# Blackfin Embedded Processor

# ADSP-BF542/ADSP-BF544/ADSP-BF547/ADSP-BF548/ADSP-BF549

## FEATURES

- Up to 600 MHz high performance Blackfin processor
  - Two 16-bit MACs, two 40-bit ALUs, four 8-bit video ALUs
  - RISC-like register and instruction model
- Wide range of operating voltages and flexible booting options
- Programmable on-chip voltage regulator
- 400-ball CSP_BGA, RoHS compliant package

## MEMORY

- Up to 324K bytes of on-chip memory comprised of instruction SRAM/cache; dedicated instruction SRAM; data SRAM/cache; dedicated data SRAM; scratchpad SRAM
- External sync memory controller supporting either DDR SDRAM or mobile DDR SDRAM
- External async memory controller supporting 8-/16-bit async memories and burst flash devices
- NAND flash controller
- 4 memory-to-memory DMA pairs, 2 with ext. requests
- Memory management unit providing memory protection
- Code security with Lockbox secure technology and 128-bit AES/ARC4 data encryption
- One-time-programmable (OTP) memory

## PERIPHERALS

- High speed USB On-the-Go (OTG) with integrated PHY
- SD/SDIO controller
- ATA/ATAPI-6 controller
- Up to 4 synchronous serial ports (SPORTs)
- Up to 3 serial peripheral interfaces (SPI-compatible)
- Up to 4 UARTs, two with automatic H/W flow control
- Up to 2 CAN (controller area network) 2.0B interfaces
- Up to 2 TWI (2-wire interface) controllers
- 8- or 16-bit asynchronous host DMA interface
- Multiple enhanced parallel peripheral interfaces (EPPIs), supporting ITU-R BT.656 video formats and 18-/24-bit LCD connections
- Media transceiver (MXVR) for connection to a MOST network
- Pixel compositor for overlays, alpha blending, and color conversion
- Up to eleven 32-bit timers/counters with PWM support
- Real-time clock (RTC) and watchdog timer
- Up/down counter with support for rotary encoder
- Up to 152 general-purpose I/O (GPIOs)
- On-chip PLL capable of 0.5× to 64× frequency multiplication
- Debug/JTAG interface

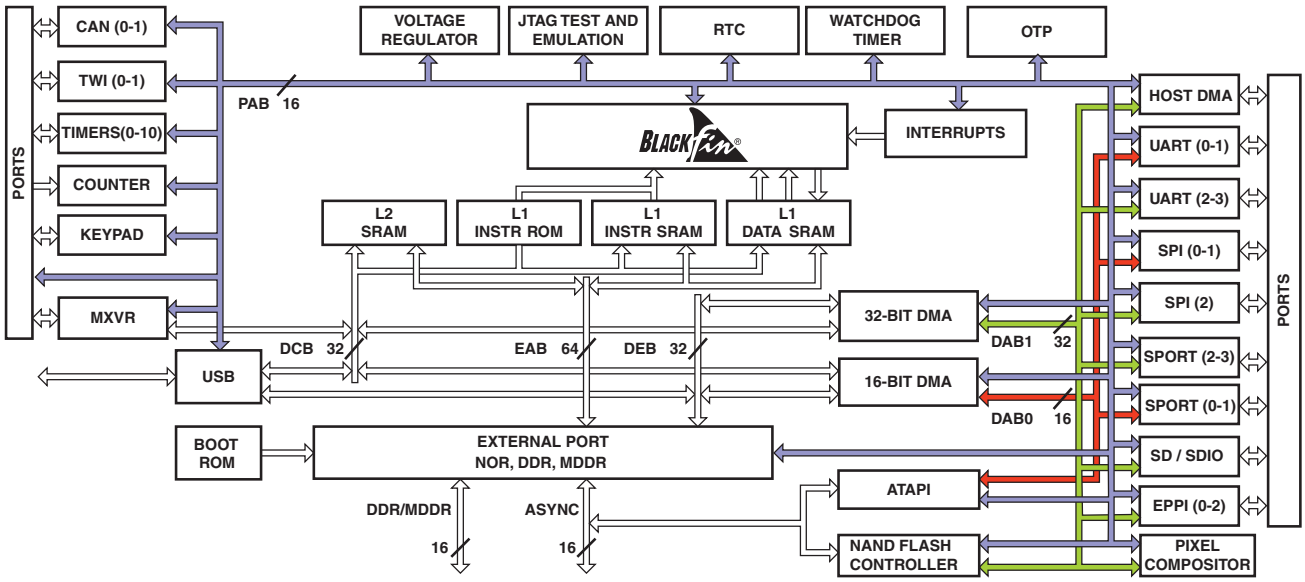



*Figure 1. ADSP-BF549 Functional Block Diagram*

Blackfin and the Blackfin logo are registered trademarks of Analog Devices, Inc.

**Rev. C**

**Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.**

**One Technology Way, P.O.Box 9106, Norwood, MA 02062-9106 U.S.A.**
**Tel: 781.329.4700** **www.analog.com**
**Fax: 781.461.3113** **© 2010 Analog Devices, Inc. All rights reserved.**

# TABLE OF CONTENTS


General Description ..... 3
Low Power Architecture ..... 4
System Integration ..... 4
Blackfin Processor Peripherals ..... 4
Blackfin Processor Core ..... 4
Memory Architecture ..... 6
DMA Controllers ..... 10
Real-Time Clock ..... 11
Watchdog Timer ..... 12
Timers ..... 12
Up/Down Counter and Thumbwheel Interface ..... 12
Serial Ports (SPORTs) ..... 12
Serial Peripheral Interface (SPI) Ports ..... 13
UART Ports (UARTs) ..... 13
Controller Area Network (CAN) ..... 13
TWI Controller Interface ..... 14
Ports ..... 14
Pixel Compositor (PIXC) ..... 14
Enhanced Parallel Peripheral Interface (EPPI) ..... 14
USB On-the-Go Dual-Role Device Controller ..... 15
ATA/ATAPI-6 Interface ..... 15
Keypad Interface ..... 15
Secure Digital (SD)/SDIO Controller ..... 16
Code Security ..... 16
Media Transceiver MAC Layer (MXVR) ..... 16
Dynamic Power Management ..... 16
Voltage Regulation ..... 18
Clock Signals ..... 18
Booting Modes ..... 19
Instruction Set Description ..... 22
Development Tools ..... 23
EZ-KIT Lite Evaluation Board ..... 23
Designing an Emulator-Compatible Processor Board ..... 23
MXVR Board Layout Guidelines ..... 23
Related Documents ..... 24
Lockbox Secure Technology Disclaimer ..... 24
Pin Descriptions ..... 25
Specifications ..... 34
Operating Conditions ..... 34
Electrical Characteristics ..... 36
Absolute Maximum Ratings ..... 40
ESD Sensitivity ..... 41
Package Information ..... 41
Timing Specifications ..... 42
Output Drive Currents ..... 86
Test Conditions ..... 88
Capacitive Loading ..... 88
Typical Rise and Fall Times ..... 89
Thermal Characteristics ..... 91
400-Ball CSP_BGA Package ..... 92
Outline Dimensions ..... 98
Surface-Mount Design ..... 98
Automotive Products ..... 99
Ordering Guide ..... 100


## REVISION HISTORY

**2/10—Rev. B to Rev. C**

Added $V_{IHTWI}$ and $V_{ILTWI}$ data to Operating Conditions ..... 34

Added $I_{OH}/I_{OL}$ per pin group data to Absolute Maximum Ratings ..... 40

Added Table 23 (Total Current Pin Groups) ..... 40

Revised all timing diagrams for clarity/consistency in Timing Specifications ..... 42

Updated specifications (reference PCN 09_0173) in the Clock and Reset Timing section to accurately describe processor cold-startup/reset timing..... 42

Added $t_{SUDTE}$ and $t_{SUDRE}$ data to Table 41 (Serial Ports—External Clock) ..... 61

Added $t_{SCLKIW}$ and $t_{SCLK}$ data to Table 42 (Serial Ports—Internal Clock) .....61

Added Figure 34 (Serial Port Start-Up with External Clock and Frame Sync) and Figure 36 (Serial Ports—Enable and Three-State) .....62

To view product/process change notifications (PCNs) related to this data sheet revision, please visit the processor's product page on the www.analog.com website and use the **View PCN** link.

# GENERAL DESCRIPTION

The ADSP-BF54x Blackfin® processors are members of the Blackfin family of products, incorporating the Analog Devices/ Intel Micro Signal Architecture (MSA). Blackfin processors combine a dual-MAC state-of-the-art signal processing engine, the advantages of a clean, orthogonal RISC-like microprocessor instruction set, and single-instruction, multiple-data (SIMD) multimedia capabilities into a single instruction-set architecture.

Specific performance, memory configurations, and features of ADSP-BF54x Blackfin processors are shown in Table 1.

**Table 1. ADSP-BF54x Processor Features**

| Processor Features | | ADSP-BF549 | ADSP-BF548 | ADSP-BF547 | ADSP-BF544 | ADSP-BF542 |
|---|---|---|---|---|---|---|
| Lockbox® [1] code security | | 1 | 1 | 1 | 1 | 1 |
| 128-bit AES/ ARC4 data encryption | | 1 | 1 | 1 | 1 | 1 |
| SD/SDIO controller | | 1 | 1 | 1 | – | 1 |
| Pixel compositor | | 1 | 1 | 1 | 1 | 1 |
| 18- or 24-bit EPPI0 with LCD | | 1 | 1 | 1 | 1 | – |
| 16-bit EPPI1, 8-bit EPPI2 | | 1 | 1 | 1 | 1 | 1 |
| Host DMA port | | 1 | 1 | 1 | 1 | – |
| NAND flash controller | | 1 | 1 | 1 | 1 | 1 |
| ATAPI | | 1 | 1 | 1 | – | 1 |
| High Speed USB OTG | | 1 | 1 | 1 | – | 1 |
| Keypad interface | | 1 | 1 | 1 | – | 1 |
| MXVR | | 1 | – | – | – | – |
| CAN ports | | 2 | 2 | – | 2 | 1 |
| TWI ports | | 2 | 2 | 2 | 2 | 1 |
| SPI ports | | 3 | 3 | 3 | 2 | 2 |
| UART ports | | 4 | 4 | 4 | 3 | 3 |
| SPORTs | | 4 | 4 | 4 | 3 | 3 |
| Up/Down counter | | 1 | 1 | 1 | 1 | 1 |
| Timers | | 11 | 11 | 11 | 11 | 8 |
| General-Purpose I/O pins | | 152 | 152 | 152 | 152 | 152 |
| Memory Configurations (K Bytes) | L1 Instruction SRAM/Cache | 16 | 16 | 16 | 16 | 16 |
| | L1 Instruction SRAM | 48 | 48 | 48 | 48 | 48 |
| | L1 Data SRAM/Cache | 32 | 32 | 32 | 32 | 32 |
| | L1 Data SRAM | 32 | 32 | 32 | 32 | 32 |
| | L1 Scratchpad SRAM | 4 | 4 | 4 | 4 | 4 |
| | L1 ROM[2] | 64 | 64 | 64 | 64 | 64 |
| | L2 | 128 | 128 | 128 | 64 | – |
| | L3 Boot ROM[2] | 4 | 4 | 4 | 4 | 4 |
| Maximum Core Instruction Rate (MHz) | | 533 | 533 | 600 | 533 | 600 |

[1] Lockbox is a registered trademark of Analog Devices, Inc.
[2] This ROM is not customer-configurable.

Specific peripherals for ADSP-BF54x Blackfin processors are shown in Table 2.

**Table 2. Specific Peripherals for ADSP-BF54x Processors**

| Module | ADSP-BF549 | ADSP-BF548 | ADSP-BF547 | ADSP-BF544 | ADSP-BF542 |
|---|---|---|---|---|---|
| EBIU (async) | P | P | P | P | P |
| NAND flash controller | P | P | P | P | P |
| ATAPI | P | P | P | – | P |
| Host DMA port (HOSTDP) | P | P | P | P | – |
| SD/SDIO controller | P | P | P | – | P |
| EPPI0 | P | P | P | P | – |
| EPPI1 | P | P | P | P | P |
| EPPI2 | P | P | P | P | P |
| SPORT0 | P | P | P | – | – |
| SPORT1 | P | P | P | P | P |
| SPORT2 | P | P | P | P | P |
| SPORT3 | P | P | P | P | P |
| SPI0 | P | P | P | P | P |
| SPI1 | P | P | P | P | P |
| SPI2 | P | P | P | – | – |
| UART0 | P | P | P | P | P |
| UART1 | P | P | P | P | P |
| UART2 | P | P | P | – | – |
| UART3 | P | P | P | P | P |
| High Speed USB OTG | P | P | P | – | P |
| CAN0 | P | P | – | P | P |
| CAN1 | P | P | – | P | – |
| TWI0 | P | P | P | P | P |
| TWI1 | P | P | P | P | – |
| Timer 0–7 | P | P | P | P | P |
| Timer 8–10 | P | P | P | P | – |
| Up/Down counter | P | P | P | P | P |
| Keypad interface | P | P | P | – | P |
| MXVR | P | – | – | – | – |
| GPIOs | P | P | P | P | P |

The ADSP-BF54x Blackfin processors are completely code- and pin-compatible. They differ only with respect to their performance, on-chip memory, and selection of I/O peripherals. Specific performance, memory, and feature configurations are shown in Table 1.

By integrating a rich set of industry-leading system peripherals and memory, Blackfin processors are the platform of choice for next-generation applications that require RISC-like programmability, multimedia support, and leading-edge signal processing in one integrated package.

## LOW POWER ARCHITECTURE

Blackfin processors provide world-class power management and performance. Blackfin processors are designed in a low power and low voltage design methodology and feature on-chip dynamic power management, the ability to vary both the voltage and frequency of operation to significantly lower overall power consumption. Reducing both voltage and frequency can result in a substantial reduction in power consumption as compared to reducing only the frequency of operation. This translates into longer battery life for portable appliances.

## SYSTEM INTEGRATION

The ADSP-BF54x Blackfin processors are highly integrated system-on-a-chip solutions for the next generation of embedded network connected applications. By combining industry-standard interfaces with a high performance signal processing core, users can develop cost-effective solutions quickly without the need for costly external components. The system peripherals include a high speed USB OTG (On-the-Go) controller with integrated PHY, CAN 2.0B controllers, TWI controllers, UART ports, SPI ports, serial ports (SPORTs), ATAPI controller, SD/SDIO controller, a real-time clock, a watchdog timer, LCD controller, and multiple enhanced parallel peripheral interfaces.

## BLACKFIN PROCESSOR PERIPHERALS

The ADSP-BF54x processors contain a rich set of peripherals connected to the core via several high bandwidth buses, providing flexibility in system configuration as well as excellent overall system performance (see Figure 1 on Page 1). The general-purpose peripherals include functions such as UARTs, SPI, TWI, timers with pulse width modulation (PWM) and pulse measurement capability, general-purpose I/O pins, a real-time clock, and a watchdog timer. This set of functions satisfies a wide variety of typical system support needs and is augmented by the system expansion capabilities of the part. The ADSP-BF54x processors contain dedicated network communication modules and high speed serial and parallel ports, an interrupt controller for flexible management of interrupts from the on-chip peripherals or external sources, and power management control functions to tailor the performance and power characteristics of the processor and system to many application scenarios.

All of the peripherals, except for general-purpose I/O, CAN, TWI, real-time clock, and timers, are supported by a flexible DMA structure. There are also separate memory DMA channels dedicated to data transfers between the processor's various memory spaces, including external DDR (either standard or mobile, depending on the device) and asynchronous memory. Multiple on-chip buses running at up to 133 MHz provide enough bandwidth to keep the processor core running along with activity on all of the on-chip and external peripherals.

The ADSP-BF54x Blackfin processors include an on-chip voltage regulator in support of the dynamic power management capability. The voltage regulator provides a range of core voltage levels when supplied from $V_{DDEXT}$. The voltage regulator can be bypassed at the user's discretion.

## BLACKFIN PROCESSOR CORE

As shown in Figure 2 on Page 5, the Blackfin processor core contains two 16-bit multipliers, two 40-bit accumulators, two 40-bit ALUs, four video ALUs, and a 40-bit shifter. The computation units process 8-, 16-, or 32-bit data from the register file.

The compute register file contains eight 32-bit registers. When performing compute operations on 16-bit operand data, the register file operates as 16 independent 16-bit registers. All operands for compute operations come from the multiported register file and instruction constant fields.

Each MAC can perform a 16-bit by 16-bit multiply in each cycle, accumulating the results into the 40-bit accumulators. Signed and unsigned formats, rounding, and saturation are supported.

The ALUs perform a traditional set of arithmetic and logical operations on 16- or 32-bit data. In addition, many special instructions are included to accelerate various signal processing tasks. These include bit operations such as field extract and population count, modulo $2^{32}$ multiply, divide primitives, saturation and rounding, and sign/exponent detection. The set of video instructions include byte alignment and packing operations, 16-bit and 8-bit adds with clipping, 8-bit average operations, and 8-bit subtract/absolute value/accumulate (SAA) operations. Also provided are the compare/select and vector search instructions.

For certain instructions, two 16-bit ALU operations can be performed simultaneously on register pairs (a 16-bit high half and 16-bit low half of a compute register). By also using the second ALU, quad 16-bit operations are possible.

The 40-bit shifter can perform shifts and rotates and is used to support normalization, field extract, and field deposit instructions.

The program sequencer controls the flow of instruction execution, including instruction alignment and decoding. For program flow control, the sequencer supports PC relative and indirect conditional jumps (with static branch prediction), and subroutine calls. Hardware is provided to support zero-overhead looping. The architecture is fully interlocked, meaning that the programmer need not manage the pipeline when executing instructions with data dependencies.

The address arithmetic unit provides two addresses for simultaneous dual fetches from memory. It contains a multiported register file consisting of four sets of 32-bit index, modify,

length, and base registers (for circular buffering), and eight additional 32-bit pointer registers (for C-style indexed stack manipulation).

Blackfin processors support a modified Harvard architecture in combination with a hierarchical memory structure. Level 1 (L1) memories are those that typically operate at the full processor speed with little or no latency. At the L1 level, the instruction memory holds instructions only. The two data memories hold data, and a dedicated scratchpad data memory stores stack and local variable information.

In addition, multiple L1 memory blocks are provided, offering a configurable mix of SRAM and cache. The memory management unit (MMU) provides memory protection for individual tasks that may be operating on the core and can protect system registers from unintended access.

The architecture provides three modes of operation: user mode, supervisor mode, and emulation mode. User mode has restricted access to certain system resources, thus providing a protected software environment, while supervisor mode has unrestricted access to the system and core resources.

The Blackfin processor instruction set has been optimized so that 16-bit opcodes represent the most frequently used instructions, resulting in excellent compiled code density. Complex DSP instructions are encoded into 32-bit opcodes, representing fully featured multifunction instructions. Blackfin processors support a limited multi-issue capability, where a 32-bit instruction can be issued in parallel with two 16-bit instructions, allowing the programmer to use many of the core resources in a single instruction cycle.

The Blackfin processor assembly language uses an algebraic syntax for ease of coding and readability. The architecture has been optimized for use in conjunction with the C/C++ compiler, resulting in fast and efficient software implementations.

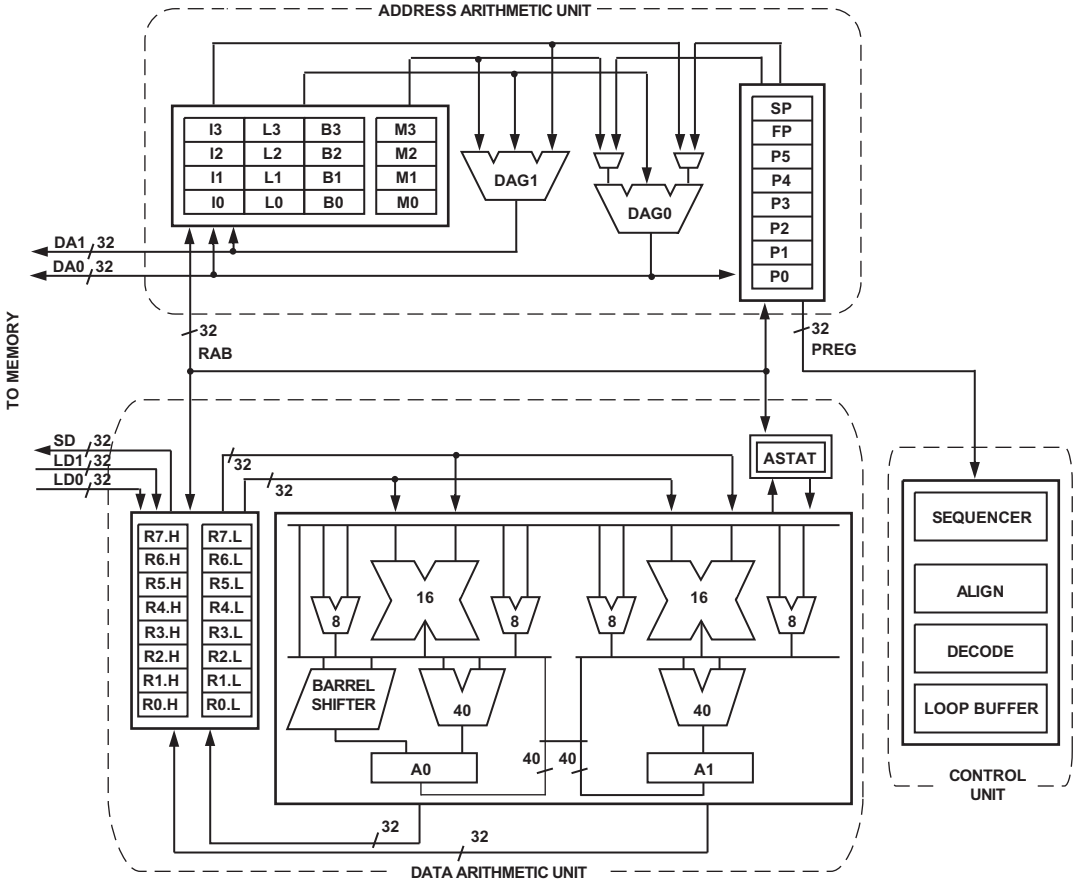



*Figure 2. Blackfin Processor Core*

## MEMORY ARCHITECTURE

The ADSP-BF54x processors view memory as a single unified 4G byte address space, using 32-bit addresses. All resources, including internal memory, external memory, and I/O control registers, occupy separate sections of this common address space. The memory portions of this address space are arranged in a hierarchical structure to provide a good cost/performance balance of some very fast, low-latency on-chip memory as cache or SRAM, and larger, lower-cost and performance off-chip memory systems. See Figure 3 on Page 6.

The on-chip L1 memory system is the highest-performance memory available to the Blackfin processor. The off-chip memory system, accessed through the external bus interface unit (EBIU), provides expansion with flash memory, SRAM, and double-rate SDRAM (standard or mobile DDR), optionally accessing up to 768M bytes of physical memory.

Most of the ADSP-BF54x Blackfin processors also include an L2 SRAM memory array which provides up to 128K bytes of high speed SRAM, operating at one half the frequency of the core and with slightly longer latency than the L1 memory banks (for information on L2 memory in each processor, see Table 1). The L2 memory is a unified instruction and data memory and can hold any mixture of code and data required by the system design. The Blackfin cores share a dedicated low latency 64-bit data path port into the L2 SRAM memory.

The memory DMA controllers (DMAC1 and DMAC0) provide high-bandwidth data-movement capability. They can perform block transfers of code or data between the internal memory and the external memory spaces.

### *Internal (On-Chip) Memory*

The ADSP-BF54x processors have several blocks of on-chip memory providing high bandwidth access to the core.

The first block is the L1 instruction memory, consisting of 64K bytes of SRAM, of which 16K bytes can be configured as a four-way set-associative cache or as SRAM. This memory is accessed at full processor speed.

The second on-chip memory block is the L1 data memory, consisting of 64K bytes of SRAM, of which 32K bytes can be configured as a two-way set-associative cache or as SRAM. This memory block is accessed at full processor speed.

The third memory block is a 4K byte scratchpad SRAM, which runs at the same speed as the L1 memories. It is only accessible as data SRAM and cannot be configured as cache memory.

The fourth memory block is the factory programmed L1 instruction ROM, operating at full processor speed. This ROM is not customer-configurable.

The fifth memory block is the L2 SRAM, providing up to 128K bytes of unified instruction and data memory, operating at one half the frequency of the core.

Finally, there is a 4K byte boot ROM connected as L3 memory. It operates at full SCLK rate.

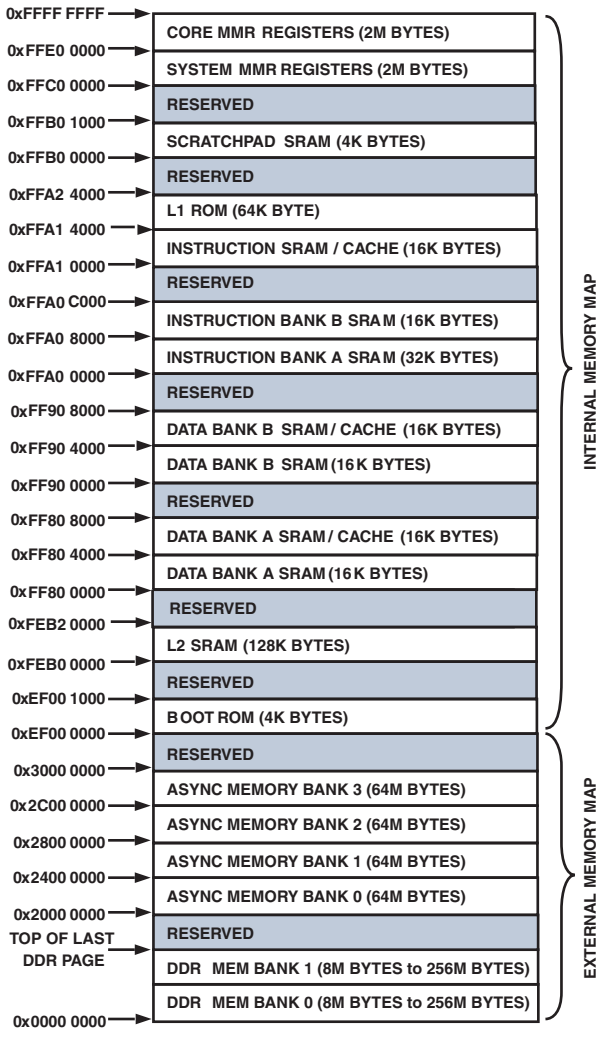



*Figure 3. ADSP-BF547/ADSP-BF548/ADSP-BF549 Internal/External Memory Map*[1]

[1] For ADSP-BF544 processors, L2 SRAM is 64K Bytes (0xFEB0000–0xFEB0FFFF). For ADSP-BF542 processors, there is no L2 SRAM.

### *External (Off-Chip) Memory*

Through the external bus interface unit (EBIU), the ADSP-BF54x Blackfin processors provide glueless connectivity to external 16-bit wide memories, such as DDR and mobile DDR SDRAM, SRAM, NOR flash, NAND flash, and FIFO devices. To provide the best performance, the bus system of the DDR and mobile DDR interface is completely separate from the other parallel interfaces. Furthermore, the DDR controller supports either standard DDR memory or mobile DDR memory. See the Ordering Guide on Page 100 for details. Throughout this document, references to "DDR" are intended to cover both the standard and mobile DDR standards.

The DDR memory controller can gluelessly manage up to two banks of double-rate synchronous dynamic memory (DDR and mobile DDR SDRAM). The 16-bit interface operates at the SCLK frequency, enabling a maximum throughput of 532M bytes/s. The DDR and mobile DDR controller is augmented with a queuing mechanism that performs efficient bursts into the DDR and mobile DDR. The controller is an industry standard DDR and mobile DDR SDRAM controller with each bank supporting from 64M bit to 512M bit device sizes and 4-, 8-, or 16-bit widths. The controller supports up to 256M bytes per external bank. With 2 external banks, the controller supports up to 512M bytes total. Each bank is independently programmable and is contiguous with adjacent banks regardless of the sizes of the different banks or their placement.

Traditional 16-bit asynchronous memories, such as SRAM, EPROM, and flash devices, can be connected to one of the four 64M byte asynchronous memory banks, represented by four memory select strobes. Alternatively, these strobes can function as bank-specific read or write strobes preventing further glue logic when connecting to asynchronous FIFO devices. See the Ordering Guide on Page 100 for a list of specific products that provide support for DDR memory.

In addition, the external bus can connect to advanced flash device technologies, such as:

- Page-mode NOR flash devices
- Synchronous burst-mode NOR flash devices
- NAND flash devices

Customers should consult the Ordering Guide when selecting a specific ADSP-BF54x component for the intended application. Products that provide support for mobile DDR memory are noted in the ordering guide footnotes.

**NAND Flash Controller (NFC)**

The ADSP-BF54x Blackfin processors provide a NAND Flash Controller (NFC) as part of the external bus interface. NAND flash devices provide high-density, low-cost memory. However, NAND flash devices also have long random access times, invalid blocks, and lower reliability over device lifetimes. Because of this, NAND flash is often used for read-only code storage. In this case, all DSP code can be stored in NAND flash and then transferred to a faster memory (such as DDR or SRAM) before execution. Another common use of NAND flash is for storage of multimedia files or other large data segments. In this case, a software file system may be used to manage reading and writing of the NAND flash device. The file system selects memory segments for storage with the goal of avoiding bad blocks and equally distributing memory accesses across all address locations. Hardware features of the NFC include:

- Support for page program, page read, and block erase of NAND flash devices, with accesses aligned to page boundaries.
- Error checking and correction (ECC) hardware that facilitates error detection and correction.
- A single 8-bit or 16-bit external bus interface for commands, addresses, and data.
- Support for SLC (single level cell) NAND flash devices unlimited in size, with page sizes of 256 bytes and 512 bytes. Larger page sizes can be supported in software.
- The ability to release external bus interface pins during long accesses.
- Support for internal bus requests of 16 bits or 32 bits.
- A DMA engine to transfer data between internal memory and a NAND flash device.

### *One-Time-Programmable Memory*

The ADSP-BF54x Blackfin processors have 64K bits of one-time-programmable (OTP) non-volatile memory that can be programmed by the developer only one time. It includes the array and logic to support read access and programming. Additionally, its pages can be write protected.

OTP enables developers to store both public and private data on-chip. In addition to storing public and private key data for applications requiring security, it also allows developers to store completely user-definable data such as a customer ID, product ID, or a MAC address. By using this feature, generic parts can be shipped, which are then programmed and protected by the developer within this non-volatile memory. The OTP memory can be accessed through an API provided by the on-chip ROM.

### *I/O Memory Space*

The ADSP-BF54x Blackfin processors do not define a separate I/O space. All resources are mapped through the flat 32-bit address space. On-chip I/O devices have their control registers mapped into memory-mapped registers (MMRs) at addresses near the top of the 4G byte address space. These are separated into two smaller blocks, one containing the control MMRs for all core functions and the other containing the registers needed for setup and control of the on-chip peripherals outside of the core. The MMRs are accessible only in supervisor mode and appear as reserved space to on-chip peripherals.

### *Booting*

The ADSP-BF54x Blackfin processors contain a small on-chip boot kernel, which configures the appropriate peripheral for booting. If the ADSP-BF54x Blackfin processors are configured to boot from boot ROM memory space, the processor starts executing from the on-chip boot ROM. For more information, see Booting Modes on Page 19.

### *Event Handling*

The event controller on the ADSP-BF54x Blackfin processors handles all asynchronous and synchronous events to the processors. The ADSP-BF54x Blackfin processors provide event handling that supports both nesting and prioritization. Nesting allows multiple event service routines to be active simultaneously. Prioritization ensures that servicing of a

higher-priority event takes precedence over servicing of a lower-priority event. The controller provides support for five different types of events:

- Emulation. An emulation event causes the processor to enter emulation mode, allowing command and control of the processor via the JTAG interface.
- Reset. This event resets the processor.
- Non-maskable interrupt (NMI). The NMI event can be generated by the software watchdog timer or by the $\overline{\text{NMI}}$ input signal to the processor. The NMI event is frequently used as a power-down indicator to initiate an orderly shutdown of the system.
- Exceptions. Events that occur synchronously to program flow (that is, the exception is taken before the instruction is allowed to complete). Conditions such as data alignment violations and undefined instructions cause exceptions.
- Interrupts. Events that occur asynchronously to program flow. They are caused by input pins, timers, and other peripherals, as well as by an explicit software instruction.

Each event type has an associated register to hold the return address and an associated return-from-event instruction. When an event is triggered, the state of the processor is saved on the supervisor stack.

The ADSP-BF54x Blackfin processor event controller consists of two stages, the core event controller (CEC) and the system interrupt controller (SIC). The core event controller works with the system interrupt controller to prioritize and control all system events. Conceptually, interrupts from the peripherals enter into the SIC and are then routed directly into the general-purpose interrupts of the CEC.

### *Core Event Controller (CEC)*

The CEC supports nine general-purpose interrupts (IVG15–7), in addition to the dedicated interrupt and exception events. Of these general-purpose interrupts, the two lowest-priority interrupts (IVG15–14) are recommended to be reserved for software interrupt handlers, leaving seven prioritized interrupt inputs to support the peripherals of the ADSP-BF54x Blackfin processors. Table 3 describes the inputs to the CEC, identifies their names in the event vector table (EVT), and lists their priorities.

**Table 3. Core Event Controller (CEC)**

| Priority (0 is Highest) | Event Class | EVT Entry |
|---|---|---|
| 0 | Emulation/Test Control | EMU |
| 1 | Reset | RST |
| 2 | Nonmaskable Interrupt | NMI |
| 3 | Exception | EVX |
| 4 | Reserved | — |
| 5 | Hardware Error | IVHW |
| 6 | Core Timer | IVTMR |
| 7 | General Interrupt 7 | IVG7 |
| 8 | General Interrupt 8 | IVG8 |
| 9 | General Interrupt 9 | IVG9 |
| 10 | General Interrupt 10 | IVG10 |
| 11 | General Interrupt 11 | IVG11 |
| 12 | General Interrupt 12 | IVG12 |
| 13 | General Interrupt 13 | IVG13 |
| 14 | General Interrupt 14 | IVG14 |
| 15 | General Interrupt 15 | IVG15 |

### *System Interrupt Controller (SIC)*

The system interrupt controller provides the mapping and routing of events from the many peripheral interrupt sources to the prioritized general-purpose interrupt inputs of the CEC. Although the ADSP-BF54x Blackfin processors provide a default mapping, the user can alter the mappings and priorities of interrupt events by writing the appropriate values into the interrupt assignment registers (SIC_IARx). Table 4 describes the inputs into the SIC and the default mappings into the CEC.

**Table 4. System Interrupt Controller (SIC)**

| Peripheral IRQ Source | IRQ ID | GP IRQ (at Reset) | Core IRQ ID |
|---|---|---|---|
| PLL Wakeup IRQ | 0 | IVG7 | 0 |
| DMAC0 Status (Generic) | 1 | IVG7 | 0 |
| EPPI0 Error IRQ | 2 | IVG7 | 0 |
| SPORT0 Error IRQ | 3 | IVG7 | 0 |
| SPORT1 Error IRQ | 4 | IVG7 | 0 |
| SPI0 Status IRQ | 5 | IVG7 | 0 |
| UART0 Status IRQ | 6 | IVG7 | 0 |
| Real-Time Clock IRQ | 7 | IVG8 | 1 |
| DMA12 IRQ (EPPI0) | 8 | IVG8 | 1 |
| DMA0 IRQ (SPORT0 RX) | 9 | IVG9 | 2 |
| DMA1 IRQ (SPORT0 TX) | 10 | IVG9 | 2 |
| DMA2 IRQ (SPORT1 RX) | 11 | IVG9 | 2 |
| DMA3 IRQ (SPORT1 TX) | 12 | IVG9 | 2 |
| DMA4 IRQ (SPI0) | 13 | IVG10 | 3 |

**Table 4. System Interrupt Controller (SIC) (Continued)**

| Peripheral IRQ Source | IRQ ID | GP IRQ (at Reset) | Core IRQ ID |
|---|---|---|---|
| DMA6 IRQ (UART0 RX) | 14 | IVG10 | 3 |
| DMA7 IRQ (UART0 TX) | 15 | IVG10 | 3 |
| Timer 8 IRQ | 16 | IVG11 | 4 |
| Timer 9 IRQ | 17 | IVG11 | 4 |
| Timer 10 IRQ | 18 | IVG11 | 4 |
| Pin IRQ 0 (PINT0) | 19 | IVG12 | 5 |
| Pin IRQ 1 (PINT1) | 20 | IVG12 | 5 |
| MDMA Stream 0 IRQ | 21 | IVG13 | 6 |
| MDMA Stream 1 IRQ | 22 | IVG13 | 6 |
| Software Watchdog Timer IRQ | 23 | IVG13 | 6 |
| DMAC1 Status (Generic) | 24 | IVG7 | 0 |
| SPORT2 Error IRQ | 25 | IVG7 | 0 |
| SPORT3 Error IRQ | 26 | IVG7 | 0 |
| MXVR Synchronous Data IRQ | 27 | IVG7 | 0 |
| SPI1 Status IRQ | 28 | IVG7 | 0 |
| SPI2 Status IRQ | 29 | IVG7 | 0 |
| UART1 Status IRQ | 30 | IVG7 | 0 |
| UART2 Status IRQ | 31 | IVG7 | 0 |
| CAN0 Status IRQ | 32 | IVG7 | 0 |
| DMA18 IRQ (SPORT2 RX) | 33 | IVG9 | 2 |
| DMA19 IRQ (SPORT2 TX) | 34 | IVG9 | 2 |
| DMA20 IRQ (SPORT3 RX) | 35 | IVG9 | 2 |
| DMA21 IRQ (SPORT3 TX) | 36 | IVG9 | 2 |
| DMA13 IRQ (EPPI1) | 37 | IVG9 | 2 |
| DMA14 IRQ (EPPI2, Host DMA) | 38 | IVG9 | 2 |
| DMA5 IRQ (SPI1) | 39 | IVG10 | 3 |
| DMA23 IRQ (SPI2) | 40 | IVG10 | 3 |
| DMA8 IRQ (UART1 RX) | 41 | IVG10 | 3 |
| DMA9 IRQ (UART1 TX) | 42 | IVG10 | 3 |
| DMA10 IRQ (ATAPI RX) | 43 | IVG10 | 3 |
| DMA11 IRQ (ATAPI TX) | 44 | IVG10 | 3 |
| TWI0 IRQ | 45 | IVG11 | 4 |
| TWI1 IRQ | 46 | IVG11 | 4 |
| CAN0 Receive IRQ | 47 | IVG11 | 4 |
| CAN0 Transmit IRQ | 48 | IVG11 | 4 |
| MDMA Stream 2 IRQ | 49 | IVG13 | 6 |
| MDMA Stream 3 IRQ | 50 | IVG13 | 6 |
| MXVR Status IRQ | 51 | IVG11 | 4 |
| MXVR Control Message IRQ | 52 | IVG11 | 4 |

**Table 4. System Interrupt Controller (SIC) (Continued)**

| Peripheral IRQ Source | IRQ ID | GP IRQ (at Reset) | Core IRQ ID |
|---|---|---|---|
| MXVR Asynchronous Packet IRQ | 53 | IVG11 | 4 |
| EPPI1 Error IRQ | 54 | IVG7 | 0 |
| EPPI2 Error IRQ | 55 | IVG7 | 0 |
| UART3 Status IRQ | 56 | IVG7 | 0 |
| Host DMA Status | 57 | IVG7 | 0 |
| Reserved | 58 | IVG7 | 0 |
| Pixel Compositor (PIXC) Status IRQ | 59 | IVG7 | 0 |
| NFC Status IRQ | 60 | IVG7 | 0 |
| ATAPI Status IRQ | 61 | IVG7 | 0 |
| CAN1 Status IRQ | 62 | IVG7 | 0 |
| DMAR0 Block IRQ | 63 | IVG7 | 0 |
| DMAR1 Block IRQ | 63 | IVG7 | 0 |
| DMAR0 Overflow Error IRQ | 63 | IVG7 | 0 |
| DMAR1 Overflow Error IRQ | 63 | IVG7 | 0 |
| DMA15 IRQ (PIXC IN0) | 64 | IVG8 | 1 |
| DMA16 IRQ (PIXC IN1) | 65 | IVG8 | 1 |
| DMA17 IRQ (PIXC OUT) | 66 | IVG8 | 1 |
| DMA22 IRQ (SDH/NFC) | 67 | IVG8 | 1 |
| Counter (CNT) IRQ | 68 | IVG8 | 1 |
| Keypad (KEY) IRQ | 69 | IVG8 | 1 |
| CAN1 RX IRQ | 70 | IVG11 | 4 |
| CAN1 TX IRQ | 71 | IVG11 | 4 |
| SDH Mask 0 IRQ | 72 | IVG11 | 4 |
| SDH Mask 1 IRQ | 73 | IVG11 | 4 |
| Reserved | 74 | IVG11 | 4 |
| USB_INT0 IRQ | 75 | IVG11 | 4 |
| USB_INT1 IRQ | 76 | IVG11 | 4 |
| USB_INT2 IRQ | 77 | IVG11 | 4 |
| USB_DMAINT IRQ | 78 | IVG11 | 4 |
| OTPSEC IRQ | 79 | IVG11 | 4 |
| Reserved | 80 | IVG11 | 4 |
| Reserved | 81 | IVG11 | 4 |
| Reserved | 82 | IVG11 | 4 |
| Reserved | 83 | IVG11 | 4 |
| Reserved | 84 | IVG11 | 4 |
| Reserved | 85 | IVG11 | 4 |
| Timer 0 IRQ | 86 | IVG11 | 4 |
| Timer 1 IRQ | 87 | IVG11 | 4 |
| Timer 2 IRQ | 88 | IVG11 | 4 |

**Table 4. System Interrupt Controller (SIC) (Continued)**

| Peripheral IRQ Source | IRQ ID | GP IRQ (at Reset) | Core IRQ ID |
|---|---|---|---|
| Timer 3 IRQ | 89 | IVG11 | 4 |
| Timer 4 IRQ | 90 | IVG11 | 4 |
| Timer 5 IRQ | 91 | IVG11 | 4 |
| Timer 6 IRQ | 92 | IVG11 | 4 |
| Timer 7 IRQ | 93 | IVG11 | 4 |
| Pin IRQ 2 (PINT2) | 94 | IVG12 | 5 |
| Pin IRQ 3 (PINT3) | 95 | IVG12 | 5 |

### *Event Control*

The ADSP-BF54x Blackfin processors provide the user with a very flexible mechanism to control the processing of events. In the CEC, three registers are used to coordinate and control events. Each register is 16 bits wide:

- CEC interrupt latch register (ILAT). The ILAT register indicates when events have been latched. The appropriate bit is set when the processor has latched the event and cleared when the event has been accepted into the system. This register is updated automatically by the controller, but it may be written only when its corresponding IMASK bit is cleared.
- CEC interrupt mask register (IMASK). The IMASK register controls the masking and unmasking of individual events. When a bit is set in the IMASK register, that event is unmasked and is processed by the CEC when asserted. A cleared bit in the IMASK register masks the event, preventing the processor from servicing the event even though the event may be latched in the ILAT register. This register may be read or written while in supervisor mode. Note that general-purpose interrupts can be globally enabled and disabled with the STI and CLI instructions, respectively.
- CEC interrupt pending register (IPEND). The IPEND register keeps track of all nested events. A set bit in the IPEND register indicates that the event is currently active or nested at some level. This register is updated automatically by the controller but may be read while in supervisor mode.

The SIC allows further control of event processing by providing three 32-bit interrupt control and status registers. Each register contains a bit corresponding to each of the peripheral interrupt events shown in Table 4 on Page 8.

- SIC interrupt mask registers (SIC_IMASKx). These registers control the masking and unmasking of each peripheral interrupt event. When a bit is set in a register, that peripheral event is unmasked and is processed by the system when asserted. A cleared bit in the register masks the peripheral event, preventing the processor from servicing the event.
- SIC interrupt status registers (SIC_ISRx). As multiple peripherals can be mapped to a single event, these registers allow the software to determine which peripheral event source triggered the interrupt. A set bit indicates the peripheral is asserting the interrupt, and a cleared bit indicates the peripheral is not asserting the event.
- SIC interrupt wakeup enable registers (SIC_IWRx). By enabling the corresponding bit in this register, a peripheral can be configured to wake up the processor, should the core be idled or in Sleep mode when the event is generated. (For more information, see Dynamic Power Management on Page 16.)

Because multiple interrupt sources can map to a single general-purpose interrupt, multiple pulse assertions can occur simultaneously, before or during interrupt processing for an interrupt event already detected on this interrupt input. The IPEND register contents are monitored by the SIC as the interrupt acknowledgement.

The appropriate ILAT register bit is set when an interrupt rising edge is detected. (Detection requires two core clock cycles.) The bit is cleared when the respective IPEND register bit is set. The IPEND bit indicates that the event has entered into the processor pipeline. At this point the CEC recognizes and queues the next rising edge event on the corresponding event input. The minimum latency from the rising edge transition of the general-purpose interrupt to the IPEND output asserted is three core clock cycles; however, the latency can be much higher, depending on the activity within and the state of the processor.

## DMA CONTROLLERS

ADSP-BF54x Blackfin processors have multiple, independent DMA channels that support automated data transfers with minimal overhead for the processor core. DMA transfers can occur between the ADSP-BF54x processors' internal memories and any of the DMA-capable peripherals. Additionally, DMA transfers can be accomplished between any of the DMA-capable peripherals and external devices connected to the external memory interfaces, including DDR and asynchronous memory controllers.

While the USB controller and MXVR have their own dedicated DMA controllers, the other on-chip peripherals are managed by two centralized DMA controllers, called DMAC1 (32-bit) and DMAC0 (16-bit). Both operate in the SCLK domain. Each DMA controller manages 12 independent peripheral DMA channels, as well as two independent memory DMA streams. The DMAC1 controller masters high-bandwidth peripherals over a dedicated 32-bit DMA access bus (DAB32). Similarly, the DMAC0 controller masters most serial interfaces over the 16-bit DAB16 bus. Individual DMA channels have fixed access priority on the DAB buses. DMA priority of peripherals is managed by a flexible peripheral-to-DMA channel assignment scheme.

All four DMA controllers use the same 32-bit DCB bus to exchange data with L1 memory. This includes L1 ROM, but excludes scratchpad memory. Fine granulation of L1 memory and special DMA buffers minimize potential memory conflicts when the L1 memory is accessed simultaneously by the core. Similarly, there are dedicated DMA buses between the external bus interface unit (EBIU) and the three DMA controllers (DMAC1, DMAC0, and USB) that arbitrate DMA accesses to external memories and the boot ROM.

The ADSP-BF54x Blackfin processors' DMA controllers support both 1-dimensional (1D) and 2-dimensional (2D) DMA transfers. DMA transfer initialization can be implemented from registers or from sets of parameters called descriptor blocks.

The 2D DMA capability supports arbitrary row and column sizes up to 64K elements by 64K elements, and arbitrary row and column step sizes up to ±32K elements. Furthermore, the column step size can be less than the row step size, allowing implementation of interleaved data streams. This feature is especially useful in video applications where data can be de-interleaved on the fly.

Examples of DMA types supported by the ADSP-BF54x Blackfin processors' DMA controllers include:

- A single, linear buffer that stops upon completion
- A circular, auto-refreshing buffer that interrupts on each full or fractionally full buffer
- 1D or 2D DMA using a linked list of descriptors
- 2D DMA using an array of descriptors, specifying only the base DMA address within a common page

In addition to the dedicated peripheral DMA channels, the DMAC1 and DMAC0 controllers each feature two memory DMA channel pairs for transfers between the various memories of the ADSP-BF54x Blackfin processors. This enables transfers of blocks of data between any of the memories—including external DDR, ROM, SRAM, and flash memory—with minimal processor intervention. Like peripheral DMAs, memory DMA transfers can be controlled by a very flexible descriptor-based methodology or by a standard register-based autobuffer mechanism.

The memory DMA channels of the DMAC1 controller (MDMA2 and MDMA3) can be controlled optionally by the external DMA request input pins. When used in conjunction with the External Bus Interface Unit (EBIU), this handshaked memory DMA (HMDMA) scheme can be used to efficiently exchange data with block-buffered or FIFO-style devices connected externally. Users can select whether the DMA request pins control the source or the destination side of the memory DMA. It allows control of the number of data transfers for memory DMA. The number of transfers per edge is programmable. This feature can be programmed to allow memory DMA to have an increased priority on the external bus relative to the core.

### *Host DMA Port Interface*

The host DMA port (HOSTDP) facilitates a host device external to the ADSP-BF54x Blackfin processors to be a DMA master and transfer data back and forth. The host device always masters the transactions, and the processor is always a DMA slave device.

The HOSTDP is enabled through the peripheral access bus. Once the port has been enabled, the transactions are controlled by the external host. The external host programs standard DMA configuration words in order to send/receive data to any valid internal or external memory location. The host DMA port controller includes the following features:

- Allows an external master to configure DMA read/write data transfers and read port status
- Uses a flexible asynchronous memory protocol for its external interface
- Allows an 8- or 16-bit external data interface to the host device
- Supports half-duplex operation
- Supports little/big endian data transfers
- Acknowledge mode allows flow control on host transactions
- Interrupt mode guarantees a burst of FIFO depth host transactions

## REAL-TIME CLOCK

The ADSP-BF54x Blackfin processors' real-time clock (RTC) provides a robust set of digital watch features, including current time, stopwatch, and alarm. The RTC is clocked by a 32.768 kHz crystal external to the ADSP-BF54x Blackfin processors. The RTC peripheral has dedicated power supply pins so that it can remain powered up and clocked even when the rest of the processor is in a low-power state. The RTC provides several programmable interrupt options, including interrupt per second, minute, hour, or day clock ticks, interrupt on programmable stopwatch countdown, or interrupt at a programmed alarm time.

The 32.768 kHz input clock frequency is divided down to a 1 Hz signal by a prescaler. The counter function of the timer consists of four counters: a 60-second counter, a 60-minute counter, a 24-hour counter, and a 32,768-day counter.

When enabled, the alarm function generates an interrupt when the output of the timer matches the programmed value in the alarm control register. There are two alarms. The first alarm is for a time of day. The second alarm is for a day and time of that day.

The stopwatch function counts down from a programmed value with one-second resolution. When the stopwatch is enabled and the counter underflows, an interrupt is generated.

Like the other peripherals, the RTC can wake up the ADSP-BF54x processor from sleep mode upon generation of any RTC wakeup event. Additionally, an RTC wakeup event can wake up the ADSP-BF54x processors from deep sleep mode, and it can wake up the on-chip internal voltage regulator from the hibernate state.

Connect RTC pins RTXI and RTXO with external components as shown in Figure 4.

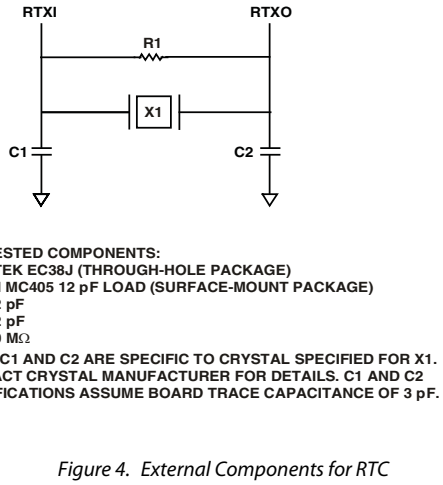



*Figure 4. External Components for RTC*

## WATCHDOG TIMER

The ADSP-BF54x processors include a 32-bit timer that can be used to implement a software watchdog function. A software watchdog can improve system reliability by forcing the processor to a known state through generation of a hardware reset, non-maskable interrupt (NMI), or general-purpose interrupt if the timer expires before being reset by software. The programmer initializes the count value of the timer, enables the appropriate interrupt, and then enables the timer. Thereafter, the software must reload the counter before it counts to zero from the programmed value. This protects the system from remaining in an unknown state where software, which would normally reset the timer, has stopped running due to an external noise condition or software error.

If configured to generate a hardware reset, the watchdog timer resets both the core and the ADSP-BF54x processors' peripherals. After a reset, software can determine if the watchdog was the source of the hardware reset by interrogating a status bit in the watchdog timer control register.

The timer is clocked by the system clock (SCLK) at a maximum frequency of $f_{SCLK}$.

## TIMERS

There are up to two timer units in the ADSP-BF54x Blackfin processors. One unit provides eight general-purpose programmable timers, and the other unit provides three. Each timer has an external pin that can be configured either as a pulse width modulator (PWM) or timer output, as an input to clock the timer, or as a mechanism for measuring pulse widths and periods of external events. These timers can be synchronized to an external clock input on the TMRx pins, an external clock TMRCLK input pin, or to the internal SCLK.

The timer units can be used in conjunction with the four UARTs and the CAN controllers to measure the width of the pulses in the data stream to provide a software auto-baud detect function for the respective serial channels.

The timers can generate interrupts to the processor core, providing periodic events for synchronization to either the system clock or to a count of external signals.

In addition to the general-purpose programmable timers, another timer is also provided by the processor core. This extra timer is clocked by the internal processor clock and is typically used as a system tick clock for generation of periodic operating system interrupts.

## UP/DOWN COUNTER AND THUMBWHEEL INTERFACE

A 32-bit up/down counter is provided that can sense the 2-bit quadrature or binary codes typically emitted by industrial drives or manual thumb wheels. The counter can also operate in general-purpose up/down count modes. Then count direction is either controlled by a level-sensitive input pin or by two edge detectors.

A third input can provide flexible zero marker support and can alternatively be used to input the push-button signal of thumb wheels. All three pins have a programmable debouncing circuit.

An internal signal forwarded to the timer unit enables one timer to measure the intervals between count events. Boundary registers enable auto-zero operation or simple system warning by interrupts when programmable count values are exceeded.

## SERIAL PORTS (SPORTS)

The ADSP-BF54x Blackfin processors incorporate up to four dual-channel synchronous serial ports (SPORT0, SPORT1, SPORT2, and SPORT3) for serial and multiprocessor communications. The SPORTs support the following features:

- $I^2S$ capable operation.
- Bidirectional operation. Each SPORT has two sets of independent transmit and receive pins, enabling up to eight channels of $I^2S$ stereo audio.
- Buffered (8-deep) transmit and receive ports. Each port has a data register for transferring data words to and from other processor components and shift registers for shifting data in and out of the data registers.
- Clocking. Each transmit and receive port can either use an external serial clock or generate its own, in frequencies ranging from ($f_{SCLK}$/131,070) Hz to ($f_{SCLK}$/2) Hz.
- Word length. Each SPORT supports serial data words from 3 to 32 bits in length, transferred most-significant-bit first or least-significant-bit first.
- Framing. Each transmit and receive port can run with or without frame sync signals for each data word. Frame sync signals can be generated internally or externally, active high or low, and with either of two pulse widths and early or late frame sync.

- Companding in hardware. Each SPORT can perform A-law or μ-law companding according to ITU recommendation G.711. Companding can be selected on the transmit and/or receive channel of the SPORT without additional latencies.
- DMA operations with single-cycle overhead. Each SPORT can receive and transmit multiple buffers of memory data automatically. The processor can link or chain sequences of DMA transfers between a SPORT and memory.
- Interrupts. Each transmit and receive port generates an interrupt upon completing the transfer of a data word or after transferring an entire data buffer or buffers through DMA.
- Multichannel capability. Each SPORT supports 128 channels out of a 1024-channel window and is compatible with the H.100, H.110, MVIP-90, and HMVIP standards.

## SERIAL PERIPHERAL INTERFACE (SPI) PORTS

The ADSP-BF54x Blackfin processors have up to three SPI-compatible ports that allow the processor to communicate with multiple SPI-compatible devices.

Each SPI port uses three pins for transferring data: two data pins (master output slave input, SPIxMOSI, and master input-slave output, SPIxMISO) and a clock pin (serial clock, SPIxSCK). An SPI chip select input pin ($\overline{\text{SPIxSS}}$) lets other SPI devices select the processor, and three SPI chip select output pins per SPI port $\overline{\text{SPIxSELy}}$ let the processor select other SPI devices. The SPI select pins are reconfigured general-purpose I/O pins. Using these pins, the SPI ports provide a full-duplex, synchronous serial interface, which supports both master/slave modes and multimaster environments.

The SPI port's baud rate and clock phase/polarities are programmable, and it has an integrated DMA controller, configurable to support transmit or receive data streams. The SPI's DMA controller can only service unidirectional accesses at any given time.

The SPI port's clock rate is calculated as

$$SPI\ Clock\ Rate = \frac{f_{SCLK}}{2 \times SPI_BAUD}$$

Where the 16-bit SPI_BAUD register contains a value of 2 to 65,535.

During transfers, the SPI port transmits and receives simultaneously by serially shifting data in and out on its two serial data lines. The serial clock line synchronizes the shifting and sampling of data on the two serial data lines.

## UART PORTS (UARTS)

The ADSP-BF54x Blackfin processors provide up to four full-duplex universal asynchronous receiver/transmitter (UART) ports. Each UART port provides a simplified UART interface to other peripherals or hosts, supporting full-duplex, DMA-supported, asynchronous transfers of serial data. A UART port includes support for five to eight data bits, one or two stop bits, and none, even, or odd parity. Each UART port supports two modes of operation:

- PIO (programmed I/O). The processor sends or receives data by writing or reading I/O-mapped UART registers. The data is double-buffered on both transmit and receive.
- DMA (direct memory access). The DMA controller transfers both transmit and receive data. This reduces the number and frequency of interrupts required to transfer data to and from memory. Each UART has two dedicated DMA channels, one for transmit and one for receive. These DMA channels have lower default priority than most DMA channels because of their relatively low service rates. Flexible interrupt timing options are available on the transmit side.

Each UART port's baud rate, serial data format, error code generation and status, and interrupts are programmable:

- Supporting bit rates ranging from ($f_{SCLK}$/1,048,576) to ($f_{SCLK}$) bits per second.
- Supporting data formats from seven to 12 bits per frame.
- Both transmit and receive operations can be configured to generate maskable interrupts to the processor.

The UART port's clock rate is calculated as

$$UART\ Clock\ Rate = \frac{f_{SCLK}}{16^{(1-EDBO)} \times UART_Divisor}$$

Where the 16-bit UART divisor comes from the UARTx_DLH register (most significant 8 bits) and UARTx_DLL register (least significant eight bits), and the EDBO is a bit in the UARTx_GCTL register.

In conjunction with the general-purpose timer functions, autobaud detection is supported.

UART1 and UART3 feature a pair of $\overline{\text{UARTxRTS}}$ (request to send) and $\overline{\text{UARTxCTS}}$ (clear to send) signals for hardware flow purposes. The transmitter hardware is automatically prevented from sending further data when the $\overline{\text{UARTxCTS}}$ input is de-asserted. The receiver can automatically de-assert its $\overline{\text{UARTxRTS}}$ output when the enhanced receive FIFO exceeds a certain high-water level. The capabilities of the UARTs are further extended with support for the Infrared Data Association (IrDA®) Serial Infrared Physical Layer Link Specification (SIR) protocol.

## CONTROLLER AREA NETWORK (CAN)

The ADSP-BF54x Blackfin processors offer up to two CAN controllers that are communication controllers that implement the controller area network (CAN) 2.0B (active) protocol. This protocol is an asynchronous communications protocol used in both industrial and automotive control systems. The CAN protocol is well suited for control applications due to its capability to communicate reliably over a network since the protocol incorporates CRC checking, message error tracking, and fault node confinement.

The ADSP-BF54x Blackfin processors' CAN controllers offer the following features:

- 32 mailboxes (8 receive only, 8 transmit only, 16 configurable for receive or transmit).
- Dedicated acceptance masks for each mailbox.
- Additional data filtering on first two bytes.
- Support for both the standard (11-bit) and extended (29-bit) identifier (ID) message formats.
- Support for remote frames.
- Active or passive network support.
- CAN wakeup from hibernation mode (lowest static power consumption mode).
- Interrupts, including: TX complete, RX complete, error and global.

The electrical characteristics of each network connection are very demanding, so the CAN interface is typically divided into two parts: a controller and a transceiver. This allows a single controller to support different drivers and CAN networks. The ADSP-BF54x Blackfin processors' CAN module represents only the controller part of the interface. The controller interface supports connection to 3.3 V high speed, fault-tolerant, single-wire transceivers.

An additional crystal is not required to supply the CAN clock, as the CAN clock is derived from the processor system clock (SCLK) through a programmable divider.

## TWI CONTROLLER INTERFACE

The ADSP-BF54x Blackfin processors include up to two 2-wire interface (TWI) modules for providing a simple exchange method of control data between multiple devices. The modules are compatible with the widely used $I^2C$ bus standard. The TWI modules offer the capabilities of simultaneous master and slave operation and support for both 7-bit addressing and multimedia data arbitration. Each TWI interface uses two pins for transferring clock (SCLx) and data (SDAx), and supports the protocol at speeds up to 400K bits/sec. The TWI interface pins are compatible with 5 V logic levels.

Additionally, the ADSP-BF54x Blackfin processors' TWI modules are fully compatible with serial camera control bus (SCCB) functionality for easier control of various CMOS camera sensor devices.

## PORTS

Because of their rich set of peripherals, the ADSP-BF54x Blackfin processors group the many peripheral signals to ten ports—referred to as Port A to Port J. Most ports contain 16 pins, though some have fewer. Many of the associated pins are shared by multiple signals. The ports function as multiplexer controls. Every port has its own set of memory-mapped registers to control port muxing and GPIO functionality.

### *General-Purpose I/O (GPIO)*

Every pin in Port A to Port J can function as a GPIO pin, resulting in a GPIO pin count up to 154. While it is unlikely that all GPIO pins will be used in an application, as all pins have multiple functions, the richness of GPIO functionality guarantees unrestrictive pin usage. Every pin that is not used by any function can be configured in GPIO mode on an individual basis.

After reset, all pins are in GPIO mode by default. Since neither GPIO output nor input drivers are active by default, unused pins can be left unconnected. GPIO data and direction control registers provide flexible write-one-to-set and write-one-to-clear mechanisms so that independent software threads do not need to protect against each other because of expensive read-modify-write operations when accessing the same port.

### *Pin Interrupts*

Every port pin on ADSP-BF54x Blackfin processors can request interrupts in either an edge-sensitive or a level-sensitive manner with programmable polarity. Interrupt functionality is decoupled from GPIO operation. Four system-level interrupt channels (PINT0, PINT1, PINT2 and PINT3) are reserved for this purpose. Each of these interrupt channels can manage up to 32 interrupt pins. The assignment from pin to interrupt is not performed on a pin-by-pin basis. Rather, groups of eight pins (half ports) can be flexibly assigned to interrupt channels.

Every pin interrupt channel features a special set of 32-bit memory-mapped registers that enables half-port assignment and interrupt management. This not only includes masking, identification, and clearing of requests, it also enables access to the respective pin states and use of the interrupt latches regardless of whether the interrupt is masked or not. Most control registers feature multiple MMR address entries to write-one-to-set or write-one-to-clear them individually.

## PIXEL COMPOSITOR (PIXC)

The pixel compositor (PIXC) provides image overlays with transparent-color support, alpha blending, and color space conversion capabilities for output to TFT LCDs and NTSC/PAL video encoders. It provides all of the control to allow two data streams from two separate data buffers to be combined, blended, and converted into appropriate forms for both LCD panels and digital video outputs. The main image buffer provides the basic background image, which is presented in the data stream. The overlay image buffer allows the user to add multiple foreground text, graphics, or video objects on top of the main image or video data stream.

## ENHANCED PARALLEL PERIPHERAL INTERFACE (EPPI)

The ADSP-BF54x Blackfin processors provide up to three enhanced parallel peripheral interfaces (EPPIs), supporting data widths up to 24 bits. The EPPI supports direct connection to TFT LCD panels, parallel analog-to-digital and digital-to-analog converters, video encoders and decoders, image sensor modules and other general-purpose peripherals.

The following features are supported in the EPPI module:

- Programmable data length: 8 bits, 10 bits, 12 bits, 14 bits, 16 bits, 18 bits, and 24 bits per clock.
- Bidirectional and half-duplex port.
- Clock can be provided externally or can be generated internally.
- Various framed and non-framed operating modes. Frame syncs can be generated internally or can be supplied by an external device.
- Various general-purpose modes with zero to three frame syncs for both receive and transmit directions.
- ITU-656 status word error detection and correction for ITU-656 receive modes.
- ITU-656 preamble and status word decode.
- Three different modes for ITU-656 receive modes: active video only, vertical blanking only, and entire field mode.
- Horizontal and vertical windowing for GP 2 and 3 frame sync modes.
- Optional packing and unpacking of data to/from 32 bits from/to 8, 16 and 24 bits. If packing/unpacking is enabled, endianness can be changed to change the order of packing/unpacking of bytes/words.
- Optional sign extension or zero fill for receive modes.
- During receive modes, alternate even or odd data samples can be filtered out.
- Programmable clipping of data values for 8-bit transmit modes.
- RGB888 can be converted to RGB666 or RGB565 for transmit modes.
- Various de-interleaving/interleaving modes for receiving/transmitting 4:2:2 YCrCb data.
- FIFO watermarks and urgent DMA features.
- Clock gating by an external device asserting the clock gating control signal.
- Configurable LCD data enable (DEN) output available on Frame Sync 3.

## USB ON-THE-GO DUAL-ROLE DEVICE CONTROLLER

The USB OTG dual-role device controller (USBDRC) provides a low-cost connectivity solution for consumer mobile devices such as cell phones, digital still cameras, and MP3 players, allowing these devices to transfer data using a point-to-point USB connection without the need for a PC host. The USBDRC module can operate in a traditional USB peripheral-only mode as well as the host mode presented in the On-the-Go (OTG) supplement to the USB 2.0 specification. In host mode, the USB module supports transfers at high speed (480 Mbps), full speed (12 Mbps), and low speed (1.5 Mbps) rates. Peripheral-only mode supports the high and full speed transfer rates.

The USB clock (USB_XI) is provided through a dedicated external crystal or crystal oscillator. See Table 62 for related timing requirements. If using a fundamental mode crystal to provide the USB clock, connect the crystal between USB_XI and USB_XO with a circuit similar to that shown in Figure 7. Use a parallel-resonant, fundamental mode, microprocessor-grade crystal. If a third-overtone crystal is used, follow the circuit guidelines outlined in Clock Signals on Page 18 for third-overtone crystals.

The USB On-the-Go dual-role device controller includes a Phase Locked Loop with programmable multipliers to generate the necessary internal clocking frequency for USB. The multiplier value should be programmed based on the USB_XI clock frequency to achieve the necessary 480 MHz internal clock for USB high speed operation. For example, for a USB_XI crystal frequency of 24 MHz, the USB_PLLOSC_CTRL register should be programmed with a multiplier value of 20 to generate a 480 MHz internal clock.

## ATA/ATAPI-6 INTERFACE

The ATAPI interface connects to CD/DVD and HDD drives and is ATAPI-6 compliant. The controller implements the peripheral I/O mode, the multi-DMA mode, and the Ultra DMA mode. The DMA modes enable faster data transfer and reduced host management. The ATAPI controller supports PIO, multi-DMA, and ultra DMA ATAPI accesses. Key features include:

- Supports PIO modes 0, 1, 2, 3, 4
- Supports multiword DMA modes 0, 1, 2
- Supports ultra DMA modes 0, 1, 2, 3, 4, 5 (up to UDMA 100)
- Programmable timing for ATA interface unit
- Supports CompactFlash cards using true IDE mode

By default, the ATAPI_A0-2 address signals and the ATAPI_D0-15 data signals are shared on the asynchronous memory interface with the asynchronous memory and NAND flash controllers. The data and address signals can be remapped to GPIO ports F and G, respectively, by setting PORTF_MUX[1:0] to b#01.

## KEYPAD INTERFACE

The keypad interface is a 16-pin interface module that is used to detect the key pressed in a 8 × 8 (maximum) keypad matrix. The size of the input keypad matrix is programmable. The interface is capable of filtering the bounce on the input pins, which is common in keypad applications. The width of the filtered bounce is programmable. The module is capable of generating an interrupt request to the core once it identifies that any key has been pressed.

The interface supports a press-release-press mode and infrastructure for a press-hold mode. The former mode identifies a press, release and press of a key as two consecutive presses of the same key, whereas the latter mode checks the input key's state in periodic intervals to determine the number of times the same

key is meant to be pressed. It is possible to detect when multiple keys are pressed simultaneously and to provide limited key resolution capability when this happens.

## SECURE DIGITAL (SD)/SDIO CONTROLLER

The SD/SDIO controller is a serial interface that stores data at a data rate of up to 10M bytes per second using a 4-bit data line.

The SD/SDIO controller supports the SD memory mode only. The interface supports all the power modes and performs error checking by CRC.

## CODE SECURITY

An OTP/security system, consisting of a blend of hardware and software, provides customers with a flexible and rich set of code security features with Lockbox® secure technology. Key features include:

- OTP memory
- Unique chip ID
- Code authentication
- Secure mode of operation

The security scheme is based upon the concept of authentication of digital signatures using standards-based algorithms and provides a secure processing environment in which to execute code and protect assets. See Lockbox Secure Technology Disclaimer on Page 24.

## MEDIA TRANSCEIVER MAC LAYER (MXVR)

The ADSP-BF549 Blackfin processors provide a media transceiver (MXVR) MAC layer, allowing the processor to be connected directly to a MOST® [1] network through an FOT. See Figure 5 on Page 17 for an example of a MXVR MOST connection.

The MXVR is fully compatible with industry-standard standalone MOST controller devices, supporting 22.579 Mbps or 24.576 Mbps data transfer. It offers faster lock times, greater jitter immunity, and a sophisticated DMA scheme for data transfers. The high speed internal interface to the core and L1 memory allows the full bandwidth of the network to be utilized. The MXVR can operate as either the network master or as a network slave.

The MXVR supports synchronous data, asynchronous packets, and control messages using dedicated DMA channels that operate autonomously from the processor core moving data to and from L1 and/or L2 memory. Synchronous data is transferred to or from the synchronous data physical channels on the MOST bus through eight programmable DMA channels. The synchronous data DMA channels can operate in various modes including modes that trigger DMA operation when data patterns are detected in the receive data stream. Furthermore, two DMA channels support asynchronous traffic, and two others support control message traffic.

Interrupts are generated when a user-defined amount of synchronous data has been sent or received by the processor or when asynchronous packets or control messages have been sent or received.

The MXVR peripheral can wake up the ADSP-BF549 Blackfin processor from sleep mode when a wakeup preamble is received over the network or based on any other MXVR interrupt event. Additionally, detection of network activity by the MXVR can be used to wake up the ADSP-BF549 Blackfin processor from the hibernate state. These features allow the ADSP-BF549 processor to operate in a low-power state when there is no network activity or when data is not currently being received or transmitted by the MXVR.

The MXVR clock is provided through a dedicated external crystal or crystal oscillator. The frequency of the external crystal or crystal oscillator can be 256 Fs, 384 Fs, 512 Fs, or 1024 Fs for Fs = 38 kHz, 44.1 kHz, or 48 kHz. If using a crystal to provide the MXVR clock, use a parallel-resonant, fundamental mode, microprocessor-grade crystal.

## DYNAMIC POWER MANAGEMENT

The ADSP-BF54x Blackfin processors provide five operating modes, each with a different performance/power profile. In addition, dynamic power management provides the control functions to dynamically alter the processor core supply voltage, further reducing power dissipation. Control of clocking to each of the ADSP-BF54x Blackfin processors' peripherals also reduces power consumption. See Table 5 for a summary of the power settings for each mode.

### *Full-On Operating Mode—Maximum Performance*

In the full-on mode, the PLL is enabled and is not bypassed, providing the capability to run at the maximum operational frequency. This is the power-up default execution state in which maximum performance can be achieved. The processor core and all enabled peripherals run at full speed.

### *Active Operating Mode—Moderate Power Savings*

In the active mode, the PLL is enabled but bypassed. Because the PLL is bypassed, the processor's core clock (CCLK) and system clock (SCLK) run at the input clock (CLKIN) frequency. DMA access is available to appropriately configured L1 memories.

In the active mode, it is possible to disable the control input to the PLL by setting the PLL_OFF bit in the PLL control register. This register can be accessed with a user-callable routine in the on-chip ROM called bfrom_SysControl(). For more information, see the "Dynamic Power Management" chapter in the *ADSP-BF54x Blackfin Processor Hardware Reference*. If disabled, the PLL must be re-enabled before transitioning to the full-on or sleep modes.

[1] MOST is a registered trademark of Standard Microsystems, Corp.

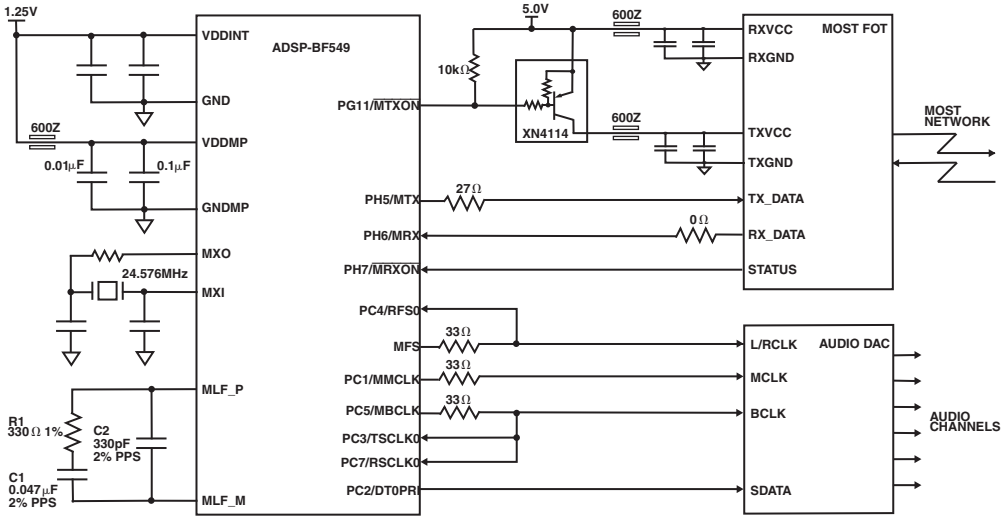



*Figure 5. MXVR MOST Connection*

**Table 5. Power Settings**

| Mode/State | PLL | PLL Bypassed | Core Clock (CCLK) | System Clock (SCLK) | Core Power |
|---|---|---|---|---|---|
| Full On | Enabled | No | Enabled | Enabled | On |
| Active | Enabled/ Disabled | Yes | Enabled | Enabled | On |
| Sleep | Enabled | - | Disabled | Enabled | On |
| Deep Sleep | Disabled | - | Disabled | Disabled | On |
| Hibernate | Disabled | - | Disabled | Disabled | Off |

### *Sleep Operating Mode—High Dynamic Power Savings*

The sleep mode reduces dynamic power dissipation by disabling the clock to the processor core (CCLK). The PLL and system clock (SCLK), however, continue to operate in this mode. Typically an external event or RTC activity will wake up the processor. In the sleep mode, assertion of a wakeup event enabled in the SIC_IWRx register causes the processor to sense the value of the BYPASS bit in the PLL control register (PLL_CTL). If BYPASS is disabled, the processor transitions to the full on mode. If BYPASS is enabled, the processor transitions to the active mode.

In the sleep mode, system DMA access to L1 memory is not supported.

### *Deep Sleep Operating Mode—Maximum Dynamic Power Savings*

The deep sleep mode maximizes dynamic power savings by disabling the clocks to the processor core (CCLK) and to all synchronous peripherals (SCLK). Asynchronous peripherals, such as the RTC, may still be running but will not be able to access internal resources or external memory. This powered-down mode can only be exited by assertion of the reset interrupt ($\overline{\text{RESET}}$) or by an asynchronous interrupt generated by the RTC. In deep sleep mode, an asynchronous RTC interrupt causes the processor to transition to the active mode. Assertion of $\overline{\text{RESET}}$ while in deep sleep mode causes the processor to transition to the full on mode.

### *Hibernate State—Maximum Static Power Savings*

The hibernate state maximizes static power savings by disabling the voltage and clocks to the processor core (CCLK) and to all the synchronous peripherals (SCLK). The internal voltage regulator for the processor can be shut off by using the bfrom_SysControl() function in the on-chip ROM. This sets the internal power supply voltage ($V_{DDINT}$) to 0 V to provide the greatest power savings mode. Any critical information stored internally (memory contents, register contents, and so on) must be written to a non-volatile storage device prior to removing power if the processor state is to be preserved.

Since $V_{DDEXT}$ is still supplied in this mode, all of the external pins three-state, unless otherwise specified. This allows other devices that may be connected to the processor to have power still applied without drawing unwanted current.

The internal supply regulator can be woken up by CAN, by the MXVR, by the keypad, by the up/down counter, by the USB, and by some GPIO pins. It can also be woken up by a real-time clock wakeup event or by asserting the $\overline{\text{RESET}}$ pin. Waking up from hibernate state initiates the hardware reset sequence.

With the exception of the VR_CTL and the RTC registers, all internal registers and memories lose their content in hibernate state. State variables may be held in external SRAM or DDR memory.

### *Power Domains*

As shown in Table 6, the ADSP-BF54x Blackfin processors support different power domains. The use of multiple power domains maximizes flexibility while maintaining compliance with industry standards and conventions. By isolating the internal logic of the ADSP-BF54x Blackfin processors into its own power domain separate from the RTC and other I/O, the processors can take advantage of dynamic power management without affecting the RTC or other I/O devices. There are no sequencing requirements for the various power domains.

**Table 6. Power Domains**

| Power Domain | VDD Range |
|---|---|
| All internal logic, except RTC, DDR, and USB | $V_{DDINT}$ |
| RTC internal logic and crystal I/O | $V_{DDRTC}$ |
| DDR external memory supply | $V_{DDDDR}$ |
| USB internal logic and crystal I/O | $V_{DDUSB}$ |
| Internal voltage regulator | $V_{DDVR}$ |
| MXVR PLL and logic | $V_{DDMP}$ |
| All other I/O | $V_{DDEXT}$ |

## VOLTAGE REGULATION

The ADSP-BF54x Blackfin processors provide an on-chip voltage regulator that can generate processor core voltage levels from an external supply (see specifications in Operating Conditions on Page 34). Figure 6 on Page 18 shows the typical external components required to complete the power management system. The regulator controls the internal logic voltage levels and is programmable with the voltage regulator control register (VR_CTL) in increments of 50 mV. This register can be accessed using the bfrom_SysControl() function in the on-chip ROM. To reduce standby power consumption, the internal voltage regulator can be programmed to remove power to the processor core while keeping I/O power supplied. While in hibernate state, $V_{DDEXT}$, $V_{DDRTC}$, $V_{DDDDR}$, $V_{DDUSB}$, and $V_{DDVR}$ can still be applied, eliminating the need for external buffers. The voltage regulator can be activated from this power-down state by assertion of the $\overline{\text{RESET}}$ pin, which then initiates a boot sequence. The regulator can also be disabled and bypassed at the user's discretion. For all 600 MHz speed grade models and all automotive grade models, the internal voltage regulator must not be used and $V_{DDVR}$ must be tied to $V_{DDEXT}$. For additional information regarding design of the voltage regulator circuit, see *Switching Regulator Design Considerations for the ADSP-BF533 Blackfin Processors (EE-228)*.

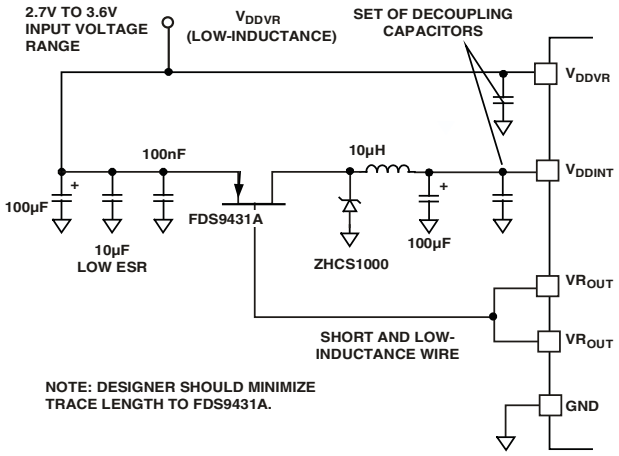



*Figure 6. Voltage Regulator Circuit*

## CLOCK SIGNALS

The ADSP-BF54x Blackfin processors can be clocked by an external crystal, a sine wave input, or a buffered, shaped clock derived from an external clock oscillator.

If an external clock is used, it should be a TTL-compatible signal and must not be halted, changed, or operated below the specified frequency during normal operation. This signal is connected to the processor's CLKIN pin. When an external clock is used, the XTAL pin must be left unconnected.

Alternatively, because the ADSP-BF54x Blackfin processors include an on-chip oscillator circuit, an external crystal may be used. For fundamental frequency operation, use the circuit shown in Figure 7. A parallel-resonant, fundamental frequency, microprocessor-grade crystal is connected across the CLKIN and XTAL pins. The on-chip resistance between CLKIN and the XTAL pin is in the 500 kΩ range. Typically, further parallel resistors are not recommended. The two capacitors and the series resistor shown in Figure 7 fine-tune phase and amplitude of the sine frequency. The 1MOhm pull-up resistor on the XTAL pin guarantees that the clock circuit is properly held inactive when the processor is in the hibernate state.

The capacitor and resistor values shown in Figure 7 are typical values only. The capacitor values are dependent upon the crystal manufacturers' load capacitance recommendations and the PCB physical layout. The resistor value depends on the drive level specified by the crystal manufacturer. System designs should verify the customized values based on careful investigations on multiple devices over temperature range.

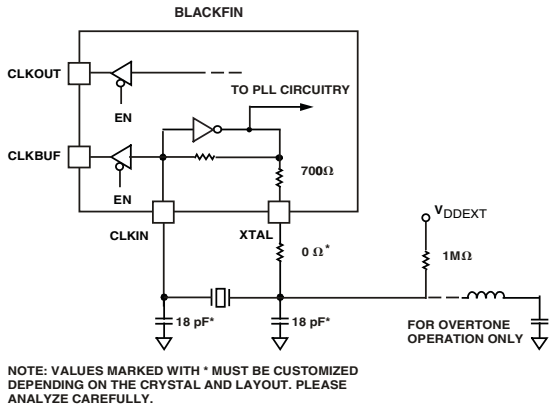


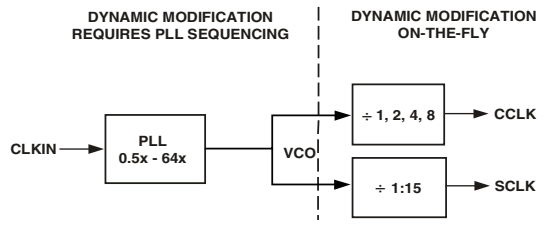

*Figure 7. External Crystal Connections*

A third-overtone crystal can be used at frequencies above 25 MHz. The circuit is then modified to ensure crystal operation only at the third overtone by adding a tuned inductor circuit as shown in Figure 7. A design procedure for third-overtone operation is discussed in detail in an Application Note, *Using Third Overtone Crystals (EE-168)*.

The Blackfin core runs at a different clock rate than the on-chip peripherals. As shown in Figure 8 on Page 19, the core clock (CCLK) and system peripheral clock (SCLK) are derived from the input clock (CLKIN) signal. An on-chip PLL is capable of multiplying the CLKIN signal by a programmable 0.5× to 64× multiplication factor (bounded by specified minimum and maximum VCO frequencies). The default multiplier is 8×, but it can be modified by a software instruction sequence. This sequence is managed by the bfrom_SysControl() function in the on-chip ROM.

On-the-fly CCLK and SCLK frequency changes can be applied by using the bfrom_SysControl() function in the on-chip ROM. Whereas the maximum allowed CCLK and SCLK rates depend on the applied voltages $V_{DDINT}$ and $V_{DDEXT}$, the VCO is always permitted to run up to the frequency specified by the part's speed grade.

The CLKOUT pin reflects the SCLK frequency to the off-chip world. It functions as a reference for many timing specifications. While inactive by default, it can be enabled using the EBIU_AMGCTL register.




Note: For CCLK and SCLK specifications, see Table 16.

*Figure 8. Frequency Modification Methods*

All on-chip peripherals are clocked by the system clock (SCLK). The system clock frequency is programmable by means of the SSEL3–0 bits of the PLL_DIV register. The values programmed into the SSEL fields define a divide ratio between the PLL output (VCO) and the system clock. SCLK divider values are 1 through 15. Table 7 illustrates typical system clock ratios. The default ratio is 4.

**Table 7. Example System Clock Ratios**

| Signal Name SSEL3–0 | Divider Ratio VCO/SCLK | Example Frequency Ratios (MHz) | |
|---|---|---|---|
| | | VCO | SCLK |
| 0010 | 2:1 | 200 | 100 |
| 0110 | 6:1 | 300 | 50 |
| 1010 | 10:1 | 500 | 50 |

Note that the divisor ratio must be chosen to limit the system clock frequency to its maximum of $f_{SCLK}$. The SSEL value can be dynamically changed without any PLL lock latencies by writing the appropriate values to the PLL divisor register (PLL_DIV) using the bfrom_SysControl() function in the on-chip ROM.

The core clock (CCLK) frequency can also be dynamically changed by means of the CSEL1–0 bits of the PLL_DIV register. Supported CCLK divider ratios are 1, 2, 4, and 8, as shown in Table 8. The default ratio is 1. This programmable core clock capability is useful for fast core frequency modifications.

The maximum CCLK frequency not only depends on the part's speed grade, it also depends on the applied $V_{DDINT}$ voltage. See Table 13 on Page 35 for details.

**Table 8. Core Clock Ratios**

| Signal Name CSEL1–0 | Divider Ratio VCO/CCLK | Example Frequency Ratios (MHz) | |
|---|---|---|---|
| | | VCO | CCLK |
| 00 | 1:1 | 300 | 300 |
| 01 | 2:1 | 300 | 150 |
| 10 | 4:1 | 500 | 125 |
| 11 | 8:1 | 200 | 25 |

## BOOTING MODES

The ADSP-BF54x Blackfin processors have many mechanisms (listed in Table 9) for automatically loading internal and external memory after a reset. The boot mode is specified by four BMODE input pins dedicated to this purpose. There are two categories of boot modes: master and slave. In master boot

modes, the processor actively loads data from parallel or serial memories. In slave boot modes, the processor receives data from an external host device.

**Table 9. Booting Modes**

| BMODE3–0 | Description |
|---|---|
| 0000 | Idle-no boot |
| 0001 | Boot from 8- or 16-bit external flash memory |
| 0010 | Boot from 16-bit asynchronous FIFO |
| 0011 | Boot from serial SPI memory (EEPROM or flash) |
| 0100 | Boot from SPI host device |
| 0101 | Boot from serial TWI memory (EEPROM or flash) |
| 0110 | Boot from TWI host |
| 0111 | Boot from UART host |
| 1000 | Reserved |
| 1001 | Reserved |
| 1010 | Boot from DDR SDRAM/Mobile DDR SDRAM |
| 1011 | Boot from OTP memory |
| 1100 | Reserved |
| 1101 | Boot from 8- or 16-bit NAND flash memory via NFC |
| 1110 | Boot from 16-bit host DMA |
| 1111 | Boot from 8-bit host DMA |

The boot modes listed in Table 9 provide a number of mechanisms for automatically loading the processor's internal and external memories after a reset. By default, all boot modes use the slowest allowed configuration settings. Default settings can be altered via the initialization code feature at boot time or by proper OTP programming at pre-boot time. Some boot modes require a boot host wait (HWAIT) signal, which is a GPIO output signal that is driven and toggled by the boot kernel at boot time. If pulled high through an external pull-up resistor, the HWAIT signal behaves active high and will be driven low when the processor is ready for data. Conversely, when pulled low, HWAIT is driven high when the processor is ready for data. When the boot sequence completes, the HWAIT pin can be used for other purposes. By default, HWAIT functionality is on GPIO port B (PB11). However, if PB11 is otherwise utilized in the system, an alternate boot host wait (HWAITA) signal can be enabled on GPIO port H (PH7) by programming the OTP_ALTERNATE_HWAIT bit in the PBS00L OTP memory page.

The BMODE pins of the reset configuration register, sampled during power-on resets and software-initiated resets, implement the following modes:

- Idle-no boot mode (BMODE = 0x0)—In this mode, the processor goes into the idle state. The idle boot mode helps to recover from illegal operating modes, in case the OTP memory is misconfigured.
- Boot from 8- or 16-bit external flash memory—(BMODE = 0x1)—In this mode, the boot kernel loads the first block header from address 0x2000 0000 and, depending on instructions contained in the header, the boot kernel performs an 8- or 16-bit boot or starts program execution at the address provided by the header. By default, all configuration settings are set for the slowest device possible (3-cycle hold time; 15-cycle R/W access times; 4-cycle setup).

  The ARDY pin is not enabled by default. It can, however, be enabled by OTP programming. Similarly, all interface behavior and timings can be customized through OTP programming. This includes activation of burst-mode or page-mode operation. In this mode, all asynchronous interface signals are enabled at the port muxing level.
- Boot from 16-bit asynchronous FIFO (BMODE = 0x2)—In this mode, the boot kernel starts booting from address 0x2030 0000. Every 16-bit word that the boot kernel has to read from the FIFO must be requested by a low pulse on the DMAR1 pin.
- Boot from serial SPI memory, EEPROM or flash (BMODE = 0x3)—8-, 16-, 24- or 32-bit addressable devices are supported. The processor uses the PE4 GPIO pin to select a single SPI EEPROM or flash device and uses SPI0 to submit a read command and successive address bytes (0x00) until a valid 8-, 16-, 24-, or 32-bit addressable device is detected. Pull-up resistors are required on the $\overline{\text{SPI0SEL1}}$ and SPI0MISO pins. By default, a value of 0x85 is written to the SPI0_BAUD register.
- Boot from SPI host device (BMODE = 0x4)—The processor operates in SPI slave mode (using SPI0) and is configured to receive the bytes of the .LDR file from an SPI host (master) agent. The HWAIT signal must be interrogated by the host before every transmitted byte. A pull-up resistor is required on the $\overline{\text{SPI0SS}}$ input. A pull-down resistor on the serial clock (SPI0SCK) may improve signal quality and booting robustness.
- Boot from serial TWI memory, EEPROM or flash (BMODE = 0x5)—The processor operates in master mode (using TWI0) and selects the TWI slave with the unique ID 0xA0. The processor submits successive read commands to the memory device starting at two-byte internal address 0x0000 and begins clocking data into the processor. The TWI memory device should comply with Philips $I^2C$ Bus Specification version 2.1 and have the capability to autoincrement its internal address counter such that the contents of the memory device can be read sequentially. By default, a prescale value of 0xA and CLKDIV value of 0x0811 is used. Unless altered by OTP settings, an $I^2C$ memory that takes two address bytes is assumed. Development tools ensure that data that is booted to memories that cannot be accessed by the Blackfin core is written to an intermediate storage place and then copied to the final destination via memory DMA.
- Boot from TWI host (BMODE = 0x6)—The TWI host agent selects the slave with the unique ID 0x5F. The processor (using TWI0) replies with an acknowledgement, and the host can then download the boot stream. The TWI host agent should comply with Philips $I^2C$ Bus Specification ver-

sion 2.1. An $I^2C$ multiplexer can be used to select one processor at a time when booting multiple processors from a single TWI.

- Boot from UART host (BMODE = 0x7)—In this mode, the processor uses UART1 as the booting source. Using an autobaud handshake sequence, a boot-stream-formatted program is downloaded by the host. The host agent selects a bit rate within the UART's clocking capabilities.

  When performing the autobaud, the UART expects an "@" (0x40) character (eight data bits, one start bit, one stop bit, no parity bit) on the UART1RX pin to determine the bit rate. It then replies with an acknowledgement, which is composed of four bytes (0xBF, the value of UART1_DLL, the value of UART1_DLH, and finally 0x00). The host can then download the boot stream. The processor deasserts the $\overline{\text{UART1RTS}}$ output to hold off the host; $\overline{\text{UART1CTS}}$ functionality is not enabled at boot time.
- Boot from (DDR) SDRAM (BMODE = 0xA)—In this mode, the boot kernel starts booting from address 0x0000 0010. This is a warm boot scenario only. The SDRAM is expected to contain a valid boot stream and the SDRAM controller must have been configured by the OTP settings.
- Boot from 8-bit and 16-bit external NAND flash memory (BMODE = 0xD)—In this mode, auto detection of the NAND flash device is performed. The processor configures PORTJ GPIO pins PJ1 and PJ2 to enable the $\overline{\text{ND_CE}}$ and $\overline{\text{ND_RB}}$ signals, respectively. For correct device operation, pull-up resistors are required on both $\overline{\text{ND_CE}}$ (PJ1) and $\overline{\text{ND_RB}}$ (PJ2) signals. By default, a value of 0x0033 is written to the NFC_CTL register. The booting procedure always starts by booting from byte 0 of block 0 of the NAND flash device. In this boot mode, the HWAIT signal does not toggle. The respective GPIO pin remains in the high-impedance state.

  NAND flash boot supports the following features:
  - Device auto detection
  - Error detection and correction for maximum reliability
  - No boot stream size limitation
  - Peripheral DMA via channel 22, providing efficient transfer of all data (excluding the ECC parity data)
  - Software-configurable boot mode for booting from boot streams expanding multiple blocks, including bad blocks
  - Software-configurable boot mode for booting from multiple copies of the boot stream allowing for handling of bad blocks and uncorrectable errors
  - Configurable timing via OTP memory

  Small page NAND flash devices must have a 512-byte page size, 32 pages per block, a 16-byte spare area size and a bus configuration of eight bits. By default, all read requests from the NAND flash are followed by four address cycles.

  If the NAND flash device requires only three address cycles, then the device must be capable of ignoring the additional address cycle.

  The small page NAND flash device must comply with the following command set:

| |
|---|
| Reset: 0xFF |
| Read lower half of page: 0x00 |
| Read upper half of page: 0x01 |
| Read spare area: 0x50 |

  For large page NAND flash devices, the 4-byte electronic signature is read in order to configure the kernel for booting. This allows support for multiple large page devices. The fourth byte of the electronic signature must comply with the specifications in Table 10.

  Any configuration from Table 10 that also complies with the command set listed below is directly supported by the boot kernel. There are no restrictions on the page size or block size as imposed by the small-page boot kernel.

**Table 10. Byte 4 Electronic Signature Specification**

| | | | |
|---|---|---|---|
| Page Size (excluding spare area) | D1:D0 | 00 | 1K bytes |
| | | 01 | 2K bytes |
| | | 10 | 4K bytes |
| | | 11 | 8K bytes |
| Spare Area Size | D2 | 0 | 8 bytes/512 bytes |
| | | 1 | 16 bytes/512 bytes |
| Block Size (excluding spare area) | D5:4 | 00 | 64K bytes |
| | | 01 | 128K bytes |
| | | 10 | 256K bytes |
| | | 11 | 512K bytes |
| Bus Width | D6 | 0 | x8 |
| | | 1 | x16 |
| Not Used for Configuration | D3, D7 | | |

  Large page devices must support the following command set:

| |
|---|
| Reset: 0xFF |
| Read Electronic Signature: 0x90 |
| Read: 0x00, 0x30 (confirm command) |

  Large page devices must not support or react to NAND flash command 0x50. This is a small page NAND flash command used for device auto detection.

By default, the boot kernel will always issue five address cycles; therefore, if a large page device requires only four cycles, the device must be capable of ignoring the additional address cycle.

16-bit NAND flash memory devices must only support the issuing of command and address cycles via the lower eight bits of the data bus. Devices that use the full 16-bit bus for command and address cycles are not supported.

- Boot from OTP memory (BMODE = 0xB)—This provides a standalone booting method. The boot stream is loaded from on-chip OTP memory. By default, the boot stream is expected to start from OTP page 0x40 and can occupy all public OTP memory up to page 0xDF (2560 bytes). Since the start page is programmable, the maximum size of the boot stream can be extended to 3072 bytes.
- Boot from 16-bit host DMA (BMODE = 0xE)—In this mode, the host DMA port is configured in 16-bit acknowledge mode with little endian data format. Unlike other modes, the host is responsible for interpreting the boot stream. It writes data blocks individually into the host DMA port. Before configuring the DMA settings for each block, the host may either poll the ALLOW_CONFIG bit in HOST_STATUS or wait to be interrupted by the HWAIT signal. When using HWAIT, the host must still check ALLOW_CONFIG at least once before beginning to configure the host DMA port. After completing the configuration, the host is required to poll the READY bit in HOST_STATUS before beginning to transfer data. When the host sends an HIRQ control command, the boot kernel issues a CALL instruction to address 0xFFA0 0000. It is the host's responsibility to ensure valid code has been placed at this address. The routine at address 0xFFA0 0000 can be a simple initialization routine to configure internal resources, such as the SDRAM controller, which then returns using an RTS instruction. The routine may also be the final application, which will never return to the boot kernel.
- Boot from 8-bit host DMA (BMODE = 0xF)—In this mode, the host DMA port is configured in 8-bit interrupt mode with little endian data format. Unlike other modes, the host is responsible for interpreting the boot stream. It writes data blocks individually to the host DMA port. Before configuring the DMA settings for each block, the host may either poll the ALLOW_CONFIG bit in HOST_STATUS or wait to be interrupted by the HWAIT signal. When using HWAIT, the host must still check ALLOW_CONFIG at least once before beginning to configure the host DMA port. The host will receive an interrupt from the HOST_ACK signal every time it is allowed to send the next FIFO depth's worth (sixteen 32-bit words) of information. When the host sends an HIRQ control command, the boot kernel issues a CALL instruction to address 0xFFA0 0000. It is the host's responsibility to ensure valid code has been placed at this address. The routine at address 0xFFA0 0000 can be a simple initialization routine to configure internal resources, such as the SDRAM controller, which then returns using an RTS instruction. The routine may also be the final application, which will never return to the boot kernel.

For each of the boot modes, a 16-byte header is first read from an external memory device. The header specifies the number of bytes to be transferred and the memory destination address. Multiple memory blocks may be loaded by any boot sequence. Once all blocks are loaded, program execution commences from the address stored in the EVT1 register.

Prior to booting, the pre-boot routine interrogates the OTP memory. Individual boot modes can be customized or disabled based on OTP programming. External hardware, especially booting hosts, may monitor the HWAIT signal to determine when the pre-boot has finished and the boot kernel starts the boot process. However, the HWAIT signal does not toggle in NAND boot mode. By programming OTP memory, the user can instruct the preboot routine to also customize the PLL, voltage regulator, DDR controller, and/or asynchronous memory interface controller.

The boot kernel differentiates between a regular hardware reset and a wakeup-from-hibernate event to speed up booting in the later case. Bits 6-4 in the system reset configuration (SYSCR) register can be used to bypass the pre-boot routine and/or boot kernel in case of a software reset. They can also be used to simulate a wakeup-from-hibernate boot in the software reset case.

The boot process can be further customized by "initialization code." This is a piece of code that is loaded and executed prior to the regular application boot. Typically, this is used to configure the DDR controller or to speed up booting by managing PLL, clock frequencies, wait states, and/or serial bit rates.

The boot ROM also features C-callable function entries that can be called by the user application at run time. This enables second-stage boot or booting management schemes to be implemented with ease.

## INSTRUCTION SET DESCRIPTION

The Blackfin processor family assembly language instruction set employs an algebraic syntax designed for ease of coding and readability. The instructions have been specifically tuned to provide a flexible, densely encoded instruction set that compiles to a very small final memory size. The instruction set also provides fully featured multifunction instructions that allow the programmer to use many of the processor core resources in a single instruction. Coupled with many features more often seen on microcontrollers, this instruction set is very efficient when compiling C and C++ source code. In addition, the architecture supports both user (algorithm/application code) and supervisor (O/S kernel, device drivers, debuggers, ISRs) modes of operation, allowing multiple levels of access to core processor resources.

The assembly language, which takes advantage of the processor's unique architecture, offers the following advantages:

- Seamlessly integrated DSP/MCU features are optimized for both 8-bit and 16-bit operations.
- A multi-issue load/store modified-Harvard architecture, which supports two 16-bit MAC or four 8-bit ALU + two load/store + two pointer updates per cycle.
- All registers, I/O, and memory are mapped into a unified 4G byte memory space, providing a simplified programming model.
- Microcontroller features, such as arbitrary bit and bit-field manipulation, insertion, and extraction; integer operations on 8-, 16-, and 32-bit data-types; and separate user and supervisor stack pointers.
- Code density enhancements, which include intermixing of 16- and 32-bit instructions (no mode switching, no code segregation). Frequently used instructions are encoded in 16 bits.

## DEVELOPMENT TOOLS

The ADSP-BF54x Blackfin processors are supported with a complete set of CROSSCORE® software and hardware development tools, including Analog Devices emulators and VisualDSP++® development environment. The same emulator hardware that supports other Blackfin processors also fully emulates the ADSP-BF54x Blackfin processors.

### *EZ-KIT Lite Evaluation Board*

For evaluation of ADSP-BF54x Blackfin processors, use the ADSP-BF548 EZ-KIT Lite® board available from Analog Devices. Order part number ADZS-BF548-EZLITE. The board comes with on-chip emulation capabilities and is equipped to enable software development. Multiple daughter cards are available.

## DESIGNING AN EMULATOR-COMPATIBLE PROCESSOR BOARD

The Analog Devices family of emulators are tools that every system developer needs to test and debug hardware and software systems. Analog Devices has supplied an IEEE 1149.1 JTAG test access port (TAP) on each JTAG processor. The emulator uses the TAP to access the internal features of the processor, allowing the developer to load code, set breakpoints, observe variables, observe memory, and examine registers. The processor must be halted to send data and commands, but once an operation has been completed by the emulator, the processor is set running at full speed with no impact on system timing.

To use these emulators, the target board must include a header that connects the processor's JTAG port to the emulator.

For details on target board design issues including mechanical layout, single processor connections, multiprocessor scan chains, signal buffering, signal termination, and emulator pod logic, see *Analog Devices JTAG Emulation Technical Reference (EE-68)* on the Analog Devices web site under www.analog.com/ee-notes. This document is updated regularly to keep pace with improvements to emulator support.

## MXVR BOARD LAYOUT GUIDELINES

The MXVR Loop Filter RC network is connected between the MLF_P and MLF_M pins in the following manner:

Capacitors:

- C1: 0.047 µF (PPS type, 2% tolerance recommended)
- C2: 330 pF (PPS type, 2% tolerance recommended)

Resistor:

- R1: 330 Ω (1% tolerance)

The RC network should be located physically close to the MLF_P and MLF_M pins on the board.

The RC network should be shielded using $GND_{MP}$ traces.

Avoid routing other switching signals near the RC network to avoid crosstalk.

MXI driven with external clock oscillator IC:

- MXI should be driven with the clock output of a clock oscillator IC running at a frequency of 49.152 MHz or 45.1584 MHz.
- MXO should be left unconnected.
- Avoid routing other switching signals near the oscillator and clock output trace to avoid crosstalk. When not possible, shield traces with ground.

MXI/MXO with external crystal:

- The crystal must be a fundamental mode crystal running at a frequency of 49.152 MHz or 45.1584 MHz.
- The crystal and load capacitors should be placed physically close to the MXI and MXO pins on the board.
- Board trace capacitance on each lead should not be more than 3 pF.
- Trace capacitance plus load capacitance should equal the load capacitance specification for the crystal.
- Avoid routing other switching signals near the crystal and components to avoid crosstalk. When not possible, shield traces and components with ground.

$V_{DDMP}$/$GND_{MP}$—MXVR PLL power domain:

- Route $V_{DDMP}$ and $GND_{MP}$ with wide traces or as isolated power planes.
- Drive $V_{DDMP}$ to same level as $V_{DDINT}$.
- Place a ferrite bead between the $V_{DDINT}$ power plane and the $V_{DDMP}$ pin for noise isolation.
- Locally bypass $V_{DDMP}$ with 0.1 µF and 0.01 µF decoupling capacitors to $GND_{MP}$.
- Avoid routing switching signals near to $V_{DDMP}$ and $GND_{MP}$ traces to avoid crosstalk.

Fiber optic transceiver (FOT) connections:

- Keep the traces between the ADSP-BF549 processor and the FOT as short as possible.
- The receive data trace connecting the FOT receive data output pin to the ADSP-BF549 PH6/MRX input pin should have a 0 Ω series termination resistor placed close to the FOT receive data output pin. Typically, the edge rate of the FOT receive data signal driven by the FOT is very slow, and further degradation of the edge rate is not desirable.
- The transmit data trace connecting the ADSP-BF549 PH5/MTX output pin to the FOT transmit data input pin should have a 27 Ω series termination resistor placed close to the ADSP-BF549 PH5/MTX pin.
- The receive data trace and the transmit data trace between the ADSP-BF549 processor and the FOT should not be routed close to each other in parallel over long distances to avoid crosstalk.

## RELATED DOCUMENTS

The following publications that describe the ADSP-BF54x Blackfin processors (and related processors) can be ordered from any Analog Devices sales office or accessed electronically on www.analog.com:

- *ADSP-BF54x Blackfin Processor Hardware Reference, Volume 1 and Volume 2*
- *Blackfin Processor Programming Reference*
- *ADSP-BF542/BF544/BF547/BF548/BF549 Blackfin Anomaly List*

## LOCKBOX SECURE TECHNOLOGY DISCLAIMER

Analog Devices products containing Lockbox Secure Technology are warranted by Analog Devices as detailed in the Analog Devices Standard Terms and Conditions of Sale. To our knowledge, the Lockbox secure technology, when used in accordance with the data sheet and hardware reference manual specifications, provides a secure method of implementing code and data safeguards. However, Analog Devices does not guarantee that this technology provides absolute security. ACCORDINGLY, ANALOG DEVICES HEREBY DISCLAIMS ANY AND ALL EXPRESS AND IMPLIED WARRANTIES THAT THE LOCKBOX SECURE TECHNOLOGY CANNOT BE BREACHED, COMPROMISED, OR OTHERWISE CIRCUMVENTED AND IN NO EVENT SHALL ANALOG DEVICES BE LIABLE FOR ANY LOSS, DAMAGE, DESTRUCTION, OR RELEASE OF DATA, INFORMATION, PHYSICAL PROPERTY, OR INTELLECTUAL PROPERTY.

# PIN DESCRIPTIONS

ADSP-BF54x Blackfin processors' pin multiplexing scheme is listed in Table 11 and the pin definitions are listed in Table 12.

**Table 11. Pin Multiplexing**

| Primary Pin Function (Number of Pins)[1, 2] | First Peripheral Function | Second Peripheral Function | Third Peripheral Function | Fourth Peripheral Function | Interrupt Capability |
|---|---|---|---|---|---|
| **Port A** | | | | | |
| GPIO (16 pins) | SPORT2 (8 pins) | TMR4 (1 pin) | TACI7 (1 pin) | | Interrupts (16 pins) |
| | | TMR5 (1 pin) | TACLK7–0 (8 pins) | | |
| | SPORT3 (8 pins) | TMR6 (1 pin) | | | |
| | | TMR7 (1 pin) | | | |
| **Port B** | | | | | |
| GPIO (15 pins) | TWI1 (2 pins)<br>UART2 or 3 CTL (2 pins)<br>UART2 (2 pins)<br>UART3 (2 pins) | | TACI2-3 (2 pins) | | Interrupts (15 pins) |
| | $\overline{\text{SPI2 SEL1-3}}$ (3 pins) | TMR0–2 (3 pins) | | | |
| | SPI2 (3 pins) | TMR3 (1 pin) | HWAIT (1 pin) | | |
| **Port C** | | | | | |
| GPIO (16 pins) | SPORT0 (8 pins) | MXVR MMCLK, MBCLK (2 pins) | | | Interrupts (8 pins)[3] |
| | SDH (6 pins) | | | | Interrupts (8 pins) |
| **Port D** | | | | | |
| GPIO (16 pins) | PPI1 D0–15 (16 pins) | Host D0–15 (16 pins) | SPORT1 (8 pins) | PPI0 D18– 23 (6 pins) | Interrupts (8 pins) |
| | | | PPI2 D0–7 (8 pins) | Keypad<br>Row 0–3<br>Col 0–3 (8 pins) | Interrupts (8 pins) |
| **Port E** | | | | | |
| GPIO (16 pins) | SPI0 (7 pins) | Keypad<br>Row 4–6<br>Col 4–7 (7 pins) | TACI0 (1 pin) | | Interrupts (8 pins) |
| | UART0 TX (1 pin) | Keypad R7 (1 pin) | | | |
| | UART0 RX (1 pin)<br>UART0 or 1 CTL (2 pins)<br>PPI1 CLK,FS (3 pins)<br>TWI0 (2 pins) | | | | Interrupts (8 pins) |
| **Port F** | | | | | |
| GPIO (16 pins) | PPI0 D0–15 (16 pins) | ATAPI D0-15A | | | Interrupts (8 pins)<br>Interrupts (8 pins) |
| **Port G** | | | | | |
| GPIO (16 pins) | PPI0 CLK,FS (3 pins)<br>DATA 16–17 (2 pins) | TMRCLK (1 pin) | | | Interrupts (8 pins) |
| | | ATAPI A0-2A | | | |
| | $\overline{\text{SPI1 SEL1–3}}$ (3 pins) | Host CTL (3 pins) | PPI2 CLK,FS (3 pins) | CZM (1 pin) | |
| | SPI1 (4 pins) | MXVR $\overline{\text{MTXON}}$ (1 pin) | TACI4-5 (2 pins) | | Interrupts (8 pins) |
| | CAN0 (2 pins) | | | | |
| | CAN1 (2 pins) | | | | |

**Table 11. Pin Multiplexing (Continued)**

| Primary Pin Function (Number of Pins)[1, 2] | First Peripheral Function | Second Peripheral Function | Third Peripheral Function | Fourth Peripheral Function | Interrupt Capability |
|---|---|---|---|---|---|
| **Port H** | | | | | |
| GPIO (14 pins) | UART1 (2 pins) | PPI0-1_FS3 (2 pins) | TACI1 (1 pin) | | Interrupts (8 pins) |
| | $\overline{\text{ATAPI_RESET}}$ (1 pin) | TMR8 (1 pin) | PPI2_FS3 (1 pin) | | |
| | HOST_ADDR (1 pin) | TMR9 (1 pin) | Counter Down/Gate (1 pin) | | |
| | HOST_ACK (1 pin) | TMR10 (1 pin) | Counter Up/Dir (1 pin) | | |
| | MXVR MRX, MTX, $\overline{\text{MRXON}}$/$\overline{\text{GPW}}$ (3 pins)[4] | | DMAR 0–1 (2 pins) | TACI8–10 (3 pins) TACLK8–10 (3 pins) HWAITA | |
| | | AMC Addr 4-9 (6 pins) | | | Interrupts (6 pins) |
| **Port I** | | | | | |
| GPIO (16 pins) | Async Addr10–25 (16 pins) | | | | Interrupts (8 pins) |
| | | | | | Interrupts (8 pins) |
| **Port J** | | | | | |
| GPIO (14 pins) | Async CTL and MISC | | | | Interrupts (8 pins) |
| | | | | | Interrupts (6 pins) |

[1] Port connections may be inputs or outputs after power up depending on the model and boot mode chosen.
[2] All port connections always power up as inputs for some period of time and require resistive termination to a safe condition if used as outputs in the system.
[3] A total of 32 interrupts at once are available from ports C through J, configurable in byte-wide blocks.
[4] $\overline{\text{GPW}}$ functionality available when MXVR is not present or unused.

ADSP-BF54x processor pin definitions are listed in Table 12. To see the pin multiplexing scheme, see Table 11.

**Table 12. Pin Descriptions**

| Pin Name | I/O[1] | Function (First/Second/Third/Fourth) | Driver Type[2] |
|---|---|---|---|
| **Port A:** GPIO/*SPORT2–3/TMR4–7* | | | |
| PA0/*TFS2* | I/O | GPIO/*SPORT2 Transmit Frame Sync* | C |
| PA1/*DT2SEC/TMR4* | I/O | GPIO/*SPORT2 Transmit Data Secondary/Timer 4* | C |
| PA2/*DT2PRI* | I/O | GPIO/*SPORT2 Transmit Data Primary* | C |
| PA3/*TSCLK2* | I/O | GPIO/*SPORT2 Transmit Serial Clock* | A |
| PA4/*RFS2* | I/O | GPIO/*SPORT2 Receive Frame Sync* | C |
| PA5/*DR2SEC/TMR5* | I/O | GPIO/*SPORT2 Receive Data Secondary/Timer 5* | C |
| PA6/*DR2PRI* | I/O | GPIO/*SPORT2 Receive Data Primary* | C |
| PA7/*RSCLK2/TACLK0* | I/O | GPIO/*SPORT2 Receive Serial Clock/Alternate Input Clock 0* | A |
| PA8/*TFS3/TACLK1* | I/O | GPIO/*SPORT3 Transmit Frame Sync/Alternate Input Clock 1* | C |
| PA9/*DT3SEC/TMR6* | I/O | GPIO/*SPORT3 Transmit Data Secondary/Timer 6* | C |
| PA10/*DT3PRI/TACLK2* | I/O | GPIO/*SPORT3 Transmit Data Primary/Alternate Input Clock 2* | C |
| PA11/*TSCLK3/TACLK3* | I/O | GPIO/*SPORT3 Transmit Serial Clock/Alternate Input Clock 3* | A |
| PA12/*RFS3/TACLK4* | I/O | GPIO/*SPORT3 Receive Frame Sync/Alternate Input Clock 4* | C |
| PA13/*DR3SEC/TMR7/TACLK5* | I/O | GPIO/*SPORT3 Receive Data Secondary/Timer 7/Alternate Input Clock 5* | C |
| PA14/*DR3PRI/TACLK6* | I/O | GPIO/*SPORT3 Receive Data Primary/Alternate Input Clock 6* | C |
| PA15/*RSCLK3/TACLK7 and TACI7* | I/O | GPIO/*SPORT3 Receive Serial Clock/Alt Input Clock 7 and Alt Capture Input 7* | A |

**Table 12. Pin Descriptions (Continued)**

| Pin Name | I/O[1] | Function (First/Second/Third/Fourth) | Driver Type[2] |
|---|---|---|---|
| **Port B:** GPIO/*TWI1/UART2–3/SPI2/TMR0–3* | | | |
| PB0/*SCL1* | I/O | GPIO/*TWI1 Serial Clock* (Open-drain output: requires a pull-up resistor.) | E |
| PB1/*SDA1* | I/O | GPIO/*TWI1 Serial Data* (Open-drain output: requires a pull-up resistor.) | E |
| PB2/$\overline{UART3RTS}$ | I/O | GPIO/*UART3 Request to Send* | C |
| PB3/$\overline{UART3CTS}$ | I/O | GPIO/*UART3 Clear to Send* | A |
| PB4/*UART2TX* | I/O | GPIO/*UART2 Transmit* | A |
| PB5/*UART2RX/TACI2* | I/O | GPIO/*UART2 Receive/Alternate Capture Input 2* | A |
| PB6/*UART3TX* | I/O | GPIO/*UART3 Transmit* | A |
| PB7/*UART3RX/TACI3* | I/O | GPIO/*UART3 Receive/Alternate Capture Input 3* | A |
| PB8/$\overline{SPI2SS}$/*TMR0* | I/O | GPIO/*SPI2 Slave Select Input/Timer 0* | A |
| PB9/$\overline{SPI2SEL1}$/*TMR1* | I/O | GPIO/*SPI2 Slave Select Enable 1/Timer 1* | A |
| PB10 $\overline{SPI2SEL2}$/*TMR2* | I/O | GPIO/*SPI2 Slave Select Enable 2/Timer 2* | A |
| PB11/$\overline{SPI2SEL3}$/*TMR3/HWAIT* | I/O | GPIO/*SPI2 Slave Select Enable 3/Timer 3/Boot Host Wait* | A |
| PB12/*SPI2SCK* | I/O | GPIO/*SPI2 Clock* | A |
| PB13/*SPI2MOSI* | I/O | GPIO/*SPI2 Master Out Slave In* | C |
| PB14/*SPI2MISO* | I/O | GPIO/*SPI2 Master In Slave Out* | C |
| **Port C:** GPIO/*SPORT0/SD Controller/MXVR (MOST)* | | | |
| PC0/*TFS0* | I/O | GPIO/*SPORT0 Transmit Frame Sync* | C |
| PC1/*DT0SEC/MMCLK* | I/O | GPIO/*SPORT0 Transmit Data Secondary/MXVR Master Clock* | C |
| PC2/*DT0PRI* | I/O | GPIO/*SPORT0 Transmit Data Primary* | C |
| PC3/*TSCLK0* | I/O | GPIO/*SPORT0 Transmit Serial Clock* | A |
| PC4/*RFS0* | I/O | GPIO/*SPORT0 Receive Frame Sync* | C |
| PC5/*DR0SEC/MBCLK* | I/O | GPIO/*SPORT0 Receive Data Secondary/MXVR Bit Clock* | C |
| PC6/*DR0PRI* | I/O | GPIO/*SPORT0 Receive Data Primary* | C |
| PC7/*RSCLK0* | I/O | GPIO/*SPORT0 Receive Serial Clock* | C |
| PC8/*SD_D0* | I/O | GPIO/*SD Data Bus* | A |
| PC9/*SD_D1* | I/O | GPIO/*SD Data Bus* | A |
| PC10/*SD_D2* | I/O | GPIO/*SD Data Bus* | A |
| PC11/*SD_D3* | I/O | GPIO/*SD Data Bus* | A |
| PC12/*SD_CLK* | I/O | GPIO/*SD Clock Output* | A |
| PC13/*SD_CMD* | I/O | GPIO/*SD Command* | A |

**Table 12. Pin Descriptions (Continued)**

| Pin Name | I/O[1] | Function (First/Second/Third/Fourth) | Driver Type[2] |
|---|---|---|---|
| **Port D:** GPIO/*PPI0–2/SPORT 1/Keypad/Host DMA* | | | |
| PD0/*PPI1_D0/HOST_D8/ TFS1/PPI0_D18* | I/O | GPIO/*PPI1 Data/Host DMA/SPORT1 Transmit Frame Sync/PPI0 Data* | C |
| PD1/*PPI1_D1/HOST_D9/ DT1SEC/PPI0_D19* | I/O | GPIO/*PPI1 Data/Host DMA/SPORT1 Transmit Data Secondary/PPI0 Data* | C |
| PD2/*PPI1_D2/HOST_D10/ DT1PRI/PPI0_D20* | I/O | GPIO/*PPI1 Data/Host DMA/SPORT1 Transmit Data Primary/PPI0 Data* | C |
| PD3/*PPI1_D3/HOST_D11/ TSCLK1/PPI0_D21* | I/O | GPIO/*PPI1 Data/Host DMA/SPORT1 Transmit Serial Clock/PPI0 Data* | A |
| PD4/*PPI1_D4/HOST_D12/RFS1/PPI0_D22* | I/O | GPIO/*PPI1 Data/Host DMA/SPORT1 Receive Frame Sync/PPI0 Data* | C |
| PD5/*PPI1_D5/HOST_D13/DR1SEC/PPI0_D23* | I/O | GPIO/*PPI1 Data/Host DMA/SPORT1 Receive Data Secondary/PPI0 Data* | C |
| PD6/*PPI1_D6/HOST_D14/DR1PRI* | I/O | GPIO/*PPI1 Data/Host DMA/SPORT1 Receive Data Primary* | C |
| PD7/*PPI1_D7/HOST_D15/RSCLK1* | I/O | GPIO/*PPI1 Data/Host DMA/SPORT1 Receive Serial Clock* | A |
| PD8/*PPI1_D8/HOST_D0/ PPI2_D0/KEY_ROW0* | I/O | GPIO/*PPI1 Data/Host DMA/PPI2 Data/Keypad Row Input* | A |
| PD9/*PPI1_D9/HOST_D1/PPI2_D1/KEY_ROW1* | I/O | GPIO/*PPI1 Data/Host DMA/PPI2 Data/Keypad Row Input* | A |
| PD10/*PPI1_D10/HOST_D2/PPI2_D2/KEY_ROW2* | I/O | GPIO/*PPI1 Data/Host DMA/PPI2 Data/Keypad Row Input* | A |
| PD11/*PPI1_D11/HOST_D3/PPI2_D3/KEY_ROW3* | I/O | GPIO/*PPI1 Data/Host DMA/PPI2 Data/Keypad Row Input* | A |
| PD12/*PPI1_D12/HOST_D4/PPI2_D4/KEY_COL0* | I/O | GPIO/*PPI1 Data/Host DMA/PPI2 Data/Keypad Column Output* | A |
| PD13/*PPI1_D13/HOST_D5/PPI2_D5/KEY_COL1* | I/O | GPIO/*PPI1 Data/Host DMA/PPI2 Data/Keypad Column Output* | A |
| PD14/*PPI1_D14/HOST_D6/PPI2_D6/KEY_COL2* | I/O | GPIO/*PPI1 Data/Host DMA/PPI2 Data/Keypad Column Output* | A |
| PD15/*PPI1_D15/HOST_D7/PPI2_D7/KEY_COL3* | I/O | GPIO/*PPI1 Data/Host DMA/PPI2 Data/Keypad Column Output* | A |
| **Port E:** GPIO/*SPI0/UART0-1/PPI1/TWI0/Keypad* | | | |
| PE0/*SPI0SCK/KEY_COL7*[3] | I/O | GPIO/*SPI0 Clock/Keypad Column Output* | A |
| PE1/*SPI0MISO/KEY_ROW6*[3] | I/O | GPIO/*SPI0 Master In Slave Out/Keypad Row Input* | C |
| PE2/*SPI0MOSI/KEY_COL6* | I/O | GPIO/*SPI0 Master Out Slave In/Keypad Column Output* | C |
| PE3/$\overline{\textit{SPI0SS}}$/*KEY_ROW5* | I/O | GPIO/*SPI0 Slave Select Input/Keypad Row Input* | A |
| PE4/$\overline{\textit{SPI0SEL1}}$/*KEY_COL*[3] | I/O | GPIO/*SPI0 Slave Select Enable 1/Keypad Column Output* | A |
| PE5/$\overline{\textit{SPI0SEL2}}$/*KEY_ROW4* | I/O | GPIO/*SPI0 Slave Select Enable 2/Keypad Row Input* | A |
| PE6/$\overline{\textit{SPI0SEL3}}$/*KEY_COL4* | I/O | GPIO/*SPI0 Slave Select Enable 3/Keypad Column Output* | A |
| PE7/*UART0TX/KEY_ROW7* | I/O | GPIO/*UART0 Transmit/Keypad Row Input* | A |
| PE8/*UART0RX/TACI0* | I/O | GPIO/*UART0 Receive/Alternate Capture Input 0* | A |
| PE9/$\overline{\textit{UART1RTS}}$ | I/O | GPIO/*UART1 Request to Send* | A |
| PE10/$\overline{\textit{UART1CTS}}$ | I/O | GPIO/*UART1 Clear to Send* | A |
| PE11/*PPI1_CLK* | I/O | GPIO/*PPI1 Clock* | A |
| PE12/*PPI1_FS1* | I/O | GPIO/*PPI1 Frame Sync 1* | A |
| PE13/*PPI1_FS2* | I/O | GPIO/*PPI1 Frame Sync 2* | A |
| PE14/*SCL0* | I/O | GPIO/*TWI0 Serial Clock* (Open-drain output: requires a pull-up resistor.) | E |
| PE15/*SDA0* | I/O | GPIO/*TWI0 Serial Data* (Open-drain output: requires a pull-up resistor.) | E |

**Table 12. Pin Descriptions (Continued)**

| Pin Name | I/O[1] | Function (First / Second/Third/ Fourth) | Driver Type[2] |
|---|---|---|---|
| **Port F:** *GPIO/PPI0/Alternate ATAPI Data* | | | |
| PF0 / *PPI0_D0/ATAPI_D0A* | I/O | GPIO / *PPI0 Data/Alternate ATAPI Data* | A |
| PF1 / *PPI0_D1/ATAPI_D1A* | I/O | GPIO / *PPI0 Data/Alternate ATAPI Data* | A |
| PF2 / *PPI0_D2/ATAPI_D2A* | I/O | GPIO / *PPI0 Data/Alternate ATAPI Data* | A |
| PF3 / *PPI0_D3/ATAPI_D3A* | I/O | GPIO / *PPI0 Data/Alternate ATAPI Data* | A |
| PF4 / *PPI0_D4/ATAPI_D4A* | I/O | GPIO / *PPI0 Data/Alternate ATAPI Data* | A |
| PF5 / *PPI0_D5/ATAPI_D5A* | I/O | GPIO / *PPI0 Data/Alternate ATAPI Data* | A |
| PF6 / *PPI0_D6/ATAPI_D6A* | I/O | GPIO / *PPI0 Data/Alternate ATAPI Data* | A |
| PF7 / *PPI0_D7/ATAPI_D7A* | I/O | GPIO / *PPI0 Data/Alternate ATAPI Data* | A |
| PF8 / *PPI0_D8/ATAPI_D8A* | I/O | GPIO / *PPI0 Data/Alternate ATAPI Data* | A |
| PF9 / *PPI0_D9/ATAPI_D9A* | I/O | GPIO / *PPI0 Data/Alternate ATAPI Data* | A |
| PF10 / *PPI0_D10/ATAPI_D10A* | I/O | GPIO / *PPI0 Data/Alternate ATAPI Data* | A |
| PF11 / *PPI0_D11/ATAPI_D11A* | I/O | GPIO / *PPI0 Data/Alternate ATAPI Data* | A |
| PF12 / *PPI0_D12/ATAPI_D12A* | I/O | GPIO / *PPI0 Data/Alternate ATAPI Data* | A |
| PF13 / *PPI0_D13/ATAPI_D13A* | I/O | GPIO / *PPI0 Data/Alternate ATAPI Data* | A |
| PF14 / *PPI0_D14/ATAPI_D14A* | I/O | GPIO / *PPI0 Data/Alternate ATAPI Data* | A |
| PF15 / *PPI0_D15/ATAPI_D15A* | I/O | GPIO / *PPI0 Data/Alternate ATAPI Data* | A |
| **Port G:** GPIO / *PPI0 / SPI1 / PPI2/Up-Down Counter/CAN0–1/ Host DMA/MXVR (MOST)/ATAPI* | | | |
| PG0 / *PPI0_CLK / TMRCLK* | I/O | GPIO / *PPI0 Clock / External Timer Reference* | A |
| PG1 / *PPI0_FS1* | I/O | GPIO / *PPI0 Frame Sync 1* | A |
| PG2 / *PPI0_FS2/ATAPI_A0A* | I/O | GPIO / *PPI0 Frame Sync 2/Alternate ATAPI Address* | A |
| PG3 / *PPI0_D16/ATAPI_A1A* | I/O | GPIO / *PPI0 Data/Alternate ATAPI Address* | A |
| PG4 / *PPI0_D17/ATAPI_A2A* | I/O | GPIO / *PPI0 Data/Alternate ATAPI Address* | A |
| PG5 / $\overline{\textit{SPI1SEL1}}$ / $\overline{\textit{HOST_CE}}$ / *PPI2_FS2 / CZM* | I/O | GPIO / *SPI1 Slave Select / Host DMA Chip Enable / PPI2 Frame Sync 2 / Counter Zero Marker* | A |
| PG6 / $\overline{\textit{SPI1SEL2}}$ / $\overline{\textit{HOST_RD}}$ / *PPI2_FS1* | I/O | GPIO / *SPI1 Slave Select / Host DMA Read / PPI2 Frame Sync* 1 | A |
| PG7 / $\overline{\textit{SPI1SEL3}}$ / $\overline{\textit{HOST_WR}}$ / *PPI2_CLK* | I/O | GPIO / *SPI1 Slave Select / Host DMA Write / PPI2 Clock* | A |
| PG8 / *SPI1SCK* | I/O | GPIO / *SPI1 Clock* | C |
| PG9 / *SPI1MISO* | I/O | GPIO / *SPI1 Master In Slave Out* | C |
| PG10 / *SPI1MOSI* | I/O | GPIO / *SPI1 Master Out Slave In* | C |
| PG11 / $\overline{\textit{SPI1SS}}$ / $\overline{\textit{MTXON}}$ | I/O | GPIO / *SPI1 Slave Select Input / MXVR Transmit Phy On* | A |
| PG12 / *CAN0TX* | I/O | GPIO / *CAN0 Transmit* | A |
| PG13 / *CAN0RX / TACI4* | I/O | GPIO / *CAN0 Receive / Alternate Capture Input 4* | A |
| PG14 / *CAN1TX* | I/O | GPIO / *CAN1 Transmit* | A |
| PG15 / *CAN1RX / TACI5* | I/O | GPIO / *CAN1 Receive / Alternate Capture Input 5* | A |

**Table 12. Pin Descriptions (Continued)**

| Pin Name | I/O[1] | Function (First/Second/Third/Fourth) | Driver Type[2] |
|---|---|---|---|
| **Port H:** GPIO/*AMC*/*EXTDMA*/*UART1*/*PPI0–2*/*ATAPI*/*Up-Down Counter*/*TMR8-10*/*Host DMA*/*MXVR (MOST)* | | | |
| PH0/*UART1TX*/*PPI1_FS3_DEN* | I/O | GPIO/*UART1 Transmit*/*PPI1 Frame Sync* 3 | A |
| PH1/*UART1RX*/*PPI0_FS3_DEN*/*TACI1* | I/O | GPIO/*UART 1 Receive*/ *PPI0 Frame Sync* 3/*Alternate Capture Input 1* | A |
| PH2/$\overline{ATAPI_RESET}$/*TMR8*/*PPI2_FS3_DEN* | I/O | GPIO/*ATAPI Interface Hard Reset Signal*/*Timer 8*/*PPI2 Frame Sync* 3 | A |
| PH3/*HOST_ADDR*/*TMR9*/*CDG* | I/O | GPIO/*HOST Address*/*Timer 9*/*Count Down and Gate* | A |
| PH4/*HOST_ACK*/*TMR10*/*CUD* | I/O | GPIO/*HOST Acknowledge*/*Timer 10*/*Count Up and Direction* | A |
| PH5/*MTX*/*DMAR0*/*TACI8 and TACLK8* | I/O | GPIO/*MXVR Transmit Data*/*Ext. DMA Request*/*Alt Capt. In. 8*/*Alt In. Clk 8* | C |
| PH6/*MRX*/*DMAR1*/*TACI9 and TACLK9* | I/O | GPIO/*MXVR Receive Data*/*Ext. DMA Request*/*Alt Capt. In. 9*/*Alt In. Clk 9* | A |
| PH7/$\overline{MRXON}$/$\overline{GPW}$/*TACI10 and TACLK10*/*HWAITA*[4,5] | I/O | GPIO/*MXVR Receive Phy On*/*Alt Capt. In. 10*/*Alt In. Clk 10*/*Alternate Boot Host Wait* | A |
| PH8/*A4*[6] | I/O | GPIO/*Address Bus for Async Access* | A |
| PH9/*A5*[6] | I/O | GPIO/*Address Bus for Async Access* | A |
| PH10/*A6*[6] | I/O | GPIO/*Address Bus for Async Access* | A |
| PH11/*A7*[6] | I/O | GPIO/*Address Bus for Async Access* | A |
| PH12/*A8*[6] | I/O | GPIO/*Address Bus for Async Access* | A |
| PH13/*A9*[6] | I/O | GPIO/*Address Bus for Async Access* | A |
| **Port I:** GPIO/*AMC* | | | |
| PI0/*A10*[6] | I/O | GPIO/*Address Bus for Async Access* | A |
| PI1/*A11*[6] | I/O | GPIO/*Address Bus for Async Access* | A |
| PI2/*A12*[6] | I/O | GPIO/*Address Bus for Async Access* | A |
| PI3/*A13*[6] | I/O | GPIO/*Address Bus for Async Access* | A |
| PI4/*A14*[6] | I/O | GPIO/*Address Bus for Async Access* | A |
| PI5/*A15*[6] | I/O | GPIO/*Address Bus for Async Access* | A |
| PI6/*A16*[6] | I/O | GPIO/*Address Bus for Async Access* | A |
| PI7/*A17*[6] | I/O | GPIO/*Address Bus for Async Access* | A |
| PI8/*A18*[6] | I/O | GPIO/*Address Bus for Async Access* | A |
| PI9/*A19*[6] | I/O | GPIO/*Address Bus for Async Access* | A |
| PI10/*A20*[6] | I/O | GPIO/*Address Bus for Async Access* | A |
| PI11/*A21*[6] | I/O | GPIO/*Address Bus for Async Access* | A |
| PI12/*A22*[6] | I/O | GPIO/*Address Bus for Async Access* | A |
| PI13/*A23*[6] | I/O | GPIO/*Address Bus for Async Access* | A |
| PI14/*A24*[6] | I/O | GPIO/*Address Bus for Async Access* | A |
| PI15/*A25/NR_CLK*[6] | I/O | GPIO/*Address Bus for Async Access/ NOR clock* | A |

**Table 12. Pin Descriptions (Continued)**

| Pin Name | I/O[1] | Function (First/Second/Third/Fourth) | Driver Type[2] |
|---|---|---|---|
| **Port J:** GPIO/*AMC/ATAPI* | | | |
| PJ0/*ARDY*/$\overline{WAIT}$ | I/O | GPIO/ *Async Ready/NOR Wait* | A |
| PJ1/$\overline{ND_CE}$[7] | I/O | GPIO/*NAND Chip Enable* | A |
| PJ2/$\overline{ND_RB}$ | I/O | GPIO/*NAND Ready Busy* | A |
| PJ3/$\overline{ATAPI_DIOR}$ | I/O | GPIO/*ATAPI Read* | A |
| PJ4/$\overline{ATAPI_DIOW}$ | I/O | GPIO/*ATAPI Write* | A |
| PJ5/$\overline{ATAPI_CS0}$ | I/O | GPIO/*ATAPI Chip Select/Command Block* | A |
| PJ6/$\overline{ATAPI_CS1}$ | I/O | GPIO/*ATAPI Chip Select* | A |
| PJ7/$\overline{ATAPI_DMACK}$ | I/O | GPIO/*ATAPI DMA Acknowledge* | A |
| PJ8/$\overline{ATAPI_DMARQ}$ | I/O | GPIO/*ATAPI DMA Request* | A |
| PJ9/*ATAPI_INTRQ* | I/O | GPIO/*Interrupt Request from the Device* | A |
| PJ10/$\overline{ATAPI_IORDY}$ | I/O | GPIO/*ATAPI Ready Handshake* | A |
| PJ11/$\overline{BR}$[8] | I/O | GPIO/*Bus Request* | A |
| PJ12/$\overline{BG}$[6] | I/O | GPIO/*Bus Grant* | A |
| PJ13/$\overline{BGH}$[6] | I/O | GPIO/*Bus Grant Hang* | A |
| **DDR Memory Interface** | | | |
| DA0–12 | O | DDR Address Bus | D |
| DBA0–1 | O | DDR Bank Active Strobe | D |
| DQ0–15 | I/O | DDR Data Bus | D |
| DQS0–1 | I/O | DDR Data Strobe | D |
| DQM0–1 | O | DDR Data Mask for Reads and Writes | D |
| DCLK0–1 | O | DDR Output Clock | D |
| $\overline{DCLK0–1}$ | O | DDR Complementary Output Clock | D |
| $\overline{DCS0–1}$ | O | DDR Chip Selects | D |
| DCLKE[9] | O | DDR Clock Enable | D |
| $\overline{DRAS}$ | O | DDR Row Address Strobe | D |
| $\overline{DCAS}$ | O | DDR Column Address Strobe | D |
| $\overline{DWE}$ | O | DDR Write Enable | D |
| DDR_VREF | I | DDR Voltage Reference | |
| DDR_VSSR | I | DDR Voltage Reference Shield (Must be connected to GND.) | |
| **Asynchronous Memory Interface** | | | |
| A1-3 | O | Address Bus for Async and ATAPI Addresses | A |
| D0-15/ND_D0-15/ATAPI_D0-15 | I/O | Data Bus for Async, NAND and ATAPI Accesses | A |
| $\overline{AMS0–3}$ | O | Bank Selects (Pull high with a resistor when used as chip select.) | A |
| $\overline{ABE0}$/*ND_CLE* | O | Byte Enables: Data Masks for Asynchronous Access/*NAND Command Latch Enable* | A |
| $\overline{ABE1}$/*ND_ALE* | O | Byte Enables: Data Masks for Asynchronous Access/*NAND Address Latch Enable* | A |
| $\overline{AOE}$/*NR_ADV* | O | Output Enable/*NOR Address Data Valid* | A |
| $\overline{ARE}$ | O | Read Enable/*NOR Output Enable* | A |
| $\overline{AWE}$ | O | Write Enable | A |
| **ATAPI Controller Pins** | | | |
| $\overline{ATAPI_PDIAG}$ | I | Determines if an 80-pin cable is connected to the host. (Pull high or low when unused.) | |

**Table 12. Pin Descriptions (Continued)**

| Pin Name | I/O[1] | Function (First/Second/Third/Fourth) | Driver Type[2] |
|---|---|---|---|
| **High Speed USB OTG Pins** | | | |
| USB_DP | I/O | USB D+ Pin (Pull low when unused.) | |
| USB_DM | I/O | USB D- Pin (Pull low when unused.) | |
| USB_XI | C | Clock XTAL Input (Pull high or low when unused.) | |
| USB_XO | C | Clock XTAL Output (Leave unconnected when unused.) | |
| USB_ID[10] | I | USB OTG ID Pin (Pull high when unused.) | |
| USB_VBUS[11] | I/O | USB VBUS Pin (Pull high or low when unused.) | |
| USB_VREF | A | USB Voltage Reference (Connect to GND through a 0.1 mF capacitor or leave unconnected when not used.) | |
| USB_RSET | A | USB Resistance Set (Connect to GND through an unpopulated resistor pad.) | |
| **MXVR (MOST) Interface** | | | |
| MFS | O | MXVR Frame Sync (Leave unconnected when unused.) | C |
| MLF_P | A | MXVR Loop Filter Plus (Leave unconnected when unused.) | |
| MLF_M | A | MXVR Loop Filter Minus (Leave unconnected when unused.) | |
| MXI | C | MXVR Crystal Input (Pull high or low when unused.) | |
| MXO | C | MXVR Crystal Output (Pull high or low when unused.) | |
| **Mode Control Pins** | | | |
| BMODE0–3 | I | Boot Mode Strap 0–3 | |
| **JTAG Port Pins** | | | |
| TDI | I | JTAG Serial Data In | |
| TDO | O | JTAG Serial Data Out | C |
| $\overline{\text{TRST}}$ | I | JTAG Reset (Pull low when unused.) | |
| TMS | I | JTAG Mode Select | |
| TCK | I | JTAG Clock | |
| $\overline{\text{EMU}}$ | O | Emulation Output | C |
| **Voltage Regulator** | | | |
| $VR_{OUT}0$, $VR_{OUT}1$ | O | External FET/BJT Drivers (Always connect together to reduce signal impedance.) | |
| **Real Time Clock** | | | |
| RTXO | C | RTC Crystal Output (Leave unconnected when unused.) | |
| RTXI | C | RTC Crystal Input (Pull high or low when unused.) | |
| **Clock (PLL) Pins** | | | |
| CLKIN | C | Clock/Crystal Input | |
| CLKOUT | O | Clock Output | B |
| XTAL | C | Crystal Output | |
| CLKBUF | O | Buffered Oscillator Output | C |
| EXT_WAKE | O | External Wakeup from Hibernate Output | A |
| $\overline{\text{RESET}}$ | I | Reset | |
| $\overline{\text{NMI}}$ | I | Non-maskable Interrupt (Pull high when unused.) | |
| **Supplies** | | | |
| $V_{DDINT}$ | P | Internal Power Supply | |
| $V_{DDEXT}$[12] | P | External Power Supply | |
| $V_{DDDDR}$[12] | P | External DDR Power Supply | |
| $V_{DDUSB}$[12] | P | External USB Power Supply | |
| $V_{DDRTC}$[12] | P | RTC Clock Supply | |

**Table 12. Pin Descriptions (Continued)**

| Pin Name | I/O[1] | Function (First/Second/Third/Fourth) | Driver Type[2] |
|---|---|---|---|
| $V_{DDVR}$[13] | P | Internal Voltage Regulator Power Supply (Connect to $V_{DDEXT}$ when unused.) | |
| GND | G | Ground | |
| $V_{DDMP}$[12] | P | MXVR PLL Power Supply. (Must be driven to same level as $V_{DDINT}$. Connect to $V_{DDINT}$ when unused or when MXVR is not present.) | |
| $GND_{MP}$[12] | G | MXVR PLL Ground (Connect to GND when unused or when MXVR is not present.) | |

[1] I = Input, O = Output, P =Power, G = Ground, C = Crystal, A = Analog.
[2] Refer to Table 61 on Page 86 through Table 70 on Page 87 for driver types.
[3] To use the SPI memory boot, SPI0SCK should have a pulldown, SPI0MISO should have a pullup, and $\overline{SPI0SEL1}$ is used as the $\overline{CS}$ with a pullup.
[4] HWAIT/HWAITA should be pulled high or low to configure polarity. See Booting Modes on Page 19.
[5] $\overline{GPW}$ functionality is available when MXVR is not present or unused.
[6] This pin should not be used as GPIO if booting in mode 1.
[7] This pin should always be enabled as $\overline{ND_CE}$ in software and pulled high with a resistor when using NAND flash.
[8] This pin should always be enabled as $\overline{BR}$ in software and pulled high to enable asynchronous access.
[9] This pin must be pulled low through a 10kOhm resistor if self-refresh mode is desired during hibernate state or deep-sleep mode.
[10] If the USB is used in device mode only, the USB_ID pin should be either pulled high or left unconnected.
[11] This pin is an output only during initialization of USB OTG session request pulses. Therefore, host mode or OTG type A mode requires that an external voltage source of 5 V, at 8 mA or more per the OTG specification, be applied to this pin. Other OTG modes require that this external voltage be disabled.
[12] To ensure proper operation, the power pins should be driven to their specified level even if the associated peripheral is not used in the application.
[13] This pin must always be connected. If the internal voltage regulator is not being used, this pin may be connected to $V_{DDEXT}$. Otherwise it should be powered according to the VDDVR specification. For automotive grade models, the internal voltage regulator must not be used and this pin must be tied to $V_{DDEXT}$.

# SPECIFICATIONS

Component specifications are subject to change without notice.

## OPERATING CONDITIONS

| Parameter | | Conditions | Min | Nominal | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{DDINT}$[1, 2] | Internal Supply Voltage | Nonautomotive grade models | 0.9 | | 1.43 | V |
| | Internal Supply Voltage | Automotive grade models | 1.0 | | 1.38 | V |
| | Internal Supply Voltage | Mobile DDR SDRAM models | 1.14 | | 1.31 | V |
| $V_{DDEXT}$[3] | External Supply Voltage | Nonautomotive 3.3 V I/O | 2.7 | 3.3 | 3.6 | V |
| | External Supply Voltage | Nonautomotive 2.5 V I/O | 2.25 | 2.5 | 2.75 | V |
| | External Supply Voltage | Automotive grade models | 2.7 | 3.3 | 3.6 | V |
| $V_{DDUSB}$ | USB External Supply Voltage | | 3.0 | 3.3 | 3.6 | V |
| $V_{DDMP}$ | MXVR PLL Supply Voltage | Nonautomotive grade models | 0.9 | | 1.43 | V |
| | MXVR PLL Supply Voltage | Automotive grade models | 1.0 | | 1.38 | V |
| $V_{DDRTC}$ | Real Time Clock Supply Voltage | Nonautomotive grade models | 2.25 | | 3.6 | V |
| | Real Time Clock Supply Voltage | Automotive grade models | 2.7 | 3.3 | 3.6 | V |
| $V_{DDDDR}$ | DDR Memory Supply Voltage | DDR SDRAM models | 2.5 | 2.6 | 2.7 | V |
| | DDR Memory Supply Voltage | Mobile DDR SDRAM models | 1.8 | 1.875 | 1.95 | V |
| $V_{DDVR}$[4] | Internal Voltage Regulator Supply Voltage | | 2.7 | 3.3 | 3.6 | V |
| $V_{IH}$ | High Level Input Voltage[5, 6] | $V_{DDEXT}$ = maximum | 2.0 | | 3.6 | V |
| $V_{IHDDR}$ | High Level Input Voltage[7] | DDR SDRAM models | $V_{DDR_VREF}$ + 0.15 | | $V_{DDDDR}$ + 0.3 | V |
| | High Level Input Voltage[7] | Mobile DDR SDRAM models | $V_{DDR_VREF}$ + 0.125 | | $V_{DDDDR}$ + 0.3 | V |
| $V_{IH5V}$[12] | High Level Input Voltage[8] | $V_{DDEXT}$ = maximum | 2.0 | | 5.5 | V |
| $V_{IHTWI}$ | High Level Input Voltage[9, 13] | $V_{DDEXT}$ = maximum | 0.7 x $V_{DDEXT}$ | | 5.5 | V |
| $V_{IHUSB}$ | High Level Input Voltage[10] | | | | 5.25 | V |
| $V_{IL}$ | Low Level Input Voltage[5, 11] | $V_{DDEXT}$ = minimum | –0.3 | | 0.6 | V |
| $V_{IL5V}$ | Low Level Input Voltage[12] | 3.3 V I/O, $V_{DDEXT}$ = minimum | –0.3 | | 0.8 | V |
| | Low Level Input Voltage[12] | 2.5 V I/O, $V_{DDEXT}$ = minimum | –0.3 | | 0.6 | V |
| $V_{ILDDR}$ | Low Level Input Voltage[7] | DDR SDRAM models | –0.3 | | $V_{DDR_VREF}$ – 0.15 | V |
| | Low Level Input Voltage[7] | Mobile DDR SDRAM models | –0.3 | | $V_{DDR_VREF}$ – 0.125 | V |
| $V_{ILTWI}$ | Low Level Input Voltage[9, 13] | | –0.3 | | 0.3 x $V_{DDEXT}$ | V |
| $V_{DDR_VREF}$ | DDR_VREF Pin Input Voltage | | 0.49 x $V_{DDDDR}$ | 0.50 x $V_{DDDDR}$ | 0.51 x $V_{DDDDR}$ | V |
| $T_J$[14] | Junction Temperature (400/533 MHz) | 400-Ball Chip Scale Package Ball Grid Array (CSP_BGA) @$T_{AMBIENT}$ = –40°C to +85°C | –40 | | +105 | °C |
| | Junction Temperature (600 MHz) | 400-Ball Chip Scale Package Ball Grid Array (CSP_BGA) @$T_{AMBIENT}$ = 0°C to +70°C | 0 | | +90 | °C |

[1] See Table 13 on Page 35 for frequency/voltage specifications.

[2] $V_{DDINT}$ maximum is 1.10 V during one-time-programmable (OTP) memory programming operations.

[3] $V_{DDEXT}$ minimum is 3.0 V and maximum is 3.6 V during OTP memory programming operations.

[4] Use of the internal voltage regulator is not supported on 600 MHz speed grade models or on automotive grade models. An external voltage regulator must be used.

[5] Bidirectional pins (D15–0, PA15–0, PB14–0, PC15–0, PD15–0, PE15–0, PF15–0, PG15–0, PH13–0, PI15–0, PJ14–0) and input pins ($\overline{ATAPI_PDIAG}$, USB_ID, TCK, TDI, TMS, $\overline{TRST}$, CLKIN, $\overline{RESET}$, $\overline{NMI}$, and BMODE3–0) of the ADSP-BF54x Blackfin processors are 3.3 V-tolerant (always accept up to 3.6 V maximum $V_{IH}$). Voltage compliance (on outputs, $V_{OH}$) is limited by the $V_{DDEXT}$ supply voltage. The regulator can generate $V_{DDINT}$ at levels of 0.90 V to 1.30 V with -5% to +5% tolerance.

[6] Parameter value applies to all input and bidirectional pins except PB1-0, PE15-14, PG15–11, PH7-6, DQ0-15, and DQS0-1.

[7] Parameter value applies to pins DQ0–15 and DQS0–1.
[8] PB1-0, PE15-14, PG15-11, and PH7-6 are 5.0 V-tolerant (always accept up to 5.5 V maximum $V_{IH}$ when power is applied to $V_{DDEXT}$ pins). Voltage compliance (on output $V_{OH}$) is limited by $V_{DDEXT}$ supply voltage.
[9] SDA and SCL are 5.0V tolerant (always accept up to 5.5V maximum $V_{IH}$). Voltage compliance on outputs ($V_{OH}$) is limited by the $V_{DDEXT}$ supply voltage.
[10] Parameter value applies to USB_DP, USB_DM, and USB_VBUS pins. See Absolute Maximum Ratings on Page 40.
[11] Parameter value applies to all input and bidirectional pins, except PB1-0, PE15-14, PG15–11, and PH7-6.
[12] Parameter value applies to pins PG15–11 and PH7-6.
[13] Parameter value applies to pins PB1-0 and PE15-14. Consult the $I^2C$ specification version 2.1 for the proper resistor value and other open drain pin electrical parameters.
[14] $T_J$ must be in the range: 0°C < $T_J$ < 55°C during OTP memory programming operations.

Table 13 and Table 16 describe the voltage/frequency requirements for the ADSP-BF54x Blackfin processors' clocks. Take care in selecting MSEL, SSEL, and CSEL ratios so as not to exceed the maximum core clock and system clock. Table 15 describes the phase-locked loop operating conditions.

**Table 13. Core Clock Requirements—533 MHz and 600 MHz Speed Grade[1]**

| Parameter | | Condition | Internal Regulator Setting[2] | Max | Unit |
|---|---|---|---|---|---|
| $f_{CCLK}$ | Core Clock Frequency | $V_{DDINT}$ = 1.30 V minimum | N/A[2] | 600 | MHz |
| $f_{CCLK}$ | Core Clock Frequency | $V_{DDINT}$ = 1.20 V minimum | 1.25 V | 533 | MHz |
| $f_{CCLK}$ | Core Clock Frequency | $V_{DDINT}$ = 1.14 V minimum | 1.20 V | 500 | MHz |
| $f_{CCLK}$ | Core Clock Frequency | $V_{DDINT}$ = 1.045 V minimum | 1.10 V | 444 | MHz |
| $f_{CCLK}$ | Core Clock Frequency | $V_{DDINT}$ = 0.95 V minimum | 1.00 V | 400 | MHz |
| $f_{CCLK}$ | Core Clock Frequency | $V_{DDINT}$ = 0.90 Vminimum | 0.95 V | 333 | MHz |

[1] See the Ordering Guide on Page 100.
[2] Use of an internal voltage regulator is not supported on automotive grade and 600 MHz speed grade models

**Table 14. Core Clock Requirements—400 MHz Speed Grade[1]**

| Parameter | | Condition | Internal Regulator Setting[2] | Max | Unit |
|---|---|---|---|---|---|
| $f_{CCLK}$ | Core Clock Frequency | $V_{DDINT}$ = 1.14 V minimum | 1.20 V | 400 | MHz |
| $f_{CCLK}$ | Core Clock Frequency | $V_{DDINT}$ = 1.045 V minimum | 1.10 V | 364 | MHz |
| $f_{CCLK}$ | Core Clock Frequency | $V_{DDINT}$ = 0.95 V minimum | 1.00 V | 333 | MHz |
| $f_{CCLK}$ | Core Clock Frequency | $V_{DDINT}$ = 0.90 V minimum | 0.95 V | 300 | MHz |

[1] See Ordering Guide on Page 100
[2] Use of an internal voltage regulator is not supported on automotive grade models

**Table 15. Phase-Locked Loop Operating Conditions**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| $f_{VCO}$ | Voltage Controlled Oscillator (VCO) Frequency | 50 | Maximum $f_{CCLK}$ | MHz |

**Table 16. System Clock Requirements**

| Parameter | Condition | DDR SDRAM Models | Mobile DDR SDRAM Models | | Unit |
|---|---|---|---|---|---|
| | | Max | Min | Max | |
| $f_{SCLK}$ | $V_{DDINT} \geq 1.14$ V[1] | 133[2] | 120[3] | 133[2] | MHz |
| $f_{SCLK}$ | $V_{DDINT} < 1.14$ V[1] | 100 | N/A[4] | N/A[4] | MHz |

[1] $f_{SCLK}$ must be less than or equal to $f_{CCLK}$.
[2] Rounded number. Actual test specification is SCLK period of 7.5 ns. See Table 26 on Page 43.
[3] Rounded number. Actual test specification is SCLK period of 8.33 ns.
[4] $V_{DDINT}$ must be greater than or equal to 1.14 V for mobile DDR SDRAM models. See Operating Conditions on Page 34.

## ELECTRICAL CHARACTERISTICS

| Parameter | | Test Conditions | Nonautomotive 400 MHz[1] Min | Typ | Max | All Other Devices[2] Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|---|---|---|
| $V_{OH}$ | High Level Output Voltage for 3.3 V I/O[3] | $V_{DDEXT}$ = 2.7 V, $I_{OH}$ = –0.5 mA | 2.4 | | | 2.4 | | | V |
| | High Level Output Voltage for 2.5 V I/O[3] | $V_{DDEXT}$ = 2.25 V, $I_{OH}$ = –0.5 mA | 2.0 | | | 2.0 | | | V |
| $V_{OHDDR}$ | High Level Output Voltage for DDR SDRAM[4] | $V_{DDDDR}$ = 2.5 V, $I_{OH}$ = –8.1 mA | 1.74 | | | 1.74 | | | V |
| | High Level Output Voltage for Mobile DDR SDRAM[4] | $V_{DDDDR}$ = 1.8 V, $I_{OH}$ = –0.1 mA | 1.62 | | | 1.62 | | | V |
| $V_{OL}$ | Low Level Output Voltage for 3.3 V I/O[3] | $V_{DDEXT}$ = 2.7 V, $I_{OL}$ = 2.0 mA | | | 0.4 | | | 0.4 | V |
| | Low Level Output Voltage for 2.5 V I/O[3] | $V_{DDEXT}$ = 2.25 V, $I_{OL}$ = 2.0 mA | | | 0.4 | | | 0.4 | V |
| $V_{OLDDR}$ | Low Level Output Voltage for DDR SDRAM[4] | $V_{DDDDR}$ = 2.5 V, $I_{OL}$ = 8.1 mA | | | 0.56 | | | 0.56 | V |
| | Low Level Output Voltage for Mobile DDR SDRAM[4] | $V_{DDDDR}$ = 1.8 V, $I_{OL}$ = 0.1 mA | | | 0.18 | | | 0.18 | V |
| $I_{IH}$ | High Level Input Current[5] | $V_{DDEXT}$ = 3.6 V, $V_{IN}$ = $V_{IN}$ Max | | | 10.0 | | | 10.0 | µA |
| $I_{IHP}$ | High Level Input Current[6] | $V_{DDEXT}$ = 3.6 V, $V_{IN}$ = $V_{IN}$ Max | | | 50.0 | | | 50.0 | µA |
| $I_{IHDDR_VREF}$ | High Level Input Current for DDR SDRAM[7] | $V_{DDDDR}$ = 2.7 V, $V_{IN}$ = 0.51 × $V_{DDDDR}$ | | | 30.0 | | | 30.0 | µA |
| | High Level Input Current for Mobile DDR SDRAM[7] | $V_{DDDDR}$ = 1.95 V, $V_{IN}$ = 0.51 × $V_{DDDDR}$ | | | 30.0 | | | 30.0 | µA |
| $I_{IL}$[8] | Low Level Input Current | $V_{DDEXT}$ = 3.6 V, $V_{IN}$ = 0 V | | | 10.0 | | | 10.0 | µA |
| $I_{OZH}$[9] | Three-State Leakage Current[10] | $V_{DDEXT}$ = 3.6 V, $V_{IN}$ = $V_{IN}$ Max | | | 10.0 | | | 10.0 | µA |
| $I_{OZL}$[11] | Three-State Leakage Current[10] | $V_{DDEXT}$ = 3.6 V, $V_{IN}$ = 0 V | | | 10.0 | | | 10.0 | µA |
| $C_{IN}$ | Input Capacitance[12] | $f_{IN}$ = 1 MHz, $T_{AMBIENT}$ = 25°C, $V_{IN}$ = 2.5 V | | 4[12] | 8[12] | | 4[12] | 8[12] | pF |
| $I_{DDDEEPSLEEP}$[13] | $V_{DDINT}$ Current in Deep Sleep Mode | $V_{DDINT}$ = 1.0 V, $f_{CCLK}$ = 0 MHz, $f_{SCLK}$ = 0 MHz, $T_J$ = 25°C, ASF = 0.00 | | 22 | | | 37 | | mA |
| $I_{DDSLEEP}$ | $V_{DDINT}$ Current in Sleep Mode | $V_{DDINT}$ = 1.0 V, $f_{SCLK}$ = 25 MHz, $T_J$ = 25°C | | 35 | | | 50 | | mA |
| $I_{DD-IDLE}$ | $V_{DDINT}$ Current in Idle | $V_{DDINT}$ = 1.0 V, $f_{CCLK}$ = 50 MHz, $T_J$ = 25°C, ASF = 0.47 | | 44 | | | 59 | | mA |

| Parameter | | Test Conditions | Nonautomotive 400 MHz[1] Min | Typ | Max | All Other Devices[2] Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|---|---|---|
| $I_{DD-TYP}$ | $V_{DDINT}$ Current | $V_{DDINT}$ = 1.10 V, $f_{CCLK}$ = 300 MHz, $f_{SCLK}$ = 25 MHz, $T_J$ = 25°C, ASF = 1.00 | | 145 | | | 178 | | mA |
| $I_{DD-TYP}$ | $V_{DDINT}$ Current | $V_{DDINT}$ = 1.20 V, $f_{CCLK}$ = 400 MHz, $f_{SCLK}$ = 25 MHz, $T_J$ = 25°C, ASF = 1.00 | | 199 | | | 239 | | mA |
| $I_{DD-TYP}$ | $V_{DDINT}$ Current | $V_{DDINT}$ = 1.25 V, $f_{CCLK}$ = 533 MHz, $f_{SCLK}$ = 25 MHz, $T_J$ = 25°C, ASF = 1.00 | | | | | 301 | | mA |
| $I_{DD-TYP}$ | $V_{DDINT}$ Current | $V_{DDINT}$ = 1.35 V, $f_{CCLK}$ = 600 MHz, $f_{SCLK}$ = 25 MHz, $T_J$ = 25°C, ASF = 1.00 | | | | | 360 | | mA |
| $I_{DDHIBERNATE}$[13, 14] | Hibernate State Current | $V_{DDEXT}$ = $V_{DDVR}$ = $V_{DDUSB}$ = 3.30 V, $V_{DDDDR}$ = 2.5 V, $T_J$ = 25°C, CLKIN= 0 MHz with voltage regulator off ($V_{DDINT}$ = 0 V) | | 60 | | | 60 | | µA |
| $I_{DDRTC}$ | $V_{DDRTC}$ Current | $V_{DDRTC}$ = 3.3 V, $T_J$ = 25°C | | 20 | | | 20 | | µA |
| $I_{DDUSB-FS}$ | $V_{DDUSB}$ Current in Full/Low Speed Mode | $V_{DDUSB}$ = 3.3 V, $T_J$ = 25°C, Full Speed USB Transmit | | 9 | | | 9 | | mA |
| $I_{DDUSB-HS}$ | $V_{DDUSB}$ Current in High Speed Mode | $V_{DDUSB}$ = 3.3 V, $T_J$ = 25°C, High Speed USB Transmit | | 25 | | | 25 | | mA |
| $I_{DDDEEPSLEEP}$[13, 15] | $V_{DDINT}$ Current in Deep Sleep Mode | $f_{CCLK}$ = 0 MHz, $f_{SCLK}$ = 0 MHz | | | Table 17 | | | Table 18 | mA |
| $I_{DDSLEEP}$[13, 15] | $V_{DDINIT}$ Current in Sleep Mode | $f_{CCLK}$ = 0 MHz, $f_{SCLK}$ > 0 MHz | | | $I_{DDDEEPSLEEP}$ + (0.77 × $V_{DDINT}$ × $f_{SCLK}$)[16] | | | $I_{DDDEEPSLEEP}$ + (0.77 × $V_{DDINT}$ × $f_{SCLK}$)[16] | mA[16] |
| $I_{DDINT}$[15, 17] | $V_{DDINT}$ Current | $f_{CCLK}$ > 0 MHz, $f_{SCLK}$ > 0 MHz | | | $I_{DDSLEEP}$ + (Table 20 × ASF) | | | $I_{DDSLEEP}$ + (Table 20 × ASF) | mA |

[1] Applies to all nonautomotive 400 MHz speed grade models. See Ordering Guide.
[2] Applies to all 533 MHz and 600 MHz speed grade models and automotive 400 MHz speed grade models. See Ordering Guide.
[3] Applies to output and bidirectional pins, except USB_VBUS and the pins listed in table note 4.
[4] Applies to pins DA0–12, DBA0–1, DQ0–15, DQS0–1, DQM0–1, DCLK1–2, $\overline{\text{DCLK1–2}}$, $\overline{\text{DCS0–1}}$, DCLKE, $\overline{\text{DRAS}}$, $\overline{\text{DCAS}}$, and $\overline{\text{DWE}}$.
[5] Applies to all input pins except JTAG inputs.
[6] Applies to JTAG input pins (TCK, TDI, TMS, $\overline{\text{TRST}}$).
[7] Applies to DDR_VREF pin.
[8] Absolute value.
[9] For DDR pins (DQ0-15, DQS0-1), test conditions are $V_{DDDDR}$ = Maximum, $V_{IN}$ = $V_{DDDDR}$ Maximum.
[10] Applies to three-statable pins.
[11] For DDR pins (DQ0-15, DQS0-1), test conditions are $V_{DDDDR}$ = Maximum, $V_{IN}$ = 0V.
[12] Guaranteed, but not tested

[13]See the *ADSP-BF54x Blackfin Processor Hardware Reference Manual* for definition of sleep, deep sleep, and hibernate operating modes.
[14]Includes current on $V_{DDEXT}$, $V_{DDUSB}$, $V_{DDVR}$, and $V_{DDDDR}$ supplies. Clock inputs are tied high or low.
[15]Guaranteed maximum specifications.
[16]Unit for $V_{DDINT}$ is V (volts). Unit for $f_{SCLK}$ is MHz. Example: 1.2 V, 133 MHz would be $0.77 \times 1.2 \times 133 = 122.9$ mA added to $I_{DDDEEPSLEEP}$.
[17]See Table 19 for the list of $I_{DDINT}$ power vectors covered.

Total power dissipation has two components:

- Static, including leakage current
- Dynamic, due to transistor switching characteristics

Many operating conditions can also affect power dissipation, including temperature, voltage, operating frequency, and processor activity. Electrical Characteristics on Page 36 shows the current dissipation for internal circuitry ($V_{DDINT}$). $I_{DDDEEPSLEEP}$ specifies static power dissipation as a function of voltage ($V_{DDINT}$) and temperature (see Table 17 and Table 18), and $I_{DDINT}$ specifies the total power specification for the listed test conditions, including the dynamic component as a function of voltage ($V_{DDINT}$) and frequency (Table 20).

There are two parts to the dynamic component. The first part is due to transistor switching in the core clock (CCLK) domain. This part is subject to an activity scaling factor (ASF) which represents application code running on the processor core and L1/L2 memories (Table 19). The ASF is combined with the CCLK frequency and $V_{DDINT}$ dependent data in Table 20 to calculate this part. The second part is due to transistor switching in the system clock (SCLK) domain, which is included in the $I_{DDINT}$ specification equation.

**Table 17. Static Current—Nonautomotive 400 MHz Speed Grade Devices (mA)[1]**

| | Voltage ($V_{DDINT}$)[2] | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $T_J$ (°C)[2] | 0.90 V | 0.95 V | 1.00 V | 1.05 V | 1.10 V | 1.15 V | 1.20 V | 1.25 V | 1.30 V | 1.35 V | 1.38 V | 1.40 V | 1.43 V |
| -40 | 11.9 | 13.5 | 15.5 | 17.7 | 20.3 | 23.3 | 26.8 | 30.6 | 35.0 | 39.9 | 43.2 | 45.5 | 49.5 |
| 0 | 20.1 | 22.3 | 24.7 | 27.8 | 31.1 | 34.9 | 39.3 | 44.2 | 49.6 | 55.7 | 59.8 | 62.5 | 67.2 |
| 25 | 31.2 | 34.2 | 37.5 | 41.3 | 45.6 | 50.3 | 55.7 | 61.7 | 68.2 | 75.4 | 80.3 | 83.6 | 88.6 |
| 45 | 47.0 | 51.0 | 55.5 | 60.6 | 66.0 | 72.0 | 78.8 | 86.1 | 94.2 | 102.9 | 108.9 | 112.8 | 118.2 |
| 55 | 58.6 | 63.1 | 68.3 | 74.1 | 80.3 | 87.1 | 94.9 | 103.0 | 112.0 | 122.0 | 128.4 | 132.8 | 140.0 |
| 70 | 80.7 | 86.6 | 93.0 | 100.2 | 108.1 | 116.7 | 125.9 | 136.0 | 146.8 | 158.7 | 166.4 | 171.6 | 179.5 |
| 85 | 107.0 | 114.3 | 122.5 | 131.5 | 141.2 | 151.7 | 163.1 | 175.3 | 188.5 | 202.7 | 211.8 | 218.0 | 226.7 |
| 100 | 153.9 | 163.0 | 173.3 | 184.8 | 197.0 | 210.0 | 224.1 | 239.0 | 255.1 | 272.4 | 283.4 | 290.8 | 300.6 |
| 105 | 171.7 | 181.5 | 192.7 | 205.1 | 218.3 | 232.4 | 247.5 | 263.6 | 280.9 | 299.3 | 308.7 | 314.9 | 325.7 |

[1] Values are guaranteed maximum $I_{DDDEEPSLEEP}$ for nonautomotive 400 MHz speed-grade devices.
[2] Valid temperature and voltage ranges are model-specific. See Operating Conditions on Page 34.

**Table 18. Static Current—Automotive 400 MHz and All 533 MHz/600 MHz Speed Grade Devices (mA)[1]**

| | Voltage ($V_{DDINT}$)[2] | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $T_J$ (°C)[2] | 0.90 V | 0.95 V | 1.00 V | 1.05 V | 1.10 V | 1.15 V | 1.20 V | 1.25 V | 1.30 V | 1.35 V | 1.38 V | 1.40 V | 1.43 V |
| -40 | 19.7 | 22.1 | 24.8 | 27.9 | 31.4 | 35.4 | 39.9 | 45.0 | 50.6 | 57.0 | 61.2 | 64.0 | 70.4 |
| 0 | 45.2 | 49.9 | 55.2 | 61.3 | 67.9 | 75.3 | 83.5 | 92.6 | 102.6 | 113.6 | 121.0 | 125.8 | 135.0 |
| 25 | 80.0 | 87.5 | 96.2 | 105.8 | 116.4 | 127.9 | 140.4 | 154.1 | 169.2 | 185.4 | 196.1 | 203.3 | 218.0 |
| 45 | 124.2 | 134.8 | 147.1 | 160.7 | 175.3 | 191.2 | 208.6 | 227.3 | 247.6 | 269.6 | 284.0 | 293.6 | 312.0 |
| 55 | 154.6 | 167.2 | 181.7 | 197.7 | 214.9 | 233.8 | 254.2 | 276.1 | 299.7 | 325.9 | 343.1 | 354.6 | 374.0 |
| 70 | 209.8 | 225.6 | 243.9 | 264.1 | 285.8 | 309.4 | 334.8 | 363.5 | 394.3 | 427.7 | 449.4 | 463.9 | 489.0 |
| 85 | 281.8 | 301.3 | 323.5 | 350.2 | 378.5 | 408.9 | 442.1 | 477.9 | 516.5 | 557.5 | 584.2 | 602.0 | 629.0 |
| 100 | 366.5 | 390.5 | 419.4 | 452.1 | 486.9 | 524.4 | 564.8 | 608.2 | 654.8 | 704.7 | 737.0 | 758.5 | 793.0 |
| 105 | 403.8 | 428.3 | 459.5 | 494.3 | 531.7 | 571.9 | 614.9 | 661.5 | 711.1 | 763.9 | 798.5 | 821.6 | 864.0 |

[1] Values are guaranteed maximum $I_{DDDEEPSLEEP}$ for automotive 400 MHz and all 533 MHz and 600 MHz speed grade devices.
[2] Valid temperature and voltage ranges are model-specific. See Operating Conditions on Page 34.

**Table 19. Activity Scaling Factors[1]**

| $I_{DDINT}$ Power Vector | Activity Scaling Factor (ASF) |
|---|---|
| $I_{DD-PEAK}$ | 1.29 |
| $I_{DD-HIGH}$ | 1.24 |
| $I_{DD-TYP}$ | 1.00 |
| $I_{DD-APP}$ | 0.87 |
| $I_{DD-NOP}$ | 0.74 |
| $I_{DD-IDLE}$ | 0.47 |

[1] See *Estimating Power for ADSP-BF534/BF536/BF537 Blackfin Processors (EE-297)*. The power vector information also applies to the ADSP-BF542/ADSP-BF544/ADSP-BF547/ADSP-BF548/ADSP-BF549 processors.

**Table 20. Dynamic Current in CCLK Domain (mA, with ASF = 1.0)[1]**

| | Voltage ($V_{DDINT}$)[2] | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $f_{CCLK}$ (MHz)[2] | 0.90 V | 0.95 V | 1.00 V | 1.05 V | 1.10 V | 1.15 V | 1.20 V | 1.25 V | 1.30 V | 1.35 V | 1.38 V | 1.40 V | 1.43 V |
| 100 | 29.7 | 31.6 | 33.9 | 35.7 | 37.9 | 40.5 | 42.9 | 45.5 | 48.2 | 50.8 | 52.0 | 53.5 | 54.6 |
| 200 | 55.3 | 58.9 | 62.5 | 66.0 | 70.0 | 74.0 | 78.3 | 82.5 | 86.7 | 91.3 | 93.3 | 95.6 | 97.6 |
| 300 | 80.8 | 85.8 | 91.0 | 96.0 | 101.3 | 107.0 | 112.8 | 118.7 | 124.6 | 130.9 | 133.8 | 137.0 | 140.0 |
| 400 | N/A | 112.2 | 119.4 | 125.5 | 132.4 | 139.6 | 146.9 | 154.6 | 162.3 | 170.0 | 173.8 | 177.8 | 181.6 |
| 500 | N/A | N/A | N/A | N/A | N/A | 171.9 | 180.6 | 189.9 | 199.1 | 205.7 | 210.3 | 213.0 | 217.6 |
| 533 | N/A | N/A | N/A | N/A | N/A | N/A | 191.9 | 201.6 | 211.5 | 218.0 | 222.8 | 225.7 | 230.5 |
| 600 | N/A | N/A | N/A | N/A | N/A | N/A | N/A | N/A | 233.1 | 241.4 | 246.7 | 252.7 | 258.1 |

[1] The values are not guaranteed as stand-alone maximum specifications. They must be combined with static current per the equations of Electrical Characteristics on Page 36.
[2] Valid frequency and voltage ranges are model-specific. See Operating Conditions on Page 34.

## ABSOLUTE MAXIMUM RATINGS

Stresses greater than those listed in Table 21 may cause permanent damage to the device. These are stress ratings only. Functional operation of the device at these or any other conditions greater than those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. Table 22 details the maximum duty cycle for input transient voltage.

**Table 21. Absolute Maximum Ratings**

| | |
|---|---|
| Internal (Core) Supply Voltage ($V_{DDINT}$) | –0.3 V to +1.43 V |
| External (I/O) Supply Voltage ($V_{DDEXT}$) | –0.3 V to +3.8 V |
| Input Voltage[1, 2, 3] | –0.5 V to +3.6 V |
| Output Voltage Swing | –0.5 V to $V_{DDEXT}$ + 0.5 V |
| $I_{OH}/I_{OL}$ Current per Single Pin[4] | 40 mA (max) |
| $I_{OH}/I_{OL}$ Current per Pin Group[4] | 80 mA (max) |
| Storage Temperature Range | –65°C to +150°C |
| Junction Temperature Underbias | +125°C |

[1] Applies to all bidirectional and input only pins except PB1-0, PE15-14, PG15–11, and PH7-6, where the absolute maximum input voltage range is –0.5 V to +5.5 V.

[2] Pins USB_DP, USB_DM, and USB_VBUS are 5 V-tolerant when VDDUSB is powered according to the operating conditions table. If VDDUSB supply voltage does not meet the specification in the operating conditions table, these pins could suffer long-term damage when driven to +5 V. If this condition is seen in the application, it can be corrected with additional circuitry to use the external host to power only the $V_{DDUSB}$ pins. Contact factory for application detail and reliability information.

[3] Applies only when $V_{DDEXT}$ is within specifications. When $V_{DDEXT}$ is outside specifications, the range is $V_{DDEXT}$ ± 0.2 V.

[4] For more information, see description preceding Table 23.

**Table 22. Maximum Duty Cycle for Input[1] Transient Voltage**

| $V_{IN}$ Max (V)[2] | $V_{IN}$ Min (V) | Maximum Duty Cycle |
|---|---|---|
| 3.63 | –0.33 | 100% |
| 3.80 | –0.50 | 48% |
| 3.90 | –0.60 | 30% |
| 4.00 | –0.70 | 20% |
| 4.10 | –0.80 | 10% |
| 4.20 | –0.90 | 8% |
| 4.30 | –1.00 | 5% |

[1] Does not apply to CLKIN. Absolute maximum for pins PB1-0, PE15-14, PG15-11, and PH7-6 is +5.5V.

[2] Only one of the listed options can apply to a particular design.

The Absolute Maximum Ratings table specifies the maximum total source/sink ($I_{OH}/I_{OL}$) current for a group of pins. Permanent damage can occur if this value is exceeded. To understand this specification, if pins PA4, PA3, PA2, PA1 and PA0 from group 1 in the Total Current Pin Groups table were sourcing or sinking 2 mA each, the total current for those pins would be 10 mA. This would allow up to 70 mA total that could be sourced or sunk by the remaining pins in the group without damaging the device. For a list of all groups and their pins, see the Total Current Pin Groups table. Note that the $V_{OL}$ and $V_{OH}$ specifications have separate per-pin maximum current requirements, see the Electrical Characteristics table.

**Table 23. Total Current Pin Groups**

| Group | Pins in Group |
|---|---|
| 1 | PA0, PA1, PA2, PA3, PA4, PA5, PA6, PA7, PA8, PA9, PA10, PA11 |
| 2 | PA12, PA13, PA14, PA15, PB8, PB9, PB10, PB11, PB12, PB13, PB14 |
| 3 | PB0, PB1, PB2, PB3, PB4, PB5, PB6, PB7, BMODE0, BMODE1, BMODE2, BMODE3 |
| 4 | TCK, TDI, TDO, TMS, $\overline{\text{TRST}}$, PD14, $\overline{\text{EMU}}$ |
| 5 | PD8, PD9, PD10, PD11, PD12, PD13, PD15 |
| 6 | PD0, PD1, PD2, PD3, PD4, PD5, PD6, PD7 |
| 7 | PE11, PE12, PE13, PF12, PF13, PF14, PF15, PG3, PG4 |
| 8 | PF4, PF5, PF6, PF7, PF8, PF9, PF10, PF11 |
| 9 | PF0, PF1, PF2, PF3, PG0, PG1, PG2 |
| 10 | PC0, PC1, PC2, PC3, PC4, PC5, PC6, PC7 |
| 11 | PH5, PH6, PH7 |
| 12 | A1, A2, A3 |
| 13 | PH8, PH9, PH10, PH11, PH12, PH13 |
| 14 | PI0, PI1, PI2, PI3, PI4, PI5, PI6, PI7 |
| 15 | PI8, PI9, PI10, PI11, PI12, PI13, PI14, PI15 |
| 16 | $\overline{\text{AMS0}}$, $\overline{\text{AMS1}}$, $\overline{\text{AMS2}}$, $\overline{\text{AMS3}}$, $\overline{\text{AOE}}$, CLKBUF, $\overline{\text{NMI}}$ |
| 17 | CLKIN, XTAL, $\overline{\text{RESET}}$, RTXI, RTXO, $\overline{\text{ARE}}$, $\overline{\text{AWE}}$ |
| 18 | D0, D1, D2, D3, D4, D5, D6, D7 |
| 19 | D8, D9, D10, D11, D12 |
| 20 | D13, D14, D15, $\overline{\text{ABE0}}$, $\overline{\text{ABE1}}$ |
| 21 | EXT_WAKE, CLKOUT, PJ11, PJ12, PJ13 |
| 22 | PJ0, PJ1, PJ2, PJ3, PJ4, PJ5, PJ6, PJ7, $\overline{\text{ATAPI_PDIAG}}$ |
| 23 | PJ8, PJ9, PJ10, PE7, PG12, PG13 |
| 24 | PE0, PE1, PE2, PE4, PE5, PE6, PE8, PE9, PE10, PH3, PH4 |
| 25 | PH0, PH2, PE14, PE15, PG5, PG6, PG7, PG8, PG9, PG10, PG11 |
| 26 | PC8, PC9, PC10, PC11, PC12, PC13, PE3, PG14, PG15, PH1 |

## ESD SENSITIVITY

**ESD (electrostatic discharge) sensitive device.** Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

## PACKAGE INFORMATION

The information presented in Figure 9 and Table 24 provides information related to specific product features. For a complete listing of product offerings, see the Ordering Guide on Page 100.

*Figure 9. Product Information on Package*

**Table 24. Package Information**

| Brand Key | Description |
|---|---|
| BF54x | x = 2, 4, 7, 8 or 9 |
| (M) | Mobile DDR Indicator (optional) |
| t | Temperature Range |
| pp | Package Type |
| Z | RoHS Compliant part |
| cc | See Ordering Guide |
| vvvvvv.x-q | Assembly Lot Code |
| n.n | Silicon Revision |
| # | RoHS Compliant Designation |
| yyww | Date Code |

## TIMING SPECIFICATIONS

Timing specifications are detailed in this section.

### *Clock and Reset Timing*

Table 25 and Figure 10 describe Clock Input and Reset Timing.
Table 26 and Figure 11 describe Clock Out Timing.

**Table 25. Clock Input and Reset Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{CKIN}$ | CLKIN Period[1, 2, 3, 4] | 20.0 | 100.0 | ns |
| $t_{CKINL}$ | CLKIN Low Pulse[2] | 8.0 | | ns |
| $t_{CKINH}$ | CLKIN High Pulse[2] | 8.0 | | ns |
| $t_{BUFDLAY}$ | CLKIN to CLKBUF Delay | | 10 | ns |
| $t_{WRST}$ | $\overline{\text{RESET}}$ Asserted Pulsewidth Low[5] | 11 $t_{CKIN}$ | | ns |
| $t_{RHWFT}$ | RESET High to First HWAIT/HWAITA Transition (Boot Host Wait Mode)[6,7,8,9] | 6100 $t_{CKIN}$ + 7900 $t_{SCLK}$ | | ns |
| $t_{RHWFT}$ | RESET High to First HWAIT/HWAITA Transition (Reset Output Mode)[7,10,11] | 6100 $t_{CKIN}$ | 7000 $t_{CKIN}$ | ns |

[1] Combinations of the CLKIN frequency and the PLL clock multiplier must not exceed the allowed $f_{VCO}$, $f_{CCLK}$, and $f_{SCLK}$ settings discussed in Table 16 and Table 13 on Page 35.
[2] Applies to PLL bypass mode and PLL non-bypass mode.
[3] CLKIN frequency and duty cycle must not change on the fly.
[4] If the DF bit in the PLL_CTL register is set, then the maximum $t_{CKIN}$ period is 50 ns.
[5] Applies after power-up sequence is complete. See Table 27 and Figure 12 for more information about power-up reset timing.
[6] Maximum value not specified due to variation resulting from boot mode selection and OTP memory programming.
[7] Values specified assume no invalidation preboot settings in OTP page PBS00L. Invalidating a PBS set will increase the value by 1875 $t_{CKIN}$ (typically).
[8] Applies only to boot modes BMODE=1, 2, 4, 6, 7, 10, 11, 14, 15.
[9] Use default $t_{SCLK}$ value unless PLL is reprogrammed during preboot. In case of PLL reprogramming use the new $t_{SCLK}$ value and add PLL_LOCKCNT settle time.
[10] When enabled by OTP_RESETOUT_HWAIT bit. If regular HWAIT is not required in an application, the OTP_RESETOUT_HWAIT bit in the same page instructs the HWAIT or HWAITA to simulate reset output functionality. Then an external resistor is expected to pull the signal to the reset level, as the pin itself is in high performance mode during reset.
[11] Variances are mainly dominated by PLL programming instructions in PBS00L page and boot code differences between silicon revisions. The earlier is bypassed in boot mode BMODE = 0. Maximum value assumes PLL programming instructions do not cause the SCLK frequency to decrease.

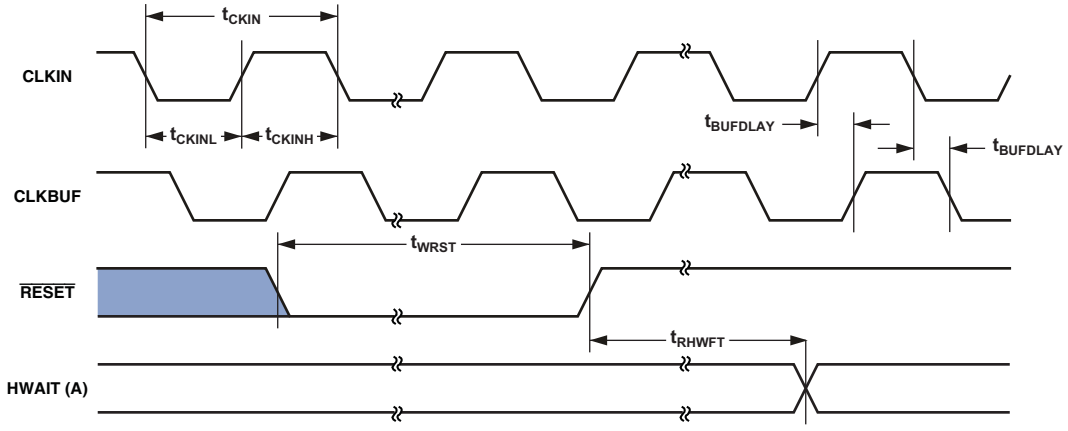



*Figure 10. Clock and Reset Timing*

**Table 26. Clock Out Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Switching Characteristics* | | | | |
| $t_{SCLK}$ | CLKOUT Period[1,2] | 7.5 | | ns |
| $t_{SCLKH}$ | CLKOUT Width High | 2.5 | | ns |
| $t_{SCLKL}$ | CLKOUT Width Low | 2.5 | | ns |

[1] The $t_{SCLK}$ value is the inverse of the $f_{SCLK}$ specification. Reduced supply voltages affect the best-case value of 7.5 ns listed here.
[2] The $t_{SCLK}$ value does not account for the effects of jitter.

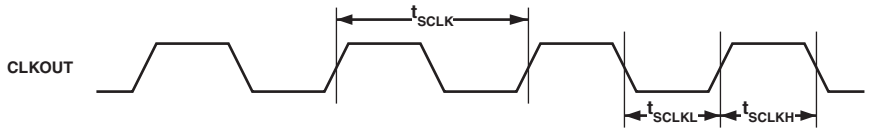
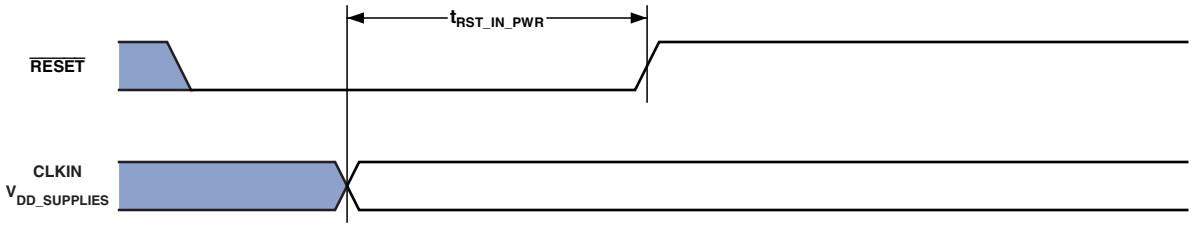



*Figure 11. CLKOUT Interface Timing*

**Table 27. Power-Up Reset Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{RST_IN_PWR}$ | $\overline{\text{RESET}}$ Deasserted After the $V_{DDINT}$, $V_{DDEXT}$, $V_{DDDDR}$,$V_{DDUSB}$,$V_{DDRTC}$,$V_{DDVR}$,$V_{DDMP}$, and CLKIN Pins Are Stable and Within Specification | 3500 × $t_{CKIN}$ | | ns |




In Figure 12, $V_{DD_SUPPLIES}$ is $V_{DDINT}$, $V_{DDEXT}$, $V_{DDDDR}$, $V_{DDUSB}$, $V_{DDRTC}$, $V_{DDVR}$, and $V_{DDMP}$.

*Figure 12. Power-Up Reset Timing*

### *Asynchronous Memory Read Cycle Timing*

Table 28 and Table 29 on Page 45 and Figure 13 and Figure 14 on Page 45 describe asynchronous memory read cycle operations for synchronous and for asynchronous ARDY.

**Table 28. Asynchronous Memory Read Cycle Timing with Synchronous ARDY**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SDAT}$ | DATA15–0 Setup Before CLKOUT | 5.0 | | ns |
| $t_{HDAT}$ | DATA15–0 Hold After CLKOUT | 0.8 | | ns |
| $t_{SARDY}$ | ARDY Setup Before the Falling Edge of CLKOUT | 5.0 | | ns |
| $t_{HARDY}$ | ARDY Hold After the Falling Edge of CLKOUT | 0.0 | | ns |
| *Switching Characteristics* | | | | |
| $t_{DO}$ | Output Delay After CLKOUT[1] | | 6.0 | ns |
| $t_{HO}$ | Output Hold After CLKOUT[1] | 0.3 | | ns |

[1] Output pins include $\overline{AMS3-0}$, $\overline{ABE1-0}$, ADDR19–1, $\overline{AOE}$, and $\overline{ARE}$.

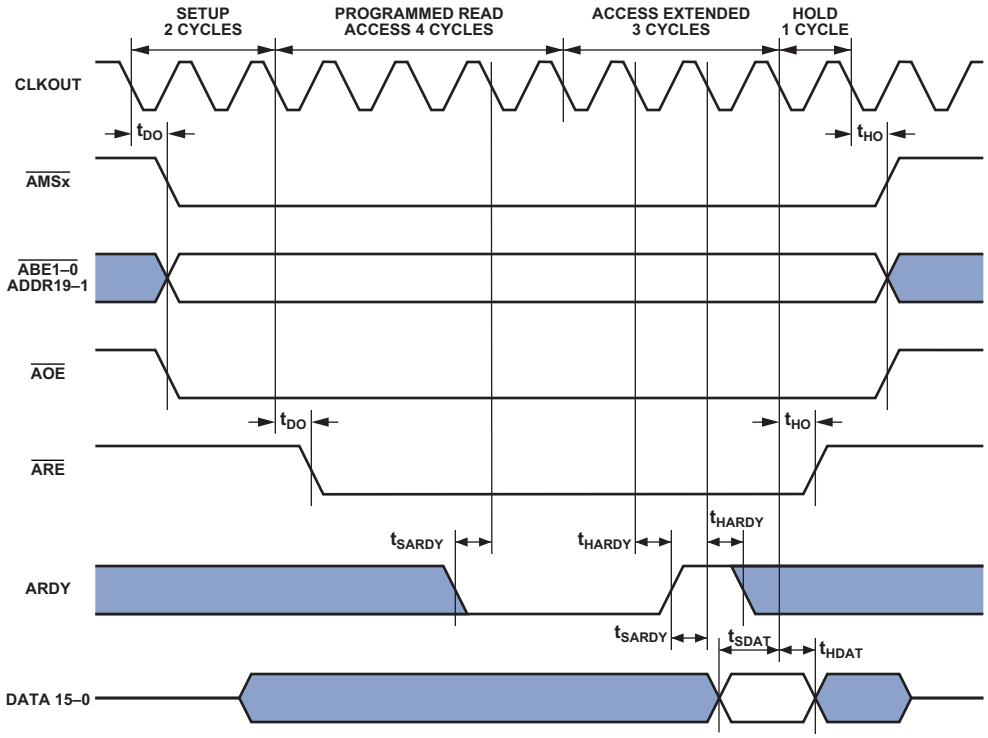



*Figure 13. Asynchronous Memory Read Cycle Timing with Synchronous ARDY*

**Table 29. Asynchronous Memory Read Cycle Timing with Asynchronous ARDY**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SDAT}$ | DATA15–0 Setup Before CLKOUT | 5.0 | | ns |
| $t_{HDAT}$ | DATA15–0 Hold After CLKOUT | 0.8 | | ns |
| $t_{DANR}$ | ARDY Negated Delay from $\overline{AMSx}$ Asserted[1] | | $(S + RA - 2) \times t_{SCLK}$ | ns |
| $t_{HAA}$ | ARDY Asserted Hold After $\overline{ARE}$ Negated | 0.0 | | ns |
| *Switching Characteristics* | | | | |
| $t_{DO}$ | Output Delay After CLKOUT[2] | | 6.0 | ns |
| $t_{HO}$ | Output Hold After CLKOUT[2] | 0.3 | | ns |

[1] S = number of programmed setup cycles, RA = number of programmed read access cycles.
[2] Output pins include $\overline{AMS3-0}$, $\overline{ABE1-0}$, ADDR19–1, $\overline{AOE}$, and $\overline{ARE}$.

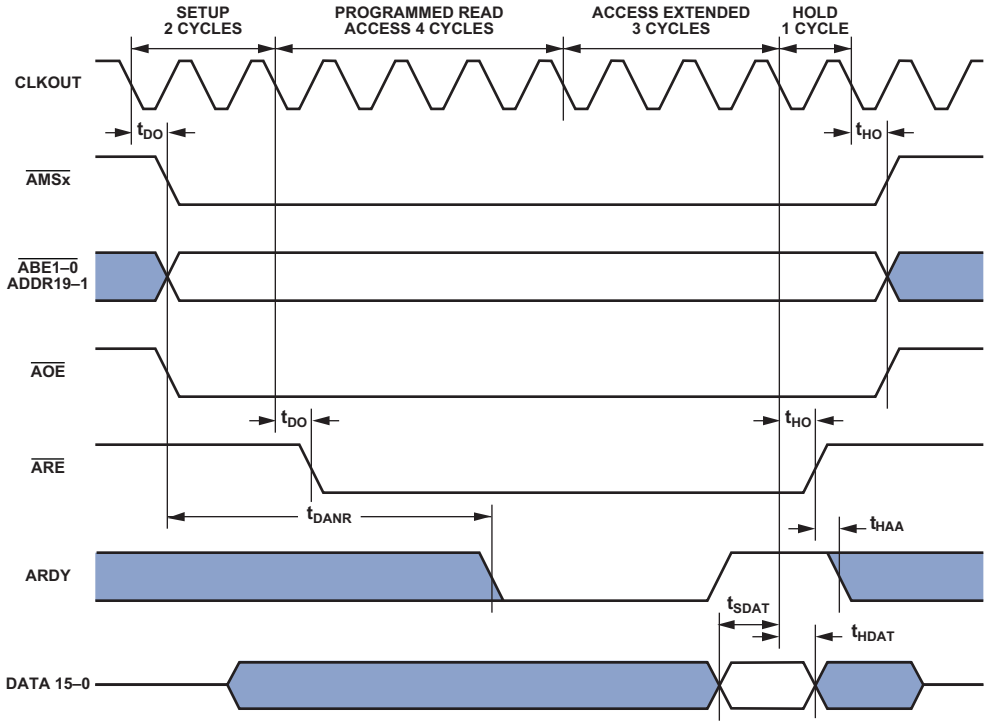



*Figure 14. Asynchronous Memory Read Cycle Timing with Asynchronous ARDY*

### *Asynchronous Memory Write Cycle Timing*

Table 30 and Table 31 on Page 47 and Figure 15 and Figure 16 on Page 47 describe asynchronous memory write cycle operations for synchronous and for asynchronous ARDY.

**Table 30. Asynchronous Memory Write Cycle Timing with Synchronous ARDY**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SARDY}$ | ARDY Setup Before the Falling Edge of CLKOUT | 5.0 | | ns |
| $t_{HARDY}$ | ARDY Hold After the Falling Edge of CLKOUT | 0.0 | | ns |
| *Switching Characteristics* | | | | |
| $t_{DDAT}$ | DATA15–0 Disable After CLKOUT | | 6.0 | ns |
| $t_{ENDAT}$ | DATA15–0 Enable After CLKOUT | 0.0 | | ns |
| $t_{DO}$ | Output Delay After CLKOUT[1] | | 6.0 | ns |
| $t_{HO}$ | Output Hold After CLKOUT[1] | 0.3 | | ns |

[1] Output pins include $\overline{AMS3-0}$, $\overline{ABE1-0}$, ADDR19–1, and $\overline{AWE}$.

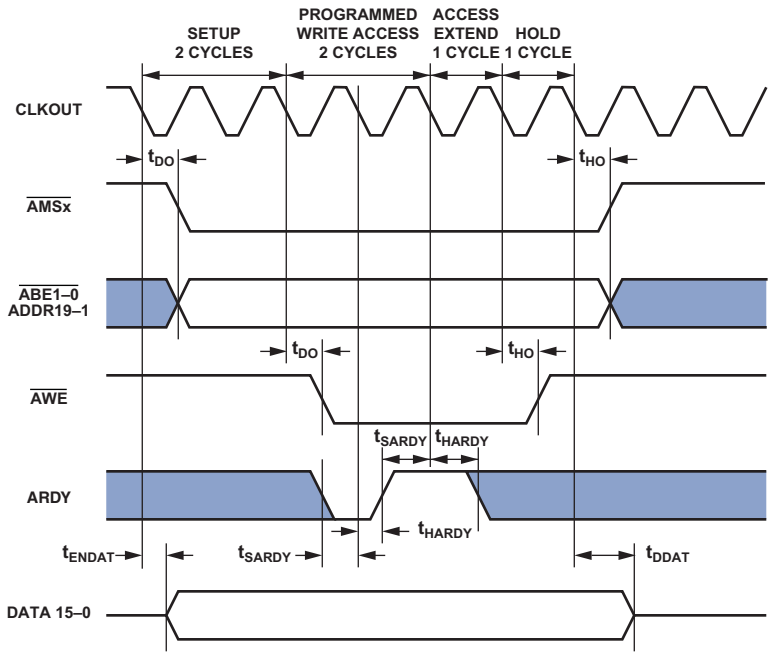



*Figure 15. Asynchronous Memory Write Cycle Timing with Synchronous ARDY*

**Table 31. Asynchronous Memory Write Cycle Timing with Asynchronous ARDY**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{DANW}$ | ARDY Negated Delay from $\overline{AMSx}$ Asserted[1] | | (S + WA – 2) × $t_{SCLK}$ | ns |
| $t_{HAA}$ | ARDY Asserted Hold After $\overline{AWE}$ Negated | 0.0 | | ns |
| *Switching Characteristics* | | | | |
| $t_{DDAT}$ | DATA15–0 Disable After CLKOUT | | 6.0 | ns |
| $t_{ENDAT}$ | DATA15–0 Enable After CLKOUT | 0.0 | | ns |
| $t_{DO}$ | Output Delay After CLKOUT[2] | | 6.0 | ns |
| $t_{HO}$ | Output Hold After CLKOUT[2] | 0.3 | | ns |

[1] S = number of programmed setup cycles, WA = number of programmed write access cycles.
[2] Output pins include $\overline{AMS3–0}$, $\overline{ABE1–0}$, ADDR19–1, $\overline{AOE}$, and $\overline{AWE}$.

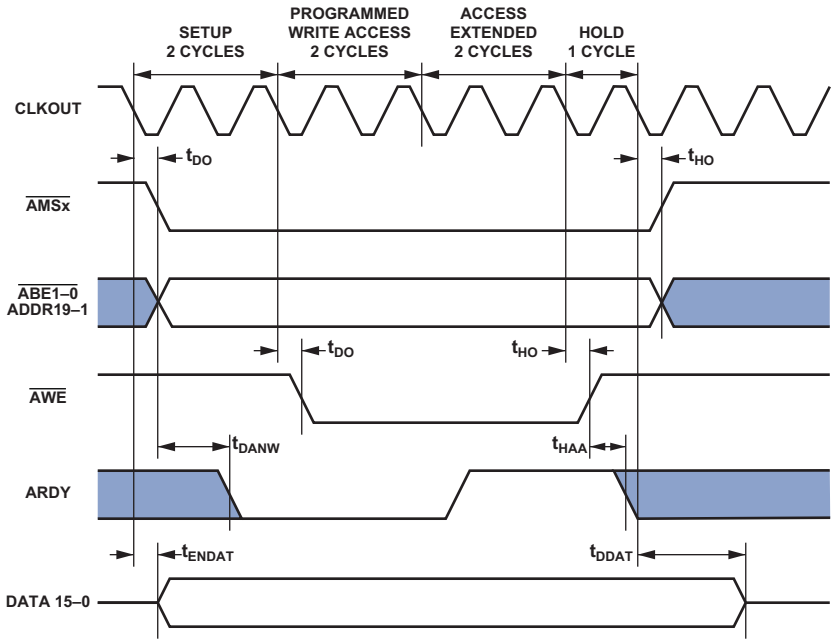



*Figure 16. Asynchronous Memory Write Cycle Timing with Asynchronous ARDY*

### *DDR SDRAM/Mobile DDR SDRAM Clock and Control Cycle Timing*

Table 32 and Figure 17 describe DDR SDRAM/mobile DDR SDRAM clock and control cycle timing.

**Table 32. DDR SDRAM/Mobile DDR SDRAM Clock and Control Cycle Timing**

| Parameter | | DDR SDRAM Min | DDR SDRAM Max | Mobile DDR SDRAM Min | Mobile DDR SDRAM Max | Unit |
|---|---|---|---|---|---|---|
| *Switching Characteristics* | | | | | | |
| $t_{CK}$[1] | DCK0-1 Period | 7.50 | | 7.50 | 8.33 | ns |
| $t_{CH}$ | DCK0-1 High Pulse Width | 0.45 | 0.55 | 0.45 | 0.55 | $t_{CK}$ |
| $t_{CL}$ | DCK0-1 Low Pulse Width | 0.45 | 0.55 | 0.45 | 0.55 | $t_{CK}$ |
| $t_{AS}$[2,3] | Address and Control Output SETUP Time Relative to CK | 1.00 | | 1.00 | | ns |
| $t_{AH}$[2,3] | Address and Control Output HOLD Time Relative to CK | 1.00 | | 1.00 | | ns |
| $t_{OPW}$[2,3] | Address and Control Output Pulse Width | 2.20 | | 2.30 | | ns |

[1] The $t_{CK}$ specification does not account for the effects of jitter.
[2] Address pins include DA0-12 and DBA0-1.
[3] Control pins include $\overline{\text{DCS0-1}}$, DCLKE, $\overline{\text{DRAS}}$, $\overline{\text{DCAS}}$, and $\overline{\text{DWE}}$.

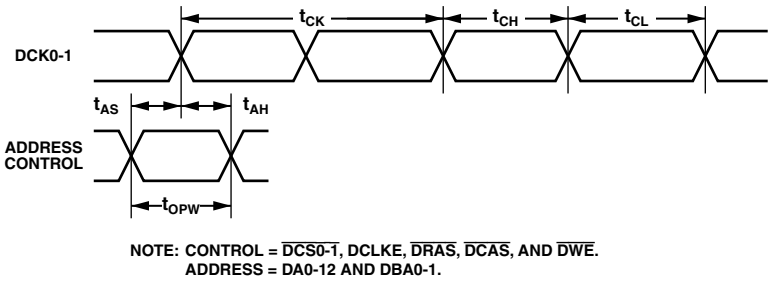



*Figure 17. DDR SDRAM /Mobile DDR SDRAM Clock and Control Cycle Timing*

### *DDR SDRAM/Mobile DDR SDRAM Timing*

Table 33 and Figure 18/Figure 19 describe DDR SDRAM/mobile DDR SDRAM read cycle timing.

**Table 33. DDR SDRAM/Mobile DDR SDRAM Read Cycle Timing**

| Parameter | | DDR SDRAM Min | DDR SDRAM Max | Mobile DDR SDRAM Min | Mobile DDR SDRAM Max | Unit |
|---|---|---|---|---|---|---|
| *Timing Requirements* | | | | | | |
| $t_{AC}$ | Access Window of DQ0-15 to DCK0-1 | –1.25 | +1.25 | 0.0 | 6.00 | ns |
| $t_{DQSCK}$ | Access Window of DQS0-1 to DCK0-1 | –1.25 | +1.25 | 0.0 | 6.00 | ns |
| $t_{DQSQ}$ | DQS0-1 to DQ0-15 Skew, DQS0-1 to Last DQ0-15 Valid | | 0.90 | | 0.85 | ns |
| $t_{QH}$ | DQ0-15 to DQS0-1 Hold, DQS0-1 to First DQ0-15 to Go Invalid | $t_{CK}/2 - 1.25$[1]<br>$t_{CK}/2 - 1.75$[2] | | $t_{CK}/2 - 1.25$ | | ns |
| $t_{RPRE}$ | DQS0-1 Read Preamble | 0.9 | 1.1 | 0.9 | 1.1 | $t_{CK}$ |
| $t_{RPST}$ | DQS0-1 Read Postamble | 0.4 | 0.6 | 0.4 | 0.6 | $t_{CK}$ |

[1] For 7.50 ns $\leq t_{CK} < 10$ ns.
[2] For $t_{CK} \geq 10$ ns.

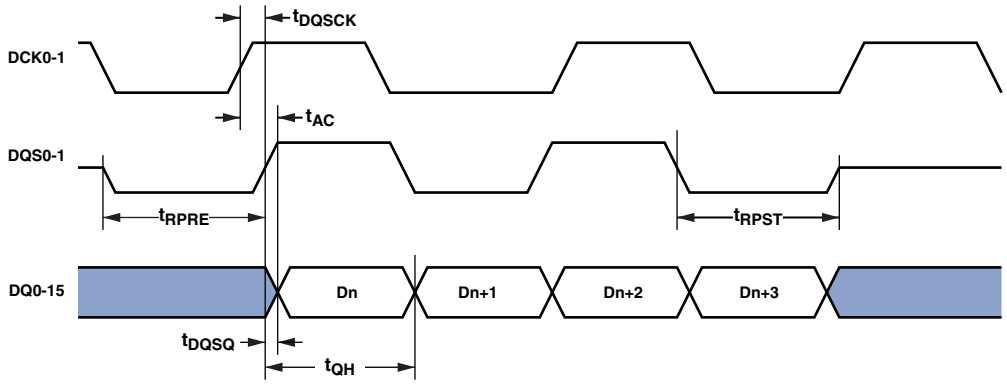



*Figure 18. DDR SDRAM Controller Read Cycle Timing*

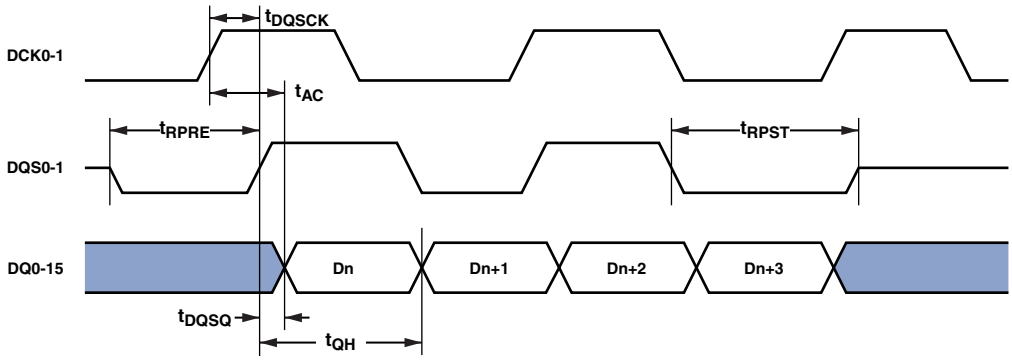



*Figure 19. Mobile DDR SDRAM Controller Read Cycle Timing*

### *DDR SDRAM/Mobile DDR SDRAM Write Cycle Timing*

Table 34 and Figure 20 describe DDR SDRAM/mobile DDR SDRAM write cycle timing.

**Table 34. DDR SDRAM/Mobile DDR SDRAM Write Cycle Timing**

| Parameter | | DDR SDRAM Min | DDR SDRAM Max | Mobile DDR SDRAM Min | Mobile DDR SDRAM Max | Unit |
|---|---|---|---|---|---|---|
| *Switching Characteristics* | | | | | | |
| $t_{DQSS}$ | Write CMD to First DQS0-1 | 0.75 | 1.25 | 0.75 | 1.25 | $t_{CK}$ |
| $t_{DS}$ | DQ0-15/DQM0-1 Setup to DQS0-1 | 0.90 | | 0.90 | | ns |
| $t_{DH}$ | DQ0-15/DQM0-1 Hold to DQS0-1 | 0.90 | | 0.90 | | ns |
| $t_{DSS}$ | DQS0-1 Falling to DCK0-1 Rising (DQS0-1 Setup) | 0.20 | | 0.20 | | $t_{CK}$ |
| $t_{DSH}$ | DQS0-1 Falling from DCK0-1 Rising (DQS0-1 Hold) | 0.20 | | 0.20 | | $t_{CK}$ |
| $t_{DQSH}$ | DQS0-1 High Pulse Width | 0.35 | | 0.40 | 0.60 | $t_{CK}$ |
| $t_{DQSL}$ | DQS0-1 Low Pulse Width | 0.35 | | 0.40 | 0.60 | $t_{CK}$ |
| $t_{WPRE}$ | DQS0-1 Write Preamble | 0.25 | | 0.25 | | $t_{CK}$ |
| $t_{WPST}$ | DQS0-1 Write Postamble | 0.40 | 0.60 | 0.40 | 0.60 | $t_{CK}$ |
| $t_{DOPW}$ | DQ0-15 and DQM0-1 Output Pulse Width (for Each) | 1.75 | | 1.75 | | ns |

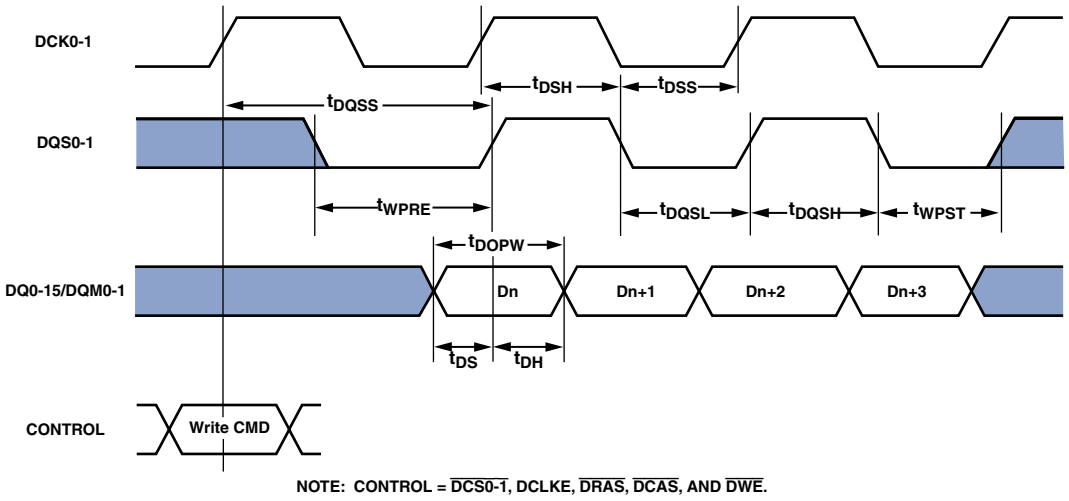



*Figure 20. DDR SDRAM /Mobile DDR SDRAM Controller Write Cycle Timing*

### *External Port Bus Request and Grant Cycle Timing*

Table 35 and Table 36 on Page 52 and Figure 21 and Figure 22 on Page 52 describe external port bus request and grant cycle operations for synchronous and for asynchronous $\overline{BR}$.

**Table 35. External Port Bus Request and Grant Cycle Timing with Synchronous $\overline{BR}$**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{BS}$ | $\overline{BR}$ Asserted to CLKOUT Low Setup | 5.0 | | ns |
| $t_{BH}$ | CLKOUT Low to $\overline{BR}$ Deasserted Hold Time | 0.0 | | ns |
| *Switching Characteristics* | | | | |
| $t_{SD}$ | CLKOUT Low to $\overline{AMSx}$, Address, and $\overline{ARE}$/$\overline{AWE}$ Disable | | 5.0 | ns |
| $t_{SE}$ | CLKOUT Low to $\overline{AMSx}$, Address, and $\overline{ARE}$/$\overline{AWE}$ Enable | | 5.0 | ns |
| $t_{DBG}$ | CLKOUT Low to $\overline{BG}$ Asserted Output Delay | | 4.0 | ns |
| $t_{EBG}$ | CLKOUT Low to $\overline{BG}$ Deasserted Output Hold | | 4.0 | ns |
| $t_{DBH}$ | CLKOUT Low to $\overline{BGH}$ Asserted Output Delay | | 3.6 | ns |
| $t_{EBH}$ | CLKOUT Low to $\overline{BGH}$ Deasserted Output Hold | | 3.6 | ns |

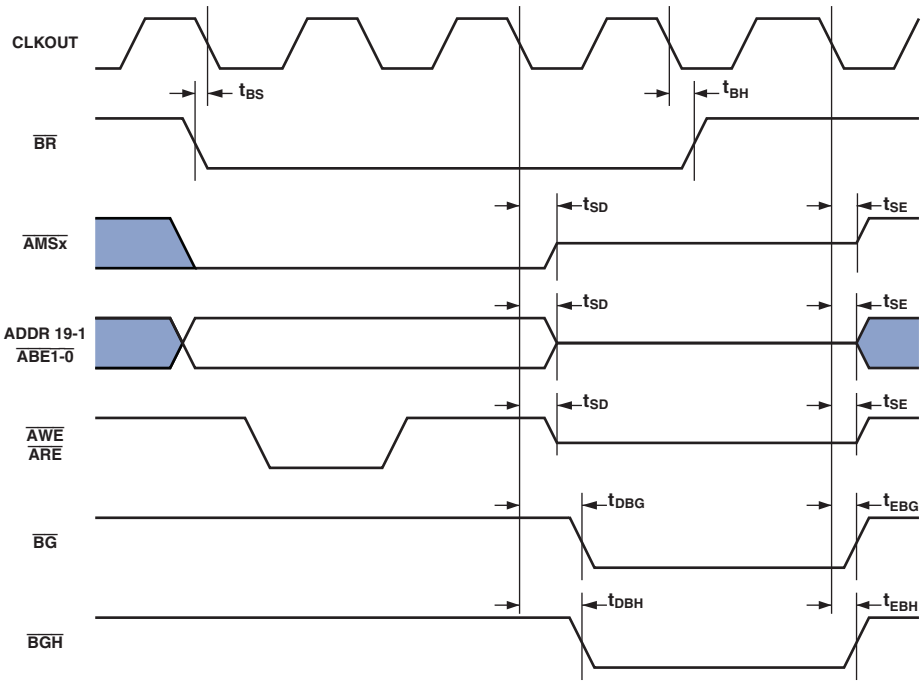



*Figure 21. External Port Bus Request and Grant Cycle Timing with Synchronous $\overline{BR}$*

**Table 36. External Port Bus Request and Grant Cycle Timing with Asynchronous $\overline{BR}$**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{WBR}$ | $\overline{BR}$ Pulsewidth | 2 x $t_{SCLK}$ | | ns |
| *Switching Characteristics* | | | | |
| $t_{SD}$ | CLKOUT Low to $\overline{AMSx}$, Address, and $\overline{ARE}$/$\overline{AWE}$ Disable | | 5.0 | ns |
| $t_{SE}$ | CLKOUT Low to $\overline{AMSx}$, Address, and $\overline{ARE}$/$\overline{AWE}$ Enable | | 5.0 | ns |
| $t_{DBG}$ | CLKOUT Low to $\overline{BG}$ Asserted Output Delay | | 4.0 | ns |
| $t_{EBG}$ | CLKOUT Low to $\overline{BG}$ Deasserted Output Hold | | 4.0 | ns |
| $t_{DBH}$ | CLKOUT Low to $\overline{BGH}$ Asserted Output Delay | | 3.6 | ns |
| $t_{EBH}$ | CLKOUT Low to $\overline{BGH}$ Deasserted Output Hold | | 3.6 | ns |

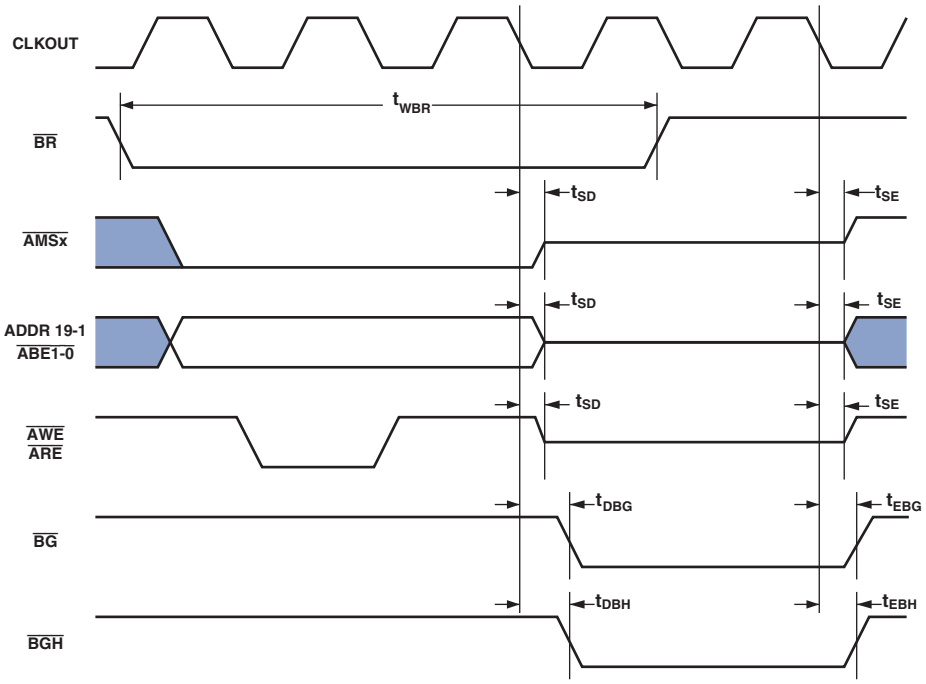



*Figure 22. External Port Bus Request and Grant Cycle Timing with Asynchronous $\overline{BR}$*

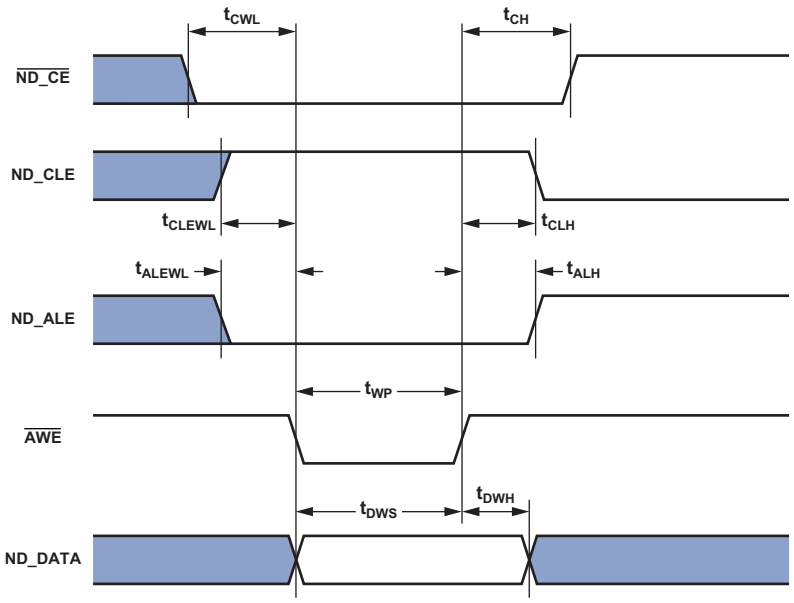
### *NAND Flash Controller Interface Timing*

Table 37 and Figure 23 on Page 54 through Figure 27 on Page 56 describe NAND flash controller interface operations.

**Table 37. NAND Flash Controller Interface Timing**

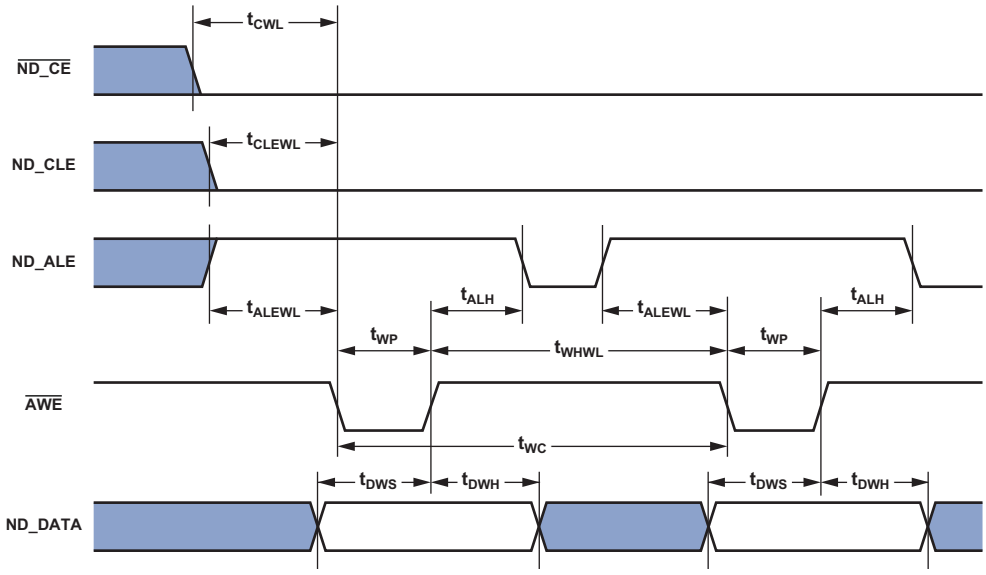
| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| **Write Cycle** | | | | |
| *Switching Characteristics* | | | | |
| $t_{CWL}$ | $\overline{ND_CE}$ Setup Time to $\overline{AWE}$ Low | $1.0 \times t_{SCLK} - 4$ | | ns |
| $t_{CH}$ | $\overline{ND_CE}$ Hold Time from $\overline{AWE}$ High | $3.0 \times t_{SCLK} - 4$ | | ns |
| $t_{CLHWL}$ | ND_CLE Setup Time High to $\overline{AWE}$ Low | 0.0 | | ns |
| $t_{CLH}$ | ND_CLE Hold Time from $\overline{AWE}$ High | $2.5 \times t_{SCLK} - 4$ | | ns |
| $t_{ALLWL}$ | ND_ALE Setup Time Low to $\overline{AWE}$ Low | 0.0 | | ns |
| $t_{ALH}$ | ND_ALE Hold Time from $\overline{AWE}$ High | $2.5 \times t_{SCLK} - 4$ | | ns |
| $t_{WP}$[1] | $\overline{AWE}$ Low to $\overline{AWE}$ High | $(WR_DLY + 1.0) \times t_{SCLK} - 4$ | | ns |
| $t_{WHWL}$ | $\overline{AWE}$ High to $\overline{AWE}$ Low | $4.0 \times t_{SCLK} - 4$ | | ns |
| $t_{WC}$[1] | $\overline{AWE}$ Low to $\overline{AWE}$ Low | $(WR_DLY + 5.0) \times t_{SCLK} - 4$ | | ns |
| $t_{DWS}$[1] | Data Setup Time for a Write Access | $(WR_DLY + 1.5) \times t_{SCLK} - 4$ | | ns |
| $t_{DWH}$ | Data Hold Time for a Write Access | $2.5 \times t_{SCLK} - 4$ | | ns |
| **Read Cycle** | | | | |
| *Switching Characteristics* | | | | |
| $t_{CRL}$ | $\overline{ND_CE}$ Setup Time to $\overline{ARE}$ Low | $1.0 \times t_{SCLK} - 4$ | | ns |
| $t_{CRH}$ | $\overline{ND_CE}$ Hold Time from $\overline{ARE}$ High | $3.0 \times t_{SCLK} - 4$ | | ns |
| $t_{RP}$[1] | $\overline{ARE}$ Low to $\overline{ARE}$ High | $(RD_DLY + 1.0) \times t_{SCLK} - 4$ | | ns |
| $t_{RHRL}$ | $\overline{ARE}$ High to $\overline{ARE}$ Low | $4.0 \times t_{SCLK} - 4$ | | ns |
| $t_{RC}$[1] | $\overline{ARE}$ Low to $\overline{ARE}$ Low | $(RD_DLY + 5.0) \times t_{SCLK} - 4$ | | ns |
| *Timing Requirements* | | | | |
| $t_{DRS}$ | Data Setup Time for a Read Transaction | 8.0 | | ns |
| $t_{DRH}$ | Data Hold Time for a Read Transaction | 0.0 | | ns |
| **Write Followed by Read** | | | | |
| *Switching Characteristic* | | | | |
| $t_{WHRL}$ | $\overline{AWE}$ High to $\overline{ARE}$ Low | $5.0 \times t_{SCLK} - 4$ | | ns |

[1] WR_DLY and RD_DLY are defined in the NFC_CTL register.



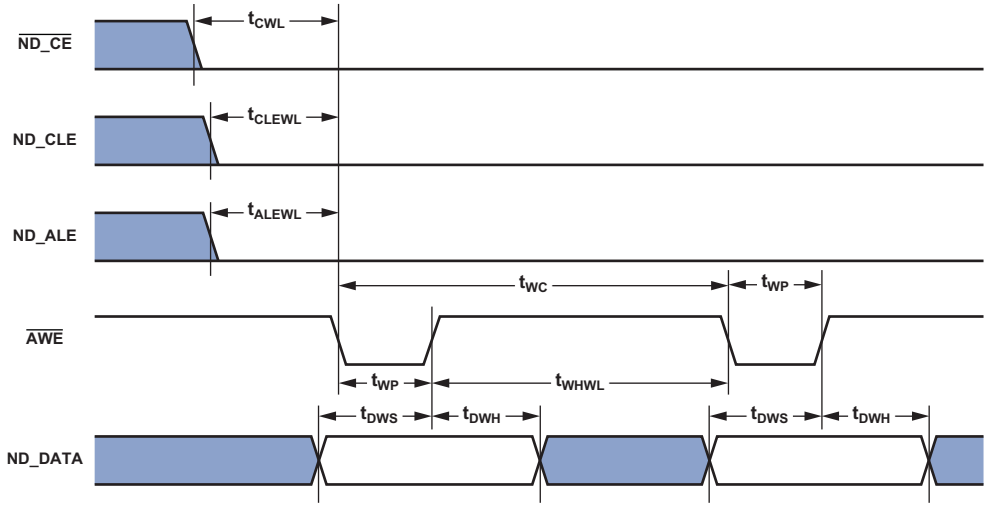

In Figure 23, ND_DATA is ND_D0–D15.

*Figure 23. NAND Flash Controller Interface Timing—Command Wri Cycle*



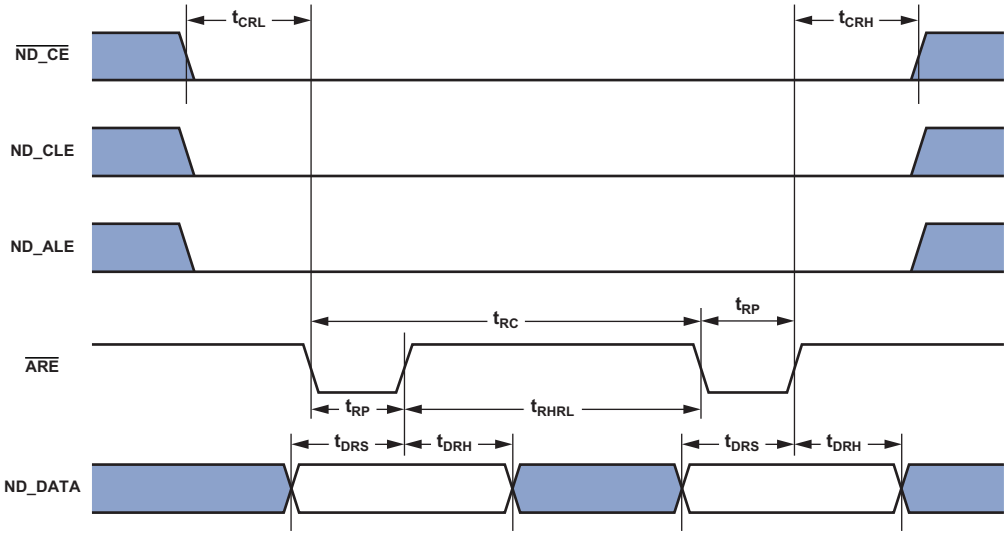

In Figure 24, ND_DATA is ND_D0–D15.

*Figure 24. NAND Flash Controller Interface Timing—Address Write Cycle*



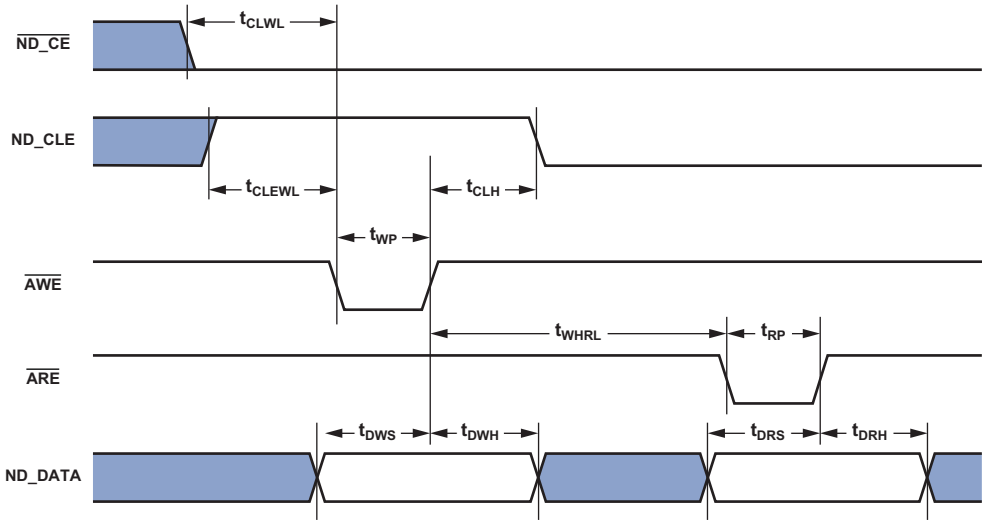

In Figure 25, ND_DATA is ND_D0–D15.

*Figure 25. NAND Flash Controller Interface Timing—Data Write Operation*




In Figure 26, ND_DATA is ND_D0–D15.

*Figure 26. NAND Flash Controller Interface Timing—Data Read Operation*




In Figure 27, ND_DATA is ND_D0–D15.

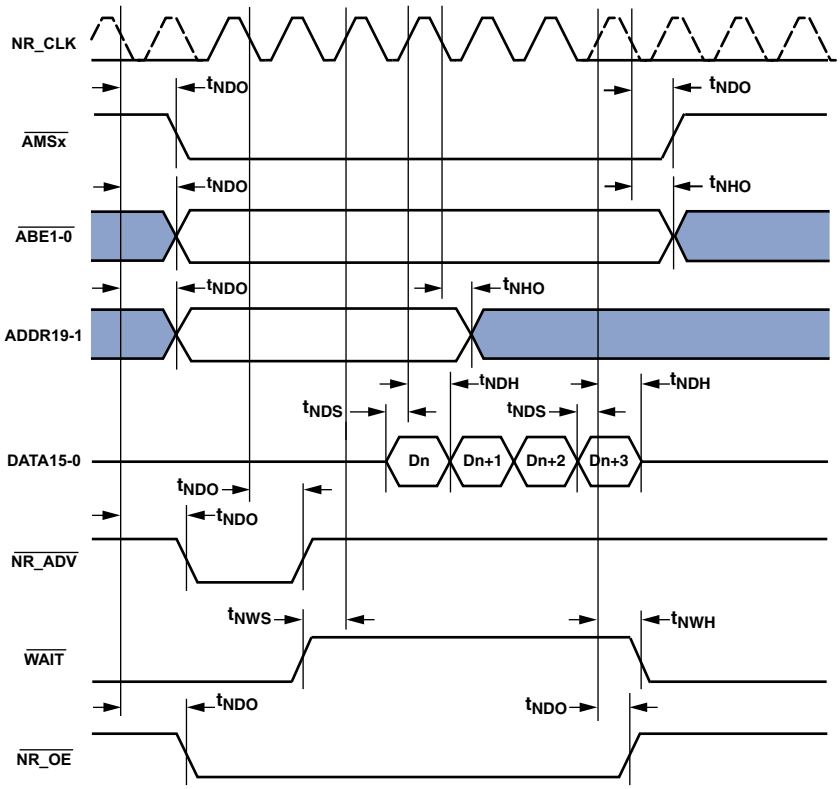
*Figure 27. NAND Flash Controller Interface Timing—Write Followed by Read Operation*

### *Synchronous Burst AC Timing*

Table 38 and Figure 28 on Page 57 describe Synchronous Burst AC operations.

**Table 38. Synchronous Burst AC Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{NDS}$ | DATA15-0 Setup Before NR_CLK | 4.0 | | ns |
| $t_{NDH}$ | DATA15-0 Hold After NR_CLK | 2.0 | | ns |
| $t_{NWS}$ | WAIT Setup Before NR_CLK | 8.0 | | ns |
| $t_{NWH}$ | WAIT Hold After NR_CLK | 0.0 | | ns |
| *Switching Characteristics* | | | | |
| $t_{NDO}$ | $\overline{AMSx}$, $\overline{ABE1-0}$, ADDR19-1, $\overline{NR_ADV}$, $\overline{NR_OE}$ Output Delay After NR_CLK | 6.0 | | ns |
| $t_{NHO}$ | $\overline{ABE1-0}$, ADDR19-1 Output Hold After NR_CLK | –3.0 | | ns |




**NOTE: NR_CLK dotted line represents a free running version of NR_CLK that is not visible on the NR_CLK pin.**

*Figure 28. Synchronous Burst AC Interface Timing*

### *External DMA Request Timing*

Table 39 and Figure 29 describe the external DMA request timing operations.

**Table 39. External DMA Request Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Parameters* | | | | |
| $t_{DR}$ | DMARx Asserted to CLKOUT High Setup | 6.0 | | ns |
| $t_{DH}$ | CLKOUT High to DMARx Deasserted Hold Time | 0.0 | | ns |
| $t_{DMARACT}$ | DMARx Active Pulse Width | $1.0 \times t_{SCLK}$ | | ns |
| $t_{DMARINACT}$ | DMARx Inactive Pulse Width | $1.75 \times t_{SCLK}$ | | ns |

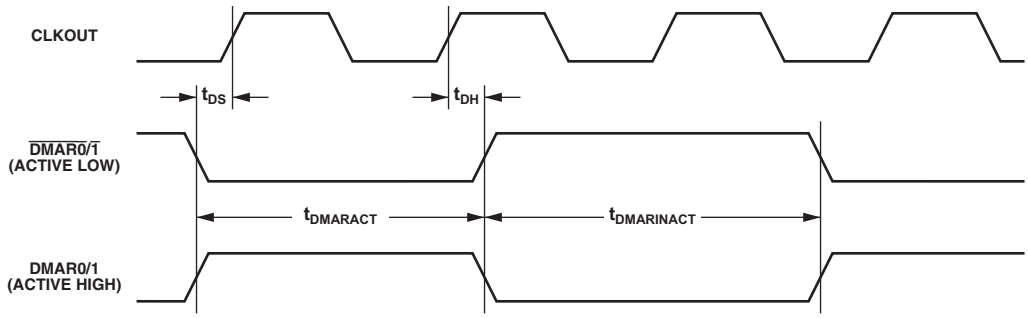



*Figure 29. External DMA Request Timing*

### *Enhanced Parallel Peripheral Interface Timing*

Table 40 and Figure 32 on Page 60, Figure 30 on Page 59, Figure 33 on Page 60, and Figure 31 on Page 59 describe enhanced parallel peripheral interface timing operations.

**Table 40. Enhanced Parallel Peripheral Interface Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{PCLKW}$ | PPIx_CLK Width | 6.0 | | ns |
| $t_{PCLK}$ | PPIx_CLK Period | 13.3 | | ns |
| *Timing Requirements—GP Input and Frame Capture Modes* | | | | |
| $t_{SFSPE}$ | External Frame Sync Setup Before PPIx_CLK | 0.9 | | ns |
| $t_{HFSPE}$ | External Frame Sync Hold After PPIx_CLK | 1.9 | | ns |
| $t_{SDRPE}$ | Receive Data Setup Before PPIx_CLK | 1.6 | | ns |
| $t_{HDRPE}$ | Receive Data Hold After PPIx_CLK | 1.5 | | ns |
| *Switching Characteristics—GP Output and Frame Capture Modes* | | | | |
| $t_{DFSPE}$ | Internal Frame Sync Delay After PPIx_CLK | | 10.5 | ns |
| $t_{HOFSPE}$ | Internal Frame Sync Hold After PPIx_CLK | 2.4 | | ns |
| $t_{DDTPE}$ | Transmit Data Delay After PPIx_CLK | | 9.9 | ns |
| $t_{HDTPE}$ | Transmit Data Hold After PPIx_CLK | 2.4 | | ns |

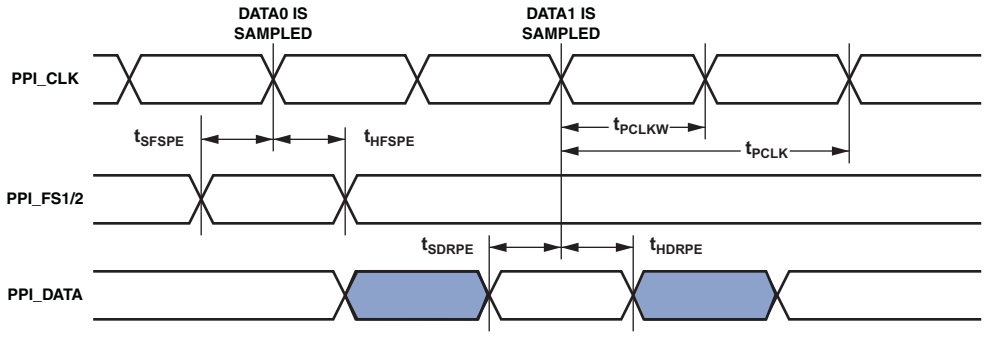



*Figure 30. EPPI GP Rx Mode with External Frame Sync Timing*

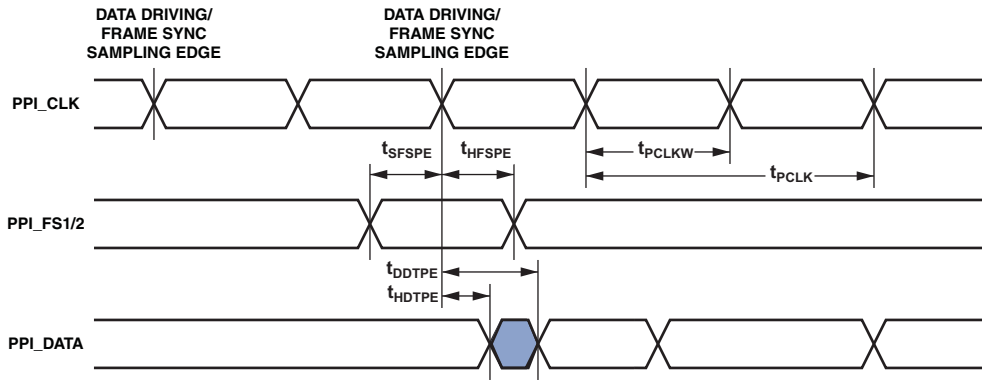



*Figure 31. EPPI GP Tx Mode with External Frame Sync Timing*

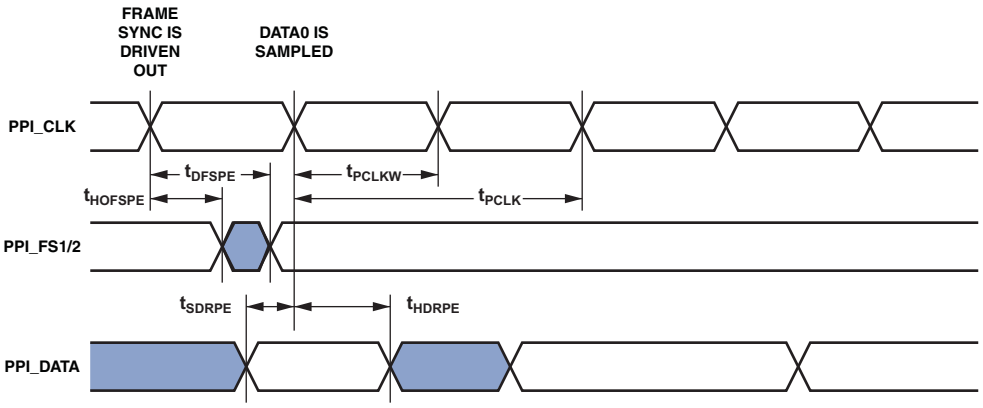



*Figure 32. EPPI GP Rx Mode with Internal Frame Sync Timing*

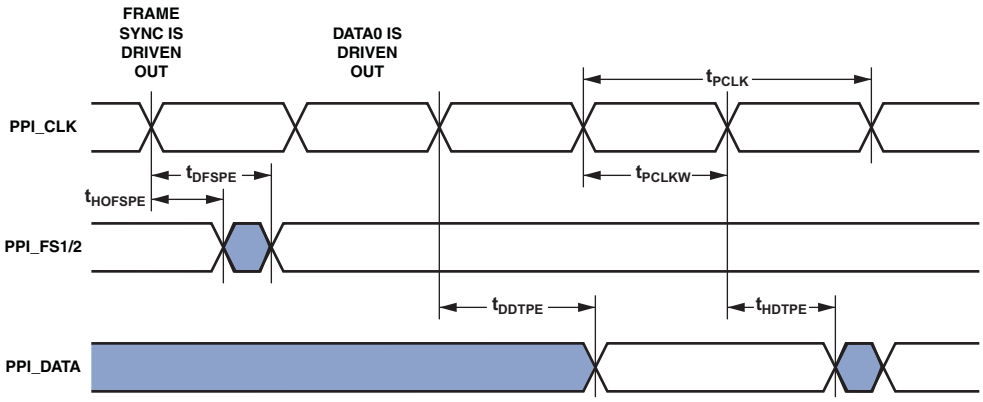



*Figure 33. EPPI GP Tx Mode with Internal Frame Sync Timing*

### *Serial Ports Timing*

Table 41 through Table 44 on Page 63 and Figure 34 on Page 62 through Figure 37 on Page 63 describe serial port operations.

**Table 41. Serial Ports—External Clock**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SFSE}$ | TFSx/RFSx Setup Before TSCLKx/RSCLKx (Externally Generated TFSx/RFSx)[1] | 3.0 | | ns |
| $t_{HFSE}$ | TFSx/RFSx Hold After TSCLKx/RSCLKx (Externally Generated TFSx/RFSx)[1] | 3.0 | | ns |
| $t_{SDRE}$ | Receive Data Setup Before RSCLKx[1] | 3.0 | | ns |
| $t_{HDRE}$ | Receive Data Hold After RSCLKx[1] | 3.0 | | ns |
| $t_{SCLKEW}$ | TSCLKx/RSCLKx Width | 4.5 | | ns |
| $t_{SCLKE}$ | TSCLKx/RSCLKx Period | 15.0 | | ns |
| $t_{RCLKE}$ | RSCLKx Period[2] | 11.1 | | ns |
| $t_{SUDTE}$ | Start-Up Delay From SPORT Enable To First External TFSx | $4 \times t_{SCLKE}$ | | ns |
| $t_{SUDRE}$ | Start-Up Delay From SPORT Enable To First External RFSx | $4 \times t_{RCLKE}$ | | ns |
| *Switching Characteristics* | | | | |
| $t_{DFSE}$ | TFSx/RFSx Delay After TSCLKx/RSCLKx (Internally Generated TFSx/RFSx)[3] | | 10.0 | ns |
| $t_{HOFSE}$ | TFSx/RFSx Hold After TSCLKx/RSCLKx (Internally Generated TFSx/RFSx)[3] | 0.0 | | ns |
| $t_{DDTE}$ | Transmit Data Delay After TSCLKx[3] | | 10.0 | ns |
| $t_{HDTE}$ | Transmit Data Hold After TSCLKx[3] | 0.0 | | ns |

[1] Referenced to sample edge.
[2] For serial port receive with external clock and external frame sync only.
[3] Referenced to drive edge.

**Table 42. Serial Ports—Internal Clock**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SFSI}$ | TFSx/RFSx Setup Before TSCLKx/RSCLKx (Externally Generated TFSx/RFSx)[1] | 10.0 | | ns |
| $t_{HFSI}$ | TFSx/RFSx Hold After TSCLKx/RSCLKx (Externally Generated TFSx/RFSx)[1] | –1.5 | | ns |
| $t_{SDRI}$ | Receive Data Setup Before RSCLKx[1] | 10.0 | | ns |
| $t_{HDRI}$ | Receive Data Hold After RSCLKx[1] | –1.5 | | ns |
| *Switching Characteristics* | | | | |
| $t_{DFSI}$ | TFSx/RFSx Delay After TSCLKx/RSCLKx (Internally Generated TFSx/RFSx)[2] | | 3.0 | ns |
| $t_{HOFSI}$ | TFSx/RFSx Hold After TSCLKx/RSCLKx (Internally Generated TFSx/RFSx)[2] | –1.0 | | ns |
| $t_{DDTI}$ | Transmit Data Delay After TSCLKx[2] | | 3.0 | ns |
| $t_{HDTI}$ | Transmit Data Hold After TSCLKx[2] | –2.0 | | ns |
| $t_{SCLKIW}$ | TSCLKx/RSCLKx Width | 4.5 | | ns |

[1] Referenced to sample edge.
[2] Referenced to drive edge.

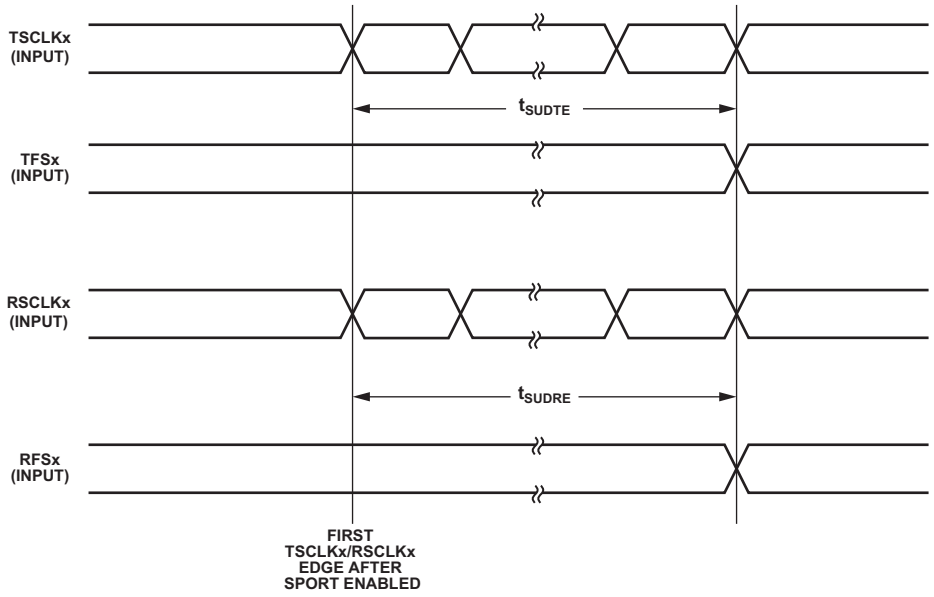



*Figure 34. Serial Port Start-Up with External Clock and Frame Sync*

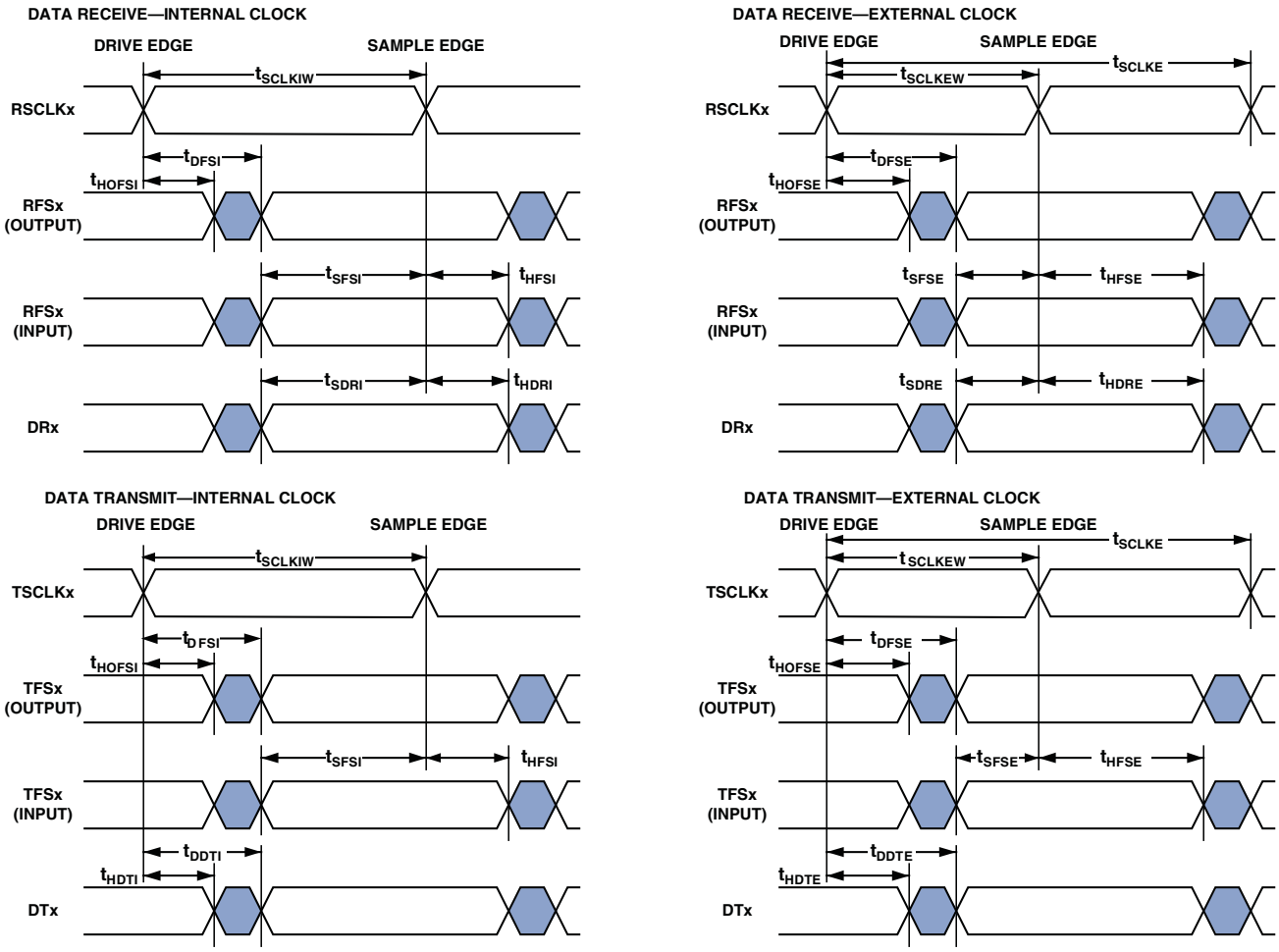



*Figure 35. Serial Ports*

**Table 43. Serial Ports—Enable and Three-State**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Switching Characteristics* | | | | |
| $t_{DTENE}$ | Data Enable Delay from External TSCLKx[1] | 0 | | ns |
| $t_{DDTTE}$ | Data Disable Delay from External TSCLKx[1, 2] | | 10.0 | ns |
| $t_{DTENI}$ | Data Enable Delay from Internal TSCLKx[1] | –2.0 | | ns |
| $t_{DDTTI}$ | Data Disable Delay from Internal TSCLKx[1, 2] | | 3.0 | ns |

[1] Referenced to drive edge.
[2] Applicable to multichannel mode only.

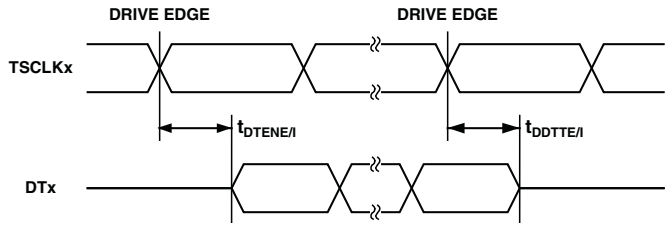



*Figure 36. Serial Ports—Enable and Three-State*

**Table 44. Serial Ports—External Late Frame Sync**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Switching Characteristics* | | | | |
| $t_{DDTLFSE}$ | Data Delay from Late External TFSx or External RFSx in multi-channel mode with MFD = 01[1, 2] | | 10.0 | ns |
| $t_{DTENLFSE}$ | Data Enable from External RFSx in multi-channel mode with MFD = 0[1, 2] | 0 | | ns |

[1] In multichannel mode, TFSx enable and TFSx valid follow $t_{DTENLFS}$ and $t_{DDTLFSE}$.
[2] If external RFS/TFS setup to RSCLK/TSCLK > $t_{SCLKE}/2$, then $t_{DDTE/I}$ and $t_{DTENE/I}$ apply; otherwise $t_{DDTLFSE}$ and $t_{DTENLFS}$ apply.

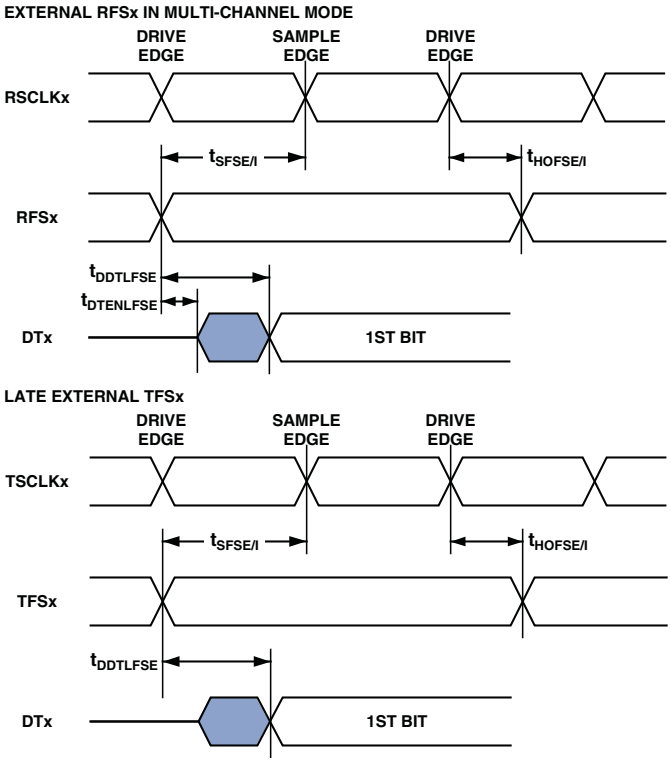



*Figure 37. Serial Ports—External Late Frame Sync*

### *Serial Peripheral Interface (SPI) Port—Master Timing*

Table 45 and Figure 38 describe SPI port master operations.

**Table 45. Serial Peripheral Interface (SPI) Port—Master Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SSPIDM}$ | Data Input Valid to SPIxSCK Edge (Data Input Setup) | 9.0 | | ns |
| $t_{HSPIDM}$ | SPIxSCK Sampling Edge to Data Input Invalid | –1.5 | | ns |
| *Switching Characteristics* | | | | |
| $t_{SDSCIM}$ | $\overline{\text{SPIxSELy}}$ Low to First SPIxSCK Edge | $2t_{SCLK}$–1.5 | | ns |
| $t_{SPICHM}$ | SPIxSCK High Period | $2t_{SCLK}$–1.5 | | ns |
| $t_{SPICLM}$ | SPIxSCK Low Period | $2t_{SCLK}$–1.5 | | ns |
| $t_{SPICLK}$ | SPIxSCK Period | $4t_{SCLK}$–1.5 | | ns |
| $t_{HDSM}$ | Last SPIxSCK Edge to $\overline{\text{SPIxSELy}}$ High | $2t_{SCLK}$–1.5 | | ns |
| $t_{SPITDM}$ | Sequential Transfer Delay | $2t_{SCLK}$–1.5 | | ns |
| $t_{DDSPIDM}$ | SPIxSCK Edge to Data Out Valid (Data Out Delay) | | 6 | ns |
| $t_{HDSPIDM}$ | SPIxSCK Edge to Data Out Invalid (Data Out Hold) | –1.0 | | ns |

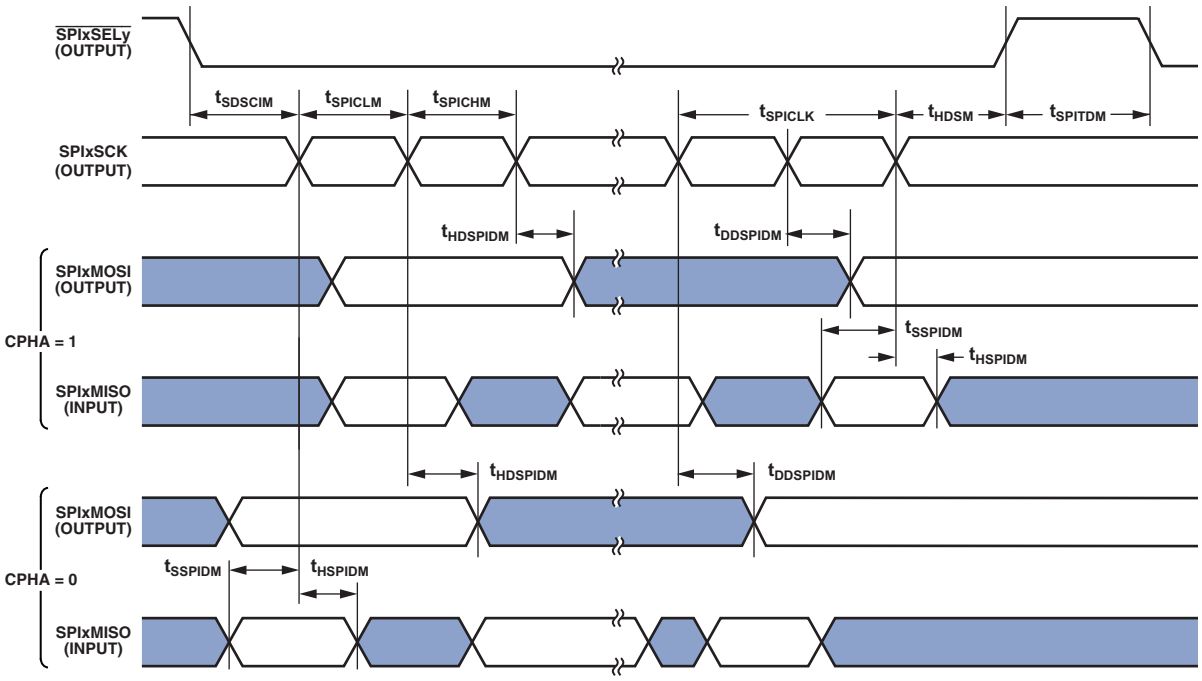



*Figure 38. Serial Peripheral Interface (SPI) Port—Master Timing*

### *Serial Peripheral Interface (SPI) Port—Slave Timing*

Table 46 and Figure 39 describe SPI port slave operations.

**Table 46. Serial Peripheral Interface (SPI) Port—Slave Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SPICHS}$ | SPIxSCK High Period | $2t_{SCLK}-1.5$ | | ns |
| $t_{SPICLS}$ | SPIxSCK Low Period | $2t_{SCLK}-1.5$ | | ns |
| $t_{SPICLK}$ | SPIxSCK Period | $4t_{SCLK}$ | | ns |
| $t_{HDS}$ | Last SPIxSCK Edge to $\overline{\text{SPIxSS}}$ Not Asserted | $2t_{SCLK}-1.5$ | | ns |
| $t_{SPITDS}$ | Sequential Transfer Delay | $2t_{SCLK}-1.5$ | | ns |
| $t_{SDSCI}$ | $\overline{\text{SPIxSS}}$ Assertion to First SPIxSCK Edge | $2t_{SCLK}-1.5$ | | ns |
| $t_{SSPID}$ | Data Input Valid to SPIxSCK Edge (Data Input Setup) | 1.6 | | ns |
| $t_{HSPID}$ | SPIxSCK Sampling Edge to Data Input Invalid | 1.6 | | ns |
| *Switching Characteristics* | | | | |
| $t_{DSOE}$ | $\overline{\text{SPIxSS}}$ Assertion to Data Out Active | 0 | 8 | ns |
| $t_{DSDHI}$ | $\overline{\text{SPIxSS}}$ Deassertion to Data High Impedance | 0 | 8 | ns |
| $t_{DDSPID}$ | SPIxSCK Edge to Data Out Valid (Data Out Delay) | | 10 | ns |
| $t_{HDSPID}$ | SPIxSCK Edge to Data Out Invalid (Data Out Hold) | 0 | | ns |

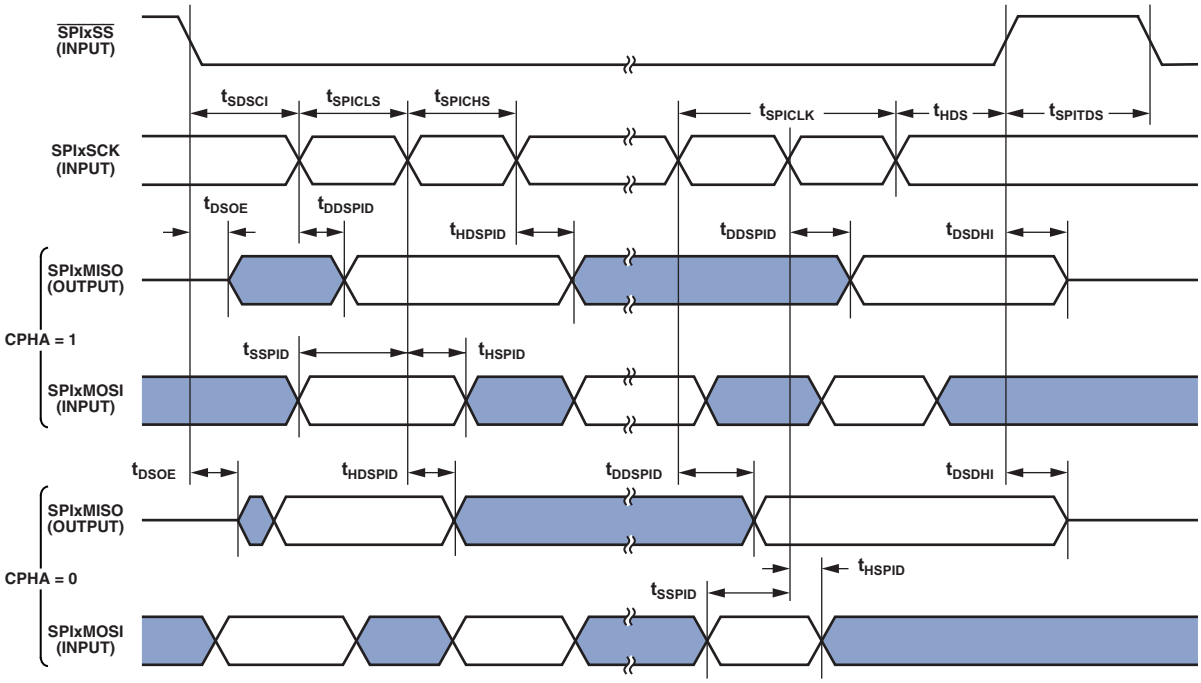



*Figure 39. Serial Peripheral Interface (SPI) Port—Slave Timing*

### *Universal Asynchronous Receiver-Transmitter (UART) Ports—Receive and Transmit Timing*

The UART ports have a maximum baud rate of SCLK/16. There is some latency between the generation of internal UART interrupts and the external data operations. These latencies are negligible at the data transmission rates for the UART. For more information, see the *ADSP-BF54x Blackfin Processor Hardware Reference*.

### *General-Purpose Port Timing*

Table 47 and Figure 40 describe general-purpose port operations.

**Table 47. General-Purpose Port Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirement* | | | | |
| $t_{WFI}$ | General-Purpose Port Pin Input Pulse Width | $t_{SCLK}$ + 1 | | ns |
| *Switching Characteristics* | | | | |
| $t_{GPOD}$ | General-Purpose Port Pin Output Delay from CLKOUT Low | –0.3 | 6 | ns |

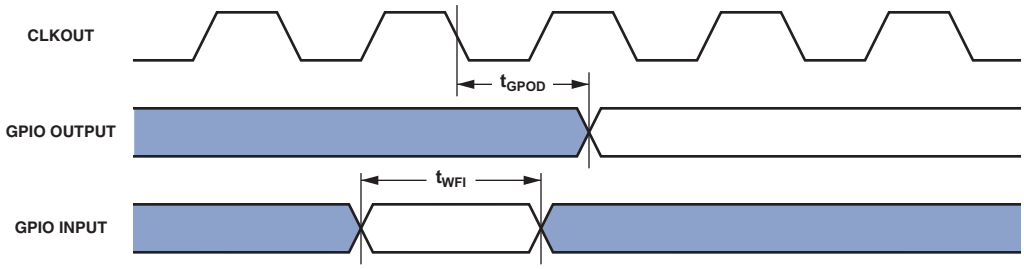



*Figure 40. General-Purpose Port Timing*

### *Timer Cycle Timing*

Table 48 and Figure 41 describe timer expired operations. The input signal is asynchronous in "width capture mode" and "external clock mode" and has an absolute maximum input frequency of ($f_{SCLK}$/2) MHz.

**Table 48. Timer Cycle Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Characteristics* | | | | |
| $t_{WL}$ | Timer Pulse Width Input Low[1] | $t_{SCLK}$ +1 | | ns |
| $t_{WH}$ | Timer Pulse Width Input High[1] | $t_{SCLK}$ +1 | | ns |
| $t_{TIS}$ | Timer Input Setup Time Before CLKOUT Low[2] | 6.5 | | ns |
| $t_{TIH}$ | Timer Input Hold Time After CLKOUT Low[2] | –1 | | ns |
| *Switching Characteristics* | | | | |
| $t_{HTO}$ | Timer Pulse Width Output | 1×$t_{SCLK}$ | ($2^{32}$ – 1)×$t_{SCLK}$ | ns |
| $t_{TOD}$ | Timer Output Delay After CLKOUT High | | 6 | ns |

[1] The minimum pulse widths apply for TMRx signals in width capture and external clock modes.
[2] Either a valid setup and hold time or a valid pulse width is sufficient. There is no need to resynchronize timer flag inputs.

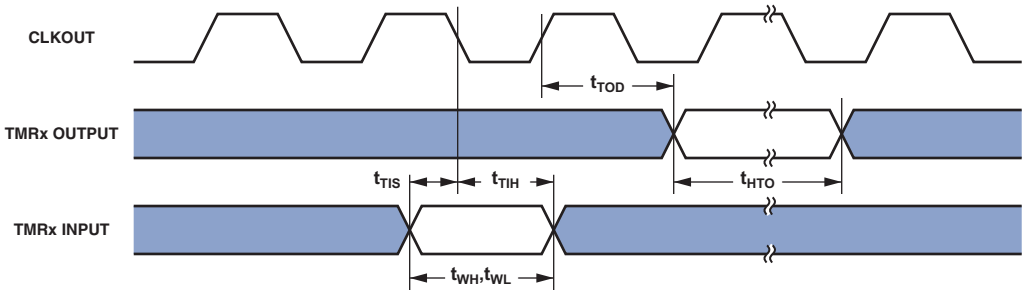



*Figure 41. Timer Cycle Timing*

### *Up/Down Counter/Rotary Encoder Timing*

Table 49 and Figure 42 describe up/down counter/rotary encoder timing.

**Table 49. Up/Down Counter/Rotary Encoder Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{WCOUNT}$ | CUD/CDG/CZM Input Pulse Width | $t_{SCLK}$ + 1 | | ns |
| $t_{CIS}$ | CUD/CDG/CZM Input Setup Time Before CLKOUT High[1] | 7.2 | | ns |
| $t_{CIH}$ | CUD/CDG/CZM Input Hold Time After CLKOUT High[1] | 0.0 | | ns |

[1] Either a valid setup and hold time or a valid pulse width is sufficient. There is no need to resynchronize counter inputs.

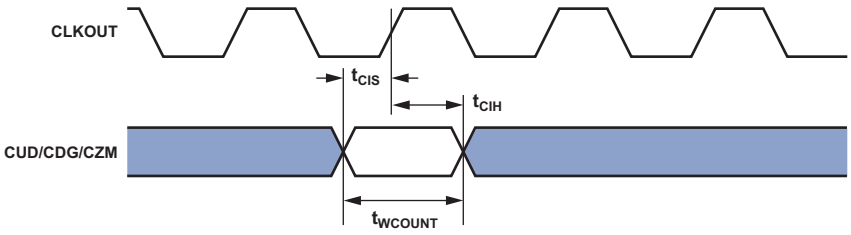



*Figure 42. Up/Down Counter/Rotary Encoder Timing*

### *SD/SDIO Controller Timing*

Table 50 and Figure 43 describe SD/SDIO controller timing.
Table 51 and Figure 44 describe SD/SDIO controller (high-speed mode) timing.

**Table 50. SD/SDIO Controller Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{ISU}$ | SD_Dx and SD_CMD Input Setup Time | 7.2 | | ns |
| $t_{IH}$ | SD_Dx and SD_CMD Input Hold Time | 2 | | ns |
| *Switching Characteristics* | | | | |
| $f_{PP}$ | SD_CLK Frequency During Data Transfer Mode[1] | 0 | 20 | MHz |
| $f_{OD}$ | SD_CLK Frequency During Identification Mode | 100[2] | 400 | kHz |
| $t_{WL}$ | SD_CLK Low Time | 15 | | ns |
| $t_{WH}$ | SD_CLK High Time | 15 | | ns |
| $t_{TLH}$ | SD_CLK Rise Time | | 10 | ns |
| $t_{THL}$ | SD_CLK Fall Time | | 10 | ns |
| $t_{ODLY}$ | SD_Dx and SD_CMD Output Delay Time During Data Transfer Mode | –1 | 14 | ns |
| $t_{ODLY}$ | SD_Dx and SD_CMD Output Delay Time During Identification Mode | –1 | 50 | ns |

[1] $t_{PP}=1/f_{PP}$
[2] Spec can be 0 kHz, meaning to stop the clock. The given minimum frequency range is for cases where a continuous clock is required.

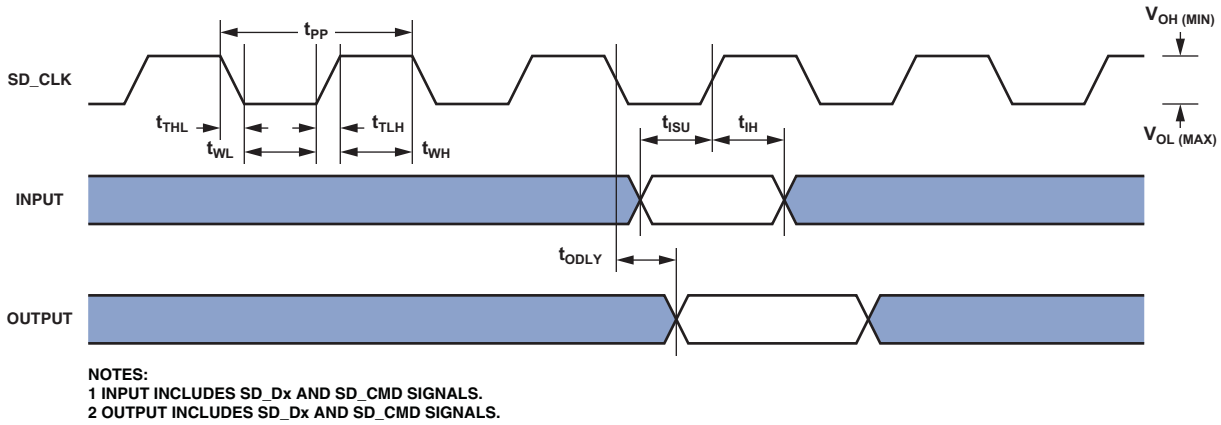



*Figure 43. SD/SDIO Controller Timing*

**Table 51. SD/SDIO Controller Timing (High Speed Mode)**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{ISU}$ | SD_Dx and SD_CMD Input Setup Time | 7.2 | | ns |
| $t_{IH}$ | SD_Dx and SD_CMD Input Hold Time | 2 | | ns |
| *Switching Characteristics* | | | | |
| $f_{PP}$ | SD_CLK Frequency During Data Transfer Mode[1] | 0 | 40 | MHz |
| $t_{WL}$ | SD_CLK Low Time | 9.5 | | ns |
| $t_{WH}$ | SD_CLK High Time | 9.5 | | ns |
| $t_{TLH}$ | SD_CLK Rise Time | | 3 | ns |
| $t_{THL}$ | SD_CLK Fall Time | | 3 | ns |
| $t_{ODLY}$ | SD_Dx and SD_CMD Output Delay Time During Data Transfer Mode | | 2 | ns |
| $t_{OH}$ | SD_Dx and SD_CMD Output Hold Time | 2.5 | | ns |

[1] $t_{PP}=1/f_{PP}$

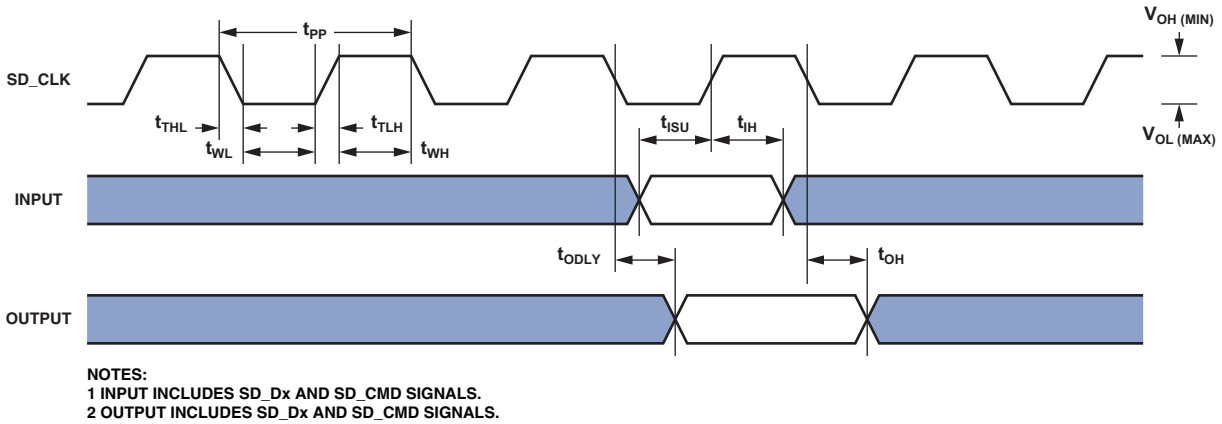



*Figure 44. SD/SDIO Controller Timing (High Speed Mode)*

### *MXVR Timing*

Table 52 and Table 53 describe the MXVR timing requirements.
Figure 5 illustrates the MOST connection.

**Table 52. MXVR Timing—MXI Center Frequency Requirements**

| Parameter | | Fs = 38 kHz | Fs = 44.1 kHz | Fs = 48 kHz | Unit |
|---|---|---|---|---|---|
| $f_{MXI_256}$ | MXI Center Frequency (256 Fs) | 9.728 | 11.2896 | 12.288 | MHz |
| $f_{MXI_384}$ | MXI Center Frequency (384 Fs) | 14.592 | 16.9344 | 18.432 | MHz |
| $f_{MXI_512}$ | MXI Center Frequency (512 Fs) | 19.456 | 22.5792 | 24.576 | MHz |
| $f_{MXI_1024}$ | MXI Center Frequency (1024 Fs) | 38.912 | 45.1584 | 49.152 | MHz |

**Table 53. MXVR Timing— MXI Clock Requirements**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $FS_{MXI}$ | MXI Clock Frequency Stability | –50 | +50 | ppm |
| $FT_{MXI}$ | MXI Frequency Tolerance Over Temperature | –300 | +300 | ppm |
| $DC_{MXI}$ | MXI Clock Duty Cycle | +40 | +60 | % |

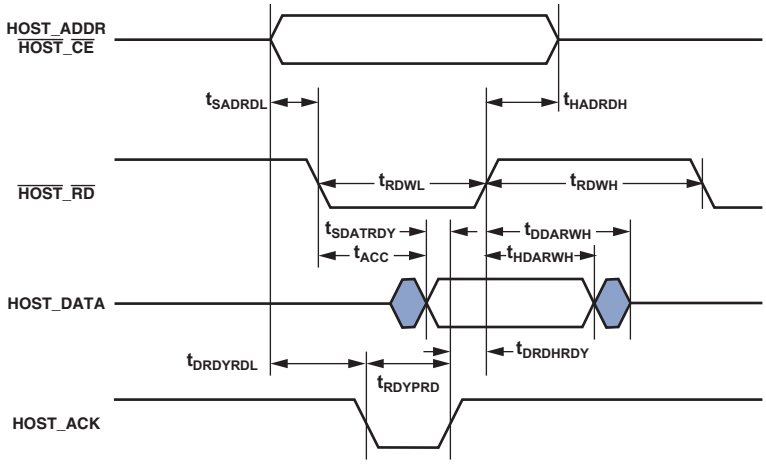
### *HOSTDP A/C Timing-Host Read Cycle*

Table 54 and Figure 45 describe the HOSTDP A/C host read cycle timing requirements.

**Table 54. Host Read Cycle Timing Requirements**

| Parameter | | Min | Max | Units |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SADRDL}$ | HOST_ADDR and $\overline{\text{HOST_CE}}$ Setup Before $\overline{\text{HOST_RD}}$ Falling Edge | 4 | | ns |
| $t_{HADRDH}$ | HOST_ADDR and $\overline{\text{HOST_CE}}$ Hold After $\overline{\text{HOST_RD}}$ Rising Edge | 2.5 | | ns |
| $t_{RDWL}$ | $\overline{\text{HOST_RD}}$ Pulse Width Low (ACK Mode) | $t_{DRDYRDL} + t_{RDYPRD} + t_{DRDHRDY}$ | | ns |
| $t_{RDWL}$ | $\overline{\text{HOST_RD}}$ Pulse Width Low (INT Mode) | $1.5 \times t_{SCLK} + 8.7$ | | ns |
| $t_{RDWH}$ | $\overline{\text{HOST_RD}}$ Pulse Width High or Time Between $\overline{\text{HOST_RD}}$ Rising Edge and $\overline{\text{HOST_WR}}$ Falling Edge | $2 \times t_{SCLK}$ | | ns |
| $t_{DRDHRDY}$ | $\overline{\text{HOST_RD}}$ Rising Edge Delay After HOST_ACK Rising Edge (ACK Mode) | 0 | | ns |
| *Switching Characteristics* | | | | |
| $t_{SDATRDY}$ | HOST_D15–0 Valid Prior HOST_ACK Rising Edge (ACK Mode) | $t_{SCLK} - 4.0$ | | ns |
| $t_{DRDYRDL}$ | HOST_ACK Falling Edge After $\overline{\text{HOST_CE}}$ (ACK Mode) | | 11.25 | ns |
| $t_{RDYPRD}$ | HOST_ACK Low Pulse-Width for Read Access (ACK Mode) | | NM[1] | ns |
| $t_{DDARWH}$ | HOST_D15–0 Disable After $\overline{\text{HOST_RD}}$ | | 8.0 | ns |
| $t_{ACC}$ | HOST_D15–0 Valid After $\overline{\text{HOST_RD}}$ Falling Edge (INT Mode) | | $1.5 \times t_{SCLK}$ | ns |
| $t_{HDARWH}$ | HOST_D15–0 Hold After $\overline{\text{HOST_RD}}$ Rising Edge | 1.0 | | ns |

[1] NM (Not Measured) — This parameter is based on $t_{SCLK}$. It is not measured because the number of SCLK cycles for which HOST_ACK remains low depends on the Host DMA FIFO status. This is system design dependent.




In Figure 45, HOST_DATA is HOST_D0–D15.

*Figure 45. HOSTDP A/C—Host Read Cycle*

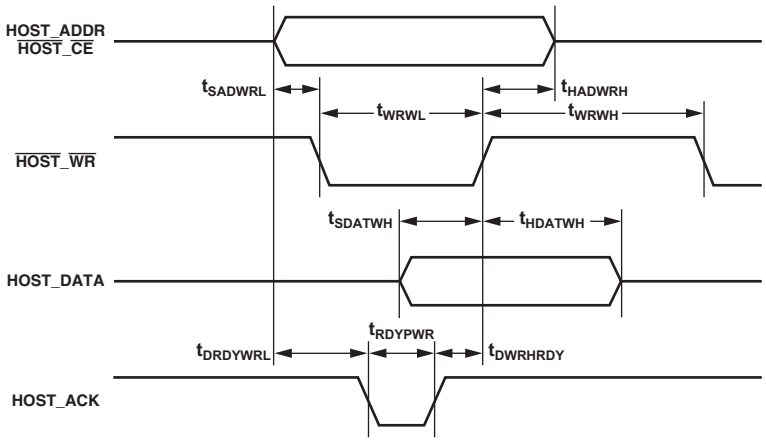
### *HOSTDP A/C Timing-Host Write Cycle*

Table 55 and Figure 46 describe the HOSTDP A/C host write cycle timing requirements.

**Table 55. Host Write Cycle Timing Requirements**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SADWRL}$ | HOST_ADDR/$\overline{\text{HOST_CE}}$ Setup Before $\overline{\text{HOST_WR}}$ Falling Edge | 4 | | ns |
| $t_{HADWRH}$ | HOST_ADDR/$\overline{\text{HOST_CE}}$ Hold After $\overline{\text{HOST_WR}}$ Rising Edge | 2.5 | | ns |
| $t_{WRWL}$ | $\overline{\text{HOST_WR}}$ Pulse Width Low (ACK Mode) | $t_{DRDYWRL} + t_{RDYPRD} + t_{DWRHRDY}$ | | ns |
| | $\overline{\text{HOST_WR}}$ Pulse Width Low (INT Mode) | $1.5 \times t_{SCLK} + 8.7$ | | ns |
| $t_{WRWH}$ | $\overline{\text{HOST_WR}}$ Pulse Width High or Time Between $\overline{\text{HOST_WR}}$ Rising Edge and $\overline{\text{HOST_RD}}$ Falling Edge | $2 \times t_{SCLK}$ | | ns |
| $t_{DWRHRDY}$ | $\overline{\text{HOST_WR}}$ Rising Edge Delay After HOST_ACK Rising Edge (ACK Mode) | 0 | | ns |
| $t_{HDATWH}$ | HOST_D15–0 Hold After $\overline{\text{HOST_WR}}$ Rising Edge | 2.5 | | ns |
| $t_{SDATWH}$ | HOST_D15–0 Setup Before $\overline{\text{HOST_WR}}$ Rising Edge | 3.5 | | ns |
| *Switching Characteristics* | | | | |
| $t_{DRDYWRL}$ | HOST_ACK Falling Edge After $\overline{\text{HOST_CE}}$ Asserted (ACK Mode) | | 11.25 | ns |
| $t_{RDYPWR}$ | HOST_ACK Low Pulse-Width for Write Access (ACK Mode) | | NM[1] | ns |

[1] NM (not measured)—This parameter is based on $t_{SCLK}$. It is not measured because the number of SCLK cycles for which HOST_ACK remains low depends on the Host DMA FIFO status. This is system design dependent.




In Figure 46, HOST_DATA is HOST_D0–D15.

*Figure 46. HOSTDP A/C- Host Write Cycle*

### *ATA/ATAPI-6 Interface Timing*

The following tables and figures specify ATAPI timing parameters. For detailed parameter descriptions, refer to the ATAPI specification (ANSI INCITS 361-2002). Table 58 to Table 61 include ATAPI timing parameter equations. System designers should use these equations along with the parameters provided in Table 56 and Table 57. ATAPI timing control registers should be programmed such that ANSI INCITS 361-2002 specifications are met for the desired transfer type and mode.

**Table 56. ATA/ATAPI-6 Timing Parameters**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| $t_{SK1}$ | Difference in output delay after CLKOUT for ATAPI output pins[1] | | 6 | ns |
| $t_{OD}$ | Output delay after CLKOUT for outputs[1] | | 12 | ns |
| $t_{SUD}$ | ATAPI_D0-15 or ATAPI_D0-15A Setup Before CLKOUT | 6 | | ns |
| $t_{SUI}$ | $\overline{\text{ATAPI_IORDY}}$ Setup Before CLKOUT | 6 | | ns |
| $t_{SUDU}$ | ATAPI_D0-15 or ATAPI_D0-15A Setup Before $\overline{\text{ATAPI_IORDY}}$ (UDMA-in only) | 2 | | ns |
| $t_{HDU}$ | ATAPI_D0-15 or ATAPI_D0-15A Hold After $\overline{\text{ATAPI_IORDY}}$ (UDMA-in only) | 2.6 | | ns |

[1] ATAPI output pins include $\overline{\text{ATAPI_CS0}}$, $\overline{\text{ATAPI_CS1}}$, A1-3, $\overline{\text{ATAPI_DIOR}}$, $\overline{\text{ATAPI_DIOW}}$, $\overline{\text{ATAPI_DMACK}}$, ATAPI_D0-15, ATAPI_A0-2A, and ATAPI_D0-15A.

**Table 57. ATA/ATAPI-6 System Timing Parameters**

| Parameter | | Source |
|---|---|---|
| $t_{SK2}$ | Maximum difference in board propagation delay between any 2 ATAPI output pins[1] | System Design |
| $t_{BD}$ | Maximum board propagation delay. | System Design |
| $t_{SK3}$ | Maximum difference in board propagation delay during a read between $\overline{\text{ATAPI_IORDY}}$ and ATAPI_D0-15/ATAPI_D0-15A. | System Design |
| $t_{SK4}$ | Maximum difference in ATAPI cable propagation delay between output pin group A and output pin group B[2] | ATAPI Cable Specification |
| $t_{CDD}$ | ATAPI cable propagation delay for ATAPI_D0-15 and ATAPI_D0-15A signals. | ATAPI Cable Specification |
| $t_{CDC}$ | ATAPI cable propagation delay for $\overline{\text{ATAPI_DIOR}}$, $\overline{\text{ATAPI_DIOW}}$, $\overline{\text{ATAPI_IORDY}}$, and $\overline{\text{ATAPI_DMACK}}$ signals. | ATAPI Cable Specification |

[1] ATAPI output pins include $\overline{\text{ATAPI_CS0}}$, $\overline{\text{ATAPI_CS1}}$, A1-3, $\overline{\text{ATAPI_DIOR}}$, $\overline{\text{ATAPI_DIOW}}$, $\overline{\text{ATAPI_DMACK}}$, ATAPI_D0-15, ATAPI_A0-2A, and ATAPI_D0-15A.
[2] Output pin group A includes $\overline{\text{ATAPI_DIOR}}$, $\overline{\text{ATAPI_DIOW}}$, and $\overline{\text{ATAPI_DMACK}}$. Output pin group B includes $\overline{\text{ATAPI_CS0}}$, $\overline{\text{ATAPI_CS1}}$, A1-3, ATAPI_D0-15, ATAPI_A0-2A, and ATAPI_D0-15A.

### Register and PIO

Table 58 and Figure 47 describe the ATAPI register and the PIO data transfer timing.

**Table 58. ATAPI Register and PIO Data Transfer Timing**

| ATAPI Parameter/Description | | ATAPI_REG/PIO_TIM_x Timing Register Setting[1] | Timing Equation |
|---|---|---|---|
| $t_0$ | Cycle time | T2_PIO, TEOC_PIO | $(T2_PIO + TEOC_PIO) \times t_{SCLK}$ |
| $t_1$ | ATAPI_ADDR valid to $\overline{ATAPI_DIOR}$/$\overline{ATAPI_DIOW}$ setup | T1 | $T1 \times t_{SCLK} - (t_{SK1} + t_{SK2} + t_{SK4})$ |
| $t_2$ | $\overline{ATAPI_DIOR}$/$\overline{ATAPI_DIOW}$ pulse width | T2_PIO | $T2_PIO \times t_{SCLK}$ |
| $t_{2i}$ | $\overline{ATAPI_DIOR}$/$\overline{ATAPI_DIOW}$ recovery time | TEOC_PIO | $TEOC_PIO \times t_{SCLK}$ |
| $t_3$ | $\overline{ATAPI_DIOW}$ data setup | T2_PIO | $T2_PIO \times t_{SCLK} - (t_{SK1} + t_{SK2} + t_{SK4})$ |
| $t_4$ | $\overline{ATAPI_DIOW}$ data hold | T4 | $T4 \times t_{SCLK} - (t_{SK1} + t_{SK2} + t_{SK4})$ |
| $t_5$ | $\overline{ATAPI_DIOR}$ data setup | N/A | $t_{OD} + t_{SUD} + 2 \times t_{BD} + t_{CDD} + t_{CDC}$ |
| $t_6$ | $\overline{ATAPI_DIOR}$ data hold | N/A | 0 |
| $t_9$ | $\overline{ATAPI_DIOR}$/$\overline{ATAPI_DIOW}$ to ATAPI_ADDR valid hold | TEOC_PIO | $TEOC_PIO \times t_{SCLK} - (t_{SK1} + t_{SK2} + t_{SK4})$ |
| $t_A$ | $\overline{ATAPI_IORDY}$ setup time | T2_PIO | $T2_PIO \times t_{SCLK} - (t_{OD} + t_{SUI} + 2 \times t_{CDC} + 2 \times t_{BD})$ |

[1] ATAPI timing register setting should be programmed with a value that guarantees parameter compliance with the ATA ANSI specification for the ATA device mode of operation.

Figure 47 displays the REG and PIO data transfer timing. Note that ATAPI_ADDR pins include A1-3, $\overline{ATAPI_CS0}$, and $\overline{ATAPI_CS1}$. Alternate ATAPI port ATAPI _ADDR pins include ATAPI_A0A, ATAPI_A1A, ATAPI_A2A, $\overline{ATAPI_CS0}$, and $\overline{ATAPI_CS1}$. Note that an alternate ATAPI_D0-15 port bus is ATAPI_D0-15A

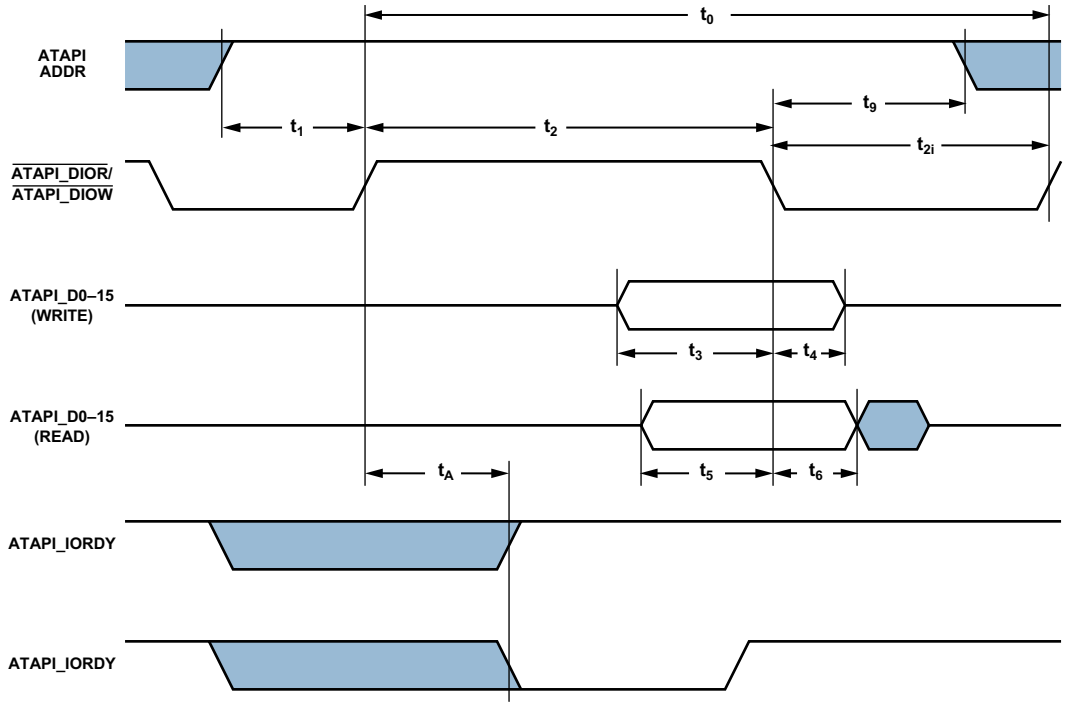



*Figure 47. REG and PIO Data Transfer Timing[1]*

[1] This material is adapted from ATAPI-6 (INCITS 361-2002[R2007] and is used with permission of the American National Standards Institute (ANSI) on behalf of the Information Technology Industry Council ("ITIC"). Copies of ATAPI-6 (INCITS 361-2002[R2007] can be purchased from ANSI.

### ATAPI Multiword DMA Transfer Timing

Table 59 and Figure 48 through Figure 51 describe the ATAPI multiword DMA transfer timing.

**Table 59. ATAPI Multiword DMA Transfer Timing**

| ATAPI Parameter/Description | | ATAPI_MULTI_TIM_x Timing Register Setting[1] | Timing Equation |
|---|---|---|---|
| $t_0$ | Cycle time | TD, TK | $(TD + TK) \times t_{SCLK}$ |
| $t_D$ | $\overline{ATAPI_DIOR}$/$\overline{ATAPI_DIOW}$ asserted Pulse Width | TD | $TD \times t_{SCLK}$ |
| $t_F$ | $\overline{ATAPI_DIOR}$ data hold | N/A | 0 |
| $t_{G(write)}$ | $\overline{ATAPI_DIOW}$ data setup | TD | $TD \times t_{SCLK} - (t_{SK1} + t_{SK2} + t_{SK4})$ |
| $t_{G(read)}$ | $\overline{ATAPI_DIOR}$ data setup | TD | $t_{OD} + t_{SUD} + 2 \times t_{BD} + t_{CDD} + t_{CDC}$ |
| $t_H$ | $\overline{ATAPI_DIOW}$ data hold | TK | $TK \times t_{SCLK} - (t_{SK1} + t_{SK2} + t_{SK4})$ |
| $t_I$ | $\overline{ATAPI_DMACK}$ to $\overline{ATAPI_DIOR}$/$\overline{ATAPI_DIOW}$ setup | TM | $TM \times t_{SCLK} - (t_{SK1} + t_{SK2} + t_{SK4})$ |
| $t_J$ | $\overline{ATAPI_DIOR}$/$\overline{ATAPI_DIOW}$ to $\overline{ATAPI_DMACK}$ hold | TK, TEOC_MDMA | $(TK + TEOC_MDMA) \times t_{SCLK} - (t_{SK1} + t_{SK2} + t_{SK4})$ |
| $t_{KR}$ | $\overline{ATAPI_DIOR}$ negated pulse width | TKR | $TKR \times t_{SCLK}$ |
| $t_{KW}$ | $\overline{ATAPI_DIOW}$ negated pulse width | TKW | $TKW \times t_{SCLK}$ |
| $t_{LR}$ | $\overline{ATAPI_DIOR}$ to $\overline{ATAPI_DMARQ}$ delay | N/A | $(TD + TK) \times t_{SCLK} - (t_{OD} + 2 \times t_{BD} + 2 \times t_{CDC})$ |
| $t_M$ | $\overline{ATAPI_CS0\text{-}1}$ valid to $\overline{ATAPI_DIOR}$/$\overline{ATAPI_DIOW}$ | TM | $TM \times t_{SCLK} - (t_{SK1} + t_{SK2} + t_{SK4})$ |
| $t_N$ | $\overline{ATAPI_CS0\text{-}1}$ hold | TK, TEOC_MDMA | $(TK + TEOC_MDMA) \times t_{SCLK} - (t_{SK1} + t_{SK2} + t_{SK4})$ |

[1] ATAPI timing register setting should be programmed with a value that guarantees parameter compliance with the ATA ANSI specification for an ATA device mode of operation.

Figure 48 displays the initiation of a multiword DMA data burst. Note that an alternate ATAPI_D0-15 port bus is ATAPI_D0-15A.

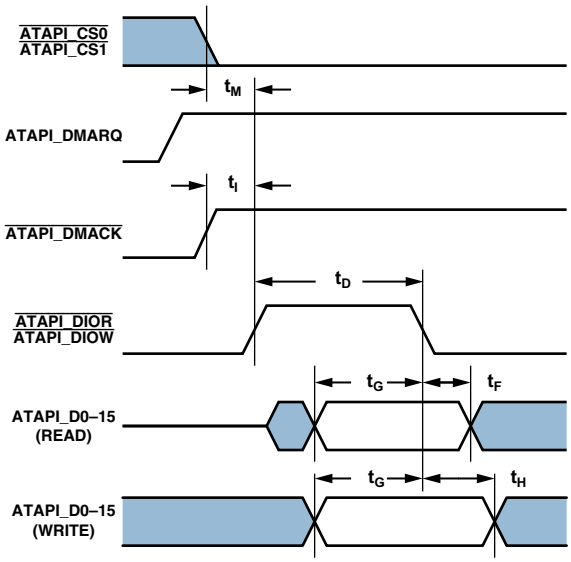



*Figure 48. Initiating a Multiword DMA Data Burst*[1]

[1] This material is adapted from ATAPI-6 (INCITS 361-2002[R2007] and is used with permission of the American National Standards Institute (ANSI) on behalf of the Information Technology Industry Council ("ITIC"). Copies of ATAPI-6 (INCITS 361-2002[R2007] can be purchased from ANSI.

Figure 49 displays a sustained multiword DMA data burst.

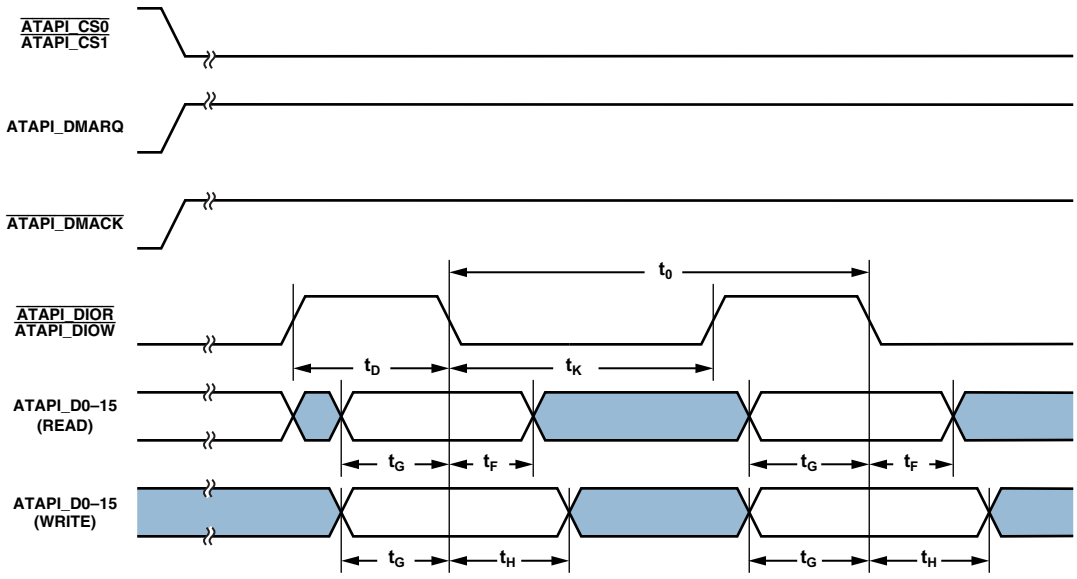



*Figure 49. Sustained Multiword DMA Data Burst*[1]

[1] This material is adapted from ATAPI-6 (INCITS 361-2002[R2007] and is used with permission of the American National Standards Institute (ANSI) on behalf of the Information Technology Industry Council ("ITIC"). Copies of ATAPI-6 (INCITS 361-2002[R2007] can be purchased from ANSI.

Figure 50 displays a device terminating a multiword DMA data burst.

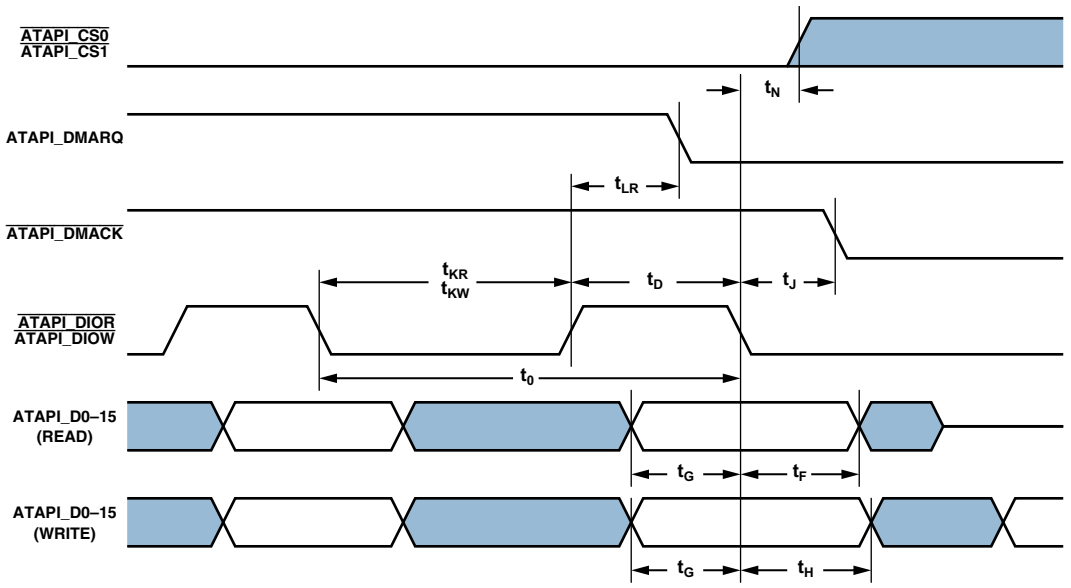



*Figure 50. Device Terminating a Multiword DMA Data Burst*[1]

[1] This material is adapted from ATAPI-6 (INCITS 361-2002[R2007] and is used with permission of the American National Standards Institute (ANSI) on behalf of the Information Technology Industry Council ("ITIC"). Copies of ATAPI-6 (INCITS 361-2002[R2007] can be purchased from ANSI.

Figure 51 displays a host terminating a multiword DMA data burst.

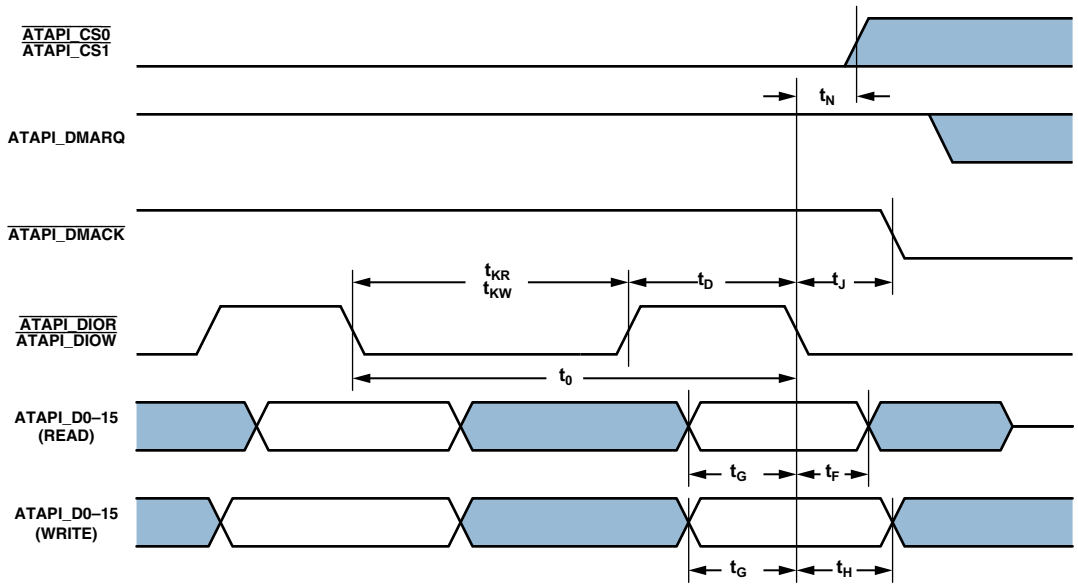



*Figure 51. Host Terminating a Multiword DMA Data Burst*[1]

[1] This material is adapted from ATAPI-6 (INCITS 361-2002[R2007] and is used with permission of the American National Standards Institute (ANSI) on behalf of the Information Technology Industry Council ("ITIC"). Copies of ATAPI-6 (INCITS 361-2002[R2007] can be purchased from ANSI.

### ATAPI Ultra DMA Data-In Transfer Timing

Table 60 and Figure 52 through Figure 55 describe the ATAPI ultra DMA data-in data transfer timing.

**Table 60. ATAPI Ultra DMA Data-In Transfer Timing**

| ATAPI Parameter | | ATAPI_ULTRA_TIM_x Timing Register Setting[1] | Timing Equation |
|---|---|---|---|
| $t_{DS}$ | Data setup time at host | N/A | $T_{SK3} + t_{SUDU}$ |
| $t_{DH}$ | Data hold time at host | N/A | $T_{SK3} + t_{HDU}$ |
| $t_{CVS}$ | CRC word valid setup time at host | TDVS | $TDVS \times t_{SCLK} - (t_{SK1} + t_{SK2})$ |
| $t_{CVH}$ | CRC word valid hold time at host | TACK | $TACK \times t_{SCLK} - (t_{SK1} + t_{SK2})$ |
| $t_{LI}$ | Limited interlock time | N/A | $2 \times t_{BD} + 2 \times t_{SCLK} + t_{OD}$ |
| $t_{MLI}$ | Interlock time with minimum | TZAH, TCVS | $(TZAH + TCVS) \times t_{SCLK} - (4 \times t_{BD} + 4 \times t_{SCLK} + 2 \times t_{OD})$ |
| $t_{AZ}$ | Maximum time allowed for output drivers to release | N/A | 0 |
| $t_{ZAH}$ | Minimum delay time required for output | TZAH | $2 \times t_{SCLK} + TZAH \times t_{SCLK} + t_{SCLK}$ |
| $t_{ENV}$[2] | $\overline{\text{ATAPI_DMACK}}$ to $\overline{\text{ATAPI_DIOR}}/\overline{\text{DIOW}}$ | TENV | $(TENV \times t_{SCLK})$ +/- $(t_{SK1} + t_{SK2})$ |
| $t_{RP}$ | $\overline{\text{ATAPI_DMACK}}$ to $\overline{\text{ATAPI_DIOR}}/\overline{\text{DIOW}}$ | TRP | $TRP \times t_{SCLK} - (t_{SK1} + t_{SK2} + t_{SK4})$ |
| $t_{ACK}$ | Setup and hold times for $\overline{\text{ATAPI_DMACK}}$ | TACK | $TACK \times t_{SCLK} - (t_{SK1} + t_{SK2})$ |

[1] ATAPI Timing Register Setting should be programmed with a value that guarantees parameter compliance with the ATA ANSI specification for ATA device mode of operation.
[2] This timing equation can be used to calculate both the minimum and maximum $t_{ENV}$.

Figure 52 displays the initiation of an ultra DMA data-in burst. Note that an alternate ATAPI_D0-15 port bus is ATAPI_D0-15A.

Also note that ATAPI_ADDR pins include A1-3, $\overline{\text{ATAPI_CS0}}$, and $\overline{\text{ATAPI_CS1}}$. Alternate ATAPI port ATAPI _ADDR pins include ATAPI_A0A, ATAPI_A1A, ATAPI_A2A, $\overline{\text{ATAPI_CS0}}$, and $\overline{\text{ATAPI_CS1}}$.

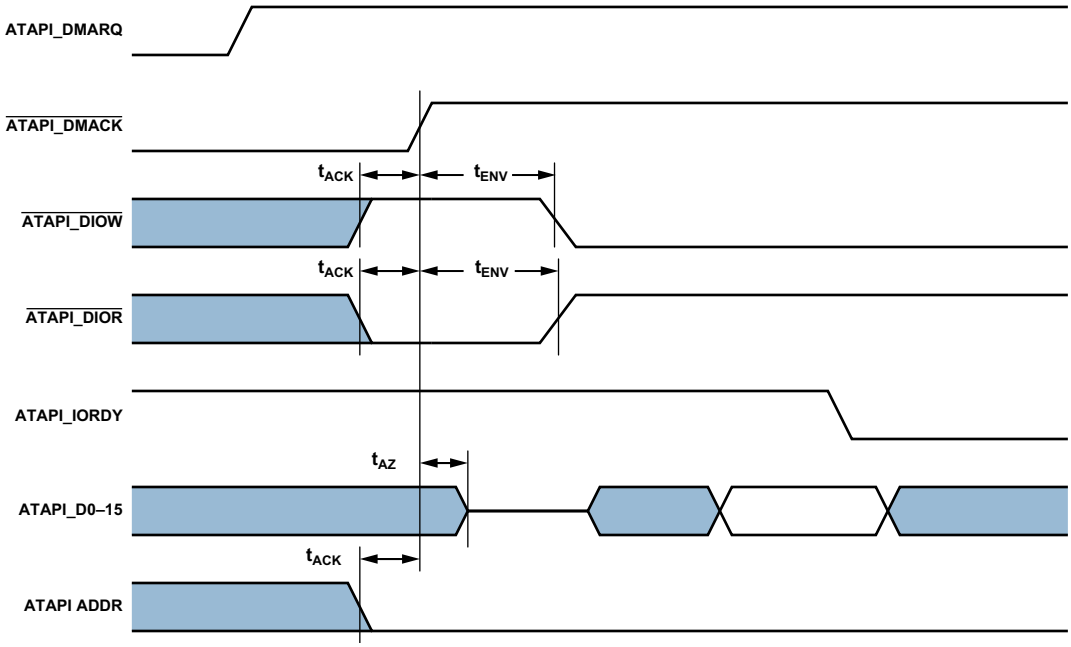



*Figure 52. Initiating an Ultra DMA Data-In Burst*[1]

[1] This material is adapted from ATAPI-6 (INCITS 361-2002[R2007] and is used with permission of the American National Standards Institute (ANSI) on behalf of the Information Technology Industry Council ("ITIC"). Copies of ATAPI-6 (INCITS 361-2002[R2007] can be purchased from ANSI.

Figure 53 displays a sustained ultra DMA data-in burst. Note that an alternate ATAPI_D0-15 port bus is ATAPI_D0-15A.

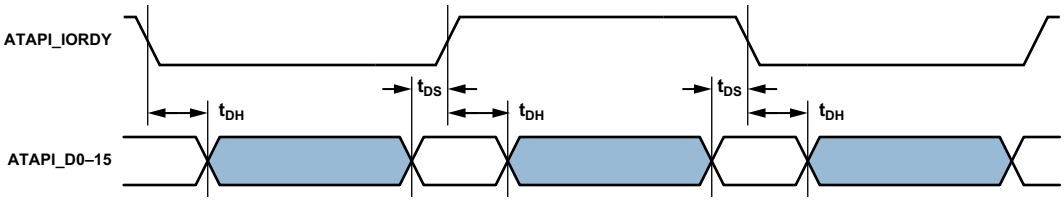



*Figure 53. Sustained Ultra DMA Data-In Burst*[1]

[1] This material is adapted from ATAPI-6 (INCITS 361-2002[R2007] and is used with permission of the American National Standards Institute (ANSI) on behalf of the Information Technology Industry Council ("ITIC"). Copies of ATAPI-6 (INCITS 361-2002[R2007] can be purchased from ANSI.

Figure 54 displays a device terminating an ultra DMA data-in burst.

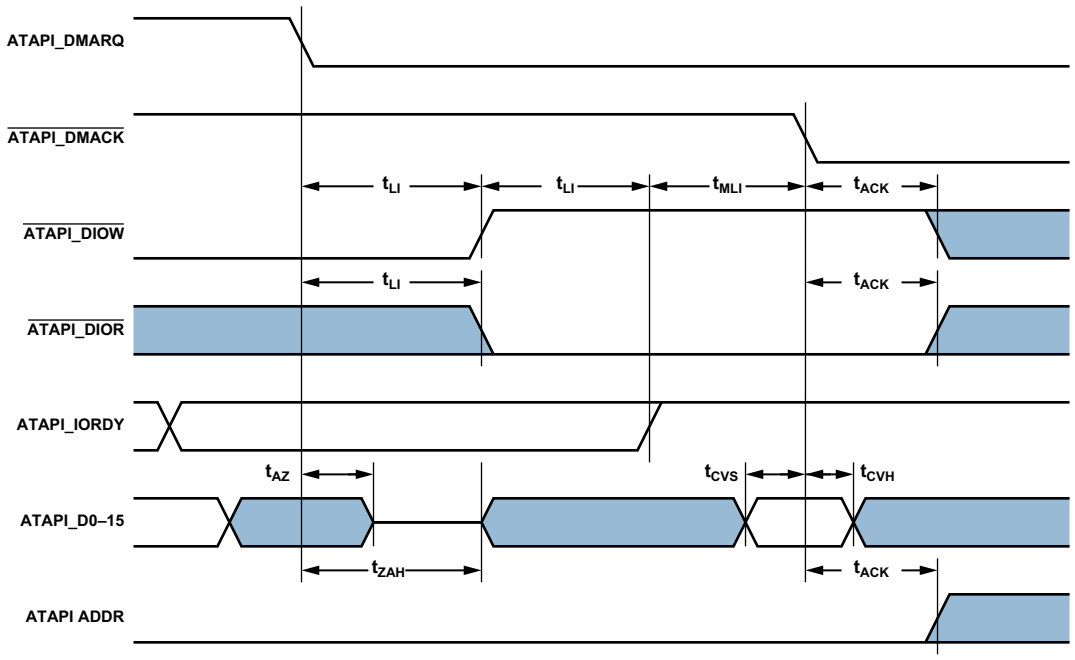



*Figure 54. Device Terminating an Ultra DMA Data-In Burst*[1]

[1] This material is adapted from ATAPI-6 (INCITS 361-2002[R2007] and is used with permission of the American National Standards Institute (ANSI) on behalf of the Information Technology Industry Council ("ITIC"). Copies of ATAPI-6 (INCITS 361-2002[R2007] can be purchased from ANSI.

Figure 55 displays a host terminating an ultra DMA data-in burst.

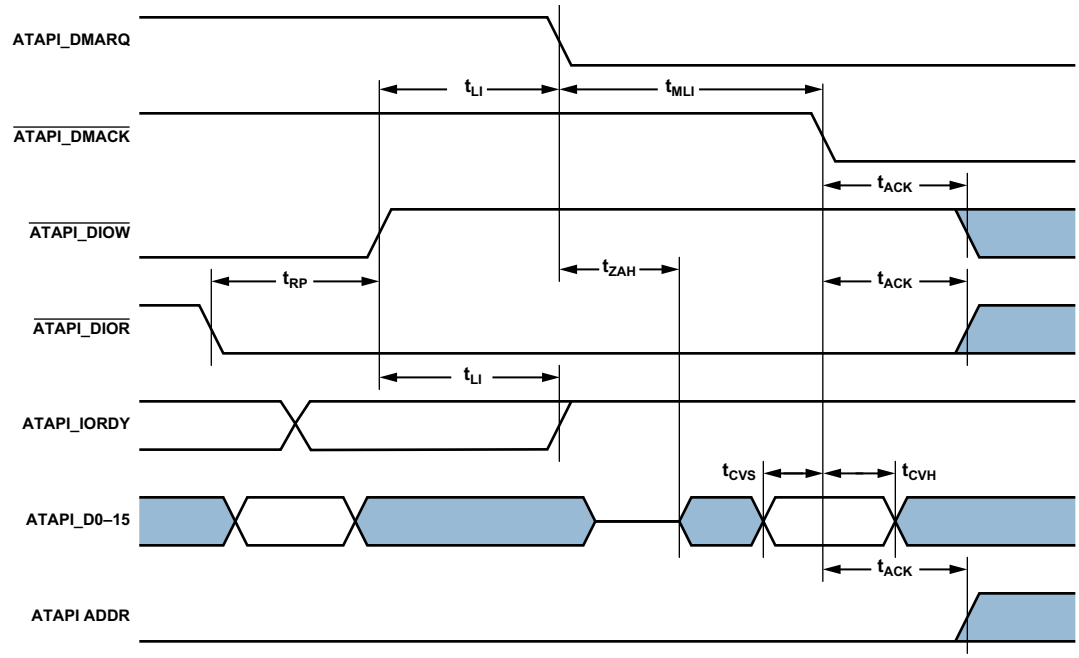



*Figure 55. Host Terminating an Ultra DMA Data-In Burst*[1]

[1] This material is adapted from ATAPI-6 (INCITS 361-2002[R2007] and is used with permission of the American National Standards Institute (ANSI) on behalf of the Information Technology Industry Council ("ITIC"). Copies of ATAPI-6 (INCITS 361-2002[R2007] can be purchased from ANSI.

### ATAPI Ultra DMA Data-Out Transfer Timing

Table 61 and Figure 56 through Figure 59 describes the ATAPI ultra DMA data-out transfer timing.

**Table 61. ATAPI Ultra DMA Data-Out Transfer Timing**

| ATAPI Parameter | | ATAPI_ULTRA_TIM_x Timing Register Setting[1] | Timing Equation |
|---|---|---|---|
| $t_{CYC}$[2] | Cycle time | TDVS, TCYC_TDVS | $(TDVS + TCYC_TDVS) \times t_{SCLK}$ |
| $t_{2CYC}$ | Two cycle time | TDVS, TCYC_TDVS | $2 \times (TDVS + TCYC_TDVS) \times t_{SCLK}$ |
| $t_{DVS}$ | Data valid setup time at sender | TDVS | $TDVS \times t_{SCLK} - (t_{SK1} + t_{SK2})$ |
| $t_{DVH}$ | Data valid hold time at sender | TCYC_TDVS | $TCYC_TDVS \times t_{SCLK} - (t_{SK1} + t_{SK2})$ |
| $t_{CVS}$ | CRC word valid setup time at host | TDVS | $TDVS \times t_{SCLK} - (t_{SK1} + t_{SK2})$ |
| $t_{CVH}$ | CRC word valid hold time at host | TACK | $TACK \times t_{SCLK} - (t_{SK1} + t_{SK2})$ |
| $t_{DZFS}$ | Time from data output released-to-driving to first strobe timing | TDVS | $TDVS \times t_{SCLK} - (t_{SK1} + t_{SK2})$ |
| $t_{LI}$ | Limited interlock time | N/A | $2 \times t_{BD} + 2 \times t_{SCLK} + t_{OD}$ |
| $t_{MLI}$ | Interlock time with minimum | TMLI | $TMLI \times t_{SCLK} - (t_{SK1} + t_{SK2})$ |
| $t_{ENV}$[3] | $\overline{ATAPI_DMACK}$ to $\overline{ATAPI_DIOR}$/$\overline{DIOW}$ | TENV | $(TENV \times t_{SCLK}) +/- (t_{SK1} + t_{SK2})$ |
| $t_{RFS}$ | Ready to final strobe time | N/A | $2 \times t_{BD} + 2 \times t_{SCLK} + t_{OD}$ |
| $t_{ACK}$ | Setup and Hold time for $\overline{ATAPI_DMACK}$ | TACK | $TACK \times t_{SCLK} - (t_{SK1} + t_{SK2})$ |
| $t_{SS}$ | Time from STROBE edge to assertion of $\overline{ATAPI_DIOW}$ | TSS | $TSS \times t_{SCLK} - (t_{SK1} + t_{SK2})$ |

[1] ATAPI Timing Register Setting should be programmed with a value that guarantees parameter compliance with the ATA ANSI specification for ATA device mode of operation.
[2] ATA/ATAPI-6 compliant functionality with limited speed.
[3] This timing equation can be used to calculate both the minimum and maximum $t_{ENV}$.

Figure 56 displays the initiation of an ultra DMA data-out burst. Note that an alternate ATAPI_D0-15 port bus is ATAPI_D0-15A.

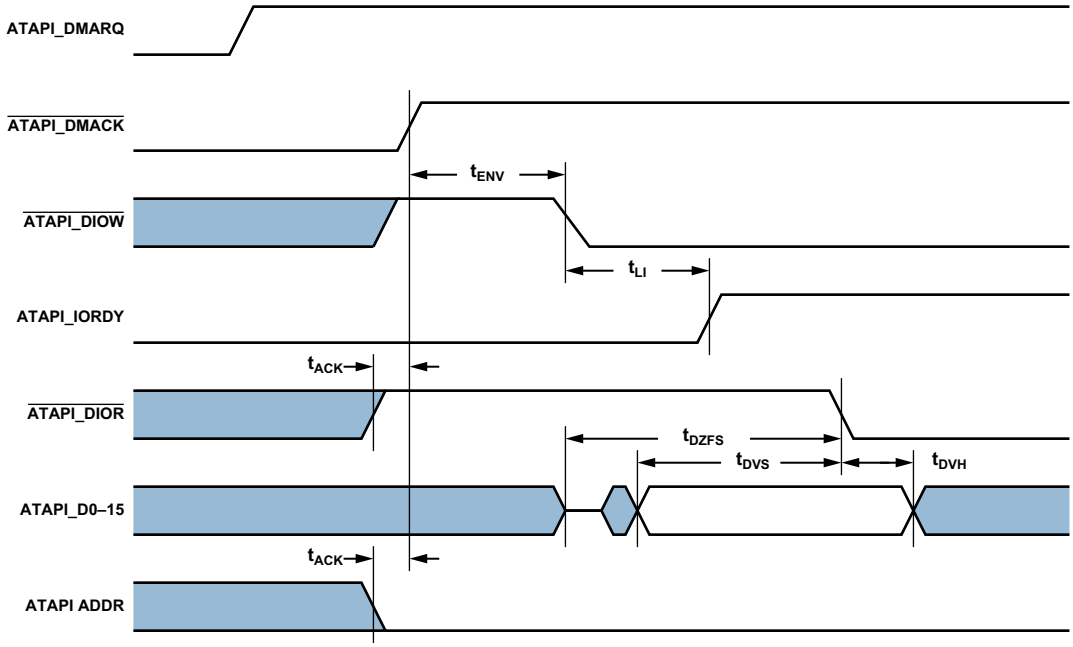



*Figure 56. Initiating an Ultra DMA Data-Out Burst*[1]

[1] This material is adapted from ATAPI-6 (INCITS 361-2002[R2007] and is used with permission of the American National Standards Institute (ANSI) on behalf of the Information Technology Industry Council ("ITIC"). Copies of ATAPI-6 (INCITS 361-2002[R2007] can be purchased from ANSI.

Figure 57 displays a sustained ultra DMA data-out burst. Note that an alternate ATAPI_D0-15 port bus is ATAPI_D0-15A.

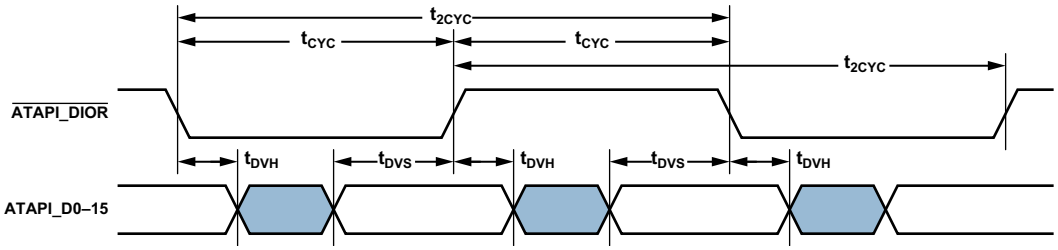



*Figure 57. Sustained Ultra DMA Data-Out Burst*[1]

[1] This material is adapted from ATAPI-6 (INCITS 361-2002[R2007] and is used with permission of the American National Standards Institute (ANSI) on behalf of the Information Technology Industry Council ("ITIC"). Copies of ATAPI-6 (INCITS 361-2002[R2007] can be purchased from ANSI.

Figure 58 displays a host terminating an ultra DMA data-out burst.

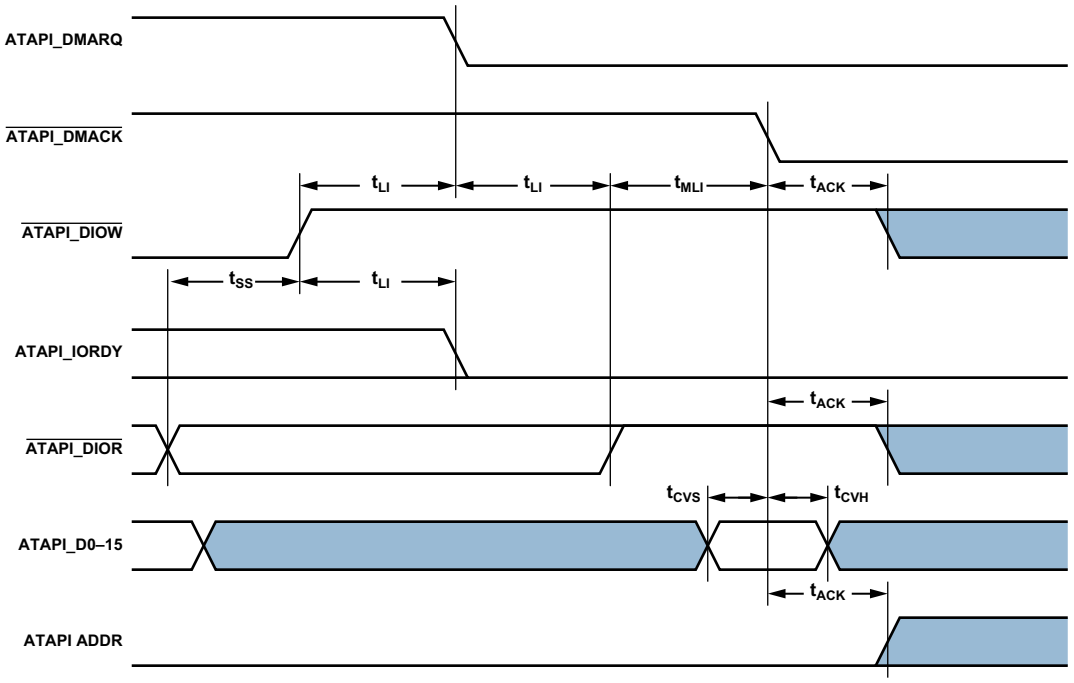



*Figure 58. Host terminating an Ultra DMA Data-Out Burst*[1]

[1] This material is adapted from ATAPI-6 (INCITS 361-2002[R2007] and is used with permission of the American National Standards Institute (ANSI) on behalf of the Information Technology Industry Council ("ITIC"). Copies of ATAPI-6 (INCITS 361-2002[R2007] can be purchased from ANSI.

Figure 59 displays a device terminating an ultra DMA data-out burst.

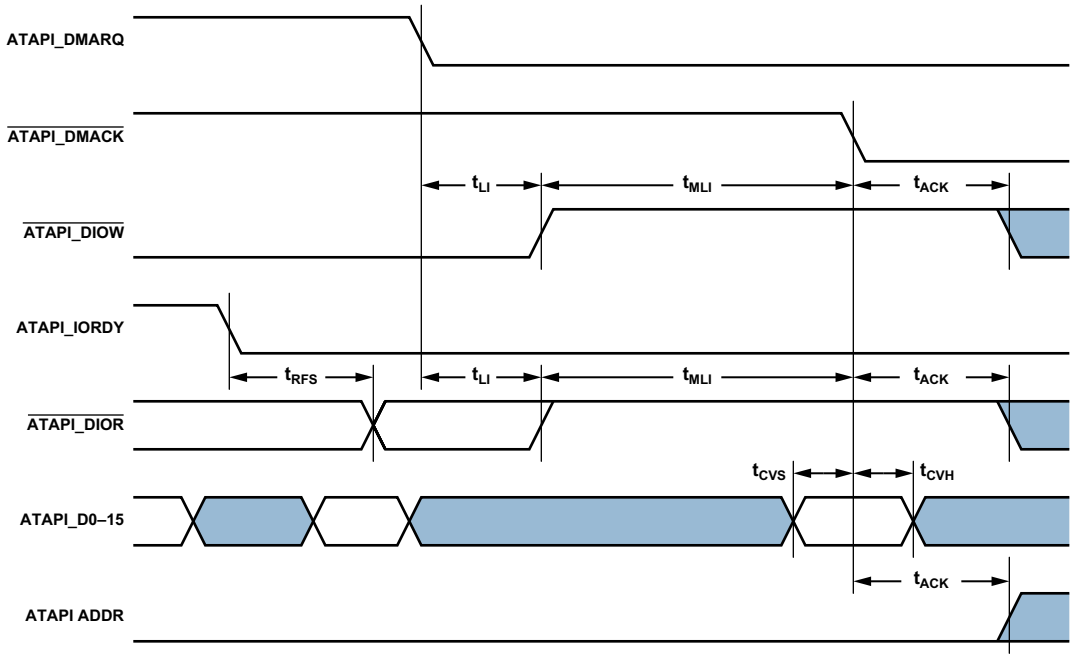



*Figure 59. Device Terminating an Ultra DMA Data-Out Burst*[1]

[1] This material is adapted from ATAPI-6 (INCITS 361-2002[R2007] and is used with permission of the American National Standards Institute (ANSI) on behalf of the Information Technology Industry Council ("ITIC"). Copies of ATAPI-6 (INCITS 361-2002[R2007] can be purchased from ANSI.

### *USB On-The-Go-Dual-Role Device Controller Timing*

Table 62 describes the USB On-The-Go Dual-Role Device Controller timing requirements.

**Table 62. USB On-The-Go Dual-Role Device Controller Timing Requirements**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $f_{USB}$ | USB_XI frequency | 9 | 33.3 | MHz |
| $FS_{USB}$ | USB_XI Clock Frequency Stability | –50 | +50 | ppm |

### *JTAG Test And Emulation Port Timing*

Table 63 and Figure 60 describe JTAG port operations.

**Table 63. JTAG Port Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Parameters* | | | | |
| $t_{TCK}$ | TCK Period | 20 | | ns |
| $t_{STAP}$ | TDI, TMS Setup Before TCK High | 4 | | ns |
| $t_{HTAP}$ | TDI, TMS Hold After TCK High | 4 | | ns |
| $t_{SSYS}$ | System Inputs Setup Before TCK High[1] | 4 | | ns |
| $t_{HSYS}$ | System Inputs Hold After TCK High[1] | 11 | | ns |
| $t_{TRSTW}$ | $\overline{\text{TRST}}$ Pulse-Width[2] (measured in TCK cycles) | 4 | | $t_{TCK}$ |
| *Switching Characteristics* | | | | |
| $t_{DTDO}$ | TDO Delay from TCK Low | | 10 | ns |
| $t_{DSYS}$ | System Outputs Delay After TCK Low[3] | 0 | 16.5 | ns |

[1] System inputs = PA15–0, PB14–0, PC13–0, PD15–0, PE15–0, PF15–0, PG15–0, PH13–0, PI15–0, PJ13–0, DQ15–0, DQS1–0, D15–0, $\overline{\text{ATAPI_PDIAG}}$, $\overline{\text{RESET}}$, $\overline{\text{NMI}}$, and BMODE3–0.

[2] 50 MHz Maximum

[3] System outputs = PA15–0, PB14–0, PC13–0, PD15–0, PE15–0, PF15–0, PG15–0, PH13–0, PI15–0, PJ13–0, DQ15–0, DQS1–0, D15–0, DA12–0, DBA1–0, DQM1–0, DCLK0-1, $\overline{\text{DCLK0–1}}$, $\overline{\text{DCS1–0}}$, DCLKE, $\overline{\text{DRAS}}$, $\overline{\text{DCAS}}$, $\overline{\text{DWE}}$, $\overline{\text{AMS3–0}}$, $\overline{\text{ABE1–0}}$, $\overline{\text{AOE}}$, $\overline{\text{ARE}}$, $\overline{\text{AWE}}$, CLKOUT, A3–1, and MFS.

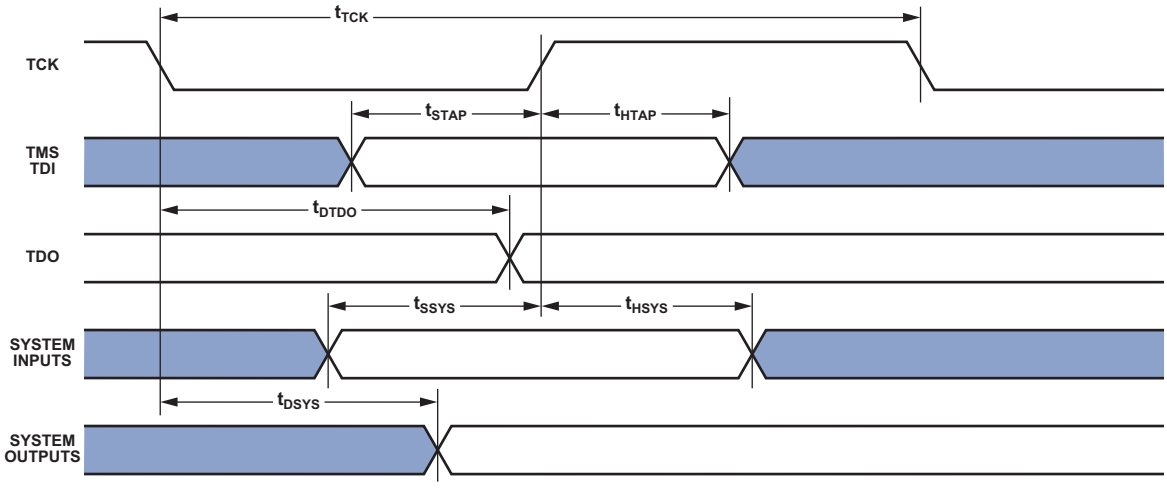



*Figure 60. JTAG Port Timing*

## OUTPUT DRIVE CURRENTS

Figure 61 through Figure 70 show typical current-voltage characteristics for the output drivers of the ADSP-BF54x Blackfin processors. The curves represent the current drive capability of the output drivers as a function of output voltage.

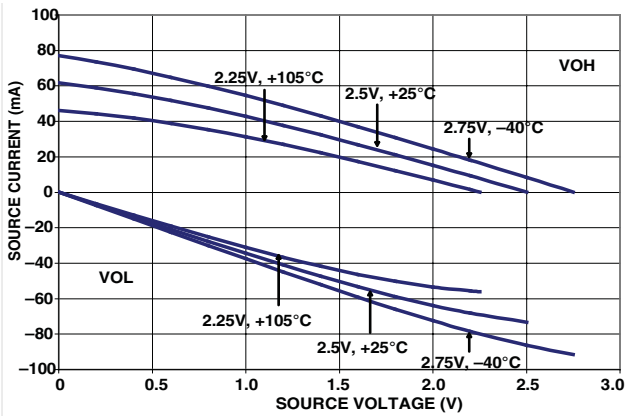



*Figure 61. Drive Current A (Low $V_{DDEXT}$)*

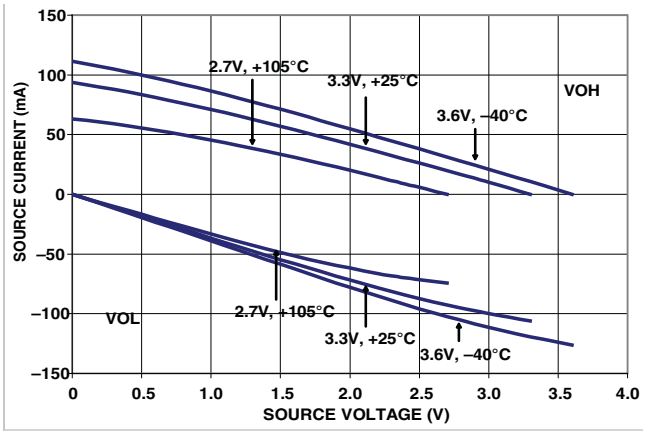



*Figure 62. Drive Current A (High $V_{DDEXT}$)*

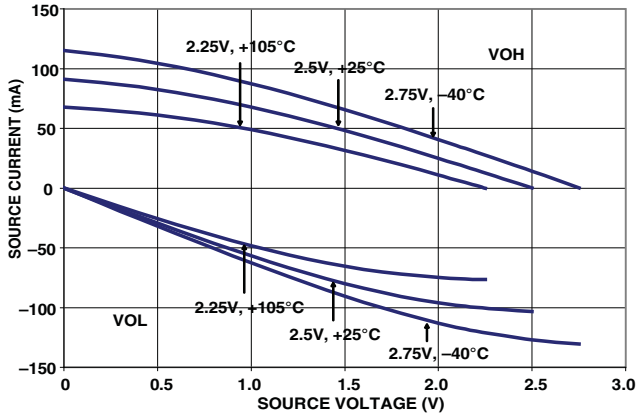



*Figure 63. Drive Current B (Low $V_{DDEXT}$)*

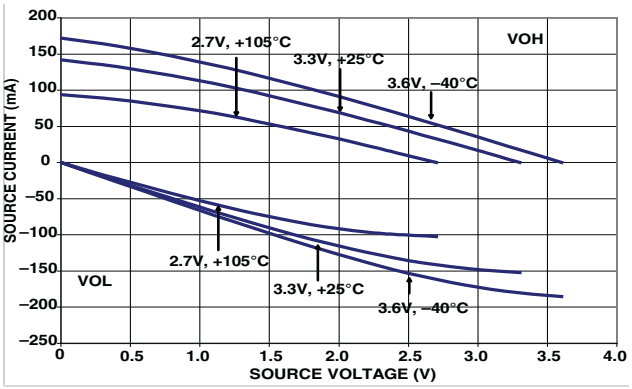



*Figure 64. Drive Current B (High $V_{DDEXT}$)*

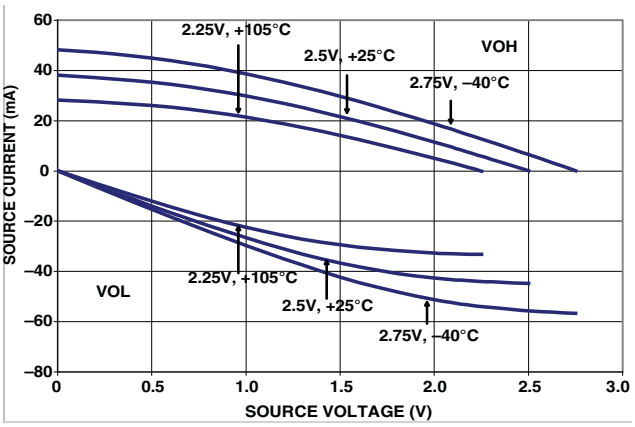



*Figure 65. Drive Current C (Low $V_{DDEXT}$)*

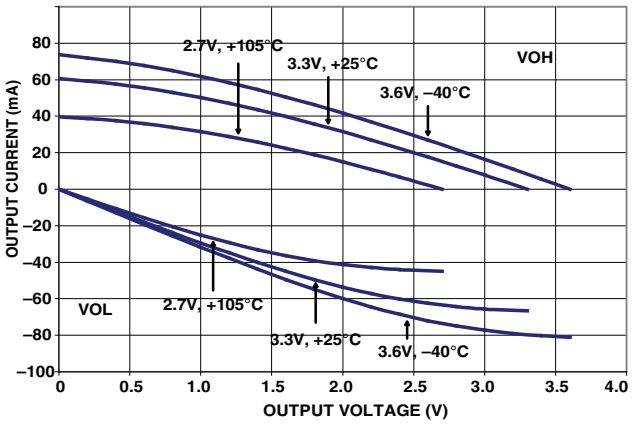



*Figure 66. Drive Current C (High $V_{DDEXT}$)*

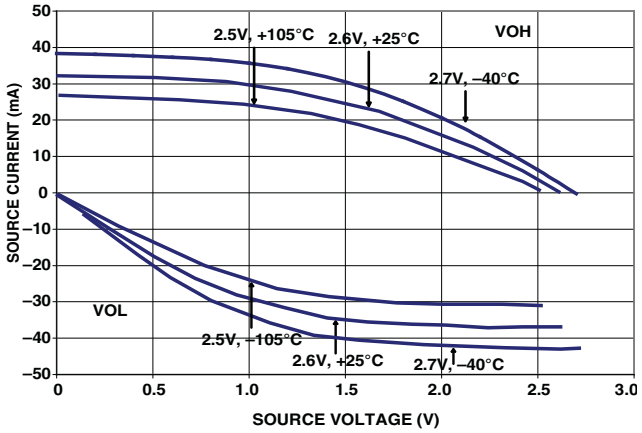



*Figure 67. Drive Current D (DDR SDRAM)*

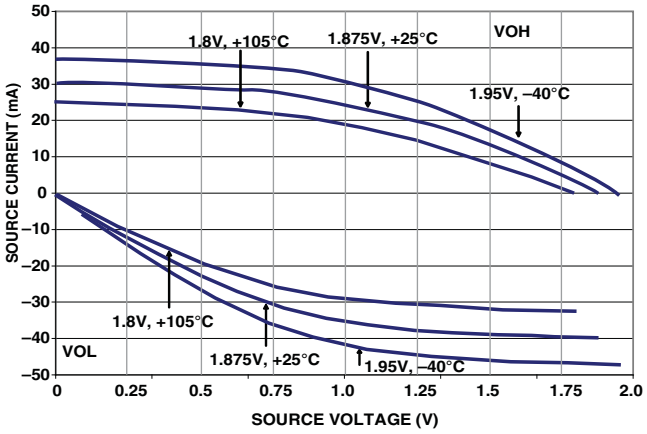



*Figure 68. Drive Current D (Mobile DDR SDRAM)*

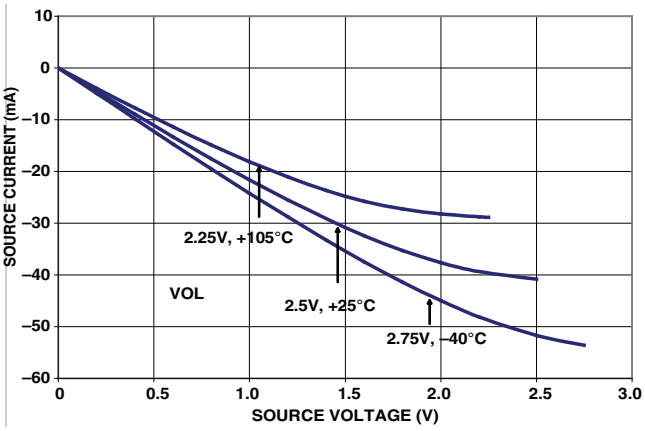



*Figure 69. Drive Current E (Low $V_{DDEXT}$)*

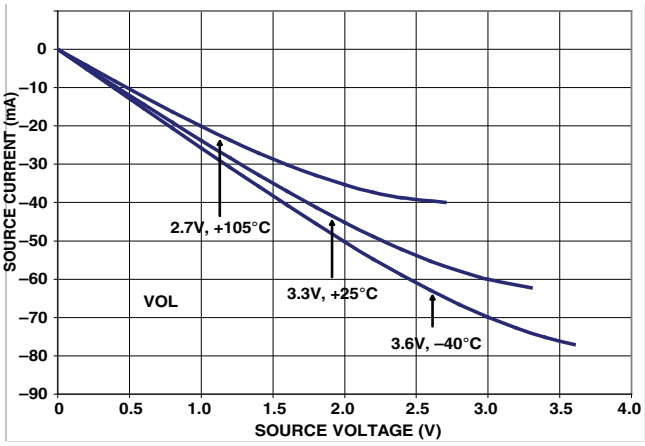



*Figure 70. Drive Current E (High $V_{DDEXT}$)*

## TEST CONDITIONS

All timing parameters appearing in this data sheet were measured under the conditions described in this section. Figure 71 shows the measurement point for AC measurements (except output enable/disable). The measurement point $V_{MEAS}$ is $V_{DDEXT}/2$ or $V_{DDDDR}/2$, depending on the pin under test.

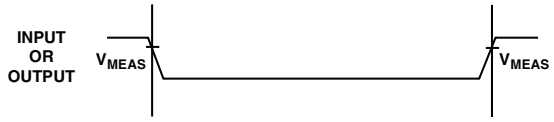



*Figure 71. Voltage Reference Levels for AC Measurements (Except Output Enable/Disable)*

### Output Enable Time

Output pins are considered to be enabled when they have made a transition from a high-impedance state to the point when they start driving. The output enable time $t_{ENA}$ is the interval from the point when a reference signal reaches a high or low voltage level to the point when the output starts driving as shown in the output enable/disable diagram (Figure 72). The time, $t_{ENA_MEASURED}$, is the interval from the point when the reference signal switches to the point when the output voltage reaches either 1.75 V (output high) or 1.25 V (output low). Time $t_{TRIP}$ is the interval from when the output starts driving to when the output reaches the 1.25 V or 1.75 V trip voltage. Time $t_{ENA}$ is calculated as shown in the equation:

$$t_{ENA} = t_{ENA_MEASURED} - t_{TRIP}$$

If multiple pins (such as the data bus) are enabled, the measurement value is that of the first pin to start driving.

### Output Disable Time

Output pins are considered to be disabled when they stop driving, go into a high-impedance state, and start to decay from their output high or low voltage. The time for the voltage on the bus to decay by ΔV is dependent on the capacitive load, $C_L$ and the load current, $I_L$. This decay time can be approximated by the equation:

$$t_{DECAY} = (C_L \Delta V)/I_L$$

The output disable time $t_{DIS}$ is the difference between $t_{DIS_MEASURED}$ and $t_{DECAY}$ as shown in Figure 72. The time $t_{DIS_MEASURED}$ is the interval from when the reference signal switches to when the output voltage decays ΔV from the measured output high or output low voltage. The time $t_{DECAY}$ is calculated with test loads $C_L$ and $I_L$, and with ΔV equal to 0.25 V.

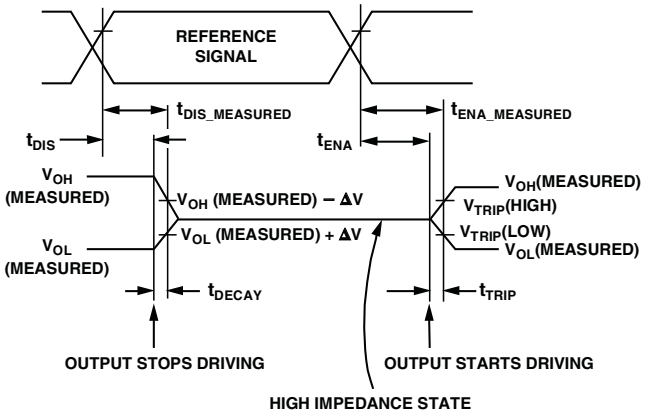


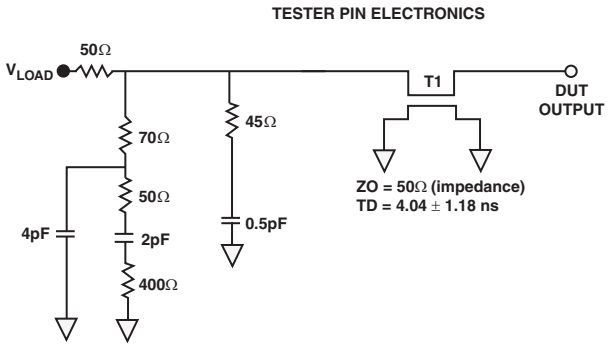

*Figure 72. Output Enable/Disable*

### Example System Hold Time Calculation

To determine the data output hold time in a particular system, first calculate $t_{DECAY}$ using the equation given above. Choose ΔV to be the difference between the ADSP-BF54x Blackfin processors' output voltage and the input threshold for the device requiring the hold time. A typical ΔV will be 0.4 V. $C_L$ is the total bus capacitance (per data line), and $I_L$ is the total leakage or three-state current (per data line). The hold time will be $t_{DECAY}$ plus the minimum disable time (for example, $t_{DDAT}$ for an asynchronous memory write cycle).

## CAPACITIVE LOADING

Output delays and holds are based on standard capacitive loads of an average of 6 pF on all balls (see Figure 73).




NOTES:
THE WORST-CASE TRANSMISSION LINE DELAY IS SHOWN AND CAN BE USED FOR THE OUTPUT TIMING ANALYSIS TO REFELECT THE TRANSMISSION LINE EFFECT AND MUST BE CONSIDERED. THE TRANSMISSION LINE (TD), IS FOR LOAD ONLY AND DOES NOT AFFECT THE DATA SHEET TIMING SPECIFICATIONS.

ANALOG DEVICES RECOMMENDS USING THE IBIS MODEL TIMING FOR A GIVEN SYSTEM REQUIREMENT. IF NECESSARY, A SYSTEM MAY INCORPORATE EXTERNAL DRIVERS TO COMPENSATE FOR ANY TIMING DIFFERENCES.

*Figure 73. Equivalent Device Loading for AC Measurements (Includes All Fixtures)*

$V_{LOAD}$ is equal to $V_{DDEXT}/2$ or $V_{DDDDR}/2$, depending on the pin under test. Figure 74 through Figure 85 on Page 91 show how output rise time varies with capacitance. The delay and hold specifications given should be derated by a factor derived from these figures. The graphs in these figures may not be linear outside the ranges shown.

## TYPICAL RISE AND FALL TIMES

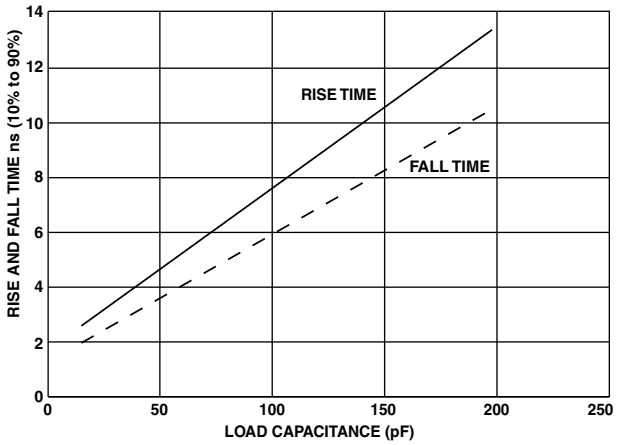



*Figure 74. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver A at $V_{DDEXT}$ = 2.25 V*

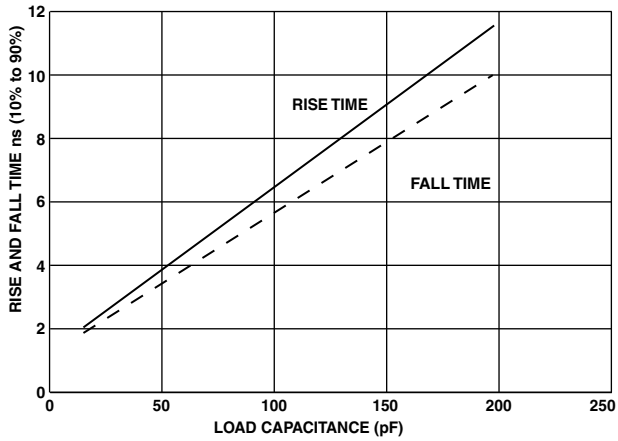



*Figure 75. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver A at $V_{DDEXT}$ = 3.65 V*

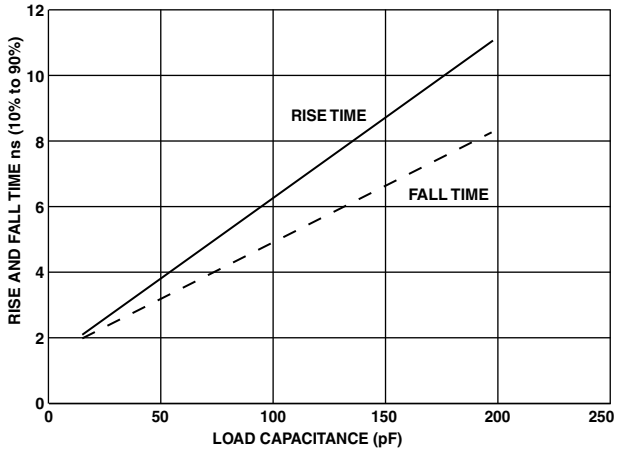



*Figure 76. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver B at $V_{DDEXT}$ = 2.25 V*

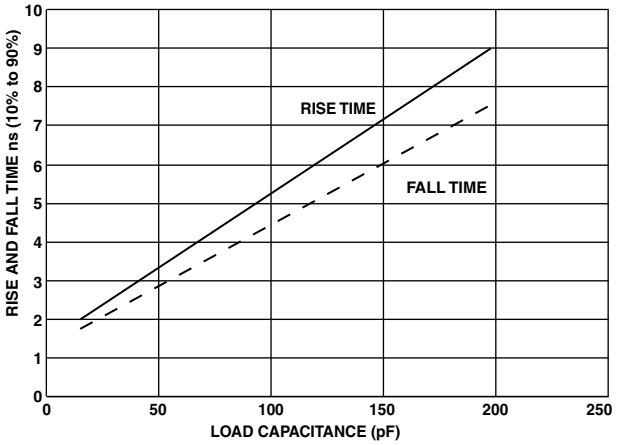



*Figure 77. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver B at $V_{DDEXT}$ = 3.65 V*

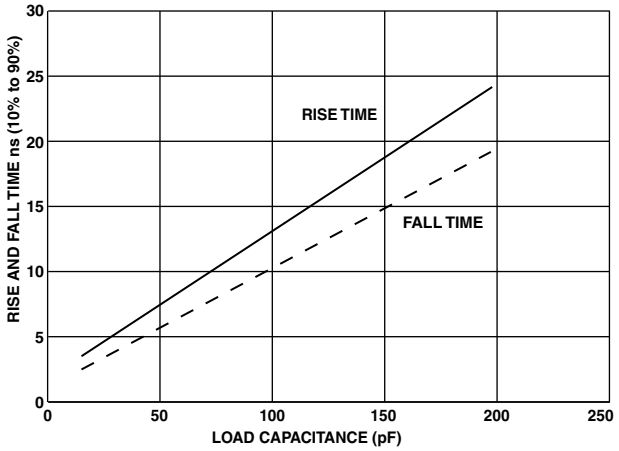



*Figure 78. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver C at $V_{DDEXT}$ = 2.25 V*

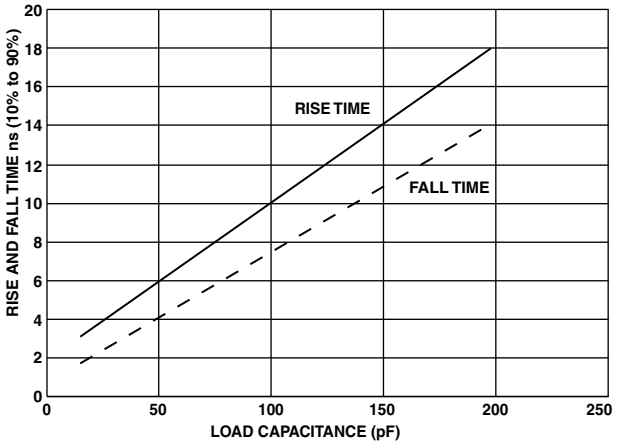



*Figure 79. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver C at $V_{DDEXT}$ = 3.65 V*

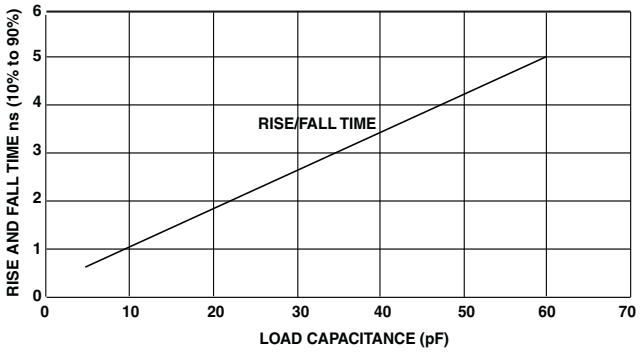



*Figure 80. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver D DDR SDRAM at $V_{DDDDR}$= 2.5V*

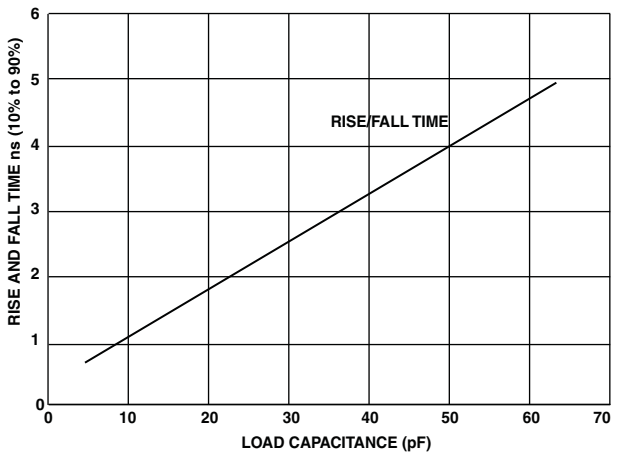



*Figure 81. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver D DDR SDRAM at $V_{DDDDR}$= 2.7V*

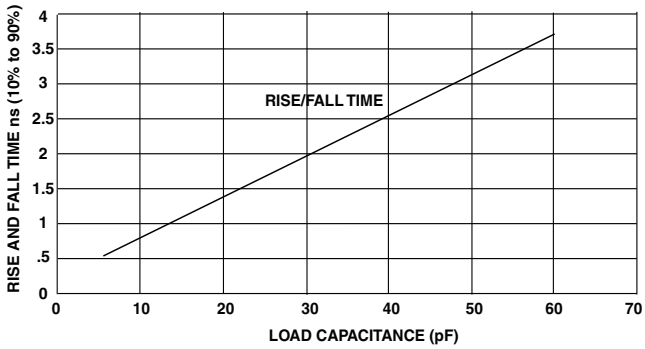



*Figure 82. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver D Mobile DDR SDRAM at $V_{DDDDR}$ = 1.8V*

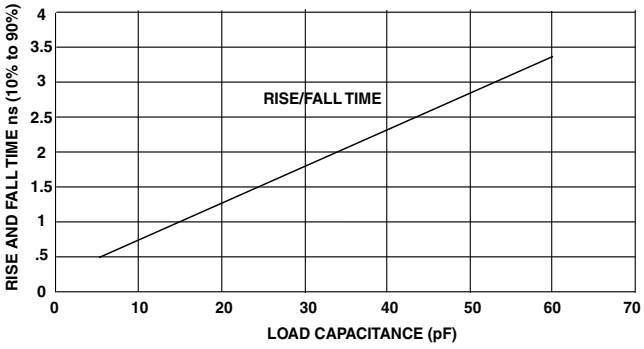



*Figure 83. Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance for Driver D Mobile DDR SDRAM at $V_{DDDDR}$ = 1.95V*

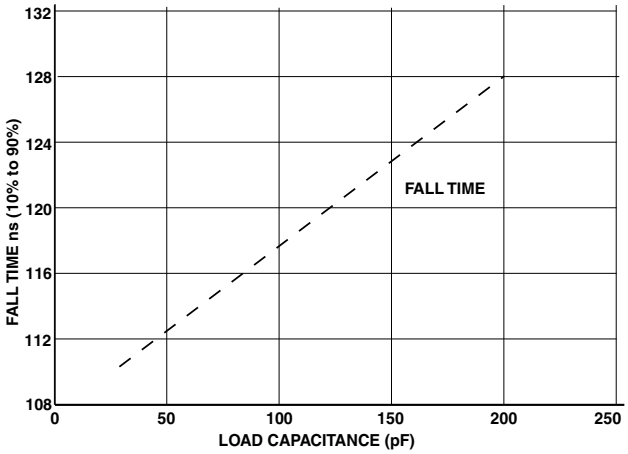



*Figure 84. Typical Fall Time (10% to 90%) vs. Load Capacitance for Driver E at $V_{DDEXT}$ = 2.7 V*

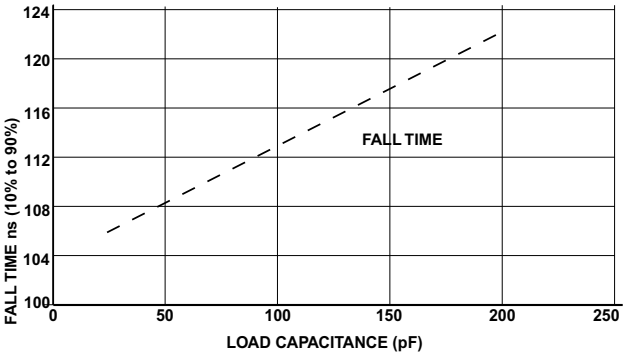



*Figure 85. Typical Fall Time (10% to 90%) vs. Load Capacitance for Driver E at $V_{DDEXT}$ = 3.65 V*

## THERMAL CHARACTERISTICS

To determine the junction temperature on the application printed circuit board use

$$T_J = T_{CASE} + (\Psi_{JT} \times P_D)$$

where:

$T_J$ =junction temperature (°C)

$T_{CASE}$ = case temperature (°C) measured by customer at top center of package.

$\Psi_{JT}$ = from Table 71

$P_D$ = power dissipation. (See Table 18 on Page 38 for a method to calculate $P_D$.)

Values of $\theta_{JA}$ are provided for package comparison and printed circuit board design considerations. $\theta_{JA}$ can be used for a first order approximation of $T_J$ by the equation

$$T_J = T_A + (\theta_{JA} \times P_D)$$

where:

$T_A$ = ambient temperature (°C)

Table 64 lists values for $\theta_{JC}$ and $\theta_{JB}$ parameters. These values are provided for package comparison and printed circuit board design considerations. Airflow measurements in Table 64 comply with JEDEC standards JESD51-2 and JESD51-6, and the junction-to-board measurement complies with JESD51-8. The junction-to-case measurement complies with MIL-STD-883 (Method 1012.1). All measurements use a 2S2P JEDEC testboard.

**Table 64. Thermal Characteristics, 400-Ball CSP_BGA**

| Parameter | Condition | Typical | Unit |
|---|---|---|---|
| $\theta_{JA}$ | 0 linear m/s air flow | 18.4 | °C/W |
| | 1 linear m/s air flow | 15.8 | °C/W |
| | 2 linear m/s air flow | 15.0 | °C/W |
| $\theta_{JB}$ | | 9.75 | °C/W |
| $\theta_{JC}$ | | 6.37 | °C/W |
| $\Psi_{JT}$ | 0 linear m/s air flow | 0.27 | °C/W |
| | 1 linear m/s air flow | 0.60 | °C/W |
| | 2 linear m/s air flow | 0.66 | °C/W |

# 400-BALL CSP_BGA PACKAGE

Table 65 lists the CSP_BGA package by signal for the ADSP-BF549. Table 66 on Page 95 lists the CSP_BGA package by ball number.

**Table 65. 400-Ball CSP_BGA Ball Assignment (Alphabetically by Signal)**

| Signal | Ball No. | Signal | Ball No. | Signal | Ball No. | Signal | Ball No. |
|---|---|---|---|---|---|---|---|
| A1 | B2 | DA4 | G16 | DQS1 | H18 | GND | L10 |
| A2 | A2 | DA5 | F19 | $\overline{\text{DRAS}}$ | E17 | GND | L11 |
| A3 | B3 | DA6 | D20 | $\overline{\text{DWE}}$ | E18 | GND | L12 |
| $\overline{\text{ABE0}}$ | C17 | DA7 | C20 | $\overline{\text{EMU}}$ | R5 | GND | L13 |
| $\overline{\text{ABE1}}$ | C16 | DA8 | F18 | EXT_WAKE | M18 | GND | L14 |
| $\overline{\text{AMS0}}$ | A10 | DA9 | E19 | GND | A1 | GND | M6 |
| $\overline{\text{AMS1}}$ | D9 | DA10 | B20 | GND | A13 | GND | M7 |
| $\overline{\text{AMS2}}$ | B10 | DA11 | F17 | GND | A20 | GND | M8 |
| $\overline{\text{AMS3}}$ | D10 | DA12 | D19 | GND | B11 | GND | M9 |
| $\overline{\text{AOE}}$ | C10 | DBA0 | H17 | GND | D1 | GND | M10 |
| $\overline{\text{ARE}}$ | B12 | DBA1 | H16 | GND | D4 | GND | M11 |
| $\overline{\text{ATAPI_PDIAG}}$ | P19 | $\overline{\text{DCAS}}$ | F16 | GND | E3 | GND | M12 |
| $\overline{\text{AWE}}$ | D12 | $\overline{\text{DCLK0}}$ | E16 | GND | F3 | GND | M13 |
| BMODE0 | W1 | DCLK0 | D16 | GND | F6 | GND | M14 |
| BMODE1 | W2 | DCLK1 | C18 | GND | F14 | GND | N6 |
| BMODE2 | W3 | $\overline{\text{DCLK1}}$ | D18 | GND | G9 | GND | N7 |
| BMODE3 | W4 | DCLKE | B18 | GND | G10 | GND | N8 |
| CLKBUF | D11 | $\overline{\text{DCS0}}$ | C19 | GND | G11 | GND | N9 |
| CLKIN | A11 | $\overline{\text{DCS1}}$ | B19 | GND | H7 | GND | N10 |
| CLKOUT | L16 | DDR_VREF | M20 | GND | H8 | GND | N11 |
| D0 | D13 | DDR_VSSR | N20 | GND | H9 | GND | N12 |
| D1 | C13 | DQ0 | L18 | GND | H10 | GND | N13 |
| D2 | B13 | DQ1 | M19 | GND | H11 | GND | N14 |
| D3 | B15 | DQ2 | L19 | GND | H12 | GND | P8 |
| D4 | A15 | DQ3 | L20 | GND | J7 | GND | P9 |
| D5 | B16 | DQ4 | L17 | GND | J8 | GND | P10 |
| D6 | A16 | DQ5 | K16 | GND | J9 | GND | P11 |
| D7 | B17 | DQ6 | K20 | GND | J10 | GND | P12 |
| D8 | C14 | DQ7 | K17 | GND | J11 | GND | P13 |
| D9 | C15 | DQ8 | K19 | GND | J12 | GND | R9 |
| D10 | A17 | DQ9 | J20 | GND | K7 | GND | R13 |
| D11 | D14 | DQ10 | K18 | GND | K8 | GND | R14 |
| D12 | D15 | DQ11 | H20 | GND | K9 | GND | R16 |
| D13 | E15 | DQ12 | J19 | GND | K10 | GND | U8 |
| D14 | E14 | DQ13 | J18 | GND | K11 | GND | V6 |
| D15 | D17 | DQ14 | J17 | GND | K12 | GND | Y1 |
| DA0 | G19 | DQ15 | J16 | GND | K13 | GND | Y20 |
| DA1 | G17 | DQM0 | G20 | GND | L7 | $\text{GND}_{\text{MP}}$ | E7 |
| DA2 | E20 | DQM1 | H19 | GND | L8 | MFS | E6 |
| DA3 | G18 | DQS0 | F20 | GND | L9 | MLF_M | F4 |

**Table 65. 400-Ball CSP_BGA Ball Assignment (Alphabetically by Signal) (Continued)**

| Signal | Ball No. | Signal | Ball No. | Signal | Ball No. | Signal | Ball No. |
|---|---|---|---|---|---|---|---|
| MLF_P | E4 | PC5 | G1 | PE15 | W17 | PH7 | H4 |
| MXI | C2 | PC6 | J5 | PF0 | K3 | PH8 | D5 |
| MXO | C1 | PC7 | H3 | PF1 | J1 | PH9 | C4 |
| $\overline{\text{NMI}}$ | C11 | PC8 | Y14 | PF2 | K2 | PH10 | C7 |
| PA0 | U12 | PC9 | V13 | PF3 | K1 | PH11 | C5 |
| PA1 | V12 | PC10 | U13 | PF4 | L2 | PH12 | D7 |
| PA2 | W12 | PC11 | W14 | PF5 | L1 | PH13 | C6 |
| PA3 | Y12 | PC12 | Y15 | PF6 | L4 | PI0 | A3 |
| PA4 | W11 | PC13 | W15 | PF7 | K4 | PI1 | B4 |
| PA5 | V11 | PD0 | P3 | PF8 | L3 | PI2 | A4 |
| PA6 | Y11 | PD1 | P4 | PF9 | M1 | PI3 | B5 |
| PA7 | U11 | PD2 | R1 | PF10 | M2 | PI4 | A5 |
| PA8 | U10 | PD3 | R2 | PF11 | M3 | PI5 | B6 |
| PA9 | Y10 | PD4 | T1 | PF12 | M4 | PI6 | A6 |
| PA10 | Y9 | PD5 | R3 | PF13 | N4 | PI7 | B7 |
| PA11 | V10 | PD6 | T2 | PF14 | N1 | PI8 | A7 |
| PA12 | Y8 | PD7 | R4 | PF15 | N2 | PI9 | C8 |
| PA13 | W10 | PD8 | U1 | PG0 | J4 | PI10 | B8 |
| PA14 | Y7 | PD9 | U2 | PG1 | K5 | PI11 | A8 |
| PA15 | W9 | PD10 | T3 | PG2 | L5 | PI12 | A9 |
| PB0 | W5 | PD11 | V1 | PG3 | N3 | PI13 | C9 |
| PB1 | Y2 | PD12 | T4 | PG4 | P1 | PI14 | D8 |
| PB2 | T6 | PD13 | V2 | PG5 | V15 | PI15 | B9 |
| PB3 | U6 | PD14 | U4 | PG6 | Y17 | PJ0 | R20 |
| PB4 | Y4 | PD15 | U3 | PG7 | W16 | PJ1 | N18 |
| PB5 | Y3 | PE0 | V19 | PG8 | V16 | PJ2 | M16 |
| PB6 | W6 | PE1 | T17 | PG9 | Y19 | PJ3 | T20 |
| PB7 | V7 | PE2 | U18 | PG10 | Y18 | PJ4 | N17 |
| PB8 | W8 | PE3 | V14 | PG11 | U15 | PJ5 | U20 |
| PB9 | V8 | PE4 | Y16 | PG12 | P16 | PJ6 | P18 |
| PB10 | U7 | PE5 | W20 | PG13 | R18 | PJ7 | N16 |
| PB11 | W7 | PE6 | W19 | PG14 | Y13 | PJ8 | R19 |
| PB12 | Y6 | PE7 | R17 | PG15 | W13 | PJ9 | P17 |
| PB13 | V9 | PE8 | V20 | PH0 | W18 | PJ10 | T19 |
| PB14 | Y5 | PE9 | U19 | PH1 | U14 | PJ11 | M17 |
| PC0 | H2 | PE10 | T18 | PH2 | V17 | PJ12 | P20 |
| PC1 | J3 | PE11 | P2 | PH3 | V18 | PJ13 | N19 |
| PC2 | J2 | PE12 | M5 | PH4 | U17 | $\overline{\text{RESET}}$ | C12 |
| PC3 | H1 | PE13 | P5 | PH5 | C3 | RTXI | A14 |
| PC4 | G2 | PE14 | U16 | PH6 | D6 | RTXO | B14 |

**Table 65. 400-Ball CSP_BGA Ball Assignment (Alphabetically by Signal) (Continued)**

| Signal | Ball No. | Signal | Ball No. | Signal | Ball No. | Signal | Ball No. |
|---|---|---|---|---|---|---|---|
| TCK | V3 | $V_{DDDDR}$ | J14 | $V_{DDEXT}$ | N5 | $V_{DDINT}$ | G13 |
| TDI | V5 | $V_{DDDDR}$ | J15 | $V_{DDEXT}$ | N15 | $V_{DDINT}$ | J6 |
| TDO | V4 | $V_{DDDDR}$ | K14 | $V_{DDEXT}$ | P15 | $V_{DDINT}$ | J13 |
| TMS | U5 | $V_{DDDDR}$ | K15 | $V_{DDEXT}$ | R6 | $V_{DDINT}$ | L6 |
| $\overline{TRST}$ | T5 | $V_{DDEXT}$ | E5 | $V_{DDEXT}$ | R7 | $V_{DDINT}$ | L15 |
| USB_DM | E2 | $V_{DDEXT}$ | E9 | $V_{DDEXT}$ | R8 | $V_{DDINT}$ | P6 |
| USB_DP | E1 | $V_{DDEXT}$ | E10 | $V_{DDEXT}$ | R15 | $V_{DDINT}$ | P7 |
| USB_ID | G3 | $V_{DDEXT}$ | E11 | $V_{DDEXT}$ | T7 | $V_{DDINT}$ | P14 |
| USB_RSET | D3 | $V_{DDEXT}$ | E12 | $V_{DDEXT}$ | T8 | $V_{DDINT}$ | R10 |
| USB_VBUS | D2 | $V_{DDEXT}$ | F7 | $V_{DDEXT}$ | T9 | $V_{DDINT}$ | R11 |
| USB_VREF | B1 | $V_{DDEXT}$ | F8 | $V_{DDEXT}$ | T10 | $V_{DDINT}$ | R12 |
| USB_XI | F1 | $V_{DDEXT}$ | F13 | $V_{DDEXT}$ | T11 | $V_{DDINT}$ | U9 |
| USB_XO | F2 | $V_{DDEXT}$ | G5 | $V_{DDEXT}$ | T12 | $V_{DDMP}$ | E8 |
| $V_{DDDDR}$ | F10 | $V_{DDEXT}$ | G6 | $V_{DDEXT}$ | T13 | $V_{DDRTC}$ | E13 |
| $V_{DDDDR}$ | F11 | $V_{DDEXT}$ | G7 | $V_{DDEXT}$ | T14 | $V_{DDUSB}$ | F5 |
| $V_{DDDDR}$ | F12 | $V_{DDEXT}$ | G14 | $V_{DDEXT}$ | T15 | $V_{DDUSB}$ | G4 |
| $V_{DDDDR}$ | G15 | $V_{DDEXT}$ | H5 | $V_{DDEXT}$ | T16 | $V_{DDVR}$ | F15 |
| $V_{DDDDR}$ | H13 | $V_{DDEXT}$ | H6 | $V_{DDINT}$ | F9 | $VR_{OUT0}$ | A18 |
| $V_{DDDDR}$ | H14 | $V_{DDEXT}$ | K6 | $V_{DDINT}$ | G8 | $VR_{OUT1}$ | A19 |
| $V_{DDDDR}$ | H15 | $V_{DDEXT}$ | M15 | $V_{DDINT}$ | G12 | XTAL | A12 |

Table 66 lists the CSP_BGA package by ball number for the ADSP-BF549. Table 65 on Page 92 lists the CSP_BGA package by signal.

**Table 66. 400-Ball CSP_BGA Ball Assignment (Numerically by Ball Number)**

| Ball No. | Signal | Ball No. | Signal | Ball No. | Signal | Ball No. | Signal |
|---|---|---|---|---|---|---|---|
| A1 | GND | C1 | MXO | E1 | USB_DP | G1 | PC5 |
| A2 | A2 | C2 | MXI | E2 | USB_DM | G2 | PC4 |
| A3 | PI0 | C3 | PH5 | E3 | GND | G3 | USB_ID |
| A4 | PI2 | C4 | PH9 | E4 | MLF_P | G4 | $V_{DDUSB}$ |
| A5 | PI4 | C5 | PH11 | E5 | $V_{DDEXT}$ | G5 | $V_{DDEXT}$ |
| A6 | PI6 | C6 | PH13 | E6 | MFS | G6 | $V_{DDEXT}$ |
| A7 | PI8 | C7 | PH10 | E7 | $GND_{MP}$ | G7 | $V_{DDEXT}$ |
| A8 | PI11 | C8 | PI9 | E8 | $V_{DDMP}$ | G8 | $V_{DDINT}$ |
| A9 | PI12 | C9 | PI13 | E9 | $V_{DDEXT}$ | G9 | GND |
| A10 | $\overline{AMS0}$ | C10 | $\overline{AOE}$ | E10 | $V_{DDEXT}$ | G10 | GND |
| A11 | CLKIN | C11 | $\overline{NMI}$ | E11 | $V_{DDEXT}$ | G11 | GND |
| A12 | XTAL | C12 | $\overline{RESET}$ | E12 | $V_{DDEXT}$ | G12 | $V_{DDINT}$ |
| A13 | GND | C13 | D1 | E13 | $V_{DDRTC}$ | G13 | $V_{DDINT}$ |
| A14 | RTXI | C14 | D8 | E14 | D14 | G14 | $V_{DDEXT}$ |
| A15 | D4 | C15 | D9 | E15 | D13 | G15 | $V_{DDDDR}$ |
| A16 | D6 | C16 | $\overline{ABE1}$ | E16 | $\overline{DCLK0}$ | G16 | DA4 |
| A17 | D10 | C17 | $\overline{ABE0}$ | E17 | $\overline{DRAS}$ | G17 | DA1 |
| A18 | $VROUT_0$ | C18 | DCLK1 | E18 | $\overline{DWE}$ | G18 | DA3 |
| A19 | $VROUT_1$ | C19 | $\overline{DCS0}$ | E19 | DA9 | G19 | DA0 |
| A20 | GND | C20 | DA7 | E20 | DA2 | G20 | DQM0 |
| B1 | USB_VREF | D1 | GND | F1 | USB_XI | H1 | PC3 |
| B2 | A1 | D2 | USB_VBUS | F2 | USB_XO | H2 | PC0 |
| B3 | A3 | D3 | USB_RSET | F3 | GND | H3 | PC7 |
| B4 | PI1 | D4 | GND | F4 | MLF_M | H4 | PH7 |
| B5 | PI3 | D5 | PH8 | F5 | $V_{DDUSB}$ | H5 | $V_{DDEXT}$ |
| B6 | PI5 | D6 | PH6 | F6 | GND | H6 | $V_{DDEXT}$ |
| B7 | PI7 | D7 | PH12 | F7 | $V_{DDEXT}$ | H7 | GND |
| B8 | PI10 | D8 | PI14 | F8 | $V_{DDEXT}$ | H8 | GND |
| B9 | PI15 | D9 | $\overline{AMS1}$ | F9 | $V_{DDINT}$ | H9 | GND |
| B10 | $\overline{AMS2}$ | D10 | $\overline{AMS3}$ | F10 | $V_{DDDDR}$ | H10 | GND |
| B11 | GND | D11 | CLKBUF | F11 | $V_{DDDDR}$ | H11 | GND |
| B12 | $\overline{ARE}$ | D12 | $\overline{AWE}$ | F12 | $V_{DDDDR}$ | H12 | GND |
| B13 | D2 | D13 | D0 | F13 | $V_{DDEXT}$ | H13 | $V_{DDDDR}$ |
| B14 | RTXO | D14 | D11 | F14 | GND | H14 | $V_{DDDDR}$ |
| B15 | D3 | D15 | D12 | F15 | $V_{DDVR}$ | H15 | $V_{DDDDR}$ |
| B16 | D5 | D16 | DCLK0 | F16 | $\overline{DCAS}$ | H16 | DBA1 |
| B17 | D7 | D17 | D15 | F17 | DA11 | H17 | DBA0 |
| B18 | DCLKE | D18 | $\overline{DCLK1}$ | F18 | DA8 | H18 | DQS1 |
| B19 | $\overline{DCS1}$ | D19 | DA12 | F19 | DA5 | H19 | DQM1 |
| B20 | DA10 | D20 | DA6 | F20 | DQS0 | H20 | DQ11 |

**Table 66. 400-Ball CSP_BGA Ball Assignment (Numerically by Ball Number) (Continued)**

| Ball No. | Signal | Ball No. | Signal | Ball No. | Signal | Ball No. | Signal |
|---|---|---|---|---|---|---|---|
| J1 | PF1 | L1 | PF5 | N1 | PF14 | R1 | PD2 |
| J2 | PC2 | L2 | PF4 | N2 | PF15 | R2 | PD3 |
| J3 | PC1 | L3 | PF8 | N3 | PG3 | R3 | PD5 |
| J4 | PG0 | L4 | PF6 | N4 | PF13 | R4 | PD7 |
| J5 | PC6 | L5 | PG2 | N5 | $V_{DDEXT}$ | R5 | $\overline{EMU}$ |
| J6 | $V_{DDINT}$ | L6 | $V_{DDINT}$ | N6 | GND | R6 | $V_{DDEXT}$ |
| J7 | GND | L7 | GND | N7 | GND | R7 | $V_{DDEXT}$ |
| J8 | GND | L8 | GND | N8 | GND | R8 | $V_{DDEXT}$ |
| J9 | GND | L9 | GND | N9 | GND | R9 | GND |
| J10 | GND | L10 | GND | N10 | GND | R10 | $V_{DDINT}$ |
| J11 | GND | L11 | GND | N11 | GND | R11 | $V_{DDINT}$ |
| J12 | GND | L12 | GND | N12 | GND | R12 | $V_{DDINT}$ |
| J13 | $V_{DDINT}$ | L13 | GND | N13 | GND | R13 | GND |
| J14 | $V_{DDDDR}$ | L14 | GND | N14 | GND | R14 | GND |
| J15 | $V_{DDDDR}$ | L15 | $V_{DDINT}$ | N15 | $V_{DDEXT}$ | R15 | $V_{DDEXT}$ |
| J16 | DQ15 | L16 | CLKOUT | N16 | PJ7 | R16 | GND |
| J17 | DQ14 | L17 | DQ4 | N17 | PJ4 | R17 | PE7 |
| J18 | DQ13 | L18 | DQ0 | N18 | PJ1 | R18 | PG13 |
| J19 | DQ12 | L19 | DQ2 | N19 | PJ13 | R19 | PJ8 |
| J20 | DQ9 | L20 | DQ3 | N20 | DDR_VSSR | R20 | PJ0 |
| K1 | PF3 | M1 | PF9 | P1 | PG4 | T1 | PD4 |
| K2 | PF2 | M2 | PF10 | P2 | PE11 | T2 | PD6 |
| K3 | PF0 | M3 | PF11 | P3 | PD0 | T3 | PD10 |
| K4 | PF7 | M4 | PF12 | P4 | PD1 | T4 | PD12 |
| K5 | PG1 | M5 | PE12 | P5 | PE13 | T5 | $\overline{TRST}$ |
| K6 | $V_{DDEXT}$ | M6 | GND | P6 | $V_{DDINT}$ | T6 | PB2 |
| K7 | GND | M7 | GND | P7 | $V_{DDINT}$ | T7 | $V_{DDEXT}$ |
| K8 | GND | M8 | GND | P8 | GND | T8 | $V_{DDEXT}$ |
| K9 | GND | M9 | GND | P9 | GND | T9 | $V_{DDEXT}$ |
| K10 | GND | M10 | GND | P10 | GND | T10 | $V_{DDEXT}$ |
| K11 | GND | M11 | GND | P11 | GND | T11 | $V_{DDEXT}$ |
| K12 | GND | M12 | GND | P12 | GND | T12 | $V_{DDEXT}$ |
| K13 | GND | M13 | GND | P13 | GND | T13 | $V_{DDEXT}$ |
| K14 | $V_{DDDDR}$ | M14 | GND | P14 | $V_{DDINT}$ | T14 | $V_{DDEXT}$ |
| K15 | $V_{DDDDR}$ | M15 | $V_{DDEXT}$ | P15 | $V_{DDEXT}$ | T15 | $V_{DDEXT}$ |
| K16 | DQ5 | M16 | PJ2 | P16 | PG12 | T16 | $V_{DDEXT}$ |
| K17 | DQ7 | M17 | PJ11 | P17 | PJ9 | T17 | PE1 |
| K18 | DQ10 | M18 | EXT_WAKE | P18 | PJ6 | T18 | PE10 |
| K19 | DQ8 | M19 | DQ1 | P19 | $\overline{ATAPI_PDIAG}$ | T19 | PJ10 |
| K20 | DQ6 | M20 | DDR_VREF | P20 | PJ12 | T20 | PJ3 |

**Table 66. 400-Ball CSP_BGA Ball Assignment (Numerically by Ball Number) (Continued)**

| Ball No. | Signal | Ball No. | Signal | Ball No. | Signal | Ball No. | Signal |
|---|---|---|---|---|---|---|---|
| U1 | PD8 | V1 | PD11 | W1 | BMODE0 | Y1 | GND |
| U2 | PD9 | V2 | PD13 | W2 | BMODE1 | Y2 | PB1 |
| U3 | PD15 | V3 | TCK | W3 | BMODE2 | Y3 | PB5 |
| U4 | PD14 | V4 | TDO | W4 | BMODE3 | Y4 | PB4 |
| U5 | TMS | V5 | TDI | W5 | PB0 | Y5 | PB14 |
| U6 | PB3 | V6 | GND | W6 | PB6 | Y6 | PB12 |
| U7 | PB10 | V7 | PB7 | W7 | PB11 | Y7 | PA14 |
| U8 | GND | V8 | PB9 | W8 | PB8 | Y8 | PA12 |
| U9 | $V_{DDINT}$ | V9 | PB13 | W9 | PA15 | Y9 | PA10 |
| U10 | PA8 | V10 | PA11 | W10 | PA13 | Y10 | PA9 |
| U11 | PA7 | V11 | PA5 | W11 | PA4 | Y11 | PA6 |
| U12 | PA0 | V12 | PA1 | W12 | PA2 | Y12 | PA3 |
| U13 | PC10 | V13 | PC9 | W13 | PG15 | Y13 | PG14 |
| U14 | PH1 | V14 | PE3 | W14 | PC11 | Y14 | PC8 |
| U15 | PG11 | V15 | PG5 | W15 | PC13 | Y15 | PC12 |
| U16 | PE14 | V16 | PG8 | W16 | PG7 | Y16 | PE4 |
| U17 | PH4 | V17 | PH2 | W17 | PE15 | Y17 | PG6 |
| U18 | PE2 | V18 | PH3 | W18 | PH0 | Y18 | PG10 |
| U19 | PE9 | V19 | PE0 | W19 | PE6 | Y19 | PG9 |
| U20 | PJ5 | V20 | PE8 | W20 | PE5 | Y20 | GND |

Figure 86 lists the top view of the BGA ball configuration.

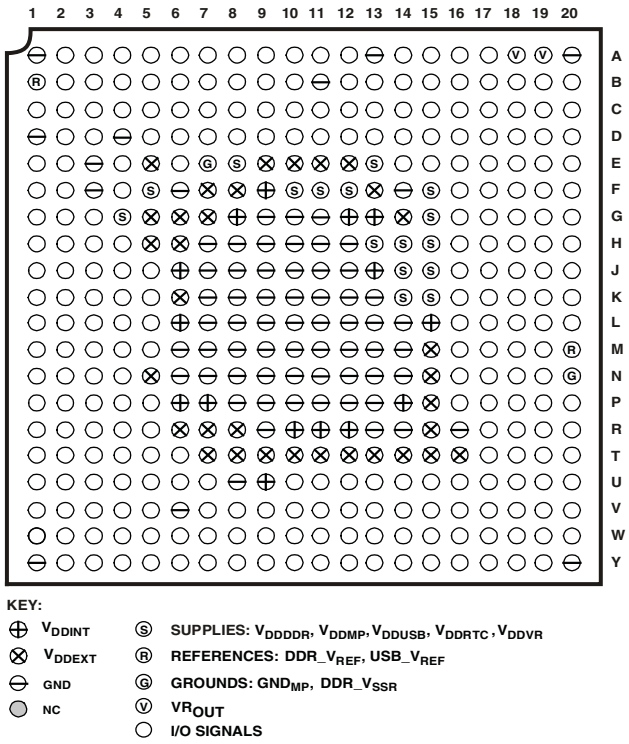
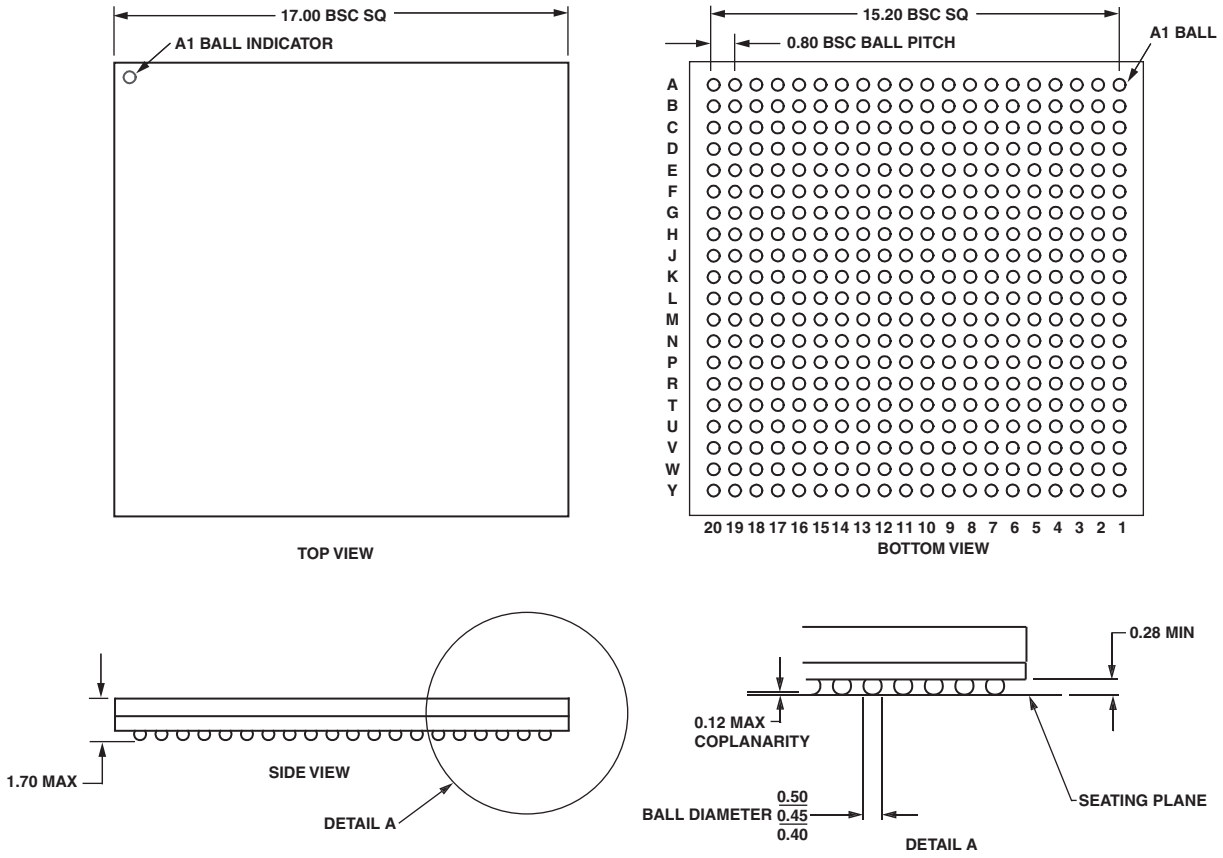



*Figure 86. 400-Ball CSP_BGA Configuration (Top View)*

# OUTLINE DIMENSIONS

Dimensions for the 17 mm × 17 mm CSP_BGA package in Figure 87 are shown in millimeters.




NOTES:
1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. COMPLIANT TO JEDEC REGISTERED OUTLINE MO-205, VARIATION AM, WITH THE EXCEPTION OF BALL DIAMETER.
3. CENTER DIMENSIONS ARE NOMINAL.

*Figure 87. 400-Ball, 17 mm × 17 mm CSP_BGA (Chip Scale Package Ball Grid Array) (BC-400-1)*

## SURFACE-MOUNT DESIGN

Table 67 is provided as an aid to PCB design. For industry-standard design recommendations, refer to IPC-7351, *Generic Requirements for Surface-Mount Design and Land Pattern Standard.*

**Table 67. BGA Data for Use with Surface-Mount Design**

| Package | Package Ball Attach Type | Package Solder Mask Opening | Package Ball Pad Size |
|---|---|---|---|
| 400-Ball CSP_BGA (Chip Scale Package Ball Grid Array) BC-400-1 | Solder Mask Defined | 0.40 mm Diameter | 0.50 mm Diameter |

## AUTOMOTIVE PRODUCTS

Some ADSP-BF54x Blackfin processor models are available for automotive applications with controlled manufacturing. Note that these special models may have specifications that differ from the general release models.

The automotive grade products shown in Table 68 are available for use in automotive applications. Contact your local ADI account representative or authorized ADI product distributor for specific product ordering information. Note that all automotive products are RoHS compliant.

**Table 68. Automotive Products**

| Product Family[1] | Temperature Range[2] | Speed Grade (Max) | Package Description | Package Option |
|---|---|---|---|---|
| ADBF542WBBCZ-4xx | –40°C to +85°C | 400 MHz | 400-Ball CSP_BGA | BC-400-1 |
| ADBF542WBBCZ-5xx | –40°C to +85°C | 533 MHz | 400-Ball CSP_BGA | BC-400-1 |
| ADBF544WBBCZ-5xx | –40°C to +85°C | 533 MHz | 400-Ball CSP_BGA | BC-400-1 |
| ADBF549WBBCZ-5xx | –40°C to +85°C | 533 MHz | 400-Ball CSP_BGA | BC-400-1 |
| ADBF549MWBBCZ-5xx | –40°C to +85°C | 533 MHz | 400-Ball CSP_BGA | BC-400-1 |

[1] The use of xx designates silicon revision
[2] Referenced temperature is ambient temperature.

## ORDERING GUIDE

| Model[1, 2, 3] | Temperature Range[4] | Speed Grade (Max) | Package Description | Package Option |
|---|---|---|---|---|
| ADSP-BF542BBCZ-4A | –40°C to +85°C | 400 MHz | 400-Ball CSP_BGA | BC-400-1 |
| ADSP-BF542BBCZ-5A | –40°C to +85°C | 533 MHz | 400-Ball CSP_BGA | BC-400-1 |
| ADSP-BF542MBBCZ-5M | –40°C to +85°C | 533 MHz | 400-Ball CSP_BGA | BC-400-1 |
| ADSP-BF542KBCZ-6A | 0°C to +70°C | 600 MHz | 400-Ball CSP_BGA | BC-400-1 |
| ADSP-BF544BBCZ-4A | –40°C to +85°C | 400 MHz | 400-Ball CSP_BGA | BC-400-1 |
| ADSP-BF544BBCZ-5A | –40°C to +85°C | 533 MHz | 400-Ball CSP_BGA | BC-400-1 |
| ADSP-BF544MBBCZ-5M | –40°C to +85°C | 533 MHz | 400-Ball CSP_BGA | BC-400-1 |
| ADSP-BF547BBCZ-5A | –40°C to +85°C | 533 MHz | 400-Ball CSP_BGA | BC-400-1 |
| ADSP-BF547MBBCZ-5M | –40°C to +85°C | 533 MHz | 400-Ball CSP_BGA | BC-400-1 |
| ADSP-BF547KBCZ-6A | 0°C to +70°C | 600 MHz | 400-Ball CSP_BGA | BC-400-1 |
| ADSP-BF548MBBCZ-5M | –40°C to +85°C | 533 MHz | 400-Ball CSP_BGA | BC-400-1 |
| ADSP-BF548BBCZ-5A | –40°C to +85°C | 533 MHz | 400-Ball CSP_BGA | BC-400-1 |

[1] Each ADSP-BF54xM model contains a mobile DDR controller and does not support the use of standard DDR memory.
[2] Z = RoHS Compliant Part.
[3] The ADSP-BF549 is available for automotive use only. Please contact your local ADI product representative or authorized distributor for specific automotive product ordering information.
[4] Referenced temperature is ambient temperature.

©2010 Analog Devices, Inc. All rights reserved. Trademarks and registered trademarks are the property of their respective owners.
D06512-0-2/10(C)

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