool list unless the *TXSWC.LLE* bit is set.
14. Ingress Mirroring - If the pool list is not empty, each of the 4 mirroring rules adds its destination pool (VMRCTL.MP) to the pool list if the following applies:
    a. Pool mirroring - *VMRCTL.VPME* is set and one of the bits in the pool list matches one of the bits in the VMRVM register.
    b. VLAN port mirroring - *VMRCTL.VLME* is set and the index of the VLAN of the packet in the *VLVF* table matches one of the bits in the *VMVLAN* register.
    c. VFRE — If any bit in the *VFRE* register is cleared, clear the respective bit in the pool list.
15. Egress mirroring - For each of the 4 mirroring rules, if *VMRCTL.DPME* is set and the packet is sent to the network add the destination pool (VMRCTL.MP) to the pool list.
16. Length Limit: If the packet is longer than a legal Ethernet packet, remove from the pool list all the pools for which the *VMOLR.LPE* bit is not set or for which the packet length is larger than the value in the *VMOLR.RLPML* field.

7.8.3.5.2 Replication Mode Disabled

When replication mode is disabled, the software should take care of multicast and broadcast packets and check which of the VMs should get them. In this mode the pool list for Tx packets always contains at the most one pool according to the following steps:

1. Exact unicast or multicast match - If the packet DA matches one of the exact filters (RAL/RAH), take the *RAH.PLSEL* field as a candidate for the pool list.
2. VFRE — If any bit in the VFRE register is cleared, clear the respective bit in the pool list.
3. Unicast hash - If the packet is a unicast packet, and the prior steps yielded no VMs, check it against the Unicast hash table (UTA). If there is a match, add pools for which their *VMOLR.ROPE* bit (Receive Overflow packet enable) is set.
4. Ignore MAC (VLAN only filtering) - If *VT_CTL.IGMAC* bit is set, then the previous steps are ignored and a full pool list is assumed for the next step.
5. VLAN groups - This step is relevant only if the *RCTL.VFE* bit is set, otherwise it is skipped. Packets should be sent only to VMs that belong to the packet's VLAN group. Thus the following rules are applied only to entries set in the pool list by previous steps:
   a. Tagged packets: enable only pools in the packet's VLAN group as defined by the VLAN filters - *VLVF[n].VLAN_id* and their pool list - *VLVF[n].POOLSEL[7:0]* or pools for which the VMOLR.VPE (VLAN Promiscuous Enable) is set.
   b. Untagged packets: enable only pools with their VMOLR.AUPE bit set.
   c. If there is no match, the pool list should be empty.
6. Forwarding to the Network:
   a. All broadcast and multicast packets are sent to the network also.
   b. Unicast packets: If *VT_CTL.IGMAC* bit is cleared and after step 2. the pool list is empty or the matching *RAH.TRMCST* bit is set, the packet is sent to the network also. If *VT_CTL.IGMAC* bit is set and the pool list at this stage is empty the packet is sent to the network also.
   c. If the *VLVF.LVLAN* is set, then the packet is not sent to the network even if one of the previous conditions is met.
7. Broadcast or Multicast - If the packet is a Multicast or Broadcast packet and V*T_CTL.IGMAC* is not set and no pool was chosen, set the default pool bit in the pool list (from *VT_CTL.Def_PL*).
8. Length Limit: If the packet is longer than a legal Ethernet packet, remove from the pool list all the pools for which the *VMOLR.LPE* bit is not set or for which the packet length is larger than the value in the *VMOLR.RLPML* field.
9. VFRE — If any bit in the *VFRE* register is cleared, clear the respective bit in the pool list.
10. Filter source port - The pool from which the packet was sent is removed from the pool list unless the *TXSWC.LLE* bit is set.

The limitations listed in Section 7.8.3.4.2 applies for Tx traffic also.

7.8.3.6 Mirroring Support

The I350 supports 4 mirroring rules. Each rule can be of one of 5 types. Mirroring is supported only to virtual ports and not to the uplink (i.e. a mirrored packet can not be sent back to the Network).

Mirroring should be activated only when one of the VMDq queueing modes is used.

The following types of rules are supported:

1. Virtual port mirroring - reflects all the packets sent to a set of given VMs.
2. Uplink port mirroring - reflects all the traffic received from the network.
3. Downlink port mirroring - reflects all the traffic transmitted to the network.

4. Receive mirroring - reflects all the traffic received by any of the VMs. Either from the network or from local VMs. This is supported by enabling mirroring of all VMs.
5. VLAN mirroring - reflects all the traffic received in a set of given VLANs. Either from the network or from local VMs.

All the modes can be accumulated into a single rule.

These mirroring rules are controlled by a set of rule control registers:

- *VMRCTL* - controls the rules to be applied and the destination port.
- *VMRVLAN* - controls the VLAN ports as listed in the *VLVF* table taking part in the VLAN mirror rule.
- *VMRVM* - controls the VMs ports taking part of the Virtual port mirror rule.

Mirroring is supported only when replication is enabled. The exact flow of mirroring is described in step 15. in Section 7.8.3.4.1.

7.8.3.7 Offloads

7.8.3.7.1 Split Header Offload

In case of packets directed to one VM only, the split header size is determined by the specific VM setting. However, the I350 can not apply different split header size to different replication of the same packet. The following sections describes the rules used to decide which split header size to apply in case of replicated packets.

7.8.3.7.1.1 Replication by Exact MAC Address

As mentioned above, the same MAC address can be assigned to more than one VM. This is used for the following cases:

- Multicast address - In this case, the different VMs might be part of the same VLAN. The header size applied to packets matching this address is defined in the *RPLPSRTYPE* register.
- Unicast - Same MAC different VLAN - In this case, each VM should belong to different VLAN(s). The applied offloads is according to the pool selected by the MAC/VLAN pair.

7.8.3.7.1.2 Replication by Promiscuous Modes

A packet might be replicated to multiple VMs because part of the VMs are set to receive all multicast or broadcast packets or because of a packet matching one of the hash tables (*UTA* or *MTA*). The header size applied to packet is defined in the *RPLPSRTYPE* register.

In case of unicast packet, the header size is applied according the first of the pools selected to receive the packet.

7.8.3.7.1.3 Replication by Mirroring

- Header size of mirrored packets are determined according to the original pool.

7.8.3.7.2 Local VM to VM Traffic Offload

Most of the offloads available for regular incoming traffic are not available in case of loop back traffic. The driver might handle the lack of the offloads as follows:

1. **Checksum** - The transmit path always adds a checksum - either by the driver or by the I350, but this checksum is not validated by the receive path. As this packet wasn't sent over the network, the receive side might assume the TCP and IP checksums are valid.
2. **Packet types identification** - The L3 packet type identification is provided only if at least one of the following offloads is requested for the transmitted packet: IP checksum or L4 checksum. The L4 packet type identification is provided only if L4 checksum is requested for the transmitted packet. A packet might be identified as IPv4 with extensions only if IP checksum was requested on this packet. L5 Packet type identification is not valid for loop back packets.
3. **Header split & replication** - Available only for part of the local packets. It is available only if the header split boundary is at the L4 level (TCP/UDP), in cases where the Tx side provided a valid L4 packet type (in packets for which L4 checksum is requested). In all other cases the SPH is set to zero.
4. **Error bits** - The error bits are also fixed to zero - although most of the errors are not relevant for loop back packets.
5. **Special queueing filters** such as 5-tuple filter or ether-type filter are not applied to the local traffic. A driver using such filters should check if a packet belongs to a special queue and redirect it accordingly.

7.8.3.7.3 Small Packets Padding

In Virtualized systems, the driver receiving the packet in the VM might not be aware of all the hardware offloads applied to the packet. Thus, in case of stripping actions by the hardware (VLAN strip), it might receive packets which are smaller than a legal packet. The I350 provides an option to pad small packets in such cases so that all packets have a legal size. This option can be enabled only if the CRC is stripped. In these cases, all packets are padded to 60 bytes (legal packet - 4 bytes CRC). The padding is done with zero data. This function is enabled via the *RCTL.PSP* bit.

7.8.3.8 Security Features

The I350 allows some security checks on the inbound and outbound traffic of the switch.

7.8.3.8.1 Inbound Security

Each incoming packet (either from the LAN or from a local VM) is filtered according to the VLAN tag so that packets from one VLAN can not be received by VMs that are not members of that VLAN.

When the VLAN is inserted by the switch, it is preferable to hide the received VLAN from the guest OS, as exposing it can create a security hole. The I350 allows removal of the VLAN tag from received packet, so that the receiving VM is not aware of the VLAN network it belongs to.

This mode is controlled using the *hide VLAN* bit in the *DVMOLR* register. If this bit is set, the VLAN is always stripped, a value of zero is written in the *RDESC.VLAN tag* and in the *RDESC.STATUS.VP* fields of the received descriptor.

7.8.3.8.2 Outbound Security

7.8.3.8.2.1 Anti-Spoofing

The source MAC address of each outgoing packet can be compared to the MAC address the sending VM uses for packets reception. A packet with a non matching SA is dropped. Thus preventing spoofing of the MAC address. In this mode, the SA of a transmit packet is compared with the addresses stored in

the *RAH/RAL* registers. If a match is found and the pool from which the packet was sent is enabled in the *RAH.POOLSEL*, the packet can be forwarded, otherwise the packet is dropped and a notification is sent to the VMM via the *ICR.MDDET* bit and *WVBR.WVM* field.

This feature is enabled in the *TXSWC.MACAS* field, and can be enabled per VF.

Note: MAC Anti Spoofing is not available for VMs that hides behind them other VMs whose MAC addresses are not part of the RAH/RAL MAC address registers. In this case anti-spoofing should be done by the software switching handling these VMs.

If VLAN anti spoofing is set, a check is done to validate that sender is a member of the VLAN set in the packet. In this mode, the VLAN ID of a transmit packet is compared with the VLAN tag stored in the VLVF registers. If a match is found and the pool from which the packet was sent is enabled in the *VLVF.POOLSEL*, the packet can be forwarded, otherwise the packet is dropped and a notification is sent to the VMM via the *ICR.MDDET* bit and *WVBR.WVM* field. The VMOLR.AUPE bit is used to decide if untagged packets can be forwarded.

This feature is enabled via the *TXWSC.VLANAS* field, and can be enabled per VF.

Note: VLAN anti spoofing is not available for pools programmed to receive all VLANs (VMOLR.VPE is set).

7.8.3.8.2.2 VLAN Insertion From Register Instead of Descriptor

There are cases, where the VLAN should be inserted by the switch without intervention from the guest operating system. In VMDq mode, where the physical driver is controlled by a trusted central entity, we can assume the software requests inserting the right tag. However, in IOV scenarios, the driver might be malicious, and thus we can not assume it uses the right VLAN tags. In order to overcome this issue, default VLAN tags are defined per VM, and a default behavior is defined. The possible behaviors that can be set in the *VMVIR.vlana* field are:

1. Use descriptor value - to be used in case of a trusted VM that can decide which VLAN to send. This option should be used also in case one VM is member of multiple VLANs.
2. Always insert default VLAN value defined in *VMVIR.Port VLAN ID* field - this mode should be used for non trusted or non VLAN aware VMs. In this case any VLAN insertion command from the VM is ignored. If a packet is received with a VLAN, the packet should be dropped.
3. Never insert VLAN - This mode should be used in non VLAN network. In this case any VLAN insertion command from the VM is ignored. If a packet is received with a VLAN, the packet should be dropped.

Note: The VLAN insertion settings should be done before any of the queues of the VM are enabled.

7.8.3.8.2.3 Egress VLAN Filtering

Part of the VLANs used by VMM vendors are VLAN local to the virtualized server. Packets sent with a private VLAN should not forwarded to the external network. Local VLANs are indicated by setting the *LVLAN* bit in the adequate VLVF entry.

Note: A packet with a local VLAN tag whose destination is not in the server is dropped. This means that a local VLAN should be confined to one physical port and can not have member VMs connected to different ports even in the same NIC.

7.8.3.8.3 Interrupt on Misbehavior of VM (Malicious Driver Detection)

The hardware can be programmed to take some action as a result of some misbehavior of a VM. For example upon detection of a packet with a wrong source MAC address, the hardware might block the packet. These actions might hint to the fact that some VM is malicious and the VMM should remedy the situation. In order to inform the VMM of this fact, an interrupt bit exists in the ICR register (*ICR.MDDET* bit) to indicate the occurrence of such behavior. The *LVMMC* register contains information on which queue (*LVMMC.Last_Q*) and port (*LVMMC.Mal_PF*) the malicious behavior was detected. The *LVMMC* register is clear by read.

Malicious driver behavior detection is enabled by setting the *DTXCTL.MDP_EN* bit to 1. On detection of a malicious driver event the I350 stops activity of the offending queue, asserts relevant bit in the *MDFB.Block Queue* field and generates an interrupt by asserting the *ICR.MDDET* bit. Cause of Malicious driver activation is reported in the *LVMMC* register. To re-activate offending queue, driver should either reset the offending VF or re-enable the relevant VFTE bit.

7.8.3.8.3.1 Transmit Descriptor Validity Checks

The table below describes the checks are done by the I350 to define if a transmit packet descriptor is valid. All the checks are done on the descriptors. The checks on the packet header are described in the previous sections.

Table 7-68 Malicious Driver - TX descriptor checks

Check type	Description	Action
Mac Header size	Checks that the MAC header size in the context descriptor is at least 14 (or 18 in case of offloaded packet and double VLAN).	Drop Packet and stop offending queue
IPV4 Header	If a checksum or TSO offload is required, checks that the IPv4 header size in the context descriptor is at least 20.	Drop Packet and stop offending queue
IPV6 Header	If a TSO offload is required, checks that the IPv6 header size in the context descriptor is at least 40.	Drop Packet and stop offending queue
Wrong MAC_IP	Check that the MAC+ IP header size is not bigger than the packet size.	Drop Packet and stop offending queue
TCP header size	If a TCP TSO offload is required, checks that the TCP header size in the context descriptor is at least 20.	Drop Packet and stop offending queue
UDP header size	If a UDP TSO offload is required, checks that the TCP header size in the context descriptor is at least 8.	Drop Packet and stop offending queue
SCTP data size	If a SCTP checksum offload is required, checks that the SCTP L4 packet size (including header and data) is at least 12.	Drop Packet and stop offending queue
Packet too big	In case of a single send, check that the packet is not larger than the value set in the *DTXMXPKTSZ* register.	Drop Packet and stop offending queue
Packet too small	Check that the total length of the packet transmitted, not including FCS is at least 13 bytes (or 17 if double VLAN is enabled).	Silently drop packet.
Illegal offload request.	Check that TSO is no requested for SCTP or that a checksum offload is not requested for IPv6 packets.	Drop Packet and stop offending queue
SCTP alignment	If an SCTP CRC offload is requested, check that the data size is 4 byte aligned.	Drop Packet and stop offending queue
Zero MSS	Check that the MSS size is larger than zero.	Drop Packet and stop offending queue
Context in middle of packet	Check that a context descriptor is not sent in the middle of a packet.	Drop Packet and stop offending queue
Number of large send header buffers	Check that the Large send header is contained in at most 4 buffers.	Drop Packet and stop offending queue
Buffers size and length match - Single Send	For single send, check that the total of all buffers size and the packet length match.	Drop Packet and stop offending queue

Table 7-68 Malicious Driver - TX descriptor checks

Check type	Description	Action
Buffers size and length match - Large Send	For LSO, check that the total of all buffers size and the packet length match.	Drop Packet and stop offending queue
UDP data size	If a UDP checksum offload is required, checks that the UDP L4 packet size in the context descriptor is at least 8.	Drop Packet and stop offending queue
TCP data size	If a TCP checksum offload is required, checks that the TCP L4 packet size in the context descriptor is at least 20.	Drop Packet and stop offending queue
Descriptor Type	Check that only descriptor types 2 (context) or 3 (advanced data descriptor) are used.	Drop Packet and stop offending queue
Null packet check	Check that a Null packet has the EOP bit set.	Drop Packet and stop offending queue
Packet without EOP	Check that a only entire packets are provided by the driver	Drop Packet and stop offending queue
Burst of contexts	Check that less than *DTXCTL.Cswthres* Contiguous context descriptor are sent by the driver.	Drop Packet and stop offending queue

7.8.3.8.3.2 Reactive Malicious behavior detection

The table below describes the checks are done by the I350 to detect a malicious behavior, even if the packet seems valid.

Table 7-69 Reactive malicious checks

Check type	Description	Action
Malicious VF memory access	A PCIe DMA access initiated by a VF ended with Unsupported Request (UR) or Completer Abort (CA). This check is done for both Tx and Rx queues.	Drop Packet and stop offending queue
Invalid Queue parameters	Check that the queue length is not null before accepting a queue enable.	Ignore queue enable.

7.8.3.8.4 Storm Control

As there is no separate path for multicast & broadcast packets, too much replicated packets might cause congestions in the data path. In order to avoid such scenarios, broadcast and multicast storm control rate limiters are added. The rate controllers define windows and the maximal allowed number of multicast or broadcast bytes/packets per window. Once the threshold is crossed different types of policies can be applied.

7.8.3.8.4.1 Assumptions

- Only one interval size and interval counter is used for both broadcast & multicast storm control mechanisms.
- The threshold and actions for each mechanism are separate.
- The traffic used to calculate the broadcast & multicast rate is all the traffic with a local destination - either Tx or Rx.
- The storm control does not block traffic to the network.
- The basic unit of traffic counted is 64 bytes of data.

7.8.3.8.4.2 Storm Control Functionality

The time interval over which Broadcast Storm control is performed is controlled by three factors.

- *SCBI* register
- Port speed.
- The value in *SCCRL.INTERVAL*

The first two factors determine the Unit time interval as described in Table 7-70. The interval is automatically chosen internal to hardware based on port speed. The third factor (*Interval* field) determines how many of such unit intervals are considered for one Storm Control Interval.

Table 7-70 Storm Control Interval by Speed

Port Speed	MIN Time Interval	MAX Time Interval
1 Gb/s	100 µs	100 ms
100 Mb/s	1 ms	1 s
10 Mb/s	10 ms	10 s

The number of 64 bytes chunk of Broadcast or Multicast packets that are allowed in a given interval is determined by setting the *BSCTRH* or *MSCTRH* register respectively.

The I350 supports two modes of reactions to storm event:

1. Block all Multicast or Broadcast packets from the moment the threshold is crossed until the end of the interval. The block is removed at the end of the interval until the threshold is crossed again. This mode is set by asserting *SCCRL.MDICW* (for multicast) or *SCCRL.BDICW* (for broadcasts). This mode is used as a rate limiter.
2. Block all Multicast or Broadcast packets from the moment the threshold is crossed until a full interval without threshold crossing is registered. The block is removed at the end of the interval until the threshold is crossed again. This mode is set by asserting *SCCRL.MDICW* and SCCRL.MDIPW (for multicast) or *SCCRL.BDICW* and *SCCRL.BDIPW* (for broadcasts). This mode is used for storm blocking.

The I350 can be programmed to add all packets for which a queue was not found for storm control calculation. For example, packets that passed the 1st stage of L2 filtering but didn't pass the 2nd stage of pooling, or were sent to the default pool, as broadcast packets. This mode is activated by setting the *SCCRL.BIDU* field.

Any change in the storm control state (block or pass of multicast or broadcast packets) is indicated to the software via the *ICR.SCE* interrupt cause. The current state is reflected in the *SCSTS* register.

For diagnostic purpose only, the storm control timer and counters can be read via the *SCTC*, *MSCCNT* & *BSCCNT* registers.

7.8.3.9 External Switch Loopback Support

One of the solutions for the switching issue is a mode where an external switch would do the loopback of VM to VM traffic and the NIC is responsible for the replication of multicast packets only. In order to support this mode, the internal loopback mode should be disabled and received packets SA should be compared to the exact MAC addresses to check if the packet originated from a local source, so that the packet is not forwarded to the VM originator. This mode is enabled by the *VT_CTL.FLP* bit.

7.8.3.10 Switch Control

The VMM/IOVM driver has some control of the switch logic. The following registers are available to the VMM/IOVM for this purpose:

VLVF: VLAN queuing table: A set of 32 VLAN entries with an associated per VM/VF bit map allowing allocation of each VF or VM to each of the 32 VLAN tags.

TXSWC: DM Tx Switch control register - controls the security setting of the switch such as MAC & VLAN anti spoof filters, local loopback enable and the loopback enable mode.

QDE: Queue Drop Enable register(s): A register defining wether receive packets destined to a specific queue is dropped if no descriptor are available. This register overrides the individual SRRCTL.DROP_EN bits.

VT_CTL: VT Control register - contains the following fields:

- Replication enable - allows replication of multicast & broadcast packets - both in incoming & outgoing traffic. If this bit is cleared, Tx multicast & broadcast packets are sent only to the network and Rx multicast & broadcast packets are sent to the default VM.
- Default pool - defines where to send packets that passed L2 filtering but didn't pass any of the queueing mechanisms.
- Default pool disable- defines whether to drop packets that passed L2 filtering but didn't pass any of the queueing mechanisms.

VMVIR: A set of registers used to control VLAN insertion of outgoing packets.

VMOLR/DVMOLR: Defines the offloads and pool selection options for each VF or VM.

In addition the storm control mechanism is programmed as described in Section 7.8.3.8.4.2.

7.8.4 Virtualization of the Hardware

This section describes additional features used in both IOV & Next Generation VMDq modes.

7.8.4.1 Per Pool Statistics

Part of the statistics are by definition shared and can not be allocated to a specific VM. For example, CRC error count can not be allocated to a specific VM, as the destination of such a packet is not known if the CRC is wrong.

All the non specific statistics is handled by the PF driver in the same way as it is done in non virtualized systems. A VM might require a statistic from the PF driver but might not access it directly.

The conceptual model used to gather statistics in a virtualization context is that each queue pool is considered as a virtual link and the Ethernet link is considered as the uplink of the switch. Thus any packet sent by a VM is counted in the Tx statistics, even if it was forwarded to another VM internally or dropped by the MAC from some reason. In the same way, a replicated packet is counted in each of the VMs receiving it.

The following statistics are provided per VM:

1. Good Packet received count (*VFGPRC*).
2. Good Packet transmitted count (*VFGPTC*).
3. Good octets received count (*VFGORC*).

4. Good octets transmitted count (*VFGOTC*).
5. Rx Packet dropped because of queue descriptors not available (RQDPC).
6. Tx Packet dropped because of packet buffer (TQDPC).
7. Multicast Packets Received Count (*VFMPRC*).
8. Good Packet received from local VM count (*VFGPRLBC*).
9. Good Packet transmitted to local VM count (*VFGPTLBC*).
10. Good octets received from local VM count (*VFGORLBC*).
11. Good octets transmitted to local VM count (*VFGOTLBC*).

Note: All the per VM statistics are read only (RO) and wrap around after reaching their maximal value.

7.8.4.1.1 Byte Count Statistics

The component of a packet that are taken into account when calculating the per VF byte statistics (*VFGORC, VFGOTC, VFGORLBC, and VFGOTLBC*) varies according to the following rules:

- For transmit statistics (*VFGOTC, and VFGOTLBC*), VLAN tag is part of the byte count only if inserted by the VM. I.e. if the *VMVIR.VLANA* for the VM equals 00 (use descriptor command) and the packet contains a VLAN either in the packet or in the descriptor. CRC is part of the byte count if *DMATXCTRL.Count CRC* is set.
- For receive statistics (*VFGORC, and VFGORLBC*), VLAN tag is part of the byte count only if reported to the VM. I.e. if the *DVMOLR.HIDE VLAN* is not set for this VM. CRC is part of the byte count if *DMATXCTRL.Count CRC* is set.

Note:

The following tables summarize the size of the packet used for the byte count statistics based on different system settings and packet properties.

Table 7-71 Tx Packet Size for VFGOTC, and VFGOTLBC

VLAN source	CRC source	Count CRC	Source Packet size[1]	Statistics Packet size
VMVIR (port based)	IFCS = 1 (Hardware inserted)	Yes	< 60	64
			>60	Source packet size +4
		No	< 60	60
			>60	Source packet size
	IFCS = 0 (Software inserted)	Illegal configuration		
VLE (From descriptor)	IFCS = 1 (Hardware inserted)	Yes	< 56	64
			>56	Source packet size +8
		No	< 56	60
			>56	Source packet size + 4
	IFCS = 0 (Software inserted)	Illegal configuration		

Table 7-71 Tx Packet Size for VFGOTC, and VFGOTLBC (Continued)

VLAN source	CRC source	Count CRC	Source Packet size[1]	Statistics Packet size
In packet	IFCS = 1 (Hardware inserted)	Yes	< 60	64
			>60	Source packet size +4
		No	< 60	60
			>60	Source packet size
	IFCS = 0 (Software inserted)	Yes	< 64	64
			> 64	Source packet size
		No	< 64	60
			> 64	Source Packet size - 4

1. The source packet size defines the packet size as sent by the software.

Table 7-72 Rx packet size for VFGORC, and VFGORLBC

VLAN source	VLAN in Packet?	Count CRC	Statistics Packet size
Hide VLAN (port based)	Yes	Yes	Source packet size[1]-4
		No	Source packet size-8
	No	Yes	Source packet size
		No	Source Packet size - 4
Do not hide VLAN (In descriptor or in packet)	Yes	Yes	Source packet size
		No	Source packet size-4
	No	Yes	Source packet size
		No	Source Packet size - 4

1. The source packet size defines the packet size as sent by the network.

7.9 Time SYNC (IEEE1588 and IEEE 802.1AS)

7.9.1 Overview

IEEE 1588 addresses the clock synchronization requirements of measurement and control systems. The protocol supports system-wide synchronization accuracy in the sub-microsecond range with minimal network and local clock computing resources. The protocol is spatially localized and allows simple systems to be installed and operated without requiring the administrative attention of users.

The IEEE802.1AS standard specifies the protocol used to ensure that synchronization requirements are met for time sensitive applications, such as audio and video, across Bridged and Virtual Bridged Local Area Networks consisting of LAN media where the transmission delays are fixed and symmetrical; for example, IEEE 802.3 full duplex links. This includes the maintenance of synchronized time during normal operation and following addition, removal, or failure of network components and network reconfiguration. It specifies the use of IEEE 1588 specifications where applicable.

Activation of the I350 Time Sync mechanism is possible in full duplex mode only. No limitations on wire speed exist, although wire speed might affect the accuracy. Time Sync protocol is tolerant of dropping packets as well as missing timestamps.

7.9.2 Flow and Hardware/Software Responsibilities

The operation of a PTP (Precision Time Protocol) enabled network is divided into two stages, initialization and time synchronization.

At the initialization stage every master enabled node starts by sending Sync packets that include the clock parameters of its clock. Upon reception of a Sync packet a node compares the received clock parameters to its own. If the received clock parameters of a peer are better, the node moves to Slave state and stops sending Sync packets. When in slave state the node continuously compares the incoming packet clock parameters to its currently chosen master. If the new clock parameters are better then the current master selection, it changes master clock source. Eventually the best master clock source is chosen. Every node has a defined Sync packet time-out interval. If no Sync packet is received from its chosen master clock source during the interval it moves back to master state and starts sending Sync packets until a new Best Master Clock (BMC) is chosen.

The time synchronization stage is different for master and slave nodes. If a node is in master state it should periodically send a Sync packet which is time stamped by hardware on the transmit path (as close as possible to the PHY). After the Sync packet a Follow_up packet is sent which includes the value of the timestamp kept from the Sync packet. In addition the master should timestamp Delay_Req packets on its RX path and return to the slave that sent it the timestamp value using a Delay_Response packet. A node in Slave state should timestamp every incoming Sync packet that is received from its selected master, software uses this value for time offset calculation. In addition it should periodically send Delay_Req packets in order to calculate the path delay from its master. Every sent Delay_Req packet sent by the slave is time stamped and kept. Using the value received from the master Delay_Response packet the slave can now calculate the path delay from the master to the slave. The synchronization protocol flow and the offset calculation are described in the following figure.

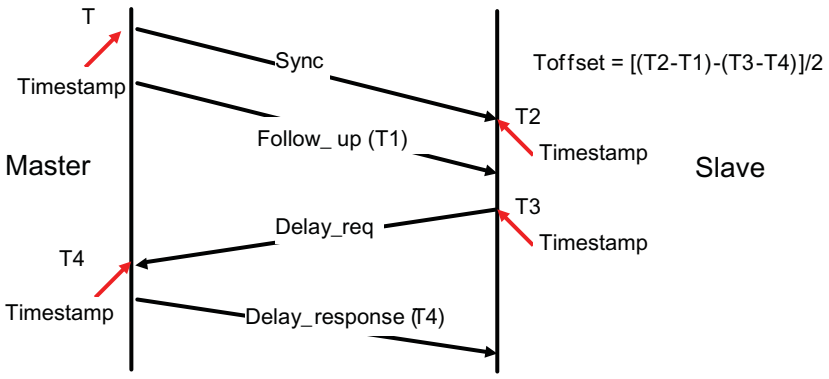


Figure 7-26 Sync Flow and Offset Calculation

The hardware's responsibilities are:

1. Identify the packets that require time stamping.
2. Time stamp the packets on both receive and transmit paths.
3. Store the time stamp value for software.
4. Keep the system time in hardware and give a time adjustment service to the software.

5. Maintain auxiliary features related to the system time.

The software's responsibilities are:

1. Best Master Clock protocol execution, which determines which clock is the highest quality clock within the network. Result of protocol sets the node state (master or slave). If node is slave, software also selects the master clock.
2. Generate PTP (Precision Time Protocol) packets, consume PTP packets.
3. Calculate the time offset and adjust the system time using the hardware mechanism.
4. Enable configuration and usage of the auxiliary features.

Table 7-73 Chronological Order of Events for Sync and Path Delay

Action	Responsibility	Node Role
Generate a Sync packet with timestamp notification in descriptor	Software	Master
Timestamp the packet and store the value in registers (T1)	Hardware	Master
Timestamp incoming Sync packet, store the value in register and store the sourceID and sequenceID in registers (T2)	Hardware	Slave
Read the timestamp from register, prepare a Follow_Up packet and send	Software	Master
Once Follow_Up packet is received, load T2 from registers and T1 from Follow_up packet	Software	Slave
Generate a Delay_Req packet with timestamp notification in descriptor	Software	Slave
Timestamp the packet and store the value in registers (T3)	Hardware	Slave
Timestamp incoming Delay_Req packet, store the value in register and store the sourceID and sequenceID in registers (T4)	Hardware	Master
Read the timestamp from register and send back to Slave using a Delay_Response packet	Software	Master
Once Delay_Response packet is received, calculate offset using T1, T2, T3 and T4 values	Software	Slave

7.9.2.1 TimeSync Indications in Receive and Transmit Packet Descriptors

Certain indications are transferred between software and hardware regarding PTP packets.

On the transmit path the software should set the *1588* bit in the transmit packet descriptor (MAC field bit 1). To indicate that the transmit packet time stamp should be taken and placed in the *TXSTMPH* and *TXSTMPL* time stamp registers.

On the receive path the hardware transfers three indications to software in the receive descriptor:

1. An indication in *RDESC.Packet Type* that this packet is a PTP packet (no matter if timestamp is sampled or not). This indication is used also by PTP packets required for protocol management.

Note: This indication is only relevant for L2 type packets (the PTP packet is identified according to its Ethertype). PTP packets have the *L2Type* bit in the *Packet Type* field set (bit 11) and the Etype matches the filter number set by the software to filter PTP packets. UDP type PTP packets don't require such an indication since the port number (319 for event and 320 for all other PTP packets) directs the packets toward the time sync application.

2. A second indication in the *RDESC.STATUS.TS* bit to indicate to the software that time stamp was taken for this packet and placed in the *RXSTMPH* and *RXSTMPL* time stamp registers. Software needs to access the time stamp registers to get the time stamp values.
3. A third indication in the *RDESC.STATUS.TSIP* bit to indicate that a time stamp was taken for this packet and placed at the start of the receive buffer (For further information see Section 7.1.6).

7.9.3 Hardware Time Sync Elements

All time sync hardware elements are reset to their initial values as defined in the registers section upon MAC reset.

7.9.3.1 System Time Structure and Mode of Operation

The time sync logic contains the SYSTIM counter to maintain the system time value. This is a 72 bit counter that is built of the SYSTIMR, SYSTIML and SYSTIMH registers. Operation of the counter is enabled by clearing the TSAUXC.Disable systime bit. When in Master state the SYSTIMH, SYSTIML and SYSTIMR registers should be set once by the software according to general system requirements in the following manner:

1. Disable SYSTIM timer operation by setting the TSAUXC.Disable systime bit.
2. Program the SYSTIMH, SYSTIML and SYSTIMR registers.
3. Enable SYSTIM timer operation by clearing the TSAUXC.Disable systime bit.

When in slave state software should update the system time on every sync event as described in Section 7.9.3.3. Setting the system time is done by direct write to the SYSTIMH register as described above, enabling SYSTIM operation by clearing the TSAUXC.Disable systime bit and fine tuning the setting of the SYSTIM register, using the adjustment mechanism described in Section 7.9.3.3.

Read access to the SYSTIMH, SYSTIML and SYSTIMR registers should be executed in the following order:

1. Software reads register SYSTIMR. At this stage the hardware latches the value of SYSTIMH and SYSTIML registers.
2. Software reads register SYSTIML.
3. Software reads register SYSTIMH. The latched SYSTIMH and SYSTIML values (from last read of SYSTIMR register) should be returned by hardware when reading the SYSTIML and SYSTIMH registers.

The SYSTIM timer value in the SYSTIMH, SYSTIML and SYSTIMR registers, is updated periodically each 8 nS clock cycle according to the following formula:

$$\text{New SYSTIM} = \text{Old SySTIM} + 8\ \text{nS} +/- \textbf{\textit{TIMINCA}}.\textit{Incvalue} * 2^{-32}\ \text{nS}$$

Where subtraction or addition of the TIMINCA.Incvalue value is defined according to the TIMINCA.ISGN value (0 - Add, 1 - Subtract). For the TIMINCA register description refer to section 8.16.12.

7.9.3.2 Time Stamp Mechanism

The time stamp logic is located on transmit and receive paths at a location as close as possible to the PHY, to reduce delay uncertainties originating from implementation differences. The time stamp logic operation is slightly different on transmit and on receive paths.

When the TSAUXC.Disable systime bit is cleared the transmit logic decides to timestamp a packet if the transmit timestamp is enabled (TSYNCTXCTL.EN = 1) and the time stamp bit in the packet descriptor (TDESD.MAC.1588 = 1) is set. On the transmit side only the time is captured in the TXSTMPL and TXSTMPH registers.

The receive logic parses the received frame and timestamps the receive packet according to the conditions defined in the following fields:

1. TSYNCRXCTL.Type field that defines type of packets to be sampled.
2. TSYNCRXCFG.CTRLT field that defines message type criteria for timestamping V1 type packets when TSYNCRXCTL.Type register field equals 001b.
3. TSYNCRXCFG.MSGT field that defines message type criteria for timestamping V2 type packets when TSYNCRXCTL.Type register field equals 000b or 010b.

When the TSAUXC.Disable systime bit is cleared and above conditions to timestamp a receive packet are met:

1. The timestamp is latched in the RXSTMPL and RXSTMPH registers.
2. The packet's sourceId and sequenceId fields are latched in the RXSATRL and RXSATRH timestamp registers.
3. When the SRRCTL[n].Timestamp bit is set to 1, packets received to the queue will have a timestamp value added to the beginning of the receive buffer (See Section 7.1.6 for additional information).

Three indications are placed in the receive descriptor to support TimeSync operation:

1. RDESC.Packet Type - Value in this field identifies that this is a PTP packet (this indication is only for L2 packets since on the UDP packets the port number directs the packet to the application).
2. RDESC.STATUS.TS - Bit identifies that a time stamp was taken for this packet and latched in RXSTMPL and RXSTMPH registers and the packet's sourceId and sequenceId are latched in the RXSATRL and RXSATRH timestamp registers.
3. RDESC.STATUS.TSIP - Bit identifies that a time stamp was taken for this packet and placed at the start of the receive buffer (For further information see Section 7.1.6).

For more details please refer to the timestamp registers in Section 8.16. Figure 7-27 defines the exact point where the time value is captured.

On both transmit and receive sides the timestamp values are locked in registers until software reads the TXSTMPH register to unlock Transmit timestamp registers or reads the RXSTMPH register to unlock the receive timestamp registers. As a result, if a new PTP packet that needs to be time stamped arrives before software accesses the timestamp registers, it is not time stamped. In some cases on the receive path a packet that was timestamped might be lost and not reach the host. To avoid a deadlock condition on the time stamp registers the software should keep a watch dog timer to clear locking of the time stamp register. The interval counted by such a timer should be higher then the expected interval between two Sync or Delay_Req packets depends on the node state (Master or Slave).

Note: When TSYNCRXCTL.Type value is 100b, the Receive timestamp registers are not locked after a timestamp event.

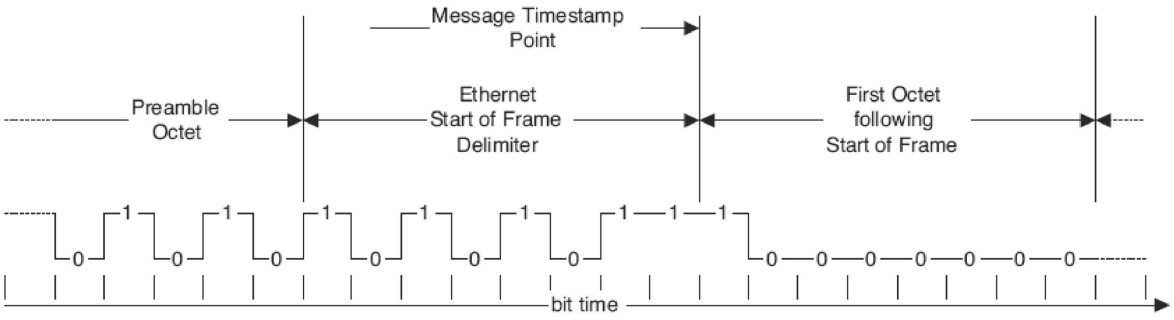


Figure 7-27 Time Stamp Point

7.9.3.3 Time Adjustment Mode of Operation

A Node in a Time Sync network can be in one of two states Master or Slave. When a Time Sync entity is in the Master state it should synchronize other entities to its System Clock. In this case no time adjustments are needed. When the entity is in slave state it should adjust its system clock by using the data arriving in the Follow_Up and Delay_Response packets and the time stamp values of the Sync and Delay_Req packets. When all the values are available the software in the slave entity can calculate its offset in the following manner:

```
Toffset = [(T2-T1) - (T3-T4)]/2

T1 - Timing data in Follow_Up packet
T2 - Sync Time Stamp
T3 - Delay_Req Time Stamp
T4 - Timing data in Delay_Response packet
```

To increase or decrease the system time value located in the SYSTIMH and SYSTIML registers by the calculated Toffset value, software should write the calculated Toffset value to the TIMADJH and TIMADJL registers in the following order:

1. Write the low portion of the Toffset value to the TIMADJL.TADJL field.
2. Write the high portion of the Toffset value to the TIMADJH.TADJH field with the correct sign in the TIMADJH.Sign bit, to indicate if the Toffset value should be added or subtracted.

A write to the TIMADJH register causes the Toffset value in the TIMADJH and TIMADJL registers to be added or subtracted (depending on the sign bit) to the system time registers (SYSTIMH and SYSTIML), resulting in a new system time that equals previous system time + Toffset.

7.9.4 Time Sync Related Auxiliary Elements

The time sync logic implements three types of auxiliary elements using the precise system timer (SYSTIML and SYSTIMH).

7.9.4.1 Target Time

The two target time registers TRGTTIML/H0 and TRGTTIML/H1 enable generating a time triggered event to external hardware using one of the SDP pins according to the setup defined in the TSSDP and TSAUXC registers (See Section 8.16.15 and Section 8.16.27). Each target time register is structured the same as the system time register. If the value of the system time is equal or has passed the value written to one of the target time registers, a change in level or a pulse is generated on the programmed SDP outputs.

7.9.4.1.1 SYSTIM Synchronized Level Change Generation on SDP Pins

To generate a level change on one of the SDP pins when System Time (SYSTIM) reaches a pre-defined value, driver should:

1. Select SDPx pin functionality using the appropriate TSSDP.TS_SDPx_SEL field (where x is 0, 1, 2 or 3).
   - Program 00b to the TSSDP.TS_SDPx_SEL field if level change should occur when SYSTIM equals TRGTTIML/H0.
   - Program 01b to the TSSDP.TS_SDPx_SEL field if level change should occur when SYSTIM equals TRGTTIML/H1.
2. Define selected SDPx pin as output, by setting the appropriate SDPx_IODIR bit (where x is 0,1,2 or 3) in the CTRL or CTRL_EXT registers.
3. To define that level change is generated on selected SDP pin when SYSTIM is equal or greater than Target Time, program TSAUXC.PLSGx bit (where x is 0 or 1) to 0.
4. Program TRGTTIML/Hx (where x is 0 or 1) to define SYSTIM time where level change should occur.
5. Enable level change or pulse generation by setting the TSAUXC.EN_TTx bit (where x is 0 or 1).

Each target time register has an enable bit located in the auxiliary control register (TSAUXC.EN_TTx). When the SDP level has changed (if TSAUXC.PLSGx = 0) on the selected SDP pin, the enable bit is cleared and needs to be set again by software to get another target time event.

7.9.4.1.2 SYSTIM Synchronized Pulse Generation on SDP Pins

To generate a pulse on one of the SDP pins when System Time (SYSTIM) reaches a pre-defined value, driver should:

1. Select SDPx pin functionality using the appropriate TSSDP.TS_SDPx_SEL field (where x is 0, 1, 2 or 3).
   - Program 00b to the TSSDP.TS_SDPx_SEL field if pulse start should occur when SYSTIM equals TRGTTIML/H0.
   - Program 01b to the TSSDP.TS_SDPx_SEL field if pulse start should occur when SYSTIM equals TRGTTIML/H1.
2. Define selected SDPx pin as output, by setting the appropriate SDPx_IODIR bit (where x is 0,1,2 or 3) in the CTRL or CTRL_EXT registers.
3. Program TSAUXC.PLSGx bit (where x is 0 or 1) to 1 to define that pulse is generated on the selected SDP pin, when Target Time is equal or greater than System Time (SYSTIM).
4. Program the TSAUXC.PLSNegx bit to define if generated pulse is positive or negative.
5. Program TRGTTIML/H0 and TRGTTIML/H1 registers to define pulse start time and pulse duration if pulse generation.

— Note that for pulse generation both the TRGTTIML/H0 and TRGTTIML/H1 Target Time registers are used. Depending on the TSSDP.TS_SDPx_SEL field value, one Target Time register is used to define start of pulse and the other is used to define end of pulse.

6. Enable pulse generation by setting both the TSAUXC.EN_TT0 and TSAUXC.EN_TT1 bits to 1.

Each target time register has an enable bit located in the auxiliary control register (*TSAUXC.EN_TTx*). When a pulse is generated on the selected SDP pin, the enable bits are cleared and need to be set again by software to get another target time event.

7.9.4.1.3 Start of Clock Generation on SDP Pins Synchronized to SYSTIM

The I350 supports driving a configurable Clock on the SDP pins. The output clocks generated are synchronized to the global System (SYSTIM) clock. The Target Time registers (TRGTTIML/H0 or TRGTTIML/H1) can be used to trigger toggle of the configurable clock output on a certain system time.

Setting the appropriate TSAUXC.STx (where x is 0 or 1) bit to 1 enables start of clock generation only after Target Time defined in the TRGTTIML/Hx registers is reached (further information can be found in Section 7.9.4.2).

7.9.4.2 Configurable Frequency Clock

This feature enables to generate up to 2 programmable clocks on the appropriate SDP pins by configuring the SDP pins using the TSSDP register and by programming appropriate values to the Frequency out Control registers (FREQOUT0 and FREQOUT1). The output clocks are synchronized to the global System (SYSTIM) clock and are affected by System time corrections programmed in the TIMINCA register and the TIMADJL/H registers.

When clock generation is enabled, the error correction programmed in the TIMADJL/H registers is compensated from the clock output gradually, at a rate of 1 ns per 8 nS internal clock cycle. The gradual compensation is done to avoid large duty cycle variations in the output clock.

Note: Before updating the TIMADJL/H registers, software should verify that the appropriate TSICR.TADJ0/1 register bit was set to indicate that previous one time adjustment has completed.

To generate either Clock 0 or Clock 1 on one of the SDP pins, the following steps should be taken:

1. Program the CHCT field in the relevant FREQOUT0/1 register to define clock half cycle time (See Section 8.16.20 and Section 8.16.21 for additional information).
2. Define SDPx pin functionality to drive clock by programming the appropriate TSSDP.TS_SDPx_SEL field and setting the TSSDP.TS_SDPx_EN bit to 1 (where x is 0,1,2 or 3).
   — Program 10b to the TSSDP.TS_SDPx_SEL field if the FREQOUT0 register is used to define clock frequency.
   — Program 11b to the TSSDP.TS_SDPx_SEL field if the FREQOUT1 register is used to define clock frequency.
3. Define selected SDPx pin as output, by setting the appropriate SDPx_IODIR bit (where x is 0,1,2 or 3) in the CTRL or CTRL_EXT registers.
4. If clock start needs to be aligned to the system time (SYSTIM), program start of clock toggle in the appropriate Target Time (TRGTTIML0/1 and TRGTTIMH0/1) registers and set the relevant TSAUXC.ST0/1 field to 1.
5. To start clock operation, set the relevant *TSAUXC.EN_CLK0/1* bit to 1.

Clock out drives initially a logical 0. Clock value toggles each time a System Time duration of *FREQOUT0*/1 is reached or passed.

Note: Clock output mechanism should be activated only after *SYSTIMH/L* timer is aligned to global system clock and SYSTIM timer error correction entered using the *TIMADJL/H* registers is below 64 μs.

7.9.4.3 Time Stamp Events

Upon a change in the input level of one of the SDP pins that was configured to detect Time stamp events using the TSSDP register, a time stamp of the system time is captured into one of the two auxiliary time stamp registers (*AUXSTMPL/H0* or *AUXSTMPL/H1*).

For example to define timestamping of events in the *AUXSTMPL*0 and *AUXSTMPH0* registers, Software should:

1. Set the TSSDP.AUX0_SDP_SEL field to select the SDP pin that detects the level change and set the TSSDP.AUX0_TS_SDP_EN bit to 1.
2. Set the TSAUXC.EN_TS0 bit to 1 to enable timestamping.

7.9.5 Time SYNC Interrupts

Time Sync related interrupts can be generated by programming the TSICR, TSIM and TSIS registers. The TSICR register logs the interrupt cause, the TSIM register enables masking specific TSICR bits and the TSIS register enables Software generated Time Sync interrupts. Detailed description of the Time Sync interrupt registers can be found in Section 8.16.28. Occurrence of a Time Sync interrupt sets the ICR.Time_Sync interrupt bit.

7.9.6 PTP Packet Structure

The time sync implementation supports both the 1588 V1 and V2 PTP frame formats. The V1 structure can come only as UDP payload over IPv4 while the V2 can come over L2 with its Ethertype or as a UDP payload over IPv4 or IPv6. The 802.1AS uses only the layer 2 V2 format.

Table 7-74 V1 and V2 PTP Message Structure

Offset in Bytes	V1 Fields	V2 Fields	
Bits	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0	
0	***version PTP***	transport Specific[1]	***message Type***
1		Reserved	***version PTP***
2	version Network	message Length	
3			

Table 7-74 V1 and V2 PTP Message Structure (Continued)

Offset in Bytes	V1 Fields	V2 Fields
Bits	7 6 5 4 3 2 1 0	7 6 5 4 3 2 1 0
4	Subdomain	domain Number
5		Reserved
6		flags
7		
8		correction Field
9		
10		
11		
12		
13		
14		
15		
16		reserved
17		
18		
19		
20	message Type	Source Port Identity
21	Source communication technology	
22	Source UUID	
23		
24		
25		
26		
27		
28	source port id	
29		
30	***sequenceId***	***sequenceId***
31		
32	***control***	control
33	reserved	Log Message Interval
34	flags	N/A
35		

1. Should be all zero.

Note: Only the fields with the bold italic format colored red are of interest to the hardware.

Table 7-75 PTP Message Over Layer 2

Ethernet (L2)	VLAN (Optional)	PTP Ethertype	PTP message

Table 7-76 PTP Message Over Layer 4

Ethernet (L2)	IP (L3)	UDP	PTP message

When a PTP packet is recognized (by Ethertype or UDP port address) on the receive side then if the version is V1 then the Control field at offset 32 should be compared to the TSYNCRXCFG.CTRLT message field (see Section 8.16.26) otherwise the byte at offset zero should be used for comparison to the TSYNCRXC*FG*.MSGT field. The rest of the required fields are at the same location and size for both V1 and V2.

Table 7-77 Message Decoding for V1 (Control Field at Offset 32)

Enumeration	Value
PTP_SYNC_MESSAGE	0
PTP_DELAY_REQ_MESSAGE	1
PTP_FOLLOWUP_MESSAGE	2
PTP_DELAY_RESP_MESSAGE	3
PTP_MANAGEMENT_MESSAGE	4
reserved	5–255

Table 7-78 Message Decoding for V2 (Message ID Field at Offset 0)

Message ID	Message Type	Value (hex)
PTP_SYNC_MESSAGE	Event	0
PTP_DELAY_REQ_MESSAGE	Event	1
PTP_PATH_DELAY_REQ_MESSAGE	Event	2
PTP_PATH_DELAY_RESP_MESSAGE	Event	3
Unused	Event	4-7
PTP_FOLLOWUP_MESSAGE	General	8
PTP_DELAY_RESP_MESSAGE	General	9
PTP_PATH_DELAY_FOLLOWUP_MESSAGE	General	A
PTP_ANNOUNCE_MESSAGE	General	B
PTP_SIGNALLING_MESSAGE	General	C
PTP_MANAGEMENT_MESSAGE	General	D
Unused	General	E-F

If V2 mode is configured in the TSYNCRXCTL.Type field (see Section 8.16.1) then the time stamp should be taken on PTP_PATH_DELAY_REQ_MESSAGE and PTP_PATH_DELAY_RESP_MESSAGE according to the value in the *TSYNCRXCFG*.MSGT message field described in Section 8.16.26.

7.10 Statistic Counters

The I350 supports different statistic counters as described in Section 8.18. The statistic counters can be used to create statistic reports as required by different standards. The I350 statistic counters allow support for the following standards:

- IEEE 802.3 clause 30 management – DTE section.
- NDIS 6.0 OID_GEN_STATISTICS.
- RFC 2819 – RMON Ethernet statistics group.
- Linux Kernel (version 2.6) net_device_stats

The following section describes the match between the internal the I350 statistic counters and the counters requested by the different standards.

7.10.1 IEEE 802.3 clause 30 management

The I350 supports the Basic and Mandatory Packages defined in clause 30 of the IEEE 802.3 spec. The following table describes the matching between the internal statistics and the counters requested by these packages.

Table 7-79 IEEE 802.3 Mandatory Package Statistics

Mandatory package capability	I350 counter	Notes and limitations
FramesTransmittedOK	GPTC	The I350 doesn't include flow control packets.
SingleCollisionFrames	SCC	
MultipleCollisionFrames	MCC	
FramesReceivedOK	GPRC	The I350 doesn't include flow control packets.
FrameCheckSequenceErrors	CRCERRS	
AlignmentErrors	ALGNERRC	

In addition, part of the recommended package is also implemented as described in the following table

Table 7-80 IEEE 802.3 Recommended Package Statistics

Recommended package capability	I350 counter	Notes and limitations
OctetsTransmittedOK	GOTCH/GOTCL	The I350 counts also the DA/SA/LT/CRC as part of the octets. The I350 doesn't count Flow control packets.
FramesWithDeferredXmissions	DC	
LateCollisions	LATECOL	
FramesAbortedDueToXSColls	ECOL	
FramesLostDueToIntMACXmitError	HTDMPC	The I350 counts the excessive collisions in this counter, while 802.3 increments no other counters, while this counter is incremented
CarrierSenseErrors	TNCRS	The I350 doesn't count cases of CRS de-assertion in the middle of the packet. However, such cases are not expected when the internal PHY is used.
OctetsReceivedOK	TORL+TORH	The I350 counts also the DA/SA/LT/CRC as part of the octets. Doesn't count Flow control packets.
FramesLostDueToIntMACRcvError	RNBC	
SQETestErrors	N/A	
MACControlFramesTransmitted	N/A	
MACControlFramesReceived	N/A	
UnsupportedOpcodesReceived	FCURC	
PAUSEMACCtrlFramesTransmitted	XONTXC + XOFFTXC	
PAUSEMACCtrlFramesReceived	XONRXC + XOFFRXC	

Part of the optional package is also implemented as described in the following table

Table 7-81 IEEE 802.3 Optional Package Statistics

Optional package capability	I350 counter	Notes
MulticastFramesXmittedOK	MPTC	The I350 doesn't count FC packets
BroadcastFramesXmittedOK	BPTC	
MulticastFramesReceivedOK	MPRC	The I350 doesn't count FC packets
BroadcastFramesReceivedOK	BPRC	
InRangeLengthErrors	LENERRS	
OutOfRangeLengthField	N/A	Packets parsed as Ethernet II packets
FrameTooLongErrors	ROC + RJC	

7.10.2 OID_GEN_STATISTICS

The I350 supports the part of the OID_GEN_STATISTICS as defined by Microsoft* NDIS 6.0 spec. The following table describes the matching between the internal statistics and the counters requested by this structure.

Table 7-82 Microsoft* OID_GEN_STATISTICS

OID entry	I350 counters	Notes
ifInDiscards;	CRCERRS + RLEC + RXERRC + MPC + RNBC + ALGNERRC	
ifInErrors;	CRCERRS + RLEC + RXERRC + ALGNERRC	
ifHCInOctets;	GORCL/GOTCL	
ifHCInUcastPkts;	GPRC - MPRC - BPRC	
ifHCInMulticastPkts;	MPRC	
ifHCInBroadcastPkts;	BPRC	
ifHCOutOctets;	GOTCL/GOTCH	
ifHCOutUcastPkts;	GPTC - MPTC - BPTC	
ifHCOutMulticastPkts;	MPTC	
ifHCOutBroadcastPkts;	BPTC	
ifOutErrors;	ECOL + LATECOL	
ifOutDiscards;	ECOL	
ifHCInUcastOctets;	N/A	
ifHCInMulticastOctets;	N/A	
ifHCInBroadcastOctets;	N/A	
ifHCOutUcastOctets;	N/A	
ifHCOutMulticastOctets;	N/A	
ifHCOutBroadcastOctets;	N/A	

7.10.3 RMON

The I350 supports the part of the RMON Ethernet statistics group as defined by IETF RFC 2819. The following table describes the matching between the internal statistics and the counters requested by this group.

Table 7-83 RMON Statistics

RMON statistic	I350 counters	Notes
etherStatsDropEvents	*MPC + RNBC*	
etherStatsOctets	*TOTL + TOTH*	
etherStatsPkts	TPR	
etherStatsBroadcastPkts	BPRC	
etherStatsMulticastPkts	MPRC	The I350 don't count FC packets
etherStatsCRCAlignErrors	*CRCERRS + ALGNERRC*	
etherStatsUndersizePkts	RUC	
etherStatsOversizePkts	ROC	
etherStatsFragments	RFC	Should count bad aligned fragments as well
etherStatsJabbers	RJC	Should count bad aligned jabbers as well
etherStatsCollisions	COLC	
etherStatsPkts64Octets	PRC64	RMON counts bad packets as well
etherStatsPkts65to127Octets	PRC127	RMON counts bad packets as well
etherStatsPkts128to255Octets	PRC255	RMON counts bad packets as well
etherStatsPkts256to511Octets	PRC511	RMON counts bad packets as well
etherStatsPkts512to1023Octets	PRC1023	RMON counts bad packets as well
etherStatsPkts1024to1518Octets	PRC1522	RMON counts bad packets as well

7.10.4 Linux* net_device_stats

The I350 supports part of the net_device_stats as defined by Linux* Kernel version 2.6 (defined in <linux/netdevice.h>). The following table describes the matching between the internal statistics and the counters requested by this structure./

Table 7-84 Linux net_device_stats

net_device_stats field	I350 counters	Notes
rx_packets	GPRC	The I350 doesn't count flow controls - can be accounted for by using the XONRXC and XOFFRXC counters
tx_packets	GPTC	The I350 doesn't count flow controls - can be accounted for by using the XONTXC and XOFFTXC counters
rx_bytes	*GORCL + GORCH*	
tx_bytes	*GOTCL + GOTCH*	
rx_errors	*CRCERRS + RLEC + RXERRC + ALGNERRC*	
tx_errors	*ECOL + LATECOL*	
rx_dropped	N/A	
tx_dropped	N/A	
multicast	MPTC	
collisions	COLC	

Table 7-84 Linux net_device_stats (Continued)

net_device_stats field	I350 counters	Notes
rx_length_errors	RLEC	
rx_over_errors	SDPC	Used an empty stat to expose the packets dropped due to switch drops.
rx_crc_errors	CRCERRS	
rx_frame_errors	ALGNERRC	
rx_fifo_errors	Sum (RQDPC)	
rx_missed_errors	MPC	
tx_aborted_errors	ECOL	
tx_carrier_errors	N/A	
tx_fifo_errors	N/A	
tx_heartbeat_errors	N/A	
tx_window_errors	LATECOL	
rx_compressed	N/A	
tx_compressed	N/A	

7.10.5 Statistics Hierarchy.

The following diagram describes the relations between the packet flow and the different statistic counters.

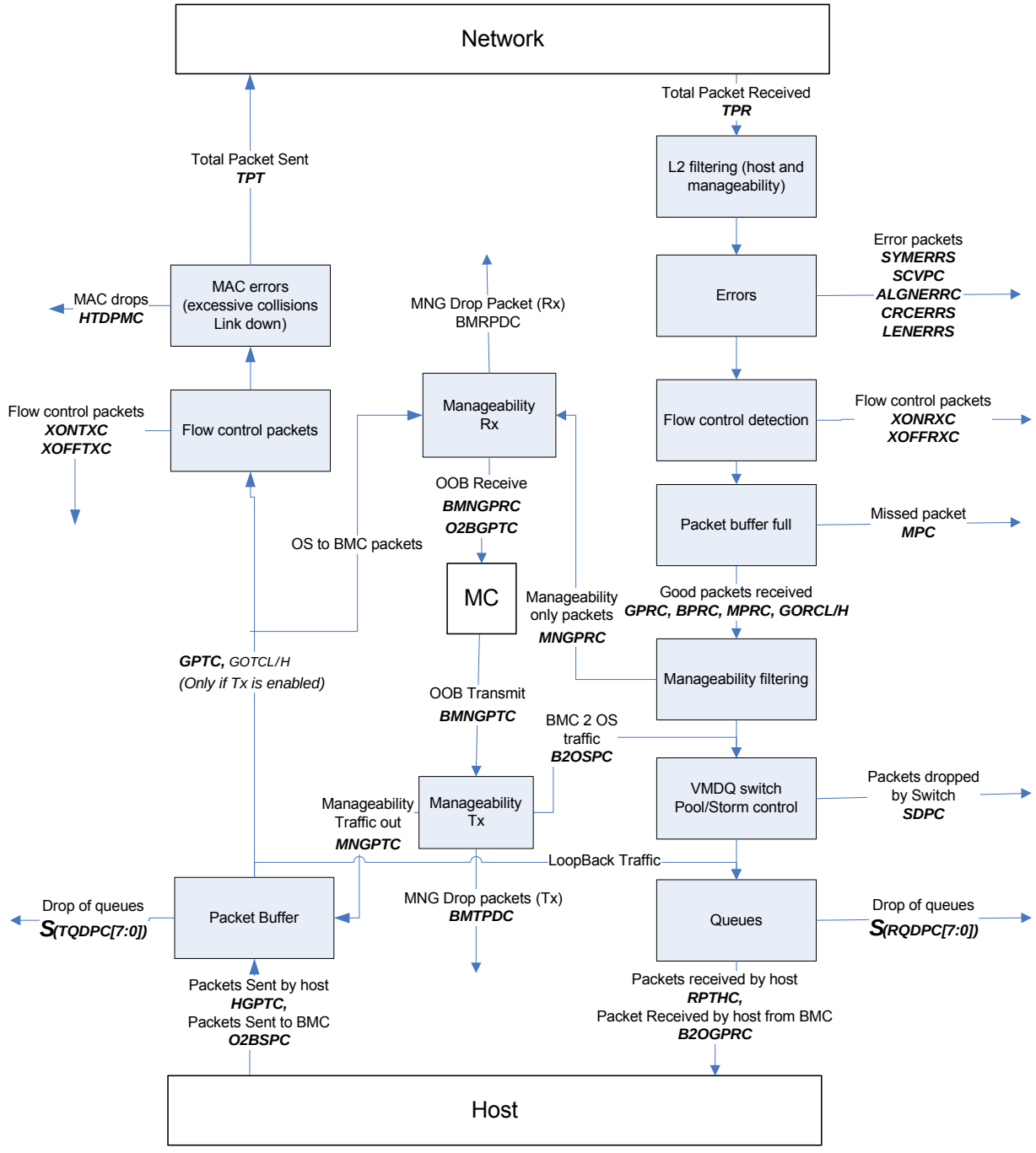


Figure 7-30 Flow Statistics

§ §

8 Programming Interface

8.1 Introduction

This chapter details the programmer visible state inside the I350. In some cases, it describes hardware structures invisible to software in order to clarify a concept. The I350's address space is mapped into four regions with PCI Base Address Registers described in Section 9.4.11. These regions are listed in the table below.

Table 8-1 Address Space Regions

Addressable Content	How Mapped	Size of Region
Internal registers, memories and FLASH ("Memory BAR")	Direct memory-mapped	128K + FLASH Size [1]
Flash (optional)	Direct memory-mapped	64K-8M
Expansion ROM (optional)	Direct memory-mapped	64K-8M[1]
Internal registers and memories, Flash (optional)	I/O Window mapped	32 bytes[2]
MSI-X (optional)	Direct memory-mapped	16K

1. The FLASH space in the "Memory CSR" and Expansion ROM Base Address map the same FLASH memory. Accessing the "memory BAR" at offset 128K and Expansion ROM at offset 0x0 are mapped to the FLASH device at offset 0x0.
2. The internal registers and memories can be accessed though I/O space indirectly as explained below.

The internal register/memory space is described in the following sections. The PHY registers are accessed through the MDIO interface.

8.1.1 Memory, I/O Address and Configuration Decoding

8.1.1.1 Memory-Mapped Access to Internal Registers and Memories

The internal registers and memories might be accessed as direct memory-mapped offsets from the base address register (BAR0 or BAR 0/1; refer to Section 9.4.11). Refer to Section 8.1.3 for the appropriate offset for each specific internal register.

In IOV mode, this area is partially duplicated per VF. All replications contain only the subset of the register set that is available for VF programming.

8.1.1.2 Memory-Mapped Access to Flash

The external Flash may be accessed using direct memory-mapped offsets from the Memory base address register (BAR0 in 32bit addressing or BAR0/BAR1 in 64 bit addressing; refer to Section 9.4.11). For accesses, the offset from the Memory BAR minus 128KB corresponds to the physical address within the external Flash device. Memory mapped accesses to the external Flash are enabled when the value of the *Flash Size* field in the EEPROM (refer to Section 6.2.26) is not 000b.

8.1.1.3 Memory-Mapped Access to MSI-X Tables

The MSI-X tables may be accessed as direct memory-mapped offsets from the base address register (BAR3; refer to Section 9.4.11). Refer to Section 8.1.3 for the appropriate offset for each specific internal MSIX register.

In IOV mode, this area is duplicated per VF.

8.1.1.4 Memory-Mapped Access to Expansion ROM

The external Flash might also be accessed as a memory-mapped expansion ROM. Accesses to offsets starting from the Expansion ROM Base address (refer to Section 9.4.11) reference the Flash provided that access is enabled thorough the *LAN Boot Disable* bit in the *Initialization Control 3* EEPROM word, and if the Expansion ROM Base Address register contains a valid (non-zero) base memory address.

8.1.1.5 I/O-Mapped Access to Internal Registers and Memories

To support pre-boot operation (prior to the allocation of physical memory base addresses), all internal registers and memories can be accessed using I/O operations. I/O accesses are supported only if an I/O Base Address is allocated and mapped (BAR2; refer to Section 9.4.11), the BAR contains a valid (non-zero value), and I/O address decoding is enabled in the PCIe configuration.

When an I/O BAR is mapped, the I/O address range allocated opens a 32-byte "window" in the system I/O address map. Within this window, two I/O addressable registers are implemented: IOADDR and IODATA. The IOADDR register is used to specify a reference to an internal register or memory, and then the IODATA register is used as a "window" to the register or memory address specified by IOADDR:

Table 8-2 IOADDR and IODATA in I/O Address Space

Offset	Abbreviation	Name	RW	Size
0x00	IOADDR	Internal Register or Internal Memory location address. 0x00000-0x1FFFF – Internal Registers and Memories 0x20000-0xFFFFFFFF – Undefined	RW	4 bytes
0x04	IODATA	Data field for reads or writes to the Internal Register or Internal Memory Location as identified by the current value in IOADDR. All 32 bits of this register are read/write-able.	RW	4 bytes
0x08 – 0x1F	Reserved	Reserved	RO	4 bytes

8.1.1.5.1 IOADDR (I/O offset 0x00)

The IOADDR register must always be written as a DWORD access. Writes that are less than 32 bits are ignored. Reads of any size return a DWORD of data; however, the chipset or CPU might only return a subset of that DWORD.

For software programmers, the IN and OUT instructions must be used to cause I/O cycles to be used on the PCIe bus. Because writes must be to a 32-bit quantity, the source register of the OUT instruction must be EAX (the only 32-bit register supported by the OUT command). For reads, the IN instruction can have any size target register, but it is recommended that the 32-bit EAX register be used.

Because only a particular range is addressable, the upper bits of this register are hard coded to zero. Bits 31 through 20 are not write-able and always read back as 0b.

At hardware reset (LAN_PWR_GOOD) or PCI Reset, this register value resets to 0x00000000. Once written, the value is retained until the next write or reset.

8.1.1.5.2 IODATA (I/O offset 0x04)

The IODATA register must always be written as a DWORD access when the IOADDR register contains a value for the Internal Register and Memories (for example, 0x00000-0x1FFFC). In this case, writes that are less than 32 bits are ignored.

Reads to IODATA of any size returns a DWORD of data. However, the chipset or CPU might only return a subset of that DWORD.

For software programmers, the IN and OUT instructions must be used to cause I/O cycles to be used on the PCIe bus. Where 32-bit quantities are required on writes, the source register of the OUT instruction must be EAX (the only 32-bit register supported by the OUT command).

Writes and reads to IODATA when the IOADDR register value is in an undefined range (0x20000-0xFFFFFFFC) should not be performed. Results cannot be determined.

Notes: There are no special software timing requirements on accesses to IOADDR or IODATA. All accesses are immediate, except when data is not readily available or acceptable. In this case, the I350 delays the results through normal bus methods (for example, split transaction or transaction retry).

Because a register/memory read or write takes two IO cycles to complete, software must provide a guarantee that the two IO cycles occur as an atomic operation. Otherwise, results can be non-deterministic from the software viewpoint.

Software should access CSRs via IO address space or Configuration address space but should not use both mechanisms at the same time.

8.1.1.5.3 Undefined I/O Offsets

I/O offsets 0x08 through 0x1F are considered to be reserved offsets with the I/O window. Dword reads from these addresses returns 0xFFFF; writes to these addresses are discarded.

8.1.1.6 Configuration Access to Internal Registers and Memories

To support 'legacy' pre-boot 16-bit operating environments without requiring IO address space, the I350 enables accessing CSRs via configuration address space by mapping the *IOADDR* and *IODATA* registers into configuration address space. The registers mapping in this case is shown in Table 8-3.

Table 8-3 IOADDR and IODATA in Configuration Address space

Configuration Address	Abbreviation	Name	RW	Size
0x98	IOADDR	Internal register or internal memory location address. 0x00000-0x1FFFF – Internal Registers and Memories 0x20000-0x7FFFFF – Undefined	RW	4 bytes
0x9C	IODATA	Data field for reads or writes to the internal register or internal memory location as identified by the current value in IOADDR. All 32 bits of this register are read/write-able.	RW	4 bytes

Software writes data to an internal CSR via Configuration space in the following manner:

1. CSR address is written to the IOADDR register where:
   a. Bit 31 (IOADDR.Configuration IO Access Enable) of the IOADDR register should be set to 1.
   b. Bits 30:0 of IOADDR should hold the actual address of the internal register or memory being written to.
2. Data to be written is written into the IODATA register.
   — The IODATA register is used as a "window" to the register or memory address specified by IOADDR register. As a result the data written to the IODATA register is written into the CSR pointed to by bits 30:0 of the IOADDR register.
3. IOADDR.Configuration IO Access Enable is cleared, to avoid un-intentional CSR read operations (that may cause clear by read) by other applications scanning the configuration space.

Software reads data from an internal CSR via Configuration space in the following manner:

1. CSR address is written to the IOADDR register where:
   a. Bit 31 (IOADDR.Configuration IO Access Enable) of the IOADDR register should be set to 1.
   b. Bits 30:0 of IOADDR should hold the actual address of the internal register or memory being read.
2. CSR value is read from the IODATA register.
   a. The IODATA register is used as a "window" to the register or memory address specified by IOADDR register. As a result the data read from the IODATA register is the data of the CSR pointed to by bits 30:0 of the IOADDR register
3. IOADDR.Configuration IO Access Enable is cleared, to avoid un-intentional CSR read operations (that may cause clear by read) by other applications scanning the configuration space.

Notes:

- In the event that the CSR_conf_en bit in the PCIe Init Configuration 2 EEPROM word is cleared, accesses to the IOADDR and IODATA registers via the configuration address space are ignored and have no effect on the register and the CSRs referenced by the IOADDR register. In this case any read access to these registers returns a value of 0.
- When Function is in D3 state Software should not attempt to access CSRs via the IOADDR and IODATA Configuration registers.
- To enable CSR access via configuration space, Software should set to 1 bit 31 (IOADDR.Configuration IO Access Enable) of the IOADDR register. Software should clear bit 31 of the IOADDR register after completing CSR access to avoid an unintentional "clear by read" operation, by another application scanning the configuration address space.
- Bit 31 of the *IOADDR* register (*IOADDR.Configuration IO Access Enable*) has no effect when initiating access via IO Address space.

- Software should access CSRs via IO address space or Configuration address space but should not use both mechanisms at the same time.

8.1.2 Register Conventions

All registers in the I350 are defined to be 32 bits, should be accessed as 32 bit double-words, There are some exceptions to this rule:

- Register pairs where two 32 bit registers make up a larger logical size
- Accesses to Flash memory (via Expansion ROM space, secondary BAR space, or the I/O space) might be byte, word or double word accesses.

Reserved bit positions: Some registers contain certain bits that are marked as "reserved". These bits should never be set to a value of "one" by software. Reads from registers containing reserved bits might return indeterminate values in the reserved bit-positions unless read values are explicitly stated. When read, these reserved bits should be ignored by software.

Reserved and/or undefined addresses: any register address not explicitly declared in this specification should be considered to be reserved, and should not be written to. Writing to reserved or undefined register addresses might cause indeterminate behavior. Reads from reserved or undefined configuration register addresses might return indeterminate values unless read values are explicitly stated for specific addresses.

Initial values: most registers define the initial hardware values prior to being programmed. In some cases, hardware initial values are undefined and is listed as such via the text "undefined", "unknown", or "X". Such configuration values might need to be set via EEPROM configuration or via software in order for proper operation to occur; this need is dependent on the function of the bit. Other registers might cite a hardware default which is overridden by a higher-precedence operation. Operations which might supersede hardware defaults might include a valid EEPROM load, completion of a hardware operation (such as hardware auto-negotiation), or writing of a different register whose value is then reflected in another bit.

For registers that should be accessed as 32 bit double words, partial writes (less than a 32 bit double word) do not take effect (the write is ignored). Partial reads returns all 32 bits of data regardless of the byte enables.

Note: Partial reads to read-on-clear registers (*ICR*) can have unexpected results since all 32 bits are actually read regardless of the byte enables. Partial reads should not be done.

All statistics registers are implemented as 32 bit registers. Though some logical statistics registers represent counters in excess of 32-bits in width, registers must be accessed using 32-bit operations (for example, independent access to each 32-bit field). When reading 64 bits statistics registers the least significant 32 bit register should be read first.

See special notes for VLAN Filter Table, Multicast Table Arrays and Packet Buffer Memory which appear in the specific register definitions.

The I350 register fields are assigned one of the attributes described in Table 8-4.

Table 8-4 I350 Register Field Attributes

Attribute	Description
RW	Read-Write field: Register bits are read-write and can be either set or cleared by software to the desired state.
RWS	Read-Write Status field: Register bits are read-write and can be either set or cleared by software to the desired state. However, the value of this field might be changed by the hardware to reflect a status change.
RO	Read-only register: Register bits are read-only and should not be altered by software. Register bits might be initialized by hardware mechanisms such as pin strapping, serial EEPROM or reflect a status of the hardware state.
R/W1C	Read-only status, Write-1-to-clear status register: Register bits indicate status when read, a set bit indicating a status event can be cleared by writing a 1b. Writing a 0b to R/W1C bit has no effect.
Rsv	Reserved. Write 0 to these fields and ignore read.
RC	Read-only status, Read-to-clear status register: Register bits indicate status when read, a set bit indicating a status event is cleared by reading it.
SC	Self Clear field: a command field that is self clearing. These field are always read as zero.
WO	Write only field: a command field that can not be read, These field read values are undefined.
RC/W	Read-Write status, Read-to-clear status register: Read-to-clear status register. Register bits indicate status when read. Register bits are read-write and can be either set or cleared by software to the desired state.
RC/W1C	Read-only status, Write-1-to-clear status register: Read-to-clear status register. Register bits indicate status when read, a set bit indicating a status event can be cleared by writing a 1b or by reading the register. Writing a 0b to RC/W1C bit has no effect.
RS	Read Set – This is the attribute used for Semaphore bits. These bits are set by read in case the previous values were zero. In this case the read value is zero; otherwise the read value is one. Cleared by write zero.

PHY registers described in Section 8.26 use a special nomenclature to define the read/write mode of individual bits in each register. See table bellow for details.

Table 8-5 PHY Register Nomenclature

Register Mode	Description
LH	Latched High. Event is latched and erased when read.
LL	Latched Low. Event is latched and erased when read. For example, Link Loss is latched when the PHY Control Register bit 2 = 0b. After read, if the link is good, the PHY Control Register bit 2 is set to 1b.
RO	Read Only.
R/W	Read and Write.
SC	Self-Clear. The bit is set, automatically executed, and then reset to normal operation.
CR	Clear after Read. For example, 1000BASE-T Status Register bits 7:0 (Idle Error Counter).
Update	Value written to the register bit does not take effect until software PHY reset is executed.

Note: For all binary equations appearing in the register map, the symbol "|" is equivalent to a binary OR operation.

8.1.2.1 Registers Byte Ordering

This section defines the structure of registers that contain fields carried over the network. Some examples are L2, L3 and L4 fields.

The following example is used to describe byte ordering over the wire (hex notation):

```
Last                    First
...,06, 05, 04, 03, 02, 01, 00
```

Each byte is sent with the LSbit first. That is, the bit order over the wire for this example is

```
Last                                  First
..., 0000 0011, 0000 0010, 0000 0001, 0000 0000
```

The general rule for register ordering is to use Host Ordering (also called little endian). Using the above example, a 6-byte fields (MAC address) is stored in a CSR in the following manner:

```
                  Byte 3 Byte 2 Byte 1 Byte0
DW address (N)    0x03   0x02   0x01   0x00
DW address (N+4) ...     ...    0x05   0x04
```

The exceptions listed below use network ordering (also called big endian). Using the above example, a 16-bit field (EtherType) is stored in a CSR in the following manner:

```
                  Byte 3 Byte 2 Byte 1 Byte0
(DW aligned)      ...    ...    0x00   0x01
or
(Word aligned)    0x00   0x01   ...    ...
```

The following exceptions use network ordering:

All ETherType fields. For example, the VET EXT field in the VET register, the EType field in the ETQF register, the EType field in the METF register.

Note: The “normal” notation as it appears in text books, etc. is to use network ordering. Example: Suppose a MAC address of 00-A0-C9-00-00-00. The order on the network is 00, then A0, then C9, etc. However, the host ordering presentation would be:

```
                  Byte 3 Byte 2 Byte 1 Byte0
DW address (N)    00     C9     A0     00
DW address (N+4) ...     ...    00     00
```

8.1.3 Register Summary

All the I350's non-PCIe configuration registers, except for the MSI-X register, are listed in the table below. These registers are ordered by grouping and are not necessarily listed in order that they appear in the address space. In an IOV system, this list refers to the PF registers, the VF register space is listed in Section 8.27.

Table 8-6 Register Summary

Offset	Alias Offset	Abbreviation	Name	RW
General				
0x0000	0x0004	CTRL	Device Control Register	RW
0x0008	N/A	STATUS	Device Status Register	RO
0x0018	N/A	CTRL_EXT	Extended Device Control Register	RW
0x0020	N/A	MDIC	MDI Control Register	RW
0x0E08	N/A	P1GCTRL0	Serdes Control 0	RW

Table 8-6 Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0x0028	N/A	FCAL	Flow Control Address Low	RO
0x002C	N/A	FCAH	Flow Control Address High	RO
0x0030	N/A	FCT	Flow Control Type	RW
0x0034	N/A	CONNSW	Copper/Fiber switch control	RW
0x0038	N/A	VET	VLAN Ether Type	RW
0x0E04	N/A	MDICNFG	MDC/MDIO Configuration Register	RW
0x0170	N/A	FCTTV	Flow Control Transmit Timer Value	RW
0x0E00	N/A	LEDCTL	LED Control Register	RW
0x1028	N/A	I2CCMD	SFP I2C command	RW
0x102C	N/A	I2CPARAMS	SFP I2C Parameter	RW
0x1040	N/A	WDSTP	Watchdog setup register	RW
0x1044	N/A	WDSWSTS	Watchdog Software	RW
0x1048	N/A	FRTIMER	Free Running Timer	RWS
0x104C	N/A	TCPTimer	TCP timer	RW
0x5B70	N/A	DCA_ID	DCA Requester ID Information Register	RO
0x5B50	N/A	SWSM	Software Semaphore Register	RW
0x5B54	N/A	FWSM	Firmware Semaphore Register	RWS
0x5B5C	N/A	SW_FW_SYNC	Software-Firmware Synchronization	RWS
0x5B04	N/A	SWMBWR	Software Mailbox Write	RW
0x5B08	N/A	SWMB0	Software Mailbox Port 0	RO
0x5B0C	N/A	SWMB1	Software Mailbox Port 1	RO
0x5B18	N/A	SWMB2	Software Mailbox Port 2	RO
0x5B1C	N/A	SWMB3	Software Mailbox Port 3	RO
0x0E38	N/A	IPCNFG	Internal PHY Configuration	RW
0x0E14	N/A	PHPM	PHY Power Management	RW
Flash/EEPROM Registers				
0x0010	N/A	EEC	EEPROM/Flash Control Register	RW
0x0014	N/A	EERD	EEPROM Read Register	RW
0x001C	N/A	FLA	Flash Access Register	RW
0x1010	N/A	EEMNGCTL	MNG EEPROM Control Register	RW
0x1014	N/A	EEMNGDATA	MNG EEPROM Read/Write data	RW
0x1024	N/A	EEARBC	EEPROM Auto Read Bus Control	RW
0x103C	N/A	FLASHOP	Flash Opcode Register	RW
Interrupts				
0x1500	0x00C0	ICR	Interrupt Cause Read	RC/W1C
0x1504	0x00C8	ICS	Interrupt Cause Set	WO
0x1508	0x00D0	IMS	Interrupt Mask Set/Read	RW
0x150C	0x00D8	IMC	Interrupt Mask Clear	WO
0x1510	0x00E0	IAM	Interrupt Acknowledge Auto Mask	RW
0x1520	N/A	EICS	Extended Interrupt Cause Set	WO
0x1524	N/A	EIMS	Extended Interrupt Mask Set/Read	RWS
0x1528	N/A	EIMC	Extended Interrupt Mask Clear	WO
0x152C	N/A	EIAC	Extended Interrupt Auto Clear	RW

Table 8-6 Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0x1530	N/A	EIAM	Extended Interrupt Auto Mask	RW
0x1580	N/A	EICR	Extended Interrupt Cause Read	RC/W1C
0x1700 - 0x170C	N/A	IVAR	Interrupt Vector Allocation Registers	RW
0x1740	N/A	IVAR_MISC	Interrupt Vector Allocation Registers - MISC	RW
0x1680 - 0x16A0	N/A	EITR	Extended Interrupt Throttling Rate 0 - 24	RW
0x1514	N/A	GPIE	General Purpose Interrupt Enable	RW
0x5B68	N/A	PBACL	MSI-X PBA Clear	R/W1C
Receive				
0x0100	N/A	RCTL	RX Control	RW
0x2160	0x0168	FCRTL0	Flow Control Receive Threshold Low	RW
0x2168	0x0160	FCRTH0	Flow Control Receive Threshold High	RW
0x2404	N/A	IRPBS	Internal Receive Packet Buffer Size	RO
0x2460	N/A	FCRTV	Flow control Refresh timer value	RW
0xC000	0x0110, 0x2800	RDBAL[0]	RX Descriptor Base Low queue 0	RW
0xC004	0x0114, 0x2804	RDBAH[0]	RX Descriptor Base High queue 0	RW
0xC008	0x0118, 0x2808	RDLEN[0]	RX Descriptor Ring Length queue 0	RW
0xC00C	0x280C	SRRCTL[0]	Split and Replication Receive Control Register queue 0	RW
0xC010	0x0120, 0x2810	RDH[0]	RX Descriptor Head queue 0	RO
0xC018	0x0128, 0x2818	RDT[0]	RX Descriptor Tail queue 0	RW
0xC028	0x02828	RXDCTL[0]	Receive Descriptor Control queue 0	RW
0xC014	0x2814	RXCTL[0]	Receive Queue 0 DCA CTRL Register	RW
0xC040 + 0x40 * (n-1)	0x2900+ 0x100 * (n-1)	RDBAL[1 - 3]	RX Descriptor Base Low queue 1 - 3	RW
0xC044 + 0x40 * (n-1)	0x2904 + 0x100 * (n-1)	RDBAH[1 - 3]	RX Descriptor Base High queue 1 - 3	RW
0xC048 + 0x40 * (n-1)	0x2908 + 0x100 * (n-1)	RDLEN[1 - 3]	RX Descriptor Ring Length queue 1 - 3	RW
0xC04C + 0x40 * (n-1)	0x290C + 0x100 * (n-1)	SRRCTL[1 - 3]	Split and Replication Receive Control Register queue 1 - 3	RW
0xC050 + 0x40 * (n-1)	0x2910 + 0x100 * (n-1)	RDH[1 - 3]	RX Descriptor Head queue 1 - 3	RO
0xC058 + 0x40 * (n-1)	0x2918 + 0x100 * (n-1)	RDT[1 - 3]	RX Descriptor Tail queue 1 - 3	RW
0xC068 + 0x40 * (n-1)	0x2928 + 0x100 * (n-1)	RXDCTL[1 - 3]	Receive Descriptor Control queue 1 - 3	RW
0xC054 + 0x40 * (n-1)	0x2914 + 0x100 * (n-1)	RXCTL[1 - 3]	Receive Queue 1 - 3 DCA CTRL Register	RW

Table 8-6 Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0xC100 + 0x40 * (n- 4)	N/A	RDBAL[4-7]	RX Descriptor Base Low queue 4 - 7	RW
0xC104 + 0x40 * (n- 4)	N/A	RDBAH[4-7]	RX Descriptor Base High queue 4 - 7	RW
0xC108 + 0x40 * (n- 4)	N/A	RDLEN[4-7]	RX Descriptor Ring Length queue 4 - 7	RW
0xC10C + 0x40 * (n- 4)	N/A	SRRCTL[4 -7]	Split and Replication Receive Control Register queue 4 - 7	RW
0xC110 + 0x40 * (n- 4)	N/A	RDH[4 - 7]	RX Descriptor Head queue 4 - 7	RO
0xC118 + 0x40 * (n- 4)	N/A	RDT[4 - 7]	RX Descriptor Tail queue 4 - 7	RW
0xC128 + 0x40 * (n- 4)	N/A	RXDCTL[4 - 7]	Receive Descriptor Control queue 4 - 7	RW
0xC114 + 0x40 * (n- 4)	N/A	RXCTL[4 - 7]	Receive Queue 4 - 7 DCA CTRL Register	RW
0x5000	N/A	RXCSUM	Receive Checksum Control	RW
0x5004	N/A	RLPML	Receive Long packet maximal length	RW
0x5008	N/A	RFCTL	Receive Filter Control Register	RW
0x5200- 0x53FC	0x0200- 0x03FC	MTA[127:0]	Multicast Table Array (n)	RW
0x5400 + 8*n	0x0040 + 8*n	RAL[0-15]	Receive Address Low (15:0)	RW
0x5404 + 8 *n	0x0044 + 8 *n	RAH[0-15]	Receive Address High (15:0)	RW
0x54E0 + 8*n	N/A	RAL[16-31]	Receive Address Low (31:16)	RW
0x54E4 + 8 *n	N/A	RAH[16-31]	Receive Address High (31:16)	RW
0x5480 – 0x549C	N/A	PSRTYPE[7:0]	Packet Split Receive type (n)	RW
0x54C0	N/A	RPLPSRTYPE	Replicated Packet Split Receive type	RW
0x581C	N/A	VT_CTL	VMDq Control register	RW
0x5600-0x57FC	0x0600- 0x07FC	VFTA[127:0]	VLAN Filter Table Array (n)	RW
0x5818	N/A	MRQC	Multiple Receive Queues Command	RW
0x5C00-0x5C7C	N/A	RETA	Redirection Table	RW
0x5C80-0x5CA4	N/A	RSSRK	RSS Random Key Register	RW
Transmit				
0x0400	N/A	TCTL	TX Control	RW
0x0404	N/A	TCTL_EXT	TX Control extended	RW
0x0410	N/A	TIPG	TX IPG	RW
0x041C	N/A	RETX_CTL	Retry Buffer Control	RW
0x3404	N/A	ITPBS	Internal Transmit Packet Buffer Size	RO
0x359C	N/A	DTXTCPFLGL	DMA TX TCP Flags Control Low	RW
0x35A0	N/A	DTXTCPFLGH	DMA TX TCP Flags Control High	RW
0x3540	N/A	DTXMXSZRQ	DMA TX Max Total Allow Size Requests	RW
0x355C	N/A	DTXMXPKTSZ	DMA TX Max Allowable packet size	RW
0x3590	N/A	DTXCTL	DMA TX Control	RW
0x35A4	N/A	DTXBCTL	DMA TX Behavior Control	RW

Table 8-6 Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0xE000	0x0420, 0x3800	TDBAL[0]	TX Descriptor Base Low 0	RW
0xE004	0x0424, 0x3804	TDBAH[0]	TX Descriptor Base High 0	RW
0xE008	0x0428, 0x3808	TDLEN[0]	TX Descriptor Ring Length 0	RW
0xE010	0x0430, 0x3810	TDH[0]	TX Descriptor Head 0	RO
0xE018	0x0438, 0x3818	TDT[0]	TX Descriptor Tail 0	RW
0xE028	0x3828	TXDCTL[0]	Transmit Descriptor Control queue 0	RW
0xE014	0x3814	TXCTL[0]	TX DCA CTRL Register Queue 0	RW
0xE038	0x3838	TDWBAL[0]	Transmit Descriptor WB Address Low queue 0	RW
0xE03C	0x383C	TDWBAH[0]	Transmit Descriptor WB Address High queue 0	RW
0xE040 + 0x40 * (n-1)	0x3900 + 0x100 * (n-1)	TDBAL[1-3]	TX Descriptor Base Low queue 1 - 3	RW
0xE044 + 0x40 * (n-1)	0x3904 + 0x100 * (n-1)	TDBAH[1-3]	TX Descriptor Base High queue 1 - 3	RW
0xE048 + 0x40 * (n-1)	0x3908 + 0x100 * (n-1)	TDLEN[1-3]	TX Descriptor Ring Length queue 1 - 3	RW
0xE050 + 0x40 * (n-1)	0x3910 + 0x100 * (n-1)	TDH[1-3]	TX Descriptor Head queue 1 - 3	RO
0xE058 + 0x40 * (n-1)	0x3918 + 0x100 * (n-1)	TDT[1-3]	TX Descriptor Tail queue 1 - 3	RW
0xE068 + 0x40 * (n-1)	0x3928 + 0x100 * (n-1)	TXDCTL[1-3]	Transmit Descriptor Control 1 - 3	RW
0xE054 + 0x40 * (n-1)	0x3914 + 0x100 * (n-1)	TXCTL[1-3]	TX DCA CTRL Register Queue 1 - 3	RW
0xE078 + 0x40 * (n-1)	0x3938 + 0x100 * (n-1)	TDWBAL[1-3]	Transmit Descriptor WB Address Low queue 1 - 3	RW
0xE07C + 0x40 * (n-1)	0x393C + 0x100 * (n-1)	TDWBAH[1-3]	Transmit Descriptor WB Address High queue 1 - 3	RW
0xE180 + 0x40 * (n - 4)	N/A	TDBAL[4 - 7]	TX Descriptor Base Low queue 4 - 7	RW
0xE184 + 0x40 * (n - 4)	N/A	TDBAH[4 - 7]	TX Descriptor Base High queue 4 - 7	RW
0xE188 + 0x40 * (n - 4)	N/A	TDLEN[4 - 7]	TX Descriptor Ring Length queue 4 - 7	RW
0xE190 + 0x40 * (n - 4)	N/A	TDH[4 - 7]	TX Descriptor Head queue 4 - 7	RO
0xE198 + 0x40 * (n - 4)	N/A	TDT[4 - 7]	TX Descriptor Tail queue 4 - 7	RW
0xE1A8 + 0x40 * (n - 4)	N/A	TXDCTL[4 - 7]	Transmit Descriptor Control 4 - 7	RW
0xE194 + 0x40 * (n - 4)	N/A	TXCTL[4 - 7]	TX Queue 4 - 7DCA CTRL Register	RW

Table 8-6 Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0xE1B8 + 0x40 * (n - 4)	N/A	TDWBAL[4 - 7]	Transmit Descriptor WB Address Low queue 4 - 7	RW
0xE1BC + 0x40 * (n - 4)	N/A	TDWBAH[4 - 7]	Transmit Descriptor WB Address High queue 4 - 7	RW
Filters				
0x5CB0 + 4*n	N/A	ETQF[0 - 7]	EType Queue Filter 0 - 7	RW
0x5A80 + 4*n	N/A	IMIR[0 - 7]	Immediate Interrupt Rx 0 - 7	RW
0x5AA0 + 4*n	N/A	IMIREXT[0 - 7]	Immediate Interrupt Rx Extended 0 - 7	RW
0x5AC0	N/A	IMIRVP	Immediate Interrupt Rx VLAN Priority	RW
0x59E0 + 4*n	N/A	TTQF[0 - 7]	Two-Tuple Queue Filter 0 - 7	RW
0x55FC	N/A	SYNQF	SYN Packet Queue Filter	RW
Virtualization				
0x0C40 - 0x0C5C	N/A	VFMailbox[0 - 7]	VF mailbox register	RW
0x0C00 - 0x0C1C	N/A	PFMailbox[0 - 7]	PF Mailbox register	RW
0x0800 - 0x09FC	N/A	VMBMEM	Virtual machines Mailbox Memory	RW
0x0C80	N/A	MBVFICR	Mailbox VF interrupt causes	R/W1C
0x0C84	N/A	MBVFIMR	Mailbox VF interrupt mask	RW
0x0C88	N/A	VFLRE	VFLR Events	R/W1C
0x0C8C	N/A	VFRE	VF Receive Enable	RW
0x0C90	N/A	VFTE	VF Transmit Enable	RW
0x3554	N/A	WVBR	Wrong VM Behavior Register	RC/W1C
0x3510	N/A	VMECM	VM Error count mask	RW
0x3548	N/A	LVMMC	Last VM Misbehavior Cause	RC
0x3558	N/A	MDFB	Malicious Driver Free Block	RWS
0x2408	N/A	QDE	Queue drop enable register	RW
0x5ACC	N/A	TXSWC	TX Switch Control	RW
0x5D00 - 0x5D7C	N/A	VLVF	VLAN VM Filter	RW
0x5AD0 - 0x5ADC	N/A	VMOLR[0 - 7]	VM Offload register[0-7]	RW
0xC038 + 0x40*n	N/A	DVMOLR[0 - 7]	DMA VM Offload register[0-7]	RW
0x3700	N/A	VMVIR	VM VLAN insert register	RW
0xA000 - 0xA1FC	N/A	UTA	Unicast Table Array	RW
0x5D80 - 0x5D8C	N/A	VMRCTL	Virtual Mirror rule control	RW
0x5D90 - 0x5D9C	N/A	VMRVLAN	Virtual Mirror rule VLAN	RW
0x5DA0 - 0x5DAC	N/A	VMRVM	Virtual Mirror rule VM	RW
0x5DB0	N/A	SCCRL	Storm Control control register	RW
0x5DB4	N/A	SCSTS	Storm Control status	RO
0x5DB8	N/A	BSCTRH	Broadcast Storm control Threshold	RW
0x5DBC	N/A	MSCTRH	Multicast Storm control Threshold	RW

Table 8-6 Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0x5DC0	N/A	BSCCNT	Broadcast Storm Control Current Count	RO
0x5DC4	N/A	MSCCNT	Multicast Storm control Current Count	RO
0x5DC8	N/A	SCTC	Storm Control Time Counter	RO
0x5DCC	N/A	SCBI	Storm Control Basic interval	RW
VMDq Statistics				
0xC030 + 0x40 * n	0x2830 + 0x100 * n	RQDPC[0 - 3]	Receive Queue drop packet count Register 0 - 3	RW
0xC130 + 0x40 * (n- 4)	N/A	RQDPC[4 - 7]	Receive Queue drop packet count Register 4 - 7	RW
0xE030 + 0x40 * n	N/A	TQDPC[0 - 7]	Transmit Queue drop packet count Register 0 - 7	RW
0x10010 + 0x100*n	N/A	VFGPRC[0 - 7]	Per queue Good Packets Received Count	RO
0x10014 + 0x100*n	N/A	VFGPTC[0 - 7]	Per queue Good Packets Transmitted Count	RO
0x10018 + 0x100*n	N/A	VFGORC[0 - 7]	Per queue Good Octets Received Count	RO
0x10034 + 0x100*n	N/A	VFGOTC[0 - 7]	Per queue Octets Transmitted Count	RO
0x10038 + 0x100*n	N/A	VFMPRC[0 - 7]	Per queue Multicast Packets Received Count	RO
Statistics				
0x4000	N/A	CRCERRS	CRC Error Count	RC
0x4004	N/A	ALGNERRC	Alignment Error Count	RC
0x4008	N/A	SYMERRS	Symbol Error Count	RC
0x400C	N/A	RXERRC	RX Error Count	RC
0x4010	N/A	MPC	Missed Packets Count	RC
0x4014	N/A	SCC	Single Collision Count	RC
0x4018	N/A	ECOL	Excessive Collisions Count	RC
0x401C	N/A	MCC	Multiple Collision Count	RC
0x4020	N/A	LATECOL	Late Collisions Count	RC
0x4028	N/A	COLC	Collision Count	RC
0x4030	N/A	DC	Defer Count	RC
0x4034	N/A	TNCRS	Transmit - No CRS	RC
0x403C	N/A	HTDPMC	Host Transmit Discarded Packets by MAC Count	RC
0x4040	N/A	RLEC	Receive Length Error Count	RC
0x4048	N/A	XONRXC	XON Received Count	RC
0x404C	N/A	XONTXC	XON Transmitted Count	RC
0x4050	N/A	XOFFRXC	XOFF Received Count	RC
0x4054	N/A	XOFFTXC	XOFF Transmitted Count	RC
0x4058	N/A	FCRUC	FC Received Unsupported Count	RC
0x405C	N/A	PRC64	Packets Received (64 Bytes) Count	RC
0x4060	N/A	PRC127	Packets Received (65-127 Bytes) Count	RC
0x4064	N/A	PRC255	Packets Received (128-255 Bytes) Count	RC
0x4068	N/A	PRC511	Packets Received (256-511 Bytes) Count	RC
0x406C	N/A	PRC1023	Packets Received (512-1023 Bytes) Count	RC

Table 8-6 Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0x4070	N/A	PRC1522	Packets Received (1024-1522 Bytes)	RC
0x4074	N/A	GPRC	Good Packets Received Count	RC
0x4078	N/A	BPRC	Broadcast Packets Received Count	RC
0x407C	N/A	MPRC	Multicast Packets Received Count	RC
0x4080	N/A	GPTC	Good Packets Transmitted Count	RC
0x4088	N/A	GORCL	Good Octets Received Count (Lo)	RC
0x408C	N/A	GORCH	Good Octets Received Count (Hi)	RC
0x4090	N/A	GOTCL	Good Octets Transmitted Count (Lo)	RC
0x4094	N/A	GOTCH	Good Octets Transmitted Count (Hi)	RC
0x40A0	N/A	RNBC	Receive No Buffers Count	RC
0x40A4	N/A	RUC	Receive Under size Count	RC
0x40A8	N/A	RFC	Receive Fragment Count	RC
0x40AC	N/A	ROC	Receive Oversize Count	RC
0x40B0	N/A	RJC	Receive Jabber Count	RC
0x40B4	N/A	MNGPRC	Management Packets Receive Count	RC
0x40B8	N/A	MPDC	Management Packets Dropped Count	RC
0x40BC	N/A	MNGPTC	Management Packets Transmitted Count	RC
0x40C0	N/A	TORL	Total Octets Received (Lo)	RC
0x8FE0	N/A	B2OSPC	BMC2OS packets sent by BMC	RC
0x4158	N/A	B2OGPRC	BMC2OS packets received by host	RC
0x8FE4	N/A	O2BGPTC	OS2BMC packets received by BMC	RC
0x415C	N/A	O2BSPC	OS2BMC packets transmitted by host	RC
0x40C4	N/A	TORH	Total Octets Received (Hi)	RC
0x40C8	N/A	TOTL	Total Octets Transmitted (Lo)	RC
0x40CC	N/A	TOTH	Total Octets Transmitted (Hi)	RC
0x40D0	N/A	TPR	Total Packets Received	RC
0x40D4	N/A	TPT	Total Packets transmitted	RC
0x40D8	N/A	PTC64	Packets Transmitted (64 Bytes) Count	RC
0x40DC	N/A	PTC127	Packets Transmitted (65-127 Bytes) Count	RC
0x40E0	N/A	PTC255	Packets Transmitted (128-256 Bytes) Count	RC
0x40E4	N/A	PTC511	Packets Transmitted (256-511 Bytes) Count	RC
0x40E8	N/A	PTC1023	Packets Transmitted (512-1023 Bytes) Count	RC
0x40EC	N/A	PTC1522	Packets Transmitted (1024-1522 Bytes) Count	RC
0x40F0	N/A	MPTC	Multicast Packets Transmitted Count	RC
0x40F4	N/A	BPTC	Broadcast Packets Transmitted Count	RC
0x40F8	N/A	TSCTC	TCP Segmentation Context Transmitted Count	RC
0x4100	N/A	IAC	Interrupt Assertion Count	RC
0x4104	N/A	RPTHC	Rx Packets to host count	RC
0x4148	N/A	TLPIC	EEE TX LPI Count	RC
0x414C	N/A	RLPIC	EEE RX LPI Count	RC
0x4118	N/A	HGPTC	Host Good Packets Transmitted Count	RC
0x4120	N/A	RXDMTC	Rx Descriptor Minimum Threshold Count	RC
0x4128	N/A	HGORCL	Host Good Octets Received Count (Lo)	RC
0x412C	N/A	HGORCH	Host Good Octets Received Count (Hi)	RC

Table 8-6 Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0x4130	N/A	HGOTCL	Host Good Octets Transmitted Count (Lo)	RC
0x4134	N/A	HGOTCH	Host Good Octets Transmitted Count (Hi)	RC
0x4138	N/A	LENERRS	Length Errors count register	RC
0x4228	N/A	SCVPC	SerDes/SGMII/1000BASE-KX Code Violation Packet Count Register	RW
0x41A0	N/A	SSVPC	Switch Security Violation Packet Count	RC
0x41A4	N/A	SDPC	Switch Drop Packet Count	RC/W
0x4154	N/A	MNGFBDPC	Management full buffer drop packet count	RC/W
0x4150	N/A	LPBKFBDPC	Loopback full buffer drop packet count	RC/W
Manageability Statistics				
0x413C	N/A	BMNGPRC	BMC Management Packets Receive Count	RC
0x4140	N/A	BMRPDC	BMC Management Receive Packets Dropped Count	RC
0x8FDC	N/A	BMTPDC	BMC Management Transmit Packets Dropped Count	RC
0x4144	N/A	BMNGPTC	BMC Management Packets Transmitted Count	RC
0x4400	N/A	BUPRC	BMC Total Unicast Packets Received	RC
0x4404	N/A	BMPRC	BMC Total Multicast Packets Received	RC
0x4408	N/A	BBPRC	BMC Total Broadcast Packets Received	RC
0x440C	N/A	BUPTC	BMC Total Unicast Packets Transmitted	RC
0x4410	N/A	BMPTC	BMC Total Multicast Packets Transmitted	RC
0x4414	N/A	BBPTC	BMC Total Broadcast Packets Transmitted	RC
0x4418	N/A	BCRCERRS	BMC FCS Receive Errors	RC
0x441C	N/A	BALGNERRC	BMC Alignment Errors	RC
0x4420	N/A	BXONRXC	BMC Pause XON Frames Received	RC
0x4424	N/A	BXOFFRXC	BMC Pause XOFF Frames Received	RC
0x4428	N/A	BXONTXC	BMC Pause XON Frames Transmitted	RC
0x442C	N/A	BXOFFTXC	BMC Pause XOFF Frames Transmitted	RC
0x4430	N/A	BSCC	BMC Single Collision Transmit Frames	RC
0x4434	N/A	BMCC	BMC Multiple Collision Transmit Frames	RC
Wake up and Proxying				
0x5800	N/A	WUC	Wake Up Control	RW
0x5808	N/A	WUFC	Wake Up Filter Control	RW
0x5810	N/A	WUS	Wake Up Status	R/W1C
0x5F60	N/A	PROXYFC	Proxying Filter Control	RW
0x5F64	N/A	PROXYS	Proxying Status	R/W1C
0x5838	N/A	IPAV	IP Address Valid	RW
0x5840- 0x5858	N/A	IP4AT	IPv4 Address Table	RW
0x5880- 0x588F	N/A	IP6AT	IPv6 Address Table	RW
0x5900	N/A	WUPL	Wake Up Packet Length	RO
0x5A00- 0x5A7C	N/A	WUPM	Wake Up Packet Memory	RO
0x9000-0x93FC	N/A	FHFT	Flexible Host Filter Table registers	RW
0x9A00-0x9DFC	N/A	FHFT_EXT	Flexible Host Filter Table registers extended	RW
Manageability				
0x5010 - 0x502C	N/A	MAVTV[7:0]	VLAN TAG Value 7 - 0	RW

Table 8-6 Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0x5030 - 0x503C	N/A	MFUTP[3:0]	Management Flex UDP/TCP Ports	RW
0x5060 - 0x506C	N/A	METF[3:0]	Management Ethernet Type Filters	RW
0x5820	N/A	MANC	Management Control	RW
0x5864	N/A	MNGONLY	Management Only Traffic Register	RW
0x5890 - 0x58AC	N/A	MDEF[7:0]	Manageability Decision Filters	RW
0x5930 – 0x594C	N/A	MDEF_EXT[7:0]	Manageability Decision Filters	RW
0x58B0 - 0x58EC	N/A	MIPAF[15:0]	Manageability IP Address Filter	RW
0x5910 + 8*n	N/A	MMAL[3:0]	Manageability MAC Address Low 3:0	RW
0x5914 + 8*n	N/A	MMAH[3:0]	Manageability MAC Address High 3:0	RW
0x9400-0x94FC	N/A	FTFT	Flexible TCO Filter Table	RW
0x8800-0x8EFC	N/A	Flex MNG	Flex manageability memory address space	RW
0x8F00	N/A	HICR	HOST Interface Control Register	RW
0x8F0C	N/A	FWSTS	Firmware Status register	RW
Thermal Sensor				
0x8100	N/A	THMJT	Thermal Sensor Measured Junction Temperature	RO
0X8104	N/A	THLOWTC	Thermal Sensor Low Threshold Control	RW
0X8108	N/A	THMIDTC	Thermal Sensor Mid Threshold Control	RW
0X810C	N/A	THHIGHTC	Thermal Sensor High Threshold Control	RW
0x8110	N/A	THSTAT	Thermal Sensor Status	RO
0x8114	N/A	THACNFG	Thermal Sensor Auxiliary Configuration	RW
PCIe				
0x5B00	N/A	GCR	PCIe Control Register	RW
0x5B10	N/A	GSCL_1	PCIe statistics control #1	RW
0x5B14	N/A	GSCL_2	PCIe statistics control #2	RW
0x5B90 - 0x5B9C	N/A	GSCL_5_8	PCIe statistics control Leaky Bucket Timer	RW
0x5B20	N/A	GSCN_0	PCIe counter register #0	RW
0x5B24	N/A	GSCN_1	PCIe counter register #1	RW
0x5B28	N/A	GSCN_2	PCIe counter register #2	RW
0x5B2C	N/A	GSCN_3	PCIe counter register #3	RW
0x5B30	N/A	FACTPS	Function Active and Power State	RW
0x5B64	N/A	MREVID	Mirrored Revision ID	RO
0x5B6C	N/A	GCR_EXT	PCIe Control Extended Register	RW
0x5B74	N/A	DCA_CTRL	DCA Control Register	RW
0x5B88	N/A	PICAUSE	PCIe Interrupt Cause	R/W1C
0x5B8C	N/A	PIENA	PCIe Interrupt Enable	RW
0x5BFC	N/A	BARCTRL	PCIe BAR Control	RW
Memory Error Detection				
0x1084	N/A	PEIND	Parity and ECC Indication	RC
0x1088	N/A	PEINDM	Parity and ECC Indication Mask	RW

Table 8-6 Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0x245C	N/A	RPBECCSTS	Receive Packet buffer ECC control	RW
0x345C	N/A	TPBECCSTS	Transmit Packet buffer ECC control	RW
0x5BA0	N/A	PCIEERRCTL	PCIe Parity Control Register	RW
0x5BA4	N/A	PCIEECCCTL	PCIe ECC Control Register	RW
0x5BA8	N/A	PCIEERRSTS	PCIe Parity status Register	R/W1C
0x5BAC	N/A	PCIEECCSTS	PCIe ECC Status Register	R/W1C
0x5F54	N/A	LANPERRCTL	LAN Port Parity Error Control register	RW
0x5F58	N/A	LANPERRSTS	LAN Port Parity Error Status register	R/W1C
0x3F00	N/A	DTPARC	DMA Transmit ECC and Parity Control	RW
0x3F10	N/A	DTPARS	DMA Transmit ECC and Parity Status	R/W1C
0x3F04	N/A	DRPARC	DMA Receive ECC and Parity Control	RW
0x3F14	N/A	DRPARS	DMA Receive ECC and Parity Status	R/W1C
0x3F08	N/A	DDECCC	Dhost ECC Control	RW
0x3F18	N/A	DDECCS	Dhost ECC Status	R/W1C
Power Management Registers				
0x2508	N/A	DMACR	DMA Coalescing Control Register	RW
0x2514	N/A	DMCTLX	DMA Coalescing Time to LX Request	RW
0x3550	N/A	DMCTXTH	DMA Coalescing Transmit Threshold	RW
0x5DD0	N/A	DMCRTRH	DMA Coalescing Receive Packet Rate Threshold	RW
0x5DD4	N/A	DMCCNT	DMA Coalescing Current RX Count	RO
0x2170	N/A	FCRTC	Flow Control Receive Threshold Coalescing	RW
0x5BB0	N/A	LTRMINV	Latency Tolerance Reporting (LTR) Minimum Values	RW
0x5BB4	N/A	LTRMAXV	Latency Tolerance Reporting (LTR) Maximum Values	RW
0x01A0	N/A	LTRC	Latency Tolerance Reporting (LTR) Control	RW
0x0E30	N/A	EEER	Energy Efficient Ethernet (EEE) Register	RW
0x0E34	N/A	EEE_SU	Energy Efficient Ethernet (EEE) Setup Register	RW
0x24A4	N/A	SWDFPC	Switch Data FIFO Packet Count	RO
0x24E8	N/A	PBRWAC	Receive Packet Buffer wrap around counter	RO
PCS				
0x4200	N/A	PCS_CFG	PCS Configuration 0 Register	RW
0x4208	N/A	PCS_LCTL	PCS Link Control Register	RW
0x420C	N/A	PCS_LSTS	PCS Link Status Register	RO
0x4210	N/A	PCS_DBG0	PCS Debug 0 register	RO
0x4214	N/A	PCS_DBG1	PCS Debug 1 register	RO
0x4218	N/A	PCS_ANADV	AN advertisement Register	RW
0x421C	N/A	PCS_LPAB	Link Partner Ability Register	RO
0x4220	N/A	PCS_NPTX	AN Next Page Transmit Register	RW
0x4224	N/A	PCS_LPABNP	Link Partner Ability Next Page Register	RO
Time Sync				
0xB620	N/A	TSYNCRXCTL	RX Time Sync Control register	RW
0xB624	N/A	RXSTMPL	RX timestamp Low	RO
0xB628	N/A	RXSTMPH	RX timestamp High	RO
0xB62C	N/A	RXSATRL	RX timestamp attributes low	RO

Table 8-6 Register Summary (Continued)

Offset	Alias Offset	Abbreviation	Name	RW
0xB630	N/A	RXSATRH	RX timestamp attributes low	RO
0xB614	N/A	TSYNCTXCTL	TX Time Sync Control register	RW
0xB618	N/A	TXSTMPL	TX timestamp value Low	RO
0xB61C	N/A	TXSTMPH	TX timestamp value High	RO
0xB6F8	N/A	SYSTIMR	System time residue register	RW
0xB600	N/A	SYSTIML	System time register Low	RW
0xB604	N/A	SYSTIMH	System time register High	RW
0xB608	N/A	TIMINCA	Increment attributes register	RW
0xB60C	N/A	TIMADJL	Time adjustment offset register low	RW
0xB610	N/A	TIMADJH	Time adjustment offset register high	RW
0xB640	N/A	TSAUXC	Auxiliary Control Register	RW
0xB644	N/A	TRGTTIML0	Target Time register 0 Low	RW
0xB648	N/A	TRGTTIMH0	Target Time register 0 High	RW
0xB64C	N/A	TRGTTIML1	Target Time register 1 Low	RW
0xB650	N/A	TRGTTIMH1	Target Time register 1 High	RW
0xB654	N/A	FREQOUT0	Frequency out 0 Control register	RW
0xB658	N/A	FREQOUT1	Frequency out 1 Control register	RW
0xB65C	N/A	AUXSTMPL0	Auxiliary Time Stamp 0 register Low	RO
0xB660	N/A	AUXSTMPH0	Auxiliary Time Stamp 0 register High	RO
0xB664	N/A	AUXSTMPL1	Auxiliary Time Stamp 1 register Low	RO
0xB668	N/A	AUXSTMPH1	Auxiliary Time Stamp 1 register High	RO
0x5F50	N/A	TSYNCRXCFG	Time Sync RX Configuration	RW
0x003C	N/A	TSSDP	Time Sync SDP Configuration Reg	RW
0xB66C	N/A	TSICR	Time Sync Interrupt Cause Register	RC/W1C
0xB670	N/A	TSIS	Time Sync Interrupt Set Register	WO
0xB674	N/A	TSIM	Time Sync Interrupt Mask Register	RW

8.1.3.1 Alias Addresses

Certain registers maintain an alias address designed for backward compatibility with software written for previous GbE controllers. For these registers, the alias address is shown in the table above. Those registers can be accessed by software at either the new offset or the alias offset. It is recommended that software that is written solely for the I350, use the new address offset.

Note:

8.1.4 MSI-X BAR Register Summary

Table 8-7 MSI-X Register Summary

Category	Offset	Abbreviation	Name	RW	Page
MSI-X Table	0x0000 + n*0x10 [n=0...24]	MSIXTADD	MSI–X Table Entry Lower Address	RW	page 535
MSI-X Table	0x0004 + n*0x10 [n=0...24]	MSIXTUADD	MSI–X Table Entry Upper Address	RW	page 535
MSI-X Table	0x0008 + n*0x10 [n=0...24]	MSIXTMSG	MSI–X Table Entry Message	R/W	page 535
MSI-X Table	0x02000	MSIXPBA	MSIXPBA Bit Description	RO	page 536

8.2 General Register Descriptions

8.2.1 Device Control Register - CTRL (0x00000; R/W)

This register, as well as the Extended Device Control register (CTRL_EXT), controls the major operational modes for the device. While software write to this register to control device settings, several bits (such as FD and SPEED) can be overridden depending on other bit settings and the resultant link configuration determined by the PHY's Auto- Negotiation resolution. See Section 4.6.7 for details on the setup of these registers in the different link modes.

Note: This register is also aliased at address 0x0004.

Field	Bit(s)	Initial Value	Description
FD	0	1b[1]	Full-Duplex Controls the MAC duplex setting when explicitly set by software. 0b = half duplex. 1b = full duplex.
Reserved	1	0b	Reserved Write 0 ignore on read.
GIO Master Disable	2	0b	When set to 1b, the function of this bit blocks new master requests including manageability requests. If no master requests are pending by this function, the *STATUS.GIO Master Enable Status* bit is set. See Section 5.2.3.3 for further information.
Reserved	5:3	0x0	Reserved Write 0, ignore on read.

Field	Bit(s)	Initial Value	Description
SLU	6	0b[1]	Set Link Up Set Link Up must be set to 1 to permit the MAC to recognize the LINK signal from the PHY, which indicates the PHY has gotten the link up, and is ready to receive and transmit data. See Section 3.7.4 for more information about Auto-Negotiation and link configuration in the various modes. ***Notes:*** 1. The *CTRL.SLU* bit is normally initialized to 0. However, if the *APM Enable* bit is set in the EEPROM then it is initialized to 1b. 2. The *CTRL.SLU* bit will be set to 1b if the *Enable All PHYs in D3* bit in the Common Firmware Parameters 2 EEPROM word is set to 1 (See Section 6.3.7.3). 3. The *CTRL.SLU* bit is set in NCSI mode according to the "enable channel command" to the port. 4. In SerDes and 1000Base-KX modes Link up can be forced by setting this bit as described in Section 3.7.4.1.4.
ILOS	7	0b[1]	Invert Loss-of-Signal (LOS/LINK) Signal Bit controls the polarity of the SRDS_[n]_SIG_DET signal or internal Link up signal. 0b = Do not invert (active high input signal). 1b = Invert signal (active low input signal). ***Notes:*** 1. Source of the link-up signal (SRDS_[n]_SIG_DET signal or internal Link up signal) is set via the *CONNSW.ENRGSRC* bit. When using internal link-up signal bit should be 0. 2. Should be set to zero when using internal copper PHY or when working in SGMII, 1000BASE-BX or 1000BASE-KX modes.
SPEED	9:8	10b	Speed selection. These bits determine the speed configuration and are written by software after reading the PHY configuration through the MDIO interface. These signals are ignored when Auto-Speed Detection is enabled. 00b = 10 Mb/s. 01b = 100 Mb/s. 10b = 1000 Mb/s. 11b = not used.
Reserved	10	0b	Reserved. Write 0, ignore on read.
FRCSPD	11	0b[1]	Force Speed This bit is set when software needs to manually configure the MAC speed settings according to the SPEED bits. Note that MAC and PHY must resolve to the same speed configuration or software must manually set the PHY to the same speed as the MAC. Software must clear this bit to enable the PHY or ASD function to control the MAC speed setting. Note that this bit is superseded by the *CTRL_EXT.SPD_BYPS* bit which has a similar function.
FRCDPLX	12	0b	Force Duplex When set to 1b, software can override the duplex indication from the PHY that is indicated in the FDX to the MAC. Otherwise, in 10/100/1000Base-T link mode, the duplex setting is sampled from the PHY FDX indication into the MAC on the asserting edge of the PHY LINK signal. When asserted, the *CTRL.FD* bit sets duplex.
Reserved	15:13	0x0	Reserved Write 0, ignore on read.
SDP0_GPIEN	16	0b	General Purpose Interrupt Detection Enable for SDP0 If software-controlled IO pin SDP0 is configured as an input, this bit (when 1b) enables the use for GPI interrupt detection.
SDP1_GPIEN	17	0b	General Purpose Interrupt Detection Enable for SDP1 If software-controlled IO pin SDP1 is configured as an input, this bit (when 1b) enables the use for GPI interrupt detection.

Field	Bit(s)	Initial Value	Description
SDP0 DATA (RWS)	18	0b[1]	SDP0 Data Value Used to read or write the value of software-controlled IO pin SDP0. If SDP0 is configured as an output (*SDP0_IODIR* = 1b), this bit controls the value driven on the pin (initial value EEPROM-configurable). If SDP0 is configured as an input, reads return the current value of the pin. When the SDP0_WDE bit is set, this field indicates the polarity of the watchdog indication.
SDP1 DATA (RWS)	19	0b[1]	SDP1 Data Value Used to read or write the value of software-controlled IO pin SDP1. If SDP1 is configured as an output (SDP1_IODIR = 1b), this bit controls the value driven on the pin (initial value EEPROM-configurable). If SDP1 is configured as an input, reads return the current value of the pin.
ADVD3WUC	20	1b[1]	D3Cold Wake up Capability Enable When bit is 0b PME (WAKE#) is not generated in D3Cold. Bit loaded from EEPROM (refer to Section 6.2.21).
SDP0_WDE	21	0b[1]	SDP0 used for Watchdog indication When set, SDP0 is used as a watchdog indication. When set, the SDP0_DATA bit indicates the polarity of the watchdog indication. In this mode, S*DP0_IODIR* must be set to an output.
SDP0_IODIR	22	0b[1]	SDP0 Pin Direction Controls whether software-controllable pin SDP0 is configured as an input or output (0b = input, 1b = output). Initial value is EEPROM-configurable. This bit is not affected by software or system reset, only by initial power-on or direct software writes.
SDP1_IODIR	23	0b[1]	SDP1 Pin Direction Controls whether software-controllable pin SDP1 is configured as an input or output (0b = input, 1b = output). Initial value is EEPROM-configurable. This bit is not affected by software or system reset, only by initial power-on or direct software writes.
Reserved	25:24	0x0	Reserved. Write 0, ignore on read.
RST (SC)	26	0b	Port Software Reset This bit performs reset to the respective port, resulting in a state nearly approximating the state following a power-up reset or internal PCIe reset, except for system PCI configuration and logic used by all ports. 0b = Normal. 1b = Reset. This bit is self clearing and is referred to as software reset or global reset.
RFCE	27	1b	Receive Flow Control Enable When set, indicates that the I350 responds to the reception of flow control packets. If Auto-Negotiation is enabled, this bit is set to the negotiated flow control value. In SerDes mode the resolution is done by the hardware. In internal PHY, SGMII or 1000BASE-KX modes it should be done by the software.
TFCE	28	0b	Transmit Flow Control Enable When set, indicates that the I350 transmits flow control packets (XON and XOFF frames) based on the receiver fullness. If Auto-Negotiation is enabled, this bit is set to the negotiated duplex value. In SerDes mode the resolution is done by the hardware. In internal PHY, SGMII or 1000BASE-KX modes it should be done by the software.

Field	Bit(s)	Initial Value	Description
DEV_RST (SC)	29	0b	Device Reset This bit performs a reset of the entire controller device, resulting in a state nearly approximating the state following a power-up reset or internal PCIe reset, except for system PCI configuration. 0b = Normal. 1b = Reset. This bit is self clearing. ***Notes:*** 1. Assertion of *DEV_RST* generates an interrupt on all ports via the ICR.DRSTA interrupt bit. 2. Device Reset (*CTRL.DEV_RST*) can be used to globally reset the entire component if the *DEV_RST_EN* bit in Initialization Control 4 EEPROM word is set. 3. Assertion of *DEV_RST* sets on all ports the *STATUS.DEV_RST_SET* bit. For additional information, refer to Section 4.3.4.
VME	30	0b	VLAN Mode Enable When set to 1b, VLAN information is stripped from all received 802.1Q packets. ***Note:*** If this bit is set the *RCTL.SECRC* bit should also be set as the CRC is not valid anymore.
PHY_RST	31	0b	PHY Reset Generates a hardware-level reset to the internal 1000BASE-T PHY. 0b = Normal operation. 1b = Internal PHY reset asserted.

1. These bits are loaded from EEPROM.

8.2.2 Device Status Register - STATUS (0x0008; RO)

Field	Bit(s)	Initial Value	Description
FD	0	X	Full Duplex. 0 = Half duplex (HD). 1= Full duplex (FD). Reflects duplex setting of the MAC and/or link. FD reflects the actual MAC duplex configuration. This normally reflects the duplex setting for the entire link, as it normally reflects the duplex configuration negotiated between the PHY and link partner (copper link) or MAC and link partner (fiber link).
LU	1	X	Link up. 0 = no link established; 1 = link established. For this bit to be valid, the Set Link Up bit of the Device Control Register (*CTRL.SLU*) must be set. Link up provides a useful indication of whether something is attached to the port. Successful negotiation of features/link parameters results in link activity. The link startup process (and consequently the duration for this activity after reset) can be several 100's of ms. When the internal PHY is used, this reflects whether the PHY's LINK indication is present. When the SerDes, SGMII or 1000BASE-KX interface is used, this indicates loss-of-signal; if Auto-Negotiation is also enabled, this can also indicate successful Auto-Negotiation. Refer to Section 3.7.4 for more details. Note: Bit is valid only when working in Internal PHY mode. In SerDes mode bit is always 0.

Field	Bit(s)	Initial Value	Description
LAN ID	3:2	Port 0 = 00b Port 1 = 01b Port 2 = 10b Port 3 = 11b	LAN ID Provides software a mechanism to determine the LAN identifier for the MAC. 00b = LAN 0. 01b = LAN 1. 10b = LAN 2. 11b = LAN 3.
TXOFF	4	X	Transmission Paused This bit indicates the state of the transmit function when symmetrical flow control has been enabled and negotiated with the link partner. This bit is set to 1b when transmission is paused due to the reception of an XOFF frame. It is cleared (0b) upon expiration of the pause timer or the receipt of an XON frame.
Reserved	5	X	Reserved. Write 0, ignore on read.
SPEED	7:6	X	Link Speed Setting Reflects the speed setting of the MAC and/or link when it is operating in 10/100/1000BASE-T mode (internal PHY). When the MAC is operating in 10/100/1000BASE-T mode with the internal PHY, these bits normally reflect the speed of the actual link, negotiated by the PHY and link partner and reflected internally from the PHY to the MAC (*SPD_IND*). These bits also might represent the speed configuration of the MAC only, if the MAC speed setting has been forced via software (*CTRL.SPEED*) or if MAC auto-speed detection is used. If Auto-Speed Detection is enabled, the I350's speed is configured only once after the LINK signal is asserted by the PHY. 00b = 10 Mb/s. 01b = 100 Mb/s. 10b = 1000 Mb/s. 11b = 1000 Mb/s.
ASDV	9:8	X	Auto-Speed Detection Value Speed result sensed by the I350’s MAC auto-detection function. These bits are provided for diagnostics purposes only. The ASD calculation can be initiated by software writing a logic 1b to the *CTRL_EXT.ASDCHK* bit. The resultant speed detection is reflected in these bits. Refer to Section 8.2.3 for details.
PHYRA	10	1b	PHY Reset Asserted This read/write bit is set by hardware following the assertion of an internal PHY reset; it is cleared by writing a 0b to it. This bit is also used by firmware indicating a required initialization of the I350’s PHY.
Reserved	13:11	0x0	Reserved. Write 0, ignore on read.
Num VFs (RO)	17:14	0x0	Reflects the value of the Num VFs in the IOV capability structure.
IOV mode (RO)	18	0b	Reflects the value of the VF enable (VFE) bit in the IOV capability structure.
GIO Master Enable Status	19	1b	Cleared by the I350 when the *CTRL.GIO Master Disable* bit is set and no master requests are pending by this function and is set otherwise. Indicates that no master requests are issued by this function as long as the *CTRL.GIO Master Disable* bit is set.
DEV_RST_SET (R/W1C)	20	0b	Device Reset Set When set indicates that a device reset (*CTRL.DEV_RST*) was initiated by one of the software drivers. Note: Bit cleared by write 1.
PF_RST_DONE	21	1b	PF _RST_DONE When set indicates that Software reset (*CTRL.RST*) or Device reset (*CTRL.DEV_RST*) has completed and Software driver can begin initialization process.
Reserved	30:22	0x0	Reserved. Write 0, ignore on read.

Field	Bit(s)	Initial Value	Description
MAC clock gating Enable	31	1b[1]	MAC clock gating Enable bit loaded from the EEPROM- indicates the device support gating of the MAC clock.

1. If the signature bits of the EEPROM's *Initialization Control Word 1* match (01b), this bit is read from the EEPROM.

8.2.3 Extended Device Control Register - CTRL_EXT (0x0018; R/W)

This register provides extended control of the I350's functionality beyond that provided by the Device Control register (*CTRL*).

Field	Bit(s)	Initial Value	Description
Reserved	1:0	0b	Reserved. Write 0, ignore on read.
SDP2_GPIEN	2	0b	General Purpose Interrupt Detection Enable for SDP2. If software-controllable IO pin SDP2 is configured as an input, this bit (when set to 1b) enables use for GPI interrupt detection.
SDP3_GPIEN	3	0b	General Purpose Interrupt Detection Enable for SDP3. If software-controllable IO pin SDP3 is configured as an input, this bit (when set to 1b) enables use for GPI interrupt detection.
Reserved	5:4	00b	Reserved. Write 0, ignore on read.
SDP2_DATA	6	0b[1]	SDP2 Data Value. Used to read (write) the value of software-controllable IO pin SDP2. If SDP2 is configured as an output (SDP2_IODIR = 1b), this bit controls the value driven on the pin (initial value EEPROM-configurable). If SDP2 is configured as an input, reads return the current value of the pin.
SDP3_DATA	7	0b[1]	SDP3 Data Value. Used to read (write) the value of software-controllable IO pin SDP3. If SDP3 is configured as an output (SDP3_IODIR = 1b), this bit controls the value driven on the pin (initial value EEPROM-configurable). If SDP3 is configured as an input, reads return the current value of the pin.
Reserved	9:8	0x0[1]	Reserved. Write 0, ignore on read.
SDP2_IODIR	10	0b[1]	SDP2 Pin Direction. Controls whether software-controllable pin SDP2 is configured as an input or output (0b = input, 1b = output). Initial value is EEPROM-configurable. This bit is not affected by software or system reset, only by initial power-on or direct software writes.
SDP3_IODIR	11	0b[1]	SDP3 Pin Direction. Controls whether software-controllable pin SDP3 is configured as an input or output (0b = input, 1b = output). Initial value is EEPROM-configurable. This bit is not affected by software or system reset, only by initial power-on or direct software writes.
ASDCHK	12	0b	ASD Check Initiates an Auto-Speed-Detection (ASD) sequence to sense the frequency of the PHY receive clock (RX_CLK). The results are reflected in STATUS.ASDV. This bit is self-clearing.
EE_RST (SC)	13	0b	EEPROM Reset When set, initiates a reset-like event to the EEPROM function. This causes an EEPROM Auto-load operation as if a software reset (*CTRL.RST*) had occurred. This bit is self-clearing.
PFRSTD (SC)	14	0b	PF reset Done. When set, the RSTI bit in all the VFMailbox registers is cleared and the RSTD bit in all the VFMailbox regs is set (Refer to Section 4.6.11.2.3 for additional information).

Field	Bit(s)	Initial Value	Description
SPD_BYPS	15	0b	Speed Select Bypass When set to 1b, all speed detection mechanisms are bypassed, and the I350 is immediately set to the speed indicated by *CTRL.SPEED*. This provides a method for software to have full control of the speed settings of the I350 and when the change takes place, by overriding the hardware clock switching circuitry.
NS_DIS	16	0	No Snoop Disable When set to 1b, the I350 does not set the no snoop attribute in any PCIe packet, independent of PCIe configuration and the setting of individual no snoop enable bits. When set to 0b, behavior of no snoop is determined by PCIe configuration and the setting of individual no snoop enable bits.
RO_DIS	17	0b	Relaxed Ordering Disabled When set to 1b, the I350 does not request any relaxed ordering transactions on the PCIe interface regardless of the state of bit 4 in the *PCIe Device Control* register. When this bit is cleared and bit 4 of the *PCIe Device Control* register is set, the I350 requests relaxed ordering transactions as specified by registers *RXCTL* and *TXCT*L (per queue and per flow).
SerDes Low Power Enable	18	0b[1]	When set, allows the SerDes to enter a low power state when the function is in Dr state.
Dynamic MAC Clock Gating	19	0b[1]	When set, enables Dynamic MAC Clock Gating.
PHY Power Down Enable	20	1b[1]	When set, enables the PHY to enter a low-power state as described in Section 5.4.3.
Reserved	21	0b	Reserved. Write 0, ignore on read.
LINK_MODE	23:22	0x00[1] 0x11[2]	Link Mode Controls interface on the link. 00b = Direct copper (1000Base-T) interface (10/100/1000 BASE-T internal PHY mode). 01b = 1000BASE-KX. 10b = SGMII. 11b = SerDes interface. ***Note:*** 1. This bit is reset only on Power-up or PCIe reset. 2.
Reserved	24	0b	Reserved. Write 0, ignore on read.
I2C Enabled	25	0b[1]	Enable I2C This bit enables the SFPx_I2C pins that can be used to access external SFP modules or an external 1000BASE-T PHY via the MDIO interface. If cleared, the SFPx_I2C pads are isolated and accesses to the SFPx_I2C pins through the *I2CCMD* register or the *MDIC* register are ignored.
EXT_VLAN	26	0b[1]	External VLAN Enable When set, all incoming Rx packets are expected to have at least one VLAN with the Ether type as defined in *VET.EXT_VET* that should be ignored. The packets can have a second internal VLAN that should be used for all filtering purposes. All Tx packets are expected to have at least one VLAN added to them by the host. In the case of an additional VLAN request (*VLE* - VLAN Enable is set in transmit descriptor) the second VLAN is added after the first external VLAN is added by the host. This bit is reset only by a power up reset or by an EEPROM full auto load and should only be changed while Tx and Rx processes are stopped.

Field	Bit(s)	Initial Value	Description
Reserved	27	0b	Reserved. Write 0, ignore on read.
DRV_LOAD	28	0b	Driver Loaded This bit should be set by the driver after it is loaded. This bit should be cleared when the driver unloads or after a PCIe reset. The Management controller reads this bit to indicate to the manageability controller (BMC) that the driver has loaded. ***Note:*** Bit is reset on Power-up or PCIe reset only.
Reserved	31:29	0b	Reserved Write 0, Ignore on read.

1. These bits are read from the EEPROM.
2. Default value for SerDes only SKU.

The I350 allows up to four externally controlled interrupts. All software-definable pins, these can be mapped for use as GPI interrupt bits. Mappings are enabled by the SDPx_GPIEN bits only when these signals are also configured as inputs via SDPx_IODIR. When configured to function as external interrupt pins, a GPI interrupt is generated when the corresponding pin is sampled in an active-high state.

The bit mappings are shown in the table bellow for clarity.

Table 8-8 Mappings for SDI Pins Used as GPI

SDP Pin Used as GPI	CTRL_EXT Field Settings		Resulting ICR Bit (GPI)
	Direction	Enable as GPI interrupt	
3	SDP3_IODIR	SDP3_GPIEN	14
2	SDP2_IODIR	SDP2_GPIEN	13
1	SDP1_IODIR	SDP1_GPIEN	12
0	SDP0_IODIR	SDP0_GPIEN	11

Note: If software uses the EE_RST function and desires to retain current configuration information, the contents of the control registers should be read and stored by software. Control register values are changed by a read of the EEPROM which occurs upon assertion of the EE_RST bit.

The EEPROM reset function can read configuration information out of the EEPROM which affects the configuration of PCIe space BAR settings. The changes to the BARs are not visible unless the system reboots and the BIOS is allowed to re-map them.

The SPD_BYPS bit performs a similar function to the CTRL.FRCSPD bit in that the I350's speed settings are determined by the value software writes to the CRTL.SPEED bits. However, with the SPD_BYPS bit asserted, the settings in CTRL.SPEED take effect immediately rather than waiting until after the I350's clock switching circuitry performs the change.

8.2.4 MDI Control Register - MDIC (0x0020; R/W)

Software uses this register to read or write Media Dependent Interface (MDI) registers in the internal PHY or an external SGMII PHY.

Field	Bit(s)	Initial Value	Description
DATA	15:0	X	Data In a Write command, software places the data bits and the MAC shifts them out to the PHY. In a Read command, the MAC reads these bits serially from the PHY and software can read them from this location.
REGADD	20:16	0x0	PHY Register Address: Reg. 0, 1, 2,...31
Reserved	25:21	0x0	Reserved. Write 0, ignore on read.
OP	27:26	0x0	Opcode 01b = MDI Write 10b = MDI Read All other values are reserved.
R (RWS)	28	1b	Ready Bit Set to 1b by the I350 at the end of the MDI transaction (for example, indication of a Read or Write completion). It should be reset to 0b by software at the same time the command is written.
MDI_IE	29	0b	Interrupt Enable When set to 1 an Interrupt is generated at the end of an MDI cycle to indicate an end of a read or write operation to the PHY.
MDI_ERR (RWS)	30	0b	Error This bit is set to 1b by hardware when it fails to complete an MDI read. Software should make sure this bit is clear (0b) before issuing an MDI read or write command. Note: bit is valid only when the Ready bit is set.
Reserved	31	0b	Reserved. Write 0, ignore on read.

8.2.5 MDC/MDIO Configuration Register – MDICNFG (0x0E04; R/W)

Note: This register is used to configure the MDIO connection that is accessed via the *MDIC* register. Refer to Section 3.7.2.2.2 for details on usage of this register.

Field	Bit(s)	Initial Value	Description
Reserved	20:0	0x0	Reserved. Write 0, ignore on read.
PHYADD[1]	25:21	0x00 - LAN 0 0x01 - LAN 1 0x02 - LAN 2 0x03 - LAN 3	External PHY Address When *MDICNFG.Destination* bit is 0b, default PHYADD accesses internal PHY.

Field	Bit(s)	Initial Value	Description
Reserved	29:26	0x0	Reserved. Write 0, ignore on read.
Com_MDIO[2]	30	0b	When interfacing an external SGMII PHY bit defines if MDIO access is routed to the common MDIO port on LAN 0, to support multi port external PHYs, or to the dedicated per function MDIO port. 0b - MDIO access routed to the LAN port's MDIO interface. 1b - MDIO accesses on this LAN port routed to LAN port 0 MDIO interface
Destination[3]	31	0b	Destination 0b = The MDIO transaction is to the internal PHY. 1b = The MDIO transaction is directed to the external MDIO pins (I^2C Interface). Note: • When PHY registers access is initiated via the I2CCMD interface, access is always via the external I^2C Interface. In this case the destination field should always be 0.

1. PHYADD Loaded from *Initialization Control 4* EEPROM word to allocate per port address when using external MDIO port.
2. Common MDIO usage configuration bit is loaded from *Initialization Control 3* EEPROM word.
3. Destination Loaded from EEPROM Initialization Control 3 word. When external PHY supports a MDIO interface bit is 1, otherwise bit is 0.

8.2.6 SERDES Control 0 - P1GCTRL0 (0x0E08; RW)

Field	Bit(s)	Initial Value	Description
Reserved	0	0b	Reserved. Write 0, ignore on read.
LPBKBUF	1	0b	External SerDes LoopBack 0b - Normal operation. 1b - Enable External SerDes loopback (RX to TX).
Reserved	4:2	0x0	Reserved. Write 0, ignore on read.
Reserved_1	6:5	11b	Reserved. Write 11b, ignore on read.
Reserved	31:7	0x0	Reserved. Write 0, ignore on read.

8.2.7 Copper/Fiber Switch Control - CONNSW (0x0034; R/W)

Field	Bit(s)	Initial Value	Description
AUTOSENSE_EN	0	0b	Auto Sense Enable When set, the auto sense mode is active. In this mode the non-active link is sensed by hardware as follows PHY Sensing: The electrical idle detector of the receiver of the PHY is activated while in SerDes, SGMII or 1000BASE-KX mode. SerDes sensing: The electrical idle detector of the receiver of the SerDes is activated while in internal PHY mode, assuming the *ENRGSRC* bit is cleared If energy is detected in the non active media, the OMED bit in the ICR register is set and this bit is cleared. This includes the case where energy was present at the non-active media when this bit is being set.
AUTOSENSE_CONF	1	0b	Auto Sense Configuration Mode This bit should be set during the configuration of the PHY/SerDes towards the activation of the auto-sense mode to avoid spurious interrupts. While this bit is set, the PHY/SerDes is active even though the active link is set to SerDes, 1000BASE-KX or SGMII/PHY. Energy detection while this bit is set is not reflected to the *ICR.OMED* interrupt.
ENRGSRC	2	0b[1]	SerDes Energy Detect Source 0b - SerDes Energy detect source is internal. 1b - SerDes Energy detect source is from SRDS_[n]_SIG_DET pin. If set, the OMED interrupt cause is set after asserting the external signal detect pin. If cleared, the OMED interrupt cause is set after exiting from electrical idle of the SerDes receiver. This bit also defines the source of the signal detect indication used to set link up while in SerDes mode. ***Note:*** In SGMII and 1000BASE-KX modes energy detect source is internal and value of *CONNSW.ENRGSRC* bit should be 0b.
ASCLR_DIS	3	0b	Reserved
Reserved	8:4	0x0	Reserved. Write 0, ignore on read.
SerDesD (RO)	9	X	SerDes Signal Detect Indication Indicates the SerDes signal detect value according to the selected source (either external or internal). Valid only if LINK_MODE is SerDes, 1000BASE-KX or SGMII.
PHYSD (RO)	10	X	PHY Signal Detect Indication Valid only if LINK_MODE is the PHY and the receiver is not in electrical idle.
PHY_PDN (RO)	11	X	This bit indicates that the internal GbE PHY is in power down state. 0 = Internal GbE PHY not in power down. 1 = Internal GbE PHY in power down.
Reserved	31:12	0x0	Reserved. Write 0, ignore on read.

1. The default value of the ENRGSRC bit in this register is defined in the *Initialization Control 3* (Offset 0x24) EEPROM word (bit 15).

8.2.8 VLAN Ether Type - VET (0x0038; R/W)

This register contains the type field hardware matches against to recognize an 802.1Q (VLAN) Ethernet packet. To be compliant with the 802.3ac standard, the *VET.VET* field has a value of 0x8100.

Field	Bit(s)	Initial Value	Description
VET (RO)	15:0	0x8100	VLAN EtherType
VET EXT	31:16	0x8100	External VLAN Ether Type.

8.2.9 LED Control - LEDCTL (0x0E00; RW)

This register controls the setup of the LEDs. Refer to Section 7.5.1 for details of the MODE fields encoding.

Field	Bit(s)	Initial Value	Description
LED0_MODE	3:0	0010b[1]	LED0/LINK# Mode This field specifies the control source for the LED0 output. An initial value of 0010b selects LINK_UP# indication.
LED_PCI_MODE	4	0b	0b = Use LEDs as defined in the other fields of this register. 1b = Use LEDs to indicate PCI-E Lanes Idle status in SDP mode (only when the led_mode is set to 0x8 – SDP mode) For Port 0 LED0 3-0 indicates RX lanes 3- 0 Electrical Idle status For Port 1 LED1 3-0 indicates TX lanes 3- 0 Electrical Idle status
GLOBAL_BLINK_MODE	5	0b[1]	Global Blink Mode This field specifies the blink mode of all the LEDs. 0b = Blink at 200 ms on and 200 ms off. 1b = Blink at 83 ms on and 83 ms off.
LED0_IVRT	6	0b[1]	LED0/LINK# Invert This field specifies the polarity/ inversion of the LED source prior to output or blink control. 0b = Do not invert LED source (LED active low). 1b = Invert LED source (LED active High).
LED0_BLINK	7	0b[1]	LED0/LINK# Blink This field specifies whether to apply blink logic to the (possibly inverted) LED control source prior to the LED output. 0b = Do not blink asserted LED output. 1b = Blink asserted LED output.
LED1_MODE	11:8	0011b[1]	LED1/ACTIVITY# Mode This field specifies the control source for the LED1 output. An initial value of 0011b selects FILTER ACTIVITY# indication.
Reserved	12	0b	Reserved Write as 0 ignore on read.
Reserved	13	0b	Reserved Write 0 ignore on read.
LED1_IVRT	14	0b[1]	LED1/ACTIVITY# Invert This field specifies the polarity/ inversion of the LED source prior to output or blink control. 0b = Do not invert LED source (LED active low). 1b = Invert LED source (LED active High).
LED1_BLINK	15	1b[1]	LED1/ACTIVITY# Blink

Field	Bit(s)	Initial Value	Description
LED2_MODE	19:16	0110b[1]	LED2/LINK100# Mode This field specifies the control source for the LED2 output. An initial value of 0011b selects LINK100# indication.
Reserved	21:20	0x0	Reserved. Write 0, ignore on read.
LED2_IVRT	22	0b[1]	LED2/LINK100# Invert This field specifies the polarity/ inversion of the LED source prior to output or blink control. 0b = Do not invert LED source (LED active low). 1b = Invert LED source (LED active High).
LED2_BLINK	23	0b[1]	LED2/LINK100# Blink
LED3_MODE	27:24	0111b[1]	LED3/LINK1000# Mode This field specifies the control source for the LED3 output. An initial value of 0111b selects LINK1000# indication.
Reserved	29:28	0x0	Reserved Write 0, ignore on read.
LED3_IVRT	30	0b[1]	LED3/LINK1000# Invert This field specifies the polarity/ inversion of the LED source prior to output or blink control. 0b = Do not invert LED source (LED active low). 1b = Invert LED source (LED active High).
LED3_BLINK	31	0b[1]	LED3/LINK1000# Blink

1. These bits are read from the EEPROM.

8.3 Internal Packet Buffer Size Registers

The following registers define the size of the on-chip receive and transmit buffers used to receive and transmit packets. The overall available internal buffer size in the I350 for all ports is 144 KB for receive buffers and 80 KB for transmit Buffers. Disabled ports memory can be shared between active ports and sharing can be asymmetric. The default buffer size for each port is loaded from the EEPROM on initialization.

8.3.1 Internal Receive Packet Buffer Size - IRPBS (0x2404; RO)

Field	Bit(s)	Initial Value	Description
RXPbsize[1]	3:0	0x0	Receive internal buffer size: 0x0 - 36 KB 0x1 - 72 KB 0x2 - 144 KB 0x3 - 1 KB 0x4 - 2 KB 0x5 - 4 KB 0x6 - 8 KB 0x7 - 16 KB 0x8 - 35 KB 0x9 - 70 KB 0xA - 140 KB 0xB:0xF - reserved ***Notes:*** 1. When 4 ports are active maximum buffer size can be 36 KB. When 2 ports are active maximum buffer size can be 72 KB. When only a single port is active maximum buffer size can be 144 KB. For further information, refer to Section 7.1.3.2. 2. Values bellow 35 KB should be used for diagnostic purposes only. 3. When port is disabled for both PCIe and Management access, the buffer size allocated to the port is 0 Bytes. Available internal memory can be used by other ports.4. When BMC to Host traffic is enabled maximum available receive buffer for all ports is 140 KB. 4. Field loaded from EEPROM following Power-up, PCIe reset and software reset.
Reserved	31:4	0x0	Reserved Write 0, ignore on read.

1. Value loaded from *Initialization Control 4* EEPROM word.In Dual port SKUs, the NVM default should be set to 1 (72 KB).

8.3.2 Internal Transmit Packet Buffer Size - ITPBS (0x3404; RO)

Field	Bit(s)	Initial Value	Description
TXPbsize[1]	3:0	0x0	Transmit internal buffer size: 0x0 - 20 KB 0x1 - 40 KB 0x2 - 80 KB 0x3 - 1 KB 0x4 - 2 KB 0x5 - 4 KB 0x6 - 8 KB 0x7 - 16 KB 0x8 - 19 KB 0x9 - 38 KB 0xA - 76 KB 0xB:0xF - reserved ***Notes:*** 1. When 4 ports are active maximum buffer size can be 20KB. When only 2 ports are active maximum buffer size is 40KB. When only a single port is active maximum buffer size is 80KB. For further information, refer to Section 7.2.1.2. 2. Values bellow 20 KB should be used for diagnostic purposes only. 3. When port is disabled for both PCIe and management access, the buffer size allocated to the port is 0 Bytes. Available internal memory can be used by other ports.4. When BMC to Host traffic is enabled maximum available buffers for all ports is 126 KB. 4. Transmit Internal Buffer size should be greater than the maximum transmit packet size defined in the *DTXMXPKTSZ* register. 5. Field loaded from EEPROM following Power-up, PCIe reset and software reset.
Reserved	31:4	0x0	Reserved Write 0, ignore on read.

1. Value loaded from Initialization Control 4 EEPROM word.In Dual port SKUs, the NVM default should be set to 1 (40 KB).

8.4 EEPROM/Flash Register Descriptions

8.4.1 EEPROM/Flash Control Register - EEC (0x0010; R/W)

This register provides software direct access to the EEPROM. Software can control the EEPROM by successive writes to this register. Data and address information is clocked into the EEPROM by software toggling the *EE_SK* and *EE_DI* bits (0 and 2) of this register with *EE_CS* set to 0b. Data output from the EEPROM is latched into the *EE_DO* bit (bit 3) via the internal 62.5 MHz clock and can be accessed by software via reads of this register.

Field	Bit(s)	Initial Value	Description
EE_SK	0	0b	Clock input to the EEPROM When *EE_GNT* = 1b, the *EE_SK* output signal is mapped to this bit and provides the serial clock input to the EEPROM. Software clocks the EEPROM via toggling this bit with successive writes.
EE_CS	1	1b	Chip select input to the EEPROM When *EE_GNT* = 1b, the *EE_CS* output signal is mapped to the chip select of the EEPROM device. Software enables the EEPROM by writing a 0b to this bit.
EE_DI	2	1b	Data input to the EEPROM When *EE_GNT* = 1b, the *EE_DI* output signal is mapped directly to this bit. Software provides data input to the EEPROM via writes to this bit.
EE_DO (RO)	3	X[1]	Data output bit from the EEPROM The *EE_DO* input signal is mapped directly to this bit in the register and contains the EEPROM data output. This bit is RO from a software perspective; writes to this bit have no effect.
FWE	5:4	01b	Flash Write Enable Control These two bits, control whether writes to Flash memory are allowed. 00b = Flash erase (along with bit 31 in the *FLA* register). 01b = Flash writes disabled. 10b = Flash writes enabled. 11b = Reserved.
EE_REQ	6	0b	Request EEPROM Access The software must write a 1b to this bit to get direct EEPROM access. It has access when *EE_GNT* is 1b. When the software completes the access it must write a 0b.
EE_GNT	7	0b	Grant EEPROM Access When this bit is 1b the software can access the EEPROM using the SK, CS, DI, and DO bits.
EE_PRES (RO)	8	X	EEPROM Present and Signature is valid This bit indicates that an EEPROM is present and the value of the *Signature* field in the *EEPROM Sizing and Protected Fields* EEPROM word (Word 0x12) is 01b 0b = Signature field invalid 1b = EEPROM present and signature is valid.
Auto_RD (RO)	9	0b	EEPROM Auto Read Done When set to 1b, this bit indicates that the auto read by hardware from the EEPROM is done. This bit is also set when the EEPROM is not present or when its signature is not valid.
EE_ADDR_SIZE (RO)	10	1b	EEPROM Address Size This field defines the address size of the EEPROM. This bit is set by the EEPROM size auto-detect mechanism. If no EEPROM is present or the signature is not valid, a 16-bit address is assumed. 0b = 8- and 9-bit. 1b = 16-bit.
EE_SIZE (RO)	14:11[2]	0111b	EEPROM Size This field defines the size of the EEPROM: Field Value EEPROM Size EEPROM Address Size 0000b - 0110b Reserved 0111b 16 Kbytes 2 bytes 1000b 32 Kbytes 2 bytes

Field	Bit(s)	Initial Value	Description
EE_BLOCKED (RO)	15	0b	EEPROM access blocked EEPROM Bit Banging access blocked - Bit is set by HW when detecting an EEPROM access violation during bit banging access using the *EEC* register or detecting an EEPROM access violation when accessing the EPROM using the *EERD* register. When bit is set further Bit Banging operations from the function are disabled until bit is cleared. Type of violations that can cause the bit to be set are write to read-only sections, access to a hidden area or any other EEPROM protection violation detected. ***Note:*** Bit is cleared by write one to the *EEC.EE_CLR_ERR* bit.
EE_ABORT (RO)	16	0b	EEPROM access Aborted Bit is set by HW when EEPROM access was aborted due to deadlock avoidance, management reset or EEPROM reset via CTRL_EXT.EE_RST. When bit is set further Bit Banging operations from the Function are disabled until bit is cleared. ***Note:*** Bit is cleared by write one to the EEC.EE_CLR_ERR bit.
EE_RD_TIMEOUT (RO)	17	0b	EERD access timeout When bit is set to 1b indicates the EEPROM access via EERD register timed out while trying to read EEPROM status (Can occur when no EEPROM exists). ***Note:*** Bit is cleared by write one to the *EEC.EE_CLR_ERR* bit.
EE_CLR_ERR (SC)	18	0b	Clear EEPROM Access Error A write 1b to the EE_CLR_ERR bit clears the *EEC.EE_ABORT* bit, *EE_BLOCKED* bit and *EE_RD_TIMEOUT* bit. ***Note:*** Clearing the *EEC.EE_ABORT* bit and *EE_BLOCKED* bit enables further Bit Banging access to the EEPROM from the function.
EE_DET (RO)	19	X	EEPROM Detected ***Note:*** Bit is set to 1b when EEPROM responded correctly to a get status opcode following power-up.
Reserved	31:20	0x0	Reserved Write 0 ignore on read.

1. Value depends on voltage level on EE_DO pin following initialization
2. These bits are read from the EEPROM.

8.4.2 EEPROM Read Register - EERD (0x0014; RW)

This register is used by software to cause the I350 to read individual words in the EEPROM. To read a word, software writes the address to the *Read Address* field and simultaneously writes a 1b to the *Start Read* field. The I350 reads the word from the EEPROM and places it in the *Read Data* field, setting the *Read Done* field to 1b. Software can poll this register, looking for a 1b in the *Read Done* field, and then using the value in the *Read Data* field.

When this register is used to read a word from the EEPROM, that word does not influence any of the I350's internal registers even if it is normally part of the auto-read sequence.

Note: The EERD register can access only the first 32 KB of the NVM.

Note:

Field	Bit(s)	Initial Value	Description
START	0	0b	Start Read Writing a 1b to this bit causes the EEPROM to read a (16-bit) word at the address stored in the EE_ADDR field and then storing the result in the EE_DATA field. This bit is self-clearing.
DONE (RO)	1	0b	Read Done Set to 1b when the EEPROM read completes. Set to 0b when the EEPROM read is not completed. Writes by software are ignored. Reset by setting the START bit.
ADDR	15:2	0x0	Read Address This field is written by software along with *Start Read* to indicate the word to read.
DATA (RO)	31:16	X	Read Data. Data returned from the EEPROM read.

8.4.3 Flash Access - FLA (0x001C; R/W)

This register provides software direct access to the Flash. Software can control the Flash by successive writes to this register. Data and address information is clocked into the Flash by software toggling the *FL_SCK* bit (bit 0) of this register with *FL_CE* set to 0b. Data output from the Flash is latched into the *FL_SO* bit (bit 3) of this register via the internal 125 MHz clock and can be accessed by software via reads of this register.

Field	Bit(s)	Initial Value	Description
FL_SCK	0	0b	Clock Input to the Flash When *FL_GNT* is 1b, the *FL_SCK* out signal is mapped to this bit and provides the serial clock input to the Flash device. Software clocks the Flash memory via toggling this bit with successive writes.
FL_CE	1	1b	Chip Select Input to the Flash When *FL_GNT* is 1b, the *FL_CE* output signal is mapped to the chip select of the Flash device. Software enables the Flash by writing a 0b to this bit.
FL_SI	2	1b	Data Input to the Flash When *FL_GNT* is 1b, the *FL_SI* output signal is mapped directly to this bit. Software provides data input to the Flash via writes to this bit.
FL_SO	3	X	Data Output Bit from the Flash The FL_SO input signal is mapped directly to this bit in the register and contains the Flash memory serial data output. This bit is read only from the software perspective — writes to this bit have no effect.
FL_REQ	4	0b	Request Flash Access The software must write a 1b to this bit to get direct Flash memory access. It has access when *FL_GNT* is 1b. When the software completes the access it must write a 0b.
FL_GNT	5	0b	Grant Flash Access When this bit is 1b, the software can access the Flash memory using the *FL_SCK*, *FL_CE*, *FL_SI*, and *FL_SO* bits.
FLA_add_size	6	0b	Flash Address Size 0b - Flash devices are accessed using 2 bytes of address. 1b - Flash devices (including 64 KB) are accessed using 3 bytes of the address. ***Notes:*** 1. If this bit is set by one of the functions, it is also reflected in all other functions. 2. If value of *BARCTRL.FLSize* field is greater than 0x0, bit is read as 1b.

Field	Bit(s)	Initial Value	Description
FLA_ABORT (RO)	7	0b	Flash Access Aborted Bit is set by HW when Flash access was aborted due to deadlock avoidance. When bit is set further Flash Bit Banging access from the function are blocked. ***Note:*** Bit is cleared by write 1b to the FLA.FLA_CLR_ERR bit.
FLA_CLR_ERR (SC)	8	0b	Clear Flash Access Error A write 1b to the FLA_CLR_ER bit clears the FLA.FLA_ABORT bit and enables further Bit Banging access to the Flash from the function.
Reserved	28:9	0x0	Reserved Write 0 ignore on read.
FL_BAR_WR (RO)	29	0b	Flash Write via BAR in Progress This bit is set to 1b while a write to the Flash memory is in progress or is pending as a result of a direct Memory access (not bit banging access). When this bit is clear (read as 0b) software can initiate a byte write operation to the Flash device.
FL_BUSY (RO)	30	0b	Flash Busy When set to 1b indicates that a Flash memory access is in progress.
FL_ER (SC)	31	0b	Flash Erase Command When bit is set to 1b an erase command is sent to the Flash component only if the *EEC.FWE* field is 00b (Flash Erase). This bit is automatically cleared when flash erase has completed.

8.4.4 Flash Opcode - FLASHOP (0x103C; R/W)

This register enables the host or the firmware to define the op-code used in order to erase a sector of the flash or the complete flash. This register is reset only at power on assertion.

This register is common to all ports and manageability. Register should be programmed according to the parameters of the flash used.

Note: The default values fit to Atmel* Serial Flash Memory devices.
The flash device erase opcode default (0x62) matches older Atmel Flash Memory devices. For newer Atmel devices or for SST devices a value of 0x60 should be used.

Field	Bit(s)	Initial Value	Description
DERASE	7:0	0x0062	Flash Device Erase Instruction The op-code for the Flash erase instruction.
SERASE	15:8	0x0052	Flash Block Erase Instruction The op-code for the Flash block erase instruction. Relevant only to Flash access by manageability.
Reserved	31:16	0x0	Reserved Write 0 ignore on read.

8.4.5 EEPROM Auto Read Bus Control - EEARBC (0x1024; R/W)

In EEPROM-less implementations, this register is used to program the I350 the same way it should be programmed if an EEPROM was present. Refer to Section 3.3.1.8.1 for details of this register usage.

This register is common to all functions and should be accessed only following access coordination with the other ports.

Field	Bit(s)	Initial Value	Description
VALID_CORE0	0	0b	Valid Write Active to Core 0 Write strobe to Core 0. Firmware/software sets this bit for write access to registers loaded from EEPROM words in LAN0 section. Software should clear this bit to terminate the write transaction.
VALID_CORE1	1	0b	Valid Write Active to Core 1 Write strobe to Core 1. Firmware/software sets this bit for write access to registers loaded from EEPROM words in LAN1 section. Software should clear this bit to terminate the write transaction.
VALID_COMMON	2	0b	Valid Write Active to Common Write strobe to Common. Firmware/software sets this bit for write access to registers loaded from EEPROM words that are common to all sections. Software should clear this bit to terminate the write transaction.
VALID_PCIE	3	0b	Valid Write Active to PCIe PHY Write strobe to PCI PHY. Firmware/software sets this bit for write access to registers loaded from EEPROM words pointed by word 0x10 that are directed to the PCIe PHY. Software should clear this bit to terminate the write transaction.
ADDR	12:4	0x0	Write Address This field specifies the address offset of the EEPROM word from the start of the EEPROM Section. Sections supported are: • Common and LAN0 • LAN1 • LAN2 • LAN3
VALID_CORE2	13	0b	Valid Write Active to Core 2 Write strobe to Core 2. Firmware/software sets this bit for write access to registers loaded from EEPROM words in LAN2 section. Software should clear this bit to terminate the write transaction.
VALID_CORE3	14	0b	Valid Write Active to Core 3 Write strobe to Core 3. Firmware/software sets this bit for write access to registers loaded from EEPROM words in LAN3 section. Software should clear this bit to terminate the write transaction.
Reserved	15	0b	Reserved Write 0, ignore on read.
DATA	31:16	0x0	Data written into the EEPROM auto read bus.

1. Not all EEPROM addresses are part of the auto read. By using this register software can write to the hardware registers that are configured during auto read only.
2. Host access via *EEARBC* can be done only when no EEPROM presence is detected. Management can access the internal registers via *EEARBC* also when EEPROM presence is detected and EEPROM load is done.

8.4.6 Management-EEPROM CSR I/F

The following registers are reserved for Firmware access to the EEPROM and are not writable by the host.

8.4.6.1 Management EEPROM Control Register - EEMNGCTL (0x1010; RW)

Field	Bit(s)	Initial Value	Description
ADDR	14:0	0x0	Address - This field is written by MNG along with Start Read or Start write to indicate the EEPROM word address to read or write.
START	15	0b	Start - Writing a 1b to this bit causes the EEPROM to start the read or write operation according to the write bit. ***Note:*** Bit is not cleared by Firmware reset.
WRITE	16	0b	Write - This bit tells the EEPROM if the current operation is read or write: 0b = read 1b = write
EEBUSY	17	0b	EEPROM Busy - This bit indicates that the EEPROM is busy processing an EEPROM transaction and EEPROM access will be delayed.
CFG_DONE 0[1]	18	0b	Configuration cycle is done for port 0 – This bit indicates that configuration cycle (configuration of SerDes, PHY, PCIe and PLLs) is done for port 0. This bit is set to 1b to indicate configuration done, and cleared by hardware on any of the reset sources that causes initialization of the PHY. Note: Port 0 driver should not try to access the PHY for configuration before this bit is set.
CFG_DONE 1[1]	19	0b	Configuration cycle is done for port 1 – This bit indicates that configuration cycle (configuration of SerDes, PHY, PCIe and PLLs) is done for port 1. This bit is set to 1b to indicate configuration done, and cleared by hardware on any of the reset sources that cause initialization of the PHY. Note: Port 1 driver should not try to access the PHY for configuration before this bit is set.
CFG_DONE 2[1]	20	0b	Configuration cycle is done for port 2 – This bit indicates that the configuration cycle (configuration of SerDes, PCIe and PLLs) is done for port 2. This bit is set to 1b to indicate configuration done, and cleared by hardware on any of the reset sources that cause initialization of the PHY. Note: Port 2 driver should not try to access the PHY for configuration before this bit is set.
CFG_DONE 3[1]	21	0b	Configuration cycle is done for port 3 – This bit indicates that the configuration cycle (configuration of SerDesPHY, PCIe and PLLs) is done for port 3. This bit is set to 1b to indicate configuration done, and cleared by hardware on any of the reset sources that cause initialization of the PHY. Note: Port 3 driver should not try to access the PHY for configuration before this bit is set.
Reserved	28:22	0x0	Reserved Write 0, ignore on read.
EEMNGCTL_CLR_ERR (SC)	29	0b	Clear Timeout Error A write 1b to the *EEMNGCTL.EEMNGCTL_CLR_ERR* bit clears the error reported in the *EEMNGCTL.TIMEOUT* bit.
TIMEOUT	30	0b	When bit is set to 1b indicates that a transaction timed out while trying to read the EEPROM status (Occurs when no EEPROM exists). ***Notes:*** 1. To clear the bit Firmware should write 1b to the *EEMNGCTL.EEMNGCTL_CLR_ERR* bit. 2. Bit is not cleared by Firmware reset.
DONE	31	1b	Transaction Done - This bit is cleared after Start Write or Start Read bit is set by the MNG and is set back again when the EEPROM write or read transaction is done. ***Note:*** Bit is not cleared by Firmware reset.

1. Bit relates to physical port. If LAN Function Swap (*FACTPS.LAN Function Sel* = 1) is done, Software should poll *CFG_DONE* bit of original port to detect end of PHY configuration operation.

8.4.6.2 Management EEPROM Read/Write Data - EEMNGDATA (0x1014; RW)

Field	Bit(s)	Initial Value	Description
WRDATA	15:0	0x0	Write Data Data to be written to the EEPROM.
RDDATA (RO)	31:16	–	Read Data Data returned from the EEPROM read.

8.5 Flow Control Register Descriptions

8.5.1 Flow Control Address Low - FCAL (0x0028; RO)

Flow control packets are defined by 802.3X to be either a unique multicast address or the station address with the Ether Type field indicating PAUSE. The FCA registers provide the value hardware uses to compare incoming packets against, to determine that it should PAUSE its output.

The FCAL register contains the lower bits of the internal 48-bit Flow Control Ethernet address. All 32 bits are valid. Software can access the High and Low registers as a register pair if it can perform a 64-bit access to the PCIe bus. The complete flow control multicast address is: 0x01_80_C2_00_00_01; where 0x01 is the first byte on the wire, 0x80 is the second, etc.

Note: Any packet matching the contents of {*FCAH*, *FCAL*, *FCT*} when *CTRL.RFCE* is set is acted on by the I350. Whether flow control packets are passed to the host (software) depends on the state of the *RCTL.DPF* bit and whether the packet matches any of the normal filters.

Field	Bit(s)	Initial Value	Description
FCAL	31:0	0x00C28001	Flow Control Address Low

8.5.2 Flow Control Address High - FCAH (0x002C; RO)

This register contains the upper bits of the 48-bit Flow Control Ethernet address. Only the lower 16 bits of this register have meaning. The complete Flow Control address is {*FCAH*, *FCAL*}.

The complete flow control multicast address is: 0x01_80_C2_00_00_01; where 0x01 is the first byte on the wire, 0x80 is the second, etc.

Field	Bit(s)	Initial Value	Description
FCAH	15:0	0x0100	Flow Control Address High Should be programmed with 0x01_00.
Reserved	31:16	0x0	Reserved Write 0, ignore on read.

8.5.3 Flow Control Type - FCT (0x0030; R/W)

This register contains the type field that hardware matches to recognize a flow control packet. Only the lower 16 bits of this register have meaning. This register should be programmed with 0x88_08. The upper byte is first on the wire *FCT*[15:8].

Field	Bit(s)	Initial Value	Description
FCT	15:0	0x8808	Flow Control Type
Reserved	31:16	0x0	Reserved Write 0, ignore on read.

8.5.4 Flow Control Transmit Timer Value - FCTTV (0x0170; R/W)

The 16-bit value in the *TTV* field is inserted into a transmitted frame (either XOFF frames or any PAUSE frame value in any software transmitted packets). It counts in units of slot time of 64 bytes. If software needs to send an XON frame, it must set *TTV* to 0 prior to initiating the PAUSE frame.

Field	Bit(s)	Initial Value	Description
TTV	15:0	X	Transmit Timer Value
Reserved	31:16	0b	Reserved Write 0, ignore on read.

8.5.5 Flow Control Receive Threshold Low - FCRTL0 (0x2160; R/W)

This register contains the receive threshold used to determine when to send an XON packet The complete register reflects the threshold in units of bytes. The lower 4 bits must be programmed to 0b (16 byte granularity). Software must set *XONE* to enable the transmission of XON frames. Each time hardware crosses the receive-high threshold (becoming more full), and then crosses the receive-low threshold and *XONE* is enabled (1b), hardware transmits an XON frame.

When XONE is set, the *RTL* field should be programmed to at least 3b (at least 48 bytes).

Flow control reception/transmission are negotiated capabilities by the Auto-Negotiation process. When the I350 is manually configured, flow control operation is determined by the *CTRL.RFCE* and *CTRL.TFCE* bits.

Field	Bit(s)	Initial Value	Description
Reserved	3:0	0000b	Reserved Write 0, ignore on read.

RTL	16:4	0x0	Receive Threshold Low. FIFO low water mark for flow control transmission when Transmit Flow control is enabled (*CTRL.TFCE* = 1b). An XON packet is sent if the occupied space in the packet buffer is smaller or equal than this watermark. This field is in 16 bytes granularity. This value should be set to at least 0x5 if transmit flow control is enabled
Reserved	30:17	0x0	Reserved Write 0, ignore on read.
XONE	31	0b	XON Enable 0b = Disabled. 1b = Enabled. If FCRTL0.XONE is 1, the minimum value allowed in FCRTL0.RTL is 3 (48 bytes).

8.5.6 Flow Control Receive Threshold High - FCRTH0 (0x2168; R/W)

This register contains the receive threshold used to determine when to send an XOFF packet. The complete register reflects the threshold in units of bytes. This value must be at maximum 48 bytes less than the maximum number of bytes allocated to the Receive Packet Buffer (*IRPBS.RXPbsize*), and the lower 4 bits must be programmed to 0b (16 byte granularity). The value of *RTH* should also be bigger than *FCRTL0.RTL*. Each time the receive FIFO reaches the fullness indicated by *RTH*, hardware transmits a PAUSE frame if the transmission of flow control frames is enabled.

Flow control reception/transmission are negotiated capabilities by the Auto-Negotiation process. When the I350 is manually configured, flow control operation is determined by the *CTRL.RFCE* and *CTRL.TFCE* bits.

Field	Bit(s)	Initial Value	Description
Reserved	3:0	000b	Reserved Write 0, ignore on read.
RTH	17:4	0x0	Receive Threshold High FIFO high water mark for flow control transmission when Transmit Flow control is enabled (*CTRL.TFCE* = 1b). An XOFF packet is sent if the occupied space in the packet buffer is bigger or equal than this watermark. This field is in 16 bytes granularity. Refer to Section 3.7.5.3.1 for calculation of *FCRTH0.RTH* value. ***Notes:*** 1. When in DMA coalescing operation and internal Transmit buffer is empty threshold high value defined in *FCRTC.RTH_Coal* is used instead of the *FCRTH0.RTH* value to allow increase of Receive Threshold High value by maximum supported Jumbo frame size. 2. Value programmed should be greater than maximum packet size.
Reserved	31:18	0x0	Reserved Write 0, ignore on read.

8.5.7 Flow Control Refresh Threshold Value - FCRTV (0x2460; R/W)

Field	Bit(s)	Initial Value	Description
FC_refresh_th	15:0	0x0	Flow Control Refresh Threshold This value indicates the threshold value of the flow control shadow counter when Transmit Flow control is enabled (*CTRL.TFCE* = 1b). When the counter reaches this value, and the conditions for PAUSE state are still valid (buffer fullness above low threshold value), a PAUSE (XOFF) frame is sent to link partner. If this field contains zero value, the Flow Control Refresh is disabled.
Reserved	31:16	-	Reserved Write 0, ignore on read.

8.5.8 Flow Control Status - FCSTS0 (0x2464; RO)

This register describes the status of the flow control machine.

Field	Bit(s)	Initial Value	Description
Flow_control state	0	0b	Flow control state machine signal 0b = XON 1b = XOFF
Above high	1	0b	The size of data in the memory is above the high threshold
Below low	2	1b	The size of data in the memory is below the low threshold
Reserved	15:3	0x0	Reserved Write 0, ignore on read.
Refresh counter	31:16	0x0	Flow control refresh counter

8.6 PCIe Register Descriptions

8.6.1 PCIe Control - GCR (0x5B00; RW)

Field	Bit(s)	Initial Value	Description
Reserved	1:0	0x0	Reserved. Write 0, ignore on read.
Discard on BME de-assert	2	1b	When set, on BME fall, the PCIE discards all requests of this function
Reserved	8:3	0x0	Reserved. Write 0, ignore on read.
Completion Timeout resend enable	9	1b[1]	When set, enables a resend request after the completion timeout expires. 0b = Do not resend request after completion timeout 1b = Resend request after completion timeout. Note: This field is loaded from the “Completion Timeout Resend” bit in the EEPROM.

Field	Bit(s)	Initial Value	Description
Reserved	10	0b	Reserved Write 0, ignore on read.
Number of resends	12:11	11b	The number of resends in case of Timeout or Poisoned.
Reserved	17:13	0x0	Reserved Write 0, ignore on read.
PCIe Capability Version (RO)	18	1b[2]	Reports the PCIe capability version supported. 0b = Capability version: 0x1. 1b = Capability version: 0x2.
Reserved	30:19	0x0	Reserved.
DEV_RST In progress	31	0b	Device reset in progress Bit is set following Device Reset assertion (*CTRL.DEV_RST* = 1) until no pending requests exist in PCI-E. Software driver should wait for bit to be cleared before re-initializing the port (Refer to Section 4.3.1).

1. Loaded from *PCIe Completion Timeout Configuration* EEPROM word (word 0x15).
2. The default value for this field is read from the *PCIe Init Configuration 3* EEPROM word (address 0x1A) bits 11:10. If these bits are set to 10b, then this field is set to 1, otherwise field is reset to zero.

8.6.2 PCIe Statistics Control #1 - GSCL_1 (0x5B10; RW)

Field	Bit(s)	Initial Value	Description
GIO_COUNT_EN_0	0	0b	Enable PCIe Statistic Counter Number 0.
GIO_COUNT_EN_1	1	0b	Enable PCIe Statistic Counter Number 1.
GIO_COUNT_EN_2	2	0b	Enable PCIe Statistic Counter Number 2.
GIO_COUNT_EN_3	3	0b	Enable PCIe Statistic Counter Number 3.
LBC Enable 0	4	0b	When set, statistics counter 0 operates in Leaky Bucket mode.
LBC Enable 1	5	0b	When set, statistics counter 1 operates in Leaky Bucket mode.
LBC Enable 2	6	0b	When set, statistics counter 2 operates in Leaky Bucket mode.
LBC Enable 3	7	0b	When set, statistics counter 3 operates in Leaky Bucket mode.
Reserved	26:8	0b	Reserved. Write 0, ignore on read.
GIO_COUNT_TEST	27	0b	Test Bit Forward counters for testability.
GIO_64_BIT_EN	28	0b	Enable two 64-bit counters instead of four 32-bit counters.
GIO_COUNT_RESET	29	0b	Reset indication of PCIe statistical counters.
GIO_COUNT_STOP	30	0b	Stop indication of PCIe statistical counters.
GIO_COUNT_START	31	0b	Start indication of PCIe statistical counters.

8.6.3 PCIe Statistics Control #2 - GSCL_2 (0x5B14; RW)

This register configures the events counted by the GSCN_0, GSCN_1, GSCN_2 and GSCN_3 counters.

Note:

Field	Bit(s)	Initial Value	Description
GIO_EVENT_NUM_0	7:0	0x0	Event type that counter 0 (GSCN_0) counts.
GIO_EVENT_NUM_1	15:8	0x0	Event type that counter 1 (GSCN_1) counts.
GIO_EVENT_NUM_2	23:16	0x0	Event type that counter 2 (GSCN_2) counts.
GIO_EVENT_NUM_3	31:24	0x0	Event type that counter 3 (GSCN_3) counts.

Table 8-9 lists the encoding of possible event types counted by GSCN_0, GSCN_1, GSCN_2 and GSCN_3.

Table 8-9 PCIe Statistic Events Encoding

Transaction layer Events	Event Mapping (Hex)	Description
Bad TLP from LL	0x0	Each cycle, the counter increase in 1, if bad TLP is received (bad CRC, error reported by AL, misplaced special char, reset in middle of received TLP).
Requests that reached timeout	0x10	Number of requests that reached Time Out.
NACK DLLP received	0x20	For each cycle, the counter increase by one, if a message was transmitted.
Replay happened in Retry-Buffer	0x21	Occurs when a replay happened due to timeout (not asserted when replay initiated due to NACK
Receive Error	0x22	Set when one of the following occurs: 1. Decoder error occurred during training in the PHY. It is reported only when training ends. 2. Decoder error occurred during link-up or till the end of the current packet (in case the link failed). This error is masked when entering/exiting EI.
Replay Roll-Over	0x23	Occurs when replay was initiated for more than 3 times [threshold is configurable by the PHY CSRs]
Re-Sending Packets	0x24	Occurs when TLP is resend in case of completion timeout
Surprise Link Down	0x25	Occurs when link is unpredictably down (Not because of reset or DFT)
LTSSM in L0s in both Rx & Tx	0x30	Occurs when LTSSM enters L0s state in both Tx & Rx
LTSSM in L0s in Rx	0x31	Occurs when LTSSM enters L0s state in Rx
LTSSM in L0s in Tx	0x32	Occurs when LTSSM enters L0s state in Tx
LTSSM in L1 active	0x33	Occurs when LTSSM enters L1-Active state (Requested from Host side)
LTSSM in L1 SW	0x34	Occurs when LTSSM enters L1-Switch (Requested from Switch side)
LTSSM in recovery	0x35	Occurs when LTSSM enters Recovery state

8.6.4 PCIe Statistic Control Register #5...#8 - GSCL_5_8 (0x5B90 + 4*n[n=0...3]; RW)

These registers control the operation of the statistical counters GSCN_0, GSCN_1, GSCN_2 and GSCN_3 when operating Leaky Bucket mode:

- GSCL_5 controls operation of GSCN_0.
- GSCL_6 controls operation of GSCN_1.
- GSCL_7 controls operation of GSCN_2.

- GSCL_8 controls operation of GSCN_3.

Field	Bit(s)	Initial Value	Description
LBC threshold n	15:0	0x0	Threshold for the Leaky Bucket Counter n
LBC timer n	31:16	0x0	Time period between decrements of the value in Leaky Bucket Counter n.

8.6.5 PCIe Counter #0 - GSCN_0 (0x5B20; RC)

Field	Bit(s)	Initial Value	Description
EVC	31:0	0x0	Event Counter. Type of event counted is defined by the *GSCL_2.GIO_EVENT_NUM_0* field. Count value does not wrap around and remains stuck at the maximum value of 0xFF...F. Value is cleared by read.

8.6.6 PCIe Counter #1 - GSCN_1 (0x5B24; RC)

Field	Bit(s)	Initial Value	Description
EVC	31:0	0x0	Event Counter. Type of event counted is defined by the *GSCL_2.GIO_EVENT_NUM_1* field. Count value does not wrap around and remains stuck at the maximum value of 0xFF...F. Value is cleared by read.

8.6.7 PCIe Counter #2 - GSCN_2 (0x5B28; RC)

Field	Bit(s)	Initial Value	Description
EVC	31:0	0x0	Event Counter. Type of event counted is defined by the *GSCL_2.GIO_EVENT_NUM_2* field. Count value does not wrap around and remains stuck at the maximum value of 0xFF...F. Value is cleared by read.

8.6.8 PCIe Counter #3 - GSCN_3 (0x5B2C; RC)

Field	Bit(s)	Initial Value	Description
EVC	31:0	0x0	Event Counter. Type of event counted is defined *by the* GSCL_2.GIO_EVENT_NUM_3 field. Count value does not wrap around and remains stuck at the maximum value of 0xFF...F. Value is cleared by read.

8.6.9 Function Active and Power State to MNG - FACTPS (0x5B30; RO)

Field	Bit(s)	Initial Value	Description
Func0 Power State	1:0	00b	Power state indication of Function 0 00b → DR 01b → D0u 10b →D0a 11b → D3
LAN0 Valid	2	0b	LAN 0 Enable When set to 0b, it indicates that the LAN 0 function is disabled. When the function is enabled, the bit is set to 1b. The LAN 0 enable bit is set by the LAN0_DIS_N strapping pin.
Func0 Aux_En	3	0b	Function 0 Auxiliary (AUX) Power PM Enable bit shadow from the configuration space.
Reserved	5:4	0x0	Reserved. Write 0, ignore on read.
Func1 Power State	7:6	00b	Power state indication of Function 1 00b → DR 01b → D0u 10b → D0a 11b → D3
LAN1 Valid	8	0b	LAN 1 Enable When set to 0b, it indicates that the LAN 1 function is disabled. When the function is enabled, the bit is set to 1b. The LAN 1 enable bit is set by the LAN1_DIS_N strapping pin.
Func1 Aux_En	9	0b	Function 1 Auxiliary (AUX) Power PM Enable bit shadow from the configuration space.
Func2 Power State	11:10	00b	Power state indication of Function 2 00b → DR 01b → D0u 10b → D0a 11b → D3
LAN2 Valid	12	0b	LAN 2 Enable When set to 0b, it indicates that the LAN 2 function is disabled. When the function is enabled, the bit is set to 1b. The LAN 2 enable bit is set by the LAN2_DIS_N strapping pin.
Func2 Aux_En	13	0b	Function 2 Auxiliary (AUX) Power PM Enable bit shadow from the configuration space.

Field	Bit(s)	Initial Value	Description
Func3 Power State	15:14	00b	Power state indication of Function 3 00b → DR 01b → D0u 10b → D0a 11b → D3
LAN3 Valid	16	0b	LAN 3 Enable When set to 0b, it indicates that the LAN 3 function is disabled. When the function is enabled, the bit is set to 1b. The LAN 3 enable bit is set by the LAN3_DIS_N strapping pin.
Func3 Aux_En	17	0b	Function 3 Auxiliary (AUX) Power PM Enable bit shadow from the configuration space.
Reserved	28:18	0x0	Reserved Write 0, ignore on read.
MNGCG	29	0b	MNG Clock Gated When set, indicates that the manageability clock is gated.
LAN Function Sel	30[1]	0b	When all LAN ports are enabled and LAN Function Sel = 0b, LAN 0 is routed to PCIe Function 0, LAN 1 is routed to PCIe Function 1, etc. If LAN Function Sel = 1b, LAN 0 is routed to PCIe Function 3, LAN 1 is routed to PCIe Function 2, LAN 2 is routed to PCIe Function 1 and LAN 3 is routed to PCIe Function 0. If a port is disabled a description of the mapping between LAN port and PCIe function can be found in Section 4.4.2. ***Note:*** PCIe Functions mapping of the dual port SKU and 25x25 package SKU behave like the 4 port SKU with LAN 2 and LAN 3 disabled.
PM State Changed (RC)	31	0b	Indication that one or more of the functions power states had changed. This bit is also a signal to the MNG unit to create an interrupt. This bit is cleared on read by Management.

1. This bit is initiated from EEPROM word "*Functions Control*" (0x21)..

8.6.10 Mirrored Revision ID - MREVID (0x5B64; R/W)

Field	Bit(s)	Initial Value	Description
EEPROM RevID	7:0	0x0	Mirroring of the Revision ID modifier loaded from the EEPROM (from word Device Rev ID word, address 0x1E).
Step REV ID	15:8	0x1	Revision ID from function configuration space before NVM overriding. This values is XORed with eprom_rev_id and stored as the actual device revision ID in PCIE config space, address 0x8
Reserved	31:16	0x0	Reserved Write 0, ignore on read.

8.6.11 PCIe Control Extended Register - GCR_EXT (0x5B6C; RW)

Field	Bit(s)	Initial Value	Description
Reserved	3:0	0x0	Reserved Write 0, ignore on read.
APBACD	4	0b	Auto PBA Clear Disable. When set to 1, Software can clear the PBA only by direct write to clear access to the PBA bit. When set to 0, any active PBA entry is cleared on the falling edge of the appropriate interrupt request to the PCIe block. The appropriate interrupt request is cleared when software sets the associated interrupt mask bit in the EIMS (re-enabling the interrupt) or by direct write to clear to the PBA.
Reserved	31:5	0x00	Reserved

8.6.12 PCIe BAR Control - BARCTRL (0x5BFC; R/W) Target

Field	Bit(s)	Initial Value	Description
Reserved	7:0	0x0	Reserved Write 0, ignore on read.
FLSize	10:8	000b[1]	This field indicates the size of the external Flash device equals to 64KB x 2^{FLSize}. Refer to the table below for the usable FLASH size. ***Note:*** Value is loaded from *PCIe Control 2* EEPROM word (Refer to Section 6.2.27).
Reserved	12:11	0x0	Reserved Write 0, ignore on read.
CSRSize	13	0b[1]	The CSRSize and FLSize fields define the usable FLASH size and CSR mapping window size as shown in Table 8-10 below. ***Note:*** Value is loaded from *PCIe Control 2* EEPROM word (Refer to Section 6.2.26).
PREFBAR	14	0b[1]	Prefetchable bit indication in the memory BARs (should be set when 64bit BARs are used) 0 BARs are marked as non prefetchable 1 BARs are marked as prefetchable Note: Value is loaded from EEPROM word *Functions Control*.
BAR32	15	1b[1]	BAR 32bit Enable. When set 32bit BARs are enabled. At 0b 64 bit BAR addressing mode is selected. When *PREFBAR* bit is set *BAR32* bit value should always be 0. Note: Value is loaded from EEPROM word *Functions Control*.
Reserved	31:16	0x0	Reserved Write 0, ignore on read.

1. These bits are loaded from EEPROM.

Table 8-10 Usable FLASH Size and CSR Mapping Window Size

FLSize	CSRSize	Resulted CSR + FLASH BAR Size	Installed FLASH Device	Usable FLASH Space
000b	0	128KB	No Flash	0
000b	1	256KB	64KB	64KB
001b	0	256KB	128KB	128KB
001b	1	n/a	n/a	Reserved
010b	0	256KB	256KB	256KB minus 128KB
010b	1	512KB	256KB	256KB
011b	0	512KB	512KB	512KB minus 128KB
011b	1	1MB	512KB	512KB
100b	0	1MB	1MB	1MB minus 128KB
100b	1	2MB	1MB	1MB
101b	0	2MB	2MB	2MB minus 128KB
101b	1	4MB	2MB	2MB
110b	0	4MB	4MB	4MB minus 128KB
110b	1	8MB	4MB	4MB
111b	0	8MB	8MB	8MB minus 128KB
111b	1	16MB	8MB	8MB

8.7 Semaphore Registers

This section contains registers common to all ports used to coordinate between all functions. The usage of these registers is described in Section 4.6.12

8.7.1 Software Semaphore - SWSM (0x5B50; R/W)

Field	Bit(s)	Initial Value	Description
SMBI (RS)	0	0x0	Software/Software Semaphore Bit This bit is set by hardware when this register is read by the device driver and cleared when the HOST driver writes a 0b to it. The first time this register is read, the value is 0b. In the next read the value is 1b (hardware mechanism). The value remains 1b until the software device driver clears it. This bit can be used as a semaphore between all the device's drivers in the I350. This bit is cleared on PCIe reset.
SWESMBI	1	0x0	Software/Firmware Semaphore bit This bit should be set only by the device driver (read only to firmware). The bit is not set if bit 0 in the FWSM register is set. The device driver should set this bit and then read it to verify that it was set. If it was set, it means that the device driver can access the SW_FW_SYNC register. The device driver should clear this bit after modifying the SW_FW_SYNC register. Notes: • If Software takes ownership of the *SWSM.SWESMBI* bit for a duration longer than 100 mS, Firmware may take ownership of the bit. • Hardware clears this bit on PCIe reset.
Reserved	30:2	0x0	Reserved Write 0, ignore on read.
SWMB_CLR (SC)	31	0b	Software Mailbox clear When bit is set to 1b, the *SWMBWR*, *SWMB0*, *SWMB1*, *SWMB2* and *SWMB3* Software Mailbox registers are reset.

8.7.2 Firmware Semaphore - FWSM (0x5B54; R/WS)

Field[1]	Bit(s)	Initial Value	Description
EEP_FW_Semaphore	0	0b	Software/Firmware Semaphore Firmware should set this bit to 1b before accessing the *SW_FW_SYNC* register. If the software is using the SWSM register and does not lock the SW_FW_SYNC, firmware is able to set this bit to 1b. Firmware should set this bit back to 0b after modifying the *SW_FW_SYNC* register. Note: If Software takes ownership of the *SWSM.SWESMBI* bit for a duration longer than 100 mS, Firmware may take ownership of the bit.
FW_Mode	3:1	0x0	Firmware Mode Indicates the firmware mode as follows: 000b = No MNG. 001b = Reserved. 010b = PT mode. 011b = Reserved. 100b = Reserved. ***Notes:*** if FW_Mode = No MNG, proxy may still be supported if MANC. MPROXYE is set.
Reserved	5:4	00b	Reserved Write 0, ignore on read.

Field[1]	Bit(s)	Initial Value	Description
EEP_Reload_Ind	6	0b	EEPROM reloaded indication Set to 1b after firmware reloads the EEPROM. Cleared by firmware once the "Clear Bit" host command is received from host software.
Reserved	14:7	0x0	Reserved Write 0, ignore on read.
FW_Val_Bit	15	0b	Firmware Valid Bit Hardware clears this bit in reset de-assertion so software can know firmware mode (bits 1-3) bits are invalid. Firmware should set this bit to 1b when it is ready (end of boot sequence).
Reset_Cnt	18:16	0b	Reset Counter Firmware increments the count on every Firmware reset. After 7 Firmware reset events counter stays stuck at 7 and does not wrap around.
Ext_Err_Ind	24:19	0x0	External error indication Firmware writes here the reason that the firmware operation has stopped. For example, EEPROM CRC error, etc. Possible values: 0x00: No Error 0x01: Reserved 0x02: Reserved. 0x03: EEPROM CRC Error in Common Firmware Parameters Module. 0x04: EEPROM CRC error in Pass Through LAN 0 Module. 0x05: EEPROM CRC error in Pass Through LAN 1 Module. 0x06: EEPROM CRC error in Pass Through LAN 2 Module. 0x07: EEPROM CRC error in Pass Through LAN 3 Module. 0x08: EEPROM CRC error in Sideband Configuration Module. 0x09: EEPROM CRC Error in Flexible TCO Filter Configuration Module. 0x0A: EEPROM CRC Error in NC-SI Microcode Download Module. 0x0B: EEPROM CRC Error in NC-SI Configuration Module. 0x0C: EEPROM CRC Error in Traffic Type Parameters Module. 0x0D: EEPROM CRC Error in Inventory NVM Structure Module. 0x0E: EEPROM CRC Error in PHY Configuration structure Module. 0x0F to 0x15: Reserved. 0x16: TLB table exceeded. 0x17: DMA load failed. 0x18: Reserved. 0x19: Flash device not supported. 0x1A: Invalid Flash checksum. 0x1B: Unspecified Error. 0x1C to 0x1F: Reserved. 0x20: EEPROM CRC Error in HW Auto-load. 0x21: No Manageability (No EEPROM) 0x22: TCO isolate mode active. 0x23: Management memory Parity error. 0x24: FW EEPROM Access Failure. 0x25: Other Management Error detected. 0x26 to 0x03F: Reserved ***Note:*** Following error detection and *FWSM.Ext_Err_ind* update, the *ICR.MGMT* bit is set and an interrupt is sent to the Host. However when values of 0x00 or 0x21 are placed in the *FWSM.Ext_Err_ind* field the *ICR.MGMT* bit is not set and an interrupt is not generated.
PCIe_Config_Err_Ind	25	0b	PCIe configuration error indication Set to 1b by firmware when it fails to configure PCIe interface. Cleared by firmware upon successful configuration of PCIe interface.

Field[1]	Bit(s)	Initial Value	Description
PHY_SERDES0_Config_ Err_Ind	26	0b	PHY/SerDes0 configuration error indication Set to 1b by firmware when it fails to configure LAN0 PHY/SerDes. Cleared by firmware upon successful configuration of LAN0 PHY/SerDes.
PHY_SERDES1_Config_ Err_Ind	27	0b	PHY/SerDes1 configuration error indication Set to 1b by firmware when it fails to configure LAN1 PHY/SerDes. Cleared by firmware upon successful configuration of LAN1 PHY/SerDes.
Reserved	28	0b	Reserved. Write 0, ignore on read.
SERDES2_Config_ Err_Ind	29	0b	SerDes2 configuration error indication Set to 1b by firmware when it fails to configure LAN2 SerDes. Cleared by firmware upon successful configuration of LAN2 SerDes.
SERDES3_Config_ Err_Ind	30	0b	SerDes3 configuration error indication Set to 1b by firmware when it fails to configure LAN3 SerDes. Cleared by firmware upon successful configuration of LAN3 SerDes.
Factory MAC address restored	31	0b	This bit is set if the internal Firmware restored the factory MAC address at power up or if the factory MAC address and the regular MAC address were the same. This bit is common to all ports.

Notes:
1. This register should be written only by the manageability firmware. The device driver should only read this register.
2. Firmware ignores the EEPROM semaphore in operating system hung states.
3. Bits 15:0 are cleared on firmware reset.

8.7.3 Software–Firmware Synchronization - SW_FW_SYNC (0x5B5C; RWS)

This register is intended to synchronize between software and firmware. This register is common to all ports.

Note: If Software takes ownership of bits in the *SW_FW_SYNC* register for a duration longer than 1 Second, Firmware may take ownership of the bit.

Field	Bit(s)	Initial Value	Description
SW_EEP_SM	0	0b	When set to 1b, EEPROM access is owned by software
SW_PHY_SM0	1	0b	When set to 1b, SerDes/PHY 0 access is owned by software
SW_PHY_SM1	2	0b	When set to 1b, SerDes/PHY 1 access is owned by software
SW_MAC_CSR_SM	3	0b	When set to 1b, software owns access to shared CSRs
SW_FLASH_SM	4	0	When set to 1b, software owns access to the flash.
SW_PHY_SM2	5	0b	When set to 1b, SerDes/PHY 2 access is owned by software
SW_PHY_SM3	6	0b	When set to 1b, SerDes/PHY 3 access is owned by software
SW_PWRTS_SM	7	0b	When set to 1b Thermal Sensor Registers are owned by software driver.
SW_MB_SM	8	0b	When Set to 1b, *SWMBWR* mailbox write register, is owned by software driver.
Reserved	9	0b	Reserved. Write 0, ignore on read

Field	Bit(s)	Initial Value	Description
SW_MNG_SM	10	0b	When set to 1b Management Host interface is owned by port driver. Bit can be used by port driver when updating teaming or proxying information.
Reserved	15:11	0b	Reserved. Write 0, ignore on read
FW_EEP_SM	16	0b	When set to 1b, EEPROM access is owned by firmware
FW_PHY_SM0	17	0b	When set to 1b, PHY 0 access is owned by firmware
FW_PHY_SM1	18	0b	When set to 1b, PHY 1 access is owned by Firmware
FW_MAC_CSR_SM	19	0b	When set to 1b, Firmware owns access to shared CSRs
FW_FLASH_SM	20	0	When set to 1b, Firmware owns access to the flash.
FW_PHY_SM2	21	0b	When set to 1b, PHY 2 access is owned by Firmware.
FW_PHY_SM3	22	0b	When set to 1b, PHY 3 access is owned by Firmware.
FW_PWRTS_SM	23	0b	When set to 1b Thermal Sensor Registers are owned by Firmware.
Reserved	31:24	0x0	Reserved Write 0, ignore on read.

Reset conditions:

- The software-controlled bits 15:0 are reset as any other CSR on global resets, D3hot exit and Forced TCO. Software is expected to clear the bits on entry to D3 state.
- The Firmware controlled bits (bits 31:16) are reset on LAN_PWR_GOOD (power-up) and firmware reset.

8.7.4 Software Mailbox Write - SWMBWR (0x5B04; R/W)

Field	Bit(s)	Initial Value	Description
Mailbox	31:0	0x0	Message sent from driver to the other drivers. The interpretation of this field is defined by the drivers. ***Note:*** This register is reset by power on reset and by a write 1b to the *SWSM.SWMB_CLR* field.

8.7.5 Software Mailbox 0 - SWMB0 (0x5B08; RO)

Field	Bit(s)	Initial Value	Description
Mailbox	31:0	0x0	Message sent from the driver of port 0 via write to the *SWMBWR* register. The interpretation of this field is defined by the drivers. ***Note:*** This register is reset by power on reset and by a write 1b to the *SWSM.SWMB_CLR* field.

8.7.6 Software Mailbox 1 - SWMB1 (0x5B0C; RO)

Field	Bit(s)	Initial Value	Description
Mailbox	31:0	0x0	Message sent from the driver of port 1 via write to the *SWMBWR* register. The interpretation of this field is defined by the drivers. ***Note:*** This register is reset by power on reset and by a write 1b to the *SWSM.SWMB_CLR* field.

8.7.7 Software Mailbox 2 - SWMB2 (0x5B18; RO)

Field	Bit(s)	Initial Value	Description
Mailbox	31:0	0x0	Message sent from the driver of port 2 via write to the *SWMBWR* register. The interpretation of this field is defined by the drivers. ***Note:*** This register is reset by power on reset and by a write 1b to the *SWSM.SWMB_CLR* field.

8.7.8 Software Mailbox 3 - SWMB3 (0x5B1C; RO)

Field	Bit(s)	Initial Value	Description
Mailbox	31:0	0x0	Message sent from the driver of port 3 via write to the *SWMBWR* register. The interpretation of this field is defined by the drivers. ***Note:*** This register is reset by power on reset and by a write 1b to the *SWSM.SWMB_CLR* field.

8.8 Interrupt Register Descriptions

8.8.1 PCIe Interrupt Cause - PICAUSE (0x5B88; RW1/C)

Field	Bit(s)	Init.	Description
CA	0	0b	PCI Completion Abort exception issued.
UA	1	0b	Unsupported IO address exception. Bit is set when: IO access to address outside of the allocated address space is detected. IO access to non-CSR address space (Flash access). Write access to IOADDR with partial Byte-Enable.
BE	2	0b	Wrong Byte-Enable exception in the FUNC unit.
TO	3	0b	PCI Timeout exception in the FUNC unit.

Field	Bit(s)	Init.	Description
BMEF	4	0b	Asserted when bus-master-enable (BME) of the PF is de-asserted.
ABR	5	0b	PCI Completer Abort Received. PCI Completer Abort (CA) or Unsupported Request (UR) received (Set on reception of CA or UR). Note: When bit is set all PCIe master activity is stopped. Software should issue a software (*CTRL.RST*) reset to enable PCIe activity on all ports.
Reserved	31:6	0x0	Reserved Write 0, ignore on read.

8.8.2 PCIe Interrupt Enable - PIENA (0x5B8C; R/W)

Field	Bit(s)	Init.	Description
CA	0	0b	When set to 1 the PCI Completion Abort interrupt is enabled.
UA	1	0b	When set to 1 the Unsupported IO address interrupt is enabled.
BE	2	0b	When set to 1 the Wrong Byte-Enable interrupt is enabled.
TO	3	0b	When set to 1 the PCI Timeout interrupt is enabled.
BMEF	4	0b	When set to 1 the Bus Master Enable interrupt is enabled.
ABR	5	0b	When set to 1 the PCI completion abort received interrupt is enabled.
Reserved	31:6	0x0	Reserved Write 0, ignore on read.

8.8.3 Extended Interrupt Cause - EICR (0x1580; RC/W1C)

This register contains the frequent interrupt conditions for the I350. Each time an interrupt causing event occurs, the corresponding interrupt bit is set in this register. An interrupt is generated each time one of the bits in this register is set and the corresponding interrupt is enabled via the Interrupt Mask Set/Read register. The interrupt might be delayed by the selected Interrupt Throttling register.

Note that the software device driver cannot determine from the RxTxQ bits what was the cause of the interrupt. The possible causes for asserting these bits are:

- Receive Descriptor Write Back, Receive Descriptor Minimum Threshold hit, low latency interrupt for Rx, Transmit Descriptor Write Back.

Writing a 1b to any bit in the register clears that bit. Writing a 0b to any bit has no effect on that bit.

Register bits are cleared on register read if GPIE.Multiple_MSIX = 0

Auto clear can be enabled for any or all of the bits in this register.

Table 8-11 EICR Register Bit Description - non MSI-X mode (GPIE.Multiple_MSIX = 0)

Field	Bit(s)	Initial Value	Description
RxTxQ	7:0	0x0	Receive/Transmit Queue Interrupts One bit per queue or a bundle of queues, activated on receive/transmit queue events for the corresponding bit, such as: Receive Descriptor Write Back, Receive Descriptor Minimum Threshold hit Transmit Descriptor Write Back. The mapping of actual queue to the appropriate RxTxQ bit is according to the IVAR registers.
Reserved	29:8	0x0	Reserved Write 0, ignore on read.
TCP Timer	30	0b	TCP Timer Expired Activated when the TCP timer reaches its terminal count.
Other Cause	31	0b	Interrupt Cause Active Activated when any bit in the ICR register is set.

Note: Bits are not reset by Device Reset (*CTRL.DEV_RST*).

Table 8-12 EICR Register Bit Description - MSI-X mode (GPIE.Multiple_MSIX = 1)

Field	Bit(s)	Initial Value	Description
MSIX	24:0	0x0	Indicates an interrupt cause mapped to MSI-X vectors 24:0 Note: Bits are not reset by Device Reset (*CTRL.DEV_RST*).
Reserved	31:25	0x0	Reserved Write 0, ignore on read.

Note: In IOV mode bit 0 of this vector is available for the PF function, additional bits may be allocated per VF MSI-X vector usage.

8.8.4 Extended Interrupt Cause Set - EICS (0x1520; WO)

Software uses this register to set an interrupt condition. Any bit written with a 1b sets the corresponding bit in the Extended Interrupt Cause Read register. An interrupt is then generated if one of the bits in this register is set and the corresponding interrupt is enabled via the Extended Interrupt Mask Set/Read register. Bits written with 0b are unchanged.

Table 8-13 EICS Register Bit Description - non MSI-X mode (GPIE.Multiple_MSIX = 0)

Field	Bit(s)	Initial Value	Description
RxTxQ	7:0	0x0	Sets to corresponding EICR RxTXQ interrupt condition.
Reserved	29:8	0x0	Reserved Write 0, ignore on read.
TCP Timer	30	0b	Sets the corresponding EICR TCP Timer interrupt condition.
Reserved	31	0b	Reserved Write 0, ignore on read.

Note: In order to set bit 31 of the *EICR* (*Other Causes*), the ICS and IMS registers should be used in order to enable one of the legacy causes.

Table 8-14 EICS Register Bit Description - MSI-X mode (GPIE.Multiple_MSIX = 1)

Field	Bit(s)	Initial Value	Description
MSIX	24:0	0x0	Sets the corresponding *EICR* bit of MSI-X vectors 24:0
Reserved	31:25	0x0	Reserved Write 0, ignore on read.

8.8.5 Extended Interrupt Mask Set/Read - EIMS (0x1524; RWS)

Reading of this register returns which bits have an interrupt mask set. An interrupt in *EICR* is enabled if its corresponding mask bit is set to 1b and disabled if its corresponding mask bit is set to 0b. A PCI interrupt is generated each time one of the bits in this register is set and the corresponding interrupt condition occurs (subject to throttling). The occurrence of an interrupt condition is reflected by having a bit set in the Extended Interrupt Cause Read register.

An interrupt might be enabled by writing a 1b to the corresponding mask bit location (as defined in the *EICR* register) in this register. Any bits written with a 0b are unchanged. As a result, if software needs to disable an interrupt condition that had been previously enabled, it must write to the *Extended Interrupt Mask Clear* register rather than writing a 0b to a bit in this register.

Table 8-15 EIMS Register Bit Description - non MSI-X mode (GPIE.Multiple_MSIX = 0)

Field	Bit(s)	Initial Value	Description
RxTxQ	7:0	0x0	Set Mask bit for the corresponding EICR RxTXQ interrupt.
Reserved	29:8	0x0	Reserved Write 0, ignore on read.
TCP Timer	30	0b	Set Mask bit for the corresponding EICR TCP timer interrupt condition.
Other Cause	31	1b	Set Mask bit for the corresponding EICR other cause interrupt condition.

Note: Bits are not reset by Device Reset (*CTRL.DEV_RST*).

Table 8-16 EIMS Register Bit Description - MSI-X mode (GPIE.Multiple_MSIX = 1)

Field	Bit(s)	Initial Value	Description
MSIX	24:0	0x0	Set Mask bit for the corresponding *EICR* bit of MSI-X vectors 24:0. Note: Bits are not reset by Device Reset (*CTRL.DEV_RST*).
Reserved	31:25	0x0	Reserved Write 0, ignore on read.

8.8.6 Extended Interrupt Mask Clear - EIMC (0x1528; WO)

This register provides software a way to disable certain or all interrupts. Software disables a given interrupt by writing a 1b to the corresponding bit in this register.

On interrupt handling, the software device driver should set all the bits in this register related to the current interrupt request even though the interrupt was triggered by part of the causes that were allocated to this vector.

Interrupts are presented to the bus interface only when the mask bit is set to 1b and the cause bit is set to 1b. The status of the mask bit is reflected in the Extended Interrupt Mask Set/Read register and the status of the cause bit is reflected in the Interrupt Cause Read register.

Software blocks interrupts by clearing the corresponding mask bit. This is accomplished by writing a 1b to the corresponding bit location (as defined in the *EICR* register) of that interrupt in this register. Bits written with 0b are unchanged (their mask status does not change).

Table 8-17 EIMC Register Bit Description - non MSI-X mode (GPIE.Multiple_MSIX = 0)

Field	Bit(s)	Initial Value	Description
RxTxQ	7:0	0x0	Clear Mask bit for the corresponding *EICR* RxTXQ interrupt.
Reserved	29:8	0x0	Reserved Write 0, ignore on read.
TCP Timer	30	0b	Clear Mask bit for the corresponding *EICR* TCP timer interrupt.
Other Cause	31	1b	Clear Mask bit for the corresponding *EICR* other cause interrupt.

Table 8-18 EIMC Register Bit Description - MSI-X mode (GPIE.Multiple_MSIX = 1)

Field	Bit(s)	Initial Value	Description
MSIX	24:0	0x0	clear Mask bit for the corresponding *EICR* bit of MSI-X vectors 24:0
Reserved	31:25	0x0	Reserved Write 0, ignore on read.

8.8.7 Extended Interrupt Auto Clear - EIAC (0x152C; R/W)

This register is mapped like the *EICS*, *EIMS*, and *EIMC* registers, with each bit mapped to the corresponding MSI-X vector.

This register is relevant to MSI-X mode only, where read-to-clear can not be used, as it might erase causes tied to other vectors. If any bits are set in *EIAC*, the *EICR* register should not be read. Bits without auto clear set, need to be cleared with write-to-clear.

Note: *EICR* bits that have auto clear set are cleared by the internal emission of the corresponding MSI-X message even if this vector is disabled by the operating system.

The MSI-X message can be delayed by *EITR* moderation from the time the *EICR* bit is activated.

Table 8-19 EIAC Register Bit Description - MSI-X mode (GPIE.Multiple_MSIX = 1)

Field	Bit(s)	Initial Value	Description
MSIX	24:0	0x0	Auto clear bit for the corresponding *EICR* bit of MSI-X vectors 24:0. Notes: • Bits are not reset by Device Reset (*CTRL.DEV_RST*). • When *GPIE.Multiple_MSIX* = 0 (Non MSI-X mode) bits 8 and 9 are read only and should be ignored.
Reserved	31:25	0x0	Reserved Write 0, ignore on read.

8.8.8 Extended Interrupt Auto Mask Enable - EIAM (0x1530; R/W)

Each bit in this register enables clearing of the corresponding bit in *EIMS* register following read- or write-to-clear to *EICR* or setting of the corresponding bit in *EIMS* following a write-to-set to *EICS*.

In MSI-X mode, this register controls which of the bits in the *EIMS* register to clear upon interrupt generation if enabled via the *GPIE.EIAME* bit.

Note: When operating in MSI mode than setting any bit in the *EIAM* register will cause clearing of all bits in the *EIMS* register and masking of all interrupts following generation of a MSI interrupt.

Table 8-20 EIAM Register Bit Description - non MSI-X mode (GPIE.Multiple_MSIX = 0)

Field	Bit(s)	Initial Value	Description
RxTxQ	7:0	0x0	Auto Mask bit for the corresponding *EICR RxTxQ* interrupt.
Reserved	29:8	0x0	Reserved Write 0, ignore on read.
TCP Timer	30	0b	Auto mask bit for the corresponding *EICR TCP timer* interrupt condition.
Other Cause	31	0b	Auto mask bit for the corresponding *EICR other cause* interrupt condition.

Note: Bits are not reset by Device Reset (*CTRL.DEV_RST*).

Table 8-21 EIAM Register Bit Description - MSI-X mode (GPIE.Multiple_MSIX = 1)

Field	Bit(s)	Initial Value	Description
MSIX	24:0	0x0	Auto Mask bit for the corresponding *EICR* bit of MSI-X vectors 24:0. Note: Bits are not reset by Device Reset (*CTRL.DEV_RST*).
Reserved	31:25	0x0	Reserved Write 0, ignore on read.

8.8.9 Interrupt Cause Read Register - ICR (0x1500; RC/W1C)

This register contains the interrupt conditions for the I350 that are not present directly in the EICR. Each time an ICR interrupt causing event occurs, the corresponding interrupt bit is set in this register. The *EICR.Other* bit reflects the setting of interrupt causes from *ICR* as masked by the Interrupt Mask Set/Read register. Each time all un-masked causes in *ICR* are cleared, the *EICR.Other* bit is also cleared.

ICR bits are cleared on register read. Clear-on-read can be enabled/disabled through a general configuration register bit. Refer to Section 7.3.3 for additional information.

Auto clear is not available for the bits in this register.

In order to prevent unwanted *LSC* (*Link Status Change*) interrupts during initialization, software should disable this interrupt until the end of initialization.

Field	Bit(s)	Initial Value	Description
TXDW	0	0b	Transmit Descriptor Written Back Set when the I350 writes back a Tx descriptor to memory.
Reserved	1	0b	Reserved Write 0, ignore on read.
LSC	2	0b	Link Status Change This bit is set each time the link status changes (either from up to down, or from down to up). This bit is affected by the LINK indication from the PHY (internal PHY mode).
Reserved	3	0b	Reserved Write 0, ignore on read.
RXDMT0	4	0b	Receive Descriptor Minimum Threshold Reached Indicates that the minimum number of receive descriptors are available and software should load more receive descriptors.
Reserved	5	0b	Reserved Write 0, ignore on read.
Rx Miss	6	0b	Missed packet interrupt is activated for each received packet that overflows the Rx packet buffer (overrun). Note that the packet is dropped and also increments the associated MPC counter. Note: Could be caused by no available receive buffers or because PCIe receive bandwidth is inadequate.
RXDW	7	0b	Receiver Descriptor Write Back Set when the I350 writes back an Rx descriptor to memory.
SWMB	8	0b	Set when one of the drivers wrote a message using the SWMBWR mailbox register. Set in IOV mode when a VF sends a message or an acknowledge of a message to the PF. Also set, when an FLR is asserted for one of the VFs.
Reserved	9	0b	Reserved
GPHY	10	0b	Internal 1000/100/10BASE-T PHY interrupt. Refer to Section 8.26.3.23 for further information.
GPI_SDP0	11	0b	General Purpose Interrupt on SDP0 If GPI interrupt detection is enabled on this pin (via *CTRL.SDP0_GPIEN*), this interrupt cause is set when the SDP0 is sampled high.
GPI_SDP1	12	0b	General Purpose Interrupt on SDP1 If GPI interrupt detection is enabled on this pin (via *CTRL.SDP1_GPIEN*), this interrupt cause is set when the SDP1 is sampled high.

Field	Bit(s)	Initial Value	Description
GPI_SDP2	13	0b	General Purpose Interrupt on SDP2 If GPI interrupt detection is enabled on this pin (via *CTRL_EXT.SDP2_GPIEN*), this interrupt cause is set when the SDP2 is sampled high.
GPI_SDP3	14	0b	General Purpose Interrupt on SDP3 If GPI interrupt detection is enabled on this pin (via *CTRL_EXT.SDP3_GPIEN*), this interrupt cause is set when the SDP3 is sampled high.
Reserved	17:15	000b	Reserved
MNG	18	0b	Manageability Event Detected Indicates that a manageability event happened. When bit is set due to detection of error by Management, *FWSM.Ext_Err_Ind* field is updated with the error cause.
Time_Sync	19	0b	Time_Sync Interrupt This interrupt cause is set if Interrupt is generated by the Time Sync Interrupt registers (*TSICR*, *TSIM* and *TSIS*).
OMED	20	0b	Other Media Energy Detect When in SerDes/SGMII mode, indicates that link status has changed on the 10/100/1000BASE-T link or when in internal 1000BASE-T PHY mode, there is a change in the external media link status.
Reserved	21	0b	Reserved Write 0, ignore on read.
FER	22	0b	Fatal Error This bit is set when a fatal error is detected in one of the memories
THS	23	0b	Thermal Sensor Event This bit is set when thermal trip point was crossed.
PCI Exception	24	0b	The PCI timeout Exception is activated by one of the following events when the specific PCI event is reported in the *PICAUSE* register and the appropriate bit in the *PIENA* register is set: (1) IO completion abort. (2) Unsupported IO request (Wrong address). (3) Byte-Enable error - Access to client that does not support Partial BE access (All but Flash, MSIX & PCIE-target). (4) Timeout occurred in the FUNC block. (5) Bus-master-enable (*BME*) of the PF is cleared. ***Note:*** This event can occur if a CSR access timeout is detected during one of the internal clients (firmware or NVM) access.
SCE	25	0b	Storm Control Event This bit is set when multicast or broadcast storm control mechanism is activated or de-activated.
Software WD	26	0b	Software Watchdog This bit is set after a software watchdog timer times out.
Reserved	27	0b	Reserved Write 0, ignore on read.
MDDET	28	0b	Detected Malicious driver behavior Occurs when one of the queues used malformed descriptors. In virtualized systems, might indicate a malicious or buggy driver. **Note:** This bit should never rise during normal operation.
TCP timer	29	0b	TCP timer interrupt Activated when the TCP timer reaches its terminal count.
DRSTA	30	0b	Device Reset Asserted Indicates the *CTRL.DEV_RST* was asserted on another port or on this port. When device reset occurs all ports should re-initialize registers and descriptor rings. Note: Bit is not reset by Device Reset (*CTRL.DEV_RST*).
INTA	31	0b	Interrupt Asserted: Indicates that the INT line is asserted. Can be used by driver in shared interrupt scenario to decide if the received interrupt was emitted by the I350. This bit is not valid in MSI/MSI-X environments

8.8.10 Interrupt Cause Set Register - ICS (0x1504; WO)

Software uses this register to set an interrupt condition. Any bit written with a 1b sets the corresponding interrupt. This results in the corresponding bit being set in the Interrupt Cause Read Register (refer to Section 8.8.9). A PCIe interrupt is generated if one of the bits in this register is set and the corresponding interrupt is enabled through the Interrupt Mask Set/Read Register (refer to Section 8.8.11). Bits written with 0b are unchanged. Refer to Section 7.3.3 for additional information.

Field	Bit(s)	Initial Value	Description
TXDW	0	0b	Sets the Transmit Descriptor Written Back Interrupt.
Reserved	1	0b	Reserved Write 0, ignore on read.
LSC	2	0b	Sets the Link Status Change Interrupt.
Reserved	3	0b	Reserved Write 0, ignore on read.
RXDMT0	4	0b	Sets the Receive Descriptor Minimum Threshold Hit Interrupt.
Reserved	5	0b	Reserved Write 0, ignore on read.
Rx Miss	6	0b	Sets the Rx Miss Interrupt.
RXDW	7	0b	Sets the Receiver Descriptor Write Back Interrupt.
SWMB	8	0b	Sets the SWMB mailbox interrupt.
Reserved	9	0b	Reserved
GPHY	10	0b	Sets the Internal 1000/100/10BASE-T PHY interrupt.
GPI_SDP0	11	0b	Sets the General Purpose Interrupt, related to SDP0 pin.
GPI_SDP1	12	0b	Sets the General Purpose Interrupt, related to SDP1 pin.
GPI_SDP2	13	0b	Sets the General Purpose Interrupt, related to SDP2 pin.
GPI_SDP3	14	0b	Sets the General Purpose Interrupt, related to SDP3 pin.
Reserved	17:15	000b	Reserved.
MNG	18	0b	Sets the Management Event Interrupt.
Time_Sync	19	0b	Sets the Time_Sync interrupt.
OMED	20	0b	Sets the Other Media Energy Detected Interrupt.
Reserved	21	0b	Reserved Write 0, ignore on read.
FER	22	0b	Sets the Fatal Error Interrupt.
THS	23	0b	Sets the Thermal Sensor Event Interrupt.
PCI Exception	24	0b	Sets the PCI Exception Interrupt.
SCE	25	0b	Set the Storm Control Event Interrupt
Software WD	26	0b	Sets the Software Watchdog Interrupt.
Reserved	27	0b	Reserved. Write 0, ignore on read.
MDDET	28	0b	Sets the Detected Malicious driver behavior Interrupt.
TCP timer	29	0b	Sets the TCP timer interrupt.
DRSTA	30	0b	Sets the Device Reset Asserted Interrupt. Note that when setting this bit a DRSTA interrupt is generated on this port only.
Reserved	31	0b	Reserved. Write 0, ignore on read.

8.8.11 Interrupt Mask Set/Read Register - IMS (0x1508; R/W)

Reading this register returns bits that have an interrupt mask set. An interrupt is enabled if its corresponding mask bit is set to 1b and disabled if its corresponding mask bit is set to 0b. A PCIe interrupt is generated each time one of the bits in this register is set and the corresponding interrupt condition occurs. The occurrence of an interrupt condition is reflected by having a bit set in the Interrupt Cause Read Register (refer to Section 8.8.9).

A particular interrupt can be enabled by writing a 1b to the corresponding mask bit in this register. Any bits written with a 0b are unchanged. As a result, if software desires to disable a particular interrupt condition that had been previously enabled, it must write to the Interrupt Mask Clear Register (refer to Section 8.8.12) rather than writing a 0b to a bit in this register. Refer to Section 7.3.3 for additional information.

Field	Bit(s)	Initial Value	Description
TXDW	0	0b	Sets/Reads the mask for Transmit Descriptor Written Back Interrupt.
Reserved	1	0b	Reserved Write 0, ignore on read.
LSC	2	0b	Sets/Reads the mask for Link Status Change Interrupt.
Reserved	3	0b	Reserved Write 0, ignore on read.
RXDMT0	4	0b	Sets/Reads the mask for Receive Descriptor Minimum Threshold Hit Interrupt.
Reserved	5	0b	Reserved Write 0, ignore on read.
Rx Miss	6	0b	Sets/Reads the mask for the Rx Miss Interrupt.
RXDW	7	0b	Sets/Reads the mask for Receiver Descriptor Write Back Interrupt.
SWMB	8	0b	Sets/Reads the mask for Software Mailbox Interrupt.
Reserved	9	0b	Reserved
GPHY	10	0b	Sets/Reads the mask for Internal 1000/100/10BASE-T PHY interrupt.
GPI_SDP0	11	0b	Sets/Reads the mask for General Purpose Interrupt, related to SDP0 pin.
GPI_SDP1	12	0b	Sets/Reads the mask for General Purpose Interrupt, related to SDP1 pin.
GPI_SDP2	13	0b	Sets/Reads the mask for General Purpose Interrupt, related to SDP2 pin.
GPI_SDP3	14	0b	Sets/Reads the mask for General Purpose Interrupt, related to SDP3 pin.
Reserved	17:15	000b	Reserved.
MNG	18	0b	Sets/Reads the mask for Management Event Interrupt.
Time_Sync	19	0b	Sets/Reads the mask for Time_Sync Interrupt.
OMED	20	0b	Sets/Reads the mask for Other Media Energy Detected Interrupt.
Reserved	21	0b	Reserved Write 0, ignore on read.
FER	22	0b	Sets/Reads the mask for the Fatal Error Interrupt.
THS	23	0b	Sets/Reads the mask for the Thermal Sensor Event Interrupt.
PCI Exception	24	0b	Sets/Reads the mask for the PCI Exception Interrupt.
SCE	25	0b	Sets/Reads the mask for the Storm Control Event Interrupt.
Software WD	26	0b	Sets/Reads the mask for the Software Watchdog Interrupt.
Reserved	27	0b	Reserved. Write 0, ignore on read.

Field	Bit(s)	Initial Value	Description
MDDET	28	0b	Sets/Reads the mask for Detected Malicious driver behavior Interrupt.
TCP timer	29	0b	Sets/Reads the mask for TCP timer interrupt.
DRSTA	30	0b	Sets/Reads the mask for Device Reset Asserted Interrupt. Note: Bit is not reset by Device Reset (CTRL.DEV_RST).
Reserved	31	0b	Reserved. Write 0, ignore on read.

8.8.12 Interrupt Mask Clear Register - IMC (0x150C; WO)

Software uses this register to disable an interrupt. Interrupts are presented to the bus interface only when the mask bit is set to 1b and the cause bit set to 1b. The status of the mask bit is reflected in the Interrupt Mask Set/Read Register (refer to Section 8.8.11), and the status of the cause bit is reflected in the Interrupt Cause Read Register (refer to Section 8.8.9). Reading this register returns the value of the IMS register.

Software blocks interrupts by clearing the corresponding mask bit. This is accomplished by writing a 1b to the corresponding bit in this register. Bits written with 0b are unchanged (their mask status does not change).

Software device driver should set all the bits in this register related to the current interrupt request when handling interrupts, even though the interrupt was triggered by part of the causes that were allocated to this vector. Refer to Section 7.3.3 for additional information.

Field	Bit(s)	Initial Value	Description
TXDW	0	0b	Clears the mask for Transmit Descriptor Written Back Interrupt.
Reserved	1	0b	Reserved Write 0, ignore on read.
LSC	2	0b	Clears the mask for Link Status Change Interrupt.
Reserved	3	0b	Reserved Write 0, ignore on read.
RXDMT0	4	0b	Clears the mask for Receive Descriptor Minimum Threshold Hit Interrupt.
Reserved	5	0b	Reserved Write 0, ignore on read.
Rx Miss	6	0b	Clears the mask for the Rx Miss Interrupt.
RXDW	7	0b	Clears the mask for the Receiver Descriptor Write Back Interrupt.
SWMB	8	0b	Clears the mask for the Software Mailbox interrupt.
Reserved	9	0b	Reserved
GPHY	10	0b	Clears the mask for the Internal 1000/100/10BASE-T PHY interrupt.
GPI_SDP0	11	0b	Clears the mask for the General Purpose Interrupt, related to SDP0 pin.
GPI_SDP1	12	0b	Clears the mask for the General Purpose Interrupt, related to SDP1 pin.
GPI_SDP2	13	0b	Clears the mask for the General Purpose Interrupt, related to SDP2 pin.
GPI_SDP3	14	0b	Clears the mask for the General Purpose Interrupt, related to SDP3 pin.
Reserved	17:15	000b	Reserved.
MNG	18	0b	Clears the mask for the Management Event Interrupt.
Time_Sync	19	0b	Clears the mask for the Time_Sync Interrupt.

Field	Bit(s)	Initial Value	Description
OMED	20	0b	Clears the mask for the Other Media Energy Detected Interrupt.
Reserved	21	0b	Reserved Write 0, ignore on read.
FER	22	0b	Clears the mask for the Fatal Error Interrupt.
THS	23	0b	Clears the mask for the Thermal Sensor Event Interrupt.
PCI Exception	24	0b	Clears the mask for the PCI Exception Interrupt.
SCE	25	0b	Clears the mask for the Storm Control Event Interrupt.
Software WD	26	0b	Clears the mask for Software Watchdog Interrupt.
Reserved	27	0b	Reserved. Write 0, ignore on read.
MDDET	28	0b	Clears the mask for Detected Malicious driver behavior Interrupt.
TCP timer	29	0b	Clears the mask for TCP timer interrupt.
DRSTA	30	0b	Clears the mask for Device Reset Asserted Interrupt.
Reserved	31	0b	Reserved. Write 0, ignore on read.

8.8.13 Interrupt Acknowledge Auto Mask Register - IAM (0x1510; R/W)

Field	Bit(s)	Initial Value	Description
IAM_VALUE	30:0	0b	An *ICR* read or write will have the side effect of writing the contents of this register to the *IMC* register. If *GPIE.NSICR* = 0, then the copy of this register to the *IMC* register will occur only if at least one bit is set in the *IMS* register and there is a true interrupt as reflected in the *ICR.INTA* bit. Refer to Section 7.3.3 for additional information. ***Note:*** Bit 30 of this register is not reset by Device Reset (*CTRL.DEV_RST*).
Reserved	31	0b	Reserved. Write 0, ignore on read.

8.8.14 Interrupt Throttle - EITR (0x1680 + 4*n [n = 0...24]; R/W)

Each *EITR* is responsible for an interrupt cause (RxTxQ, TCP timer and Other Cause). The allocation of EITR-to-interrupt cause is through the *IVAR* registers.

Software uses this register to pace (or even out) the delivery of interrupts to the host processor. This register provides a guaranteed inter-interrupt delay between interrupts asserted by the I350, regardless of network traffic conditions. To independently validate configuration settings, software can use the following algorithm to convert the inter-interrupt interval value to the common interrupts/sec. performance metric:

$$\text{interrupts/sec} = (1 * 10^{-6}\text{sec} \times \text{interval})^{-1}$$

A counter counts in units of $1*10^{-6}$ sec. After counting "interval "number of units, an interrupt is sent to the software. The above equation gives the number of interrupts per second. The equation below time in seconds between consecutive interrupts.

For example, if the interval is programmed to 125 (decimal), the I350 guarantees the processor does not receive an interrupt for 125 μs from the last interrupt. The maximum observable interrupt rate from the I350 should never exceed 8000 interrupts/sec.

Inversely, inter-interrupt interval value can be calculated as:

$$\text{inter-interrupt interval} = (1 * 10^{-6}\text{sec} \times \text{interrupt/sec})^{-1}$$

The optimal performance setting for this register is very system and configuration specific. An initial suggested range is 2 to 175 (0x02 to 0xAF).

Note: Setting EITR to a non zero value can cause an interrupt cause Rx/Tx statistics miscount.

Field	Bit(s)	Initial Value	Description
Reserved	1:0	0x0	Reserved Write 0, ignore on read.
Interval	14:2	0x0	Minimum inter-interrupt interval. The interval is specified in 1 μs increments. A zero disables interrupt throttling logic.
LLI_EN	15	0b	LLI moderation enable.
LL Counter (RWS)	20:16	0x0	Reflects the current credits for that *EITR* for LL interrupts. If the *CNT_INGR* is not set this counter can be directly written by software at any time to alter the throttles performance
Moderation Counter (RWS)	30:21	0x0	Down counter, exposes only the 10 most significant bits of the real 12-bit counter. Loaded with Interval value whenever the associated interrupt is signaled. Counts down to 0 and stops. The associated interrupt is signaled whenever this counter is zero and an associated (via the Interrupt Select register) EICR bit is set. If the CNT_INGR is not set this counter can be directly written by software at any time to alter the throttles performance.
CNT_INGR (WO)	31	0b	When set the hardware does not override the counters fields (ITR counter and LLI credit counter), so they keep their previous value. Relevant for the current write only and is always read as zero

Note: EITR register and interrupt mechanism is not reset by Device Reset (*CTRL.DEV_RST*). Occurrence of Device Reset interrupt causes immediate generation of all pending interrupts.

8.8.15 Interrupt Vector Allocation Registers - IVAR (0x1700 + 4*n [n=0...3]; RW)

These registers have three modes of operation:

1. In MSI-X mode these registers define the allocation of the different interrupt causes as defined in Table 7-48 to one of the MSI-X vectors. Each *INT_Alloc[i]* (i=0…15) field is a byte indexing an entry in the MSI-X Table Structure and MSI-X PBA Structure.
2. In non MSI-X mode these registers define the allocation of the Rx and Tx queues interrupt causes to one of the RxTxQ bits in the *EICR* register. Each *INT_Alloc[i]* (i=0…15) field is a byte indexing the appropriate RxTxQ bit as defined in Table 7-47.
3. In SR-IOV mode MSI-X mode is always activated. These registers define the allocation of the different interrupt causes as defined in Table 7-48 to one of the MSI-X vectors. Each *INT_Alloc[i]*

(i=0…15) field is a byte indexing an entry in the MSI-X Table Structure and MSI-X PBA Structure. Any *INT_Alloc[i]* field that is allocated to a VF is Read only (RO), while any *INT_Alloc[i]* fields allocated to the PF are Read/Write (R/W). Fields allocated to the VF can be accessed by the PF using the VF address space (See *VTIVAR* addresses under the Physical Address Base column in Section 8.27.3).

Entries are mapped as follows:

a. Queues RX0, TX0, RX1, TX1 are mapped in *IVAR[0]* Register.

b. Queues RX2, TX2, RX3, TX3 are mapped in *IVAR[1]* Register.

c. Queues RX4, TX4, RX5, TX5 are mapped in *IVAR[2]* Register.

d. Queues RX6, TX6, RX7, TX7 are mapped in *IVAR[3]* Register.

Field	Bit(s)	Initial Value	Description
INT_Alloc[0]	4:0	0x0	Defines the MSI-X vector assigned to the interrupt cause associated with this entry, as defined in Table 7-48. Valid values are 0 to 24 for MSI-X mode and 0 to 7 in non MSI-X mode.
Reserved	6:5	0x0	Reserved Write 0 ignore on read.
INT_Alloc_val[0]	7	0b	Valid bit for INT_Alloc[0]
INT_Alloc[1]	12:8	0x0	Defines the MSI-X vector assigned to the interrupt cause associated with this entry, as defined in Table 7-48. Valid values are 0 to 24 for MSI-X mode and 0 to 7 in non MSI-X mode.
Reserved	14:13	0x0	Reserved Write 0 ignore on read
INT_Alloc_val[1]	15	0b	Valid bit for INT_Alloc[1]
INT_Alloc[2]	20:16	0x0	Defines the MSI-X vector assigned to the interrupt cause associated with this entry, as defined in Table 7-48. Valid values are 0 to 24 for MSI-X mode and 0 to 7 in non MSI-X mode.
Reserved	22:21	0x0	Reserved Write 0, ignore on read.
INT_Alloc_val[2]	23	0b	Valid bit for INT_Alloc[2]
INT_Alloc[3]	28:24	0x0	Defines the MSI-X vector assigned to the interrupt cause associated with this entry, as defined in Table 7-48. Valid values are 0 to 24 for MSI-X mode and 0 to 7 in non MSI-X mode.
Reserved	30:29	0x0	Reserved Write 0, ignore on read.
INT_Alloc_val[3]	31	0b	Valid bit for INT_Alloc[3]

Note: If invalid values are written to the INT_Alloc fields the result is unexpected.

DW	31 24	23 16	15 8	7 0
0	INT_ALLOC[3]	INT_ALLOC[2]	INT_ALLOC[1]	INT_ALLOC[0]
1			…	…
2		…		
3	INT_ALLOC[15]	INT_ALLOC[14]	INT_ALLOC[13]	INT_ALLOC[12]

8.8.16 Interrupt Vector Allocation Registers - MISC IVAR_MISC (0x1740; RW)

This register is used only in MSI-X mode. This register defines the allocation of the *Other Cause* and *TCP Timer* interrupts to one of the MSI-X vectors.

Field	Bit(s)	Initial Value	Description
INT_Alloc[16]	4:0	0x0	Defines the MSI-X vector assigned to the TCP timer interrupt cause. Valid values are 0 to 24.
Reserved	6:5	0x0	Reserved Write 0, ignore on read.
INT_Alloc_val[16]	7	0b	Valid bit for INT_Alloc[16]
INT_Alloc[17]	12:8	0x0	Defines the MSI-X vector assigned to the "Other Cause" interrupt. Valid values are 0 to 24.
Reserved	14:13	0b	Reserved Write 0, ignore on read.
INT_Alloc_val[17]	15	0b	Valid bit for INT_Alloc[17]
Reserved	31:16	0x0	Reserved Write 0, ignore on read.

8.8.17 General Purpose Interrupt Enable - GPIE (0x1514; RW)

Field	Bit(s)	Initial Value	Description
NSICR	0	0b	Non Selective Interrupt clear on read: When set, every read of *ICR* clears it. When this bit is cleared, an *ICR* read causes it to be cleared only if an actual interrupt was asserted or *IMS* = 0x0. Refer to Section 7.3.3 for additional information.
Reserved	3:1	0x0	Reserved Write 0, ignore on read.
Multiple MSIX	4	0b	0 = on-MSI mode, or MSI-X with single vector, *IVAR* maps Rx/Tx causes, to 8 EICR bits, but MSIX[0] is asserted for all. 1 = MSIX mode, *IVAR* maps Rx/Tx causes, TCP Timer and "Other Cause" interrupts to 25 MSI-x vectors reflected in 25 *EICR* bits. ***Note:*** When set, the *EICR* register is not cleared on read.
Reserved	6:5	0x0	Reserved Write 0, ignore on read.
LL Interval	11:7	0x0	Low latency credits increment rate. The interval is specified in 4 µs increments. A value of 0x0 disables moderation of LLI for all interrupt vectors.
Reserved	29:12	0x0	Reserved Write 0, ignore on read.

Field	Bit(s)	Initial Value	Description
EIAME	30	0b	Extended Interrupt Auto Mask enable: When set (usually in MSI-X mode); upon firing of a MSI-X message, if bits in *EIAM* register associated with this message are set then the corresponding bits in the *EIMS* register will be cleared. Otherwise, *EIAM* is used only upon read or write of EICR/EICS registers. ***Note:*** When bit is set in MSI mode than setting of any bit in the *EIAM* register will cause clearing of all bits in the *EIMS* register and masking of all interrupts following generation of a MSI interrupt.
PBA_support	31	0b	PBA Support: When set, setting one of the extended interrupts masks via *EIMS* causes the PBA bit of the associated MSI-X vector to be cleared. Otherwise, the I350 behaves in a way that supports legacy INT-x interrupts. **Note:** Should be cleared when working in INT-x or MSI mode and set in MSI-X mode.

8.9 MSI-X Table Register Descriptions

These registers are used to configure the MSI-X mechanism. The *message address* and *message upper address* registers sets the address for each of the vectors. The message register sets the data sent to the relevant address. The vector control registers are used to enable specific vectors.

The pending bit array register indicates which vectors have pending interrupts. The structure is listed in Table 8-22.

Table 8-22 MSI-X Table Structure

DWORD3 MSIXTVCTRL	DWORD2 MSIXTMSG	DWORD1 MSIXTUADD	DWORD0 MSIXTADD	Entry Number	BAR 3 - Offset
Vector Control	Msg Data	Msg Upper Addr.	Msg Addr	Entry 0	Base (0x0000)
Vector Control	Msg Data	Msg Upper Addr	Msg Addr	Entry 1	Base + 1*16
Vector Control	Msg Data	Msg Upper Addr	Msg Addr	Entry 2	Base + 2*16
…	…	…	…	…	
Vector Control	Msg Data	Msg Upper Addr	Msg Addr	Entry (N-1)	Base + (N-1) *16

Note: N = 25.

Table 8-23 MSI-X PBA Structure

MSIXPBA[63:0]	QWORD Number	BAR 3 - Offset
Pending Bits 0 through 63	QWORD0	Base (0x2000)
Pending Bits 64 through 127	QWORD1	Base+1*8
…	…	…
Pending Bits ((N-1) div 64)*64 through N-1	QWORD((N-1) div 64)	BASE + ((N-1) div 64)*8

Note: N = 25. As a result, only QWORD0 is implemented.

8.9.1 MSI–X Table Entry Lower Address - MSIXTADD (BAR3: 0x0000 + 0x10*n [n=0…24]; R/W)

Field	Bit(s)	Initial Value	Description
Message Address LSB (RO)	1:0	0x0	For proper DWORD alignment, software must always write 0b's to these two bits. Otherwise, the result is undefined.
Message Address	31:2	0x0	System-Specific Message Lower Address For MSI-X messages, the contents of this field from an MSI-X table entry specifies the lower portion of the DWORD-aligned address for the memory write transaction.

8.9.2 MSI–X Table Entry Upper Address - MSIXTUADD (BAR3: 0x0004 + 0x10*n [n=0…24]; R/W)

Field	Bit(s)	Initial Value	Description
Message Address	31:0	0x0	System-Specific Message Upper Address

8.9.3 MSI–X Table Entry Message - MSIXTMSG (BAR3: 0x0008 + 0x10*n [n=0…24]; R/W)

Field	Bit(s)	Initial Value	Description
Message Data	31:0	0x0	System-Specific Message Data For MSI-X messages, the contents of this field from an MSI-X table entry specifies the data written during the memory write transaction. In contrast to message data used for MSI messages, the low-order message data bits in MSI-X messages are not modified by the function.

8.9.4 MSI–X Table Entry Vector Control - MSIXTVCTRL (BAR3: 0x000C + 0x10*n [n=0...24]; R/W)

Field	Bit(s)	Initial Value	Description
Mask	0	1b	When this bit is set, the function is prohibited from sending a message using this MSI-X table entry. However, any other MSI-X table entries programmed with the same vector are still capable of sending an equivalent message unless they are also masked.
Reserved	31:1	0x0	Reserved Write 0, ignore on read.

8.9.5 MSIXPBA Bit Description – MSIXPBA (BAR3: 0x2000; RO)

Field	Bit(s)	Initial Value	Description
Pending Bits	24:0	0x0	For each pending bit that is set, the function has a pending message for the associated MSI-X Table entry. Pending bits that have no associated MSI-X table entry are reserved.
Reserved	31:25	0x0	Reserved Write 0, ignore on read.

8.9.6 MSI-X PBA Clear – PBACL (0x5B68; R/W1C)

Field	Bit(s)	Initial Value	Description
PENBITCLR	24:0	0x0	MSI-X Pending bits Clear Writing a 1b to any bit clears the corresponding MSIXPBA bit; writing a 0b has no effect. **Note:** Bits are set for a single PCIe clock cycle and then cleared.
Reserved	31:25	0x0	Reserved Write 0, ignore on read.

8.10 Receive Register Descriptions

8.10.1 Receive Control Register - RCTL (0x0100; R/W)

This register controls all the I350 receiver functions.

Field	Bit(s)	Initial Value	Description
Reserved	0	0b	Reserved Write 0, ignore on read.
RXEN	1	0b	Receiver Enable The receiver is enabled when this bit is set to 1b. Writing this bit to 0b stops reception after receipt of any in progress packet. All subsequent packets are then immediately dropped until this bit is set to 1b.
SBP	2	0b	Store Bad Packets 0b = do not store. 1b = store bad packets. This bit controls the MAC receive behavior. A packet is required to pass the address (or normal) filtering before the SBP bit becomes effective. If SBP = 0b, then all packets with layer 1 or 2 errors are rejected. The appropriate statistic would be increased. If SBP = 1b, then these packets are received (and transferred to host memory). The receive descriptor error field (RDESC.ERRORS) should have the corresponding bit(s) set to signal the software device driver that the packet is erred. In some operating systems the software device driver passes this information to the protocol stack. In either case, if a packet only has layer 3+ errors, such as IP or TCP checksum errors, and passes other filters, the packet is always received (layer 3+ errors are not used as a packet filter). ***Note:*** symbol errors before the SFD are ignored. Any packet must have a valid SFD (RX_DV with no RX_ER in 10/100/1000BASE-T mode) in order to be recognized by the I350 (even bad packets). Also, erred packets are not routed to the MNG even if this bit is set.
UPE	3	0b	Unicast Promiscuous Enabled 0b = Disabled. 1b = Enabled.
MPE	4	0b	Multicast Promiscuous Enabled 0b = Disabled. 1b = Enabled.
LPE	5	0b	Long Packet Reception Enable 0b = Disabled. 1b = Enabled. LPE controls whether long packet reception is permitted. If LPE is 0b Hardware discards long packets over 1518, 1522 or 1526 bytes depending on the *CTRL_EXT.EXT_VLAN* bit and the detection of a VLAN tag in the packet. If LPE is 1b, the maximum packet size that the I350 can receive is defined in the *RLPML.RLPML* register.
LBM	7:6	00b	Loopback mode. Controls the loopback mode of the I350. 00b = Normal operation (or PHY loopback in 10/100/1000BASE-T mode). 01b = MAC loopback (test mode). 10b = Undefined. 11b = Loopback via internal SerDes (SerDes/SGMII/KX mode only). When using the internal PHY, LBM should remain set to 00b and the PHY instead configured for loopback through the MDIO interface. **Note:** PHY devices require programming for loopback operation using MDIO accesses.
Reserved	11:8	0x0	Reserved Write 0, ignore on read.
MO	13:12	00b	Multicast Offset Determines which bits of the incoming multicast address are used in looking up the bit vector. 00b = bits [47:36] of received destination multicast address. 01b = bits [46:35] of received destination multicast address. 10b = bits [45:34] of received destination multicast address. 11b = bits [43:32] of received destination multicast address.
Reserved	14	0b	Reserved Write 0, ignore on read.

Field	Bit(s)	Initial Value	Description
BAM	15	0b	Broadcast Accept Mode. 0b = Ignore broadcast (unless it matches through exact or imperfect filters). 1b = Accept broadcast packets.
BSIZE	17:16	00b	Receive Buffer Size BSIZE controls the size of the receive buffers and permits software to trade-off descriptor performance versus required storage space. Buffers that are 2048 bytes require only one descriptor per receive packet maximizing descriptor efficiency. 00b = 2048 Bytes. 01b = 1024 Bytes. 10b = 512 Bytes. 11b = 256 Bytes. **Notes:** 1. BSIZE should not be modified when RXEN is set to 1b. Set RXEN =0 when modifying the buffer size by changing this field. 2. BSIZE value only defines receive buffer size of queues with a *SRRCTL.BSIZEPACKET* value of 0.
VFE	18	0b	VLAN Filter Enable 0b = Disabled (filter table does not decide packet acceptance). 1b = Enabled (filter table decides packet acceptance for 802.1Q packets). Three bits [20:18] control the VLAN filter table. The first determines whether the table participates in the packet acceptance criteria. The next two are used to decide whether the CFI bit found in the 802.1Q packet should be used as part of the acceptance criteria.
CFIEN	19	0b	Canonical Form Indicator Enable 0b = Disabled (CFI bit found in received 802.1Q packet's tag is not compared to decide packet acceptance). 1b = Enabled (CFI bit found in received 802.1Q packet's tag must match RCTL.CFI to accept 802.1Q type packet.
CFI	20	0b	Canonical Form Indicator bit value 0b = 802.1Q packets with CFI equal to this field are accepted. 1b = 802.1Q packet is discarded.
PSP	21	0b	Pad Small Receive packets. If this field is set, in virtualized operating mode, strip CRC (D*VMOLR.CRC Strip)* should be set for all functions otherwise *RCTL.SECRC* should be set.
DPF	22	0b	Discard Pause Frames with Station MAC Address Controls whether pause frames directly addressed to this station are forwarded to the host. 0b = incoming pause frames with station MAC address are forwarded to the host. 1b = incoming pause frames with station MAC address are discarded. **Note:** Pause frames with other MAC addresses (multicast address) are always discarded unless the specific address is added to the accepted MAC addresses (either multicast or unicast).
PMCF	23	0b	Pass MAC Control Frames Filters out unrecognized pause and other control frames. 0b = Filter MAC Control frames. 1b = Pass/forward MAC control frames to the Host that are not XON/XOFF Flow Control packets. The *PMCF* bit controls the DMA function of the MAC control frames (other than flow control). A MAC control frame in this context must be addressed to either the MAC control frame multicast address or the station address, match the type field, and NOT match the PAUSE opcode of 0x0001. If *PMCF* = 1b then frames meeting this criteria are transferred to Host memory.

Field	Bit(s)	Initial Value	Description
Reserved	25:24	0x0	Reserved Write 0, ignore on read.
SECRC	26	0b	Strip Ethernet CRC from incoming packet Causes the CRC to be stripped from all packets. 0b = Does not strip CRC 1b = Strips CRC. This bit controls whether the hardware strips the Ethernet CRC from the received packet. This stripping occurs prior to any checksum calculations. The stripped CRC is not transferred to host memory and is not included in the length reported in the descriptor. ***Notes:*** 1. This bit should not be set in virtualization mode. 2. If the *CTRL.VME* bit is set the *RCTL.SECRC* bit should also be set as the CRC is not valid anymore. 3. Even when bit is set CRC strip is not done on runt packets (smaller than 64 Bytes).
Reserved	31:27	0x0	Reserved Write 0, ignore on read.

8.10.2 Split and Replication Receive Control - SRRCTL (0xC00C + 0x40*n [n=0...7]; R/W)

Field	Bit(s)	Initial Value	Description
BSIZEPACKET	6:0	0x0	Receive Buffer Size for Packet Buffer The value is in 1 KB resolution. Valid values can be from 1 KB to 16 KB. Default buffer size is 0 KB. If this field is equal 0x0, then *RCTL.BSIZE* determines the packet buffer size.
DMACQ_Dis	7	0b	DMA Coalescing disable 0 - Enable DMA Coalescing on this queue if *DMACR.DMAC_EN* is set to 1. 1 - Disable DMA Coalescing on this queue. When packet is destined to this queue and device is in coalescing mode, Coalescing mode is exited immediately and PCIe moves to L0 link power management state.
BSIZEHEADER	13:8	0x4	Receive Buffer Size for Header Buffer The value is in 64 bytes resolution. Valid value can be from 64 bytes to 2048 bytes (BSIZEHEADER = 0x1 to 0x20). Default buffer size is 256 bytes. This field must be greater than 0 if the value of *DESCTYPE* is greater or equal to 2. Note: When *SRRCTL.Timestamp* is set to 1 and the value of *SRRCTL.DESCTYPE* is greater or equal to 2, *BSIZEHEADER* size should be equal or greater than 2 (128 bytes).
Reserved	19:14	0x0	Reserved. Write 0 ignore on read.
RDMTS	24:20	0x0	Receive Descriptor Minimum Threshold Size A low latency interrupt (LLI) associated with this queue is asserted whenever the number of free descriptors becomes equal to *RDMTS* multiplied by 16.
DESCTYPE	27:25	000b	Defines the descriptor in Rx 000b = Legacy. 001b = Advanced descriptor one buffer. 010b = Advanced descriptor header splitting. 011b = Advanced descriptor header replication - replicate always. 100b = Advanced descriptor header replication large packet only (larger than header buffer size). 101b = Reserved. 111b = Reserved.

Field	Bit(s)	Initial Value	Description
Reserved	29:28	0x0	Reserved. Write 0, ignore on read.
Timestamp	30	0b	Timestamp Received packet 0 - Do not place timestamp at beginning of receive buffer. 1- Place timestamp at beginning of receive buffer. Timestamp is placed only in buffers of received packets that meet the criteria defined in the *TSYNCRXCTL.Type* field, 2-tuple filters or *ETQF* registers. When set a 40 bit time stamp generated from the value in *SYSTIMH* and *SYSTIML* registers is placed in the receive buffer before the MAC header of the packets defined in the *TSYNCRXCTL.Type* field.
Drop_En	31	0b/1b	Drop Enabled If set, packets received to the queue when no descriptors are available to store them are dropped. The packet is dropped only if there are not enough free descriptors in the host descriptor ring to store the packet. If there are enough descriptors in the host, but they are not yet fetched by the I350, then the packet is not dropped and there are no release of packets until the descriptors are fetched. Default is 0b for queue 0 and 1b for the other queues.

8.10.3 Packet Split Receive Type - PSRTYPE (0x5480 + 4*n [n=0...7]; R/W)

This register enables or disables each type of header that needs to be split or replicated (refer to Section 7.1.5 for additional information on header split support). Each register controls the behavior of 1 queue.

- Packet Split Receive Type Register (queue 0) - PSRTYPE0 (0x5480)
- Packet Split Receive Type Register (queue 1) - PSRTYPE1 (0x5484)
- Packet Split Receive Type Register (queue 2) - PSRTYPE2 (0x5488)
- Packet Split Receive Type Register (queue 3) - PSRTYPE3 (0x548C)
- Packet Split Receive Type Register (queue 4) - PSRTYPE4 (0x5490)
- Packet Split Receive Type Register (queue 5) - PSRTYPE5 (0x5494)
- Packet Split Receive Type Register (queue 6) - PSRTYPE6 (0x5498)
- Packet Split Receive Type Register (queue 7) - PSRTYPE7 (0x549C)

Field	Bit(s)	Initial Value	Description
PSR_type0	0	0b	Header includes MAC (VLAN/SNAP).
PSR_type1	1	1b	Header includes MAC, (VLAN/SNAP) Fragmented IPv4 only
PSR_type2	2	1b	Header includes MAC, (VLAN/SNAP) IPv4, TCP only
PSR_type3	3	1b	Header includes MAC, (VLAN/SNAP) IPv4, UDP only
PSR_type4	4	1b	Header includes MAC, (VLAN/SNAP) IPv4, Fragmented IPv6 only
PSR_type5	5	1b	Header includes MAC, (VLAN/SNAP) IPv4, IPv6, TCP only
PSR_type6	6	1b	Header includes MAC, (VLAN/SNAP) IPv4, IPv6, UDP only
PSR_type7	7	1b	Header includes MAC, (VLAN/SNAP) Fragmented IPv6 only
PSR_type8	8	1b	Header includes MAC, (VLAN/SNAP) IPv6, TCP only
PSR_type9	9	1b	Header includes MAC, (VLAN/SNAP) IPv6, UDP only
Reserved_1	10	1b	Reserved Write 1, ignore on read.
PSR_type11	11	1b	Header includes MAC, (VLAN/SNAP) IPv4, TCP, NFS only

Field	Bit(s)	Initial Value	Description
PSR_type12	12	1b	Header includes MAC, (VLAN/SNAP) IPv4, UDP, NFS only
Reserved_1	13	1b	Reserved Write 1, ignore on read.
PSR_type14	14	1b	Header includes MAC, (VLAN/SNAP) IPv4, IPv6, TCP, NFS only
PSR_type15	15	1b	Header includes MAC, (VLAN/SNAP) IPv4, IPv6, UDP, NFS only
Reserved_1	16	1b	Reserved Write 1, ignore on read.
PSR_type17	17	1b	Header includes MAC, (VLAN/SNAP) IPv6, TCP, NFS only
PSR_type18	18	1b	Header includes MAC, (VLAN/SNAP) IPv6, UDP, NFS only
Reserved	31:19	0x0	Reserved Write 0, ignore on read.

8.10.4 Replicated Packet Split Receive Type - RPLPSRTYPE (0x54C0; R/W)

This register enables or disables each type of header that needs to be split. This register controls the behavior of packets that are replicated to multiple queues.

Field	Bit(s)	Initial Value	Description
PSR_type0	0	0b	Header includes MAC (VLAN/SNAP) only
PSR_type1	1	1b	Header includes MAC, (VLAN/SNAP) IPv4 only
PSR_type2	2	1b	Header includes MAC, (VLAN/SNAP) IPv4, TCP only
PSR_type3	3	1b	Header includes MAC, (VLAN/SNAP) IPv4, UDP only
PSR_type4	4	1b	Header includes MAC, (VLAN/SNAP) IPv4, IPv6 only
PSR_type5	5	1b	Header includes MAC, (VLAN/SNAP) IPv4, IPv6, TCP only
PSR_type6	6	1b	Header includes MAC, (VLAN/SNAP) IPv4, IPv6, UDP only
PSR_type7	7	1b	Header includes MAC, (VLAN/SNAP) IPv6 only
PSR_type8	8	1b	Header includes MAC, (VLAN/SNAP) IPv6, TCP only
PSR_type9	9	1b	Header includes MAC, (VLAN/SNAP) IPv6, UDP only
Reserved_1	10	1b	Reserved Write 1, ignore on read.
PSR_type11	11	1b	Header includes MAC, (VLAN/SNAP) IPv4, TCP, NFS only
PSR_type12	12	1b	Header includes MAC, (VLAN/SNAP) IPv4, UDP, NFS only
Reserved_1	13	1b	Reserved Write 1, ignore on read.
PSR_type14	14	1b	Header includes MAC, (VLAN/SNAP) IPv4, IPv6, TCP, NFS only
PSR_type15	15	1b	Header includes MAC, (VLAN/SNAP) IPv4, IPv6, UDP, NFS only
Reserved_1	16	1b	Reserved Write 1, ignore on read.
PSR_type17	17	1b	Header includes MAC, (VLAN/SNAP) IPv6, TCP, NFS only
PSR_type18	18	1b	Header includes MAC, (VLAN/SNAP) IPv6, UDP, NFS only
Reserved	31:19	0x0	Reserved Write 0, ignore on read.

8.10.5 Receive Descriptor Base Address Low - RDBAL (0xC000 + 0x40*n [n=0...7]; R/W)

This register contains the lower bits of the 64-bit descriptor base address. The lower four bits are always ignored. The Receive Descriptor Base Address must point to a 128 byte-aligned block of data.

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x2800, 0x2900, 0x2A00 & 0x2B00 respectively.

Field[1]	Bit(s)	Initial Value	Description
Lower_0	6:0	0x0	Ignored on writes. Returns 0x0 on reads.
RDBAL	31:7	X	Receive Descriptor Base Address Low

1. Software should program *RDBAL[n]* register only when queue is disabled (*RXDCTL[n].Enable* = 0).

8.10.6 Receive Descriptor Base Address High - RDBAH (0xC004 + 0x40*n [n=0...7]; R/W)

This register contains the upper 32 bits of the 64-bit descriptor base address.

Field[1]	Bit(s)	Initial Value	Description
RDBAH	31:0	X	Receive Descriptor Base Address [63:32]

1. Software should program *RDBAH[n]* register only when queue is disabled (*RXDCTL[n].Enable* = 0).

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x2804, 0x2904, 0x2A04 & 0x2B04 respectively.

8.10.7 Receive Descriptor Ring Length - RDLEN (0xC008 + 0x40*n [n=0...7]; R/W)

This register sets the number of bytes allocated for descriptors in the circular descriptor buffer. It must be 128-byte aligned.

Field[1]	Bit(s)	Initial Value	Description
0	6:0	0x0	Ignore on writes. Bits 6:0 must be set to zero. Bits 4:0 always read as zero.
LEN	19:7	0x0	Descriptor Ring Length (number of 8 descriptor sets) Note: maximum allowed value in *RDLEN* field 19:0 is 0x80000 (32K descriptors).
Reserved	31:20	0x0	Reserved Write 0, ignore on read.

1. Software should program *RDLEN[n]* register only when queue is disabled (*RXDCTL[n].Enable* = 0).

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x2808, 0x2908, 0x2A08 & 0x2B08 respectively.

8.10.8 Receive Descriptor Head - RDH (0xC010 + 0x40*n [n=0...7]; RO)

The value in this register might point to descriptors that are still not in host memory. As a result, the host cannot rely on this value in order to determine which descriptor to process.

Field	Bit(s)	Initial Value	Description
RDH	15:0	0x0	Receive Descriptor Head
Reserved	31:16	0x0	Reserved Write 0, ignore on read.

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x2810, 0x2910, 0x2A10 & 0x2B10 respectively.

8.10.9 Receive Descriptor Tail - RDT (0xC018 + 0x40*n [n=0...7]; R/W)

This register contains the tail pointers for the receive descriptor buffer. The register points to a 16-byte datum. Software writes the tail register to add receive descriptors to the hardware free list for the ring.

Note: Writing the RDT register while the corresponding queue is disabled is ignored by the I350.

In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x2818, 0x2918, 0x2A18& 0x2B18 respectively.

Field	Bit(s)	Initial Value	Description
RDT	15:0	0x0	Receive Descriptor Tail
Reserved	31:16	0x0	Reserved. Write 0, ignore on read.

8.10.10 Receive Descriptor Control - RXDCTL (0xC028 + 0x40*n [n=0...7]; R/W)

This register controls the fetching and write-back of receive descriptors. The three threshold values are used to determine when descriptors are read from and written to host memory. The values are in units of descriptors (each descriptor is 16 bytes).

Field	Bit(s)	Initial Value	Description
PTHRESH	4:0	0xC	Prefetch Threshold PTHRESH is used to control when a prefetch of descriptors is considered. This threshold refers to the number of valid, unprocessed receive descriptors the I350 has in its on-chip buffer. If this number drops below PTHRESH, the algorithm considers pre-fetching descriptors from host memory. This fetch does not happen unless there are at least HTHRESH valid descriptors in host memory to fetch. **Note:** HTHRESH should be given a non zero value each time PTHRESH is used. Possible values for this field are 0 to 16.
Reserved	7:5	0x0	Reserved Write 0, ignore on read.
HTHRESH	12:8	0xA	Host Threshold Field defines when receive descriptor prefetch is performed. Each time enough valid descriptors, as defined in the HTHRESH field, are available in host memory a prefetch is performed. Possible values for this field are 0 to 16.
Reserved	15:13	0x0	Reserved Write 0, ignore on read.
WTHRESH	20:16	0x1	Write-Back Threshold *WTHRES*H controls the write-back of processed receive descriptors. This threshold refers to the number of receive descriptors in the on-chip buffer that are ready to be written back to host memory. In the absence of external events (explicit flushes), the write-back occurs only after at least *WTHRESH* descriptors are available for write-back. Possible values for this field are 0 to 15. **Note:** Since the default value for write-back threshold is 1b, the descriptors are normally written back as soon as one cache line is available. *WTHRESH* must contain a non-zero value to take advantage of the write-back bursting capabilities of the I350. **Note**: It's recommended not to place a value above 0xC in the *WTHRESH* field.
Reserved	24:21	0x0	Reserved
ENABLE	25	0b	Receive Queue Enable When set, the *Enable* bit enables the operation of the specific receive queue. 1b =Enables queue. 0b =Disables queue. Setting this bit initializes Head and Tail registers (*RDH*[n] and *RDT*[n]) of the specific queue. Until then, the state of the queue is kept and can be used for debug purposes. When disabling a queue, this bit is cleared only after all activity in the queue has stopped. **Note:** When receive queue is enabled and descriptors exist, descriptors and are fetched immediately. Actual receive activity on port starts only if the *RCTL.RXEN* bit is set.
SWFLUSH (WC)	26	0b	Receive Software Flush Enables software to trigger receive descriptor write-back flushing, independently of other conditions. This bit is cleared by hardware after write-back flush is triggered (may take a number of cycles).
Reserved	31:27	0x0	Reserved

Note: In order to keep compatibility with 82575, for queues 0-3, these registers are aliased to addresses 0x2828, 0x2928, 0x2A28 & 0x2B28 respectively.

8.10.11 Receive Queue drop packet count - RQDPC (0xC030 + 0x40*n [n=0...7]; RC)

Field	Bit(s)	Initial Value	Description
RQDPC	31:0	0x0	Receive Queue drop packet count - counts the number of packets dropped by a queue due to lack of descriptors available. Note: Counter is stuck when reaching a value of 0xFFFFFFFF.

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x2830, 0x2930, 0x2A30 & 0x2B30 respectively.

Packets dropped due to the queue being disabled may not be counted by this register.

8.10.12 Receive Checksum Control - RXCSUM (0x5000; R/W)

The Receive Checksum Control register controls the receive checksum off loading features of the I350. The I350 supports the off loading of three receive checksum calculations: the Packet Checksum, the IP Header Checksum, and the TCP/UDP Checksum.

Note: This register should only be initialized (written) when the receiver is not enabled (only write this register when RCTL.RXEN = 0b)

Field	Bit(s)	Initial Value	Description
PCSS	7:0	0x0	Packet Checksum Start Controls the packet checksum calculation. The packet checksum shares the same location as the RSS field and is reported in the receive descriptor when the *RXCSUM.PCSD* bit is cleared. If the *RXCSUM.IPPCSE* is set, the Packet checksum is aimed to accelerate checksum calculation of fragmented UDP packets. Please refer to section Section 7.1.7.2 for detailed explanation. If *RXCSUM.IPPCSE* is cleared (the default value), the checksum calculation that is reported in the Rx Packet checksum field is the unadjusted 16-bit ones complement of the packet. The packet checksum starts from the byte indicated by *RXCSUM.PCSS* (0b corresponds to the first byte of the packet), after VLAN stripping if enabled by the *CTRL.VME*. For example, for an Ethernet II frame encapsulated as an 802.3ac VLAN packet and with *RXCSUM.PCSS* set to 14, the packet checksum would include the entire encapsulated frame, excluding the 14-byte Ethernet header (DA, SA, Type/Length) and the 4-byte VLAN tag. The packet checksum does not include the Ethernet CRC if the *RCTL.SECRC* bit is set. Software must make the required offsetting computation (to back out the bytes that should not have been included and to include the pseudo-header) prior to comparing the packet checksum against the L4 checksum stored in the packet checksum. The partial checksum in the descriptor is aimed to accelerate checksum calculation of fragmented UDP packets. ***Note:*** The PCSS value should point to a field that is before or equal to the IP header start. Otherwise the IP header checksum or TCP/UDP checksum is not calculated correctly.
IPOFLD	8	1b	IP Checksum Off-load Enable *RXCSUM.IPOFLD* is used to enable the IP Checksum off-loading feature. If *RXCSUM.IPOFLD* is set to 1b, the I350 calculates the IP checksum and indicates a pass/fail indication to software via the IP Checksum Error bit (*IPE*) in the *Error* field of the receive descriptor. Similarly, if *RXCSUM.TUOFLD* is set to 1b, the I350 calculates the TCP or UDP checksum and indicates a pass/fail indication to software via the TCP/UDP Checksum Error bit (*RDESC.L4E*). This applies to checksum off loading only. Supported frame types: Ethernet II Ethernet SNAP
TUOFLD	9	1b	TCP/UDP Checksum Off-load Enable
ICMPv6XSUM	10	1b	ICMPv6 Checksum Enable 0b - Disable ICMPv6 checksum calculation. 1b - Enable ICMPv6 checksum calculation. ***Note:*** ICMPv6 checksum offload is supported only for packets sent to Firmware for Proxying.
CRCOFL	11	0b	CRC32 Offload Enable Enables the SCTP CRC32 checksum off-loading feature. If *RXCSUM.CRCOFL* is set to 1b, the I350 calculates the CRC32 checksum and indicates a pass/fail indication to software via the CRC32 *Checksum Valid* bit (*RDESC.L4I*) in the *Extended Status* field of the receive descriptor. In non I/OAT, this bit is read only as 0b.
IPPCSE	12	0b	IP Payload Checksum Enable See PCSS description.
PCSD	13	0b	Packet Checksum Disable The packet checksum and IP identification fields are mutually exclusive with the RSS hash. Only one of the two options is reported in the Rx descriptor. *RXCSUM.PCSD* Legacy Rx Descriptor (*SRRCTL.DESCTYPE* = 000b): 0b (checksum enable) - Packet checksum is reported in the Rx descriptor. 1b (checksum disable) - Not supported. *RXCSUM.PCSD* Extended or Header Split Rx Descriptor (*SRRCTL.DESCTYPE* not equal 000b): 0b (checksum enable) - checksum and IP identification are reported in the Rx descriptor. 1b (checksum disable) - RSS Hash value is reported in the Rx descriptor.
Reserved	31:14	0x0	Reserved Write 0, ignore on read.

8.10.13 Receive Long Packet Maximum Length - RLPML (0x5004; R/W)

Field	Bit(s)	Initial Value	Description
RLPML	13:0	0x2600	Maximum allowed long packet length. This length is the global length of the packet including all the potential headers of suffixes in the packet.
Reserved	31:14	0x0	Reserved Write 0, ignore on read.

8.10.14 Receive Filter Control Register - RFCTL (0x5008; R/W)

Field	Bit(s)	Initial Value	Description
Reserved	5:0	1b	Reserved Write 0, ignore on read.
NFSW_DIS	6	0b	NFS Write Disable Disables filtering of NFS write request headers.
NFSR_DIS	7	0b	NFS Read Disable Disables filtering of NFS read reply headers.
NFS_VER	9:8	00b	NFS Version 00b = NFS version 2. 01b = NFS version 3. 10b = NFS version 4. 11b = Reserved for future use.
Reserved	10	0b	Reserved Write 0, ignore on read.
IPv6XSUM_DIS	11	0b	IPv6 XSUM Disable Disables XSUM on IPv6 packets.
Reserved	14:12	0x0	Reserved Write 0, ignore on read.
Reserved	17:15	0x0	Reserved Write 0, ignore on read.
LEF	18	0b	Forward Length Error Packet 0b = packet with length error are dropped. 1b = packets with length error are forwarded to the host. A packet with length error is a 802.3 packet where the Length field value doesn't match the actual length of the packet.
SYNQFP	19	0b	Defines the priority between SYNQF & 2 tuple filter 0b = 2-tuple filter priority 1b = SYN filter priority.
Reserved	31:20	0x0	Reserved Write 0, ignore on read.
Reserved	31:20	0x08	Reserved Should be written with 0b to ensure future capability.

8.10.15 Multicast Table Array - MTA (0x5200 + 4*n [n=0...127]; R/W)

There is one register per 32 bits of the Multicast Address Table for a total of 128 registers. Software must mask to the desired bit on reads and supply a 32-bit word on writes. The first bit of the address used to access the table is set according to the RX_CTRL.MO field.

Note: All accesses to this table must be 32 bit.

Field	Bit(s)	Initial Value	Description
Bit Vector	31:0	X	Word wide bit vector specifying 32 bits in the multicast address filter table.

Figure 8-1 shows the multicast lookup algorithm. The destination address shown represents the internally stored ordering of the received DA. Note that bit 0 indicated in this diagram is the first on the wire.

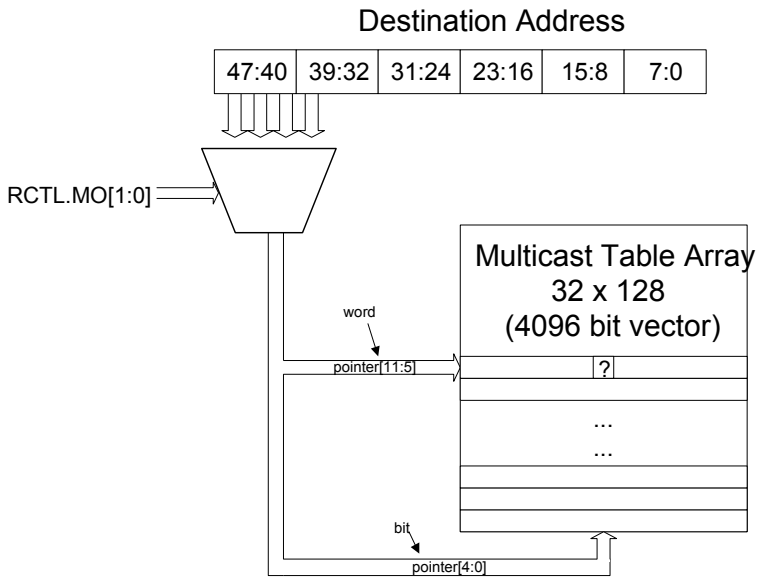


Figure 8-1 Multicast Table Array

8.10.16 Receive Address Low - RAL (0x5400 + 8*n [n=0...15]; 0x54E0 + 8*n [n=0...15]; R/W)

While "n" is the exact unicast/multicast address entry and it is equal to 0,1,...15.

These registers contain the lower bits of the 48 bit Ethernet address. All 32 bits are valid.

These registers are reset by a software reset or platform reset. If an EEPROM is present, the first register (RAL0) is loaded from the EEPROM after a software or platform reset.

Note: The RAL field should be written in network order.

Field	Bit(s)	Initial Value	Description
RAL	31:0	X	Receive address low Contains the lower 32-bit of the 48-bit Ethernet address.

8.10.17 Receive Address High - RAH (0x5404 + 8*n [n=0...15]; 0x54E4 + 8*n [n=0...15]; R/W)

These registers contain the upper bits of the 48 bit Ethernet address. The complete address is [RAH, RAL]. The RAH.AV bit determines whether this address is compared against the incoming packet.

The RAH.ASEL field enables the I350 to perform special filtering on receive packets.

After reset, if an EEPROM is present, the first register (Receive Address Register 0) is loaded from the IA field in the EEPROM with its Address Select field set to 00b and its Address Valid field set to 1b. If no EEPROM is present, the Address Valid field is set to 0b and the Address Valid field for all of the other registers is set to 0b.

Note: The RAH field should be written in network order.

The first receive address register (RAH[0]) is also used for exact match pause frame checking (DA matches the first register). As a result, RAH[0] should always be used to store the individual Ethernet MAC address of the I350.

Field	Bit(s)	Initial Value	Description
RAH	15:0	X	Receive address High Contains the upper 16 bits of the 48-bit Ethernet address.
ASEL	17:16	X	Address Select Selects how the address is to be used in the address filtering. 00b = Destination address (required for normal mode) 01b = Source address. This mode should not be used in virtualization mode. 10b = Reserved 11b = Reserved
POOLSEL	25:18	0x0	Pool Select In virtualization modes (*MRQC.Multiple Receive Queues Enable* = 011b) indicates which Pool should get the packets matching this MAC address. This field is a bit map (bit per VM) where more than one bit can be set according to the limitations defined in Section 7.8.3.4. If all the bits are zero, this address is used only for L2 filtering and is not used as part of the queueing decision.
Reserved	29:26	0x0	Reserved Write 0, Ignore on reads.
TRMCST	30	0x0	Treat as Multicast. A transmit packet matching this destination address will be forwarded to the network even if forwarded also to a local pool.
AV	31	1b for RAH[0] if NVM is present, 0b otherwise 0b for all others.	Address Valid Cleared after master reset. If an EEPROM is present, the *Address Valid* field of the Receive Address Register 0 is set to 1b after a software or PCI reset or EEPROM read.

8.10.18 VLAN Filter Table Array - VFTA (0x5600 + 4*n [n=0...127]; R/W)

There is one register per 32 bits of the VLAN Filter Table. The size of the word array depends on the number of bits implemented in the VLAN Filter Table. Software must mask to the desired bit on reads and supply a 32-bit word on writes.

Note: All accesses to this table must be 32 bit.

The algorithm for VLAN filtering using the VFTA is identical to that used for the Multicast Table Array. Refer to Section 8.10.15 for a block diagram of the algorithm. If VLANs are not used, there is no need to initialize the VFTA.

Field	Bit(s)	Initial Value	Description
Bit Vector	31:0	X	Double-word wide bit vector specifying 32 bits in the VLAN Filter table.

8.10.19 Multiple Receive Queues Command Register - MRQC (0x5818; R/W)

Field	Bit(s)	Initial Value	Description
Multiple Receive Queues Enable	2:0	0x0	Multiple Receive Queues Enable Enables support for Multiple Receive Queues and defines the mechanism that controls queue allocation. 000b = Multiple receive queues as defined by filters (2-tuple filters, L2 Ether-type filters, SYN filter and Flex Filters). 001b = Reserved.Reserved 010b = Multiple receive queues as defined by filters and RSS for 8 queues[1]. 011b = Multiple receive queues as defined by VMDq based on packet destination MAC address (*RAH.POOLSEL*) and Ether-type queuing decision filters. 100b = Reserved 101b = Reserved. 110b = Reserved 111b = Reserved. If VT is not supported, the only allowed values for this field are 000b, and 010b. Writing any other value is ignored. The allowed values for this field are 000b, 010b and 011b. Any other value is ignored.
Def_Q	5:3	0x0	Defines the default queue in non VMDq mode according to value of the *Multiple Receive Queues Enable* field. If Multiple Receive Queues Enable = 000b: Def_Q defines the destination of all packets not forwarded by filters. 001b: Def_Q field is ignored 010b: Def_Q defines the destination of all packets not forwarded by RSS or filters. 011b - Def_Q field is ignored. Queueing decision of all packets not forwarded by MAC address and Ether-type filters is according to VT_CTL.DEF_PL field. 100-101b: Def_Q field is ignored. 110b: Def_Q field is ignored. Note: In VMDq mode (*Multiple Receive Queues Enable* = 011b) the default queue is set according to the *VT_CTL.DEF_PL* if packet passes MAC Address filtering of a filter with *RAH.POOLSEL* = 0x0 or is a broadcast or multicast packet and does not match Ether-type queuing decision filters.

Field	Bit(s)	Initial Value	Description
Reserved	15:6	0x0	Reserved. Write 0, ignore on read.
RSS Field Enable	31:16	0x0	Each bit, when set, enables a specific field selection to be used by the hash function. Several bits can be set at the same time. Bit[16] = Enable TcpIPv4 hash function Bit[17] = Enable IPv4 hash function Bit[18] = Enable TcpIPv6Ex hash function Bit[19] = Enable IPv6Ex hash function Bit[20] = Enable IPv6 hash function Bit[21] = Enable TCPIPv6 hash function Bit[22] = Enable UDPIPv4 Bit[23] = Enable UDPIPv6 Bit[24] = Enable UDPIPv6Ext ReservedBits[31:26] - Reserved Zero

1. Note that the *RXCSUM.PCSD* bit should be set to enable reception of the RSS hash value in the receive descriptor.

Note: The *MRQC.Multiple Receive Queues Enable* field is used to enable/disable RSS hashing and also to enable multiple receive queues. Disabling this feature is not recommended. Model usage is to reset the I350 after disabling the RSS.

8.10.20 RSS Random Key Register - RSSRK (0x5C80 + 4*n [n=0...9]; R/W)

Field	Bit(s)	Initial Value	Description
K0	7:0	0x0	Byte n*4 of the RSS random key (n=0,1,...9).
K1	15:8	0x0	Byte n*4+1 of the RSS random key (n=0,1,...9).
K2	23:16	0x0	Byte n*4+2 of the RSS random key (n=0,1,...9).
K3	31:24	0x0	Byte n*4+3 of the RSS random key (n=0,1,...9).

The RSS Random Key Register stores a 40 byte key used by the RSS hash function.

31 24	23 16	15 8	7 0
K[3]	K[2]	K[1]	K[0]
...	...	...	...
K[39]	...	...	K[36]

8.10.21 Redirection Table - RETA (0x5C00 + 4*n [n=0...31]; R/W)

The redirection table is a 128-entry table with each entry being eight bits wide. Only 1 to 3 bits of each entry are used to store the queue index. The table is configured through the following R/W registers.

Field	Bit(s)	Initial Value	Description
Entry 0	7:0	0x0	Determines the tag value and physical queue for index 4*n+0 (n=0...31).
Entry 1	15:8	0x0	Determines the tag value and physical queue for index 4*n+1 (n=0...31).
Entry 2	23:16	0x0	Determines the tag value and physical queue for index 4*n+2 (n=0...31).
Entry 3	31:24	0x0	Determines the tag value and physical queue for index 4*n+3 (n=0...31).

31	24	23	16	15	8	7	0
Tag 3		Tag 2		Tag 1		Tag 0	
...		...		...		...	
Tag 127		...		...		...	

Each entry (byte) of the redirection table contains the following:

7:3	2:0
Reserved	Queue index

- Bits [7:3] - Reserved
- Bits [2:0] - Queue index for all pools or in regular RSS. In RSS only mode, all bits are used.

The contents of the redirection table are not defined following reset of the Memory Configuration Registers. System software must initialize the Table prior to enabling multiple receive queues. It can also update the redirection table during run time. Such updates of the table are not synchronized with the arrival time of received packets. Therefore, it is not guaranteed that a table update takes effect on a specific packet boundary.

Note: In case the operating system provides a redirection Table whose size is smaller than 128 bytes, the software usually replicates the operating system-provided redirection table to span the whole 128 bytes of the hardware's redirection table.

8.11 Filtering Register Descriptions

8.11.1 Immediate Interrupt RX - IMIR (0x5A80 + 4*n [n=0...7]; R/W)

This *IMIR[n]* register, *TTQF[n]* register and the *IMIREXT[n]* register define the filtering required to indicate which packet triggers a low latency interrupt (immediate interrupt). The registers can also be used for queuing and deciding on timestamp of a packet.

Notes:

1. The *Port* field should be written in network order.

2. If one of the actions for this filter is set, then at least one of the *IMIR[n].PORT_BP*, *IMIR[n].Size_BP*, the Mask bits in the *TTQF[n]* register or the *IMIREXT.CtrlBit_BP* bits should be cleared.
3. The value of the *IMIR* and *IMIREXT* registers after reset is unknown (apart from the *IMIR.Immediate Interrupt* bit which is guaranteed to be cleared). Therefore, both registers should be programmed before *IMIR.Immediate Interrupt* is set for a given flow.

Field	Bit(s)	Initial Value	Description
Destination Port	15:0	0x0	Destination TCP port This field is compared with the Destination TCP port in incoming packets. Only a packet with a matching destination TCP port will trigger an immediate interrupt (if *IMIR[n].Immediate Interrupt* is set to 1b) and trigger the actions defined in the appropriate *TTQF[n]* register if all other filtering conditions are met. ***Note:*** Enabled by the *IMIR.PORT_BP* bit.
Immediate Interrupt	16	0b	Enables issuing an immediate interrupt when the following conditions are met: • The 2-tuple filter associated with this register matches • The Length filter associated with this filter matches • The TCP flags filter associated with this filter matches
PORT_BP	17	X	Port Bypass When set to 1b, the TCP port check is bypassed and only other conditions are checked. When set to 0b, the TCP port is checked to fit the port field.
Reserved	28:18	0x0	Reserved Write 0, ignore on read.
Filter Priority	31:29	000b	Defines the priority of the filter assuming two filters with same priority don't match. If two filters with the same priority match the incoming packet, the first filter (lowest ordinal number) is used in order to define the queue destination of this packet.

8.11.2 Immediate Interrupt Rx Ext. - IMIREXT (0x5AA0 + 4*n [n=0...7]; R/W)

Field	Bit(s)	Initial Value	Description
Size_Thresh	11:0	X	Size Threshold These 12 bits define a size threshold. Only a packet with a length below this threshold will trigger an immediate interrupt (if *IMIR[n].Immediate Interrupt* is set to 1b) and trigger the actions defined in the appropriate *TTQF[n]* register (if *TTQF[n].Queue Enable* is set to 1b) if all other filtering conditions are met. ***Notes:*** 1. Enabled by the *IMIREXT.Size_BP* bit. 2. The size used for this comparison is the size of the packet as forwarded to the host and does not include any of the fields stripped by the MAC (VLAN or CRC). As a result, setting the *RCTL.SECRC* and *CTRL.VME* bits should be taken into account while calculating the size threshold. 3. In virtualization mode, where *DVMOLR.CRC* strip and *DVMOLR.STRVLAN* are used, the Size_thresh should include the VLAN and the CRC.
Size_BP	12	X	Size Bypass When 1b, the size check is bypassed. When 0b, the size check is performed.
CtrlBit	18:13	X	Control Bit Defines TCP control bits used to generate immediate interrupt and trigger filter. Only a received packet with the corresponding TCP control bits set to 1b will trigger an immediate interrupt (if *IMIR[n].Immediate Interrupt* is set to 1b) and trigger the actions defined in the appropriate *TTQF[n]* register (if *TTQF[n].Queue Enable* is set to 1b) if all other filtering conditions are met. Bit 13 (URG): Urgent pointer field significant Bit 14 (ACK): Acknowledgment field Bit 15 (PSH): Push function Bit 16 (RST): Reset the connection Bit 17 (SYN): Synchronize sequence numbers Bit 18 (FIN): No more data from sender ***Note:*** Enabled by the *IMIREXT.CtrlBit_BP* bit.
CtrlBit_BP	19	X	Control Bits Bypass When set to 1b, the control bits check is bypassed. When set to 0b, the control bits check is performed.
Reserved	31:20	0x0	Reserved Write 0, ignore on read.

8.11.3

8.11.4 2-tuples Queue Filter - TTQF (0x59E0 +

4*n[n=0...7]; RW)

Field	Bit(s)	Initial Value	Description
Protocol	7:0	0x0	IP L4 protocol, part of the 2-tuple queue filters. This field is compared with the IP L4 protocol in incoming packets. Only a packet with a matching IP L4 protocol will trigger an immediate interrupt (if *IMIR[n].Immediate Interrupt* is set to 1b) and trigger the actions defined in the appropriate *TTQF[n]* register (if *TTQF[n].Queue Enable* is set to 1b) if all other filtering conditions are met.
Queue Enable	8	0b	When set, enables filtering of Rx packets by the 2-tuples defined in this filter to the queue indicated in this register.
Reserved	14:9	0x0	Reserved Write 0, ignore on read.
Reserved_1	15	1b	Reserved Write 1b, ignore on read.
Rx Queue	18:16	0x0	Identifies the Rx queue associated with this 2-tuple filter. If the VF Mask bit is set, the queue number is used as an offset to the VM list and does not override it.
Reserved	26:19	0x0	Reserved Write 0, ignore on read.
1588 time stamp	27	0b	When set, packets that match this filter are time stamped according to the IEEE 1588 specification. ***Note:*** Packet will be time stamped only if it matches IEEE 1588 protocol according to the definition in the *TSYNCRXCTL.Type* field.
Mask	31:28	0xF	Mask bits for the 2-tuple fields. The corresponding field participates in the match if the bit below is cleared: Bit 28 - Mask protocol comparison Bits 31 - 29 Reserved

8.11.5 Immediate Interrupt Rx VLAN Priority - IMIRVP (0x5AC0; R/W)

Field	Bit(s)	Initial Value	Description
Vlan_Pri	2:0	000b	VLAN Priority This field includes the VLAN priority threshold. When Vlan_pri_en is set to 1b, then an incoming packet with a VLAN tag with a priority field equal or higher to VlanPri triggers an immediate interrupt, regardless of the EITR moderation.
Vlan_pri_en	3	0b	VLAN Priority Enable When set to 1b, an incoming packet with VLAN tag with a priority equal or higher to Vlan_Pri triggers an immediate interrupt, regardless of the EITR moderation. When set to 0b, the interrupt is moderated by EITR.
Reserved	31:4	0x0	Reserved Write 0, ignore on read.

8.11.6 SYN Packet Queue Filter - SYNQF (0x55FC; RW)

Field	Bit(s)	Initial Value	Description
Queue Enable	0	0b	When set, enables forwarding of Rx packets to the queue indicated in this register.
Rx Queue	3:1	0x0	Identifies an Rx queue associated with SYN packets.
Reserved	31:4	0x0	Reserved Write 0, ignore on read.

8.11.7 EType Queue Filter - ETQF (0x5CB0 + 4*n[n=0...7]; RW)

Field	Bit(s)	Initial Value	Description
EType	15:0	0x0	Identifies the protocol running on top of IEEE 802. Used to forward Rx packets containing this EType to a specific Rx queue.
Rx Queue	18:16	0x0	Identifies the receive queue associated with this EType.
Reserved	25:19	0x0	Reserved Write 0, ignore on read.
Filter enable	26	0b	When set, this filter is valid. Any of the actions controlled by the following fields are gated by this field.
Reserved	28:27	0x0	Reserved Write 0, ignore on read.
Immediate Interrupt	29	0x0	When set, packets that match this filter generate an immediate interrupt.
1588 time stamp	30	0b	When set, packets with this EType are time stamped according to the IEEE 1588 specification. ***Note:*** Packet will be time stamped only if it matches IEEE 1588 protocol according to the definition in the *TSYNCRXCTL.Type* field.
Queue Enable	31	0b	When set, enables filtering of Rx packets by the EType defined in this register to the queue indicated in this register.

8.12 Transmit Register Descriptions

8.12.1 Transmit Control Register - TCTL (0x0400; R/W)

This register controls all transmit functions for the I350.

Field	Bit(s)	Initial Value	Description
Reserved	0	0b	Reserved Write 0, ignore on read.
EN	1	0b	Transmit Enable The transmitter is enabled when this bit is set to 1b. Writing 0b to this bit stops transmission after any in progress packets are sent. Data remains in the transmit FIFO until the device is re-enabled. Software should combine this operation with reset if the packets in the TX FIFO should be flushed.
Reserved	2	0b	Reserved Write 0, ignore on read.
PSP	3	1b	Pad Short Packets 0b = Do not pad. 1b = Pad. Padding makes the packet 64 bytes long. This is not the same as the minimum collision distance. If padding of short packets is allowed, the total length of a packet not including FCS should be not less than 17 bytes.
CT	11:4	0xF	Collision Threshold This determines the number of attempts at retransmission prior to giving up on the packet (not including the first transmission attempt). While this can be varied, it should be set to a value of 15 in order to comply with the IEEE specification requiring a total of 16 attempts. The Ethernet back-off algorithm is implemented and clamps to the maximum number of slot-times after 10 retries. This field only has meaning when in half-duplex operation. Note: Software can choose to abort packet transmission in less than the Ethernet mandated 16 collisions. For this reason, hardware provides CT support.
BST	21:12	0x40	Back-Off Slot Time This value determines the back-off slot time value in byte time.
SWXOFF	22	0b	Software XOFF Transmission When set to 1b, the I350 schedules the transmission of an XOFF (PAUSE) frame using the current value of the PAUSE timer (*FCTTV.TTV*). This bit self-clears upon transmission of the XOFF frame. Note: While 802.3x flow control is only defined during full duplex operation, the sending of PAUSE frames via the SWXOFF bit is not gated by the duplex settings within the I350. Software should not write a 1b to this bit while the I350 is configured for half-duplex operation.
Reserved	23	0b	Reserved
RTLC	24	0b	Re-transmit on Late Collision When set, enables the I350 to re-transmit on a late collision event. Note: RTLC configures the I350 to perform retransmission of packets when a late collision is detected. Note that the collision window is speed dependent: 64 bytes for 10/100 Mb/s and 512 bytes for 1000 Mb/s operation. If a late collision is detected when this bit is disabled, the transmit function assumes the packet has successfully transmitted. This bit is ignored in full-duplex mode.
Reserved	25	0b	Reserved
Reserved	27:26	0x1	Reserved
Reserved	31:28	0xA	Reserved

8.12.2 Transmit Control Extended - TCTL_EXT (0x0404; R/W)

This register controls late collision detection.

The COLD field is used to determine the latest time in which a collision indication is considered as a valid collision and not a late collision. When using the internal PHY, the default value of 0x40 provides a behavior consistent with the 802.3 spec requested behavior. However, when using an SGMII connected external PHY, the SGMII interface adds some delay on top of the time budget allowed by the specification (collisions in valid network topographies even after 512 bit time can be expected). In order to accommodate this condition, COLD should be updated to take the SGMII inbound and outbound delays.

Field	Bit(s)	Initial Value	Description
Reserved	9:0	0x40	Reserved. Write 0, ignore on read.
COLD	19:10	0x42	Collision Distance Used to determine the latest time in which a collision indication is considered as a valid collision and not a late collision.
Reserved	31:20	0x0	Reserved. Write 0, ignore on read.

8.12.3 Transmit IPG Register - TIPG (0x0410; R/W)

This register controls the Inter Packet Gap (IPG) timer.

Field	Bit(s)	Initial Value	Description
IPGT	9:0	0x08	IPG Back to Back Specifies the IPG length for back to back transmissions in both full and half duplex. Measured in increments of the MAC clock: 8 ns MAC clock when operating @ 1 Gb/s. 80 ns MAC clock when operating @ 100 Mb/s. 800 ns MAC clock when operating @ 10 Mb/s. IPGT specifies the IPG length for back-to-back transmissions in both full duplex and half duplex. Note that an offset of 4 byte times is added to the programmed value to determine the total IPG. As a result, a value of 8 is recommended to achieve a 12 byte time IPG.
IPGR1	19:10	0x04	IPG Part 1 Specifies the portion of the IPG in which the transmitter defers to receive events. IPGR1 should be set to 2/3 of the total effective IPG (8). Measured in increments of the MAC clock: 8 ns MAC clock when operating @ 1 Gb/s. 80 ns MAC clock when operating @ 100 Mb/s 800 ns MAC clock when operating @ 10 Mb/s.

Field	Bit(s)	Initial Value	Description
IPGR	29:20	0x06	IPG After Deferral Specifies the total IPG time for non back-to-back transmissions (transmission following deferral) in half duplex. Measured in increments of the MAC clock: 8 ns MAC clock when operating @ 1 Gb/s. 80 ns MAC clock when operating @ 100 Mb/s 800 ns MAC clock when operating @ 10 Mb/s. An offset of 5-byte times must be added to the programmed value to determine the total IPG after a defer event. A value of 7 is recommended to achieve a 12-byte effective IPG. Note that the IPGR must never be set to a value greater than IPGT. If IPGR is set to a value equal to or larger that IPGT, it overrides the IPGT IPG setting in half duplex resulting in inter-packet gaps that are larger then intended by IPGT. In this case, full duplex is unaffected and always relies on IPGT.
Reserved	31:30	00b	Reserved Write 0, ignore on read.

8.12.4 Retry Buffer Control – RETX_CTL (0x041C; RW)

This register controls the collision retry buffer.

Field	Bit(s)	Initial Value	Description
Water Mark	3:0	0x3	Retry buffer water mark. This parameters defines the minimal number of QWords that should be present in the retry buffer before transmission is started.
Reserved	31:4	0x0	Reserved. Write 0, ignore on read.

8.12.5 DMA TX Control - DTXCTL (0x3590; R/W)

This register is used for controlling the DMA TX behavior.

Field	Bit(s)	Initial Value	Description
Reserved	1:0	0x0	Reserved. Write 0, ignore on read.
Enable_spoof_queue	2	0b	Enable Spoofing Queue 0b - Disable queue that exhibited spoofing behavior. 1b - Do not disable port that exhibited spoofing behavior.
disable_malicious_block	3	0b	Disable Malicious blocking 0b - Block queue that exhibited malicious behavior on Transmit path. 1b - Do not block queue that exhibited malicious behavior on Transmit path.
OutOfSyncDisable	4	0b	Disable Out Of Sync mechanism 0b = Out Of Sync mechanism is enabled. 1b = Out Of Sync mechanism is disabled.
MDP_EN	5	0b	Malicious driver protection enable 0b = mechanism is disabled. 1b = mechanism is enabled.

Field	Bit(s)	Initial Value	Description
SPOOF_INT	6	0b	Interrupt on spoof behavior detection 0b = mechanism is disabled. 1b = mechanism is enabled.
Count CRC	7	1b	If set, the CRC is counted as part of the packet bytes statistics in per VF statistics (VFGORC, VFGOTC, VFGORLBC and VFGOTLBC).
Reserved	31:8	0x0	Reserved Write 0, ignore on read.

8.12.6 DMA TX TCP Flags Control Low - DTXTCPFLGL (0x359C; RW)

This register holds the buses that "AND" the control flags in TCP header for the first and middle segments of a TSO packet. Refer to Section 7.2.4.7.1 and Section 7.2.4.7.2 for details on the use of this register.

Field	Bit(s)	Initial Value	Description
TCP_flg_first_seg	11:0	0xFF6	TCP Flags first segment. Bits that are used to execute an AND operation with the TCP flags in the TCP header in the first segment
Reserved	15:12	0x0	Reserved Write 0, ignore on read.
TCP_Flg_mid_seg	27:16	0xF76	TCP Flags middle segments. Bits that are used to execute an AND operation with the TCP flags in the TCP header in the middle segments
Reserved	31:28	0x0	Reserved Write 0, ignore on read.

8.12.7 DMA TX TCP Flags Control High - DTXTCPFLGH (0x35A0; RW)

This register holds the buses that "AND" the control flags in TCP header for the last segment of a TSO packet. Refer to Section 7.2.4.7.3 for details of use of this register

Field	Bit(s)	Initial Value	Description
TCP_Flg_lst_seg	11:0	0xF7F	TCP Flags last segment. Bits that are used to execute an AND operation with the TCP flags at TCP header in the last segment
Reserved	31:12	0x0	Reserved. Write 0, ignore on read.

8.12.8 DMA TX Max Total Allow Size Requests - DTXMXSZRQ (0x3540; RW)

This register limits the allowable size of concurrent outstanding TX read requests from the host memory on the PCIe. Limiting the size of concurrent outstanding PCIe requests allows low latency packet read requests to be serviced in a timely manner, as the low latency request is serviced right after current outstanding requests are completed.

Field	Bit(s)	Initial Value	Description
Max_bytes_num_req	11:0	0x10	Maximum allowable size of concurrent TX outstanding requests on PCIe. Field defines maximum size in 256 byte resolution of outstanding TX requests to be sent on PCIe. If total amount of outstanding TX requests is higher than defined in this field, no further TX outstanding requests are sent.
Reserved	31:12	0x0	Reserved Write 0, ignore on read.

8.12.9 DMA TX Maximum Packet Size - DTXMXPKTSZ (0x355C; RW)

This register limits the total number of data bytes that might be transmitted in a single frame. Reducing packet size enables better utilization of transmit buffer.

Field	Bit(s)	Initial Value	Description
MAX_TPKT_SIZE	8:0	0x98	Maximum transmit packet size that is allowed to be transmitted by the driver. Value entered is in 64 Bytes resolution. ***Notes:*** 1. Default value enables transmission of maximum sized 9,728 Byte Jumbo frames. 2. Values programmed in this field should not exceed 9,728 Bytes. 3. Value programmed should not exceed the TX buffer size programmed in the *ITPBS.TXPbsize* register.
Reserved	31:9	0x0	Reserved Write 0, ignore on read.

8.12.10 Transmit Descriptor Base Address Low - TDBAL (0xE000 + 0x40*n [n=0...7]; R/W)

These registers contain the lower 32 bits of the 64-bit descriptor base address. The lower 7 bits are ignored. The Transmit Descriptor Base Address must point to a 128-byte aligned block of data.

Field[1]	Bit(s)	Initial Value	Description
Lower_0	6:0	0x0	Ignored on writes. Returns 0x0 on reads.
TDBAL	31:7	X	Transmit Descriptor Base Address Low

1. Software should program *TDBAL[n]* register only when queue is disabled (*TXDCTL[n].Enable* = 0).

Note: In order to keep compatibility with 82575, for queues 0-3, these registers are aliased to addresses 0x3800, 0x3900, 0x3A00 & 0x3B00 respectively.

8.12.11 Transmit Descriptor Base Address High - TDBAH (0xE004 + 0x40*n [n=0...7]; R/W)

These registers contain the upper 32 bits of the 64-bit descriptor base address.

Field[1]	Bit(s)	Initial Value	Description
TDBAH	31:0	X	Transmit Descriptor Base Address [63:32]

1. Software should program *TDBAH[n]* register only when queue is disabled (*TXDCTL[n].Enable* = 0).

Note: In order to keep compatibility with 82575 for queues 0-3, these registers are aliased to addresses 0x3804, 0x3904, 0x3A04 & 0x3B04 respectively.

8.12.12 Transmit Descriptor Ring Length - TDLEN (0xE008 + 0x40*n [n=0...7]; R/W)

These registers contain the descriptor ring length. The registers indicates the length in bytes and must be 128-byte aligned.

Field[1]	Bit(s)	Initial Value	Description
0	6:0	0x0	Ignore on writes. Read back as 0x0.
LEN	19:7	0x0	Descriptor Ring Length (number of 8 descriptor sets) Note: maximum allowed value in *TDLEN* field 19:0 is 0x80000 (32K descriptors).
Reserved	31:20	0x0	Reserved Write 0, ignore on read.

1. Software should program *TDLEN[n]* register only when queue is disabled (*TXDCTL[n].Enable* = 0).

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x3808, 0x3908, 0x3A08 & 0x3B08 respectively.

8.12.13 Transmit Descriptor Head - TDH (0xE010 + 0x40*n [n=0...7]; RO)

These registers contain the head pointer for the transmit descriptor ring. It points to a 16-byte datum. Hardware controls this pointer.

Note: The values in these registers might point to descriptors that are still not in host memory. As a result, the host cannot rely on these values in order to determine which descriptor to release.

Field	Bit(s)	Initial Value	Description
TDH	15:0	0x0	Transmit Descriptor Head
Reserved	31:16	0x0	Reserved Write 0, ignore on read.

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x3810, 0x3910, 0x3A10 & 0x3B10 respectively.

8.12.14 Transmit Descriptor Tail - TDT (0xE018 + 0x40*n [n=0...7]; R/W)

These registers contain the tail pointer for the transmit descriptor ring and points to a 16-byte datum. Software writes the tail pointer to add more descriptors to the transmit ready queue. Hardware attempts to transmit all packets referenced by descriptors between head and tail.

Field	Bit(s)	Initial Value	Description
TDT	15:0	0x0	Transmit Descriptor Tail
Reserved	31:16	0x0	Reserved Write 0, ignore on read.

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x3818, 0x3918, 0x3A18 & 0x3B18 respectively.

8.12.15 Transmit Descriptor Control - TXDCTL (0xE028 + 0x40*n [n=0...7]; R/W)

These registers control the fetching and write-back operations of transmit descriptors. The three threshold values are used to determine when descriptors are read from and written to host memory. The values are in units of descriptors (each descriptor is 16 bytes).

Since write-back of transmit descriptors is optional (under the control of *RS* bit in the descriptor), not all processed descriptors are counted with respect to *WTHRESH*. Descriptors start accumulating after a descriptor with *RS* set is processed. In addition, with transmit descriptor bursting enabled, some descriptors are written back that did not have *RS* set in their respective descriptors.

Note: When *WTHRESH* = 0x0, only descriptors with the *RS* bit set are written back.

Field	Bit(s)	Initial Value	Description
PTHRESH	4:0	0x0	Prefetch Threshold Controls when a prefetch of descriptors is considered. This threshold refers to the number of valid, unprocessed transmit descriptors the I350 has in its on-chip buffer. If this number drops below PTHRESH, the algorithm considers pre-fetching descriptors from host memory. However, this fetch does not happen unless there are at least HTHRESH valid descriptors in host memory to fetch. **Note:** When PTHRESH is 0x0 a Transmit descriptor fetch operation is done when any valid descriptors are available in Host memory and space is available in internal buffer.
Reserved	7:5	0x0	Reserved Write 0, ignore on read.
HTHRESH	12:8	0x0	Host Threshold Prefetch of transmit descriptors is considered when number of valid transmit descriptors in host memory is at least *HTHRESH*. **Note:** *HTHRESH* should be given a non zero value each time PTHRESH is used.
Reserved	15:13	0x0	Reserved Write 0, ignore on read.

Field	Bit(s)	Initial Value	Description
WTHRESH	20:16	0x0	Write-Back Threshold Controls the write-back of processed transmit descriptors. This threshold refers to the number of transmit descriptors in the on-chip buffer that are ready to be written back to host memory. In the absence of external events (explicit flushes), the write-back occurs only after at least *WTHRESH* descriptors are available for write-back. Possible values for this field are 0 to 23. **Note:** Since the default value for write-back threshold is 0b, descriptors are normally written back as soon as they are processed. *WTHRESH* must be written to a non-zero value to take advantage of the write-back bursting capabilities of the I350.
Reserved	23:21	0x0	Reserved
Reserved	24	0b	Reserved Write 0, ignore on read.
ENABLE	25	0b	Transmit Queue Enable When set, this bit enables the operation of a specific transmit queue. Setting this bit initializes the Tail and Head registers (*TDT*[n] and *TDH*[n]) of a specific queue. Until then, the state of the queue is kept and can be used for debug purposes. When disabling a queue, this bit is cleared only after all transmit activity on this queue is stopped. **Note:** When transmit queue is enabled and descriptors exist, descriptors and data are fetched immediately. Actual transmit activity on port starts only if the *TCTL.EN* bit is set.
SWFLSH (WC)	26	0b	Transmit Software Flush This bit enables software to trigger descriptor write-back flushing, independently of other conditions. This bit is self cleared by hardware. Bit will clear after write-back flush is triggered (may take a number of cycles). **Note:** When working in head write-back mode (*TDWBAL.Head_WB_En* = 1) *TDWBAL.WB_on_EITR* bit should be set for transmit descriptor flush to occur.
Priority	27	0b	Transmit queue priority 0 - Low priority 1 - High priority When set, Transmit DMA resources are always allocated to the queue before low priority queues. Arbitration between transmit queues with same priority is done in a Round Robin fashion.
HWBTHRESH	31:28	0x0	Transmit Head writeback threshold If value of field is greater than 0x0, head writeback to host will occur only when the amount of internal pending write backs exceeds this threshold. Refer to Section 7.2.3 for additional information. **Note:** When activating this mode the *WB_on_EITR* bit in the *TDWBAL* register should be set to guarantee a write back after a timeout even if the threshold has not been reached.

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x3828, 0x3928, 0x3A28 and 0x3B28 respectively.

8.12.16 Tx Descriptor Completion Write – Back Address Low - TDWBAL (0xE038 + 0x40*n [n=0...7]; R/W)

Field[1]	Bit(s)	Initial Value	Description
Head_WB_En	0	0b	Head Write-Back Enable 1b = Head write-back is enabled. 0b = Head write-back is disabled. When *head_WB_en* is set, *TXDCTL.SWFLSH* is ignored and no descriptor write-back is executed.
WB_on_EITR	1	0b	When set, a head write back is done upon EITR expiration.
HeadWB_Low	31:2	0x0	Bits 31:2 of the head write-back memory location (DWORD aligned). Last 2 bits of this field are ignored and are always interpreted as 00b, meaning that the actual address is QWORD aligned. Bits 1:0 are always 00b.

1. Software should program TDWBAL[n] register only when queue is disabled (*TXDCTL[n].Enable* = 0).

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x3838, 0x3938, 0x3A38 & 0x3B38 respectively.

8.12.17 Tx Descriptor Completion Write–Back Address High - TDWBAH (0xE03C + 0x40*n [n=0...7];R/W)

Field[1]	Bit(s)	Initial Value	Description
HeadWB_High	31:0	0x0	Highest 32 bits of the head write-back memory location.

1. Software should program TDWBAH[n] register only when queue is disabled (*TXDCTL[n].Enable* = 0).

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x383C, 0x393C, 0x3A3C & 0x3B3C respectively.

8.12.18 Transmit Queue drop packet count - TQDPC (0xE030 + 0x40*n [n=0...7]; RW)

Field	Bit(s)	Initial Value	Description
TQDPC	31:0	0x0	Transmit Queue drop packet count - counts the number of packets dropped by a queue due to lack of space in the loopback buffer or due to security (anti - spoof) issues. A multicast packet dropped by some of the destinations, but sent to others is counted by this counter Note: Counter wraps around when reaching a value of 0xFFFFFFFF.

8.13 DCA and TPH Register Descriptions

8.13.1 Rx DCA Control Registers - RXCTL (0xC014 + 0x40*n [n=0...7]; R/W)

Note: RX data write no-snoop is activated when the NSE bit is set in the receive descriptor.

Field	Bit(s)	Initial Value	Description
RX Descriptor Fetch TPH EN	0	0b	Receive Descriptor Fetch TPH Enable When set, hardware enables TPH for all Rx descriptors fetch from memory. When cleared, hardware does not enable TPH for descriptor fetches. This bit is cleared as a default.
RX Descriptor Writeback TPH EN[1]	1	0b	Receive Descriptor Writeback TPH Enable When set, hardware enables TPH for all Rx descriptors written back into memory. When cleared, hardware does not enable TPH for descriptor write-backs. This bit is cleared as a default. The hint used is the hint set in the Socket ID field.
Rx Header TPH EN[1]	2	0b	Receive Header TPH Enable When set, hardware enables TPH for all received header buffers. When cleared, hardware does not enable TPH for Rx headers. This bit is cleared as a default. The hint used is the hint set in the Socket ID field.
Rx Payload TPH EN[1]	3	0b	Receive Payload TPH Enable When set, hardware enables TPH for all Ethernet payloads written into memory. When cleared, hardware does not enable TPH for Ethernet payloads. This bit is cleared as a default. The hint used is the hint set in the Socket ID field.
Reserved	4	0b	Reserved Write 0, ignore on read.
RX Descriptor DCA EN[1]	5	0b	Descriptor DCA Enable When set, hardware enables DCA for all Rx descriptors written back into memory. When cleared, hardware does not enable DCA for descriptor write-backs. This bit is cleared as a default.
Rx Header DCA EN[1]	6	0b	Receive Header DCA Enable When set, hardware enables DCA for all received header buffers. When cleared, hardware does not enable DCA for Rx headers. This bit is cleared as a default.
Rx Payload DCA EN[1]	7	0b	Receive Payload DCA Enable When set, hardware enables DCA for all Ethernet payloads written into memory. When cleared, hardware does not enable DCA for Ethernet payloads. This bit is cleared as a default.

Field	Bit(s)	Initial Value	Description
RXdescRead NSEn	8	0b	Receive Descriptor Read No Snoop Enable This bit must be reset to 0b to ensure correct functionality (Except if the software driver can guarantee the data is present in the main memory before the DMA process occurs). Note: When TPH is enabled No Snoop bit should be 0.
RXdescRead ROEn	9	1b	Receive Descriptor Read Relax Order Enable
RXdescWBNSen	10	0b	Receive Descriptor Write-Back No Snoop Enable This bit must be reset to 0b to ensure correct functionality of descriptor write-back. Note: When TPH is enabled No Snoop bit should be 0.
RXdescWBROen (RO)	11	0b	Receive Descriptor Write-Back Relax Order Enable This bit must be reset to 0b to ensure correct functionality of descriptor write-back.
RXdataWrite NSEn	12	0b	Receive Data Write No Snoop Enable (header replication: header and data) When set to 0b, the last bit of the *Packet Buffer Address* field in the advanced receive descriptor is used as the LSB of the packet buffer address (A0), thus enabling Byte alignment of the buffer. When set to 1b, the last bit of the *Packet Buffer Address* field in advanced receive descriptor is used as the No-Snoop Enabling (NSE) bit (buffer is Word aligned). If also set to 1b, the NSE bit determines whether the data buffer is snooped or not. Note: When TPH is enabled No Snoop bit should be 0.
RXdataWrite ROEn	13	1b	Receive Data Write Relax Order Enable (header replication: header and data)
RxRepHeader NSEn	14	0b	Receive Replicated/Split Header No Snoop Enable This bit must be reset to 0b to ensure correct functionality of header write to host memory. Note: When TPH is enabled No Snoop bit should be 0.
RxRepHeader ROEn	15	1b	Receive Replicated/Split Header Relax Order Enable
Reserved	23:16	0x0	Reserved Write 0, ignore on read.
CPUID	31:24	0x0	Physical ID Legacy DCA capable platforms - the device driver, upon discovery of the physical CPU ID and CPU Bus ID, programs the CPUID field with the Physical CPU and Bus ID associated with this Rx queue. DCA 1.0 capable platforms - the device driver programs a value, based on the relevant APIC ID, associated with this Tx queue. Refer to Table 3.1.3.1.2.3 for details. TPH capable platforms - the device driver programs a value, based on the relevant Socket ID, associated with this receive queue. Note that for TPH platforms, bits 31:27 of this field should always be set to zero. Refer to Section 7.7.2 for details.

1. Both DCA Enable bit and TPH Enable bit should not be set for the same type of traffic.

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x2814, 0x2914, 0x2A14 & 0x2B14 respectively.

8.13.2 Tx DCA Control Registers - TXCTL (0xE014 + 0x40*n [n=0...7]; R/W)

Field	Bit(s)	Initial Value	Description
Tx Descriptor Fetch TPH EN[1]	0	0b	Transmit Descriptor Fetch TPH Enable When set, hardware enables TPH for all Tx descriptors fetch from memory. When cleared, hardware does not enable TPH for descriptor fetches. This bit is cleared as a default.
Tx Descriptor Writeback TPH EN	1	0b	Transmit Descriptor Writeback TPH Enable When set, hardware enables TPH for all Tx descriptors written back into memory. When cleared, hardware does not enable TPH for descriptor write-backs. This bit is cleared as a default. The hint used is the hint set in the Socket ID field.
Reserved	2	0b	Reserved Write 0, ignore on read.
Tx Packet TPH EN	3	0b	Transmit Packet TPH Enable When set, hardware enables TPH for all Ethernet payloads read from memory. When cleared, hardware does not enable TPH for Ethernet payloads. This bit is cleared as a default.
Reserved	4	0b	Reserved Write 0, ignore on read.
TX Descriptor DCA EN[1]	5	0b	Descriptor DCA Enable When set, hardware enables DCA for all Tx descriptors written back into memory. When cleared, hardware does not enable DCA for descriptor write-backs. This bit is cleared as a default and also applies to head write-back when enabled.
Reserved	7:6	00b	Reserved Write 0, ignore on read.
TXdescRDNSen	8	0b	Tx Descriptor Read No Snoop Enable This bit must be reset to 0b to ensure correct functionality (unless the software device driver has written this bit with a write-through instruction). Note: When TPH is enabled No Snoop bit should be 0.
TXdescRDROEn	9	1b	Tx Descriptor Read Relax Order Enable
TXdescWBNSen	10	0b	Tx Descriptor Write-Back No Snoop Enable This bit must be reset to 0b to ensure correct functionality of descriptor write-back. Also applies to head write-back, when enabled. Note: When TPH is enabled No Snoop bit should be 0.
TXdescWBROen	11	1b	Tx Descriptor Write-Back Relax Order Enable Applies to head write-back, when enabled.
TXDataReadNSEn	12	0b	Tx Data Read No Snoop Enable Note: When TPH is enabled No Snoop bit should be 0.
TXDataReadROEn	13	1b	Tx Data Read Relax Order Enable
Reserved	23:14	0b	Reserved Write 0 ignore on read.
CPUID	31:24	0x0	Physical ID Legacy DCA capable platforms - the device driver, upon discovery of the physical CPU ID and CPU Bus ID, programs the CPUID field with the Physical CPU and Bus ID associated with this Tx queue. DCA 1.0 capable platforms - the device driver programs a value, based on the relevant APIC ID, associated with this Tx queue. Refer to Table 3.1.3.1.2.3 for details TPH capable platforms - the device driver programs a value, based on the relevant Socket ID, associated with this transmit queue. Note that for TPH platforms, bits 31:27 of this field should always be set to zero. Refer to Section 7.7.2 for details.

1. Both DCA Enable bit and TPH Enable bit should not be set for the same type of traffic.

Note: In order to keep compatibility with the 82575, for queues 0-3, these registers are aliased to addresses 0x3814, 0x3914, 0x3A14 & 0x3B14 respectively.

8.13.3 DCA Requester ID Information - DCA_ID (0x5B70; RO)

The DCA requester ID field, composed of Device ID, Bus #, and Function # is set up in MMIO space for software to program the DCA Requester ID Authentication register.

Field	Bit(s)	Initial Value	Description
Function Number	2:0	000b	Function Number Function number assigned to the function based on BIOS/operating system enumeration.
Device Number	7:3	0x0	Device Number Device number assigned to the function based on BIOS/operating system enumeration.
Bus Number	15:8	0x0	Bus Number Bus number assigned to the function based on BIOS/operating system enumeration.
Reserved	31:16	0x0	Reserved Write 0, ignore on read.

8.13.4 DCA Control - DCA_CTRL (0x5B74; R/W)

This CSR is common to all functions.

Field	Bit(s)	Initial Value	Description
DCA_DIS	0	1b	DCA Disable 0b = DCA tagging is enabled. 1b = DCA tagging is disabled.
DCA_MODE	4:1	0x0	DCA Mode 000b = Legacy DCA is supported. The TAG field in the TLP header is based on the following coding: bit 0 is DCA enable; bits 3:1 are CPU ID). 001b = DCA 1.0 is supported. When DCA is disabled for a given message, the TAG field is 0000,0000b. If DCA is enabled, the TAG is set per queue as programmed in the relevant DCA Control register. All other values are undefined.
Reserved	8:5	0x0	Reserved Write 0, ignore on read.
Desc_PH	10:9	00b	Descriptor PH - defines the PH field used when a TPH hint is given for descriptor associated traffic (descriptor fetch, descriptor write back or head write back).
Data_PH	12:11	10b	Data PH - defines the PH field used when a TPH hint is given for data associated traffic (Tx data read, Rx data write).
Reserved	31:13	0x0	Reserved Write 0, ignore on read.

8.14 Virtualization Register Descriptions

8.14.1 VMDq Control Register – VT_CTL (0x581C; R/W)

Field	Bit(s)	Initial Value	Description
Reserved	6:0	0x0	Reserved Write 0, ignore on read.
DEF_PL	9:7	000b	Default pool - used to queue packets that did not pass any VM queuing decision.
Reserved	26:10	0x0	Reserved Write 0, ignore on read.
FLP	27	0b	Filter local packets - filter incoming packets whose MAC source address matches one of the LAN port DA MAC addresses. If the SA of the received packet matches one of the DA in the RAH/RAL registers, then the VM tied to this DA does not receive the packet. Other VMs can still receive it.
IGMAC	28	0b	If set, MAC address is ignored during pool decision. Pooling is based on VLAN only. This bit can be set only if the RCTL.VFE bit is set.
Dis_Def_Pool	29	0b	Drop if no pool is found. If this bit is asserted, then in a RX switching virtualized environment, if there is no destination pool, the packet is discarded and not sent to the default pool. Otherwise, it is sent to the pool defined by the DEF_PL field.
Rpl_En	30	0b	Replication Enable
Reserved	31	0b	Reserved Write 0, ignore on read.

8.14.2 Physical Function Mailbox - PFMailbox (0x0C00 + 4*n[n=0...7]; RW)

Field	Bit(s)	Initial Value	Description
Sts (WO)	0	0b	Status/Command from PF ready. Setting this bit, causes an interrupt to the relevant VF. This bit always read as zero. Setting this bit sets the PFSTS bit in VFMailbox.
Ack (WO)	1	0b	VF message received. Setting this bit, causes an interrupt to the relevant VF. This bit always read as zero. Setting this bit sets the PFACK bit in VFMailbox.
VFU	2	0b	Buffer is taken by VF. This bit is RO for the PF and is a mirror of the VFU bit of the VFMailbox register.
PFU	3	0b	Buffer is taken by PF. This bit can be set only if the VFU bit is cleared and is mirrored in the PFU bit of the VFMailbox register.
RVFU (WO)	4	0b	Reset VFU - setting this bit clears the VFU bit in the corresponding VFMailbox register - this bit should be used only if the VF driver is stuck. Setting this bit is also reset the corresponding bits in the *MBVFICR VFREQ* and *VFACK* fields.
Reserved	31:5	0x0	Reserved Write 0, ignore on read.

The usage of the mailbox register set is described in Section 7.8.2.9.1.

8.14.3 Virtual Function Mailbox - VFMailbox (0x0C40 + 4*n [n=0...7]; RW)

Field	Bit(s)	Initial Value	Description
Req (WO)	0	0b	Request for PF ready. Setting this bit, causes an interrupt to the PF. This bit always read as zero. Setting this bit sets the corresponding bit in *VFREQ* field in *MBVFICR* register.
Ack (WO)	1	0b	PF message received. Setting this bit, causes an interrupt to the PF. This bit always read as zero. Setting this bit sets the corresponding bit in *VFACK* field in *MBVFICR* register.
VFU	2	0b	Buffer is taken by VF. This bit can be set only if the PFU bit is cleared and is mirrored in the VFU bit of the PFMailbox register.
PFU	3	0b	Buffer is taken by PF. This bit is RO for the VF and is a mirror of the PFU bit of the *PFMailbox* register.
PFSTS (RC)	4	0b	PF wrote a message in the mailbox.
PFACK (RC)	5	0b	PF acknowledged the VF previous message.
RSTI (RO)	6	1b	Indicates that the PF has reset the shared resources and the reset sequence is in progress. ***Notes:*** 1. Refer to Section 4.6.11.2.3 for additional information on *RSTI* usage. 2. When *CTRL_EXT.PFRSTD* is set, the *RSTI* bit in all the VFMailbox registers is cleared and the *RSTD* bit in all the VFMailbox register is set.
RSTD (RC)	7	0b	Indicates that a PF software reset completed and the VF can start to use the device. ***Note:*** When *CTRL_EXT.PFRSTD* is set, the *RSTI* bit in all the *VFMailbox* registers is cleared and the *RSTD* bit in all the VFMailbox register is set.
Reserved	31:8	0x0	Reserved Write 0, ignore on read.

8.14.4 Virtualization Mailbox Memory - VMBMEM (0x0800:0x083C + 0x40*n [n=0...7]; R/W)

Mailbox memory for PF and VF drivers communication. Locations can be accessed as 32-bit or 64-bit words.

The memory is accessible to the PF and the VFs according to the following mapping.

RAM address	Function	PF BAR 0 mapping[1]	VF BAR 0 mapping
0 - 63	VF0 ↔ PF	0 - 63	VMBMEM:VMBMEM + 63
64 - 127	VF1 ↔ PF	64 - 127	VMBMEM:VMBMEM + 63
....			
448 - 511	VF7↔ PF	448 - 511	VMBMEM:VMBMEM + 63

1. Relative to VMBMEM register.

Field	Bit(s)	Initial Value	Description
Mailbox Data	31:0	X	Mailbox Data

8.14.5 Mailbox VF Interrupt Causes Register - MBVFICR (0x0C80; R/W1C)

Field	Bit(s)	Initial Value	Description
VFREQ	7:0	0x0	VF #n wrote a message.
Reserved	15:8	0x0	Reserved Write 0, ignore on read.
VFACK	23:16	0x0	VF #n acknowledged a PF message.
Reserved	31:24	0x0	Reserved Write 0, ignore on read.

8.14.6 Mailbox VF Interrupt Mask Register - MBVFIMR (0x0C84; RW)

Field	Bit(s)	Initial Value	Description
VFIM	7:0	0xFF	Mailbox indication from VF #n can cause an interrupt to the PF.
Reserved	31:8	0x0	Reserved Write 0, ignore on read.

8.14.7 FLR Events - VFLRE (0x0C88; R/W1C)

This register reflects the VFLR events of the different VFs. It is accessible only to the PF. These bits are cleared by writing 1.

Field	Bit(s)	Initial Value	Description
VFLR	7:0	X	Reflects a VFLR event in VF7 to VF0 respectively.
Reserved	31:8	0x0	Reserved Write 0, ignore on read.

8.14.8 VF Receive Enable- VFRE (0x0C8C; RW)

Field	Bit(s)	Initial Value	Description
VFRE	7:0	0xFF	Enables filtering process to forward packets to VF7 to VF0 respectively. Each bit is cleared by the relevant VFLR or by a VF SW reset.
Reserved	31:8	0x0	Reserved Write 0, ignore on read.

8.14.9 VF Transmit Enable - VFTE (0x0C90; RW)

Field	Bit(s)	Initial Value	Description
VFTE	7:0	0xFF	Enables transmit process to forward packets from VF7 to VF0 respectively. Each bit is cleared by the relevant VFLR or by a VF SW reset.
Reserved	31:8	0x0	Reserved Write 0, ignore on read.

Note: Clearing one of *VFTE* bits may cause a transmit packet drop from the disabled queue.

8.14.10 Wrong VM Behavior Register - WVBR (0x3554; RC)

Field	Bit(s)	Initial Value	Description
WVM	7:0	0x0	Bitmap indicating against which queue an anti spoof action or malicious action was taken. Once the register is read, the indication is cleared, but the queue is still blocked. The queue will be released only by a reset of the VF.
Reserved	31:8	0x0	Reserved

8.14.11 Malicious Driver Free Block - MDFB (0x3558; RO)

Field	Bit(s)	Initial Value	Description
Block Queue	7:0	0x0	Indicates queue that was blocked due to malicious behavior. When bit is set, to commence activity on offending queue, Software should toggle the corresponding bit in VFTE to release the queue. After that, the queue should be re-initialized.
Reserved	31:8	0x0	Reserved Write 0, ignore on read.

8.14.12 VM Error Count Mask – VMECM (0x3510; RW)

Field	Bit(s)	Initial Value	Description
Filter	7:0	0x0	Defines if a packet dropped from pools 0 to 7 respectively is counted in the SSVPC counter.
Reserved	31:8	0x0	Reserved Write 0, ignore on read.

8.14.13 Last VM Misbehavior Cause – LVMMC (0x3548; RC)

Bits in LVMMC register define the cause for blocking the malicious queue that was reported in the *MDFB.Block Queue* field when *DTXCTL.MDP_EN* is set. Refer to Section 7.8.3.8.3 for details of the different bits.

Note: Only the first malicious event is registered for each packet, so if a bit is not set it doesn't mean that this event didn't occur, only that another malicious behavior was detected first.

Field	Bit(s)	Initial Value	Description
Mac Header	0	0b	Illegal MAC header size.
IPV4 Header	1	0b	Illegal IPV4 header size.
IPV6 Header	2	0b	Illegal IPV6 header size.
Wrong MAC_IP	3	0b	Wrong MAC +IP header size
TCP LSO	4	0b	Illegal TCP header was detected in a large send operation.
Reserved	5	0b	Reserved Write 0, ignore on read.
UDP LSO	6	0b	Illegal UDP header was detected in a large send operation.
SCTP SSO	7	0b	Illegal SCTP header was detected in a single send operation.
Leg_Size	8	0b	Illegal legacy descriptor size.
Adv_Size	9	0b	Illegal advanced descriptor size.
Off_Ill	10	0b	Illegal offload request.
SCTP_aligned	11	0b	CRC request of non 4 byte aligned data.
Reserved	15:5	0b	Reserved
SSO_UDP	17	0b	Wrong parameter of headers for UDP SSO
SSO_TCP	18	0b	Wrong parameter of headers for TCP SSO
MVF_MACC	19	0b	Malicious VF memory access A PCIe DMA access initiated by a VF ended with Unsupported Request (UR) or Completer Abort (CA). When a Malicious DMA access is detected queue (VF) that initiated the access is disabled (both RX and TX) and corresponding MDFB.Block Queue bit is set.
DESC_TYPE	20	0b	Wrong descriptor type (other than 2,3)
Wrong_null	21	0b	Null without EOP
No EOP	22	0b	Packet without EOP (i.e. bigger than the ring size)
ILL_DBU	24	0b	Illegal DBU configuration.
MAC VLAN spoof	25	0b	A MAC spoof or VLAN spoof attempt was detected
VLAN IERR	26	0b	VLAN Insertion Error. VLE bit set in TX descriptor when VMVIR[n].VLANA register field is not 0 causing drop of TX packets.
Legacy desc in IOV	27	0b	A legacy desc in IOV was detected
Mal_PF	28	0b	Malicious Driver behavior detected on current PF
Last_Q	31:29	0x0	Last queue that detected malicious behavior.

8.14.14 Queue Drop Enable Register - QDE (0x2408;RW)

This register allows the PF to override the *SRRCTL.drop_en* bit set by the VF, to avoid head of line blocking issues if an un-trusted VF does not provide a receive descriptor to the hardware.

.

Field	Bit(s)	Initial Value	Description
QDE	7:0	0x0	Enable drop packets from queue 7 to 0 respectively. This bit overrides the *SRRCTL.drop_en* bit of each queue. If either of the bits is set, a packet received when no is descriptor available is dropped.
Reserved	15:8	0x0	Reserved Write 0, ignore on read.
Reserved	31:16	0x0	Reserved Write 0, ignore on read.

8.14.15 TX Switch Control - TXSWC (0x5ACC; R/W)

This register controls the security settings of the switch and enables the loopback mode.

Field	Bit(s)	Initial Value	Description
MACAS	7:0	0x0	Enable anti spoofing filter on MAC addresses for VF7 to VF0 respectively.
VLANAS	15:8	0x0	Enable anti spoofing filter on VLAN tags for VF7 to VF0 respectively.
LLE	23:16	0x0	Local loopback enable - when set, a packet originating from pool N and destined to pool N is looped back. If clear, the packet is dropped.
Reserved	30:24	0x0	Reserved Write 0, ignore on read.
Loopback_en	31	0b	Enable VMDQ loopback.

8.14.16 VM VLAN Insert Register – VMVIR (0x3700 + 4 *n [n=0...7]; RW)

Field	Bit(s)	Initial Value	Description
Port VLAN ID	15:0	0x0	Port VLAN tag to insert when VLAN action is to always insert default VLAN (*VMVIR[n].vlana* = 01b).
Reserved	29:16	0x0	Reserved Write 0, ignore on read.
vlana	31:30	0x0	VLAN action: 00b = Use descriptor command (Refer to Section 7.2.2). 01b = Always insert Default VLAN 10b = Never insert VLAN 11b = Reserved.

8.14.17 VM Offload Register - VMOLR (0x5AD0 + 4*n [n=0...7]; RW)

This register controls part of the offload and queueing options applied to each pool (VM).

Field	Bit(s)	Initial Value	Description
rlpml	13:0	0x2600	Long packet size (9k default) If the packet is longer than a legal Ethernet packet, packet is removed from the pool list all the pools for which the *VMOLR.LPE* bit is not set or the packet length is larger than the value stated in the *VMOLR.RLPML* field (Refer to Section 7.8.3.4 for additional information). ***Note:*** Packet is longer than length of a legal Ethernet Packet if: 1. The packet is longer than 1518 bytes and there are no VLAN tags in the packet. 2. The packet is longer than 1522 bytes and there is one VLAN tag in the packet. 3. The packet is longer than 1526 bytes and there are two VLAN tags in the packet.
Reserved	15:14	0x0	Reserved Write 0, ignore on read.
lpe	16	0b	Long packet enable
Reserved	22:17	0x0	Reserved Write 0, ignore on read.
vpe	23	0b	VLAN Promiscuous Enable
aupe	24	0b	Accept Untagged packets enable. When set, packets without VLAN tag can be forwarded to this queue, assuming they pass the MAC address queueing mechanism.
rompe	25	0b	receive overflow multicast packets - accept packets that match the MTA table.
rope	26	0b	receive overflow packets - accept packets that match the UTA table.
bam	27	0b	Broadcast accept
mpe	28	0b	multicast promiscuous
upe	29	0b	Unicast Promiscuous
Reserved	31:30	0x0	Reserved Write 0, ignore on read.

8.14.18 DMA VM Offload Register - DVMOLR (0xC038 + 0x40*n[n=0...7]; RW)

This register controls part of the offload and queueing options applied to each pool (VM).

Field	Bit(s)	Initial Value	Description
Reserved	28:0	0x0	Reserved Write 0, ignore on read.
Hide VLAN	29	0b	If this bit is set, a value of zero is written in the *RDESC.VLAN tag* and in the *RDESC.STATUS.VP* fields of the received descriptor. If this bit is set for a VM, the *DVMOLR.STRVLAN* bit for this VM should be set also.
STRVLAN	30	0b	VLAN strip If this bit is set, the VLAN is removed from the packet, and may be inserted in the receive descriptor (depending on the value of the Hide VLAN field). ***Note:*** If this bit is set the *DVMOLR[n].CRC strip* bit should be set as the CRC is not valid anymore.

Field	Bit(s)	Initial Value	Description
CRC strip	31	1b	CRC strip If this bit is set, the CRC is removed from the packet. ***Notes:*** 1. If the *DVMOLR[n].STRVLAN* bit is set the *DVMOLR[n].CRC strip* bit should also be set as the CRC is not valid anymore. 2. Even when bit is set, CRC strip is not done on runt packets (smaller than 64 Bytes). 3. These bits should be cleared in non virtualized mode.

8.14.19 VLAN VM Filter - VLVF (0x5D00 + 4*n [n=0...31]; RW)

This register set describes which VLANs the local VMs are part of. Each of the 32 registers contains a VLAN tag and a list of the VMs (VFs) which are part of it. Only packets with a VLAN matching one of the VLAN tags of which the VM is member of are forwarded to this VM.

Field	Bit(s)	Initial Value	Description
VLAN_Id	11:0	0x0	Defines a VLAN tag, to which each VM whose bit is set in the POOLSEL field, belongs to.
POOLSEL	19:12	0x0	Pool Select (bitmap) Field defines to which VMs a packet with the VLAN_Id should be forwarded to. A bit is allocated to each of the 8 VMs, enabling forwarding the packet with the VLAN_Id to multiple VMs.
LVLAN	20	0x0	This VLAN is local and packets with this VLAN should not be forwarded to the external NIC.
Reserved	30:21	0x0	Reserved Write 0, ignore on read.
VI_En	31	0b	VLAN Id Enable - this filter is valid. Note: If *RCTL.VFE* is 0 all *VLVF* filters are disabled.

8.14.20 Unicast Table Array - UTA (0xA000 + 4*n [n=0...127]; R/W)

There is one register per 32 bits of the Unicast Address Table for a total of 128 registers (the UTA[127:0] designation). Software must mask to the desired bit on reads and supply a 32-bit word on writes. The first bit of the address used to access the table is set according to the *RCTL.MO* field.

Note: All accesses to this table must be 32 bit.
The lookup algorithm is the same one used for the MTA table.
This table should be zeroed by software before start of work.

Field	Bit(s)	Initial Value	Description
Bit Vector	31:0	X	Word wide bit vector specifying 32 bits in the unicast destination address filter table.

8.14.21 Storm Control - Control Register- SCCRL (0x5DB0; RW)

Field	Bit(s)	Initial Value	Description
MDIPW	0	0b	Drop multicast packets (excluding flow control and manageability packets) if multicast threshold is exceeded in previous window
MDICW	1	0b	Drop multicast packets (excluding flow control and manageability packets) if multicast threshold is exceeded in current window
BDIPW	2	0b	Drop broadcast packets (excluding flow control and manageability packets) if broadcast threshold is exceeded in previous window
BDICW	3	0b	Drop broadcast packets (excluding flow control and manageability packets) if broadcast threshold is exceeded in current window
BIDU	4	0b	BSC Includes Destination Unresolved packets: If bit is set, unicast received packets with no destination pool and sent to the default pool are included in IBSC
Reserved	7:5	0x0	Reserved Write 0, ignore on read.
INTERVAL	17:8	0x8	BSC/MSC Time-interval-specification: The interval size for applying Ingress Broadcast or Multicast Storm Control. Interrupt decisions are made at the end of each interval (and most flags are also set at interval end). Setting this field resets the counter. Refer to additional information in Section 7.8.3.8.4.
Reserved	31:18	0x0	Reserved Write 0, ignore on read.

8.14.22 Storm Control status - SCSTS (0x5DB4;RO)

Field	Bit(s)	Initial Value	Description
BSCA	0	0b	Broadcast storm control active
BSCAP	1	0b	Broadcast storm control active in previous window
MSCA	2	0b	Multicast storm control active
MSCAP	3	0b	Multicast storm control active in previous window
Reserved	31:4	0x0	Reserved Write 0, ignore on read.

8.14.23 Broadcast Storm Control Threshold - BSCTRH (0x5DB8; RW)

Field	Bit(s)	Initial Value	Description
UTRESH	18:0	0x0	Traffic Upper Threshold-size: Represents the upper threshold for broadcast storm control.
Reserved	31:19	0x0	Reserved Write 0, ignore on read.

8.14.24 Multicast Storm Control Threshold - MSCTRH (0x5DBC; RW)

Field	Bit(s)	Initial Value	Description
UTRESH	18:0	0x0	Traffic Upper Threshold-size: Represents the upper threshold for multicast storm control.
Reserved	31:19	0x0	Reserved Write 0, ignore on read.

8.14.25 Broadcast Storm Control Current Count - BSCCNT (0x5DC0; RO)

Field	Bit(s)	Initial Value	Description
CCOUNT	24:0	0x0	IBSC Traffic Current Count: Represents the count of broadcast traffic received in the current time interval in units of 64-byte segments.
Reserved	31:25	0x0	Reserved. Write 0, ignore on read.

8.14.26 Multicast Storm Control Current Count - MSCCNT (0x5DC4; RO)

Field	Bit(s)	Initial Value	Description
CCOUNT	24:0	0x0	IMSC Traffic Current Count: Represents the count of multicast traffic received in the current time interval in units of f 64-byte segments.
Reserved	31:25	0x0	Reserved. Write 0, ignore on read.

8.14.27 Storm Control Time Counter - SCTC (0x5DC8; RO)

This register keeps track of the number of time units elapsed since the end of last time interval.

Field	Bit(s)	Initial Value	Description
COUNT	9:0	0x0	SC Time Counter: The counter for number of time units elapsed since the end of the last time interval.
Reserved	31:10	0x0	Reserved. Write 0, ignore on read.

8.14.28 Storm Control Basic Interval- SCBI (0x5DCC; RW)

This register defines the basic interval used as the base for the *SCCRL.Interval* counting in 10 Mb/s speed.

Field	Bit(s)	Initial Value	Description
BI	24:0	0x5F5E10	Basic interval in 10 Mb/s port link rate in 16 ns clock cycles. ***Notes:*** 1. Initial value defines a basic interval of 100 mS at 10 Mbps link speed. 2. The interval in 1Gb/s and 100Mb/s port link rates is 100 and 10 times smaller respectively.
Reserved	31:25	0x0	Reserved. Write 0, ignore on read.

8.14.29 Virtual Mirror Rule Control - VMRCTL (0x5D80 + 0x4*n [n= 0...3]; RW)

This register controls the rules to be applied and the destination port.

Field	Bit(s)	Initial Value	Description
VPME	0	0b	Virtual pool mirroring enable- reflects all the packets sent to a set of given VMs.
UPME	1	0b	Uplink port mirroring enable - reflects all the traffic received from the network.
DPME	2	0b	Downlink port mirroring enable - reflects all the traffic transmitted to the network.
VLME	3	0b	VLAN mirroring enable - reflects all the traffic received in a set of given VLANs. bit enables mirroring operation based Either from the network or from local VMs.
Reserved	7:4	0x0	Reserved Write 0, ignore on read.
MP	10:8	0x0	VM Mirror port destination. Packets destined to certain VLAN groups, are mirrored to the queue defined by the MP field, according to the *VMRVLAN* register. Packets destined to certain VMs, are mirrored to the queue defined by the MP field, according to the *VMRVM* register. Note: If the *VMRCTL.UPME* bit is set to 1 all packets received on the port will be forwarded to the queue defined in the *MP* field.
Reserved	31:11	0x0	Reserved Write 0, ignore on read.

8.14.30 Virtual Mirror Rule VLAN - VMRVLAN (0x5D90 + 0x4*n [n= 0...3]; RW)

This register controls the VLAN tags as listed in the VLVF table taking part in the VLAN mirror rule.

Field	Bit(s)	Initial Value	Description
VLAN	31:0	0x0	Bitmap listing that defines which of the 32 VLANs defined in the *VLVF* registers participate in the mirror rule. Packets that have a matching Vlan_ID as defined by the VLVF registers will also be forwarded (mirrored) to the queue defined in the *VMRCTL.MP* field, if the *VMRCTL.VLME bit* is set to 1.

8.14.31 Virtual Mirror Rule VM - VMRVM (0x5DA0 + 0x4*n [n= 0...3]; RW)

This register controls the VM as listed in the RAH registers taking part in the VM mirror rule.

Field	Bit(s)	Initial Value	Description
VM	7:0	0x0	Bitmap listing of VMs participating in the mirror rule. Packets that are forwarded to the queues defined in the VMRVM.VM field, will also be forwarded (mirrored) to the queue defined in the *VMRCTL.MP* field, if the *VMRCTL.VPME bit* is set to 1.
Reserved	31:8	0x0	Reserved Write 0, ignore on read.

8.15 Timer Registers Description

8.15.1 Watchdog Setup - WDSTP (0x1040; R/W)

Field	Bit(s)	Initial Value	Description
WD_Enable	0	0b[1]	Enable Watchdog Timer
WD_Timer_ Load_enable (SC)	1	0b	Enables the load of the watchdog timer by writing to WD_Timer field. If this bit is not set, the WD_Timer field is loaded by the value of WD_Timeout. **Note:** Writing to this field is only for DFX purposes.
Reserved	15:2	0x0	Reserved Write 0, ignore on read.
WD_Timer (RWS)	23:16	WD_Timeout	Indicates the current value of the timer. Resets to the timeout value each time the I350 functional bit in Software Device Status register is set. If this timer expires, the WD interrupt to the firmware and the WD SDP is asserted. As a result, this timer is stuck at zero until it is re-armed. **Note:** Writing to this field is only for DFX purposes.
WD_Timeout	31:24	0x2[1]	Defines the number of seconds until the watchdog expires. The granularity of this timer is 1 sec. The minimal value allowed for this register when the watchdog mechanism is enabled is two. Setting this field to 1b might cause the watchdog to expire immediately. Note: Only 4 LSB bits loaded from EEPROM.

1. Value read from the EEPROM.

8.15.2 Watchdog Software Device Status - WDSWSTS (0x1044; R/W)

Field	Bit(s)	Initial Value	Description
Dev_Functional (SC)	0	0b	Each time this bit is set, the watchdog timer is re-armed. This bit is self clearing
Force_WD (SC)	1	0b	Setting this bit causes the WD timer to expire immediately. The WD_timer field is set to 0b. It can be used by software in order to indicate some fatal error detected in the software or in the hardware. This bit is self clearing.
Reserved	23:2	0x0	Reserved Write 0, ignore on read.
Stuck Reason	31:24	0x0	This field can be used by software to indicate to the firmware the reason the I350 is malfunctioning. The encoding of this field is software/firmware dependent. A value of 0 indicates a functional the I350.

8.15.3 Free Running Timer - FRTIMER (0x1048; RWS)

This register reflects the value of a free running timer that can be used for various timeout indications. The register is reset by a PCI reset and/or software reset.

Note: Writing to this register is for DFX purposes only.

Field	Bit(s)	Initial Value	Description
Microsecond	9:0	X	Number of microseconds in the current millisecond.
Millisecond	19:10	X	Number of milliseconds in the current second.
Seconds	31:20	X	Number of seconds from the timer start (up to 4095 seconds).

8.15.4 TCP Timer - TCPTIMER (0x104C; R/W)

Field	Bit(s)	Initial Value	Description
Duration	7:0	0x0	Duration Duration of the TCP interrupt interval in msec.
KickStart (WS)	8	0b	Counter Kick-Start Writing a 1b to this bit kick-starts the counter down-count from the initial value defined in the *Duration* field. Writing a 0b has no effect.

Field	Bit(s)	Initial Value	Description
TCPCountEn	9	0b	TCP Count Enable 1b = TCP timer counting enabled. 0b = TCP timer counting disabled. Once enabled, the TCP counter counts from its internal state. If the internal state is equal to 0b, the down-count does not restart until KickStart is activated. If the internal state is not 0b, the down-count continues from internal state. This enables a pause in the counting for debug purpose.
TCPCountFinish (WS)	10	0b	TCP Count Finish This bit enables software to trigger a TCP timer interrupt, regardless of the internal state. Writing a 1b to this bit triggers an interrupt and resets the internal counter to its initial value. Down-count does not restart until either KickStart is activated or Loop is set. Writing a 0b has no effect.
Loop	11	0b	TCP Loop When set to 1b, the TCP counter reloads duration each time it reaches zero, and continues down-counting from this point without kick-starting. When set to 0b, the TCP counter stops at a zero value and does not re-start until KickStart is activated. **Note:** Setting this bit alone is not enough to start the timer activity. The KickStart bit should also be set.
Reserved	31:12	0x0	Reserved. Write 0, ignore on read.

8.16 Time Sync Register Descriptions

8.16.1 RX Time Sync Control Register - TSYNCRXCTL (0xB620;RW)

Field	Bit(s)	Initial Value	Description
RXTT(ROS)	0	0x0	Rx timestamp valid Bit is set when a valid value for Rx timestamp is captured in the Rx timestamp registers. Bit is cleared by read of Rx timestamp high register (*RXSTMPH*)).
Type	3:1	0x0	Type of packets to timestamp - 000b – time stamp L2 (V2) packets only (Sync or Delay_req depends on message type in Section 8.16.26 and packets with Pdelay_Req and Pdelay_Resp message ID values) 001b – time stamp L4 (V1) packets only (Sync or Delay_req depends on message type in Section 8.16.26) 010b – time stamp V2 (L2 and L4) packets (Sync or Delay_req depends on message type in Section 8.16.26 and packets with Pdelay_Req and Pdelay_Resp message ID values) 100b – time stamp all packets. In this mode no locking is done to the timestamp value in the *RXSTMPL/H* timestamp registers, the *RDESC.STATUS.TS* bit in the receive descriptor stays 0, while the *RDESC.STATUS.TSIP* bit in the receive descriptor is always 1 if placing timestamp in receive buffer is enabled (refer to Section 7.1.6). 101b - Time stamp all packets which have a Message Type bit 3 zero, which means timestamp all event packets. This is applicable for V2 packets only. 011b, 110b and 111b – reserved Note: Field is also used for defining packets that have timestamp captured in receive buffer (refer to Section 7.1.6). ***Notes:*** 1. When time stamping a L2 packet then the Ethernet Type should match at least one EtherType filter with the *ETQF[n].1588 time stamp* bit set to 1b. 2. When time stamping a L4 packet then packet should match at least one 2-tuples filter (defined by the *TTQF[n]*, *IMIREXT[n]* and *IMIR[n]* registers) with the *TTQF[n].1588 time stamp* bit set to 1b.
En	4	0b	Enable RX timestamp 0 = time stamping disabled. 1 = time stamping enabled.
RSV	31:6	0x0	Reserved Write 0, ignore on read.

8.16.2 RX Timestamp Low - RXSTMPL (0xB624; RO)

Field	Bit(s)	Initial Value	Description
RTSL	31:0	0x0	Rx timestamp LSB value Value in 1 nS resolution.

8.16.3 RX Timestamp High - RXSTMPH (0xB628; RO)

Field	Bit(s)	Initial Value	Description
RTSH	7:0	0x0	Rx timestamp MSB value Value in 2^{32} nS resolution.
Reserved	31:8	0x0	Reserved. Write 0, ignore on read.

8.16.4 RX Timestamp Attributes Low - RXSATRL(0xB62C; RO)

Field	Bit(s)	Initial Value	Description
SourceIDL	31:0	0x0	Sourceuuid low

8.16.5 RX Timestamp Attributes High- RXSATRH (0xB630; RO)

Field	Bit(s)	Initial Value	Description
SourceIDH	15:0	0x0	Sourceuuid high
SequenceID	31:16	0x0	SequenceId

8.16.6 TX Time Sync Control Register - TSYNCTXCTL (0xB614; RW)

Field	Bit(s)	Initial Value	Description
TXTT(ROS)	0	0b	Transmit timestamp valid (equals 1b when a valid value for Tx timestamp is captured in the Tx timestamp register, clear by read of Tx timestamp register TXSTMPH)
RSV	3:1	0x0	Reserved Write 0, ignore on read.
EN	4	0b	Enable Transmit timestamp 0b = time stamping disabled. 1b = time stamping enabled.
RSV	31:5	0x0	Reserved Write 0, ignore on read.

8.16.7 TX Timestamp Value Low - TXSTMPL (0xB618; RO)

Field	Bit(s)	Initial Value	Description
TTSL	31:0	0x0	Transmit timestamp LSB value Value in 1 nS resolution.

8.16.8 TX Timestamp Value High - TXSTMPH(0xB61C; RO)

Field	Bit(s)	Initial Value	Description
TTSH	7:0	0x0	Transmit timestamp MSB value Value in 2^{32} nS resolution.
Reserved	31:8	0x0	Reserved Write 0 ignore on read.

8.16.9 System Time Register Residue - SYSTIMR (0xB6F8; RW)

Field	Bit(s)	Initial Value	Description
STR	31:0	0x0	System time Residue value. Value in 2^{-32} nS resolution.

8.16.10 System Time Register Low - SYSTIML (0xB600; RW)

Field	Bit(s)	Initial Value	Description
STL	31:0	0x0	System time LSB value Value in 1 nS resolution.

8.16.11 System Time Register High - SYSTIMH (0xB604; RW)

Field	Bit(s)	Initial Value	Description
STH	7:0	0x0	System time MSB value Value in 2^{32} nS resolution.
Reserved	31:8	0x0	Reserved Write 0 ignore on read.

8.16.12 Increment Attributes Register - TIMINCA (0xB608; RW)

Field	Bit(s)	Initial Value	Description
Incvalue	30:0	0x0	Increment value. Value to be added or subtracted (depending on ISGN value) from 8 nS clock cycle in resolution of 2^{-32} nS.
ISGN	31	0b	Increment sign. 0 - Each 8 nS cycle add to SYSTIM a value of 8 nS + Incvalue * 2^{-32} nS. 1 - Each 8 nS cycle add to SYSTIM a value of 8 nS -Incvalue * 2^{-32} nS.

8.16.13 Time Adjustment Offset Register Low - TIMADJL (0xB60C; RW)

Field	Bit(s)	Initial Value	Description
TADJL	31:0	0x0	Time adjustment value – Low Value in 1 nS resolution.

8.16.14 Time Adjustment Offset Register High - TIMADJH (0xB610;RW)

Field	Bit(s)	Initial Value	Description
TADJH	7:0	0x0	Time adjustment value - High Value in 2^{32} resolution.
Reserved	30:8	0x0	Reserved. Write 0 ignore on read.
Sign	31	0b	Sign (0b=”+”, 1b =”-“)

8.16.15 TimeSync Auxiliary Control Register - TSAUXC (0xB640; RW)

Field	Bit(s)	Initial Value	Description
EN_TT0	0	0b	Enable target time 0. Enable bit is set by software to 1b, to enable pulse or level change generation as a function of the *TSAUXC.PLSG0* and *TSAUXC.PLSNeg0* bits. The bit is cleared by hardware when the target time is hit and Pulse or level change occurs.
EN_TT1	1	0b	Enable target time 1. Enable bit is set by software to 1b, to enable pulse or level change generation as a function of the *TSAUXC.PLSG1* and *TSAUXC.PLSNeg1* bits. The bit is cleared by hardware when the target time is hit and Pulse or level change occurs.
EN_CLK0	2	0b	Enable Configurable Frequency Clock 0 Clock is generated according to frequency defined in the *FREQOUT0* register on the SDP pin (0 to 3) that has both: 1. *TSSDP.TS_SDPx_SEL* field with a value of 10b. 2. *TSSDP.TS_SDPx_EN* value of 1b.
Reserved	3	0b	Reserved Write 0, ignore on read.
ST0	4	0b	Start Clock 0 Toggle on Target Time 0 Enable Clock 0 toggle only after target time 0, that's defined in the *TRGTTIML0* and *TRGTTIMH0* registers, has passed. The clock output is initially 0 and toggles with a frequency defined in the *FREQOUT0* register.
EN_CLK1	5	0b	Enable Configurable Frequency Clock 1 Clock is generated according to frequency defined in the *FREQOUT1* register on the SDP pin (0 to 3) that has both: 1. *TSSDP.TS_SDPx_SEL* field with a value of 11b. 2. *TSSDP.TS_SDPx_EN* value of 1b.
Reserved	6	0b	Reserved Write 0, ignore on read.
ST1	7	0b	Start Clock 1 Toggle on Target Time 1 Enable Clock 1 toggle only after Target Time 1, that's defined in the *TRGTTIML1* and *TRGTTIMH1* registers, has passed. The clock output is initially 1 and toggles with a frequency defined in the *FREQOUT1* register
EN_TS0	8	0b	Enable hardware time stamp 0 Enable Time stamping occurrence of change in SDP pin into the *AUXSTMPL0* and *AUXSTMPH0* registers. SDP pin (0 to 3) is selected for time stamping, if the SDP pin is selected via the *TSSDP.AUX0_SDP_SEL* field and the *TSSDP.AUX0_TS_SDP_EN* bit is set to 1b.
AUTT0	9	0b	Auxiliary timestamp taken - cleared when read from auxiliary timestamp 0 occurred
EN_TS1	10	0b	Enable hardware time stamp 1 Enable Time stamping occurrence of change in SDP pin into the *AUXSTMPL*1 and *AUXSTMPH1* registers. SDP pin (0 to 3) is selected for time stamping, if the SDP pin is selected via the *TSSDP.AUX1_SDP_SEL* field and the *TSSDP.AUX1_TS_SDP_EN* bit is set to 1b.
AUTT1	11	0b	Auxiliary timestamp taken - cleared when read from auxiliary timestamp 1 occurred
Reserved	16:12	0x0	Reserved Write 0, ignore on read.
PLSG0	17	0b	Use Target Time 0 to generate start of pulse and Target Time 1 to generate end of pulse. SDP pin selected to drive pulse or level change is set according to the *TSSDP.TS_SDPx_SEL* field with a value of 00b and *TSSDP.TS_SDPx_EN* bit with a value of 1b. 0 – Target Time 0 generates change in SDP level. 1 – Target time 0 generates start of pulse on SDP pin. Note: Pulse or level change is generated when *TSAUXC.EN_TT0* is set to 1.

Field	Bit(s)	Initial Value	Description
PLSNeg0	18	0b	Generate Negative pulse on Target Time 0 when PLSG0 is 1. 0 – Generate positive pulse on Target Time 0 when PLSG0 is 1. 1 – Generate Negative pulse on target time 0 when PLSG0 is 1. Note: If PLSNeg0 = 1, at start the selected SDP pin is set to 1.
PLSG1	19	0b	Use Target Time 1 to generate start of pulse and Target Time 0 to generate end of pulse. SDP pin selected to drive pulse or level change is set according to the *TSSDP.TS_SDPx_SEL* field with a value of 01b and *TSSDP.TS_SDPx_EN* bit with a value of 1b. 0 – Target Time 1 generates change in SDP level. 1 – Target time 1 generates start of pulse on SDP pin. Note: Pulse or level change is generated when *TSAUXC.EN_TT1* is set to 1.
PLSNeg1	20	0b	Generate Negative pulse on Target Time 1 when *PLSG1* is 1. 0 – Generate positive pulse on Target Time 1when *PLSG1* is 1. 1 – Generate Negative pulse on target time 1 when *PLSG1* is 1. Note: If *PLSNeg1* = 1, at start the selected SDP pin is set to 1.
Reserved	30:21	0x0	Reserved Write 0, ignore on read
Disable systime	31	1b	Disable SYSTIM count operation 0b - SYSTIM timer activated 1b - *SYSTIM* timer disabled. Value of *SYSTIMH*, *SYSTIML* and *SYSTIMR* remains constant.

8.16.16 Target Time Register 0 Low - TRGTTIML0 (0xB644; RW)

Field	Bit(s)	Initial Value	Description
TTL	31:0	0x0	Target Time 0 LSB register Value in 1 nS resolution.

8.16.17 Target Time Register 0 High - TRGTTIMH0 (0xB648; RW)

Field	Bit(s)	Initial Value	Description
TTH	7:0	0x0	Target Time 0 MSB register Value in 2^{32} nS resolution.
Reserved	31:8	0x0	Reserved Write 0 ignore on read.

8.16.18 Target Time Register 1 Low - TRGTTIML1 (0xB64C; RW)

Field	Bit(s)	Initial Value	Description
TTL	31:0	0x0	Target Time 1 LSB register Value in 1 nS resolution.

8.16.19 Target Time Register 1 High - TRGTTIMH1 (0xB650; RW)

Field	Bit(s)	Initial Value	Description
TTH	7:0	0x0	Target Time 1 MSB register Value in 2^{32} nS resolution.
Reserved	31:8	0x0	Reserved Write 0 ignore on read.

8.16.20 Frequency Out 0 Control Register FREQOUT0 (0xB654; RW)

Field	Bit(s)	Initial Value	Description
CHCT	7:0	0x0	Clock Out Half Cycle Time Half Cycle time of Clock 0 in 8 nS resolution. Clock is generated on SDP pin when TS*AUXC.EN_CLK0* is set to 1. SDP pin (0 to 3) that drives Clock 0 is selected according to the *TSSDP.TS_SDPx_SEL* field that has a value of 10b and a *TSSDP.TS_SDPx_EN* value of 1b. If TS*AUXC.*ST0 is set to 1, start of clock toggle is defined by Target Time 0 (*TRGTTIML0* and *TRGTTIMH0*) registers. Notes: 1. Setting this register to zero while using the frequency out feature, is illegal. 2. Clock 0 generation should not be enabled so long as absolute value of SYSTIM error correction using the *TRGTTIMH0* and *TRGTTIML0* registers is greater then 2 msec.
Reserved	31:8	0x0	Reserved Write 0 ignore on read.

8.16.21 Frequency Out 1 Control Register - FREQOUT1 (0xB658; RW)

Field	Bit(s)	Initial Value	Description
CHCT	7:0	0x0	Clock Out Half Cycle Time Half Cycle time of Clock 1 in 8 nS resolution. Clock is generated on SDP pin when TS*AUXC.EN_CLK1* is set to 1. SDP pin (0 to 3) that drives Clock 1 is selected according to the *TSSDP.TS_SDPx_SEL* field that has a value of 11b and a *TSSDP.TS_SDPx_EN* value of 1b. If TS*AUXC.*ST1 is set to 1, start of clock toggle is defined by Target Time 1 (*TRGTTIML1* and *TRGTTIMH1*) registers. Notes: 1. Setting this register to zero while using the frequency out feature, is illegal. 2. Clock 1 generation should not be enabled so long as absolute value of SYSTIM error correction using the *TRGTTIMH1* and *TRGTTIML1* registers is greater then 2 msec.
Reserved	31:8	0x0	Reserved Write 0 ignore on read.

8.16.22 Auxiliary Time Stamp 0 Register Low - AUXSTMPL0 (0xB65C; RO)

Field	Bit(s)	Initial Value	Description
TSTL	31:0	0x0	Auxiliary Time Stamp 0 LSB value Value in 1 nS resolution.

8.16.23 Auxiliary Time Stamp 0 Register High - AUXSTMPH0 (0xB660; RO)

Reading this register will release the value stored in AUXSTMPH/L0 and will allow time stamping of the next value.

Field	Bit(s)	Initial Value	Description
TSTH	7:0	0x0	Auxiliary Time Stamp 0 MSB value Value in 2^{32} nS resolution.
Reserved	31:8	0x0	Reserved Write 0 ignore on read.

8.16.24 Auxiliary Time Stamp 1 Register Low AUXSTMPL1 (0xB664; RO)

Field	Bit(s)	Initial Value	Description
TSTL	31:0	0x0	Auxiliary Time Stamp 1 LSB value Value in 1 nS resolution.

8.16.25 Auxiliary Time Stamp 1 Register High - AUXSTMPH1 (0xB668; RO)

Reading this register will release the value stored in AUXSTMPH/L1 and will allow stamping of the next value.

Field	Bit(s)	Initial Value	Description
TSTH	7:0	0x0	Auxiliary Time Stamp 1 MSB value Value in 2^{32} nS resolution.
Reserved	31:8	0x0	Reserved Write 0 ignore on read.

8.16.26 Time Sync RX Configuration - TSYNCRXCFG (0x5F50; R/W)

Field	Bit(s)	Initial Value	Description
CTRLT	7:0	0x0	V1 control to timestamp
MSGT	15:8	0x0	V2 Message Type to timestamp
Reserved	31:16	0x0	Reserved Write 0, ignore on read.

8.16.27 Time Sync SDP Configuration Register - TSSDP (0x003C; R/W)

This register defines the assignment of SDP pins to the Time sync auxiliary capabilities.

Field	Bit(s)	Initial Value	Description
AUX0_SDP_SEL	1:0	00b	Select one of the SDPs to serve as the trigger for auxiliary time stamp 0 (AUXSTMPL0 and AUXSTMPH0 registers) 00b = SDP0 is assigned 01b = SDP1 is assigned 10b = SDP2 is assigned 11b = SDP3 is assigned
AUX0_TS_SDP_EN	2	0b	When set indicates that one of the SDPs can be used as an external trigger to Aux timestamp 0 (note that if this bit is set to one of the SDP pins, the corresponding pin should be configured to input mode using SPD_DIR)
AUX1_SDP_SEL	4:3	00b	Select one of the SDPs to serve as the trigger for auxiliary time stamp 1 (in AUXSTMPL1 and AUXSTMPH1 registers) 00b = SDP0 is assigned 01b = SDP1 is assigned 10b = SDP2 is assigned 11b = SDP3 is assigned
AUX1_TS_SDP_EN	5	0b	When set indicates that one of the SDPs can be used as an external trigger to Aux timestamp 1 (note that if this bit is set to one of the SDP pins, the corresponding pin should be configured to input mode using SPD_DIR)
TS_SDP0_SEL	7:6	00b	SDP0 allocation to Tsync event – when TS_SDP0_EN is set, these bits select the Tsync event that is routed to SDP0. 00b = Target Time 0 is output on SDP0 01b = Target Time 1 is output on SDP0 10b = Freq Clock 0 is output on SDP0 11b = Freq Clock 1 is output on SDP0
TS_SDP0_EN	8	0b	When set indicates that SDP0 is assigned to Tsync.
TS_SDP1_SEL	10:9	00b	SDP1 allocation to Tsync event – when TS_SDP1_EN is set, these bits select the Tsync event that is routed to SDP1. 00b = Target Time 0 is output on SDP1 01b = Target Time 1 is output on SDP1 10b = Freq Clock 0 is output on SDP1 11b = Freq Clock 1 is output on SDP1
TS_SDP1_EN	11	0b	When set indicates that SDP1 is assigned to Tsync.
TS_SDP2_SEL	13:12	00b	SDP2 allocation to Tsync event – when TS_SDP2_EN is set, these bits select the Tsync event that is routed to SDP2. 00b = Target Time 0 is output on SDP2 01b = Target Time 1 is output on SDP2 10b = Freq Clock 0 is output on SDP2 11b = Freq Clock 1 is output on SDP2
TS_SDP2_EN	14	0b	When set indicates that SDP2 is assigned to Tsync.
TS_SDP3_SEL	16:15	00b	SDP3 allocation to Tsync event – when TS_SDP3_EN is set, these bits select the Tsync event that is routed to SDP3. 00b = Target Time 0 is output on SDP3 01b = Target Time 1 is output on SDP3 10b = Freq Clock 0 is output on SDP3 11b = Freq Clock 1 is output on SDP3
TS_SDP3_EN	17	0b	When set indicates that SDP3 is assigned to Tsync.
Reserved	31:18	0x0	Reserved Write 0, ignore on read.

8.16.28 Time Sync Interrupt Registers

8.16.28.1 Time Sync Interrupt Cause Register - TSICR (0xB66C; RC/W1C)

Note: Once ICR.Time_Sync is set, TSICR should be read to determine the actual interrupt cause and to enable reception of an additional ICR.Time_Sync interrupt.

Field	Bit(s)	Initial Value	Description
SYS WARP	0	0b	SYSTIM Warp around. Set when SYSTIM Warp Around occurs. Warp around occurrence can be used by Software to update software time sync time.
TXTS	1	0b	Transmit Time Stamp Set when new timestamp is loaded into *TXSTMP* register
RXTS	2	0b	Receive Time Stamp Set when new timestamp is loaded into *RXSTMP* register
TT0	3	0b	Target time 0 trigger. Set when Target Time 0 (*TRGTTIML/H0*) trigger occurs.
TT1	4	0b	Target time 1 trigger. Set when Target Time 1 (*TRGTTIML/H1*) trigger occurs.
AUTT0	5	0b	Auxiliary timestamp 0 taken. Set when new timestamp is loaded into AUXSTMP 0 (auxiliary timestamp 0) register.
AUTT1	6	0b	Auxiliary timestamp 1 taken. Set when new timestamp is loaded into AUXSTMP 1 (auxiliary timestamp 1) register.
TADJ	7	0b	Time Adjust 0 done Set when Time Adjust to clock out 0 or 1 completed
Reserved	31:8	0x0	Reserved. Write 0, ignore on read.

8.16.28.2 Time Sync Interrupt Mask Register - TSIM (0xB674; RW)

Field	Bit(s)	Initial Value	Description
SYS WARP	0	0b	SYSTIM Warp around Mask 0 – No Interrupt generated when *TSICR.SWARP* is set. 1 – Interrupt generated when *TSICR.SWARP* is set.
TXTS	1	0b	Transmit Time Stamp Mask 0 – No Interrupt generated when *TSICR.TXTS* is set. 1 – Interrupt generated when *TSICR.TXTS* is set.
RXTS	2	0b	Receive Time Stamp Mask 0 – No Interrupt generated when *TSICR.RXTS* is set. 1 – Interrupt generated when *TSICR.RXTS* is set.
TT0	3	0b	Target time 0 Trigger Mask 0 – No Interrupt generated when *TSICR.TT0* is set. 1 – Interrupt generated when *TSICR.TT0* is set.

Field	Bit(s)	Initial Value	Description
TT1	4	0b	Target time 1 Trigger Mask 0 – No Interrupt generated when *TSICR.TT1* is set. 1 – Interrupt generated when *TSICR.TT1* is set.
AUTT0	5	0b	Auxiliary timestamp 0 taken Mask 0 – No Interrupt generated when *TSICR.AUTT0* is set. 1 – Interrupt generated when *TSICR.AUTT0* is set.
AUTT1	6	0b	Auxiliary timestamp 1 taken Mask 0 – No Interrupt generated when *TSICR.AUTT1* is set. 1 – Interrupt generated when *TSICR.AUTT1* is set.
TADJ	7	0b	Time Adjust 0 done mask 0 – No Interrupt generated when *TSICR.TADJ* is set. 1 – Interrupt generated when *TSICR.TADJ* is set.
Reserved	31:8	0x0	Reserved. Write 0, ignore on read.

8.16.28.3 Time Sync Interrupt Set Register - TSIS (0xB670; WO)

TSIS register is Write Only. Writing 1 to a bit sets respective interrupt bit in *TSICR* register. Write operation causes a single interrupt event and bit is cleared internally.

Field	Bit(s)	Initial Value	Description
SYS WARP (SC)	0	0b	Set SYSTIM Warp Around Interrupt 0 – No *TSICR.SWARP* Interrupt set. 1 – *TSICR.SWARP* interrupt set.
TXTS (SC)	1	0b	Set Transmit Time Stamp Interrupt 0 – No *TSICR.TXTS* Interrupt set. 1 – *TSICR.TXTS* interrupt set.
RXTS (SC)	2	0b	Set Receive Time Stamp Interrupt 0 – No *TSICR.RXTS* Interrupt set. 1 – *TSICR.RXTS* Interrupt set.
TT0 (SC)	3	0b	Set Target Time 0 Trigger Interrupt 0 – No *TSICR.TT0* Interrupt set. 1 – *TSICR.TT0* Interrupt set.
TT1 (SC)	4	0b	Set Target Time 1 Trigger Interrupt 0 – No *TSICR.TT1* Interrupt set. 1 – *TSICR.TT1* Interrupt set.
AUTT0 (SC)	5	0b	Set Auxiliary Timestamp 0 Taken Interrupt 0 – No *TSICR.AUTT0* Interrupt set. 1 – *TSICR.AUTT0* Interrupt set.
AUTT1 (SC)	6	0b	Set Auxiliary Timestamp 1 Taken Interrupt 0 – No *TSICR.AUTT1* Interrupt set. 1 – *TSICR.AUTT1* Interrupt set.
TADJ (SC)	7	0b	Set Time Adjust 0 done Interrupt 0 – No *TSICR.TADJ* interrupt set. 1 – *TSICR.TADJ* interrupt set.
Reserved	31:8	0x0	Reserved. Write 0, ignore on read.

8.17 PCS Register Descriptions

These registers are used to configure the SerDes, SGMII and 1000BASE-KX PCS logic. Usage of these registers is described in Section 3.7.4.1 & Section 3.7.4.3.

8.17.1 PCS Configuration - PCS_CFG (0x4200; R/W)

Field	Bit(s)	Initial Value	Description
Reserved	2:0	000b	Reserved Write 0, ignore on read.
PCS Enable	3	1b	PCS Enable Enables the PCS logic of the MAC. Should be set in SGMII, 1000BASE-KX and SerDes mode for normal operation. Clearing this bit disables RX/TX of both data and control codes. Use this to force link down at the far end.
Reserved	29:4	0x0	Reserved Write 0, ignore on read.
PCS Isolate	30	0b	PCS Isolate Setting this bit isolates the PCS logic from the MAC's data path. PCS control codes are still sent and received.
SRESET	31	0b	Soft Reset Setting this bit puts all modules within the MAC in reset except the Host Interface. The Host Interface is reset via HRST. This bit is NOT self clearing; GMAC is in a reset state until this bit is cleared.

8.17.2 PCS Link Control - PCS_LCTL (0x4208; RW)

Field	Bit(s)	Initial Value	Description
FLV	0	0b	Forced Link Value This bit denotes the link condition when force link is set. 0b = Forced link down. 1b = Forced link up.
FSV	2:1	10b	Forced Speed Value These bits denote the speed when force speed and duplex (*PCS_LCTL.FSD*) bit is set. This value is also used when AN is disabled or when in SerDes mode. 00b = 10 Mb/s (SGMII). 01b = 100 Mb/s (SGMII). 10b = 1000 Mb/s (SerDes/SGMII/1000BASE-KX). 11b = Reserved.
FDV	3	1b	Forced Duplex Value This bit denotes the duplex mode when force speed and duplex (*PCS_LCTL.FSD*) bit is set. This value is also used when AN is disabled or when in SerDes mode. 1b = Full duplex (SerDes/SGMII/1000BASE-KX). 0b = Half duplex (SGMII).
FSD	4	0b	Force Speed and Duplex If this bit is set, then speed and duplex mode is forced to forced speed value and forced duplex value, respectively. Otherwise, speed and duplex mode are decided by internal AN/SYNC state machines.

Field	Bit(s)	Initial Value	Description
Force Link	5	0b	Force Link If this bit is set, then the internal LINK_OK variable is forced to forced link value (bit 0 of this register). Otherwise, LINK_OK is decided by internal AN/SYNC state machines.
LINK LATCH LOW (LL)	6	0b	Link Latch Low Enable If this bit is set, then link OK going LOW (negative edge) is latched until a processor read. Afterwards, link OK is continuously updated until link OK again goes LOW (negative edge is seen).
Force Flow Control	7	0b	0 = Flow control mode is set according to the AN process by following Table 37-4 in the IEEE 802.3 spec. 1 = Flow control is set according to FC_TX_EN / FC_RX_EN bits in CTRL register.
Reserved	15:8	0x0	Reserved Write 0, ignore on read.
AN_ENABLE	16	0b[1]	AN Enable Setting this bit enables the AN process in SerDes operating mode. Note: When link-up is forced (*CTRL.SLU*=1) the AN_ENABLE bit should be 0.
AN RESTART (SC)	17	0b	AN Restart Used to reset/restart the link auto-negotiation process when using SerDes mode. Setting this bit restarts the Auto-negotiation process. This bit is self clearing.
AN TIMEOUT EN	18	1b[1]	AN Timeout Enable This bit enables the AN Timeout feature. During AN, if the link partner does not respond with AN pages, but continues to send good IDLE symbols, then LINK UP is assumed. (This enables LINK UP condition when link partner is not AN-capable and does not affect otherwise). This bit should not be set in SGMII mode.
AN SGMII BYPASS	19	0b	AN SGMII Bypass If this bit is set, then IDLE detect state is bypassed during AN in SGMII mode. This reduces the acknowledge time in SGMII mode.
AN SGMII TRIGGER	20	1b	AN SGMII Trigger If this bit is cleared, then AN is not automatically triggered in SGMII mode even if SYNC fails. AN is triggered only in response to PHY messages or by a manual setting like changing the AN Enable/Restart bits.
Reserved	23:21	000b	Reserved Write 0, ignore on read.
FAST LINK TIMER	24	0b	Fast Link Timer AN timer is reduced if this bit is set.
LINK OK FIX EN	25	1b	Link OK Fix Enable Control for enabling/disabling LinkOK/SyncOK fix. Should be set for normal operation.
Reserved	26	0b	Reserved
Reserved	31:27	0x0	Reserved Write 0, ignore on read.

1. Bit loaded from EEPROM.

8.17.3 PCS Link Status - PCS_LSTS (0x420C; RO)

Field	Bit(s)	Initial Value	Description
LINK OK	0	0b	Link OK This bit denotes the current link ok status. 0b = Link down. 1b = Link up/OK.
SPEED	2:1	10b	Speed This bit denotes the current operating Speed. 00b = 10 Mb/s. 01b = 100 Mb/s. 10b = 1000 Mb/s. 11b = Reserved.
DUPLEX	3	1b	Duplex This bit denotes the current duplex mode. 1b = Full duplex. 0b = Half duplex.
SYNC OK	4	0b	Sync OK This bit indicates the current value of Sync OK from the PCS Sync state machine.
Reserved	15:5	0x0	Reserved Write 0, ignore on read.
AN COMPLETE	16	0b	AN Complete This bit indicates that the AN process has completed.This bit is set when the AN process reached the Link OK state. It is reset upon AN restart or reset. It is set even if the AN negotiation failed and no common capabilities were found.
AN PAGE RECEIVED	17	0b	AN Page Received This bit indicates that a link partner's page was received during an AN process. This bit is cleared on reads.
AN TIMEDOUT	18	0b	AN Timed Out This bit indicates an AN process was timed out. Valid after the *AN Complete* bit is set.
AN REMOTE FAULT	19	0b	AN Remote Fault This bit indicates that an AN page was received with a remote fault indication during an AN process. This bit cleared on reads.
AN ERROR (RWS)	20	0B	AN Error This bit indicates that a AN error condition was detected in SerDes/SGMII mode. Valid after the *AN Complete* bit is set. AN error conditions: SerDes mode: Both nodes not Full Duplex SGMII mode: PHY is set to 1000 Mb/s Half Duplex mode. Software can also force a AN error condition by writing to this bit (or can clear a existing AN error condition). This bit is cleared at the start of AN.
Reserved	31:21	0x0	Reserved Write 0, ignore on read.

8.17.4 AN Advertisement - PCS_ANADV (0x4218; R/W)

Field	Bit(s)	Initial Value	Description
Reserved	4:0	0x0	Reserved Write 0, ignore on read.
FDCAP	5	1b	Full Duplex Setting this bit indicates that the I350 is capable of full duplex operation. This bit should be set to 1b for normal operation.
HDCAP (RO)	6	0b	Half Duplex This bit indicates that the I350 is capable of half duplex operation. This bit is tied to 0b because the I350 does not support half duplex in SerDes mode.
ASM	8:7	00b[1]	Local PAUSE Capabilities The I350's PAUSE capability is encoded in this field. 00b = No PAUSE. 01b = Symmetric PAUSE. 10b = Asymmetric PAUSE to link partner. 11b = Both symmetric and asymmetric PAUSE to the I350.
Reserved	11:9	0x0	Reserved Write 0, ignore on read.
RFLT	13:12	00b	Remote Fault The I350's remote fault condition is encoded in this field. The I350 might indicate a fault by setting a non-zero remote fault encoding and re-negotiating. 00b = No error, link OK. 01b = Link failure. 10b = Offline. 11b = Auto-negotiation error.
Reserved	14	0x0	Reserved Write 0, ignore on read.
NEXTP	15	0b	Next Page Capable The I350 asserts this bit to request a next page transmission. The I350 clears this bit when no subsequent next pages are requested.
Reserved	31:16	0x0	Reserved

1. Loaded from EEPROM word 0x0F, bits 13:12.

8.17.5 Link Partner Ability - PCS_LPAB (0x421C; RO)

Field	Bit(s)	Initial Value	Description
Reserved	4:0	0x0	Reserved
LPFD	5	0b	LP Full Duplex (SerDes) When set to 1b, the link partner is capable of full duplex operation. When set to 0b, the link partner is not capable of full duplex mode. This bit is reserved while in SGMII mode.
LPHD	6	0b	LP Half Duplex (SerDes) When set to 1b, the link partner is capable of half duplex operation. When set to 0b, the link partner is not capable of half duplex mode. This bit is reserved while in SGMII mode.

Field	Bit(s)	Initial Value	Description
LPASM	8:7	00b	LP ASMDR/LP PAUSE (SerDes) The link partner's PAUSE capability is encoded in this field. 00b = No PAUSE. 01b = Symmetric PAUSE. 10b = Asymmetric PAUSE to link partner. 11b = Both symmetric and asymmetric PAUSE to the I350. These bits are reserved while in SGMII mode.
Reserved	9	0b	Reserved Write 0, ignore on read.
SGMII SPEED	11:10	00b	SerDes: reserved. Speed (SGMII): Speed indication from the PHY.
PRF	13:12	00b	LP Remote Fault (SerDes) The link partner's remote fault condition is encoded in this field. 00b = No error, link ok. 10b = Link failure. 01b = Offline. 11b = Auto-negotiation error. SGMII [13]: Reserved SGMII [12]: Duplex mode indication from the PHY.
ACK	14	0b	Acknowledge (SerDes) The link partner has acknowledge page reception. SGMII: Reserved.
LPNEXTP	15	0b	LP Next Page Capable (SerDes) The link partner asserts this bit to indicate its ability to accept next pages. SGMII: Link-OK indication from the PHY.
Reserved	31:16	0x0	Reserved Write 0, ignore on read.

8.17.6 Next Page Transmit - PCS_NPTX (0x4220; RW)

Field	Bit(s)	Initial Value	Description
CODE	10:0	0x0	Message/Un-formatted Code Field The Message Field is an 11-bit wide field that encodes 2048 possible messages. Un-formatted Code Field is an 11-bit wide field that might contain an arbitrary value.
TOGGLE	11	0b	Toggle This bit is used to ensure synchronization with the Link Partner during Next Page exchange. This bit always takes the opposite value of the *Toggle* bit in the previously exchanged Link Code Word. The initial value of the *Toggle* bit in the first Next Page transmitted is the inverse of bit 11 in the base Link Code Word and, therefore, can assume a value of 0b or 1b. The *Toggle* bit is set as follows: 0b = Previous value of the transmitted Link Code Word when 1b 1b = Previous value of the transmitted Link Code Word when 0b.
ACK2	12	0b	Acknowledge 2 Used to indicate that a device has successfully received its Link Partners' Link Code Word.
PGTYPE	13	0b	Message/Un-formatted Page This bit is used to differentiate a Message Page from an Un-formatted Page. The encoding is: 0b = Un-formatted page. 1b = Message page.

Field	Bit(s)	Initial Value	Description
Reserved	14	-	Reserved Write 0, ignore on read.
NXTPG	15	0b	Next Page Used to indicate whether or not this is the last Next Page to be transmitted. The encoding is: 0b = Last page. 1b = Additional Next Pages follow.
Reserved	31:16	-	Reserved Write 0, ignore on read.

8.17.7 Link Partner Ability Next Page - PCS_LPABNP (0x4224; RO)

Field	Bit(s)	Initial Value	Description
CODE	10:0	-	Message/Un-formatted Code Field The Message Field is an 11-bit wide field that encodes 2048 possible messages. Un-formatted Code Field is an 11-bit wide field that might contain an arbitrary value.
TOGGLE	11	-	Toggle This bit is used to ensure synchronization with the Link Partner during Next Page exchange. This bit always takes the opposite value of the *Toggle* bit in the previously exchanged Link Code Word. The initial value of the *Toggle* bit in the first Next Page transmitted is the inverse of bit 11 in the base Link Code Word and, therefore, can assume a value of 0b or 1b. The *Toggle* bit is set as follows: 0b = Previous value of the transmitted Link Code Word when 1b 1b = Previous value of the transmitted Link Code Word when 0b.
ACK2	12	-	Acknowledge 2 Used to indicate that a device has successfully received its Link Partners' Link Code Word.
MSGPG	13	-	Message Page This bit is used to differentiate a Message Page from an Un-formatted Page. The encoding is: 0b = Un-formatted page. 1b = Message page.
ACK	14	-	Acknowledge The Link Partner has acknowledged Next Page reception.
NXTPG	15	-	Next Page Used to indicate whether or not this is the last Next Page to be transmitted. The encoding is: 0b = Last page. 1b = Additional Next Pages follow.
Reserved	31:16	-	Reserved Write 0, ignore on read.

8.17.8 SFP I2C Command- I2CCMD (0x1028; R/W)

This register is used by software to read or write to the configuration registers in an SFP module when the *CTRL_EXT.I2C Enabled* bit is set to 1.

Note: According to the SFP specification, only reads are allowed from this interface; however, SFP vendors also provide a writable register through this interface (for example, PHY registers). As a result, write capability is also supported.

Field	Bit(s)	Initial Value	Description
DATA	15:0	X	Data In a write command, software places the data bits and then the MAC shifts them out to the I^2C bus. In a read command, the MAC reads these bits serially from the I^2C bus and then software reads them from this location. ***Note:*** **This field is read in byte order and not in word order.**
REGADD	23:16	0x0	I^2C Register Address For example, register 0, 1, 2... 255.
PHYADD	26:24	0x0	Device Address bits 3 -1 The actual address used is b{1010, PHYADD[2:0], 0}.
OP	27	0b	Op Code 0b = I^2C write. 1b = I^2C read.
Reset	28	0b	Reset Sequence If set, sends a reset sequence before the actual read or write. This bit is self clearing. A reset sequence is defined as nine consecutive stop conditions.
R	29	0b	Ready Bit Set to 1b by the I350 at the end of the I^2C transaction. For example, indicates a read or write has completed. Reset by a software write of a command.
IE	30	0b	Interrupt Enable When set to 1b by software, it causes an Interrupt to be asserted to indicate the end of an I^2C cycle (*ICR.MDAC*).Reserved
E	31	0b	Error This bit set is to 1b by hardware when it fails to complete an I^2C read. Reset by a software write of a command. Note: Bit is valid only when Ready bit is set.

8.17.9 SFP I2C Parameters - I2CPARAMS (0x102C; R/W)

This register is used to set the parameters for the I^2C access to the SFP module and to allow bit banging access to the I^2C interface.

Field	Bit(s)	Initial Value	Description
Write Time	4:0	110b	Write Time Defines the delay between a write access and the next access. The value is in milliseconds. A value of zero is not valid.
Read Time	7:5	010b	Read Time Defines the delay between a read access and the next access. The value is in microseconds. A value of Zero is not valid
I2CBB_EN	8	0b	I^2C Bit Bang Enable If set, the I^2C_CLK and I^2C_DATA lines are controlled via the CLK, DATA and DATA_OE_N fields of this register. Otherwise, they are controlled by the hardware machine activated via the I2CCMD or MDIC registers.

Field	Bit(s)	Initial Value	Description
CLK	9	0b	I^2C Clock While in bit bang mode, controls the value driven on the I2C_CLK pad of this port.
DATA_OUT	10	0b	I^2C_DATA While in bit bang mode and when the DATA_OE_N field is zero, controls the value driven on the I2C_DATA pad of this port.
DATA_OE_N	11	0b	I^2C_DATA_OE_N While in bit bang mode, controls the direction of the I2C_DATA pad of this port. 0b = Pad is output. 1b = Pad is input.
DATA_IN (RO)	12	X	I^2C_DATA_IN Reflects the value of the I2C_DATA pad. While in bit bang mode when the DATA_OE_N field is zero, this field reflects the value set in the DATA_OUT field.
CLK_OE_N	13	0b	I^2C Clock Output Enable While in bit bang mode, controls the direction of the I2C_CLK pad of this port. 0b = Pad is output. 1b = Pad is input.
CLK_IN (RO)	14	X	I^2C Clock In Value Reflects the value of the I2C_CLK pad. While in bit bang mode when the CLK_OE_N field is zero, this field reflects the value set in the CLK_OUT field.
clk_stretch_dis	15	0b	0b - Enable slave clock stretching support in I^2C access. 1b - Disable clock stretching support in I^2C access.
Reserved	31:16	0x0	Reserved

8.18 Statistics Register Descriptions

All Statistics registers reset when read. In addition, they stick at 0xFFFF_FFFF when the maximum value is reached.

For the receive statistics it should be noted that a packet is indicated as received if it passes the I350's filters and is placed into the packet buffer memory. A packet does not have to be transferred to host memory in order to be counted as received.

Due to divergent paths between interrupt-generation and logging of relevant statistics counts, it might be possible to generate an interrupt to the system for a noteworthy event prior to the associated statistics count actually being increased. This is extremely unlikely due to expected delays associated with the system interrupt-collection and ISR delay, but might be observed as an interrupt for which statistics values do not quite make sense. Hardware guarantees that any event noteworthy of inclusion in a statistics count is reflected in the appropriate count within 1 μs; a small time-delay prior to a read of statistics might be necessary to avoid the potential for receiving an interrupt and observing an inconsistent statistics count as part of the ISR.

8.18.1 CRC Error Count - CRCERRS (0x4000; RC)

Counts the number of receive packets with CRC errors. In order for a packet to be counted in this register, it must pass address filtering and must be 64 bytes or greater (from <Destination Address> through <CRC>, inclusively) in length. If receives are not enabled, then this register does not increment.

Field	Bit(s)	Initial Value	Description
CEC	31:0	0x0	CRC error count

8.18.2 Alignment Error Count - ALGNERRC (0x4004; RC)

Counts the number of receive packets with alignment errors (the packet is not an integer number of bytes in length). In order for a packet to be counted in this register, it must pass address filtering and must be 64 bytes or greater (from <Destination Address> through <CRC>, inclusive) in length. If receives are not enabled, then this register does not increment. This register is valid only in MII mode during 10/100 Mb/s operation.

Field	Bit(s)	Initial Value	Description
AEC	31:0	0x0	Alignment error count

8.18.3 Symbol Error Count - SYMERRS (0x4008; RC)

Counts the number of symbol errors between reads. The count increases for every bad symbol received, whether or not a packet is currently being received and whether or not the link is up. When working in SerDes/SGMII/1000BASE-KX mode these statistics can be read from the SCVPC register.

Field	Bit(s)	Initial Value	Description
SYMERRS	31:0	0x0	Symbol Error Count

8.18.4 RX Error Count - RXERRC (0x400C; RC)

Counts the number of packets received in which RX_ER was asserted by the PHY. In order for a packet to be counted in this register, it must pass address filtering and must be 64 bytes or greater (from <Destination Address> through <CRC>, inclusive) in length. If receives are not enabled, then this register does not increment.

This register is not available in SerDes/SGMII/1000BASE-KX modes.

Field	Bit(s)	Initial Value	Description
RXEC	31:0	0x0	RX error count

8.18.5 Missed Packets Count - MPC (0x4010; RC)

Counts the number of missed packets. Packets are missed when the receive FIFO has insufficient space to store the incoming packet. This can be caused because of too few buffers allocated, or because there is insufficient bandwidth on the PCI bus. Events setting this counter causes *ICR.Rx Miss*, the Receiver Overrun Interrupt, to be set. This register does not increment if receives are not enabled.

These packets are also counted in the Total Packets Received register as well as in Total Octets Received.

Field	Bit(s)	Initial Value	Description
MPC	31:0	0x0	Missed Packets Count

8.18.6 Single Collision Count - SCC (0x4014; RC)

This register counts the number of times that a successfully transmitted packet encountered a single collision. This register only increments if transmits are enabled (*TCTL.EN* is set) and the I350 is in half-duplex mode.

Field	Bit(s)	Initial Value	Description
SCC	31:0	0x0	Number of times a transmit encountered a single collision.

8.18.7 Excessive Collisions Count - ECOL (0x4018; RC)

When 16 or more collisions have occurred on a packet, this register increments, regardless of the value of collision threshold. If collision threshold is set below 16, this counter won't increment. This register only increments if transmits are enabled (*TCTL.EN* is set) and the I350 is in half-duplex mode.

Field	Bit(s)	Initial Value	Description
ECC	31:0	0x0	Number of packets with more than 16 collisions.

8.18.8 Multiple Collision Count - MCC (0x401C; RC)

This register counts the number of times that a transmit encountered more than one collision but less than 16. This register only increments if transmits are enabled (*TCTL.EN* is set) and the I350 is in half-duplex mode.

Field	Bit(s)	Initial Value	Description
MCC	31:0	0x0	Number of times a successful transmit encountered multiple collisions.

8.18.9 Late Collisions Count - LATECOL (0x4020; RC)

Late collisions are collisions that occur after one slot time. This register only increments if transmits are enabled (*TCTL.EN* is set) and the I350 is in half-duplex mode.

Field	Bit(s)	Initial Value	Description
LCC	31:0	0x0	Number of packets with late collisions.

8.18.10 Collision Count - COLC (0x4028; RC)

This register counts the total number of collisions seen by the transmitter. This register only increments if transmits are enabled (*TCTL.EN* is set) and the I350 is in half-duplex mode.

Field	Bit(s)	Initial Value	Description
CCC	31:0	0x0	Total number of collisions experienced by the transmitter.

8.18.11 Defer Count - DC (0x4030; RC)

This register counts defer events. A defer event occurs when the transmitter cannot immediately send a packet due to the medium being busy either because another device is transmitting, the IPG timer has not expired, half-duplex deferral events, reception of XOFF frames, or the link is not up. This register only increments if transmits are enabled (*TCTL.EN* is set). This counter does not increment for streaming transmits that are deferred due to TX IPG.

Field	Bit(s)	Initial Value	Description
CDC	31:0	0x0	Number of defer events.

8.18.12 Transmit with No CRS - TNCRS (0x4034; RC)

This register counts the number of successful packet transmissions in which the CRS input from the PHY was not asserted within one slot time of start of transmission from the MAC. Start of transmission is defined as the assertion of TX_EN to the PHY.

The PHY should assert CRS during every transmission. Failure to do so might indicate that the link has failed, or the PHY has an incorrect link configuration. This register only increments if transmits are enabled (*TCTL.EN* is set). This register is not valid in SGMII mode and is only valid when the I350 is operating at half duplex.

Field	Bit(s)	Initial Value	Description
TNCRS	31:0	0x0	Number of transmissions without a CRS assertion from the PHY.

8.18.13 Host Transmit Discarded Packets by MAC Count - HTDPMC (0x403C; RC)

This register counts the number of packets sent by the host (and not the manageability engine) that are dropped by the MAC. This can include packets dropped because of excessive collisions or link fail events.

Field	Bit(s)	Initial Value	Description
HTDPMC	31:0	0x0	Number of packets sent by the host but discarded by the MAC

8.18.14 Receive Length Error Count - RLEC (0x4040; RC)

This register counts receive length error events. A length error occurs if an incoming packet passes the filter criteria but is undersized or oversized. Packets less than 64 bytes are undersized. Packets over 1518, 1522 or 1526 bytes (according to the number of VLAN tags present) are oversized if Long Packet Enable (*RCTL.LPE*) is 0b. If *LPE* is 1b, then an incoming, packet is considered oversized if it exceeds the size defined in *RLPML.RLPML* field.

If receives are not enabled, this register does not increment. These lengths are based on bytes in the received packet from <Destination Address> through <CRC>, inclusive. Packets sent to the manageability engine are included in this counter.

Note: Runt packets smaller than 25 bytes may not be counted by this counter.

Field	Bit(s)	Initial Value	Description
RLEC	31:0	0x0	Number of packets with receive length errors.

8.18.15 XON Received Count - XONRXC (0x4048; RC)

This register counts the number of valid XON packets received. XON packets can use the global address, or the station address. This register only increments if receives are enabled (*RCTL.RXEN* is set).

Field	Bit(s)	Initial Value	Description
XONRXC	31:0	0x0	Number of XON packets received.

8.18.16 XON Transmitted Count - XONTXC (0x404C; RC)

This register counts the number of XON packets transmitted. These can be either due to a full queue or due to software initiated action (using TCTL.SWXOFF). This register only increments if transmits are enabled (*TCTL.EN* is set).

Field	Bit(s)	Initial Value	Description
XONTXC	31:0	0x0	Number of XON packets transmitted.

8.18.17 XOFF Received Count - XOFFRXC (0x4050; RC)

This register counts the number of valid XOFF packets received. XOFF packets can use the global address or the station address. This register only increments if receives are enabled (*RCTL.RXEN* is set).

Field	Bit(s)	Initial Value	Description
XOFFRXC	31:0	0x0	Number of XOFF packets received.

8.18.18 XOFF Transmitted Count - XOFFTXC (0x4054; RC)

This register counts the number of XOFF packets transmitted. These can be either due to a full queue or due to software initiated action (using TCTL.SWXOFF). This register only increments if transmits are enabled (*TCTL.EN* is set).

Field	Bit(s)	Initial Value	Description
XOFFTXC	31:0	0x0	Number of XOFF packets transmitted.

8.18.19 FC Received Unsupported Count - FCRUC (0x4058; RC)

This register counts the number of unsupported flow control frames that are received.

The *FCRUC* counter increments when a flow control packet is received that matches either the reserved flow control multicast address (in the *FCAH/L* register) or the MAC station address, and has a matching flow control type field match (value in the *FCT* register), but has an incorrect opcode field. This register only increments if receives are enabled (*RCTL.RXEN* is set).

Note: When the *RCTL.PMCF* bit is set to 1b then the *FCRUC* counter will increment on reception of packets that don't match standard address filtering.

Field	Bit(s)	Initial Value	Description
FCRUC	31:0	0x0	Number of unsupported flow control frames received.

8.18.20 Packets Received [64 Bytes] Count - PRC64 (0x405C; RC)

This register counts the number of good packets received that are exactly 64 bytes (from <Destination Address> through <CRC>, inclusive) in length. Packets that are counted in the Missed Packet Count register are not counted in this register. Packets sent to the manageability engine are included in this counter. This register does not include received flow control packets and increments only if receives are enabled (*RCTL.RXEN* is set).

Field	Bit(s)	Initial Value	Description
PRC64	31:0	0x0	Number of packets received that are 64 bytes in length.

8.18.21 Packets Received [65—127 Bytes] Count - PRC127 (0x4060; RC)

This register counts the number of good packets received that are 65-127 bytes (from <Destination Address> through <CRC>, inclusive) in length. Packets that are counted in the Missed Packet Count register are not counted in this register. Packets sent to the manageability engine are included in this counter. This register does not include received flow control packets and increments only if receives are enabled (*RCTL.RXEN* is set).

Field	Bit(s)	Initial Value	Description
PRC127	31:0	0x0	Number of packets received that are 65-127 bytes in length.

8.18.22 Packets Received [128—255 Bytes] Count - PRC255 (0x4064; RC)

This register counts the number of good packets received that are 128-255 bytes (from <Destination Address> through <CRC>, inclusive) in length. Packets that are counted in the Missed Packet Count register are not counted in this register. Packets sent to the manageability engine are included in this counter. This register does not include received flow control packets and increments only if receives are enabled (*RCTL.RXEN* is set).

Field	Bit(s)	Initial Value	Description
PRC255	31:0	0x0	Number of packets received that are 128-255 bytes in length.

8.18.23 Packets Received [256—511 Bytes] Count - PRC511 (0x4068; RC)

This register counts the number of good packets received that are 256-511 bytes (from <Destination Address> through <CRC>, inclusive) in length. Packets that are counted in the Missed Packet Count register are not counted in this register. Packets sent to the manageability engine are included in this counter. This register does not include received flow control packets and increments only if receives are enabled (*RCTL.RXEN* is set).

Field	Bit(s)	Initial Value	Description
PRC511	31:0	0x0	Number of packets received that are 256-511 bytes in length.

8.18.24 Packets Received [512—1023 Bytes] Count - PRC1023 (0x406C; RC)

This register counts the number of good packets received that are 512-1023 bytes (from <Destination Address> through <CRC>, inclusive) in length. Packets that are counted in the Missed Packet Count register are not counted in this register. Packets sent to the manageability engine are included in this counter. This register does not include received flow control packets and increments only if receives are enabled (*RCTL.RXEN* is set).

Field	Bit(s)	Initial Value	Description
PRC1023	31:0	0x0	Number of packets received that are 512-1023 bytes in length.

8.18.25 Packets Received [1024 to Max Bytes] Count - PRC1522 (0x4070; RC)

This register counts the number of good packets received that are from 1024 bytes to the maximum (from <Destination Address> through <CRC>, inclusive) in length. The maximum is dependent on the current receiver configuration (for example, *RCTL.LPE*, etc.) and the type of packet being received. If a packet is counted in Receive Oversized Count, it is not counted in this register (refer to Section 8.18.37). This register does not include received flow control packets and only increments if the packet has passed address filtering and receives are enabled (*RCTL.RXEN* is set). Packets sent to the manageability engine are included in this counter.

Due to changes in the standard for maximum frame size for VLAN tagged frames in 802.3, the I350 accepts packets that have a maximum length of 1522 bytes. The RMON statistics associated with this range has been extended to count 1522 byte long packets. If *CTRL_EXT.EXT_VLAN* is set, packets up to 1526 bytes are counted by this counter.

Field	Bit(s)	Initial Value	Description
PRC1522	31:0	0x0	Number of packets received that are 1024-Max bytes in length.

8.18.26 Good Packets Received Count - GPRC (0x4074; RC)

This register counts the number of good packets received of any legal length. The legal length for the received packet is defined by the value of Long Packet Enable (*RCTL.LPE*) (refer to Section 8.18.37). This register does not include received flow control packets and only counts packets that pass filtering. This register only increments if receives are enabled (*RCTL.RXEN* is set). This register does not count packets counted by the Missed Packet Count (*MPC*) register. Packets sent to the manageability engine (*MNGPRC*) or dropped by the VMDq queueing process (*SDPC*) are included in this counter.

Note: *GPRC* can count packets interrupted by a link disconnect although they have a CRC error.

Field	Bit(s)	Initial Value	Description
GPRC	31:0	0x0	Number of good packets received (of any length).

8.18.27 Broadcast Packets Received Count - BPRC (0x4078; RC)

This register counts the number of good (no errors) broadcast packets received. This register does not count broadcast packets received when the broadcast address filter is disabled. This register only increments if receives are enabled (*RCTL.RXEN* is set). This register does not count packets counted by the Missed Packet Count (*MPC*) register. Packets sent to the manageability engine (*MNGPRC*) or dropped by the VMDq queueing process (*SDPC*) are included in this counter.

Field	Bit(s)	Initial Value	Description
BPRC	31:0	0x0	Number of broadcast packets received.

8.18.28 Multicast Packets Received Count - MPRC (0x407C; RC)

This register counts the number of good (no errors) multicast packets received. This register does not count multicast packets received that fail to pass address filtering nor does it count received flow control packets. This register only increments if receives are enabled (*RCTL.RXEN* is set). This register does not count packets counted by the Missed Packet Count (MPC) register. Packets sent to the manageability engine (*MNGPRC*) or dropped by the VMDq queueing process (*SDPC*) are included in this counter.

Field	Bit(s)	Initial Value	Description
MPRC	31:0	0x0	Number of multicast packets received.

8.18.29 Good Packets Transmitted Count - GPTC (0x4080; RC)

This register counts the number of good (no errors) packets transmitted. A good transmit packet is considered one that is 64 or more bytes in length (from <Destination Address> through <CRC>, inclusively) in length. This does not include transmitted flow control packets. This register only increments if transmits are enabled (*TCTL.EN* is set).

Field	Bit(s)	Initial Value	Description
GPTC	31:0	0x0	Number of good packets transmitted.

8.18.30 Good Octets Received Count - GORCL (0x4088; RC)

These registers make up a 64-bit register that counts the number of good (no errors) octets received. This register includes bytes received in a packet from the <Destination Address> field through the <CRC> field, inclusive; *GORCL* must be read before *GORCH*.

In addition, it sticks at 0xFFFF_FFFF_FFFF_FFFF when the maximum value is reached. Only octets of packets that pass address filtering are counted in this register. This register does not count octets of packets counted by the Missed Packet Count (*MPC*) register. Octets of packets sent to the manageability engine are included in this counter. This register only increments if receives are enabled (*RCTL.RXEN* is set).

These octets do not include octets of received flow control packets.

Field	Bit(s)	Initial Value	Description
GORCL	31:0	0x0	Number of good octets received – lower 4 bytes.

8.18.31 Good Octets Received Count - GORCH (0x408C; RC)

Field	Bit(s)	Initial Value	Description
GORCH	31:0	0x0	Number of good octets received – upper 4 bytes.

8.18.32 Good Octets Transmitted Count - GOTCL (0x4090; RC)

These registers make up a 64-bit register that counts the number of good (no errors) packets transmitted. This register must be accessed using two independent 32-bit accesses; *GOTCL* must be read before *GOTCH*.

In addition, it sticks at 0xFFFF_FFFF_FFFF_FFFF when the maximum value is reached. This register includes bytes transmitted in a packet from the <Destination Address> field through the <CRC> field, inclusive. This register counts octets in successfully transmitted packets that are 64 or more bytes in length. This register only increments if transmits are enabled (*TCTL.EN* is set).

These octets do not include octets in transmitted flow control packets.

Field	Bit(s)	Initial Value	Description
GOTCL	31:0	0x0	Number of good octets transmitted – lower 4 bytes.

8.18.33 Good Octets Transmitted Count - GOTCH (0x4094; RC)

Field	Bit(s)	Initial Value	Description
GOTCH	31:0	0x0	Number of good octets transmitted – upper 4 bytes.

8.18.34 Receive No Buffers Count - RNBC (0x40A0; RC)

This register counts the number of times that frames were received when there were no available buffers in host memory to store those frames (receive descriptor head and tail pointers were equal). The packet is still received if there is space in the FIFO. This register only increments if receives are enabled (*RCTL.RXEN* is set).

Notes:

1. This register does not increment when flow control packets are received.
2. If a packet is replicated, this counter counts each replication of the packet that is dropped.

Field	Bit(s)	Initial Value	Description
RNBC	31:0	0x0	Number of receive no buffer conditions.

8.18.35 Receive Undersize Count - RUC (0x40A4; RC)

This register counts the number of received frames that passed address filtering, and were less than minimum size (64 bytes from <Destination Address> through <CRC>, inclusive), and had a valid CRC. This register only increments if receives are enabled (*RCTL.RXEN* is set). Packets sent to the manageability engine are included in this counter.

Note: Runt packets smaller than 25 bytes may not be counted by this counter.

Field	Bit(s)	Initial Value	Description
RUC	31:0	0x0	Number of receive undersize errors.

8.18.36 Receive Fragment Count - RFC (0x40A8; RC)

This register counts the number of received frames that passed address filtering, and were less than minimum size (64 bytes from <Destination Address> through <CRC>, inclusive), but had a bad CRC (this is slightly different from the Receive Undersize Count register). This register only increments if receives are enabled (*RCTL.RXEN* is set). Packets sent to the manageability engine are included in this counter.

Note: Runt packets smaller than 25 bytes may not be counted by this counter.

Field	Bit(s)	Initial Value	Description
RFC	31:0	0x0	Number of receive fragment errors.

8.18.37 Receive Oversize Count - ROC (0x40AC; RC)

This register counts the number of received frames with valid CRC field that passed address filtering, and were greater than maximum size. For definition of oversized packets, refer to Section 7.1.1.4.

If receives are not enabled, this register does not increment. These lengths are based on bytes in the received packet from <Destination Address> through <CRC>, inclusive. Packets sent to the manageability engine are included in this counter.

Field	Bit(s)	Initial Value	Description
ROC	31:0	0x0	Number of receive oversize errors.

8.18.38 Receive Jabber Count - RJC (0x40B0; RC)

This register counts the number of received frames that passed address filtering, and were greater than maximum size and had a bad CRC (this is slightly different from the Receive Oversize Count register). For definition of oversized packets, refer to Section 7.1.1.4.

If receives are not enabled, this register does not increment. These lengths are based on bytes in the received packet from <Destination Address> through <CRC>, inclusive. Packets sent to the manageability engine are included in this counter.

Field	Bit(s)	Initial Value	Description
RJC	31:0	0x0	Number of receive jabber errors.

8.18.39 Management Packets Received Count - MNGPRC (0x40B4; RC)

This register counts the total number of packets received that pass the management filters as described in Section 10.3. Any packets with errors are not counted, except packets that are dropped because the management receive FIFO is full.

Packets sent to both the host and the management interface are not counted by this counter.

Field	Bit(s)	Initial Value	Description
MNGPRC	31:0	0x0	Number of management packets received.

8.18.40 Management Packets Dropped Count - MPDC (0x40B8; RC)

This register counts the total number of packets received that pass the management filters as described in Section 10.3, that are dropped because the management receive FIFO is full. Management packets include any packet directed to the manageability console (for example, BMC and ARP packets).

Field	Bit(s)	Initial Value	Description
MPDC	31:0	0x0	Number of management packets dropped.

8.18.41 Management Packets Transmitted Count - MNGPTC (0x40BC; RC)

This register counts the total number of transmitted packets originating from the manageability path.

Field	Bit(s)	Initial Value	Description
MPTC	31:0	0x0	Number of management packets transmitted.

8.18.42 BMC2OS Packets Sent by BMC - B2OSPC (0x8FE0; RC)

This register counts the total number of transmitted packets sent from the manageability path that were sent to host. This includes packets received by the host and packet dropped in the I350 due to congestion conditions.

Counter is cleared when read by driver. Counter is also cleared by PCIe reset and Software reset. When reaching maximum value counter does not wrap-around.

Field	Bit(s)	Initial Value	Description
B2OSPC	31:0	0x0	BMC2OS packets sent by BMC.

8.18.43 BMC2OS Packets Received by host - B2OGPRC (0x4158; RC)

This register counts the total number of packets originating from the BMC that reached the host.

If a packet is replicated, this counter counts each replication of the packet.

Counter is cleared when read by driver. Counter is also cleared by PCIe reset and Software reset. When reaching maximum value counter does not wrap-around.

Field	Bit(s)	Initial Value	Description
B2OGPRC	31:0	0x0	BMC2OS packets received by host.

8.18.44 OS2BMC Packets Received by BMC - O2BGPTC (0x8FE4; RC)

This register counts the total number of packets originating from the host that reached the NC-SI interface.

Counter is cleared when read by driver. Counter is also cleared by PCIe reset and Software reset. When reaching maximum value counter does not wrap-around.

Field	Bit(s)	Initial Value	Description
O2BGPRC	31:0	0x0	OS2BMC good packets received count.

8.18.45 OS2BMC Packets Transmitted by Host - O2BSPC (0x415C; RC)

This register counts the total number of packets originating from the function that were sent to the manageability path. This includes packets received by the BMC and packet dropped in the I350 due to congestion conditions.

Packets dropped due to security reasons, for example anti spoofing, are not counted by this counter.

Counter is cleared when read by driver. Counter is also cleared by PCIe reset and Software reset. When reaching maximum value counter does not wrap-around.

Field	Bit(s)	Initial Value	Description
O2BGPRC	31:0	0x0	OS2BMC good packets received count.

8.18.46 Total Octets Received - TORL (0x40C0; RC)

These registers make up a logical 64-bit register which counts the total number of octets received. This register must be accessed using two independent 32-bit accesses; TORL must be read before TORH. This register sticks at 0xFFFF_FFFF_FFFF_FFFF when the maximum value is reached.

All packets received have their octets summed into this register, regardless of their length, whether they are erred, or whether they are flow control packets. This register includes bytes received in a packet from the <Destination Address> field through the <CRC> field, inclusive. This register only increments if receives are enabled (*RCTL.RXEN* is set).

Note: Broadcast rejected packets are counted in this counter (as opposed to all other rejected packets that are not counted).

Field	Bit(s)	Initial Value	Description
TORL	31:0	0x0	Number of total octets received – lower 4 bytes.

8.18.47 Total Octets Received - TORH (0x40C4; RC)

Field	Bit(s)	Initial Value	Description
TORH	31:0	0x0	Number of total octets received – upper 4 bytes.

8.18.48 Total Octets Transmitted - TOTL (0x40C8; RC)

These registers make up a 64-bit register that counts the total number of octets transmitted. This register must be accessed using two independent 32-bit accesses; TOTL must be read before TOTH. This register sticks at 0xFFFF_FFFF_FFFF_FFFF when the maximum value is reached.

All transmitted packets have their octets summed into this register, regardless of their length or whether they are flow control packets. This register includes bytes transmitted in a packet from the <Destination Address> field through the <CRC> field, inclusive.

Octets transmitted as part of partial packet transmissions (for example, collisions in half-duplex mode) are not included in this register. This register only increments if transmits are enabled (*TCTL.EN* is set).

Field	Bit(s)	Initial Value	Description
TOTL	31:0	0x0	Number of total octets transmitted – lower 4 bytes.

8.18.49 Total Octets Transmitted - TOTH (0x40CC; RC)

Field	Bit(s)	Initial Value	Description
TOTH	31:0	0x0	Number of total octets transmitted – upper 4 bytes.

8.18.50 Total Packets Received - TPR (0x40D0; RC)

This register counts the total number of all packets received. All packets received are counted in this register, regardless of their length, whether they have errors, or whether they are flow control packets. This register only increments if receives are enabled (*RCTL.RXEN* is set).

Notes:

1. Broadcast rejected packets are counted in this counter (as opposed to all other rejected packets that are not counted).
2. Runt packets smaller than 25 bytes may not be counted by this counter.
3. *TPR* can count packets interrupted by a link disconnect although they have a CRC error.

Field	Bit(s)	Initial Value	Description
TPR	31:0	0x0	Number of all packets received.

8.18.51 Total Packets Transmitted - TPT (0x40D4; RC)

This register counts the total number of all packets transmitted. All packets transmitted are counted in this register, regardless of their length, or whether they are flow control packets.

Partial packet transmissions (collisions in half-duplex mode) are not included in this register. This register only increments if transmits are enabled (*TCTL.EN* is set). This register counts all packets, including standard packets, packets received over the SMBus, and packets generated by the PT function.

Field	Bit(s)	Initial Value	Description
TPT	31:0	0x0	Number of all packets transmitted.

8.18.52 Packets Transmitted [64 Bytes] Count - PTC64 (0x40D8; RC)

This register counts the number of packets transmitted that are exactly 64 bytes (from <Destination Address> through <CRC>, inclusive) in length. Partial packet transmissions (collisions in half-duplex mode) are not included in this register. This register does not include transmitted flow control packets (which are 64 bytes in length). This register only increments if transmits are enabled (*TCTL.EN* is set). This register counts all packets, including standard packets, packets received over the SMBus, and packets generated by the PT function.

Field	Bit(s)	Initial Value	Description
PTC64	31:0	0x0	Number of packets transmitted that are 64 bytes in length.

8.18.53 Packets Transmitted [65—127 Bytes] Count - PTC127 (0x40DC; RC)

This register counts the number of packets transmitted that are 65-127 bytes (from <Destination Address> through <CRC>, inclusive) in length. Partial packet transmissions (for example, collisions in half-duplex mode) are not included in this register. This register only increments if transmits are enabled (*TCTL.EN* is set). This register counts all packets, including standard packets, packets received over the SMBus, and packets generated by the PT function.

Field	Bit(s)	Initial Value	Description
PTC127	31:0	0x0	Number of packets transmitted that are 65-127 bytes in length.

8.18.54 Packets Transmitted [128—255 Bytes] Count - PTC255 (0x40E0; RC)

This register counts the number of packets transmitted that are 128-255 bytes (from <Destination Address> through <CRC>, inclusive) in length. Partial packet transmissions (collisions in half-duplex mode) are not included in this register. This register only increments if transmits are enabled (*TCTL.EN* is set). This register counts all packets, including standard packets, packets received over the SMBus, and packets generated by the PT function.

Field	Bit(s)	Initial Value	Description
PTC255	31:0	0x0	Number of packets transmitted that are 128-255 bytes in length.

8.18.55 Packets Transmitted [256—511 Bytes] Count - PTC511 (0x40E4; RC)

This register counts the number of packets transmitted that are 256-511 bytes (from <Destination Address> through <CRC>, inclusive) in length. Partial packet transmissions (for example, collisions in half-duplex mode) are not included in this register. This register only increments if transmits are enabled (*TCTL.EN* is set). This register counts all packets. Management packets must never be more than 200 bytes.

Field	Bit(s)	Initial Value	Description
PTC511	31:0	0x0	Number of packets transmitted that are 256-511 bytes in length.

8.18.56 Packets Transmitted [512—1023 Bytes] Count - PTC1023 (0x40E8; RC)

This register counts the number of packets transmitted that are 512-1023 bytes (from <Destination Address> through <CRC>, inclusive) in length. Partial packet transmissions (for example, collisions in half-duplex mode) are not included in this register. This register only increments if transmits are enabled (*TCTL.EN* is set). This register counts all packets. Management packets must never be more than 200 bytes.

Field	Bit(s)	Initial Value	Description
PTC1023	31:0	0x0	Number of packets transmitted that are 512-1023 bytes in length.

8.18.57 Packets Transmitted [1024 Bytes or Greater] Count - PTC1522 (0x40EC; RC)

This register counts the number of packets transmitted that are 1024 or more bytes (from <Destination Address> through <CRC>, inclusive) in length. Partial packet transmissions (for example, collisions in half-duplex mode) are not included in this register. This register only increments if transmits are enabled (*TCTL.EN* is set).

Due to changes in the standard for maximum frame size for VLAN tagged frames in 802.3, the I350 transmits packets that have a maximum length of 1522 bytes. The RMON statistics associated with this range has been extended to count 1522 byte long packets. This register counts all packets. Management packets must never be more than 200 bytes. If CTRL.*EXT_VLAN* is set, packets up to 1526 bytes are counted by this counter.

Field	Bit(s)	Initial Value	Description
PTC1522	31:0	0x0	Number of packets transmitted that are 1024 or more bytes in length.

8.18.58 Multicast Packets Transmitted Count - MPTC (0x40F0; RC)

This register counts the number of multicast packets transmitted. This register does not include flow control packets and increments only if transmits are enabled (*TCTL.EN* is set).

Field	Bit(s)	Initial Value	Description
MPTC	31:0	0x0	Number of multicast packets transmitted.

8.18.59 Broadcast Packets Transmitted Count - BPTC (0x40F4; RC)

This register counts the number of broadcast packets transmitted. This register only increments if transmits are enabled (*TCTL.EN* is set). This register counts all packets. Management packets must never be more than 200 bytes.

Field	Bit(s)	Initial Value	Description
BPTC	31:0	0x0	Number of broadcast packets transmitted count.

8.18.60 TCP Segmentation Context Transmitted Count - TSCTC (0x40F8; RC)

This register counts the number of TCP segmentation offload transmissions and increments once the last portion of the TCP segmentation context payload is segmented and loaded as a packet into the on-chip transmit buffer. Note that it is not a measurement of the number of packets sent out (covered by other registers). This register only increments if transmits and TCP segmentation offload are enabled.

This counter only counts pure TSO transmissions.

Field	Bit(s)	Initial Value	Description
TSCTC	31:0	0x0	Number of TCP Segmentation contexts transmitted count.

8.18.61 Interrupt Assertion Count - IAC (0x4100; RC)

This counter counts the total number of LAN interrupts generated in the system. In case of MSI-X systems, this counter reflects the total number of MSI-X messages that are emitted.

Field	Bit(s)	Initial Value	Description
IAC	31:0	0x0	This is a count of all the LAN interrupt assertions that have occurred.

8.18.62 Rx Packets to Host Count - RPTHC (0x4104; RC)

Field	Bit(s)	Initial Value	Description
RPTHC	31:0	0x0	This is a count of all the received packets sent to the host.

8.18.63 EEE TX LPI Count - TLPIC (0x4148; RC)

This register counts EEE TX LPI entry events. A EEE TX LPI event occurs when the transmitter enters EEE (IEEE802.3az) LPI state. This register only increments if transmits are enabled (*TCTL.EN* is set) and Link Mode is internal Copper PHY (*CTRL_EXT.LINK_MODE* = 00b).

Field	Bit(s)	Initial Value	Description
ETLPIC	31:0	0x0	Number of EEE TX LPI events.

8.18.64 EEE RX LPI Count - RLPIC (0x414C; RC)

This register counts EEE RX LPI entry events. A EEE RX LPI event occurs when the receiver detects link partner entry into EEE (IEEE802.3az) LPI state. This register only increments if receives are enabled (*RCTL.RXEN* is set) and Link Mode is internal Copper PHY (*CTRL_EXT.LINK_MODE* = 00b).

Field	Bit(s)	Initial Value	Description
ERLPIC	31:0	0x0	Number of EEE RX LPI events.

8.18.65 Host Good Packets Transmitted Count-HGPTC (0x4118; RC)

Field	Bit(s)	Initial Value	Description
HGPTC	31:0	0x0	Number of good packets transmitted by the host.

This register counts the number of good (non-erred) packets transmitted sent by the host. A good transmit packet is considered one that is 64 or more bytes in length (from <Destination Address> through <CRC>, inclusively) in length. This does not include transmitted flow control packets or packets sent by the manageability engine. This register only increments if transmits are enabled (*TCTL.EN* is set).

8.18.66 Receive Descriptor Minimum Threshold Count-RXDMTC (0x4120; RC)

Field	Bit(s)	Initial Value	Description
RXDMTC	31:0	0x0	This is a count of the receive descriptor minimum threshold events

This register counts the number of events where the number of descriptors in one of the Rx queues was lower than the threshold defined for this queue.

8.18.67 Host Good Octets Received Count - HGORCL (0x4128; RC)

Field	Bit(s)	Initial Value	Description
HGORCL	31:0	0x0	Number of good octets received by host – lower 4 bytes

8.18.68 Host Good Octets Received Count - HGORCH (0x412C; RC)

Field	Bit(s)	Initial Value	Description
HGORCH	31:0	0x0	Number of good octets received by host – upper 4 bytes

These registers make up a logical 64-bit register which counts the number of good (non-erred) octets received. This register includes bytes received in a packet from the <Destination Address> field through the <CRC> field, inclusive. This register must be accessed using two independent 32-bit accesses.; HGORCL must be read before HGORCH.

In addition, it sticks at 0xFFFF_FFFF_FFFF_FFFF when the maximum value is reached. Only packets that pass address filtering are counted in this register. This register counts only octets of packets that reached the host. The only exception is packets dropped by the DMA because of lack of descriptors in one of the queues. These packets are included in this counter.

This register only increments if receives are enabled (*RCTL.RXEN* is set).

8.18.69 Host Good Octets Transmitted Count - HGOTCL (0x4130; RC)

Field	Bit(s)	Initial Value	Description
HGOTCL	31:0	0x0	Number of good octets transmitted by host – lower 4 bytes

8.18.70 Host Good Octets Transmitted Count - HGOTCH (0x4134; RC)

Field	Bit(s)	Initial Value	Description
HGOTCH	31:0	0x0	Number of good octets transmitted by host – upper 4 bytes

These registers make up a logical 64-bit register which counts the number of good (non-erred) packets transmitted. This register must be accessed using two independent 32-bit accesses. This register resets whenever the upper 32 bits are read (HGOTCH).

In addition, it sticks at 0xFFFF_FFFF_FFFF_FFFF when the maximum value is reached. This register includes bytes transmitted in a packet from the <Destination Address> field through the <CRC> field, inclusive. This register counts octets in successfully transmitted packets which are 64 or more bytes in length. This register only increments if transmits are enabled (*TCTL.EN* is set).

These octets do not include octets in transmitted flow control packets or manageability packets.

8.18.71 Length Error Count - LENERRS (0x4138; RC)

Field	Bit(s)	Initial Value	Description
LENERRS	31:0	0x0	Length error count.

Counts the number of receive packets with Length errors. For example, valid packets (no CRC error) with a length/Type field with a value smaller or equal to 1500 greater than the frame size. In order for a packet to be counted in this register, it must pass address filtering and must be 64 bytes or greater (from <Destination Address> through <CRC>, inclusive) in length. If receives are not enabled, then this register does not increment.

8.18.72 SerDes/SGMII/KX Code Violation Packet Count - SCVPC (0x4228; RW)

This register contains the number of code violation packets received. Code violation is defined as an invalid received code in the middle of a packet.

Field	Bit(s)	Initial Value	Description
CODEVIO	31:0	0x0	Code Violation Packet Count: At any point of time this field specifies number of unknown protocol packets received. Valid only in SGMII/SerDes/1000BASE-KX modes.

8.18.73 Switch Security Violation Packet Count - SSVPC (0x41A0; RC)

This register counts Tx packets dropped due to switch security violations such as an SA anti spoof filtering. Relevant only in VMDq or IOV mode.

Field	Bit(s)	Initial Value	Description
SSVPC	31:0	0x0	Switch Security Violation Packet Count: This register counts Tx packets dropped due to switch security violations such as an SA anti spoof filtering.

8.18.74 Switch Drop Packet Count - SDPC (0x41A4; RC/W)

Field	Bit(s)	Initial Value	Description
SDPC	31:0	0x0	Switch Drop Packet Count: This register counts Rx packets dropped at the pool selection stage of the switch or by the storm control mechanism. For example, packets that were not routed to any of the pools and the *VT_CTL.Dis_Def_Pool* is set. Relevant only in VMDq or IOV mode.

8.18.75 Loopback Full Buffer Drop Packet Count - LPBKFBDPC (0x4150; RC/W)

Field	Bit(s)	Initial Value	Description
LPBKFDPC	31:0	0x0	Loopback buffer full drop packet count - counts the number of packets destined to the VM to VM loopback buffer that were dropped due to lack of space in the Loopback buffer. Note: Counter does not wrap around when reaching a value of 0xFFFFFFFF.

8.18.76 Management Full Buffer Drop Packet Count - MNGFBDPC (0x4154; RC/W)

Field	Bit(s)	Initial Value	Description
MNGFDPC	31:0	0x0	Management buffer full drop packet count - counts the number of packets destined to Management that were dropped due to lack of space in the Management buffer. Note: Counter does not wrap around when reaching a value of 0xFFFFFFFF

8.18.77 Statistical Counters Per Queue

The I350 supports 9 statistical counters per queue to reduce processing overhead in virtualization operating mode.

These counters are reset only when the physical function is reset. When a VF is enabled, the PF should initialize these registers to zero.

8.18.77.1 Per Queue Good Packets Received Count - VFGPRC (0x10010 + n*0x100 [n=0...7]; RW)

This register counts the number of legal length good packets received in queue[n]. The legal length for the received packet is defined by the value of Long Packet Enable (RCTL.LPE) (refer to Section 8.18.37). This register does not include received flow control packets and only counts packets that pass filtering. This register only increments if receive is enabled.

Note: VFGPRC may count packets interrupted by a link disconnect although they have a CRC error.

Unlike some other statistics registers that are not allocated per VM, this register is not cleared on read. Furthermore, the register wraps around back to 0x0000 on the next increment when reaching a value of 0xFFFF and then continues normal count operation.

Field	Bit(s)	Initial Value	Description
GPRC	31:0	0x0	Number of good packets received (of any length).

8.18.77.2 Per Queue Good Packets Transmitted Count - VFGPTC (0x10014 + n*0x100 [n=0...7]; RW)

This register counts the number of good (no errors) packets transmitted on queue[n]. A good transmit packet is considered one that is 64 or more bytes in length (from <Destination Address> through <CRC>, inclusively) in length. This does not include transmitted flow control packets. This register only increments if transmits are enabled (TCTL.EN is set). This counter includes loopback packets or packets later dropped by the MAC.

A multicast packet dropped by some of the destinations, but sent to others is counted by this counter

Note: Unlike some other statistic registers that are not allocated per VM, this register is not cleared on read. Furthermore, the register wraps around back to 0x0000 on the next increment when reaching a value of 0xFFFF and then continues normal count operation.

Field	Bit(s)	Initial Value	Description
GPTC	31:0	0x0	Number of good packets transmitted.

8.18.77.3 Per Queue Good Octets Received Count - VFGORC (0x10018 + n*0x100 [n=0...7]; RW)

This register counts the number of good (no errors) octets received on queue[n]. This register includes bytes received in a packet from the <Destination Address> field through the <CRC> field, inclusive.

Only octets of packets that pass address filtering are counted in this register. This register only increments if receive is enabled.

Note: VLAN tag is part of the byte count only if reported to the VM. I.e. if the DVMOLR.HIDE VLAN is not set for this VM. CRC is part of the byte count if DTXCTL.Count CRC is set.

Unlike some other statistic registers that are not allocated per VM, this register is not cleared on read. Furthermore, the register wraps around back to 0x0000 on the next increment when reaching a value of 0xFFFF and then continues normal count operation.

Field	Bit(s)	Initial Value	Description
GORC	31:0	0x0	Number of good octets received.

8.18.77.4 Per Queue Good Octets Transmitted Count - VFGOTC (0x10034 + n*0x100 [n=0...7]; RW)

This register counts the number of good (no errors) packets transmitted on queue[n]. This register includes bytes transmitted in a packet from the <Destination Address> field through the <CRC> field, inclusive. Register also counts any padding that were added by the hardware. This register counts octets in successfully transmitted packets that are 64 or more bytes in length. Octets counted do not include octets in transmitted flow control packets. This register only increments if transmit is enabled.

A multicast packet dropped by some of the destinations, but sent to others is counted by this counter

Note: VLAN tag is part of the byte count only if inserted by the VM. I.e. if the VMVIR[n].VLANA field for the VM equals 00 (use descriptor command) and the packet contains a VLAN either in the packet or in the descriptor. CRC is part of the byte count if DTXCTL.Count CRC is set.

Unlike some other statistic registers that are not allocated per VM, this register is not cleared on read. Furthermore, the register wraps around back to 0x0000 on the next increment when reaching a value of 0xFFFF and then continues normal count operation.

Field	Bit(s)	Initial Value	Description
GOTC	31:0	0x0	Number of good octets transmitted – lower 4 bytes.

8.18.77.5 Per Queue Multicast Packets Received Count - VFMPRC (0x10038 + n*0x100 [n=0...7]; RO)

This register counts the number of good (no errors) multicast packets received on queue[n]. This register does not count multicast packets received that fail to pass address filtering nor does it count received flow control packets. This register only increments if receive is enabled.

Note: Unlike some other statistic registers that are not allocated per VM, this register is not cleared on read. Furthermore, the register wraps around back to 0x0000 on the next increment when reaching a value of 0xFFFF and then continues normal count operation.

Field	Bit(s)	Initial Value	Description
MPRC	31:0	0x0	Number of multicast packets received.

8.18.77.6 Good TX Octets loopback Count - VFGOTLBC (0x10050 + n*0x100 [n=0...7]; RW)

This register counts the number of good (no errors) octets transmitted by the queues allocated to this VF that were sent to local VF. This counter includes packets that are sent to the LAN and to a local VM.

This register includes bytes transmitted in a packet from the <Destination Address> field through the <CRC> field, inclusive, including any padding added by the hardware. The VLAN tag added by the hardware is counted as part of the packet.

Note: VLAN tag is part of the byte count only if inserted by the VM. I.e. if the VMVIR[n].VLANA field for the VM equals 00 (use descriptor command) and the packet contains a VLAN either in the packet or in the descriptor. CRC is part of the byte count if DTXCTL.Count CRC is set.

Unlike some other statistics registers that are not allocated per VF, this register is not cleared on read. Furthermore, the register continues to count from 0x0000 on stepping beyond 0xFFFF.

Field	Bit(s)	Initial Value	Description
GOTLBC	31:0	0x0	Number of good octets transmitted to loopback

8.18.77.7 Good TX packets loopback Count - VFGPTLBC (0x10044+ n*0x100 [n=0...7]; RW)

This register counts the number of good (no errors) packets transmitted by the queues allocated to this VF that were sent to local VF. This counter includes packets that are sent to the LAN and to a local VM.

Note: Unlike some other statistics registers that are not allocated per VF, this register is not cleared on read. Furthermore, the register continues to count from 0x0000 on stepping beyond 0xFFFF.

Field	Bit(s)	Initial Value	Description
GPTLBC	31:0	0x0	Number of good packets transmitted to loopback

8.18.77.8 Good RX Octets loopback Count - VFGORLBC (0x10048+ n*0x100 [n=0...7]; RW)

This register counts the number of good (no errors) octets received by the queues allocated to this VF that were sent from some local VFs.

Note: VLAN tag is part of the byte count only if reported to the VM. I.e. if the DVMOLR.HIDE VLAN is not set for this VM. CRC is part of the byte count if DTXCTL.Count CRC is set.

Unlike some other statistics registers that are not allocated per VF, this register is not cleared on read. Furthermore, the register continues to count from 0x0000 on stepping beyond 0xFFFF.

Field	Bit(s)	Initial Value	Description
GORLBC	31:0	0x0	Number of good octets received from loopback

8.18.77.9 Good RX Packets loopback Count - VFGPRLBC (0x10040+ n*0x100 [n=0...7]; RW)

This register counts the number of good (no errors) packets received by the queues allocated to this VF that were sent from some local VFs.

Note: Unlike some other statistics registers that are not allocated per VF, this register is not cleared on read. Furthermore, the register continues to count from 0x0000 on stepping beyond 0xFFFF.

Field	Bit(s)	Initial Value	Description
GPRLBC	31:0	0x0	Number of good packets received from loopback

8.19 Manageability Statistics

This section describes a set of statistics counters used by the NC-SI interface and are not accessible to the host driver.

8.19.1 BMC Management Receive Packets Dropped Count - BMRPDC (0x4140; RC)

This register counts the total number of packets received that pass the management filters as described in Section 10.3, that are dropped because the management receive FIFO is full. Management packets include any packet directed to the manageability console (for example, BMC and ARP packets). This register is available to the firmware only.

Field	Bit(s)	Initial Value	Description
MPDC	31:0	0x0	Number of management packets dropped.

8.19.2 BMC Management Transmit Packets Dropped Count - BMTPDC (0x8FDC; RC)

This register counts the total number of packets received from the Out of Band Management interface, that are dropped because the management transmit FIFO is full or if the relevant NC-SI channel is disabled. This counter increases only if the packet is not sent to any destination (Host or Network).

Note: This register is available to Firmware only and is shared for BMC TX traffic destined to any of the ports.

Field	Bit(s)	Initial Value	Description
BMTPDC	31:0	0x0	Number of management packets dropped.

8.19.3 BMC Management Packets Transmitted Count - BMNGPTC (0x4144; RC)

This register counts the total number of transmitted packets originating from the manageability path. This counter increases once if the packet is sent to any destination (host or network).

This register is available to the firmware only.

Field	Bit(s)	Initial Value	Description
MPTC	31:0	0x0	Number of management packets transmitted.

8.19.4 BMC Management Packets Received Count - BMNGPRC (0x413C; RC)

This register counts the total number of packets received that pass the management filters as described in Section 10.3. Any packets with errors are not counted, except packets that are dropped because the management receive FIFO is full.

This register is available to the firmware only.

Field	Bit(s)	Initial Value	Description
MNGPRC	31:0	0x0	Number of management packets received.

8.19.5 BMC Total Unicast Packets Received - BUPRC (0x4400; RC)

This register counts the number of good (no errors) unicast packets received. This register does not count unicast packets received that fail to pass address filtering. This register does not count packets counted by the Missed Packet Count (*MPC*) register. Packets sent to the manageability engine are included in this counter.

This register is available to the firmware only.

Field	Bit(s)	Initial Value	Description
BUPRC	31:0	0x0	Number of Unicast packets received.

8.19.6 BMC Total Multicast Packets Received - BMPRC (0x4404; RC)

This register counts the same events as the *MPRC* register (Section 8.18.28) for the BMC usage.This register is available to the firmware only.

8.19.7 BMC Total Broadcast Packets Received - BBPRC (0x4408; RC)

This register counts the same events as the *BPRC* register (Section 8.18.27) for the BMC usage. This register is available to the firmware only.

8.19.8 BMC Total Unicast Packets Transmitted - BUPTC (0x440C; RC)

This register counts the number of unicast packets transmitted. This register is available to the firmware only.

Field	Bit(s)	Initial Value	Description
BUPTC	31:0	0x0	Number of unicast packets transmitted.

8.19.9 BMC Total Multicast Packets Transmitted - BMPTC (0x4410; RC)

This register counts the same events as the *MPTC* register (Section 8.18.58) for the BMC usage.This register is available to the firmware only.

8.19.10 BMC Total Broadcast Packets Transmitted - BBPTC (0x4414; RC)

This register counts the same events as the *BPTC* register (Section 8.18.59) for the BMC usage. This register is available to the firmware only.

8.19.11 BMC FCS Receive Errors - BCRCERRS (0x4418; RC)

This register counts the same events as the *CRCERRS* register (Section 8.18.1) for the BMC usage. This register is available to the firmware only.

8.19.12 BMC Alignment Errors - BALGNERRC (0x441C; RC)

This register counts the same events as the *ALGNERRC* register (Section 8.18.2) for the BMC usage. This register is available to the firmware only.

8.19.13 BMC Pause XON Frames Received - BXONRXC (0x4420; RC)

This register counts the same events as the *XONRXC* register (Section 8.18.15) for the BMC usage. This register is available to the firmware only.

8.19.14 BMC Pause XOFF Frames Received - BXOFFRXC (0x4424; RC)

This register counts the same events as the *XOFFRXC* register (Section 8.18.17) for the BMC usage. This register is available to the firmware only.

8.19.15 BMC Pause XON Frames Transmitted - BXONTXC (0x4428; RC)

This register counts the same events as the *XONTXC* register (Section 8.18.16) for the BMC usage. This register is available to the firmware only.

8.19.16 BMC Pause XOFF Frames Transmitted - BXOFFTXC (0x442C; RC)

This register counts the same events as the *XOFFTXC* register (Section 8.18.18) for the BMC usage. This register is available to the firmware only.

8.19.17 BMC Single Collision Transmit Frames- BSCC (0x4430; RC)

This register counts the same events as the *SCC* register (Section 8.18.6) for the BMC usage. This register is available to the firmware only.

8.19.18 BMC Multiple Collision Transmit Frames - BMCC (0x4434; RC)

This register counts the same events as the *MCC* register (Section 8.18.8) for the BMC usage. This register is available to the firmware only.

8.20 Wake Up Control Register Descriptions

8.20.1 Wakeup Control Register - WUC (0x5800; R/W)

The *PME_En* and *PME_Status* bits of this register are reset when LAN_PWR_GOOD is 0b. When AUX_PWR = 0b, these register bits also reset by de-asserting PE_RST_N and during a D3 to D0 transition.

Field	Bit(s)	Initial Value	Description
APME	0	0b[1]	Advance Power Management Enable If set to 1b, APM Wakeup is enabled. If this bit is set and the *APMPME* bit is cleared, reception of a magic packet asserts the *WUS.MAG* bit but does not assert a PME. ***Note:*** Bit is reset on Power on reset only.
PME_En	1	0b	PME_En This read/write bit is used by the software device driver to enable generation of a PME event without writing to the Power Management Control / Status Register (*PMCSR*) in the PCIe configuration space. Note: Bit reflects value of *PMCSR.PME_En* bit when the bit in the *PMCSR* register is modified. However when value of *WUC.PME_En* bit is modified by software device driver, value is not reflected in the *PMCSR.PME_En* bit. ***Note:*** Bit is reset only on power-on reset When AUX_PWR = 0 bit is reset also on de-assertion of PE_RST_N and during D3 to D0 transition.
PME_Status (R/W1C)	2	0b	PME_Status This bit is set when the I350 receives a wakeup event. It is the same as the PME_Status bit in the Power Management Control / Status Register (*PMCSR*). Writing a 1b to this bit clears also the PME_Status bit in the *PMCSR*. ***Note:*** Bit is reset only on power-on reset When AUX_PWR = 0 bit is reset also on de-assertion of PE_RST_N and during D3 to D0 transition.
APMPME	3	0b[1]	Assert PME On APM Wakeup If set to 1b, the I350 sets the PME_Status bit in the Power Management Control / Status Register (*PMCSR*) and asserts PE_WAKE_N and sends a PM_PME PCIe message when APM Wakeup is enabled (*WUC.APME* = 1) and the I350 receives a matching Magic Packet. ***Notes:*** 1. When *WUC.APMPME* is set PE_WAKE_N is asserted and a PM_PME message is sent even if *PMCSR.PME_En* is cleared. 2. Bit is reset on Power on reset only.
PPROXYE	4	0b	Port Proxying Enable When set to 1b Proxying of packets is enabled when device is in D3 low power state. ***Note:*** Proxy information and requirements is passed by Software driver to Firmware via the shared RAM Host interface (refer to Section 10.8, Section 8.22 and Section 10.8.2.4.2).
EN_APM_D0	5	0b[1]	Enable APM wake on D0 0b - Enable APM wake only when function is in D3 and *WUC.APME* is set to 1b. 1b - Always enable APM wake when *WUC.APME* is set to 1b. ***Note:*** Bit is reset on Power on reset only.
Reserved	31:6	0x0	Reserved Write 0, ignore on read.

1. Loaded from the EEPROM.

8.20.2 Wakeup Filter Control Register - WUFC (0x5808; R/W)

This register is used to enable each of the pre-defined and flexible filters for wakeup support. A value of 1b means the filter is turned on.; A value of 0b means the filter is turned off.

If the *NoTCO* bit is set, then any packet that passes the manageability packet filtering as described in Section 10.3, does not cause a Wake Up event.

Field	Bit(s)	Initial Value	Description
LNKC	0	0b	Link Status Change Wakeup Enable.
MAG	1	0b	Magic Packet Wakeup Enable.
EX	2	0b	Directed Exact Wakeup Enable.[1]
MC	3	0b	Directed Multicast Wakeup Enable.
BC	4	0b	Broadcast Wakeup Enable.
ARP Directed	5	0b	ARP Request Packet and IP4AT Match Wakeup Enable. Wake on match of any ARP request packet that passed main filtering and Target IP address also matches one of the valid *IP4AT* filters.
IPv4	6	0b	Directed IPv4 Packet Wakeup Enable.
IPv6	7	0b	Directed IPv6 Packet Wakeup Enable.
Reserved	8	0b	Reserved. Write 0, ignore on read.
NS	9	0b	IPV6 Neighbor Solicitation wakeup enable Wake on match of any NS packet that passed main filtering.
NS Directed	10	0b	IPV6 Neighbor Solicitation and directed DA match wakeup enable Wake on match of NS packet and Target IP address also matches *IPV6AT* filter.
ARP	11	0b	ARP Request Packet Wakeup Enable. Wake on match of any ARP request packet that passed main filtering.
Reserved	12	0b	Reserved. Write 0, ignore on read.
THS_WK	13	0b	Thermal Sensor Wakeup Enable.
FLEX_HQ	14	0b	Flex filters Host Queuing 0b - Do not use Flex filters for queueing decisions in D0 state. 1b - Use Flex filters enabled in *WUFC* register for queuing decisions in D0 state. Note: Should be enabled only when multi queueing is enabled (MRQC.Multiple Receive Queues = 010b or 000b).
NoTCO	15	0b	Ignore TCO/management packets for wake up. 0b - Ignore only TCO/management packets for wake up that meet the criteria defined in the *MNGONLY* register (I.e. are intended only for the BMC and not the Host). 1b - Ignore any TCO/management packets for wake up, even if in normal operation it's forwarded to the Host in addition to the BMC.
FLX0	16	0b	Flexible Filter 0 Enable.
FLX1	17	0b	Flexible Filter 1 Enable.
FLX2	18	0b	Flexible Filter 2 Enable.
FLX3	19	0b	Flexible Filter 3 Enable.
FLX4	20	0b	Flexible Filter 4 Enable.
FLX5	21	0b	Flexible Filter 5 Enable.
FLX6	22	0b	Flexible Filter 6 Enable.
FLX7	23	0b	Flexible Filter 7 Enable.
Reserved	30:24	0x0	Reserved. Write 0, ignore on read.
FW_RST_WK	31	0b	Enable Wake on Firmware Reset assertion. When set a Firmware reset causes system wake so that Software driver can re-send Proxying information to Firmware.

1. If the *RCTL.UPE* is set, and the EX bit is set also, any unicast packet wakes up the system.

8.20.3 Wakeup Status Register - WUS (0x5810; R/W1C)

This register is used to record statistics about all wakeup packets received. If a packet matches multiple criteria then multiple bits could be set. Writing a 1b to any bit clears that bit.

This register is not cleared when RST# is asserted. It is only cleared when LAN_PWR_GOOD is de-asserted or when cleared by the software device driver.

Note: If additional packets are received that match one of the wakeup filters, after the original wakeup packet is received, the *WUS* register is not updated with the new match detection until the register is cleared.

Field	Bit(s)	Initial Value	Description
LNKC	0	0b	Link Status Change.
MAG	1	0b	Magic Packet Received.
EX	2	0b	Directed Exact Packet Received The packet's address matched one of the 32 pre-programmed exact values in the Receive Address registers (*RAL[n]/RAH[n]*), the packet was a unicast packet and *RCTL.UPE* is set to 1b or the packet was a unicast packet hashed to a value that corresponded to a 1 bit in the Unicast Table Array (*UTA*).
MC	3	0b	Directed Multicast Packet Received The packet was a multicast packet hashed to a value that corresponded to a 1 bit in the Multicast Table Array (*MTA*) or the packet was a multicast packet and *RCTL.MPE* is set to 1b.
BC	4	0b	Broadcast Packet Received.
ARP Directed	5	0b	ARP Request Packet with IPVA4AT filter Received. When set to 1b indicates a match on any ARP request packet that passed main filtering and Target IP address also matches one of the valid *IP4AT* filters.
IPv4	6	0b	Directed IPv4 Packet Received.
IPv6	7	0b	Directed IPv6 Packet Received.
MNG	8	0b	Indicates that a manageability event that should cause a PME happened.
NS	9	0b	IPV6 Neighbor Solicitation Received. When set to 1b indicates a match on any ICMPv6 packet such as Neighbor Solicitation (NS) packet or Multicast Listener Discovery (MLD) packet that passed main filtering.
NS Directed	10	0b	IPV6 Neighbor Solicitation with directed DA match Received. When set to 1b indicates a match on any ICMPv6 packet such as Neighbor Solicitation (NS) packet or Multicast Listener Discovery (MLD) packet that passed main filtering and the field placed in the Target IP address of a Neighbor Solicitation (NS) packet (9'th byte to 24'th byte of the ICMPv6 header) also matches a valid IPV6AT filter.
ARP	11	0b	ARP Request Packet Received. When set to 1b indicates a match on ARP request packet that passed main filtering.
Reserved	12	0b	Reserved. Write 0, ignore on read.
THS_WK	13	0b	Thermal Sensor Event.
Reserved	15:14	0x0	Reserved. Write 0b, ignore on read.
FLX0[1]	16	0b	Flexible Filter 0 Match.
FLX1[1]	17	0b	Flexible Filter 1 Match.
FLX2[1]	18	0b	Flexible Filter 2 Match.
FLX3[1]	19	0b	Flexible Filter 3 Match.
FLX4[1]	20	0b	Flexible Filter 4 Match.

Field	Bit(s)	Initial Value	Description
FLX5[1]	21	0b	Flexible Filter 5 Match.
FLX6[1]	22	0b	Flexible Filter 6 Match.
FLX7[1]	23	0b	Flexible Filter 7 Match.
Reserved	30:24	0b	Reserved. Write 0, ignore on read.
FW_RST_WK	31	0b	Wake due to Firmware Reset assertion event. When set to 1b, indicates that Firmware reset assertion caused system wake so that Software driver can re-send Proxying information to Firmware.

1. Bit is set only when flex filter match is detected and *WUFC.FLEX_HQ* is 0.

8.20.4 Wakeup Packet Length - WUPL (0x5900; RO)

This register indicates the length of the first wakeup packet received. It is valid if one of the bits in the Wakeup Status register (WUS) is set. It is not cleared by any reset.

Field	Bit(s)	Initial Value	Description
LEN	11:0	X	Length of wakeup packet. (If jumbo frames are enabled and the packet is longer than 2047 bytes then this field is 2047.)
Reserved	31:12	0x0	Reserved Write 0, ignore on read.

8.20.5 Wakeup Packet Memory - WUPM (0x5A00 + 4*n [n=0...31]; RO)

This register is read-only and it is used to store the first 128 bytes of the wakeup packet for software retrieval after system wakeup. It is not cleared by any reset.

Field	Bit(s)	Initial Value	Description
WUPD	31:0	X	Wakeup Packet Data

8.20.6 Proxying Filter Control Register - PROXYFC (0x5F60; R/W)

This register is used to enable each of the pre-defined and flexible filters for Proxying support. A value of 1b means the filter is turned on. A value of 0b means the filter is turned off.

If the *NoTCO* bit is set, then any packet that passes the manageability packet filtering as described in Section 10.3, is not forwarded to management for protocol offload even if it passes one of the Proxying Filters.

Field	Bit(s)	Initial Value	Description
D0_PROXY	0	0b	Enable Protocol Offload in D0. 0b - Enable Protocol offload only when device is in D3 low power state. 1b - Enable Protocol offload always. ***Note:*** Protocol offload is enabled only when the *WUC.PPROXYE* and *MANC.MPROXYE* bits are set to 1b.
Reserved	1	0b	Reserved. Write 0, ignore on read.
EX	2	0b	Directed Exact Proxy Enable.[1]
MC	3	0b	Directed Multicast Proxy Enable.
BC	4	0b	Broadcast Proxy Enable.
ARP Directed	5	0b	ARP Request Packet and IP4AT match Proxy Enable. If set to 1b forward to Management for proxying on match of any ARP request packet that passed main filtering and Target IP address also matches one of the valid *IP4AT* filters.
IPv4	6	0b	Directed IPv4 Packet Proxy Enable.
IPv6	7	0b	Directed IPv6 Packet Proxy Enable.
Reserved	8	0b	Reserved. Write 0, ignore on read.
NS	9	0b	IPV6 Neighbor Solicitation Proxy enable If set to 1b forward to Management for proxying on match of any ICMPv6 packet such as Neighbor Solicitation (NS) packet or Multicast Listener Discovery (MLD) packet that passed main filtering.
NS Directed	10	0b	IPV6 Neighbor Solicitation and directed DA match Proxy enable If set to 1b forward to Management for proxying on match of any ICMPv6 packet such as Neighbor Solicitation (NS) packet or Multicast Listener Discovery (MLD) packet that passed main filtering and the field placed in the Target IP address of a Neighbor Solicitation (NS) packet (9'th byte to 24'th byte of the ICMPv6 header) also matches a valid *IPV6AT* filter.
ARP	11	0b	ARP Request Packet Proxy Enable. If set to 1b forward to Management for proxying on match of any ARP request packet that passed main filtering.
Reserved	14:12	0x0	Reserved. Write 0, ignore on read.
NoTCO	15	0b	Ignore TCO/management packets for proxying. 0b - Ignore only TCO/management packets for Proxying that meet the criteria defined in the *MNGONLY* register (I.e. are intended only for the BMC and not the Host). 1b - Ignore any TCO/management packets for Proxying, even if in normal operation it's forwarded to the Host in addition to the BMC.
FLX0	16	0b	Flexible Filter 0 Enable.
FLX1	17	0b	Flexible Filter 1 Enable.
FLX2	18	0b	Flexible Filter 2 Enable.
FLX3	19	0b	Flexible Filter 3 Enable.
FLX4	20	0b	Flexible Filter 4 Enable.
FLX5	21	0b	Flexible Filter 5 Enable.
FLX6	22	0b	Flexible Filter 6 Enable.
FLX7	23	0b	Flexible Filter 7 Enable.
Reserved	31:24	0x0	Reserved. Write 0, ignore on read.

1. If the *RCTL.UPE* is set, and the EX bit is set also, any unicast packet is sent to Management for proxying.

8.20.7 Proxying Status Register - PROXYS (0x5F64; R/W1C)

This register is used to record statistics about all Proxying packets received. If a packet matches multiple criteria then multiple bits could be set. Writing a 1b to any bit clears that bit.

This register is not cleared when RST# is asserted. It is only cleared when LAN_PWR_GOOD is de-asserted or when cleared by the software device driver.

Note: If additional packets are received that matches one of the wakeup filters, after the original wakeup packet is received, the *PROXYS* register is updated with the matching filters accordingly.

Field	Bit(s)	Initial Value	Description
Reserved	1:0	0x0	Reserved. Write 0, ignore on read.
EX	2	0b	Directed Exact Packet Received The packet's address matched one of the 32 pre-programmed exact values in the Receive Address registers, the packet was a unicast packet and *RCTL.UPE* is set to 1b or the packet was a unicast packet hashed to a value that corresponded to a 1 bit in the Unicast Table Array (*UTA*).
MC	3	0b	Directed Multicast Packet Received The packet was a multicast packet hashed to a value that corresponded to a 1 bit in the Multicast Table Array or the packet was a multicast packet and *RCTL.MPE* is set to 1b.
BC	4	0b	Broadcast Packet Received.
ARP Directed	5	0b	ARP Request Packet with IP4AT filter match Received. When set to 1b indicates a match on any ARP request packet that passed main filtering and Target IP address also matches one of the valid *IP4AT* filters.
IPv4	6	0b	Directed IPv4 Packet Received.
IPv6	7	0b	Directed IPv6 Packet Received.
Reserved	8	0b	Reserved. Write 0, ignore on read.
NS	9	0b	IPV6 Neighbor Solicitation Received. When set to 1b indicates a match on NS packet that passed main filtering.
NS Directed	10	0b	IPV6 Neighbor Solicitation with directed DA filter match received. When set to 1b indicates a match on NS packet and Target IP address also matches a valid *IPV6AT* filter.
ARP	11	0b	ARP Request Packet Received. When set to 1b indicates a match on any ARP request packet that passed main filtering.
Reserved	15:12	0x0	Reserved. Write 0, ignore on read.
FLX0[1]	16	0b	Flexible Filter 0 Match.
FLX1[1]	17	0b	Flexible Filter 1 Match.
FLX2[1]	18	0b	Flexible Filter 2 Match.
FLX3[1]	19	0b	Flexible Filter 3 Match.
FLX4[1]	20	0b	Flexible Filter 4 Match.
FLX5[1]	21	0b	Flexible Filter 5 Match.

Field	Bit(s)	Initial Value	Description
FLX6[1]	22	0b	Flexible Filter 6 Match.
FLX7[1]	23	0b	Flexible Filter 7 Match.
Reserved	31:24	0b	Reserved. Write 0, ignore on read.

1. Bit is set only when flex filter match is detected and *WUFC.FLEX_HQ* is 0.

8.20.8 IP Address Valid - IPAV (0x5838; R/W)

The IP Address Valid indicates whether the IP addresses in the IP Address Table are valid.

Field	Bit(s)	Initial Value	Description
V40	0	0b	IPv4 Address 0 Valid.
V41	1	0b	IPv4 Address 1 Valid.
V42	2	0b	IPv4 Address 2 Valid.
V43	3	0b	IPv4 Address 3 Valid.
Reserved	15:4	0x0	Reserved. Write 0, ignore on read.
V60	16	0b	IPv6 Address 0 Valid.
Reserved	31:17	0b	Reserved. Write 0, ignore on read.

8.20.9 IPv4 Address Table - IP4AT (0x5840 + 8*n [n=0...3]; R/W)

The IPv4 Address Table is used to store the four IPv4 addresses for the ARP/IPv4 Request packet and Directed IP packet wakeup.

Field	Bit(s)	Initial Value	Description
IP Address	31:0	X	IPv4 Address n Note: These registers are written in Big Endian order (LS byte is first on the wire and is the MS byte of the IPV4 Address).

Field	Dword #	Address	Bit(s)	Initial Value	Description
IPV4ADDR0	0	0x5840	31:0	X	IPv4 Address 0
IPV4ADDR1	2	0x5848	31:0	X	IPv4 Address 1
IPV4ADDR2	4	0x5850	31:0	X	IPv4 Address 2
IPV4ADDR3	6	0x5858	31:0	X	IPv4 Address 3

8.20.10 IPv6 Address Table - IP6AT (0x5880 + 4*n [n=0...3]; R/W)

The IPv6 Address Table is used to store the IPv6 addresses for neighbor Discovery packet filtering and Directed IP packet wakeup.

Field	Bit(s)	Initial Value	Description
IP Address	31:0	X	IPv6 Address bytes 4*n+1:4*n +4 Note: These registers appear in Big Endian order (LS byte, LS address is first on the wire and is the MS byte of the IPV6 Address).

Field	Dword #	Address	Bit(s)	Initial Value	Description
IPV6ADDR0	0	0x5880	31:0	X	IPv6 Address 0, bytes 1-4
	1	0x5884	31:0	X	IPv6 Address 0, bytes 5-8
	2	0x5888	31:0	X	IPv6 Address 0, bytes 9-12
	3	0x588C	31:0	X	IPv6 Address 0, bytes 16-13

8.20.11 Flexible Host Filter Table Registers - FHFT (0x9000 - 0x93FC; RW)

Each of the 4 Flexible Host Filters Table registers (FHFT) contains a 128 byte pattern and a corresponding 128-bit mask array. If enabled, the first 128 bytes of the received packet are compared against the non-masked bytes in the FHFT register.

Each 128 byte filter is composed of 32 DW entries, where each 2 DWs are accompanied by an 8-bit mask, one bit per filter byte. When a bit in the 8-bit mask field is set the corresponding Byte in the filter is compared.

The 8 LSB bits of the last DW of each filter contains a length field defining the number of bytes from the beginning of the packet compared by this filter, the length field should be 8 bytes aligned value. If actual packet length is less than (length - 8) (length is the value specified by the length field), the filter fails. Otherwise, it depends on the result of actual byte comparison. The value should not be greater than 128.

Note: The length field must be 8 bytes aligned. For filtering packets shorter than 8 bytes aligned the values should be rounded up to the next 8 bytes aligned value, the hardware implementation compares 8 bytes at a time so it should get extra zero masks (if needed) until the end of the length value.

Bits 31:8 of the last DW of each filter also includes a Queueing field (refer to Section 8.20.11.1). When the I350 is in D0 state, the *WUFC.FLEX_HQ* bit is set to 1, *MRQC.Multiple Receive Queues* = 010b or 000b and the packet matches the flex filter, the Queueing field defines the receive queue for the packet, priority of the filter and actions to be initiated.

31 0	31 8	7 0	31 0	31 0
Reserved	Reserved	Mask [7:0]	DW 1	DW 0

Reserved	Reserved	Mask [15:8]	DW 3	DW 2
Reserved	Reserved	Mask [23:16]	DW 5	DW 4
Reserved	Reserved	Mask [31:24]	DW 7	DW 6

....

31 8	7 0	31 8	7 0	31 0	31 0
Reserved	Reserved	Reserved	Mask [119:112]	DW 29	DW 28
Queueing	Length	Reserved	Mask [127:120]	DW 31	DW 30

Accessing the *FHFT* registers during filter operation can result in a packet being mis-classified if the write operation collides with packet reception. It is therefore advised that the flex filters are disabled prior to changing their setup.

8.20.11.1 Flex Filter Queueing Field

Queueing field resides in bits 31:8 of last DW (DW 63) of flex filter. The queueing field defines the receive queue to forward the packet (*RQUEUE*), the filter priority (*FLEX_PRIO*) and additional filter actions. Operations defined in queueing field are enabled when the I350 is in D0 state, *MRQC.Multiple Receive Queues* = 010b or 000b, *WUFC.FLEX_HQ* is 1 and relevant *WUFC.FLX[n]* bit is set.

Field	Bit(s)	Initial Value	Description
Length	7:0	X	Length Filter Length in bytes. Should be 8 bytes aligned and not greater then 128 bytes.
RQUEUE	10:8	X	Receive Queue Defines receive queue associated with this Flex filter. When match occurs in D0 state, packet is forwarded to the receive queue.
Reserved	15:11	X	Reserved Write 0, ignore on read.
FLEX_PRIO	18:16	X	Flex filter Priority Defines the priority of the filter assuming two filters with same priority don't match. If two filters with the same priority match the incoming packet, the first filter (lowest address) is used in order to define the queue destination of this packet.
Reserved	23:19	X	Reserved Write 0, ignore on read.
Immediate Interrupt	24	X	Enables issuing an immediate interrupt when the Flex filter matches incoming packet.
Reserved	31:25	X	Reserved Write 0, ignore on read.

8.20.11.2 Flex Filter 0 - Example

Field	Dword	Address	Bit(s)	Initial Value
Filter 0 DW0	0	0x9000	31:0	X
Filter 0 DW1	1	0x9004	31:0	X
Filter 0 Mask[7:0]	2	0x9008	7:0	X
Reserved	3	0x900C	31:0	X
Filter 0 DW2	4	0x9010	31:0	X
…				
Filter 0 DW30	60	0x90F0	31:0	X
Filter 0 DW31	61	0x90F4	31:0	X
Filter 0 Mask[127:120]	62	0x90F8	7:0	X
Length	63	0x90FC	7:0	X
Filter 0 Queueing	63	0x90FC	31:8	X

8.20.12 Flexible Host Filter Table Extended Registers - FHFT_EXT (0x9A00 - 0x9DFC; RW)

Each of the 4 additional Flexible Host Filters Table extended registers (FHFT_EXT) contains a 128 Byte pattern and a corresponding 128-bit mask array. If enabled, the first 128 Bytes of the received packet are compared against the non-masked bytes in the FHFT_EXT register. The layout and access rules of this table are the same as in FHFT.

8.21 Management Register Descriptions

All management registers are controlled by the remote BMC for both read and write. Host accesses to the management registers are blocked for write. The attributes for the fields in this chapter refer to the BMC access rights.

Note: All the registers described in this section can get their default values from EEPROM when manageability pass through works in legacy SMBus mode. The only exception being the MANC register where part of the bits are masked. The specific MANC bits that can be loaded from EEPROM are indicated in the register description.

8.21.1 Management VLAN TAG Value - MAVTV (0x5010 +4*n [n=0...7]; RW)

Where “n” is the VLAN filter serial number, equal to 0,1,…7.

Field	Bit(s)	Initial Value	Description
VID	11:0	0x0	Contains the VLAN ID that should be compared with the incoming packet if the corresponding bit in *MDEF* is set.
Reserved	31:12	0x0	Reserved Write 0, ignore on read

The *MAVTV* registers are written by the BMC and are not accessible to the host for writing. The registers are used to filter manageability packets as described in the Management chapter.

8.21.2 Management Flex UDP/TCP Ports - MFUTP (0x5030 + 4*n [n=0...3]; RW)

Where each 32-bit register (n=0,...,3) refers to two UDP/TCP port filters (register at address offset n=0 refers to UDP/TCP ports 0 and 1, register at address offset n=1 refers to UDP/TCP ports 2 and 3, etc.).

Field	Bit(s)	Initial Value	Description
MFUTP_even	15:0	0x0	2*n Management Flex UDP/TCP port
MFUTP_odd	31:16	0x0	2*n+1 Management Flex UDP/TCP port

The *MFUTP* registers are written by the BMC and are not accessible to the host for writing. The registers are used to filter manageability packets. See section 10.3 .

Reset - The *MFUTP* registers are cleared on LAN_PWR_GOOD only. The initial values for this register can be loaded from the EEPROM after power-up reset.

Note: The *MFUTP_even* and *MFUTP_odd* fields should be written in network order.

8.21.3 Management Ethernet Type Filters- METF (0x5060 + 4*n [n=0...3]; RW)

Field	Bit(s)	Initial Value	Description
METF	15:0	0x0	EtherType value to be compared against the L2 EtherType field in the Rx packet.
Reserved	29:16	0x0	Reserved Write 0, ignore on read.
Polarity	30	0b	0b = Positive filter - forward packets matching this filter to the manageability block. 1b = Negative filter - block packets matching this filter from the manageability block.
Reserved	31	0b	Reserved Write 0, ignore on read.

The *METF* registers are written by the BMC and are not accessible to the host for writing. The registers are used to filter manageability packets. See section 10.3 .

Reset - The *METF* registers are cleared on LAN_PWR_GOOD only. The initial values for this register might be loaded from the EEPROM after power-up reset.

8.21.4 Management Control Register - MANC (0x5820; RW)

The *MANC* register can be written by the BMC and is not accessible to the host for writing.

Field	Bit(s)	Initial Value	Description
Reserved	13:0	0x0	Reserved. Write 0, ignore on read.
FW_RESET (R/W1C)	14	0b	FW Reset occurred. Set to 1b on a TCO Firmware Reset. Cleared by write 1b.
TCO_Isolate (RO)	15	0b	Set to 1 on a TCO Isolate command. When the "TCO_Isolate" bit is set. Host write cycles are completed successfully on the PCIe but silently ignored by internal logic. Note that when FW initiates the TCO Isolate command it also initiates a FW interrupt via the *ICR.MNG* bit to the host and writes a value of 0x22 to the *FWSM.Ext_Err_Ind* field. This bit is Read Only and mirrors the value of the Isolate bit in the internal Management registers.
TCO_RESET (R/W1C)	16	0b	TCO Reset occurred. Set to 1b on a TCO Reset, to reset LAN port by BMC. Cleared by write 1b.
RCV_TCO_EN	17	0b[1]	TCO Receive Traffic Enabled. When bit is set receive traffic to the manageability is enabled. This bit should be set only if either the MANC.EN_BMC2OS or MANC.EN_BMC2NET bits are set.
KEEP_PHY_LINK_UP	18	0b[1]	Block PHY reset and power state changes. When this bit is set the PHY reset and power state changes do not effect the PHY, This bit can not be written to unless the *Keep_PHY_Link_Up_En* EEPROM bit is set.
Reserved	22:2019	0b	Reserved. Write 0 ignore on read.
EN_XSUM_FILTER	23	0b[1]	Enable checksum filtering to MNG When this bit is set, only packets that pass L3, L4 checksum are sent to the MNG block.
EN_IPv4_FILTER	24	0b[1]	Enable IPv4 address Filters – when set, the last 128 bits of the MIPAF register are used to store 4 IPv4 addresses for IPv4 filtering. When cleared, these bits store a single IPv6 filter.
FIXED_NET_TYPE	25	0b[1]	Fixed net type: If set, only packets matching the net type defined by the NET_TYPE field passes to manageability. Otherwise, both tagged and un-tagged packets can be forwarded to the manageability engine.

Field	Bit(s)	Initial Value	Description
NET_TYPE	26	0b[1]	NET TYPE: 0b = pass only un-tagged packets. 1b = pass only VLAN tagged packets. Valid only if FIXED_NET_TYPE is set.
Reserved	27	0b	Reserved Write 0 ignore on read.
EN_BMC2OS (RO)	28	0b[1]	Enable BMC to OS and OS to BMC traffic 0 = The BMC can not communicate with the OS. 1 = The BMC can communicate with the OS. When cleared the BMC traffic is not forwarded to the OS, even if the Host MAC address filter and VLANs (RAH/L, MTA, VFTA and VLVF registers) indicate that it should. When cleared the OS traffic is not forwarded to the BMC even if the decision filters indicates it should. This bit does not impact the BMC to Network traffic. ***Note:*** 1. Initial value loaded according to value of *Port n traffic types* field in EEPROM (refer to Section 6.3.12.2)
EN_BMC2NET (RO)	29	1b[1]	Enable BMC to network and network to BMC traffic 0 = The BMC can not communicate with the network. 1 = The BMC can communicate with the network. When cleared the BMC traffic is not forwarded to the network and the network traffic is not forwarded to the BMC even if the decision filters indicates it should. This bit does not impact the host to BMC traffic. ***Note:*** 1. Initial value loaded according to value of *Port n traffic types* field in EEPROM (refer to Section 6.3.12.2)
MPROXYE (RO)	30	0b[1]	Management Proxying Enable When set to 1b Proxying of packets is enabled when device is in D3 low power state. 0 = Manageability does not support Proxying. 1 = Manageability supports Proxying. ***Note:*** Proxy information and requirements is passed by Software driver to Firmware via the shared RAM Host interface (Refer to Section 10.8, Section 8.22 and Section 10.8.2.4.2). ***Note:*** Proxying traffic from and to Firmware is not affected by the *MANC.RCV_TCO_EN* bit or the MANC.EN_BMC2NET bit.
Reserved	31	0b	Reserved Write 0 ignore on read.

1. Bit loaded from EEPROMin legacy SMBus mode.

8.21.5 Management Only Traffic Register - MNGONLY (0x5864; RW)

The MNGONLY register allows exclusive filtering of certain type of traffic to the BMC. Exclusive filtering enables the BMC to define certain packets that are forwarded to the BMC but not to the host. The packets will not be forwarded to the host even if they pass the host L2 filtering process.

Each manageability decision filter (*MDEF* and *MDEF_EXT*) has a corresponding bit in the *MNGONLY* register. When a manageability decision filter (*MDEF* and *MDEF_EXT*) forwards a packet to manageability, it may also block the packet from being forwarded to the host if the corresponding *MNGONLY* bit is set.

Field	Bit(s)	Initial Value[1]	Description
Exclusive to MNG	7:0	0x0	Exclusive to MNG – when set, indicates that packets forwarded by the manageability filters to manageability are not sent to the host. Bits 0...7 correspond to decision rules defined in registers *MDEF*[0...7] and *MDEF_EXT*[0...7].
Reserved	31:8	0x0	Reserved Write 0, ignore on read.

1. The initial values for this register can be loaded from the EEPROM after power-up reset or firmware reset.

8.21.6 Manageability Decision Filters- MDEF (0x5890 + 4*n [n=0...7]; RW)

Where "n" is the decision filter

Field	Bit(s)	Initial Value[1]	Description
Exact AND	3:0	0x0	Exact - Controls the inclusion of Exact MAC address 0 to 3 In the manageability filter decision (AND section). Bit 0 corresponds to exact MAC address 0 (MMAL0 and MMAH0), etc.
Broadcast AND	4	0b	Broadcast - Controls the inclusion of broadcast address filtering in the manageability filter decision (AND section).
VLAN AND	12:5	0x0	VLAN - Controls the inclusion of VLAN tag 0 to 7 respectively In the manageability filter decision (AND section). Bit 5 corresponds to VLAN tag 0, etc.
IPv4 Address	16:13	0x0	IPv4 Address - Controls the inclusion of IPV4 address 0 to 3 respectively in the manageability filter decision (AND section). Bit 13 corresponds to IPV4 address 0, etc. ***Notes:*** 1. This field is relevant only if *MANC.EN_IPv4_FILTER* is set. 2. Supported only for Network traffic.
IPv6 Address	20:17	0b	IPv6 Address - Controls the inclusion of IPV6 address 0 to 3 respectively in the manageability filter decision (AND section). Bit 17 corresponds to IPV6 address 0, etc ***Notes:*** 1. Bit 20 is relevant only if *MANC.EN_IPv4_FILTER* is cleared. 2. Supported only for Network traffic.
Exact OR	24:21	0x0	Exact - Controls the inclusion of exact MAC address 0 to 3 In the manageability filter decision (OR section). Bit 21 corresponds to exact MAC address 0 (MMAL0 and MMAH0), etc.
Broadcast OR	25	0b	Broadcast - Controls the inclusion of broadcast address filtering in the manageability filter decision (OR section).
Multicast AND	26	0b	Multicast - Controls the inclusion of Multicast address filtering in the manageability filter decision (AND section). Broadcast packets are not included by this bit.
ARP Request	27	0b	ARP Request - Controls the inclusion of ARP Request filtering in the manageability filter decision (OR section). ***Note:*** Supported only for Network traffic.
ARP Response	28	0b	ARP Response - Controls the inclusion of ARP Response filtering in the manageability filter decision (OR section). ***Note:*** Supported only for Network traffic.

Field	Bit(s)	Initial Value[1]	Description
Neighbor Discovery	29	0b	neighbor Discovery - Controls the inclusion of neighbor Discovery filtering in the manageability filter decision (OR section). ***Notes:*** 1. Supported only for Network traffic. 2. Neighbor Discovery types supported are: 0x86 (134d) - Router Advertisement 0x87 (135d) - Neighbor Solicitation 0x88 (136d) - Neighbor Advertisement 0x89 (137d) - Redirect
Port 0x298	30	0b	Port 0x298 - Controls the inclusion of Port 0x298 filtering in the manageability filter decision (OR section). ***Note:*** Supported only for Network traffic.
Port 0x26F	31	0b	Port 0x26F - Controls the inclusion of Port 0x26F filtering in the manageability filter decision (OR section). ***Note:*** Supported only for Network traffic.

1. Default values are read from EEPROM.

8.21.7 Manageability Decision Filters- MDEF_EXT (0x5930 + 4*n[n=0...7]; RW)

Field	Bit(s)	Initial Value[1]	Description
L2 EtherType AND	3:0	0x0	L2 EtherType - Controls the inclusion of L2 EtherType filtering in the manageability filter decision (AND section). ***Note:*** Supported only for Network traffic.
Reserved	7:4	0x0	Reserved for additional L2 EtherType AND filters. Write 0, ignore on read.
L2 EtherType OR	11:8	0x0	L2 EtherType - Controls the inclusion of L2 EtherType filtering in the manageability filter decision (OR section). ***Note:*** Supported only for Network traffic.
Reserved	15:12	0x0	Reserved for additional L2 EtherType OR filters.
Flex port	23:16	0x0	Flex port - Controls the inclusion of Flex port filtering in the manageability filter decision (OR section). Bit 16 corresponds to flex port 0, etc. ***Note:*** Supported only for Network traffic.
Flex TCO	24	0b	Flex TCO - Controls the inclusion of Flex TCO filtering in the manageability filter decision (OR section). Bit 24 corresponds to Flex TCO filter 0. ***Note:*** Supported only for Network traffic.
Reserved	27:25	0x0	Reserved Write 0, ignore on read.
NC-SI Discard	28	1b	0 = Apply filtering rules to packets with NC-SI EtherType. 1 = Discard packets with NC-SI EtherType. ***Notes:*** 1. NC-SI EtherType is 0x88F8 2. Supported only for Network traffic.

Field	Bit(s)	Initial Value[1]	Description
Flow Control Discard	29	1b	0 = Apply filtering rules to packets with Flow Control EtherType. 1 = Discard packets with Flow Control EtherType. ***Notes:*** 1. Flow Control EtherType is 0x8808 2. Supported only for Network traffic.
Apply_to_network_traffic	30	0b	0= Do not apply this decision filter to traffic received from the network. 1= Apply this decision filter to traffic received from the network.
Apply_to_host_traffic	31	0b	0 = This decision filter does not apply to traffic received from the host. 1 = This decision filter applies to traffic received from the host.

1. Default values are read from EEPROM.

8.21.8 Manageability IP Address Filter - MIPAF (0x58B0 + 4*n [n=0...15]; RW)

The Manageability IP Address Filter register stores IP addresses for manageability filtering. The *MIPAF* register can be used in two configurations, depending on the value of the *MANC.EN_IPv4_FILTER* bit:

- *EN_IPv4_FILTER* = 0: the last 128 bits of the register store a single IPv6 address (*IPV6ADDR3*)
- *EN_IPv4_FILTER* = 1: the last 128 bits of the register store 4 IPv4 addresses (*IPV4ADDR[3:0]*)

MANC.EN_IPv4_FILTER = 0:

DWORD#	Address	31	0
0	0x58B0	IPV6ADDR0	
1	0x58B4		
2	0x58B8		
3	0x58BC		
4	0x58C0	IPV6ADDR1	
5	0x58C4		
6	0x58C8		
7	0x58CC		
8	0x58D0	IPV6ADDR2	
9	0x58D4		
10	0x58D8		
11	0x58DC		
12	0x58E0	IPV6ADDR3	
13	0x58E4		
14	0x58E8		
15	0x58EC		

Field definitions for 0 setting:

Field	Dword #	Address	Bit(s)	Initial Value	Description
IPV6ADDR0	0	0x58B0	31:0	X*	IPv6 Address 0, bytes 1-4 (LS byte is first on the wire)
	1	0x58B4	31:0	X*	IPv6 Address 0, bytes 5-8
	2	0x58B8	31:0	X*	IPv6 Address 0, bytes 9-12
	3	0x58BC	31:0	X*	IPv6 Address 0, bytes 13-16
IPV6ADDR1	0	0x58C0	31:0	X*	IPv6 Address 1, bytes 1-4 (LS byte is first on the wire)
	1	0x58C4	31:0	X*	IPv6 Address 1, bytes 5-8
	2	0x58C8	31:0	X*	IPv6 Address 1, bytes 9-12
	3	0x58CC	31:0	X*	IPv6 Address 1, bytes 13-16
IPV6ADDR2	0	0x58D0	31:0	X*	IPv6 Address 2, bytes 1-4 (LS byte is first on the wire)
	1	0x58D4	31:0	X*	IPv6 Address 2, bytes 5-8
	2	0x58D8	31:0	X*	IPv6 Address 2, bytes 9-12
	3	0x58DC	31:0	X*	IPv6 Address 2, bytes 13-16
IPV6ADDR3	0	0x58E0	31:0	X*	IPv6 Address 3, bytes 1-4 (LS byte is first on the wire)
	1	0x58E4	31:0	X*	IPv6 Address 3, bytes 5-8
	2	0x58E8	31:0	X*	IPv6 Address 3, bytes 9-12
	3	0x58EC	31:0	X*	IPv6 Address 3, bytes 13-16

MANC.EN_IPv4_FILTER = 1:

DWORD#	Address	31	0
0	0x58B0	IPV6ADDR0	
1	0x58B4		
2	0x58B8		
3	0x58BC		
4	0x58C0	IPV6ADDR1	
5	0x58C4		
6	0x58C8		
7	0x58CC		
8	0x58D0	IPV6ADDR2	
9	0x58D4		
10	0x58D8		
11	0x58DC		
12	0x58E0	IPV4ADDR0	
13	0x58E4	IPV4ADDR1	
14	0x58E8	IPV4ADDR2	
15	0x58EC	IPV4ADDR3	

Field definitions for 1 Setting:

Field	Dword #	Address	Bit(s)	Initial Value[1]	Description
IPV6ADDR0	0	0x58B0	31:0	X	IPv6 Address 0, bytes 1-4 (LS byte is first on the wire)
	1	0x58B4	31:0	X	IPv6 Address 0, bytes 5-8
	2	0x58B8	31:0	X	IPv6 Address 0, bytes 9-12
	3	0x58BC	31:0	X	IPv6 Address 0, bytes 16-13
IPV6ADDR1	0	0x58C0	31:0	X	IPv6 Address 1, bytes 1-4 (LS byte is first on the wire)
	1	0x58C4	31:0	X	IPv6 Address 1, bytes 5-8
	2	0x58C8	31:0	X	IPv6 Address 1, bytes 9-12
	3	0x58CC	31:0	X	IPv6 Address 1, bytes 16-13
IPV6ADDR2	0	0x58D0	31:0	X	IPv6 Address 2, bytes 1-4 (LS byte is first on the wire)
	1	0x58D4	31:0	X	IPv6 Address 2, bytes 5-8
	2	0x58D8	31:0	X	IPv6 Address 2, bytes 9-12
	3	0x58DC	31:0	X	IPv6 Address 2, bytes 16-13
IPV4ADDR0	0	0x58E0	31:0	X	IPv4 Address 0 (LS byte is first on the wire)
IPV4ADDR1	1	0x58E4	31:0	X	IPv4 Address 1 (LS byte is first on the wire)
IPV4ADDR2	2	0x58E8	31:0	X	IPv4 Address 2 (LS byte is first on the wire)
IPV4ADDR3	3	0x58EC	31:0	X	IPv4 Address 3 (LS byte is first on the wire)

1. The initial values for these registers can be loaded from the EEPROM after power-up reset. The registers are written by the BMC and not accessible to the host for writing.

Initial value:

Field	Bit(s)	Initial Value[1]	Description
IP_ADDR 4 bytes	31:0	X	4 bytes of IP (v6 or v4) address. i mod 4 = 0 to bytes 1 – 4 i mod 4 = 1 to bytes 5 – 8 i mod 4 = 0 to bytes 9 – 12 i mod 4 = 0 to bytes 13 – 16 where i div 4 is the index of IP address (0...3)

1. The initial values for these registers can be loaded from the EEPROM after power-up reset. The registers are written by the BMC and not accessible to the host for writing.

Reset - The registers are cleared on LAN_PWR_GOOD only.

Note: These registers should be written in network order.

8.21.9 Manageability MAC Address Low - MMAL (0x5910 + 8*n [n= 0…3]; RW)

Where “n” is the exact unicast/Multicast address entry, equal to 0…3.

Field	Bit(s)	Initial Value[1]	Description
MMAL	31:0	X	Manageability MAC Address Low. The lower 32 bits of the 48 bit Ethernet address.

1. The initial values for these registers can be loaded from the EEPROM after power-up reset. The registers are written by the BMC and not accessible to the host for writing.

These registers contain the lower bits of the 48 bit Ethernet address. The *MMAL* registers are written by the BMC and are not accessible to the host for writing. The registers are used to filter manageability packets. See section 10.3 .

Reset - The *MMAL* registers are cleared on LAN_PWR_GOOD only. The initial values for this register can be loaded from the EEPROM after power-up reset.

Note: The *MMAL.MMAL* field should be written in network order.

8.21.10 Manageability MAC Address High - MMAH (0x5914 + 8*n [n=0…3]; RW)

Where "n" is the exact unicast/Multicast address entry, equal to 0…3.

Field	Bit(s)	Initial Value[1]	Description
MMAH	15:0	X	Manageability MAC Address High. The upper 16 bits of the 48 bit Ethernet address.
Reserved	31:16	0x0	Reserved. Write 0, ignore on read.

1. The initial values for these registers can be loaded from the EEPROM after power-up reset. The registers are written by the BMC and not accessible to the host for writing.

These registers contain the upper bits of the 48 bit Ethernet address. The complete address is {*MMAH, MMAL*}. The MMAH registers are written by the BMC and are not accessible to the host for writing. The registers are used to filter manageability packets. See section 10.3 .

Reset - The *MMAL* registers are cleared on LAN_PWR_GOOD only. The initial values for this register can be loaded from the EEPROM after power-up reset or firmware reset.

Note: The *MMAH.MMAH* field should be written in network order.

8.21.11 Flexible TCO Filter Table registers - FTFT (0x9400 - 0x94FC; RW)

The Flexible TCO Filter Table registers (*FTFT*) contains a 128 byte pattern and a corresponding 128-bit mask array. If enabled, the first 128 bytes of the received packet are compared against the non-masked bytes in the FTFT register.

The 128 byte filter is composed of 32 DW entries, where each 2 DWs are accompanied by an 8-bit mask, one bit per filter byte. The bytes in each 2 DWs are written in network order i.e. byte0 written to bits [7:0], byte1 to bits [15:8] etc. The mask field is set so that bit0 in the mask masks byte0, bit 1 masks byte 1 etc. A value of 1 in the mask field means that the appropriate byte in the filter should be compared to the appropriate byte in the incoming packet.

Note: The mask field must be 8 bytes aligned even if the length field is not 8 bytes aligned, as the hardware implementation compares 8 bytes at a time so it should get extra masks until the end of the next quad word. Any mask bit that is located after the length should be set to 0 indicating no comparison should be done.

In case the actual length which is defined by the length field register and the mask bits is not 8 bytes aligned there may be a case that a packet which is shorter then the actual required length passes the flexible filter. This may occur due to comparison of up to 7 bytes that come after the packet, but are not a real part of the packet.

The last DW of the filter contains a length field defining the number of bytes from the beginning of the packet compared by this filter. If actual packet length is less than length specified by this field, the filter fails. Otherwise, it depends on the result of actual byte comparison. The value should not be greater than 128.

The initial values for the *FTFT* registers can be loaded from the EEPROM after power-up reset. The *FTFT* registers are written by the BMC and are not accessible to the host for writing. The registers are used to filter manageability packets as described in Section 10.3.3.5.

Note: The *FTFT* registers are cleared on LAN_PWR_GOOD and Firmware reset only.

31 0	31 8	7 0	31 0	31 0
Reserved	Reserved	Mask [7:0]	DW 1	DW 0
Reserved	Reserved	Mask [15:8]	DW 3	DW 2
Reserved	Reserved	Mask [23:16]	DW 5	DW 4
Reserved	Reserved	Mask [31:24]	DW 7	DW 6

....

31 8	7 0	31 8	7 0	31 0	31 0
Reserved	Reserved	Reserved	Mask [127:120]	DW 29	DW 28
Reserved	Length	Reserved	Mask [127:120]	DW 31	DW 30

Field definitions for Filter Table Registers:

Field	Dword	Address	Bit(s)	Initial Value
Filter 0 DW0	0	0x9400	31:0	X
Filter 0 DW1	1	0x9404	31:0	X
Filter 0 Mask[7:0]	2	0x9408	7:0	X
Reserved	3	0x940C		X
Filter 0 DW2	4	0x9410	31:0	X
…				

Field	Dword	Address	Bit(s)	Initial Value
Filter 0 DW30	60	0x94F0	31:0	X
Filter 0 DW31	61	0x94F4	31:0	X
Filter 0 Mask[127:120]	62	0x94F8	7:0	X
Length	63	0x94FC	7:0	X

8.22 Management-Host Interface Register Descriptions

The device driver of functions 0,1,2 and 3 communicates with the MMS block through CSR access. The manageability block is mapped to address 0x8800 -0x8FFF on the slave bus of each function.

8.22.1 Host Slave Command Interface to MMS

This interface is used by the device driver for several of commands and for delivering various types of data structure in both directions (MNG →Host, Host → MNG).

The address space is separated into two areas:

Direct access to the internal Management data ram: The internal DATA RAM is mapped to address 0x8800-0x8EFF. Writing to this address space goes directly to the RAM.

Control registers located at address 0x8F00.

8.22.1.1 Host Slave Command I/F flow.

This interface is used for the external HOST software to access the MMS sub-system. The HOST software can write a command block or read data structure directly from the data ram. The HOST software controls these transactions through a slave access to the control register.

The flow below describes the process of initiating a command to the MMS:

1. The device driver takes ownership of the *SW_FW_SYNC.SW_MNG_SM* bit according to the flow described in Section 4.7.1.
2. The device driver reads the *HICR* register and checks that the enable bit is set.
3. The device driver writes the relevant command block into the shared ram area.
4. The device driver sets the "command" bit in the control register. Setting this bit causes an interrupt to the Management.
5. The device driver polls the control register until the command bit is cleared by hardware.
6. When the MMS is done with the command it clears the command bit (if the MMS should reply with a data, it should clear the bit only after the data is in the ram area where the device driver can read it).
7. If the device driver reads control register and the "SV" bit is set, it means that there is a valid status of the last command in the RAM. If the "SV is not set it means that the command has failed with no status in the RAM.

8.22.1.2 HOST Interface Control Register - HICR (0x8F00; RW)

Field	Bit(s)	Initial Value	Description
En (RO)	0	0b	Enable. When set it indicates that a RAM area is provided for device driver accesses. This bit is read only for the device driver.
C	1	0b	Command. The device driver sets this bit when it has finished putting a command block in the Management internal data ram. This bit should be cleared by the firmware after the command's processing has been completed.
SV (RO)	2	0b	Status Valid. Indicates that there is a valid status in CSR area that the device driver can read. 1b – status valid. 0b – status not valid. The value of the bit is valid only when the C bit is cleared. Only the device driver reads this bit.
Reserved	31:3	0x0	Reserved Write 0, ignore on read.

8.22.2 General Manageability HOST CSR Registers

Field	Bit(s)	Initial Value	Description
FWSW (RO)	15:0	0x0	Firmware Status word This bit is read only through the HOST interface.
Reserved	30:16	0x0	Reserved Write 0, ignore on read.
FWRI (R/W1C)	31	1b	Firmware Reset Indication. Set when firmware reset is asserted. Cleared when HOST writes 1b to it. Writing 0b to this bit does not change its value. Note: This bit is also set after LAN_POWER_GOOD.

8.23 Memory Error Registers Description

Main internal memories are protected by error correcting code (ECC) or parity bits. The I350 contains several registers that enable and report detection of internal memory errors. Description and usage of these registers can be found in Section 7.6.

8.23.1 Parity Error Indication- PEIND (0x1084; RC)

Field	Bit(s)	Initial Value	Description
lanport_parity_fatal_ind (RC)	0	0b	Fatal Error detected in LAN port Memory. Bit is latched high and cleared on read.
mng_parity_fatal_ind (RC)	1	0b	Fatal Error detected in Management Memory. Bit is latched high and cleared on read.
pcie_parity_fatal_ind (RC)	2	0b	Fatal Error detected in PCIe Memory. Bit is latched high and cleared on read.
dma_parity_fatal_ind (RC)	3	0b	Fatal Error detected in DMA Memory Bit is latched high and cleared on read.
Reserved	31:4	0x0	Reserved Write 0 ignore on read.

8.23.2 Parity and ECC Indication Mask – PEINDM (0x1088; RW)

Field	Bit(s)	Initial Value	Description
lanport_parity_fatal_ind	0	1b	When set and *PEIND.lanport_parity_fatal_ind* is set, enable interrupt generation by setting the ICR.FER bit.
mng_parity_fatal_ind	1	1b	When set and *PEIND.mng_parity_fatal_ind* is set, enable interrupt generation by setting the *ICR.FER bit.*
pcie_parity_fatal_ind	2	1b	When set and *PEIND.pcie_parity_fatal_ind* is set, enable interrupt generation by setting the *ICR.FER bit.*
dma_parity_fatal_ind	3	1b	When set and *PEIND.dma_parity_fatal_ind* is set, enable interrupt generation by setting the *ICR.FER bit.*
Reserved	31:4	0x0	Reserved Write 0 ignore on read.

8.23.3 DMA Transmit Parity and ECC Control - DTPARC (0x3F00; RW)

Field	Bit(s)	Initial Value	Description
ram_dtx_lso_ecc_en	0	1b	ram_dtx_lso ECC check enable.
Reserved	1	0b	Reserved. Write 0, ignore on read.
ram_dtx_cntxt_ecc_en	2	1b	ram_dtx_cntxt ECC check enable.
Reserved	3	0b	Reserved. Write 0, ignore on read.

Field	Bit(s)	Initial Value	Description
Reserved	4:5	10b	Reserved. Write 0, ignore on read.
ram_dtx_temp_ecc_en	6	1b	ram_dtx_temp ECC check enable.
Reserved	7	0b	Reserved. Write 0, ignore on read.
ram_dtx_cmd_ecc_en	8	1b	ram_dtx_cmd ECC check enable.
Reserved	9	0b	Reserved. Write 0, ignore on read.
ram_dtx_dhrf_par_en	10	1b	ram_dtx_dhrf parity check enable.
Reserved	11	0b	Reserved. Write 0, ignore on read.
ram_dtx_icache_ecc_en	12	1b	ram_dtx_icache ECC check enable.
Reserved	31:13	0x0	Reserved. Write 0, ignore on read.

8.23.4 DMA Transmit Parity and ECC Status- DTPARS (0x3F10; R/W1C)

Field	Bit(s)	Initial Value	Description
ecc_ind_dtx_lso	0	0b	dtx_lso memory ECC error indication Indicates detection of correctable ECC error in ram if *DTPARC.ram_dtx_lso_ecc_en* is set.
ecc_ind_dtx_cntxt	1	0b	dtx_cntxt memory ECC error indication Indicates detection of correctable ECC error in ram if *DTPARC.ram_dtx_cntxt_ecc_en* is set.
Reserved	2	0b	Reserved
ecc_ind_dtx_temp	3	0b	dtx_temp memory ECC error indication Indicates detection of correctable ECC error in ram if *DTPARC.ram_dtx_temp_ecc_en* is set.
ecc_ind_dtx_cmd	4	0b	dtx_cmd memory ECC error indication Indicates detection of correctable ECC error in ram if *DTPARC.ram_dtx_cmd_ecc_en* is set.

Field	Bit(s)	Initial Value	Description
par_ind_dtx_dhrf	5	0b	dtx_dhrf memory parity error indication Indicates detection of parity error in ram if *DTPARC.ram_dtx_dhrf_par_en* is set. When set stops all TX activity from the port. To recover from this condition Software Driver should issue a SW reset by asserting *CTRL.RST* and re-initializing the port. Will cause assertion of *PEIND.dma_parity_fatal_ind* and *ICR.FER* interrupt if bits are not masked. ***Note:***
ecc_ind_dtx_icache	6	0b	dtx_icache memory ECC error indication Indicates detection of correctable ECC error in ram if *DTPARC.ram_dtx_icache_ecc_en* is set.
Reserved	31:7	0x0	Reserved. Write 0, ignore on read.

8.23.5 DMA Receive Parity and ECC Control - DRPARC (0x3F04; RW)

Field	Bit(s)	Initial Value	Description
ram_drx_desc_ecc_en	0	1b	ram_drx_desc ECC check enable.
Reserved	1	0b	Reserved. Write 0, ignore on read.
ram_drx_dhrf_par_en	2	1b	ram_drx_dhrf parity check enable.
Reserved	3	0b	Reserved. Write 0, ignore on read.
ram_drx_icache_ecc_en	4	1b	ram_drx_icache ECC check enable. r
Reserved	5	0b	Reserved. Write 0, ignore on read.
ram_drx_srrctl_ecc_en	6	1b	ram_drx_srrctl ECC check enable.
Reserved	31:7	0x0	Reserved. Write 0, ignore on read.

8.23.6 DMA Receive Parity and ECC Status - DRPARS (0x3F14; R/W1C)

Field	Bit(s)	Initial Value	Description
ecc_ind_drx_desc	0	0b	drx_desc memory ECC error indication. Indicates detection of correctable ECC error in ram if *DRPARC.ram_drx_desc_ecc_en* is set.
par_ind_drx_dhrf	1	0b	drx_dhrf memory parity error indication. Indicates detection of parity error in ram if *DRPARC.ram_drx_dhrf_par_en* is set. When set stops all DMA RX activity from the port. To recover from this condition Software Driver should issue a SW reset by asserting *CTRL.RST* and re-initializing the port. Will cause assertion of *PEIND.dma_parity_fatal_ind* and *ICR.FER* interrupt if bits are not masked. ***Note:***
ecc_ind_drx_icache	2	0b	drx_icache memory ECC error indication. Indicates detection of correctable ECC error in ram if *DRPARC.ram_drx_icache_ecc_en* is set.
ecc_ind_drx_srrctl	3	0b	drx_srrctl memory ECC error indication. Indicates detection of correctable ECC error in ram if *DRPARC.ram_drx_srrctl_ecc_en* is set.
Reserved	31:4	0x0	Reserved. Write 0, ignore on read.

8.23.7 Dhost ECC Control - DDECCC (0x3F08; RW)

Field	Bit(s)	Initial Value	Description
ram_dhost_tx_data_comp_ecc_en	0	1b	ram_dhost_tx_data_comp ECC check enable.
Reserved	1	0b	Reserved. Write 0, ignore on read.
ram_dhost_tx_desc_comp_ecc_en	2	1b	ram_dhost_tx_desc_comp ECC check enable.
Reserved	3	0b	Reserved. Write 0, ignore on read.
ram_dhost_rx_desc_comp_ecc_en	4	1b	ram_dhost_rx_desc_comp ECC check enable.
Reserved	5	0b	Reserved. Write 0, ignore on read.
ram_dstat_ecc_en	6	1b	ram_dstat ECC check enable.

Field	Bit(s)	Initial Value	Description
Reserved	31:7	0x0	Reserved. Write 0, ignore on read.

8.23.8 Dhost ECC Status - DDECCS (0x3F18; R/W1C)

Field	Bit(s)	Initial Value	Description
ecc_ind_dhost_tx_data_comp	0	0b	dhost_tx_data_comp memory correctable ECC error indication. Indicates detection of correctable ECC error in ram if *DDECCC.ram_dhost_tx_data_comp_ecc_en* is set. ***Note:*** Bit cleared by write 1b.
ecc_ind_dhost_tx_desc_comp	1	0b	dhost_tx_desc_comp memory correctable ECC error indication. Indicates detection of correctable error in ram if *DDECCC.ram_dhost_tx_desc_comp_ecc_en* is set. ***Note:*** Bit cleared by write 1b.
ecc_ind_dhost_rx_desc	2	0b	dhost_rx_desc_comp memory correctable ECC error indication. Indicates detection of correctable error in ram if *DDECCC.ram_dhost_rx_desc_comp_ecc_en* is set. ***Note:*** Bit cleared by write 1b.
ecc_ind_dstat	3	0b	dstat memory correctable ECC error indication. Indicates detection of correctable ECC error in ram if *DDECCC.ram_dstat_ecc_en* is set. ***Note:*** Bit cleared by write 1b.
Reserved	31:4	0x0	Reserved. Write 0, ignore on read.

8.23.9 Rx Packet Buffer ECC Status - RPBECCSTS (0x245C; RW)

Field	Bit(s)	Initial Value	Description
Reserved	15:0	0x0	Reserved Write 0, ignore on read.
RX PB ECC Enable (RW)	16	1b	ECC Enable for RX Packet Buffer.
LPB ECC Enable (RW)	17	1b	ECC Enable for Loopback Packet Buffer
Reserved	25:18	0x0	Reserved Write 0, ignore on read.
Pb_cor_err_sta (R/W1C)	26	0b	RX Packet Buffer Correctable Error indication ***Note:*** Bit is cleared by write 1b.

Field	Bit(s)	Initial Value	Description
Reserved	27	0b	Reserved Write 0, ignore on read.
lpb_cor_err_sta (R/W1C)	28	0b	Loopback Packet Buffer Correctable Error indication ***Note:*** Bit is cleared by write 1b.
Reserved	31:29	0x0	Reserved Write 0, ignore on read.

8.23.10 Tx Packet Buffer ECC Status - TPBECCSTS (0x345C; RW)

Field	Bit(s)	Initial Value	Description
Reserved	15:0	0x0	Reserved Write 0, ignore on read.
TX PB ECC Enable	16	1b	ECC Enable for TX Packet Buffer.
MPB ECC Enable	17	1b	ECC Enable for Management TX Packet Buffer.
Reserved	25:18	0x0	Reserved Write 0, ignore on read.
Pb_cor_err_sta (R/W1C)	26	0b	TX Packet Buffer Correctable Error indication Bit is cleared by write 1b.
Reserved	27	0b	Reserved Write 0, ignore on read.
mpb_cor_err_sta (R/W1C)	28	0b	Management TX Packet Buffer Correctable Error indication Bit is cleared by write 1b.
Reserved	31:29	0x0	Reserved Write 0, ignore on read.

8.23.11 PCIe Parity Control Register - PCIEERRCTL (0x5BA0; RW)

Field	Bit(s)	Initial Value	Description
GPAR_EN	0	0b[1]	Global Parity Enable When this bit is cleared parity checking of all RAMs is disabled.
Reserved	3:1	0x0	Reserved Write 0, ignore on read.
ERR EN RX ICLUT ERR	4	1b	RX ICLUT ERR parity check Enable
Reserved	5	0b	Reserved Write 0, ignore on read
ERR EN RX CDQ 0	6	1b	RX CDQ 0 parity check Enable

Field	Bit(s)	Initial Value	Description
Reserved	7	0b	Reserved Write 0, ignore on read
ERR EN RX CDQ 1	8	1b	RX CDQ 1 parity check Enable
Reserved	9	0b	Reserved Write 0, ignore on read
ERR EN RX CDQ 2	10	1b	RX CDQ 2 parity check Enable
Reserved	11	0b	Reserved Write 0, ignore on read
ERR EN RX CDQ 3	12	1b	RX CDQ 3 parity check Enable
Reserved	31:13	0x0	Reserved Write 0, Ignore on read.

1. Bit loaded from EEPROM.

8.23.12 PCIe Parity Status Register - PCIEERRSTS (0x5BA8; R/W1C)

Register logs uncorrectable parity errors detected in PCIe logic.

Field	Bit(s)	Initial Value	Description
Reserved	1:0	0x0	Reserved Write 0, ignore on read.
PAR ERR RX ICLUT ERR	2	0b	RX ICLUT ERR Parity Error
PAR ERR RX CDQ 0	3	0b	RX CDQ 0 Parity Error Indicates detection of parity error in RAM if *PCIEERRCTL.ERR EN RX CDQ 0* is set. When set stops all PCIe and DMA RX and TX activity from the function. To recover from this condition Software Driver should issue a software reset by asserting *CTRL.RST* and re-initializing the port (refer to Section 7.6.1.1). Will cause assertion of *PEIND.pcie_parity_fatal_ind* and *ICR.FER* interrupt if bits are not masked.
PAR ERR RX CDQ 1	4	0b	RX CDQ 1 Parity Error Indicates detection of parity error in RAM if *PCIEERRCTL.ERR EN RX CDQ 1*is set. When set stops all PCIe and DMA RX and TX activity from the function. To recover from this condition Software Driver should issue a software reset by asserting *CTRL.RST* and re-initializing the port (refer to Section 7.6.1.1). Will cause assertion of *PEIND.pcie_parity_fatal_ind* and *ICR.FER* interrupt if bits are not masked.

Field	Bit(s)	Initial Value	Description
PAR ERR RX CDQ 2	5	0b	RX CDQ 2 Parity Error Indicates detection of parity error in RAM if *PCIEERRCTL.ERR EN RX CDQ 2* is set. When set stops all PCIe and DMA RX and TX activity from the function. To recover from this condition Software Driver should issue a software reset by asserting *CTRL.RST* and re-initializing the port (refer to Section 7.6.1.1). Will cause assertion of *PEIND.pcie_parity_fatal_ind* and *ICR.FER* interrupt if bits are not masked.
PAR ERR RX CDQ 3	6	0b	RX CDQ 3 Parity Error Indicates detection of parity error in RAM if *PCIEERRCTL.ERR EN RX CDQ 3* is set. When set stops all PCIe and DMA RX and TX activity from the function. To recover from this condition Software Driver should issue a software reset by asserting *CTRL.RST* and re-initializing the port (refer to Section 7.6.1.1). Will cause assertion of *PEIND.pcie_parity_fatal_ind* and *ICR.FER* interrupt if bits are not masked.
Reserved	31:7	0x0	Reserved Write 0, ignore on read.

8.23.13 PCIe ECC Control Register - PCIEECCCTL (0x5BA4; RW)

Field	Bit(s)	Initial Value	Description
ERR EN TX WR CMD	0	1b	TX Write Request Command ECC ERR Enable
Reserved	3:1	0x0	Reserved Write 0, Ignore on read
ERR EN TX RD CMD	4	1b	TX Read Request Command ECC ERR Enable
Reserved	7:5	0x0	Reserved Write 0, Ignore on read.
ERR EN MSIX	8	1b	MSIX ECC check Enable
Reserved	9	0b	Reserved Write 0, ignore on read
ERR EN RX PNP	10	1b	RX PNP ECC check Enable
Reserved	11	0b	Reserved Write 0, ignore on read
ERR EN TX WR DATA	12	1b	TX Write Request Data ECC check Enable
Reserved	13	0b	Reserved Write 0, ignore on read
ERR EN RETRY BUF	14	1b	TX Retry Buffer ECC check Enable
Reserved	31:15	0x0	Reserved Write 0, Ignore on read.

8.23.14 PCIe ECC Status Register - PCIEECCSTS (0x5BAC; R/W1C)

Field	Bit(s)	Initial Value	Description
ECC ERR TX WR CMD	0	0b	TX Write Request Command Correctable ECC Error
ECC ERR TX RD CMD	1	0b	TX Read Request Command Correctable ECC Error
ECC ERR MSIx	2	0b	MSIx ECC Correctable Error
ECC ERR RX PNP	3	0b	RX PNP ECC Error
ECC ERR TX WR DATA	4	0b	TX Write Request Data ECC Correctable Error
ECC ERR RETRY BUF	5	0b	TX Retry Buffer ECC Correctable Error
Reserved	31:6	0x0	Reserved Write 0, ignore on read

8.23.15 LAN Port Parity Error Control Register - LANPERRCTL (0x5F54; RW)

Field	Bit(s)	Initial Value	Description
Reserved	0	0b	Reserved. Write 0, ignore on read.
mrx_flx_en	8:1	0xFF	Enable mrx_flx parity error indication When set to 0xFF enables Flex filter memory parity error detection and indication.
retx_buf_en	9	1b	Enable retx_buf parity error indication When set to 1b enables RETX buffer (retransmit buffer) parity error detection and indication.
stat_regs_en	10	1b	Enable stat_regs parity error indication When set to 1b enables statistics memory parity error detection and indication.
f_uta_en	11	1b	Enable f_uta parity error indication When set to 1b enable unicast filter table parity error detection and indication.
f_rss_en	12	1b	Enable f_rss parity error indication When set to 1b enables RSS memory parity error detection and indication.
f_mulc _en	13	1b	Enable f_mulc parity error indication When set to 1b enables multicast filter table parity error detection and indication.
f_vlan_en	14	1b	Enable f_vlan parity error indication When set to 1b enables VLAN filter table parity error detection and indication.
vf_mailbox_en	15	1b	Enable vf mailbox parity error indication When set to 1b enables VF mailbox parity error detection and indication.
Reserved	31:16	0x0	Reserved. Write 0, ignore on read.

8.23.16 LAN Port Parity Error Status Register - LANPERRSTS (0x5F58; R/W1C)

Field	Bit(s)	Initial Value	Description
Reserved	0	0b	Reserved. Write 0, ignore on read.
mrx_flx	8:1	0x0	mrx_flx parity error indication When set to 1b indicates detection of parity error in Flex filter ram if respective *LANPERRCTL.mrx_flx_en* bit is set. When set disables packet reception. To recover from this condition Software Driver should issue a SW reset by asserting *CTRL.RST* and re-initializing the port. Will cause assertion of *PEIND.lanport_parity_fatal_ind* and *ICR.FER* interrupt if bits are not masked.
retx_buf	9	0b	retx_buf parity error indication When set to 1b indicates detection of parity error in RETX buffer (retransmit buffer) ram if *LANPERRCTL.retx_buf_en* is set. When set disables packet transmission. To recover from this condition Software Driver should issue a SW reset by asserting *CTRL.RST* and re-initializing the port. Will cause assertion of *PEIND.lanport_parity_fatal_ind* and *ICR.FER* interrupt if bits are not masked.
stat_regs	10	0b	stat_regs parity error indication When set to 1b indicates detection of parity error in statistics ram if *LANPERRCTL.stat_regs_en* bit is set. To recover from this condition Software Driver should discard any statists read and clear the bit. Will cause assertion of *PEIND.lanport_parity_fatal_ind* and *ICR.FER* interrupt if bits are not masked.
f_uta	11	0b	f_uta parity error indication When set to 1b indicates detection of parity error in Unicast Filter Table ram if *LANPERRCTL.f_uta_en* bit is set. When set disables packet reception. To recover from this condition Software Driver should issue a SW reset by asserting *CTRL.RST* and re-initializing the port. Will cause assertion of *PEIND.lanport_parity_fatal_ind* and *ICR.FER* interrupt if bits are not masked.
f_rss	12	0b	f_rss parity error indication When set to 1b indicates detection of parity error in RSS ram if *LANPERRCTL.f_rss_en* bit is set. When set disables packet reception. To recover from this condition Software Driver should issue a SW reset by asserting *CTRL.RST* and re-initializing the port. Will cause assertion of *PEIND.lanport_parity_fatal_ind* and *ICR.FER* interrupt if bits are not masked.
f_mulc	13	0b	f_mulc parity error indication Indicates detection of parity error in Multicast Filter Table ram if *LANPERRCTL.f_mulc_en* bit is set. When set disables packet reception. To recover from this condition Software Driver should issue a SW reset by asserting *CTRL.RST* and re-initializing the port. Will cause assertion of *PEIND.lanport_parity_fatal_ind* and *ICR.FER* interrupt if bits are not masked.
f_vlan	14	0b	f_vlan parity error indication When set to 1b indicates detection of parity error in VLAN filter table ram if *LANPERRCTL.f_vlan_en* bit is set. When set disables packet reception. To recover from this condition Software Driver should issue a SW reset by asserting *CTRL.RST* and re-initializing the port. Will cause assertion of *PEIND.lanport_parity_fatal_ind* and *ICR.FER* interrupt if bits are not masked.

Field	Bit(s)	Initial Value	Description
vf_mailbox	15	0b	vf mailbox parity error indication When set to 1b indicates detection of parity error in vf_mailbox ram if LANPERRCTL.ram_vf_mailbox_en bit is set. To recover from this condition Software Driver should discard data read and clear bit. Will cause assertion of PEIND.lanport_parity_fatal_ind and ICR.FER interrupt if bits are not masked.
Reserved	31:16	0x0	Reserved Write 0, ignore on read.

8.24 Power Management Register Description

Following registers enable control of various power saving features.

8.24.1 DMA Coalescing Control Register - DMACR (0x2508; R/W)

Field	Bit(s)	Initial Value	Description
DMACWT	13:0	0x20	DMA Coalescing Watchdog Timer When in DMA coalescing value in *DMACR.DMACWT* counter sets the upper limit in 32 µsec units between arrival of receive packet, request to transmit or an interrupt cause and move out of DMA coalescing. ***Note:*** If value is 0x0 condition to move out of DMA coalescing as a result of watchdog timer expiration is disabled.
DC_LPBKW_EN	14	1b	DMA Coalescing VM to VM Loopback watchdog enable. When set to 1b, VM to VM loopback traffic activate DMA Coalescing Watchdog timer (*DMACR.DMACWT*). ***Note:*** If DMA Coalescing watchdog timer is disabled and bit is 1b any VM to VM loopback traffic causes move out of DMA coalescing state.
DC_BMC2OSW_EN	15	1b	DMA Coalescing BMC to OS watchdog enable. When set to 1b, BMC to OS traffic activate DMA Coalescing Watchdog timer (*DMACR.DMACWT*). ***Note:*** If DMA Coalescing watchdog timer is disabled and bit is 1b any BMC to OS traffic causes move out of DMA coalescing state.
DMACTHR	23:16	0x0	DMA Coalescing Receive Threshold This value defines the DMA coalescing Receive threshold in 1 Kilobyte units. When amount of data in internal receive buffer exceeds *DMACTHR* value, DMA coalescing is stopped and PCIe moves to L0 state. ***Notes:*** 1. Value should be lower than *FCRTC.RTH_Coal* threshold value, to avoid needless generation of flow control packets when in DMA coalescing operating mode and flow control is enabled. 2. Receive threshold size should be smaller than internal receive buffer area reported in *IRPBS.RXPbsize* field. 3. If value is 0x0, condition to move out of DMA coalescing as a result of passing DMA Coalescing Receive Threshold is disabled. 4. Value programmed should be greater than Maximum packet size.
Reserved	26:24	0x0	Reserved Write 0 ignore on read.

Field	Bit(s)	Initial Value	Description
Reserved	24	0b	Reserved Write 0, ignore on read.
EXIT_DC (SC)	25	0b	Exit DMA coalescing Software can initiate a one time move out of DMA coalescing state by setting bit to 1b.
Reserved	27:26	0x0	Reserved Write 0, ignore on read.
DMAC_Lx	29:28	11b	Move to Lx low power link state when no PCIe transactions detected. Entry to Lx low power state occurs following detection of link idle state for a duration that exceeds time defined in the *DMCTLX.TTLX* field. 00b – Stay in L0. 01b – Move to L0s when no PCIe transactions 10b - Move to L1 when no PCIe transactions 11b - Move to deepest possible Lx state (L0s or L1). ***Notes:*** 1. When DMA coalescing is enabled (*DMACR.DMAC_EN* = 1) value of field should be 10b or 11b. 2. Field enables move into PCIe link low power ASPM state (L1 or L0s) even when DMA coalescing is disabled (*DMACR.DMAC_EN* = 0). 3. When ASPM L0s is enabled in the PCIe configuration *Link Control Register* (refer to Section 9.5.6.8) the I350 will transition to a L0s low power link state after detecting a link idle state for a duration that does not exceed 7μS (as defined in the *Latency_To_Enter_L0s* EEPROM field), in compliance with the PCIe specification, even if the value programmed to the *DMACR.DMAC_Lx* field does not specify entry to L0s and time defined in the *DMCTLX.TTLX* field has not expired. 4. The I350 will transition into the specified Lx low power state when PCIe link is idle only if entry to the Lx state is enabled in bits 1:0 of the PCIe configuration *Link Control Register* (refer to Section 9.5.6.8).
Reserved	30	0b	Reserved Write 0, ignore on read.
DMAC_EN	31	0b	DMA Coalescing Enable 0 - Disable DMA Coalescing 1 - Enable DMA Coalescing

8.24.2 DMA Coalescing Transmit Threshold - DMCTXTH (0x3550;RW)

Field	Bit(s)	Initial Value	Description
DMCTTHR	11:0	0xE4	DMA Coalescing Transmit Threshold This value defines the DMA coalescing transmit threshold in 64 byte units. When amount of empty space in internal transmit buffer exceeds *DMCTTHR* value and additional transmit data is available in main memory, DMA coalescing is stopped and PCIe moves to L0 state. ***Notes:*** 1. If value is 0x0 or smaller than maximum transmit packet size as defined in the *DTXMXPKTSZ.MAX_TPKT_SIZE* field condition to move out of DMA Coalescing due to passing the *DMA Coalescing Transmit Threshold* level is disabled. 2. Transmit threshold size should be smaller than Internal Transmit Buffer area reported in *ITPBS.TXPbsize* field.
Reserved	31:12	0b	Reserved Write 0, ignore on read.

8.24.3 DMA Coalescing Time to Lx Request - DMCTLX (0x2514;RW)

This CSR is common to all functions.

Field	Bit(s)	Initial Value	Description
TTLX	11:0	0x20	Time to LX request: Controls the time between detection of low power Link condition to actual request to move into Lx (L0s or L1) low power link state (as defined in the *DMACR.DMAC_Lx* field) and entry into DMA Coalescing state. Timer counts is in 1µsec intervals. ***Notes:*** 1. Timer adds delay to decision on when to enter DMA Coalescing state, flush any pending descriptor writeback operations, flush any pending interrupts and when to move PCIe link into low power Lx state as a function of the value placed in the *DMCTLX.DC_FLUSH* register bit. 2. When DMA coalescing is disabled the *DMCTLX.TTLX* value delays entry into the PCIe low power Lx state, as defined in the *DMACR.DMAC_Lx* field, after all conditions to enter PCIe low power ASPM link state exist.
Reserved	29:12	0x0	Reserved Write 0, ignore on read.
DC_FLUSH	30	0b	DMA Coalescing Flush Defines when pending descriptor write-backs and Interrupt flush occur relative to *DMCTLX.TTLX* timer expiration. 0b - Flush occurs on start of *DMCTLX.TTLX* count and entry into DMA coalescing state and move of PCIe link into low power Lx state occurs on *DMCTLX.TTLX* expiration. 1b - Flush occurs on first expiration of *DMCTLX.TTLX* timer. Entry into DMA coalescing mode and move of PCIe link into low power Lx state occurs on second *DMCTLX.TTLX* expiration. ***Notes:*** 1. When DMA coalescing is disabled (DMACR.DMAC_EN = 0) bit does not affect decision on entry into PCIe low power Lx state. 2. Functionality enabled by clearing the DMCTLX.*DCFLUSH_DIS* bit to 0b.

DCFLUSH_DIS	31	0b	Disable DMA Coalescing Flush When bit is set, Flush of pending Interrupts and Pending descriptor Writeback operations before entry into DMA Coalescing (refer to Section 5.8.2.1), is disabled.

8.24.4 DMA Coalescing Receive Packet Rate Threshold - DMCRTRH (0x5DD0;RW)

Field	Bit(s)	Initial Value	Description
UTRESH	18:0	0x0	Receive Traffic Threshold-size Defines the minimum RX packet rate to activate DMA coalescing in 64 Byte chunks of data. RX packet rate below this value will not allow entry into DMA coalescing. Refer to Section 5.8 for additional information. ***Notes:*** 1. Threshold is measured in 64 Byte chunks of data received during a time window defined by the port link speed (10Mbps, 100Mbps or 1Gbps), *SCCRL.INTERVAL* field and the time units defined in the *SCBI* register. 2. When amount of RX bytes received in the previous time window is below the value specified in the *DMCRTRH.UTRESH* field then DMA coalescing will not be entered in current time window. 3. When value of *UTRESH* field is 0x0, packet rate is not considered as a condition to enter or exit DMA coalescing state. 4. RX packet rate calculation includes both Network traffic to host and Host VF to VF traffic.
RSVD	30:19	0b	Reserved Write 0, ignore on read.
LRPRPW (RO)	31	0b	Low Receive packet rate in previous window 0b - Packet rate above *DMCRTRH.UTRESH* threshold detected. 1b - Packet rate below *DMCRTRH.UTRESH* threshold detected.

8.24.5 DMA Coalescing Current RX Count - DMCCNT (0x5DD4;RO)

Field	Bit(s)	Initial Value	Description
CCOUNT	24:0	0x0	DMA Coalescing Receive Traffic Current Count: Represents the count of receive traffic in the current time interval in units of 64-byte segments. Refer to Section 5.8 for additional information. ***Note:*** Counter does not wrap around.
RSVD	31:25	0x0	Reserved. Write 0, ignore on read.

8.24.6 Flow Control Receive Threshold Coalescing - FCRTC (0x2170; R/W)

Field	Bit(s)	Initial Value	Description
Reserved	3:0	0x0	Reserved Write 0 ignore on read.
RTH_Coal	17:4	0x0	Flow control Receive Threshold High watermark value used to generate XOFF flow control packet when executing DMA coalescing, internal transmit fifo is empty and Transmit Flow control is enabled (*CTRL.TFCE* = 1b). When previous conditions exist a XOFF packet is sent if the occupied space in the RX packet buffer is more or equal to this watermark. This field is in 16 bytes granularity. Refer to Section 3.7.5.3.1 for calculation of *FCRTC.RTH_Coal* value. ***Notes:*** 1. To avoid sending XOFF flow control packets needlessly when executing DMA Coalescing and internal transmit buffer is empty, value should be higher than threshold defined in *DMACR.DMACTHR* field. Maximum threshold value can be up to *FCRTH0.RTH* + maximum allowable packet size * 1.25. 2. *RTH_Coal* threshold value is used as watermark for sending flow control packets when DMA Coalescing is enabled and internal transmit buffer is empty. 3. Value programmed should be greater than maximum packet size.
Reserved	31:18	0x0	Reserved Write 0 ignore on read.

8.24.7 Latency Tolerance Reporting (LTR) Minimum Values - LTRMINV (0x5BB0; R/W)

Field	Bit(s)	Initial Value	Description
LTRV	9:0	0x5	Latency Tolerance Value Field indicates latency tolerance supported when conditions for minimum latency tolerance exist (Refer to Section 5.9.2.1). *LTRV* values are multiplied by 32,768ns or 1,024ns depending on the Scale field, to indicate latency tolerance supported in nanoseconds. A value of 0 indicates that the device will be impacted by any delay and that best possible service is requested. The I350 reports the same value for both Snoop and No Snoop requirements. If no memory latency requirement exists for either Snoop or No Snoop accesses the appropriate Requirement bit is cleared. Note: Software should subtract time required to move from L1 to L0 from LTR value.
Scale	12:10	011b	Latency Scale This field provides a scale for the value contained within the *LTRMINV.LTRV* field. Encoding: 010 – *LTRV* value times 1,024ns 011 – *LTRV* value times 32,768ns Others - Reserved
Reserved	14:13	0x0	Reserved Write 0 ignore on read.
LSNP Requirement	15	0b	LTR Snoop requirement 0 - No Latency requirements in Snoop memory access. 1 - Latency tolerance in Snoop memory access specified in *LTRMINV.LTRV* field.

Field	Bit(s)	Initial Value	Description
Reserved	30:16	0x0	Reserved Write 0 ignore on read.
LNSNP Requirement	31	0b	LTR Non-Snoop Requirement 0 - No Latency requirements in Non-Snoop memory access. 1 - Latency tolerance in Non-Snoop memory access specified in *LTRMINV.LTRV* field.

8.24.8 Latency Tolerance Reporting (LTR) Maximum Values - LTRMAXV (0x5BB4; R/W)

Field	Bit(s)	Initial Value	Description
LTRV	9:0	0x5	Latency Tolerance Value Field indicates latency tolerance supported when conditions for maximum latency tolerance exist (Refer to Section 5.9.2.2). *LTRV* values are multiplied by 32,768ns or 1,024ns depending on the Scale field, to indicate latency tolerance supported in nanoseconds. A value of 0 indicates that the device will be impacted by any delay and that best possible service is requested. The I350 reports the same value for both Snoop and No Snoop requirements. If no memory latency requirement exists for either Snoop or No Snoop accesses the appropriate Requirement bit is cleared. Note: Software should subtract time required to move from L1 to L0 from LTR value.
Scale	12:10	011b	Latency Scale This field provides a scale for the value contained within the *LTRMAXV.LTRV* field. Encoding: 010 – *LTRV* value times 1,024ns 011 – *LTRV* value times 32,768ns Others - Reserved
Reserved	14:13	0x0	Reserved Write 0 ignore on read.
LSNP Requirement	15	0b	LTR Snoop requirement 0 - No Latency requirements in Snoop memory access. 1 - Latency tolerance in Snoop memory access specified in *LTRMAXV.LTRV* field.
Reserved	30:16	0x0	Reserved Write 0 ignore on read.
LNSNP Requirement	31	0b	LTR Non-Snoop Requirement 0 - No Latency requirements in Non-Snoop memory access. 1 - Latency tolerance in Non-Snoop memory access specified in *LTRMAXV.LTRV* field.

8.24.9 Latency Tolerance Reporting (LTR) Control - LTRC (0x01A0; R/W)

Field	Bit(s)	Initial Value	Description
Reserved	0	0b	Reserved Write 0, ignore on read
LTR_MIN	1	0b	LTR Send Minimum Values When set to 1 the I350 sends a PCIe LTR message with the LTR Snoop value, LTR No-snoop value and LTR requirement bits as defined in the *LTRMINV* register. ***Notes:*** 1. To resend a LTR message with the minimum value defined in the *LTRMINV* register, bit should be cleared and set again. 2. LTR_MIN and LTR_MAX bits are exclusive. 3. A new PCIe LTR message will be sent only if the last PCIe LTR message sent had a latency tolerance value different then the value specified in the *LTRMINV* register
LTR_MAX	2	0b	LTR Send Maximum Values When set to 1 the I350 sends a PCIe LTR message with the LTR Snoop value, LTR No-snoop value and LTR requirement bits as defined in the *LTRMAXV* register. ***Notes:*** 1. To resend a LTR message with the maximum value defined in the *LTRMAXV* register, bit should be cleared and set again. 2. LTR_MIN and LTR_MAX bits are exclusive. 3. A new PCIe LTR message will be sent only if the last PCIe LTR message sent had a latency tolerance value different then the value specified in the *LTRMAXV* register
PDLS_EN	3	1b	Port Disable LTR send enable 0 - Do not issue PCIe LTR message with requirement bits cleared on port disable (RX and TX disabled). 1 - Issue PCIe LTR message with requirement bits cleared on port disable (RX and TX disabled).
LNKDLS_EN	4	1b	Link Disconnect LTR send enable 0 - Do not issue PCIe LTR message with requirement bits cleared on link disconnect. 1 - Issue PCIe LTR message with requirement bits cleared on link disconnect.
EEEMS_EN	5	0b	EEE LPI LTR Max send enable When bit is set and Link is in RX EEE LPI (Low Power Idle) state the I350 sends a PCIe LTR message with the LTR Snoop value, LTR No-snoop value and LTR requirement bits as defined in the LTRMAXV register. For further information, refer to Section 5.9. 0 - Do not issue PCIe LTR messages with LTRMAXV value as a result of RX link entering EEE LPI state. 1- Issue PCIe LTR messages with LTRMAXV value as a result of RX link entering EEE LPI state. ***Note:*** Bit is reset to 0 by Hardware following link disconnect to allow SW to re-negotiate Tw_system time and update LTRMAXV value.
Reserved	31:6	0x0	Reserved Write 0 ignore on read.

8.24.10 Energy Efficient Ethernet (EEE) Register - EEER (0x0E30; R/W)

Field	Bit(s)	Initial Value	Description
Tw_system	15:0	0x0	Time expressed in microseconds that no data will be transmitted following move from EEE TX LPI link state to Link Active state. Field holds the Transmit Tw_sys_tx value negotiated during EEE LLDP negotiation. Notes: 1. If value is lower than minimum Tw_sys_tx value defined in IEEE802.3az clause 78.5 (30 µsec for 100BASE-TX and 16.5 µsec for 1000BASE-T) then interval where no data is transmitted following move out of EEE TX LPI state defaults to minimum Tw_sys_tx. 2. Following link disconnect or Auto-negotiation value of this field returns to default value, until SW re-negotiates new tw_sys_tx value via EEE LLDP. ***Note:*** When transmitting flow control frames the I350 waits the minimum time defined in the IEEE802.3az standard before transmitting the flow control packet. The I350 does not wait the Tw_system time following exit of LPI before transmitting the flow control frame.
TX_LPI_EN	16	0b[1]	Enable entry into EEE LPI on TX path 0b - Disable entry into EEE LPI on TX path. 1b - Enable entry into EEE LPI on TX path. Refer to Section 3.7.7.1 for additional information on EEE TX LPI entry. ***Notes:*** 1. Even when *TX_LPI_EN* is 1b the I350 will not enable entry into TX LPI state for at least 1 second following the change of link_status to OK as defined in IEEE802.3az clause 78.1.2.1. 2. Even if the *TX_LPI_EN* bit is set, the I350 will initiate entry into TX EEE LPI link state only if EEE support at the link speed was negotiated during Auto-negotiation.
RX_LPI_EN	17	1b	Enable entry into EEE LPI on RX path 0b - Disable entry into EEE LPI on RX path. 1b - Enable entry into EEE LPI on RX path. ***Notes:*** 1. Even if the *RX_LPI_EN* bit is set, the I350 will recognize entry into RX EEE LPI link state only if EEE support at the link speed was negotiated during Auto-negotiation. 2. When set and link moves into RX LPI, a LTR message with the value defined in the *LTRMAXV* register is sent on the PCIe, if *LTRC.EEEMS_EN* is set.
LPI_FC	18	1b	Enable EEE TX LPI entry on Flow control Enable EEE TX LPI state entry when link partner sent a PAUSE Flow control frame, even if internal Transmit buffer is not empty, transmit descriptors are available or management traffic is pending. ***Notes:*** 1. The I350 enters TX LPI state when no data is transmitted and not in mid-packet. 2. Entry into TX LPI on flow control is enabled only if either *EEER.TX_LPI_EN* is set to 1b or *EEER.Force_TLPI* is set to 1b. 3. Reception of XON frame causes move out of LPI if transmit is pending.
Force_TLPI	19	0b	Force TX LPI When set PHY is forced into EEE TX LPI state if there is no TX management traffic. ***Notes:*** 1. The I350 enters TX LPI state when no data is transmitted and not in mid-packet. 2. When set the I350 will enter TX LPI even if *EEER.TX_LPI_EN* is 0b.

Field	Bit(s)	Initial Value	Description
Reserved	27:20	0x0	Reserved Write 0 ignore on read.
EEE_FRC_AN	28	0b	Force EEE Auto-negotiation When bit is set to 1 enables EEE operation in internal MAC logic even if link partner does not support EEE. Should be set to 1b to enable testing of EEE operation via MAC loopback (refer to Section 3.7.6.2).
EEE NEG (RO)	29	X	EEE support Negotiated on link 0b - EEE operation not supported on link. 1b - EEE operation supported on link.
RX LPI Status (RO)	30	X	RX Link in LPI state 0b - RX in Active state 1b - RX in LPI state
TX LPI Status (RO)	31	X	TX Link in LPI state 0b - TX in Active state 1b - TX in LPI state

1. Loaded from EEPROM.

8.25 Thermal Sensor Registers Description

Some of the thermal registers are read only and some of them are read write. Most values are loaded from the EEPROM but the BMC or Host may modify the settings.

The Thermal sensor registers are common to all functions. Before accessing any of the registers defined in this section Firmware or Software should take ownership of thermal Sensor semaphore bits (*SW_FW_SYNC.SW_PWRTS_SM* for software and *SW_FW_SYNC.FW_PWRTS_SM* for Firmware) according to the flow defined in Section 4.7.1 and release ownership of the Thermal sensor semaphore bits according to the flow defined in Section 4.7.2.

8.25.1 Thermal Sensor Measured Junction Temperature - THMJT (0x8100; RO)

Field	Bit(s)	Initial Value	Description
Tj	8:0	X	Measured Junction temperature Measured junction temperature with 1°C resolution represented in 2's complement format. ***Note:*** Thermal Sensor precision +/-5°C.
Reserved	30:9	0x0	Reserved Write 0 ignore on read.
TH Valid	31	X	Thermal Sensor Valid When bit is set to 1b, thermal temperature indicated in the *THMJT.Tj* field is valid. ***Note:*** In Diagnostic operating mode when the *THMJT.Tj* temperature value is forced via the THDIAG register the TH Valid bit is set to 1b.

8.25.2 Thermal Sensor Low Threshold Control - THLOWTC (0x8104; RW)

Field	Bit(s)	Initial Value	Description
Threshold	8:0	0x6E (110 °C)	Low Junction Temperature threshold Low Junction temperature threshold with 1°C resolution represented in 2's complement format. When Junction temperature passes the *THLOWTC.Threshold* value thermal mitigation actions specified in bits 31:24 of the register are initiated. Actions are discontinued when Junction temperature is below the *THLOWTC.Threshold - THLOWTC.Hystresis* value. ***Notes:*** 1. Values placed in this field should be positive and value of *THLOWTC.Threshold - THLOWTC.Hystresis* should be greater than 0. 2. There should be no overlap between the Temperature ranges defined in the *THLOWTC*, *THMIDTC* and *THHIGHTC* registers.
Reserved	15:9	0x0	Reserved Write 0 ignore on read.
Hysteresis	19:16	0xA (10 °C)	Low Junction Temperature Hysteresis Field defines value below *THLOWTC.Threshold* where thermal mitigation actions like link speed reduction (defined in *THLOWTC.TTHROTLE* field) and Power down (defined in *THLOWTC.PWR_DN* field) are stopped. Resolution of value specified is 1°C. ***Notes:*** 1. Values placed in this field should be positive and value of *THLOWTC.Threshold - THLOWTC.Hystresis* should be greater than 0. 2. There should be no overlap between the Temperature ranges defined in the *THLOWTC*, *THMIDTC* and *THHIGHTC* registers.
Reserved	20	0b	Reserved Write 0 ignore on read.
Wake_TH	21	0b	Wake on Thermal Sensor Event Wakeup when the I350 is in D3 PM state and TS interrupt to Host is generated as a result of conditions defined in the *THLOWTC* register. 0b - No wakeup on D3 as a result of *THLOWTC* thermal event. 1b - Initiate wakeup on D3 as a result of *THLOWTC* thermal event. ***Note:*** Wakeup is generated in D3 only if *WUFC.THS_WK* is set to 1b.
THSDP_IN	22	0b	Thermal Sensor SDP Input Enable If bit is set actions defined in bits 31:28, 26, 24 and 21 of the *THLOWTC* register are controlled by the SDP pin only. The SDP pin used for this functionality and polarity of the input signal is defined in the *THACNFG* register. ***Notes:*** 1. A different SDP pin can be configured for each threshold register. 2. When bit is set actions defined by the *HINTR Hyst* bit and the *BMCAL Hyst* bit are disabled. 3. When SDP pin is asserted and the *HINTR Thres* bit is set an interrupt is sent to Host (*ICR.THS* bit is set). 4. When SDP pin is asserted and the *BMCAL Thres* bit is set an Alert is sent to BMC.

Field	Bit(s)	Initial Value	Description
THSDP_OUT	23	0b	Thermal Sensor SDP Output Enable If bit is set occurrence of a thermal event is indicated on an SDP pin. When Junction temperature passes the *THLOWTC.Threshold* value the SDP pin is asserted, when Junction temperature is below the *THLOWTC.Threshold - THLOWTC.Hystresis* value the SDP pin is de-asserted. SDP pin used for this functionality and polarity of this pin is defined in the *THACNFG* register. ***Notes:*** 1. A different SDP pin can be configured for each threshold register. 2. SDP pin defined as an output for the Thermal Sensor, is configured as an Open-Drain I/O. 3. SDP pin will remain asserted so long as the thermal throttling action is taking place. For the cases where the *THLOWTC.TTHROTLE* field is programmed to remain in thermal throttling state until the *THSTAT.TL_TEVENT* bit is cleared, the SDP pin will remain asserted until the bit is cleared.
HINTR Thres	24	0b	Host Interrupt Threshold Enable Generate interrupt to Host by asserting *ICR.THS* bit when junction Temperature passes *THLOWTC.Threshold* value. 0b - No interrupt. 1b - Send interrupt.
HINTR Hyst	25	0b	Host Interrupt Hysteresis Enable Generate interrupt to Host by asserting *ICR.THS* bit when junction Temperature moves below the *THLOWTC.Threshold - THLOWTC.Hystresis* value after being above the *THLOWTC.Threshold* value. 0b - No interrupt. 1b - Send interrupt.
BMCAL Thres (RO)[1]	26	0b	BMC Alert Threshold Enable Send Alert to BMC when junction Temperature passes *THLOWTC.Threshold* value. 0b - Do not send Alert. 1b - Send Alert.
BMCAL Hyst (RO)[1]	27	0b	BMC Alert Hysteresis Enable Send Alert to BMC when junction Temperature moves below the *THLOWTC.Threshold - THLOWTC.Hystresis* value after being above the *THLOWTC.Threshold* value. 0b - Do not send Alert. 1b - Send Alert.
TTHROTLE	30:28	000b	Thermal Throttle Field defines Link rate reduction on all ports when configured to use internal copper PHY when junction temperature is above the *THLOWTC.Threshold* value. When value of field is 001b or 010b and junction temperature moves below the *THLOWTC.Threshold - THLOWTC.Hystresis* value, PHYs return to the original link rate. 000b - No thermal throttling. 001b - Move to 10M link rate on thermal event and return to original rate on end of thermal event. 010b - Disable 1G on thermal event (Move to 10M or 100M link rate) and return to original rate on end of thermal event. 011b - Move to 10M link rate on thermal event and do not return to original rate on end of thermal event (Return to original rate to be handled by BMC or Host after by clearing the *THSTAT.TL_TEVENT* bit). 100b - Disable 1G on thermal event (Move to 10M or 100M link rate) and do not return to original rate on end of thermal event (Return to original rate to be handled by BMC or Host by clearing the *THSTAT.TL_TEVENT* bit). 101b to 111b - Reserved.
PWR_DN	31	0b	Power Down on thermal event When junction temperature is above the *THLOWTC.Threshold* value all LAN ports are powered down (PHY and SerDes interfaces are powered down). When junction temperature moves below the *THLOWTC.Threshold - THLOWTC.Hystresis* value, PHYs return to the original link rate. ***Note:*** This bit should not be modified while a power down event is asserted

1. Bit is R/W from EEPROM or BMC and RO only by Host.

8.25.3 Thermal Sensor Mid Threshold Control - THMIDTC (0x8108; RW)

Field	Bit(s)	Initial Value	Description
Threshold	8:0	0x73 (115 °C)	Mid Junction Temperature threshold Mid Junction temperature threshold with 1°C resolution represented in 2's complement format. When Junction temperature passes the *THMIDTC.Threshold* value thermal mitigation actions specified in bits 31:24 of the register are initiated. Actions are discontinued when Junction temperature is below the *THMIDTC.Threshold - THMIDTC.Hystresis* value. ***Notes:*** 1. Values placed in this field should be positive and value of *THMIDTC.Threshold - THMIDTC.Hystresis* should be greater than 0. 2. There should be no overlap between the Temperature ranges defined in the *THLOWTC*, *THMIDTC* and *THHIGHTC* registers.
Reserved	15:9	0x0	Reserved Write 0 ignore on read.
Hysteresis	19:16	0xA (10 °C)	Mid Junction Temperature Hysteresis Field defines junction temperature below *THMIDTC.Threshold* where thermal mitigation actions like link speed reduction (defined in *TSLMIDTC.TTHROTLE* field) and Power down (defined in *THMIDTC.PWR_DN* field) are stopped. Resolution of value specified is 1°C. ***Notes:*** 1. Values placed in this field should be positive and value of *THMIDTC.Threshold - THMIDTC.Hystresis* should be greater than 0. 2. There should be no overlap between the Temperature ranges defined in the *THLOWTC*, *THMIDTC* and *THHIGHTC* registers.
Reserved	20	0b	Reserved Write 0 ignore on read.
Wake_TH	21	0b	Wake on Thermal Sensor Event Wakeup when the I350 is in D3 PM state and TS interrupt to Host is generated as a result of conditions defined in the *THMIDTC* register. 0b - No wakeup on D3 as a result of *THMIDTC* thermal event. 1b - Initiate wakeup on D3 as a result of *THMIDTC* thermal event. ***Note:*** Wakeup is generated in D3 only if *WUFC.THS_WK* is set to 1b.
THSDP_IN	22	0b	Thermal Sensor SDP Input Enable If bit is set actions defined in bits 31:28, 26, 24 and 21 of the *THMIDTC* register are controlled by the SDP pin only. The SDP pin used for this functionality and polarity of the input signal is defined in the *THACNFG* register. ***Notes:*** 1. A different SDP pin can be configured for each threshold register. 2. When bit is set actions defined by the *HINTR Hyst* bit and the *BMCAL Hyst* bit are disabled. 3. When SDP pin is asserted and the *HINTR Thres* bit is set an interrupt is sent to Host (*ICR.THS* bit is set). 4. When SDP pin is asserted and the *BMCAL Thres* bit is set an Alert is sent to BMC.

Field	Bit(s)	Initial Value	Description
THSDP_OUT	23	0b	Thermal Sensor SDP Output Enable If bit is set occurrence of a thermal event is indicated on an SDP pin. When Junction temperature passes the *THMIDTC.Threshold* value the SDP pin is asserted, when Junction temperature is below the *THMIDTC.Threshold - THMIDTC.Hystresis* value the SDP pin is de-asserted. SDP pin used for this functionality and polarity of this pin is defined in the *THACNFG* register. ***Notes:*** 1. A different SDP pin can be configured for each threshold register. 2. SDP pin defined as an output for the Thermal Sensor, is configured as an Open-Drain I/O. 3. SDP pin will remain asserted so long as the thermal throttling action is taking place. For the cases where the *THMIDTC.TTHROTLE* field is programmed to remain in thermal throttling state until the *THSTAT.TL_TEVENT* bit is cleared, the SDP pin will remain asserted until the bit is cleared.
HINTR Thres	24	0b	Host Interrupt Threshold Enable Generate interrupt to Host by asserting *ICR.THS* bit when junction Temperature passes *THMIDTC.Threshold* value. 0b - No interrupt. 1b - Send interrupt.
HINTR Hyst	25	0b	Host Interrupt Hysteresis Enable Generate interrupt to Host by asserting *ICR.THS* bit when junction Temperature moves below the *THMIDTC.Threshold - THMIDTC.Hystresis* value after being above the *THMIDTC.Threshold* value. 0b - No interrupt. 1b - Send interrupt.
BMCAL Thres (RO)[1]	26	0b	BMC Alert Threshold Enable Send Alert to BMC when junction Temperature passes *THMIDTC.Threshold* value. 0b - Do not send Alert. 1b - Send Alert.
BMCAL Hyst (RO)[1]	27	0b	BMC Alert Hysteresis Enable Send Alert to BMC when junction Temperature moves below the *THMIDTC.Threshold - THMIDTC.Hystresis* value after being above the *THMIDTC.Threshold* value. 0b - Do not send Alert. 1b - Send Alert.
TTHROTLE	30:28	000b	Thermal Throttle Field defines Link rate reduction on all ports when configured to use internal copper PHY when junction temperature is above the *THMIDTC.Threshold* value. When value of field is 001b or 010b and junction temperature moves below the *THMIDTC.Threshold - THMIDTC.Hystresis* value, PHYs return to the original link rate. 000b - No thermal throttling. 001b - Move to 10M link rate on thermal event and return to original rate on end of thermal event. 010b - Disable 1G on thermal event (Move to 10M or 100M link rate) and return to original rate on end of thermal event. 011b - Move to 10M link rate on thermal event and do not return to original rate on end of thermal event (Return to original rate to be handled by BMC or Host after by clearing the *THSTAT.TM_TEVENT* bit). 100b - Disable 1G on thermal event (Move to 10M or 100M link rate) and do not return to original rate on end of thermal event (Return to original rate to be handled by BMC or Host by clearing the *THSTAT.TM_TEVENT* bit). 101b to 111b - Reserved.
PWR_DN	31	0b	Power Down on thermal event When junction temperature is above the *THMIDTC.Threshold* value all LAN ports are powered down (PHY and SerDes interfaces are powered down). When junction temperature moves below the *THMIDTC.Threshold - THMIDTC.Hystresis* value, PHYs return to the original link rate. ***Note:*** This bit should not be modified while a power down event is asserted

1. Bit is R/W from EEPROM or BMC and RO only by Host.

8.25.4 Thermal Sensor High Threshold Control - THHIGHTC (0x810C; RW)

Field	Bit(s)	Initial Value	Description
Threshold	8:0	0x78 (120 °C)	High Junction Temperature threshold High Junction temperature threshold with 1°C resolution represented in 2's complement format. When Junction temperature passes the *THHIGHTC.Threshold* value thermal mitigation actions specified in bits 31:24 of the register are initiated. Actions are discontinued when Junction temperature is below the *THHIGHTC.Threshold - THHIGHTC.Hystresis* value. ***Notes:*** 1. Values placed in this field should be positive and value of *THHIGHTC.Threshold - THHIGHTC.Hysteresis* should be greater than 0. 2. There should be no overlap between the Temperature ranges defined in the *THLOWTC*, *THMIDTC* and *THHIGHTC* registers.
Reserved	15:9	0x0	Reserved Write 0 ignore on read.
Hysteresis	19:16	0xA (10 °C)	High Junction Temperature Hysteresis Field defines junction temperature below *THHIGHTC.Threshold* where thermal mitigation actions like link speed reduction (defined in *THHIGHTC.TTHROTLE* field) and Power down (defined in *THHIGHTC.PWR_DN* field*)* are stopped. Resolution of value specified is 1°C. ***Notes:*** 1. Values placed in this field should be positive and value of *THHIGHTC.Threshold - THHIGHTC.Hysteresis* should be greater than 0. 2. There should be no overlap between the Temperature ranges defined in the *THLOWTC*, *THMIDTC* and *THHIGHTC* registers.
Reserved	20	0b	Reserved Write 0 ignore on read.
Wake_TH	21	0b	Wake on Thermal Sensor Event Wakeup when the I350 is in D3 PM state and TS interrupt to Host is generated as a result of conditions defined in the *THHIGHTC* register. 0b - No wakeup on D3 as a result of *THHIGHTC* thermal event. 1b - Initiate wakeup on D3 as a result of *THHIGHTC* thermal event. ***Note:*** Wakeup is generated in D3 only if *WUFC.THS_WK* is set to 1b.
THSDP_IN	22	0b	Thermal Sensor SDP Input Enable If bit is set actions defined in bits 31:28, 26, 24 and 21 of the *THHIGHTC* register are controlled by the SDP pin only. The SDP pin used for this functionality and polarity of the input signal is defined in the *THACNFG* register. ***Notes:*** 1. A different SDP pin can be configured for each threshold register. 2. When bit is set actions defined by the *HINTR Hyst* bit and the *BMCAL Hyst* bit are disabled. 3. When SDP pin is asserted and the *HINTR Thres* bit is set an interrupt is sent to Host (*ICR.THS* bit is set). 4. When SDP pin is asserted and the *BMCAL Thres* bit is set an Alert is sent to BMC.

Field	Bit(s)	Initial Value	Description
THSDP_OUT	23	0b	Thermal Sensor SDP Output Enable If bit is set occurrence of a thermal event is indicated on an SDP pin. When Junction temperature passes the *THHIGHTC.Threshold* value the SDP pin is asserted, when Junction temperature is below the *THHIGHTC.Threshold - THHIGHTC.Hystresis* value the SDP pin is de-asserted. SDP pin used for this functionality and polarity of this pin is defined in the *THACNFG* register. ***Notes:*** 1. A different SDP pin can be configured for each threshold register. 2. SDP pin defined as an output for the Thermal Sensor, is configured as an Open-Drain I/O. 3. SDP pin will remain asserted so long as the thermal throttling action is taking place. For the cases where the *THHIGHTC.TTHROTLE* field is programmed to remain in thermal throttling state until the *THSTAT.TL_TEVENT* bit is cleared, the SDP pin will remain asserted until the bit is cleared.
HINTR Thres	24	0b	Host Interrupt Threshold Enable Generate interrupt to Host by asserting *ICR.THS* bit when junction Temperature passes *THHIGHTC.Threshold* value. 0b - No interrupt. 1b - Send interrupt.
HINTR Hyst	25	0b	Host Interrupt Hysteresis Enable Generate interrupt to Host by asserting *ICR.THS* bit when junction Temperature moves below the *THHIGHTC.Threshold - THHIGHTC.Hystresis* value after being above the *THHIGHTC.Threshold* value. 0b - No interrupt. 1b - Send interrupt.
BMCAL Thres (RO)[1]	26	0b	BMC Alert Threshold Enable Send Alert to BMC when junction Temperature passes *THHIGHTC.Threshold* value. 0b - Do not send Alert. 1b - Send Alert.
BMCAL Hyst (RO)[1]	27	0b	BMC Alert Hysteresis Enable Send Alert to BMC when junction Temperature moves below the *THHIGHTC.Threshold - THHIGHTC.Hystresis* value after being above the *THHIGHTC.Threshold* value. 0b - Do not send Alert. 1b - Send Alert.
TTHROTLE	30:28	000b	Thermal Throttle Field defines Link rate reduction on all ports when configured to use internal copper PHY when junction temperature is above the *THHIGHTC.Threshold* value. When value of field is 001b or 010b and junction temperature moves below the *THHIGHTC.Threshold - THHIGHTC.Hystresis* value, PHYs return to the original link rate. 000b - No thermal throttling. 001b - Move to 10M link rate on thermal event and return to original rate on end of thermal event. 010b - Disable 1G on thermal event (Move to 10M or 100M link rate) and return to original rate on end of thermal event. 011b - Move to 10M link rate on thermal event and do not return to original rate on end of thermal event (Return to original rate to be handled by BMC or Host after by clearing the *THSTAT.TH_TEVENT* bit). 100b - Disable 1G on thermal event (Move to 10M or 100M link rate) and do not return to original rate on end of thermal event (Return to original rate to be handled by BMC or Host by clearing the *THSTAT.TH_TEVENT* bit). 101b to 111b - Reserved.
PWR_DN	31	0b	Power Down on thermal event When junction temperature is above the *THHIGHTC.Threshold* value all LAN ports are powered down (PHY and SerDes interfaces are powered down). When junction temperature moves below the *THHIGHTC.Threshold - THHIGHTC.Hystresis* value, PHYs return to the original link rate. ***Note:*** This bit should not be modified while a power down event is asserted

1. Bit is R/W from EEPROM or BMC and RO only by Host.

8.25.5 Thermal Sensor Status - THSTAT (0x8110; RO)

Field	Bit(s)	Initial Value	Description
TPWR_DNE	0	0b	Thermal Power Down Event When bit is set indicates all LAN ports were powered down due to a thermal mitigation action defined in the *THHIGHTC*, *THMIDTC* or *THLOWTC* registers. Bit is cleared once internal logic detects that thermal condition for power down of all LAN ports does not exist. 0b - No Thermal Power Down event in progress 1b - Thermal Power Down event in progress
TTHROTE	1	0b	Thermal Link Speed Throttling Event When bit is set indicates link speed was reduced on all LAN ports due to a thermal mitigation action defined in the *THHIGHTC*, *THMIDTC* or *THLOWTC* registers. Bit is cleared once internal logic detects that thermal condition for link speed throttling of all LAN ports does not exist. 0b - No Thermal throttling event in progress 1b - Thermal throttling event in progress
Reserved	12:2	0x0	Reserved Write 0 ignore on read.
TH_Thresh	13	0b	Threshold High Threshold if set indicates that junction temperature is above value defined in *THHIGHTC.Threshold* field.
TH_TEVENT (RW1C)	14	0b	Threshold High Thermal Event Indicates thermal event in progress as defined in *THHIGHTC* register. 0b - No thermal event in progress. 1b - Thermal Event in progress. ***Note:*** Bit cleared by write 1b to return to original link rate when *THHIGHTC.TTHROTLE* value is 011b or 100b and no thermal event exists.
TH_SDP_IN	15	X	Threshold High SDP_IN Status Status of SDP input that activates thermal event actions defined in *THHIGHTC* register. 0b - SDP does not indicate need to activate thermal event actions. 1b - SDP indicates need to activate thermal event actions.
Reserved	20:16	0x0	Reserved Write 0, ignore on read.
TM_Thresh	21	0b	Threshold Mid Threshold if set indicates that junction temperature is above value defined in *THMIDTC.Threshold* field.
TM_TEVENT (RW1C)	22	0b	Threshold Mid Thermal Event Indicates thermal event in progress as defined in *THMIDTC* register. 0b - No thermal event in progress. 1b - Thermal Event in progress. ***Note:*** Bit cleared by write 1b to return to original link rate when *THMIDTC.TTHROTLE* value is 011b or 100b and no thermal event exists.
TM_SDP_IN	23	X	Threshold Mid SDP_IN Status Status of SDP input that activates thermal event actions defined in *THMIDTC* register. 0b - SDP does not indicate need to activate thermal event actions. 1b - SDP indicates need to activate thermal event actions.
Reserved	28:24	0x0	Reserved Write 0, ignore on read.

Field	Bit(s)	Initial Value	Description
TL_Thresh	29	0b	Threshold Low Threshold if set indicates that junction temperature is above value defined in *THLOWTC.Threshold* field.
TL_TEVENT (RW1C)	30	0b	Threshold Low Thermal Event Indicates thermal event in progress as defined in *THLOWTC* register. 0b - No thermal event in progress. 1b - Thermal Event in progress. ***Note:*** Bit cleared by write 1b to return to original link rate when *THLOWTC.TTHROTLE* value is 011b or 100b and no thermal event exists.
TL_SDP_IN	31	X	Threshold Low SDP_IN Status Status of SDP input that activates thermal event actions defined in *THLOWTC* register. 0b - SDP does not indicate need to activate thermal event actions. 1b - SDP indicates need to activate thermal event actions.

8.25.6 Thermal Sensor Auxiliary Configuration - THACNFG (0x8114; RW)

Field	Bit(s)	Initial Value	Description
TH_RST	0	0b	Thermal Sensor Reset When set to 1b Thermal Sensor is reset and is powered down. 0b - Thermal Sensor active. 1b - Thermal Sensor reset.
Reserved	1	0b	Reserved Write 0, ignore on read.
TH_SDPO_POL	2	0b	Threshold High SDP_OUT Pin polarity Indicates output polarity of SDP_OUT pin defined in *THHIGHTC* register. 0b - When SDP pin is low thermal event defined in *THHIGHTC* register is in progress. 1b - When SDP pin is high thermal event defined in the *THHIGHTC* register is in progress.
TH_SDPO_PIN	4:3	10b	Threshold High SDP_OUT Pin Select Defines SDP pin (0 to 3) in port defined by *THACNFG.TL_SDPO_PORT* where the O/D SDP output pin that indicates occurrence of thermal event according to the definitions in the *THHIGHTC* register is located. ***Note:*** Field affects configuration of SDP pins only if *THHIGHTC.THSDP_OUT* is set to 1.
TH_SDPO_PORT	6:5	01b	Threshold High SDP_OUT Port Select Defines port (0 to 3) where the O/D SDP output pin that indicates occurrence of thermal event according to the definitions in the *THHIGHTC* register is located. ***Note:*** Field affects configuration of SDP pins only if *THHIGHTC.THSDP_OUT* is set to 1.
TH_SDPI_POL	7	0b	Threshold High SDP_IN Pin polarity Indicates input polarity of SDP_IN pin defined in *THHIGHTC* register. 0b - When SDP pin is low activate thermal events defined in *THHIGHTC* register. 1b - When SDP pin is high activate thermal events defined in *THHIGHTC* register.
TH_SDPI_PIN	9:8	10b	Threshold High SDP_IN Pin Select Defines SDP pin (0 to 3) in port defined by THACNFG.TL_SDPI_PORT where SDP pin that activates actions defined in *THHIGHTC* register is located. ***Note:*** Field affects configuration of SDP pins only if *THHIGHTC.THSDP_IN* is set to 1.
TH_SDPI_PORT	11:10	00b	Threshold High SDP_IN Port Select Defines port (0 to 3) where SDP pin that activates actions defined in *THHIGHTC* register is located. ***Note:*** Field affects configuration of SDP pins only if *THHIGHTC.THSDP_IN* is set to 1.

Field	Bit(s)	Initial Value	Description
TM_SDPO_POL	12	0b	Threshold Mid SDP_OUT Pin polarity Indicates output polarity of SDP_OUT pin defined in *THMIDTC* register. 0b - When SDP pin is low thermal event defined in *THMIDTC* register is in progress. 1b - When SDP pin is high thermal event defined in the *THMIDTC* register is in progress.
TM_SDPO_PIN	14:13	01b	Threshold Mid SDP_OUT Pin Select Defines SDP pin (0 to 3) in port defined by *THACNFG.TL_SDPO_PORT* where the O/D SDP output pin that indicates occurrence of thermal event according to the definitions in the *THMIDTC* register is located. ***Note:*** Field affects configuration of SDP pins only if *THMIDTC.THSDP_OUT* is set to 1.
TM_SDPO_PORT	16:15	01b	Threshold Mid SDP_OUT Port Select Defines port (0 to 3) where the O/D SDP output pin that indicates occurrence of thermal event according to the definitions in the *THMIDTC* register is located. ***Note:*** Field affects configuration of SDP pins only if *THMIDTC.THSDP_OUT* is set to 1.
TM_SDPI_POL	17	0b	Threshold Mid SDP_IN Pin polarity Indicates input polarity of SDP_IN pin defined in *THMIDTC* register. 0b - When SDP pin is low activate thermal events defined in *THMIDTC* register. 1b - When SDP pin is high activate thermal events defined in *THMIDTC* register.
TM_SDPI_PIN	19:18	01b	Threshold Mid SDP_IN Pin Select Defines SDP pin (0 to 3) in port defined by THACNFG.TL_SDPI_PORT where SDP pin that activates actions defined in *THMIDTC* register is located. ***Note:*** Field affects configuration of SDP pins only if *THMIDTC.THSDP_IN* is set to 1.
TM_SDPI_PORT	21:20	00b	Threshold Mid SDP_IN Port Select Defines port (0 to 3) where SDP pin that activates actions defined in *THMIDTC* register is located. ***Note:*** Field affects configuration of SDP pins only if *THMIDTC.THSDP_IN* is set to 1.
TL_SDPO_POL	22	0b	Threshold Low SDP_OUT Pin polarity Indicates output polarity of SDP_OUT pin defined in *THLOWTC* register. 0b - When SDP pin is low thermal event defined in *THLOWTC* register is in progress. 1b - When SDP pin is high thermal event defined in the *THLOWTC* register is in progress.
TL_SDPO_PIN	24:23	00b	Threshold Low SDP_OUT Pin Select Defines SDP pin (0 to 3) in port defined by *THACNFG.TL_SDPO_PORT* where the O/D SDP output pin that indicates occurrence of thermal event according to the definitions in the *THLOWTC* register is located. ***Note:*** Field affects configuration of SDP pins only if *THLOWTC.THSDP_OUT* is set to 1.
TL_SDPO_PORT	26:25	01b	Threshold Low SDP_OUT Port Select Defines port (0 to 3) where the O/D SDP output pin that indicates occurrence of thermal event according to the definitions in the *THLOWTC* register is located. ***Note:*** Field affects configuration of SDP pins only if *THLOWTC.THSDP_OUT* is set to 1.
TL_SDPI_POL	27	0b	Threshold Low SDP_IN Pin polarity Indicates input polarity of SDP_IN pin defined in *THLOWTC* register. 0b - When SDP pin is low activate thermal events defined in *THLOWTC* register. 1b - When SDP pin is high activate thermal events defined in *THLOWTC* register.
TL_SDPI_PIN	29:28	00b	Threshold Low SDP_IN Pin Select Defines SDP pin (0 to 3) in port defined by THACNFG.TL_SDPI_PORT where SDP pin that activates actions defined in *THLOWTC* register is located. ***Note:*** Field affects configuration of SDP pins only if *THLOWTC.THSDP_IN* is set to 1.
TL_SDPI_PORT	31:30	00b	Threshold Low SDP_IN Port Select Defines port (0 to 3) where SDP pin that activates actions defined in *THLOWTC* register is located. ***Note:*** Field affects configuration of SDP pins only if *THLOWTC.THSDP_IN* is set to 1.

8.25.7 Rx Packet Buffer Wrap Around Counter - PBRWAC (0x24E8; RO)

Field	Bit(s)	Initial Value	Description
WALPB	2:0	0x0	Counts the wrap around events of the LAN Rx packet's buffer. Reflects the wrap around events of the entire LAN Port Packet Buffer.
PBE	3	1b	Rx Packet buffer is empty
Reserved	7:4	0x0	Reserved. Write 0, ignore on read.
WAMNGPB	10:8	0x0	Counts the wrap around events of the Management Rx packet buffer. Reflects the wrap around events of the entire Management Packet Buffer.
MNGPBE	11	1b	Management Rx Packet buffer is empty
Reserved	15:12	0x0	Reserved. Write 0, ignore on read.
WALPBKPB	18:16	0x0	Reflects the wrap around of the entire Packet buffer. Counts the wrap around events of Loopback Rx Packet Buffer. Reflects the wrap around of the entire Loopback Packet buffer.
LPBKPBE	19	1b	Loopback Rx Packet buffer is empty
Reserved	31:20	0x0	Reserved. Write 0, ignore on read.

8.26 PHY Software Interface

8.26.1 Internal PHY Configuration - IPCNFG (0x0E38, RW)

The *IPCNFG* register controls PHY configuration.

Field	Bit(s)	Initial Value	Description	Mode
Reserved	0		Reserved	
MDI_Flip	0	0b[1]	MDI Flip When set MDI Channel D is exchanged with MDI Channel A and MDI Channel C is exchanged with MDI Channel B.	R/W
10BASE-TE	1	0b[2]	Enable low amplitude 10BASE-T operation Setting this bit enables the I350 to operate in IEEE802.3az 10BASE-Te low power operation. 0b - 10BASE-Te operation disabled. 1b - 10BASE-Te operation enabled ***Note:*** When operating in 10BASE-T mode and bit is set supported cable length is reduced.	R/W
EEE_100M_AN	2	1b[2]	Report EEE 100M capability in Auto-negotiation 0b - Do not report EEE 100M capability in Auto-negotiation. 1b - Report EEE 100M capability in Auto-negotiation. ***Note:*** Changing value of bit causes link drop and re-negotiation.	R/W

Field	Bit(s)	Initial Value	Description	Mode
EEE_1G_AN	3	1b[2]	Report EEE 1G capability in Auto-negotiation 0b - Do not report EEE 1G capability in Auto-negotiation. 1b - Report EEE 1G capability in Auto-negotiation. ***Note:*** Changing value of bit causes link drop and re-negotiation.	R/W
Reserved	31:4	0x0	Reserved. Write 0, ignore on read.	R/W

1. Bit Loaded from *MDI_Flip* bit in *Initialization Control 3* EEPROM word on power-up.
2. Loaded from EEPROM.

8.26.2 PHY Power Management - PHPM (0x0E14, RW)

The *PHPM* register controls Internal PHY Power management operation.

Field	Bit(s)	Initial Value	Description	Mode
SPD_EN[1]	0	1b	Smart Power Down When set, enables PHY Smart Power Down mode. ***Note:*** Bit 3 in PHMIC (21d) register should be 0b to allow the *PHPM.SPD_EN* bit to disable Smart Power Down operation.	R/W
Reserved	0		Reserved	
D0LPLU	1	0b	D0 Low Power Link Up When set, configures the PHY to negotiate for a low speed link in all states. ***Note:*** Bit 8 in *PHCTRL1* (23d) register should be 0b for the *PHPM.D0LPLU* bit to disable Low Power Link Up operation.	R/W
LPLU[2]	2	1b	Low Power on Link Up When set, enables the decrease in link speed while in non-D0a states when the power policy and power management state specify it. ***Note:*** Bit 8 in *PHCTRL1* (23d) register should be 0b for the *PHPM.LPLU* bit to disable Low Power Link Up operation.	R/W
Disable 1000 in non-D0a[3]	3	1b	Disables 1000 Mb/s operation in non-D0a states. ***Note:*** Bit 0 in PHMIC (21d) register should be 0b to allow the *PHPM.Disable 1000 in non-D0a* bit to enable 1G operation in non-D0a PM state.	R/W
Link Energy Detect	4	0b	This bit is set when the PHY detects energy on the link. Note that this bit is valid only if *PHPM.Go Link disconnect* is set to 1b.	RO, LH
Go Link disconnect	5	0b	Setting this bit will cause the PHY to enter link disconnect mode immediately.	R/W
Disable 1000[4]	6	0b	When set, disables 1000 Mb/s in all power modes. ***Note:*** Bit 0 in PHMIC (21d) register should be 0b to allow the *PHPM.Disable 1000* bit to enable 1G operation.	R/W
SPD_B2B_EN	7	1b	SPD back-to-back enable. ***Note:*** Bit 2 in PHMIC (21d) register should be 0b to allow the *PHPM.SPD_B2B_EN* bit to disable Smart Power Down Back to Back operation.	R/W
rst_compl	8	0b	Indicates PHY internal reset cleared.	RO, LH
Disable 100 in non-D0a[5]	9	0b	Disables 100 Mb/s and 1000 Mb/s operation in non-D0a states.	R/W
Reserved	31:10	0x0	Reserved. Write 0, ignore on read.	R/W

1. Bit Loaded from *SPD Enable* bit in *Initialization Control 4* EEPROM word on reset
2. Bit Loaded from *LPLU* bit in *Initialization Control 4* EEPROM word on reset
3. Bit Loaded from *Disable 1000 in non-D0a* bit in *Software Defined Pins Control* EEPROM word on reset
4. Bit Loaded from *Giga Disable* bit in *Software Defined Pins Control* EEPROM word on reset
5. Bit Loaded from *Disable 100 in non-D0a* bit in *Software Defined Pins Control* EEPROM word on reset

8.26.3 Internal PHY Software Interface (PHYREG)

1. Base Registers (0 through 10 and 15) are defined in accordance with the "Reconciliation Sub layer and Media Independent Interface" and "Physical Layer Link Signaling for 10/100/ 1000 Mb/s Auto-Negotiation" sections of the IEEE 802.3.
2. Additional registers (PHYREG.16 through 31) are defined in accordance with the IEEE 802.3 specification for adding unique chip functions.
3. Registers in Table 8-24 are accessed using the internal MDIO interface via the MDIC register (Refer to Section 3.7.2.2).

Table 8-24 Table of PHYREG Registers

Offset	Abbreviation	Name	RW	Link to Page
00d	PCTRL	PHY Control Register	R/W	page 686
01d	PSTATUS	PHY Status Register	RO	page 687
02d	PHY ID 1	PHY Identifier Register 1 (LSB)	RO	page 688
03d	PHY ID 2	PHY Identifier Register 2 (MSB)	RO	page 688
04d	ANA	Auto–Negotiation Advertisement Register	R/W	page 688
05d	ANLPA	Auto–Negotiation link partner Ability Register	RO	page 689
06d	ANE	Auto–Negotiation Expansion Register	RO	page 690
07d	NPT	Auto–Negotiation Next Page Transmit Register	R/W	page 690
08d	LPN	Auto–Negotiation Next Page Link Partner Register	RO	page 691
09d	GCON	1000BASE–T/100BASE–T2 Control Register	R/W	page 691
10d	GSTATUS	1000BASE–T/100BASE–T2 Status Register	RO	page 692
11d - 14d		Reserved		
15d	ESTATUS	Extended Status Register	RO	page 692
16d	EMIADD	Extended Memory Indirect Address Register	R/W	page 693
17d	EMIDATA	Extended Memory Indirect Address Register	R/W	page 693
18d	PHCTRL2	PHY control register 2	R/W	page 693
19d	PHLBKC	Loopback control register.	R/W	page 695
20d	PHRERRC	RX error counter register	RO, SC	page 696
21d	PHMIC	Management interface (MI) control register.	R/W	page 696
22d	PHCNFG	PHY configuration register.	R/W	page 697
23d	PHCTRL1	PHY control register 1.	R/W	page 698
24d	PHINTM	Interrupt mask register.	R/W	page 699
25d	PHINTR	Interrupt status register.	RO	page 699
26d	PHSTAT	PHY status register.	RO	page 700
29d - 27d		Reserved.		
31d	PHDSTAT	Diagnostics status register.	RO	page 702

8.26.3.1 PHY Control Register - PCTRL (00d; R/W)

Field	Bit(s)	Description	Mode	Default
Reserved	5:0	Reserved. Write 0, ignore on read.	RW	0x0
Speed Selection 1000 Mb/s (MSB)[5]	6	Speed Selection is determined by bits 6 (MSB) and 13 (LSB) as follows. 11b = Reserved 10b = 1000 Mb/s 01b = 100 Mb/s 00b = 10 Mb/s **Note:** If auto-negotiation is enabled, this bit is ignored.	R/W	00b
Collision Test[1]	7	1b = Enable COL signal test. 0b = Disable COL signal test.	R/W	0b
Duplex Mode[2]	8	1b = Full Duplex. 0b = Half Duplex. **Note:** If auto-negotiation is enabled, this bit is ignored.	R/W	1b
Restart Auto-Negotiation	9	1b = Restart Auto-Negotiation Process. 0b = Normal operation. Auto-Negotiation automatically restarts after hardware or software reset regardless of whether or not the restart bit is set.	R/W, SC	0b
Isolate[3]	10	1b = Isolates PHY from internal MII/GMII interface. 0b = Normal operation.	R/W	0b
Power Down	11	1b = Power down. 0b = Normal operation.	R/W	0b
Auto-Negotiation Enable[4]	12	1b = Enable Auto-Negotiation Process. 0b = Disable Auto-Negotiation Process. This bit must be enabled for 1000BASE-T operation.	R/W	1b
Speed Selection (LSB)[5]	13	Refer to Speed Selection (MSB), bit 6. **Note:** If auto-negotiation is enabled, this bit is ignored.	R/W	0b
Loopback[6]	14	1b = Enable loopback. 0b = Disable loopback.	R/W	0b
Reset[7]	15	1b = PHY reset. 0b = Normal operation. **Note:** When using PHY Reset, the PHY default configuration is not loaded from the EEPROM. The preferred way to reset the I350 PHY is using the *CTRL.PHY_RST* field. Refer to	WO, SC	0b

1. Enables IEEE 802.3 Clause 22.2.4.1.9 collision test.
2. This bit may be used to configure the link manually. Setting this bit has no effect unless address 0d, bit 12 is clear.
3. Setting this bit isolates the PHY from the internal MII or GMII interfaces.
4. When this bit is cleared, the link configuration is determined manually.
5. The speed selection address 0d, bits 13 and 6 may be used to configure the link manually. Setting these bits has no effect unless address 0d bit 12 is clear.
6. This is the master enable for digital and analog loopback as defined by the standard. The exact type of loopback is determined by the loopback control register (address 19d).
7. The reset bit is automatically cleared upon completion of the reset sequence. This bit is set to 1 during reset.

8.26.3.2 PHY Status Register - PSTATUS (01d; RO)

Field	Bit(s)	Description	Mode	Default
Extended Capability[1]	0	1b = Extended register capabilities.	RO	1b
Jabber Detect	1	1b = Jabber condition detected. 0b = Jabber condition not detected.	RO LH	0b
Link Status[2]	2	1b = Link is up. 0b = Link is down.	RO, LL	0b
Auto-Negotiation Ability	3	1b = PHY able to perform Auto-Negotiation. 0b = PHY is not able to perform Auto-Negotiation.	RO	1b
Remote Fault[3]	4	1b = Remote fault condition detected. 0b = Remote fault condition not detected.	RO LH	0b
Auto-Negotiation [4] Complete	5	1b = Auto-Negotiation process complete. 0b = Auto-Negotiation process not complete.	RO	0b
MF Preamble Suppression	6	1b = PHY accepts management frames with preamble suppressed. 0b = PHY does not accept management frames with preamble suppressed.	RO	1b
Reserved	7	Reserved. Write 0, ignore on read.	RO	0b
Extended Status	8	1b = Extended status information in the Extended PHY Status Register (15d). 0b = No extended status information in the Extended PHY Status Register (15d).	RO	1b
100BASE-T2 Half Duplex	9	1b = PHY able to perform half duplex 100BASE-T2 (not supported). 0b = PHY not able to perform half duplex 100BASE-T2.	RO	0b
100BASE-T2 Full Duplex	10	1b = PHY able to perform full duplex 100BASE-T2 (not supported). 0b = PHY not able to perform full duplex 100BASE-T2.	RO	0b
10 Mb/s Half Duplex	11	1b = PHY able to perform half duplex 10BASE-T. 0b = PHY not able to perform half duplex 10BASE-T.	RO	1b
10 Mb/s Full Duplex	12	1b = PHY able to perform full duplex 10BASE-T. 0b = PHY not able to perform full duplex 10BASE-T.	RO	1b
100BASE-X Half Duplex	13	1b = PHY able to perform half duplex 100BASE-X. 0b = PHY not able to perform half duplex 100BASE-X.	RO	1b
100BASE-X Full Duplex	14	1b = PHY able to perform full duplex 100BASE-X. 0b = PHY not able to perform full duplex 100BASE-X.	RO	1b
100BASE-T4	15	1b = PHY able to perform 100BASE-T4. 0b = PHY not able to perform 100BASE-T4.	RO	0b

1. Indicates that the PHY provides an extended set of capabilities that may be accessed through the extended register set. For a PHY that incorporates a GMII/RGMII, the extended register set consists of all management registers except registers 0, 1, and 15.
2. This bit indicates that a valid link has been established. Once cleared due to link failure, this bit remains cleared until register 1d is read via the management interface.
3. This bit indicates that a remote fault has been detected. Once set, it remains set until it is cleared by reading register 1d via the management interface or by PHY reset.
4. Upon completion of auto negotiation, this bit becomes set.

8.26.3.3 PHY Identifier Register 1 (LSB) - PHY ID 1 (02d; RO)

Field	Bit(s)	Description	Mode	Default
PHY ID Number[1]	15:0	The PHY identifier composed of bits 3 through 18 of the Organizationally Unique Identifier (OUI)	RO	0x0154

1. PHY ID Number based on Intel assigned OUI number of 00-AA-00 following bit reversal.

8.26.3.4 PHY Identifier Register 2 (MSB) - PHY ID 2 (03d; RO)

Field	Bit(s)	Description	Mode	Default
Manufacturer's Revision Number	3:0	4 bits containing the manufacturer's revision number.	RO	0x0
Manufacturer's Model Number	9:4	6 bits containing the manufacturer's part number.	RO	0x3B
PHY ID Number[1]	15:10	The PHY identifier composed of bits 19 through 24 of the OUI	RO	0x0

1. PHY ID Number based on Intel assigned OUI number of 00-AA-00 following bit reversal.

8.26.3.5 Auto–Negotiation Advertisement Register - ANA (04d; R/W)

Field	Bit(s)	Description	Mode	Default
Selector Field	4:0	00001b = 802.3 Other combinations are reserved. Unspecified or reserved combinations should not be transmitted. **Note:** Setting this field to a value other than 00001b can cause auto negotiation to fail.	R/W	00001b
10Base-T Half Duplex	5	1b = DTE is 10BASE-T Half Duplex capable. 0b = DTE is not 10BASE-T Half Duplex capable.	R/W	1b
10Base-T Full Duplex	6	1b = DTE is 10BASE-T Full duplex capable. 0b = DTE is not 10BASE-T Full duplex capable.	R/W	1b
100Base-TX Half Duplex	7	1b = DTE is 100BASE-TX Half Duplex capable. 0b = DTE is not 100BASE-TX Half Duplex capable.	R/W	1b
100BASE-TX Full Duplex	8	1b = DTE is 100BASE-TX Full duplex capable. 0b = DTE is not 100BASE-TX Full duplex capable.	R/W	1b
100BASE-T4	9	1b = Capable of 100BASE-T4 (not supported). 0b = Not capable of 100BASE-T4.	RO	0b
PAUSE	10	Advertise to Partner that Pause operation (as defined in 802.3x) is desired.	R/W	1b
ASM_DIR	11	Advertise Asymmetric Pause direction bit. This bit is used in conjunction with PAUSE.	R/W	1b
Reserved	12	Reserved. Write 0, ignore on read.	R/W	0b

Field	Bit(s)	Description	Mode	Default
Remote Fault	13	1b = Set Remote Fault bit. 0b = Do not set Remote Fault bit.	R/W	0b
Reserved	14	Reserved. Write 0, ignore on read.	R/W	0b
Next Page	15	1 = Advertises next page ability supported. 0 = Advertises next page ability not supported.	R/W	0b

8.26.3.6 Auto–Negotiation Link Partner Ability Register - ANLPA (05d; RO)

Field	Bit(s)	Description	Mode	Default
Selector Fields[4:0]	4:0	<00001> = IEEE 802.3 Other combinations are reserved. Unspecified or reserved combinations must not be transmitted. If field does not match PHY Register 04d, bits 4:0, the AN process does not complete and no HCD is selected.	RO	N/A
10BASE-T Half Duplex	5	1b = Link Partner is 10BASE-T Half Duplex capable. 0b = Link Partner is not 10BASE-T Half Duplex capable.	RO	N/A
10BASE-T Full Duplex	6	1b = Link Partner is 10BASE-T Full duplex capable. 0b = Link Partner is not 10BASE-T Full duplex capable.	RO	N/A
100BASE-TX Half Duplex	7	1b = Link Partner is 100BASE-TX Half Duplex capable. 0b = Link Partner is not 100BASE-TX Half Duplex capable.	RO	N/A
100BASE-TX Full Duplex	8	1b = Link Partner is 100BASE-TX Full duplex capable. 0b = Link Partner is not 100BASE-TX Full duplex capable.	RO	N/A
100BASE-T4	9	1b = Link Partner is 100BASE-T4 capable. 0b = Link Partner is not 100BASE-T4 capable.	RO	N/A
LP Pause	10	Link Partner uses Pause Operation as defined in 802.3x.	RO	N/A
LP ASM_DIR	11	Asymmetric Pause Direction Bit 1b = Link Partner is capable of asymmetric pause. 0b = Link Partner is not capable of asymmetric pause.	RO	N/A
Reserved	12	Reserved. Write 0, ignore on read.		
Remote Fault	13	1b = Remote fault. 0b = No remote fault.	RO	N/A
Acknowledge	14	1b = Link Partner has received Link Code Word from the PHY. 0b = Link Partner has not received Link Code Word from the PHY.	RO	N/A
Next Page	15	1b = Link Partner has ability to send multiple pages. 0b = Link Partner has no ability to send multiple pages.	RO	N/A

8.26.3.7 Auto–Negotiation Expansion Register - ANE (06d; RO)

Field	Bit(s)	Description	Mode	Default
Link Partner Auto-Negotiation Able	0	1b = Link Partner is Auto-Negotiation able. 0b = Link Partner is not Auto-Negotiation able.	RO	0b
Page Received	1	Indicates that a new page has been received and the received code word has been loaded into PHY register 05d (base pages) or PHY register 08d (next pages) as specified in clause 28 of 802.3. 1 = New page has been received from link partner. 0 = New page has not been received.	RO/LH	0b
Next Page Able	2	1b = Local device is next page able. 0b = Local device is not next page able.	RO	1b
Link Partner Next Page Able	3	1b = Link Partner is next page able. 0b = Link Partner is not next page able.	RO	0b
Parallel Detection Fault	4	1b = Parallel detection fault has occurred. 0b = Parallel detection fault has not occurred.	RO/LH	0b
Reserved	15:5	Reserved. Write 0, ignore on read.		

8.26.3.8 Auto–Negotiation Next Page Transmit Register - NPT (07d; R/W)

Field	Bit(s)	Description	Mode	Default
Message/Un-formatted Field	10:0	11-bit Next page message code or Un-formatted data.	R/W	0x1
Toggle	11	1b = Previous value of the transmitted Link Code Word = 0b. 0b = Previous value of the transmitted Link Code Word = 1b.	RO	0b
Acknowledge 2	12	1b = Complies with message. 0b = Cannot comply with message.	R/W	0b
Message Page	13	1b = Message page. 0b = Un-formatted page.	R/W	1b
Reserved	14	Reserved. Write 0, ignore on read.		
Next Page	15	1b = Additional next pages follow. 0b = Last page.	R/W	0b

8.26.3.9 Auto–Negotiation Next Page Link Partner Register - LPN (08d; RO)

Bit(s)	Field	Description	Mode	Default
10:0	Message/Un-formatted Field	11-bit Next page message code or Un-formatted data.	RO	0x0
11	Toggle	1b = Previous value of the transmitted Link Code Word = 0b. 0b = Previous value of the transmitted Link Code Word = 1b.	RO	0b
12	Acknowledge 2	1b = Link Partner complies with the message. 0b = Link Partner cannot comply with the message.	RO	0b
13	Message Page	1b = Page sent by the Link Partner is a Message Page. 0b = Page sent by the Link Partner is an Un-formatted Page.	RO	0b
14	Acknowledge	1b = Link Partner has received Link Code Word from the PHY. 0b = Link Partner has not received Link Code Word from the PHY.	RO	0b
15	Next Page	1b = Link Partner has additional next pages to send. 0b = Link Partner has no additional next pages to send.	RO	0b

8.26.3.10 1000BASE–T/100BASE–T2 Control Register - GCON (09d; R/W)

Bit(s)	Field	Description	Mode	Default
7:0	Reserved	Reserved. Write 0, ignore on read.		
8	1000BASE-T Half Duplex	1b = DTE is 1000BASE-T Half Duplex capable. 0b = DTE is not 1000BASE-T Half Duplex capable. This bit is used by Smart Negotiation.	R/W	0b
9	1000BASE-T Full Duplex	1b = DTE is 1000BASE-T full duplex capable. 0b = DTE is not 1000BASE-T full duplex capable. This bit is used by Smart Negotiation.	R/W	1b
10	Port Type	1b = Prefer multi-port device (Master). 0b = Prefer single port device (Slave). This bit is only used when PHY register 9, bit 12 is set to 0b.	R/W	1b
11	Master/Slave Config Value[1]	1b = Configure PHY as MASTER during MASTER-SLAVE negotiation (only when PHY register 9, bit 12 is set to 1b. 0b = Configure PHY as SLAVE during MASTER-SLAVE negotiation (only when PHY register 9, bit 12 is set to 1b.	R/W	0b
12	Master/Slave Config Enable	1b = Manual Master/Slave configuration. 0b = Automatic Master/Slave configuration.	R/W	0b
15:13	Test mode	000b = Normal Mode. 001b = Pulse and Droop Template. 010b = Jitter Template. 011b = Jitter Template. 100b = Distortion Packet. 101b, 110b, 111b = Reserved.	R/W	000b

1. Setting this bit has no effect unless address 9d, bit 12 is set.

8.26.3.11 1000BASE–T/100BASE–TX Status Register - GSTATUS (10d; RO)

Bit(s)	Field	Description	Mode	Default
7:0	Idle Error Count[1]	MSB of idle error count.	RO, SC	0x0
9:8	Reserved	Reserved. Write 0, ignore on read.		
10	LP 1000T HD	1b = Link Partner is capable of 1000BASE-T half duplex. 0b = Link Partner is not capable of 1000BASE-T half duplex. Value in bit 10 are not valid until the ANE Register Page Received bit equals 1b.	RO	0b
11	LP 1000T FD	1b = Link Partner is capable of 1000BASE-T full duplex. 0b = Link Partner is not capable of 1000BASE-T full duplex. Value in bit 11 are not valid until the ANE Register Page Received bit equals 1b.	RO	0b
12	Remote Receiver Status	1b = Remote Receiver OK. 0 b = Remote Receiver Not OK.	RO	0b
13	Local Receiver Status[2]	1b = Local Receiver OK. 0b = Local Receiver Not OK.	RO	0b
14	Master/Slave Resolution[3]	1b = Local PHY configuration resolved to Master. 0b = Local PHY configuration resolved to Slave. Value in bit 14 is not valid until the ANE Register Page Received bit equals 1b.	RO	0b
15	Master/Slave Config Fault[4]	1b = Master/Slave configuration fault detected. 0b = No Master/Slave configuration fault detected.	RO, LH, SC	0b

1. These bits contain a cumulative count of the errors detected when the receiver is receiving idles and both local and remote receiver status are OK. The count is held at 255 in the event of overflow and is reset to zero by reading register 10d via the management interface or by reset.
2. In the context of Energy Efficient Ethernet, during the Quiet periods, there is no signal transmitted to the line, hence the receiver will get no signal and will not be ready to receive (e.g. clock will be lost). This means that its local receiver status shall be set to NOT_OK to convey the information, in this case, to the PMA PHY Control function, as is implied by Figure 40-15b of the IEEE Draft P802.3az-D2.3.
3. This bit is not valid when bit 15 is set.
4. Once set, this bit remains set until cleared by the following actions:
   - Read of register 10d via the management interface.
   - Reset.
   - Completion of auto negotiation.
   - Enable of auto negotiation

8.26.3.12 Extended Status Register - ESTATUS (15d; RO)

Field	Bit(s)	Description	Mode	Default
Reserved	11:0	Reserved. Write 0, ignore on read.		
1000BASE-T Half Duplex	12	1b = 1000BASE-T half duplex capable. 0b = not 1000BASE-T half duplex capable.	RO	1b
1000BASE-T Full Duplex	13	1b = 1000BASE-T full duplex capable. 0b = Not 1000BASE-T full duplex capable.	RO	1b
1000BASE-X Half Duplex	14	1b =1000BASE-X half duplex capable. 0b = Not 1000BASE-X half duplex capable.	RO	0b

Field	Bit(s)	Description	Mode	Default
1000BASE-X Full Duplex	15	1b =1000BASE-X full duplex capable. 0b = Not 1000BASE-X full duplex capable.	RO	0b

8.26.3.13 Extended Memory Indirect Address Register - EMIADD (16d; R/W)

The *EMIADD* and *EMIDATA* registers enable indirect access to registers internal to the PHY at addresses greater than 31d. To read or write registers in the extended address space register address should be written to the *EMIADD* register and data should be written or read from the *EMIDATA* register.

Field	Bit(s)	Description	Mode	Default
EADD	15:0	Extended Register Address	R/W	0x0

8.26.3.14 Extended Memory Indirect Data Register - EMIDATA (17d; R/W)

Field	Bit(s)	Description	Mode	Default
EDATA	15:0	Extended Register Data	R/W	0x0

8.26.3.15 EEE MMD Extended Register support

Following Energy Efficient Ethernet (EEE) register bits defined in IEEE802.3az clause 45 are placed in extended PHY memory space and can be accessed using the *EMIADD* and *EMIDATA* registers.

Table 8-25 shows the mapping between the IEEE802.3az EEE MMD register bits and the registers inside the I350 PHY.

Table 8-25 IEEE802.3az EEE PHY registers

MMD register	MMD Bits	Default	Name	Hex EMI Address	Comments
3.0	10	1b	Clock stoppable	0x182A	Read on bit 0 of EMI address
3.1	11	0b	Tx LPI received (RO/LH)	0x182E	Read on bit 3 of EMI address
	10	0b	Rx LPI received (RO/LH)		Read on bit 2 of EMI address
	9	0b	TX LPI indication (RO)		Read on bit 1 of EMI address
	8	0b	RX LPI indication (RO)		Read on bit 0 of EMI address
3.20	15:0	11b	EEE capability register	0x0410	
3.22	15:0	0x0	EEE wake error counter (RO/ NR)	0x4C08	In 100BASE-TX mode
				0x4802	In 1000BASE-T mode
7.60	1:0	11b	EEE advertisement	0x040E	
7.61	1:0	00b	EEE LP advertisement	0x040F	

In addition to the IEEE specified MMD registers tabulated above, the I350 PHY provides a single bit EMI register to force EEE mode for a 1000BASE-T or 100BASE-TX link. Since the IEEE802.3az specification mandates auto-negotiation for EEE, this register is intended for verification purposes only.

	Bits	Name	Hex EMI Address	Comments
	0	eee_en_frc_emi	040C	forces the EEE mode for a 1000BASE-T or 100BASE-TX link

8.26.3.16 PHY Control Register 2 - PHCTRL2(18d; R/W)

Field	Bit(s)	Description	Mode	Default
Reserved	0	Reserved. Write 0, ignore on read.		0b
Reserved_1	1	Reserved. Ignore on read, write 1.		1b
Enable Diagnostics[1]	2	1b = Enables diagnostics. 0b = Disables diagnostics.	R/W	0b
Reserved_1	3	Reserved. Ignore on read, write 1.		1b
Reserved	8:4	Reserved. Write 0, ignore on read.		
MDI/MDI-X Configuration	9	1b = Manual MDI-X configuration. 0b = Manual MDI configuration.	R/W	0b
Automatic MDI/MDI-X	10	1b = Enables automatic MDI/MDI-X detection. 0b = Disables automatic MDI/MDI-X detection.	R/W	1b
Reserved	12:11	Reserved. Write 0, ignore on read.		
Count Symbol Errors	13[2]	1b = Rx error counter counts symbol errors. 0b = Rx error counter counts CRC errors.	R/W	0b
Count False Carrier Events	14[2]	1b = Rx error counter counts false carrier events. 0b = Rx error counter does not count false carrier events.	R/W	0b
Resolve MDI/MDI-X before Forced Speed	15	1b = Resolves MDI/MDI-X configuration before forcing speed. 0b = Does not resolve MDI/MDI-X configuration before forcing speed.	R/W	1b

1. This bit enables PHY diagnostics, which include IP phone detection and TDR cable diagnostics. It is not recommended to enable this bit in normal operation (when the link is active). This bit does not need to be set for link analysis cable diagnostics.
2. Count symbol errors (18.13) and count false carrier events (18.14) control the type of errors that the Rx error counter (20.15:0) counts (settings are shown below). The default is to count CRC errors (refer to Table 8-26).

Table 8-26 RX Error Counter Programming

Count False Carrier Events	Count Symbol Errors	Rx Error Counter
1	1	Counts symbol errors and false carrier events.
1	0	Counts CRC errors and false carrier events
0	1	Counts symbol errors.
0	0	Counts CRC errors.

Bit 9, PHY Control Register 2, manually sets the MDI/MDI-X configuration if automatic MDIX is disabled, as indicated below.

Table 8-27 MDI/MDI-X Configuration

Automatic MDI/MDI-X	MDI/MDI-X Configuration	MDI/MDI-X Mode
1	X	Automatic MDI/MDI-X detection.
0	0	MDI configuration (NIC/DTE).
0	1	MDI-X configuration (switch).

The mapping of the transmitter and receiver to pins, for MDI and MDI-X configurations for 10Base-T, 100Base-TX, and 1000Base-T is shown in Table 8-28. Note that even in manual MDI/MDI-X configuration, the PHY automatically detects and corrects for C and D pair swaps.

Table 8-28 MDI/MDI-X Pin Mapping

Pin	MDI Pin Mapping			MDI-X Pin Mapping		
	10Base-T	100Base-TX	1000Base-T	10Base-T	100Base-TX	1000Base-T
MDI_0_P/N	Transmit +/–	Transmit +/–	Transmit A+/– Receive B+/–	Receive +/–	Receive +/–	Transmit B+/– Receive A+/–
MDI_1_P/N	Receive +/–	Receive +/–	Transmit B+/– Receive A+/–	Transmit +/–	Transmit +/–	Transmit A+/– Receive B+/–
MDI_2_P/N			Transmit C+/– Receive D+/–			Transmit D+/– Receive C+/–
MDI_3_P/N			Transmit D+/– Receive C+/–			Transmit C+/– Receive D+/–

8.26.3.17 Loopback Control Register - PHLBKC (19d; R/W)

Field	Bit(s)	Description	Mode	HW Rst
Force Link Status[1]	0	1b = Forces link status okay in MII loopback. 0b = Forces link status not okay in MII loopback.	R/W	1b
Reserved	5:1	Reserved. Write 0, ignore on read.		
Tx Suppression	6	1b = Suppress Tx during all digital loopback. 0b = Do not suppress Tx during all digital loopback.	R/W	1b
External Cable	7	1b = External cable loopback enabled. 0b = External cable loopback disabled.	R/W	0b
Reserved_1	8	Reserved. Write 1, ignore on read.		1b
Remote	9	1b = Remote loopback enabled. 0b = Remote loopback disabled. ***Note:*** When bit is set to 1b, PHY does not strip 10BASE-T preamble.	R/W	0b
Line Driver	10	1b = Line driver loopback selected. 0b = Line driver loopback not selected.	R/W	0b
Reserved	11	Reserved. Write 0, ignore on read.		

Field	Bit(s)	Description	Mode	HW Rst
All Digital	12	1b = All digital loopback selected. 0b = All digital loopback not selected.	R/W	1b
Reserved	14:13	Reserved. Write 0, ignore on read.		
MII	15	1b = MII loopback selected. 0b = MII loopback not selected.	R/W	0b

1. This bit can be used to force link status okay during MII loopback. In MII loopback, the link status bit is not set unless force link status is used. In all other loopback modes, the link status bit is set when the link comes up.

8.26.3.17.1 Loopback Mode Setting

Table 8-29 shows how the loopback bit (0.14) and the LNK_EN bit (23.13) should be set for each loopback mode. It also indicates whether the loopback mode sets the link status bit and when the PHY is ready to receive data.

Table 8-29 Loopback Bit (0.14) Settings for Loopback Mode

Loopback	Bit 0.14 = 1 Loopback Required	Bit 26.6 Link Status Set	PHY Ready for Data
MII	Yes	19.0	After a few ms
All Digital	Yes	Yes	Link Status
Line Driver	Yes	Yes	Link Status
Ext Cable	No	Yes	Link Status
Remote	No	Yes	Never

8.26.3.18 RX Error Counter Register - PHRERRC (20d; RO)

Field	Bit(s)	Description	Mode	Default
Rx Error Counter[1]	15:0	16-bit Rx error counter. This register is clear-on-read.	RO, SC	0x0

1. Refer to Register 18d (bits 13 and 14) for error type descriptions.

8.26.3.19 Management Interface (MI) Control Register - PHMIC (21d; R/W)

Field	Bit(s)	Description	Mode	Default
Auto Negotiation to 1000 disable	0	Disable auto-negotiation to 1000BASE-T. ***Note:*** Bit should be 0b to allow enabling 1G operation via the *PHPM.Disable 1000* and *PHPM.Disable 1000 in non-D0a* bits.	R/W	0b
phy_in_nrg_pd	1	Energy Detect (Cable Disconnect) Status. When set it indicates that the PHY has entered the energy detect power-down mode because energy detect power-down is enabled and no energy has been detected on the line for 4s (cable disconnected).	RO	x

Field	Bit(s)	Description	Mode	Default
nrg_pd_tx_en	2	Software enable for periodic NLP transmission in energy detect power-down. 1b = Enables NLP transmission during energy-detect power down. 0b = Disables NLP transmission during energy-detect power down. ***Note:*** Bit should be 0b to allow the *PHPM.SPD_B2B_EN* bit to disable Smart Power Down Back to Back operation.	R/W	1b
Energy Detect Power Down Enable	3	1b = Enables energy detect power down. 0b = Disables energy detect power down ***Note:*** Bit should be 0b to allow the *PHPM.SPD_EN* bit to disable Smart Power Down operation.	R/W	1b
Reserved	15:4	Reserved. Write 0, ignore on read.		

8.26.3.20 PHY Configuration Register - PHCNFG (22d; R/W)

Field	Bit(s)	Description	Mode	Default
Reserved	2:0	Reserved. Write 0, ignore on read.		
Reserved_1	3	Reserved. Write 1, ignore on read.		
Reserved	4	Reserved. Write 0, ignore on read.		
Transmit Clock Enable	5	1b = Enables output of mixer clock (transmit clock in 1000Base-T). 0b = Disables output.	R/W	0b
Group MDIO Mode Enable	6	When this bit is set, the PHY processes MDIO accesses to the group address 31 as if they are accesses to it's own PHY address. 1b = Enables group MDIO mode. 0b = Disables Group MDIO mode.	R/W	0b
Alternate Next-Page	7	1b = Enables manual control of 1000Base-T next pages only. 0b = Normal operation of 1000Base-T next page exchange.	R/W	0b
Reserved	9:8	Reserved. Write 11b, ignore on read.		11b
Automatic Speed Downshift Mode[1]	11:10	00b = Automatic speed downshift disabled. 01b = 10Base-T downshift enabled. 10b = 100Base-TX downshift enabled. 11b = 100Base-TX and 10Base-T enabled.	R/W	11b
Transmit FIFO depth (1000Base-T)	13:12	00b= ±8. 01b = ±16. 10b = ±24. 11b = ±32.	R/W	00b
Ignore 10G Frames	14	1b = Management frames with ST = <00> are ignored. 0b = Management frames with ST = <00> are treated as wrong frames	R/W	1b
CRS Transmit Enable	15	1b = Enables CRS on transmit in half-duplex mode. 0b = Disables CRS on transmit.	R/W	0b

1. If automatic speed downshift is enabled and the PHY fails to auto negotiate at 1000Base-T, the PHY falls back to attempt connection at 100Base-TX and, subsequently, 10Base-T. This cycle repeats. If the link is broken at any speed, the PHY restarts this process by re-attempting connection at the highest possible speed (e.g., 1000Base-T).

8.26.3.21 PHY Control Register 1 - PHCTRL1 (23d; R/W)

Field	Bit(s)	Description	Mode	Default
Force Interrupt	0	1b = Assert PHY interrupt. 0b = De-assert PHY interrupt.	R/W	0b
Reserved	1	Reserved. Write 0, ignore on read.		
10BASE-T Preamble Length	3:2	00b = 10BASE-T preamble length of 0 octets in the received frames sent over the MII. 01b = 10BASE-T preamble length of 1 octets. 10b = 10BASE-T preamble length of 2 octets. 11b = 10BASE-T preamble length of 7 octets.	R/W	10b
10BASE-T MAU Loopback Function Enable (10BASE-T)	4	1b = Enables MAU loopback function (half-duplex only). 0b = Disables MAU loopback function.	R/W	0b
SQE Test Enable (10Base-T)	5	1b = Enables heartbeat. 0b = Disables heartbeat.	R/W	0b
Jabber Enable (10Base-T)	6	1b = Disables jabber. 0b = Normal operation.	R/W	1b
Link Partner Detected[1]	7	1b = Link partner detected. 0b = Link partner not detected	RO, LH	0b
Reverse Auto-negotiation (LPLU)	8	1b = Reverse Auto-negotiation (LCD). 0b = Normal Auto-negotiation (HCD)	R/W	0b
Reserved	9	Reserved. Write 0, ignore on read.		
Link Attempts Before Automatic Speed Downshift	12:10	000b = 1. 001b = 2. 010b = 3. 011b = 4. 100b = 5. 101b = 6. 110b = 7. 111b = 8.	R/W	100b
LNK_EN[2]	13	1b = Enables linking. 0b = Disables linking.	R/W	1b
IP Phone Detect Enable[3]	14	1b = Enables automatic IP phone detect. 0b = Disables automatic IP phone detect.	R/W, SC	0b
IP Phone Detected[4]	15	1b = IP phone detected. 0b = IP phone not detected.	RO	0b

1. When linking is disabled, the PHY automatically monitors for the appearance of a link partner and sets this bit if detected. Linking is disabled when LNK_EN is cleared (23.13 = 0).
2. If LNK_EN is set, the PHY attempts to bring up a link with a remote partner and will monitor the MDI for link pulses. If LNK_EN is cleared the PHY takes down any active link, goes into standby, and does not respond to link pulses from a remote link partner. In standby, IP phone detect and TDR functions are available.
3. When this bit is set, the PHY performs automatic IP phone detection whenever linking is disabled. Linking is disabled when LNK_EN is cleared (23.13 = 0). If an IP phone is detected it is indicated in 23.15.
4. When linking is disabled, the PHY automatically monitors for the appearance of a link partner and sets this bit if detected. Linking is disabled when LNK_EN is cleared (23.13 = 0).

8.26.3.22 Interrupt Mask Register - PHINTM (24d; R/W)

Field	Bit(s)	Description	Mode	Default
MDINT_N Enable	0	1b = PHY interrupt enabled. 0b = PHY interrupt disabled.	R/W	0b
Automatic Speed Downshift	1	1b = Interrupt enabled. 0b = Interrupt disabled.	R/W	0b
Link Status Change	2	1b = Interrupt enabled. 0b = Interrupt disabled.	R/W	0b
Receive Status Change	3	1b = Interrupt enabled. 0b = Interrupt disabled.	R/W	0b
FIFO Overflow/Underflow	4	1b = Interrupt enabled. 0b = Interrupt disabled.	R/W	0b
Error Counter Full	5	1b = Interrupt enabled. 0b = Interrupt disabled.	R/W	0b
Next Page Received	6	1b = Interrupt enabled. 0b = Interrupt disabled.	R/W	0b
CRC Errors	7	1b = Interrupt enabled. 0b = Interrupt disabled.	R/W	0b
Auto negotiation Status Change	8	1b = Interrupt enabled. 0b = Interrupt disabled.	R/W	0b
MDIO Sync Lost	9	1b = Interrupt enabled. 0b = Interrupt disabled.	R/W	0b
TDR/IP Phone	10	1b = Interrupt enabled. 0b = Interrupt disabled.	R/W	0b
Reserved	15:11	Reserved. Write 0, ignore on read.		

8.26.3.23 Interrupt Status Register - PHINT (25d; RC)

The Interrupt Status Register reports interrupt conditions detected in the internal PHY. An interrupt bit that is set and is not masked via the *PHINTM* register will assert the *ICR.GPHY* bit and generate a Host interrupt if the bit is not masked. To clear the interrupt the Host should read the *PHINT* register before clearing the *ICR.GPHY* bit.

Field	Bit(s)	Description	Mode	Default
MII Interrupt Pending[1]	0	1b = Interrupt pending. 0b = No interrupt pending.	RC, LH	0b
Automatic Speed Downshift	1	1b = Event has occurred. 0b = Event has not occurred.	RC, LH	0b
Link Status Change	2	1b = Event has occurred. 0b = Event has not occurred.	RC, LH	0b
Receive Status Change	3	1b = Event has occurred. 0b = Event has not occurred.	RC, LH	0b
FIFO Overflow/Underflow	4	1b = Event has occurred. 0b = Event has not occurred.	RC, LH	0b
Error Counter Full	5	1b = Event has occurred. 0b = Event has not occurred.	RC, LH	0b

Field	Bit(s)	Description	Mode	Default
Next Page Received	6	1b = Event has occurred. 0b = Event has not occurred.	RC, LH	0b
CRC Errors	7	1b = Event has occurred. 0b = Event has not occurred.	RC, LH	0b
Auto negotiation Status Change	8	1b = Event has occurred. 0b = Event has not occurred.	RC, LH	0b
MDIO Sync Lost[2]	9	1b = Event has occurred. 0b = Event has not occurred.	RC, LH	0b
TDR/IP Phone	10	1b = Event has occurred. 0b = Event has not occurred.	RC, LH	0b
Reserved	15:11	Reserved. Write 0, ignore on read.		

1. An event has occurred and the corresponding interrupt mask bit is enabled (set = 1).
2. If the management frame preamble is suppressed (MF preamble suppression, register 0, bit 6), it is possible for the PHY to lose synchronization if there is a glitch at the interface. The PHY can recover if a single frame with a preamble is sent to the PHY. The MDIO sync lost interrupt can be used to detect loss of synchronization and, thus, enable recovery.

8.26.3.24 PHY Status Register - PHSTAT (26d; RO)

Field	Bit(s)	Description	Mode	Default
Link partner advertised asymmetric PAUSE	0	1b = Link partner advertised asymmetric PAUSE. 0b = Link partner did not advertised asymmetric PAUSE.	RO	0b
Link partner advertised PAUSE	1	1b = Link partner advertised PAUSE. 0b = Link partner did not advertised PAUSE.	RO	0b
Auto negotiation Enabled	2	1b = Both partners have auto negotiation enabled. 0b = Both partners do not have auto negotiation enabled.	RO	0b
Collision Status	3	1b = Collision occurring. 0b = Collision not occurring.	RO	0b
Receive Status	4	1b = PHY receiving a packet. 0b = PHY not receiving a packet.	RO	0b
Transmit Status	5	1b = PHY transmitting a packet. 0b = PHY not transmitting a packet.	RO	0b
Link Status	6	1b = Link up. 0b = Link down.	RO	0b
Duplex Status	7	1b = Full duplex. 0b = Half duplex.	RO	0b
Speed Status	9:8	11b = Undetermined. 10b = 1000Base-T. 01b = 100Base-TX. 00b = 10Base-T.	RO	11b
Polarity Status	10	1b = Polarity inverted (10Base-T only). 0b = Polarity normal (10Base-T only).	RO	0b
Pair Swap on Pairs A and B	11	1b = Pairs A and B swapped. 0b = Pairs A and B not swapped.	RO	0b

Field	Bit(s)	Description	Mode	Default
Auto negotiation Status	12	1b = Auto negotiation complete. 0b = Auto negotiation not complete.	RO	0b
Auto negotiation Fault Status	14:13	11b = Reserved. 10b = Master/slave auto negotiation fault. 01b = Parallel detect auto negotiation fault. 00b = No auto negotiation fault.	RO	00b
PHY in Standby Mode[1]	15	1b = PHY in standby mode. 0b = PHY not in standby mode.	RO	0b

1. This bit indicates that the PHY is in standby mode and is ready to perform IP phone detection or TDR cable diagnostics. The PHY enters standby mode when LNK_EN is cleared (23.13 = 0) and exits standby mode and attempts to auto negotiate a link when LNK_EN is set (23.13= 1).

8.26.3.25 Diagnostics Control Register (Linking Disabled) - PHDIAG (30d; R/W)

Field	Bit(s)	Description	Mode	Default
Reserved	9:0	Reserved. Write 0, ignore on read.		
TDR Rx Dim[1]	11:10	Receive dimension for single-pair TDR analysis: 00b = TDR receive on pair A. 01b = TDR receive on pair B. 10b = TDR receive on pair C. 11b = TDR receive on pair D.	R/W	00b
TDR Tx Dim[2]	13:12	Transmit dimension for single-pair TDR analysis/first dimension to be reported for automatic TDR analysis: 00b = TDR transmit on pair A. 01b = TDR transmit on pair B. 10b = TDR transmit on pair C. 11b = TDR transmit on pair D.	R/W	00b
TDR Request[3]	15:14	11b = Automatic TDR analysis in progress. 10b = Single-pair TDR analysis in progress. 01b = TDR analysis complete, results valid. 00b = TDR analysis complete, results invalid.	R/W, SC	00b

1. The TDR receive dimension is only valid for single-pair TDR analysis. It is ignored for automatic TDR analysis when all ten pair combinations are analyzed.
2. The TDR transmit dimension is only valid for single-pair TDR analysis. For automatic TDR analysis, these bits specify the first dimension to be reported in register 31.
3. Automatic TDR analysis is enabled by setting TDR request to {11}. All ten combinations of pairs are analyzed in sequence, and the results are available in register 31. TDR analysis for a single pair combination can be enabled by setting TDR request to {10}. Linking must be disabled (23.13 = 0) and IP phone detect must be disabled (23.14 = 0) to do TDR operations. Bit 15 self-clears when the TDR operation is complete. When TDR is complete, bit 14 indicates if the results are valid.

8.26.3.26 Diagnostics Status Register (Linking Disabled) - PHDSTAT (31d; RO

)

Field	Bit(s)	Description	Mode	Default
Pair Indication[1]	1:0	00b = Results are for pair A. 01b = Results are for pair B. 10b = Results are for pair C. 11b = Results are for pair D.	RO	00b
Distance to Fault[2,3]	9:2	Distance to first open, short, or SIM fault on pair X.	RO	0x0
Short Between Pairs X and A[4]	10	1b = Short between pairs X and A. 0b = No short between pairs X and A.	RO	0b
Short Between Pairs X and B[4]	11	1b = Short between pairs X and B. 0b = No short between pairs X and B.	RO	0b
Short Between Pairs X and C[4]	12	1b = Short between pairs X and C. 0b = No short between pairs X and C.	RO	0b
Short Between Pairs X and D[4]	13	1b = Short between pairs X and D. 0b = No short between pairs X and D.	RO	0b
TDR Fault Type Pair X[5,6,7]	15:14	11b = Result invalid. 10b = Open or short found on pair X. 01b = Strong impedance mismatch found on pair X. 00b = Good termination found on pair X.	RO	11b

1. This indicates the pair to which the results in register bits 31.15:2 correspond.
2. The first time this register is read after automatic TDR analysis has completed, it indicates the distance to the first fault on pair A. The second time it is read, it indicates the distance to the first fault on pair B; the third time, on pair C; and the fourth time, on pair D. It then cycles back to pair A. Pair indication bits 31.1:0 indicate to which pair the results correspond. Bits 30.13:12 can be used to specify a pair other than pair A as the first dimension to be reported.
3. This 8-bit integer value is the distance in meters. The value 0xff indicates an unknown result.
4. The first time these bits are read after automatic TDR analysis has completed, they indicate a short between pair A and pair A, B, C, and D, respectively. The second time they are read, they indicate a short between pair B and pair A, B, C, and D, respectively. The third time, with pair C; and the fourth time, with pair D. It then cycles back to pair A. Pair indication bits 31.1:0 indicate to which pair the results correspond. Bits 30.13:12 can be used to specify a pair other than pair A as the first dimension to be reported.
5. The first time this register is read after automatic TDR analysis has completed, it indicates the fault type for pair A. The second time it is read, it indicates the fault type for pair B; the third, for pair C; and the fourth time, for pair D. It then cycles back to pair A. Pair indication bits 31.1:0 indicate pair to which the results correspond. Bits 30.13:12 can be used to specify a pair other than pair A as the first dimension to be reported.
6. A value of 01b indicates either an open or a short. If 31.13:10 = 0000b, it is an open. For all other values of 31.13:10, each bit indicates a short to pair A, B, C and D.
7. A value of 11 indicates that the results for this pair are invalid. An invalid result usually occurs when unexpected pulses are received during the TDR operation, e.g., from a remote PHY also doing TDR or trying to bring up a link. When an invalid result is indicated, the distance in bits 31.9:2 is 0xff and should be ignored.

8.26.3.27 Diagnostics Status Register (Linking Enabled) - PHDSTAT (31d; RO)

Field	Bit(s)	Description	Mode	Default
Excessive Pair Skew	0	1b = Excessive pair skew (1000BASE-T only). 0b = Not excessive pair skew (1000BASE-T only).	RO	0b
Reserved	1	Reserved. Write 0, ignore on read.		
Cable Length	9:2	Cable length when the link is active. This 8-bit integer value is the cable length in meters when the link is active. The value 0xFF indicates an unknown result.	RO	0xFF

Field	Bit(s)	Description	Mode	Default
Polarity on Pair A	10	1b = Polarity on pair A is inverted (10BASE-T or 1000BASE-T). 0b = Polarity on pair A is normal (10BASE-T or 1000BASE-T).	RO	0b
Polarity on Pair B	11	1b = Polarity on pair B is inverted (10BASE-T or 1000BASE-T). 0b = Polarity on pair B is normal (10BASE-T or 1000BASE-T).	RO	0b
Polarity on Pair C	12	1b = Polarity on pair C is inverted (1000BASE-T only). 0b = Polarity on pair C is normal (1000BASE-T only).	RO	0b
Polarity on Pair D	13	1b = Polarity on pair D is inverted (1000BASE-T only). 0b = Polarity on pair D is normal (1000BASE-T only).	RO	0b
Pair Swap on Pairs C and D[1]	14	1b = Pairs C and D are swapped (1000BASE-T only). 0b = Pairs C and D are not swapped (1000BASE-T only).	RO	0b
Reserved	15	Reserved. Write 0, ignore on read.		

1. When bit is set, the PHY detects the crossover of the received pair 2 (RJ-45 pins 4 and 5) and pair 3 (RJ-45 pins 7 and 8).

8.27 Virtual Function Device Registers

8.27.1 Queues Registers

Each VF has one queue – Q0. These queues are also used by the PF when working in non-IOV mode or if not all VFs are allocated to VMs. The mapping between the Virtual Queue number (VQn) and the Physical Queue number (PQn) is given by the equation: **PQn = VFn** (where VFn is the VF number).

For example: Q0 of VF0 is actually Q0, Q0 of VF1 are actually Q1, etc.

The virtual address of Q0 registers is always the same (RX: 0x2800, TX: 0x3800) – like the physical Q0 in the 82575 aliased area.

8.27.2 Non-Queue Registers

Non-queue registers get a virtual address that are equal to the same registers that belong to the PF. These registers are mapped to the physical address space at 0x10000, where each VM gets 0x100 bytes for its registers:

- VF0 registers: 0x10000 – 0x100FF
- VF1 registers: 0x10100 – 0x101FF
- …

8.27.2.1 EITR Registers

The I350 supports 25 EITR registers. In non IOV mode, all the EITR registers can be used by the PF. In IOV mode, 3 EITR registers are allocated to each VF and the PF should only use the remaining EITR registers. EITR0-2 registers are accessed by the VFs at addresses 0x1680 - 0x1688 and matches the PF EITRs according to the following table. EITR0 is always allocated to the PF.

VF	PF EITR	Physical Address
0	EITR22 - EITR24	0x16D8 - 0x16E0
1	EITR19 - EITR21	0x16CC - 0x16D4
2	EITR16 - EITR18	0x16C0 - 0x16C8
...		
7	EITR1 - EITR3	0x1684 - 0x168C

8.27.2.2 MSI-X Registers

The MSI-X vectors of each VF are reflected in its BAR3.

The PBA bits of the VFs are not replicated in the PF.

8.27.3 Register Set - CSR BAR

Virtual Address[1]	Physical Address Base	Abbreviation	Name	VF access	PF access
0x0000/ 0x0004	0x10000 + VFn * 0x100	VTCTRL	Control (only RST bit)	WO	WO
0x0008	0x0008 (Common – RO)	VTStatus	Status (mirror of PF status register).	RO	R/W1C
0x1048	0x1048 (common - RO)	VTFRTIMER	Free running timer (mirror of PF timer).	RO	RWS
0x1520	0x10020 + VFn * 0x100[2]	VTEICS	Extended Interrupt Cause Set Register	WO	WO
0x1524	0x10024 + VFn * 0x100[2]	VTEIMS	Extended Interrupt Mask Set/Read Register	RWS	RWS
0x1528	0x10028 + VFn * 0x100[2]	VTEIMC	Extended Interrupt Mask Clear Register	WO	WO
0x152C	0x1002C + VFn * 0x100[2]	VTEIAC	Extended Interrupt Auto Clear Register	RW	RW
0x1530	0x10030 + VFn * 0x100[2]	VTEIAM	Extended Interrupt Auto Mask Enable register	RW	RW
0x1580	0x10080 + VFn * 0x100[2]	VTEICR	Extended Interrupt Cause Set Register	RC/W1C	RC/W1C
0x1680 – 0x1688	0x16D8 - 0x16E0 - VFn * 0xC	EITR 0-2	Interrupt Throttle Registers 0-2	RW	RW
0x1700	0x10084 + VFn * 0x100[2]	VTIVAR	Interrupt vector allocation register Queues	RW	RW
0x1740	0x10088 + VFn * 0x100[2]	VTIVAR_MISC	Interrupt vector allocation register Misc.	RW	RW
0x0F04	0x5B68	PBACL	PBA clear	R/W1C	R/W1C
0x0F0C	0x5480 + VFn * 0x4	PSRTYPE	Replication Packet Split Receive Type	RW	RW
0x0C40	0x0C40 + VFn*0x4	VFMailbox	Virtual Function Mailbox	RW	RW
0x0800 – 0x083F	0x0800 – 0x083F + VFn * 0x40	VMBMEM	Virtualization Mail Box Memory	RW	RW
0x2800	0xC000,+VFn * 0x40	RDBAL0	Receive Descriptor Base Address Low	RW	RW
0x2804	0xC004+VFn * 0x40	RDBAH0	Receive Descriptor Base Address High	RW	RW
0x2808	0xC008+VFn * 0x40	RDLEN0	Receive Descriptor Length	RW	RW
0x280C	0xC00C+VFn * 0x40	SRRCTL0	Split and Replication Receive Control Register	RW	RW
0x2810	0xC010+VFn * 0x40	RDH0	Receive Descriptor Head	RW	RW
0x2814	0xC014+VFn * 0x40	RXCTL0	Rx DCA control registers	RW	RW
0x2818	0xC018+VFn * 0x40	RDT0	Receive Descriptor Tail	RW	RW

Virtual Address[1]	Physical Address Base	Abbreviation	Name	VF access	PF access
0x2828	0xC028+VFn * 0x40	RXDCTL0	Receive Descriptor Control	RW	RW
0x2830	0xC030+VFn * 0x40	RQDPC0	Receive Queue drop packet count	RO	RW
0x3800	0xE000+VFn * 0x40	TDBAL0	Transmit Descriptor Base Address Low	RW	RW
0x3804	0xE004,+VFn * 0x40	TDBAH0	Transmit Descriptor Base Address High	RW	RW
0x3808	0xE008+VFn * 0x40	TDLEN0	Transmit Descriptor Ring Length	RW	RW
0x3810	0xE010+VFn * 0x40	TDH0	Transmit Descriptor Head	RW	RW
0x3814	0xE014+VFn * 0x40	TXCTL0	Tx DCA control registers	RW	RW
0x3818	0xE018+VFn * 0x40	TDT0	Transmit Descriptor Tail	RW	RW
0x3828	0xE028+VFn * 0x40	TXDCTL0	Transmit Descriptor Control	RW	RW
0x3830	0xE030+VFn * 0x40	TQDPC0	Transmit Queue drop packet count	RO	RW
0x3838	0xE038+VFn * 0x40	TDWBAL0	Tx Descriptor Completion writeback Address Low	RW	RW
0x383C	0xE03C+VFn * 0x40	TWBAH0	Tx Descriptor Completion writeback Address High	RW	RW
0x0F10	0x10010 + VFn * 0x100	VFGPRC	Good Packets Received Count	RO	RW
0x0F14	0x10014 + VFn * 0x100	VFGPTC	Good Packets Transmitted Count	RO	RW
0x0F18	0x10018 + VFn * 0x100	VFGORC	Good Octets Received Count	RO	RW
0x0F34	0x10034 + VFn * 0x100	VFGOTC	Good Octets Transmitted Count	RO	RW
0x0F38	0x10038 + VFn * 0x100	VFMPRC	Multicast Packets Received Count	RO	RW
0x0F40	0x10040 + VFn * 0x100	VFGPRLBC	Good RX Packets loopback Count	RO	RW
0x0F44	0x10044 + VFn * 0x100	VFGPTLBC	Good TX packets loopback Count	RO	RW
0x0F48	0x10048 + VFn * 0x100	VFGORLBC	Good RX Octets loopback Count	RO	RW
0x0F50	0x10050 + VFn * 0x100	VFGOTLBC	Good TX Octets loopback Count	RO	RW
0x34E8	0x34E8	PBTWAC	Tx packet buffer wrap around counter	RO	RO
0x24E8	0x24E8	PBRWAC	Rx packet buffer wrap around counter	RO	RO

1. Addresses in **Bold** indicates registers whose virtual addresses are different from their physical addresses due to the need to maintain a virtual address space of 16KBytes.
2. Accesses through this addresses are executed as if accessed via the VF BAR.

8.27.4 Register Set - MSI-X BAR

Virtual Address	Physical Address Base (+ VFn *0x30)	Abbreviation	Name
0x0000 - 0x0020	0x0010	MSIXTADD	MSIX table entry lower address
0x0004 - 0x0024	0x0018	MSIXTUADD	MSIX table entry upper address
0x0008 - 0x0028	0x0028	MSIXTMSG	MSIX table entry message
0x000C - 0x002C	N/A	MSIXTVCTRL	MSIX table vector control
Max (Page Size, 0x2000)	N/A	MSIXPBA	MSI-X Pending bit array

8.28 Virtual Function Register Descriptions

All the registers in this section are replicated per VF. The addresses are relative to the beginning of each VF address space. The address relative to BAR0 as programmed in the IOV structure in the PF configuration space (offset 0x180-0x184) can be found by the following formula:

```
VF BAR0 + Max(16K, system page size)* VF# + CSR offset.
```

Refer to Section 8.28.49 for the list of registers exposed to the VF detailed below.

8.28.1 VT Control Register - VTCTRL (0x0000; WO)

Field	Bit(s)	Initial Value	Description
Reserved	25:0	0x0	Reserved. Write 0, ignore on read.
RST (SC)	26	0b	VF Reset This bit performs a reset of the queue enable and the interrupt registers of the VF.
Reserved	31:27	0x0	Reserved. Write 0, ignore on read.

8.28.2 VF Status Register - STATUS (0x0008; RO)

This register is a mirror of the PF status register. Refer to Section 8.2.2 for details of this register.

8.28.3 VT Free Running Timer - VTFRTIMER (0x1048; RO)

This register reflects the value of a free running timer that can be used for various timeout indications. The register is reset by a PCI reset and/or software reset. This register is a mirror of the PF register. See description of this register in Section 8.15.3.

8.28.4 VT Extended Interrupt Cause - VTEICR (0x1580; RC/W1C)

See description of this register in Section 8.8.3.

Field	Bit(s)	Initial Value	Description
MSIX	2:0	0x0	Indicates an interrupt cause mapped to MSI-X vectors 2:0
Reserved	31:3	0x0	Reserved. Write 0, ignore on read.

8.28.5 VT Extended Interrupt Cause Set - VTEICS (0x1520; WO)

See the description of this register in Section 8.8.4.

Field	Bit(s)	Initial Value	Description
MSIX	2:0	0x0	Sets to corresponding *EICR* bit of MSI-X vectors 2:0
Reserved	31:3	0x0	Reserved. Write 0, ignore on read.

8.28.6 VT Extended Interrupt Mask Set/Read - VTEIMS (0x1524; RWS)

See the description of this register in Section 8.8.5.

Field	Bit(s)	Initial Value	Description
MSIX	2:0	0x0	Set Mask bit for the corresponding *EICR* bit of MSI-X vectors 2:0
Reserved	31:3	0x0	Reserved. Write 0, ignore on read.

8.28.7 VT Extended Interrupt Mask Clear - VTEIMC (0x1528; WO)

See the description of this register in Section 8.8.6.

Field	Bit(s)	Initial Value	Description
MSIX	2:0	0x0	clear Mask bit for the corresponding *EICR* bit of MSI-X vectors 2:0
Reserved	31:3	0x0	Reserved. Write 0, ignore on read.

8.28.8 VT Extended Interrupt Auto Clear - VTEIAC (0x152C; RW)

See the description of this register in Section 8.8.7.

Field	Bit(s)	Initial Value	Description
MSIX	2:0	0x0	Auto clear bit for the corresponding *EICR* bit of MSI-X vectors 2:0
Reserved	31:3	0x0	Reserved. Write 0, ignore on read.

8.28.9 VT Extended Interrupt Auto Mask Enable - VTEIAM (0x1530; RW)

See the description of this register in Section 8.8.8.

Field	Bit(s)	Initial Value	Description
MSIX	2:0	0x0	Auto Mask bit for the corresponding *EICR* bit of MSI-X vectors 2:0
Reserved	31:3	0x0	Reserved. Write 0, ignore on read.

8.28.10 VT Interrupt Throttle - VTEITR (0x01680 + 4*n[n = 0...2]; RW)

See the description of this register in Section 8.8.14.

8.28.11 VT Interrupt Vector Allocation Registers - VTIVAR (0x1700; RW)

These registers define the allocation of the queue pair interrupt causes as defined in Table 7-49 to one of the MSI-X vectors. Each *INT_Alloc*[i] (i=0…3) field is a byte indexing an entry in the MSI-X Table Structure and MSI-X PBA Structure.

Field	Bit(s)	Initial Value	Description
INT_Alloc[0]	1:0	X	Defines the MSI-X vector assigned to the interrupt cause associated with queue 0 Rx. Valid values are 0 to 2.
Reserved	6:2	0x0	Reserved. Write 0, ignore on read.
INT_Alloc_val[0]	7	0b	Valid bit for INT_Alloc[0]
INT_Alloc[1]	9:8	X	Defines the MSI-X vector assigned to the interrupt cause associated with queue 0 Tx. Valid values are 0 to 2.
Reserved	14:10	0x0	Reserved. Write 0, ignore on read.
INT_Alloc_val[1]	15	0b	Valid bit for INT_Alloc[1]
Reserved	31:16	0x0	Reserved. Write 0, ignore on read.

8.28.12 VT Interrupt Vector Allocation Registers - VTIVAR_MISC (0x1740; RW)

This register defines the MSI-X vector allocated to the mailbox interrupt.

A mailbox interrupt is asserted in the VF upon reception of a mailbox message or an acknowledge from the PF.

Field	Bit(s)	Initial Value	Description
INT_Alloc[2]	1:0	X	Defines the MSI-X vector assigned to the interrupt cause associated with the mailbox. Valid values are 0 to 2.
Reserved	6:2	0x0	Reserved. Write 0, ignore on read.
INT_Alloc_val[2]	7	0b	Valid bit for INT_Alloc[2]
Reserved	31:8	0x0	Reserved. Write 0, ignore on read.

8.28.13 MSI—X Table Entry Lower Address - MSIXTADD (BAR3: 0x0000 + 16*n [n=0...2]; R/W)

Refer to Section 8.9.1 for information about this register.

8.28.14 MSI—X Table Entry Upper Address - SIXTUADD (BAR3: 0x0004 + 16*n [n=0...2]; R/W)

Refer to Section 8.9.2 for information about this register.

8.28.15 MSI-X Table Entry Message - MSIXTMSG (BAR3: 0x0008 + 16*n [n=0...2]; R/W)

Refer to Section 8.9.3 for information about this register.

8.28.16 MSI-X Table Entry Vector Control - MSIXTVCTRL (BAR3: 0x000C + 16*n [n=0...2]; R/W)

Refer to Section 8.9.4 for information about this register.

8.28.17 MSI-X Pending Bits - MSIXPBA (BAR3: 0x2000; RO)

Field	Bit(s)	Initial Value	Description
Pending Bits	2:0	0x0	For each pending bit that is set, the function has a pending message for the associated MSI-X Table entry. Pending bits that have no associated MSI-X table entry are reserved.
Reserved	31:3	0x0	Reserved. Write 0, ignore on read.

Note: If a page size larger than 8K is programmed in the IOV structure, the address of the MSIX PBA table moves to be page aligned.

8.28.18 MSI-X PBA Clear - PBACL (0x0F04; R/W1C)

Field	Bit(s)	Initial Value	Description
PENBIT	2:0	0x0	MSI-X Pending bits Clear Writing a 1b to any bit clears the corresponding MSIXPBA bit; writing a 0b has no effect. Reading this register returns the PBA vector.
Reserved	31:3	0x0	Reserved. Write 0, ignore on read.

8.28.19 Receive Descriptor Base Address Low - RDBAL (0x2800; RW)

Refer to Section 8.10.5 for information about this register.

8.28.20 Receive Descriptor Base Address High - RDBAH (0x2804; RW)

Refer to Section 8.10.6 for information about this register.

8.28.21 Receive Descriptor Ring Length - RDLEN (0x2808; RW)

Refer to Section 8.10.7 for information about this register.

8.28.22 Receive Descriptor Head - RDH (0x2810; RW)

Refer to Section 8.10.8 for information about this register.

8.28.23 Receive Descriptor Tail - RDT (0x2818; RW)

Refer to Section 8.10.9 for information about this register.

8.28.24 Receive Descriptor Control - RXDCTL (0x2828; RW)

Refer to Section 8.10.10 for information about this register.

8.28.25 Split and Replication Receive Control Register queue - SRRCTL(0x280C; RW)

Refer to Section 8.10.2 for information about this register.

8.28.26 Receive Queue drop packet count - RQDPC (0x2830; RO)

Refer to Section 8.10.11 for information about this register.

8.28.27 Replication Packet Split Receive Type - PSRTYPE (0x0F0C; RW)

Refer to Section 8.10.3 for information about this register.

8.28.28 Transmit Descriptor Base Address Low - TDBAL (0x3800; RW)

Refer to Section 8.12.10 for information about this register.

8.28.29 Transmit Descriptor Base Address High - TDBAH (0x3804; RW)

Refer to Section 8.12.11 for information about this register.

8.28.30 Transmit Descriptor Ring Length - TDLEN (0x3808; RW)

Refer to Section 8.12.12 for information about this register.

8.28.31 Transmit Descriptor Head - TDH (0x3810; RW)

Refer to Section 8.12.13 for information about this register.

8.28.32 Transmit Descriptor Tail - TDT (0x3818; RW)

Refer to Section 8.12.14 for information about this register.

8.28.33 Transmit Descriptor Control - TXDCTL (0x3828; RW)

Refer to Section 8.12.15 for information about this register.

8.28.34 Transmit Queue drop packet count - TQDPC (0xE030 + 0x40*n [n=0...7]; RO)

Refer to Section 8.12.18 for information about this register.

8.28.35 Tx Descriptor Completion Write–Back Address Low - TDWBAL (0x3838; RW)

Refer to Table 8.12.16 for information about this register.

8.28.36 Tx Descriptor Completion Write–Back Address High - TDWBAH (0x383C; RW)

Refer to Section 8.12.17 for information about this register.

8.28.37 Rx DCA Control Registers - RXCTL (0x2814; RW)

Refer to Section 8.13.1 for information about this register.

8.28.38 Tx DCA Control Registers - TXCTL (0x3814; RW)

Refer to Section 8.13.2 for information about this register.

8.28.39 Good Packets Received Count - VFGPRC (0x0F10; RO)

Refer to Section 8.18.77.1 for information about this register.

8.28.40 Good Packets Transmitted Count - VFGPTC (0x0F14; RO)

Refer to Section 8.18.77.2 for information about this register.

8.28.41 Good Octets Received Count - VFGORC (0x0F18; RO)

Refer to Section 8.18.77.3 for information about this register.

8.28.42 Good Octets Transmitted Count - VFGOTC (0x0F34; RO)

Refer to Section 8.18.77.4 for information about this register.

8.28.43 Multicast Packets Received Count - VFMPRC (0x0F38; RO)

Refer to Section 8.18.77.5 for information about this register.

8.28.44 Good TX Octets loopback Count - VFGOTLBC (0x0F50; RO)

Refer to Section 8.18.77.6 for information about this register.

8.28.45 Good TX packets loopback Count - VFGPTLBC (0x0F44; RO)

Refer to Section 8.18.77.7 for information about this register.

8.28.46 Good RX Octets loopback Count - VFGORLBC (0x0F48; RO)

Refer to Section 8.18.77.8 for information about this register.

8.28.47 Good RX Packets loopback Count - VFGPRLBC (0x0F40; RO)

Refer to Section 8.18.77.9 for information about this register.

8.28.48 Virtual Function Mailbox - VFMailbox (0x0C40; RW)

Refer to Section 8.14.3 for information about this register.

8.28.49 Virtualization Mailbox memory - VMBMEM (0x0800:0x083C; RW)

A 64 bytes mailbox memory for PF and VF driver communication. Locations can be accessed as 32-bit or 64-bit words.

Refer to Section 8.14.4 for information about this register.

§ §

9 PCIe* Programming Interface

9.1 PCIe Compatibility

PCIe is completely compatible with existing deployed PCI software. To achieve this, PCIe hardware implementations conform to the following requirements:

- All devices required to be supported by deployed PCI software must be enumerable as part of a tree through PCI device enumeration mechanisms.
- Devices in their default operating state must conform to PCI ordering and cache coherency rules from a software viewpoint.
- PCIe devices must conform to PCI power management specifications and must not require any register programming for PCI-compatible power management beyond those available through PCI power management capabilities registers. Power management is expected to conform to a standard PCI power management by existing PCI bus drivers.
- PCIe devices implement all registers required by the PCI specification as well as the power management registers and capability pointers specified by the PCI power management specification. In addition, PCIe defines a PCIe capability pointer to indicate support for PCIe extensions and associated capabilities.

The I350 is a multi-function device with the following functions:

- LAN 0
- LAN 1
- LAN 2
- LAN 3

All functions contain the following regions of the PCI configuration space:

- Mandatory PCI configuration registers
- Power management capabilities
- MSI and MSI-X capabilities
- PCIe extended capabilities

Different parameters affect how LAN functions are exposed on the PCIe*. Table 9-1 describes the various mapping options.

Table 9-1 I350 Function Mapping Options

Function Number	Function Description	Disable options
0 or 3	LAN 0	Strapping option
1 or 0 or 2	LAN 1	Strapping option/ *Software Defined Pins Control* (Offset 0x20) EEPROM word in LAN1 section, bit 11
2 or 1 or 0	LAN 2	Strapping option/ *Software Defined Pins Control* (Offset 0x20) EEPROM word in LAN2 section, bit 11
3 or 0	LAN 3	Strapping option/ *Software Defined Pins Control* (Offset 0x20) EEPROM word in LAN3 section, bit 11

The mapping of each port to a PCIe function is influenced by the number of functions enabled, the dummy function mode selected and the function select word in the EEPROM. See Section 4.4 for description of the function mapping when part of the functions are disabled.

9.2 Configuration Sharing Among PCI Functions

The I350 contains a single physical PCIe core interface. The I350 is designed so that each of the logical LAN ports appears as a distinct function. Many of the fields of the PCIe header space contain hardware default values that are either fixed or might be overridden by data from the EEPROM, but may not be independently specified for each function. The following fields are considered to be common to all LAN functions:

Table 9-2 Common Fields for LAN Devices

Vendor ID	The Vendor ID of the I350 can be specified via EEPROM, but only a single value can be specified. The value is reflected identically for all functions. Default value is 0x8086. The value is reflected identically for all LAN devices.
Revision	The revision number of the I350 is reflected identically for all LAN functions.
Header Type	This field indicates if a device is single function or multifunction. The value reflected in this field is reflected identically for all LAN functions, but the actual value reflected depends on LAN disable configuration. When more than one I350 LAN function is enabled, all PCIe headers return 0x80 in this field, acknowledging being part of a multi-function device. If only a single function is enabled, then a *single-function device* is indicated (this field returns a value of 0x00) and the LAN exists as device function 0 (when dummy mode is not enabled). See Table 9-7 for details.
Subsystem ID	The subsystem ID of the I350 can be specified via EEPROM, but only a single value can be specified. The value is reflected identically for all LAN functions.
Subsystem Vendor ID	The I350 subsystem vendor ID can be specified via EEPROM, but only a single value can be specified. The value is reflected identically for all LAN functions.
Cap_Ptr, Max Latency, Min Grant	These fields reflect fixed values that are constant values reflected for all LAN functions.

The following fields are implemented individually for each LAN function:

Table 9-3 Fields Implemented Differently in LAN Devices

Device ID	The device ID reflected for each LAN function can be independently specified via EEPROM.
Command, Status	Each LAN function implements its own command/status registers.
Latency Timer, Cache Line Size	Each LAN function implements these registers individually. The system should program these fields identically for each LAN to ensure consistent behavior and performance of the device.
Memory BAR, IO BAR, Expansion ROM BAR, MSI-X BAR	Each LAN function implements its own base address registers, enabling each function to claim its own address region(s). The IO BAR and Flash BAR are supported depending on BAR32 setting in the EEPROM (See Section 9.4.11).
Interrupt Pin	Each LAN function independently indicates which interrupt pin (INTA#, INTB#, INTC# or INTD#) is used by that device's MAC to signal system interrupts. The value for each LAN device can be independently specified via EEPROM, but only if more than one LAN function is enabled.
Class Code	Different class code values (iSCSI/LAN) can be set for each function.

See Section 9.6.5 for a description of the configuration space reflected to virtual functions.

9.3 PCIe Register Map

9.3.1 Register Attributes

Configuration registers are assigned one of the attributes described in the following table.

Table 9-4 Configuration Registers

Rd/Wr	Description
RO	Read-only register: Register bits are read-only and cannot be altered by software.
RW	Read-write register: Register bits are read-write and can be either set or reset.
R/W1C	Read-only status, write-1-to-clear status register, writing a 0b to R/W1C bits has no effect.
ROS	Read-only register with sticky bits: Register bits are read-only and cannot be altered by software. Bits are not cleared by reset and can only be reset with the PWRGOOD signal. Devices that consume AUX power are not allowed to reset sticky bits when AUX power consumption (either via AUX power or PME enable) is enabled.
RWS	Read-write register: Register bits are read-write and can be either set or reset by software to the desired state. Bits are not cleared by reset and can only be reset with the PWRGOOD signal. Devices that consume AUX power are not allowed to reset sticky bits when AUX power consumption (either via AUX power or PME enable) is enabled.
R/W1CS	Read-only status, write-1-to-clear status register: Register bits indicate status when read, a set bit indicating a status event can be cleared by writing a 1b. Writing a 0b to R/W1C bits has no effect. Bits are not cleared by reset and can only be reset with the PWRGOOD signal. Devices that consume AUX power are not allowed to reset sticky bits when AUX power consumption (either via AUX power or PME enable) is enabled.
HwInit	Hardware initialized: Register bits are initialized by firmware or hardware mechanisms such as pin strapping or serial EEPROM. Bits are read-only after initialization and can only be reset (for write-once by firmware) with PWRGOOD signal.
RsvdP	Reserved and preserved: Reserved for future R/W implementations; software must preserve value read for writes to bits.
RsvdZ	Reserved and zero: Reserved for future R/W1C implementations; software must use 0b for writes to bits.

The PCI configuration registers map is listed in Table 9-5. Refer to a detailed description for registers loaded from the EEPROM at initialization time. Note that initialization values of the configuration registers are marked in parenthesis.

9.3.2 PCIe Configuration Space Summary

Table 9-5 PCIe Configuration Registers Map - LAN functions

Section	Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
Mandatory PCI register	0x0	Device ID		Vendor ID	
	0x4	Status Register		Control Register	
	0x8	Class Code (0x020000/0x010000)			Revision ID
	0xC	BIST (0x00)	Header Type (0x0/ 0x80)	Latency Timer	Cache Line Size (0x10)
	0x10	Base Address Register 0			
	0x14	Base Address Register 1			
	0x18	Base Address Register 2			
	0x1C	Base Address Register 3			
	0x20	Base Address Register 4			
	0x24	Base Address Register 5			
	0x28	CardBus CIS pointer (0x0000)			
	0x2C	Subsystem Device ID		Subsystem Vendor ID	
	0x30	Expansion ROM Base Address			
	0x34	Reserved			Cap Ptr (0x40)
	0x38	Reserved			
	0x3C	Max Latency (0x00)	Min Grant (0x00)	Interrupt Pin (0x01...0x04)	Interrupt Line (0x00)
Power management capability	0x40	Power Management Capabilities		Next Pointer (0x50)	Capability ID (0x01)
	0x44	Data	Bridge Support Extensions	Power Management Control & Status	
MSI capability	0x50	Message Control (0x0080)		Next Pointer (0x70)	Capability ID (0x05)
	0x54	Message Address			
	0x58	Message Upper Address			
	0x5C	Reserved		Message Data	
	0x60	Mask bits			
	0x64	Pending bits			
MSI-X capability	0x70	Message Control (0x00090)		Next Pointer (0xA0)	Capability ID (0x11)
	0x74	Table Offset			
	0x78	PBA offset			
CSR Access Registers	0x98	IOADDR			
	0x9C	IODATA			

Table 9-5 PCIe Configuration Registers Map (Continued)- LAN functions

Section	Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
PCIe capability	0xA0	PCIe Capability Register (0x0002)		Next Pointer (0xE0)	Capability ID (0x10)
	0xA4	Device Capability			
	0xA8	Device Status		Device Control	
	0xAC	Link Capabilities			
	0xB0	Link Status		Link Control	
	0xB4	Reserved			
	0xB8	Reserved		Reserved	
	0xBC	Reserved			
	0xC0	Reserved		Reserved	
	0xC4	Device Capability 2			
	0xC8	Reserved		Device Control 2	
	0xCC	Reserved			
	0xD0	Link Status 2		Link Control 2	
	0xD4	Reserved			
	0xD8	Reserved		Reserved	
VPD capability	0xE0	VPD address		Next Pointer (0x00)	Capability ID (0x03)
	0xE4	VPD data			
AER capability	0x100	Next Capability Ptr. (0x140/0x150/0x160/ 0x1A0)	Version (0x2)	AER Capability ID (0x0001)	
	0x104	Uncorrectable Error Status			
	0x108	Uncorrectable Error Mask			
	0x10C	Uncorrectable Error Severity			
	0x110	Correctable Error Status			
	0x114	Correctable Error Mask			
	0x118	Advanced Error Capabilities and Control Register			
	0x11C: 0x128	Header Log			
Serial ID capability	0x140	Next Capability Ptr. (0x150/0x160/0x1A0)	Version (0x1)	Serial ID Capability ID (0x0003)	
	0x144	Serial Number Register (Lower Dword)			
	0x148	Serial Number Register (Upper Dword)			
ARI capability	0x150	Next Capability Ptr. (0x160/0x1A0)	Version (0x1)	ARI Capability ID (0x000E)	
	0x154	ARI Control Register		ARI Capabilities	

Table 9-5 PCIe Configuration Registers Map (Continued)- LAN functions

Section	Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
SR-IOV capability	0x160	Next Capability offset (0x1A0)	Version (0x1)	IOV Capability ID (0x0010)	
	0x164	SR IOV Capabilities			
	0x168	SR IOV Status		SR IOV Control	
	0x16C	TotalVFs (RO)		Initial VF (RO)	
	0x170	Reserved	Function Dependency Link (RO)	Num VF (RW)	
	0x174	VF Stride (RO)		First VF Offset (RO)	
	0x178	VF Device ID		Reserved	
	0x17C	Supported Page Size (0x553)			
	0x180	system page Size (RW)			
	0x184	VF BAR0 - Low (RW)			
	0x188	VF BAR0 - High (RW)			
	0x18C	VF BAR2 (RO)			
	0x190	VF BAR3 - Low (RW)			
	0x194	VF BAR3- High (RW)			
	0x198	VF BAR5 (RO)			
	0x19C	VF Migration State Array Offset (RO)			
TPH Requester capability	0x1A0	Next Capability Ptr. (0x1C0/0x1D0)	Version (0x1)	TPH Capability ID (0x17)	
	0x1A4	TPH Requester Capability Register			
	0x1A8	TPH Requester Control Register			
	0x1AC: 0x1B8	TPH Steering Table			
LTR capability	0x1C0	Next Capability Ptr. (0x1D0)	Version (0x1)	LTR Capability ID (0x18)	
	0x1C4	Maximum Non-Snooped Platform Latency Tolerance Register		Maximum Snooped Platform Latency Tolerance Register	
ACS capability	0x1D0	Next Capability Ptr. (0x000)	Version (0x1)	ACS Capability ID (0x0D)	
	0x1D4	ACS Control Register (0x0)		ACS Capability Register (0x0)	

Table 9-6 PCIe Configuration Registers Map - Dummy Function

Section	Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
Mandatory PCI register	0x0	Device ID		Vendor ID	
	0x4	Status Register		Control Register	
	0x8	Class Code (0xFF0000)			Revision ID
	0xC	BIST (0x00)	Header Type (0x0/ 0x80)	Latency Timer	Cache Line Size (0x10)
	0x10	Base Address Register 0			
	0x14	Base Address Register 1 (0x0)			
	0x18	Base Address Register 2 (0x0)			
	0x1C	Base Address Register 3 (0x0)			
	0x20	Base Address Register 4 (0x0)			
	0x24	Base Address Register 5 (0x0)			
	0x28	CardBus CIS pointer (0x0000)			
	0x2C	Subsystem Device ID		Subsystem Vendor ID	
	0x30	Expansion ROM Base Address (0x0)			
	0x34	Reserved			Cap Ptr (0x40)
	0x38	Reserved			
	0x3C	Max Latency (0x00)	Min Grant (0x00)	Interrupt Pin (0x01...0x04)	Interrupt Line (0x00)
Power management capability	0x40	Power Management Capabilities		Next Pointer (0xA0)	Capability ID (0x01)
	0x44	Data	Bridge Support Extensions	Power Management Control & Status	
PCIe capability	0xA0	PCIe Capability Register (0x0002)		Next Pointer (0x00)	Capability ID (0x10)
	0xA4	Device Capability			
	0xA8	Device Status		Device Control	
	0xAC	Link Capabilities			
	0xB0	Link Status		Link Control	
	0xB4	Reserved			
	0xB8	Reserved		Reserved	
	0xBC	Reserved			
	0xC0	Reserved		Reserved	
	0xC4	Device Capability 2			
	0xC8	Reserved		Device Control 2	
	0xCC	Reserved			
	0xD0	Link Status 2		Link Control 2	
	0xD4	Reserved			
	0xD8	Reserved		Reserved	

Table 9-6 PCIe Configuration Registers Map - Dummy Function

Section	Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
AER capability	0x100	Next Capability Ptr. (0x140/0x150/0x1C0/ 0x1D0)	Version (0x2)	AER Capability ID (0x0001)	
	0x104	Uncorrectable Error Status			
	0x108	Uncorrectable Error Mask			
	0x10C	Uncorrectable Error Severity			
	0x110	Correctable Error Status			
	0x114	Correctable Error Mask			
	0x118	Advanced Error Capabilities and Control Register			
	0x11C: 0x128	Header Log			
Serial ID capability	0x140	Next Capability Ptr. (0x150/0x1C0/1D0)	Version (0x1)	Serial ID Capability ID (0x0003)	
	0x144	Serial Number Register (Lower Dword)			
	0x148	Serial Number Register (Upper Dword)			
ARI capability	0x150	Next Capability Ptr. (0x1C0/0x1D0)	Version (0x1)	ARI Capability ID (0x000E)	
	0x154	ARI Control Register		ARI Capabilities	
LTR capability	0x1C0	Next Capability Ptr. (0x1D0)	Version (0x1)	LTR Capability ID (0x18)	
	0x1C4	Maximum Non-Snooped Platform Latency Tolerance Register		Maximum Snooped Platform Latency Tolerance Register	
ACS capability	0x1D0	Next Capability Ptr. (0x000)	Version (0x1)	ACS Capability ID (0x0D)	
	0x1D4	ACS Control Register (0x0)		ACS Capability Register (0x0)	

A description of the registers is provided in the following sections.

9.4 Mandatory PCI Configuration Registers

9.4.1 Vendor ID (0x0; RO)

This value can be loaded automatically from EEPROM address 0x0E at power up or reset. A value of 0x8086 is the default for this field at power up if the EEPROM does not respond or is not programmed. All functions are initialized to the same value.

Note: To avoid a system hang situation, if a value of 0xFFFF is read from the EEPROM, the value of the Vendor ID field defaults back to 0x8086.

9.4.2 Device ID (0x2; RO)

This is a read-only register. This field identifies individual I350 functions. It has the same default value for all LAN functions but can be auto-loaded from the EEPROM during initialization with a different value for each port. The following table describes the possible values according to the SKU and functionality of each function.

PCI Function	Default Value	EEPROM Address	Meaning
LAN 0	0x151F	0x0D 0x1D	0x151F - EEPROM-less Device ID (Default)[1] 0x1521 - 10/100/1000 Mb/s Ethernet controller, x4 PCIe, copper.[2] 0x1522 - 10/100/1000 Mb/s Ethernet controller, x4 PCIe, Fiber.[3] 0x1523 - 10/100/1000 Mb/s Ethernet controller, x4 PCIe, 1000BASE-KX/ 1000BASE-BX backplane[4]. 0x1524 - 10/100/1000 Mb/s Ethernet controller, x4 PCIe, External SGMII PHY.[5] 0x10A6 – Dummy function[6].
LAN 1	0x151F	0x8D	Same as port zero.
LAN 2	0x151F	0xCD	Same as port zero.
LAN 3	0x151F	0x10D	Same as port zero.

1. Default ID for Embedded use and eeprom-less operation.
2. *CTRL_EXT.Link_Mode* field value 00b (10/100/1000 BASE-T internal PHY mode).
3. *CTRL_EXT.Link_Mode* field value 11b (SerDes).
4. *CTRL_EXT.Link_Mode* field value either 01b (1000BASE-KX) or 11b (SerDes - 1000BASE-BX). User option to enable Clause 37 Auto-negotiation.
5. *CTRL_EXT.Link_Mode* field value 10b (SGMII).
6. The Dummy function device ID is loaded from the *Dummy Device ID* EEPROM word and is used according to the disable status of the function. It is applicable only for function 0. See section 6.2.7 for details.

9.4.3 Command Register (0x4; R/W)

This is a read/write register. Each function has its own command register. Unless explicitly specified, functionality is the same in all functions.

Bit(s)	R/W	Initial Value	Description
0	R/W[1]	0b	I/O Access Enable For LAN and Dummy functions this field is R/W.
1	R/W	0b	Memory Access Enable For LAN and Dummy functions this field is R/W.
2	R/W	0b	Bus Master Enable (BME) For LAN functions this field is R/W. For Dummy function this field is RO as zero.
3	RO	0b	Special Cycle Monitoring Hardwired to 0b.
4	RO	0b	MWI Enable Hardwired to 0b.
5	RO	0b	Palette Snoop Enable Hardwired to 0b.
6	RW	0b	Parity Error Response
7	RO	0b	Wait Cycle Enable Hardwired to 0b.

Bit(s)	R/W	Initial Value	Description
8	RW	0b	SERR# Enable
9	RO	0b	Fast Back-to-Back Enable
10	RW	0/1b	Interrupt Disable[2] For Dummy function this field is RO as one.
15:11	RO	0x0	Reserved

1. If IO_Sup bit in PCIe Init Configuration 2 EEPROM Word (0x19) is 0, I/O Access Enable bit is RO with a value of 0. In EEPROM-less mode bit is R/W.
2. The Interrupt Disable register bit is a read-write bit that controls the ability of a PCIe device to generate a legacy interrupt message. When set, devices are prevented from generating legacy interrupt messages.

9.4.4 Status Register (0x6; RO)

Each function has its own status register. Unless explicitly specified, functionality is the same in all functions.

Bits	R/W	Initial Value	Description
2:0		000b	Reserved
3	RO	0b	Interrupt Status[1]
4	RO	1b	New Capabilities Indicates that a device implements extended capabilities. The I350 sets this bit, and implements a capabilities list, to indicate that it supports PCI power management, Message Signaled Interrupts (MSI), Enhanced Message Signaled Interrupts (MSI-X), Vital Product Data (VPD), and the PCIe extensions.
5		0b	66 MHz Capable Hardwired to 0b.
6		0b	Reserved
7		0b	Fast Back-to-Back Capable Hardwired to 0b.
8	R/W1C	0b	Data Parity Reported
10:9		00b	DEVSEL Timing Hardwired to 0b.
11	R/W1C	0b	Signaled Target Abort
12	R/W1C	0b	Received Target Abort
13	R/W1C	0b	Received Master Abort
14	R/W1C	0b	Signaled System Error
15	R/W1C	0b	Detected Parity Error

1. The *Interrupt Status* field is a RO field that indicates that an interrupt message is pending internally to the device.

9.4.5 Revision (0x8; RO)

The default revision ID of the I350 is 0x01. The value of the rev ID is a logic XOR between the default value and the value in EEPROM word 0x1E. Note that all LAN functions have the same revision ID.

Note: The default value is mirrored in the *MREVID.Step_REV_ID* internal register (Section 8.6.10).

9.4.6 Class Code (0x9; RO)

The class code is a RO hard coded value that identifies the I350's functionality.

- LAN 0...LAN3 - 0x020000/0x010000 - Ethernet/SCSI Adapter[1]
- Dummy Function - 0xFF0000 - Other device.

9.4.7 Cache Line Size (0xC; R/W)

This field is implemented by PCIe devices as a read-write field for legacy compatibility purposes but has no impact on any PCIe device functionality. Field is loaded from the *PCIe Init Configuration 3* (Word 0x1A) EEPROM word and defines cache line size in Dwords. All functions are initialized to the same value. In EEPROM-less systems, the value is 0x20.

9.4.8 Latency Timer (0xD; RO)

Not used. Hardwired to zero.

9.4.9 Header Type (0xE; RO)

This indicates if a device is single function or multifunction. If a single LAN function is the only active one then this field has a value of 0x00 to indicate a single function device. If other functions are enabled then this field has a value of 0x80 to indicate a multi-function device. The following table lists the different options to set the header type field:

Table 9-7 Header Type Settings

<table>
<tr><th>LAN 0</th><th>LAN 1</th><th>LAN 2</th><th>LAN 3</th><th>Cross Mode Enable</th><th>Dummy Function Enable</th><th>Header Type Expected Value</th></tr>
<tr><td>Disabled</td><td>Disabled</td><td>Disabled</td><td>Disabled</td><td>X</td><td>X</td><td>N/A (no function)</td></tr>
<tr><td colspan="4">Only one function enabled</td><td>X</td><td>0</td><td>0x00</td></tr>
<tr><td colspan="4">More than one function is enabled</td><td>X</td><td>X</td><td>0x80 (multi function)</td></tr>
<tr><td>Disabled</td><td colspan="3">At least on of functions 1 - 3 is enabled</td><td>0</td><td>1</td><td>0x80 (dummy exist)</td></tr>
<tr><td colspan="3">At least one of functions 0 to 3 is enabled</td><td>Disabled</td><td>1</td><td>1</td><td>0x80 (dummy exist)</td></tr>
</table>

9.4.10 BIST (0xF; RO)

BIST is not supported in the I350.

1. Selected according to bit 11, 12, 13 or 14 in *Device Rev ID* EEPROM word for LAN0, LAN 1, LAN2 or LAN3 respectively.

9.4.11 Base Address Registers (0x10...0x27; R/W)

The Base Address registers (BARs) are used to map I350 register space of the various functions. The I350 has a memory BAR, IO BAR and MSI-X BAR described in Table 9-8 below.

Table 9-8 Base Address Registers Description - LAN 0...3

Mapping Windows	Mapping Description
Memory BAR	The internal registers memories and external FLASH device are accessed as direct memory mapped offsets from the Base Address register. Software can access a Dword or 64 bits. The FLASH space in this BAR is enabled by the FLSize and CSRSize fields in the BARCTRL register. Address 0 in the FLASH device is mapped to address 128K in the Memory BAR. When the usable FLASH size + CSR space is smaller than the memory BAR, then accessing addresses above the top of the FLASH wraps back to the beginning of the FLASH.
IO BAR	All internal registers and memories can be accessed using I/O operations. There are two 4-byte registers in the IO mapping window: Addr Reg and Data Reg accessible as Dword entities. IO BAR support depends on the *IO_Sup* bit in the EEPROM *"PCIe Init Configuration 2"* word.
MSI-X BAR	The MSI-X vectors and Pending bit array (PBA) structures are accessed as direct memory mapped offsets from the MSI-X BAR. Software can access Dword entities.

9.4.11.1 32-bit LAN BARs Mode Mapping

This mapping is selected when bit 10 in the *Functions Control* EEPROM word is equal to 1b.

Table 9-9 Base Address Setting in 32bit BARs Mode (*BARCTRL.BAR32* = 1b)

BAR	Addr	31 5	4	3	2	1	0
0	0x10	Memory CSR + FLASH BAR (R/W - 31:17; RO - 16:4 (0x0))		0/1	0	0	0
1	0x14	Reserved (read as all 0b's)					
2	0x18	IO BAR (R/W - 31:5)	0	0	0	0	1
3	0x1C	MSI-X BAR (R/W - 31:14; RO - 13:4 (0x0))		0/1	0	0	0
4	0x20	Reserved (read as all 0b's)					
5	0x24	Reserved (read as all 0b's)					

9.4.11.2 64-bit LAN BARs Mode Mapping

This mapping is selected when bit 10 in the *Functions Control* EEPROM word is equal to 0b.

Table 9-10 Base Address Setting in 64bit BARs Mode (*BARCTRL.BAR32* = 0b)

BAR	Addr	31 5	4	3	2	1	0
0	0x10	Memory CSR + FLASH BAR Low (RW - 31:17;RO - 16:4 (0x0))		0/1	1	0	0
1	0x14	Memory CSR + FLASH BAR High (RW)					
2	0x18	IO BAR (R/W - 31:5)	0	0	0	0	1
3	0x1C	Reserved (RO - 0)					
4	0x20	MSI-X BAR Low (RW - 31:14; RO - 13:4 (0x0))		0/1	1	0	0
5	0x24	MSI-X BAR High (RW)					

9.4.11.3 Dummy Function BARs Mapping

BAR	Addr	31 5	4	3	2	1	0
0	0x10	Dummy Memory (RO - 0x0)		0/1	0	0	0
1	0x14	Reserved (read as all 0b's)					
2	0x18	Reserved (read as all 0b's)					
3	0x1C	Reserved (read as all 0b's)					
4	0x20	Reserved (read as all 0b's)					
5	0x24	Reserved (read as all 0b's)					

9.4.11.4 Base Address Register Fields

All base address registers have the following fields.

Table 9-11 Base Address Registers' Fields

Field	Bits	R/W	Description	
Mem / IO Space Indication	0	RO	0b = Indicates memory space. 1b = Indicates I/O.	
Memory Type	2:1	RO	00b = 32-bit BAR (BAR32 in the EEPROM equals 1b) 10b = 64-bit BAR (BAR32 in the EEPROM equals 0b)	
Prefetch Memory	3	RO	0b = Non-prefetchable space. 1b = Prefetchable space. This bit should be set only on systems that do not generate prefetchable cycles. This bit is loaded from the PREFBAR bit in the EEPROM.	
Address Space (Low register for 64bit Memory BARs)	31:4	R/W	The length of the RW bits and RO 0b bits depend on the mapping window sizes. Init value of the RW fields is 0x0.	
			Mapping Window	RO bits
			Memory CSR + FLASH BAR size depends on BARCTRL.FLSize and BARCTRL.CSRSize fields.	16:4 for 128KB 17:4 for 256KB and so on...
			MSI-X space is 16KB	13:4
			I/O spaces size is 32 bytes	4

9.4.12 CardBus CIS (0x28; RO)

Not used. Hardwired to zero.

9.4.13 Subsystem Vendor ID (0x2C; RO)

This value can be loaded automatically from EEPROM address 0x0C at power up or reset. A value of 0x8086 is the default for this field at power up if the EEPROM does not respond or is not programmed. All functions are initialized to the same value.

9.4.14 Subsystem ID (0x2E; RO)

This value can be loaded automatically from EEPROM address 0x0B at power up with a default value of 0x0000.

9.4.15 Expansion ROM Base Address (0x30; RW)

This register is used to define the address and size information for boot-time access to the optional Flash memory. Expansion ROM is enabled by placing 0b in the *LAN Boot Disable* EEPROM bit for LAN 0, LAN1, LAN2 and LAN 3, respectively. This register returns a zero value for functions without an expansion ROM window.

Note: For Dummy functions this register is Read Only with a value of zero.

Field	Bit(s)	R/W	Initial Value	Description
En	0	R/W	0b	1b = Enables expansion ROM access. 0b = Disables expansion ROM access.
Reserved	10:1	RO	0b	Always read as 0b. Writes are ignored.
Address	31:11	R/W	0b	Read-write bits are hard wired to 0b and dependent on the memory mapping window size. The LAN Expansion ROM spaces can be either 64 KB or up to 8 MB in powers of 2. Mapping window size is set by the *Flash Size* EEPROM field.

9.4.16 Cap_Ptr (0x34; RO)

The *Capabilities Pointer* field (Cap_Ptr) is an 8-bit field that provides an offset in the device's PCI configuration space for the location of the first item in the Capabilities Linked List (CLL). The I350 sets this bit and implements a capabilities list to indicate that it supports PCI power management, Message Signaled Interrupts (MSIs), and PCIe extended capabilities. Its value is 0x40, which is the address of the first entry: PCI power management.

9.4.17 Interrupt Line (0x3C; RW)

Read/write register programmed by software to indicate which of the system interrupt request lines this I350's interrupt pin is bound to. See the PCIe definition for more details. Each of the PCIe functions has its own register.

For Dummy functions this register is RO - zero.

9.4.18 Interrupt Pin (0x3D; RO)

Read only register.

- LAN 0 / LAN 1/LAN2/ LAN3 [1] - A value of 0x1, 0x2, 0x3 or 0x4 indicates that this function implements legacy interrupt on INTA#, INTB#, INTC# or INTD#, respectively. Value is loaded from

1. If only a single device/function of the I350 component is enabled, this value is ignored and the *Interrupt Pin* field of the enabled device reports INTA# usage.

Initialization Control 3 (Offset 0x24) EEPROM words from relevant LAN 0, LAN 1, LAN2 and LAN3 EEPROM sections.

Note: If only a single port is enabled while the other ports are disabled, the enabled port uses INTA, independent of the EEPROM setting.

9.4.19 Max_Lat/Min_Gnt (0x3E; RO)

Not used. Hardwired to zero.

9.5 PCI Capabilities

The first entry of the PCI capabilities link list is pointed by the Cap_Ptr register. The following tables describes the capabilities supported by the I350.

Table 9-12 PCI capabilities for LAN functions

Address	Item	Next Pointer
0x40-47	PCI Power Management	0x50
0x50-67	Message Signaled Interrupt	0x70
0x70-8B	Extended Message Signaled Interrupt	0xA0
0xA0-DB	PCIe Capabilities	0xE0/0x00[1]
0xE0-0xE7	Vital Product Data Capability	0x00

1. Next pointer is 0x00 if the VPD area in the EEPROM does not exist. In EEPROM-less mode, the PCIe capability is the last capabilities section.

Table 9-13 PCI capabilities for Dummy function

Address	Item	Next Pointer
0x40-47	PCI Power Management	0x50
0xA0-DB	PCIe Capabilities	0x00

9.5.1 PCI Power Management Capability

All fields are reset on full power-up. All of the fields except *PME_En* and *PME_Status* are reset on exit from D3cold state. If aux power is not supplied, the *PME_En* and *PME_Status* fields also reset on exit from D3cold state.

See the detailed description for registers loaded from the EEPROM at initialization time. Behavior of some fields in this section depend on the *Power Management* bit in EEPROM word 0x0A.

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
0x40	Power Management Capabilities		Next Pointer (0x50/ 0xA0)	Capability ID (0x01)
0x44	Data	Bridge Support Extensions	Power Management Control & Status	

9.5.1.1 Capability ID (0x40; RO)

This field equals 0x01 indicating the linked list item as being the PCI Power Management registers.

9.5.1.2 Next Pointer (0x41; RO)

This field provides an offset to the next capability item in the capability list. In LAN function, a value of 0x50 points to the MSI capability. In the dummy function, a value of 0xA0 points to the PCI Express capability.

9.5.1.3 Power Management Capabilities - PMC (0x42; RO)

This field describes the I350's functionality at the power management states as described in the following table. Note that each device function has its own register.

Bits	Default	R/W	Description
15:11	01001b See value in description column	RO	PME_Support - This 5-bit field indicates the power states in which the function may assert PME#. A value of 0b for any bit indicates that the function is not capable of asserting the PME# signal while in that power state. bit(11) X XXX1b - PME# can be asserted from D0 bit(12) X XX1Xb - PME# can be asserted from D1 bit(13) X X1XXb - PME# can be asserted from D2 bit(14) X 1XXXb - PME# can be asserted from D3hot bit(15) 1 XXXXb - PME# can be asserted from D3cold Value of bit 15 is a function of Aux Pwr availability and *Power Management* (PM Ena) bit in *Initialization Control Word 1* (word 0x0A) EEPROM word. Condition / Functionality / Value PM Dis in EEProm / No PME at all states / 00000b PM Ena & NoAux Pwr / PME at D0 and D3hot / 01001b PM Ena & Aux Pwr / PME at D0, D3hot and D3cold / 11001b Note: Aux Pwr is considered available if AUX_PWR pin is connected to 3.3V and *D3COLD_WAKEUP_ADVEN* EEPROM bit is set to 1b. For Dummy function, this field is RO - zero.
10	0b	RO	D2_Support The I350 does not support D2 state.
9	0b	RO	D1_Support The I350 does not support D1 state.
8:6	000b	RO	AUX Current – Required current defined in the Data Register.
5	1b	RO	DSI The I350 requires its device driver to be executed following transition to the D0 uninitialized state.
4	0b	RO	Reserved
3	0b	RO	PME_Clock Disabled. Hardwired to 0b.
2:0	011b	RO	Version The I350 complies with the PCI PM specification, revision 1.2.

9.5.1.4 Power Management Control / Status Register - PMCSR (0x44; R/W)

This register is used to control and monitor power management events in the I350. Note that each device function has its own *PMCSR* register.

Bits	Default	R/W	Description
15	0b (at power up)	R/W1CS	PME_Status This bit is set to 1b when the function detects a wake-up event independent of the state of the *PME_En* bit. Writing a 1b clears this bit.
14:13	01b	RO	Data_Scale This field indicates the scaling factor to be used when interpreting the value of the Data register. This field equals 01b (indicating 0.1 watt units) if power management is enabled in the *Power Management* (PM Ena) bit in *Initialization Control Word 1* (word 0x0A) EEPROM word and the *Data_Select* field is set to 0, 3, 4, 7, (or 8 for Function 0). Otherwise, this field equals 00b.
12:9	0000b	R/W	Data_Select This four-bit field is used to select which data is to be reported through the Data register and *Data_Scale* field. These bits are writable only when power management is enabled by setting the *Power Management* (PM Ena) bit in *Initialization Control Word 1* (word 0x0A) EEPROM word.
8	0b (at power up)	R/WS	PME_En If power management is enabled in the EEPROM, writing a 1b to this register enables wake up. If power management is disabled in the EEPROM, writing a 1b to this bit has no effect and does not set the bit to 1b.
7:4	000000b	RO	Reserved
3	1b[1]	RO	No_Soft_Reset No_Soft_Reset - When set (“1”), this bit indicates that when the I350 transitions from D3hot to D0 because of modifying *Power State* bits in the *PMCSR* register, no internal reset is issued and Configuration Context is preserved. Upon transition from the D3hot to the D0 Initialized state, no additional operating system intervention is required to preserve Configuration Context beyond writing the *Power State* bits. When clear (“0”), the I350 performs an internal reset upon transitioning from D3hot to D0 via software control of the *Power State* bits in the *PMCSR* register. Configuration Context is lost when performing the soft reset. Upon transition from the D3hot to the D0 state, full re initialization sequence is needed to return the device to D0 Initialized. Regardless of this bit, devices that transition from D3hot to D0 by a system or bus segment reset will return to the device state D0 Uninitialized with only PME context preserved if PME is supported and enabled.
2	0b	RO	Reserved for PCIe.
1:0	00b	R/W	Power State This field is used to set and report the power state of a function as follows: 00b = D0 01b = D1 (cycle ignored if written with this value) 10b = D2 (cycle ignored if written with this value) 11b = D3 (cycle ignored if power management is not enabled in the EEPROM)

1. Loaded from EEPROM (See Section 6.2.17).

9.5.1.5 Bridge Support Extensions - PMCSR_BSE (0x46; RO)

This register is not implemented in the I350. Values are set to 0x00.

9.5.1.6 Data Register (0x47; RO)

This optional register is used to report power consumption and heat dissipation. Reported register is controlled by the *Data_Select* field in the PMCSR and the power scale is reported in the *Data_Scale* field in the PMCSR. The data of this field is loaded from the EEPROM if power management is enabled in the EEPROM or with a default value of 0x00. The values for the I350 functions are read from EEPROM word 0x22.

Function	D0 (Consume/ Dissipate)	D3 (Consume/ Dissipate)	Common
PMCSR.Data Select	0x0 / 0x4	0x3 / 0x7	0x8
Function 0	EEPROM addr 0x22	EEPROM addr 0x22	EEPROM addr 0x22
Functions 1 - 3	EEPROM addr 0x22	EEPROM addr 0x22	0x00

For other *Data_Select* values, the Data register output is reserved (0x0).

9.5.2 MSI Configuration

This capability is not available for Dummy functions.

This structure is required for PCIe devices.

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
0x50	Message Control (0x0180)		Next Pointer (0x70)	Capability ID (0x05)
0x54	Message Address			
0x58	Message Upper Address			
0x5C	Reserved		Message Data	
0x60	Mask bits			
0x64	Pending bits			

9.5.2.1 Capability ID (0x50; RO)

This field equals 0x05 indicating the linked list item as being the MSI registers.

9.5.2.2 Next Pointer (0x51; RO)

This field provides an offset to the next capability item in the capability list. Its value of 0x70 points to the MSI-X capability structure.

9.5.2.3 Message Control (0x52; R/W)

The register fields are described in the following table. There is a dedicated register per PCI function to separately enable their MSI.

Bits	Default	R/W	Description
0	0b	R/W	MSI Enable If set to 1b, equals MSI. In this case, the I350 generates an MSI for interrupt assertion instead of INTx signaling.
3:1	000b	RO	Multiple Message Capable The I350 indicates a single requested message per each function.
6:4	000b	RO	Multiple Message Enable The I350 returns 000b to indicate that it supports a single message per function.
7	1b	RO	64-bit capable A value of 1b indicates that the I350 is capable of generating 64-bit message addresses.
8	1b[1]	RO	MSI per-vector masking. A value of 1b indicates that the I350 is capable of per-vector masking. This field is loaded from the *MSI-X Configuration* (Offset 0x16) EEPROM word.
15:9	0b	RO	Reserved Write 0 ignore on read.

1. Default value is read from the EEPROM

9.5.2.4 Message Address Low (0x54; R/W)

Written by the system to indicate the lower 32 bits of the address to use for the MSI memory write transaction. The lower two bits always return 0b regardless of the write operation.

9.5.2.5 Message Address High (0x58; R/W)

Written by the system to indicate the upper 32-bits of the address to use for the MSI memory write transaction.

9.5.2.6 Message Data (0x5C; R/W)

Written by the system to indicate the lower 16 bits of the data written in the MSI memory write Dword transaction. The upper 16 bits of the transaction are written as 0b.

9.5.2.7 Mask Bits (0x60; R/W)

The Mask Bits and Pending Bits registers enable software to disable or defer message sending on a per-vector basis. As the I350 supports only one message, only bit 0 of these register is implemented.

Bits	Default	R/W	Description
0	0b	R/W	MSI Vector 0 Mask If set, the I350 is prohibited from sending MSI messages.
31:1	000b	RO	Reserved

9.5.2.8 Pending Bits (0x64; R/W)

Bits	Default	R/W	Description
0	0b	RO	If set, the I350 has a pending MSI message.
31:1	000b	RO	Reserved

9.5.3 MSI-X Configuration

More than one MSI-X capability structure per function is prohibited, but a function is permitted to have both an MSI and an MSI-X capability structure.

In contrast to the MSI capability structure, which directly contains all of the control/status information for the function's vectors, the MSI-X capability structure instead points to an MSI-X table structure and a MSI-X Pending Bit Array (PBA) structure, each residing in memory space.

Each structure is mapped by a Base Address Register (BAR) belonging to the function, located beginning at 0x10 in configuration space. A BAR Indicator Register (BIR) indicates which BAR, and a Qword-aligned offset indicates where the structure begins relative to the base address associated with the BAR. The BAR is permitted to be either 32-bit or 64-bit, but must map to memory space. A function is permitted to map both structures with the same BAR, or to map each structure with a different BAR.

The MSI-X table structure, listed in Section 8.9, typically contains multiple entries, each consisting of several fields: message address, message upper address, message data, and vector control. Each entry is capable of specifying a unique vector.

The PBA structure, described in the same section, contains the function's pending bits, one per Table entry, organized as a packed array of bits within Qwords. Note that the last Qword might not be fully populated.

To request service using a given MSI-X table entry, a function performs a Dword memory write transaction using:

- The contents of the Message Data field entry for data.
- The contents of the Message Upper Address field for the upper 32 bits of the address.
- The contents of the Message Address field entry for the lower 32 bits of the address.

A memory read transaction from the address targeted by the MSI-X message produces undefined results.

The MSI-X table and MSI-X PBA are permitted to co-reside within a naturally aligned 4 KB address range, though they must not overlap with each other.

MSI-X table entries and Pending bits are each numbered 0 through N-1, where N-1 is indicated by the Table Size field in the MSI-X Message Control register. For a given arbitrary MSI-X table entry K, its starting address can be calculated with the formula:

```
Entry starting address = Table base + K*16
```

For the associated Pending bit K, its address for Qword access and bit number within that Qword can be calculated with the formulas:

```
Qword address = PBA base + (K div 64)*8
Qword bit# = K mod 64
```

Software that chooses to read Pending bit K with Dword accesses can use these formulas:

```
Dword address = PBA base + (K div 32)*4
Dword bit# = K mod 32
```

The I350 also supports the table-less MSI-X mode, where a single interrupt vector is provided. The MSI-X table and MSI-X PBA are not used. Instead, the capability structure includes several additional fields (Message Address, Message Address Upper, and Message Data) for vector configuration. The I350 embeds the number of the original MSI-X vectors (i.e. the vectors supported if the number of vectors was not limited to 1) in the LSB bits of the Message Data field.

Table 9-14 MSI-X capability Structure

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
0x70	Message Control (0x00090)		Next Pointer (0xA0)	Capability ID (0x11)
0x74	Table Offset			
0x78	PBA offset			

This capability is not available for Dummy functions.

9.5.3.1 Capability ID (0x70; RO)

This field equals 0x11 indicating the linked list item as being the MSI-X registers.

9.5.3.2 Next Pointer (0x71; RO)

This field provides an offset to the next capability item in the capability list. Its value of 0xA0 points to the PCIe capability.

9.5.3.3 Message Control (0x72; R/W)

The register fields are described in the following table. There is a dedicated register per PCI function to separately configure their MSI-X functionality.

Bits	Default	R/W	Description
10:0	0x009[1]	RO	TS - Table Size System software reads this field to determine the MSI-X Table Size N, which is encoded as N-1. For example, a returned value of 0x00F indicates a table size of 16. The I350 supports 10 MSI-X vectors. This field is loaded from the *MSI-X Configuration* (Offset 0x16) EEPROM word.
13:11	000b	RO	Reserved Always return 000b on read. Write operation has no effect.
14	0b	R/W	FM - Function Mask If set to 1b, all of the vectors associated with the function are masked, regardless of their per-vector *Mask* bit states. If set to 0b, each vector's *Mask* bit determines whether the vector is masked or not. Setting or clearing the *MSI-X Function Mask* bit has no effect on the state of the per-vector *Mask* bits.

Bits	Default	R/W	Description
15	0b	R/W	En - MSI-X Enable If set to 1b and the *MSI Enable* bit in the MSI Message Control (MMC) register is 0b, the function is permitted to use MSI-X to request service and is prohibited from using its INTx# pin. System configuration software sets this bit to enable MSI-X. A software device driver is prohibited from writing this bit to mask a function's service request. If set to 0b, the function is prohibited from using MSI-X to request service.

1. Default value is read from the EEPROM

9.5.3.4 MSI-X Table Offset (0x74; R/W)

Bits	Default	Type	Description
31:3	0x000	RO	Table Offset Used as an offset from the address contained by one of the function's BARs to point to the base of the MSI-X table. The lower three table BIR bits are masked off (set to zero) by software to form a 32-bit Qword-aligned offset.
2:0	0x3/0x4	RO	Table BIR Indicates which one of a function's BARs, located beginning at 0x10 in configuration space, is used to map the function's MSI-X table into memory space. BIR values: 0...5 correspond to BARs 0x10...0x 24 respectively. A BIR value of 3 indicates that the table is mapped in BAR 3 (address 0x1C). When *BARCTRL.BAR32* equals 0b (64 bit MMIO mapping) the table BIR equals 0x4. When *BARCTRL.BAR32* equals 1b (32 bit MMIO mapping) the table BIR equals 0x3.

9.5.3.5 MSI-X Pending Bit Array - PBA Offset (0x78; R/W)

Bits	Default	Type	Description
31:3	0x400	RO	PBA Offset Used as an offset from the address contained by one of the function's BARs to point to the base of the MSI-X PBA. The lower three PBA BIR bits are masked off (set to zero) by software to form a 32-bit Qword-aligned offset.
2:0	0x3	RO	PBA BIR: Indicates which one of a function's Base Address registers, located beginning at 10h in Configuration Space, is used to map the function's MSI-X PBA into Memory Space. BIR values: 0...5 correspond to BARs 0x10...0x 24 respectively. A BIR value of 3 indicates that the table is mapped in BAR 3 (address 0x1C). When *BARCTRL.BAR32* equals 0b (64 bit MMIO mapping) the table BIR equals 0x4. When *BARCTRL.BAR32* equals 1b (32 bit MMIO mapping) the table BIR equals 0x3.

9.5.4 CSR Access Via Configuration Address Space

These registers are not available for Dummy functions.

9.5.4.1 IOADDR Register (0x98; R/W)

This is a read/write register. Each function has its own *IOADDR* register. Functionality is the same in all functions. Register is cleared at Power-up or PCIe reset.

Note: When function is in D3 state Software should not attempt to access CSRs via the *IOADDR* and *IODATA* registers.

Bit(s)	R/W	Initial Value	Description
30:0	R/W[1]	0x0	Internal Register or Internal Memory location Address. 0x00000-0x1FFFF – Internal Registers and Memories 0x20000-0x7FFFFFFF – Undefined
31	R/W	0b	Configuration IO Access Enable. 0b - CSR configuration read or write disabled. 1b - CSR Configuration read or write enabled When bit is set accesses to the IODATA register actually generate transactions to the device. Otherwise, accesses to the IODATA register are don't-cares (write are discarded silently, reads return arbitrary results).

1. In the event that the *CSR_conf_en* bit in the *PCIe Init Configuration 2* EEPROM word is cleared, accesses to the *IOADDR* register via configuration address space is ignored and has no effect on the register and the CSRs referenced by the *IOADDR* register.

9.5.4.2 IODATA Register (0x9C; R/W)

This is a read/write register. Each function has its own *IODATA* register. Functionality is the same in all functions. Register is cleared at Power-up or PCIe reset.

Bit(s)	R/W	Initial Value	Description
31:0	R/W[1]	0x0	Data field for reads or writes to the Internal register or internal memory location as identified by the current value in IOADDR. All 32 bits of this register are read/write-able.

1. In the event that the *CSR_conf_en* bit in the *PCIe Init Configuration 2* EEPROM word is cleared, access to the *IODATA* register via configuration address space is ignored and has no effect on the register and the CSRs referenced by the *IOADDR* register.

9.5.5 Vital Product Data Registers

This capability is not available for Dummy functions.

The I350 supports access to a VPD structure stored in the EEPROM using the following set of registers.

Note: The VPD structure is available through all port functions. As the interface is common to all functions, accessing the VPD structure of one function while an access to the EEPROM is in process on another function can yield unexpected results.

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
0xE0	VPD address		Next Pointer (0x00)	Capability ID (0x03)
0xE4	VPD data			

9.5.5.1 Capability ID (0xE0; RO)

This field equals 0x3 indicating the linked list item as being the VPD registers.

9.5.5.2 Next Pointer (0xE1; RO)

Offset to the next capability item in the capability list. A 0x00 value indicates that it is the last item in the capability-linked list.

9.5.5.3 VPD Address (0xE2; RW)

Dword-aligned byte address of the VPD area in the EEPROM to be accessed. The register is read/write with the initial value at power-up indeterminate.

Bits	Default	R/W	Description
14:0	X	RW	Address Dword-aligned byte address of the VPD area in the EEPROM to be accessed. The register is read/write with the initial value at power-up indeterminate. The two LSBs are RO as zero. This is the address relative to the start of the VPD area. As the maximal size supported by the I350 is 256 bytes, bits 14:8 should always be zero.
15	0b	RW	A flag used to indicate when the transfer of data between the VPD Data register and the storage component completes. The Flag register is written when the VPD Address register is written. 0b = Read. Set by hardware when data is valid. 1b = Write. Cleared by hardware when data is written to the EEPROM. The VPD address and data should not be modified before the action completes.

9.5.5.4 VPD Data (0xE4; RW)

This register contains the VPD read/write data.

Bits	Default	R/W	Description
31:0	X	RW	VPD Data VPD data can be read or written through this register. The LSB of this register (at offset four in this capability structure) corresponds to the byte of VPD at the address specified by the VPD Address register. The data read from or written to this register uses the normal PCI byte transfer capabilities. Four bytes are always transferred between this register and the VPD storage component. Reading or writing data outside of the VPD space in the storage component is not allowed. In a write access, the data should be set before the address and the flag is set.

9.5.6 PCIe Configuration Registers

PCIe provides two mechanisms to support native features:

- PCIe defines a PCI capability pointer indicating support for PCIe.
- PCIe extends the configuration space beyond the 256 bytes available for PCI to 4096 bytes.

The I350 implements the PCIe capability structure for endpoint devices as follows:

9.5.6.1 Capability ID (0xA0; RO)

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
0xA0	PCI Express Capability Register (0x0002)		Next Pointer (0xE0/ 0x00)	Capability ID (0x10)
0xA4	Device Capability			
0xA8	Device Status		Device Control	
0xAC	Link Capabilities			
0xB0	Link Status		Link Control	
0xB4	Reserved			
0xB8	Reserved		Reserved	
0xBC	Reserved			
0xC0	Reserved		Reserved	
0xC4	Device Capabilities 2			
0xC8	Reserved		Device Control 2	
0xCC	Reserved			
0xD0	Link Status 2		Link Control 2	
0xD4	Reserved			
0xD8	Reserved		Reserved	

This field equals 0x10 indicating the linked list item as being the PCIe Capabilities registers.

9.5.6.2 Next Pointer (0xA1; RO)

Offset to the next capability item in the capability list. Its value of 0xE0 points to the VPD structure. If VPD is disabled, operation is EEPROM-less or function is a dummy function, a value of 0x00 value indicates that it is the last item in the capability-linked list.

9.5.6.3 PCIe CAP (0xA2; RO)

The PCIe capabilities register identifies the PCIe device type and associated capabilities. This is a read only register identical to all functions.

Bits	Default	R/W	Description
3:0	0010b	RO	Capability Version Indicates the PCIe capability structure version number. The I350 supports both version 1 and version 2 as loaded from the PCIe *Capability Version* bit in the EEPROM.
7:4	0000b	RO	Device/Port Type Indicates the type of PCIe functions. All functions are a native PCI function with a value of 0000b.
8	0b	RO	Slot Implemented The I350 does not implement slot options therefore this field is hardwired to 0b.
13:9	00000b	RO	Interrupt Message Number The I350 does not implement multiple MSI interrupts per function, therefore this field is hardwired to 0x0.
15:14	00b	RO	Reserved

9.5.6.4 Device Capabilities (0xA4; RO)

This register identifies the PCIe device specific capabilities. It is a read only register with the same value for all functions.

Bits	R/W	Default	Description
2:0	RO	010b	Max Payload Size Supported This field indicates the maximum payload that the I350 can support for TLPs. It is loaded from the EEPROM's *PCIe Init Configuration 3* word, 0x1A (with a default value of 512 bytes.
4:3	RO	00b	Phantom Function Supported Not supported by the I350.
5	RO	0b	Extended Tag Field Supported Max supported size of the *Tag* field. The I350 supported 5-bit *Tag* field for all functions.
8:6	RO	011b	Endpoint L0s Acceptable Latency This field indicates the acceptable latency that the I350 can withstand due to the transition from the L0s state to the L0 state. All functions share the same value loaded from the EEPROM *PCIe Init Configuration 1* word, 0x18 (See Section 6.2.14).
11:9	RO	110b	Endpoint L1 Acceptable Latency This field indicates the acceptable latency that the I350 can withstand due to the transition from the L1 state to the L0 state. All functions share the same value loaded from the EEPROM PCIe L1 Exit latencies word, 0x14 (See Section 6.2.11).
12	RO	0b	Attention Button Present Hardwired in the I350 to 0b for all functions.
13	RO	0b	Attention Indicator Present Hardwired in the I350 to 0b for all functions.
14	RO	0b	Power Indicator Present Hardwired in the I350 to 0b for all functions.
15	RO	1b	Role-Based Error Reporting This bit, when set, indicates that the I350 implements the functionality originally defined in the Error Reporting ECN for PCIe Base Specification 1.0a and later incorporated into PCIe Base Specification 1.1. Set to 1b in the I350.
17:16	RO	000b	Reserved
25:18	RO	0x00	Slot Power Limit Value Hardwired in the I350 to 0x00 for all functions, as the I350 consumes less than the 25W allowed for its form factor.
27:26	RO	00b	Slot Power Limit Scale Hardwired in the I350 to 0 for all functions, as the I350 consumes less than the 25W allowed for its form factor.
28	RO	1b[1]	Function Level Reset (FLR) Capability A value of 1b indicates the function supports the optional FLR mechanism.
31:29	RO	000b	Reserved

1. Loaded from EEPROM

9.5.6.5 Device Control (0xA8; RW)

This register controls the PCIe specific parameters. There is a dedicated register per each function.

Bits	R/W	Default	Description
0	RW	0b	Correctable Error Reporting Enable Enable report of correctable errors.
1	RW	0b	Non-Fatal Error Reporting Enable Enable report of non fatal errors.
2	RW	0b	Fatal Error Reporting Enable Enable report of fatal errors.
3	RW	0b	Unsupported Request Reporting Enable Enable report of unsupported requests error.
4	RW	1b	Enable Relaxed Ordering If this bit is set, the I350 is permitted to set the *Relaxed Ordering* bit in the attribute field of write transactions that do not need strong ordering. For more details, refer to the description about the RO_DIS bit in the CTRL_EXT register bit in See section 8.2.3 .
7:5	RW	000b (128 bytes)	Max Payload Size This field sets maximum TLP payload size for the I350 functions. As a receiver, the I350 must handle TLPs as large as the set value. As a transmitter, the I350 must not generate TLPs exceeding the set value. The max payload size supported in I350 Device capabilities register indicates permissible values that can be programmed. When ARI support is exposed, the value set in function zero (even when it is a dummy function), is the value used for all the functions transactions.
8	R0	0b	Extended Tag field Enable Not implemented in the I350.
9	R0	0b	Phantom Functions Enable Not implemented in the I350.
10	RWS	0b	Auxiliary Power PM Enable When set, enables the I350 to draw AUX power independent of PME AUX power. The I350 is a multi function device, therefore it is allowed to draw AUX power if at least one of the functions has this bit set.
11	RW	1b	Enable No Snoop Snoop is gated by *NONSNOOP* bits in the GCR register in the CSR space.
14:12	RW	010b	Max Read Request Size - this field sets maximum read request size for the Device as a requester. 000b = 128 bytes. 001b = 256 bytes. 010b = 512 bytes (the default value). 011b = 1 KB. 100b = 2 KB. 101b = Reserved. 110b = Reserved. 111b = Reserved.
15	RW	0b	Initiate Function Level Reset A write of 1b initiates an FLR to the function. The value read by software from this bit is always 0b.

9.5.6.6 Device Status (0xAA; R/W1C)

This register provides information about PCIe device's specific parameters. There is a dedicated register per each function.

Bits	R/W	Default	Description
0	R/W1C	0b	Correctable Error Detected Indicates status of correctable error detection.
1	R/W1C	0b	Non-Fatal Error Detected Indicates status of non-fatal error detection.
2	R/W1C	0b	Fatal Error Detected Indicates status of fatal error detection.
3	R/W1C	0b	Unsupported Request Detected Indicates that the I350 received an unsupported request. This field is identical in all functions. The I350 cannot distinguish which function caused an error.
4	RO	0b	Aux Power Detected If aux power is detected, this field is set to 1b. It is a strapping signal from the periphery identical for all functions. Reset on LAN_PWR_GOOD and GIO Power Good only.
5	RO	0b	Transactions Pending Indicates whether the I350 has any transaction pending.
15:6	RO	0x00	Reserved

9.5.6.7 Link Capabilities Register (0xAC; RO)

This register identifies PCIe link specific capabilities. This is a read only register identical to all functions.

Bits	Rd/Wr	Default	Description
3:0	RO	0010b	Max Link Speed This field indicates the supported Link speed(s) of the associated link port. Defined encodings are: 0001b = 2.5 Gb/s Link speed supported. 0010b = 5 Gb/s and 2.5 Gb/s Link speeds supported. Value of this field is determined by the *Disable PCIe Gen 2* bit in the *PCIe PHY Auto Configuration* EEPROM section.
9:4	RO	0x4	Max Link Width Indicates the maximum link width. The I350 can support by 1-, by 2- and by 4-link width. The field is loaded from the EEPROM (See Section 6.3.5.5.2), with a default value of four lanes. Relevant encoding: 000000b = Reserved. 000001b = x1. 000010b = x2. 000100b = x4.
11:10	RO	11b	Active State Power Management (ASPM) Support – This field indicates the level of ASPM supported on the I350 PCI Express Link. Defined encodings are: 00b = No ASPM Support. 01b = L0s Supported. 10b = L1 Supported. 11b = L0s and L1 Supported.

Bits	Rd/Wr	Default	Description
14:12	RO	Usage depended. See default values in Section 6.2.14.	L0s Exit Latency Indicates the exit latency from L0s to L0 state. 000b = Less than 64ns. 001b = 64ns – 128ns. 010b = 128ns – 256ns. 011b = 256ns - 512ns. 100b = 512ns - 1 µs. 101b = 1 µs – 2 µs. 110b = 2 µs – 4 µs. 111b = Reserved. Depending on usage of common clock or separate clock the value of this field is loaded from PCIe Init Config 1 EEPROM word, 0x18 (See Section 6.2.14).
17:15	RO	Usage depended. See default values in Section 6.2.11.	L1 Exit Latency Indicates the exit latency from L1 to L0 state. 000b = Less than 1 µs. 001b = 1 µs - 2 µs. 010b = 2 µs - 4 µs. 011b = 4 µs - 8 µs. 100b = 8 µs - 16 µs. 101b = 16 µs - 32 µs. 110b = 32 µs - 64 µs. 111b = L1 transition not supported. Depending on usage of common clock or separate clock the value of this field is loaded from *PCIe L1 Exit latencies* EEPROM word, 0x14 (See Section 6.2.11).
18	RO	0b	Clock Power Management Status Not supported in the I350. RO as zero.
19	RO	0b	Surprise Down Error Reporting Capable Status Not supported in the I350. RO as zero
20	RO	0b	Data Link Layer Link Active Reporting Capable Status Not supported in the I350. RO as zero.
21	RO	0b	Link Bandwidth Notification Capability Status Not supported in the I350. RO as zero.
22	RO	1b	ASPM Optionality Compliance Software is permitted to use the value of this bit to help determine whether to enable ASPM or whether to run ASPM compliance tests.
23	RO	00b	Reserved
31:24	HwInit	0x0	Port Number The PCIe port number for the given PCIe link. Field is set in the link training phase.

9.5.6.8 Link Control Register (0xB0; RO)

This register controls PCIe link specific parameters. There is a dedicated register per each function.

Bits	R/W	Default	Description
1:0	RW	00b	Active State Power Management (ASPM) Control – This field controls the level of Active State Power Management (ASPM) supported on the I350 PCI Express Link. For non-ARI mode, only capabilities enabled in all Functions are enabled for the component as a whole. When ARI support is exposed, ASPM Control is determined solely by the setting in Function 0 (even when it is a dummy function), regardless of Function 0's D-state. The settings in the other Functions always return whatever value software programmed for each, but otherwise are ignored by the I350. Defined encodings are: 00b = PM disabled. 01b = L0s entry supported. 10b = L1 Entry Enabled. 11b = L0s and L1 supported. Note: "L0s Entry Enabled" enables the Transmitter to enter L0s is supported. If L0s is supported, the Receiver must be capable of entering L0s even when the Transmitter is disabled from entering L0s (00b or 10b).
2	RO	0b	Reserved
3	RW	0b	Read Completion Boundary Read Completion Boundary (RCB) – Optionally Set by configuration software to indicate the RCB value of the Root Port Upstream from the Endpoint or Bridge. Defined encodings are: 0b = 64 byte 1b = 128 byte Configuration software must only Set this bit if the Root Port Upstream from the Endpoint or Bridge reports an RCB value of 128 bytes (a value of 1b in the Read Completion Boundary bit).
4	RO	0b	Link Disable Not applicable for endpoint devices; hardwired to 0b.
5	RO	0b	Retrain Clock Not applicable for endpoint devices; hardwired to 0b.
6	RW	0b	Common Clock Configuration When this bit is set, it indicates that the I350 and the component at the other end of the link are operating with a common reference clock. A value of 0b indicates that both operate with an asynchronous clock. This parameter affects the L0s exit latencies. When ARI support is exposed, the value set in function zero (even when it is a dummy function), is the value used for all the functions.
7	RW	0b	Extended Synch When this bit is set, it forces an extended Tx of a FTS ordered set in FTS and an extra TS1 at exit from L0s prior to enter L0.
8	RO	0b	Enable Clock Power Management Not supported in the I350. RO as zero.
9	RO	0b	Hardware Autonomous Width Disable Not supported in the I350. RO as zero.
10	RO	0b	Link Bandwidth Management Interrupt Enable Not supported in the I350. RO as zero.
11	RO	0b	Link Autonomous Bandwidth Interrupt Enable Not supported in the I350. RO as zero.
15:12	RO	0000b	Reserved

9.5.6.9 Link Status (0xB2; RO)

This register provides information about PCIe link specific parameters. This is a read only register identical to all functions.

Bits	R/W	Default	Description
3:0	RO	0001b	Link Speed This field indicates the negotiated link speed of the given PCIe link. Defined encodings are: 0001b = 2.5 Gb/s PCIe link. 0010b = 5 Gb/s PCIe link. All other encodings are reserved.
9:4	RO	000001b	Negotiated Link Width Indicates the negotiated width of the link. Relevant encoding for the I350 are: 000001b = x1 000010b = x2 000100b = x4
10	RO	0b	Reserved (was: Link Training Error)
11	RO	0b	Link Training Indicates that link training is in progress.
12	HwInit	1b	Slot Clock Configuration When set, indicates that the I350 uses the physical reference clock that the platform provides on the connector. This bit must be cleared if the I350 uses an independent clock. The Slot Clock Configuration bit is loaded from the *Slot_Clock_Cfg* bit in *PCIe Init Configuration 3* Word (Word 0x1A) EEPROM word.
13	RO	0b	Data Link Layer Link Active Not supported in the I350. RO as zero.
14	RO	0b	Link Bandwidth Management Status Not supported in the I350. RO as zero.
15	RO	0b	Reserved

9.5.6.10 Reserved (0xB4-0xC0; RO)

Unimplemented reserved registers not relevant to PCIe endpoint.

The following registers are supported only if the capability version is two and above.

9.5.6.11 Device Capabilities 2 (0xC4; RO)

This register identifies PCIe device specific capabilities. It is a read only register with the same value for all functions.

.

Bit Location	R/W	Default	Description
3:0	RO	1111b	Completion Timeout Ranges Supported This field indicates I350 support for the optional completion timeout programmability mechanism. This mechanism enables system software to modify the completion timeout value. Description of the mechanism can be found in Section 3.1.3.2. Four time value ranges are defined: • Range A = 50 µs to 10 ms • Range B = 10 ms to 250 ms • Range C = 250 ms to 4 s • Range D = 4 s to 64 s A value of 1111b indicates the I350 supports ranges A, B, C, & D.

Bit Location	R/W	Default	Description
4	RO	1b	Completion Timeout Disable Supported A value of 1b indicates support for the completion timeout disable mechanism. For Dummy function, this field is RO - zero.
5	RO	0b	ARI Forwarding Supported Applicable only to switch downstream ports and root ports; must be set to 0b for other function types.
6	RO	0b	AtomicOp Routing Supported - not supported in the I350.
7	RO	0b	32-bit AtomicOp Completer Supported – not supported in the I350.
8	RO	0b	64-bit AtomicOp Completer Supported – not supported in the I350.
9	RO	0b	128-bit CAS Completer Supported – not supported in the I350.
10	RO	0b	No RO-enabled PR-PR Passing – not supported in the I350.
11	RO	1b[1]	LTR Mechanism Supported – A value of 1b indicates support for the optional Latency Tolerance Requirement Reporting (LTR) mechanism capability. Note: Value loaded from LTR_EN bit in Initialization Control Word 1 EEPROM word.
13:12	RO	00b	TPH Completer supported - the I350 does not use the hints as a completer
17:14	RO	0x0	Reserved
19:18	RO	00b	Reserved
31:20	RO	0x0	Reserved

1. Value loaded from EEPROM word.

9.5.6.12 Device Control 2 (0xC8; RW)

This register controls PCIe specific parameters. There is a dedicated register per each function.

Bit location	R/W	Default	Description
3:0	RW	0000b	Completion Timeout Value[1] In devices that support completion timeout programmability, this field enables system software to modify the completion timeout value. Encoding: • 0000b = Allowable default range: 50 µs to 50 ms. It is strongly recommended that the completion timeout mechanism not expire in less than 10 ms. Actual completion timeout range supported in the I350 is 16 ms to 32 ms. Values available if Range A (50 µs to 10 ms) programmability range is supported: • 0001b = Allowable range is 50 µs to 100 µs. Actual completion timeout range supported in the I350 is 50 µs to 100 µs. • 0010b = Allowable range is 1 ms to 10 ms. Actual completion timeout range supported in the I350 is 1 ms to 2 ms. Values available if Range B (10 ms to 250 ms) programmability range is supported: • 0101b = Allowable range is 16 ms to 55 ms. Actual completion timeout range supported in the I350 is 16 ms to 32 ms. • 0110b = Allowable range is 65 ms to 210 ms. Actual completion timeout range supported in the I350 is 65 ms to 130 ms. Values available if Range C (250 ms to 4 s) programmability range is supported: • 1001b = Allowable range is 260 ms to 900 ms. Actual completion timeout range supported in the I350 is 260 ms to 520 ms. • 1010b = Allowable range is 1 s to 3.5 s. Actual completion timeout range supported in the I350 is 1 s to 2 s. Values available if the Range D (4 s to 64 s) programmability range is supported: • 1101b = Allowable range is 4 s to 13 s. Actual completion timeout range supported in the I350 is 4 s to 8 s. • 1110b = Allowable range is 17 s to 64 s. Actual completion timeout range supported in the I350 is 17 s to 34 s. Values not defined are reserved. Software is permitted to change the value in this field at any time. For requests already pending when the completion timeout value is changed, hardware is permitted to use either the new or the old value for the outstanding requests and is permitted to base the start time for each request either when this value was changed or when each request was issued. The default value for this field is 0000b. For Dummy function, this field is RO - zero.
4	RW	0b	Completion Timeout Disable When set to 1b, this bit disables the completion timeout mechanism. Software is permitted to set or clear this bit at any time. When set, the completion timeout detection mechanism is disabled. If there are outstanding requests when the bit is cleared, it is permitted but not required for hardware to apply the completion timeout mechanism to the outstanding requests. If this is done, it is permitted to base the start time for each request on either the time this bit was cleared or the time each request was issued. The default value for this bit is 0b. For Dummy function, this field is RO - zero.
5	RO	0b	Alternative RID Interpretation (ARI) Forwarding Enable Applicable only to switch devices.
6	RO	0b	AtomicOp Requester Enable - not supported in the I350.
7	RO	0b	AtomicOp Egress Blocking - not supported in the I350.
8	RW	0b	IDO Request Enable - If this bit is Set, the Function is permitted to set the ID-Based Ordering (IDO) bit (Attribute[2]) of Requests it initiates
9	RW	0b	IDO Completion Enable - If this bit is Set, the Function is permitted to set the ID-Based Ordering (IDO) bit (Attribute[2]) of Completion it initiates

Bit location	R/W	Default	Description
10	RW/ RO[2]	0b	LTR Mechanism Enable – When Set to 1b, this bit enables the Latency Tolerance Requirement Reporting (LTR) mechanism. Notes: 1. Since the I350 is a Multi-Function device, the bit in Function 0 is of type RW, and only Function 0 controls the component's Link behavior. In all other Functions this bit is of type RsvdP. 2. Field is RW and controls device behavior also when function 0 is a dummy function. 3. If Value of *LTR_EN* bit in *Initialization Control Word 1* EEPROM word is 0, then bit is RO with a value of 0b.
12:11	RO	0x0	Reserved.
14:13	RO	00b	Reserved.
15:11	RO	0x0	Reserved.

1. The completion timeout value must be programmed correctly in PCIe configuration space (in Device Control 2 Register); the value must be set above the expected maximum latency for completions in the system in which the I350 is installed. This will ensure that the I350 receives the completions for the requests it sends out, avoiding a completion timeout scenario. It is expected that the system BIOS will set this value appropriately for the system.
2. RW for Function 0, RO with a value of 0 for all other functions

9.5.6.13 Link Control 2 (0xD0; RW)

Bits	R/W	Default	Description
3:0	RWS	0010b	Target Link Speed. This field is used to set the target compliance mode speed when software is using the *Enter Compliance* bit to force a link into compliance mode. Defined encodings are: 0001b = 2.5 Gb/s Target Link Speed. 0010b = 5 Gb/s Target Link Speed. All other encodings are reserved. If a value is written to this field that does not correspond to a speed included in the *Max Link Speed* field, the result is undefined. The default value of this field is the highest link speed supported by the I350 (as reported in the *Max Link Speed* field of the Link Capabilities register). ***Notes:*** 1. For the I350 which is a Multi-Function device the field in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of the device, this field is of type RsvdP. 2. Field is RWS and controls device behavior also when function 0 is a dummy function.
4	RWS	0b	Enter Compliance. Software is permitted to force a link to enter compliance mode at the speed indicated in the *Target Link Speed* field by setting this bit to 1b in both components on a link and then initiating a hot reset on the link. The default value of this field following a fundamental reset is 0b. ***Notes:*** 1. For the I350 which is a Multi-Function device the field in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of the device, this field is of type RsvdP. 2. Field is RWS and controls device behavior also when function 0 is a dummy function.
5	RO	0b	Hardware Autonomous Speed Disable. When set to 1b, this bit disables hardware from changing the link speed for reasons other than attempting to correct unreliable link operation by reducing link speed. Bit is Hard wired to 0b.
6	RO	0b	Selectable De-emphasis This bit is not applicable and reserved for Endpoints.

Bits	R/W	Default	Description
9:7	RWS	000b	Transmit Margin This field controls the value of the non de emphasized voltage level at the Transmitter pins. Encodings: 000b - Normal operating range 001b - 800-1200 mV for full swing and 400-700 mV for half-swing 010b - (n-1) - Values must be monotonic with a non-zero slope. The value of n must be greater than 3 and less than 7. At least two of these must be below the normal operating range of n: 200-400 mV for full-swing and 100-200 mV for half-swing n - 111b reserved. ***Notes:*** 1. For the I350 which is a Multi-Function device the field in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of the device, this field is of type RsvdP. 2. Field is RWS and controls device behavior also when function 0 is a dummy function.
10	RWS	0b	Enter Modified Compliance When this bit is set to 1b, the device transmits modified compliance pattern if the LTSSM enters Polling.Compliance state. ***Notes:*** 1. For the I350 which is a Multi-Function device the field in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of the device, this field is of type RsvdP. 2. Field is RWS and controls device behavior also when function 0 is a dummy function.
11	RWS	0b	Compliance SOS When set to 1b, the LTSSM is required to send SOS periodically in between the (modified) compliance patterns. ***Notes:*** 1. For the I350 which is a Multi-Function device the field in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of the device, this field is of type RsvdP. 2. Field is RWS and controls device behavior also when function 0 is a dummy function.
12	RWS	0b	Compliance De-emphasis This bit sets the de-emphasis level in Polling.Compliance state if the entry occurred due to the Enter Compliance bit being 1b. Encodings: 1b -3.5 dB 0b -6 dB When the Link is operating at 2.5 GT/s, the setting of this bit has no effect. ***Notes:*** 1. For the I350 which is a Multi-Function device the field in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of the device, this field is of type RsvdP. 2. Field is RWS and controls device behavior also when function 0 is a dummy function.
15:13	RO	0x0	reserved

9.5.6.14 Link Status 2 (0xD2; RW)

Bits	R/W	Default	Description
0	RO	0b	Current De-emphasis Level – When the Link is operating at 5 GT/s speed, this bit reflects the level of de-emphasis. it is undefined when the Link is operating at 2.5 GT/s speed Encodings: 1b -3.5 dB 0b -6 dB
15:1	RO	0x0	Reserved

9.6 PCIe Extended Configuration Space

PCIe extended configuration space is located in a flat memory-mapped address space. PCIe extends the configuration space beyond the 256 bytes available for PCI to 4096 bytes. The I350 decodes an additional 4-bits (bits 27:24) to provide the additional configuration space as shown in Table 9-15. PCIe reserves the remaining 4 bits (bits 31:28) for future expansion of the configuration space beyond 4096 bytes.

The configuration address for a PCIe device is computed using a PCI-compatible bus, device, and function numbers as follows.

Table 9-15 PCIe Extended Configuration Space

31 28	27 20	19 15	14 12	11 2	1 0
0000b	Bus #	Device #	Fun #	Register Address (offset)	00b

PCIe extended configuration space is allocated using a linked list of optional or required PCIe extended capabilities following a format resembling PCI capability structures. The first PCIe extended capability is located at offset 0x100 in the device configuration space. The first Dword of the capability structure identifies the capability/version and points to the next capability.

The I350 supports the following PCIe extended capabilities.

Table 9-16 PCIe Extended Capability Structure

Capability	Offset	Next Header
Advanced Error Reporting	0x100	0x140/0x150/0x160/0x1A0/0x1C0/ 0x1D0[1]
Serial Number[2]	0x140	0x150/0x160/0x1A0/0x1C0/0x1D0[3]
Alternative RID Interpretation (ARI)	0x150	0x160/0x1A0/0x1C0/0x1D0
IOV support	0x160	0x1A0[4]
TLP processing hints	0x1A0	0x1C0/0x1D0[5] [4]
Latency Tolerance Requirement Reporting	0x1C0	0x1D0
Access Control Services	0x1D0	0x000

1. Depends on EEPROM settings enabling the Serial Number, ARI structure, IOV structure and LTR in EEPROM. A dummy function will point from the AER structure to the Serial Number structure. If Serial Number structure is disabled, then AER points to the ARI Structure. If ARI is also disabled a dummy function will point from the AER structure to the LTR structure, otherwise if the LTR structure is also disabled the AER structure points to the ACS structure.
2. Not available in EEPROM-less systems.
3. A dummy function will point from the Serial Number structure to the ARI Structure. If ARI is also disabled a dummy function will point from the Serial Number structure to the LTR structure, otherwise if the LTR structure is also disabled the Serial Number structure points to the ACS structure.
4. In a dummy function, the IOV and TPH structures are not exposed.
5. If LTR is enabled the TPH structure points to the LTR structure, otherwise it points to the ACS structure.

9.6.1 Advanced Error Reporting (AER) Capability

The PCIe AER capability is an optional extended capability to support advanced error reporting. The following table lists the PCIe AER extended capability structure for PCIe devices.

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
0x100	Next Capability Ptr. (0x140/0x150/0x160/ 0x1A0/0x1C0/0x1D0[1])	Version (0x2)	AER Capability ID (0x0001)	
0x104	Uncorrectable Error Status			
0x108	Uncorrectable Error Mask			
0x10C	Uncorrectable Error Severity			
0x110	Correctable Error Status			
0x114	Correctable Error Mask			
0x118	Advanced Error Capabilities and Control Register			
0x11C... 0x128	Header Log			

1. Depends on EEPROM settings, enabling the Serial Number, enabling ARI structure, IOV structure and enabling LTR. A dummy function will point from the AER structure to the ARI structure. If ARI is disabled, it will point to the LTR structure, if LTR is disabled then it will point to the ACS structure.

9.6.1.1 PCIe CAP ID (0x100; RO)

Bit Location	Attribute	Default Value	Description
15:0	RO	0x0001	Extended Capability ID PCIe extended capability ID indicating AER capability.
19:16	RO	0x2[1]	AER Capability Version PCIe AER extended capability version number.
31:20	RO	0x1D0/ 0x1C0/ 0x1A0/ 0x160/ 0x150/ 0x140	Next Capability Pointer Next PCIe extended capability pointer. A value of 0x140 points to the serial ID capability. In EEPROM-less systems or when serial ID is disabled in the EEPROM, the next pointer is 0x150 and points to the ARI capability structure. If ARI is also disabled in the EEPROM the next pointer is 0x160 and points to the IOV capability structure. If IOV is also disabled in the EEPROM this field points to the TPH capability structure (0x1A0). Note: If function is a dummy function, depending on the capability structures disabled in the EEPROM, this field can point to either 0x140 (Serial ID), 0x150 (ARI), 0x1C0 (LTR) or 0x1D0 (ACS).

1. Loaded from EEPROM (See Section 6.2.19).

9.6.1.2 Uncorrectable Error Status (0x104; R/W1CS)

The Uncorrectable Error Status register reports error status of individual uncorrectable error sources on a PCIe device. An individual error status bit that is set to 1b indicates that a particular error occurred; software can clear an error status by writing a 1b to the respective bit.

Bit Location	Attribute	Default Value	Description
3:0	RO	0x0	Reserved
4	R/W1CS	0b	Data Link Protocol Error Status
5	RO	0b	Surprise Down Error Status (Optional) Not supported in the I350.
11:6	RO	0x0	Reserved
12	R/W1CS	0b	Poisoned TLP Status
13	R/W1CS	0b	Flow Control Protocol Error Status
14	R/W1CS	0b	Completion Timeout Status
15	R/W1CS	0b	Completer Abort Status
16	R/W1CS	0b	Unexpected Completion Status
17	R/W1CS	0b	Receiver Overflow Status
18	R/W1CS	0b	Malformed TLP Status
19	R/W1CS	0b	ECRC Error Status
20	R/W1CS	0b	Unsupported Request Error Status
21	RO	0b	ACS Violation Status Not supported in the I350.
22	RO	0b	Uncorrectable Internal Error Status (Optional) Not supported in the I350.
23	RO	0b	MC Blocked TLP Status (Optional) Not supported in the I350.
24	RO	0b	AtomicOps Egress Blocked Status (Optional) Not supported in the I350.
25	RO	0b	TLP Prefix Blocked Error Status (Optional) Not supported in the I350.
31:26	RO	0x0	Reserved

9.6.1.3 Uncorrectable Error Mask (0x108; RWS)

The Uncorrectable Error Mask register controls reporting of individual uncorrectable errors by device to the host bridge via a PCIe error message. A masked error (respective bit set in mask register) is not reported to the host bridge by an individual device. There is a mask bit per bit in the Uncorrectable Error Status register.

Bit Location	Attribute	Default Value	Description
3:0	RO	0x0	Reserved
4	RWS	0b	Data Link Protocol Error Mask
5	RO	0b	Surprise Down Error Mask (Optional) Not supported in the I350.
11:6	RO	0x0	Reserved
12	RWS	0b	Poisoned TLP Mask
13	RWS	0b	Flow Control Protocol Error Mask
14	RWS	0b	Completion Timeout Mask
15	RWS	0b	Completer Abort Mask

Bit Location	Attribute	Default Value	Description
16	RWS	0b	Unexpected Completion Mask
17	RWS	0b	Receiver Overflow Mask
18	RWS	0b	Malformed TLP Mask
19	RWS	0b	ECRC Error Mask
20	RWS	0b	Unsupported Request Error Mask
21	RO	0b	ACS Violation Mask Not supported in the I350.
22	RO	0b	Uncorrectable Internal Error Mask (Optional) Not supported in the I350.
23	RO	0b	MC Blocked TLP Mask (Optional) Not supported in the I350.
24	RO	0b	AtomicOps Egress Blocked Mask (Optional) Not supported in the I350.
25	RO	0b	TLP Prefix Blocked Error Mask (Optional) Not supported in the I350.
31:26	RO	0x0	Reserved

9.6.1.4 Uncorrectable Error Severity (0x10C; RWS)

The Uncorrectable Error Severity register controls whether an individual uncorrectable error is reported as a fatal error. An uncorrectable error is reported as fatal when the corresponding error bit in the severity register is set. If the bit is cleared, the corresponding error is considered non-fatal.

Bit Location	Attribute	Default Value	Description
3:0	RO	0001b	Reserved
4	RWS	1b	Data Link Protocol Error Severity
5	RO	1b	Surprise Down Error Severity (Optional) Not supported in the I350.
11:6	RO	0x0	Reserved
12	RWS	0b	Poisoned TLP Severity
13	RWS	1b	Flow Control Protocol Error Severity
14	RWS	0b	Completion Timeout Severity
15	RWS	0b	Completer Abort Severity
16	RWS	0b	Unexpected Completion Severity
17	RWS	1b	Receiver Overflow Severity
18	RWS	1b	Malformed TLP Severity
19	RWS	0b	ECRC Error Severity
20	RWS	0b	Unsupported Request Error Severity
21	RWS	0b	ACS Violation Severity
22	RO	1b	Uncorrectable Internal Error Severity (Optional) Not supported in the I350.
23	RO	0b	MC Blocked TLP Severity (Optional) Not supported in the I350.

Bit Location	Attribute	Default Value	Description
24	RO	0b	AtomicOps Egress Blocked Severity (Optional) Not supported in the I350.
25	RO	0b	TLP Prefix Blocked Error Severity (Optional) Not supported in the I350.
31:26	RO	0x0	Reserved

9.6.1.5 Correctable Error Status (0x110; R/W1CS)

The Correctable Error Status register reports error status of individual correctable error sources on a PCIe device. When an individual error status bit is set to 1b, it indicates that a particular error occurred; software can clear an error status by writing a 1b to the respective bit.

Bit Location	Attribute	Default Value	Description
0	R/W1CS	0b	Receiver Error Status
5:1	RO	0x0	Reserved
6	R/W1CS	0b	Bad TLP Status
7	R/W1CS	0b	Bad DLLP Status
8	R/W1CS	0b	REPLAY_NUM Rollover Status
11:9	RO	000	Reserved
12	R/W1CS	0b	Replay Timer Timeout Status
13	R/W1CS	0b	Advisory Non-Fatal Error Status
14	RO	0b	Corrected Internal Error Status (Optional) Not supported in the I350.
15	RO	0b	Header Log Overflow Status (Optional) Not supported in the I350.
31:16	RO	0x0	Reserved

9.6.1.6 Correctable Error Mask (0x114; RWS)

The Correctable Error Mask register controls reporting of individual correctable errors by device to the host bridge via a PCIe error message. A masked error (respective bit set in mask register) is not reported to the host bridge by an individual device. There is a mask bit per bit in the Correctable Error Status register.

Bit Location	Attribute	Default Value	Description
0	RWS	0b	Receiver Error Mask
5:1	RO	0x0	Reserved
6	RWS	0b	Bad TLP Mask
7	RWS	0b	Bad DLLP Mask
8	RWS	0b	REPLAY_NUM Rollover Mask
11:9	RO	000b	Reserved
12	RWS	0b	Replay Timer Timeout Mask

Bit Location	Attribute	Default Value	Description
13	RWS	1b	Advisory Non-Fatal Error Mask. This bit is Set by default to enable compatibility with software that does not comprehend Role-Based Error Reporting.
14	RO	0b	Corrected Internal Error Mask (Optional) Not supported in the I350.
15	RO	0b	Header Log Overflow Mask (Optional) Not supported in the I350.
31:16	RO	0x0	Reserved

9.6.1.7 Advanced Error Capabilities and Control Register (0x118; RWS)

Bit Location	Attribute	Default Value	Description
4:0	ROS	0x0	First Error Pointer The First Error Pointer is a field that identifies the bit position of the first error reported in the Uncorrectable Error Status register.
5	RO	1b	ECRC Generation Capable This bit indicates that the I350 is capable of generating ECRC. This bit is loaded from EEPROM PCIe Control 2 word (Word 0x28).
6	RWS	0b	ECRC Generation Enable When set, enables ECRC generation.
7	RO	1b	ECRC Check Capable If Set, this bit indicates that the Function is capable of checking ECRC. This bit is loaded from EEPROM PCIe Control 2 word (Word 0x28).
8	RWS	0b	ECRC Check Enable When set, enables ECRC checking.
9	RO	0b	Multiple Header Recording Capable – If Set, this bit indicates that the Function is capable of recording more than one error header.
10	RO	0b	This bit enables the Function to record more than one error header. Functions that do not implement the associated mechanism are permitted to hardwire this bit to 0b.
11	RO	0b	TLP Prefix Log Present If Set and the First Error Pointer is valid, indicates that the TLP Prefix Log register contains valid information. If Clear or if First Error Pointer is invalid, the TLP Prefix Log register is undefined. Default value of this bit is 0b. This bit is RsvdP if the End-End TLP Prefix Supported bit is Clear.
31:12	RO	0x0	Reserved

9.6.1.8 Header Log (0x11C:0x128; RO)

The Header Log register captures the header for the transaction that generated an error. This register is 16 bytes in length.

Bit Location	Attribute	Default Value	Description
127:0	ROS	0b	Header of the packet in error (TLP or DLLP).

9.6.2 Serial Number

The PCIe device serial number capability is an optional extended capability that can be implemented by any PCIe device. The device serial number is a read-only 64-bit value that is unique for a given PCIe device.

All multi-function devices that implement this capability must implement it for function 0; other functions that implement this capability must return the same device serial number value as that reported by function 0.

Note: The I350 does not support this capability in an EEPROM-less configuration.

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
0x140	Next Capability Ptr. (0x150/0x160/0x1A0/ 0x1C0/0X1D0)[1 2]	Version (0x1)	Serial ID Capability ID (0x0003)	
0x144	Serial Number Register (Lower Dword)			
0x148	Serial Number Register (Upper Dword)			

1. If ARI structure is enabled in EEPROM value of field is 0x150 that points to ARI capability structure, otherwise if IOV structure is enabled in EEPROM value of field is 0x160 otherwise value of field is 0x1A0 that points to the TPH capability structure.
2. If function is a dummy function, if ARI structure is enabled in EEPROM value of field is 0x150 that points to ARI capability structure, otherwise if LTR is enabled in the EEPROM value of field is 0x1C0 that points to LTR capability structure otherwise value of the field is 0x1D0 that points to the ACS capability structure.

9.6.2.1 Device Serial Number Enhanced Capability Header (0x140; RO)

The following table lists the allocation of register fields in the device serial number enhanced capability header. It also lists the respective bit definitions. The extended capability ID for the device serial number capability is 0x0003.

Bit(s) Location	Default value	Attributes	Description
15:0	0x0003	RO	PCIe Extended Capability ID This field is a PCI-SIG defined ID number that indicates the nature and format of the extended capability. The extended capability ID for the device serial number capability is 0x0003.
19:16	0x1	RO	Capability Version This field is a PCI-SIG defined version number that indicates the version of the current capability structure.
31:20	0x150/ 0x160/ 0x1A0/ 0X1C0/ 0X1D0	RO	Next Capability Offset This field contains the offset to the next PCIe capability structure or 0x000 if no other items exist in the linked list of capabilities. • In a LAN function the value of this field is 0x150 to point to the ARI capability structure. If ARI is disabled value of field is 0x160 that points to the IOV capability structure. If IOV is also disabled in the EEPROM, then this field is 0x1A0 that points to the TPH capabilities structure. • In a Dummy function the value of this field is 0x150 to point to the ARI capability structure. If ARI is disabled in the EEPROM, and LTR is enabled in the EEPROM then the value of this field is 0x1C0 (LTR capability structure), otherwise the value in this field is 0x1D0 (ACS capability structure).

9.6.2.2 Serial Number Register (0x144:0x148; RO)

The Serial Number register is a 64-bit field that contains the IEEE defined 64-bit extended unique identifier (EUI-64™). Table 9-17 lists the allocation of register fields in the Serial Number register. Table 9-17 also lists the respective bit definitions.

Table 9-17 Serial Number Register

31:0
Serial Number Register (Lower Dword)
Serial Number Register (Upper word)
63:32

Serial number definition in the I350:

Table 9-18 SN Definition

Bit(s) Location	Attributes	Description
63:0	RO	PCIe Device Serial Number This field contains the IEEE defined 64-bit extended unique identifier (EUI-64™). This identifier includes a 24-bit company ID value assigned by IEEE registration authority and a 40-bit extension identifier assigned by the manufacturer.

Serial number uses the MAC address according to the following definition:

Field	Extension identifier					Company ID		
Order	Addr+0	Addr+1	Addr+2	Addr+3	Addr+4	Addr+5	Addr+6	Addr+7

Most significant byte — Least significant byte

Most significant bit — Least significant bit

The serial number can be constructed from the 48-bit MAC address in the following form:

Field	Extension identifier			MAC Label		Company ID		
Order	Addr+0	Addr+1	Addr+2	Addr+3	Addr+4	Addr+5	Addr+6	Addr+7

Most significant bytes — Least significant byte

Most significant bit — Least significant bit

The MAC label in this case is 0xFFFF.

For example, assume that the company ID is (Intel) 00-A0-C9 and the extension identifier is 23-45-67. In this case, the 64-bit serial number is:

Field	Extension identifier			MAC Label		Company ID		
Order	Addr+0	Addr+1	Addr+2	Addr+3	Addr+4	Addr+5	Addr+6	Addr+7
	67	45	23	FF	FF	C9	A0	00

Most significant byte — Least significant byte

Most significant bit — Least significant bit

The MAC address is the function 0 MAC address as loaded from the EEPROM into the RAL and RAH registers.

The translation from EEPROM words 0 to 2 to the serial number is as follows:

- Serial number ADDR+0 = EEPROM byte 5
- Serial number ADDR+1 = EEPROM byte 4
- Serial number ADDR+2 = EEPROM byte 3
- Serial number ADDR+3,4 = 0xFF 0xFF
- Serial number ADDR+5 = EEPROM byte 2
- Serial number ADDR+6 = EEPROM byte 1
- Serial number ADDR +7 = EEPROM byte 0

The official document defining EUI-64 is: http://standards.ieee.org/regauth/oui/tutorials/EUI64.html

9.6.3 ARI Capability Structure

In order to enable more than eight functions per end point without requesting an internal switch (typically needed in virtualization scenarios), the PCIsig defines a new capability that enables a different interpretation of the *Bus*, *Device*, and *Function* fields. The Alternate Requester ID Interpretation (ARI) capability structure is as follows:.

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
0x150	Next Capability Ptr. (0x160/0x1A0/0x1C0/ 0x1D0)[1] [2]	Version (0x1)	ARI Capability ID (0x000E)	
0x154	ARI Capabilities & Control Register			

1. If function is not a Dummy function next capability structure is 0x160 that points to the IOV structure and if IOV is disabled in the EEPROM value of field is 0x1A0 that points to the TPH capability structure.
2. If function is a Dummy function and LTR is enabled in the EEPROM, value in field is 0x1C0 that points to the LTR capability structure, otherwise if LTR is disabled, value of field is 0x1D0 that points to the ACS capability structure.

9.6.3.1 PCIe ARI Header Register (0x150; RO)

Bit(s)	Initial Value	Access	Description
15:0	0x000E	RO	ID - PCIe Extended Capability ID PCIe extended capability ID for the ARI.
19:16	0x1	RO	Version - Capability Version This field is a PCI-SIG defined version number that indicates the version of the current capability structure. Must be 0x1 for this version of the specification.
31:20	0x160/ 0x1A0/ 0x1C0/ 0x1D0	RO	Next Capability Ptr. - Next Capability Offset This field contains the offset to the next PCIe extended capability structure. The value of 0x160 points to the IOV structure. If IOV is disabled the value of the field is 0x1A0 that points to the TPH capability structure. For a dummy function, the next capability structure is LTR (value of 0x1C0), if LTR is disabled the next structure is ACS (Value of 0x1D0).

9.6.3.2 PCIe ARI Capabilities & Control Register (0x154; RO)

Bit(s)	Initial Value	Access	Description
0	0b	RO	M - MFVC Function Groups Capability Applicable only to function 0; must be 0b for all other functions. If 1b, indicates that the ARI device supports function group level arbitration via its Multi-Function Virtual Channel (MFVC) Capability structure. Not supported in the I350.
1	0b	RO	A - ACS Function Groups Capability (A) Applicable only to function 0; must be 0b for all other functions. If 1b, indicates that the ARI device supports function group level granularity for ACS P2P egress control via its ACS capability structures. Not supported in the I350.
7:2	0x0	RO	Reserved
15:8	0x1 (func 0) 0x2 (func 1) 0x3 (func 2) 0x0 (func 3)[1]	RO	NFP - Next Function Pointer This field contains the pointer to the next physical function configuration space or 0x0000 if no other items exist in the linked list of functions. Function 0 is the start of the link list of functions.
16	0b	RO	M_EN - MFVC Function Groups Enable (M) Applicable only for function 0; must be hardwired to 0b for all other functions. When set, the ARI device must interpret entries in its function arbitration table as function group numbers rather than function numbers. Not supported in the I350.
17	0b	RO	A_EN - ACS Function Groups Enable (A) Applicable only for function 0; must be hardwired to 0b for all other functions. When set, each function in the ARI device must associate bits within its egress control vector with function group numbers rather than function numbers. Not supported in the I350.
19:18	00b	RO	Reserved
22:20	0x0	RO	Function Group Number (FGN) Not supported in the I350.
31:23	0x0	RO	Reserved

1. If port 0, port 1, port 2 or port 3 are switched or function zero is a dummy function, this register should keep its attributes according to the function number. If part of the LANs are disabled, then the value of this field should create a valid link list between all the functions that are enabled. In the last function this field should be zero.

9.6.4 SR-IOV Capability Structure

This is the structure used to support the IOV capabilities reporting and control.

Note: This capability structure is not exposed in a Dummy function.

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
0x160	Next Capability offset (0x1A0)	Version (0x1)	IOV Capability ID (0x0010)	
0x164	SR IOV Capabilities			
0x168	SR IOV Status		SR IOV Control	
0x16C	TotalVFs (RO)		Initial VF (RO)	
0x170	Reserved	Function Dependency Link (RO)	Num VF (RW)	
0x174	VF Stride (RO)		First VF Offset (RO)	
0x178	VF Device ID		Reserved	
0x17C	Supported Page Size (0x553)			

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
0x180	system page Size (RW)			
0x184	VF BAR0 - Low (RW)			
0x188	VF BAR0 - High (RW)			
0x18C	VF BAR2 (RO)			
0x190	VF BAR3 - Low (RW)			
0x194	VF BAR3- High (RW)			
0x198	VF BAR5 (RO)			
0x19C	VF Migration State Array Offset (RO)			

9.6.4.1 PCIe SR-IOV Header Register (0x160; RO)

Bit(s)	Initial Value	Access	Description
15:0	0x0010	RO	PCIe Extended Capability ID PCIe extended capability ID of the SR-IOV capability.
19:16	0x1	RO	Capability Version This field is a PCI-SIG defined version number that indicates the version of the current capability structure. Must be 0x1 for this version of the specification.
31:20	0x1A0	RO	Next Capability Offset This field contains the offset to the next PCIe extended capability structure or 0x000 if no other items that exist in the linked list of capabilities. This pointer points to the TPH capability.

9.6.4.2 PCIe SR-IOV Capabilities Register (0x164; RO)

Bit(s)	Initial Value	Access	Description
0	0b	RO	VF Migration Capable Migration capable device running under migration capable MR-PCIM. RO as 0b in the I350.
1	1b/0b[1]	RO	ARI Capable Hierarchy Preserved - If Set, the ARI Capable Hierarchy bit is preserved across certain power state transitions.
20:2	0x0	RO	Reserved
31:21	0x0	RO	VF Migration Interrupt Message Number Indicates the MSI/MSI-X vector used for the interrupts. This field is undefined when the VF Migration Capable bit is cleared.

1. Set on first function where SR-IOV is enabled. ARI Capable Hierarchy Preserved bit is Read Only Zero in other PFs of a Device.

9.6.4.3 PCIe SR-IOV Status and Control Register (0x168; RW)

Bit(s)	Initial Value	Access	Description
0	0b	RW	VFE: VF Enable/Disable VF Enable manages the assignment of VFs to the associated PF. If VF Enable is Set, the VFs associated with the PF are accessible in the PCIe fabric. When Set, VFs respond to and may issue PCI-Express transactions following the rules for PCI-Express Endpoint Functions. If Clear, VFs are disabled and not visible in the PCI-Express fabric; VFs shall respond to Requests with UR and may not issue PCIe transactions. Setting VF Enable after it has been previously been Cleared shall result in the same VF state as if FLR had been issued to the VF.
1	0b	RO	VF ME - VF Migration Enable Enables/disables VF migration support.
2	0b	RO	VF MIE - VF Migration Interrupt Enable Enables/disables VF migration state change interrupt. Note: Not implemented in the I350.
3	0b	RW	VF MSE - Memory Space Enable for Virtual Functions VF MSE controls memory space enable for all VFs associated with this PF as with the *Memory Space Enable* bit in a functions PCI command register. The default value for this bit is 0b. When *VF Enable* = 1b, virtual function memory space access is permitted only when VF MSE is set. VFs must follow the same error reporting rules as defined in the base specification if an attempt is made to access a virtual functions memory space when *VF Enable* is 1b and VF MSE is 0b. **Implementation Note:** Virtual functions memory space cannot be accessed when the *VF Enable* bit = 0b. Thus, VF MSE is a don't care when *VF Enable* = 0b, however, software might choose to set VF MSE after programming the VF BARn registers, prior to setting *VF Enable* to 1b.
4	0b	RW (first function where SR-IOV is enabled) RO (all other functions)[1]	ARI Capable Hierarchy The I350 can locate VFs in function numbers 8 to 255 of the captured bus number. Default value is 0b. This field is RW in the lowest numbered PF. Other functions use the PF0 value as sticky. ***Notes:*** 1. If either ARI Capable Hierarchy Preserved is Set (see Section 9.6.4.2) or No_Soft_Reset is Set, a power state transition of this PF from D3hot to D0 does not affect the value of this bit. 2. This bit is not reset by a FLR reset - only by a full device reset.
15:5	0x0	RO	Reserved
16	0b	RO	VFMIS - VF Migration Event Pending Indicates a VF migration in or migration out request has been issued by MR-PCIM. To determine the cause of the event, software can scan the VF state array. Note: Not implemented in the I350.
31:17	0x0	RO	Reserved

1. If the ports are switched, this field should keep its attributes according to the function number.

9.6.4.4 PCIe SR-IOV Max/Total VFs Register (0x16C)

Table 9-19 PCIe SR-IOV Max/Total VFs Register (0x16C)

Bit(s)	Initial Value	Access	Description
15:0	0x8	RO	**InitialVFs** **InitialVFs** indicates the number of VFs that are initially associated with the PF. If VF migration capable is clear, this field must contain the same value as TotalVFs. A lower value of this field can be loaded from the IOV control word in the EEPROM.

Table 9-19 PCIe SR-IOV Max/Total VFs Register (0x16C) (Continued)

Bit(s)	Initial Value	Access	Description
31:16	0x8	RO	**TotalVFs** **TotalVFs** indicates the maximum number of VFs that could be associated with the PF. In the I350, this is equal to InitialVFs.

9.6.4.5 PCIe SR-IOV Num VFs Register (0x170; R/W)

Bit(s)	Initial Value	Access	Description
15:0	0x0	R/W	**NumVFs** **NumVFs** define the number of VFs software has assigned to the PF. Software sets NumVFs as part of the process of creating VFs. NumVFs VFs must be visible in the PCIe fabric after both NumVFs are set to a valid value and *VF Enable* is set to 1b. Visible in the PCIe fabric means that the VF must respond to PCIe transactions targeting the VF, following all other rules defined by this specification and the base specification. The results are undefined if NumVFs are set to a value greater than TotalVFs. NumVFs can only be written while *VF Enable* is clear. The NumVFs field is RO when *VF Enable* is set.
23:16	0x0 (func 0) 0x1 (func 1) 0x2 (func 2) 0x3 (func 3)[1]	RO	**FDL - Function Dependency Link** Defines dependencies between physical functions allocation. The default behavior of the I350 is not to define any such constraints.
31:24	0x0	RO	Reserved

1. If port 0, port 1, port 2 or port 3 are switched or function zero is a dummy function, this register should keep its attributes according to the function number.

9.6.4.6 PCIe SR-IOV VF RID Mapping Register (0x174; RO)

See Section 7.8.2.6 for details of the RID mapping.

Bit(s)	Initial Value	Access	Description
15:0	0x180	RO	**FVO** **First VF offset** defines the requestor ID (RID) offset of the first VF that is associated with the PF that contains this capability structure. The first VFs 16-bit RID is calculated by adding the contents of this field to the RID of the PF containing this field. The content of this field is valid only when *VF Enable* is set. If *VF Enable* is 0b, the contents are undefined. If the *ARI Enable* bit is set, this field changes to 0x80.
31:16	0x4	RO	**VFS** **VF Stride** defines the Requestor ID (RID) offset from one VF to the next one for all VFs associated with the PF that contains this capability structure. The next VFs 16-bit RID is calculated by adding the contents of this field to the RID of the current VF. The content of this field is valid only when *VF Enable* is set and NumVFs are a non-zero. If *VF Enable* is 0b or if NumVFs are zero, the contents are undefined.

9.6.4.7 PCIe SR-IOV VF device ID (0x178; RO)

Bit(s)	Initial Value	Access	Description
31:16	0x1520	RO	This field contain the Device ID that should be presented for every VF to the Virtual Machine software. The value of this field may be read from EEPROM word 0x26
15:0	0x0	RO	Reserved

9.6.4.8 PCIe SR-IOV Supported Page Size Register (0x17C; RO)

Bit(s)	Initial Value	Access	Description
31:0	0x553	RO	Supported page Size For PFs that support the stride-based BAR mechanism, this field defines the supported page sizes. This PF supports a page size of 2^(n+12) if bit n is set. For example, if bit 0 is set, the EP supports 4 KB page sizes. The I350 supports 4 KB, 8 KB, 64 KB, 256 KB, 1 MB and 4 MB page sizes.

9.6.4.9 PCIe SR-IOV System Page Size Register (0x180; R/W)

Bit(s)	Initial Value	Access	Description
31:0	0x1	R/W	Page Size This field defines the page size the system uses to map the PF's and associated VFs' memory addresses. Software must set the value of the system page size to one of the page sizes set in the *Supported Page Sizes* field. As with supported page sizes, if bit n is set in system page size, the PF and its associated VFs are required to support a page size of 2^(n+12). For example, if bit 1 is set, the system is using an 8 KB page size. The results are undefined if more than one bit is set in system page size. The results are undefined if a bit is set in a system page size that is not set in supported page sizes. When system page size is set, the PF and associated VFs are required to align all BAR resources on a system page size boundary. Each BAR size, including VF BARn size (described in the sections that follow) must be aligned on a system page size boundary. Each BAR size, including VF BARn size must be sized to consume a multiple of system page size bytes. All fields requiring page size alignment within a function must be aligned on a system page size boundary. *VF Enable* must be set to 0b when system page size is set. The results are undefined if system page size is set when *VF Enable* is set.

9.6.4.10 PCIe SR-IOV BAR 0 - Low Register (0x184; R/W)

Bit(s)	Initial Value	Access	Description
0	0b	RO	Mem 0b = Indicates memory space.

Bit(s)	Initial Value	Access	Description
2:1	10b	RO	Mem Type Indicates the address space size. 00b = 32-bit 01b = Reserved 10b = 64-bit. 11b = Reserved BAR bit sizes are set according to bit 2 in EEPROM word 0x25.
3	0b	RO	Prefetch Mem 0b = Non-prefetchable space. 1b = Prefetchable space. This BARs prefetchable bit is set according to bit 1 in EEPROM word 0x25.
31:4	0x0	R/W	Memory Address Space Which bits are R/W bits and which are read only to 0b depends on the memory mapping window size. The size is a maximum between 16 KB and the page size.

9.6.4.11 PCIe SR-IOV BAR 0 - High Register (0x188; R/W)

Bit(s)	Initial Value	Access	Description
31:0	0b	RW	BAR0 - MSB MSB part of BAR 0.

9.6.4.12 PCIe SR-IOV BAR 2 (0x18C; RO)

Bit(s)	Initial Value	Access	Description
31:0	0b	RO	BAR2 This BAR is not used.

9.6.4.13 PCIe SR-IOV BAR 3 - Low Register (0x190; R/W)

Bit(s)	Initial Value	Access	Description
0	0b	RO	Mem 0b = Indicates memory space.
2:1	10b	RO	Mem Type Indicates the address space size. 00b = 32-bit. 01b = Reserved. 10b = 64-bit. 11b = Reserved. BAR bit sizes are set according to bit 2 in EEPROM word 0x25.
3	0b	RO	Prefetch Mem 0b = Non-prefetchable space. 1b = Prefetchable space. This BAR's prefetchable bit is set according to bit 1 in EEPROM word 0x25.
31:4	0b	R/W	Memory Address Space Which bits are R/W bits and which are read only to 0b depends on the memory mapping window size. The size is a maximum between 16 KB and the page size.

9.6.4.14 PCIe SR-IOV BAR 3 - High Register (0x194; R/W)

Bit(s)	Initial Value	Access	Description
31:0	0x0	RW	BAR3 - MSB MSB part of BAR 3.

9.6.4.15 PCIe SR-IOV BAR 5 (0x198; RO)

Bit(s)	Initial Value	Access	Description
31:0	0x0	RO	BAR5 This BAR is not used.

9.6.4.16 PCIe SR-IOV VF Migration State Array Offset (0x19C;

RO)

Bit(s)	Initial Value	Access	Description
2:0	000b	RO	BIR: Indicates which PF BAR contains the VF migration state array. Not implemented in the I350.
31:0	0x0	RO	Offset, relative to the beginning of the BAR of the start of the migration array. Not implemented in the I350.

9.6.5 TLP Processing Hint Requester (TPH) Capability

The PCIe TPH Requester capability is an optional extended capability to support TLP Processing Hints. The following table lists the PCIe TPH extended capability structure for PCIe devices.

Note: This capability structure is not exposed in a Dummy function.

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
0x1A0	Next Capability Ptr. (0x1C0/0x1D0[1])	Version (0x1)	TPH Capability ID (0x17)	
0x1A4	TPH Requester Capability Register			
0x1A8	TPH Requester Control Register			
0x1AC-0x1B8	TPH ST Table			

1. Depends on EEPROM settings of the *LTR_EN* bit in *Initialization Control Word 1* EEPROM word, that controls enabling of the LTR structures.

9.6.5.1 TPH CAP ID (0x1A0; RO)

Bit Location	Attribute	Default Value	Description
15:0	RO	0x17	Extended Capability ID PCIe extended capability ID indicating TPH capability.
19:16	RO	0x1	Version Number PCIe TPH extended capability version number.
31:20	RO	0x1C0/ 0x1D0[1]	Next Capability Pointer This field contains the offset to the next PCIe capability structure. If LTR is enabled in EEPROM then value of this field is 0x1C0 to point to the LTR capability structure. If LTR is disabled then the value of this field is 0x1D0 to point to the ACS capability structure.

1. Depends on EEPROM settings of the *LTR_EN* bit in *Initialization Control Word 1* EEPROM word, that controls enabling of the LTR structures.

9.6.5.2 TPH Requester Capabilities (0x1A4; RO)

Bit Location	Attribute	Default Value	Description
0	RO	1	No ST Mode Supported: When set indicates the Function is capable of generating Requests without using ST.
1	RO	0	Interrupt Vector Mode Supported: Cleared to indicates that the I350 does not support Interrupt Vector Mode of operation
2	RO	1	Device Specific Mode: Set to indicate that the I350 supports Device Specific Mode of operation
7:3	RO	0	Reserved
8	RO	0	Extended TPH Requester Supported – Cleared to indicate that the function is not capable of generating requests with Extended TPH TLP Prefix
10:9	RO	01b	ST Table Location – Value indicates if and where the ST Table is located. Defined Encodings are: 00b – ST Table is not present. 01b – ST Table is located in the TPH Requester Capability structure. 10b – ST Table is located in the MSI-X Table structure. 11b – Reserved Default value of 01b indicates that function supports ST table that's located in the TPH Requester Capability structure.
15:11	RO	0x0	Reserved
26:16	RO	0x7	ST_Table Size – System software reads this field to determine the ST_Table_Size N, which is encoded as N-1. The I350 supports a table with 8 entries.
31:27	RO	0x0	Reserved

9.6.5.3 TPH Requester Control (0x1A8; R/W)

Bit Location	Attribute	Default Value	Description
2:0	RW	0x0	ST Mode Select – Indicates the ST mode of operation selected. The ST mode encodings are as defined below 000b – No Table Mode 001b – Interrupt Vector Mode (not supported by the I350) 010b – Device Specific Mode 0thers – reserved for future use The default value of 000 indicates No Table mode of operation.
7:3	RO	0x0	Reserved
9:8	RW	0x0	TPH Requester Enable: Defined Encodings are 00b – The I350 is not permitted to issue transactions with TPH or Extended TPH as Requester 01b – The I350 is permitted to issue transactions with TPH as Requester and is not permitted to issue transactions with Extended TPH as Requester 10b – Reserved 11b – The I350 is permitted to issue transactions with TPH and Extended TPH as Requester (The I350 does not issue transactions with Extended TPH).
31:10	RO	0x0	Reserved

9.6.5.4 TPH Steering Table (0x1AC - 0x1B8; R/W)

Bit Location	Attribute	Default Value	Description
7:0	RW	0x0	Steering Table Lower Entry 2*n (n = 0...3). A value of zero indicates the tag is not valid
15:8	RO	0x0	Steering Table Upper Entry 2*n (n = 0...3) - RO zero in the I350, as extended tags are not supported.
23:16	RW	0x0	Steering Table Entry 2*n + 1 (n = 0...3) - A value of zero indicates the tag is not valid
31:24	RO	0x0	Steering Table Upper Entry 2*n + 1 (n = 0...3) - RO zero in the I350, as extended tags are not supported.

9.6.6 Latency Tolerance Requirement Reporting (LTR) Capability

The PCI Express Latency Tolerance Requirement Reporting Capability is an optional Extended Capability that allows software to provide platform latency information to devices with upstream ports (Endpoints and Switches). This capability structure is required if the device supports Latency Tolerance Requirement Reporting (LTR).

Note: The LTR Capability structure is implemented only in Function 0 even when Function 0 is a dummy function, and controls the component's Link behavior on behalf of all the Functions of the device

The following table lists the PCIe LTR extended capability structure for PCIe devices.

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
0x1C0	Next Capability Ptr. (0x1D0)	Version (0x1)	LTR Capability ID (0x18)	
0x1C4	Maximum Non-Snooped Platform Latency Tolerance Register		Maximum Snooped Platform Latency Tolerance Register	

9.6.6.1 LTR CAP ID (0x1C0; RO)

Bit Location	Attribute	Default Value	Description
15:0	RO	0x18	LTR Capability ID PCIe extended capability ID indicating LTR capability.
19:16	RO	0x1	Version Number PCIe LTR extended capability version number.
31:20	RO	0x1D0	Next Capability Pointer Points to the ACS capability

9.6.6.2 LTR Capabilities (0x1C4; RW)

Bit Location	Attribute	Default Value	Description
9:0	RW	0x0	Maximum Snoop Latency Value Along with the Max Snoop Latency Scale field, this register specifies the maximum nosnoop latency that a device is permitted to request. Software should set this to the platform's maximum supported latency or less. Field is also an indicator of the platforms maximum latency, should an endpoint send up LTR Latency Values with the Requirement bit not set.
12:10	RW	0x0	Max Snoop Latency Scale This field provides a scale for the value contained within the Maximum Snoop Latency Value field. Encoding: 000 – Value times 1ns 001 – Value times 32ns 010 – Value times 1,024ns 011 – Value times 32,768ns 100 – Value times 1,048,576ns 101 – Value times 33,554,432ns 110-111 – Not Permitted
15:13	RO	0x0	Reserved
25:16	RW	0x0	Max No-Snoop Latency Value Along with the Max No-Snoop Latency Scale field, this register specifies the maximum no-snoop latency that a device is permitted to request. Software should set this to the platform's maximum supported latency or less. Field is also an indicator of the platforms maximum latency, should an endpoint send up LTR Latency Values with the Requirement bit not set.
28:26	RW	0x0	Max No-Snoop Latency Scale — This register provides a scale for the value contained within the Maximum Non-Snoop Latency Value field. Encoding: 000 – Value times 1 ns 001 – Value times 32 ns 010 – Value times 1,024 ns 011 – Value times 32,768 ns 100 – Value times 1,048,576 ns 101 – Value times 33,554,432 ns 110-111 – Not Permitted
31:29	RO	0x0	Reserved.

9.6.7 Access Control Services (ACS) Capability

The PCI Express ACS defines a set of control points within a PCI Express topology to determine whether a TLP should be routed normally, blocked, or redirected. ACS is applicable to RCs, Switches, and multifunction devices

The following table lists the PCIe ACS extended capability structure for PCIe devices.

Byte Offset	Byte 3	Byte 2	Byte 1	Byte 0
0x1D0	Next Capability Ptr. (0x000)	Version (0x1)	ACS Capability ID (0x0D)	
0x1D4	ACS Control Register (0x0)		ACS Capability Register (0x0)	

9.6.7.1 ACS CAP ID (0x1D0; RO)

Bit Location	Attribute	Default Value	Description
15:0	RO	0x0D	ACS Capability ID PCIe extended capability ID indicating ACS capability.
19:16	RO	0x1	Version Number PCIe ACS extended capability version number.
31:20	RO	0x000	Next Capability Pointer This is the last capability, so the next pointer is 0x000.

9.6.7.2 ACS Control and Capabilities (0x1D4; RO)

Bit Location	Attribute	Default Value	Description
0	RO	0b	ACS Source Validation (V) – Hardwired to Zero, not supported in the I350.
1	RO	0b	ACS Translation Blocking (B) – Hardwired to Zero, not supported in the I350.
2	RO	0b	ACS P2P Request Redirect (R) – Hardwired to Zero, not supported in the I350.
3	RO	0b	ACS P2P Completion Redirect (C) – Hardwired to Zero, not supported in the I350.
4	RO	0b	ACS Upstream Forwarding (U) – Hardwired to Zero, not supported in the I350.
5	RO	0b	ACS P2P Egress Control (E) – Hardwired to Zero, not supported in the I350.
6	RO	0b	ACS Direct Translated P2P (T) – Hardwired to Zero, not supported in the I350.
7	RsrvP	0b	Reserved
15:8	RO	0x0	Egress Control Vector Size – Hardwired to Zero, not supported in the I350.
16	RO	0b	ACS Source Validation Enable (V) – Hardwired to Zero, not supported in the I350.
17	RO	0b	ACS Translation Blocking Enable (B) – Hardwired to Zero, not supported in the I350.
18	RO	0b	ACS P2P Request Redirect Enable (R) – Hardwired to Zero, not supported in the I350.
19	RO	0b	ACS P2P Completion Redirect Enable (C) – Hardwired to Zero, not supported in the I350.
20	RO	0b	ACS Upstream Forwarding Enable (U) – Hardwired to Zero, not supported in the I350.
21	RO	0b	ACS P2P Egress Control Enable (E) – Hardwired to Zero, not supported in the I350.
22	RO	0b	ACS Direct Translated P2P Enable (T) – Hardwired to Zero, not supported in the I350.
31:23	RsrvP	0b	Reserved

9.7 Virtual Functions (VF) Configuration Space

The configuration space reflected to each VF is a sparse version of the physical function configuration space. Table 9-20 lists the behavior of each register in the VF configuration space.

Table 9-20 VF PCIe Configuration Space

Section	Offset	Name	VF Behavior	Notes
PCI Mandatory Registers	0x0	Vendor ID	RO - 0xFFFF	
	0x2	Device ID	RO - 0xFFFF	
	0x4	Command	Per VF	See Table 9-21 for details
	0x6	Status	Per VF	See Table 9-22 for details
	0x8	RevisionID	RO as PF	
	0x9	Class Code	RO as PF	
	0xC	Cache Line Size	RO - 0	
	0xD	LatencyTimer	RO - 0	
	0xE	Header Type	RO - 0	
	0xF	BIST	RO - 0	
	0x10 - 0x27	BARs	RO - 0	Emulated by VMM
	0x28	CardBus CIS	RO - 0	Not used
	0x2C	Sub Vendor ID	RO as PF	
	0x2E	Sub System	RO as PF	
	0x30	Expansion ROM	RO - 0	Emulated by VMM
	0x34	Cap Pointer	RO - 0x70	Points to MSI-X
	0x3C	Int Line	RO - 0	
	0x3D	Int Pin	RO - 0	
	0x3E	Max Lat/Min Gnt	RO - 0	
MSI-X Capability	0x70	MSI-X Header	RO - 0xA011	Points to PCIe capability
	0x72	MSI-x Message Control	per VF	See Table 9-23
	0x74	MSI-X Table Address	RO	See Table 9-24
	0x78	MSI-X PBA Address	RO	See Table 9-25
PCIe Capability	0xA0	PCIe Header	RO - 0x0010	Last capability
	0xA2	PCIe Capabilities	RO - as PF	0x0002
	0xA4	PCIe Dev Cap	RO - as PF	
	0xA8	PCIe Dev Ctrl	RW	RO as zero apart from FLR - see Table 9-26
	0xAA	PCIe Dev Status	per VF	See Table 9-27
	0xAC	PCIe Link Capabilities	RO - as PF	
	0xB0	PCIe Link Control	RO - 0x0	
	0xB2	PCIe Link Status	RO - 0x0	
	0xC4	PCIe Dev Cap 2	RO - as PF	
	0xC8	PCIe Dev Ctrl 2	RO - 0x0	The Timeout value and Timeout disable of the PF are used for all VFs.
	0xD0	PCIe Link Ctrl 2	RO - 0x0	
	0xD2	PCIe Link Status 2	RO - 0x0	

Table 9-20 VF PCIe Configuration Space (Continued)

Section	Offset	Name	VF Behavior	Notes
AER Capability	0x100	AER - Header	RO - 0x15010001	Points to ARI structure
	0x104	AER - Uncorr Status	per VF	See Table 9-28
	0x108	AER - Uncorr Mask	RO - 0x0	
	0x10C	AER - Uncorr Severity	RO - 0x0	
	0x110	AER - Corr Status	per VF	See Table 9-29
	0x114	AER - Corr Mask	RO - 0x0	
	0x118	AER - Cap/Ctrl	per VF	See Table 9-30
	0x11C:0x128	AER - Error Log	one log per VF	Same structure as in PF.
ARI Capability	0x150	ARI - Header	0x1A01000E	Points to TPH
	0x154	ARI - Cap/Ctrl	RO - 0	
TPH Requester capability	0x1A0	TPH - Header	0x1D010017	Points to ACS
	0x1A4	TPH - Capability	RO - 0x00000005	No table reported.
	0x1A8	TPH - Control	per VF	Same structure as in PF
ACS capability	0x1D0	ACS - Header	RO - 0x0001000D	Last
	0x1D4	ACS - Capability	RO - 0x00000000	

9.7.1 Legacy Header Details

Table 9-21 VF Control register (0x4; RW)

Bit(s)	Initial Value	R/W	Description
0	0b	RO	IOAE - I/O Access Enable RO as a zero field.
1	0b	RO	MAE - Memory Access Enable RO as a zero field.
2	0b	RW	BME - Bus Master Enable Disabling this bit prevents the associated VF from issuing any memory or I/O requests. Note that as MSI/MSI-X interrupt messages are in-band memory writes, disabling the bus master enable bit disables MSI/MSI-X interrupt messages as well. Requests other than memory or I/O requests are not controlled by this bit. **Note:** The state of active transactions is not specified when this bit is disabled after being enabled. The I350 can choose how it behaves when this condition occurs. Software cannot count on the I350 retaining state and resuming without loss of data when the bit is re-enabled. Transactions for a VF that has its *Bus Master Enable* bit set must not be blocked by transactions for VFs that have their *Bus Master Enable* bit cleared.
3	0b	RO	SCM - Special Cycle Enable Hardwired to 0b.
4	0b	RO	MWIE - MWI Enable Hardwired to 0b.
5	0b	RO	PSE - Palette Snoop Enable Hardwired to 0b.
6	0b	RO	PER - Parity Error Response Zero for VFs. Behavior is set by PF bit
7	0b	RO	WCE - Wait Cycle Enable Hardwired to 0b.

Table 9-21 VF Control register (0x4; RW) (Continued)

Bit(s)	Initial Value	R/W	Description
8	0b	RO	SERRE - SERR# Enable Zero for VFs. Behavior is set by PF bit
9	0b	RO	FB2BE - Fast Back-to-Back Enable Hardwired to 0b.
10	0b	RO	INTD - Interrupt Disable Hardwired to 0b.
15:11	0x0	RO	Reserved

Table 9-22 VF Status register (0x6; RW)

Bits	Initial Value	R/W	Description
2:0	000b	RO	Reserved
3	0b	RO	Interrupt Status Hardwired to 0b.
4	1b	RO	New Capabilities Indicates that the I350 VFs implement extended capabilities. The I350 VFs implement a capabilities list to indicate that it supports enhanced message signaled interrupts and PCIe extensions.
5	0b	RO	66MHz Capable Hardwired to 0b.
6	0b	RO	Reserved
7	0b	RO	Fast Back-to-Back Capable Hardwired to 0b.
8	0b	R/W1C	MPERR - Data Parity Reported
10:9	00b	RO	DEVSEL Timing Hardwired to 0b.
11	0b	R/W1C	STA - Signaled Target Abort
12	0b	R/W1C	RTA - Received Target Abort
13	0b	R/W1C	RMA - Received Master Abort
14	0b	R/W1C	SSERR - Signaled System Error
15	0b	R/W1C	DSERR - Detected Parity Error

9.7.2 Legacy Capabilities

9.7.2.1 MSI-X Capability

Table 9-23 MSI-X Control (0x72; RW)

Bits	Initial Value	Rd/Wr	Description
10:0	0x002[1]	RO	TS - Table Size

Table 9-23 MSI-X Control (0x72; RW) (Continued)

Bits	Initial Value	Rd/Wr	Description
13:11	000b	RO	Reserved
14	0b	RW	Mask - Function Mask
15	0b	RW	En - MSI-X Enable

1. Default value is read from I/O Virtualization (IOV) Control EEPROM word.

Table 9-24 MSI-X Table Offset (0x74; RO)

Bits	Default	Type	Description
31:3	0x000	RO	Table Offset Used as an offset from the address contained by one of the function's BARs to point to the base of the MSI-X table. The lower three table BIR bits are masked off (set to zero) by software to form a 32-bit Qword-aligned offset.
2:0	0x3	RO	Table BIR Indicates which one of a function's BARs, located beginning at 0x10 in configuration space, is used to map the function's MSI-X table into memory space. BIR values: 0...5 correspond to BARs 0x10...0x 24 respectively. A BIR value of 3 indicates that the table is mapped in BAR 3 (address 0x1C).

Table 9-25 MSI-X PBA Offset (0x78; RO)

Bits	Default	Type	Description
31:3	0x400	RO	PBA Offset Used as an offset from the address contained by one of the function's BARs to point to the base of the MSI-X PBA. The lower three PBA BIR bits are masked off (set to zero) by software to form a 32-bit Qword-aligned offset. This value changes according to the value set in the IOV System Page Size register, so that the offset of the PBA register is on a page boundary. The values by page sizes are: 4 KB: 0x400 8 KB: 0x400 64 KB: 0x1000 256 KB: 0x4000 1 MB: 0x10000 4 MB: 0x40000
2:0	0x3	RO	PBA BIR Indicates which one of a function's BARs, located beginning at 0x10 in configuration space, is used to map the function's MSI-X PBA into memory space. A BIR value of three indicates that the PBA is mapped in BAR 3.

9.7.2.2 PCIe Capability Registers

The device control and device status registers have some fields that are specific per VF.

Table 9-26 Device Control (0xA8; RW)

Bits	R/W	Default	Description
0	RO	0b	Correctable Error Reporting Enable Zero for VFs.
1	RO	0b	Non-Fatal Error Reporting Enable Zero for VFs.

Table 9-26 Device Control (0xA8; RW) (Continued)

Bits	R/W	Default	Description
2	RO	0b	Fatal Error Reporting Enable Zero for VFs.
3	RO	0b	Unsupported Request Reporting Enable Zero for VFs.
4	RO	0b	Enable Relaxed Ordering Zero for VFs.
7:5	RO	0b	Max Payload Size Zero for VFs.
8	RO	0b	Extended Tag field Enable Not implemented in the I350.
9	RO	0b	Phantom Functions Enable Not implemented in the I350.
10	RO	0b	Auxiliary Power PM Enable Zero for VFs.
11	RO	0b	Enable No Snoop Zero for VFs.
14:12	RO	000b	Max Read Request Size Zero for VFs.
15	RW	0b	Initiate Function Level Reset Specific to each VF.

Table 9-27 Device Status (0xAA; R/W1C)

Bits	R/W	Default	Description
0	R/W1C	0b	Correctable Detected Indicates status of correctable error detection.
1	R/W1C	0b	Non-Fatal Error Detected Indicates status of non-fatal error detection.
2	R/W1C	0b	Fatal Error Detected Indicates status of fatal error detection.
3	R/W1C	0b	Unsupported Request Detected Indicates that the I350 received an unsupported request. This field is identical in all functions. The I350 cannot distinguish which function caused an error.
4	RO	0b	Aux Power Detected Zero for VFs.
5	RO	0b	Transaction Pending Specific per VF. When set, indicates that a particular function (PF or VF) has issued non-posted requests that have not been completed. A function reports this bit cleared only when all completions for any outstanding non-posted requests have been received.
15:6	RO	0x00	Reserved

9.7.3 Advanced Capabilities

9.7.3.1 Advanced Error Reporting Registers

The following registers in the AER capability have a different behavior in a VF function.

Table 9-28 Uncorrectable Error Status (0x104; R/W1CS)

Bit Location	Attribute	Default Value	Description
3:0	RO	0000b	Reserved
4	RO	0b	Data Link Protocol Error Status
5	RO	0b	Surprise Down Error Status (Optional)
11:6	RO	0x0	Reserved
12	R/W1CS	0b	Poisoned TLP Status
13	RO	0b	Flow Control Protocol Error Status
14	R/W1CS	0b	Completion Timeout Status
15	R/W1CS	0b	Completer Abort Status
16	R/W1CS	0b	Unexpected Completion Status
17	RO	0b	Receiver Overflow Status
18	RO	0b	Malformed TLP Status
19	RO	0b	ECRC Error Status
20	R/W1CS	0b	Unsupported Request Error Status When caused by a function that claims a TLP.
21	RO	0b	ACS Violation Status Not supported in the I350.
22	RO	0b	Uncorrectable Internal Error Status (Optional) Not supported in the I350.
23	RO	0b	MC Blocked TLP Status (Optional) Not supported in the I350.
24	RO	0b	AtomicOps Egress Blocked Status (Optional) Not supported in the I350.
25	RO	0b	TLP Prefix Blocked Error Status (Optional) Not supported in the I350.
31:26	RO	0x0	Reserved

The Correctable Error Status register reports error status of individual correctable error sources on a PCIe device. When an individual error status bit is set to 1b, it indicates that a particular error occurred; software can clear an error status by writing a 1b to the respective bit. See the table below.

Table 9-29 Correctable Error Status (0x110; R/W1CS)

Bit Location	Attribute	Default Value	Description
0	RO	0b	Receiver Error Status
5:1	RO	0x0	Reserved
6	RO	0b	Bad TLP Status
7	RO	0b	Bad DLLP Status
8	RO	0b	REPLAY_NUM Rollover Status
11:9	RO	000b	Reserved
12	RO	0b	Replay Timer Timeout Status
13	R/W1CS	0b	Advisory Non-Fatal Error Status

Table 9-29 Correctable Error Status (0x110; R/W1CS) (Continued)

14	RO	0b	Reserved
15	RO	0b	Header Log Overflow Status (optional)
31:16	RO	0x0	Reserved

Table 9-30 Advanced Error Capabilities and Control Register (0x118; RWS)

Bit Location	Attribute	Default Value	Description
4:0	ROS	0x0	First Error Pointer The First Error Pointer is a field that identifies the bit position of the first error reported in the Uncorrectable Error Status register.
5	RO	0b	ECRC Generation Capable This bit indicates that the I350 is capable of generating ECRC. zeros for VFs. ***Note:***
6	RO	0b	ECRC Generation Enable Zero for VFs.
7	RO	0b	ECRC Check Capable Zero for VFs. ***Note:***
8	RO	0b	ECRC Check Enable Zero for VFs.
9	RO	0b	Multiple Header Recording Capable – If Set, this bit indicates that the Function is capable of recording more than one error header.
10	RO	0b	This bit enables the Function to record more than one error header. Functions that do not implement the associated mechanism are permitted to hardwire this bit to 0b.
11	RO	0b	TLP Prefix Log Present If Set and the First Error Pointer is valid, indicates that the TLP Prefix Log register contains valid information. If Clear or if First Error Pointer is invalid, the TLP Prefix Log register is undefined. Default value of this bit is 0b. This bit is RsvdP if the End-End TLP Prefix Supported bit is Clear.
31:12	RO	0x0	Reserved

NOTE: ***This page intentionally left blank.***

§ §

10 System Manageability

Network management is an important requirement in today's networked computer environment. Software-based management applications provide the ability to administer systems while the operating system is functioning in a normal power state (not in a pre-boot state or powered-down state). The Intel® System Management Bus (SMBus) Interface and the Network Controller Sideband Interface (NC-SI) fill the management void that exists when the operating system is not running or fully functional. This is accomplished by providing mechanisms by which manageability network traffic can be routed to and from a Management Controller (BMC).

This chapter describes the supported management interfaces and hardware configurations for platform system management. It describes the interfaces to an external BMC, the partitioning of platform manageability among system components, and the functionality provided by in each platform configuration.

10.1 Pass-Through (PT) Functionality

Pass-Through (PT) is the term used when referring to the process of sending and receiving Ethernet traffic over the sideband interface. The I350 has the ability to route Ethernet traffic to the host operating system as well as the ability to send Ethernet traffic over the sideband interface to an external BMC. See Figure 10-1.

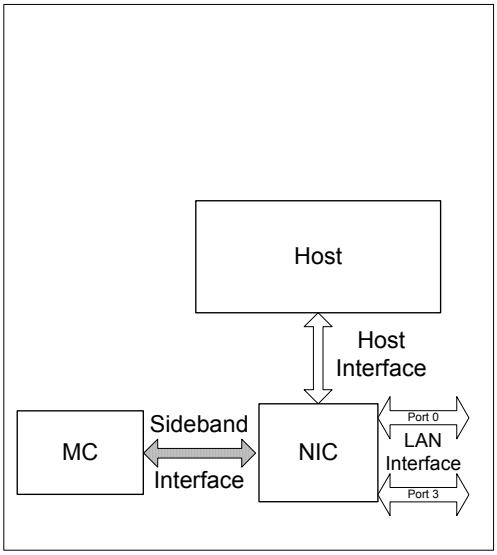


Figure 10-1 Sideband Interface

The sideband interface provides a mechanism by which the I350 can be shared between the host and the BMC. By providing this sideband interface, the BMC can communicate with the LAN without requiring a dedicated Ethernet controller. The I350 supports two sideband interfaces:

- SMBus
- NC-SI

The usable bandwidth for either direction is up to 400 Kb/s when using SMBus and 100 Mb/s for the NC-SI interface. Only one mode of sideband can be active at any given time. The configuration is done using an EEPROM setting.

10.1.1 Pass Through Packet Routing

When an Ethernet packet reaches the I350, it is examined and compared to a number of configurable filters. These filters are configurable by the BMC and include, but not limited to, filtering on:

- MAC Address
- IP Address
- UDP/IP Ports
- VLAN Tags
- EtherType

If the incoming packet matches any of the configured filters, it is passed to the BMC. Otherwise it is not passed.

The packet filtering process is described in Section 10.3.

10.2 Components of the Sideband Interface

There are two components to a sideband interface:

- Physical Layer
- Logical Layer

The BMC and the I350 must be in alignment for both components. An example issue: the NC-SI physical interface is based on the RMII interface, but there are differences between the devices at the physical level and the protocol layer is completely different.

10.2.1 Physical Layer

This is the electrical connection between the I350 and BMC.

10.2.1.1 SMBus

The SMBus physical layer is defined by the SMBus specification. The interface is made up of two connections: Data and Clock. There is also an optional third connection: the Alert line. This line is used by the I350 to notify the BMC that there is data available for reading. Refer to the SMBus specification for details.

10.2.1.2 NC-SI

The I350 uses the DMTF standard Sideband Interface. This interface consists of 6 lines for transmission and reception of Ethernet packets and two optional lines for arbitration among more than one physical network controller.

The physical layer of NC-SI is very similar to the RMII interface, although not an exact duplicate. Refer to the NC-SI specification for details of the differences.

10.2.2 Logical Layer

10.2.2.1 Legacy SMBus

The protocol layer for SMBus consists of commands the BMC issues to configure filtering for I350 management traffic and the reading and writing of Ethernet frames over the SMBus interface. There is no industry standard protocol for sideband traffic over SMBus. The protocol layer for SMBus on the I350 is Intel proprietary. The Legacy SMBus protocol is described in Section 10.5.

10.2.2.2 NC-SI

The DMTF also defines the protocol layer for the NC-SI interface. NC-SI compliant devices are required to implement a minimum set of commands. The specification also provides a mechanism for vendors to add additional capabilities through the use of OEM commands. Intel OEM NC-SI commands for the I350 are discussed in this document. For information on base NC-SI commands, see the NC-SI specification.

NC-SI traffic can run on top of two different Physical layers:

4. NC-SI Physical layer as described in Section 10.2.1.2.
5. MCTP over SMBus. This protocol allows control and traffic over SMBus of a NIC or a LOM device. The MCTP protocol is described in Section 10.7.

The I350 exposes one NC-SI package with four channels, one per port. The I350 implement a type C NC-SI interface (Single package, common bus buffers and shared RX queue) as described in section 5.2 of the NC-SI specification. The arbitration between the channels inside the package is done internally.

The package ID can be set either from the NVM *Package ID* field in the NC-SI Configuration - Offset 0x06 NVM word (Section 6.3.9.7) or from SDP0 pins of port 0 and 1. In this case, the Package ID is {0,Port1.SDP0, Port0.SDP0}. The mode used is set by the *Read NCSI Package ID from SDP* field in the NC-SI Configuration - Offset 0x07 NVM word (Section 6.3.9.8). Note that when the package ID is set from the SDP pins, the used SDPs should be set as input in the relevant Software Defined Pins Control NVM words.

10.3 Packet Filtering

Since both the host operating system and BMC use the I350 to send and receive Ethernet traffic, there needs to be a mechanism by which incoming Ethernet packets can be identified as those that should be sent to the BMC rather than the host operating system.

There are two different types of filtering available. The first is filtering based upon the MAC address. With this filtering, the BMC has at least one dedicated MAC address and incoming Ethernet traffic with the matching MAC address(es) are passed to the BMC. This is the simplest filtering mechanism to utilize and it allows an BMC to receive all types traffic (including, but not limited to, IPMI, NFS, HTTP etc).

The other mechanism available utilizes a highly configurable mechanism by which packets can be filtered using a wide range of parameters. Using this method, an BMC can share a MAC address (and IP address, if desired) with the host OS and receive only specific Ethernet traffic. This method is useful if the BMC is only interested in specific traffic, such as IPMI packets.

10.3.1 Manageability Receive Filtering

This section describes the manageability receive packet filtering flow. Packet reception by the I350 can generate one of the following results:

- Discarded
- Sent to Host memory
- Sent to the external BMC
- Sent to both the BMC and Host memory

The decisions regarding forwarding of packets to the Host and to the BMC are separate and are configured through two sets of registers. However, the BMC may define some types of traffic as exclusive. This traffic will be forwarded only to the BMC, even if it passes the filtering process of the Host. These types of traffic are defined using the MNGONLY register.

An example of packets that might be necessary to send exclusively to the BMC might be specific TCP/UDP ports of a shared MAC address or a MAC address dedicated to the BMC. If the BMC configures the manageability filters to send these ports to the BMC, it should configure the settings to not send them to the Host, otherwise, these ports will be received and handled by the Host operating system.

The BMC controls the types of packets that it receives by programming receive manageability filters. The following filters are accessible to the BMC:

Table 10-1 Filters Accessible to BMC

Filters	Functionality	When Reset?
Filters Enable	General configuration of the manageability filters	Internal Power On Reset
Manageability Only	Enables routing of packets exclusively to the manageability.	Internal Power On Reset
Manageability Decision Filters [7:0]	Configuration of manageability decision filters	Internal Power On Reset
MAC Address [3:0]	Four exact MAC manageability addresses	Internal Power On Reset
VLAN Filters [7:0]	Eight VLAN tag values	Internal Power On Reset
UDP/TCP Port Filters [3:0]	8 destination port values	Internal Power On Reset
Flexible 128 bytes TCO Filter	Length values for one flex TCO filter	Internal Power On Reset
IPv4 and IPv6 Address Filters [3:0]	IP address for manageability filtering	Internal Power On Reset

All filtering capabilities are available on both the NC-SI and legacy SMBus interfaces. However, in NC-SI mode, in order to program part of the capabilities, the Intel OEM commands described in Section 10.6.3 should be used.

All filters are reset only on Internal Power On Reset. Register filters that enable filters or functionality are also reset by firmware. These registers can be loaded from the EEPROM following a reset. See Section 6.1 for description of the location in the EEPROM map.

The high-level structure of manageability filtering is done using two steps.

1. The packet is parsed and fields in the header are compared to programmed filters.
2. A set of decision filters are applied to the result of the first step.

Some general rules apply:

- Fragmented packets are passed to manageability but not parsed beyond the IP header.
- Packets with L2 errors (CRC, alignment, etc.) are not forwarded to the BMC.
- Packets longer than 2KB are filtered out.

The following sections describe the manageability filtering, followed by the final filtering rules.

The filtering rules are created by programming the decision filters as described in Section 10.3.4.

10.3.2 L2 Filters

10.3.2.1 MAC and VLAN Filters

The manageability MAC filters allow comparison of the Destination MAC address to one of 4 filters defined in the *MMAH* and *MMAL* registers.

The VLAN filters allow comparison of the 12 bit VLAN tag to one of 8 filters defined in the *MAVTV* registers.

10.3.2.2 EtherType Filters

Manageability L2 EtherType filters allow filtering of received packets based on the Layer 2 EtherType field. The L2 type field of incoming packets is compared against the EtherType filters programmed in the Manageability EtherType Filter (*METF*; up to 4 filters); the result is incorporated into decision filters.

Each Manageability EtherType filter can be configured as pass (positive) or reject (negative) using a polarity bit. In order for the reverse polarity mode to be effective and block certain type of packets, the EtherType filter should be part of all the enabled decision filters.

An example for usage of L2 EtherType filters is to determine the destination of 802.1X control packets. The 802.1X protocol is executed at different times in either the management controller or by the Host. L2 EtherType filters are used to route these packets to the proper agent.

In addition to the flexible EtherType filters, the I350 supports 2 fixed EtherType filters used to block NC-SI control traffic and flow control traffic from reaching the manageability interface. The NC-SI EtherType is used for communication between the management controller on the NC-SI link and the I350. Packets coming from the network are not expected to carry this EtherType and such packets are blocked to prevent attacks on the management controller. Flow control packets should be consumed by the MAC and as such are not expected to be forwarded to the management interface.

Note: In order to get meaningful filtering of Ethertype packets, negative filters should be in the AND section. If more than one positive Ethertype filter is needed, then they should be set in the

OR section. A single positive Ethertype filter may be enabled both in the AND or in OR section.

10.3.3 L3/L4 Filtering

The manageability filtering stage combines checks done at previous stages with additional L3/L4 checks to make a the decision on whether to route a packet to the BMC. The following sections describe the manageability filtering done at layers L3/L4 and final filtering rules.

10.3.3.1 ARP Filtering

ARP filtering — The I350 supports filtering of ARP request packets (initiated externally) and ARP responses (to requests initiated by the BMC or Host).

In legacy SMBus mode, the ARP filters can be used as part of the ARP offload described in Section 10.5.4. ARP offload is not specifically available when using NC-SI. However, the general filtering mechanism is utilized to filter incoming ARP traffic as requested using the Enable Broadcast Filtering NC-SI command.

10.3.3.2 Neighbor Discovery Filtering

The I350 supports filtering of the following Neighbor Discovery packets:

1. 0x86 (134d) - Router Advertisement.
2. 0x87 (135d) - Neighbor Solicitation.
3. 0x88 (136d) - Neighbor Advertisement.
4. 0x89 (137d) - Redirect.

In SMBus mode, there is specific Neighborhood Discovery filter that can be enabled. The NC-SI interface does not have a filter for this. However, the general filtering mechanism can be utilized to filter this type of traffic.

10.3.3.3 RMCP Filtering

The I350 supports filtering by fixed destination port numbers, port 0x26F and port 0x298. These ports are IANA reserved for RMCP.

In SMBus mode, there are filters that can be enabled for these ports. When using NC-SI, they are not specifically available. However, the general filtering mechanism can be utilized to filter incoming ARP traffic.

10.3.3.4 Flexible Port Filtering

The I350 implements 8 flex destination port filters. The I350 directs packets whose L4 destination port matches to the BMC. The BMC must ensure that only valid entries are enabled in the decision filters.

10.3.3.5 Flexible 128 Byte Filter

The I350 provides one flex TCO filter. This filter looks for a pattern match within the first 128 bytes of the packet. The BMC must ensure that only valid entries are enabled in decision filters.

Flex filters are temporarily disabled when read from or written to by the Host. Any packet received during a read or write operation is dropped. Filter operation resumes once the read or write access completes.

10.3.3.5.1 Flexible Filter Structure

The filter is composed of the following fields:

1. Flexible Filter length — This field indicates the number of bytes in the packet header that should be inspected. The field also indicates the minimal length of packets inspected by the filter. Packet below that length will not be inspected. Valid values for this field are: 8*n, where n=1…16.
2. Data — This is a set of up to 128 bytes comprised of values that header bytes of packets are tested against.
3. Mask — This is a set of 128 bits corresponding to the 128 data bytes that indicate for each corresponding byte if is tested against its corresponding byte. The general filter is 128 bytes that the BMC configures; all of these bytes may not be needed or used for the filtering, so the mask is used to indicate which of the 128 bytes are used for the filter.

Each filter tests the first 128 bytes (or less) of a packet, where not all bytes must necessarily be tested.

10.3.3.5.2 TCO Filter Programming

Programming each filter is done using the following commands (NC-SI or SMBus) in a sequential manner:

1. Filter Mask and Length — This command configures the following fields:
   a. Mask — A set of 16 bytes containing the 128 bits of the mask. Bit 0 of the first byte corresponds to the first byte on the wire.
   b. Length — A 1-byte field indicating the length.
2. Filter Data — The filter data is divided into groups of bytes, described below:

Group	Test Bytes
0x0	0-29
0x1	30-59
0x2	60-89
0x3	90-119
0x4	120-127

   Each group of bytes need to be configured using a separate command, where the group number is given as a parameter. The command has the following parameters:
   a. Group number — A 1-byte field indicating the current group addressed
   b. Data bytes — Up to 30 bytes of test-bytes for the current group

10.3.3.6 IP Address Filtering

The I350 supports filtering by IP address using IPv4 and IPv6 address filters. These are dedicated to manageability. Two modes are possible, depending on the value of the MANC. EN_IPv4_FILTER bit:

- EN_IPv4_FILTER = 0b: the I350 provides four IPv6 address filters.
- EN_IPv4_FILTER = 1b: the I350 provides three IPv6 address filters and four IPv4 address filters.

10.3.3.7 Checksum Filtering

If bit *MANC.EN_XSUM_FILTER* is set, the I350 directs packets to the BMC only if they pass L3/L4 checksum (if they exist) in addition to matching other filters previously described.

Enabling the XSUM filter when using the SMBus interface is accomplished by setting the *Enable XSUM Filtering to Manageability* bit within the Manageability Control (MANC) register. This is done using the Update Management Receive Filter Parameters command. See Section 10.5.10.1.6**.**

To enable the XSUM filtering when using NC-SI, use the Enable Checksum Offloading command. See Section 10.6.3.13.

10.3.4 Configuring Manageability Filters

There are a number of pre-defined filters that are available for the BMC to enable, such as ARPs and IPMI ports 298h 26Fh. These are generally enabled by setting the appropriate bit within the MANC register using specific commands.

For more advanced filtering needs, the BMC has the ability to configure a number of configurable filters. It is a two-step process to use these filters. They must first be configured and then enabled.

10.3.4.1 Manageability Decision Filters (MDEF and MDEF_EXT)

Manageability decision filters are a set of eight filters, each with the same structure. The filtering rule for each decision filter is programmed by the BMC and defines which of the L2, VLAN, EtherType and L3/L4 filters participate in decision making. Any packet that passes at least one rule is directed to manageability and possibly to the Host.

With the I350, packets can also be filtered by EtherType. This is part of the Extended Manageability Decision Filters (MDEF_EXT).

The inputs to each decision filter are:

- Packet passed a valid management L2 exact address filter.
- Packet is a broadcast packet.
- Packet has a VLAN header and it passed a valid manageability VLAN filter.
- Packet matched one of the valid IPv4 or IPv6 manageability address filters.
- Packet is a multicast packet.
- Packet passed ARP filtering (request or response).
- Packet passed neighbor solicitation filtering.
- Packet passed 0x298/0x26F port filter.

- Packet passed a valid flex port filter.
- Packet passed a valid flex TCO filter.
- Packet passed or failed an L2 EtherType filter.
- Packet passed or failed Flow Control or NC-SI L2 EtherType Discard filter.

The structure of each decision filter is shown in Figure 10-2. A boxed number indicates that the input is conditioned by a mask bit defined in the MDEF register and MDEF_EXT register for this rule. Decision filter rules are as follows:

- At least one bit must be set in a register. If all bits are cleared (MDEF/MDEF_EXT = 0x0000), then the decision filter is disabled and ignored.
- All enabled AND filters must match for the decision filter to match. An AND filter not enabled in the MDEF/MDEF_EXT registers is ignored. If an AND filter is preceded by a OR filter, then at least one of the enabled OR inputs must match for the filter to pass.
- If no OR filter is enabled in the register, the OR filters are ignored in the decision (the filter might still match).
- If one or more OR filters are enabled in the register, then at least one of the enabled OR filters must match for the decision filter to match.

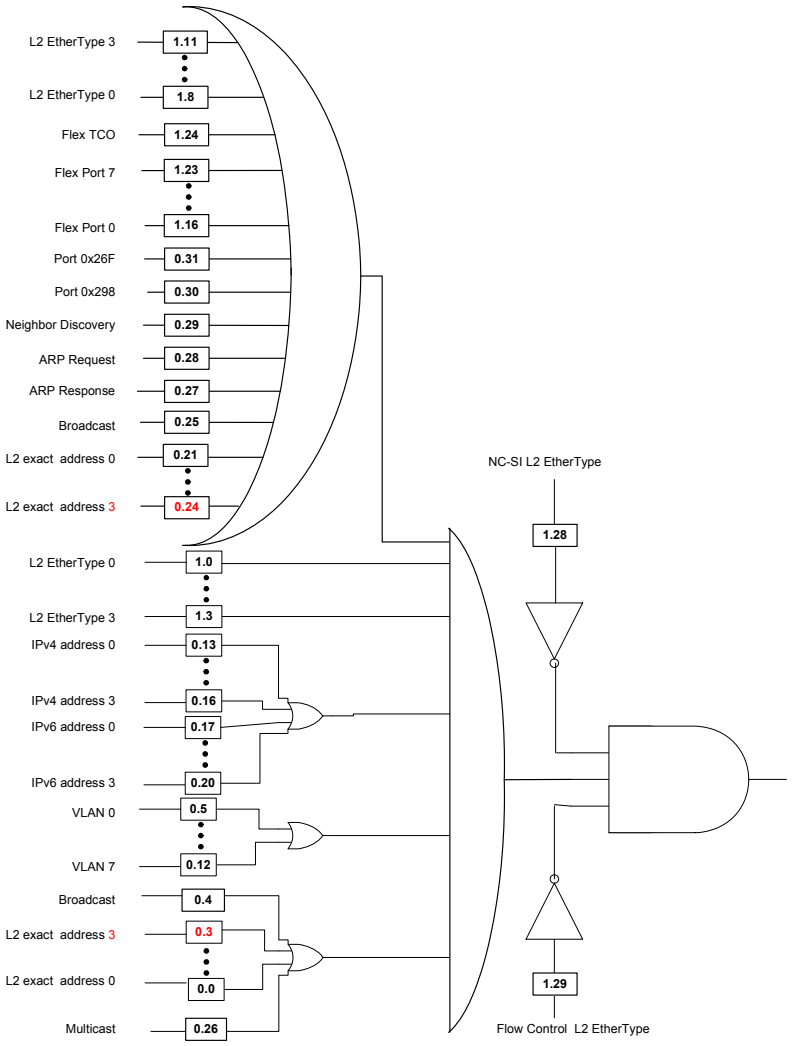


Figure 10-2 Manageability Decision Filters

A decision filter (for any of the 8 filters) defines which of the above inputs is enabled as part of a filtering rule. The BMC programs two 32-bit registers per rule (MDEF[7:0] & MDEF_EXT[7:0]) with the settings as described in Section 8.21.6 and Section 8.21.7. A set bit enables its corresponding filter to participate in the filtering decision.

10.3.4.2 Exclusive Traffic

The decisions regarding forwarding of packets to the Host for LAN traffic or to the LAN for Host traffic are independent from the management decision filters. However, the BMC may define some types of traffic as exclusive. The behavior for such traffic is defined by the using the bits corresponding to the decision filter in the *MNGONLY* register (one bit per each of the eight decision rules) and the *MDEF_EXT.apply_to_host_traffic* and *MDEF_EXT.apply_to_network_traffic* bits. Table 10-3 describes the behavior in each case. If one or more filters match the traffic and at least one of the filters is set as exclusive, the traffic is treated as exclusive.

Table 10-3 Exclusive traffic behavior

	Filter match		Filter doesn't match
traffic source	**MNGONLY = 0**	**MNGONLY = 1**	**N/A**
From network	Traffic is forwarded to the manageability. Traffic is forwarded to the Host according to Host filtering	Traffic is forwarded only to manageability.	Traffic is forwarded to the Host according to Host filtering
From Host	Traffic is forwarded to the manageability and to the LAN	Traffic is forwarded only to manageability.	Traffic is forwarded to the LAN

Any traffic matching any of the configurable filters (see Section 10.3.4.1) can be used as filters to pass traffic to the Host.

Table 10-4 *MNGONLY* register description and usage

Bits	Description	Default
0	Decision Filter 0	Determines if packets that have passed decision filter 0 are sent exclusively to the manageability path.
1	Decision Filter 1	Determines if packets that have passed decision filter 1are sent exclusively to the manageability path.
2	Decision Filter 2	Determines if packets that have passed decision filter 2 are sent exclusively to the manageability path
3	Decision Filter 3	Determines if packets that have passed decision filter 3 are sent exclusively to the manageability path
4	Decision Filter 4	Determines if packets that have passed decision filter 4 are sent exclusively to the manageability path
5	Unicast and Mixed	NC-SI mode: Determines if unicast and mixed packets are sent exclusively to the manageability path SMBus mode: Determines if packets that have passed decision filter 5 are sent exclusively to the manageability path
6	Global Multicast	NC-SI mode: Determines if multicast packets are sent exclusively to the manageability path SMBus mode: Determines if packets that have passed decision filter 6 are sent exclusively to the manageability path
7	Broadcast	NC-SI mode: Determines if broadcast packets are sent exclusively to the manageability path SMBus mode: Determines if ARP packets are sent exclusively to the manageability path
31:8	Reserved	Reserved

When using the SMBus interface, the BMC enables these filters by issuing the Update Management Receive Filter Parameters command (see Section 10.5.10.1.6) with the parameter of 0x0F.

The MNGONLY is also configurable when using NC-SI using the Set Intel Filters — Manageability Only Command (see Section 10.6.3.5.3).

All manageability filters are controlled by the BMC only and not by the LAN device driver.

10.3.5 Possible Configurations

This section describes ways of using management filters. Actual usage may vary.

10.3.5.1 Dedicated MAC Packet Filtering

- Select one of the eight rules for dedicated MAC filtering.
- Load Host MAC address to one of the management MAC address filters and set the appropriate bit in field 3:0 of the *MDEF* register.
- Set other bits to qualify which packets are allowed to pass through. For example:
  - Load one or more management VLAN filters and set the appropriate bits in field 12:5 of the *MDEF* register to qualify the relevant manageability VLANs.
  - Set relevant bits in field 20:13 of the *MDEF* register to qualify with a match to one of the IP addresses.
  - Set any L3/L4 bits (bits 31:27 in the *MDEF* register and bits 23:16 in the *MDEF_EXT register*) to filter using any set of L3/L4 filters.

10.3.5.2 Broadcast Packet Filtering

- Select one of the eight rules for broadcast filtering.
- Set bit 25 in the *MDEF* register of the decision rule to enforce broadcast filtering.
- Set other bits to qualify which broadcast packets are allowed to pass through. For example:
  - Set bit 5 in the *MDEF* register to filter with the first manageability VLAN.
  - Set relevant bits in field 20:13 of the *MDEF* register to qualify with a match to one of the IP addresses.
  - Set any L3/L4 bits (bits 31:27 in the *MDEF* register and bits 23:16 in the *MDEF_EXT* register) to filter with any set of L3/L4 filters.

10.3.5.3 VLAN Packet Filtering

- Select one of the eight rules for VLAN filtering.
- Load one or more management VLAN filters and set the appropriate bits in field 12:5 of the *MDEF* register to qualify the relevant manageability VLANs.
- Set other bits to qualify which VLAN packets are allowed to pass through. For example:
  - Set any L3/L4 bits (bits 31:27 in the *MDEF* register and bits 23:16 in the *MDEF_EXT* register) to filter using appropriate L3/L4 filter set.

10.3.5.4 IPv6 Filtering

IPv6 filtering is done using the following IPv6-specific filters:

- IP Unicast filtering — requires filtering for Link Local address and a Global address. Filtering setup might depend on whether or not the MAC address is shared with the Host or dedicated to manageability:
  - Dedicated MAC address (for example, dynamic address allocation with DHCP does not support multiple IP addresses for one MAC address). In this case, filtering can be done at L2 using two dedicated unicast MAC filters.

  - — Shared MAC address (for example, static address allocation sharing addresses with Host). In this case, filtering needs to be done at L3, requiring two IPv6 address filters, one per address.
- A neighbor Discovery filter — The I350 supports IPv6 neighbor Discovery protocol. Since the protocol relies on multicast packets, the I350 supports filtering of these packets. IPv6 multicast addresses are translated into corresponding Ethernet multicast addresses in the form of 33-33-xx-xx-xx-xx, where the last 32 bits of address are taken from the last 32 bits of the IPv6 multicast address. As a result, two direct MAC filters can be used to filter IPv6 solicited-node multicast packets as well as IPv6 all node multicast packets.

10.3.5.5 Receive Filtering with Shared IP

When using the SMBus interface, it is possible to share the Host MAC and IP address with the BMC. This functionality is also available when using NC-SI using Intel OEM commands.

When the BMC shares the MAC and IP address with the Host, receive filtering is based on identifying specific flows through port allocation. The following setting might be used:

- Select one of the eight rules for Dedicated MAC filtering.
- Load Host MAC address to one of the management MAC address filters and set the appropriate bit in field 3:0 of the *MDEF* register to enforce MAC address filtering using the MAC address.
- If VLAN is used for management, load one or more management VLAN filters and set the appropriate bits in field 12:5 of the *MDEF* register to qualify the relevant manageability VLANs.
- ARP filter/Neighbor Discovery filter is enabled when the BMC is responsible for handling the ARP protocol. Set bit 27 or bit 28 in the *MDEF* register for this functionality.
- Set other bits to qualify which packets are allowed to pass through. For example:
  - — Set any L3/L4 bits (bits 31:27 in the *MDEF* register and bits 23:16 in the *MDEF_EXT* register) to filter using the appropriate L3/L4 filters.

10.3.6 Determining Manageability MAC Address

If the BMC wishes to use a dedicated MAC address or configure the automatic ARP response mechanism (only available in SMBus mode), it may be beneficial for the BMC to be able to determine the MAC address used by the Host.

Both the NC-SI and SMBus interfaces provide an Intel OEM command to read the System MAC address.

A possible use for this is that the MAC address programmed at manufacturing time does not increment by one each time, but rather by two. In this way, the BMC can read the System MAC address and add one to it and be guaranteed of a unique MAC address.

Determining the IP address being used by the Host is beyond the scope of this document.

10.4 OS to BMC Traffic

10.4.1 Overview

Traditionally, the communication between a Host and the local BMC is not handled through the network interface and requires a dedicated interface such as an IPMI KCS interface. The I350 allows the Host and the local BMC communication via the regular pass-through interface, and thus allow management of a local console using the same interface used to manage any BMC in the network.

When this flow is used, the Host will send packets to the BMC through the network interface. The I350 will examine these packets and it will then decide if they should be forwarded to the BMC. On the inverse path, when the BMC sends a packet on the pass-through interface, the I350 will check if it should be forwarded to the network, the Host, or both. Figure 10-3 describes the flow for OS to BMC traffic for the NC-SI over RMII case. OS2BMC is not available in legacy SMBus mode.

The OS to BMC flow can be enabled using the *OS2BMC enable* field for the relevant port in the OS 2 BMC configuration structure of the EEPROM.

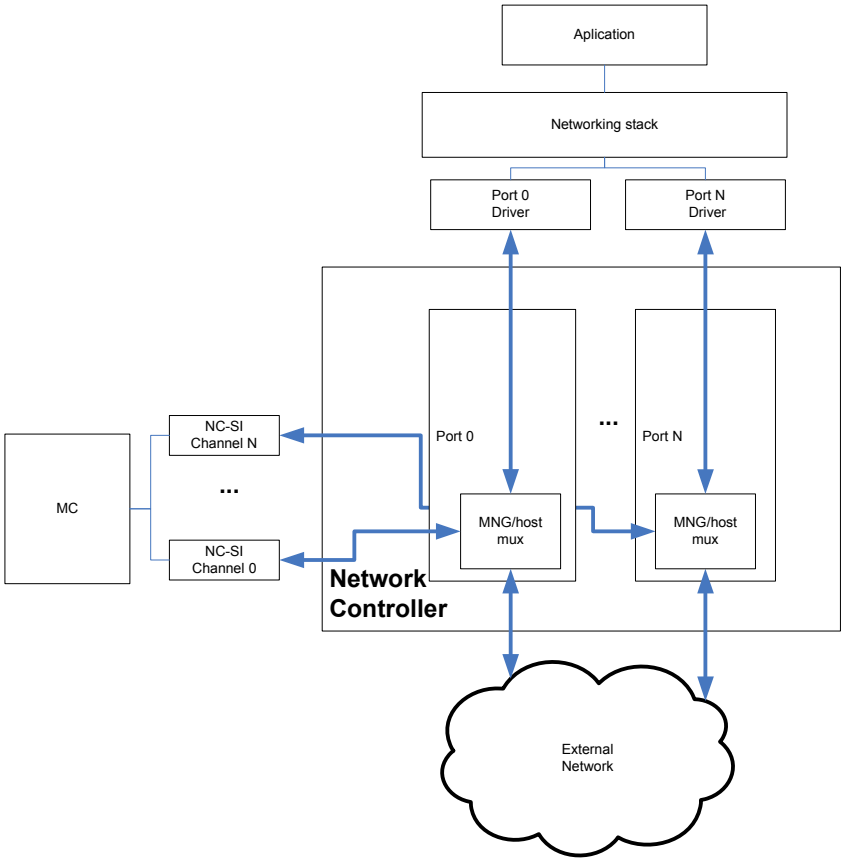


Figure 10-3 OS 2 BMC Diagram

Note: This flow assumes that the BMC does not share a MAC address with the Host.

The OS to BMC flow is enabled only for ports enabled by the NC-SI "Enable Channel" command or via the *OS to BMC Enable* field for the relevant port in the OS-to-BMC configuration structure of the EEPROM.

OS2BMC traffic shall comply with NC-SI specifications and is therefore limited to maximum sized frames of 1536 bytes (in both directions).

10.4.2 Filtering

10.4.2.1 OS2BMC Filtering

When OS to BMC traffic is enabled, the filters used for network to BMC traffic are also used for OS to BMC traffic. Traffic considered as exclusive to the BMC (Relevant bit in MNGONLY is set) is also considered as exclusive to the BMC when sent from the Host and not forwarded to the network.

VM to VM switching is considered only for packets that are forwarded to the network by the OS to BMC filtering process. Thus a packet will be sent to the network only if it wasn't defined as exclusive by the manageability path and by the VM to VM switching.

The filtering of the BMC precedes the VM to VM filtering and thus a packet is candidate for VM to VM switching only if the OS to BMC filtering decided to send the packet to the network as described in Figure 10-4.

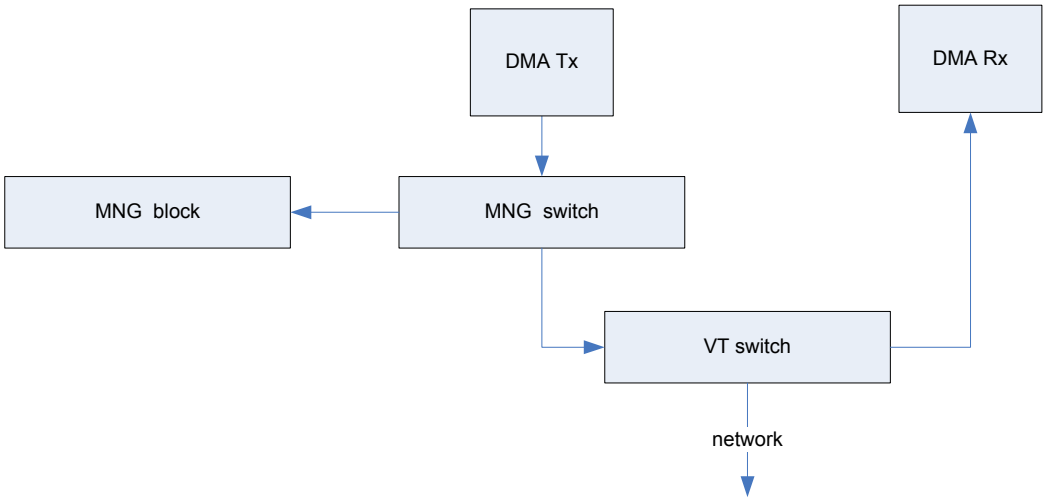


Figure 10-4 OS 2 BMC and VM to VM filtering

10.4.2.2 Handling of OS to BMC Packets

All the regular transmit offloads are available for OS to BMC packets also.

10.4.2.3 BMC to OS Filtering

When OS to BMC is enabled, as with regular BMC transmit traffic, the port (OS or network) to which the packet is sent is fixed according to the source MAC address of the packet.

After that, the BMC traffic will be filtered according to the L2 Host filters of the selected port (as described in Section 7.1.3). According to the results of the filtering the packet can be forwarded to the OS, the network or both.

The following rules apply to the forwarding of OS packets:

If *BMC to net* is disabled, all the traffic from the BMC is sent to the Host.

If *BMC to host* is disabled, all the traffic from the BMC is sent to the network.

The packet will be forwarded only according to the destination MAC address and VLAN tag.

When working in non VMDq modes (*MRQC.Multiple Receive Queues Enable* field not equal 011b), unicast packets that matches one of the exact filters (*RAH*/*RAL*) are sent only to the Host. Other packets that passes the L2 Host filtering will be sent to both the Host and the network. Packets that do not pass the L2 filtering will be sent only to the network.

When working in VMDq mode (*MRQC.Multiple Receive Queues Enable* field =011b), if a packet passes L2 filtering, the forwarding decisions are based on the Tx switching algorithm described in Section 7.8.3.5. A packet that does not pass L2 filtering is sent to the network.

Packets for which the *LVLAN* bit is set in the *VLVF* matching their VLAN are not forwarded to the host.

10.4.2.4 Queuing of Packets Received from the BMC.

The traffic of the BMC to the Host can be queued according to the following mechanisms.

1. VMDq - In this mode, the packets will be queued according to the destination MAC address and VLAN. This mode is enabled by setting the *MRQC.Multiple Receive Queues Enable* field to 011b
2. If the previous modes is not used, the packets are forwarded to queue *MRQC.Def_Q*.

10.4.2.5 Offloads of Packets received from the BMC.

Packets received from the BMC and forwarded to the OS do not pass the same path as regular network packets. Thus parts of the offloads provided for the network packets are not available for the BMC packets. Packet received from the BMC are identified by the *RDESC.STATUS.BMC* bit.

The following list describes which offloads are available for BMC packets:

- CRC is checked and removed on the BMC packets. The *RDESC.STATUS.Strip CRC* will always be set for these packets.
- The RSS type and RSS hash are not calculated for BMC packets and are always set to zero.
- The header of BMC packets is never split.
- A fragmented BMC packet will not be detected by the hardware.
- The BMC packets are not detected as time sync packet. The *RDESC.STATUS.TS* will always be clear for these packets.
- The L3 and L4 checksum are not performed on these packets. The *L4I*, *IPCS*, *UDPCS*, and *UDPV* fields will always be cleared for these packets.

- In systems where the double VLAN feature is enabled (*CTRL_EXT.EXT_VLAN* is set), the VEXT bit is valid for BMC packets.

Note: In systems that uses double VLAN, the BMC is expected to send all packets (apart from NC-SI commands) with the outer VLAN included. Failing to do so may cause corruptions to the packet received by the OS

- The *RDESC.ERRORS* field is always cleared for these packets.

Note: Traffic sent from the BMC will not cause a PME event, even if it matches one of the wake-up filters set by the port.

10.4.3 Blocking of Network to BMC Flow

In some systems the BMC may have its own private connection to the network and may use the I350 port only for the OS to BMC traffic. In this case, the BMC to network flow should be blocked while enabling the OS to BMC and OS to network flows.

This can be done by clearing the *MANC.EN_BMC2NET* bit for the relevant port. The BMC can control this functionality using the "Enable Network to BMC flow" and "Disable Network to BMC flow" NC-SI OEM commands. This can also be controlled using the *Network to BMC disable* field in the EEPROM "OS2BMC Configuration Structure".

Note: When network to BMC flow is blocked and OS to BMC flow is enabled, the traffic from the BMC is sent to the OS only if it passes the host L2 filtering. Other packets are dropped. The OS traffic filtering is still done using the regular decision filters.

10.4.4 Statistics

Packets sent from the OS to the BMC should be counted by all statistical counters as packets sent by the OS. If they are sent to both the network and to the BMC, then they are counted once.

Packets sent from the BMC to the Host are counted as packets received by the Host. If they are sent to the Host and to the network, then they are counted both as received packets and as packet transmitted to the network.

In addition, the I350 supports the following statistical counters that measure just the BMC to OS and OS to BMC traffic:

- O2BGPTC - OS2BMC packets received by BMC
- O2BSPC - OS2BMC packets transmitted by OS
- B2OSPC - BMC2OS packets sent by BMC
- B2OGPRC - BMC2OS packets received by OS.

The driver can use these statistics to count packets dropped by the I350 during the transfer between the OS and the BMC.

See Section 7.10.5 for details of the statistics hierarchy.

10.4.5 OS to BMC Enablement

The I350 supports the unified network software model for OS to BMC traffic, where the OS to BMC traffic is shared with the regular traffic. In this model, there is no need for a special configuration of the OS networking stack or the BMC stack, but if the link is down, then the OS to BMC communication is stopped.

In order to enable OS to BMC either:

- Enable OS2BMC in the port traffic type field in the Traffic type Parameters EEPROM word for the relevant port.
- Send an *EnableOS2BMC Flow* NC-SI OEM Command.

Note: When OS2BMC is enabled, OS shall avoid sending packets longer than 1.5KB to BMC. Such packets will be dropped.

10.5 SMBus Pass-Through Interface

SMBus is the system management bus defined by Intel. It is used in personal computers and servers for low-speed system management communications. This section describes how the SMBus interface operates in pass-through mode.

10.5.1 General

The SMBus sideband interface includes standard SMBus commands used for assigning a slave address and gathering device information as well as Intel proprietary commands used specifically for the pass-through interface.

10.5.2 Pass-Through Capabilities

This section details manageability capabilities the I350 provides while in SMBus mode. Pass-through traffic is carried by the sideband interface as described in Section 10.1.

These services are not available in NC-SI mode.

When operating in SMBus mode, in addition to exposing a communication channel to the LAN for the BMC, the I350 provides the following manageability services to the BMC:

- ARP handling — The I350 can be programmed to auto-ARP replying for ARP request packets to reduce the traffic over the BMC interconnect.
- Default configuration of filters by EEPROM - When working in SMBus mode, the default values of the manageability receive filters can be set according to the PT LAN and flex TCO EEPROM structures.

10.5.3 Port to SMBus Mapping

The I350 is presented on the SMBus manageability link as four different devices (for example, via four different SMBus addresses on which each device is connected to a different LAN port). There is no logical connection between the four devices.

The fail-over between the LAN ports is done by the BMC (by sending/receiving packets through different devices). The status report to the BMC, ARP handling, DHCP, and other pass-through functionality are unique for each port and configured by the BMC.

10.5.4 Automatic Ethernet ARP Operation

The I350 can offload the Ethernet Address Resolution Protocol (ARP) for the BMC in order to reduce the bandwidth required on the SMBus link.

Automatic Ethernet ARP parameters are loaded from the EEPROM when the I350 is powered up or configured through the sideband management interface. The following parameters should be configured in order to enable ARP operation:

- ARP auto-reply enabled
- ARP IP address (to filter ARP packets)
- ARP MAC addresses (for ARP responses)

These are all configurable over the sideband interface using the advanced version of the Receive Enable command.

When an ARP request packet is received and ARP auto-reply is enabled, the I350 checks the targeted IP address (after the packet has passed L2 checks and ARP checks). If the targeted IP matches the IP configuration for the I350, it replies with an ARP response.

The I350 responds to ARP request targeted to the ARP IP address with the configured ARP MAC address. In case that there is no match, the I350 silently discards the packets. If the I350 is not configured to do auto-ARP response, it can be configured to forward the ARP packets to the BMC (which can respond to ARP requests).

When the external BMC uses the same IP and MAC address of the OS, the ARP operation should be coordinated with the Host operating system.

Note: If sharing the MAC and IP with the Host operating system is possible, the I350 provides the ability to read the stem MAC address, allowing the BMC to share the MAC address. There is no mechanism however provided by the I350 to read the IP address. The Host OS (or an agent within) and BMC must coordinate the sharing of IP addresses.

10.5.4.1 ARP Packet Formats

Table 10-5 ARP Request Packet

Offset	# Of bytes	Field	Value (In Hex)	Action
0	6	Destination Address		Compare
6	6	Source Address		Stored
12	S=(0/4)	Possible VLAN Tag		Stored
12 + S	D=(0/8)	Possible Length + LLC/SNAP Header		Stored
12 + S + D	2	Type	0806	Compare
14+ S + D	2	HW Type	0001	Compare
16+ S + D	2	Protocol Type	0800	Compare

Table 10-5 ARP Request Packet (Continued)

18+ S + D	1	Hardware Size	06	Compare
19+ S + D	1	Protocol Address Length	04	Compare
20+ S + D	2	Operation	0001	Compare
22+ S + D	6	Sender HW Address	-	Stored
28+ S + D	4	Sender IP Address	-	Stored
32+ S + D	6	Target HW Address	-	Ignore
38+ S + D	4	Target IP Address	ARP IP address	Compare

Table 10-6 ARP Response Packet

Offset	# of bytes	Field	Value
0	6	Destination Address	ARP Request Source Address
6	6	Source Address	Programmed from EEPROM or BMC
12	S=(0/4)	Possible VLAN Tag	From ARP Request
12 + S	D=(0/8)	Possible Length + LLC/SNAP Header	From ARP Request
12 + S + D	2	Type	0x0806
14+ S + D	2	HW Type	0x0001
16+ S + D	2	Protocol Type	0x0800
18+ S + D	1	Hardware Size	0x06
19+ S + D	1	Protocol Address Length	0x04
20+ S + D	2	Operation	0x0002
22+ S + D	6	Sender HW Address	Programmed from EEPROM or BMC
28+ S + D	4	Sender IP Address	Programmed from EEPROM or BMC
32 +S + D	6	Target HW Address	ARP Request Sender HW Address
38 +S + D	4	Target IP Address	ARP Request Sender IP Address

10.5.5 SMBus Transactions

This section gives a brief overview of the SMBus protocol. Following is an example for a format of a typical SMBus transaction.

1	7	1	1	8	1	8	1	1
S	Slave Address	Wr	A	Command	A	PEC	A	P
	1100 001	0	0	0000 0010	0	[Data Dependent]	0	

The top row of the table identifies the bit length of the field in a decimal bit count. The middle row (bordered) identifies the name of the fields used in the transaction. The last row appears only with some transactions, and lists the value expected for the corresponding field. This value can be either hexadecimal or binary.

The SMBus controller is a master for some transactions and a slave for others. The differences are identified in this document.

Shorthand field names are listed in Table 10-8 and are fully defined in the SMBus specification.

Table 10-8 Shorthand Field Names

Field Name	Definition
S	SMBus START Symbol
P	SMBus STOP Symbol
PEC	Packet Error Code
A	ACK (Acknowledge)
N	NACK (Not Acknowledge)
Rd	Read Operation (Read Value = 1b)
Wr	Write Operation (Write Value = 0b)

10.5.5.1 SMBus Addressing

The SMBus is presented as four SMBus devices on the SMBus (four addresses). All pass-through functionality is duplicated on the SMBus address, where each SMBus address is connected to a different LAN port. Note that it is not permitted to configure different ports to the same SMBus address. When a LAN function is disabled, the corresponding SMBus address is not presented to the BMC.

SMBus addresses (enabled from the EEPROM) can be re-assigned using the SMBus ARP protocol.

In addition to the SMBus address values, all parameters of the SMBus (SMBus channel selection, address mode, and address enable) can be set only through EEPROM configuration. Note that the EEPROM is read at the I350's power up and resets.

All SMBus addresses should be in Network Byte Order (NBO); MSB first.

10.5.5.2 SMBus ARP Functionality

The I350 supports the SMBus ARP protocol as defined in the SMBus 2.0 specification. The I350 is a persistent slave address device so its SMBus address is valid after power-up and loaded from the EEPROM. The I350 supports all SMBus ARP commands defined in the SMBus specification both general and directed.

SMBus ARP capability can be disabled through the EEPROM.

10.5.5.3 SMBus ARP Flow

SMBus ARP flow is based on the status of two flags:

- AV (Address Valid): This flag is set when the I350 has a valid SMBus address.
- AR (Address Resolved): This flag is set when the I350 SMBus address is resolved (SMBus address was assigned by the SMBus ARP process).

These flags are internal I350 flags and are not exposed to external SMBus devices.

Since the I350 is a Persistent SMBus Address (PSA) device, the AV flag is always set, while the AR flag is cleared after power up until the SMBus ARP process completes. Since AV is always set, the I350 always has a valid SMBus address.

When the SMBus master needs to start an SMBus ARP process, it resets (in terms of ARP functionality) all devices on SMBus by issuing either Prepare to ARP or Reset Device commands. When the I350 accepts one of these commands, it clears its AR flag (if set from previous SMBus ARP process), but not its AV flag (the current SMBus address remains valid until the end of the SMBus ARP process).

Clearing the AR flag means that the I350 responds to SMBus ARP transactions that are issued by the master. The SMBus master issues a Get UDID command (general or directed) to identify the devices on the SMBus. The I350 always responds to the Directed command and to the General command only if its AR flag is not set.

After the Get UDID, The master assigns the I350 SMBus address by issuing an Assign Address command. The I350 checks whether the UDID matches its own UDID and if it matches, it switches its SMBus address to the address assigned by the command (byte 17). After accepting the Assign Address command, the AR flag is set and from this point (as long as the AR flag is set), the I350 does not respond to the Get UDID General command. Note that all other commands are processed even if the AR flag is set. The I350 stores the SMBus address that was assigned in the SMBus ARP process in the EEPROM, so at the next power up, it returns to its assigned SMBus address.

Figure 10-5 shows the I350 SMBus ARP flow.

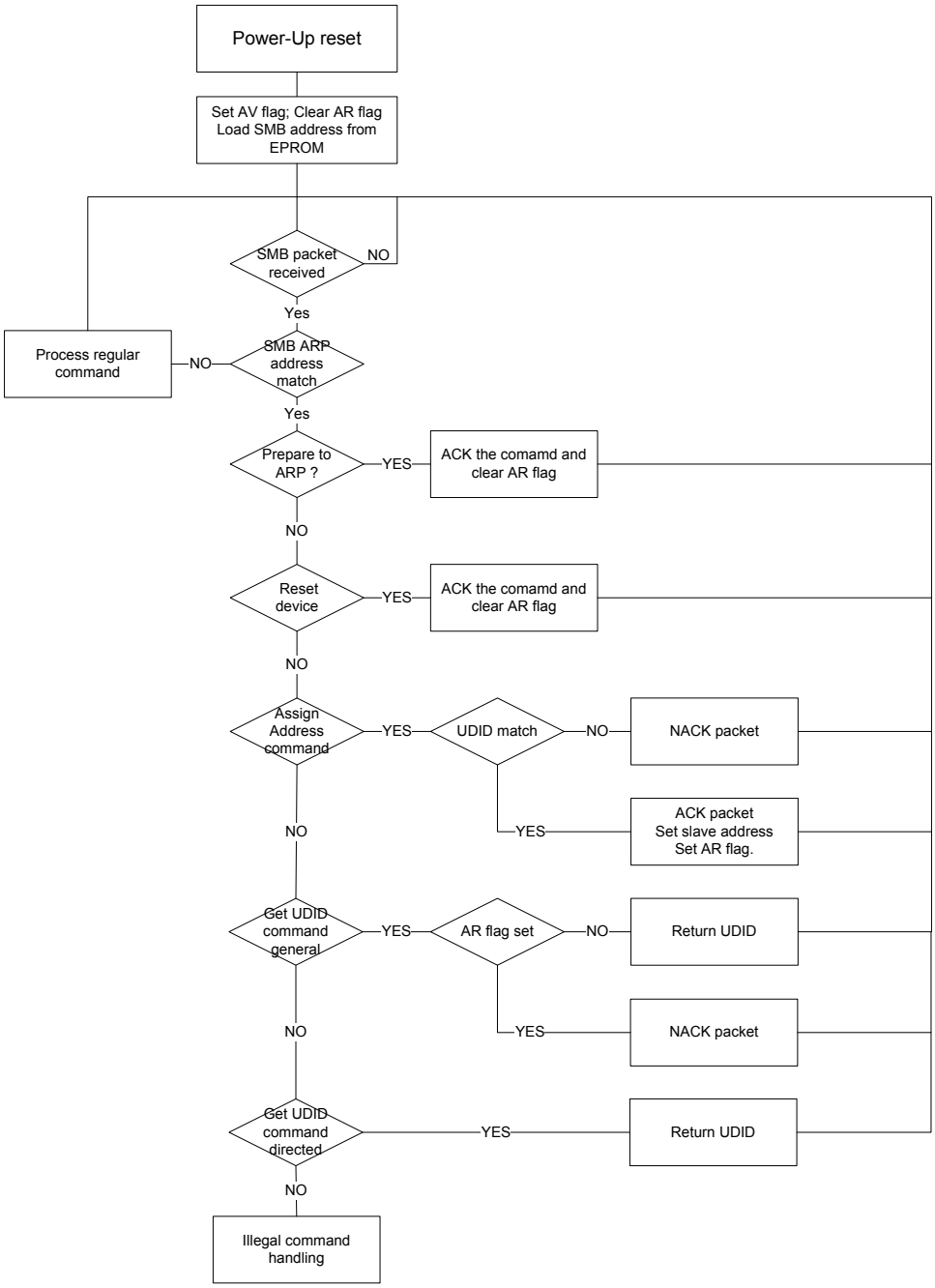


Figure 10-5 SMBus ARP Flow

10.5.5.4 SMBus ARP UDID Content

The UDID provides a mechanism to isolate each device for the purpose of address assignment. Each device has a unique identifier. The 128-bit number is comprised of the following fields:

1 Byte	1 Byte	2 Bytes	2 Bytes	2 Bytes	2 Bytes	2 Bytes	4 Bytes
Device Capabilities	Version/ Revision	Vendor ID	Device ID	Interface	Subsystem Vendor ID	Subsystem Device ID	Vendor Specific ID
See notes that follow	See notes that follow	0x8086	0x151F	0x0004/ 0x0024	0x0000	0x0000	See notes that follow
MSB							LSB

Where:

Vendor ID:	The device manufacturer's ID as assigned by the SBS Implementers' Forum or the PCI SIG. Constant value: 0x8086
Device ID:	The device ID as assigned by the device manufacturer (identified by the Vendor ID field). Constant value: 0x151F
Interface:	Identifies the protocol layer interfaces supported over the SMBus connection by the device. Bits 3:0 = 0x4 indicates SMBus Version 2.0 Bit 5 (ASF bit) = 1 in MCTP mode.
Subsystem Fields:	These fields are not supported and return zeros.

Device Capabilities: Dynamic and Persistent Address, *PEC Support* bit:

7	6	5	4	3	2	1	0
Address Type		Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)	Reserved (0)	PEC Supported
0b	1b	0b	0b	0b	0b	0b	0b
MSB							LSB

Version/Revision: UDID Version 1, Silicon Revision:

7	6	5	4	3	2	1	0
Reserved (0)	Reserved (0)	UDID Version			Silicon Revision ID		
0b	0b	001b			See the following table		
MSB							LSB

Silicon Revision ID:

Silicon Version	Revision ID
A0	000b
A1	001b
A2	010b

Vendor Specific ID: Four LSB bytes of the device Ethernet MAC address of the relevant port. The device Ethernet address is taken from the EEPROM. Note that in the I350 there are four MAC addresses (one for each port).

1 Byte	1 Byte	1 Byte	1 Byte
MAC Address, Byte 3	MAC Address, Byte 2	MAC Address, Byte 1	MAC Address, Byte 0
MSB			LSB

10.5.5.5 SMBus ARP in Multi/Single Mode

The I350 responds as four SMBus devices having four sets of AR/AV flags (one for each port). The I350 responds four times to the SMBus ARP master, once each for each port. All SMBus addresses are taken from the SMBus ARP address word in the EEPROM.

Note that the Unique Device Identifier (UDID) is different for the four ports in the version ID field (which represents the MAC address and is different for the four ports). It is recommended that the I350 first respond as port 0, and only when an address is assigned, then start responding as port 1,2 and 3 to the Get UDID command.

10.5.5.6 Concurrent SMBus Transactions

The SMBus interface is single threaded. Thus, concurrent SMBus transactions are not permitted. Once a transaction is started, it must be completed before additional transaction can be initiated.

A transaction is defined as:

- All SMBus commands used to receive a packet.
- All SMBus commands used to send a packet.
- The read and write SMBus commands used as part of read parameters described in Section 10.5.10.2.
- The single write SMBus commands described in Section 10.5.10.1.

10.5.6 SMBus Notification Methods

The I350 supports three methods of notifying the BMC that it has information that needs to be read by the BMC:

- SMBus alert - Refer to Section 10.5.6.1.
- Asynchronous notify - Refer to Section 10.5.6.2.
- Direct receive - refer to Section 10.5.6.3.

The notification method used by the I350 can be configured from the SMBus using the Receive Enable command. This default method is set by the EEPROM in the Pass-Through Init field.

The following events cause the I350 to send a notification event to the BMC:

- Receiving a LAN packet that is designated to the BMC
- The Firmware was reset and requires re-initialization if values other than the EEPROM defaults should be configured.
- Receiving a Request Status command from the BMC initiates a status response.

- The I350 is configured to notify the BMC upon status changes (by setting the EN_STA bit in the Receive Enable Command) and one of the following events happen:
  - — TCO Command Aborted
  - — Link Status changed
  - — Power state change
  - — Thermal Sensor Event

There can be cases where the BMC is hung and not responding to the SMBus notification. The I350 has a time-out value (defined in the EEPROM) to avoid hanging while waiting for the notification response. If the BMC does not respond until the time out expires, the notification is de-asserted and all pending data is silently discarded.

Note that the SMBus notification time-out value can only be set in the EEPROM. The BMC cannot modify this value.

10.5.6.1 SMBus Alert and Alert Response Method

The SMBus Alert# (SMBALERT_N) signal is an additional SMBus signal that acts as an asynchronous interrupt signal to an external SMBus master. The I350 asserts this signal each time it has a message that it needs the BMC to read and if the chosen notification method is the SMBus alert method. Note that the SMBus alert method is an open-drain signal which means that other devices besides the I350 can be connected on the same alert pin. As a result, the BMC needs a mechanism to distinguish between the alert sources.

The BMC can respond to the alert by issuing an ARA Cycle command to detect the alert source device. The I350 responds to the ARA cycle with its own SMBus slave address (if it was the SMBus alert source) and de-asserts the alert when the ARA cycle is completes. Following the ARA cycle, the BMC issues a read command to retrieve the I350 message.

Some BMCs do not implement the ARA cycle transaction. These BMCs respond to an alert by issuing a Read command to the I350 (0xC0/0xD0 or 0xDE). The I350 always responds to a Read command, even if it is not the source of the notification. The default response is a status transaction. If the I350 is the source of the SMBus Alert, it replies the read transaction and then de-asserts the alert after the command byte of the read transaction.

Note: In SMBus Alert mode, the SMBALERT_N pin is used for notification. In multiple-address mode, all devices generate alerts on events that are independent of each other.

The ARA cycle is an SMBus receive byte transaction to SMBus Address 0001-100b. Note that the ARA transaction does not support PEC. The ARA transaction format is as follows:

1	7	1	1	8	1	1	1
S	Alert Response Address	Rd	A	Slave Device Address		A	P
	0001 100	1	0	Manageability Slave SMBus Address	0	1	

10.5.6.2 Asynchronous Notify Method

When configured using the asynchronous notify method, the I350 acts as a SMBus master and notifies the BMC by issuing a modified form of the write word transaction. The asynchronous notify transaction SMBus address and data payload is configured using the Receive Enable command or using the EEPROM defaults. Note that the asynchronous notify is not protected by a PEC byte.

1	7	1	1	7	1	1	
S	Target Address	Wr	A	Sending Device Address		A	...
	BMC Slave Address	0	0	MNG Slave SMBus Address	0	0	

8	1	8	1	1
Data Byte Low	A	Data Byte High	A	P
Interface	0	Alert Value	0	

The target address and data byte low/high is taken from the Receive Enable command or EEPROM configuration.

10.5.6.3 Direct Receive Method

If configured, the I350 has the capability to send a message it needs to transfer to the external BMC as a master over the SMBus instead of alerting the BMC and waiting for it to read the message.

The message format follows. Note that the command that is used is the same command that is used by the external BMC in the Block Read command. The opcode that the I350 puts in the data is also the same as it put in the Block Read command of the same functionality. The rules for the *F* and *L* flags (bits) are also the same as in the Block Read command.

1	7	1	1	1	1	6	1	
S	Target Address	Wr	A	F	L	Command	A	...
	BMC Slave Address	0	0	First Flag	Last Flag	Receive TCO Command 01 0000b	0	

8	1	8	1		1	8	1	1
Byte Count	A	Data Byte 1	A	...	A	Data Byte N	A	P
N	0		0		0		0	

10.5.7 Receive TCO Flow

The I350 is used as a channel for receiving packets from the network link and passing them to the external BMC. The BMC configures the I350 to pass these specific packets to the BMC. Once a full packet is received from the link and identified as a manageability packet that should be transferred to the BMC, the I350 starts the receive TCO flow to the BMC.

The I350 uses the SMBus notification method to notify the BMC that it has data to deliver. Since the packet size might be larger than the maximum SMBus fragment size, the packet is divided into fragments, where the I350 uses the maximum fragment size allowed in each fragment (configured via the EEPROM). The last fragment of the packet transfer is always the status of the packet. As a result, the packet is transferred in at least two fragments. The data of the packet is transferred as part of the receive TCO LAN packet transaction.

When SMBus alert is selected as the BMC notification method, the I350 notifies the BMC on each fragment of a multi-fragment packet. When asynchronous notify is selected as the BMC notification method, the I350 notifies the BMC only on the first fragment of a received packet. It is the BMC's responsibility to read the full packet including all the fragments.

Any timeout on the SMBus notification results in discarding the entire packet. Any NACK by the BMC causes the fragment to be re-transmitted to the BMC on the next Receive Packet command.

The maximum size of the received packet is limited by the I350 hardware to 1536 bytes. Packets larger then 1536 bytes are silently discarded. Any packet smaller than 1536 bytes is processed.

10.5.8 Transmit TCO Flow

The I350 is used as the channel for transmitting packets from the external BMC to the network link. The network packet is transferred from the BMC over the SMBus and then, when fully received by the I350, is transmitted over the network link.

In quad-address mode, each SMBus address is connected to a different LAN port. When a packet is received during a SMBus transaction using SMBus address #0, it is transmitted to the network using LAN port #0; it is transmitted through LAN port #1 if received on SMBus address #1, etc.

The I350 supports packets up to an Ethernet packet length of 1536 bytes. Since SMBus transactions can only be up to 240 bytes in length, packets might need to be transferred over the SMBus in more than one fragment. This is achieved using the *F* and *L* bits in the command number of the transmit TCO packet Block Write command. When the *F* bit is set, it is the first fragment of the packet. When the *L* bit is set, it is the last fragment of the packet. When both bits are set, the entire packet is in one fragment. The packet is sent over the network link only after all its fragments are received correctly over the SMBus. The maximum SMBus fragment size is defined within the EEPROM and cannot be changed by the BMC.

The minimum packet length defined by the 802.3 spec is 64 bytes. The I350 pads packets that are less than 64 bytes to meet the specification requirements (there is no need for the external BMC to pad packets less than 64 bytes). If the packet sent by the BMC is larger than 1536 bytes, the I350 silently discards the packet.

The I350 calculates the L2 CRC on the transmitted packet and adds its four bytes at the end of the packet. Any other packet field (such as XSUM or VLAN) must be calculated and inserted by the BMC (the I350 does not change any field in the transmitted packet, other than adding padding and CRC bytes).

If the network link is down when the I350 has received the last fragment of the packet from the BMC, it silently discards the packet. Note that any link down event during the transfer of any packet over the SMBus does not stop the operation since the I350 waits for the last fragment to end to see whether the network link is up again.

10.5.8.1 Transmit Errors in Sequence Handling

Once a packet is transferred over the SMBus from the BMC to the I350, the *F* and *L* flags should follow specific rules. The *F* flag defines the first fragment of the packet; the *L* flag that the transaction contains the last fragment of the packet. Table 10-9 lists the different flag options in transmit packet transactions.

Table 10-9 Flag Options During Transmit Packet Transactions

Previous	Current	Action/Notes
Last	First	Accept both.
Last	Not First	Error for the current transaction. Current transaction is discarded and an abort status is asserted.
Not Last	First	Error in previous transaction. Previous transaction (until previous First) is discarded. Current packet is processed. No abort status is asserted.
Not Last	Not First	Process the current transaction.

Note: Since every other Block Write command in TCO protocol has both *F* and *L* flags on, they cause flushing any pending transmit fragments that were previously received. When running the TCO transmit flow, no other Block Write transactions are allowed in between the fragments.

10.5.8.2 TCO Command Aborted Flow

The I350 indicates to the BMC an error or an abort condition by setting the *TCO Abort* bit (See Section 10.5.10.2.2) in the general status. The I350 might also be configured to send a notification to the BMC (see Section 10.5.10.1.3.3).

Following is a list of possible error and abort conditions:

- Any error in the SMBus protocol (NACK, SMBus timeouts, etc.).
- Any error in compatibility between required protocols to specific functionality (for example, RX Enable command with a byte count not equal to 1/14, as defined in the command specification).
- If the I350 does not have space to store the transmitted packet from the BMC (in its internal buffer space) before sending it to the link, the packet is discarded and the external BMC is notified via the *Abort* bit.
- Error in the *F*/*L* bit sequence during multi-fragment transactions.
- An internal reset to the I350's firmware.

10.5.9 SMBus ARP Transactions

All SMBus ARP transactions include the PEC byte.

10.5.9.1 Prepare to ARP

This command clears the *Address Resolved* flag (set to false). It does not affect the status or validity of the dynamic SMBus address and is used to inform all devices that the ARP master is starting the ARP process:

1	7	1	1	8	1	8	1	1
S	Slave Address	Wr	A	Command	A	PEC	A	P
	1100 001	0	0	0000 0001	0	[Data Dependent Value]	0	

10.5.9.2 Reset Device (General)

This command clears the *Address Resolved* flag (set to false). It does not affect the status or validity of the dynamic SMBus address.

1	7	1	1	8	1	8	1	1
S	Slave Address	Wr	A	Command	A	PEC	A	P
	1100 001	0	0	0000 0010	0	[Data Dependent Value]	0	

10.5.9.3 Reset Device (Directed)

The Command field is NACKed if bits 7:1 do not match the current SMBus address. This command clears the *Address Resolved* flag (set to false) and does not affect the status or validity of the dynamic SMBus address.

1	7	1	1	8	1	8	1	1
S	Slave Address	Wr	A	Command	A	PEC	A	P
	1100 001	0	0	Targeted Slave Address | 0	0	[Data Dependent Value]	0	

10.5.9.4 Assign Address

This command assigns SMBus address. The address and command bytes are always acknowledged.

The transaction is aborted (NACKed) immediately if any of the UDID bytes is different from I350 UDID bytes. If successful, the manageability system internally updates the SMBus address. This command also sets the *Address Resolved* flag (set to true).

1	7	1	1	8	1	8	1	
S	Slave Address	Wr	A	Command	A	Byte Count	A	· · ·
	1100 001	0	0	0000 0100	0	0001 0001	0	

8	1	8	1	8	1	8	1	
Data 1	A	Data 2	A	Data 3	A	Data 4	A	· · ·
UDID Byte 15 (MSB)	0	UDID Byte 14	0	UDID Byte 13	0	UDID Byte 12	0	

8	1	8	1	8	1	8	1	
Data 5	A	Data 6	A	Data 7	A	Data 8	A	· · ·
UDID Byte 11	0	UDID Byte 10	0	UDID Byte 9	0	UDID Byte 8	0	

8	1	8	1	8	1	
Data 9	A	Data 10	A	Data 11	A	· · ·
UDID Byte 7	0	UDID Byte 6	0	UDID Byte 5	0	

8	1	8	1	8	1	8	1	
Data 12	A	Data 13	A	Data 14	A	Data 15	A	· · ·
UDID Byte 4	0	UDID Byte 3	0	UDID Byte 2	0	UDID Byte 1	0	

8	1	8	1	8	1	1
Data 16	A	Data 17	A	PEC	A	P
UDID Byte 0 (LSB)	0	Assigned Address	0	[Data Dependent Value]	0	

Note: The Assigned address is not checked by the I350, so if the bus master assigns an invalid address (for example a reserved address), the I350 will use it as its address. The only exception is address 0xC2 used as part of the SMBus process.

10.5.9.5 Get UDID (General and Directed)

The general get UDID SMBus transaction supports a constant command value of 0x03 and, if directed, supports a Dynamic command value equal to the dynamic SMBus address.

If the SMBus address has been resolved (*Address Resolved* flag set to true), the manageability system does not acknowledge (NACK) this transaction. If it's a General command, the manageability system always acknowledges (ACKs) as a directed transaction.

This command does not affect the status or validity of the dynamic SMBus address or the *Address Resolved* flag.

S	Slave Address	Wr	A	Command	A	S	• • •
	1100 001	0	0	See Below	0		

7	1	1	8	1	
Slave Address	Rd	A	Byte Count	A	· · ·
1100 001	1	0	0001 0001	0	

8	1	8	1	8	1	8	1	
Data 1	A	Data 2	A	Data 3	A	Data 4	A	· · ·
UDID Byte 15 (MSB)	0	UDID Byte 14	0	UDID Byte 13	0	UDID Byte 12	0	

8	1	8	1	8	1	8	1	
Data 5	A	Data 6	A	Data 7	A	Data 8	A	· · ·
UDID Byte 11	0	UDID Byte 10	0	UDID Byte 9	0	UDID Byte 8	0	

8	1	8	1	8	1	
Data 9	A	Data 10	A	Data 11	A	· · ·
UDID Byte 7	0	UDID Byte 6	0	UDID Byte 5	0	

8	1	8	1	8	1	8	1	
Data 12	A	Data 13	A	Data 14	A	Data 15	A	· · ·
UDID Byte 4	0	UDID Byte 3	0	UDID Byte 2	0	UDID Byte 1	0	

8	1	8	1	8	1	1
Data 16	A	Data 17	A	PEC	~Ã	P
UDID Byte 0 (LSB)	0	Device Slave Address	0	[Data Dependent Value]	1	

The Get UDID command depends on whether or not this is a Directed or General command.

The General Get UDID SMBus transaction supports a constant command value of 0x03.

The Directed Get UDID SMBus transaction supports a Dynamic command value equal to the dynamic SMBus address with the LSB bit set.

Note: Bit 0 (LSB) of Data byte 17 is always 1b.

10.5.10 SMBus Pass-Through Transactions

This section details commands (both read and write) that the I350 SMBus interface supports for pass-through.

10.5.10.1 Write SMBus Transactions

This section details the commands that the BMC can send to the I350 over the SMBus interface. The SMBus write transactions table lists the different SMBus write transactions supported by the I350.

TCO Command	Transaction	Command	Fragmentation	Section
Transmit Packet	Block Write	First: 0x84 Middle: 0x04 Last: 0x44	Multiple	10.5.10.1.1
Transmit Packet	Block Write	Single: 0xC4	Single	10.5.10.1.1
Request Status	Block Write	Single: 0xDD	Single	10.5.10.1.2
Receive Enable	Block Write	Single: 0xCA	Single	10.5.10.1.3
Force TCO	Block Write	Single: 0xCF	Single	10.5.10.1.4
Management Control	Block Write	Single: 0xC1	Single	10.5.10.1.5
Update MNG RCV Filter Parameters	Block Write	Single: 0xCC	Single	10.5.10.1.6
Set Thermal Sensor Configuration	Block Write	Single: 0xCB (opcode = 1)	Single	10.5.10.1.7
Perform Thermal Sensor Action	Block Write	Single: 0xCB (opcode = 2)	Single	10.5.10.1.8

10.5.10.1.1 Transmit Packet Command

The Transmit Packet command behavior is detailed in Section 10.5.8. The Transmit Packet fragments have the following format.

The payload length is limited to the maximum payload length set in the EEPROM. If the overall packet length is bigger than 1536 bytes, the packet is silently discarded.

Function	Command	Byte Count	Data 1	...	Data N
Transmit first fragment	0x84	N	Packet data MSB	...	Packet data LSB
Transmit middle fragment	0x04				
Transmit last fragment	0x44				
Transmit single fragment	0xC4				

10.5.10.1.2 Request Status Command

An external BMC can initiate a request to read I350 manageability status by sending a Request Status command. When received, the I350 initiates a notification to an external BMC when status is ready. After this, the external controller will be able to read the status, by issuing a read status command (see Section 10.5.10.2.2).

The format is as follows:

Function	Command	Byte Count	Data 1
Request Status	0xDD	1	0

10.5.10.1.3 Receive Enable Command

The Receive Enable command is a single fragment command used to configure the I350. This command has two formats: short, 1-byte legacy format (providing backward compatibility with previous components) and long, 14-byte advanced format (allowing greater configuration capabilities). The Receive Enable command format is as follows:

Function	CMD	Byte Count	Data 1	Data 2	...	Data 7	Data 8	...	Data 11	Data 12	Data 13	Data 14
Legacy Receive Enable	0xCA	1	Receive Control Byte	-	...	-	-	...	-	-	-	-
Advanced Receive Enable		14 (0x0E)		MAC Addr MSB		MAC Addr LSB	IP Addr MSB		IP Addr LSB	BMC SMBus Addr	I/F Data Byte	Alert Value Byte

Table 10-10 Receive Control Byte

Field	Bit(s)	Description
RCV_EN	0	Receive TCO Enable. 0b: Disable receive TCO packets. 1b: Enable Receive TCO packets. Setting this bit enables all manageability receive filtering operations. Enabling specific filters is done via the EEPROM or through special configuration commands. **Note:** When the *RCV_EN* bit is cleared, all receive TCO functionality is disabled, not just the packets that are directed to the BMC (also auto ARP packets).
RCV_ALL	1	Receive All Enable. 0b: Disable receiving all packets. 1b: Enable receiving all packets. Forwards all packets received over the wire that passed L2 filtering to the external BMC. This flag has no effect if bit 0 (Enable TCO packets) is disabled.
EN_STA	2	Enable Status Reporting. 0b: Disable status reporting. 1b: Enable status reporting.

Table 10-10 Receive Control Byte

Field	Bit(s)	Description
EN_ARP_RES	3	Enable ARP Response. 0b: Disable the I350 ARP response. The I350 treats ARP packets as any other packet, for example, packet is forwarded to the BMC if it passed other (non-ARP) filtering. 1b: Enable the I350 ARP response. The I350 automatically responds to all received ARP requests that match its IP address. Note that setting this bit does not change the Rx filtering settings. Appropriate Rx filtering to enable ARP request packets to reach the BMC should be set by the BMC or by the EEPROM. The BMC IP address is provided as part of the Receive Enable message (bytes 8:11). If a short version of the command is used, the I350 uses IP address configured in the most recent long version of the command in which the EN_ARP_RES bit was set. If no such previous long command exists, then the I350 uses the IP address configured in the EEPROM as ARP Response IPv4 Address in the pass-through LAN configuration structure. If the *CBDM* bit is set, the I350 uses the BMC dedicated MAC address in ARP response packets. If the CBDM bit is not set, the BMC uses the Host MAC address.
NM	5:4	Notification Method. Define the notification method the I350 uses. 00b: SMBUS Alert. 01b: Asynchronous notify. 10b: Direct receive. 11b: Not supported.
Reserved	6	Reserved. Must be set to 1b.
CBDM	7	Configure the BMC Dedicated MAC Address. **Note:** This bit should be 0b when the *RCV_EN* bit (bit 0) is not set. 0b: The I350 shares the MAC address for MNG traffic with the Host MAC address, which is specified in EEPROM words 0x0-0x2. 1b: The I350 uses the BMC dedicated MAC address as a filter for incoming receive packets. The BMC MAC address is set in bytes 2-7 in this command. If a short version of the command is used, the I350 uses the MAC address configured in the most recent long version of the command in which the *CBDM* bit was set. When the dedicated MAC address feature is activated, the I350 uses the following registers to filter in all the traffic addressed to the BMC MAC. BMC should not modify these registers: Manageability Decision Filter – MDEF7 (and corresponding bit 7 in Management Only traffic Register – *MNGONLY*) Manageability MAC Address Low – *MMAL[3]* Manageability MAC Address High – *MMAH[3]*

10.5.10.1.3.1 Management MAC Address (Data Bytes 7:2)

Ignored if the *CBDM* bit is not set. This MAC address is used to configure the dedicated MAC address. In addition, it is used in the ARP response packet when the *EN_ARP_RES* bit is set. This MAC address is also used when *CBDM* bit is set in subsequent short versions of this command.

10.5.10.1.3.2 Management IP Address (Data Bytes 11:8)

This IP address is used to filter ARP request packets.

10.5.10.1.3.3 Asynchronous Notification SMBus Address (Data Byte 12)

This address is used for the asynchronous notification SMBus transaction and for direct receive.

10.5.10.1.3.4 Interface Data (Data Byte 13)

Interface data byte used in asynchronous notification.

10.5.10.1.3.5 Alert Value Data (Data Byte 14)

Alert Value data byte used in asynchronous notification.

10.5.10.1.4 Force TCO Command

This command causes the I350 to perform a TCO reset, TCO isolate, or Firmware Reset

TCO Reset: if Force TCO reset is enabled in the EEPROM. The force TCO reset clears the data path (Rx/Tx) of the I350 to enable the BMC to transmit/receive packets through the I350. Force TCO reset is asserted only to the port related to the SMBus address the command. This command should only be used when the BMC is unable to transmit receive and suspects that the I350 is inoperable. The command also causes the LAN device driver to unload. It is recommended to perform a system restart to resume normal operation.

TCO isolate: if TCO isolate is enabled in the EEPROM (See Section 6.3.7.3). The TCO Isolate command will disable PCIe write operations to the LAN port. If TCO Isolate is disabled in EEPROM the I350 does not execute the command but sends a response to the BMC with successful completion. Following TCO Isolate management sets *MANC.TCO_Isolate* to 1.

Firmware Reset: This command will cause re-initialization of all the manageability functions and re-load of manageability related EEPROM words (e.g. Firmware patch code).

The I350 considers the Force TCO reset command as an indication that the operating system is hung and clears the *DRV_LOAD* flag. The Force TCO command format is as follows:

Function	Command	Byte Count	Data 1
Force TCO Reset	0xCF	1	TCO Mode

Where TCO Mode is:

Field	Bit(s)	Description
DO_TCO_RST	0	Perform TCO Reset. 0b: Do nothing. 1b: Perform TCO reset.
DO_TCO_ISOLATE[1]	1	Do TCO Isolate 0b = Enable PCIe write access to LAN port. 1b = Isolate Host PCIe write operation to the port Note: Should be used for debug only.
RESET_MGMT	2	Reset manageability; re-load manageability EEPROM words. 0b = Do nothing 1b = Issue firmware reset to manageability. Setting this bit generates a one-time firmware reset. Following the reset, management related data from EEPROM is loaded.
Reserved	7:3	Reserved (set to 0x00).

1. TCO Isolate Host Write operation enabled in EEPROM.

10.5.10.1.5 Management Control

This command is used to set generic manageability parameters. The parameters list is shown in Table 10-11. The command is 0xC1 stating that it is a Management Control command. The first data byte is the parameter number and the data afterwards (length and content) are parameter specific as shown in Management Control Command Parameters/Content.

Note: If the parameter that the BMC sets is not supported by the I350. The I350 does not NACK the transaction. After the transaction ends, the I350 discards the data and asserts a transaction abort status.

The Management Control command format is as follows:

Function	Command	Byte Count	Data 1	Data 2	...	Data N
Management Control	0xC1	N	Parameter Number	Parameter Dependent		

Table 10-11 Management Control Command Parameters/Content

Parameter	#	Parameter Data
Keep PHY Link Up	0x00	A single byte parameter: Data 2: Bit 0: Set to indicate that the PHY link for this port should be kept up throughout system resets. This is useful when the server is reset and the BMC needs to keep connectivity for a manageability session. Bit [7:1] Reserved. 0b: Disabled. 1b: Enabled.

10.5.10.1.6 Update Management Receive Filter Parameters

This command is used to set the manageability receive filters parameters. The command is 0xCC. The first data byte is the parameter number and the data that follows (length and content) are parameter specific as listed in management RCV filter parameters.

If the parameter that the BMC sets is not supported by the I350, then the I350 does not NACK the transaction. After the transaction ends, the I350 discards the data and asserts a transaction abort status.

The update management RCV receive filter parameters command format is as follows:

Function	Command	Byte Count	Data 1	Data 2	...	Data N
Update Manageability Filter Parameters	0xCC	N	Parameter Number	Parameter Dependent		

Table 10-12 lists the different parameters and their content.

Table 10-12 Management Receive Filter Parameters

Parameter	Number	Parameter Data
Filters Enables	0x1	Defines the generic filters configuration. The structure of this parameter is four bytes as the Manageability Control (MANC) register. **Note:** The general filter enable is in the Receive Enable command that enables receive filtering.

Table 10-12 Management Receive Filter Parameters

Parameter	Number	Parameter Data
MNGONLY configuration	0xF	This parameter defines which of the packets types identified as manageability packets in the receive path will never be directed to the Host memory. Data 2:5: MNGONLY register bytes - Data 2 is the MSB
Flex Filter 0 Enable Mask and Length	0x10	Flex Filter 0 Mask. Data 17:2 = Mask. Bit 0 in data 2 is the first bit of the mask. Data 19:18 = Reserved. Should be set to 00b. Date 20 = Flexible filter length.
Flex Filter 0 Data	0x11	Data 2 — Group of flex filter's bytes: 0x0 = bytes 0-29 0x1 = bytes 30-59 0x2 = bytes 60-89 0x3 = bytes 90-119 0x4 = bytes 120-127 Data 3:32 = Flex filter data bytes. Data 3 is LSB. Group's length is not a mandatory 30 bytes; it might vary according to filter's length and must NOT be padded by zeros.
Decision Filters	0x61	Five bytes are required to load the manageability decision filters (MDEF). Data 2: Decision filter number. Data 3: MSB of MDEF register for this decision filter. ... Data 6: LSB of MDEF register for this decision filter.
VLAN Filters	0x62	Three bytes are required to load the VLAN tag filters. Data 2: VLAN filter number. Data 3: MSB of VLAN filter. Data 4: LSB of VLAN filter.
Flex Port Filters	0x63	Three bytes are required to load the manageability flex port filters. Data 2: Flex port filter number. Data 3: MSB of flex port filter. Data 4: LSB of flex port filter.
IPv4 Filters	0x64	Five bytes are required to load the IPv4 address filter. Data 2: IPv4 address filter number (3:0). Data 3: LSB of IPv4 address filter. … Data 6: MSB of IPv4 address filter.
IPv6 Filters	0x65	17 bytes are required to load the IPv6 address filter. Data 2 — IPv6 address filter number (3:0). Data 3 — LSB of IPv6 address filter. … Data 18 — MSB of IPv6 address filter.
MAC Filters	0x66	Seven bytes are required to load the MAC address filters. Data 2 — MAC address filters pair number (3:0). Data 3 — MSB of MAC address. … Data 8: LSB of MAC address.
EtherType Filters	0x67	5 bytes to load Ethertype Filters (METF) Data 2 — METF filter index (valid values are 0, 1, 2, 3) Data 3 — MSB of METF ... Data 6 — LSB of METF

Table 10-12 Management Receive Filter Parameters

Parameter	Number	Parameter Data
Extended Decision Filter	0x68	9 bytes to load the extended decision filters (MDEF_EXT & MDEF) Data 2 — MDEF filter index (valid values are 0...6) Data 3 — MSB of MDEF_EXT (DecisionFilter1) .... Data 6 — LSB of MDEF_EXT (DecisionFilter1) Data 7 — MSB of MDEF (DecisionFilter0) .... Data 10 — LSB of MDEF (DecisionFilter0) The command shall overwrite any previously stored value

Table 10-13 Filter Enable Parameters

Bit	Name	Description
16:0	Reserved	Reserved
17	RCV_TCO_EN	TCO Receive Traffic Enabled. When bit is set receive traffic to the manageability block is enabled. This bit should be set only if at least one of EN_BMC2OS or EN_BMC2NET bits are set. This bit is usually set using the receive enable command (see Section 10.5.10.1.3).
18	KEEP_PHY_LINK_UP	Block PHY reset and power state changes. When this bit is set the PHY reset and power state changes does not get to the PHY, This bit can not be written unless Keep_PHY_Link_Up_En EEPROM bit is set.
22:19	Reserved	Reserved
23	Enable Xsum Filtering to MNG	When this bit is set, only packets that pass the L3 and L4 checksum are send to the manageability block.
24	Enable IPv4 Address Filters	When set, the last 128 bits of the MIPAF register are used to store four IPv4 addresses for IPv4 filtering. When cleared, these bits store a single IPv6 filter.
25	FIXED_NET_TYPE	Fixed net type: If set, only packets matching the net type defined by the NET_TYPE field passes to manageability. Otherwise, both tagged and un-tagged packets can be forwarded to the manageability engine.
26	NET_TYPE	NET TYPE: 0b = pass only un-tagged packets. 1b = pass only VLAN tagged packets. Valid only if FIXED_NET_TYPE is set.
31:27	Reserved	Reserved.

10.5.10.1.7 Set Thermal Sensor Configuration

This command sets the thermal sensor configuration for threshold “Index” in direction “Direction”.

Where the threshold is measured in “unit types” and the “Actions” field describes the actions to activate upon crossing of the threshold in the requested direction according to Table 10-44.

Direction is encoded as follow:

- 0 = High Going
- 1 = Low Going

Function	Command	Byte Count	Data 1	Data 2	Data
Set Thermal Sensor Configuration	0xCB	14	Opcode (0x1)	Index	Direction

Data 4-5	Data 6-9	Data 10-13	Data 14
Threshold	Actions "Going High"	Actions "Going Low"	Hysteresis

10.5.10.1.8 Perform Thermal Sensor Action

This command executes actions immediately.

The "Actions" field describes the actions to activate according to Table 10-44.

Function	Command	Byte Count	Data 1	Data 2-5
Perform Thermal Sensor Action	0xCB	5	Opcode (0x2)	Actions

10.5.10.2 Read SMBus Transactions

This section details the pass-through read transactions that the BMC can send to the I350 over SMBus.

SMBus read transactions lists the different SMBus read transactions supported by the I350. All the read transactions are compatible with SMBus read block protocol format.

Table 10-14 SMBus Read Transactions

TCO Command	Transaction	Command	Opcode	Fragments	Section
Receive TCO Packet	Block Read	0xD0 or 0xC0	First: 0x90 Middle: 0x10 Last[1]: 0x50	Multiple	10.5.10.2.1
Read Status	Block Read	0xD0 or 0xC0 or 0xDE	Single: 0xDD	Single	10.5.10.2.2
Get System MAC Address	Block Read	0xD4	Single: 0xD4	Single	10.5.10.2.3
Read Management Parameters	Block Read	0xD1	Single: 0xD1	Single	10.5.10.2.4
Read Management RCV Filter Parameters	Block Read	0xCD	Single: 0xCD	Single	10.5.10.2.5
Read Receive Enable Configuration	Block Read	0xDA	Single: 0xDA	Single	10.5.10.2.6
Get Thermal Sensor Capabilities	Block Read	0xDB (index = 0)	Single: 0xDB	Single	10.5.10.2.6
Get Thermal Sensor Configuration	Block Read	0xDB (index = 1)	Single: 0xDB	Single	10.5.10.2.6
Get Thermal Sensor Status	Block Read	0xDB (index = 2)	Single: 0xDB	Single	10.5.10.2.6

1. The last fragment of the receive TCO packet is the packet status.

0xC0 or 0xD0 commands are used for more than one payload. If BMC issues these read commands, and the I350 has no pending data to transfer, it always returns as default opcode 0xDD with the I350 status and does not NACK the transaction.

If an SMBus quick read command is received, it is handled as a I350 Request Status command (See Section 10.5.10.1.2 for details).

10.5.10.2.1 Receive TCO LAN Packet Transaction

The BMC uses this command to read packets received on the LAN and its status. When the I350 has a packet to deliver to the BMC, it asserts the SMBus notification for the BMC to read the data (or direct receive). Upon receiving notification of the arrival of a LAN receive packet, the BMC begins issuing a Receive TCO packet command using the block read protocol.

A packet can be transmitted to the BMC in at least two fragments (at least one for the packet data and one for the packet status). As a result, BMC should follow the *F* and *L* bit of the op-code.

The op-code can have these values:

- 0x90 — First Fragment
- 0x10 — Middle Fragment
- When the opcode is 0x50, this indicates the last fragment of the packet, which contains packet status.

If a notification timeout is defined (in the EEPROM) and the BMC does not finish reading the whole packet within the timeout period, since the packet has arrived, the packet is silently discarded.

Following is the receive TCO packet format and the data format returned from the I350.

Function	Command
Receive TCO Packet	0xC0 or 0xD0

Function	Byte Count	Data 1 (Op-Code)	Data 2	...	Data N
Receive TCO First Fragment	N	0x90	Packet Data Byte	...	Packet Data Byte
Receive TCO Middle Fragment		0x10			
Receive TCO Last Fragment	17 (0x11)	0x50	See Section 10.5.10.2.1.1		

10.5.10.2.1.1 Receive TCO LAN Status Payload Transaction

This transaction is the last transaction that the I350 issues when a packet received from the LAN is transferred to the BMC. The transaction contains the status of the received packet.

The format of the status transaction is as follows:

Function	Byte Count	Data 1 (Op-Code)	Data 2 – Data 17 (Status Data)
Receive TCO Long Status	17 (0x11)	0x50	See Below

The status is 16 bytes where byte 0 (bits 7:0) is set in Data 2 of the status and byte 15 in Data 17 of the status. Table 10-15 lists the content of the status data.

Table 10-15 TCO LAN Packet Status Data

Name	Bits	Description
Packet Length	13:0	Packet length including CRC, only 14 LSB bits.
Packet status	36:14	See Table 10-16
Reserved	42:37	Reserved
Error	47:43	See Table 10-17
VLAN	63:48	The two bytes of the VLAN header tag.
Reserved	67:64	Reserved
Packet type	80:68	See Table 10-19
Reserved	84:81	Reserved
MNG status	127:85	See Table 10-20. This field should be ignored if Receive TCO is not enabled,

The meaning of the bits inside of each field can be found in Section 7.1.4.2.

Table 10-16 Packet Status Info

Field	Bit(s)	Description
LAN#	22:21	Indicates the source port of the packet
Reserved	20	Reserved
VP	19	VLAN Stripped –insertion of VLAN TAG is needed.
VEXT	18	Additional VLAN present in packet
Reserved	17:15	Reserved
Reserved	14:12	Reserved
CRC stripped	11	Insertion of CRC is needed.
Reserved	10:6	Reserved
UDPV	5	UDP checksum valid
Reserved	4:3	Reserved
IPCS	2	Ipv4 Checksum Calculated on packet
L4I	1	L4 (TCP/UDP) Checksum calculated on packet
UDPCS	0	UDP checksum calculated on packet

Table 10-17 Error Status Info

Field	Bit(s)	Description
RXE	4	RX Data Error
IPE	3	Ipv4 Checksum Error
L4E	2	L4 (TCP/UDP) Checksum Error
Reserved	1:0	Reserved

Table 10-19 Packet type

Bit Index	Bit 11 = 0b	Bit 11 = 1b (L2 packet)
12	VLAN packet indication	
11	Packet matched one of the ETQF filters.	
10:8	Reserved	Reserved
7	Reserved	Reserved
6		
5		
4		
3		
2		EtherType - ETQF register index that matches the packet. Special types might be defined for 1588, 802.1X, or any other requested type.
1		
0		

Table 10-20 MNG Status

Name	Bits	Description
Decision Filter match	42:35	Set when there is a match to one of the Decision filters
IPv4/IPv6 match	34	Set when there is an IPv6 match and cleared when there's an IPV4 match. This bit is valid only if bit 33 (IP match bit) is set.
IP address match	33	Set when there is a match to any of the IP address filters
IP address Index	32:31	Set when there is a match to the IP filter number. (IPv4 or IPv6)
Flex TCO filter match	30	Set when there is a match to the Flex port filter
Reserved	29:27	Reserved
L4 port match	26	Set when there is a match to any of the UDP / TCP port filters
L4 port Filter Index	25:19	Indicate the flex filter number
Exact Address match	18	Set when there is a match to any of the 4 Exact MAC addresses.
Exact Address Index	17:15	Indicates which of the 4 Exact MAC addresses match the packet. Valid only if the Exact Address match is set.
MNG VLAN Address Match	14	Set when the MNG packet matches one of the MNG VLAN filters
Pass MNG VLAN Filter Index	13:11	Indicates which of the Vlan filters match the packet.
Reserved	10:8	Reserved
Pass ARP req / ARP resp	7	Set when the MNG packet is an ARP response/request packet
Pass MNG neighbor	6	Set when the MNG packet is a neighbor discovery packet.
Pass MNG broadcast	5	Set when the MNG packet is a broadcast packet
Pass RMCP 0x0298	4	Set when the UDP/TCP port of the MNG packet is 0x298
Pass RMCP 0x026F	3	Set when the UDP/TCP port of the MNG packet is 0x26F
Manageability Ethertype filter passed	2	Indicates that one of the METF filters matched
manageability Ethertype filter index	1:0	Indicates which of the METF filters matched

10.5.10.2.2 Read Status Command

The BMC should use this command after receiving a notification from the I350 (such as SMBus Alert). The I350 also sends a notification to the BMC in either of the following two cases:

- The BMC asserts a request for reading the status.

- The I350 detects a change in one of the Status Data 1 bits (and was set to send status to the BMC on status change) in the Receive Enable command.

Note: Commands 0xC0/0xD0 are for backward compatibility and can be used for other payloads. The I350 defines these commands in the opcode as well as which payload this transaction is. When the 0XDE command is set, the I350 always returns opcode 0XDD with the I350 status. The BMC reads the event causing the notification, using the Read Status command as follows.

The I350 response to one of the commands (0xC0 or 0xD0) in a given time as defined in the SMBus Notification Timeout and Flags word in the EEPROM.

Function	Command
Read Status	0XC0 or 0XD0 or 0XDE

Function	Byte Count	Data 1 (Op-Code)	Data 2 (Status Data 1)	Data 3 (Status Data 2)
Receive TCO Partial Status	3	0XDD	See Below	

Table 10-21 lists the status data byte 1 parameters.

Table 10-21 Status Data Byte 1

Bit	Name	Description
7	LAN Port Lsb	LAN port Lsb together with Lan Port Msb define port that sent status. See further information in description of Lan Port Msb (bit 2).
6	TCO Command Aborted	1b = A TCO command abort event occurred since the last read status cycle. 0b = A TCO command abort event did not occur since the last read status cycle.
5	Link Status Indication	0b = LAN link down. 1b = LAN link up.
4	PHY Link Forced Up	Contains the value of the *PHY_Link_Up* bit. When set, indicates that the PHY link is configured to keep the link up.
3	Initialization Indication	0b = An EEPROM reload event has not occurred since the last Read Status cycle. 1b = An EEPROM reload event has occurred since the last Read Status cycle[1].
2	LAN Port Msb	Defines together with LAN Port Lsb the port that sent the Status: Lan Port Msb / Lan Port Lsb 0 / 0 — Status came from LAN port 0. 0 / 1 — Status came from LAN port 1. 1 / 0 — Status came from LAN port 2. 1 / 1 — Status came from LAN port 3.
1:0	Power State	00b = Dr state. 01b = D0u state. 10b = D0 state. 11b = D3 state.

1. This indication is asserted when the I350 manageability block reloads the EEPROM and its internal database is updated to the EEPROM default values. This is an indication that the external BMC should reconfigure the I350, if other values other than the EEPROM default should be configured.

Status data byte 2 is used by the BMC to indicate whether the LAN device driver is alive and running.

The LAN device driver valid indication is a bit set by the LAN device driver during initialization; the bit is cleared when the LAN device driver enters a Dx state or is cleared by the hardware on a PCI reset.

Bits 2 and 1 indicate that the LAN device driver is stuck. Bit 2 indicates whether the interrupt line of the LAN function is asserted. Bit 1 indicates whether the LAN device driver dealt with the interrupt line before the last Read Status cycle. Table 10-22 lists status data byte 2.

Table 10-22 Status Data Byte 2

Bit	Name	Description
7:5	Reserved	Reserved.
4	Reserved	Reserved
3	Driver Valid Indication	0b = LAN driver is not alive. 1b = LAN driver is alive.
2	Interrupt Pending Indication	1b = LAN interrupt line is asserted. 0b = LAN interrupt line is not asserted.
1	Interrupt Cause Register (ICR0 Read/Write	1b = ICR register was read since the last read status cycle. 0b = ICR register was not read since the last read status cycle. Reading the ICR indicates that the driver has dealt with the interrupt that was asserted.
0	Thermal Sensor event	0b = No thermal event 1b = Thermal event asserted.

Table 10-23 lists the possible values of bits 2 and 1 and what the BMC can assume from the bits:

Table 10-23 Status Data Byte 2 (Bits 2 and 1)

Previous	Current	Description
Don't Care	00b	Interrupt is not pending (OK).
00b	01b	New interrupt is asserted (OK).
10b	01b	New interrupt is asserted (OK).
11b	01b	Interrupt is waiting for reading (OK).
01b	01b	Interrupt is waiting for reading by the driver for more than one read cycle (not OK). Possible drive hang state.
Don't Care	11b	Previous interrupt was read and current interrupt is pending (OK).
Don't Care	10b	Interrupt is not pending (OK).

BMC reads should consider the time it takes for the LAN device driver to deal with the interrupt (in µs). Note that excessive reads by the BMC can give false indications.

10.5.10.2.3 Get System MAC Address Command

The Get System MAC Address returns the system MAC address over to the SMBus. This command is a single-fragment Read Block transaction that returns the following the MAC address configured in RAL0, RAH0 registers.

Get system MAC address format:

Function	Command
Get system MAC address	0xD4

Data returned from the I350:

Function	Byte Count	Data 1 (Op-Code)	Data 2	...	Data 7
Get system MAC address	7	0xD4	MAC address MSB	...	MAC address LSB

10.5.10.2.4 Read Management Parameters Command

In order to read the management parameters the BMC should execute two SMBus transactions. The first transaction is a block write that sets the parameter that the BMC wants to read. The second transaction is block read that reads the parameter.

Block write transaction:

Function	Command	Byte Count	Data 1
Management control request	0xC1	1	Parameter number

Following the block write the BMC should issue a block read that reads the parameter that was set in the Block Write command:

Function	Command
Read management parameter	0xD1

Data returned:

Function	Byte Count	Data 1 (Op-Code)	Data 2	Data 3	...	Data N
Read management parameter	N	0xD1	Parameter number	Parameter dependent		

The returned data is in the same format of the BMC command. The returned data is as follow:

Parameter	#	Parameter Data
Keep PHY Link Up	0x00	A single byte parameter: Data 2 — Bit 0 Set to indicate that the PHY link for this port should be kept up. Sets the keep_PHY_link_up bit. When cleared, clears the keep_PHY_link_up bit. Bit [7:1] Reserved.
Wrong parameter request	0xFE	Returned by the I350 only. This parameter is returned on read transaction, if in the previous read command the BMC sets a parameter that is not supported by the I350.
The I350 is not ready	0xFF	Returned by the I350 only, on read parameters command when the data that should have been read is not ready. This parameter has no data. The BMC should retry the read transaction. This value is also returned if the byte count is illegal or if the read command is not preceded by a write command.

The parameter that is returned might not be the parameter requested by the BMC. The BMC should verify the parameter number (default parameter to be returned is 0x1).

If the parameter number is 0xFF, it means that the data that was requested from the I350 is not ready yet.The BMC should retry the read transaction.

It is responsibility of the BMC to follow the procedure previously defined. When the BMC sends a Block Read command (as previously described) that is not preceded by a Block Write command with bytecount=1, the I350 sets the parameter number in the read block transaction to be 0xFF.

10.5.10.2.5 Read Management Receive Filter Parameters Command

In order to read the management receive filter parameters, the BMC should execute two SMBus transactions. The first transaction is a block write that sets the parameter that the BMC wants to read. The second transaction is block read that read the parameter.

Block write transaction:

Function	Command	Byte Count	Data 1	Data 2
Update MNG RCV filter parameters	0xCC	1 or 2	Parameter number	Parameter data

The different parameters supported for this command are the same as the parameters supported for update Management receive filter parameters.

Following the block write the BMC should issue a block read that reads the parameter that was set in the Block Write command:

Function	Command
Request MNG RCV filter parameters	0xCD

Data returned from the I350:

Function	Byte Count	Data 1 (Op-Code)	Data 2	Data 3	...	Data N
Read MNG RCV filter parameters	N	0xCD	Parameter number	Parameter dependent		

The parameter that is returned might not be the parameter requested by the BMC. The BMC should verify the parameter number (default parameter to be returned is 0x1).

If the parameter number is 0xFF, it means that the data that was requested from the I350 should supply is not ready yet. The BMC should retry the read transaction.

It is BMC responsibility to follow the procedure previously defined. When the BMC sends a Block Read command (as previously described) that is not preceded by a Block Write command with bytecount=1 or 2, the I350 sets the parameter number in the read block transaction to be 0xFF.

Parameter	#	Parameter Data
Filters Enable	0x01	None
MNGONLY Configuration	0x0F	None
Flex Filter Enable Mask and Length	0x10	None
Flex Filter Data	0x11	Data 2 — Group of Flex Filter's Bytes: 0x0 = bytes 0-29 0x1 = bytes 30-59 0x2 = bytes 60-89 0x3 = bytes 90-119 0x4 = bytes 120-127
Decision Filters	0x61	One byte to define the accessed manageability decision filter (MDEF) Data 2 — Decision Filter number
VLAN Filters	0x62	One byte to define the accessed VLAN tag filter (MAVTV) Data 2 — VLAN Filter number
Flex Ports Filters	0x63	One byte to define the accessed manageability flex port filter (MFUTP). Data 2 — Flex Port Filter number
IPv4 Filter	0x64	One byte to define the accessed IPv4 address filter (MIPAF) Data 2 — IPv4 address filter number
IPv6 Filters	0x65	One byte to define the accessed IPv6 address filter (MIPAF) Data 2 — Pv6 address filter number
MAC Filters	0x66	One byte to define the accessed MAC address filters pair (MMAL, MMAH) Data 2 — MAC address filters pair number (0 - 3)
EtherType Filters	0x67	1 byte to define Ethertype filters (METF) Data 2 — METF filter index (valid values are 0 - 3)
Extended Decision Filter	0x68	1 byte to define the extended decisions filters (MDEF_EXT & MDEF) Data 2 — MDEF filter index (valid values are 0 - 6)
Wrong parameter request	0xFE	Returned by the I350 only. This parameter is returned on read transaction, if in the previous read command the BMC sets a parameter that is not supported by the I350.
The I350 is not ready	0xFF	Returned by the I350 only, on read parameters command when the data that should have been read is not ready. This parameter has no data. This value is also returned if the byte count is illegal or if the read command is not preceded by a write command.

10.5.10.2.6 Read Receive Enable Configuration Command

The BMC uses this command to read the receive configuration data. This data can be configured when using Receive Enable command or through the EEPROM.

Read Receive Enable Configuration command format (SMBus Read Block) is as follows:

Function	Command
Read Receive Enable	0xDA

Data returned from the I350:

Function	Byte Count	Data 1 (Op-Code)	Data 2	Data 3	...	Data 8	Data 9	...	Data 12	Data 13	Data 14	Data 15

Read Receive Enable	15 (0x0F)	0xDA	Receive Control Byte	MAC Addr MSB	...	MAC Addr LSB	IP Addr MSB	...	IP Addr LSB	BMC SMBus Addr	I/F Data Byte	Alert Value Byte

The detailed description of each field is specified in the receive enable command description in Section 10.5.10.1.3.

10.5.10.2.7 Get Thermal Sensor Capabilities Command

The BMC can use this function to read the thermal sensor capabilities. It uses a write command to set the needed index and then a read block command to read the data.

The write command is

Function	Command	Byte Count	Data 1
Get Thermal Capabilities Request	0xCB	1	0x0

The read command is:

Function	Command
Get Thermal Capabilities Request	0xDB

Data returned from the I350 is

Function	Byte Count	Data 1 (Op-Code)	Data 2 (index)	Data 3 (Version)	Data 4 (Unit Types)	Data 5-6 (Number of Thresholds)
Get Thermal Sensor capabilities	24	0xDB	0x0x	1	See Table 10-45	See below

Data 7 (Accuracy)	Data 8 (Hysteresis)	Data 9 (M)	Data 10 (B)	Data 11 (K1)	Data 12 (K2)	Data 13-16 (Valid High going Actions)	Data 17-20 (Valid Low going Actions)	Data 21-24 (Valid Immediate Actions)	Data 25-26 (TJunction Max)
	See Below	See below				See Table 10-44	See Table 10-44	See Table 10-44	See below

- Version should always be 1.
- Unit Types describes the unit types measured according to the encoding in Table 10-45.
- Accuracy describes the accuracy of the reported measurements as follow:
  - 7:4: Max deviation of actual value above measurement in “unit types”.
  - 3:0: Max deviation of actual value below measurement in “unit types”.
- Max hysteresis - defines the max hysteresis value allowed in the implementation. A value of zero means hysteresis is not supported.
- Number of thresholds describes the number of up and down thresholds as follow:
  - 15:12: Reserved

  - 11:8: Max Number of mixed thresholds.
  - 7:4: Max number of up thresholds.
  - 3:0: Max number of down thresholds.
- Valid Actions “High going” thresholds - describes the actions that can be activated by the device as described in Table 10-44 when an high going threshold is crossed or the upper hysteresis of a “Low Going” Threshold is crossed.
- Valid Actions “Low going” thresholds - describes the actions that can be activated by the device as described in Table 10-44 when an low going threshold is crossed or the lower hysteresis of a “Low Going” Threshold is crossed.
- Valid Actions “Immediate” - describes the actions that can be activated by the device as described in Table 10-44 using a “Perform Thermal Sensor Action” command.
- M, B, K1, K2 - parameters used to translate the raw data read to a meaningful value according to the following formula:

$$Y = (MX + (B * 10^{K1})) * 10^{K2}$$

  where X is the measured value and Y is the value presented to the user.

Note: This formula is compliant with the definition of section 36.3 “Sensor Reading Conversion Formula” in IPMI 2.0

- Tjunction Max - The maximal junction temperature supported (125 C)

10.5.10.2.8 Get Thermal Sensor Configuration Command

The BMC can use this function to read the thermal sensor configuration for a given threshold. It uses a write command to set the needed index and then a read block command to read the data.

The write command is

Function	Command	Byte Count	Data 1	Data 1
Get Thermal configuration Request	0xCB	2	0x1	Index

The read command is

Function	Byte Count	Data 1 (Op-Code)	Data 2 (command)	Data 3 (index)	Data 4-5 (Threshold)
Get Thermal Sensor configuration	15	0xDB	0x1	index	The programmed threshold

Data 6-9 (Action “Going High”)	Data 10-13 (Action “Going Low”)	Data 14 (Direction)	Data 15 (Hysteresis)
The programmed actions as described in Table 10-44	The programmed actions as described in Table 10-44	See below	See below

The threshold and the Hysteresis are measured in “unit types”.

The "Actions Going High" field describes the actions to activate upon crossing of the threshold for "Going High" thresholds or when crossing the hysteresis for "Going Low" thresholds according to Table 10-44.

The "Actions Going Low" field describes the actions to activate upon crossing of the threshold for "Going Low" thresholds or when crossing the hysteresis for "Going High" thresholds according to Table 10-44.

Direction is encoded as follow:

- 0 = High Going
- 1 = Low Going

10.5.10.2.9 Get Thermal Sensor Status Command

The BMC can use this function to read the thermal sensor status. It uses a write command to set the needed index and then a read block command to read the data.

The write command is

Function	Command	Byte Count	Index
Get Thermal Sensor Status Request	0xCB	1	0x2

The read command is

Function	Byte Count	Data 1 (Op-Code)	Data 2 (command)	Data 3-4 (Measured Value)	Data 5-8 (Active Actions)	Data 9-10 (Threshold cross event)
Get Thermal Sensor Status	10	0xDB	0x2	The value measured in "unit types"	The currently active actions as described in Table 10-44	See below

"Threshold cross events" is a bitmap that describes which events where crossed since the last read of the status or since the activation of the thermal sensor (the latest of the two).

10.5.11 Example Configuration Steps

This section provides sample configuration settings for common filtering configurations. Three examples are presented. The examples are in pseudo code format, with the name of the SMBus command followed by the parameters for that command and an explanation.

10.5.11.1 Example 1 - Shared MAC, RMCP Only Ports

This example is the most basic configuration. The MAC address filtering is shared with the Host operating system and only traffic directed the RMCP ports (26Fh & 298h) is filtered. For this example, the BMC must issue gratuitous ARPs because no filter is enabled to pass ARP requests to the BMC.

10.5.11.1.1 Example 1 Pseudo Code

Step 1: Disable existing filtering

Receive Enable[00]

Utilizing the simple form of the Receive Enable command, this prevents any packets from reaching the BMC by disabling filtering:

Receive Enable Control 00h:

— Bit 0 [0] – Disable Receiving of packets

Step 2: Configure MDEF[0]

Update Manageability Filter Parameters [61, 0, C0000000]

Use the Update Manageability Filter Parameters command to update Decision Filters (MDEF) (parameter 61h). This will update MDEF[0], as indicated by the 2nd parameter (0).

MDEF[0] value of C0000000h:

— Bit 30 [1] – port 298h

— Bit 31 [1] – port 26Fh

Step 3: Configure *MNGONLY*

Update Manageability Filter Parameters [F, 0, 00000001]

Use the Update Manageability Filter Parameters command to update Manageability Only (*MNGONLY*) (parameter Fh) so that port 298h and 26Fh would not be sent to the Host.

— Bit [0] - *MDEF[0]* is exclusive to the BMC.

Step 4: - Enable Filtering

Receive Enable [05]

Using the simple form of the Receive Enable command:

Receive Enable Control 05h:

— Bit 0 [1] – Enable Receiving of packets

— Bit 2 [1] – Enable status reporting (such as link lost)

— Bit 5:4 [00] – Notification method = SMB Alert

— Bit 7 [0] – Use shared MAC

The resulting *MDEF* filters are as follows:

Table 10-25 Example 1 MDEF Results

		Manageability Decision Filter (MDEF)							
Filter		0	1	2	3	4	5	6	7
L2 Exact Address[3:0]	AND								
Broadcast	AND								
Manageability VLAN[7:0]	AND								
IPv6 Address[3:0]	AND								
IPv4 Address[3:0]	AND								
L2 Exact Address[3:0]	OR								
Broadcast	OR								
Multicast	AND								

Table 10-25 Example 1 MDEF Results

ARP Request	OR								
ARP Response	OR								
Neighbor Discovery	OR								
Port 0x298	OR	X							
Port 0x26F	OR	X							
Flex Port 7:0	OR								
Flex TCO	OR								

10.5.11.2 Example 2 - Dedicated MAC, Auto ARP Response and RMCP Port Filtering

This example shows a common configuration; the BMC has a dedicated MAC and IP address. Automatic ARP responses will be enabled as well as RMCP port filtering. By enabling Automatic ARP responses the BMC is not required to send the gratuitous ARPs as it did in Example 1.

For demonstration purposes, the dedicated MAC address will be calculated by reading the System MAC address and adding 1 do it, assume the System MAC is AABBCCDC. The IP address for this example will be 1.2.3.4. Additionally, the XSUM filtering will be enabled.

Note that not all Intel Ethernet Controllers support automatic ARP responses, please refer to product specific documentation.

10.5.11.2.1 Example 2 - Pseudo Code

Step 1: Disable existing filtering

`Receive Enable[00]`

Utilizing the simple form of the Receive Enable command, this prevents any packets from reaching the BMC by disabling filtering:

Receive Enable Control 00h:

— Bit 0 [0] – Disable Receiving of packets

Step 2: Read System MAC Address

`Get System MAC Address []`

Reads the System MAC address. Assume returned AABBCCDC for this example.

Step 3: Configure XSUM Filter

`Update Manageability Filter Parameters [01, 00800000]`

Use the Update Manageability Filter Parameters command to update Filters Enable settings (parameter 1). This set the Manageability Control (MANC) Register.

MANC Register 00800000h:

— Bit 23 [1] - XSUM Filter enable

Note that some of the following configuration steps manipulate the MANC register indirectly, this command sets all bits except XSUM to 0. It is important to either do this step before the others, or to read the value of the MANC and then write it back with only bit 32 changed. Also note that the XSUM enable bit may differ between Ethernet Controllers, refer to product specific documentation.

Step 4: Configure MDEF[0]

Update Manageability Filter Parameters [61, 0, C0000000]

Use the Update Manageability Filter Parameters command to update Decision Filters (MDEF) (parameter 61h). This will update MDEF[0], as indicated by the 2nd parameter (0).

MDEF value of 00000C00h:

— Bit 30 [1] – port 298h

— Bit 31 [1] – port 26Fh

Step 5: Configure MDEF[1]

Update Manageability Filter Parameters [61, 1, 10000000]

Use the Update Manageability Filter Parameters command to update Decision Filters (MDEF) (parameter 61h). This will update MDEF[1], as indicated by the 2nd parameter (1).

MDEF value of 10000000:

— Bit 28 [1] – ARP Requests

When Enabling Automatic ARP responses, the ARP requests still go into the manageability filtering system and as such need to be designated as also needing to be sent to the Host. For this reason a separate MDEF is created with only ARP request filtering enabled.

Refer to the next step for more details.

Step 6: Configure Manageability only

Update Manageability Filter Parameters [F, 0, 00000001]

Use the Update Manageability Filter Parameters command to update Manageability Only (MNGONLY) (parameter Fh) so that port 298h and 26Fh would not be sent to the Host.

— Bit [0] - MDEF[0] is exclusive to the BMC.

This allows ARP requests to be passed to both manageability and to the Host. Specified separate MDEF filter for ARP requests. If ARP requests had been added to *MDEF[0]* and then *MDEF[0]* specified in Management Only configuration then not only would RMCP traffic (ports 26Fh and 298h) be sent only to the BMC, ARP requests would have also been sent to the BMC only.

Step 7: Enable Filtering

Receive Enable [8D, AABBCCDD, 01020304, 00, 00, 00]

Using the advanced version Receive Enable command, the first parameter:

Receive Enable Control 8Dh:

— Bit 0 [1] – Enable Receiving of packets

— Bit 2 [1] – Enable status reporting (such as link lost)

— Bit 3 [1] – Enable Automatic ARP Responses

— Bit 5:4 [00] – Notification method = SMB Alert

— Bit 7 [1] - Use dedicated MAC

Second parameter is the MAC address (AABBCCDD).

Third Parameter is the IP address(01020304).

The last three parameters are zero when the notification method is SMB Alert.

The resulting *MDEF* filters are as follows:

Table 10-26 Example 2 MDEF Results

		Manageability Decision Filter (MDEF)							
Filter		0	1	2	3	4	5	6	7
L2 Exact Address[3:0]	AND								
Broadcast	AND								
Manageability VLAN[7:0]	AND								
IPv6 Address[3:0]	AND								
IPv4 Address[3:0]	AND								
L2 Exact Address[3:0]	OR								
Broadcast	OR								
Multicast	AND								
ARP Request	OR		X						
ARP Response	OR								
Neighbor Discovery	OR								
Port 0x298	OR	X							
Port 0x26F	OR	X							
Flex Port 7:0	OR								
Flex TCO	OR								

10.5.11.3 Example 3 - Dedicated MAC & IP Address

This example provided the BMC with a dedicated MAC and IP address and allows it to receive ARP requests. The BMC is then responsible for responding to ARP requests.

For demonstration purposes, the dedicated MAC address will be calculated by reading the System MAC address and adding 1 do it, assume the System MAC is AABBCCDC. The IP address for this example will be 1.2.3.4. For this example, the Receive Enable command is used to configure the MAC address filter.

In order for the BMC to be able to receive ARP Requests, it will need to specify a filter for this, and that filter will need to be included in the Manageability To Host filtering so that the Host OS may also receive ARP Requests.

10.5.11.3.1 Example 3 - Pseudo Code

Step 1: Disable existing filtering

`Receive Enable[00]`

Utilizing the simple form of the Receive Enable command, this prevents any packets from reaching the BMC by disabling filtering:

Receive Enable Control 00h:

— Bit 0 [0] – Disable Receiving of packets

Step 2: Read System MAC Address

`Get System MAC Address []`

Reads the System MAC address. Assume returned AABBCCDC for this example.

Step 3: Configure IP Address Filter

Update Manageability Filter Parameters [64, 00, 01020304]

Use the Update Manageability Filter Parameters to configure an IPv4 filter.

The 1st parameter (64h) specifies that we are configuring an IPv4 filter.

The 2nd parameter (00h) indicates which IPv4 filter is being configured, in this case filter 0.

The 3rd parameter is the IP address – 1.2.3.4.

Step 4: Configure MAC Address Filter

Update Manageability Filter Parameters [66, 00, AABBCCDD]

Use the Update Manageability Filter Parameters to configure a MAC Address filter.

The 1st parameter (66h) specifies that we are configuring a MAC Address filter.

The 2nd parameter (00h) indicates which MAC Address filter is being configured, in this case filter 0.

The 3rd parameter is the MAC Address - AABBCCDD

Step 5: Configure *MDEF[0]* for IP and MAC Filtering

Update Manageability Filter Parameters [61, 0, 00002001]

Use the Update Manageability Filter Parameters command to update Decision Filters (*MDEF*) (parameter 61h). This will update MDEF[0], as indicated by the 2nd parameter (0).

MDEF value of 00002001:

- Bit 0 [1] – MAC[0] Address Filtering
- Bit 13 [1] – IP[0] Address Filtering

Step 6: Configure *MDEF[1]*

Update Manageability Filter Parameters [61, 1, 10000000]

Use the Update Manageability Filter Parameters command to update Decision Filters (MDEF) (parameter 61h). This will update *MDEF[1]*, as indicated by the 2nd parameter (1).

MDEF value of 10000000:

- Bit 28 [1] – ARP Requests

Step 7: Configure the Management to Host Filter

Update Manageability Filter Parameters [F, 0, 00000001]

Use the Update Manageability Filter Parameters command to update Manageability Only (*MNGONLY*) (parameter Fh) so that the dedicated MAC/IP traffic would not be sent to the Host. Note that given the Host will not program this address in its L2 filtering, this step is not a must, unless the Host chooses to work in promiscuous mode.

- Bit [0] - MDEF[0] is exclusive to the BMC.

Step 8: Enable Filtering

Receive Enable [05]

Using the simple form of the Receive Enable command,:

Receive Enable Control 05h:

— Bit 0 [1] – Enable Receiving of packets
— Bit 2 [1] – Enable status reporting (such as link lost)
— Bit 5:4 [00] – Notification method = SMB Alert

The resulting *MDEF* filters are as follows:

Table 10-27 Example 3 MDEF Results

		Manageability Decision Filter (MDEF)							
Filter		0	1	2	3	4	5	6	7
L2 Exact Address[3:0]	AND	0001							
Broadcast	AND								
Manageability VLAN[7:0]	AND								
IPv6 Address[3:0]	AND								
IPv4 Address[3:0]	AND	0001							
L2 Exact Address[3:0]	OR								
Broadcast	OR								
Multicast	AND								
ARP Request	OR		X						
ARP Response	OR								
Neighbor Discovery	OR								
Port 0x298	OR								
Port 0x26F	OR								
Flex Port 7:0	OR								
Flex TCO	OR								

10.5.11.4 Example 4 - Dedicated MAC and VLAN Tag

This example shows an alternate configuration; the BMC has a dedicated MAC and IP address, along with a VLAN tag of 32h will be required for traffic to be sent to the BMC. This means that all traffic with VLAN a matching tag will be sent to the BMC.

For demonstration purposes, the dedicated MAC address will be calculated by reading the System MAC address and adding 1 do it, assume the System MAC is AABBCCDC. The IP address for this example will be 1.2.3.4 and the VLAN tag will be 0032h.

Additionally, the XSUM filtering will be enabled.

10.5.11.4.1 Example 4 - Pseudo Code

Step 1: Disable existing filtering

`Receive Enable[00]`

Utilizing the simple form of the Receive Enable command, this prevents any packets from reaching the BMC by disabling filtering:

Receive Enable Control 00h:
— Bit 0 [0] – Disable Receiving of packets

Step 2: - Read System MAC Address

Get System MAC Address []

Reads the System MAC address. Assume returned AABBCCDC for this example.

Step 3: Configure XSUM Filter

Update Manageability Filter Parameters [01, 00800000]

Use the Update Manageability Filter Parameters command to update Filters Enable settings (parameter 1). This set the Manageability Control (MANC) Register.

MANC Register 00800000h:

— Bit 23 [1] – XSUM Filter enable

Note that some of the following configuration steps manipulate the MANC register indirectly, this command sets all bits except XSUM to 0. It is important to either do this step before the others, or to read the value of the MANC and then write it back with only bit 32 changed. Also note that the XSUM enable bit may differ between Ethernet Controllers, refer to product specific documentation.

Step 4: Configure VLAN 0 Filter

Update Manageability Filter Parameters [62, 0, 0032]

Use the Update Manageability Filter Parameters command to configure VLAN filters. Parameter 62h indicates update to VLAN Filter, the 2nd parameter indicates which VLAN filter (0 in this case), the last parameter is the VLAN ID (0032h).

Step 5: Configure *MDEF[0]*

Update Manageability Filter Parameters [61, 0, 00000020]

Use the Update Manageability Filter Parameters command to update Decision Filters (MDEF) (parameter 61h). This will update MDEF[0], as indicated by the 2nd parameter (0).

MDEF value of 00000020:

— Bit 5 [1] – VLAN[0] AND

Step 6: Enable Filtering

Receive Enable [85, AABBCCDD, 01020304, 00, 00, 00]

Using the advanced version Receive Enable command, the first parameter:

Receive Enable Control 85h:

— Bit 0 [1] – Enable Receiving of packets

— Bit 2 [1] – Enable status reporting (such as link lost)

— Bit 5:4 [00] – Notification method = SMB Alert

— Bit 7 [1] – Use Dedicated MAC

Second parameter is the MAC address: AABBCCDD.

Third Parameter is the IP address: 01020304.

The last three parameters are zero when the notification method is SMBus Alert.

The resulting MDEF filters are as follows:

Table 10-28 Example 4 MDEF Results

		Manageability Decision Filter (MDEF)							
Filter		0	1	2	3	4	5	6	7
L2 Exact Address[3:0]	AND								0001
Broadcast	AND								
Manageability VLAN[7:0]	AND	X							
IPv6 Address[3:0]	AND								
IPv4 Address[3:0]	AND								
L2 Exact Address[3:0]	OR								
Broadcast	OR								
Multicast	AND								
ARP Request	OR								
ARP Response	OR								
Neighbor Discovery	OR								
Port 0x298	OR								
Port 0x26F	OR								
Flex Port 7:0	OR								
Flex TCO	OR								

10.5.12 SMBus Troubleshooting

This section outlines the most common issues found while working with pass-through using the SMBus sideband interface.

10.5.12.1 TCO Alert Line Stays Asserted After a Power Cycle

After the I350 resets, all its ports indicate a status change. If the BMC only reads status from one port (slave address), the other ports will continue to assert the TCO alert line.

Ideally, the BMC should use the ARA transaction (see Section 10.5.9) to determine which slave asserted the TCO alert. Many customers only wish to use one port for manageability thus using ARA might not be optimal.

An alternate to using ARA is to configure one of the ports to not report status and to set its SMBus timeout period. In this case, the SMBus timeout period determines how long a port asserts the TCO alert line awaiting a status read from a BMC; by default this value is zero (indicates an infinite timeout).

The SMBus configuration section of the EEPROM has a SMBus Notification Timeout (ms) field that can be set to a recommended value of 0xFF (for this issue). Note that this timeout value is for all slave addresses. Along with setting the SMBus Notification Timeout to 0xFF, it is recommended that the other ports be configured in the EEPROM to disable status alerting. This is accomplished by having the *Enable Status Reporting* bit set to 0b for the desired port in the LAN configuration section of the EEPROM.

The third solution for this issue is to have the BMC hard-code the slave addresses to always read from all ports. As with the previous solution, it is recommend that the other ports have status reporting disabled.

10.5.12.2 When SMBus Commands Are Always NACK'd

There are several reasons why all commands sent to the I350 from a BMC could be NACK'd. The following are most common:

- Invalid EEPROM Image — The image itself might be invalid or it could be a valid image and is not a pass-through image, as such SMBus connectivity is disabled.
- The BMC is not using the correct SMBus address — Many BMC vendors hard-code the SMBus address(es) into their firmware. If the incorrect values are hard-coded, the I350 does not respond.
  - The SMBus address(es) can be dynamically set using the SMBus ARP mechanism.
- The BMC is using the incorrect SMBus interface — The EEPROM might be configured to use one physical SMBus port; however, the BMC is physically connected to a different one.
- Bus Interference — the bus connecting the BMC and the I350 might be unstable.

10.5.12.3 SMBus Clock Speed Is 16.6666 KHz

This can happen when the SMBus connecting the BMC and the I350 is also tied into another device (such as an ICH) that has a maximum clock speed of 16.6666 KHz. The solution is to not connect the SMBus between the I350 and the BMC to this device.

10.5.12.4 A Network Based Host Application Is Not Receiving Any Network Packets

Reports have been received about an application not receiving any network packets. The application in question was NFS under Linux. The problem was that the application was using the RMPC/RMCP+ IANA reserved port 0x26F (623) and the system was also configured for a shared MAC and IP address with the OS and BMC.

The management control to Host configuration, in this situation, was setup not to send RMCP traffic to the OS (this is typically the correct configuration). This means that no traffic send to port 623 was being routed.

The solution in this case is to configure the problematic application NOT to use the reserved port 0x26F.

10.5.12.5 Unable to Transmit Packets from the BMC

If the BMC has been transmitting and receiving data without issue for a period of time and then begins to receive NACKs from the I350 when it attempts to write a packet, the problem is most likely due to the fact that the buffers internal to the I350 are full of data that has been received from the network but has yet to be read by the BMC.

Being an embedded device, the I350 has limited buffers that are shared for receiving and transmitting data. If a BMC does not keep the incoming data read, the I350 can be filled up This prevents the BMC form transmitting more data, resulting in NACKs.

If this situation occurs, the recommended solution is to have the BMC issue a Receive Enable command to disable more incoming data, read all the data from the I350, and then use the Receive Enable command to enable incoming data.

10.5.12.6 SMBus Fragment Size

The SMBus specification indicates a maximum SMBus transaction size of 32 bytes. Most of the data passed between the I350 and the BMC over the SMBus is RMCP/RMCP+ traffic, which by its very nature (UDP traffic) is significantly larger than 32 bytes in length. Multiple SMBus transactions may therefore be required to move data from the I350 to the BMC or to send a data from the BMC to the I350.

Recognizing this bottleneck, the I350 handles up to 240 bytes of data in a single transaction. This is a configurable setting in the EEPROM. The default value in the EEPROM images is 32, per the SMBus specification. If performance is an issue, increase this size.

During initialization, firmware within the I350 allocates buffers based upon the SMBus fragment size setting within the EEPROM. The I350 firmware has a finite amount of RAM for its use: the larger the SMBus fragment size, the fewer buffers it can allocate. Because this is true, BMC implementations must take care to send data over the SMBus efficiently.

For example, the I350 firmware has 3 KB of RAM it can use for buffering SMBus fragments. If the SMBus fragment size is 32 bytes then the firmware could allocate 96 buffers of size 32 bytes each. As a result, the BMC could then send a large packet of data (such as KVM) that is 800 bytes in size in 25 fragments of size 32 bytes apiece.

However, this might not be the most efficient way because the BMC must break the 800 bytes of data into 25 fragments and send each one at a time.

If the SMBus fragment size is changed to 240 bytes, the I350 firmware can create 12 buffers of 240 bytes each to receive SMBus fragments. The BMC can now send that same 800 bytes of KVM data in only four fragments, which is much more efficient.

The problem of changing the SMBus fragment size in the EEPROM is if the BMC does not also reflect this change. If a programmer changes the SMBus fragment size in the I350 to 240 bytes and then wants to send 800 bytes of KVM data, the BMC can still only send the data in 32 byte fragments. As a result, firmware runs out of memory.

This is because firmware created the 12 buffers of 240 bytes each for fragments; however, the BMC is only sending fragments of size 32 bytes. This results in a memory waste of 208 bytes per fragment. Then when the BMC attempts to send more than 12 fragments in a single transaction, the I350 NACKs the SMBus transaction due to not enough memory to store the KVM data.

In summary, if a programmer increases the size of the SMBus fragment size in the EEPROM (recommended for efficiency purposes) take care to ensure that the BMC implementation reflects this change and uses that fragment size to its fullest when sending SMBus fragments.

10.5.12.7 Losing Link

Normal behavior for the Ethernet Controller when the system powers down or performs a reset is for the link to temporarily go down and then back up again to re-negotiate the link speed. This behavior can have adverse affects on manageability.

For example if there is an active FTP or Serial Over LAN session to the BMC, this connection may be lost. In order to avoid this possible situation, the BMC can use the Management Control command detailed in Section 10.5.10.1.5 to ensure the link stays active at all times.

This command is available when using the NC-SI sideband interface as well.

Care should be taken with this command, if the driver negotiates the maximum link speed, the link speed will remain the same when the system powers down or resets. This may have undesirable power consumption consequences. Currently, when using NC-SI, the BMC can re-negotiate the link speed. That functionality is not available when using the SMBus interface.

10.5.12.8 Enable XSum Filtering

If XSum filtering is enabled, the BMC does not need to perform the task of checking this checksum for incoming packets. Only packets that have a valid XSum is passed to the BMC. All others are silently discarded.

This is a way to offload some work from the BMC.

10.5.12.9 Still Having Problems?

If problems still exist, contact your field representative. Be prepared to provide the following:

- A SMBus trace if possible
- A dump of the EEPROM image. This should be taken from the actual I350, rather than the EEPROM image provided by Intel. Parts of the EEPROM image are changed after writing (such as the physical EEPROM size).

10.6 NC-SI Pass Through Interface

The Network Controller Sideband Interface (NC-SI) is a DMTF industry standard protocol for the sideband interface. NC-SI uses a modified version of the industry standard RMII interface for the physical layer as well as defining a new logical layer.

The NC-SI specification can be found at:

http://www.dmtf.org/

10.6.1 Overview

10.6.1.1 Terminology

The terminology in this document is taken from the NC-SI specification.

Table 10-29 NC-SI Terminology

Term	Definition
Frame Versus Packet	Frame is used in reference to Ethernet, whereas packet is used everywhere else.
External Network Interface	The interface of the network controller that provides connectivity to the external network infrastructure (port).
Internal Host Interface	The interface of the network controller that provides connectivity to the Host OS running on the platform.
Management Controller (BMC)	An intelligent entity comprising of HW/FW/SW, that resides within a platform and is responsible for some or all management functions associated with the platform (BMC, service processor, etc.).

Table 10-29 NC-SI Terminology

Term	Definition
Network Controller (NC)	The component within a system that is responsible for providing connectivity to the external Ethernet network world.
Remote Media	The capability to allow remote media devices to appear as if they were attached locally to the Host.
Network Controller Sideband Interface	The interface of the network controller that provides connectivity to a management controller. It can be shorten to sideband interface as appropriate in the context.
Interface	This refers to the entire physical interface, such as both the transmit and receive interface between the management controller and the network controller.
Integrated Controller	The term integrated controller refers to a network controller device that supports two or more channels for NC-SI that share a common NC-SI physical interface. For example, a network controller that has two or more physical network ports and a single NC-SI bus connection.
Multi-Drop	Multi-drop commonly refers to the case where multiple physical communication devices share an electrically common bus and a single device acts as the master of the bus and communicates with multiple slave or target devices. In NC-SI, a management controller serves the role as the master, and the network controllers are the target devices.
Point-to-Point	Point-to-point commonly refers to the case where only two physical communication devices are interconnected via a physical communication medium. The devices might be in a master/slave relationship, or could be peers. In NC-SI, point-to-point operation refers to the situation where only a single management controller and single network controller package are used on the bus in a master/slave relationship where the management controller is the master.
Channel	The control logic and data paths supporting NC-SI pass-through operation on a single network interface (port). A network controller that has multiple network interface ports can support an equivalent number of NC-SI channels.
Package	One or more NC-SI channels in a network controller that share a common set of electrical buffers and common buffer control for the NC-SI bus. Typically, there will be a single, logical NC-SI package for a single physical network controller package (chip or module). However, the specification allows a single physical chip or module to hold multiple NC-SI logical packages.
Control Traffic/Messages/Packets	Command, response and notification packets transmitted between the BMC and the I350 for the purpose of managing NC-SI.
Pass-Through Traffic/Messages/ Packets	Non-control packets passed between the external network and the BMC through the I350.
Channel Arbitration	Refer to operations where more than one of the network controller channels can be enabled to transmit pass-through packets to the BMC at the same time, where arbitration of access to the RXD, CRS_DV, and RX_ER signal lines is accomplished either by software of hardware means.
Logically Enabled/Disabled NC	Refers to the state of the network controller wherein pass-through traffic is able/unable to flow through the sideband interface to and from the management controller, as a result of issuing Enable/Disable Channel command.
NC RX	Defined as the direction of ingress traffic on the external network controller interface
NC TX	Defined as the direction of egress traffic on the external network controller interface
NC-SI RX	Defined as the direction of ingress traffic on the sideband enhanced NC-SI Interface with respect to the network controller.
NC-SI TX	Defined as the direction of egress traffic on the sideband enhanced NC-SI Interface with respect to the network controller.

10.6.1.2 System Topology

In NC-SI each physical endpoint (NC package) can have several logical slaves (NC channels).

NC-SI defines that one management controller and up to four network controller packages can be connected to the same NC-SI link.

Figure 10-6 shows an example topology for a single BMC and a single NC package. In this example, the NC package has two NC channels.

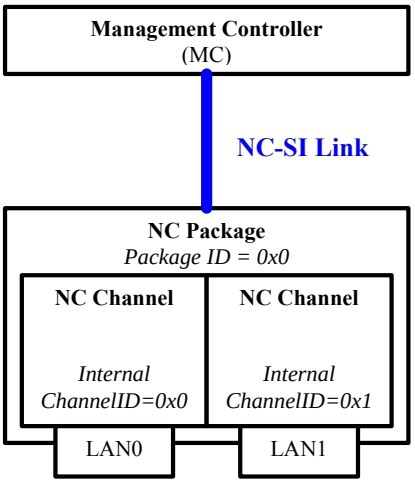


Figure 10-6 Single NC Package, Two NC Channels

Figure 10-7 shows an example topology for a single BMC and two NC packages. In this example, one NC package has two NC channels and the other has only one NC channel. Scenarios in which the NC-SI lines are shared by multiple NCs (Figure 10-7) mandate an arbitration mechanism. The arbitration mechanism is described in Section 10.6.8.1.

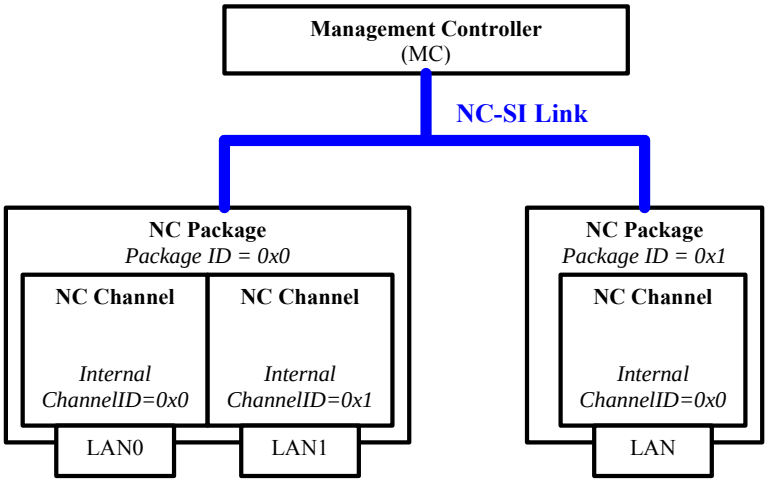


Figure 10-7 Two NC Packages (Left, with Two NC Channels and Right, with One NC Channel)

Note: Channel numbers should match PCI function numbers. So when PCI functions are swapped (FACTPS.LAN Function Sel == 1), then the channels should be swapped also.

10.6.1.3 Data Transport

Since NC-SI is based upon the RMII transport layer, data is transferred in the form of Ethernet frames.

NC-SI defines two types of transmitted frames:

1. Control frames:
   a. Configures and control the interface
   b. Identified by a unique EtherType in their L2 header
2. Pass-through frames:
   a. Actual LAN pass-through frames transferred from/to the BMC
   b. Identified as not being a control frame
   c. Attributed to a specific NC channel by their source MAC address (as configured in the NC by the BMC)

Note: The NC-SI spec allows reception of data packets up to 1536 bytes. However, the I350 allows only legal Ethernet packets to pass. So packets larger than 1518 bytes plus optional VLAN headers will be dropped.

10.6.1.3.1 Control Frames

NC-SI control frames are identified by a unique NC-SI EtherType (0x88F8).

Control frames are used in a single-threaded operation, meaning commands are generated only by the BMC and can only be sent one at a time. Each command from the BMC is followed by a single response from the NC (command-response flow), after which the BMC is allowed to send a new command.

The only exception to the command-response flow is the Asynchronous Event Notification (AEN). These control frames are sent unsolicited from the NC to the BMC.

AEN functionality by the NC must be disabled by default, until activated by the BMC using the Enable AEN commands.

In order to be considered a valid command, a control frame must:

1. Comply with the NC-SI header format.
2. Be targeted to a valid channel in the package via the Package ID and Channel ID fields. For example, to target a NC channel with package ID of 0x2 and internal channel ID of 0x5, the BMC must set the channel ID inside the control frame to 0x45. The channel ID is composed of three bits of package ID and five bits of internal channel ID.
3. Contain a correct payload checksum (if used).
4. Meet any other condition defined by NC-SI.

There are also commands (such as select package) targeted to the package as a whole. These commands must use an internal channel ID of 0x1F.

For details, refer to the NC-SI specification.

10.6.1.3.2 NC-SI Frames Receive Flow

Figure 10-8 shows the flow for frames received on the NC from the BMC.

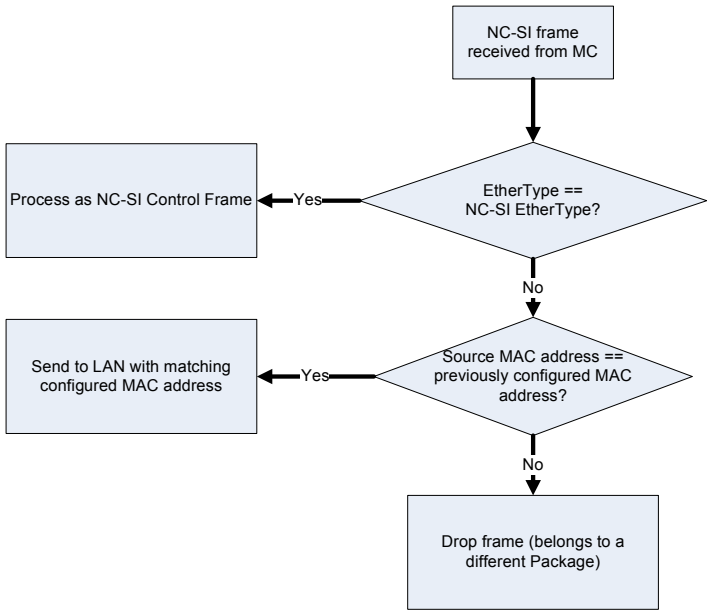


Figure 10-8 NC-SI Frames Receive Flow for the NC

10.6.2 Supported Features

The I350 supports all the mandatory features of the NC-SI specification (rev 1.0.0). Table 10-30 lists the supported commands.

Table 10-31 lists optional features supported.

Table 10-30 Supported NC-SI Commands

Command	Supported over RMII	Supported over MCTP
Clear initial state	Yes	Yes
Get Version ID	Yes	Yes
Get Parameters	Yes	Yes
Get Controller Packet Statistics	Yes, partially	Yes, partially
Get Link Status[1]	Yes	Yes
Enable Channel	Yes	Yes[2]
Disable Channel	Yes	Yes[2]
Reset Channel	Yes	Yes[2]
Enable VLAN	Yes[1,3,4]	No[2]
Disable VLAN	Yes	No[2]
Enable Broadcast Filter	Yes	No[2]
Disable Broadcast Filter	Yes	No[2]
Set MAC Address	Yes	No[2]

Table 10-30 Supported NC-SI Commands (Continued)

Command	Supported over RMII	Supported over MCTP
Get NC-SI Statistics	Yes, partially	Yes, partially
Set NC-SI Flow-Control	Yes	No
Set Link Command	Yes	Yes
Enable Global multicast Filter	Yes	No[2]
Disable Global multicast Filter	Yes	No[2]
Get Capabilities	Yes	Yes[5]
Set VLAN Filters	Yes	No[2]
AEN Enable	Yes	Yes
Get NC-SI Pass-Through Statistics	Yes, partially	No[2]
Select Package	Yes	Yes
Deselect Package	Yes	Yes
Enable Channel Network TX	Yes	No
Disable Channel Network TX	Yes	No
OEM Command[6]	Yes	Yes

1. When working with SGMII interface, this command is not supported.
2. In MCTP over SMBus mode, only control commands are supported and not pass through traffic - thus many of the regular NC-SI commands are not supported or are supported in a limited manner, only to allow control and status reporting for the device.
3. When one of the LAN devices is assigned for the sole use of the manageability and its LAN PCI-E function is disabled, using the NC-SI Set Link command while advertising multiple speeds and enabling Auto-Negotiation, will result in the lowest possible speed chosen. To enable higher link speed, the BMC should not advertise speeds that are below the desired link speed. When doing it, changing the power state of the LAN device will have no effect and the link speed will not be re-negotiated.
4. The I350 does not support filtering of User priority/CFI Bits of VLAN
5. When the "Get Capabilities" command is received over MCTP, the I350 returns the full filtering capabilities reported over RMII even that pass through is not available via MCTP.
6. See Section 10.6.3 for details.

Table 10-31 Optional NC-SI Features Support

Feature	Implement	Details
AENs	Yes	The Driver state AEN may be emitted up to 15 sec. after actual driver change.
Get Controller Packet Statistics command	Yes, partially	Supports the following counters [1]: 2-9,13-16[2]
Get NC-SI statistics	Yes, partially	Support the following counters:[3] 1-4, 7[4].
Get NC-SI Pass-Through Statistics	Yes, partially	Support the following counters: 2. Support the following counters only when the OS is down: 1, 6, 7.
VLAN Modes	Yes, partially	Support only modes 1, 3.
Buffering Capabilities	Yes	8Kb
MAC Address Filters	Yes	Supports 2 mixed MAC addresses per port.
Channel Count	Yes	Supports 4 channels.
VLAN Filters	Yes	Support 8 VLAN filters per port. Filtering is ignoring the CFI bit and the 802.1P priority bits

Table 10-31 Optional NC-SI Features Support

Feature	Implement	Details
Broadcast Filters	Yes	Support the following filters: ARP DHCP Net BIOS
Multicast Filters	Yes	Supports the following filters[5]: IPv6 Neighbor Advertisement IPv6 Router Advertisement DHCPv6 relay and server multicast
Hardware Arbitration	Yes	Supports NC-SI HW arbitration.

1. *TCTL.EN* should be set to 1b to activate TX related counters and *RCTL.RXEN*, *MANC.RCV_EN* or *WUC.APME* should be set to enable RX related counters.
2. As described in the Get Controller Packet Statistics Counter Numbers table in NC-SI spec.
3. The I350 does not increment the NC-SI Control Packets Dropped counter when packets with Checksum errors are dropped. In this case, only the NC-SI Command Checksum Errors counter is updated.
4. As described in Get NC-SI Statistics Response Counters table in NC-SI spec.
5. Supports only when all three filters are enabled.

10.6.2.1 Set Link Error Codes

The following rules are used to define the error code returned for Set Link command in case an invalid configuration is requested:

1. Host Driver Check: If host device driver is present, return a Command Specific Response (0x9) with a Set Link Host OS/Driver Conflict Reason (0x1).
2. Speed Present Check: If no speed is selected, return a General Reason Code for a failed command (0x1) with Parameter Is Invalid, Unsupported, or Out-of-Range Reason (0x2).
3. Parameter Validity:
   a. Auto Negotiation Parameter Validation: If Auto Negotiation is requested and none of the selected parameters are valid for the device, return a General Reason Code for a failed command (0x1) with a Parameter Is Invalid, Unsupported, or Out-of-Range Reason (0x2).

Note: This means that, for example, a command requesting 10G on a 1G device will succeed provided that the command requests at least one other supported speed. The same goes for an unsupported duplex setting (a device with no HD support will accept a command with both FD and HD set), and also for HD being requested with speeds of 1G and higher as long as a speed below 1G is also requested (and is supported in HD). The device will simply ignore the unsupported parameters.

   b. Force Mode Parameter Validation:
      1. If more than one link speed is being forced, then return a General Reason Code for a failed command (0x1) and a Command Specific Reason with a Set Link Speed Conflict Reason (0x0905).
      2. If more than one duplex setting is being forced, then return a General Reason Code for a failed command (0x1) with Parameter Is Invalid, Unsupported, or Out-of-Range Reason (0x2).
      3. If 1G and above is requested with HD, then return a General Reason Code for a failed command (0x1) and a Command Specific Reason with Set Link Parameter Conflict Error (0x0903).
4. Media Type Compatibility Check: If current media type is not compatible for the requested link parameters, return a General Reason Code for a failed command (0x1) and a Command Specific Reason with Set Link Media Conflict Error (0x0902).

5. Power State Compatibility Check: If current power state does not allow for the requested link parameters, return a General Reason Code for a failed command (0x1) and a Command Specific Reason with Set Link Power Mode Conflict Error (0x0904).
6. If for some reason the hardware cannot perform the flow required for the command, return a General Reason Code for a failed command (0x1) and a Command Specific Reason with Link Command Failed-Hardware Access Error (0x0906).

10.6.3 NC-SI Mode — Intel Specific Commands

In addition to regular NC-SI commands, the following Intel vendor specific commands are supported. The purpose of these commands is to provide a means for the BMC to access some of the Intel-specific features present in the I350.

10.6.3.1 Overview

The following features are available via the NC-SI OEM specific commands:

- Receive filters:
  - Packet Addition Decision Filters 0x0…0x4
  - Packet Reduction Decision Filters 0x5…0x7
  - *MNGONLY* register (controls the forwarding of manageability packets to the Host)
  - Flex 128 filter
  - Flex TCP/UDP port filters 0x0...0x2
  - IPv4/IPv6 filters
- Get System MAC Address — This command enables the BMC to retrieve the system MAC address used by the NC. This MAC address can be used for a shared MAC address mode.
- Keep PHY Link Up (*Veto* bit) Enable/Disable — This feature enables the BMC to block PHY reset, which might cause session loss.
- TCO Reset — Enables the BMC to reset the I350.
- Checksum offloading — Offloads IP/UDP/TCP checksum checking from the BMC.

These commands are designed to be compliant with their corresponding SMBus commands (if existing). All of the commands are based on a single DMTF defined NC-SI command, known as OEM Command. This command is as follows.

10.6.3.1.1 OEM Command (0x50)

The OEM command can be used by the BMC to request the sideband interface to provide vendor-specific information.

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...	Intel Command Number	Optional Data		

10.6.3.1.2 OEM Response (0xD0)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...27	Intel Command Number	Optional Return Data		

Note: Responses have no command-specific reason code, unless otherwise specified within the command.

10.6.3.2 Command Summary

Table 10-32 OEM Specific Command Response and Reason Codes

Response Code		Reason Code	
Value	**Description**	**Value**	**Description**
0x1	Command Failed	0x5081	Invalid Intel Command Number
		0x5082	Invalid Intel Command Parameter Number
		0x5087	Invalid Driver State
		0x5088	Invalid EEPROM

Table 10-33 Command Summary

Intel Command	Parameter	Command Name	Supported in MCTP
0x00	0x00	Set IP Filters Control	No
0x01	0x00	Get IP Filters Control	No
0x02	0x0F	Set Manageability Only	No
	0x10	Set Flexible 128 Filter Mask and Length	
	0x11	Set Flexible 128 Filter Data	
	0x61	Set Packet Addition Filters	
	0x63	Set Flex TCP/UDP Port Filters	
	0x64	Set Flex IPv4 Address Filters	
	0x65	Set Flex IPv6 Address Filters	
	0x67	Set EtherType Filter	
	0x68	Set Packet Addition Extended Filter	

Table 10-33 Command Summary (Continued)

Intel Command	Parameter	Command Name	Supported in MCTP
0x03	0x0F	Get Manageability Only	No
	0x10	Get Flexible 128 Filter Mask and Length	
	0x11	Get Flexible 128 Filter Data	
	0x61	Get Packet Addition Filters	
	0x63	Get Flex TCP/UDP Port Filters	
	0x64	Get Flex IPv4 Address Filters	
	0x65	Get Flex IPv6 Address Filters	
	0x67	Get EtherType Filter	
	0x68	Get Packet Addition Extended Filter	
0x04	0x00	Set Unicast Packet Reduction	No
	0x01	Set Multicast Packet Reduction	
	0x02	Set Broadcast Packet Reduction	
	0x10	Set Extended Unicast Packet Reduction	
	0x11	Set Extended Multicast Packet Reduction	
	0x12	Set Extended Broadcast Packet Reduction	
0x05	0x00	Get Unicast Packet Reduction	No
	0x01	Get Multicast Packet Reduction	
	0x02	Get Broadcast Packet Reduction	
	0x10	Get Extended Unicast Packet Reduction	
	0x11	Get Extended Multicast Packet Reduction	
	0x12	Get Extended Broadcast Packet Reduction	
0x06	N/A	Get System MAC Address	Yes
0x20	N/A	Set Intel Management Control	No
0x21	N/A	Get Intel Management Control	No
0x22	N/A	Perform TCO Reset	Yes
0x23	N/A	Enable IP/UDP/TCP Checksum Offloading	No
0x24	N/A	Disable IP/UDP/TCP Checksum Offloading	No
0x40	0x01	Enable OS2BMC flow	No
	0x02	Enable Network to BMC flow	
	0x03	Enable Both Network to BMC and Host to BMC flow	
0x41	N/A	Get OS2BMC parameters	No
0x48	0x1	Get Controller information	Yes
0x4C	0x0	Get Thermal Sensor Capabilities	Yes
	0x1	Get Thermal Sensor Configuration	Yes
	0x2	Get Thermal Sensor Status	Yes
0x4D	0x1	Set Thermal Sensor Configuration	Yes
	0x2	Perform Thermal Action	Yes

10.6.3.3 Set Intel Filters Control — IP Filters Control Command (Intel Command 0x00, Filter Control Index 0x00)

This command controls different aspects of the Intel filters.

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...23	0x00	0x00	IP Filters control (3-2)	
24...25	IP Filters Control (1-0)			

Where "IP Filters Control" has the following format.

Table 10-34 IP Filters Control

Bit #	Name	Description	Default Value
0	IPv4/IPv6 Mode	IPv6 (0b): There are zero IPv4 filters and four IPv6 filters IPv4 (1b): There are four IPv4 filters and three IPv6 filters. See Section 8.21.8 or Section 10.3.3.6 for details.	1b
1...31	Reserved		

10.6.3.3.1 Set Intel Filters Control — IP Filters Control Response (Intel Command 0x00, Filter Control Index 0x00)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...27	0x00	0x00		

10.6.3.4 Get Intel Filters Control Commands (Intel Command 0x01)

10.6.3.4.1 Get Intel Filters Control — IP Filters Control Command (Intel Command 0x01, Filter Control Index 0x00)

This command reflects different aspects of the Intel filters.

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**

00...15	NC-SI Header		
16...19	Manufacturer ID (Intel 0x157)		
20...21	0x01	0x00	

10.6.3.4.2 Get Intel Filters Control — IP Filters Control Response (Intel Command 0x01, Filter Control Index 0x00)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...27	0x01	0x00	IP Filters Control (3-2)	
28...29	IP Filters Control (1-0)			

IP Filter Control: See Table 10-34.

10.6.3.5 Set Intel Filters Formats

10.6.3.5.1 Set Intel Filters Command (Intel Command 0x02)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...21	0x02	Parameter Number	Filters Data (optional)	

10.6.3.5.2 Set Intel Filters Response (Intel Command 0x02)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...	0x02	Filter Control Index	Return Data (Optional)	

10.6.3.5.3 Set Intel Filters — Manageability Only Command (Intel Command 0x02, Filter Parameter 0x0F)

This command sets the *MNGONLY* register. The *MNGONLY* register controls whether pass-through packets destined to the BMC are not forwarded to the Host OS. The *MNGONLY* register structure is described in Table 10-4.

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...23	0x02	0x0F	Manageability Only (3-2)	
24...25	Manageability Only (1-0)			

10.6.3.5.4 Set Intel Filters — Manageability Only Response (Intel Command 0x02, Filter Parameter 0x0F)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...25	0x02	0x0F		

10.6.3.5.5 Set Intel Filters — Flex Filter Enable Mask and Length Command (Intel Command 0x02, Filter Parameter 0x10)

The following command sets the Intel flex filters mask and length. See Section 10.3.3.5 for details of the programming.

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...23	0x02	0x10	Mask Byte 1	Mask Byte 2
24...27	...	...	...	...
28...31	...	...	...	...
32...35	...	...	...	...
36...37	...	Mask Byte 16	Reserved	Reserved
38	Length			

10.6.3.5.6 Set Intel Filters — Flex Filter Enable Mask and Length Response (Intel Command 0x02, Filter Parameter 0x10)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...25	0x02	0x10		

10.6.3.5.7 Set Intel Filters — Flex Filter Data Command (Intel Command 0x02, Filter Parameter 0x11)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...	0x02	0x11	Filter Data Group	Filter Data 1
	...	Filter Data N		

The Filter Data Group parameter defines which bytes of the Flex filter are set by this command:

Table 10-35 Filter Data Group

Code	Bytes programmed	Filter Data Length
0x0	bytes 0-29	1 - 30
0x1	bytes 30-59	1 - 30
0x2	bytes 60-89	1 - 30
0x3	bytes 90-119	1 - 30
0x4	bytes 120-127	1 - 8

Note: Using this command to configure the filters data must be done after the flex filter mask command is issued and the mask is set.

10.6.3.5.8 Set Intel Filters — Flex Filter Data Response (Intel Command 0x02, Filter Parameter 0x11)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...25	0x02	0x11		

Note: If Filter Data Length is larger than specified in Table 10-35 an Out of Range Reason code is returned.

10.6.3.5.9 Set Intel Filters — Packet Addition Decision Filter Command (Intel Command 0x02, Filter Parameter 0x61)

Note: This command is kept and supported for legacy reasons, however it is recommended to use the Set Intel Filters - Packet Addition Extended Decision Filter Command (Intel Command 0x02, Filter parameter 0x68 - Section 10.6.3.5.19) instead.

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...23	0x02	0x61	Filter index	Decision Filter (MSB)
24...25	.....		Decision Filter (LSB)	

Filter index range: 0x0...0x4.

Note: If the filter index is bigger than 4, a command failed Response Code is returned with no reason.

Table 10-36 Filter Values

Bit #	Name	Description
3:0	Exact (AND)	If set, packets must match exact filter 0 to 3 respectively.
4	Broadcast (AND)	If set, packets must match the broadcast filter.
12:5	VLAN (AND)	If set, packets must match VLAN filter 0 to 7 respectively.
16:13	IPv4 Address (AND)	If set, packets must match IPv4 filter 0 to 3 respectively
20:17	IPv6 Address (AND)	If set, packets must match IPv4 filter 0 to 3 respectively
24:21	Exact (OR)	If set, packets must match exact filter 0 to 3 respectively or a different OR filter.
25	Broadcast (OR)	If set, packets can pass if match the broadcast filter or a different OR filter.
26	Multicast (AND)	If set, packets must match the multicast filter.
27	ARP Request (OR)	If set, packets can pass if match the ARP request filter or a different OR filter.
28	ARP Response (OR)	If set, packets can pass if match the ARP response filter or a different OR filter.
29	Neighbor Discovery (OR)	If set, packets can pass if match the neighbor discovery filter or a different OR filter.
30	Port 0x298 (OR)	If set, packets can pass if match a fixed TCP/UDP port 0x298 filter or a different OR filter.
31	Port 0x26F (OR)	If set, packets can pass if match a fixed TCP/UDP port 0x26F filter or a different OR filter.

The filtering is done according to the mechanism described in Section 10.3.4.1.

Note: These filter settings operate according to the VLAN mode, as configured according to the DMTF NC-SI specification. After disabling packet reduction filters, the BMC must re-set the VLAN mode using the Set VLAN command.

10.6.3.5.10 Set Intel Filters — Packet Addition Decision Filter Response (Intel Command 0x02, Filter Parameter 0x61)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...25	0x02	0x61		

10.6.3.5.11 Set Intel Filters — Flex TCP/UDP Port Filter Command (Intel Command 0x02, Filter Parameter 0x63)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...23	0x02	0x63	Port filter index	TCP/UDP Port MSB
24	TCP/UDP Port LSB			

Filter index range: 0x0...0x2.

If the filter index is bigger than 2, a command failed Response Code is returned with no reason.

10.6.3.5.12 Set Intel Filters — Flex TCP/UDP Port Filter Response (Intel Command 0x02, Filter Parameter 0x63)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...25	0x02	0x63		

10.6.3.5.13 Set Intel Filters — IPv4 Filter Command (Intel Command 0x02, Filter Parameter 0x64)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...23	0x02	0x64	IP filter index	IPv4 Address (MSB)
24...25	...		IPv4 Address (LSB)	

IPv4 Mode: Filter index range: 0x0...0x3.

IPv6 Mode: This command should not be used in IPv6 mode.

10.6.3.5.14 Set Intel Filters — IPv4 Filter Response (Intel Command 0x02, Filter Parameter 0x64)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...25	0x02	0x64		

10.6.3.5.15 Set Intel Filters — IPv6 Filter Command (Intel Command 0x02, Filter Parameter 0x65)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...23	0x02	0x65	IP filter index	...IPv6 Address (MSB, byte 15)
24...27	...	...	...	...
28...31	...	...	...	...
32...35	...	...	...	...
36...37	...		IPv6 Address (LSB, byte 0)	

Note: The filters index range can vary according to the IPv4/IPv6 mode setting in the Filters Control command.

IPv4 Mode: Filter index range: 0x0...0x2.

IPv6 Mode: Filter index range: 0x0...0x3.

10.6.3.5.16 Set Intel Filters — IPv6 Filter Response (Intel Command 0x02, Filter Parameter 0x65)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...25	0x02	0x65		

If the IP filter index is larger the 3, a command failed Response Code will be returned, with no reason.

10.6.3.5.17 Set Intel Filters - EtherType Filter Command (Intel Command 0x02, Filter parameter 0x67)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...23	0x02	0x67	EtherType Filter Index	EtherType Filter MSB
24...27	...	...	EtherType Filter LSB	

Where the EtherType Filter has the format as described in Section 8.21.3.

10.6.3.5.18 Set Intel Filters - EtherType Filter Response (Intel Command 0x02, Filter parameter 0x67)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...25	0x02	0x67		

If the Ethertype filter Index is greater than 3, a command failed Response Code is returned with no reason.

10.6.3.5.19 Set Intel Filters - Packet Addition Extended Decision Filter Command (Intel Command 0x02, Filter parameter 0x68)

See Figure 10-2 for description of the decision filters structure.

The command shall overwrite any previously stored value.

Note: Previous "Set Intel Filters – Packet Addition Decision Filter" command (0x61) is kept and supported for legacy reasons - If previous "Decision Filter" command is called – it should set the Decision Filter 0 as provided. The extended Decision Filter remains unchanged. However, it is recommended to use this set of commands for packet addition.

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...23	0x02	0x68	Extended Decision filter Index	Extended Decision filter 1 MSB
24...27	...	...	Extended Decision filter 1 LSB	Extended Decision filter 0 MSB
28...30	...	...	Extended Decision filter 0 LSB	

Extended Decision filter Index Range: 0...4

Filter 0: See Table 10-37.

Filter 1: See Table 10-38.

Table 10-37 Filter Values

Bit #	Name	Description
3:0	Exact (AND)	If set, packets must match exact filter 0 to 3 respectively.
4	Broadcast (AND)	If set, packets must match the broadcast filter.
12:5	VLAN (AND)	If set, packets must match VLAN filter 0 to 7 respectively.
16:13	IPv4 Address (AND)	If set, packets must match IPv4 filter 0 to 3 respectively
20:17	IPv6 Address (AND)	If set, packets must match IPv4 filter 0 to 3 respectively
24:21	Exact (OR)	If set, packets must match exact filter 0 to 3 respectively or a different OR filter.
25	Broadcast (OR)	If set, packets can pass if match the broadcast filter or a different OR filter.
26	Multicast (AND)	If set, packets must match the multicast filter.
27	ARP Request (OR)	If set, packets can pass if match the ARP request filter or a different OR filter.
28	ARP Response (OR)	If set, packets can pass if match the ARP response filter or a different OR filter.
29	Neighbor Discovery (OR)	If set, packets can pass if match the neighbor discovery filter or a different OR filter.
30	Port 0x298 (OR)	If set, packets can pass if match a fixed TCP/UDP port 0x298 filter or a different OR filter.
31	Port 0x26F (OR)	If set, packets can pass if match a fixed TCP/UDP port 0x26F filter or a different OR filter.

Table 10-38 Extended Filter Values

Bit #	Name	Description
3:0	Ethertype 0 -3 (AND)	If set, packets must match the Ethertype filter 0 to 3 respectively.
7:4	Reserved	Reserved
11:8	Ethertype 0 -3 (OR)	If set, packets can pass if match the Ethertype filter 0 to 3 respectively or a different OR filter.
15:12	Reserved	Reserved
16	Flex port 0 (OR)	If set, packets can pass if match the TCP/UDP Port filter 0

Table 10-38 Extended Filter Values (Continued)

Bit #	Name	Description
17	Flex port 1 (OR)	If set, packets can pass if match the TCP/UDP Port filter 1
18	Flex port 2 (OR)	If set, packets can pass if match the TCP/UDP Port filter 2
19	DHCPv6 (OR)	If set, packets can pass if match the DHCPv6 port (0x0223)
20	DHCP Client (OR)	If set, packets can pass if match the DHCP Server port (0x0043)
21	DHCP Server (OR)	If set, packets can pass if match the DHCP Client port (0x0044)
22	NetBIOS Name Service (OR)	If set, packets can pass if match the NetBIOS Name Service port (0x0089)
23	NetBIOS Datagram Service (OR)	If set, packets can pass if match the NetBIOS Datagram Service port (0x008A)
24	Flex TCO (OR)	If set, packets can pass if match the Flex 128 TCO filter
31:25	Reserved	

10.6.3.5.20 Set Intel Filters – Packet Addition Extended Decision Filter Response (Intel Command 0x02, Filter parameter 0x68)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...25	0x02	0x68		

If the Extended Decision filter Index is bigger than 5, a command failed Response Code is returned with no reason.

10.6.3.6 Get Intel Filters Formats

10.6.3.6.1 Get Intel Filters Command (Intel Command 0x03)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...21	0x03	Parameter Number		

10.6.3.6.2 Get Intel Filters Response (Intel Command 0x03)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			

16...19	Response Code		Reason Code
20...23	Manufacturer ID (Intel 0x157)		
24...25	0x03	Parameter Number	Optional Return Data

10.6.3.6.3 Get Intel Filters — Manageability Only Command (Intel Command 0x03, Filter Parameter 0x0F)

This command retrieves the *MNGONLY* register. The *MNGONLY* register controls whether pass-through packets destined to the BMC are also be forwarded to the Host OS.

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...21	0x03	0x0F		

10.6.3.6.4 Get Intel Filters — Manageability Only Response (Intel Command 0x03, Filter Parameter 0x0F)

The *MNGONLY* register structure is described in Table 10-4.

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...27	0x03	0x0F	Manageability Only (3-2)	
28...29	Manageability Only(1-0)			

10.6.3.6.5 Get Intel Filters — Flex Filter 0 Enable Mask and Length Command (Intel Command 0x03, Filter Parameter 0x10)

The following command retrieves the Intel flex filters mask and length. See Section 10.3.3.5 for details of the values returned by this command.

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...21	0x03	0x10		

10.6.3.6.6 Get Intel Filters — Flex Filter 0 Enable Mask and Length Response (Intel Command 0x03, Filter Parameter 0x10)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...27	0x03	0x10	Mask Byte 1	Mask Byte 2
28...31	...	...	...	...
32...35	...	...	...	...
36...39	...	...	...	...
40...43	...	Mask Byte 16	Reserved	Reserved
44	Flexible Filter Length			

10.6.3.6.7 Get Intel Filters — Flex Filter 0 Data Command (Intel Command 0x03, Filter Parameter 0x11)

The following command retrieves the Intel flex filters data.

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...22	0x03	0x11	Filter Data Group 0...4	

The Filter Data Group parameter defines which bytes of the Flex filter are returned by this command:

Table 10-39 Filter Data Group

Code	Bytes Returned
0x0	bytes 0-29
0x1	bytes 30-59
0x2	bytes 60-89
0x3	bytes 90-119
0x4	bytes 120-127

10.6.3.6.8 Get Intel Filters — Flex Filter 0 Data Response (Intel Command 0x03, Filter Parameter 0x11)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	

20...23	Manufacturer ID (Intel 0x157)			
24...	0x03	0x11	Filter Group Number	Filter Data 1
	...	Filter Data N		

10.6.3.6.9 Get Intel Filters — Packet Addition Decision Filter Command (Intel Command 0x03, Filter Parameter 0x61)

Note: This command is kept and supported for legacy reasons, however it is recommended to use the Get Intel Filters - Packet Addition Extended Decision Filter Command (Intel Command 0x03, Filter parameter 0x68 - Section 10.6.3.6.19) instead

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...21	0x03	0x61	Decision filter index	

Filter index range: 0x0...0x4.

10.6.3.6.10 Get Intel Filters — Packet Addition Decision Filter Response (Intel Command 0x03, Filter Parameter 0x61)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...27	0x03	0x61	Decision Filter (MSB)	
28...29	Decision Filter (LSB)			

The Decision filter structure returned is described in Table 10-37.

If the Decision filter index is bigger than 4, a command failed Response Code is returned with no reason.

10.6.3.6.11 Get Intel Filters — Flex TCP/UDP Port Filter Command (Intel Command 0x03, Filter Parameter 0x63)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...22	0x03	0x63	TCP/UDP Filter Index	

Filter index range: 0x0...0x2.

10.6.3.6.12 Get Intel Filters — Flex TCP/UDP Port Filter Response (Intel Command 0x03, Filter Parameter 0x63)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...27	0x03	0x63	TCP/UDP Filter Index	TCP/UDP Port (1)
28	TCP/UDP Port (0)			

If the TCP/UDP Filter Index is bigger than 2, a command failed Response Code is returned with no reason

10.6.3.6.13 Get Intel Filters — IPv4 Filter Command (Intel Command 0x03, Filter Parameter 0x64)

	Bits			
Bytes	31...24	23...16	15...08	07...00
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...22	0x03	0x64	IPv4 Filter Index	

Note: The filters index range can vary according to the IPv4/IPv6 mode setting in the Filters Control command.

IPv4 Mode: Filter index range: 0x0...0x3.

IPv6 Mode: This command should not be used in IPv6 mode.

10.6.3.6.14 Get Intel Filters — IPv4 Filter Response (Intel Command 0x03, Filter Parameter 0x64)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...27	0x03	0x64	IPv4 Filter Index	IPv4 Address (3)
28...29	IPv4 Address (2-0)			

10.6.3.6.15 Get Intel Filters — IPv6 Filter Command (Intel Command 0x03, Filter Parameter 0x65)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...22	0x03	0x65	IPv6 Filter Index	

Note: The filters index range can vary according to the IPv4/IPv6 mode setting in the Filters Control command

IPv4 Mode: Filter index range: 0x0...0x2.

IPv6 Mode: Filter index range: 0x0...0x3.

10.6.3.6.16 Get Intel Filters — IPv6 Filter Response (Intel Command 0x03, Filter parameter 0x65)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...27	0x03	0x65	IPv6 Filter Index	IPv6 Address (MSB, Byte 16)
28...31	...	...	...	...
32...35	...	...	...	...
36...39	...	...	...	...
40...42	...	...	IPv6 Address (LSB, Byte 0)	

If the IPv6 Filter Index is bigger than 3, a command failed Response Code is returned with no reason.

10.6.3.6.17 Get Intel Filters - EtherType Filter Command (Intel Command 0x03, Filter parameter 0x67)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...22	0x03	0x67	EtherType Filter Index	

Valid indices: 0...3

10.6.3.6.18 Get Intel Filters - EtherType Filter Response (Intel Command 0x03, Filter parameter 0x67)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...27	0x03	0x67	EtherType Filter Index	EtherType Filter MSB
28...30	..	..	EtherType Filter LSB	

If the Ethertype filter Index is larger than 3, a command failed Response Code is returned with no reason.

10.6.3.6.19 Get Intel Filters – Packet Addition Extended Decision Filter Command (Intel Command 0x03, Filter parameter 0x68)

This command allows the BMC to retrieve the Extended Decision Filter.

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...22	0x03	0x68	Extended Decision Filter Index	

10.6.3.6.20 Get Intel Filters – Packet Addition Extended Decision Filter Response (Intel Command 0x03, Filter parameter 0x68)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...27	0x03	0x68	Decision Filter Index	Decision Filter 1 MSB
28...31	..	..	Decision Filter 1 LSB	Decision Filter 0 MSB
32...34	..	..	Decision Filter 0 LSB	

Where Decision Filter 0 & Decision Filter 1 have the structure as detailed in the respective "Set" commands.

If the Extended Decision Filter Index is bigger than 4, a command failed Response Code is returned with no reason.

10.6.3.7 Set Intel Packet Reduction Filters Formats

Note: The non extended commands (Section 10.6.3.7.3 to Section 10.6.3.8.8) are kept and supported for legacy reasons, however it is recommended to use the extended commands (Section 10.6.3.7.9 to Section 10.6.3.7.14) instead

10.6.3.7.1 Set Intel Packet Reduction Filters Command (Intel Command 0x04)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...23	0x04	Packet Reduction Index	Packet Reduction Data...	

10.6.3.7.2 Set Intel Packet Reduction Filters Response (Intel Command 0x04)

	Bits			
Bytes	**31...24**	**23...16**	**15...08**	**07...00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...	0x04	Packet Reduction Index	Optional Return Data	

The *Packet Reduction* Data field has the following structure:

Table 10-40 Packet Reduction field description

Bit #	Name	Description
12:0	Reserved	Reserved
16:13	IPv4 Address (AND)	If set, packets must match IPv4 filter 0 to 3 respectively
20:17	IPv6 Address (AND)	If set, packets must match IPv6 filter 0 to 3 respectively
27:21	Reserved	Reserved
28	ARP Response (OR)	If set, packets can pass if match the ARP response filter or a different OR filter.
29	Reserved	Reserved
30	Port 0x298	If set, packets can pass if match a fixed TCP/UDP port 0x298 filter.
31	Port 0x26F	If set, packets can pass if match a fixed TCP/UDP port 0x26F filter.

Table 10-41 Extended Packet Reduction field description

Bit #	Name	Description
3:0	Ethertype 0 -3 (AND)	If set, packets must match the Ethertype filter 0 to 3 respectively.
7:4	Reserved	Reserved
11:8	Ethertype 0-3 (OR)	If set, packets can pass if match the Ethertype filter 0 to 3 respectively.
15:12	Reserved	Reserved
16	Flex port 0 (OR)	If set, packets can pass if match the TCP/UDP Port filter 0
17	Flex port 1 (OR)	If set, packets can pass if match the TCP/UDP Port filter 1
18	Flex port 2 (OR)	If set, packets can pass if match the TCP/UDP Port filter 2
23:19	Reserved	
24	Flex TCO (OR)	If set, packets can pass if match the Flex 128 TCO filter
31:25	Reserved	

The filtering is divided into two decisions:

- Bit 20:13 in Table 10-40 and Bits 3:2 in Table 10-41 works in an AND manner; it must be true in order for a packet to pass (if was set).

Bits 28 in Table 10-40 and Bits 24:10 in Table 10-41 work in an OR manner; at least one of them must be true for a packet to pass (if any were set).

10.6.3.7.3 Set Unicast Packet Reduction Command (Intel Command 0x04, Reduction Filter Index 0x00)

This command causes the NC to filter packets that have passed due to the unicast filter (MAC address filters, as specified in the DMTF NC-SI). Note that unicast filtering might be affected by other filters, as specified in the DMTF NC-SI.

The filtering of these packets are done such that the BMC might add a logical condition that a packet must match, or it must be discarded.

Note: Packets that might have been blocked can still pass due to other decision filters.

In order to disable unicast packet reduction, the BMC should set all reduction filters to 0b. Following such a setting the NC must forward, to the BMC, all packets that have passed the unicast filters (MAC address filtering) as specified in the DMTF NC-SI.

The *Unicast Packet Reduction* field structure is described in Table 10-40.

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...23	0x04	0x00	Unicast Packet Reduction (3-2)	
24...25	Unicast Packet Reduction (1-0)			

10.6.3.7.4 Set Unicast Packet Reduction Response (Intel Command 0x04, Reduction Filter Index 0x00)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...25	0x04	0x00		

10.6.3.7.5 Set Multicast Packet Reduction Command (Intel Command 0x04, Reduction Filter Index 0x01)

This command causes the NC to filter packets that have passed due to the multicast filter (MAC address filters, as specified in the DMTF NC-SI).

The filtering of these packets are done such that the BMC might add a logical condition that a packet must match, or it must be discarded.

Note: Packets that might have been blocked can still pass due to other decision filters.

In order to disable multicast packet reduction, the BMC should set all reduction filters to 0b. Following such a setting, the NC must forward, to the BMC, all packets that have passed the multicast filters (global multicast filtering) as specified in the DMTF NC-SI.

The *Multicast Packet Reduction* field structure is described in Table 10-40.

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...23	0x04	0x01	Multicast Packet Reduction (3-2)	
24...25	Multicast Packet Reduction (1-0)			

10.6.3.7.6 Set Multicast Packet Reduction Response (Intel Command 0x04, Reduction Filter Index 0x01)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...25	0x04	0x01		

10.6.3.7.7 Set Broadcast Packet Reduction Command (Intel Command 0x04, Reduction Filter Index 0x02)

This command causes the NC to filter packets that have passed due to the broadcast filter (MAC address filters, as specified in the DMTF NC-SI).

The filtering of these packets are done such that the BMC might add a logical condition that a packet must match, or it must be discarded.

Note: Packets that might have been blocked can still pass due to other decision filters.

In order to disable broadcast packet reduction, the BMC should set all reduction filters to 0b. Following such a setting, the NC must forward, to the BMC, all packets that have passed the broadcast filters as specified in the DMTF NC-SI.

The *Broadcast Packet Reduction* field structure is described in Table 10-40.

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...23	0x04	0x02	Broadcast Packet Reduction (3-2)	
24...25	Broadcast Packet Reduction (1-0)			

10.6.3.7.8 Set Broadcast Packet Reduction Response (Intel Command 0x08)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...25	0x04	0x02		

10.6.3.7.9 Set Unicast Extended Packet Reduction Command (Intel Command 0x04, Reduction Filter Index 0x10)

In "Set Intel Reduction Filters" add another parameter "Unicast Extended Packet Reduction (Intel Command 0x04, Filter parameter 0x10)" such that the byte count is 0xE. The command shall have the following format:

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			

20..23	0x04	0x10	Extended Unicast Reduction Filter MSB	..
24..27	..	Extended Unicast Reduction Filter LSB	Unicast Reduction Filter MSB	..
28..29	..	Unicast Reduction Filter LSB		

The command shall overwrite any previously stored value.

Note: See Table 10-40 and Table 10-41 for description of the Unicast Extended Packet Reduction format.

10.6.3.7.10 Set Unicast Extended Packet Reduction Response (Intel Command 0x04, Reduction Filter Index 0x10)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..25	0x04	0x10		

10.6.3.7.11 Set Multicast Extended Packet Reduction Command (Intel Command 0x04, Reduction Filter Index 0x11)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x04	0x11	Extended Multicast Reduction Filter MSB	..
24..27	..	Extended Multicast Reduction Filter LSB	Multicast Reduction Filter MSB	..
28..29	..	Multicast Reduction Filter LSB		

Note: See Table 10-40 and Table 10-41 for description of the Multicast Extended Packet Reduction format.

The command shall overwrite any previously stored value.

10.6.3.7.12 Set Multicast Extended Packet Reduction Response (Intel Command 0x04, Reduction Filter Index 0x11)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..25	0x04	0x11		

10.6.3.7.13 Set Broadcast Extended Packet Reduction Command (Intel Command 0x04, Reduction Filter Index 0x12)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00..15	NC-SI Header			
16..19	Manufacturer ID (Intel 0x157)			
20..23	0x04	0x12	Extended Broadcast Reduction Filter MSB	..
24..27	..	Extended Broadcast Reduction Filter LSB	Broadcast Reduction Filter MSB	..
28..29	..	Broadcast Reduction Filter LSB		

Note: See Table 10-40 and Table 10-41 for description of the Broadcast Extended Packet Reduction format.

The command shall overwrite any previously stored value.

10.6.3.7.14 Set Broadcast Extended Packet Reduction Response (Intel Command 0x04, Reduction Filter Index 0x12)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00..15	NC-SI Header			
16..19	Response Code		Reason Code	
20..23	Manufacturer ID (Intel 0x157)			
24..25	0x04	0x12		

10.6.3.8 Get Intel Packet Reduction Filters Formats

Note: The non extended commands (Section 10.6.3.8.3 to Section 10.6.3.8.8) are kept and supported for legacy reasons, however it is recommended to use the extended commands (Section 10.6.3.8.9 to Section 10.6.3.8.14) instead

10.6.3.8.1 Get Intel Packet Reduction Filters Command (Intel Command 0x05)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...21	0x05	Reduction Filter Index		

10.6.3.8.2 Get Intel Packet Reduction Filters Response (Intel Command 0x05)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...	0x05	Reduction Filter Index	Return Data	

Note: See Table 10-40 and Table 10-41 for description of the Return Data format.

10.6.3.8.3 Get Unicast Packet Reduction Command (Intel Command 0x05, Reduction Filter Index 0x00)

This command causes the NC to disable any packet reductions for unicast address filtering.

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...21	0x05	0x00		

10.6.3.8.4 Get Unicast Packet Reduction Response (Intel Command 0x05, Reduction Filter Index 0x00)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	

20...23	Manufacturer ID (Intel 0x157)		
24...27	0x05	0x00	Unicast Packet Reduction (MSB)
28...29	Unicast Packet Reduction (LSB)		

10.6.3.8.5 Get Multicast Packet Reduction Command (Intel Command 0x05, Reduction Filter Index 0x01)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...21	0x05	0x01		

10.6.3.8.6 Get Multicast Packet Reduction Response (Intel Command 0x05, Reduction Filter Index 0x01)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...27	0x05	0x01	Multicast Packet Reduction (MSB)	
28...29	Multicast Packet Reduction (LSB)			

10.6.3.8.7 Get Broadcast Packet Reduction Command (Intel Command 0x05, Reduction Filter Index 0x02)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...21	0x05	0x02		

10.6.3.8.8 Get Broadcast Packet Reduction Response (Intel Command 0x05, Reduction Filter Index 0x02)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...27	0x05	0x02	Broadcast Packet Reduction (MSB)	
28...29	Broadcast Packet Reduction (LSB)			

10.6.3.8.9 Get Unicast Extended Packet Reduction Command (Intel Command 0x05, Reduction Filter Index 0x10)

\

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...21	0x05	0x10		

10.6.3.8.10 Get Unicast Extended Packet Reduction Response (Intel Command 0x05, Reduction Filter Index 0x10)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...27	0x05	0x10	Extended Unicast Packet Reduction (MSB)	
28...31	Extended Unicast Packet Reduction (LSB)		Unicast Packet Reduction (MSB)	
32...33	Unicast Packet Reduction (LSB)			

10.6.3.8.11 Get Multicast Extended Packet Reduction Command (Intel Command 0x05, Reduction Filter Index 0x11)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...21	0x05	0x11		

10.6.3.8.12 Get Multicast Extended Packet Reduction Response (Intel Command 0x05, Reduction Filter Index 0x11)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...27	0x05	0x11	Extended Multicast Packet Reduction (MSB)	
28...31	Extended Multicast Packet Reduction (LSB)		Multicast Packet Reduction (MSB)	
32...33	Multicast Packet Reduction (LSB)			

10.6.3.8.13 Get Broadcast Extended Packet Reduction Command (Intel Command 0x05, Reduction Filter Index 0x12)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...21	0x05	0x12		

10.6.3.8.14 Get Broadcast Extended Packet Reduction Response (Intel Command 0x05, Reduction Filter Index 0x12)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...27	0x05	0x12	Extended Broadcast Packet Reduction (MSB)	
28...31	Extended Broadcast Packet Reduction (LSB)		Broadcast Packet Reduction (MSB)	
32...33	Broadcast Packet Reduction (LSB)			

10.6.3.9 System MAC Address

10.6.3.9.1 Get System MAC Address Command (Intel Command 0x06)

In order to support a system configuration that requires the NC to hold the MAC address for the BMC (such as shared MAC address mode), the following command is provided to enable the BMC to query the NC for a valid MAC address.

The NC must return the system MAC addresses. The BMC should use the returned MAC addressing as a shared MAC address by setting it using the Set MAC Address command as defined in NC-SI 1.0.

It is also recommended that the BMC use packet reduction and Manageability-to-Host command to set the proper filtering method.

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20	0x06			

10.6.3.9.2 Get System MAC Address Response (Intel Command 0x06)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...27	0x06	MAC Address		
28...30	MAC Address			

10.6.3.10 Set Intel Management Control Formats

10.6.3.10.1 Set Intel Management Control Command (Intel Command 0x20)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...22	0x20	0x00	Intel Management Control 1	

Where Intel Management Control 1 is as follows:

Bit #	Default value	Description
0	0b	Enable Critical Session Mode (Keep PHY Link Up and Veto Bit) 0b — Disabled 1b — Enabled When critical session mode is enabled, the following behaviors are disabled: • The PHY is not reset on PE_RST# and PCIe resets (in-band and link drop). Other reset events are not affected — Internal_Power_On_Reset, device disable, Force TCO, and PHY reset by software. • The PHY does not change its power state. As a result link speed does not change. • The device does not initiate configuration of the PHY to avoid losing link.
1…7	0x0	Reserved

10.6.3.10.2 Set Intel Management Control Response (Intel Command 0x20)

	Bits			
Bytes	31:24	23:16	15:08	07:00
00…15	NC-SI Header			
16…19	Response Code		Reason Code	
20…23	Manufacturer ID (Intel 0x157)			
24…25	0x20	0x00		

10.6.3.11 Get Intel Management Control Formats

10.6.3.11.1 Get Intel Management Control Command (Intel Command 0x21)

	Bits			
Bytes	31:24	23:16	15:08	07:00
00…15	NC-SI Header			
16…19	Manufacturer ID (Intel 0x157)			
20…21	0x21	0x00		

Where Intel Management Control 1 is as described in Section 10.6.3.10.2.

10.6.3.11.2 Get Intel Management Control Response (Intel Command 0x21)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...26	0x21	0x00	Intel Management Control 1	

10.6.3.12 TCO Reset

Depending on the bit set in the TCO mode field this command will cause the I350 to perform either:

1. TCO Reset, if Force TCO reset is enabled in the EEPROM (see Section 6.3.7). The Force TCO reset will clear the data-path (RX/TX) of the I350 to enable the BMC to transmit/receive packets through the I350.
   - If the BMC has detected that the OS is hung and has blocked the RX/TX path The Force TCO reset will clear the data-path (RX/TX) of the Network Controller to enable the BMC to transmit/ receive packets through the Network Controller.
   - When this command is issued to a channel in a package, it applies only to the specific channel.
   - After successfully performing the command the Network Controller will consider Force TCO command as an indication that the OS is hung and will clear the DRV_LOAD flag (disable the driver). If TCO reset is disabled in EEPROM the I350 clears the *CTRL_EXT.DRV_LOAD* bit but does not reset the data-path and notifies BMC on successful completion.
   - Following TCO reset management sets *MANC.TCO_RESET* to 1.
2. TCO isolate, if TCO isolate is enabled in the EEPROM (See Section 6.3.7.3). The TCO Isolate command will disable PCIe write operations to the LAN port.
   - If TCO Isolate is disabled in EEPROM the I350 does not execute the command but sends a response to the BMC with successful completion.
   - Following TCO Isolate management sets *MANC.TCO_Isolate* to 1.
3. Firmware Reset. This command will cause re-initialization of all the manageability functions and re-load of manageability related EEPROM words (e.g. Firmware patch code).
   - When the BMC has loaded new management related EEPROM image (e.g. Firmware patch) the Firmware Reset command will load management related EEPROM information without need to power down the system.
   - This command is issued to the package and affects all channels. After the Firmware reset the FW Semaphore register (FWSM) is re-initialized.

Notes: Force TCO reset and TCO Isolate will affect only the channel (port) that the command was issued to.

Following firmware reset, BMC will need to re-initialize all ports.

10.6.3.12.1 Perform Intel TCO Reset Command (Intel Command 0x22)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20	0x22	TCO Mode		

Where TCO Mode is:

Field	Bit(s)	Description
DO_TCO_RST	0	Perform TCO Reset. 0b: Do nothing. 1b: Perform TCO reset.
DO_TCO_ISOLATE[1]	1	Do TCO Isolate 0b = Enable PCIe write access to LAN port. 1b = Isolate Host PCIe write operation to the port Note: Should be used for debug only. Note: When Isolate is set, the OS2BMC flow is disabled also.
RESET_MGMT	2	Reset manageability; re-load manageability EEPROM words. 0b = Do nothing 1b = Issue firmware reset to manageability Setting this bit generates a one-time firmware reset. Following the reset, management related data from EEPROM is loaded.
Reserved	7:3	Reserved (set to 0x00).

Note: For compatibility, the TCO reset command without the TCO Mode parameter is accepted (TCO reset is performed).

1. TCO Isolate Host Write operation enabled in EEPROM.

10.6.3.12.2 Perform Intel TCO Reset Response (Intel Command 0x22)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...26	0x22			

10.6.3.13 Checksum Offloading

This command enables the checksum offloading filters in the NC.

When enabled, these filters block any packets that did not pass IP, UDP and TCP checksums from being forwarded to the BMC.

10.6.3.13.1 Enable Checksum Offloading Command (Intel Command 0x23)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20	0x23			

10.6.3.13.2 Enable Checksum Offloading Response (Intel Command 0x23)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...26	0x23			

10.6.3.13.3 Disable Checksum Offloading Command (Intel Command 0x24)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20	0x24			

10.6.3.13.4 Disable Checksum Offloading Response (Intel Command 0x24)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...26	0x24			

10.6.3.14 OS 2 BMC Configuration

These commands control enabling of the OS 2 BMC flow.

Note: If OS2BMC is disabled these commands will fail with response code 0x0001 (command failed) and reason code 0x0000.

10.6.3.14.1 EnableOS2BMC Flow Command (Intel Command 0x40, Index 0x1)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...21	0x40	0x01		

10.6.3.14.2 EnableOS2BMC Flow Response (Intel Command 0x40, Index 0x1)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...25	0x40	0x01		

10.6.3.14.3 Enable Network to BMC Flow Command (Intel Command 0x40, Index 0x2)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...21	0x40	0x02		

10.6.3.14.4 Enable Network to BMC Flow Response (Intel Command 0x40, Index 0x2)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...25	0x40	0x02		

10.6.3.14.5 Enable both Host and Network to BMC flows Command (Intel Command 0x40, Index 0x3)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...21	0x40	0x03		

10.6.3.14.6 Enable both Host and Network to BMC Flows Response (Intel Command 0x40, Index 0x3)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...25	0x40	0x03		

10.6.3.14.7 Get OS2BMC parameters Command (Intel Command 0x41)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20	0x41			

10.6.3.14.8 Get OS2BMC parameters Response (Intel Command 0x41)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...27	0x41	Status		

Where the Status byte partition is as follow:

Table 10-42 Status byte description

Bits	Content
1:0	Reserved
2	Network to BMC status 0 = network 2 BMC flow is disabled 1 = network 2 BMC flow is enabled.
3	OS2BMC status 0 = OS 2 BMC flow is disabled 1 = OS 2 BMC flow is enabled.
7:4	Reserved.

10.6.3.15 Inventory and Update System Parameters Commands

10.6.3.15.1 Get Controller Information Command (Intel Command 0x48, Index 0x1)

This command gather the controller identification information and return it back to the BMC.

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...23	0x48	0x1		

10.6.3.15.2 Get Controller information Response (Intel Command 0x48, Index 0x1)

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...27	0x48	0x01	Reserved	Number of Inventory entries

	Bits			
Bytes	**31:24**	**23:16**	**15:08**	**07:00**
28...31	FW version Item 1ID	FW version Item 1 length	FW version Item 1 Data	
...	....			
...	FW version Item 2 ID	FW version Item 2 length	FW version Item 2 Data	
...	....			
...	FW version Item n ID	FW version Item n length	FW version Item n Data	
...	....			

Where the possible inventory items are as described below. Note that not all the inventory items would be present in all the implementations of this command.

Table 10-43 Controller Information Items

ID	Length (in bytes)	Data	Notes	Available without driver?
0x0D	2	NC-SI FW version	Major.Minor	Yes
0x0E	2	PXE FW version	MajorVersion.MinorVersion.Build.	Yes
0x0F	2	iSCSI FW version	MajorVersion.MinorVersion.Build.	Yes
0x10	2	uEFI FW version	MajorVersion.MinorVersion.Build.	Yes
0x11	2	Loader FW Patch Version	Major.Minor	Yes
0x12	2	Application FW Patch Version	Should reflect the version of the patch currently used by the FW and not the FW version of the patch stored in the NVM.	Yes

10.6.3.16 Thermal Sensor Commands

Note: Most of the actions exposed here are available only when working with an internal PHY. There is no support for controlling of external PHY behaviors according to thermal sensor inputs.

Note: Thermal Sensor configuration can be done only through NC-SI channel 0.

10.6.3.16.1 Get Thermal Sensor Commands (Intel Command 0x4C)

10.6.3.16.1.1 Common Tables

The following tables are used in the various commands:

Table 10-44 Actions word definition

Bit	Action	Notes
0	Measure Only[1]	Activate the thermal sensor, but no automatic action.
1	Notify BMC[1]	
2	Power off PHY	
3	Power on PHY	
4	Restore Speed	Restore regular speed. When used as "Active Action", should be set if there are no limitations on the speed setting.
5	Set speed to 10 Mbps Max	These actions set a maximum on the speed and do not force a specific speed. The Set Link command should be used to set a specific link speed.
6	Set speed to 100 Mbps Max	
7	Set speed to 1 Gbps Max	
8	Set speed to 10 Gbps Max	
9	Indicate on SDP (set)[1]	
10	Indicate on SDP (clear)[1]	
11	HW autonomous algorithm	
12	Cancel all active actions[1]	
13	Rearm Event[1]	
31:14	Reserved	

1. These bits are not relevant when reporting active actions. Values should be ignored.

Table 10-45 Unit Types word definition

Value	Unit Types	Notes
0x0	Generic Number	No specific indication of measured value
0x1	Celsius	Temperature
0x2	Volts	
0x3	rpm	Speed
0x4-0xFF	Reserved	Reserved.

10.6.3.16.1.2 Threshold and Hysteresis

Each threshold event includes a direction. For example, a thermal event with a threshold of 100 C and an "up" direction is defined as the temperature crossing from less than 100 C to more than 100 C.

For each threshold an hysteresis may be defined. This hysteresis direction is opposite to the threshold direction. So if. for the previous example, an hysteresis of 10 C is defined, it will be activated when the temperature crosses from more than 90 C to less than 90 C.

The following figure describes the thresholds and hysteresis modes and the actions activated for each of them.

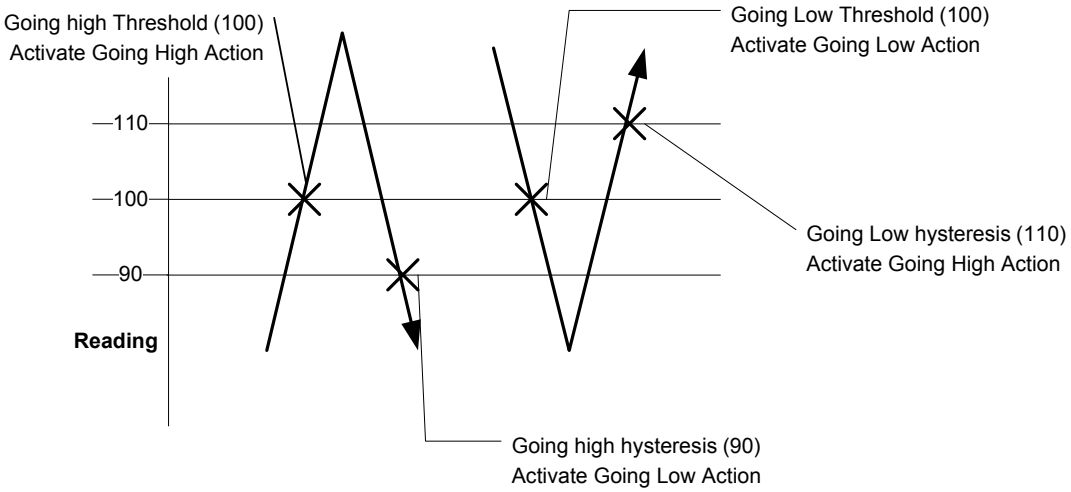


Figure 10-9 Thresholds, hysteresis and actions.

10.6.3.16.2 Get Thermal Sensor Capabilities Command (Intel Command 0x4C, Index 0x0)

This command requests the thermal sensor capabilities supported by this device.

	Bits			
Bytes	**31...24**	**23...16**	**15...08**	**07...00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...21	0x4C	0x00		

10.6.3.16.3 Get Thermal Sensor Capabilities Response (Intel Command 0x4c, Index 0x0)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...27	0x4C	0x0	Version	Unit Types
28...31	Number of Thresholds		Accuracy	Max hysteresis
32...35	M	B	K1	K2

36...39	Available actions - "High going" thresholds		
40..43	Available actions - "Low going" thresholds		
44..47	Available actions - "Immediate"		
48	TJunction Max		

- Version should always be 1.
- Unit Types = 1 - we report temperature in Celsius.
- Accuracy describes the accuracy of the reported measurements as follow:
  - 7:4: Max deviation of actual value above measurement in "unit types" = 5 C
  - 3:0: Max deviation of actual value below measurement in "unit types" = 5 C
- Max hysteresis - defines the max hysteresis value allowed in the implementation = 0xF - an hysteresis of up to 15 degrees is supported
- Number of thresholds describes the number of up and down thresholds as follow:
  - 15:12: Reserved
  - 11:8: Max Number of mixed thresholds = 0.
  - 7:4: Max number of up thresholds = 3
  - 3:0: Max number of down thresholds = 0
- Valid Actions "High going" thresholds - describes the actions that can be activated by the device as described in Table 10-44 when an high going threshold is crossed. The I350 supports Do Nothing, Notify BMC, Power off PHY, Set speed to 10 Mbps, Set speed to 100 Mbps, Indicate on SDP (set), and Indicate on SDP (clear).
- Valid Actions "Low going" thresholds - describes the actions that can be activated by the device as described in Table 10-44 when an low going threshold is crossed. The I350 supports Notify BMC, Power on PHY, Restore Speed, Indicate on SDP (set), and Indicate on SDP (clear).
- Valid Actions "Immediate" - describes the actions that can be activated by the device as described in Table 10-44 using a "Perform Thermal Sensor Action" command. The I350 supports Measure only, Reset or Speed, Cancel all active actions and reset thermal sensor.
- M = 1; B = 0, K1 = K2 = 0 - The assumption is that the thermal sensor in the I350 is the readable value.

Note: This formula is compliant with the definition of section 36.3 "Sensor Reading Conversion Formula" in IPMI 2.0

- TjunctionMax - The maximal junction temperature supported (125 C).

10.6.3.16.4 Get Thermal Sensor Configuration Command (Intel Command 0x4C, Index 0x1)

This command requests the thermal sensor configuration for threshold "Index".

	Bits			
Bytes	**31...24**	**23...16**	**15...08**	**07...00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...22	0x4C	0x01	Index	

Note: If Index points to a non valid threshold as described in the Get Thermal Sensor Capabilities Response, the command fails with an Invalid Parameter reason.

10.6.3.16.5 Get Thermal Sensor Configuration Response (Intel Command 0x4c, Index 0x1)

	Bits			
Bytes	**31..24**	**23..16**	**15..08**	**07..00**
00...15	NC-SI Header			
16...19	Response Code		Reason Code	
20...23	Manufacturer ID (Intel 0x157)			
24...27	0x4C	0x1	Index	Threshold (1)
28...31	Threshold (0)	Actions "Going High" (3-1)		
32...35	Actions "Going High" (0)	Actions "Going Low" (3-1)		
36...38	Actions "Going Low" (0)	Direction	Hysteresis	

The threshold and the Hysteresis are measured in "unit types".

The "Actions Going High" field describes the actions to activate upon crossing of the threshold for "Going High" thresholds or when crossing the hysteresis for "Going Low" thresholds according to Table 10-44.

The "Actions Going Low" field describes the actions to activate upon crossing of the threshold for "Going Low" thresholds or when crossing the hysteresis for "Going High" thresholds according to Table 10-44.

Direction is encoded as follow:

- 0 = High Going
- 1 = Low Going

10.6.3.16.6 Get Thermal Sensor Status Command (Intel Command 0x4C, Index 0x2)

This command requests the current status of the Thermal sensor.

	Bits			
Bytes	**31...24**	**23...16**	**15...08**	**07...00**
00...15	NC-SI Header			
16...19	Manufacturer ID (Intel 0x157)			
20...21	0x4C	0x02		

10.6.3.16.7 Get Thermal Sensor Status Response (Intel Command 0x4c, Index 0x2)

	Bits			
Bytes	**31…24**	**23…16**	**15…08**	**07…00**
00…15	NC-SI Header			
16…19	Response Code		Reason Code	
20…23	Manufacturer ID (Intel 0x157)			
24…27	0x4C	0x2	Measured Value	
28…31	Active Actions			
32…33	Threshold cross events			

Where "Threshold cross events" is a bitmap that describes which events where crossed since the last read of the status or since the activation of the thermal sensor (the latest of the two).

10.6.3.16.8 Set Thermal Sensor Commands (Intel Command, Index 0x4D)

10.6.3.16.8.1 Set Thermal Sensor Configuration Command (Intel Command 0x4D, Index 0x1)

This command sets the thermal sensor configuration for threshold "Index"

The threshold and the Hysteresis are measured in "unit types".

The "Actions Going High" field describes the actions to activate upon crossing of the threshold for "Going High" thresholds or when crossing the hysteresis for "Going Low" thresholds according to Table 10-44.

The "Actions Going Low" field describes the actions to activate upon crossing of the threshold for "Going Low" thresholds or when crossing the hysteresis for "Going High" thresholds according to Table 10-44.

Direction is encoded as follow:

- 0 = High Going
- 1 = Low Going

	Bits			
Bytes	**31…24**	**23…16**	**15…08**	**07…00**
00…15	NC-SI Header			
16…19	Manufacturer ID (Intel 0x157)			
20…23	0x4D	0x01	Index	Direction
24…27	Threshold		Actions "Going High" (MSB)	
28…31	Actions "Going High" (LSB)		Actions "Going Low" (MSB)	
32…34	Actions "Going Low" (LSB)		Hysteresis	

Note: If Index and Direction points to a non valid threshold as described in the Get Thermal Sensor Capabilities Response, the command fails with an Invalid Parameter reason.

If the requested action is not supported, the command fails with an Invalid Parameter reason.

Actions set can not be contradictory - so for a given set of actions, the following combinations are invalid and will result in a command fails with an Invalid Parameter reason:

- Indicate on SDP (set) and Indicate on SDP (clear) both set,
- Power off PHY and Power up PHY both set.
- Increase and Reduce Speed both set.
- Both a maximal speed and a PHY power off action are requested.
- More than one maximal speed is requested.
- Do Nothing and another option are requested.
- HW independent algorithm and another option are requested.

10.6.3.16.8.2 Set Thermal Sensor Configuration Response (Intel Command 0x4c, Index 0x1)

	Bits			
Bytes	**31…24**	**23…16**	**15…08**	**07…00**
00…15	NC-SI Header			
16…19	Response Code		Reason Code	
20…23	Manufacturer ID (Intel 0x157)			
24…25	0x4D	0x1		

10.6.3.16.8.3 Set Thermal Sensor Action Command (Intel Command 0x4D, Index 0x2)

This command executes actions immediately.

The “Actions” field describes the actions to activate according to Table 10-44.

	Bits			
Bytes	**31…24**	**23…16**	**15…08**	**07…00**
00…15	NC-SI Header			
16…19	Manufacturer ID (Intel 0x157)			
20…23	0x4D	0x02	Actions (MSB)	
24…25	Actions (LSB)			

Note: If one of the requested actions is not supported, the command fails with an Invalid Parameter reason.

Note: Actions set can not be contradictory - so the following combinations are invalid and will result in a command fails with an Invalid Parameter reason:

— Indicate on SDP (set) and Indicate on SDP (clear) both set,

— Power Down PHY and Power up PHY both set.

— Restore and one of the Reduce Speed both set.

— Both a maximal speed and a PHY power down action are requested.

— More than one maximal speed is requested.

— Do Nothing and another option are requested.

— HW independent algorithm and another option are requested.

10.6.3.16.8.4 Set Thermal Sensor Configuration Response (Intel Command 0x4c, Index 0x1)

	Bits			
Bytes	**31…24**	**23…16**	**15…08**	**07…00**
00…15	NC-SI Header			
16…19	Response Code		Reason Code	
20…23	Manufacturer ID (Intel 0x157)			
24…25	0x4D	0x2		

10.6.3.16.9 Thermal Sensor AEN (Intel AEN 0x81)

The following is the AEN that may be sent by the NC following a Thermal Sensor event.

This AEN must be enabled using the NC-SI "AEN Enable" command, using bit 17 (0x20000) of the AEN Enable mask.

	Bits			
Bytes	**31…24**	**23…16**	**15…08**	**07…00**
00…15	NC-SI AEN Header			
20…23	Reserved			0x81
24…27	Measured Value		Active Actions (3-2)	
28…31	Active Actions (1-0)		Threshold cross events	

Where "Threshold cross events" is a bitmap that describes which events where crossed since the last read of the status, AEN emission or since the activation of the thermal sensor (the latest of the two).

10.6.4 Basic NC-SI Workflows

10.6.4.1 Package States

A NC package can be in one of the following two states:

1. Selected — The package is allowed to use the NC-SI lines, meaning the NC package might send data to the BMC.
2. De-selected — The package is not allowed to use the NC-SI lines, meaning, the NC package cannot send data to the BMC.

The BMC must select no more than one NC package at any given time. Package selection can be accomplished in one of two methods:

1. Select Package command — This command explicitly selects the NC package.
2. Any other command targeted to a channel in the package also implicitly selects that NC package.

Package de-select can be accomplished only by issuing the De-Select Package command. The BMC should always issue the Select Package command as the first command to the package before issuing channel-specific commands. For further details on package selection, refer to the NC-SI specification.

10.6.4.2 Channel States

A NC channel can be in one of the following states:

1. Initial State — The channel only accepts the Clear Initial State command (the package also accepts the Select Package and De-Select Package commands).
2. Active state — This is the normal operational mode. All commands are accepted.

For normal operation mode, the BMC should always send the Clear Initial State command as the first command to the channel.

10.6.4.3 Discovery

After interface power-up, the BMC should perform a discovery process to discover the NCs that are connected to it. This process should include an algorithm similar to the following:

1. For package_id=0x0 to MAX_PACKAGE_ID
   a. Issue Select Package command to package ID package_id
   b. If a response was received then

      For internal_channel_id = 0x0 to MAX_INTERNAL_CHANNEL_ID

      Issue a Clear Initial State command for package_id | internal_channel_id (the combination of package_id and internal_channel_id to create the channel ID).

      If a response was received then

      Consider internal_channel_id as a valid channel for the package_id package

      The BMC can now optionally discover channel capabilities and version ID for the channel

      Else (If not a response was not received, then issue a Clear Initial State command three times.

      Issue a De-Select Package command to the package (and continue to the next package).
   c. Else, if a response was not received, issue a Select Packet command three times.

10.6.4.4 Configurations

This section details different configurations that should be performed by the BMC.

The BMC should not consider any configuration valid unless the BMC has explicitly configured it after every reset (entry into the initial state). As a result, the BMC should re-configure everything at power-up and channel/package resets.

10.6.4.4.1 NC Capabilities Advertisement

NC-SI defines the Get Capabilities command. It is recommended that the BMC use this command and verify that the capabilities match its requirements before performing any configurations. For example, the BMC should verify that the NC supports a specific AEN before enabling it.

10.6.4.4.2 Receive Filtering

In order to receive traffic, the BMC must configure the NC with receive filtering rules. These rules are checked on every packet received on the LAN interface (such as from the network). Only if the rules matched, will the packet be forwarded to the BMC.

10.6.4.4.2.1 MAC Address Filtering

NC-SI defines three types of MAC address filters: unicast, multicast and broadcast. To be received (not dropped) a packet must match at least one of these filters. The BMC should set one MAC address using the Set MAC Address command and enable broadcast and global multicast filtering.

10.6.4.4.2.1.1 Unicast/Exact Match (Set MAC Address Command)

This filter filters on specific 48-bit MAC addresses. The BMC must configure this filter with a dedicated MAC address.

The NC might expose three types of unicast/exact match filters (such as MAC filters that match on the entire 48 bits of the MAC address): unicast, multicast and mixed. The I350 exposes two mixed filters, which might be used both for unicast and multicast filtering. The BMC should use one mixed filter for its MAC address.

Note: The MNGONLY bit matching the unicast filter (bit 5) is set by the first set MAC address command received from the BMC. It will not be cleared by further commands. If the MAC address is shared with the host and filter reductions are applied, the MNGONLY bit of the unicast filter should be cleared after each Set MAC address command using the *Set Intel Filters — Manageability Only Command* (Section 10.6.3.5.3).

Refer to NC-SI specification — Set MAC Address for further details.

10.6.4.4.2.1.2 Broadcast (Enable/Disable Broadcast Filter Command)

NC-SI defines a broadcast filtering mechanism which has the following states:

1. Enabled — All broadcast traffic is blocked (not forwarded) to the BMC, except for specific filters (such as ARP request, DHCP, and NetBIOS).
2. Disabled — All broadcast traffic is forwarded to the BMC, with no exceptions.

Refer to NC-SI specification Enable/Disable Broadcast Filter command.

10.6.4.4.2.1.3 Global Multicast (Enable/Disable Global Multicast Filter)

NC-SI defines a multicast filtering mechanism which has the following states:

1. Enabled — All multicast traffic is blocked (not forwarded) to the BMC.
2. Disabled — All multicast traffic is forwarded to the BMC, with no exceptions.

The recommended operational mode is Enabled, with specific filters set. Not all multicast filtering modes are necessarily supported. Refer to NC-SI specification Enable/Disable Global Multicast Filter command for further details.

10.6.4.4.3 VLAN

NC-SI defines the following VLAN work modes:

Mode	Command and Name	Descriptions
Disabled	Disable VLAN command	In this mode, no VLAN frames are received.
Enabled #1	Enable VLAN command with VLAN only	In this mode, only packets that matched a VLAN filter are forwarded to the BMC.
Enabled #2	Enable VLAN command with VLAN only + non-VLAN	In this mode, packets from mode 1 + non-VLAN packets are forwarded.
Enabled #3	Enable VLAN command with Any-VLAN + non-VLAN	In this mode, packets are forwarded regardless of their VLAN state.

Refer to NC-SI specification — Enable VLAN command for further details.

The I350 only supports modes #1 and #3. Recommendation:

1. Modes:
   a. If VLAN is not required — Use the disabled mode.
   b. If VLAN is required — Use the enabled #1 mode.
2. If enabling VLAN, The BMC should also set the active VLAN ID filters using the NC-SI Set VLAN Filter command prior to setting the VLAN mode.

10.6.4.5 Pass-Through Traffic States

The BMC has independent, separate controls for enablement states of the receive (from LAN) and of the transmit (to LAN) pass-through paths.

10.6.4.6 Channel Enable

This mode controls the state of the receive path:

1. Disabled — The channel does not pass any traffic from the network to the BMC.
2. Enabled — The channel passes any traffic from the network (that matched the configured filters) to the BMC.

This state also affects AENs: AENs is only sent in the enabled state.

The default state is disabled.

It is recommended that the BMC complete all filtering configuration before enabling the channel.

10.6.4.7 Network Transmit Enable

This mode controls the state of the transmit path:

1. Disabled — the channel does not pass any traffic from the BMC to the network.
2. Enabled — the channel passes any traffic from the BMC (that matched the source MAC address filters) to the network.

The default state is disabled.

The NC filters pass-through packets according to their source MAC address. The NC tries to match that source MAC address to one of the MAC addresses configured by the Set MAC Address command. As a result, the BMC should enable network transmit only after configuring the MAC address.

It is recommended that the BMC complete all filtering configuration (especially MAC addresses) before enabling the network transmit.

This feature can be used for fail-over scenarios. See Section 10.6.8.3.

10.6.5 Asynchronous Event Notifications

The asynchronous event notifications are unsolicited messages sent from the NC to the BMC to report status changes (such as link change, operating system state change, etc.).

Recommendations:

- The BMC firmware designer should use AENs. To do so, the designer must take into account the possibility that a NC-SI response frame (such as a frame with the NC-SI EtherType), arrives out-of-context (not immediately after a command, but rather after an out-of-context AEN).
- To enable AENs, the BMC should first query which AENs are supported, using the Get Capabilities command, then enable desired AEN(s) using the Enable AEN command, and only then enable the channel using the Enable Channel command.

10.6.6 Querying Active Parameters

The BMC can use the Get Parameters command to query the current status of the operational parameters.

10.6.7 Resets

In NC-SI there are two types of resets defined:

1. Synchronous entry into the initial state.
2. Asynchronous entry into the initial state.

Recommendations:

- It is very important that the BMC firmware designer keep in mind that following any type of reset, all configurations are considered as lost and thus the BMC must re-configure everything.
- As an asynchronous entry into the initial state might not be reported and/or explicitly noticed, the BMC should periodically poll the NC with NC-SI commands (such as Get Version ID, Get Parameters, etc.) to verify that the channel is not in the initial state. Should the NC channel respond to the command with a Clear Initial State Command Expected reason code, the BMC should consider the channel (and most probably the entire NC package) as if it underwent a (possibly unexpected) reset event. Thus, the BMC should re-configure the NC. See the NC-SI specification section on Detecting Pass-through Traffic Interruption.
- The Intel recommended polling interval is 2-3 seconds.

For exact details on the resets, refer to NC-SI specification.

10.6.8 Advanced Workflows

10.6.8.1 Multi-NC Arbitration

As described in Section 10.6.1.2, in a multi-NC environment, there is a need to arbitrate the NC-SI lines.

Figure 10-10 shows the system topology of such an environment.

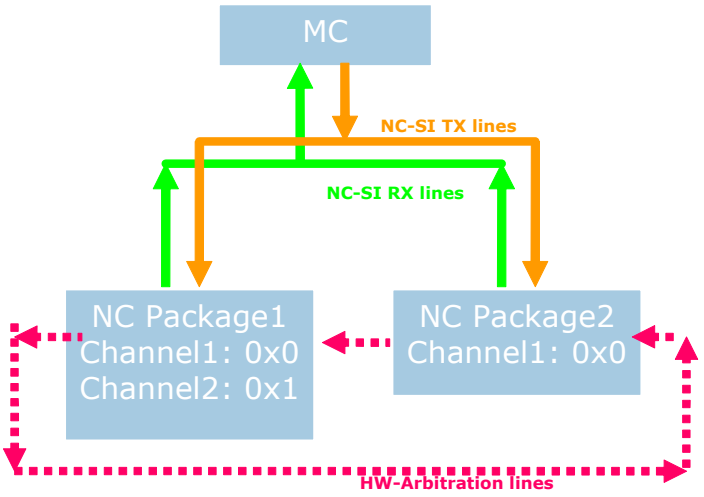


Figure 10-10 Multi-NC Environment

See Figure 10-10. The NC-SI Rx lines are shared between the NCs. To enable sharing of the NC-SI Rx lines, NC-SI has defined an arbitration scheme.

The arbitration scheme mandates that only one NC package can use the NC-SI Rx lines at any given time. The NC package that is allowed to use these lines is defined as selected. All the other NC packages are de-selected.

NC-SI has defined two mechanisms for the arbitration scheme:

1. Package selection by the BMC. In this mechanism, the BMC is responsible for arbitrating between the packages by issuing NC-SI commands (Select/De-Select Package). The BMC is responsible for having only one package selected at any given time.
2. Hardware arbitration. In this mechanism, two additional pins on each NC package are used to synchronize the NC package. Each NC package has an ARB_IN and ARB_OUT line and these lines are used to transfer Tokens. A NC package that has a token is considered selected.

Note: Hardware arbitration is enabled by the NC-SI ARB Enable EEPROM bit (See Section 6.2.22) and the NC-SI HW arbitration support EEPROM bit (see Section 6.3.9.7).

For details on Hardware arbitration, refer to the NC-SI specification.

10.6.8.2 Package Selection Sequence Example

Following is an example work flow for a BMC and occurs after the discovery, initialization, and configuration.

Assuming the BMC needs to share the NC-SI bus between packages, the BMC should:

1. Define a time-slot for each device.
2. Discover, initialize, and configure all the NC packages and channels.
3. Issue a De-Select Package command to all the channels.
4. Set active_package to 0x0 (or the lowest existing package ID).
5. At the beginning of each time slot the BMC should:
   a. Issue a De-Select Package to the active_package. The BMC must then wait for a response and then an additional timeout for the package to become de-selected (200 µs). See the NC-SI specification table 10 — parameter NC Deselect to Hi-Z Interval.
   b. Find the next available package (typically active_package = active_package + 1).
   c. Issue a Select Package command to active_package.

10.6.8.3 Multiple Channels (Fail-Over)

In order to support a fail-over scenario, it is required from the BMC to operate two or more channels. These channels might or might not be in the same package.

The key element of a fault-tolerance fail-over scenario is having two (or more) channels identifying to the switch with the same MAC address, but only one of them being active at any given time (such as switching the MAC address between channels). To accomplish this, NC-SI provides the following commands:

1. Enable Network Tx command — This command enables shutting off the network transmit path of a specific channel. This enables the BMC to configure all the participating channels with the same MAC address but only enable one of them.
2. Link Status Change AEN or Get Link Status command.

10.6.8.3.1 Fail-Over Algorithm Example

The following is a sample workflow for a fail-over scenario for the I350 quad-port GbE controller (one package and four channels):

1. BMC initializes and configures all channels after power-up. However, the BMC uses the same MAC address for all of the channels.
2. The BMC queries the link status of all the participating channels. The BMC should continuously monitor the link status of these channels. This can be accomplished by listening to AENs (if used) and/or periodically polling using the Get Link Status command.
3. The BMC then only enables channel 0 for network transmission.
4. The BMC then issues a gratuitous ARP (or any other packet with its source MAC address) to the network. This packet informs the switch that this specific MAC address is registered to channel 0's specific LAN port.
5. The BMC begins normal workflow.
6. Should the BMC receive an indication (AEN or polling) that the link status for the active channel (channel 0) has changed, the BMC should:
   a. Disable channel0 for network transmission.

b. Check if a different channel is available (link is up).

c. If found:

- Enable network TX for that specific channel.
- Issue a gratuitous ARP (or any other packet with its source MAC address) to the network. This packet informs the switch that this specific MAC address is registered to channel 0's specific LAN port.
- Resume normal workflow.
- If not found, report the error and continue polling until a valid channel is found.

The above algorithm can be generalized such that the start-up and normal workflow are the same. In addition, the BMC might need to use a specific channel (such as channel 0). In this case, the BMC should switch the network transmit to that specific channel as soon as that channel becomes valid (link is up).

Recommendations:

- Wait for a link-down-tolerance timeout before a channel is considered invalid. For example, a link re-negotiation might take a few seconds (normally 2 to 3 or might be up to 9). Thus, the link must be re-established after a short time.
- Typically, this timeout is recommended to be three seconds.
- Even when enabling and using AENs, periodically poll the link status, as dropped AENs might not be detected.

10.6.8.4 Statistics

The BMC might use the statistics commands as defined in NC-SI. These counters are meant mostly for debug purposes and are not all supported.

The statistics are divided into three commands:

1. Controller statistics — These are statistics on the primary interface (to the Host operating system). See the NC-SI specification for details.
2. NC-SI statistics — These are statistics on the NC-SI control frames (such as commands, responses, AENs, etc.). See the NC-SI specification for details.

NC-SI pass-through statistics — These are statistics on the NC-SI pass-through frames. See the NC-SI specification for details.

10.6.9 External Link Control

The BMC can use the NC-SI Set Link command to control the external interface link settings. This command enables the BMC to set the auto-negotiation, link speed, duplex, and other parameters.

This command is only available when the Host operating system is not present. Indicating the Host operating system status can be obtained via the Get Link Status command and/or Host OS Status Change AEN command.

Recommendation:

- Unless explicitly needed, it is not recommended to use this feature. The NC-SI Set Link command does not expose all the possible link settings and/or features. This might cause issues under different scenarios. Even if you decided to use this feature, use it only if the link is down (trust the I350 until proven otherwise).

- It is recommended that the BMC first query the link status using the Get Link Status command. The BMC should then use this data as a basis and change only the needed parameters when issuing the Set Link command.

For details, refer to the NC-SI specification.

10.6.9.1 Set Link While LAN PCIe Functionality is Disabled

In cases where the I350 is used solely for manageability and its LAN PCIe function is disabled, using the NC-SI Set Link command while advertising multiple speeds and enabling auto-negotiation results in the lowest possible speed chosen.

To enable link of higher a speed, the BMC should not advertise speeds that are below the desired link speed, as the lowest advertised link speed is chosen.

When the I350 is only used for manageability and the link speed advertisement is configured by the BMC, changes in the power state of the LAN device is not affected and the link speed is not re-negotiated by the LAN device.

10.7 MCTP

10.7.1 MCTP Overview

The Management Component Transport Protocol (MCTP) defines a communication model intended to facilitate communication between:

- Management controllers and other management controllers
- Management controllers and management devices

The communication model includes a message format, transport description, message exchange patterns, and configuration and initialization messages.

The basic MCTP specification is described in DMTF's DSP0236 document.

MCTP is designed so that it can potentially be used on many bus types. The protocol is intended to be used for intercommunication between elements of platform management subsystems used in computer systems, and is suitable for use in mobile, desktop, workstation, and server platforms.

Currently, specifications exists for MCTP over PCI Express (DMTF's DSP0238) and over SMBus (DMTF's DSP0237). A specification for MCTP over USB is also planned.

Management controllers such as a baseboard management controller (BMC) can use this protocol for communication between one another, as well as for accessing management devices within the platform.

10.7.1.1 NC-SI Over MCTP

MCTP is a transport layer protocol that does not include the functionality required to control the pass through traffic required for BMC connection to the network or to allow the BMC to control the network controller. This functionality is provided by encapsulating NC-SI traffic as defined in DMTF's DSP0222 document.

The details of NC-SI over MCTP protocol are defined in the NC-SI Over MCTP Specification.

10.7.1.2 MCTP Usage Model

The I350 supports NC-SI over MCTP protocol over SMBus. A BMC can be connected to a I350 NIC through MCTP as described in Figure Note: . The MCTP interface will be used by the BMC to control the NIC and not for pass through traffic.

Note: MCTP over SMBus is not active while in Dr State. In this state, the I350 will not answer to SMBus commands.

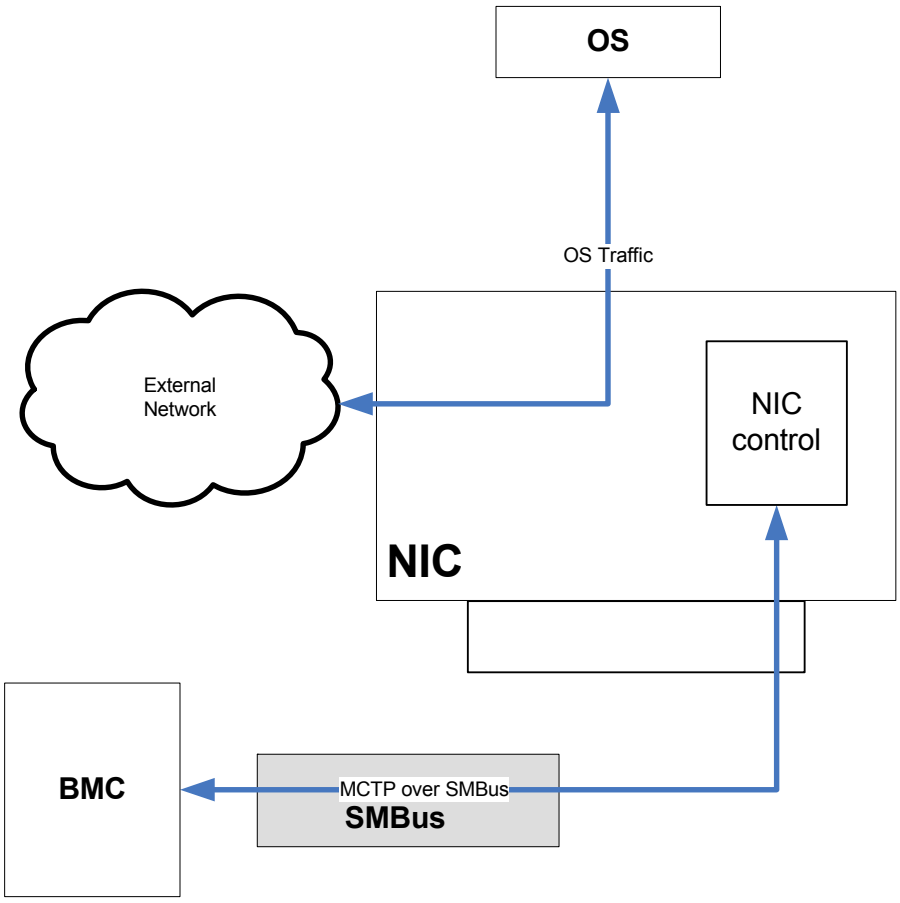


Figure 10-11 MCTP connections of the I350

10.7.1.3 Simplified MCTP Mode

For some point to point implementations of MCTP the assembly process is simplified. In this mode, the Destination EID, Source EID, Packet sequence number, Tag Owner (TO) bit and Message tag are ignored and the assembly is based only on the SOM & EOM bits. This bit is set according to the Simplified MCTP bit in the MCTP configuration word in the NVM.

This mode is relevant only for MCTP over SMBus traffic

10.7.2 NC-SI to MCTP Mappaing

The four network ports of the I350 (mapped to four NC-SI channels) are mapped to a single MCTP endpoint on SMBus.

The channels are not used for pass through traffic and are only used to define which port is currently being accessed for control or status update.

The topology used for MCTP connection is described in Figure 10-12.

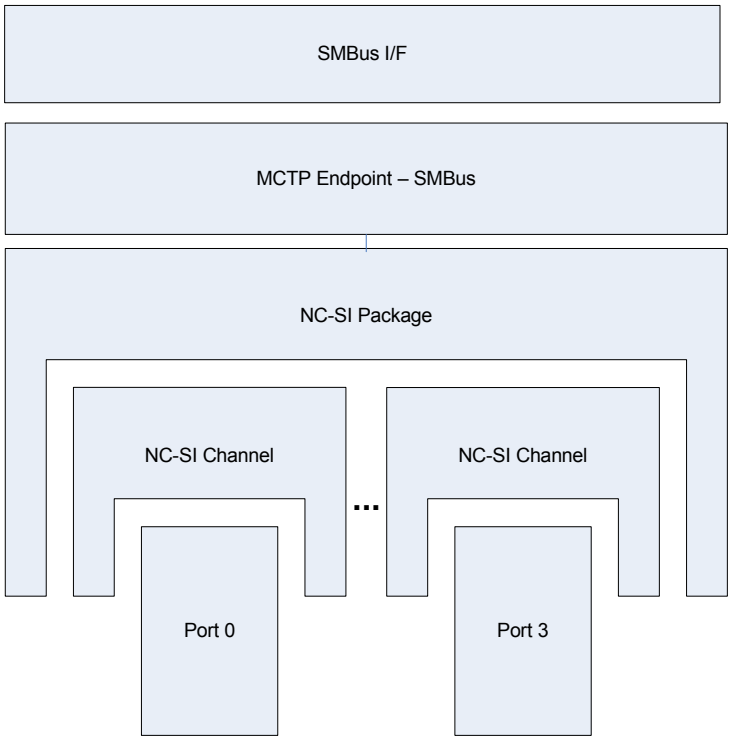


Figure 10-12 MCTP endpoints topology

10.7.3 MCTP Over SMBus

The message format used for NC-SI over MCTP over SMBus is as follows:

+0								+1								+2								+3							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Destination Slave Address							0	Command Code = MCTP = 0Fh								Byte count								Source Slave Address							
MCTP Reserved				Header version = 1				Destination endpoint ID								Source endpoint ID								SOM	EOM	SEQ#			TO	Tag	
IC [1]	Message Type = 0x02							Reserved								NC-SI Command/Pass Through data															
.....																															
NC-SI Command/Pass Through data																															
PEC																															

1. IC = 0

SMBus header/PEC
MCTP header
NC-SI header and payload

10.7.3.1 SMBus Discovery Process

The I350 follows the discovery process described in section 5.5 of the MCTP SMBus/I2C Transport Binding Specification (DSP0237). It indicates support for ASF in the SMBus getUID command (see Section 10.5.9.5). It will respond to any SMBus command using the MCTP command code - so that the bus owner knows the I350 supports MCTP.

10.7.3.2 SMBus over MCTP implementation notes

Given the limited usage of MCTP over SMBus in the I350, it is assumed that collisions on the SMBus will not occur or will be rare. Thus in case of lost arbitration, the I350 will drop the current transaction and will not try to send it again.

10.7.4 NC-SI Over MCTP

The I350 support for NC-SI over MCTP is similar to the support for NC-SI over RMII with the following exceptions that the format of the packets is modifed to account for the new transport layer as described below.

10.7.4.1 NC-SI Packets Format

NC-SI over MCTP defines two different message type for pass through and for control packets.

Packets with a message type equal to the *Control packets message type* field (default = 0x02) in the EEPROM are NC-SI control packets (commands, responses and AENs) and packets with a message type equal to the *Pass through packets message type* field (default = 0x02) in the EEPROM are NC-SI pass through packets

Note:

10.7.4.1.1 Control Packets

The format used for Control packets (Commands, Responses and AENs) is as follow:

+0								+1								+2								+3							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
SMBus or PCIe header																															
MCTP Reserved						Header version = 1		Destination endpoint ID								Source endpoint ID								SOM	EOM	SEQ#				TO = 1	Tag
IC	Message Type = Control Packets Message type (0x02)							MC ID = 0x00								Header revision								Reserved							
IID								Command								Channel ID[1]								Reserved							
Payload Length[7:0]								Reserved																Payload Length[11:8]							
Reserved																															
Reserved								Command Data																							
....																															
Command Data																Checksum															
Checksum																															

1. The channel ID is defined as described in Section 10.2.2.2

SMBus/PCIe header
MCTP header
NC-SI header
NC-SI Data

Note that the MAC header and MAC FCS present when working over NC-SI are not part of the packet in MCTP mode.

10.7.4.1.2 Command Packets

The format used for Command packets is as follow:

+0								+1								+2								+3							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Destination Slave Address							0	Command Code = MCTP = 0Fh								Byte count								Source Slave Address							
MCTP Reserved						Header version = 1		Destination endpoint ID								Source endpoint ID								SOM	EOM	SEQ#				TO = 1	Tag

+0		+1	+2	+3
IC	Message Type = 0x02	Reserved	MC ID = 0x00	Header revision
Reserved		IID	Command	Channel ID
Reserved	Payload Length		Reserved	
Reserved				
Reserved			Command Data	
....				
Command Data			Checksum	
Checksum				

SMBus header
MCTP header
NC-SI header
NC-SI Data

Note that the MAC header and MAC FCS present when working over NC-SI are not part of the packet in MCTP mode.

10.7.4.1.3 Response Packets

The format used for Response packets is as follow:

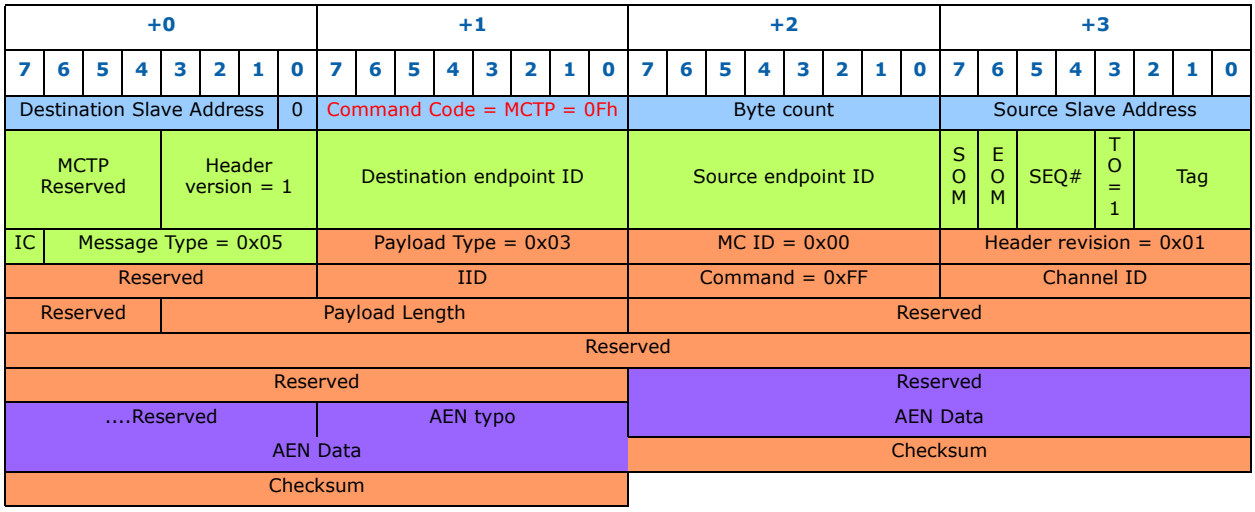
+0								+1								+2								+3							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Destination Slave Address							0	Command Code = MCTP = 0Fh								Byte count								Source Slave Address							
MCTP Reserved				Header version = 1				Destination endpoint ID								Source endpoint ID								SOM	EOM	SEQ#				TO = 0	Tag
IC	Message Type = 0x02							Reserved								MC ID = 0x00								Header revision = 0x01							
Reserved								IID								Command								Channel ID							
Reserved				Payload Length												Reserved															
Reserved																															
Reserved																Response code															
Reason code																Response Data															
....																															
Command Data																Checksum															
Checksum																															

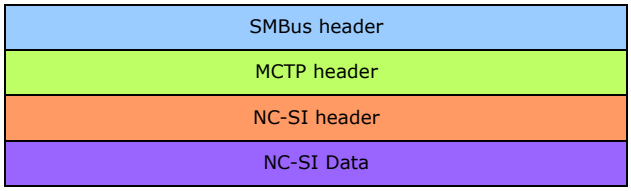
SMBus header
MCTP header
NC-SI header

Note that the MAC header and MAC FCS present when working over NC-SI are not part of the packet in MCTP mode.

10.7.4.1.4 AEN Packets

The format used for AEN packets is as follow:





Note that the MAC header and MAC FCS present when working over NC-SI are not part of the packet in MCTP mode.

10.7.5 MCTP Programming

The MCTP programming model is based on:

1. A set of MCTP commands used for the discovery process and for the link management. The list of supported commands is described in section Section 10.7.5.1.
2. A subset of the NC-SI commands used in the regular NC-SI interface, including all the OEM commands as described in Section 10.6.2 (NC-SI programming I/F). The specific commands supported are listed in Table 10-30 and Table 10-33.

Note: For all MCTP commands (both native MCTP commands and NCSI over MCTP), the response uses the Msg tag received in the request with TO bit cleared.

10.7.5.1 MCTP Commands Support

Table 10-46 lists the MCTP commands supported by the I350.

Table 10-46 MCTP commands support

Command Code	Command Name	General Description	I350 support as Initiator	I350 support as Responder
0x00	Reserved	Reserved	–	–
0x01	Set Endpoint ID	Assigns an EID to the endpoint at the given physical address.	N/A	Yes
0x02	Get Endpoint ID	Returns the EID presently assigned to an endpoint. Also returns information about what type the endpoint is and its level of use of static EIDs. See Section 10.7.5.1.1 for details.	No	Yes
0x03	Get Endpoint UUID	Retrieves a per-device unique UUID associated with the endpoint. See Section 10.7.5.1.2 for details.	No	Yes
0x04	Get MCTP Version Support	Lists which versions of the MCTP control protocol are supported on an endpoint. See Section 10.7.5.1.3 for details.	No	Yes
0x05	Get Message Type Support	Lists the message types that an endpoint supports. See Section 10.7.5.1.4 for details.	No	Yes
0x06	Get Vendor Defined Message Support	Used to discover an MCTP endpoint's vendor specific MCTP extensions and capabilities.	No	No
0x07	Resolve Endpoint ID	Used to get the physical address associated with a given EID.	No	N/A
0x08	Allocate Endpoint IDs	Used by the bus owner to allocate a pool of EIDs to an MCTP bridge.	N/A	N/A
0x09	Routing Information Update	Used by the bus owner to extend or update the routing information that is maintained by an MCTP bridge.	N/A	N/A
0x0A	Get Routing Table Entries	Used to request an MCTP bridge to return data corresponding to its present routing table entries.	No	N/A
0x0B	Prepare for Endpoint Discovery	Used to direct endpoints to clear their "discovered" flags to enable them to respond to the Endpoint Discovery command.	N/A	Yes[1]
0x0C	Endpoint Discovery	Used to discover MCTP-capable devices on a bus, provided that another discovery mechanism is not defined for the particular physical medium.	No	Yes[1]
0x0D	Discovery Notify	Used to notify the bus owner that an MCTP device has become available on the bus.	No	N/A
0x0E	Reserved	Reserved	–	–
0x0F	Query Hop	Used to discover what bridges, if any, are in the path to a given target endpoint and what transmission unit sizes the bridges will pass for a given message type when routing to the target endpoint.	No	No

1. These commands are supported only for MCTP over PCIe.

10.7.5.1.1 Get Endpoint ID

The Get Endpoint ID response of the I350 is described in the following table:

Byte	Description	Value
1	Completion Code	

Byte	Description	Value
2	Endpoint ID	0x00 - EID not yet assigned Otherwise - returns EID assigned using Set Endpoint ID command
3	Endpoint Type	0x00 (Dynamic EID, Simple Endpoint)
4	Medium Specific	SMBUs: 0x01 - Fairness arbitration protocol supported. PCIe: 0x00

10.7.5.1.2 Get Endpoint UUID

The UUID returned is calculated according to the following function:

Time Low = Read from NVM words at offset 0x9 and 0xA of Sideband Configuration Structure.

Time mid = Read from NVM word at offset 0xB of Sideband Configuration Structure

Time High and version = Read from NVM word at offset 0xC of Sideband Configuration Structure

Clock Sec and Reserved = Read from NVM word at offset 0xD of Sideband Configuration Structure

Node = Host MAC address of port 0 as stored in NVM.

10.7.5.1.3 Get MCTP Version Support

The following table describes the returned value according to the requested message type

Byte	Description	Message type			
		0xFF(Base)	0x00 (Control protocol message)	0x02 (NC-SI over MCTP)	All other
1	Completion Code				0x80
2	Version Number entry count	1	1	1	0
6:3	Version number entry	0xF1F0FF00	0xF1F0FF00	1	0

10.7.5.1.4 Get Message Type Support Command

The Get Message type support response of the I350 is described in the following table:

Byte	Description	Value
1	Completion Code	0x00
2	MCTP Message Type Count	0x01 - The I350 supports one additional message type
3	List of Message Type numbers	0x02 (NC-SI over MCTP)

10.7.5.1.5 Set Endpoint ID Command

The I350 supports the Set EID and Force EID operations defined in the Set Endpoint ID command. As endpoints in the I350 can be set only through their own interface, Set EID and Force EID are equivalent. The Reset EID and Set Discovered Flag operations are not relevant to the I350.

The Set Endpoint ID response of the I350 is described in the following table:

Byte	Description	Value
1	Completion Code	0x00
2	Completion Status	[7:6] = 00 - Reserved
		[5:4] = 00 - EID assignment accepted
		[3:2] = 00 - Reserved
		[1:0] = 00 - Device does not use an EID pool.

10.8 Manageability Host Interface

This section details host interaction with the manageability portion of the I350. The information within this section is only available to the host driver, the BMC does not have access.

10.8.1 HOST CSR Interface (All Functions)

The software device driver of all functions communicates with the manageability block through CSR access. The manageability is mapped to address space 0x8800 to 0x8FFF on the slave bus of each function.

Note: Writing to address 0x8800 from any function is targeted to the same address in the RAM.

10.8.2 Host Slave Command Interface to Manageability

This interface is used by the software device driver for several of the commands and for delivering various types of data in both directions (Manageability-to-Host and Host-to-Manageability).

The address space is separated into two areas:

- Direct access to the internal data RAM: The internal shared (between Firmware and Software) RAM is mapped to address space 0x8800 to 0x8EFF. Writing/reading to this address space goes directly to the RAM.
- Control register is located at address 0x8F00.

10.8.2.1 Host Slave Command Interface Low Level Flow

This interface is used for the external host software to access the manageability subsystem. Host software writes a command block or read data structure directly from the data RAM. Host software controls these transactions through a slave access to the control register.

The following flow shows the process of initiating a command to the manageability block:

1. The Software clears the *FWSTS.FWRI* flag (clear by write one) to clear any previous firmware reset indications.
2. The Software device driver takes ownership of the Management Host interface using the flow described in Section 4.7.1.

3. The Software device driver reads the *HOST Interface Control* Register (See Section 8.22.1.2) and checks that the *Enable* (*HICR.En*) bit is set.
4. The Software device driver writes the relevant command block into the RAM area that is mapped to addresses 0x8800-0x8EFF.
5. The Software device driver sets the *Command* (*HICR.C*) bit in the *HOST Interface Control* Register (See Section 8.22.1.2). Setting this bit causes an interrupt to the ARC (can be masked).
6. The Software checks the *FWSTS.FWRI* flag to make sure a firmware reset didn't occur during the command processing. If this bit is set, the command may have failed.
7. The Software device driver polls the *HOST Interface Control* register for the *Command* (*HICR.C*) bit to be cleared by Firmware.
8. When Firmware finishes with the command, it clears the *Command* (*HICR.C*) bit (if Firmware replies with data, it should clear the bit only after the data is placed in the shared RAM area where the software device driver can read it).

If the Software device driver reads the *HOST Interface Control* register and the *HICR.SV* bit is set to 1b, then there is a valid status of the last command in the shared RAM. If the *HICR.SV* bit is not set, then the command has failed with no status in the RAM.

On completion of access to the shared RAM Software device driver should release ownership of the shared RAM using the flow described in Section 4.7.2.

10.8.2.2 Host Slave Command Registers

10.8.2.2.1 Host Interface Control Register (CSR Address 0x8F00)

This register operates along with the host software/firmware interface (See Section 8.22.1.2).

10.8.2.3 Host Interface Structures

10.8.2.3.1 Host Interface Command Structure

Table 10-47 describes the structure used by the Software device driver to send a command to Firmware using the Host slave command interface (shared RAM mapped to addresses 0x8800-0x8EFF).

Table 10-47 Host Driver Command Structure

#Byte	Description	Bit	Value	Description
0	Command	7:0	Command Dependent	Specifies which host command to process.
1	Buffer Length	7:0	Command Length	Command Data Buffer length: 0 to 252, not including 32 bits of header.
2	Default/Implicit Interface	0	Command Dependent	Used for commands might refer to one of four interfaces (LAN or SMBus). 0b = Use default interface. 1b = Use specific interface.
	Interface Number	2:1	Command Dependent	Used when bit 0 (Default/Implicit interface) is set: 00b = Apply command for interface 0. 01b = Apply command for interface 1. 10b = Apply command for interface 2. 11b = Apply command for interface 3. When bit 0 is set to 0b, it is ignored.

Table 10-47 Host Driver Command Structure

#Byte	Description	Bit	Value	Description
	Reserved	7:3	0x0	Reserved
3	Checksum	7:0	Defined Below	Checksum signature.
255:4	Data Buffer	7:0	Command Dependent	Command Specific Data Minimum buffer size: 0. Maximum buffer size: 252.

10.8.2.3.2 Host Interface Status Structure

Table 10-48 lists the structure used by Firmware to return a status to the Software device driver via the Host slave command interface. A status is returned after a command has been executed.

Table 10-48 Status Structure Returned to Host Driver

#Byte	Description	Bit	Value	Description
0	Command	7:0	Command Dependent	Command ID.
1	Buffer Length	7:0	Status Dependent	Status buffer length: 252:0
2	Return Status	7:0	Depends on Command Executing Results	0x1 Status OK 0x2 Illegal command ID 0x3 Unsupported command 0x4 Illegal payload length 0x5 Checksum failed 0x6 Data Error 0x7 Invalid parameter 0x8 - 0xFF Reserved
3	Checksum	7:0	Defined Below	Checksum signature.
255:4	Data Buffer		Status Dependent	Status configuration parameters Minimum Buffer Size: 0. Maximal Buffer Size: 252.

10.8.2.3.3 Checksum Calculation Algorithm

The Host Command/Status structure is summed with this field cleared to 0b. The calculation is done using 8-bit unsigned math with no carry. The inverse of this sum is stored in this field (0b minus the result). Result: The current sum of this buffer (8-bit unsigned math) is 0b.

10.8.2.4 Host Interface Commands

10.8.2.4.1 Driver Info Host Command

This command is used to provide the driver information in NC-SI mode.

Table 10-49 Driver Info Host Command

Byte	Name	Bit	Value	Description
0	Command	7:0	0xDD	Driver info command.
1	Buffer Length	7:0	0x5	Port Number + 4 bytes of the Driver info
2	Reserved	7:0	0x0	Reserved
3	Checksum	7:0		Checksum signature of the Host command.
4	Port Number	7:0	Port Number	Indicates the port currently reporting its driver info
8:5	Driver Version	7:0	Driver Version	Numerical for driver version - should be Byte 8:Major Byte 7:Minor Byte 6:Build Byte 5:SubBuild

Following is the status returned on this command:

Table 10-50 Driver Info Host Status

Byte	Name	Bit	Value	Description
0	Command	7:0	0xDD	Driver Info command
1	Buffer Length	7:0	0x0	No data in return status
2	Return Status	7:0	0x1	0x1 for good status
3	Checksum	7:0		Checksum signature

10.8.2.4.2 Host Proxying Commands

Software Device driver will send to Firmware via shared RAM interface the following Proxying commands, using the interface described in Section 10.8.2.1:

1. Get Firmware Proxying Capabilities Command (See Table 10-51) to receive information on Protocol offloads supported.
2. Set Firmware Proxying Configuration Command (See Table 10-53) to define the required proxying behavior.
3. Send the required Proxying information for the Protocol offloads supported by Firmware via the following commands:
   a. Set ARP Proxy Table Entry (See Table 10-55).
   b. Set NS (Neighbor Solicitation) Proxy Table Entry (See Table 10-57).

Following the reception of the commands, Firmware will acknowledge execution of the command via the shared RAM interface using the following responses according to the command issued:

1. Get Firmware Proxying Capabilities Response (See Table 10-52).
2. Set Firmware Proxying Configuration Response (See Table 10-54).
3. Acknowledge reception of Proxying information via the following responses:
   a. Set ARP Proxy Table Entry Response (See Table 10-56).
   b. Set NS (Neighbor Solicitation) Proxy Table Entry Response (See Table 10-58)

10.8.2.4.2.1 Get Firmware Proxying Capabilities

This command is used to provide the driver information on protocol offload types supported by the I350.

Table 10-51 Get Firmware Proxying Capabilities Command

Byte	Name	Bit	Value	Description
0	Command	7:0	0xEA	GET Firmware Proxying Capabilities
1	Buffer length	7:0	0x2	
2	Reserved	7:0	0x0	Must be zeroed by host
3	Checksum	7:0		Checksum signature
4	Port Number	7:0	Port Number	Indicates the port number that the command is targeted at.
5	Page	7:0	0x1	Get capabilities page number If response exceeds 256 bytes including header (Maximum page size), Software should issue multiple Get Firmware Proxying Capabilities commands with increasing page number until a response with buffer length smaller than 252 is received or a response with a Status field with an Unsupported Page Number is received. ***Note:*** Maximum Page size is 256 Bytes including Header information.

Firmware returns the following status for this command:

Note: The Firmware status reply includes a series of two values

{Protocol offload capability type and version, number of entries for this type of Protocol offload}

Currently the following capabilities are defined, ARP proxy NS proxy and MLD support. If the structure is too big to transfer in one time the driver can ask for additional pages by incrementing the page field.

Table 10-52 Get Firmware Proxying Capabilities Response

Byte	Name	Bit	Value	Description
0	Command	7:0	0xEA	Get Firmware Proxying Capabilities
1	Buffer length	7:0	0x8	
2	Return Status	7:0	0x1	0x0 - Unsupported Page number 0x1 - Status OK 0x2 to 0xFF - Error
3	Checksum	7:0		Checksum signature
4	Port Number	7:0	Port Number	Indicates the port number that the response is for.
5	Page	7:0	0x1	First page of capabilities
6	Total Cap size	7:0	0x4[1]	Size of capability structure in bytes
7	ARP proxy version 1	7:0	0x1	
8	Number of ARP entries	7:0	Number of entries	Number of ARP entries supported
9	NS proxy version 1	7:0	0x2	
10	Number of NS proxy entries	7:0	Number of entries	Number of NS entries supported
11	MLD support	7:0	Version of MLD supported	0x0 - not supported 0x1 - MLD version 1 compatibility mode 0x2 - MLD version 2 compatibility mode 0x3 - Both versions supported 0x4 - x0FF: Reserved

1. Note that this number should be 5. will be fixed in next product.

10.8.2.4.2.2 Set Firmware Proxying Configuration

This command is used to provide information to Firmware on how to implement protocol offloads supported by the I350.

The Firmware Proxying Configuration command includes a series of two values

{Command type and version, Command Data for this type of command}

Currently only two Configuration commands are defined:

1. No Match - Command defines expected behavior when receiving a Proxying packet that’s not supported.
2. D3 to D0 - Command defines expected behavior when the I350 moves from D3 to D0 state.

If the structure is too big to transfer in one time the driver can ask for additional pages by incrementing the page field.

Table 10-53 Set Firmware Proxying Configuration Command

Byte	Name	Bit	Value	Description
0	Command	7:0	0xEB	Set Firmware Proxying Configuration
1	Buffer length	7:0	0x6	
2	Reserved	7:0	0x0	Must be zeroed by host
3	Checksum	7:0		Checksum signature
4	Port Number	7:0	Port Number	Indicates the port number that the command is targeted at. ***Note:*** Port Number should be programmed according to value read from the *STATUS.LAN ID* field (See Section 8.2.2).
5	No Match	7:0	0x1	No Match command Defines how Firmware handles unsupported proxying packets.
6	No Match data	7:0	0x0 or 0x1	No Match data 0x0 - Discard unsupported proxying packets 0x1 - Issue Wake on reception of unsupported packets. ***Note:*** 0x0 is the default value if no configuration command is issued.
7	D3 to D0	7:0	0x2	D3 to D0 command Defines how to handle Proxying table entries and setting when device moves from D3 to D0 power state.
8	D3 to D0 data	7:0	0x0 or 0x1	D3 to D0 data 0x0 - Restore default settings. 0x1 - Keep Proxying settings. Defines how to handle table entries when device moves from D3 to D0 power state. ***Notes:*** 1. 0x0 is the default value if no configuration command is issued. 2. If value is 0x0 all configuration commands are also cleared on move from D3 to D0.
9	Enable MLD	7:0	0x0, 0x1 or 0x2	0x0: Do not enable MLD 0x1: Enable MLD version 1 0x2: Enable MLD version 2 0x3 - x0FF: Reserved

Firmware returns the following Response for this command:

Table 10-54 Set Firmware Proxying Configuration Response

Byte	Name	Bit	Value	Description
0	Command	7:0	0xEB	Get Firmware Proxying Capabilities
1	Buffer length	7:0	0x6	
2	Return Status	7:0	0x1	0x0 - Undefined Error 0x1 - Status OK 0x2 - Unsupported command 0x3 - Checksum Error 0x4 - Buffer Length Error 0x5 to 0xFF - Error
3	Checksum	7:0		Checksum signature
4	Port Number	7:0	Port Number	Indicates the port number that the status is from.
5	No Match	7:0	0x1	No Match command Defines how Firmware handles unsupported proxying packets.
6	No Match data	7:0	0x0 or 0x1	No Match Data 0x0 - Discard unsupported proxying packets 0x1 - Issue Wake on reception of unsupported packets. ***Note:*** 0x0 is the default value if no configuration command is issued.
7	D3 to D0	7:0	0x2	D3 to D0 command Defines how to handle Proxying table entries and setting when device moves from D3 to D0 power state.
8	D3 to D0 data	7:0	0x0 or 0x1	D3 to D0 Data 0x0 - Restore default settings and discard any Proxying table entries. 0x1 - Keep Proxying settings. Defines how to handle table entries when device moves from D3 to D0 power state. ***Notes:*** 1. 0x0 is the default value if no configuration command is issued. 2. If value is 0x0 all configuration commands are also cleared on move from D3 to D0.
9	Enable MLD	7:0	0x0, 0x1 or 0x2	0x0: Do not enable MLD 0x1: Enable MLD version 1 0x2: Enable MLD version 2 0x3 - x0FF: Reserved

10.8.2.4.2.3 Set ARP Proxy Table Entry

Note: To set an entry Software driver will post command with *Active* field set to 0x1. To disable an entry the driver must post this entry with the *Active* field set to 0x0. Driver must initially disable all unused entries.

Table 10-55 Set ARP Proxy Table Entry Command

Byte	Name	Bit	Value	Description
0	Command	7:0	0x77	Set ARP proxy command
1	Buffer length	7:0	0x13	
2	Reserved	7:0	0x0	Must be zeroed by host
3	Checksum	7:0		Checksum signature
4	Port Number	7:0	Port Number	Indicates the port number that the command is targeted at. ***Note:*** Port Number should be programmed according to value read from the *STATUS.LAN ID* field (See Section 8.2.2).
5	Sub command	7:0	0x3	Set proxy capabilities
6	ARP proxy version 1	7:0	0x1	ARP version 1 entry
7	Table index	7:0	Index	Table index Each Set proxy command is held in a separate Table index. Field defines Table Index for current command. ***Notes:*** 1. Table Index values begin at 1. 2. Only a single ARP proxy table entry is supported and only a Table index value of 1 is valid. 3. To change contents of a table entry the relevant Table index should be invalidated (Write command to the Table index with Active field = 0x0) before writing new content.
8	Active	7:0	0x1 or 0x0	Set to 0x1 to activate table index Set to 0x0 to invalidate it If set to 0, values of all following fields are ignored
14:9	MAC Address	7:0		MAC Address to reply to ARP request
18:15	Local IP Address	7:0		Local IP Address of station
22:19	Remote IP Address	7:0		Remote IP Address A value of 0x0 indicates any remote IP address

Firmware returns the following status for this command:

Table 10-56 Set ARP Proxy Table Entry Response

Byte	Name	Bit	Value	Description
0	Command	7:0	0x77	Set ARP proxy command
1	Buffer length	7:0	0x13	
2	Status	7:0	0x1	0x0 - Unsupported Table Index 0x1 - Status OK 0x2 - Table Index in use. 0x3 - Unsupported command 0x4 - Checksum Error 0x5 - Buffer Length Error 0x6 to 0xFF - Error
3	Checksum	7:0		Checksum signature
4	Port Number	7:0	Port Number	Indicates the port number that the response is for.
5	Sub command	7:0	0x3	Set proxy capabilities
6	ARP proxy version 1	7:0	0x1	ARP version 1 entry
7	Table index	7:0	Index	
8	Active	7:0	0x1 or 0x0	Set if entry is active
14:9	MAC Address	7:0		
18:15	Local IP Address	7:0		
22:19	Remote IP Address	7:0		

10.8.2.4.2.4 Set NS (Neighbor Solicitation) Proxy Table Entry

Note: To set an entry Software driver will post command with *Active* field set to 0x1. To disable an entry the driver must post this entry with the *Active* field set to 0x0. Driver must initially disable all unused entries.

Table 10-57 Set NS Proxy Table Entry Command

Byte	Name	Bit	Value	Description
0	Command	7:0	0x78	Set NS proxy command
1	Buffer length	7:0	0x4B	
2	Reserved	7:0	0x0	Must be zeroed by host
3	Checksum	7:0		Checksum signature
4	Port Number	7:0	Port Number	Indicates the port number that the command is targeted at. ***Note:*** Port Number should be programmed according to value read from the *STATUS.LAN ID* field (See Section 8.2.2).
5	Sub command	7:0	0x3	Set proxy capabilities
6	NS proxy version 1	7:0	0x2	NS version 1 entry
7	Table index	7:0	Index	Table index Each Set proxy command is held in a separate Table index. Field defines Table Index for current command. ***Notes:*** 1. Table Index values begin at 1. 2. Up to two NS proxy table entries are supported and only Table index values of 1 and 2 are valid. 3. To change contents of a table entry the relevant Table index should be invalidated (Write command to the Table index with Active field = 0x0) before writing new content.

Byte	Name	Bit	Value	Description
8	Active	7:0	0x1 or 0x0	Set to 0x1 to activate table index Set to 0x0 to invalidate it If set to 0, values of all following fields are ignored
14:9	MAC Address	7:0		MAC Address
30:15	Local IPv6 Address 1	7:0		Local IPv6 Address 1
46:31	Local IPv6 Address 2	7:0		Local IPv6 Address 2 If there is only one local address value placed is 0x0
62:47	Remote IPv6 Address	7:0		Remote IPv6 Address A value of 0x0 indicates any address.
78:63	Solicited IPv6 Address	7:0		Solicited IPv6 Address

Firmware returns the following status for this command:

Table 10-58 Set NS Proxy Table Entry Response

Byte	Name	Bit	Value	Description
0	Command	7:0	0x78	Set NS proxy command
1	Buffer length	7:0	0x4B	
2	Status	7:0	0x1	0x0 - Unsupported Table Index 0x1 - Status OK 0x2 - Table Index in use. 0x3 - Unsupported command 0x4 - Checksum Error 0x5 - Buffer Length Error 0x6 to 0xFF - Error
3	Checksum	7:0		Checksum signature
4	Port Number	7:0	Port Number	Indicates the port number that the status is for.
5	Sub command	7:0	0x3	Set proxy capabilities
6	NS proxy version 1	7:0	0x2	NS version 1 entry
7	Table index	7:0	Index	
8	Active	7:0	0x1 or 0x0	Set if entry is active
14:9	MAC Address	7:0		MAC Address
30:15	Local IPv6 Address 1	7:0		Local IPv6 Address 1
46:31	Local IPv6 Address 2	7:0		Local IPv6 Address 2 If there is only one local address value placed is 0x0
62:47	Remote IPv6 Address	7:0		Remote IPv6 Address A value of 0x0 indicates any address.
78:63	Solicited IPv6 Address	7:0		Solicited IPv6 Address

10.8.3 Host Isolate Support

If a BMC decides that a malicious software prevents its usage of the LAN, it may decide to isolate the NIC from its driver. This is done using the TCO reset command (Section 10.6.3.12).

If TCO isolate is enabled in the EEPROM (See Section 6.3.7.3), The TCO Isolate command will disable PCIe write operations to the LAN port. As the driver needs to access the CSR space in order to provide descriptors to the NIC, this operation will also stop the network traffic including OS2BMC and BMC to OS traffic as soon as the existing transmit and receive descriptor queues are exhausted.

11 Electrical/Mechanical Specification

11.1 Introduction

These specifications are subject to change without notice.

This chapter describes the I350 DC and AC (timing) electrical characteristics. This includes absolute maximum rating, recommended operating conditions, power sequencing requirements, DC and AC timing specifications. The DC and AC characteristics include generic digital 3.3V IO specification as well as other specifications supported by the I350.

11.2 Operating Conditions

Table 11-1 Absolute Maximum Ratings[1]

Symbol	Parameter	Min	Max	Units
T_{case}	Case Temperature Under Bias		100	°C
$T_{storage}$	Storage Temperature Range	–65	140	°C
Vi/Vo	3.3V Compatible I/Os Voltage Analog 1.0 I/O Voltage Analog 1.8 I/O Voltage	Vss – 0.5 Vss – 0.2 Vss – 0.3	4.6 1.68 2.52	V
VCC3P3	3.3V Periphery DC Supply Voltage	Vss – 0.5	4.6	V
VCC	1.0V Core DC Supply Voltage	Vss – 0.2	1.68V	V
VCC1P8	1.8V Analog DC Supply Voltage	Vss – 0.3	2.52	V
VCC1P0	1.0V Analog DC Supply Voltage	Vss – 0.2	1.68V	V

1. Ratings in this table are those beyond which permanent device damage is likely to occur. These values should not be used as the limits for normal device operation. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

11.2.1 Recommended Operating Conditions

Table 11-2 Recommended Operating Conditions

Symbol	Parameter	Min	Max	Units	Notes
Ta	Operating Temperature Range Commercial (Ambient; 0 CFS airflow)	-10	55 85	°C	1,2,3

Note:

1. For normal device operation, adhere to the limits in this table. Sustained operations of a device at conditions exceeding these values, even if they are within the absolute maximum rating limits, may result in permanent device damage or impaired device reliability. Device functionality to stated DC and AC limits is not guaranteed if conditions exceed recommended operating conditions.
2. Recommended operation conditions require accuracy of power supply as defined in Section 11.3.1.
3. With external heat sink. Airflow required for operation in 85'C ambient temperature.

11.3 Power Delivery

11.3.1 Power Supply Specification

VCC3P3 (3.3V) Parameters				
Parameter	**Description**	**Min**	**Max**	**Units**
Rise Time[1]	Time from 10% to 90% mark	0.1	100	mS
Monotonicity	Voltage dip allowed in ramp	N/A	0	mV
Slope	Ramp rate at any given time between 10% and 90% Min: 0.8*V(min)/Rise time (max) Max: 0.8*V(max)/Rise time (min)	24	28800	V/S
Operational Range	Voltage range for normal operating conditions	3	3.6	V
Ripple[2]	Maximum voltage ripple (peak to peak)	N/A	70	mV
Overshoot	Maximum overshoot allowed	N/A	100	mV
Overshoot Settling Time	Maximum overshoot allowed duration. (At that time delta voltage should be lower than 5mv from steady state voltage)	N/A	0.05	mS
Decoupling Capacitance	Capacitance range	15		µF
Capacitance ESR	Equivalent series resistance of output capacitance	N/A	50	mΩ
VCC1P8 (1.8V) Parameters				
Parameter	**Description**	**Min**	**Max**	**Units**
Rise Time	Time from 10% to 90% mark	0.1	100	mS
Monotonicity	Voltage dip allowed in ramp	N/A	0	mV
Slope	Ramp rate at any given time between 10% and 90% Min: 0.8*V(min)/Rise time (max) Max: 0.8*V(max)/Rise time (min)	14	60000	V/S
Operational Range	Voltage range for normal operating conditions	1.71	1.89	V
Ripple[2]	Maximum voltage ripple (peak to peak)	N/A	40	mV

Overshoot	Maximum overshoot allowed	N/A	100	mV
Overshoot Settling Time	Maximum overshoot allowed duration. (At that time delta voltage should be lower than 5mv from steady state voltage)	N/A	0.1	mS
Decoupling Capacitance	Capacitance range	15	25[3]	µF
Capacitance ESR	Equivalent series resistance of output capacitance	N/A	50	mΩ
VCC1P0 (1.0V) Parameters				
Parameter	**Description**	**Min**	**Max**	**Units**
Rise Time	Time from 10% to 90% mark	0.1	80	mS
Monotonicity	Voltage dip allowed in ramp	N/A	0	mV
Slope	Ramp rate at any given time between 10% and 90% Min: 0.8*V(min)/Rise time (max) Max: 0.8*V(max)/Rise time (min)	7.6	33600	V/S
Operational Range	Voltage range for normal operating conditions	0.93	1.08	V
Ripple[2]	Maximum voltage ripple (peak to peak)	N/A	40	mV
Overshoot	Maximum overshoot allowed	N/A	100	mV
Overshoot Duration	Maximum overshoot allowed duration. (At that time delta voltage should be lower than 5mv from steady state voltage)	0.0	0.05	mS
Decoupling Capacitance	Capacitance range	15	25[3]	µF
Capacitance ESR	Equivalent series resistance of output capacitance	5	50	mΩ

1. When using the internal voltage regulator control function, the rise time of the 3.3V rail should be at least 5 ms to avoid high current draw on startup.
2. Power supply voltage with ripple should not be below minimum power supply operating range.
3. Applies when using the internal LVR and SVR feature.

11.3.1.1 Power On/Off Sequence

On power-on, after 3.3V reaches 90% of its final value, all voltage rails (1.8V and 1.0V) are allowed 100 ms to reach their final operating values. However, to keep leakage current at a minimum, it is recommended to turn on power supplies almost simultaneously (with delay between supplies at most a few milliseconds).

For power-down, it is recommended to turn off all power rails at the same time and let power supply voltage decay.

Table 11-3 Power Sequencing

Symbol	Parameter	Min	Max	units
T_{3_1}	VCC3P3 (3.3V) power supply stable to VCC1P0 (1.0V) power supply stable		100	ms
T_{3_18}	VCC3P3 (3.3V) power supply stable to VCC1P8 (1.8V) power supply stable		100	ms
T_{18_1}	VCC1P8 (1.8V) power supply stable to VCC1P0 (1.0V) power supply stable	0		ms
Tm-per	3.3V power supply to PE_RST_N de-assertion[1]	100		ms
Tm-ppo	3.3V power supply to MAIN_PWR_OK assertion	0		ms
Tlpg	Power Supplies Stable to LAN_PWR_GOOD assertion	0		ms

Table 11-3 Power Sequencing (Continued)

Tlpg-per	LAN_PWR_GOOD assertion to PE_RST_N de-assertion[1]	100		ms
Tper-m, Tppo-m	PE_RST_N, MAIN_PWR_OK off before 3.3V power supply down	0		ms
Tlpgw	LAN_PWR_GOOD de-assertion time[2]	1		ms

1. If external LAN_PWR_GOOD is used, this time should be kept between LAN_PWR_GOOD assertion and PERST# de-assertion.
2. parameter relevant only if external LAN_PWR_GOOD used.

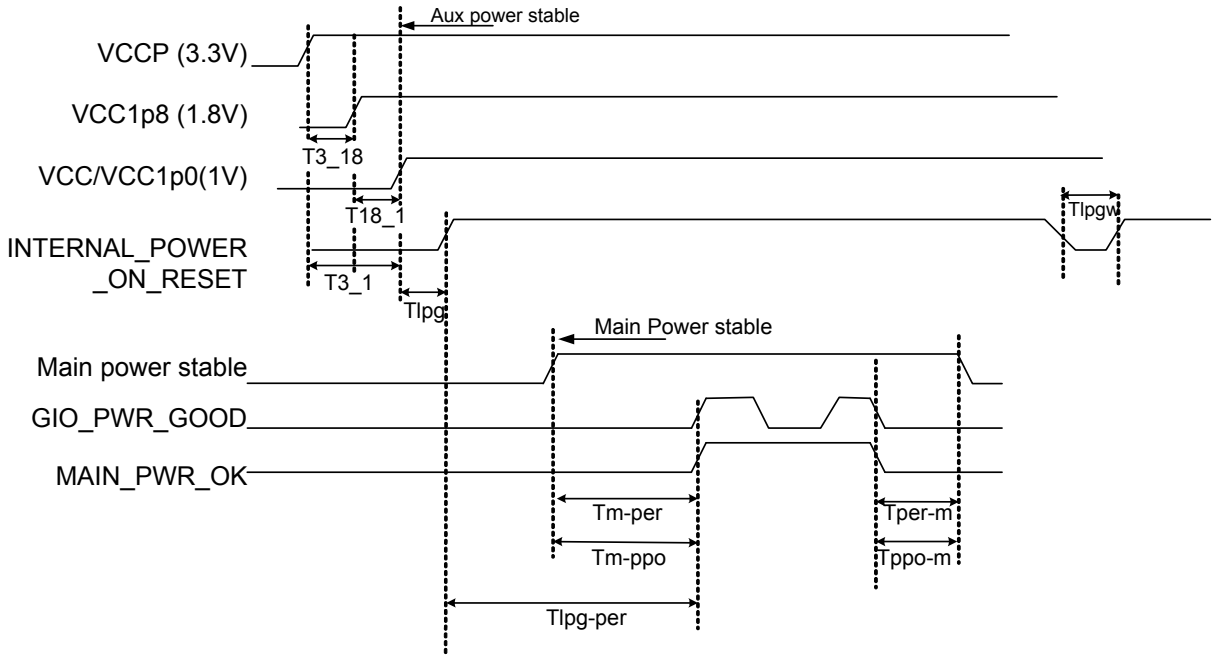


Figure 11-1 Power and Reset Sequencing

11.3.1.2 Power-On Reset Thresholds

The I350 internal Power-on Reset circuitry initiates a full chip reset when voltage levels of VCC3P3 and VCC1P0 power supplies are below certain thresholds. To avoid false power-on reset following power-up, voltage levels that trigger internal power-on reset when power supplies ramp-up and ramp-down differ.

Table 11-4 Power-on Reset Thresholds

Symbol	Parameter	Specifications			Units
		Min	Typ	Max	
V1a	Threshold for 3.3 V power supply in power-up	2.064		2.545	V
V2a	Threshold for 3.3 V power supply in power-down	2.010		2.476	V
V1b	Threshold for 1.0 V power supply in power-up	0.53		0.8	V
V2b	Threshold for 1.0 V power supply in power-down	0.38		0.64	V

The following diagram describes the reset thresholds.

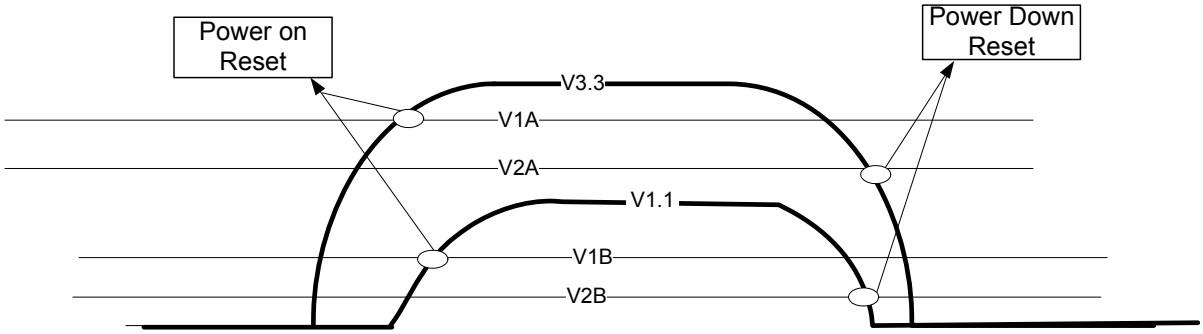


Figure 11-2 Power on Reset Thresholds

11.4 Ball Summary

See Chapter 2 for balls description and ball out map.

11.5 Current Consumption

All the numbers in this section are based on the I350 A1 measurements.

Table 11-5 Power Consumption 4 ports

			No SVR and LVR				SVR and LVR	
Condition	**Speed (MBps)**		**3.3V (mA)**	**1.8V (mA)**	**1.0V (mA)**	**Total power (mW)**	**3.3V (mA)**	**Total power (mW)**
D0a - Active Link	10	Typ	197	149	1186	2106	770	2541
	100	Typ	164	149	1239	2051	753	2485
	1000 Copper	Typ	332	150	1912	3278	1126	3717
		Max	331	150	2218	3871	1214	4372
	1000 Fiber	Typ	136	149	1168	1886	701	2313
		Max	135	149	1571	2467	816	2938
D0a - Idle Link EEE Disabled	No link	Typ	41	146	726	1125	464	1531
	10	Typ	77	146	878	1395	540	1782
	100	Typ	164	146	939	1745	646	2133
	1000 Copper	Typ	337	146	153	2906	1003	3310
	1000 Fiber	Typ	135	146	807	1517	585	1932
D0a - Idle Link EEE Enabled	100	Typ	48	146	731	1154	473	1563
	1000 Copper	Typ	79	146	771	1296	517	1706
D3cold - wake-up enabled on 4 ports	No link	Typ	41	0	317	453	193	637
	10	Typ	77	0	444	699	267	881
	100	Typ	165	0.5	500	1046	374	1234
	100 EEE Enabled	Typ	49	0	322	484	203	670

Table 11-5 Power Consumption 4 ports (Continued)

			No SVR and LVR				SVR and LVR	
Condition	**Speed (MBps)**		**3.3V (mA)**	**1.8V (mA)**	**1.0V (mA)**	**Total power (mW)**	**3.3V (mA)**	**Total power (mW)**
D3cold - wake-up enabled on 1 port only	No link	Typ	42	0	326	465	196	648
	10	Typ	51	0	358	527	213	703
	100	Typ	73	0	372	613	240	794
	100 EEE Enabled	Typ	44	0	327	473	200	660
D3cold-wake disabled (PCIe L3)	No Link	Max	41	1	738	876	320	1056
D0 Uninitialized - Disabled through LAN_DIS_N	No Link	Typ	40	149	991	1392	552	1823
D0 Uninitialized Disabled through DEV_OFF_N	No Link	Typ	31.5	0	241	345	163	538
Manageability with MCTP mode in D3Cold State - Wake disabled.	100	Typ	165	0	505	1049	891	1105
Notes: Typical conditions: room temperature (TA) = 25 C, nominal voltages and continuous network traffic at link speed at full duplex. Maximum conditions: maximum operating temperature (TJ) values, Nominal voltage values and continuous network traffic at link speed at full duplex. PCIe Configured to x4 Gen2 operation.								

Table 11-6 Power Consumption 2 Ports

			No SVR and LVR				SVR and LVR	
Condition	**Speed (MBps)**		**3.3V (mA)**	**1.8V (mA)**	**1.0V (mA)**	**Total power (mW)**	**3.3V (mA)**	**Total power (mW)**
D0a - Active Link	10	Typ	120	89	859	1416	**765**	**2524**
	100	Typ	103	89	886	1388	**875**	**2887**
	1000 Copper	Typ	187	89	1221	2000	**885**	**2920**
		Max	187	90	1558	2526	**886**	**2924**
	1000 Fiber	Typ	136	89	887	1498	554	1830
		Max	136	90	1306	2070	675	2432
D0a - Idle Link EEE disabled	No link	Typ	41	87	636	928	375	1237
	10	Typ	59	87	705.5	1057	413	1363
	100	Typ	103	87	734	1230	466	1538
	1000 Copper	Typ	190	88	1016	1801	645	2128
	1000 Fiber	Typ	136	87	698	1306	494	1630
D0a - Idle Link EEE enabled	100	Typ	45	87	637	943	380	1254
	1000 Copper	Typ	61	87	654	1012	401	1323
D3cold - wake-up enabled on 2 ports	No link	Typ	42	0	323	462	195	643
	10	Typ	60	0	386	584	230	759
	100	Typ	104	0	415	759	284	937
	100 with EEE enabled	Typ	46	0	326	478	200	660

Table 11-6 Power Consumption 2 Ports (Continued)

			No SVR and LVR				SVR and LVR	
Condition	**Speed (MBps)**		**3.3V (mA)**	**1.8V (mA)**	**1.0V (mA)**	**Total power (mW)**	**3.3V (mA)**	**Total power (mW)**
D3cold - wake-up enabled on 1 port only	No link	Typ	42	0	326	465	196	648
	10	Typ	51	0	358	527	213	703
	100	Typ	73	0	372	613	240	794
	100 with EEE enabled	Typ	44	0	327	473	200	660
D3cold-wake disabled (PCIe L3)	No Link	Max	42	1	720	861	318	1049
D0 Uninitialized - Disabled through LAN_DIS_N	No Link	Typ	40	149	991	1392	552	1823
D0 Uninitialized Disabled through DEV_OFF_N	No Link	Typ	31	0	241	345	163	538
Manageability with MCTP mode in D3Cold State - Wake disabled.	100	Typ	104	0	415	758	893	284

Notes:

Typical conditions: room temperature (TA) = 25 C, nominal voltages and continuous network traffic at link speed at full duplex.

Maximum conditions: maximum operating temperature (TJ) values, Nominal voltage values and continuous network traffic at link speed at full duplex.

PCIe configured to x2 Gen2 operation.

Table 11-7 Power Consumption 2 Ports GbE and 2 Ports SerDes

			No SVR and LVR				SVR and LVR	
Condition	**Speed (MBps)**		**3.3V (mA)**	**1.8V (mA)**	**1.0V (mA)**	**Total power (mW)**	**3.3V (mA)**	**Total power (mW)**
D0a - Active Link	10 Copper + 1000 Fiber	Typ	167	149	1178	1998	734	2424
	100 Copper + 1000 Fiber	Typ	151	149	1205	1972	726	2397
	1000 Copper + 1000 Fiber	Typ	234	149.5	1541	2582	912	3011
		Max	234	149	1878	3152	1009	3634
D0a - Idle Link EEE disabled	No link	Typ	89	146	758	1314	524	1731
	10 Copper + 1000 Fiber	Typ	107	146	825	1441	563	1858
	100 Copper + 1000 Fiber	Typ	150	146	852	1610	616	2034
	1000 Copper + 1000 Fiber	Typ	236	146	1150	2194	797	2630
D0a - Idle Link EEE enabled	100 Copper + 1000 Fiber	Typ	92	146	764	1330	529	1747
	1000 Copper + 1000 Fiber	Typ	108	146	783	1402	551	1818

Table 11-7 Power Consumption 2 Ports GbE and 2 Ports SerDes (Continued)

			No SVR and LVR				SVR and LVR	
Condition	**Speed (MBps)**		**3.3V (mA)**	**1.8V (mA)**	**1.0V (mA)**	**Total power (mW)**	**3.3V (mA)**	**Total power (mW)**
D3cold - wake-up enabled on 4 ports	No link	Typ	89	1	355	650	249	823
	10 Copper + 1000 Fiber	Typ	106	1	423	775	288	950
	100 Copper + 1000 Fiber	Typ	150	1	452	949	342	1129
	100 Copper + 1000 Fiber with EEE enabled	Typ	92	1	361	666	295	975
D3cold - wake-up enabled on 1 Cu port only	No link	Typ	90	1	359	658	250	827
	10 Copper	Typ	100	1	390	722	269	889
	100 Copper	Typ	121	1	405	808	296	978
	100 Copper with EEE enabled	Typ	92	1	360	665	253	835
D3cold-wake disabled	No Link	Max	89	1	759	1054	374	1236
D0 Uninitialized - Disabled through LAN_DIS_N	No Link	Typ	124	0	198	607	609.5	2011
D0 Uninitialized Disabled through DEV_OFF_N	No Link	Typ	33	0.5	241.5	351	164	541
Manageability with MCTP mode in D3Cold State - Wake disabled.	100 Copper	Typ	152	0	448	949.6	343	1131.9

Notes:

Typical conditions: room temperature (TA) = 25 C, nominal voltages and continuous network traffic at link speed at full duplex.

Maximum conditions: maximum operating temperature (TJ) values, Nominal voltage values and continuous network traffic at link speed at full duplex.

PCIe Configured to x4 Gen2 operation.

11.6 DC/AC Specification

11.6.1 DC Specifications

11.6.1.1 Digital I/O

Table 11-8 Digital IO DC Electrical Characteristics (Note 3)

Symbol	Parameter	Conditions	Min	Max	Units	Note
VCC3P3	Periphery supply		3.0	3.6	V	
VCC	Core supply		0.93	1.08	V	
VOH	Output High Voltage	IOH = -8mA; VCC3P3 = Min	2.4		V	
		IOH = -100µA; VCC3P3 = Min	VCC3P3–0.2			
VOL	Output Low Voltage	IOL = 8mA; VCC=Min		0.4	V	
		IOL = 100µA; VCC=Min		0.2	V	
VIH	Input High Voltage		2.0	VCC3P3 + 0.3	V	1
VIL	Input Low Voltage		-0.3	0.8	V	1
Iil	Input Current	VCC3P3 = Max; VI =3.6V/GND		+/- 10	µA	
PU	Internal pullup	VIL = 0V	40	150	kΩ	2
	Built-in hysteresis		100		mV	
Cin	Input Pin Capacitance			8	pF	
Vos	Overshoot		N/A	4	V	
Vus	Undershoot		N/A	-0.4	V	

Notes:

1. The input buffer also has hysteresis > 100mV
2. Internal pullup Max characterized at slow corner (125C, VCC3P3=min, process slow); internal pullup Min characterized at fast corner (0C, VCC3P3=max, process fast).
3. Applies to PE_RSTn, SFP0_I2C_CLK, SFP1_I2C_CLK, SFP2_I2C_CLK, SFP3_I2C_CLK, SFP0_I2C_DATA, SFP1_I2C_DATA, SFP2_I2C_DATA, SFP3_I2C_DATA, SRDS_0_SIG_DET, SRDS_1_SIG_DET, SRDS_2_SIG_DET, SRDS_3_SIG_DET LAN_PWR_GOOD DEV_OFF_N, M_PWR_OK, JTCK, JTDI, JTDO, JTMS, RSVD_ARC_JTAG, SDP0[3:0],SDP1[3:0], SDP2[3:0], SDP3[3:0], FLSH_SI, FLSH_SO, FLSH_SCK, FLSH_CE_N, EE_DI, EE_DO, EE_SK, EE_CS_N, LAN0_DIS_N, LAN1_DIS_N, LAN2_DIS_N, LAN3_DIS_N, and AUX_PWR.

11.6.1.2 LEDs I/O

Table 11-9 LED IO DC Electrical Characteristics

Symbol	Parameter	Conditions	Min	Max	Units	Note
VCC3P3	Periphery supply		3.0	3.6	V	
VCC	Core supply		0.93	1.08	V	
VOH	Output High Voltage	IOH = -16mA; VCC3P3 = Min	2.4		V	
VOL	Output Low Voltage	IOL = 16mA; VCC=Min		0.4	V	
VIH	Input High Voltage		2.0	VCC3P3 + 0.3	V	1
VIL	Input Low Voltage		-0.3	0.8	V	1
Iil	Input Current	VCC3P3 = Max; VI =3.6V/GND		+/- 20	µA	
	Built-in hysteresis		100		mV	
Vos	Overshoot		N/A	4	V	
Vus	Undershoot		N/A	-0.4	V	

Notes:
1. The input buffer also has hysteresis > 100mV
2. Applies to LED0[3:0], LED1[3:0], LED2[3:0] and LED3[3:0]

11.6.1.3 Open Drain I/Os

Table 11-10 Open Drain DC Specifications (Note 1, 4)

Symbol	Parameter	Condition	Min	Max	Units	Note
VCC3P3	Periphery supply		3.0	3.6	V	
VCC	Core supply		0.93	1.08	V	
Vih	Input High Voltage		2.1		V	
Vil	Input Low Voltage			0.8	V	
Ileakage	Output Leakage Current	0 < Vin < VCC3P3		+/-10	µA	2
Vol	Output Low Voltage	@ Ipullup		0.4	V	4
Iol	Output Low Current	Vol=0.4V	6		mA	
Cin	Input Pin Capacitance			8	pF	3
Ioffsmb	Input leakage current	VCC3P3 off or floating		+/-10	µA	2

Notes:
1. Applies to SMBD, SMBCLK, SMBALRT _N, PE_WAKE_N and VR_EN pads.
2. Device meets this whether powered or not.
3. Characterized, not tested.
4. OD no high output drive. VOL max=0.4V at 6mA, VOL max=0.2V at 0.1mA

11.6.1.4 NC-SI Input and Output Pads

Table 11-11 NC-SI Pads DC Specifications

Symbol	Parameter	Conditions	Min	Max	Units
VCC3P3	Periphery supply		3.0	3.6	V
VCC	Core supply		0.93	1.08	V
Vabs	Signal voltage range		-0.3	3.765	V
VOH	Output High Voltage	IOH = -4mA; VCC3P3 = Min	2.4		V
VOL	Output Low Voltage	IOL = 4mA; VCC3P3 = Min		0.4	V
VIH	Input High Voltage		2.0		V
VIL	Input Low Voltage			0.8	V
Vihyst	Input hysteresis		100		mV
Iil/Iih	Input Current	VCC3P3 = Max; Vin =3.6V/GND		20	µA
Cin	Input Capacitance			5	pF

Note: Applies to the NC-SI_CLK_OUT, NC-SI_CRS_DV, NC-SI_RXD[1:0], NC-SI_ARB_OUT, NC-SI_TX_EN, NC-SI_TXD[1:0], NC-SI_CLK_IN, NC-SI_ARB_IN.

11.6.2 Digital I/F AC Specifications

11.6.2.1 Reset Signals

The timing between the power up sequence and the different reset signals is described in Figure 11-1 and in Table 11-3.

11.6.2.1.1 LAN_PWR_GOOD

The I350 uses an internal power on detection circuit in order to generate the iLAN_PWR_GOOD signal. Reset can also be implemented when the external power on detection circuit determines that the device is powered up and asserts the LAN_PWR_GOOD signal to reset the device.

11.6.2.2 SMBus

The following table indicates the timing guaranteed when the driver or the agent is performing the action. Where only a typical value is specified, the actual value will be within 2% of the value indicated.

Table 11-12 SMBus Timing Parameters (Master Mode)

Symbol	Parameter	Min	Typ	Max	Units
F_{SMB}	SMBus Frequency		84	100	kHz
T_{BUF}	Time between STOP and START condition driven by the I350	4.7	6.56		µs
$T_{HD:STA}$	Hold time after Start Condition. After this period, the first clock is generated.	4	6.72		µs
$T_{SU:STA}$	Start Condition setup time	4.7			µs
$T_{SU:STO}$	Stop Condition setup time	4	6.88		µs

Table 11-12 SMBus Timing Parameters (Master Mode) (Continued)

Symbol	Parameter	Min	Typ	Max	Units
$T_{HD:DAT}$	Data hold time	0.3	0.48		µs
$T_{SU:DAT}$	Data setup time	0.25			µs
$T_{TIMEOUT}$	Detect SMBClk low timeout	26.2		31.5	ms
T_{LOW}	SMBClk low time	4.7	5.76		µs
T_{HIGH}	SMBClk high time	4	6.56		µs

The following table indicates the timing requirements of the I350 when it is the receiver of the indicated signal.

Table 11-13 SMBus Timing Parameters (Slave Mode)

Symbol	Parameter	Min 100KHz[1]	Min 400KHz[2]	Max	Units
F_{SMB}	SMBus Frequency	10	10	400	kHz
T_{BUF}	Time between STOP and START condition driven by the I350.	4.7	1.3		µs
$T_{HD:STA}$	Hold time after Start Condition. After this period, the first clock is generated.	4	0.6		µs
$T_{SU:STA}$	Start Condition setup time	4.7	0.6		µs
$T_{SU:STO}$	Stop Condition setup time	4	0.6		µs
$T_{HD:DAT}$	Data hold time	300	100		ns
$T_{SU:DAT}$	Data setup time	250	100		ns
T_{LOW}	SMBClk low time	4.7	1.3		µs
T_{HIGH}	SMBClk high time	4	0.6		µs

1. Specifications based on SMBus specification
2. Specifications based on I^2C specification for Fast-mode (400 KHz)

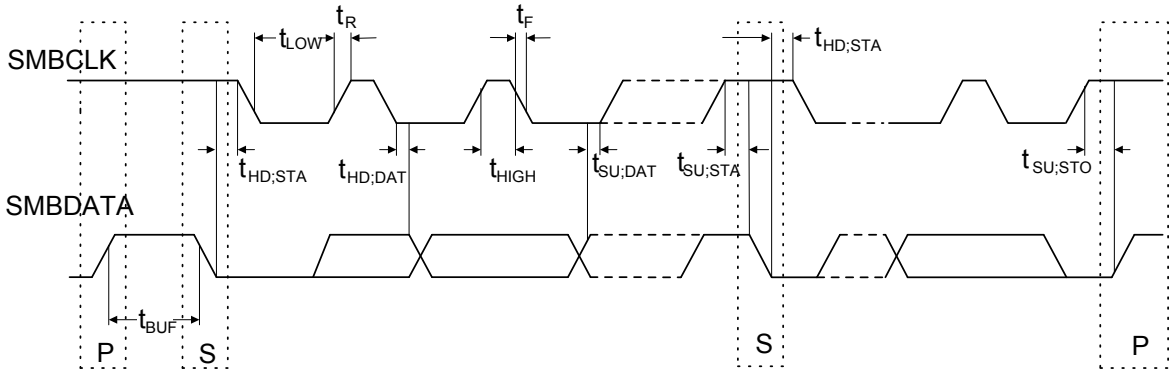


Figure 11-3 SMBus I/F Timing Diagram

11.6.2.3 I²C AC Specification

The following table indicates the timing of the I2C_CLK and I2C_DATA pins when operating in I²C mode.

Table 11-14 I²C Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units
F_{SCL}	I2C_CLK Frequency			100	kHz
T_{BUF}	Time between STOP and START condition driven by the I350	4.7			µs
$T_{HD:STA}$	Hold time after Start Condition. After this period, the first clock is generated.	4			µs
$T_{SU:STA}$	Start Condition setup time	4.7			µs
$T_{SU:STO}$	Stop Condition setup time	4			µs
$T_{HD:DAT}$	Data hold time	50[1]			ns
$T_{SU:DAT}$	Data setup time	0.25			µs
T_{LOW}	I2C_CLK low time	4.7			µs
T_{HIGH}	I2C_CLK high time	4			µs

1. According to Atmel's AT24C01A/02/04 definition of the 2 wires interface.

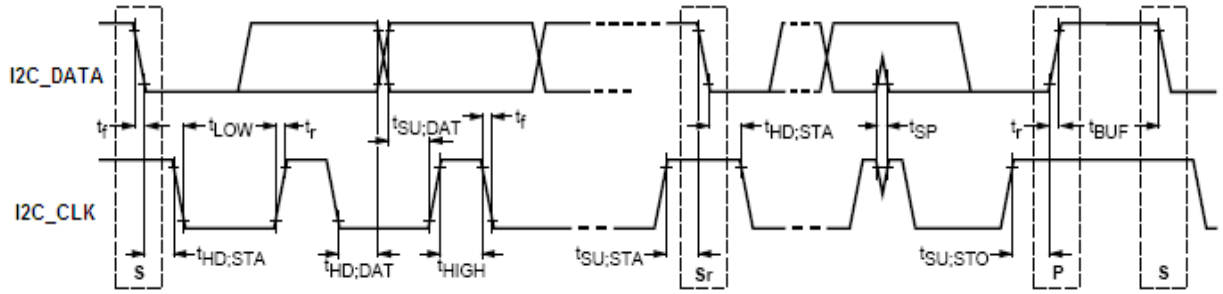


Figure 11-4 I²C I/F Timing Diagram

11.6.2.4 FLASH AC Specification

The I350 is designed to support a serial flash. Applicable over the recommended operating range from Ta = -40C to +85C, VCC3P3 = 3.3V, Cload = 1 TTL Gate and 16 pF (unless otherwise noted). For FLASH I/F timing specification Table 11-15 and Figure 11-5.

Table 11-15 FLASH I/F Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units	Note
t_{SCK}	SCK clock frequency	0	15.625	20	MHz	[1]
t_{RI}	Input rise time		2.5	20	ns	
t_{FI}	Input fall time		2.5	20	ns	
t_{WH}	SCK high time	20	32		ns	[2]

Table 11-15 FLASH I/F Timing Parameters (Continued)

Symbol	Parameter	Min	Typ	Max	Units	Note
t_{WL}	SCK low time	20	32		ns	[2]
t_{CS}	CS high time	25			ns	
t_{CSS}	CS setup time	25			ns	
t_{CSH}	CS hold time	25			ns	
t_{SU}	Data-in setup time	5			ns	
t_{H}	Data-in hold time	5			ns	
t_{V}	Output valid			20	ns	
t_{HO}	Output hold time	0			ns	
t_{DIS}	Output disable time			100	ns	

Notes:

1. Clock is 62.5MHz divided by 4. In Bit Banging mode maximum allowable frequency is 20MHz
2. 45% to 55% duty cycle.

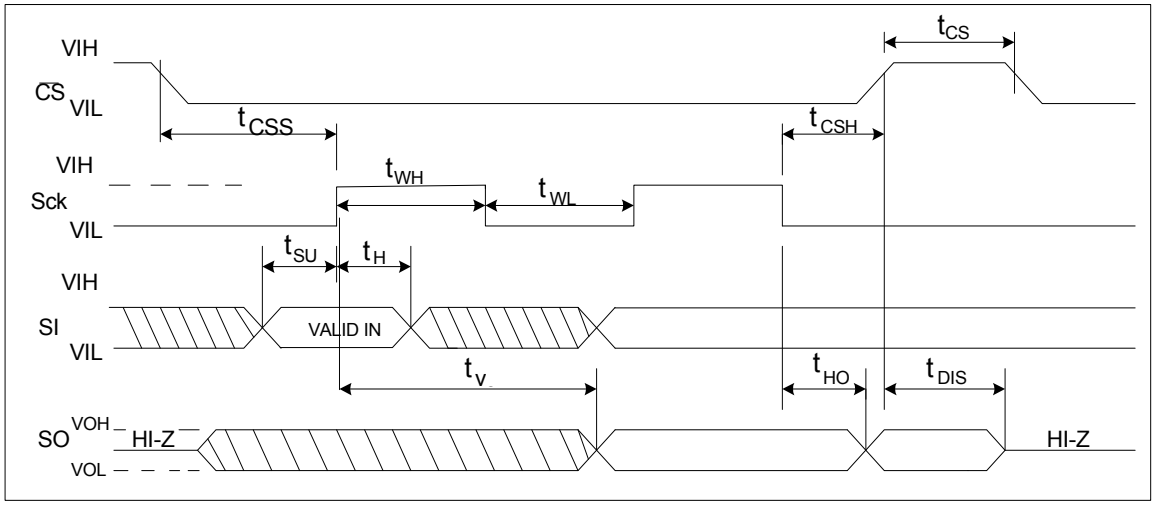


Figure 11-5 Flash Timing Diagram

11.6.2.5 EEPROM AC Specification

The I350 is designed to support a standard serial EEPROM. Applicable over recommended operating range from Ta = -40C to +85C, VCC3P3 = 3.3V, Cload = 1 TTL Gate and 16pF (unless otherwise noted). For EEPROM I/F timing specification see Table 11-16 and Figure 11-6.

Table 11-16 EEPROM I/F Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units	Note
t_{SCK}	SCK clock frequency	0	2	2.1	MHz	[1]
t_{RI}	Input rise time			2	µs	
t_{FI}	Input fall time			2	µs	
t_{WH}	SCK high time	200	250		ns	[2]
t_{WL}	SCK low time	200	250		ns	
t_{CS}	CS high time	250			ns	
t_{CSS}	CS setup time	250			ns	
t_{CSH}	CS hold time	250			ns	
t_{SU}	Data-in setup time	50			ns	
t_{H}	Data-in hold time	50			ns	
t_{V}	Output valid	0		200	ns	
t_{HO}	Output hold time	0			ns	
t_{DIS}	Output disable time			250	ns	

Notes:

1. Clock is 2MHz
2. 45% to 55% duty cycle.

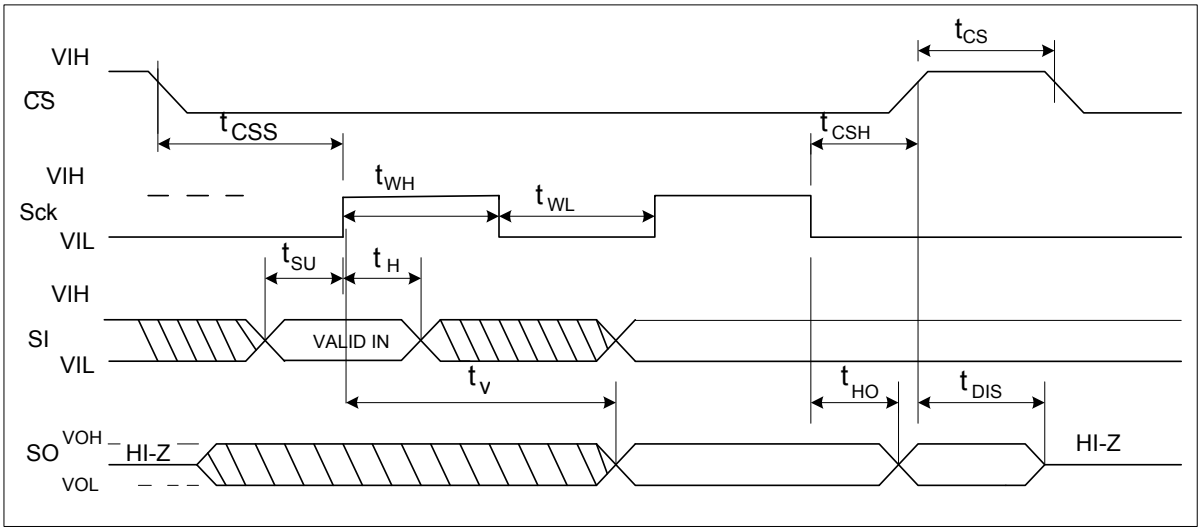


Figure 11-6 EEPROM Timing Diagram

11.6.2.6 NC-SI AC Specification

The I350 is designed to support the standard DMTF NC-SI interface. For NC-SI I/F timing specification see Table 11-17 and Figure 11-7.

Table 11-17 NC-SI AC Specifications

Symbol	Parameter	Min	Typ	Max	Units	Notes
Tckf	NCSI_CLK_IN Frequency		50		MHz	2
Rdc	NCSI_CLK_IN Duty Cycle	35		65	%	1
Racc	NCSI_CLK_IN accuracy			100	ppm	
Tco	Clock-to-out (10 pF =< cload <=50 pF) NCSI_RXD[1:0], NCSI_CRS_DV and NCSI_ARB_OUT Data valid from NCSI_CLK_IN rising edge	2.5		12.5	ns	4
Tsu	NCSI_TXD[1:0], NCSI_TX_EN and NCSI_ARB_IN Data Setup to NCSI_CLK_IN rising edge	3			ns	
Thold	NCSI_TXD[1:0], NCSI_TX_EN Data hold from NCSI_CLK_IN rising edge	1			ns	
Tor	NCSI_RXD[1:0], NCSI_CRS_DV and NCSI_ARB_OUT Output Time rise	0.5		6	ns	3
Tof	NCSI_RXD[1:0], NCSI_CRS_DV and NCSI_ARB_OUT Output Time fall	0.5		6	ns	3
Tckr/Tckf	NCSI_CLK_IN Rise/Fall Time	0.5		3.5	ns	
Tckor/Tckof	NCSI_CLK_OUT Rise/Fall Time	0.5		3.5	ns	5

Notes:

1. Clock Duty cycle measurement: High interval measured from Vih to Vil points, Low from Vil to next Vih.
2. Clock interval measurement from Vih to Vih.
3. Cload = 25 pF.
4. This timing relates to the output pins, while Tsu and Thd relate to timing at the input pins
5. 10 pF =< Cload <= 30 pF

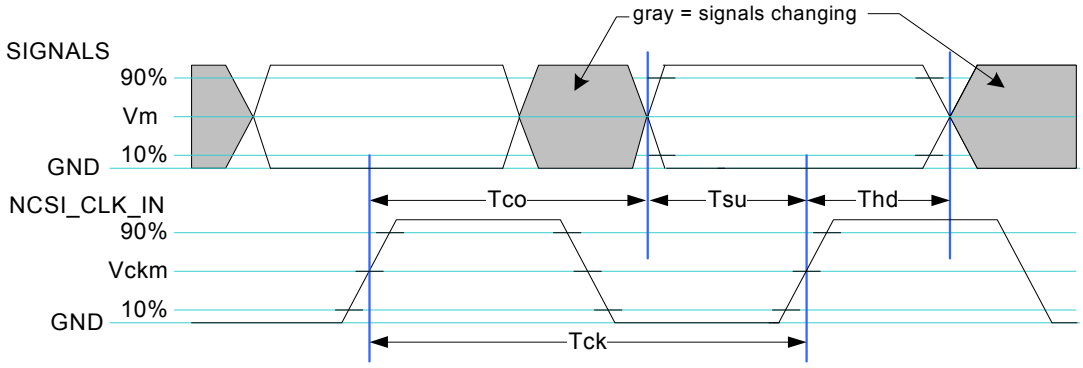


Figure 11-7 NC-SI Timing Diagram

11.6.2.7 JTAG AC Specification

The I350 is designed to support the IEEE 1149.1 standard. Following timing specifications are applicable over recommended operating range from Ta = 0°C to +70°C, VCC3P3 = 3.3V, Cload = 16pF (unless otherwise noted). For JTAG I/F timing specification see Table 11-18 and Figure 11-8.

Table 11-18 JTAG I/F Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units	Note
t_{JCLK}	JTCK clock frequency			10	MHz	
t_{JH}	JTMS and JTDI hold time	10			nS	
t_{JSU}	JTMS and JTDI setup time	10			nS	
t_{JPR}	JTDO propagation Delay			15	nS	

Notes:
1. The table above applies to JTCK, JTMS, JTDI and JTDO.
2. Timing measured relative to JTCK reference voltage of VCC3P3/2.

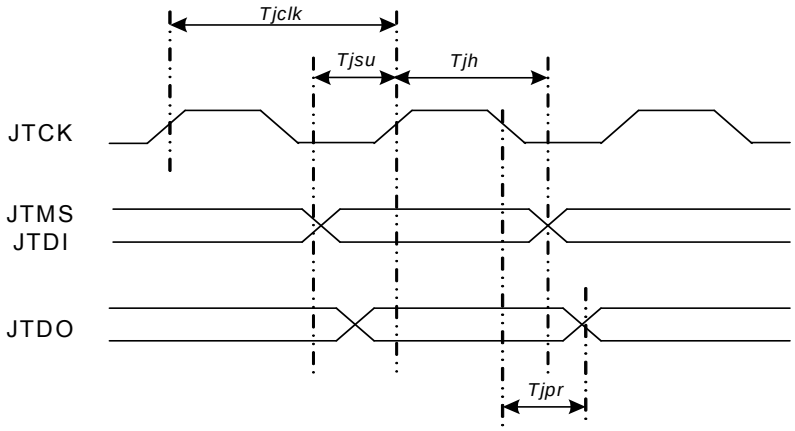


Figure 11-8 JTAG AC Timing Diagram

11.6.2.8 MDIO AC Specification

The I350 is designed to support the MDIO specifications defined in IEEE 802.3 clause 22. Following timing specifications are applicable over recommended operating range from Ta = 0°C to +70°C, VCC3P3 = 3.3V, Cload = 16pF (unless otherwise noted). For MDIO I/F timing specification see Table 11-19, Figure 11-9 and Figure 11-10.

Table 11-19 MDIO I/F Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units	Note
t_{MCLK}	MDC clock frequency			2	MHz	
t_{MH}	MDIO hold time	10			nS	

Table 11-19 MDIO I/F Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units	Note
t_{MSU}	MDIO setup time	10			nS	
t_{MPR}	MDIO propagation Delay	10		300	nS	

Notes:

1. The table above applies to MDIO0, MDC0, MDIO1, MDC1, MDIO2, MDC2, MDIO3, and MDC3.
2. Timing measured relative to MDC reference voltage of 2.0V (Vih).

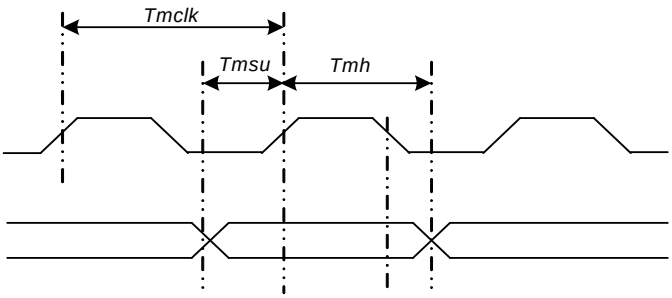


Figure 11-9 MDIO Input AC Timing Diagram

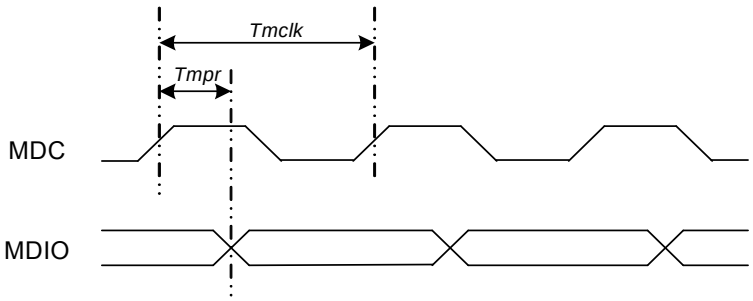


Figure 11-10 MDIO Output AC Timing Diagram

11.6.2.9 SFP 2 Wires I/F AC Specification

According to Atmel's AT24C01A/02/04 definition of the 2 wires I/F bus.

11.6.2.10 PCIe/SerDes DC/AC Specification

The transmitter and receiver specification are given per PCIe Card Electromechanical Specification rev 2.0.

11.6.2.11 PCIe Specification - Receiver

Specifications are from the PCIe v2.1 (2.5GT/s and 5GT/s) specification.

11.6.2.12 PCIe Specification - Transmitter

Specifications are from the PCI Express* 2.0 (5Gbps or 2.5Gbps) specification.

11.6.2.13 PCIe Specification - Input Clock

The input clock for PCIe must be a differential input clock in frequency of 100 MHz. For full specifications please check the PCI-Express Card Electromechanical specifications (refclk specifications).

11.6.3 Serdes DC/AC Specification

The Serdes interface supports the following standards:

1. PICMG 3.1 specification Rev 1.0 1000BASE-BX.
2. 1000BASE-KX electrical specification defined IEEE802.3ap clause 70.
3. SGMII on 1000BASE-BX or 1000BASE-KX compliant electrical interface (AC coupling with internal clock recovery).
4. SFP (Small Form factor Pluggable) Transceiver Rev 1.0

11.6.4 PHY Specification

The specifications define the interface for the back-plane board connection, Interface to external 1000BASE-T PHY and the interface to fiber or SFP module.

DC/AC specification is according to Standard 802.3 and 802.3ab.

100 Base-T parameters are also described in standard ANSI X3.263.

11.6.5 XTAL/Clock Specification

The 25 MHz reference clock of the I350 can be supplied either from a crystal or from an external oscillator. The recommended solution is to use a crystal.

11.6.5.1 Crystal Specification

Table 11-20 Specification for External Crystal

Parameter Name	Symbol	Recommended Value	Conditions
Frequency	f_o	25.000 [MHz]	@25 [°C]
Vibration mode		Fundamental	
Cut		AT	
Operating /Calibration Mode		Parallel	
Frequency Tolerance @25°C	$\Delta f/f_o$ @25°C	±30 [ppm]	@25 [°C]
Temperature Tolerance	$\Delta f/f_o$	±30 [ppm]	
Operating Temperature	T_{opr}	-20 to +70 [°C]	
Non Operating Temperature Range	T_{opr}	-40 to +90 [°C]	
Equivalent Series Resistance (ESR)	R_S	50 [Ω] maximum	@25 [MHz]
Shunt Capacitance	C_o	6 [pF] maximum	
Load Capacitance	C_{load}	20 pF	
Pullability from Nominal Load Capacitance	$\Delta f/C_{load}$	15 [ppm/pF] maximum	
Max Drive Level	D_L	0.5 [mW]	
Insulation Resistance	IR	500 [MΩ] minimum	@ 100V DC
Aging	$\Delta f/f_o$	±5 [ppm/year]	
Damp Capacitor	Cd	10 [pF]	
External Capacitors	C_1, C_2	36 [pF] to 40 [pF]	
Board Resistance	R_S	0.1 [Ω]	

11.6.5.2 External Clock Oscillator Specification

When using an external oscillator the following connection must be used:

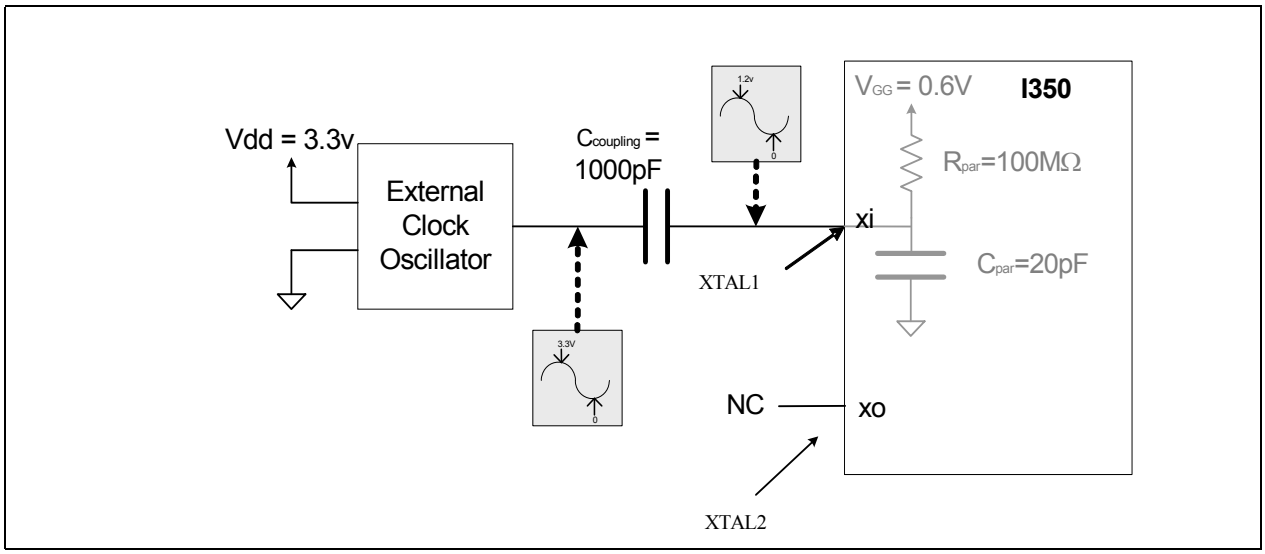


Figure 11-11 External Clock Oscillator Connectivity to the I350

Table 11-21 Specification for External Clock Oscillator

Parameter Name	Symbol	Value	Conditions
Frequency	f_o	25.0 [MHz]	@25 [°C]
External OSC Supply Swing	V_{p-p}	3.3 ± 0.3 [V]	
Frequency Tolerance	$\Delta f/f_o$	±50 [ppm]	-20 to +70 [°C]
Operating Temperature	T_{opr}	-20 to +70 [°C]	
Aging	$\Delta f/f_o$	±5 ppm per year	

11.6.6 GbE PHY GE_REXT Bias Connection

For the PHY circuit, an external resistor of 3.01KΩ (accuracy 1%) is used as reference for the internal bias currents. This resistor is connected to the GE_REXT ball and GND as shown in Figure 11-12.

Short connections for this resistor are compulsory.

Place the resistor as close as possible to the device (less than 1").

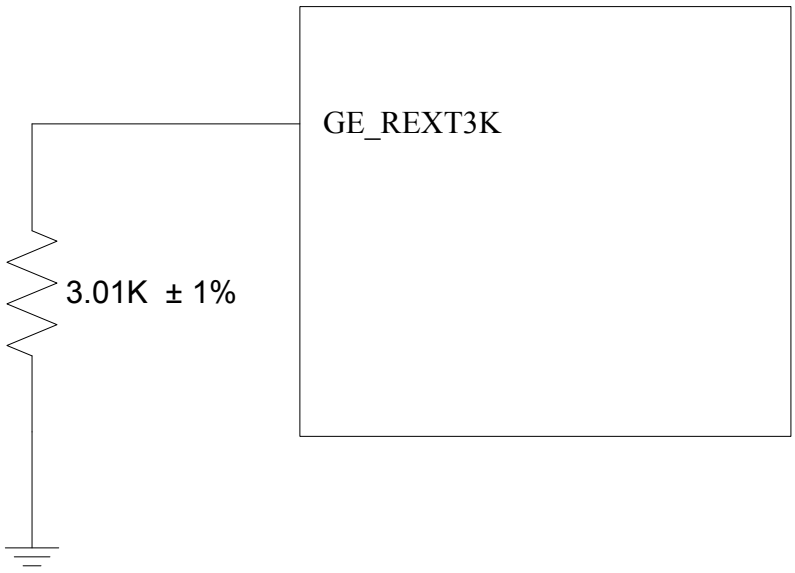


Figure 11-12 GbE PHY Bias Connection

11.6.7 SerDes SE_RSET Bias Connection

For the SerDes circuit, an external resistor of 2.37KΩ (accuracy 1%) is used as reference for the internal bias currents. This resistor is connected to the SE_RSET ball and GND as shown in Figure 11-13.

Short connections for this resistor are compulsory.

Place the resistor as close as possible to the device (less than 1").

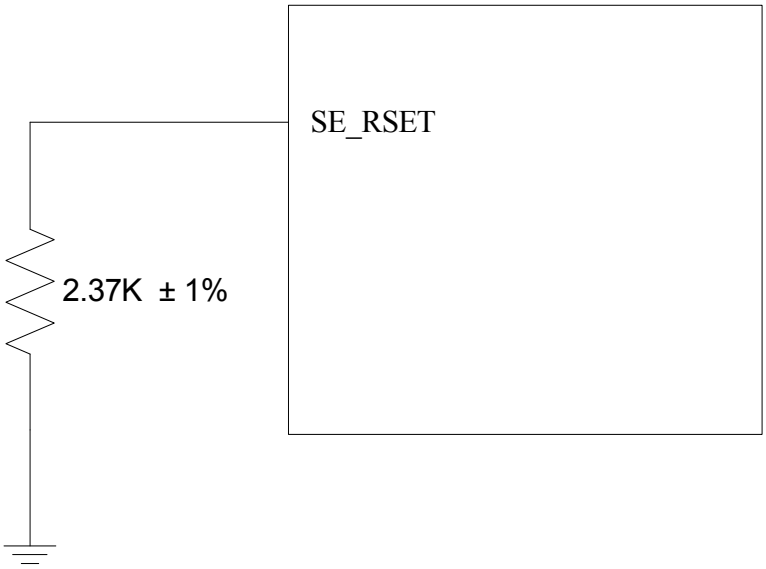


Figure 11-13 SerDes Bias Connection

11.6.8 PCIe PE_TRIM Bias Connection

For the PCIe SerDEs circuit, an external resistor of 1.5KΩ (accuracy 1%) is used as reference for the internal bias currents. This resistor is connected between the PE_TRIM1 and PE_TRIM2 balls as shown in Figure 11-14.

Short connections for this resistor are compulsory. Place the resistor as close as possible to the device (less than 1").

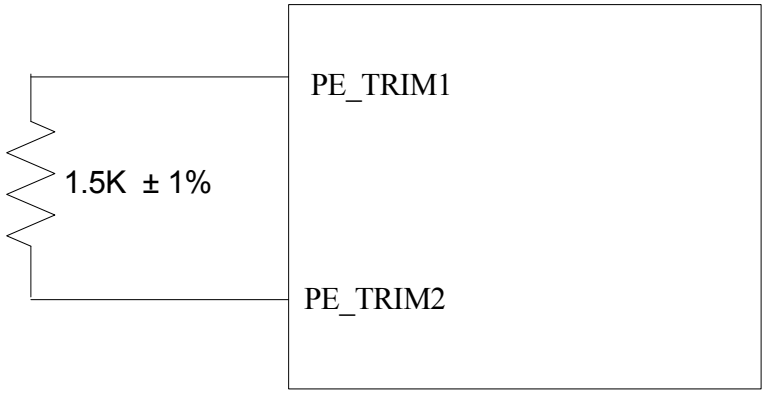


Figure 11-14 PCIe Bias Connection

11.6.9 Voltage Regulator Electrical Specifications

To reduce BOM cost the I350 supports generation of the 1.0V power supply from the 3.3V supply using an on-chip SVR (Switched Voltage Regulator) control circuit with an external inductor, PFET and NFET matched power transistors and some additional discrete components. The I350 also supports generation of the 1.8V power supply from the 3.3V power supply using a an on-chip LVR (Linear Voltage Regulator) control circuit with an external low cost BJT transistor (for additional information see Section 3.5).

11.6.9.1 1.0V SVR Electrical Specifications

Following table describes electrical performance of the 1.0V SVR when using the components specified in section Section 11.9.1.

Parameter	Min	Typ	Max	Unit	Comments
Regulator input voltage	2.9	3.3	3.7	V	
Junction Temperature	-25		125	°C	
Regulator output voltage		1.0		V	Programmable Output Voltage
Output Voltage Accuracy	-3		+3	%	Not including line and load regulation errors.
Load Current Load Current Load Current	0.05		4.0 1.0 0.8	A A A	Average Value, in PWM Mode Average Value, in PFM Mode Average Value, in Start-Up
Load regulation		0.1		%/A	Output voltage droop versus load current
Line regulation		0.2		%/V	Output voltage change verses input supply voltage (VCC3P3)
Conversion Efficiency	80			%	Depends on load current and external FETs/ Inductor
Output Filter Capacitor Value (Cout)	22		110	µF	Tolerance Limit of ±10% Use a larger inductor to reduce this value
Output Filter Capacitor ESR		5	50	mΩ	
Filter Inductor Value (L1)	1		2	µH	Sets inductor ripple current (along with switching frequency). Tolerance Limit of ± 20%. Use a larger output filter capacitor to reduce this value
Output Filter Inductor DCR			10	mΩ	Reduce DC resistance for improved efficiency
Output Filter Inductor saturation current	6			A	
Input Capacitor Value	20		110	µF	Capacitor on VCC3P3 supply pin
Top-side switch (PFET) on-resistance			40	mΩ	VCC3P3 = 2.9V. Reduce for improved efficiency
Bottom-side switch (NFET) on-resistance			20	mΩ	VCC3P3 = 2.9V Reduce for improved efficiency
Output overshoot/undershoot	-7		7	%	
Output Voltage Ripple during normal operation. Output Voltage Ripple at start-up			20 100	mV_{p-p} mV_{p-p}	Set by inductor ripple current and output capacitor Set by Hysteresis, ESR and Load Current

11.6.9.2 SVR Efficiency

Following graph depicts SVR efficiency as a function of 1.0V current consumption.

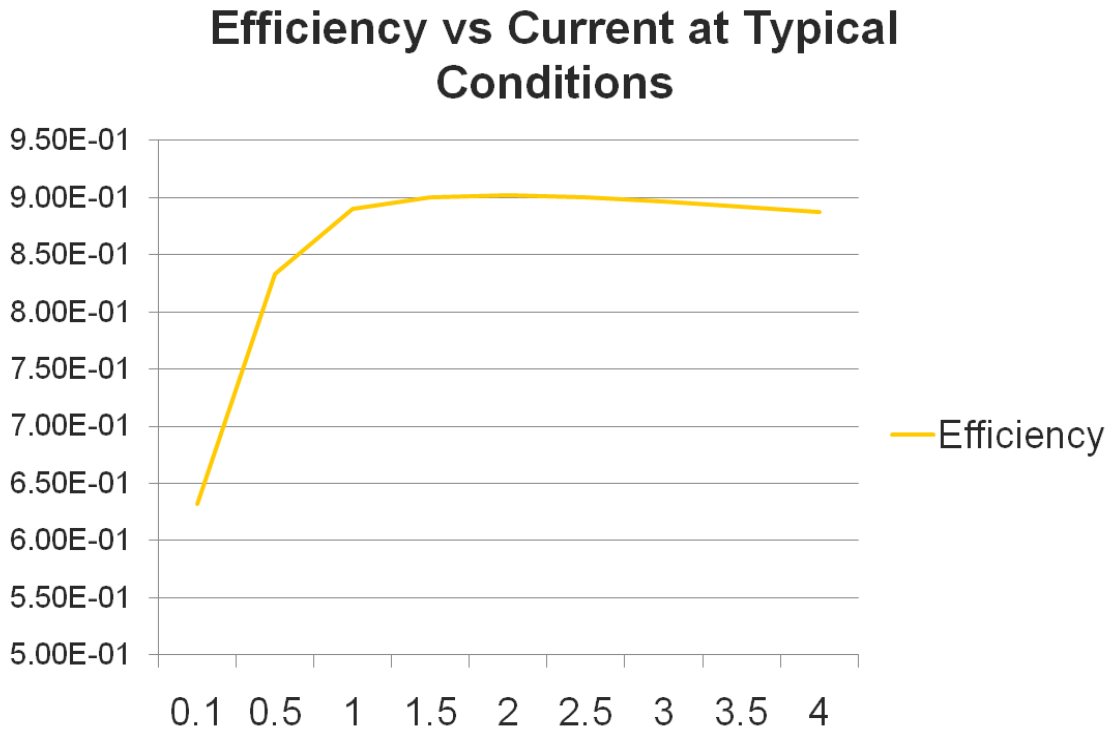


Figure 11-15 SVR Efficiency

11.6.9.3 1.8V LVR Electrical Specifications

Following table describes electrical performance of the 1.8V LVR using the components specified in section Section 11.9.2.

Table 11-22 LVR DC Specifications

Parameter	Min	Typ	Max	Unit	Comments
Regulator input voltage	2.9	3.3	3.7	V	
Junction Temperature	-25		125	°C	
Regulator output voltage	1.8			V	
Output Voltage Accuracy	-3.5		+3.5	%	Not including line and load regulation errors.
Output load capacitance	22			µF	For loop stability. To reduce capacitance ESR use multiple capacitors.
Load capacitance tolerance	-5		+5	%	For wider tolerance increase minimum load capacitance to compensate.
PNP beta	85	200	375		At I_C = 0.5A

Table 11-22 LVR DC Specifications

Parameter	Min	Typ	Max	Unit	Comments
PNP transition frequency	40	120	200	MHz	f_T
Load regulation		-2	-6	mV/A	Output voltage droop versus load current
Line regulation		0.02	0.1	%/V	Output voltage change verses input supply voltage (VCC3P3)
Load Current	10	100	600	mA	
HIZ Leakage			10	µA	On LVR_1P8_CTRL pin with 1.8V load when VR_EN pin is connected to ground.
Load dependent current	$I_{LOAD}/\beta PNP$			mA	Sunk by LVR_1P8_CTRL pin

Table 11-23 1.8V LVR AC Specifications

Parameter	Min	Typ	Max	Unit	Comments
Power supply rejection		58		dB	VCC3P3, f_{supply} < 10 KHz
Power supply rejection		34		dB	VCC3P3, 10 KHz < f_{supply} < 1 MHz
Power supply rejection		30		dB	VCC3P3, MHz < f_{supply} < 100 MHz
Output voltage overshoot		5		%	Max to min load step
Output voltage undershoot		5		%	Max to min load step
Power up time		20		µS	
Output voltage overshoot		2		%	During power up

11.7 Package

The I350 is assembled in 2 packages:

1. A 17x17 PBGA package that's footprint compatible with 82580.
2. A 25x25 PBGA package with a dual 10/100/1000BASE-T copper interface.

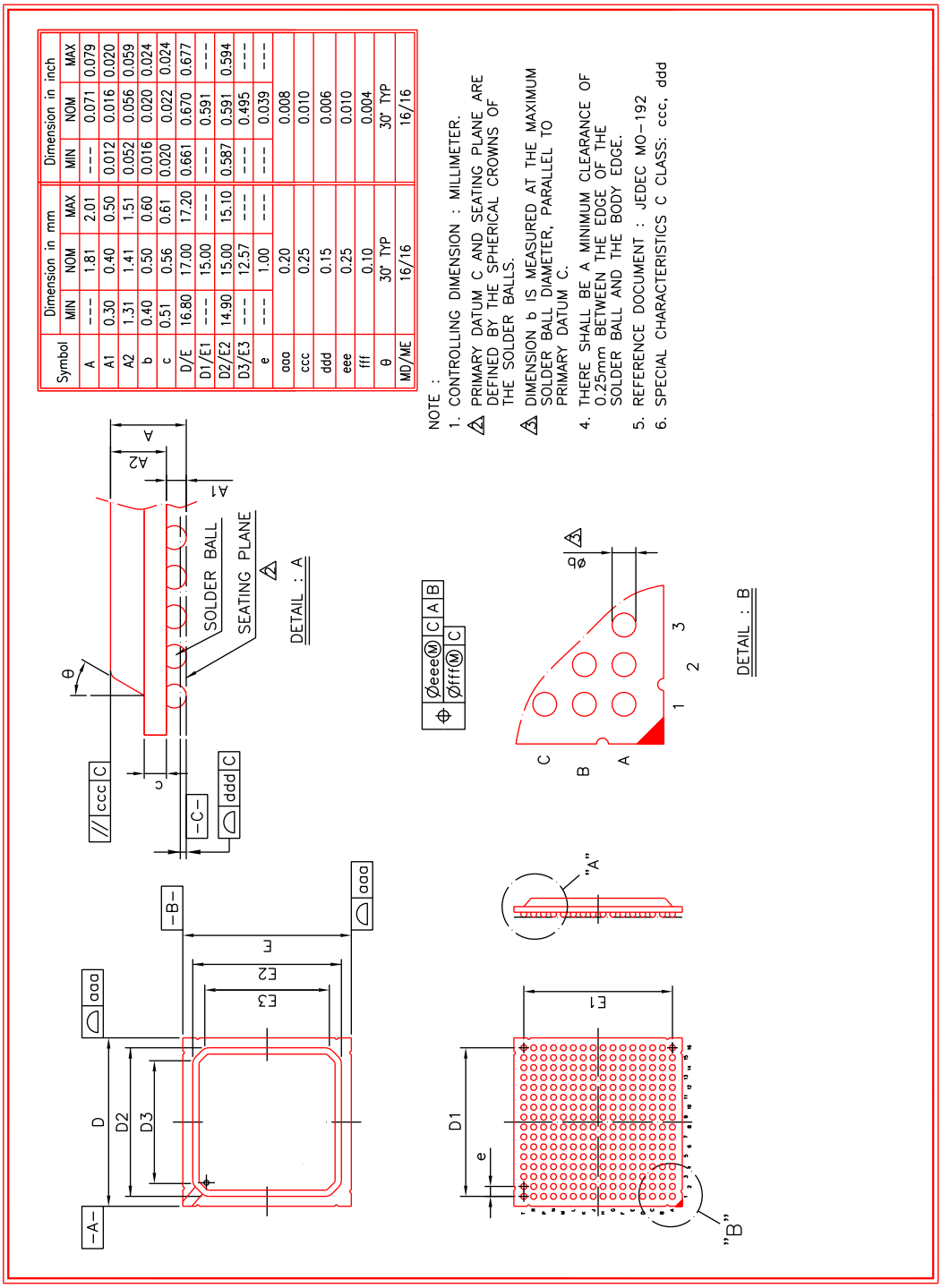
11.7.1 Mechanical Specification for the 17x17 PBGA Package.

Table 11-24 I350 17x17 Package Mechanical Specifications

Body Size	Ball Count	Ball Pitch	Ball Matrix	Substrate
17x17	256	1.0 mm	16 x 16 array, fully populated	4 layers

Note: The I350 uses the P-free SAC305 solder ball (P<2 ppm) and WF6063M5 ball attach flux. The package pad copper size is 0.6 mm in diameter. The solder mask opening is 0.45 mm in diameter.

11.7.1.1 17x17 PBGA Package Schematics



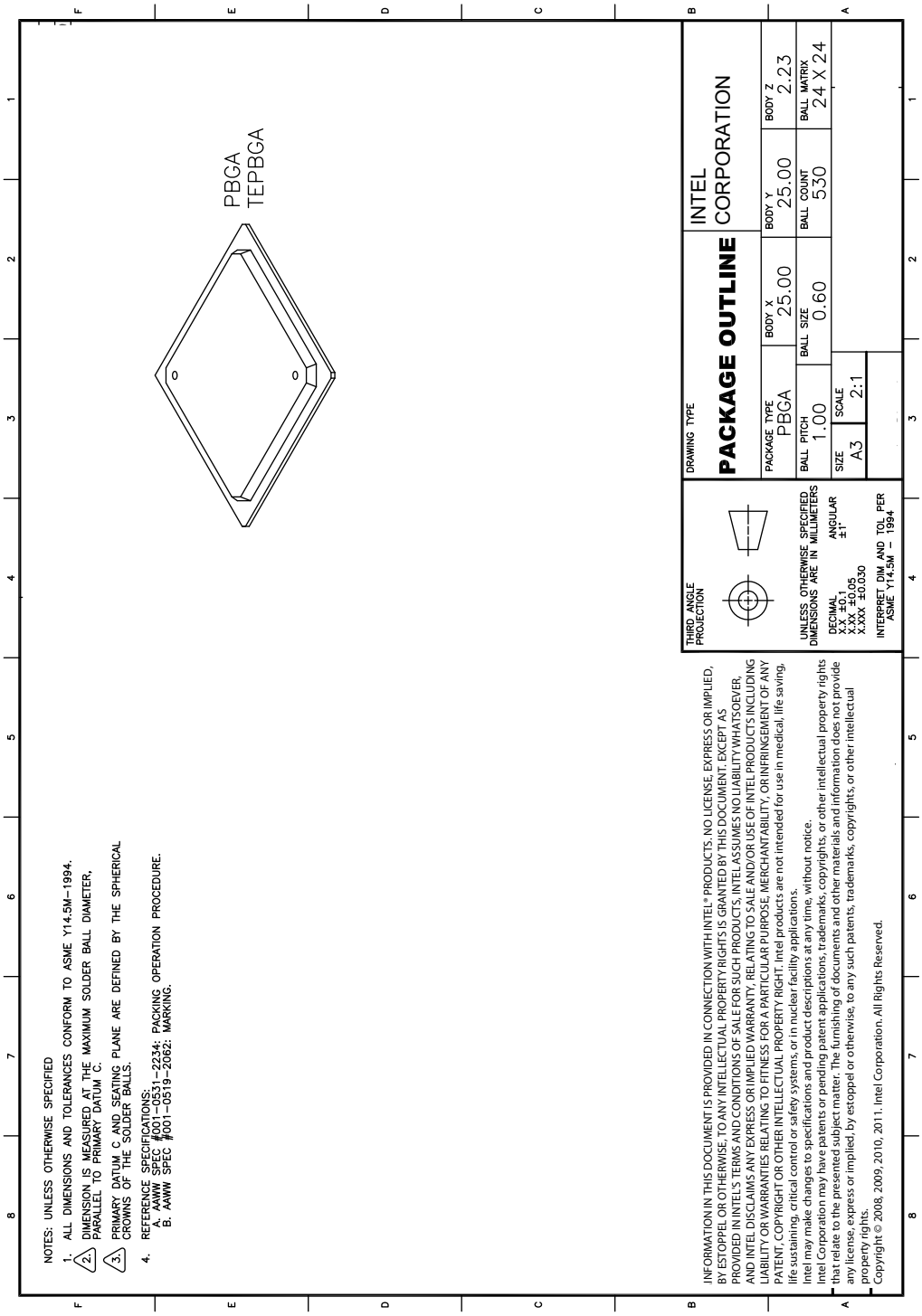
Symbol	Dimension in mm			Dimension in inch		
	MIN	NOM	MAX	MIN	NOM	MAX
A	---	1.81	2.01	---	0.071	0.079
A1	0.30	0.40	0.50	0.012	0.016	0.020
A2	1.31	1.41	1.51	0.052	0.056	0.059
b	0.40	0.50	0.60	0.016	0.020	0.024
c	0.51	0.56	0.61	0.020	0.022	0.024
D/E	16.80	17.00	17.20	0.661	0.670	0.677
D1/E1	---	15.00	---	---	0.591	---
D2/E2	14.90	15.00	15.10	0.587	0.591	0.594
D3/E3	---	12.57	---	---	0.495	---
e	---	1.00	---	---	0.039	---
aaa		0.20			0.008	
ccc		0.25			0.010	
ddd		0.15			0.006	
eee		0.25			0.010	
fff		0.10			0.004	
θ		30° TYP			30° TYP	
MD/ME		16/16			16/16	

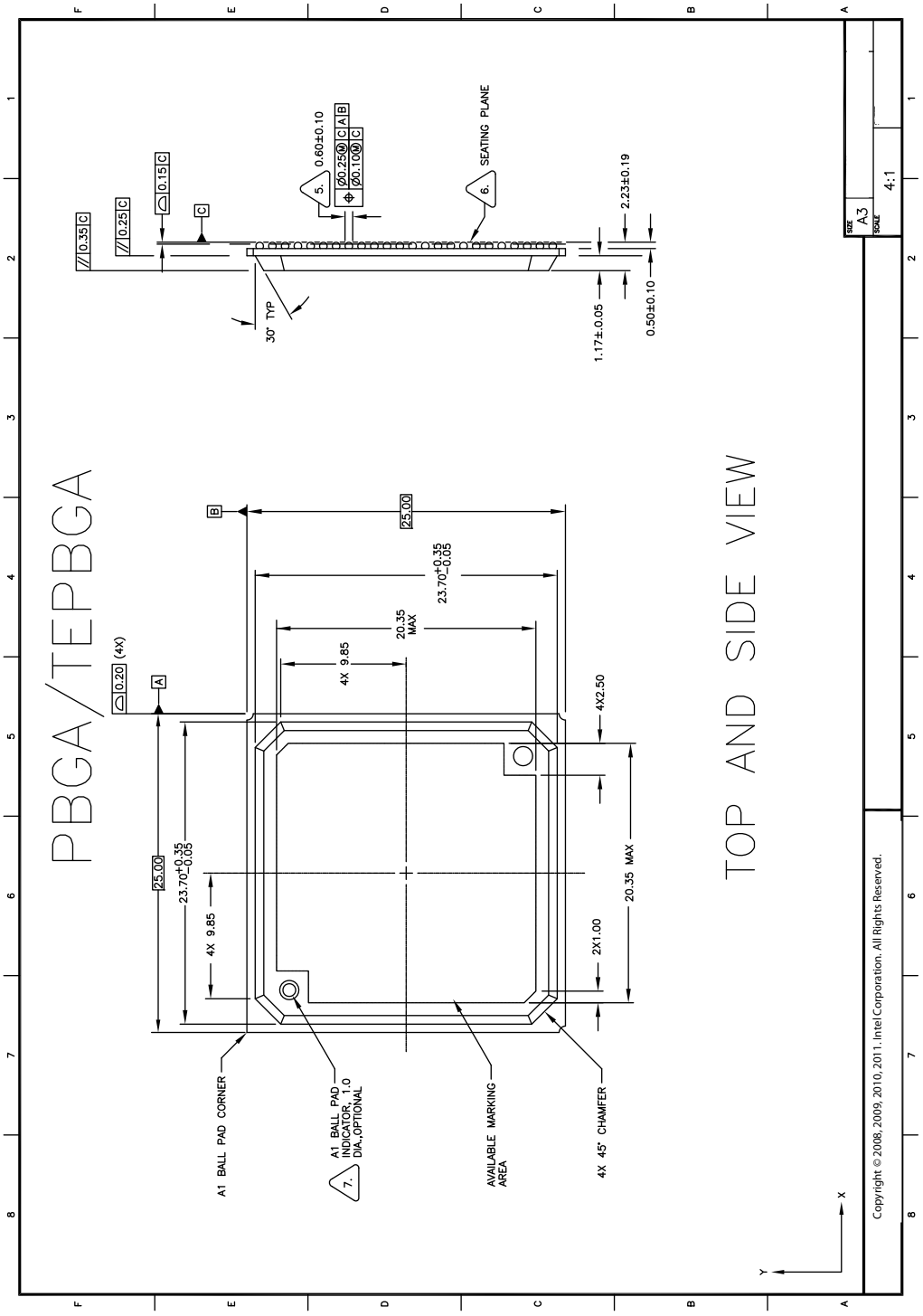
11.7.2 Mechanical Specification for the 25x25 PBGA Package

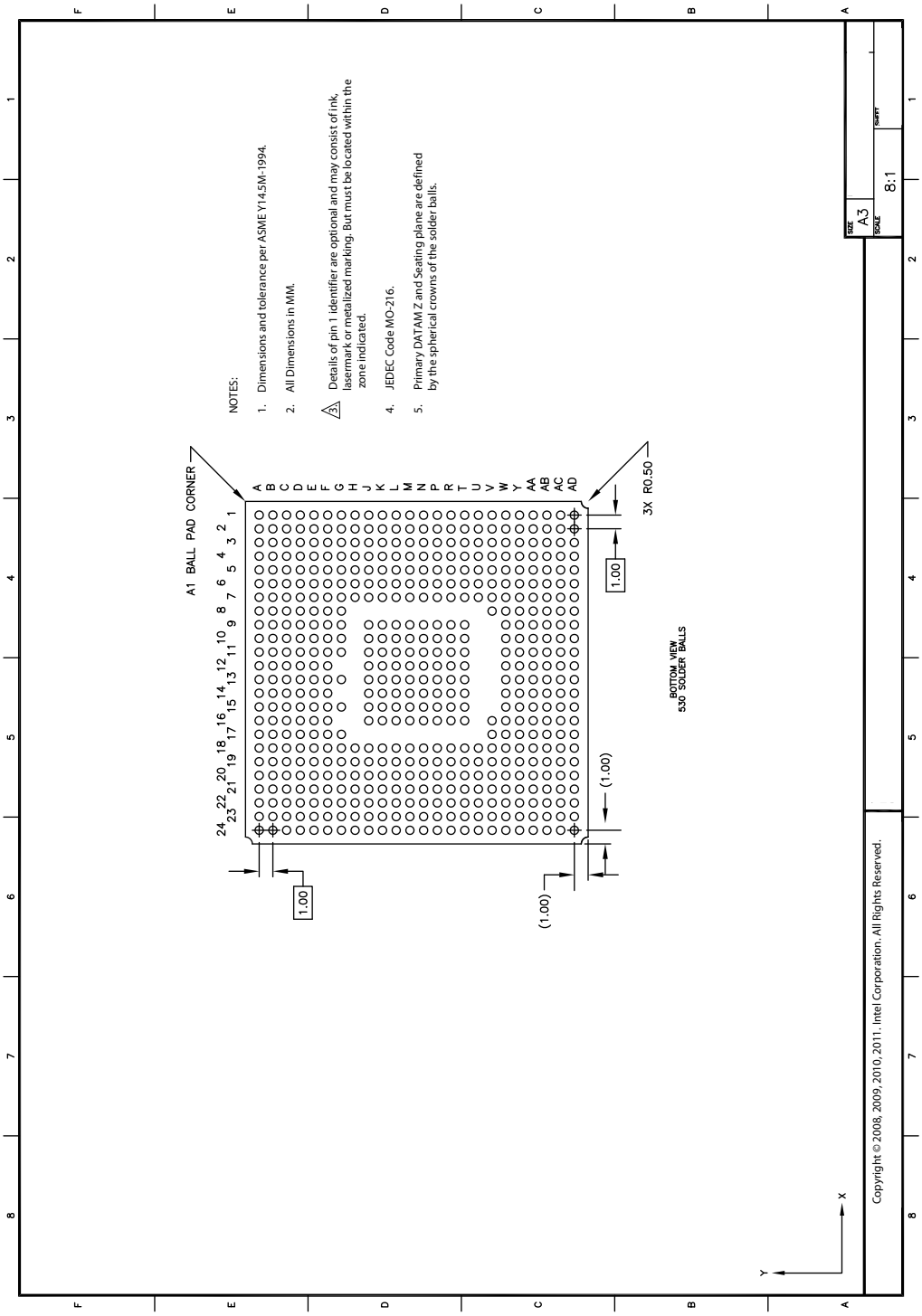
Table 11-25 I350 25x25 Package Mechanical Specifications

Body Size	Ball Count	Ball Pitch	Ball Matrix	Substrate
25x25	576	1.0 mm	16 x 16 array, fully populated	4 layers

11.7.2.1 25x25 PBGA Package Schematics




PBGA/TEPBGA
TOP AND SIDE VIEW
SEATING PLANE
30° TYP
A1 BALL PAD CORNER
A1 BALL PAD INDICATOR, 1.0 DIA.,OPTIONAL
AVAILABLE MARKING AREA
4X 45° CHAMFER
4X 9.85
20.35 MAX
4X2.50
2X1.00
25.00
0.60±0.10
2.23±0.19
1.17±0.05
0.50±0.10
4:1
A3
Copyright © 2008, 2009, 2010, 2011, Intel Corporation. All Rights Reserved.


NOTES:
1. Dimensions and tolerance per ASME Y14.5M-1994.
2. All Dimensions in MM.
3. Details of pin 1 identifier are optional and may consist of ink, lasermark or metalized marking. But must be located within the zone indicated.
4. JEDEC Code MO-216.
5. Primary DATAM Z and Seating plane are defined by the spherical crowns of the solder balls.
A1 BALL PAD CORNER
3X R0.50
1.00
(1.00)
BOTTOM VIEW
530 SOLDER BALLS
SIZE A3
SCALE 8:1
SHEET 1
Copyright © 2008, 2009, 2010, 2011, Intel Corporation. All Rights Reserved.

11.8 EEPROM Flash Devices

While Intel does not make recommendations regarding these devices, the following devices have been used successfully in previous designs.

11.8.1 Flash

Type: SPI Flash

Size: 256 Kbytes (typical), depending on application.

Table 11-26 Serial Flash Table

Density	Intel PN	Atmel PN	STM PN	SST PN
512KBit		AT25F512N-10SI-2.7	M25P05-AVMN6T	SST25VF512A
1MBit		AT25F1024N-10SI-2.7	M25P10-AVMN6T	SST25VF010A
2MBit		AT25F2048N-10SI-2.7	M25P20-AVMN6T	SST25LF020A
4MBit		AT25F4096N-10SI-2.7	M25P40-AVMN6T	SST25VF040A
8MBit			M25P80-AVMN6T	SST25VF080A
16MBit	QB25F160S33T60 QB25F160S33B60 QH25F160S33T60 QH25F160S33B60 QB25F016S33T60 QB25F016S33B60 QH25F016S33T60 QH25F016S33B60		M25P16-AVMN6T	
32Mbit	QB25F320S33T60 QB25F320S33B60 QH25F320S33T60 QH25F320S33B60		M25P32-AVMN6T	

11.8.2 EEPROM

Table 11-27 EEPROM Devices

Density [Kbits]	Atmel PN	STM PN	OnSemi PN
128	AT25128AN-10SI-2.7	M95128WMN6T	CAT25CS128-TE13
256	AT25256AN-10SI-2.7	M95256WMN6T	

11.9 Voltage Regulator External Components

Components listed in following sections can be used when the I350 1.0V power supply and 1.8V power supply are generated from the 3.3V power supply using on-chip control circuits with external power transistors and other discrete components.

11.9.1 1.0V SVR External Components

Figure 11-16 depicts the external components required for 1.0V SVR operation.

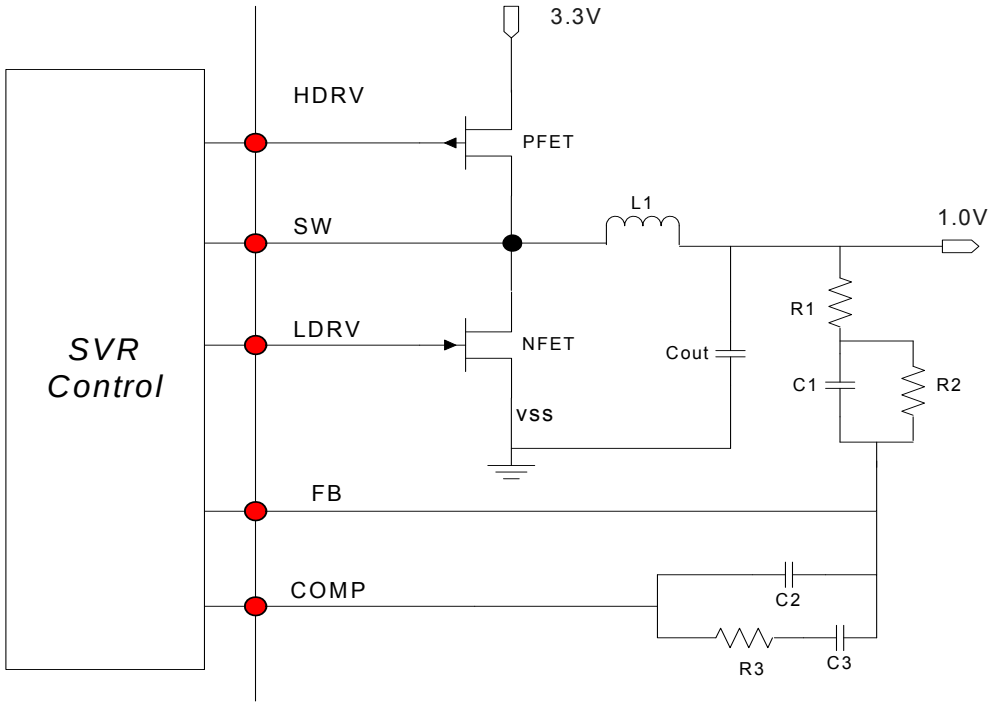


Figure 11-16 1.0V SVR Connection

Following external components can be used for the 1.0V SVR circuit.

Table 11-28 Recommended 1.0V SVR External FETs

Vendor	PFET	NFET
Vishay	SiA417DJ	SiA414DJ
Alpha and Omega	AO4437	AO4402L
Advanced Power Electronics Corp	AP9620GM-HF	AP9410GM-HF
Diodes, Inc.	DMP2022LSS-13	DMN2009LSS-13

Table 11-29 Recommended 1.0V SVR External Capacitors and Resistors

Symbol	Recommended Value	Notes
L1	1 µH ±20%	Should support saturation current of at least 6A.
R1	680 Ω ±5%	
R2	4.7 KΩ ±5%	
R3	39 KΩ ±5%	

Table 11-29 Recommended 1.0V SVR External Capacitors and Resistors

Symbol	Recommended Value	Notes
C1	2.2 nF ±10%	
C2	10 pF ±10%	
C3	680 pF ±10%	
Cout	1. 1 X 47 µF ±10% 2. 3 X 22 µF ±10% 3. 3 X 100 nF ±10% 4. 3 X 10 nF ±10% 5. 3 X 1 nF ±10%	Capacitors listed for Cout are a minimum requirement to reduce ESR. Adding additional capacitance will reduce overshoot.

11.9.2 1.8V LVR External Components

Figure 11-17 depicts the external components required for 1.8V LVR operation.

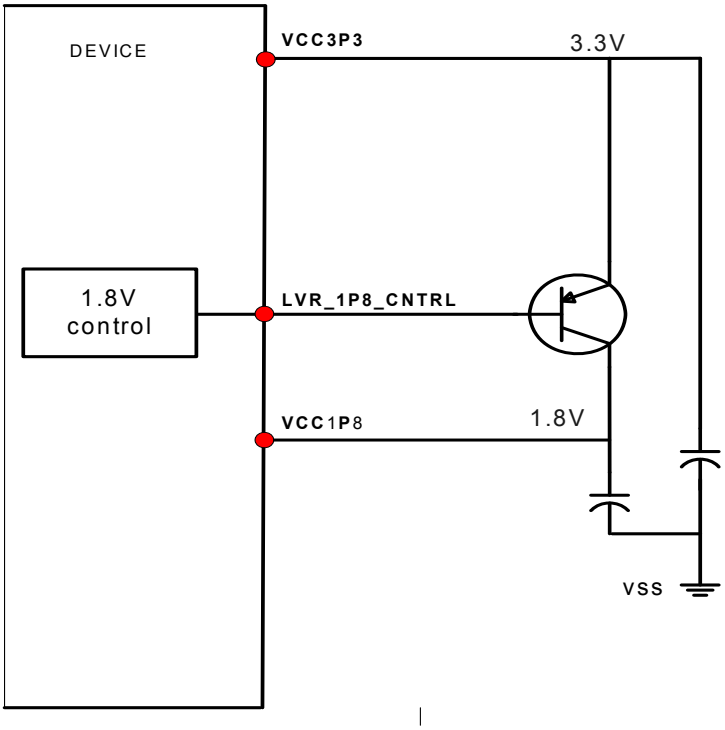


Figure 11-17 1.8V LVR Connection

Following external components can be used for the 1.8V LVR.

Table 11-30 1.8V LVR External Components

PNP BJT	Comments
BCP69	
NJT4030P	Higher f_T and β

Notes:

1. Bulk capacitors – Aluminum electrolytic, X5R, or X7R ceramic capacitors valued from 22 μF to 150 μF are required on the regulator output. The design shall be stable using values throughout this entire range although transient response may be determined by the ESR of the bulk capacitors chosen. Ceramic capacitors generally yield lower over and undershoot in response to load current changes. When all ceramic capacitors are used, a minimum of one 1 uF capacitor are required to control load impedance and insure stability in the 1 to 10 MHz range.
2. Distributed bypass capacitors – Multiple X5R, or X7R 0.1 uF chip ceramic capacitors (minimum 3) should be used. Additional smaller value caps may also be added for improved high-frequency performance.

NOTE: ***This page intentionally left blank.***

12 Design Guidelines

12.1 Ethernet Interface

12.1.1 Magnetics for 1000 BASE-T

Magnetics for the I350 can be either integrated or discrete.

The magnetics module has a critical effect on overall IEEE and emissions conformance. The device should meet the performance required for a design with reasonable margin to allow for manufacturing variation. Occasionally, components that meet basic specifications can cause the system to fail IEEE testing because of interactions with other components or the printed circuit board itself. Carefully qualifying new magnetics modules prevents this problem.

When using discrete magnetics it is necessary to use 'Bob Smith' termination: Use four 75 Ω resistors for cable-side center taps and unused pins. This method terminates pair-to-pair common mode impedance of the CAT5 cable.

Use an EFT capacitor attached to the termination plane. Suggested values are 1500 pF/2 KV or 1000 pF/3 KV. A minimum of 50-mil spacing from capacitor to traces and components should be maintained.

12.1.2 Magnetics Module Qualification Steps

The steps involved in magnetics module qualification are similar to those for crystal qualification:

1. Verify that the vendor's published specifications in the component datasheet meet or exceed specifications.
2. Independently measure the component's electrical parameters on the test bench, checking samples from multiple lots. Check that the measured behavior is consistent from sample to sample and that measurements meet the published specifications.
3. Perform physical layer conformance testing and EMC (FCC and EN) testing in real systems. Vary temperature and voltage while performing system level tests.

12.1.3 Discrete/Integrated Magnetics Specifications

Table 12-1 Discrete/Integrated Magnetics Specifications

Criteria	Condition	Values (Min/Max)
Voltage Isolation	At 50 to 60 Hz for 60 seconds	1500 Vrms (min)
	For 60 seconds	2250 V dc (min)

Table 12-1 Discrete/Integrated Magnetics Specifications

Open Circuit Inductance (OCL) or OCL (alternate)	With 8 mA DC bias at 25 °C	400 µH (min)
	With 8 mA DC bias at 0 °C to 70 °C	350 µH (min)
Insertion Loss	100 kHz through 999 kHz 1.0 MHz through 60 MHz 60.1 MHz through 80 MHz 80.1 MHz through 100 MHz 100.1 MHz through 125 MHz	1 dB (max) 0.6 dB (max) 0.8 dB (max) 1.0 dB (max) 2.4 dB (max)
Return Loss	1.0 MHz through 40 MHz 40.1 MHz through 100 MHz When reference impedance si 85 Ω, 100 Ω, and 115 Ω. Note that return loss values might vary with MDI trace lengths. The LAN magnetics might need to be measured in the platform where it is used.	 18 dB (min) 12 to 20 * LOG (frequency in MHz / 80) dB (min)
Crosstalk Isolation Discrete Modules	1.0 MHz through 29.9 MHz 30 MHz through 250 MHz 250.1 MHz through 375 MHz	-50.3+(8.8*(freq in MHz / 30)) dB (max) -26-(16.8*(LOG(freq in MHz / 250)))) dB (max) -26 dB (max)
Crosstalk Isolation Integrated Modules	1.0 MHz through 10 MHz 10.1 MHz through 100 MHz 100.1 MHz through 375 MHz	-50.8+(8.8*(freq in MHz / 10)) dB (max) -26-(16.8*(LOG(freq in MHz / 100)))) dB (max) -26 dB (max)
Diff to CMR	1.0 MHz through 29.9 MHz 30 MHz through 500 MHz	-40.2+(5.3*((freq in MHz / 30)) dB (max) -22-(14*(LOG((freq in MHz / 250)))) dB (max)
CM to CMR	1.0 MHz through 270 MHz 270.1 MHz through 300 MHz 300.1 MHz through 500 MHz	-57+(38*((freq in MHz / 270)) dB (max) -17-2*((300-(freq in MHz) / 30) dB (max) -17 dB (max)

12.1.4 Third-Party Magnetics Manufacturers

The following magnetics modules have been used successfully in previous designs.

Table 12-2 Magnetics Modules

Manufacturer	Part Number
Low Profile Discrete: Midcom Inc.	 000-7412-35R-LF1
Standard Discrete: BelFuse Pulse Eng.	 S558-5999-P3 (12-core) H5007NL (12-core)
Integrated: FOXCONN Pulse Eng. Amphenol BelFuse Tyco	 JFM38U1C-L1U1W JW0-0013NL RJMG2310 22830ER C03-002 0862-1J1T-Z4-F 6368472-1

12.1.5 Layout Considerations for the Ethernet Interface

The following sections provide recommendations for performing printed circuit board layouts. Good layout practices are essential to meet IEEE PHY conformance specifications and EMI regulatory requirements.

Critical signal traces should be kept as short as possible to decrease the likelihood of being affected by high frequency noise from other signals, including noise carried on power and ground planes. Keeping the traces as short as possible can also reduce capacitive loading.

Since the transmission line medium extends onto the printed circuit board, special attention must be paid to layout and routing of the differential signal pairs.

Designing for 1000 BASE-T Gigabit operation is very similar to designing for 10 and 100 Mb/s. For the I350, system level tests should be performed at all three speeds.

12.1.5.1 Guidelines for Component Placement

Component placement can affect signal quality, emissions, and component operating temperature This section provides guidelines for component placement.

Careful component placement can:

- Decrease potential problems directly related to electromagnetic interference (EMI), which could cause failure to meet applicable government test specifications.
- Simplify the task of routing traces. To some extent, component orientation will affect the complexity of trace routing. The overall objective is to minimize turns and crossovers between traces.

Minimizing the amount of space needed for the Ethernet LAN interface is important because other interfaces compete for physical space on a motherboard near the connector. The Ethernet LAN circuits need to be as close as possible to the connector.

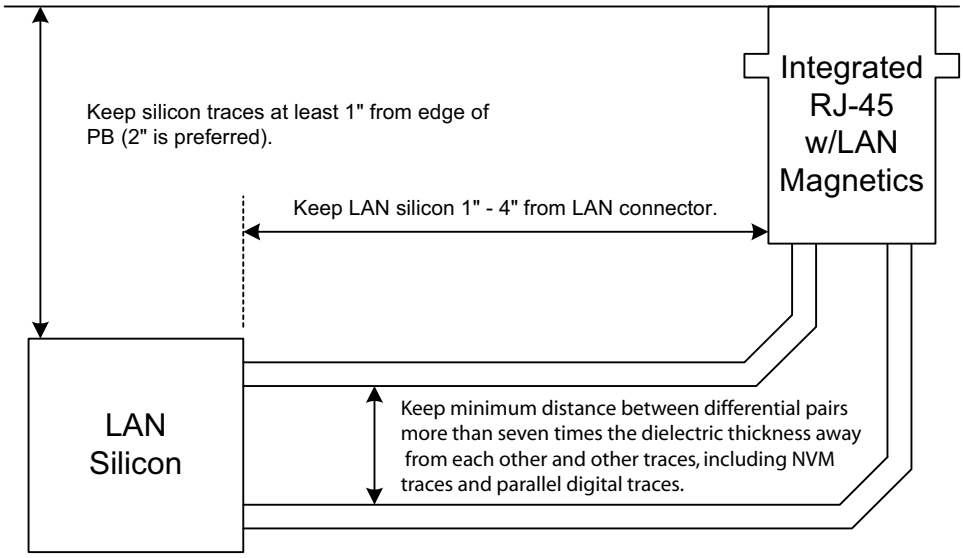


Figure 12-1 General Placement Distances for 1000 BASE-T Designs

Figure 12-1 shows some basic placement distance guidelines. Figure 12-1 shows two differential pairs, but can be generalized for a Gigabit system with four analog pairs. The ideal placement for the Ethernet silicon would be approximately one inch behind the magnetics module.

While it is generally a good idea to minimize lengths and distances, Figure 12-1 also illustrates the need to keep the LAN silicon away from the edge of the board and the magnetics module for best EMI performance.

12.1.5.2 Layout Guidelines for Use with Integrated and Discrete Magnetics

Layout requirements are slightly different when using discrete magnetics.

These include:

- Ground cut for HV isolation (not required for integrated magnetics)
- A maximum of two (2) vias
- Turns less than 45°
- Discrete terminators

Figure 12-2 shows a reference layout for discrete magnetics.

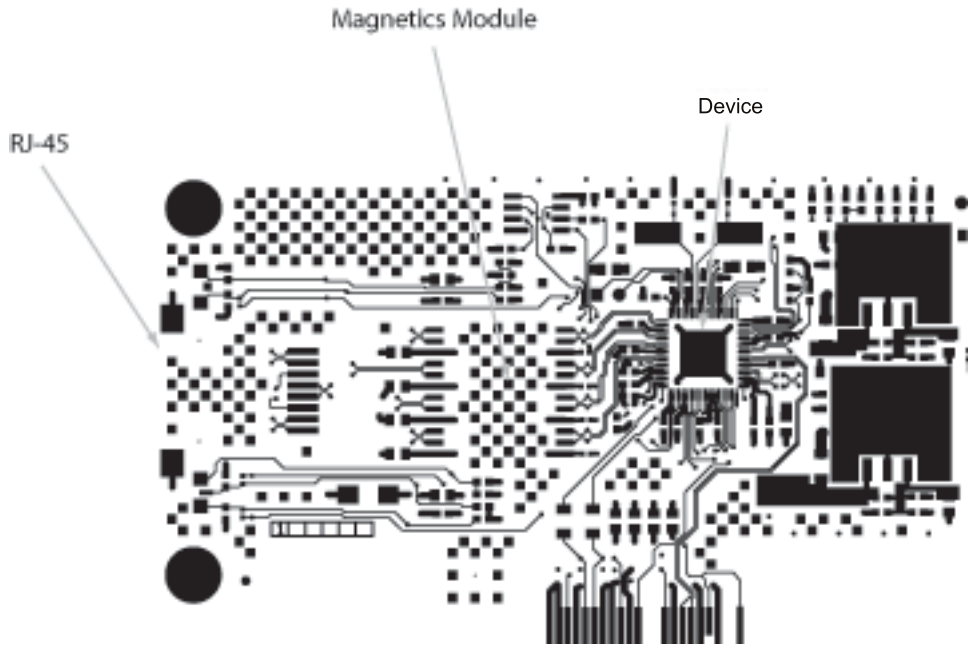


Figure 12-2 Layout for Discrete Magnetics

12.1.5.3 Board Stack-Up Recommendations

Printed circuit boards for these designs typically have four, six, eight, or more layers. Although, the I350 does not dictate the stack up, here is an example of a typical six-layer board stack up:

- Layer 1 is a signal layer. It can contain the differential analog pairs from the Ethernet device to the magnetics module, or to an optical transceiver.
- Layer 2 is a signal ground layer. Chassis ground may also be fabricated in Layer 2 under the connector side of the magnetics module.
- Layer 3 is used for power planes.
- Layer 4 is a signal layer.
- Layer 5 is an additional ground layer.
- Layer 6 is a signal layer. For 1000 BASE-T (copper) Gigabit designs, it is common to route two of the differential pairs (per port) on this layer.

This board stack up configuration can be adjusted to conform to specific OEM design rules.

12.1.5.4 Differential Pair Trace Routing for 10/100/1000 Designs

Trace routing considerations are important to minimize the effects of crosstalk and propagation delays on sections of the board where high-speed signals exist. Signal traces should be kept as short as possible to decrease interference from other signals, including those propagated through power and ground planes. Observe the following suggestions to help optimize board performance:

- Maintain constant symmetry and spacing between the traces within a differential pair.
- Minimize the difference in signal trace lengths of a differential pair.
- Keep the total length of each differential pair under 4 inches. Although possible, designs with differential traces longer than 5 inches are much more likely to have degraded receive BER (Bit Error Rate) performance, IEEE PHY conformance failures, and/or excessive EMI (Electromagnetic Interference) radiation.
- Keep differential pairs more than seven times the dielectric thickness away from each other and other traces, including NVM traces and parallel digital traces.
- Keep maximum separation within differential pairs to 7 mils.
- For high-speed signals, the number of corners and vias should be kept to a minimum. If a 90° bend is required, it is recommended to use two 45° bends instead. Refer to Figure 12-3.

Note: In manufacturing, vias are required for testing and troubleshooting purposes. The via size should be a 17-mil (±2 mils for manufacturing variance) finished hole size (FHS).

- Traces should be routed away from board edges by a distance greater than the trace height above the reference plane. This allows the field around the trace to couple more easily to the ground plane rather than to adjacent wires or boards.
- Do not route traces and vias under crystals or oscillators. This will prevent coupling to or from the clock. And as a general rule, place traces from clocks and drives at a minimum distance from apertures by a distance that is greater than the largest aperture dimension.

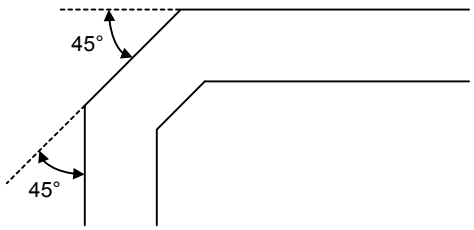


Figure 12-3 Trace Routing

- The reference plane for the differential pairs should be continuous and low impedance. It is recommended that the reference plane be either ground or 1.9 V dc (the voltage used by the PHY). This provides an adequate return path for and high frequency noise currents.
- Do not route differential pairs over splits in the associated reference plane as it may cause discontinuity in impedances.

12.1.5.5 Maximum Trace Lengths Based on Trace Geometry

Table 12-3 Maximum Trace Lengths Based on Trace Geometry and Board Stack-Up

Dielectric Thickness (mils)	Dielectric Constant (DK) at 1 MHz	Width / Space/ Width (mils)	Pair-to-Pair Space (mils)	Nominal Impedance (Ohms)	Impedance Tolerance (±%)	Maximum Trace Length (inches)[1]
2.7	4.05	4/10/4	19	95[2]	17[2]	3.5
2.7	4.05	4/10/4	19	95[2]	15[2]	4
2.7	4.05	4/10/4	19	95	10	5
3.3	4.1	4.2/9/4.2	23	100[2]	17[2]	4
3.3	4.1	4.2/9/4.2	23	100	15	4.6
3.3	4.1	4.2/9/4.2	23	100	10	6
4	4.2	5/9/5	28	100[2]	17[2]	4.5
4	4.2	5/9/5	28	100	15	5.3
4	4.2	5/9/5	28	100	10	7
4	4.2	5/7/5	28	95	10	5.4
4	4.2	5/7/5	28	95	15	4.8
4	4.2	5/7/5	28	95	17	4.3

Note:

1. Longer MDI trace lengths may be achievable, but may make it more difficult to achieve IEEE conformance. Simulations have shown deviations are possible if traces are kept short. Longer traces are possible; use cost considerations and stack-up tolerance for differential pairs to determine length requirements.
2. Deviations from 100 Ω nominal and/or tolerances greater than 15% decrease the maximum length for IEEE conformance.

Use the MDI Differential Trace Calculator to determine the maximum MDI trace length for your trace geometry and board stack-up. Contact your Intel representative for access.

The following factors can limit the maximum MDI differential trace lengths for IEEE conformance:

- Dielectric thickness
- Dielectric constant
- Nominal differential trace impedance
- Trace impedance tolerance
- Copper trace losses
- Additional devices, such as switches, in the MDI path may impact IEEE conformance.

Board geometry should also be factored in when setting trace length.

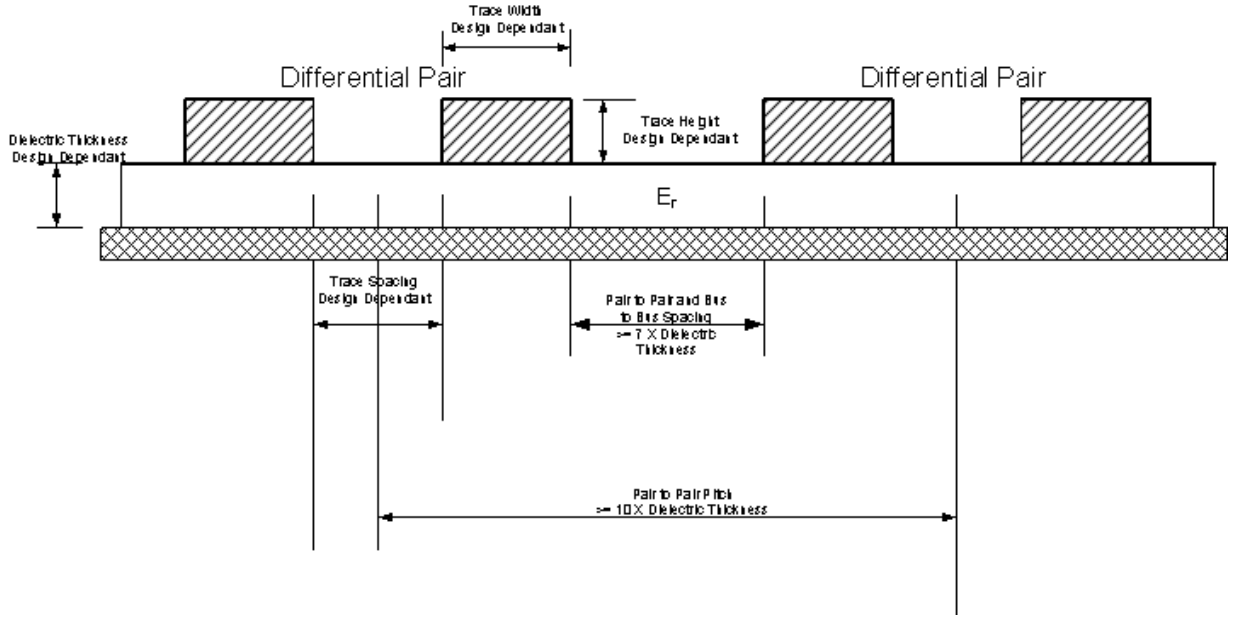


Figure 12-4 MDI Trace Geometry

12.1.5.6 Signal Termination and Coupling

The I350 has internal termination on the MDI signals. External resistors are not needed. Adding pads for external resistors can degrade signal integrity.

12.1.5.7 Signal Trace Geometry for 1000 BASE-T Designs

The key factors in controlling trace EMI radiation are the trace length and the ratio of trace-width to trace-height above the reference plane. To minimize trace inductance, high-speed signals and signal layers that are close to a reference or power plane should be as short and wide as practical. Ideally, this trace width to height above the ground plane ratio is between 1:1 and 3:1. To maintain trace impedance, the width of the trace should be modified when changing from one board layer to another if the two layers are not equidistant from the neighboring planes.

Each pair of signal should have a differential impedance of 100 Ω. +/- 15%. If a particular tool cannot design differential traces, it is permissible to specify 55-65 Ω single-ended traces as long as the spacing between the two traces is minimized. As an example, consider a differential trace pair on Layer 1 that is 8 mils (0.2 mm) wide and 2 mils (0.05 mm) thick, with a spacing of 8 mils (0.2 mm). If the fiberglass layer is 8 mils (0.2 mm) thick with a dielectric constant, E_R, of 4.7, the calculated single-ended impedance would be approximately 61 Ω and the calculated differential impedance would be approximately 100 Ω.

When performing a board layout, do not allow the CAD tool auto-router to route the differential pairs without intervention. In most cases, the differential pairs will have to be routed manually.

Note: Measuring trace impedance for layout designs targeting 100 Ω often results in lower actual impedance. Designers should verify actual trace impedance and adjust the layout accordingly. If the actual impedance is consistently low, a target of 105 – 110 Ω should compensate for second order effects.

It is necessary to compensate for trace-to-trace edge coupling, which can lower the differential impedance by up to 10 Ω, when the traces within a pair are closer than 30 mils (edge to edge).

12.1.5.8 Trace Length and Symmetry for 1000 BASE-T Designs

As indicated earlier, the overall length of differential pairs should be less than four inches measured from the Ethernet device to the magnetics.

The differential traces (within each pair) should be equal in total length to within 30 mils (0.76 mm) and as symmetrical as possible. Asymmetrical and unequal length traces in the differential pairs contribute to common mode noise. If a choice has to be made between matching lengths and fixing symmetry, more emphasis should be placed on fixing symmetry. Common mode noise can degrade the receive circuit's performance and contribute to radiated emissions.

12.1.5.8.1 Signal Detect

Each port of the I350 has a signal detect pin for connection to optical transceivers. For designs without optical transceivers, these signals can be left unconnected because they have internal pull-up resistors. Signal detect is not a high-speed signal and does not require special layout.

12.1.5.9 Impedance Discontinuities

Impedance discontinuities cause unwanted signal reflections. Minimize vias (signal through holes) and other transmission line irregularities. If vias must be used, a reasonable budget is two per differential trace. Unused pads and stub traces should also be avoided.

12.1.5.10 Reducing Circuit Inductance

Traces should be routed over a continuous reference plane with no interruptions. If there are vacant areas on a reference or power plane, the signal conductors should not cross the vacant area. This causes impedance mismatches and associated radiated noise levels. Noisy logic grounds should be separated from analog signal grounds to reduce coupling. Noisy logic grounds can sometimes affect sensitive DC subsystems such as analog to digital conversion, operational amplifiers, etc.

All ground vias should be connected to every ground plane; and similarly, every power via, to all power planes at equal potential. This helps reduce circuit inductance. Another recommendation is to physically locate grounds to minimize the loop area between a signal path and its return path. Rise and fall times should be as slow as possible. Because signals with fast rise and fall times contain many high frequency harmonics, which can radiate significantly. The most sensitive signal returns closest to the chassis ground should be connected together. This will result in a smaller loop area and reduce the likelihood of crosstalk. The effect of different configurations on the amount of crosstalk can be studied using electronics modeling software.

12.1.5.11 Signal Isolation

To maintain best signal integrity, keep digital signals far away from the analog traces. A good rule of thumb is no digital signal should be within 300 mils (7.5 mm) of the differential pairs. If digital signals on other board layers cannot be separated by a ground plane, they should be routed perpendicular to the differential pairs. If there is another LAN controller on the board, take care to keep the differential pairs from that circuit away.

Some rules to follow for signal isolation:

- Separate and group signals by function on separate layers if possible. Keep a minimum distance between differential pairs more than seven times the dielectric thickness away from each other and other traces, including NVM traces and parallel digital traces.
- Physically group together all components associated with one clock trace to reduce trace length and radiation.
- Isolate I/O signals from high-speed signals to minimize crosstalk, which can increase EMI emission and susceptibility to EMI from other signals.
- Avoid routing high-speed LAN traces near other high-frequency signals associated with a video controller, cache controller, processor, or other similar devices.

12.1.5.12 Traces for Decoupling Capacitors

Traces between decoupling and I/O filter capacitors should be as short and wide as practical. Long and thin traces are more inductive and would reduce the intended effect of decoupling capacitors. Also for similar reasons, traces to I/O signals and signal terminations should be as short as possible. Vias to the decoupling capacitors should be sufficiently large in diameter to decrease series inductance.

12.1.5.13 Light Emitting Diodes for Designs Based on the I350

The I350 provides three programmable high-current push-pull (active high) outputs to directly drive LEDs for link activity and speed indication. Each LAN device provides an independent set of LED outputs; these pins and their function are bound to a specific LAN device. Each of the four LED outputs can be individually configured to select the particular event, state, or activity, which is indicated on that output. In addition, each LED can be individually configured for output polarity, as well as for blinking versus non-blinking (steady-state) indication.

Since the LEDs are likely to be integral to a magnetics module, take care to route the LED traces away from potential sources of EMI noise. In some cases, it may be desirable to attach filter capacitors.

The LED ports are fully programmable through the NVM interface.

12.1.6 Physical Layer Conformance Testing

Physical layer conformance testing (also known as IEEE testing) is a fundamental capability for all companies with Ethernet LAN products. PHY testing is the final determination that a layout has been performed successfully. If your company does not have the resources and equipment to perform these tests, consider contracting the tests to an outside facility.

12.1.6.1 Conformance Tests for 10/100/1000 Mb/s Designs

Crucial tests are as follows, listed in priority order:

- Bit Error Rate (BER). Good indicator of real world network performance. Perform bit error rate testing with long and short cables and many link partners. The test limit is 10^{-11} errors.
- Output Amplitude, Rise and Fall Time (10/100 Mb/s), Symmetry and Droop (1000Mbps). For the 82575 controller, use the appropriate PHY test waveform.
- Return Loss. Indicator of proper impedance matching, measured through the RJ-45 connector back toward the magnetics module.
- Jitter Test (10/100 Mb/s) or Unfiltered Jitter Test (1000 Mb/s). Indicator of clock recovery ability (master and slave for Gigabit controller).

12.1.7 Troubleshooting Common Physical Layout Issues

The following is a list of common physical layer design and layout mistakes in LAN On Motherboard Designs.

1. Lack of symmetry between the two traces within a differential pair. Asymmetry can create common-mode noise and distort the waveforms. For each component and/or via that one trace encounters, the other trace should encounter the same component or a via at the same distance from the Ethernet silicon.
2. Unequal length of the two traces within a differential pair. Inequalities create common-mode noise and will distort the transmit or receive waveforms.
3. Excessive distance between the Ethernet silicon and the magnetics. Long traces on FR4 fiberglass epoxy substrate will attenuate the analog signals. In addition, any impedance mismatch in the traces will be aggravated if they are longer than the four inch guideline.
4. Routing any other trace parallel to and close to one of the differential traces. Crosstalk getting onto the receive channel will cause degraded long cable BER. Crosstalk getting onto the transmit channel can cause excessive EMI emissions and can cause poor transmit BER on long cables. At a minimum, other signals should be kept 0.3 inches from the differential traces.
5. Routing one pair of differential traces too close to another pair of differential traces. After exiting the Ethernet silicon, the trace pairs should be kept 0.3 inches or more away from the other trace pairs. The only possible exceptions are in the vicinities where the traces enter or exit the magnetics, the RJ-45 connector, and the Ethernet silicon.
6. Use of a low-quality magnetics module.
7. Re-use of an out-of-date physical layer schematic in a Ethernet silicon design. The terminations and decoupling can be different from one PHY to another.
8. Incorrect differential trace impedances. It is important to have ~100 Ω impedance between the two traces within a differential pair. This becomes even more important as the differential traces become longer. To calculate differential impedance, many impedance calculators only multiply the single-ended impedance by two. This does not take into account edge-to-edge capacitive coupling between the two traces. When the two traces within a differential pair are kept close to each other, the edge coupling can lower the effective differential impedance by 5 Ω to 20 Ω. Short traces have fewer problems if the differential impedance is slightly off target.

12.2 PCIe

The controller connects to the host system using a PCIe interface. The interface can be configured to operate in several link modes. These are detailed in the functional description chapter. A link between the ports of two devices is a collection of lanes. Each lane has to be AC-coupled between its corresponding transmitter and receiver; with the AC-coupling capacitor located close to the transmitter side (within 1 inch). Each end of the link is terminated on the die into nominal 100 differential DC impedance. Board termination is not required.

Refer to the *PCI Express* Base Specification, Revision 2.0* and *PCI Express* Card Electromechanical Specification, Revision 2.0.*

12.2.1 Link Width Configuration

The device supports link widths of x4, x2, or x1 as determined by the PCIe PHY Auto Configuration Structure. The configuration is loaded into the Maximum Link Width field of the PCIe capability Register (LCAP[11:6]; with the silicon default of a x4 link).

During link configuration, the platform and the controller negotiate a common link width. In order for this to work, the selected maximum number of PCIe lanes must be connected to the host system.

12.2.2 Polarity Inversion and Lane Reversal

To ease routing, designers have the flexibility to the lane reversal modes supported by the I350. Polarity inversion can also be used, since the polarity of each differential pair is detected during the link training sequence.

When lane reversal is used, some of the down-shift options are not available. For a description of available combinations, consult the functional description in the PCIe interconnects chapter of this document.

12.2.3 PCIe Reference Clock

The device requires a 100 MHz differential reference clock, denoted PE_CLK_p and PE_CLK_n. This signal is typically generated on the system board and routed to the PCIe port. For add-in cards, the clock will be furnished at the PCIe connector.

The frequency tolerance for the PCIe reference clock is +/- 300 ppm.

12.3 Clock Source

All designs require a 25 MHz clock source. The I350 uses the 25 MHz source to generate clocks up to 125 MHz and 1.25 GHz for the PHY circuits. For optimum results with lowest cost, connect a 25 MHz parallel resonant crystal and appropriate load capacitors at the XTAL1 and XTAL2 leads. The frequency tolerance of the timing device should be 30 ppm or better.

Refer to the Intel® Ethernet Controllers Timing Device Selection Guide for more information on choosing crystals. For further information regarding the clock for the I350, refer to the sections about frequency control, crystals, and oscillators that follow.

12.3.1 Frequency Control Device Design Considerations

This section provides information regarding frequency control devices, including crystals and oscillators, for use with all Intel Ethernet controllers. Several suitable frequency control devices are available; none of which present any unusual challenges in selection. The concepts documented herein are applicable to other data communication circuits, including Platform LAN Connect devices (PHYs).

The Intel Ethernet controllers contain amplifiers, which when used with the specific external components, form the basis for feedback oscillators. These oscillator circuits, which are both economical and reliable, are described in more detail in Section 12.4.1.

The Intel Ethernet controllers also have bus clock input functionality, however a discussion of this feature is beyond the scope of this document, and will not be addressed.

The chosen frequency control device vendor should be consulted early in the design cycle. Crystal and oscillator manufacturers familiar with networking equipment clock requirements may provide assistance in selecting an optimum, low-cost solution.

12.3.2 Frequency Control Component Types

Several types of third-party frequency reference components are currently marketed. A discussion of each follows, listed in preferred order.

12.3.2.1 Quartz Crystal

Quartz crystals are generally considered to be the mainstay of frequency control components due to their low cost and ease of implementation. They are available from numerous vendors in many package types and with various specification options.

12.3.2.2 Fixed Crystal Oscillator

A packaged fixed crystal oscillator comprises an inverter, a quartz crystal, and passive components conveniently packaged together. The device renders a strong, consistent square wave output. Oscillators used with microprocessors are supplied in many configurations and tolerances.

Crystal oscillators should be restricted to use in special situations, such as shared clocking among devices or multiple controllers. As clock routing can be difficult to accomplish, it is preferable to provide a separate crystal for each device.

12.3.2.3 Programmable Crystal Oscillators

A programmable oscillator can be configured to operate at many frequencies. The device contains a crystal frequency reference and a phase lock loop (PLL) clock generator. The frequency multipliers and divisors are controlled by programmable fuses.

A programmable oscillator's accuracy depends heavily on the Ethernet device's differential transmit lines. The Physical Layer (PHY) uses the clock input from the device to drive a differential Manchester (for 10 Mb/s operation), an MLT-3 (for 100 Mbps operation) or a PAM-5 (for 1000 Mbps operation) encoded analog signal across the twisted pair cable. These signals are referred to as self-clocking, which means the clock must be recovered at the receiving link partner. Clock recovery is performed with another PLL that locks onto the signal at the other end.

PLLs are prone to exhibit frequency jitter. The transmitted signal can also have considerable jitter even with the programmable oscillator working within its specified frequency tolerance. PLLs must be designed carefully to lock onto signals over a reasonable frequency range. If the transmitted signal has high jitter and the receiver's PLL loses its lock, then bit errors or link loss can occur.

PHY devices are deployed for many different communication applications. Some PHYs contain PLLs with marginal lock range and cannot tolerate the jitter inherent in data transmission clocked with a programmable oscillator. The American National Standards Institute (ANSI) X3.263-1995 standard test method for transmit jitter is not stringent enough to predict PLL-to-PLL lock failures, therefore, the use of programmable oscillators is not recommended.

12.3.2.4 Ceramic Resonator

Similar to a quartz crystal, a ceramic resonator is a piezoelectric device. A ceramic resonator typically carries a frequency tolerance of ±0.5%, – inadequate for use with Intel Ethernet controllers, and therefore, should not be utilized.

12.4 Crystal Support

12.4.1 Crystal Selection Parameters

All crystals used with Intel Ethernet controllers are described as AT-cut, which refers to the angle at which the unit is sliced with respect to the long axis of the quartz stone. Table 12-4 lists crystals which have been used successfully in other designs (however, no particular product is recommended):

Table 12-4 Crystal Manufacturers and Part Numbers

Manufacturer	Part No.
KDS America	DSX321G
NDK America Inc.	41CD25.0F1303018
TXC Corporation - USA	7A25000165[1] 9C25000008

1. This part footprint compatible with X540 designs.

For information about crystal selection parameters, see the electrical specification.

12.4.1.1 Vibrational Mode

Crystals in the above-referenced frequency range are available in both fundamental and third overtone. Unless there is a special need for third overtone, use fundamental mode crystals.

At any given operating frequency, third overtone crystals are thicker and more rugged than fundamental mode crystals. Third overtone crystals are more suitable for use in military or harsh industrial environments. Third overtone crystals require a trap circuit (extra capacitor and inductor) in the load circuitry to suppress fundamental mode oscillation as the circuit powers up. Selecting values for these components is beyond the scope of this document.

12.4.1.2 Nominal Frequency

Intel Ethernet controllers use a crystal frequency of 25.000 MHz. The 25 MHz input is used to generate a 125 MHz transmit clock for 100BASE-TX and 1000BASE-TX operation – 10 MHz and 20 MHz transmit clocks, for 10BASE-T operation.

12.4.1.3 Frequency Tolerance

The frequency tolerance for an Ethernet Platform LAN Connect is dictated by the IEEE 802.3 specification as ±50 parts per million (ppm). This measurement is referenced to a standard temperature of 25° C. Intel recommends a frequency tolerance of ±30 ppm.

12.4.1.4 Temperature Stability and Environmental Requirements

Temperature stability is a standard measure of how the oscillation frequency varies over the full operational temperature range (and beyond). Several optional temperature ranges are currently available, including -40° C to +85° C for industrial environments. Some vendors separate operating temperatures from temperature stability. Manufacturers may also list temperature stability as 50 ppm in their data sheets.

Note: Crystals also carry other specifications for storage temperature, shock resistance, and reflow solder conditions. Crystal vendors should be consulted early in the design cycle to discuss the application and its environmental requirements.

12.4.1.5 Calibration Mode

The terms series-resonant and parallel-resonant are often used to describe crystal oscillator circuits. Specifying parallel mode is critical to determining how the crystal frequency is calibrated at the factory.

A crystal specified and tested as series resonant oscillates without problem in a parallel-resonant circuit, but the frequency is higher than nominal by several hundred parts per million. The purpose of adding load capacitors to a crystal oscillator circuit is to establish resonance at a frequency higher than the crystal's inherent series resonant frequency.

Figure 12-5 shows the recommended placement and layout of an internal oscillator circuit. Note that pin X1 and X2 refers to XTAL1 and XTAL2 in the Ethernet device, respectively. The crystal and the capacitors form a feedback element for the internal inverting amplifier. This combination is called parallel-resonant, because it has positive reactance at the selected frequency. In other words, the crystal behaves like an inductor in a parallel LC circuit. Oscillators with piezoelectric feedback elements are also known as "Pierce" oscillators.

12.4.1.6 Load Capacitance

The formula for crystal load capacitance is as follows:

$$C_L = \frac{(C1 \cdot C2)}{(C1 + C2)} + C_{stray}$$

where C1 = C2 = 27 pF
and C_{stray} = allowance for additional capacitance in pads, traces and the chip carrier within the Ethernet device package

An allowance of 3 pF to 7 pF accounts for lumped stray capacitance. The calculated load capacitance is 16 pF with an estimated stray capacitance of about 5 pF.

Individual stray capacitance components can be estimated and added. For example, surface mount pads for the load capacitors add approximately 2.5 pF in parallel to each capacitor. This technique is especially useful if Y1, C1 and C2 must be placed farther than approximately one-half (0.5) inch from the device. It is worth noting that thin circuit boards generally have higher stray capacitance than thick circuit boards. Consult the PCIe Design Guide for more information.

The oscillator frequency should be measured with a precision frequency counter where possible. The load specification or values of C1 and C2 should be fine tuned for the design. As the actual capacitance load increases, the oscillator frequency decreases.

Note: C1 and C2 may vary by as much as 5% (approximately 1 pF) from their nominal values.

12.4.1.7 Shunt Capacitance

The shunt capacitance parameter is relatively unimportant compared to load capacitance. Shunt capacitance represents the effect of the crystal's mechanical holder and contacts. The shunt capacitance should equal a maximum of 6 pF.

12.4.1.8 Equivalent Series Resistance

Equivalent Series Resistance (ESR) is the real component of the crystal's impedance at the calibration frequency, which the inverting amplifier's loop gain must overcome. ESR varies inversely with frequency for a given crystal family. The lower the ESR, the faster the crystal starts up. Use crystals with an ESR value of 50 Ω or better.

12.4.1.9 Drive Level

Drive level refers to power dissipation in use. The allowable drive level for a Surface Mounted Technology (SMT) crystal is less than its through-hole counterpart, because surface mount crystals are typically made from narrow, rectangular AT strips, rather than circular AT quartz blanks.

Some crystal data sheets list crystals with a maximum drive level of 1 mW. However, Intel Ethernet controllers drive crystals to a level less than the suggested 0.3 mW value. This parameter does not have much value for on-chip oscillator use.

12.4.1.10 Aging

Aging is a permanent change in frequency (and resistance) occurring over time. This parameter is most important in its first year because new crystals age faster than old crystals. Use crystals with a maximum of ±5 ppm per year aging.

12.4.1.11 Reference Crystal

The normal tolerances of the discrete crystal components can contribute to small frequency offsets with respect to the target center frequency. To minimize the risk of tolerance-caused frequency offsets causing a small percentage of production line units to be outside of the acceptable frequency range, it is important to account for those shifts while empirically determining the proper values for the discrete loading capacitors, C1 and C2.

Even with a perfect support circuit, most crystals will oscillate slightly higher or slightly lower than the exact center of the target frequency. Therefore, frequency measurements (which determine the correct value for C1 and C2) should be performed with an ideal reference crystal. When the capacitive load is exactly equal to the crystal's load rating, an ideal reference crystal will be perfectly centered at the desired target frequency.

12.4.1.11.1 Reference Crystal Selection

There are several methods available for choosing the appropriate reference crystal:

- If a Saunders and Associates (S&A) crystal network analyzer is available, then discrete crystal components can be tested until one is found with zero or nearly zero ppm deviation (with the appropriate capacitive load). A crystal with zero or near zero ppm deviation will be a good reference crystal to use in subsequent frequency tests to determine the best values for C1 and C2.
- If a crystal analyzer is not available, then the selection of a reference crystal can be done by measuring a statistically valid sample population of crystals, which has units from multiple lots and approved vendors. The crystal, which has an oscillation frequency closest to the center of the distribution, should be the reference crystal used during testing to determine the best values for C1 and C2.
- It may also be possible to ask the approved crystal vendors or manufacturers to provide a reference crystal with zero or nearly zero deviation from the specified frequency when it has the specified CLoad capacitance.

When choosing a crystal, customers must keep in mind that to comply with IEEE specifications for 10/100 and 10/100/1000Base-T Ethernet LAN, the transmitter reference frequency must be precise within ±50 ppm. Intel® recommends customers to use a transmitter reference frequency that is accurate to within ±30 ppm to account for variations in crystal accuracy due to crystal manufacturing tolerance.

12.4.1.11.2 Circuit Board

Since the dielectric layers of the circuit board are allowed some reasonable variation in thickness, the stray capacitance from the printed board (to the crystal circuit) will also vary. If the thickness tolerance for the outer layers of dielectric are controlled within ±17 percent of nominal, then the circuit board should not cause more than ±2 pF variation to the stray capacitance at the crystal. When tuning crystal frequency, it is recommended that at least three circuit boards are tested for frequency. These boards should be from different production lots of bare circuit boards.

Alternatively, a larger sample population of circuit boards can be used. A larger population will increase the probability of obtaining the full range of possible variations in dielectric thickness and the full range of variation in stray capacitance.

Next, the exact same crystal and discrete load capacitors (C1 and C2) must be soldered onto each board, and the LAN reference frequency should be measured on each circuit board.

The circuit board, which has a LAN reference frequency closest to the center of the frequency distribution, should be used while performing the frequency measurements to select the appropriate value for C1 and C2.

12.4.1.11.3 Temperature Changes

Temperature changes can cause the crystal frequency to shift. Therefore, frequency measurements should be done in the final system chassis across the system's rated operating temperature range.

12.4.2 Crystal Placement and Layout Recommendations

Crystal clock sources should not be placed near I/O ports or board edges. Radiation from these devices can be coupled into the I/O ports and radiate beyond the system chassis. Crystals should also be kept away from the Ethernet magnetics module to prevent interference.

Note: Failure to follow these guidelines could result in the 25 MHz clock failing to start.

When designing the layout for the crystal circuit, the following rules must be used:

- Place load capacitors as close as possible (within design-for-manufacturability rules) to the crystal solder pads. They should be no more than 90 mils away from crystal pads.
- The two load capacitors, crystal component, the Ethernet controller device, and the crystal circuit traces must all be located on the same side of the circuit board (maximum of one via-to-ground load capacitor on each XTAL trace).
- Use 27 pF (5% tolerance) 0402 load capacitors.
- Place load capacitor solder pad directly in line with circuit trace (see Figure 12-5, point A).
- Use 50 Ω impedance single-ended microstrip traces for the crystal circuit.
- Route traces so that electro-magnetic fields from XTAL2 do not couple onto XTAL1. No differential traces.
- Route XTAL1 and XTAL2 traces to nearest inside corners of crystal pad (see Figure 12-5, point B).
- Ensure that the traces from XTAL1 and XTAL2 are symmetrically routed and that their lengths are matched.
- The total trace length of XTAL1 or XTAL2 should be less than 750 mils.

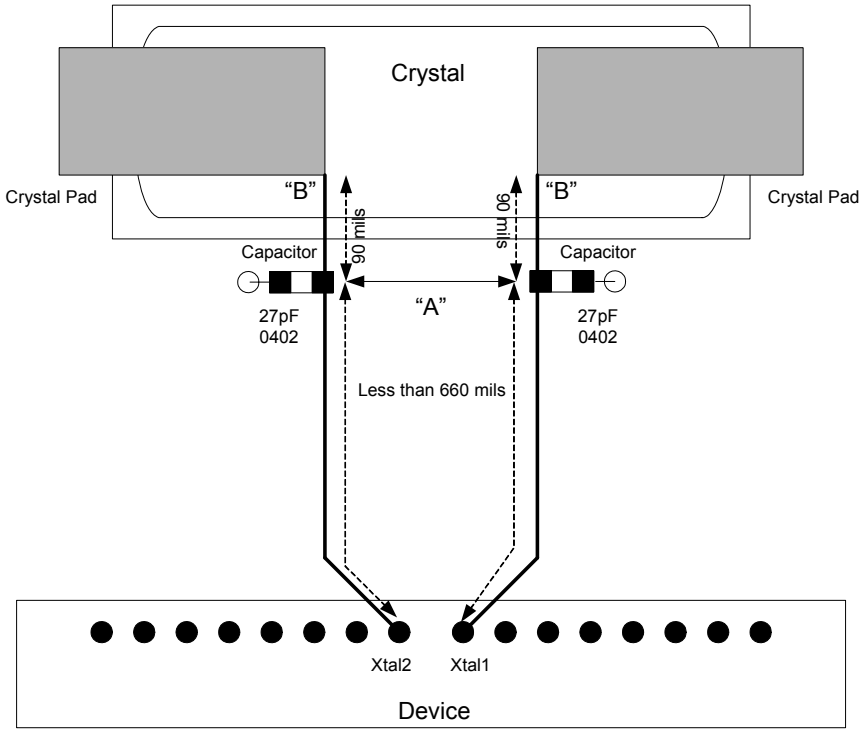


Figure 12-5 Recommended Crystal Placement and Layout

12.5 Oscillator Support

The I350 clock input circuit is optimized for use with an external crystal. However, an oscillator can also be used in place of the crystal with the proper considerations:

- We recommend adding an AC decoupling capacitor between the oscillator output and the input (XTAL1) of the LAN device.
- The input clock jitter from the oscillator can impact the I350 clock and its performance.

Note: The power consumption of additional circuitry equals about 1.5 mW.

Table 12-5 lists oscillators that can be used with the controller. Please note that no particular oscillator is recommended).

Table 12-5 Oscillator Manufacturers and Part Numbers

Manufacturer	Part No.
NDK AMERICA INC	2560TKA-25M
TXC CORPORATION - USA	6N25000160 or 7W25000025
CITIZEN AMERICA CORP	CSX750FJB25.000M-UT
Raltron Electronics Corp	CO4305-25.000-T-TR
MtronPTI	M214TCN
Kyocera Corporation	KC5032C-C3

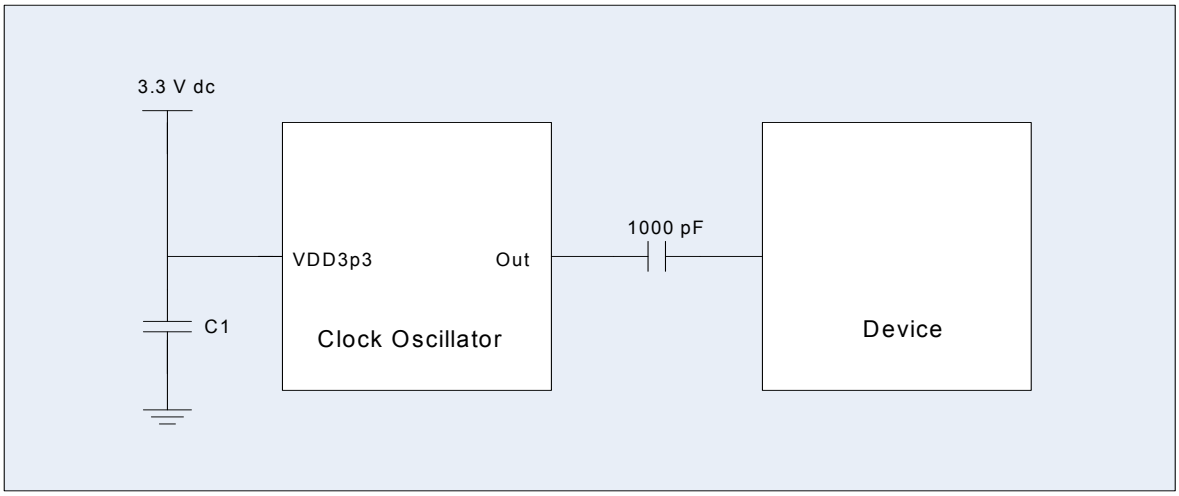


Figure 12-6 Oscillator Solution

12.5.1 Oscillator Placement and Layout Recommendations

Oscillator clock sources should not be placed near I/O ports or board edges. Radiation from these devices can be coupled into the I/O ports and radiate beyond the system chassis. Oscillators should also be kept away from the Ethernet magnetics module to prevent interference.

12.6 Device Disable

For a LOM design, it might be desirable for the system to provide BIOS-setup capability for selectively enabling or disabling LOM devices. This enables designers more control over system resource-management, avoid conflicts with add-in NIC solutions, etc. The I350 provides support for selectively enabling or disabling it.

Device disable is initiated by asserting the asynchronous DEV_OFF_N pin. The DEV_OFF_N pin has an internal pull-up resistor, so that it can be left not connected to enable device operation.

The NVM's *Device Disable Power Down En* bit enables device disable mode (hardware default is that the mode is disabled).

While in device disable mode, the PCIe link is in L3 state. The PHY is in power down mode. Output buffers are tri-stated.

Assertion or deassertion of PCIe PE_RST_N does not have any effect while the I350 is in device disable mode (that is, the I350 stays in the respective mode as long as DEV_OFF_N is asserted). However, the I350 might momentarily exit the device disable mode from the time PCIe PE_RST_N is de-asserted again and until the NVM is read.

During power-up, the DEV_OFF_N pin is ignored until the NVM is read. From that point, the I350 might enter device disable if DEV_OFF_N is asserted.

Note: The DEV_OFF_N pin should maintain its state during system reset and system sleep states. It should also insure the proper default value on system power up. For example, a designer could use a GPIO pin that defaults to 1b (enable) and is on system suspend power. For example, it maintains the state in S0-S5 ACPI states).

12.6.1 BIOS Handling of Device Disable

Assume that in the following power-up sequence the DEV_OFF_N signal is driven high (or it is already disabled)

1. The PCIe is established following the GIO_PWR_GOOD.
2. BIOS recognizes that the entire I350 should be disabled.
3. The BIOS drives the DEV_OFF_N signal to the low level.
4. As a result, the I350 samples the DEV_OFF_N signals and enters either the device disable mode.
5. The BIOS could put the link in the Electrical IDLE state (at the other end of the PCIe link) by clearing the *Link Disable* bit in the Link Control register.
6. BIOS might start with the device enumeration procedure (the entire I350 functions are invisible).
7. Proceed with normal operation

Re-enable could be done by driving high the DEV_OFF_N signal, followed later by bus enumeration.

12.7 SMBus and NC-SI

SMBus and NC-SI are interfaces for pass-through and configuration traffic between the Management Controller (MC) and the device.

Note: Intel recommends that the SMBus be connected to the ICH or MC for the EEPROM recovery solution. If the connection is to a MC, it will be able to send the EEPROM release command.

The I350 can be connected to an external MC. It operates in one of two modes:

- SMBus mode
- NC-SI mode

The Clock-out (if enabled) is provided in all power states (unless the device is disabled).

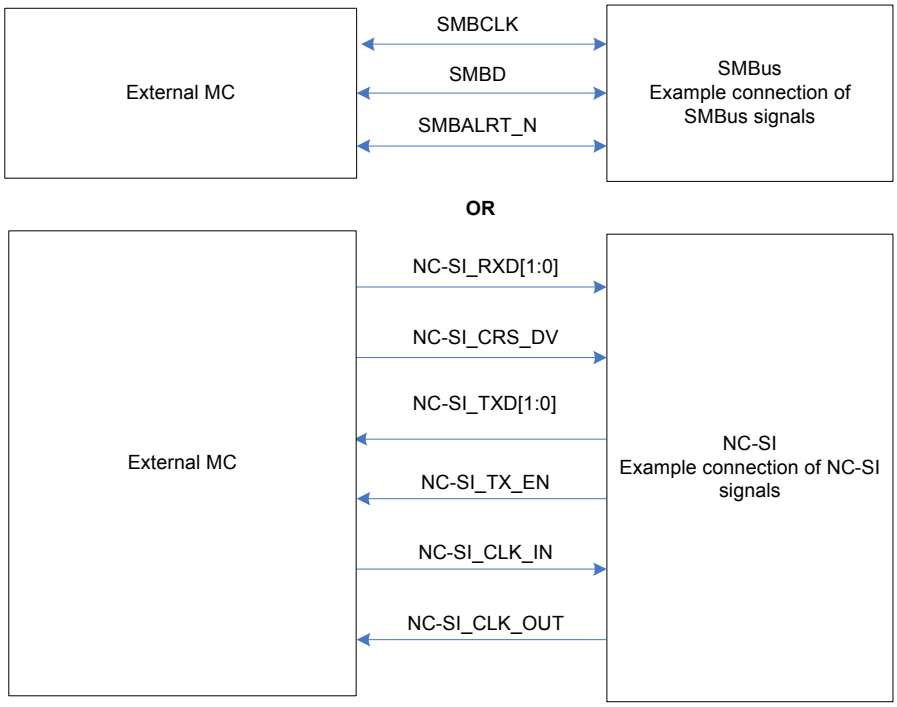


Figure 12-7 External MC Connections with NC-SI and SMBus

12.8 NC-SI

12.8.1 Design Requirements

12.8.1.1 Network Controller

The NC-SI Interface enables network manageability implementations required by information technology personnel for remote control and alerting via the LAN. Management packets can be routed to or from a management processor.

12.8.1.2 External Management Controller (MC)

An external MC is required to meet the requirements called out in the latest NC-SI specification as it relates to this interface.

12.8.1.3 Reference Schematic

The following reference schematic (provides connectivity requirements for single and multi-drop applications. This configuration only has a single connection to the MC. The network device also supports multi-drop NC-SI configuration architecture with software arbitration support from the management controller.

See the NC-SI specification for connectivity requirements for multi-drop applications.

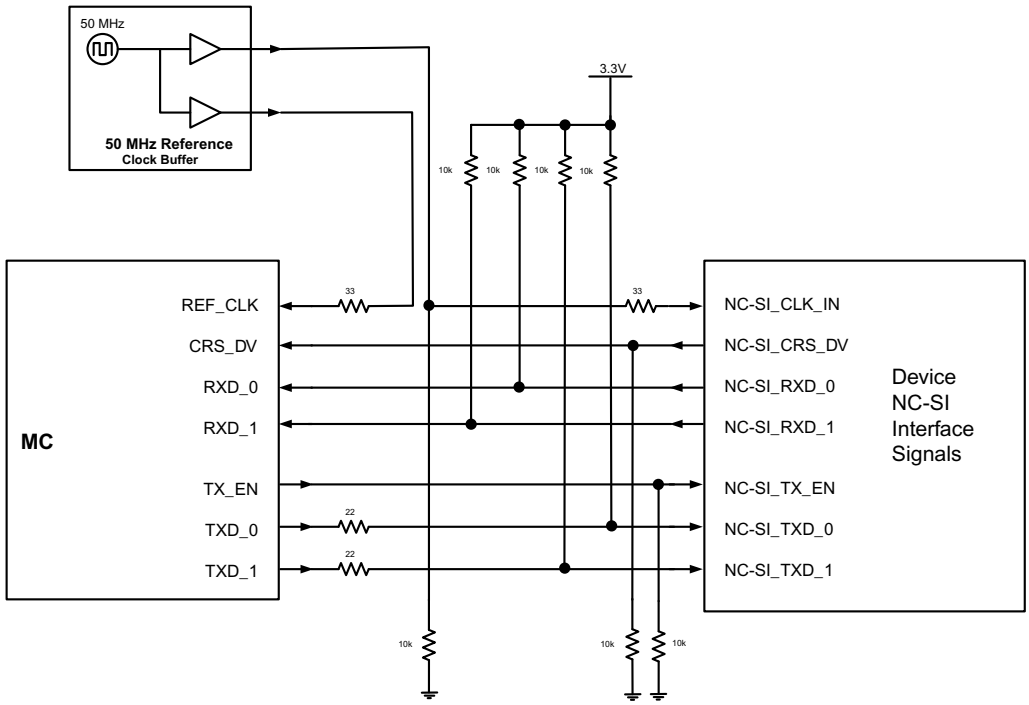


Figure 12-8 NC-SI Connection Schematic: Single-Drop Configuration

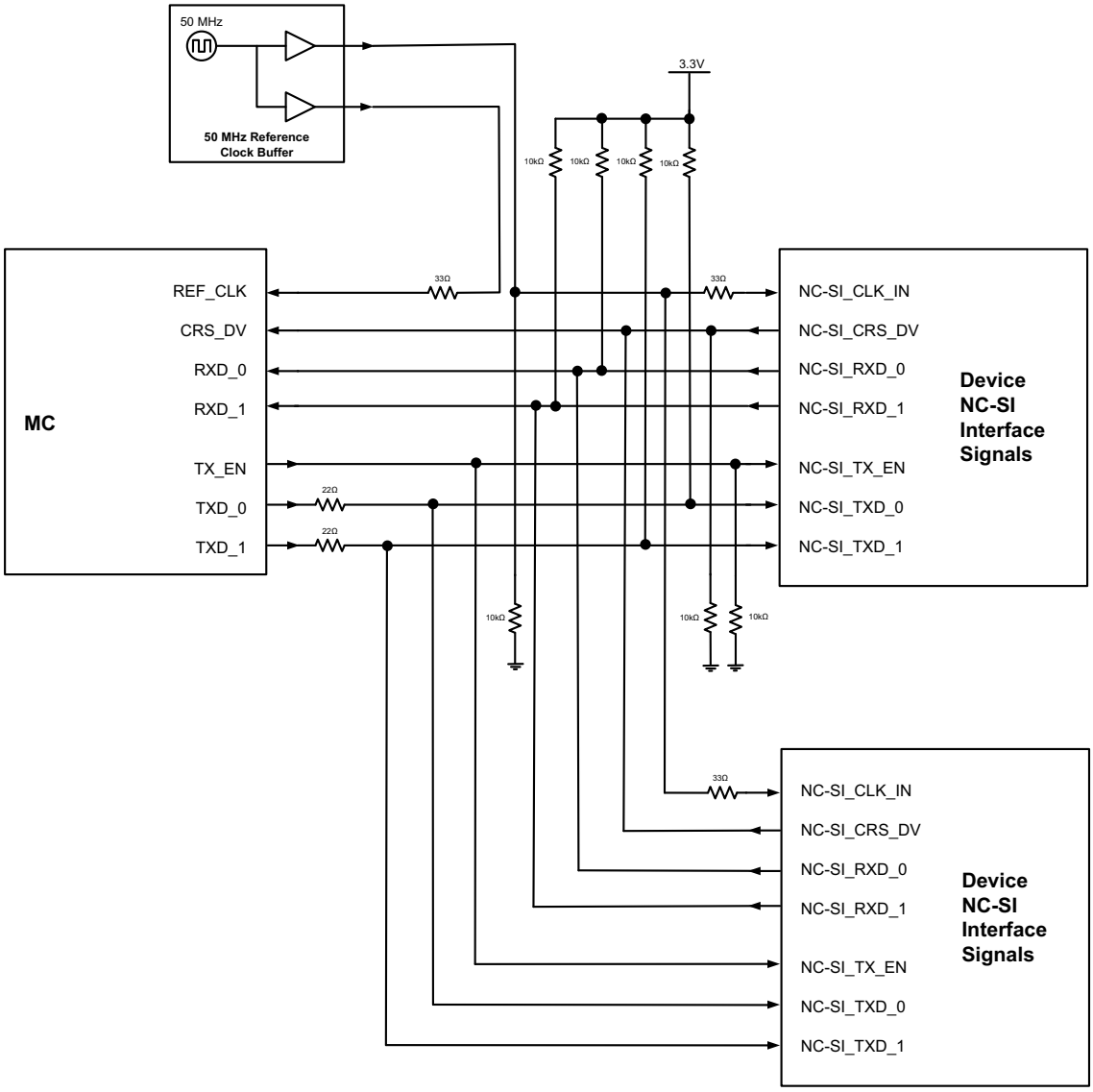


Figure 12-9 NC-SI Connection Schematic: Multi-Drop Configuration

12.8.2 Layout Requirements

12.8.2.1 Board Impedance

The NC-SI signaling interface is a single ended signaling environment and as such Intel recommends a target board and trace impedance of 50 Ohms plus 20% and minus 10%. This impedance ensures optimal signal integrity and quality.

12.8.2.2 Trace Length Restrictions

The recommended maximum trace lengths for each circuit board application is dependent on the number drops and the total capacitive loading from all the trace segments on each NC-SI signal net. The number via's must also be considered. Circuit board material variations and trace etch process variations affect the trace impedance and trace capacitance. For each fixed design, highest trace capacitance occurs when trace impedance is lowest.For the FR4 board stack-up provided in direct connect applications, the maximum length for a 50 ohm NC-SI trace would be approximately 9 inches on a minus 10% board impedance skew. This ensures that signal integrity and quality are preserved and enables the design to comply with NC-SI electrical requirements.

For special applications which require longer NC-SI traces, the total functional NC-SI trace length can be extended with non-compliant rise time by:

- providing good clock and signal alignment
- testing with the target receiver to verify it meets setup and hold requirements.

For multi-drop applications, the total capacitance and the extra resistive loading affect the rise time. A multi-drop of two devices limits the total length to 8 inches. A multi-drop of four limits the total length to 6.5 inches. Capacitive loading of extra via's have a nominal effect on the total load.

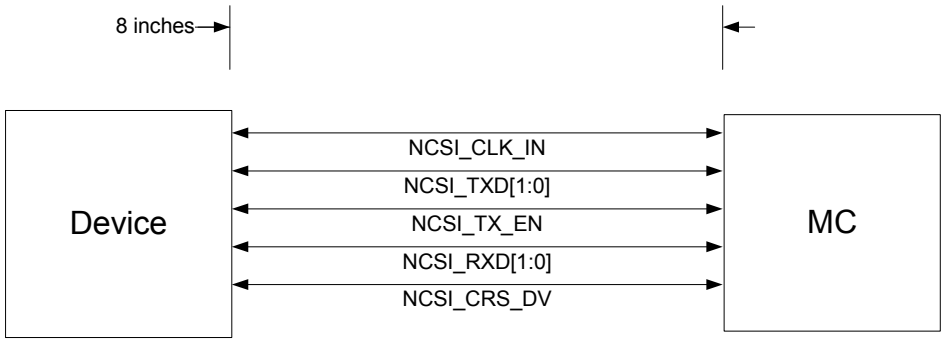


Figure 12-10 NC-SI Trace Length Requirement for Direct Connect

Table 12-6 shows how 7 more vias increase the rise time by 0.5ns. Again, longer trace lengths can be achieved.

Table 12-6 Stack Up, 7 Vias

Item	Value	Units
Trace width	4.5	mils
Trace thickness	1.9	mils
Dielectric thickness	3.0	mils
Dielectric constant	4.1	--
Loss Tangent	0.024	--
Nominal Impedance	50	Ohms
Trace Capacitance	1.39	pf/inch

Table 12-7 shows example trace lengths for the multi-drop topology of two and four represented in the figures that follow.

Table 12-7 Example Trace Lengths for Multi-Drop Topologies, 2 & 4

Multi-drop length parameter used in Figure 12-11 and Figure 12-12 .	Segment length example for multi drop configurations			
	Two drop configuration		Four drop configuration	
	Length (Inches)	Trace capacitance (Pf)	Length (Inches)	Trace capacitance (Pf)
L1	2	2.8	1.5	8.8
L2	4	5.6	2	16.1
L3	2	2.8	1	8.1
Total	8	11.1	6.5	35.6

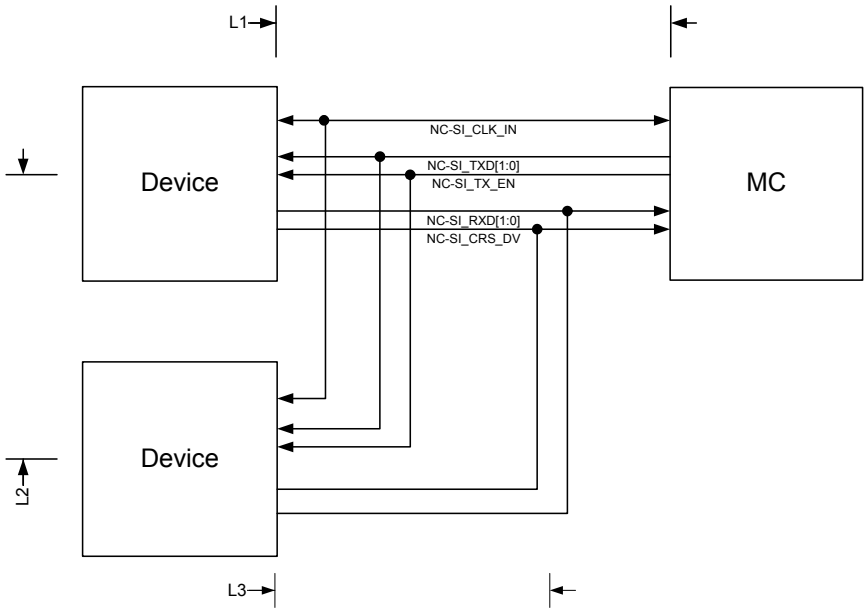


Figure 12-11 Example 2-Drop Topology

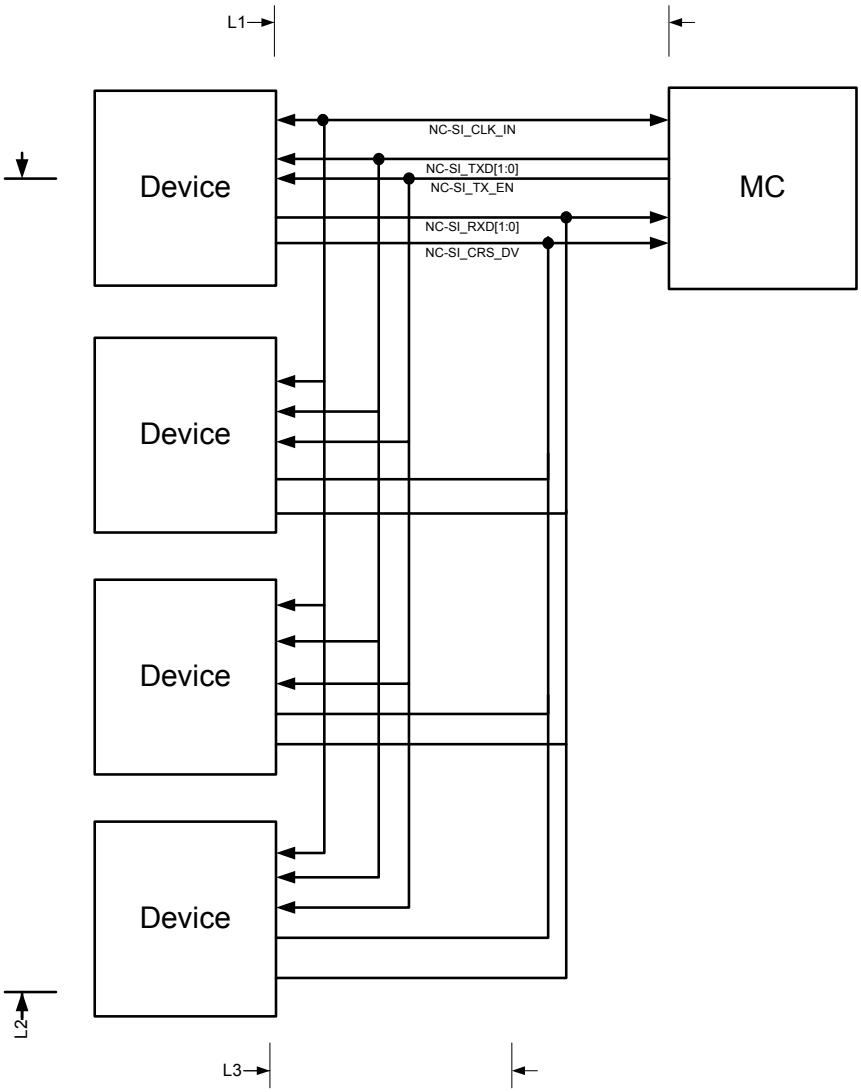


Figure 12-12 Example 4-Drop Topology

Table 12-8 Compliant NC-SI Maximum Length on a 50 ohm -10% Skew-board with Example Stack-up.

Topology	Total maximum compliant linear bus size (inches)	Number of vias	Approximate Net trace capacitance minus load capacitance (pf)
4 multi-drop	6.0	1	8.3
4 multi-drop	5.5	8	8.3
2 multi-drop	8.0	1	11.1
2 multi-drop	7.5	8	11.1
Point to point	9.0	1	12.5
Point to point	8.5	8	12.5

Extending NC-SI to a maximum 11ns rise time increases the maximum trace length.

Table 12-9 Functional NC SI maximum length on a 50 ohm -10% skew board with Example Stack-up (based on actual lab-measured solution)

Topology	Total maximum functional linear bus size (inches)	Number of vias	Approximate Net trace capacitance minus load capacitance (pf)
4 multi-drop	19	1	26.4
4 multi-drop	18	8	26.4
2 multi-drop	20	1	27.8
2 multi-drop	19	8	27.8
Point to point	22	1	30.6
Point to point	21	8	30.6

§ §

13 Thermal Management

This chapter provides methods for determining the operating temperature in a system based on case temperature. Case temperature is a function of the local ambient and internal temperatures of the component.

Properly designed solutions provide adequate cooling to maintain case temperature (Tcase) at or below what is listed in Table 13-2. This is accomplished by providing a low local ambient temperature, airflow, and creating a minimal thermal resistance to that local ambient temperature. Heatsinks and higher airflow may be required if temperatures exceed those listed.

13.1 Thermal Sensor and Thermal Diode

The I350 has both an on board thermal sensor and a thermal diode. The thermal sensor is read using the PCIe interface, the NC-SI interface, or the SMBUS interface.

The thermal diode can be read by forcing 1mA of current through the thermal diode and measuring the voltage.

The junction temperature can be determined from the following equations:

17x17mm Package
Tj=-695.834*Vd+657.277

25x25mm Package
Tj=-696.5785*Vd+657.7948

△V the voltage drop on the diode when 1mA forced

The thermal sensor and the thermal diode are not located in the same part of the die. In some situations, there can be up to a 7°C difference between the measurement on the thermal sensor and the measurement on the diode. The difference is most significant when the PCIe interface is enabled but the BASE-T interface is disabled.

See the figure below for approximate positioning.

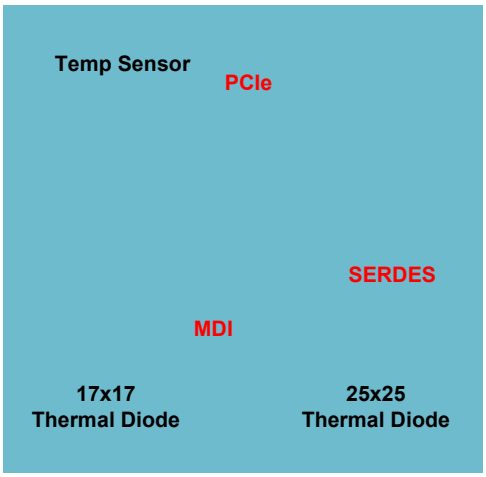


Figure 13-1 Approximate Location of Sensor and Diodes

13.2 Thermal Design Considerations

In a system, the temperature of a component is a function of both the system and component thermal characteristics. System-level thermal constraints consist of the local ambient temperature at the component, the airflow over the component and surrounding board, and the physical constraints surrounding the component that may limit the size of a thermal enhancement (heat sink).

A component's case/die temperature depends on:

- component power dissipation
- size
- packaging materials (effective thermal conductivity)
- type of interconnection to the substrate and motherboard
- presence of a thermal cooling solution
- power density of the substrate, nearby components, and motherboard

These parameters are pushed by the continued trend of technology to increase performance levels (higher operating speeds, MHz) and power density (more transistors). As operating frequencies increase and packaging size decreases, power density increases and the thermal cooling solution space and airflow become more constrained. The result is an increased emphasis on system design to ensure that thermal design requirements are met for each component in the system.

Note: Operation outside the functional limit can degrade system performance, cause logic errors, or cause device and/or system damage. Temperatures exceeding the maximum operating limits may result in irreversible changes in the device operating characteristics. Sustained operation at component maximum temperature limit may affect long-term device reliability.

13.3 Terminology

The following terminology is used in this chapter:

- **PBGA Plastic Ball Grid Array**: A surface-mount package using a BGA structure whose PCB-interconnect method consists of Pb-free solder ball array on the interconnect side of the package and attached to a plastic substrate material. An integrated heat spreader (**IHS**) may be present for larger PBGA packages for enhanced thermal performance (but IHS is not present for the I350).
- **Junction**: Refers to a P-N junction on the silicon. In this document, it is used as a temperature reference point (for example, Θ_{JA} refers to the "junction" to "ambient" thermal resistance).
- **Ambient**: Refers to local ambient temperature of the bulk air approaching the component. It can be measured by placing a thermocouple approximately 1"inch upstream from the component edge.
- **Lands**: The pads on the PCB to which BGA balls are soldered.
- **PCB**: Printed circuit board.
- **Printed Circuit Assembly (PCA)**: An assembled PCB.
- **Thermal Design Power (TDP)**: The estimated maximum possible/expected power generated in a component by a realistic application. Use Maximum power requirements listed in Table 13-2.
- **LFM:** Linear feet per minute (airflow).
- **Θ_{JA} (Theta JA)**: Thermal resistance junction-to-ambient, °C/W.
- **Ψ_{JT} (Psi JT)**: Junction-to-top (of package) thermal characterization parameter, °C/W. Ψ_{JT} does not represent thermal resistance, but instead is a characteristic parameter that can be used to convert between Tj and Tcase when knowing the total TDP. Ψ_{JT} is easy to characterize in simulations or measurements, and is equal to Tj minus Tcase divided by the total TDP. This parameter can vary by environment conditions like heat sink and airflow.

13.4 Thermal Specifications

The thermal solution must maintain a case temperature at or below the values specified in Table 13-2. System-level or component-level thermal enhancements are required to dissipate the generated heat to ensure the case temperature never exceeds the maximum temperatures listed. Table 13-1 lists the thermal performance parameters per JEDEC JESD51-2 standard.

In Table 13-1, the Θ_{JA} values should be used as reference only and can vary by system environment. Ψ_{JT} values also can vary by system environment. They are given in Table 13-1 as the maximum value for I350 simulations.

Analysis indicates that real applications are unlikely to cause the I350 to be at Tcase-max for sustained periods of time, given that Tcase can reasonably be expected to be a distribution of temperatures. Sustained operation at Tcase-max may affect long-term reliability of the I350 and the system; sustained operation at Tcase-max should be evaluated during the thermal design process and steps taken to further reduce the Tcase temperature.

Good system airflow is critical to dissipate the highest possible thermal power. The size and number of fans, vents, and/or ducts, and, their placement in relation to components and airflow channels within the system determine airflow. Acoustic noise constraints may limit the size and types of fans, vents and ducts that can be used in a particular design.

To develop a reliable, cost-effective thermal solution, all of the system variables must be considered. Use system-level thermal characteristics and simulations to account for individual component thermal requirements.

Table 13-1 Package Thermal Characteristics in Standard JEDEC Environment

Package	Θ_{JA} (°C/W)	Ψ_{JT} (°C/W)
17 mm PBGA[1]	22.6[7]	2.90[9]
17 mm PBGA - HS (19 x 6.3mm height)[2]	17.7[8]	2.90[9]
17 mm PBGA –HS (30 x 12mm height)[3]	16.6[8]	2.90[9]
17 mm PBGA–HS (25 x 7mm height)[4]	15.6[8]	2.90[9]
17 mm PBGA–HS (7 x 10mm height)[5]	14.1[8]	2.90[9]
17 mm PBGA–HS (40 x 10mm height)[6]	13.1[8]	2.90[9]

Notes:

1. Integrated Heat Spreader. The I350 is a PBGA
2. Heat sink 19 x 19 x 6.3mm
3. Heat sink 30 x 30 x 12mm
4. Heat sink 25 x 25 x 7mm
5. Heat sink 27 x 27 x 10mm
6. Heat sink 40 x 40 x 10mm
7. Integrated Circuit Thermal Measurement Method-Electrical Test Method EIA/JESD51-1, Integrated Circuits Thermal Test Method.
   Environmental Conditions - Natural Convection (Still Air), No Heat sink attached EIAJESD51-2.
8. Natural Convection (Still Air), Heat sink attached.
9. Psi_JT is given as maximum value for a worst-case I350 scenario, and may vary to a lesser value in some scenarios.

Table 13-2 I350 Line Absolute Thermal Maximum Rating (°C)

APPLICATION	Measured TDP (W)[1]	Tcase Max-hs[2] (°C)[3]
I350	4.0 @ 123 °C Tj_max	111

Notes:

1. Power value shown in Table 2 is measured maximum power, also known as Thermal Design Power (TDP). TDP is a system design target associated with the maximum component operating temperature specifications. Maximum power values are determined based on typical DC electrical specification and maximum ambient temperature for a worst-case realistic application running at maximum utilization.
2. Tcase Max-hs is defined as the maximum case temperature with the Default Enhanced Thermal Solution attached.
3. This is a not to exceed maximum allowable case temperature.

The thermal parameters defined above are based on simulated results of packages assembled on standard multi layer 2s2p 1.0-oz Cu layer boards in a natural convection environment. The maximum case temperature is based on the maximum junction temperature and defined by the relationship, maximum Tcase = Tjmax - (Ψ_{JT} x Power) where Ψ_{JT} is the junction-to-top (of package) thermal characterization parameter. If the case temperature exceeds the specified Tcase max, thermal enhancements such as heat sinks or forced air will be required. Θ_{JA} is the thermal resistance junction-to-ambient of the package.

13.4.1 Case Temperature

The I350 is designed to operate properly as long as Tcase rating is not exceeded. Section 6.1 discusses proper guidelines for measuring the case temperature.

13.5 Thermal Attributes

13.5.1 Designing for Thermal Performance

Section 13.10, "Heatsink and Attach Suppliers" and Section 13.11, "PCB Guidelines" document the PCB and system design recommendations required to achieve I350 thermal performance.

13.5.2 Typical System Definition

A system with the following attributes was used to generate thermal characteristics data:

- A heatsink case, see Section 13.6.2, "Default Enhanced Thermal Solution".
- A JEDEC JESD 51-9 standard 2s2p Board.

Keep the following in mind when reviewing the data that is included in this document:

- All data is preliminary and is not validated against physical samples.
- Your system design may be significantly different.
- A larger board (more than six copper layers) may improve I350 thermal performance.

13.5.3 Package Mechanical Attributes

For information on package attributes, see Chapter 11, "Electrical/Mechanical Specification".

13.5.4 Package Thermal Characteristics

See the table for an aid in determining the optimum airflow and heatsink combination for the I350. The table shows Tcase as a function of airflow and ambient temperature at the Thermal Design Power (TDP) for a typical system. Your system design may vary. Flotherm* models are available upon request. Contact your local Intel representative.

Table 13-3 Expected Tcase (°C) for Five Heat Sinks at 4.0 W (JEDEC Card)

Case Temperature (Max = 111C)									
No Heat Sink		Airflow (LFM)								
		0	50	100	150	200	250	300	350	400
Ambient Temperature (C)	45	120.9	117.2	113.8	111.2	109.1	107.3	106.0	104.9	103.8
	50	126.1	121.9	118.5	116.0	113.9	112.2	110.8	109.7	108.7
	55	131.2	126.7	123.3	120.8	118.7	117.0	115.7	114.6	113.6
	60	135.1	131.2	127.8	125.3	123.3	121.8	120.5	119.4	118.5
	65	139.3	135.9	132.5	130.1	128.1	126.4	125.1	124.0	123.0
	70	143.6	140.6	137.3	134.8	132.9	131.2	129.9	128.9	127.9
	75	147.9	145.2	142.0	139.6	137.7	136.0	134.8	133.7	132.8
	80	152.3	149.9	146.7	144.4	142.5	140.9	139.6	138.6	137.6
	85	156.7	154.6	151.4	149.1	147.3	145.7	144.4	143.4	142.5

Rose City Heat Sink		Airflow (LFM)								
		0	50	100	150	200	250	300	350	400
Ambient Temperature (C)	45	93.6	86.4	79.0	73.9	70.8	68.8	67.3	66.3	65.4
	50	98.4	91.2	83.9	78.9	75.7	73.7	72.3	71.2	70.4
	55	103.4	95.8	88.8	83.9	80.7	78.7	77.2	76.2	75.3
	60	107.5	100.5	93.2	88.3	85.4	83.3	82.0	81.1	80.3
	65	111.6	105.2	98.1	93.1	90.0	88.1	87.0	85.9	85.2
	70	116.0	109.8	103.0	98.0	94.9	93.0	91.7	90.7	90.1
	75	120.3	114.6	107.8	103.0	99.8	98.0	96.6	95.7	94.9
	80	124.5	119.3	112.6	107.9	104.8	102.9	101.6	100.6	99.9
	85	128.9	124.0	117.4	112.8	109.7	107.8	106.5	105.6	104.8

Table 13-3 Expected Tcase (°C) for Five Heat Sinks at 4.0 W (JEDEC Card)

AAVID-Thermalloy Heat Sink		Airflow (LFM)								
		0	50	100	150	200	250	300	350	400
Ambient Temperature (C)	45	91.1	85.2	78.3	73.1	69.8	67.3	65.5	64.0	62.8
	50	96.6	89.9	82.7	78.1	74.7	72.0	70.1	68.7	67.7
	55	100.9	94.5	87.5	82.6	79.3	76.9	75.1	73.7	72.5
	60	105.1	99.0	92.3	87.5	84.2	81.8	80.0	78.6	77.5
	65	109.4	103.5	97.1	92.3	89.1	86.8	85.0	83.6	82.4
	70	113.8	108.0	101.9	97.2	94.0	91.7	90.0	88.5	87.4
	75	118.2	112.6	106.6	102.1	98.9	96.6	94.9	93.5	92.4
	80	122.6	117.1	111.4	107.0	103.8	101.6	99.8	98.5	97.4
	85	127.0	121.8	116.2	111.8	108.7	106.5	104.8	103.4	102.3

Alpha Heat Sink LPD40-10B		Airflow (LFM)								
		0	50	100	150	200	250	300	350	400
Ambient Temperature (C)	45	89.2	83.0	78.2	75.5	73.6	72.4	71.5	70.8	70.3
	50	93.3	87.6	83.0	80.4	78.6	77.4	76.5	75.8	75.2
	55	97.7	92.2	87.9	85.2	83.5	82.3	81.4	80.7	80.2
	60	102.1	96.8	92.7	90.1	88.2	87.1	86.2	85.7	85.2
	65	106.5	101.4	97.5	95.0	93.1	92.0	91.2	90.5	90.1
	70	111.0	106.0	102.3	99.9	98.1	96.9	96.1	95.5	95.0
	75	115.5	110.7	107.2	104.8	103.0	101.9	101.1	100.4	99.9
	80	120.0	115.4	112.0	109.7	108.0	106.8	106.0	105.4	104.9
	85	124.6	120.1	116.9	114.6	112.9	111.7	111.0	110.4	109.8

Table 13-3 Expected Tcase (°C) for Five Heat Sinks at 4.0 W (JEDEC Card)

Alpha Heat Sink LPD25-7B		Airflow (LFM)								
		0	50	100	150	200	250	300	350	400
Ambient Temperature (C)	45	100.0	93.5	87.5	84.4	82.4	80.9	79.8	78.8	78.0
	50	104.4	98.2	92.3	89.3	87.1	85.8	84.7	83.7	83.0
	55	108.8	102.8	97.1	93.9	92.0	90.6	89.5	88.7	87.9
	60	113.0	107.4	101.9	98.8	96.9	95.5	94.4	93.5	92.8
	65	117.5	112.1	106.7	103.6	101.7	100.4	99.3	98.5	97.7
	70	122.0	116.7	111.6	108.5	106.6	105.3	104.2	103.4	102.6
	75	126.5	121.4	116.4	113.3	111.5	110.2	109.1	108.3	107.5
	80	131.0	126.0	121.2	118.2	116.3	115.0	114.0	113.2	112.5
	85	135.5	130.7	126.0	123.0	121.2	119.9	118.9	118.1	117.4

Alpha Heat Sink Z19-6.3B		Airflow (LFM)								
		0	50	100	150	200	250	300	350	400
Ambient Temperature (C)	45	107.6	101.6	97.1	94.1	91.7	89.7	88.2	86.8	85.7
	50	111.7	106.1	101.8	98.9	96.4	94.6	93.0	91.7	90.6
	55	115.9	110.7	106.5	103.6	101.2	99.4	97.8	96.6	95.5
	60	119.8	115.3	111.2	108.3	106.0	104.2	102.7	101.4	100.4
	65	124.1	120.0	115.9	113.1	110.9	109.0	107.5	106.3	105.2
	70	128.3	124.6	120.7	117.9	115.7	113.9	112.4	111.2	110.1
	75	132.7	129.2	125.4	122.7	120.5	118.7	117.3	116.0	115.0
	80	137.0	133.8	130.1	127.5	125.3	123.6	122.1	120.9	119.8
	85	141.4	138.5	134.8	132.2	130.1	128.4	127.0	125.8	124.7

Note: The Red blocked value(s) indicate airflow/ambient combinations that exceed the allowable case temperature for the I350 line at 4.0 W.

13.6 Thermal Enhancements

One method used to improve thermal performance is to increase the device surface area by attaching a metallic heatsink to the component top. Increasing the surface area of the heatsink reduces the thermal resistance from the heatsink to the air, increasing heat transfer.

13.6.1 Clearances

To be effective, a heatsink should have a pocket of air around it that is free of obstructions.

13.6.2 Default Enhanced Thermal Solution

If you have no control over the end-user's thermal environment or if you wish to bypass the thermal modeling and evaluation process, use the default solutions (see Section 13.6.2, "Default Enhanced Thermal Solution"). These solutions replicate the performance defined in Table 13-3 at Thermal Design Power (TDP). If after implementing the Recommended Enhanced Thermal Solution, the case temperature continues to exceed allowable values additional cooling is needed. This additional cooling may be achieved by improving airflow to the component and/or adding additional thermal enhancements.

13.6.3 Extruded Heatsinks

If required, the following extruded heatsinks are the suggested. Figure 13-2 through Figure 13-6 shows the multiple profiles.

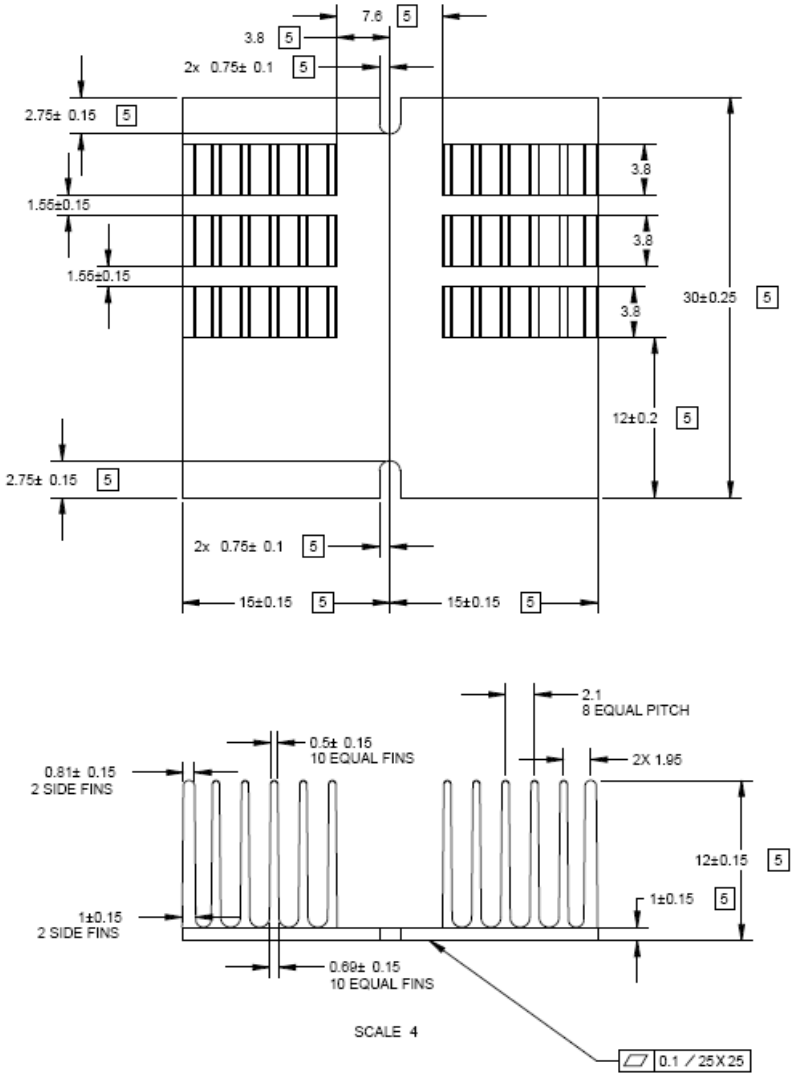


Figure 13-2 Rose City 12 mm Height Passive Heat Sink

Width	Length	Height	Fin Thickness Across Width	Fin Thickness Across Length	Base Thickness	# of fins across width	# of fins across length
27mm	27mm	10mm	0.90mm	0.93mm	1.50mm	12	13

Mechanical Outline Drawing

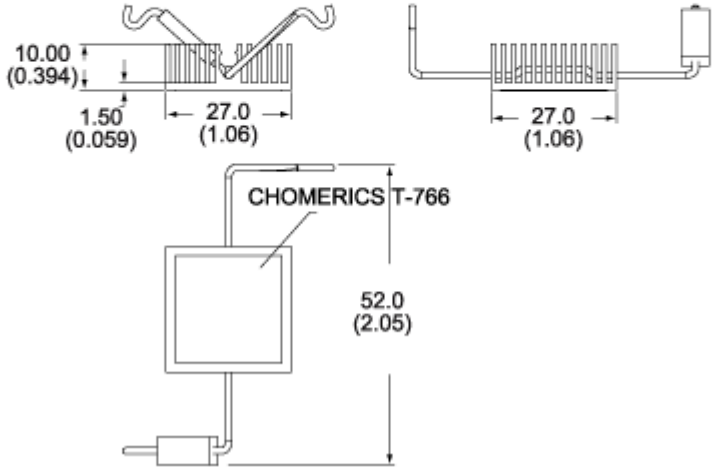
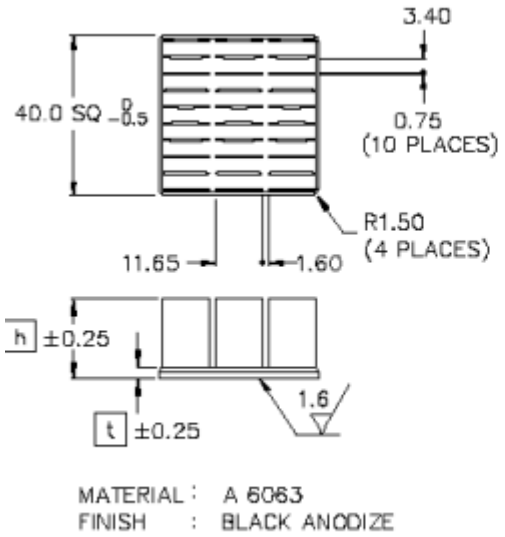


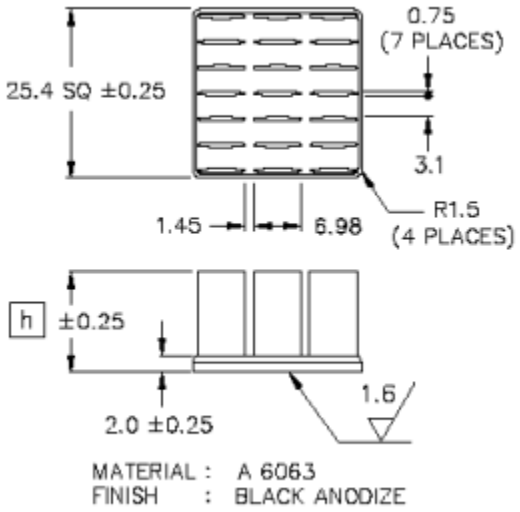
Figure 13-3 10mm Height Passive Heat Sink (AAVID Thermalloy PN: 374324B60023G)



MODEL	HEIGHT [h]	THICKNESS [t]	WEIGHT (grams)
LPD40-3B	3	2.0	9.4
LPD40-4B	4		10.0
LPD40-5B	5		10.6
LPD40-6B	6		11.2
LPD40-7B	7		11.8
LPD40-10B	10	3.0	16.4
LPD40-15B	15		19.2
LPD40-20B	20		22.0
LPD40-25B	25		24.8
LPD40-30B	30		27.6
LPD40-35B	35		30.4

Dimensions : mm

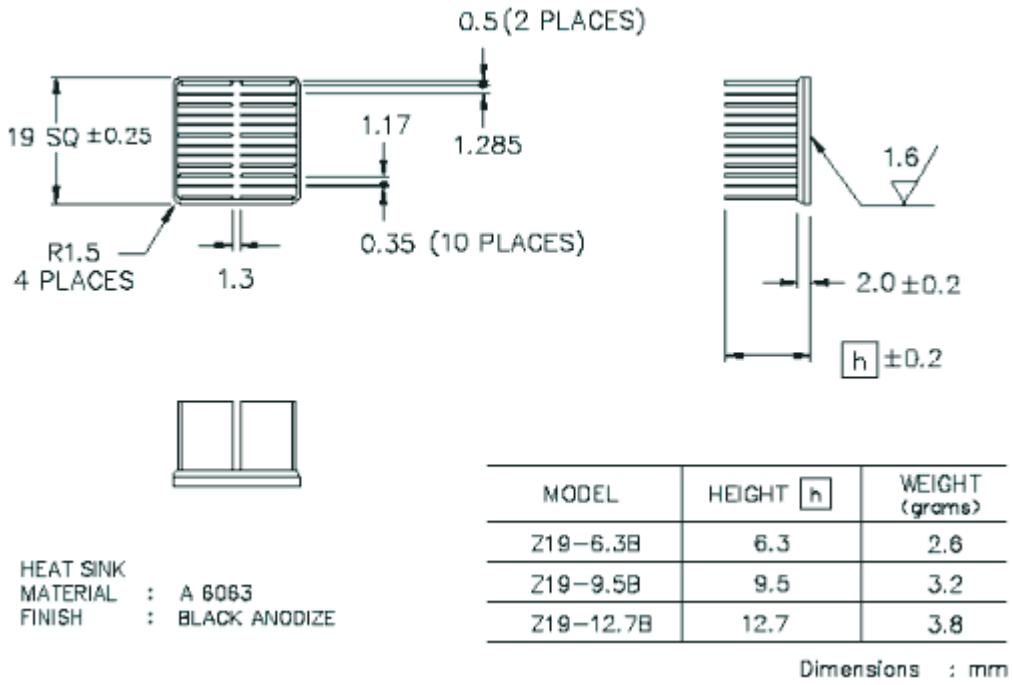
Figure 13-4 10mm Height Passive Heat Sink (Alpha Novatech, Inc. PN: LPD40-10B)



MODEL	HEIGHT [h]	WEIGHT (grams)
LPD25-3B	3	4.1
LPD25-4B	4	4.3
LPD25-5B	5	4.5
LPD25-6B	6	4.7
LPD25-7B	7	4.9
LPD25-10B	10	5.5
LPD25-15B	15	6.5
LPD25-20B	20	7.5
LPD25-25B	25	8.5

Dimensions : mm

Figure 13-5 7mm Height Passive Heat Sink (Alpha Novatech, Inc. PN: LPD25-7B)



MODEL	HEIGHT h	WEIGHT (grams)
Z19-6.3B	6.3	2.6
Z19-9.5B	9.5	3.2
Z19-12.7B	12.7	3.8

Figure 13-6 6.3mm Height Passive Heat Sink (Alpha Novatech, Inc. PN: Z19-6.3B)

13.6.4 Attaching the Extruded Heatsink

An extruded heatsink may be attached using clips with a phase change thermal interface material. For attaching methods, contact the heatsink manufacturer.

13.6.4.1 Clips

A well-designed clip, in conjunction with a thermal interface material (tape, grease, etc.) often offers the best combination of mechanical stability and reworkability. Use of a clip requires significant advance planning as mounting holes are required in the PCB.

13.6.4.2 Thermal Interface (PCM45 Series)

The recommended thermal interface is the PCM45 series from Honeywell. PCM45 Series thermal interface pads are phase change materials formulated for use in high performance devices requiring minimum thermal resistance for maximum heat sink performance and component reliability. These pads consist of an electrically non-conductive, dry film that softens at device operating temperatures resulting in “greasy-like” performance. However, Intel has not fully validated the PCM45 Series TIM.

Each PCA, system and heatsink combination varies in attach strength. Carefully evaluate the reliability of double sided thermal interface tape attachments prior to high-volume use (see Section 5.5).

13.6.4.3 Maximum Static Normal Load

Maximum applied pressure should not exceed 20 psi.

Note: The PWB under the component must be fully supported to prevent bowing or flexing of the PWB.

The force must be applied perpendicular to the component.

The pressure needs to be evenly distributed on the top side of the component.

13.6.5 Reliability

Each PCA, system and heatsink combination varies in attach strength and long-term adhesive performance. Evaluate the reliability of the completed assembly prior to high-volume use. Reliability recommendations are in Table 13-4.

Table 13-4 Reliability Validation

Test[1]	Requirement	Pass/Fail Criteria[2]
Mechanical Shock	50G trapezoidal, board level 11 ms, 3 shocks/axis	Visual and Electrical Check
Random Vibration	7.3G, board level 45 minutes/axis, 50 to 2000 Hz	Visual and Electrical Check
High-Temperature Life	85 °C 2000 hours total Checkpoints occur at 168, 500, 1000, and 2000 hours	Visual and Mechanical Check
Thermal Cycling	Per-Target Environment (for example: -40 °C to +85 °C) 500 Cycles	Visual and Mechanical Check
Humidity	85% relative humidity 85 °C, 1000 hours	Visual and Mechanical Check

Notes:

1. Performed the above tests on a sample size of at least 12 assemblies from 3 lots of material (total = 36 assemblies).
2. Additional pass/fail criteria can be added at your discretion.

13.7 Thermal Interface Management for Heat-Sink Solutions

To optimize heatsink design, it is important to understand the interface between the silicon die and the heatsink base. Thermal conductivity effectiveness depends on the following:

- Bond line thickness
- Interface material area
- Interface material thermal conductivity

13.7.1 Bond Line Management

The gap between the silicon die and the heatsink base impacts heat-sink solution performance. The larger the gap between the two surfaces, the greater the thermal resistance. The thickness of the gap is determined by the flatness of the heatsink base, the silicon die, and the package encapsulant, plus the thickness of the thermal interface material (for example, PSA, thermal grease, epoxy) used to join the two surfaces.

13.7.2 Interface Material Performance

The following factors impact the performance of the interface material between the silicon die and the heatsink base:

- Thermal resistance of the material
- Wetting/filling characteristics of the material

13.7.2.1 Thermal Resistance of the Material

Thermal resistance describes the ability of the thermal interface material to transfer heat from one surface to another. The higher the thermal resistance, the less efficient the heat transfer. The thermal resistance of the interface material has a significant impact on the thermal performance of the overall thermal solution. The higher the thermal resistance, the larger the temperature drop required across the interface.

13.7.2.2 Wetting/Filling Characteristics of the Material

The wetting/filling characteristic of the thermal interface material is its ability to fill the gap between the package's top surface and the heatsink. Since air is an extremely poor thermal conductor, the more completely the interface material fills the gaps, the lower the temperature-drop across the interface, increasing the efficiency of the thermal solution.

13.8 Measurements for Thermal Specifications

Determining the thermal properties of the system requires careful case temperature measurements. Guidelines for measuring the I350 case temperature are provided in Section 6.1.

13.8.1 Case Temperature Measurements

Maintain Tcase at or below the maximum case temperatures listed in Table 13-2 to ensure functionality and reliability. Special care is required when measuring the Tcase temperature to ensure an accurate temperature measurement. Use the following guidelines when making Tcase measurements:

- Measure the surface temperature of the case in the geometric center of the case top.
- Calibrate the thermocouples used to measure Tcase before making temperature measurements.
- Use 36-gauge (maximum) K-type thermocouples.

Care must be taken to avoid introducing errors into the measurements when measuring a surface temperature that is a different temperature from the surrounding local ambient air. Measurement errors may be due to a poor thermal contact between the thermocouple junction and the surface of the package, heat loss by radiation, convection, conduction through thermocouple leads, and/or contact between the thermocouple cement and the heat-sink base (if used).

13.8.1.1 Attaching the Thermocouple (No Heatsink)

The following approach is recommended to minimize measurement errors for attaching the thermocouple with no heatsink:

- Use 36-gauge or smaller-diameter K-type thermocouples.
- Ensure that the thermocouple has been properly calibrated.
- Attach the thermocouple bead or junction to the top surface of the package (case) in the center of the heat spreader using high thermal conductivity cements.

Note: It is critical that the entire thermocouple lead be butted tightly to the heat spreader.

Attach the thermocouple at a 0° angle if there is no interference with the thermocouple attach location or leads (see Figure 13-3). This is the preferred method and is recommended for use with packages not having a heat sink.

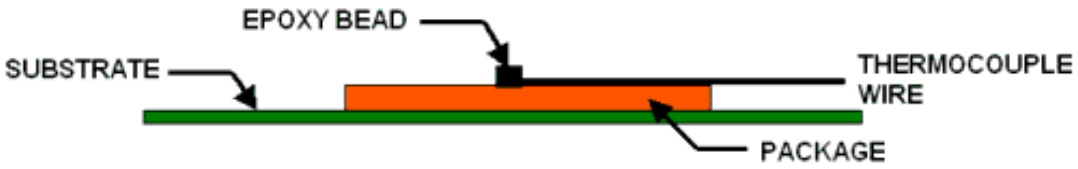


Figure 13-7 Technique for Measuring Tcase with 0° Angle Attachment, No Heatsink

13.8.1.2 Attaching the Thermocouple (Heatsink)

The following approach is recommended to minimize measurement errors for attaching the thermocouple with heatsink:

- Use 36-gauge or smaller diameter K-type thermocouples.
- Ensure that the thermocouple is properly calibrated.
- Attach the thermocouple bead or junction to the case's top surface in the geometric center using a high thermal conductivity cement.

Note: It is critical that the entire thermocouple lead be butted tightly against the case.

- Attach the thermocouple at a 90° angle if there is no interference with the thermocouple attach location or leads (see Figure 13-4). This is the preferred method and is recommended for use with packages with heatsinks.
- For testing purposes, a hole (no larger than 0.150" in diameter) must be drilled vertically through the center of the heatsink to route the thermocouple wires out.
- Ensure there is no contact between the thermocouple cement and heatsink base. Any contact affects the thermocouple reading.

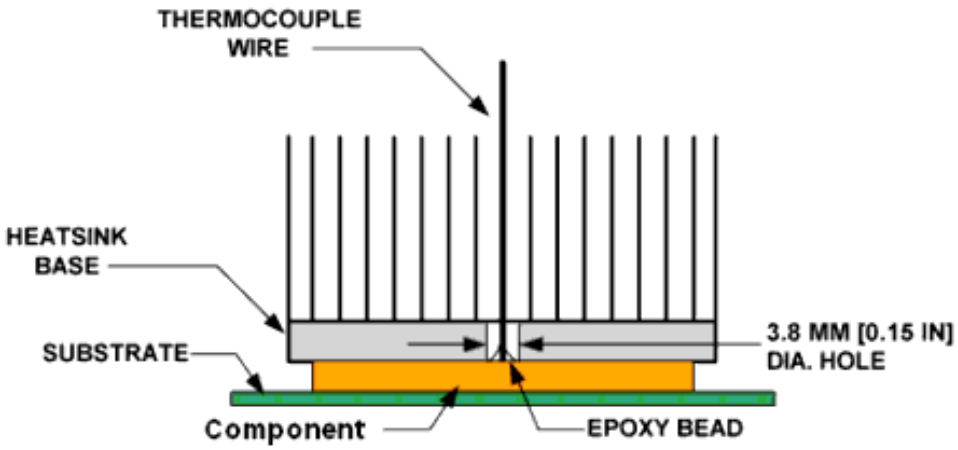


Figure 13-8 Technique for Measuring Tcase with 90° Angle Attachment

13.9 Thermal Diode

The I350 incorporates an on-die diode that may be used to monitor the die temperature (junction temperature). A thermal sensor located on the motherboard or a stand-alone measurement kit, may monitor the die temperature of the I350 for thermal management or characterization.

13.10 Heatsink and Attach Suppliers

Table 13-5 Heatsink and Attach Suppliers

Part	Part Number	Supplier	Contact
Alpha Heat Sinks	LPD40-10B LPD25-7B Z19-6.3B	Alpha Novatech, Inc	Sales Aplha Novatech, Inc. 408-567-8082 sales@alphanovtech.com
Aavid-Thermalloy Heat Sink	374324B60023G	Aavid Thermalloy	Harish Rutti 67 Primrose Dr. Suite 200 Laconia, NH 03246 Business: 972-633-9371 x27
Rose City Heat Sink	E66546-001	Intel Corp.	Please contact your Intel representative
PCM45 Series	PCM45F	Honeywell	North America Technical Contact: Paula Knoll 1349 Moffett Park Dr. Sunnyvale, CA 94089 Cell: 1-858-705-1274 Business: 858-279-2956 paula.knoll@honeywell.com

13.11 PCB Guidelines

The following general PCB design guidelines are recommended to maximize the thermal performance of PBGA packages:

- When connecting ground (thermal) vias to the ground planes, do not use thermal-relief patterns.
- Thermal-relief patterns are designed to limit heat transfer between the vias and the copper planes, thus constricting the heat flow path from the component to the ground planes in the PCB.
- As board temperature also has an effect on the thermal performance of the package, avoid placing the I350 adjacent to high-power dissipation devices.
- If airflow exists, locate the components in the mainstream of the airflow path for maximum thermal performance. Avoid placing the components downstream, behind larger devices or devices with heat sinks that obstruct the air flow or supply excessively heated air.

Note: The above information is provided as a general guideline to help maximize the thermal performance of the components.

14 Diagnostics

14.1 JTAG Test Mode Description

The I350 includes a JTAG (TAP) port compliant with the IEEE Standard Test Access Port and Boundary Scan Architecture 1149.6 Specification.

The TAP controller is accessed serially through the four dedicated pins TCK, TMS, TDI, and TDO. TMS, TDI, and TDO operate synchronously with TCK which is independent of all other clock within the I350. This interface can be used for test and debug purposes. System board interconnects can be DC tested using the boundary scan logic in pads. Table 14-1 shows TAP controller related pin descriptions. Table 14-2 describes the TAP instructions supported by the I350. The default instruction after JTAG reset is IDCODE.

Table 14-1 TAP Controller Pins

Signal	I/O	Description
TCK	In	Test clock input for the test logic defined by IEEE1149.1. Note: Signal should be connected to ground through a 1 kΩ pull-down resistor.
TDI	In	Test Data Input. Serial test instructions and data are received by the test logic at this pin. Note: Signal should be connected to VCC33 through a 1 kΩ pull-up resistor.
TDO	O/D	Test Data Output. The serial output for the test instructions and data from the test logic defined in IEEE1149.1. Note: Signal should be connected to VCC33 through a 1 kΩ pull-up resistor.
TMS	In	Test Mode Select input. The signal received at TMS is decoded by the TAP controller to control test operations. Note: Signal should be connected to VCC33 through a 1 kΩ pull-up resistor.

Table 14-2 TAP Instructions Supported

Instruction	Description	Comment
BYPASS	The BYPASS command selects the Bypass Register, a single bit register connected between TDI and TDO pins. This allows more rapid movement of test data to and from other components in the system.	IEEE 1149.1 Std. Instruction
EXTEST	The EXTEST Instruction allows circuitry or wiring external to the devices to be tested. Boundary-scan Register Cells at outputs are used to apply stimulus while Boundary-scan cells at input pins are used to capture data.	IEEE 1149.1 Std. Instruction

Table 14-2 TAP Instructions Supported (Continued)

Instruction	Description	Comment
SAMPLE / PRELOAD	The SAMPLE/PRELOAD instruction is used to allow scanning of the boundary scan register without causing interference to the normal operation of the device. Two functions can be performed by use of the Sample/Preload instruction. SAMPLE – allows a snapshot of the data flowing into and out of a device to be taken without affecting the normal operation of the device. PRELOAD – allows an initial pattern to be placed into the boundary scan register cells. This allows initial known data to be present prior to the selection of another boundary-scan test operation.	IEEE 1149.1 Std. Instruction
IDCODE	The IDCODE instruction is forced into the parallel output latches of the instruction register during the Test-Logic-Reset TAP state. This allows the device identification register to be selected by manipulation of the broadcast TMS and TCK signals for testing purposes, as well as by a conventional instruction register scan operation. The ID code value for the I350 A0 is 0x0151F013 (Intel's Vendor ID = 0x13, Device ID = 0x151F, Rev ID = 0x0) The ID code value for the I350 A1 is 0x1151F013 (Intel's Vendor ID = 0x13, Device ID = 0x151F, Rev ID = 0x1)	IEEE 1149.1 Std. Instruction
HIGHZ	The HIGHZ instruction is used to force all outputs of the device (except TDO) into a high impedance state. This instruction shall select the Bypass Register to be connected between TDI and TDO in the Shift-DR controller state.	IEEE 1149.1 Std. Instruction

§ §

Appendix A Changes from 82580

Note: This appendix summarizes changes in I350 relative to 82580.

Table A-1 Changes in the Programming Interface Relative to 82580

Feature	Registers/Descriptors	Description
Flows	Queue disable flow	• Modified queue disable flow in Section 4.6.9.2.
SR-IOV	PCIe capability Structures	• Added support for "IOV Capability Structure" registers and "Alternative RID Interpretation (ARI) capability structure" in extended PCIe configuration address space due to SR-IOV addition. Note changes relative to 82576 related to 4 ports support in ARI FDL field and IOV VF Stride field. • Added support for "Virtual Functions (VF) Configuration Space" registers defined in Section 9.7. Note new TLP and ACS capabilities not present in 82576. • Default VF device ID is now 0x1520.
	SR-IOV registers	Added back support for Virtual Function address space due to SR-IOV restore. Added following dedicated Virtual Function device registers due to SR-IOV support: • VTCTRL, VTStatus, VTFRTIMER, VTEICS, VTEIMS, VTEIMC, VTEIAC, VTEIAM, VTEICR, VTIVAR, VTIVAR_MISC, VFGPRLBC, VFGPTLBC, VFGORLBC and VFGOTLBC. • VTIVAR is different than 82576, as it supports only one queue pair events. • Only one queue pair registers per VF. Added following registers due to restore of SR-IOV: • VFMailbox[0 - 7], PFMailbox[0 - 7], VMBMEM, MBVFICR, MBVFIMR, VFLRE, VFRE, VFTE, QDE and VMVIR • CIAA and CIAD diagnostic registers. • Added GCR.IOV test mode (bit 1) • Added GCR.Ignore RID (bit 0) • Added CTRL_EXT.PFRSTD (bit14) • Added STATUS.Num VFs (bits 17:14). • Added Status.IOV Mode (bit 18) • Added to LVMMC.Legacy desc in RT/IOV (bit 27), LVMMC.Vlan Spoof (bit 26) and LVMMC.MAC Spoof (bit 25). • Removed field GCR_EXT.VT_Mode (bits 1:0).
	Receive Status descriptor	Added to receive status descriptor VM to VM loopback (LB) indication bit (bit 18) in RDESC.STATUS field.
	IVAR	Behavior of IVAR registers changed in SR-IOV mode so that all index fields allocated to the VF are read only.

Table A-1 Changes in the Programming Interface Relative to 82580 (Continued)

Feature	Registers/Descriptors	Description
Proxying and WOL	MANC	Added MPROXYE (Management Proxying Enable) bit (bit 30) to MANC register.
	WUC	Added PPROXYE (Port Proxying Enable) bit (bit 4) to WUC register. Added WUC.EN_APM_D0 bit (bit 5) to enable controlling if APM wake is generated in D0.
	WUFC and WUS	Added to WUFC and WUS registers options to wake-up or NS (bit 9), "NS Directed" (bit 10), ARP (bit 11) or FW_RST_WK (bit 31).
	PROXYFC and PROXYS	Added PROXYFC and PROXYS to define and report type of packets sent to management for Proxying.
	FWSTS	FWSTS register is per port and not shared
	RXCSUM	Added RXCSUM.ICMPv6XSUM bit (bit 10) to enable HW Neighbor Solicitation checksum calculation during Proxying.
	Host Slave Interface Commands	Added Host Proxying Commands to the FW SW interface via the Shared RAM interface.
VMDq support	Virtualization TX switch buffer registers	Added following registers due to addition of virtualization TX switch buffer: PBSWAC, TXSWC, SWDFPC, and SDPC. Added DPME (bit 2) in VMRCTL register. Added Bit *LVLAN* (Bit 20) in *VLVF* register. Added FBDPC statistic counter
	Anti spoofing	Registers added due to addition of anti spoofing protection: VMECM, SSVPC and WVBR In DTXCTL, added SPOOF_INT (bit 6)
	Advanced receive descriptors – Write Back format	Added back LB (17) bit in "Extended Status" field. Extended HDR_LEN to 12 bits to support 2K headers.
	Registers changes (relative to the 82576 virtualization support)	• Added VMOLR.UPE and VMOLR.VPE bits. • Expanded RAH/RAL to 32 sets. • Added RAH.TRMCST bit. • Expanded SRRCTL.BSIZEHEADER to support 2K buffers. • Added DVMOLR register • Added TQDPC statistics counters. • Made RQDPC RC for PF and RO for VF. • Added new registers to access VTIVAR and VTIVAR_MISC • Removed RPLOLR and VLAN strip and CRC strip fields in VMOLR. • Changed the default of previous CRC strip bits in VMOLR to zero. • Added LPBKFBDPC to count lost VM to VM packets. • Replaced DTXSWC with TXSWC (Address change - same layout) • Updated behavior of WVBR and MDFB registers. • VF assertion of VFMailbox.REQ bit causes interrupt due to ICR.SWMB assertion instead of ICR.VMMB assertion. •

Table A-1 Changes in the Programming Interface Relative to 82580 (Continued)

Feature	Registers/Descriptors	Description
Dummy function changes	PCIe capability Structures	Dummy function changes: • Command register - I/O access enable and Memory Access Enable are R/W. Interrupt Disable field is RO as one. • PCI Power Management Capability (0x40) - Next pointer is 0xA0 to skip MSi/MSI-X. • Power Management Capabilities (0x42)- PME_Support is RO 0. • CSR Access Via Configuration Address Space (0x98/0x9C) not available for dummy functions. • MSi/MSI-X/VPD capabilities not available for dummy functions. • PCI Express Capability Register (0xA0) - next pointer is 0x00 to skip VPD. • Device Capabilities 2 (0xC4) - Completion Timeout Disable Supported - set to RO 0b. • Device Control 2 (0xC8) - Completion Timeout Value and Completion Timeout Disable are RO zero. • PCIe Extended Capability Structure for dummy is AER -> ARI (if enabled) -> ACS. • Serial Number/SR-IOV/TPH/LTR not available in Dummy function.
Other PCIe changes	PCIe capability Structures	• Default Device ID is now 0x151F. • Added ACS capability (0x1D0 and 0x1D4) • Modified "Next Capability pointer" fields (Bits 31:20) in AER, Serial ID, ARI, TPH and LTR capabilities to reflect the new link list options. • Added support for ID-ordering (0xC8). • Modified AER Capability Version field (bits 19:16) to 0x2 in "PCIe CAP ID" (0x100; RO).
	PMCSR	Changed default value of No_Soft_Reset bit (bit 3) in PMCSR to 1, to meet PCIe Specification recommendations for MFD (Multi-function Devices).
IDO support	Device Control 2	Added IDO related fields to PCIe Device Control 2 register.
ASPM Optionality	Link Capabilities	Added "ASPM Optionality Compliance" bit (bit 22) to the "Link Capabilities" Register (0xAC; RO)
DMA Coalescing	DMCRTRH	Renamed DMCRTRH.LRPRCW to DMCRTRH.LRPRPW to indicate that low rate was detected in previous window and not in current window.
	DMACR	• Added move to "deepest Lx" mode (value 11b) to DMACR.DMAC_Lx field. Default value of field changed to 11b. • Added DC_LPBKW_EN (bit 14), DC_BMC2OSW_EN (bit 15), DC_FLUSH (bit 24) and EXIT_DC (bit 25) to DMACR register.
	DMCTLX	Added DC_FLUSH (bit 30) and DCFLUSH_DIS (bit 31) to the DMCTLX register.
Other registers	PSRTYPE and RPLPSRTYPE registers	Added to PSRTYPE and RPLPSRTYPE the capability to split on MAC header only.
	LVMMC	Added to the LVMMC register the VLAN IERR bit (bit 26). Added LVMMC.MVF_MACC bit (bit 19) to indicate that memory access initiated by VF terminated with an Unsupported Request (UR) or Completer Abort (CA).
	WUC	APM wake-up is disabled in D0. If APM wake-up is required software should not disable APM wake in WUC register on entry to D0, to allow for APM wake on system crash.
	DTXCTL	Changed default value of SPOOF_INT bit (bit 6) to 0b and removed OutOfSyncEnable bit (bit 4).
	STATUS	Added STATUS.PF_RST_DONE bit to indicate that internal reset sequence completed.
	BARCTRL	Changed address of BARCTRL register to 0x5BFC from 0x5BBC.
Other registers	SWSM	Added SWMB_CLR bit (bit 31) to the SWSM register to reset the SWMBWR, SWMB0, SWMB1, SWMB2 and SWMB3 Software Mailbox registers.

Table A-1 Changes in the Programming Interface Relative to 82580 (Continued)

Feature	Registers/Descriptors	Description
Flash and EEPROM	FLA	• Changed FLA_ABORT bit (bit 7) to read only and added FLA_CLR_ERR bit (bit 8) to FLA register. • Added FL_BAR_WR bit (bit 29) to the FLA register to indicate occurrence of Flash write or Erase access via direct memory (BAR) access. • Updated Flash Write Flow in Section 3.3.4.2.
	EEC	• Changed EE_BLOCKED bit (bit 15) and EE_ABORT bit (bit 16) to Read Only and added EE_CLR_ERR bit (bit 18) to EEC register. • Added EEC.EE_DET bit (bit 19) to indicate detection of EEPROM • Added EEC.EE_RD_TIMEOUT bit (bit 17) to indicate read abort when executing EEPROM read via the EERD register.
	EEARBC	Added VALID_PCIE bit (bit 3) to "EEPROM Auto Read Bus Control - EEARBC (0x1024; R/W)" register to enable write to PCIe PHY EEPROM bits.
	EEMNGCTL	Added EEMNGCTL_CLR_ERR bit (bit 29) and TIMEOUT bit (bit 30) to EEMNGCTL register.
Interrupts	ICR, ICS, IMS and IMC	• Renamed bit DOUTSYNC (bit 28) to MDDET (Malicious Driver Detect) in ICR, ICS, IMS and IMC registers. • Added the Thermal Sensor Event (TS) interrupt bit (bit 23) to the ICR, ICS, IMS and IMC registers.
Manageability Changes	MDEF/MDEF_EXT registers	Added two MAC addresses to the AND and OR sections. Added MDEF_EXT.APPLY_to _host _traffic bit.
	MANC	Added EN_BMC2NET bit. Modified description of RCV_TCO_EN bit. Changed the "TCO Reset" bit (bit 16) and "FW Reset" bit (bit 14) in the MANC register to R/W1C.
	host interface	Modified failover command to support 4 ports (relative to 82576).
	SW_FW_SYNC register	Added SW_MNG_SM bit (bit 10) to SW_FW_SYNC (0x5B5C) register to allow synchronization between drivers when accessing Management Host Interface.
	Manageability Statistics registers	Added BMTPDC register. Changed BMPDC name to BMRPDC.
	BUPTC, BMPTC, BBPTC, BSCC, BMCC, BUPRC and other BMC statistical counters.	Removed requirement for TCTL.EN or RCTL.RXEN to be set for BUPTC, BMPTC, BBPTC, BSCC, BMCC, BUPRC and other BMC statistical counters to count.
Manageability Changes (Cont')	FWSM	Added bit 31 - Factory MAC address restored. Updated Error values in FWSM.Ext_Err_Ind field. Should be read after a reset was issued and the relevant EEMNGCTL.CFG_DONE bit was set to 1b.
	THHIGHTC, THMIDTC and THLOWTC	Added Thermal Sensor BMC Threshold and Hysteresis bits (bits 26 and 27) to the THHIGHTC, THMIDTC and THLOWTC registers. Bits are R/W by management and RO by Host.
Thermal Sensor	THMJT, THLOWTC, THMIDTC, THHIGHTC, THSTAT and THACNFG	Added following thermal sensor registers: THMJT, THLOWTC, THMIDTC, THHIGHTC, THSTAT and THACNFG.
	WUFC and WUS	Added wake-up as a result of Thermal Sensor event bit (THS_WK - bit 13) to WUFC and WUS registers.
	SW_FW_SYNC	Added the SW_PWRTS_SM (bit 7) and FW_PWRTS_SM (bit 23) semaphore bits to the SW_FW_SYNC (0x5B5C) register to enable taking ownership of Thermal Sensor and LVR/SVR registers.
	ICR, ICS, IMS and IMC	Added THS bit (thermal sensor interrupt - bit 23) in ICR, ICS, IMS and IMC interrupt registers.
OS to BMC Changes	OS to BMC Statistics registers	Added B2OSPC, B2OGPRC, O2BGPTC, O2BSPC and MNGFBDPC registers.
	MANC register	Added RCV_TCO_EN, EN_BMC2OS and EN_BMC2NET
	RDESC.STATUS Descriptor Status	Added BMC (19) - Packet received from BMC bit

Table A-1 Changes in the Programming Interface Relative to 82580 (Continued)

Feature	Registers/Descriptors	Description
EEE support	LTRC register	Added EEEMS_EN bit (bit 5) to LTRC (0x01A0; RW) register.
	EEER register	Added EEER register to support EEE programmability
	EEE Statistics	Added "EEE RX LPI Count" (RLPIC) and "EEE TX LPI Count" (TLPIC) statistics counters
	PHY registers	Added PHY related EEE (IEEE802.3az) registers and EMIADD (address 16d) and EMIDATA (address 17d) registers to access these registers in extended PHY memory address space.
10/100/ 1GBASE-T PHY	IPCNFG register	Added MDI_Flip configuration bit (bit 0) to Internal PHY Configuration (IPCNFG) register. Added bits 10BASE-TE bit (bit 1), EEE_100M_AN bit (bit 2) and EEE_1G_AN (bit 3) to the "Internal PHY Configuration" - IPCNFG register.
	PHY Identifier Register 2	Changed the PHY "Manufactures Module Number" in the "PHY Identifier Register 2 (MSB) - PHY ID 2 (03d; RO)" PHY register to 0x3B.
	EMIADD and EMIDATA	Added EMIADD (address 16d) and EMIDATA (address 17d) registers to enable access of registers in extended PHY memory address space.
	EEE PHY registers in extended PHY address space.	Added PHY related EEE (IEEE802.3az) in extended PHY memory address space. Can be accessed using EMIADD (address 16d) and EMIDATA (address 17d) registers.
LTR Support	LTRC	Added LTRC.EEEMS_EN field.
ECC and Parity checks	DTPARC, DTPARS, DPARS, DDPARC, DDPARS, DDECCC, DDECCS, RPBECCSTS, TPBECCSTS, PCIEECCSTS, PCIEECCCTL, PCIEERRSTS, PCIEERRCTL, LANPERRCTL and LANPERRSTS	• Modified parity/ECC functionality in: — DTPARC, DTPARS(R/W1C) and DPARS (R/W1C) registers - Register is per function, Status bits are clear by write 1. — DDPARC and DDPARS (R/W1C) registers - DHOST Rams have ECC protection. Renamed register to DDECCC and DDECCS. DDECCS is Clear by Write 1b. — RPBECCSTS register - Added Loopback Buffer support and "RPBECCSTS.Corr_err_cnt" field was removed. — TPBECCSTS register - Added Management TX buffer support and "TPBECCSTS.Corr_err_cnt" field was removed. — PCIEECCSTS (R/W1C)- Register logs correctable ECC errors, added Rams that changed protection to ECC. — PCIEECCCTL - Added memories that have ECC protection. — PCIEERRCTL - Removed memories that previously had parity error detection and now have ECC error detection. — PCIEERRCTL - Added Global Parity Enable bit (GPAR_EN). When bit is 0b parity error detection is disabled. — PCIEERRSTS (R/W1C)- Removed memories that previously had parity error detection and now have ECC error detection. Register is per function. • Updated behavior description on reception of parity error for the various memories. • Added XTX ram bit (enable and status). •

NOTE: ***This page intentionally left blank.***

§ §

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Intel:

NHI350AM4 S LJ2Z NHI350AM2 S LJ3R NHI350AM4 S LJ3Z NHI350CM2 S LJGL NHI350BT2 S LJ3Y NHI350BT2 S LJ3X NHI350AM2 S LJ3S

Общество с ограниченной ответственностью «МосЧип» ИНН 7719860671 / КПП 771901001
Адрес: 105318, г.Москва, ул.Щербаковская д.3, офис 1107

Данный компонент на территории Российской Федерации

Вы можете приобрести в компании MosChip.

Для оперативного оформления запроса Вам необходимо перейти по данной ссылке:

http://moschip.ru/get-element

Вы можете разместить у нас заказ для любого Вашего проекта, будь то серийное производство или разработка единичного прибора.

В нашем ассортименте представлены ведущие мировые производители активных и пассивных электронных компонентов.

Нашей специализацией является поставка электронной компонентной базы двойного назначения, продукции таких производителей как XILINX, Intel (ex.ALTERA), Vicor, Microchip, Texas Instruments, Analog Devices, Mini-Circuits, Amphenol, Glenair.

Сотрудничество с глобальными дистрибьюторами электронных компонентов, предоставляет возможность заказывать и получать с международных складов практически любой перечень компонентов в оптимальные для Вас сроки.

На всех этапах разработки и производства наши партнеры могут получить квалифицированную поддержку опытных инженеров.

Система менеджмента качества компании отвечает требованиям в соответствии с ГОСТ Р ИСО 9001, ГОСТ РВ 0015-002 и ЭС РД 009

Офис по работе с юридическими лицами:

105318, г.Москва, ул.Щербаковская д.3, офис 1107, 1118, ДЦ «Щербаковский»

Телефон: +7 495 668-12-70 (многоканальный)

Факс: +7 495 668-12-70 (доб.304)

E-mail: info@moschip.ru

Skype отдела продаж:
moschip.ru
moschip.ru_4
moschip.ru_6
moschip.ru_9