# Blackfin+ Core Embedded Processor

# ADSP-BF700/701/702/703/704/705/706/707

## FEATURES

**Blackfin+ core with up to 400 MHz performance**
- **Dual 16-bit or single 32-bit MAC support per cycle**
- **16-bit complex MAC and many other instruction set enhancements**
- **Instruction set compatible with previous Blackfin products**

**Low-cost packaging**
- **88-Lead LFCSP_VQ (QFN) package (12 mm × 12 mm), RoHS compliant**
- **184-Ball CSP_BGA package (12 mm × 12 mm × 0.8 mm pitch), RoHS compliant**

**Low system power with < 100 mW core domain power at 400 MHz (< 0.25 mW/MHz) at 25°C $T_{JUNCTION}$**

## PERIPHERALS FEATURES

**See Figure 1, Processor Block Diagram and Table 1, Processor Comparison**

## MEMORY

**136 kB L1 SRAM with multi-parity-bit protection (64 kB instruction, 64 kB data, 8 kB scratchpad)**

**Large on-chip L2 SRAM with ECC protection**
- **256 kB, 512 kB, 1 MB variants**

**On-chip L2 ROM (512 kB)**

**L3 interface (CSP_BGA only) optimized for lowest system power, providing 16-bit interface to DDR2 or LPDDR DRAM devices (up to 200 MHz)**

**Security and one-time-programmable memory**
- **Crypto hardware accelerators**
- **Fast secure boot for IP protection**
- **memDMA encryption/decryption for fast run-time security**

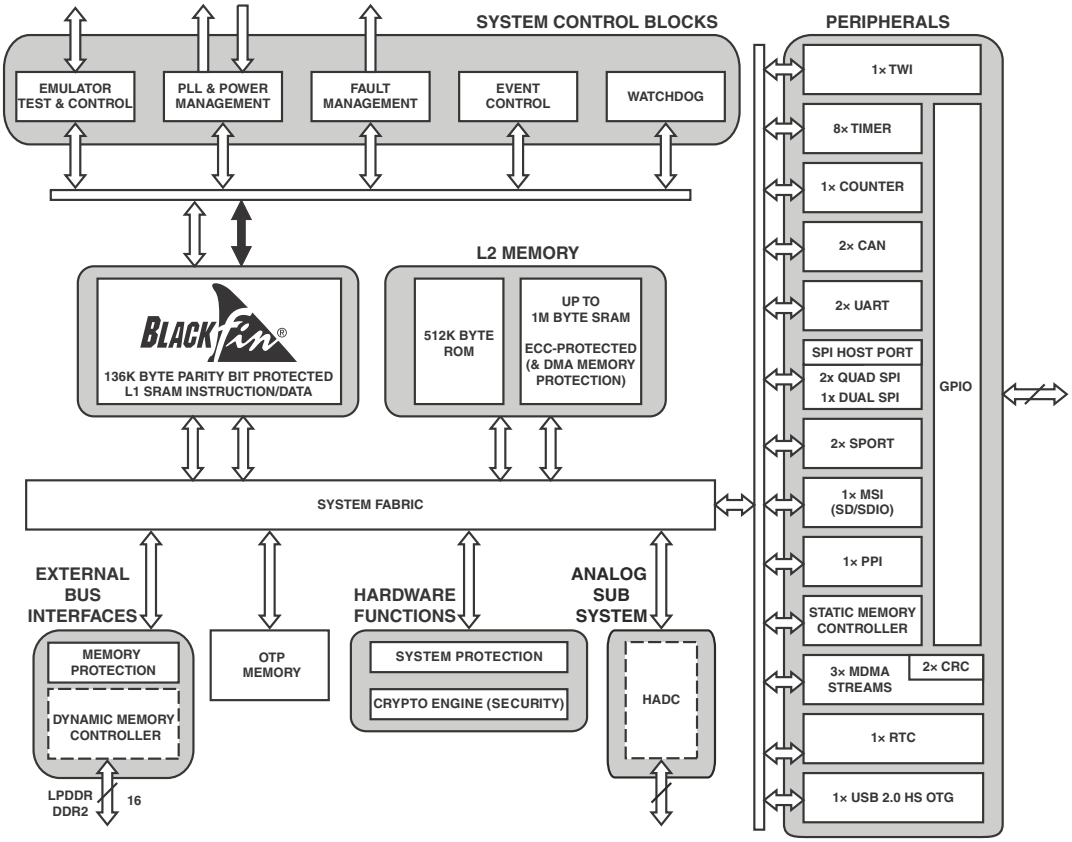



*Figure 1. Processor Block Diagram*

Blackfin+ is a trademark of Analog Devices, Inc.; Blackfin and the Blackfin logo are registered trademarks of Analog Devices, Inc.

**Rev. A** Document Feedback

**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** **©2015 Analog Devices, Inc. All rights reserved.**
**Technical Support** **www.analog.com**

# TABLE OF CONTENTS


General Description ..... 3
Blackfin+ Processor Core ..... 4
Instruction Set Description ..... 5
Processor Infrastructure ..... 5
Memory Architecture ..... 7
Security Features ..... 8
Processor Safety Features ..... 8
Additional Processor Peripherals ..... 9
Power and Clock Management ..... 12
System Debug ..... 15
Development Tools ..... 15
Additional Information ..... 16
Related Signal Chains ..... 16
Security Features Disclaimer ..... 17
ADSP-BF70x Detailed Signal Descriptions ..... 18
184-Ball CSP_BGA Signal Descriptions ..... 22
GPIO Multiplexing for 184-Ball CSP_BGA ..... 29
12 mm × 12 mm 88-Lead LFCSP (QFN) Signal Descriptions ..... 31
GPIO Multiplexing for 12 mm × 12 mm 88-Lead LFCSP (QFN) ..... 36
ADSP-BF70x Designer Quick Reference ..... 38
Specifications ..... 50
Operating Conditions ..... 50
Electrical Characteristics ..... 53
HADC ..... 58
Package Information ..... 59
Absolute Maximum Ratings ..... 59
ESD Sensitivity ..... 59
Timing Specifications ..... 60
Output Drive Currents ..... 102
Test Conditions ..... 104
Environmental Conditions ..... 106
ADSP-BF70x 184-Ball CSP_BGA Ball Assignments (Numerical by Ball Number) ..... 107
ADSP-BF70x 12 mm × 12 mm 88-Lead LFCSP (QFN) Lead Assignments (Numerical by Lead Number) ..... 110
Outline Dimensions ..... 113
Surface-Mount Design ..... 114
Planned Automotive Production Products ..... 115
Ordering Guide ..... 116


## REVISION HISTORY

**9/15—Rev. 0 to Rev. A**

Updated Processor Comparison ..... 3
Updated Serial Ports (SPORTs) ..... 10
Updated Mobile Storage Interface (MSI) ..... 11
Updated External Components for RTC ..... 13
Updated Development Tools ..... 15
Updated SPI Port—SPI_RDY Timing ..... 92
Added Models to Ordering Guide ..... 116

# GENERAL DESCRIPTION

The ADSP-BF70x processor is a member of the Blackfin® family of products. The Blackfin processor combines a dual-MAC 16-bit 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. New enhancements to the Blackfin+ core add 32-bit MAC and 16-bit complex MAC support, cache enhancements, branch prediction and other instruction set improvements—all while maintaining instruction set compatibility to previous Blackfin products.

The processor offers performance up to 400 MHz, as well as low static power consumption. Produced with a low-power and low-voltage design methodology, they provide world-class power management and performance.

By integrating a rich set of industry-leading system peripherals and memory (shown in Table 1), the Blackfin processor is the platform of choice for next-generation applications that require RISC-like programmability, multimedia support, and leading-edge signal processing in one integrated package. These applications span a wide array of markets, from automotive systems to embedded industrial, instrumentation, video/image analysis, biometric and power/motor control applications.

**Table 1. Processor Comparison**

| Processor Feature | | ADSP-BF700 | ADSP-BF701 | ADSP-BF702 | ADSP-BF703 | ADSP-BF704 | ADSP-BF705 | ADSP-BF706 | ADSP-BF707 |
|---|---|---|---|---|---|---|---|---|---|
| Maximum Speed Grade (MHz)[1] | | 200 | | 400 | | | | | |
| Maximum SYSCLK (MHz) | | 100 | | 200 | | | | | |
| Package Options | | 88-Lead LFCSP | 184-Ball CSP_BGA | 88-Lead LFCSP | 184-Ball CSP_BGA | 88-Lead LFCSP | 184-Ball CSP_BGA | 88-Lead LFCSP | 184-Ball CSP_BGA |
| GPIOs | | 43 | 47 | 43 | 47 | 43 | 47 | 43 | 47 |
| Memory (bytes) | L1 Instruction SRAM | 48K | | | | | | | |
| | L1 Instruction SRAM/Cache | 16K | | | | | | | |
| | L1 Data SRAM | 32K | | | | | | | |
| | L1 Data SRAM/Cache | 32K | | | | | | | |
| | L1 Scratchpad (L1 Data C) | 8K | | | | | | | |
| | L2 SRAM | 128K | | 256K | | 512K | | 1024K | |
| | L2 ROM | 512K | | | | | | | |
| | DDR2/LPDDR (16-bit) | No | Yes | No | Yes | No | Yes | No | Yes |
| $I^2C$ | | 1 | | | | | | | |
| Up/Down/Rotary Counter | | 1 | | | | | | | |
| GP Timer | | 8 | | | | | | | |
| Watchdog Timer | | 1 | | | | | | | |
| GP Counter | | 1 | | | | | | | |
| SPORTs | | 2 | | | | | | | |
| Quad SPI | | 2 | | | | | | | |
| Dual SPI | | 1 | | | | | | | |
| SPI Host Port | | 1 | | | | | | | |
| USB 2.0 HS OTG | | 1 | | | | | | | |
| Parallel Peripheral Interface | | 1 | | | | | | | |
| CAN | | 2 | | | | | | | |
| UART | | 2 | | | | | | | |
| Real-Time Clock | | 1 | | | | | | | |
| Static Memory Controller (SMC) | | Yes | | | | | | | |
| Security Crypto Engine | | Yes | | | | | | | |
| SD/SDIO (MSI) | | 4-bit | 8-bit | 4-bit | 8-bit | 4-bit | 8-bit | 4-bit | 8-bit |
| 4-Channel 12-Bit ADC | | No | Yes | No | Yes | No | Yes | No | Yes |

[1] Other speed grades available.

## BLACKFIN+ PROCESSOR CORE

As shown in Figure 1, the processor integrates a Blackfin+ processor core. The core, shown in Figure 2, contains two 16-bit multipliers, one 32-bit multiplier, two 40-bit accumulators (which may be used together as a 72-bit accumulator), two 40-bit ALUs, one 72-bit ALU, 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.

The core can perform two 16-bit by 16-bit multiply-accumulates or one 32-bit multiply-accumulate in each cycle. Signed and unsigned formats, rounding, saturation, and complex multiplies are supported.

The ALUs perform a traditional set of arithmetic and logical operations on 16-bit 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, 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). If a second ALU is used, 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.

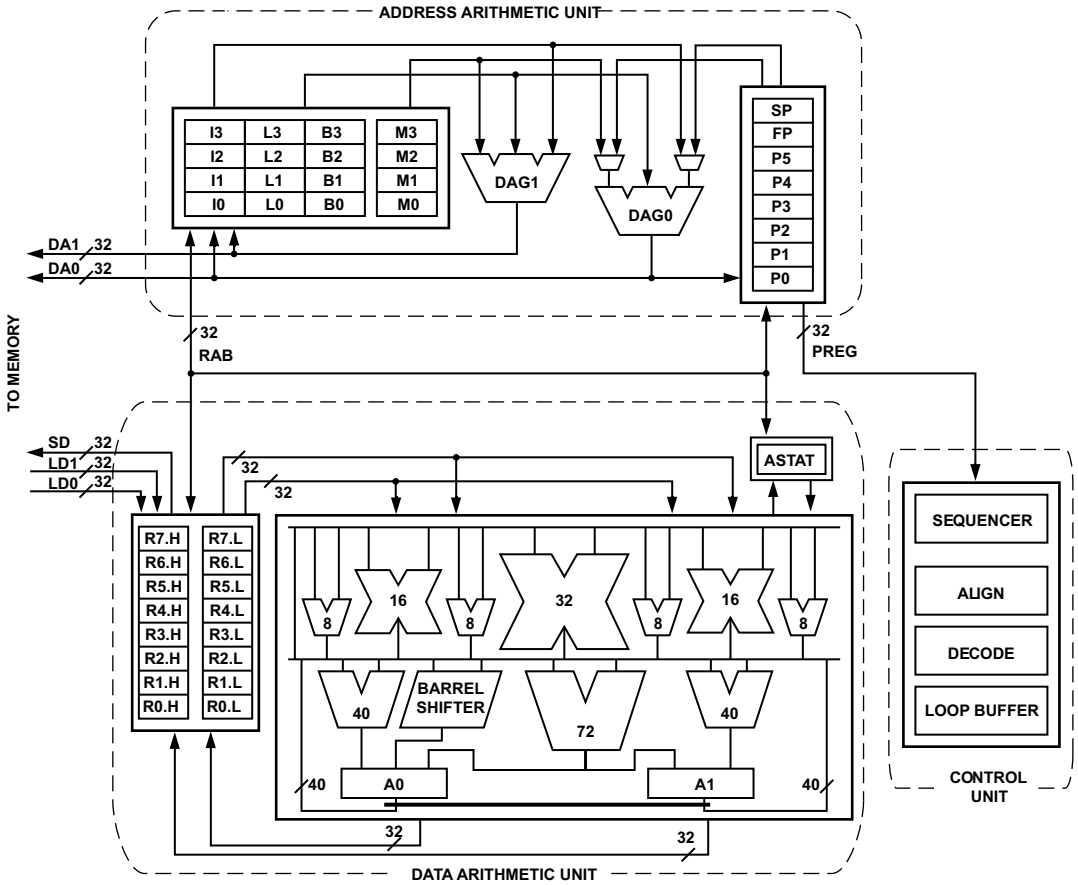



*Figure 2. Blackfin+ Processor Core*

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 dynamic branch prediction), and subroutine calls. Hardware supports 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).

The Blackfin processor supports 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 data memory holds 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.

## INSTRUCTION SET DESCRIPTION

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. The Blackfin processor supports 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 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.
- Control of all asynchronous and synchronous events to the processor is handled by two subsystems: the core event controller (CEC) and the system event controller (SEC).
- 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-bit and 32-bit instructions (no mode switching, no code segregation). Frequently used instructions are encoded in 16 bits.

## PROCESSOR INFRASTRUCTURE

The following sections provide information on the primary infrastructure components of the ADSP-BF70x processor.

### *DMA Controllers*

The processor uses direct memory access (DMA) to transfer data within memory spaces or between a memory space and a peripheral. The processor can specify data transfer operations and return to normal processing while the fully integrated DMA controller carries out the data transfers independent of processor activity.

DMA transfers can occur between memory and a peripheral or between one memory and another memory. Each memory-to-memory DMA stream uses two channels, where one channel is the source channel, and the second is the destination channel.

All DMAs can transport data to and from all on-chip and off-chip memories. Programs can use two types of DMA transfers, descriptor-based or register-based. Register-based DMA allows the processor to directly program DMA control registers to initiate a DMA transfer. On completion, the control registers may be automatically updated with their original setup values for continuous transfer. Descriptor-based DMA transfers require a set of parameters stored within memory to initiate a DMA sequence. Descriptor-based DMA transfers allow multiple DMA sequences to be chained together and a DMA channel can be programmed to automatically set up and start another DMA transfer after the current sequence completes.

The DMA controller supports the following DMA operations.

- A single linear buffer that stops on completion.
- A linear buffer with negative, positive, or zero stride length.
- A circular, auto-refreshing buffer that interrupts when each buffer becomes full.

- A similar buffer that interrupts on fractional buffers (for example, 1/2, 1/4).
- 1D DMA—uses a set of identical ping-pong buffers defined by a linked ring of two-word descriptor sets, each containing a link pointer and an address.
- 1D DMA—uses a linked list of 4 word descriptor sets containing a link pointer, an address, a length, and a configuration.
- 2D DMA—uses an array of one-word descriptor sets, specifying only the base DMA address.
- 2D DMA—uses a linked list of multi-word descriptor sets, specifying everything.

### *Event Handling*

The processor provides 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 processor 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 through the JTAG interface.
- Reset—This event resets the processor.
- Nonmaskable interrupt (NMI)—The NMI event can be generated either by the software watchdog timer, by the $\overline{\text{NMI}}$ input signal to the processor, or by software. 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 (in other words, 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 signals, timers, and other peripherals, as well as by an explicit software instruction.

**System Event Controller (SEC)**

The SEC manages the enabling, prioritization, and routing of events from each system interrupt or fault source. Additionally, it provides notification and identification of the highest priority active system interrupt request to the core and routes system fault sources to its integrated fault management unit. The SEC triggers core general-purpose interrupt IVG11. It is recommended that IVG11 be set to allow self-nesting. The four lower priority interrupts (IVG15-12) may be used for software interrupts.

### *Trigger Routing Unit (TRU)*

The TRU provides system-level sequence control without core intervention. The TRU maps trigger masters (generators of triggers) to trigger slaves (receivers of triggers). Slave endpoints can be configured to respond to triggers in various ways. Common applications enabled by the TRU include:

- Automatically triggering the start of a DMA sequence after a sequence from another DMA channel completes
- Software triggering
- Synchronization of concurrent activities

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

Each general-purpose port pin can be individually controlled by manipulation of the port control, status, and interrupt registers:

- GPIO direction control register—Specifies the direction of each individual GPIO pin as input or output.
- GPIO control and status registers—A write one to modify mechanism allows any combination of individual GPIO pins to be modified in a single instruction, without affecting the level of any other GPIO pins.
- GPIO interrupt mask registers—Allow each individual GPIO pin to function as an interrupt to the processor. GPIO pins defined as inputs can be configured to generate hardware interrupts, while output pins can be triggered by software interrupts.
- GPIO interrupt sensitivity registers—Specify whether individual pins are level- or edge-sensitive and specify—if edge-sensitive—whether just the rising edge or both the rising and falling edges of the signal are significant.

### *Pin Interrupts*

Every port pin on the processor can request interrupts in either an edge-sensitive or a level-sensitive manner with programmable polarity. Interrupt functionality is decoupled from GPIO operation. Three system-level interrupt channels (PINT0–3) 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 enable half-port assignment and interrupt management. This includes masking, identification, and clearing of requests. These registers also enable 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.

### *Pin Multiplexing*

The processor supports a flexible multiplexing scheme that multiplexes the GPIO pins with various peripherals. A maximum of 4 peripherals plus GPIO functionality is shared by each GPIO pin. All GPIO pins have a bypass path feature—that is, when the

output enable and the input enable of a GPIO pin are both active, the data signal before the pad driver is looped back to the receive path for the same GPIO pin.

## MEMORY ARCHITECTURE

The processor views 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 core-accessible memory as cache or SRAM, and larger, lower-cost and performance interface-accessible memory systems. See Figure 3.

### *Internal (Core-Accessible) Memory*

The L1 memory system is the highest-performance memory available to the Blackfin+ processor core.

The core has its own private L1 memory. The modified Harvard architecture supports two concurrent 32-bit data accesses along with an instruction fetch at full processor speed which provides high-bandwidth processor performance. In the core, a 64K byte block of data memory partners with an 64K byte memory block for instruction storage. Each data block is multibanked for efficient data exchange through DMA and can be configured as SRAM. Alternatively, 16K bytes of each block can be configured in L1 cache mode. The four-way set-associative instruction cache and the 2 two-way set-associative data caches greatly accelerate memory access performance, especially when accessing external memories.

The L1 memory domain also features a 8K byte data SRAM block which is ideal for storing local variables and the software stack. All L1 memory is protected by a multi-parity-bit concept, regardless of whether the memory is operating in SRAM or cache mode.

Outside of the L1 domain, L2 and L3 memories are arranged using a Von Neumann topology. The L2 memory domain is a unified instruction and data memory and can hold any mixture of code and data required by the system design. The L2 memory domain is accessible by the Blackfin+ core through a dedicated 64-bit interface. It operates at SYSCLK frequency.

The processor features up to 1M byte of L2 SRAM, which is ECC-protected and organized in eight banks. Individual banks can be made private to any system master. There is also a 512K byte single-bank ROM in the L2 domain. It contains boot code, security code, and general-purpose ROM space.

### *OTP Memory*

The processor features 4 kB of one-time-programmable (OTP) memory which is memory-map accessible. This memory stores a unique chip identification and is used to support secure-boot and secure operation.

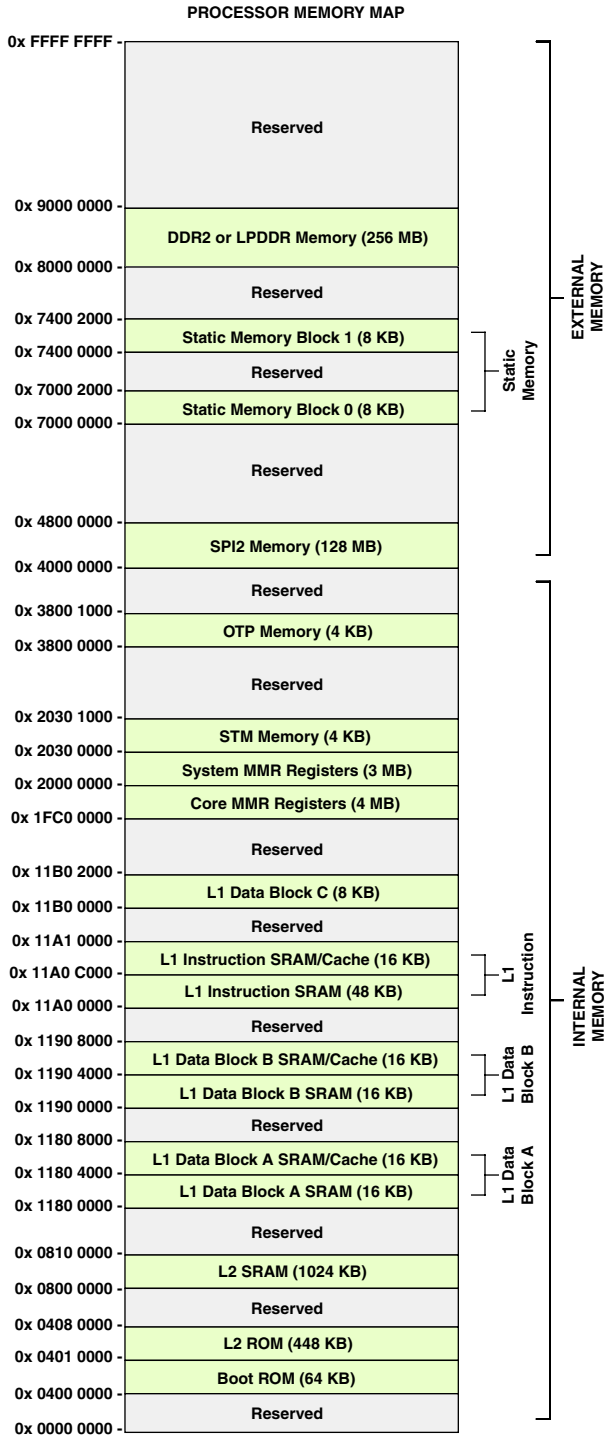



*Figure 3. ADSP-BF706/ADSP-BF707 Internal/External Memory Map*

### *Static Memory Controller (SMC)*

The SMC can be programmed to control up to two blocks of external memories or memory-mapped devices, with very flexible timing parameters. Each block occupies a 8K byte segment regardless of the size of the device used.

### *Dynamic Memory Controller (DMC)*

The DMC includes a controller that supports JESD79-2E compatible double-data-rate (DDR2) SDRAM and JESD209A low-power DDR (LPDDR) SDRAM devices. The DMC PHY features on-die termination on all data and data strobe pins that can be used during reads.

### *I/O Memory Space*

The processor does 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 in a region of the 4G byte address space. These are separated into two smaller blocks, one which contains the control MMRs for all core functions, and the other which contains 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 processor has several mechanisms for automatically loading internal and external memory after a reset. The boot mode is defined by the SYS_BMODE input pins dedicated for this purpose. There are two categories of boot modes. In master boot mode, the processor actively loads data from serial memories. In slave boot modes, the processor receives data from external host devices.

The boot modes are shown in Table 2. These modes are implemented by the SYS_BMODE bits of the reset configuration register and are sampled during power-on resets and software-initiated resets.

**Table 2. Boot Modes**

| SYS_BMODE Setting | Boot Mode |
|---|---|
| 00 | No Boot/Idle |
| 01 | SPI2 Master |
| 10 | SPI2 Slave |
| 11 | UART0 Slave |

## SECURITY FEATURES

The ADSP-BF70x processor supports standards-based hardware-accelerated encryption, decryption, authentication, and true random number generation.

The following hardware-accelerated cryptographic ciphers are supported:

- AES in ECB, CBC, ICM, and CTR modes with 128-, 192-, and 256-bit keys
- DES in ECB and CBC mode with 56-bit key
- 3DES in ECB and CBC mode with 3x 56-bit key

The following hardware-accelerated hash functions are supported:

- SHA-1
- SHA-2 with 224-bit and 256-bit digest
- HMAC transforms for SHA-1 and SHA-2

Public key accelerator is available to offload computation-intensive public key cryptography operations.

Both a hardware-based nondeterministic random number generator and pseudo-random number generator are available. The TRNG also provides HW post-processing to meet NIST requirements of FIPS 140-2, while the PRNG is ANSI X9.31 compliant.

Secure boot is also available with 224-bit elliptic curve digital signatures ensuring integrity and authenticity of the boot stream. Optionally, confidentiality is also ensured through AES-128 encryption.

**CAUTION**
This product includes security features that can be used to protect embedded nonvolatile memory contents and prevent execution of unauthorized code. When security is enabled on this device (either by the ordering party or the subsequent receiving parties), the ability of Analog Devices to conduct failure analysis on returned devices is limited. Contact Analog Devices for details on the failure analysis limitations for this device.

Secure debug is also employed to allow only trusted users to access the system with debug tools.

## PROCESSOR SAFETY FEATURES

The ADSP-BF70x processor has been designed for functional safety applications. While the level of safety is mainly dominated by the system concept, the following primitives are provided by the devices to build a robust safety concept.

### *Multi-Parity-Bit-Protected L1 Memories*

In the processor's L1 memory space, whether SRAM or cache, each word is protected by multiple parity bits to detect the single event upsets that occur in all RAMs. This applies both to L1 instruction and data memory spaces.

### *ECC-Protected L2 Memories*

Error correcting codes (ECC) are used to correct single event upsets. The L2 memory is protected with a single error correct-double error detect (SEC-DED) code. By default ECC is enabled, but it can be disabled on a per-bank basis. Single-bit errors are transparently corrected. Dual-bit errors can issue a

system event or fault if enabled. ECC protection is fully transparent to the user, even if L2 memory is read or written by 8-bit or 16-bit entities.

### CRC-Protected Memories

While parity bit and ECC protection mainly protect against random soft errors in L1 and L2 memory cells, the CRC engines can be used to protect against systematic errors (pointer errors) and static content (instruction code) of L1, L2, and even L3 memories (DDR2, LPDDR). The processor features two CRC engines which are embedded in the memory-to-memory DMA controllers. CRC checksums can be calculated or compared on the fly during memory transfers, or one or multiple memory regions can be continuously scrubbed by a single DMA work unit as per DMA descriptor chain instructions. The CRC engine also protects data loaded during the boot process.

### Memory Protection

The Blackfin+ core features a memory protection concept, which grants data and/or instruction accesses to enabled memory regions only. A supervisor mode vs. user mode programming model supports dynamically varying access rights. Increased flexibility in memory page size options supports a simple method of static memory partitioning.

### System Protection

The system protection unit (SPU) guards against accidental or unwanted access to the MMR space of a peripheral by providing a write-protection mechanism. The user is able to choose and configure the peripherals that are protected as well as configure which ones of the four system MMR masters (core, memory DMA, the SPI host port, and Coresight debug) the peripherals are guarded against.

The SPU is also part of the security infrastructure. Along with providing write-protection functionality, the SPU is employed to define which resources in the system are secure or non-secure and to block access to secure resources from non-secure masters.

Synonymously, the system memory protection unit (SMPU) provides memory protection against read and/or write transactions to defined regions of memory. There are two SMPU units in the ADSP-BF70x processors. One is for the L2 memory and the other is for the external DDR memory.

The SMPU is also part of the security infrastructure. It allows the user to not only protect against arbitrary read and/or write transactions, but it also allows regions of memory to be defined as secure and prevent non-secure masters from accessing those memory regions.

### Watchpoint Protection

The primary purpose of watchpoints and hardware breakpoints is to serve emulator needs. When enabled, they signal an emulator event whenever user-defined system resources are accessed or the core executes from user-defined addresses. Watchpoint events can be configured such that they signal the events to the fault management unit of the SEC.

### Watchdog

The on-chip software watchdog timer can supervise the Blackfin+ core.

### Bandwidth Monitor

Memory-to-memory DMA channels are equipped with a bandwidth monitor mechanism. They can signal a system event or fault when transactions tend to starve because system buses are fully loaded with higher-priority traffic.

### Signal Watchdogs

The eight general-purpose timers feature modes to monitor off-chip signals. The watchdog period mode monitors whether external signals toggle with a period within an expected range. The watchdog width mode monitors whether the pulse widths of external signals are within an expected range. Both modes help to detect undesired toggling (or lack thereof) of system-level signals.

### Up/Down Count Mismatch Detection

The GP counter can monitor external signal pairs, such as request/grant strobes. If the edge count mismatch exceeds the expected range, the GP counter can flag this to the processor or to the fault management unit of the SEC.

### Fault Management

The fault management unit is part of the system event controller (SEC). Any system event, whether a dual-bit uncorrectable ECC error, or any peripheral status interrupt, can be defined as being a fault. Additionally, the system events can be defined as an interrupt to the core. If defined as such, the SEC forwards the event to the fault management unit, which may automatically reset the entire device for reboot, or simply toggle the $\overline{\text{SYS_FAULT}}$ output pin to signal off-chip hardware. Optionally, the fault management unit can delay the action taken through a keyed sequence, to provide a final chance for the Blackfin+ core to resolve the issue and to prevent the fault action from being taken.

## ADDITIONAL PROCESSOR PERIPHERALS

The processor contains a rich set of peripherals connected to the core through several high-bandwidth buses, providing flexibility in system configuration as well as excellent overall system performance (see the block diagram on Page 1). The processor contains 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.

The following sections describe additional peripherals that were not previously described.

### Timers

The processor includes several timers which are described in the following sections.

**General-Purpose Timers**

There is one GP timer unit, and it provides eight general-purpose programmable timers. 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 TIMER_TMRx pins, an external TIMER_CLK input pin, or to the internal SCLK0.

These timer units can be used in conjunction with the UARTs and the CAN controller 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 GP timers can generate interrupts to the processor core, providing periodic events for synchronization to either the system clock or to external signals. Timer events can also trigger other peripherals through the TRU (for instance, to signal a fault). Each timer may also be started and/or stopped by any TRU master without core intervention.

**Core Timer**

The processor core also has its own dedicated timer. This extra timer is clocked by the internal processor clock and is typically used as a system tick clock for generating periodic operating system interrupts.

**Watchdog Timer**

The core includes a 32-bit timer, which may be used to implement a software watchdog function. A software watchdog can improve system availability by forcing the processor to a known state, through generation of a hardware reset, nonmaskable 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, then enables the timer. Thereafter, the software must reload the counter before it counts down to zero from the programmed value. This protects the system from remaining in an unknown state where software that would normally reset the timer has stopped running due to an external noise condition or software error.

After a reset, software can determine if the watchdog was the source of the hardware reset by interrogating a status bit in its timer control register that is set only upon a watchdog-generated reset.

### *Serial Ports (SPORTs)*

Two synchronous serial ports (comprised of four half-SPORTs) provide an inexpensive interface to a wide variety of digital and mixed-signal peripheral devices such as Analog Devices' audio codecs, ADCs, and DACs. Each half-SPORT is made up of two data lines, a clock, and frame sync. The data lines can be programmed to either transmit or receive and each data line has a dedicated DMA channel.

Serial port data can be automatically transferred to and from on-chip memory/external memory through dedicated DMA channels. Each of the serial ports can work in conjunction with another serial port to provide TDM support. In this configuration, one SPORT provides two transmit signals while the other SPORT provides the two receive signals. The frame sync and clock are shared.

Serial ports operate in six modes:

- Standard DSP serial mode
- Multichannel (TDM) mode
- $I^2S$ mode
- Packed $I^2S$ mode
- Left-justified mode
- Right-justified mode

### *General-Purpose Counters*

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

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

Internal signals forwarded to a GP timer enable this timer to measure the intervals between count events. Boundary registers enable auto-zero operation or simple system warning by interrupts when programmed count values are exceeded.

### *Parallel Peripheral Interface (PPI)*

The processor provides a parallel peripheral interface (PPI) that supports data widths up to 18 bits. The PPI 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 PPI module:

- Programmable data length: 8 bits, 10 bits, 12 bits, 14 bits, 16 bits, and 18 bits per clock.
- Various framed, non-framed, and general-purpose operating modes. Frame syncs can be generated internally or can be supplied by an external device.
- ITU-656 status word error detection and correction for ITU-656 receive modes and ITU-656 preamble and status word decode.
- Optional packing and unpacking of data to/from 32 bits from/to 8 bits, 16 bits and 24 bits. If packing/unpacking is enabled, endianness can be configured to change the order of packing/unpacking of bytes/words.
- RGB888 can be converted to RGB666 or RGB565 for transmit modes.
- Various de-interleaving/interleaving modes for receiving/transmitting 4:2:2 YCrCb data.
- Configurable LCD data enable (DEN) output available on Frame Sync 3.

### Serial Peripheral Interface (SPI) Ports

The processors have three industry-standard SPI-compatible ports that allow it to communicate with multiple SPI-compatible devices.

The baseline SPI peripheral is a synchronous, four-wire interface consisting of two data pins, one device select pin, and a gated clock pin. The two data pins allow full-duplex operation to other SPI-compatible devices. An additional two (optional) data pins are provided to support quad SPI operation. Enhanced modes of operation such as flow control, fast mode, and dual I/O mode (DIOM) are also supported. In addition, a direct memory access (DMA) mode allows for transferring several words with minimal CPU interaction.

With a range of configurable options, the SPI ports provide a glueless hardware interface with other SPI-compatible devices in master mode, slave mode, and multimaster environments. The SPI peripheral includes programmable baud rates, clock phase, and clock polarity. The peripheral can operate in a multimaster environment by interfacing with several other devices, acting as either a master device or a slave device. In a multimaster environment, the SPI peripheral uses open-drain outputs to avoid data bus contention. The flow control features enable slow slave devices to interface with fast master devices by providing an SPI Ready pin which flexibly controls the transfers.

The SPI port's baud rate and clock phase/polarities are programmable, and it has integrated DMA channels for both transmit and receive data streams.

### SPI Host Port (SPIHP)

The processor includes one SPI host port which may be used in conjunction with any available SPI port to enhance its SPI slave mode capabilities. The SPIHP allows a SPI host device access to memory-mapped resources of the processor through a SPI SRAM/FLASH style protocol. The following features are included:

- Direct read/write of memory and memory-mapped registers
- Support for pre-fetch for faster reads
- Support for SPI controllers that implement hardware-based SPI memory protocol
- Error capture and reporting for protocol errors, bus errors, and over/underflow

### UART Ports

The processor provides two full-duplex universal asynchronous receiver/transmitter (UART) ports, which are fully compatible with PC-standard UARTs. 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, and none, even, or odd parity. Optionally, an additional address bit can be transferred to interrupt only addressed nodes in multi-drop bus (MDB) systems. A frame is terminated by a configurable number of stop bits.

The UART ports support automatic hardware flow control through the clear to send (CTS) input and request to send (RTS) output with programmable assertion FIFO levels.

To help support the local interconnect network (LIN) protocols, a special command causes the transmitter to queue a break command of programmable bit length into the transmit buffer. Similarly, the number of stop bits can be extended by a programmable inter-frame space.

The capabilities of the UARTs are further extended with support for the Infrared Data Association (IrDA®) serial infrared physical layer link specification (SIR) protocol.

### 2-Wire Controller Interface (TWI)

The processor includes a 2-wire interface (TWI) module for providing a simple exchange method of control data between multiple devices. The TWI module is compatible with the widely used $I^2C$ bus standard. The TWI module offers the capabilities of simultaneous master and slave operation and support for both 7-bit addressing and multimedia data arbitration. The TWI interface utilizes two pins for transferring clock (TWI_SCL) and data (TWI_SDA) and supports the protocol at speeds up to 400k bits/sec. The TWI interface pins are compatible with 5 V logic levels.

Additionally, the TWI module is fully compatible with serial camera control bus (SCCB) functionality for easier control of various CMOS camera sensor devices.

### Mobile Storage Interface (MSI)

The mobile storage interface (MSI) controller acts as the host interface for multimedia cards (MMC), secure digital memory cards (SD), and secure digital input/output cards (SDIO). The following list describes the main features of the MSI controller:

- Support for a single MMC, SD memory, and SDIO card
- Support for 1-bit and 4-bit SD modes
- Support for 1-bit, 4-bit, and 8-bit MMC modes
- Support for eMMC 4.5 embedded NAND flash devices
- Support for power management and clock control
- An eleven-signal external interface with clock, command, optional interrupt, and up to eight data lines
- Card interface clock generation from SCLK0 or SCLK1
- SDIO interrupt and read wait features

### Controller Area Network (CAN)

A CAN controller implements the 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. This is because the protocol incorporates CRC checking, message error tracking, and fault node confinement.

The CAN controller offers 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 wake-up from hibernation mode (lowest static power consumption mode)
- Interrupts, including: TX complete, RX complete, error and global

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

### *USB 2.0 On-the-Go Dual-Role Device Controller*

The USB 2.0 on-the-go (OTG) dual-role device controller provides a low-cost connectivity solution for the growing adoption of this bus standard in industrial applications, as well as consumer mobile devices such as cell phones, digital still cameras, and MP3 players. The USB 2.0 controller allows these devices to transfer data using a point-to-point USB connection without the need for a PC host. The module can operate in a traditional USB peripheral-only mode as well as the host mode presented in the OTG supplement to the USB 2.0 specification.

The USB clock is provided through a dedicated external crystal or crystal oscillator.

The USB OTG dual-role device controller includes a phase locked loop with programmable multipliers to generate the necessary internal clocking frequency for USB.

### *Housekeeping ADC (HADC)*

The HADC provides a general-purpose, multichannel successive approximation analog-to-digital converter. It supports the following features:

- 12-bit ADC core (10-bit accuracy) with built-in sample and hold
- 4 single-ended input channels
- Throughput rates up to 1 MSPS
- Single external reference with analog inputs between 0 V and 3.3 V
- Selectable ADC clock frequency including the ability to program a prescaler
- Adaptable conversion type: allows single or continuous conversion with option of autoscan
- Auto sequencing capability with up to 4 autoconversions in a single session. Each conversion can be programmed to select any input channel.
- Four data registers (individually addressable) to store conversion values

### *System Crossbars (SCB)*

The system crossbars (SCB) are the fundamental building blocks of a switch-fabric style for (on-chip) system bus interconnection. The SCBs connect system bus masters to system bus slaves, providing concurrent data transfer between multiple bus masters and multiple bus slaves. A hierarchical model—built from multiple SCBs—provides a power and area efficient system interconnect, which satisfies the performance and flexibility requirements of a specific system.

The SCBs provide the following features:

- Highly efficient, pipelined bus transfer protocol for sustained throughput
- Full-duplex bus operation for flexibility and reduced latency
- Concurrent bus transfer support to allow multiple bus masters to access bus slaves simultaneously
- Protection model (privileged/secure) support for selective bus interconnect protection

## POWER AND CLOCK MANAGEMENT

The processor provides three operating modes, each with a different performance/power profile. Control of clocking to each of the processor peripherals also reduces power consumption. See Table 5 for a summary of the power settings for each mode.

### *System Crystal Oscillator and USB Crystal Oscillator*

The processor can be clocked by an external crystal (see Figure 4), 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 SYS_CLKIN pin of the processor. When an external clock is used, the SYS_XTAL pin must be left unconnected. Alternatively, because the processor includes an on-chip oscillator circuit, an external crystal may be used.

For fundamental frequency operation, use the circuit shown in Figure 4. A parallel-resonant, fundamental frequency, microprocessor grade crystal is connected across the SYS_CLKIN and SYS_XTAL pins. The on-chip resistance between SYS_CLKIN and the SYS_XTAL pin is in the 500 kΩ range. Further parallel resistors are typically not recommended.

The two capacitors and the series resistor shown in Figure 4 fine-tune phase and amplitude of the sine frequency. The capacitor and resistor values shown in Figure 4 are typical values only. The capacitor values are dependent upon the load capacitance recommendations of the crystal manufacturer and the PCB physical layout. The resistor value depends on the drive

level specified by the crystal manufacturer. The user should verify the customized values based on careful investigations on multiple devices over the required temperature range.

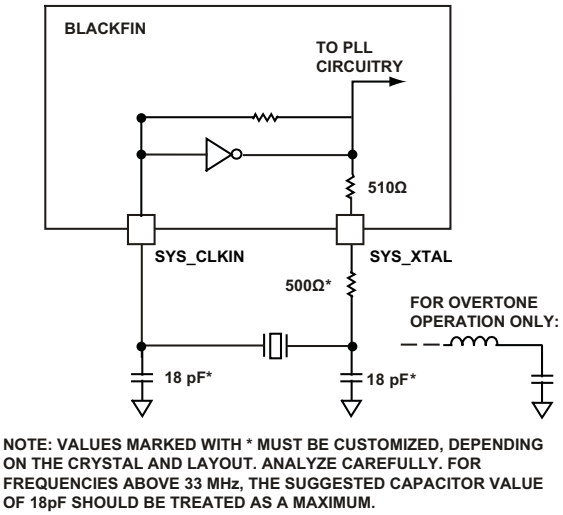



*Figure 4. External Crystal Connection*

A third-overtone crystal can be used for 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 4. A design procedure for third-overtone operation is discussed in detail in application note (EE-168) *Using Third Overtone Crystals with the ADSP-218x DSP* (www.analog.com/ee-168).

The same recommendations may be used for the USB crystal oscillator.

### Real-Time Clock

The 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 processor. Connect RTC pins RTC_CLKIN and RTC_XTAL with external components as shown in Figure 5.

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 the alarm interrupt is 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 specific day and time of that day.

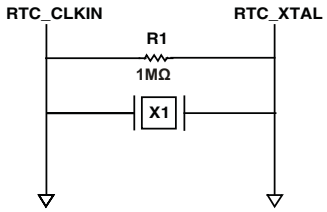



*Figure 5. External Components for RTC*

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

### Clock Generation

The clock generation unit (CGU) generates all on-chip clocks and synchronization signals. Multiplication factors are programmed to define the PLLCLK frequency. Programmable values divide the PLLCLK frequency to generate the core clock (CCLK), the system clocks (SYSCLK, SCLK0, and SCLK1), the LPDDR or DDR2 clock (DCLK), and the output clock (OCLK).

Writing to the CGU control registers does not affect the behavior of the PLL immediately. Registers are first programmed with a new value, and the PLL logic executes the changes so that it transitions smoothly from the current conditions to the new ones.

SYS_CLKIN oscillations start when power is applied to the VDD_EXT pins. The rising edge of $\overline{\text{SYS_HWRST}}$ can be applied after all voltage supplies are within specifications, and SYS_CLKIN oscillations are stable.

### Clock Out/External Clock

The SYS_CLKOUT output pin has programmable options to output divided-down versions of the on-chip clocks. By default, the SYS_CLKOUT pin drives a buffered version of the SYS_CLKIN input. Clock generation faults (for example, PLL unlock) may trigger a reset by hardware. The clocks shown in Table 3 can be output on the SYS_CLKOUT pin.

**Table 3. Clock Dividers**

| Clock Source | Divider (if Available on SYS_CLKOUT) |
|---|---|
| CCLK (Core Clock) | By 16 |
| SYSCLK (System Clock) | By 8 |
| SCLK0 (System Clock, All Peripherals not Covered by SCLK1) | Not available on SYS_CLKOUT |
| SCLK1 (System Clock for Crypto Engines and MDMA) | By 8 |
| DCLK (LPDDR/DDR2 Clock) | By 8 |
| OCLK (Output Clock) | Programmable |
| CLKBUF | None, direct from SYS_CLKIN |

### *Power Management*

As shown in Table 4, the processor supports multiple power domains, which maximizes flexibility while maintaining compliance with industry standards and conventions. There are no sequencing requirements for the various power domains, but all domains must be powered according to the appropriate Specifications table for processor operating conditions; even if the feature/peripheral is not used.

**Table 4. Power Domains**

| Power Domain | $V_{DD}$ Range |
|---|---|
| All Internal Logic | $V_{DD_INT}$ |
| DDR2/LPDDR | $V_{DD_DMC}$ |
| USB | $V_{DD_USB}$ |
| OTP Memory | $V_{DD_OTP}$ |
| HADC | $V_{DD_HADC}$ |
| RTC | $V_{DD_RTC}$ |
| All Other I/O (Includes SYS, JTAG, and Ports Pins) | $V_{DD_EXT}$ |

The dynamic power management feature of the processor allows the processor's core clock frequency ($f_{CCLK}$) to be dynamically controlled.

The power dissipated by a processor is largely a function of its clock frequency and the square of the operating voltage. For example, reducing the clock frequency by 25% results in a 25% reduction in dynamic power dissipation.

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 capability for 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.

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

The deep sleep mode maximizes dynamic power savings by disabling the clocks to the processor core and to all synchronous peripherals. Asynchronous peripherals may still be running but cannot access internal resources or external memory.

**Table 5. Power Settings**

| Mode/State | PLL | PLL Bypassed | $f_{CCLK}$ | $f_{SYSCLK}$, $f_{DCLK}$, $f_{SCLK0}$, $f_{SCLK1}$ | Core Power |
|---|---|---|---|---|---|
| Full On | Enabled | No | Enabled | Enabled | On |
| Deep Sleep | Disabled | — | Disabled | Disabled | On |
| Hibernate | Disabled | — | Disabled | Disabled | Off |

#### Hibernate State—Maximum Static Power Savings

The hibernate state maximizes static power savings by disabling the voltage and clocks to the processor core and to all of the peripherals. This setting signals the external voltage regulator supplying the VDD_INT pins to shut off using the SYS_EXTWAKE signal, which provides the lowest static power dissipation.

Any critical information stored internally (for example, memory contents, register contents, and other information) must be written to a nonvolatile storage device (or self-refreshed DRAM) prior to removing power if the processor state is to be preserved.

Because the $V_{DD_EXT}$ pins can still be 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 still have power applied without drawing unwanted current.

### *Reset Control Unit*

Reset is the initial state of the whole processor or the core and is the result of a hardware- or software-triggered event. In this state, all control registers are set to their default values and functional units are idle. Exiting a full system reset starts with the core being ready to boot.

The reset control unit (RCU) controls how all the functional units enter and exit reset. Differences in functional requirements and clocking constraints define how reset signals are generated. Programs must guarantee that none of the reset functions puts the system into an undefined state or causes resources to stall. This is particularly important when the core is reset (programs must ensure that there is no pending system activity involving the core when it is being reset).

From a system perspective, reset is defined by both the reset target and the reset source described as follows in the following list.

Target defined:

- Hardware Reset—All functional units are set to their default states without exception. History is lost.
- System Reset—All functional units except the RCU are set to their default states.
- Core-only Reset—Affects the core only. The system software should guarantee that the core, while in reset state, is not accessed by any bus master.

Source defined:

- Hardware Reset—The $\overline{\text{SYS_HWRST}}$ input signal is asserted active (pulled down).
- System Reset—May be triggered by software (writing to the RCU_CTL register) or by another functional unit such as the dynamic power management (DPM) unit (hibernate) or any of the system event controller (SEC), trigger routing unit (TRU), or emulator inputs.
- Core-only Reset—Triggered by software.
- Trigger request (peripheral).

### *Voltage Regulation*

The processor requires an external voltage regulator to power the VDD_INT pins. To reduce standby power consumption, the external voltage regulator can be signaled through SYS_EXTWAKE to remove power from the processor core. This signal is high-true for power-up and may be connected directly to the low-true shut-down input of many common regulators.

While in the hibernate state, all external supply pins (VDD_EXT, VDD_USB, and VDD_DMC) can still be powered, eliminating the need for external buffers. The external voltage regulator can be activated from this power down state by asserting the $\overline{\text{SYS_HWRST}}$ pin, which then initiates a boot sequence. SYS_EXTWAKE indicates a wake-up to the external voltage regulator.

## SYSTEM DEBUG

The processor includes various features that allow for easy system debug. These are described in the following sections.

### *System Watchpoint Unit*

The system watchpoint unit (SWU) is a single module which connects to a single system bus and provides for transaction monitoring. One SWU is attached to the bus going to each system slave. The SWU provides ports for all system bus address channel signals. Each SWU contains four match groups of registers with associated hardware. These four SWU match groups operate independently, but share common event (interrupt, trigger, and others) outputs.

### *Debug Access Port*

The debug access port (DAP) provides IEEE-1149.1 JTAG interface support through its JTAG debug and serial wire debug port (SWJ-DP). SWJ-DP is a combined JTAG-DP and SW-DP that enables either serial wire debug (SWD) or a JTAG emulator to be connected to a target. SWD signals share the same pins as JTAG. The DAP provides an optional instrumentation trace for both the core and system. It provides a trace stream that conforms to MIPI System Trace Protocol version 2 (STPv2).

## DEVELOPMENT TOOLS

Analog Devices supports its processors with a complete line of software and hardware development tools, including integrated development environments (CrossCore® Embedded Studio), evaluation products, emulators, and a wide variety of software add-ins.

### *Integrated Development Environments (IDEs)*

CrossCore Embedded Studio is based on the Eclipse™ framework. Supporting most Analog Devices processor families, it is the IDE of choice for future processors, including multicore devices. CrossCore Embedded Studio seamlessly integrates available software add-ins to support real time operating systems, file systems, TCP/IP stacks, USB stacks, algorithmic software modules, and evaluation hardware board support packages. For more information, visit www.analog.com/cces.

### *EZ-KIT Lite Evaluation Board*

For processor evaluation, Analog Devices provides a wide range of EZ-KIT Lite® evaluation boards. Including the processor and key peripherals, the evaluation board also supports on-chip emulation capabilities and other evaluation and development features. Also available are various EZ-Extenders®, which are daughter cards delivering additional specialized functionality, including audio and video processing. For more information, visit www.analog.com and search on “ezkit” or “ezextender”.

### *EZ-KIT Lite Evaluation Kits*

For a cost-effective way to learn more about developing with Analog Devices processors, Analog Devices offer a range of EZ-KIT Lite evaluation kits. Each evaluation kit includes an EZ-KIT Lite evaluation board, directions for downloading an evaluation version of the available IDE, a USB cable, and a power supply. The USB controller on the EZ-KIT Lite board connects to the USB port of the user’s PC, enabling the chosen IDE evaluation suite to emulate the on-board processor in-circuit. This permits the customer to download, execute, and debug programs for the EZ-KIT Lite system. It also supports in-circuit programming of the on-board Flash device to store user-specific boot code, enabling standalone operation. With the full version of CrossCore Embedded Studio installed (sold separately), engineers can develop software for supported EZ-KITs or any custom system utilizing supported Analog Devices processors.

### *ADSP-BF706 EZ-KIT Mini*

The ADSP-BF706 EZ-KIT Mini™ product (ADZS-BF706-EZMini) contains the ADSP-BF706 processor and is shipped with all of the necessary hardware. Users can start their evaluation immediately. The EZ-KIT Mini product includes the standalone evaluation board and USB cable. The EZ-KIT Mini ships with an on-board debug agent.

The evaluation board is designed to be used in conjunction with the CrossCore Embedded Studio (CCES) development tools to test capabilities of the ADSP-BF706 Blackfin processor.

### *Blackfin Low Power Imaging Platform (BLIP)*

The Blackfin low power imaging platform (BLIP) integrates the ADSP-BF707 Blackfin processor and Analog Devices software code libraries. The code libraries are optimized to detect the presence and behavior of humans or vehicles in indoor and outdoor environments. The BLIP hardware platform is delivered preloaded with the occupancy software module.

### *Software Add-Ins for CrossCore Embedded Studio*

Analog Devices offers software add-ins which seamlessly integrate with CrossCore Embedded Studio to extend its capabilities and reduce development time. Add-ins include board support packages for evaluation hardware, various middleware packages, and algorithmic modules. Documentation, help, configuration dialogs, and coding examples present in these add-ins are viewable through the CrossCore Embedded Studio IDE once the add-in is installed.

#### Board Support Packages for Evaluation Hardware

Software support for the EZ-KIT Lite evaluation boards and EZ-Extender daughter cards is provided by software add-ins called board support packages (BSPs). The BSPs contain the required drivers, pertinent release notes, and select example code for the given evaluation hardware. A download link for a specific BSP is located on the web page for the associated EZ-KIT or EZ-Extender product. The link is found in the *Product Download* area of the product web page.

#### Middleware Packages

Analog Devices separately offers middleware add-ins such as real time operating systems, file systems, USB stacks, and TCP/IP stacks. For more information, see the following web pages:

- www.analog.com/ucos3
- www.analog.com/ucfs
- www.analog.com/ucusbd
- www.analog.com/lwip

### *Algorithmic Modules*

To speed development, Analog Devices offers add-ins that perform popular audio and video processing algorithms. These are available for use with CrossCore Embedded Studio. For more information, visit www.analog.com and search on "Blackfin software modules" or "SHARC software modules".

### *Designing an Emulator-Compatible DSP Board (Target)*

For embedded system test and debug, Analog Devices provides a family of emulators. On each DAP-enabled processor, Analog Devices supplies an IEEE 1149.1 JTAG test access port (TAP), serial wire debug port (SWJ-DP), and trace capabilities. In-circuit emulation is facilitated by use of the JTAG or SWD interface. The emulator accesses the processor's internal features through the processor's TAP, allowing the developer to load code, set breakpoints, and view variables, memory, and registers. The emulators require the target board to include a header(s) that supports connection of the processor's DAP to the emulator for trace and debug.

Analog Devices emulators actively drive $\overline{\text{JTG_TRST}}$ high. Third-party emulators may expect a pull-up on $\overline{\text{JTG_TRST}}$ and therefore will not drive $\overline{\text{JTG_TRST}}$ high. When using this type of third-party emulator $\overline{\text{JTG_TRST}}$ must still be driven low during power-up reset, but should subsequently be driven high externally before any emulation or boundary-scan operations. See Power-Up Reset Timing on Page 61 for more information on POR specifications.

For more details on target board design issues including mechanical layout, single processor connections, signal buffering, signal termination, and emulator pod logic, contact the factory for more information.

## ADDITIONAL INFORMATION

The following publications that describe the ADSP-BF70x processors can be accessed electronically on our website:

- *ADSP-BF70x Blackfin+ Processor Hardware Reference*
- *ADSP-BF70x Blackfin+ Processor Programming Reference*
- *ADSP-BF70x Blackfin+ Processor Anomaly List*

## RELATED SIGNAL CHAINS

A signal chain is a series of signal-conditioning electronic components that receive input (data acquired from sampling either real-time phenomena or from stored data) in tandem, with the output of one portion of the chain supplying input to the next. Signal chains are often used in signal processing applications to gather and process data or to apply system controls based on analysis of real-time phenomena.

Analog Devices eases signal processing system development by providing signal processing components that are designed to work together well. A tool for viewing relationships between specific applications and related components is available on the www.analog.com website.

The application signal chains page in the Circuits from the Lab® site (http:\\www.analog.com\circuits) provides:

- Graphical circuit block diagram presentation of signal chains for a variety of circuit types and applications
- Drill down links for components in each chain to selection guides and application information
- Reference designs applying best practice design techniques

## SECURITY FEATURES DISCLAIMER

To our knowledge, the Security Features, when used in accordance with the data sheet and hardware reference manual specifications, provide 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 SECURITY FEATURES 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.

## ADSP-BF70x DETAILED SIGNAL DESCRIPTIONS

Table 6 provides a detailed description of each pin.

**Table 6. ADSP-BF70x Detailed Signal Descriptions**

| Port Name | Direction | Description |
|---|---|---|
| CAN_RX | Input | **Receive.** Typically an external CAN transceiver's RX output. |
| CAN_TX | Output | **Transmit.** Typically an external CAN transceiver's TX input. |
| CNT_DG | Input | **Count Down and Gate.** Depending on the mode of operation this input acts either as a count down signal or a gate signal Count Down - This input causes the GP counter to decrement Gate - Stops the GP counter from incrementing or decrementing. |
| CNT_UD | Input | **Count Up and Direction.** Depending on the mode of operation this input acts either as a count up signal or a direction signal Count Up - This input causes the GP counter to increment Direction - Selects whether the GP counter is incrementing or decrementing. |
| CNT_ZM | Input | **Count Zero Marker.** Input that connects to the zero marker output of a rotary device or detects the pressing of a pushbutton. |
| DMC_Ann | Output | **Address n.** Address bus. |
| DMC_BAn | Output | **Bank Address Input n.** Defines which internal bank an ACTIVATE, READ, WRITE, or PRECHARGE command is being applied to on the dynamic memory. Also defines which mode registers (MR, EMR, EMR2, and/or EMR3) are loaded during the LOAD MODE REGISTER command. |
| $\overline{\text{DMC_CAS}}$ | Output | **Column Address Strobe.** Defines the operation for external dynamic memory to perform in conjunction with other DMC command signals. Connect to the CAS input of dynamic memory. |
| DMC_CK | Output | **Clock.** Outputs DCLK to external dynamic memory. |
| $\overline{\text{DMC_CK}}$ | Output | **Clock (Complement).** Complement of DMC_CK. |
| DMC_CKE | Output | **Clock enable.** Active high clock enables. Connects to the dynamic memory's CKE input. |
| $\overline{\text{DMC_CSn}}$ | Output | **Chip Select n.** Commands are recognized by the memory only when this signal is asserted. |
| DMC_DQnn | I/O | **Data n.** Bidirectional Data bus. |
| DMC_LDM | Output | **Data Mask for Lower Byte.** Mask for DMC_DQ07:DMC_DQ00 write data when driven high. Sampled on both edges of the data strobe by the dynamic memory. |
| DMC_LDQS | I/O | **Data Strobe for Lower Byte.** DMC_DQ07:DMC_DQ00 data strobe. Output with Write Data. Input with Read Data. May be single-ended or differential depending on register settings. |
| $\overline{\text{DMC_LDQS}}$ | I/O | **Data Strobe for Lower Byte (complement).** Complement of LDQS. Not used in single-ended mode. |
| DMC_ODT | Output | **On-die termination.** Enables dynamic memory termination resistances when driven high (assuming the memory is properly configured). ODT is enabled/disabled regardless of read or write commands. |
| $\overline{\text{DMC_RAS}}$ | Output | **Row Address Strobe.** Defines the operation for external dynamic memory to perform in conjunction with other DMC command signals. Connect to the RAS input of dynamic memory. |
| DMC_UDM | Output | **Data Mask for Upper Byte.** Mask for DMC_DQ15:DMC_DQ08 write data when driven high. Sampled on both edges of the data strobe by the dynamic memory. |
| DMC_UDQS | I/O | **Data Strobe for Upper Byte.** DMC_DQ15:DMC_DQ08 data strobe. Output with Write Data. Input with Read Data. May be single-ended or differential depending on register settings. |
| $\overline{\text{DMC_UDQS}}$ | I/O | **Data Strobe for Upper Byte (complement).** Complement of UDQSb. Not used in single-ended mode. |
| DMC_VREF | Input | **Voltage Reference.** Connect to half of the VDD_DMC voltage. |
| $\overline{\text{DMC_WE}}$ | Output | **Write Enable.** Defines the operation for external dynamic memory to perform in conjunction with other DMC command signals. Connect to the $\overline{\text{WE}}$ input of dynamic memory. |
| PPI_CLK | I/O | **Clock.** Input in external clock mode, output in internal clock mode. |
| PPI_Dnn | I/O | **Data n.** Bidirectional data bus. |
| PPI_FS1 | I/O | **Frame Sync 1 (HSYNC).** Behavior depends on EPPI mode. See the EPPI HRM chapter for more details. |
| PPI_FS2 | I/O | **Frame Sync 2 (VSYNC).** Behavior depends on EPPI mode. See the EPPI HRM chapter for more details. |
| PPI_FS3 | I/O | **Frame Sync 3 (FIELD).** Behavior depends on EPPI mode. See the EPPI HRM chapter for more details. |
| HADC_VINn | Input | **Analog Input at channel n.** Analog voltage inputs for digital conversion. |

**Table 6. ADSP-BF70x Detailed Signal Descriptions (Continued)**

| Port Name | Direction | Description |
|---|---|---|
| HADC_VREFN | Input | **Ground Reference for ADC.** Connect to an external voltage reference that meets data sheet specifications. |
| HADC_VREFP | Input | **External Reference for ADC.** Connect to an external voltage reference that meets data sheet specifications. |
| $\overline{\text{MSI_CD}}$ | Input | **Card Detect.** Connects to a pull-up resistor and to the card detect output of an SD socket. |
| MSI_CLK | Output | **Clock.** The clock signal applied to the connected device from the MSI. |
| MSI_CMD | I/O | **Command.** Used to send commands to and receive responses from the connected device. |
| MSI_Dn | I/O | **Data n.** Bidirectional data bus. |
| $\overline{\text{MSI_INT}}$ | Input | **eSDIO Interrupt Input.** Used only for eSDIO. Connects to an eSDIO card's interrupt output. An interrupt may be sampled even when the MSI clock to the card is switched off. |
| Px_nn | I/O | **Position n.** General purpose input/output. See the GP Ports chapter of the HRM for programming information. |
| RTC_CLKIN | Input | **Crystal input/external oscillator connection.** Connect to an external clock source or crystal. |
| RTC_XTAL | Output | **Crystal output.** Drives an external crystal. Must be left unconnected if an external clock is driving RTC_CLKIN. |
| $\overline{\text{SMC_ABEn}}$ | Output | **Byte Enable n.** Indicate whether the lower or upper byte of a memory is being accessed. When an asynchronous write is made to the upper byte of a 16-bit memory, SMC_ABE1b=0 and SMC_ABE0b=1. When an asynchronous write is made to the lower byte of a 16-bit memory, SMC_ABE1b=1 and SMC_ABE0b=0. |
| $\overline{\text{SMC_AMSn}}$ | Output | **Memory Select n.** Typically connects to the chip select of a memory device. |
| $\overline{\text{SMC_AOE}}$ | Output | **Output Enable.** Asserts at the beginning of the setup period of a read access. |
| SMC_ARDY | Input | **Asynchronous Ready.** Flow control signal used by memory devices to indicate to the SMC when further transactions may proceed. |
| $\overline{\text{SMC_ARE}}$ | Output | **Read Enable.** Asserts at the beginning of a read access. |
| $\overline{\text{SMC_AWE}}$ | Output | **Write Enable.** Asserts for the duration of a write access period. |
| SMC_Ann | Output | **Address n.** Address bus. |
| SMC_Dnn | I/O | **Data n.** Bidirectional data bus. |
| SPI_CLK | I/O | **Clock.** Input in slave mode, output in master mode. |
| SPI_D2 | I/O | **Data 2.** Used to transfer serial data in Quad mode. Open-drain when ODM mode is enabled. |
| SPI_D3 | I/O | **Data 3.** Used to transfer serial data in Quad mode. Open-drain when ODM mode is enabled. |
| SPI_MISO | I/O | **Master In, Slave Out.** Used to transfer serial data. Operates in the same direction as SPI_MOSI in Dual and Quad modes. Open-drain when ODM mode is enabled. |
| SPI_MOSI | I/O | **Master Out, Slave In.** Used to transfer serial data. Operates in the same direction as SPI_MISO in Dual and Quad modes. Open-drain when ODM mode is enabled. |
| SPI_RDY | I/O | **Ready.** Optional flow signal. Output in slave mode, input in master mode. |
| $\overline{\text{SPI_SELn}}$ | Output | **Slave Select Output n.** Used in Master mode to enable the desired slave. |
| $\overline{\text{SPI_SS}}$ | Input | **Slave Select Input.** Slave mode - Acts as the slave select input. Master mode- Optionally serves as an error detection input for the SPI when there are multiple masters. |
| SPT_ACLK | I/O | **Channel A Clock.** Data and Frame Sync are driven/sampled with respect to this clock. This signal can be either internally or externally generated. |
| SPT_AD0 | I/O | **Channel A Data 0.** Primary bidirectional data I/O. This signal can be configured as an output to transmit serial data, or as an input to receive serial data. |
| SPT_AD1 | I/O | **Channel A Data 1.** Secondary bidirectional data I/O. This signal can be configured as an output to transmit serial data, or as an input to receive serial data. |
| SPT_AFS | I/O | **Channel A Frame Sync.** The frame sync pulse initiates shifting of serial data. This signal is either generated internally or externally. |
| SPT_ATDV | Output | **Channel A Transmit Data Valid.** This signal is optional and only active when SPORT is configured in multichannel transmit mode. It is asserted during enabled slots. |

**Table 6. ADSP-BF70x Detailed Signal Descriptions (Continued)**

| Port Name | Direction | Description |
|---|---|---|
| SPT_BCLK | I/O | **Channel B Clock.** Data and Frame Sync are driven/sampled with respect to this clock. This signal can be either internally or externally generated. |
| SPT_BD0 | I/O | **Channel B Data 0.** Primary bidirectional data I/O. This signal can be configured as an output to transmit serial data, or as an input to receive serial data. |
| SPT_BD1 | I/O | **Channel B Data 1.** Secondary bidirectional data I/O. This signal can be configured as an output to transmit serial data, or as an input to receive serial data. |
| SPT_BFS | I/O | **Channel B Frame Sync.** The frame sync pulse initiates shifting of serial data. This signal is either generated internally or externally. |
| SPT_BTDV | Output | **Channel B Transmit Data Valid.** This signal is optional and only active when SPORT is configured in multi-channel transmit mode. It is asserted during enabled slots. |
| SYS_BMODEn | Input | **Boot Mode Control n.** Selects the boot mode of the processor. |
| SYS_CLKIN | Input | **Clock/Crystal Input.** Connect to an external clock source or crystal. |
| SYS_CLKOUT | Output | **Processor Clock Output.** Outputs internal clocks. Clocks may be divided down. See the CGU chapter of the HRM for more details. |
| SYS_EXTWAKE | Output | **External Wake Control.** Drives low during hibernate and high all other times. Typically connected to the enable input of the voltage regulator controlling the VDD_INT supply. |
| $\overline{\text{SYS_FAULT}}$ | I/O | **Active-Low Fault Output.** Indicates internal faults or senses external faults depending on the operating mode. |
| $\overline{\text{SYS_HWRST}}$ | Input | **Processor Hardware Reset Control.** Resets the device when asserted. |
| $\overline{\text{SYS_NMI}}$ | Input | **Non-maskable Interrupt.** See the processor hardware and programming references for more details. |
| $\overline{\text{SYS_RESOUT}}$ | Output | **Reset Output.** Indicates that the device is in the reset or hibernate state. |
| SYS_WAKEn | Input | **Power Saving Mode Wakeup n.** Wake-up source input for deep sleep and/or hibernate mode. |
| SYS_XTAL | Output | **Crystal Output.** Drives an external crystal. Must be left unconnected if an external clock is driving CLKIN. |
| JTG_SWCLK | I/O | **Serial Wire Clock.** Clocks data into and out of the target during debug. |
| JTG_SWDIO | I/O | **Serial Wire DIO.** Sends and receives serial data to and from the target during debug. |
| JTG_SWO | Output | **Serial Wire Out.** Provides trace data to the emulator. |
| JTG_TCK | Input | **JTAG Clock.** JTAG test access port clock. |
| JTG_TDI | Input | **JTAG Serial Data In.** JTAG test access port data input. |
| JTG_TDO | Output | **JTAG Serial Data Out.** JTAG test access port data output. |
| JTG_TMS | Input | **JTAG Mode Select.** JTAG test access port mode select. |
| $\overline{\text{JTG_TRST}}$ | Input | **JTAG Reset.** JTAG test access port reset. |
| TM_ACIn | Input | **Alternate Capture Input n.** Provides an additional input for WIDCAP, WATCHDOG, and PININT modes. |
| TM_ACLKn | Input | **Alternate Clock n.** Provides an additional time base for use by an individual timer. |
| TM_CLK | Input | **Clock.** Provides an additional global time base for use by all the GP timers. |
| TM_TMRn | I/O | **Timer n.** The main input/output signal for each timer. |
| TRACE_CLK | Output | **Trace Clock.** Clock output. |
| TRACE_Dnn | Output | **Trace Data n.** Unidirectional data bus. |
| TWI_SCL | I/O | **Serial Clock.** Clock output when master, clock input when slave. |
| TWI_SDA | I/O | **Serial Data.** Receives or transmits data. |
| $\overline{\text{UART_CTS}}$ | Input | **Clear to Send.** Flow control signal. |
| $\overline{\text{UART_RTS}}$ | Output | **Request to Send.** Flow control signal. |
| $\overline{\text{UART_RX}}$ | Input | **Receive.** Receive input. Typically connects to a transceiver that meets the electrical requirements of the device being communicated with. |
| $\overline{\text{UART_TX}}$ | Output | **Transmit.** Transmit output. Typically connects to a transceiver that meets the electrical requirements of the device being communicated with. |
| USB_CLKIN | Input | **Clock/Crystal Input.** This clock input is multiplied by a PLL to form the USB clock. See data sheet specifications for frequency/tolerance information. |

**Table 6. ADSP-BF70x Detailed Signal Descriptions (Continued)**

| Port Name | Direction | Description |
|---|---|---|
| USB_DM | I/O | **Data –.** Bidirectional differential data line. |
| USB_DP | I/O | **Data +.** Bidirectional differential data line. |
| USB_ID | Input | **OTG ID.** Senses whether the controller is a host or device. This signal is pulled low when an A-type plug is sensed (signifying that the USB controller is the A device), but the input is high when a B-type plug is sensed (signifying that the USB controller is the B device). |
| USB_VBC | Output | **VBUS Control.** Controls an external voltage source to supply VBUS when in host mode. May be configured as open-drain. Polarity is configurable as well. |
| USB_VBUS | I/O | **Bus Voltage.** Connects to bus voltage in host and device modes. |
| USB_XTAL | Output | **Crystal.** Drives an external crystal. Must be left unconnected if an external clock is driving USB_CLKIN. |

## 184-BALL CSP_BGA SIGNAL DESCRIPTIONS

The processor's pin definitions are shown in Table 7. The columns in this table provide the following information:

- Signal Name: The Signal Name column in the table includes the signal name for every pin and (where applicable) the GPIO multiplexed pin function for every pin.
- Description: The Description column in the table provides a verbose (descriptive) name for the signal.
- General-Purpose Port: The Port column in the table shows whether or not the signal is multiplexed with other signals on a general-purpose I/O port pin.
- Pin Name: The Pin Name column in the table identifies the name of the package pin (at power on reset) on which the signal is located (if a single function pin) or is multiplexed (if a general-purpose I/O pin).

**Table 7. ADSP-BF70x 184-Ball CSP_BGA Signal Descriptions**

| Signal Name | Description | Port | Pin Name |
|---|---|---|---|
| CAN0_RX | CAN0 Receive | C | PC_02 |
| CAN0_TX | CAN0 Transmit | C | PC_03 |
| CAN1_RX | CAN1 Receive | A | PA_12 |
| CAN1_TX | CAN1 Transmit | A | PA_13 |
| CNT0_DG | CNT0 Count Down and Gate | A | PA_07 |
| CNT0_UD | CNT0 Count Up and Direction | A | PA_15 |
| CNT0_ZM | CNT0 Count Zero Marker | A | PA_13 |
| DMC0_A00 | DMC0 Address 0 | Not Muxed | DMC0_A00 |
| DMC0_A01 | DMC0 Address 1 | Not Muxed | DMC0_A01 |
| DMC0_A02 | DMC0 Address 2 | Not Muxed | DMC0_A02 |
| DMC0_A03 | DMC0 Address 3 | Not Muxed | DMC0_A03 |
| DMC0_A04 | DMC0 Address 4 | Not Muxed | DMC0_A04 |
| DMC0_A05 | DMC0 Address 5 | Not Muxed | DMC0_A05 |
| DMC0_A06 | DMC0 Address 6 | Not Muxed | DMC0_A06 |
| DMC0_A07 | DMC0 Address 7 | Not Muxed | DMC0_A07 |
| DMC0_A08 | DMC0 Address 8 | Not Muxed | DMC0_A08 |
| DMC0_A09 | DMC0 Address 9 | Not Muxed | DMC0_A09 |
| DMC0_A10 | DMC0 Address 10 | Not Muxed | DMC0_A10 |
| DMC0_A11 | DMC0 Address 11 | Not Muxed | DMC0_A11 |
| DMC0_A12 | DMC0 Address 12 | Not Muxed | DMC0_A12 |
| DMC0_A13 | DMC0 Address 13 | Not Muxed | DMC0_A13 |
| DMC0_BA0 | DMC0 Bank Address Input 0 | Not Muxed | DMC0_BA0 |
| DMC0_BA1 | DMC0 Bank Address Input 1 | Not Muxed | DMC0_BA1 |
| DMC0_BA2 | DMC0 Bank Address Input 2 | Not Muxed | DMC0_BA2 |
| $\overline{\text{DMC0_CAS}}$ | DMC0 Column Address Strobe | Not Muxed | $\overline{\text{DMC0_CAS}}$ |
| DMC0_CK | DMC0 Clock | Not Muxed | DMC0_CK |
| DMC0_CKE | DMC0 Clock enable | Not Muxed | DMC0_CKE |
| $\overline{\text{DMC0_CK}}$ | DMC0 Clock (complement) | Not Muxed | $\overline{\text{DMC0_CK}}$ |
| $\overline{\text{DMC0_CS0}}$ | DMC0 Chip Select 0 | Not Muxed | $\overline{\text{DMC0_CS0}}$ |
| DMC0_DQ00 | DMC0 Data 0 | Not Muxed | DMC0_DQ00 |
| DMC0_DQ01 | DMC0 Data 1 | Not Muxed | DMC0_DQ01 |
| DMC0_DQ02 | DMC0 Data 2 | Not Muxed | DMC0_DQ02 |
| DMC0_DQ03 | DMC0 Data 3 | Not Muxed | DMC0_DQ03 |
| DMC0_DQ04 | DMC0 Data 4 | Not Muxed | DMC0_DQ04 |
| DMC0_DQ05 | DMC0 Data 5 | Not Muxed | DMC0_DQ05 |
| DMC0_DQ06 | DMC0 Data 6 | Not Muxed | DMC0_DQ06 |

**Table 7. ADSP-BF70x 184-Ball CSP_BGA Signal Descriptions (Continued)**

| Signal Name | Description | Port | Pin Name |
|---|---|---|---|
| DMC0_DQ07 | DMC0 Data 7 | Not Muxed | DMC0_DQ07 |
| DMC0_DQ08 | DMC0 Data 8 | Not Muxed | DMC0_DQ08 |
| DMC0_DQ09 | DMC0 Data 9 | Not Muxed | DMC0_DQ09 |
| DMC0_DQ10 | DMC0 Data 10 | Not Muxed | DMC0_DQ10 |
| DMC0_DQ11 | DMC0 Data 11 | Not Muxed | DMC0_DQ11 |
| DMC0_DQ12 | DMC0 Data 12 | Not Muxed | DMC0_DQ12 |
| DMC0_DQ13 | DMC0 Data 13 | Not Muxed | DMC0_DQ13 |
| DMC0_DQ14 | DMC0 Data 14 | Not Muxed | DMC0_DQ14 |
| DMC0_DQ15 | DMC0 Data 15 | Not Muxed | DMC0_DQ15 |
| DMC0_LDM | DMC0 Data Mask for Lower Byte | Not Muxed | DMC0_LDM |
| DMC0_LDQS | DMC0 Data Strobe for Lower Byte | Not Muxed | DMC0_LDQS |
| $\overline{\text{DMC0_LDQS}}$ | DMC0 Data Strobe for Lower Byte (complement) | Not Muxed | $\overline{\text{DMC0_LDQS}}$ |
| DMC0_ODT | DMC0 On-die termination | Not Muxed | DMC0_ODT |
| $\overline{\text{DMC0_RAS}}$ | DMC0 Row Address Strobe | Not Muxed | $\overline{\text{DMC0_RAS}}$ |
| DMC0_UDM | DMC0 Data Mask for Upper Byte | Not Muxed | DMC0_UDM |
| DMC0_UDQS | DMC0 Data Strobe for Upper Byte | Not Muxed | DMC0_UDQS |
| $\overline{\text{DMC0_UDQS}}$ | DMC0 Data Strobe for Upper Byte (complement) | Not Muxed | $\overline{\text{DMC0_UDQS}}$ |
| DMC0_VREF | DMC0 Voltage Reference | Not Muxed | DMC0_VREF |
| $\overline{\text{DMC0_WE}}$ | DMC0 Write Enable | Not Muxed | $\overline{\text{DMC0_WE}}$ |
| GND | Ground | Not Muxed | GND |
| GND_HADC | Ground HADC | Not Muxed | GND_HADC |
| HADC0_VIN0 | HADC0 Analog Input at channel 0 | Not Muxed | HADC0_VIN0 |
| HADC0_VIN1 | HADC0 Analog Input at channel 1 | Not Muxed | HADC0_VIN1 |
| HADC0_VIN2 | HADC0 Analog Input at channel 2 | Not Muxed | HADC0_VIN2 |
| HADC0_VIN3 | HADC0 Analog Input at channel 3 | Not Muxed | HADC0_VIN3 |
| HADC0_VREFN | HADC0 Ground Reference for ADC | Not Muxed | HADC0_VREFN |
| HADC0_VREFP | HADC0 External Reference for ADC | Not Muxed | HADC0_VREFP |
| JTG_SWCLK | TAPC0 Serial Wire Clock | Not Muxed | JTG_TCK_SWCLK |
| JTG_SWDIO | TAPC0 Serial Wire DIO | Not Muxed | JTG_TMS_SWDIO |
| JTG_SWO | TAPC0 Serial Wire Out | Not Muxed | JTG_TDO_SWO |
| JTG_TCK | TAPC0 JTAG Clock | Not Muxed | JTG_TCK_SWCLK |
| JTG_TDI | TAPC0 JTAG Serial Data In | Not Muxed | JTG_TDI |
| JTG_TDO | TAPC0 JTAG Serial Data Out | Not Muxed | JTG_TDO_SWO |
| JTG_TMS | TAPC0 JTAG Mode Select | Not Muxed | JTG_TMS_SWDIO |
| $\overline{\text{JTG_TRST}}$ | TAPC0 JTAG Reset | Not Muxed | $\overline{\text{JTG_TRST}}$ |
| $\overline{\text{MSI0_CD}}$ | MSI0 Card Detect | A | PA_08 |
| MSI0_CLK | MSI0 Clock | C | PC_09 |
| MSI0_CMD | MSI0 Command | C | PC_05 |
| MSI0_D0 | MSI0 Data 0 | C | PC_08 |
| MSI0_D1 | MSI0 Data 1 | C | PC_04 |
| MSI0_D2 | MSI0 Data 2 | C | PC_07 |
| MSI0_D3 | MSI0 Data 3 | C | PC_06 |
| MSI0_D4 | MSI0 Data 4 | C | PC_10 |
| MSI0_D5 | MSI0 Data 5 | C | PC_11 |
| MSI0_D6 | MSI0 Data 6 | C | PC_12 |
| MSI0_D7 | MSI0 Data 7 | C | PC_13 |

**Table 7. ADSP-BF70x 184-Ball CSP_BGA Signal Descriptions (Continued)**

| Signal Name | Description | Port | Pin Name |
|---|---|---|---|
| $\overline{\text{MSI0_INT}}$ | MSI0 eSDIO Interrupt Input | C | PC_14 |
| PA_00-PA_15 | Position 00 through Position 15 | A | PA_00-PA_15 |
| PB_00-PB_15 | Position 00 through Position 15 | B | PB_00-PB_15 |
| PC_00-PC_14 | Position 00 through Position 14 | C | PC_00-PC_14 |
| PPI0_CLK | EPPI0 Clock | A | PA_14 |
| PPI0_D00 | EPPI0 Data 0 | B | PB_07 |
| PPI0_D01 | EPPI0 Data 1 | B | PB_06 |
| PPI0_D02 | EPPI0 Data 2 | B | PB_05 |
| PPI0_D03 | EPPI0 Data 3 | B | PB_04 |
| PPI0_D04 | EPPI0 Data 4 | B | PB_03 |
| PPI0_D05 | EPPI0 Data 5 | B | PB_02 |
| PPI0_D06 | EPPI0 Data 6 | B | PB_01 |
| PPI0_D07 | EPPI0 Data 7 | B | PB_00 |
| PPI0_D08 | EPPI0 Data 8 | A | PA_11 |
| PPI0_D09 | EPPI0 Data 9 | A | PA_10 |
| PPI0_D10 | EPPI0 Data 10 | A | PA_09 |
| PPI0_D11 | EPPI0 Data 11 | A | PA_08 |
| PPI0_D12 | EPPI0 Data 12 | C | PC_03 |
| PPI0_D13 | EPPI0 Data 13 | C | PC_02 |
| PPI0_D14 | EPPI0 Data 14 | C | PC_01 |
| PPI0_D15 | EPPI0 Data 15 | C | PC_00 |
| PPI0_D16 | EPPI0 Data 16 | B | PB_08 |
| PPI0_D17 | EPPI0 Data 17 | B | PB_09 |
| PPI0_FS1 | EPPI0 Frame Sync 1 (HSYNC) | A | PA_12 |
| PPI0_FS2 | EPPI0 Frame Sync 2 (VSYNC) | A | PA_13 |
| PPI0_FS3 | EPPI0 Frame Sync 3 (FIELD) | A | PA_15 |
| RTC0_CLKIN | RTC0 Crystal input/external oscillator connection | Not Muxed | RTC0_CLKIN |
| RTC0_XTAL | RTC0 Crystal output | Not Muxed | RTC0_XTAL |
| SMC0_A01 | SMC0 Address 1 | A | PA_08 |
| SMC0_A02 | SMC0 Address 2 | A | PA_09 |
| SMC0_A03 | SMC0 Address 3 | A | PA_10 |
| SMC0_A04 | SMC0 Address 4 | A | PA_11 |
| SMC0_A05 | SMC0 Address 5 | A | PA_07 |
| SMC0_A06 | SMC0 Address 6 | A | PA_06 |
| SMC0_A07 | SMC0 Address 7 | A | PA_05 |
| SMC0_A08 | SMC0 Address 8 | A | PA_04 |
| SMC0_A09 | SMC0 Address 9 | C | PC_01 |
| SMC0_A10 | SMC0 Address 10 | C | PC_02 |
| SMC0_A11 | SMC0 Address 11 | C | PC_03 |
| SMC0_A12 | SMC0 Address 12 | C | PC_04 |
| $\overline{\text{SMC0_ABE0}}$ | SMC0 Byte Enable 0 | A | PA_00 |
| $\overline{\text{SMC0_ABE1}}$ | SMC0 Byte Enable 1 | A | PA_01 |
| $\overline{\text{SMC0_AMS0}}$ | SMC0 Memory Select 0 | A | PA_15 |
| $\overline{\text{SMC0_AMS1}}$ | SMC0 Memory Select 1 | A | PA_02 |
| $\overline{\text{SMC0_AOE}}$ | SMC0 Output Enable | A | PA_12 |
| SMC0_ARDY | SMC0 Asynchronous Ready | A | PA_03 |

**Table 7. ADSP-BF70x 184-Ball CSP_BGA Signal Descriptions (Continued)**

| Signal Name | Description | Port | Pin Name |
|---|---|---|---|
| SMC0_ARE | SMC0 Read Enable | A | PA_13 |
| SMC0_AWE | SMC0 Write Enable | A | PA_14 |
| SMC0_D00 | SMC0 Data 0 | B | PB_07 |
| SMC0_D01 | SMC0 Data 1 | B | PB_06 |
| SMC0_D02 | SMC0 Data 2 | B | PB_05 |
| SMC0_D03 | SMC0 Data 3 | B | PB_04 |
| SMC0_D04 | SMC0 Data 4 | B | PB_03 |
| SMC0_D05 | SMC0 Data 5 | B | PB_02 |
| SMC0_D06 | SMC0 Data 6 | B | PB_01 |
| SMC0_D07 | SMC0 Data 7 | B | PB_00 |
| SMC0_D08 | SMC0 Data 8 | B | PB_08 |
| SMC0_D09 | SMC0 Data 9 | B | PB_09 |
| SMC0_D10 | SMC0 Data 10 | B | PB_10 |
| SMC0_D11 | SMC0 Data 11 | B | PB_11 |
| SMC0_D12 | SMC0 Data 12 | B | PB_12 |
| SMC0_D13 | SMC0 Data 13 | B | PB_13 |
| SMC0_D14 | SMC0 Data 14 | B | PB_14 |
| SMC0_D15 | SMC0 Data 15 | B | PB_15 |
| SPI0_CLK | SPI0 Clock | B | PB_00 |
| SPI0_CLK | SPI0 Clock | C | PC_04 |
| SPI0_D2 | SPI0 Data 2 | B | PB_03 |
| SPI0_D2 | SPI0 Data 2 | C | PC_08 |
| SPI0_D3 | SPI0 Data 3 | B | PB_07 |
| SPI0_D3 | SPI0 Data 3 | C | PC_09 |
| SPI0_MISO | SPI0 Master In, Slave Out | B | PB_01 |
| SPI0_MISO | SPI0 Master In, Slave Out | C | PC_06 |
| SPI0_MOSI | SPI0 Master Out, Slave In | B | PB_02 |
| SPI0_MOSI | SPI0 Master Out, Slave In | C | PC_07 |
| SPI0_RDY | SPI0 Ready | A | PA_06 |
| SPI0_SEL1 | SPI0 Slave Select Output 1 | A | PA_05 |
| SPI0_SEL2 | SPI0 Slave Select Output 2 | A | PA_06 |
| SPI0_SEL3 | SPI0 Slave Select Output 3 | C | PC_11 |
| SPI0_SEL4 | SPI0 Slave Select Output 4 | B | PB_04 |
| SPI0_SEL5 | SPI0 Slave Select Output 5 | B | PB_05 |
| SPI0_SEL6 | SPI0 Slave Select Output 6 | B | PB_06 |
| SPI0_SS | SPI0 Slave Select Input | A | PA_05 |
| SPI1_CLK | SPI1 Clock | A | PA_00 |
| SPI1_MISO | SPI1 Master In, Slave Out | A | PA_01 |
| SPI1_MOSI | SPI1 Master Out, Slave In | A | PA_02 |
| SPI1_RDY | SPI1 Ready | A | PA_03 |
| SPI1_SEL1 | SPI1 Slave Select Output 1 | A | PA_04 |
| SPI1_SEL2 | SPI1 Slave Select Output 2 | A | PA_03 |
| SPI1_SEL3 | SPI1 Slave Select Output 3 | C | PC_10 |
| SPI1_SEL4 | SPI1 Slave Select Output 4 | A | PA_14 |
| SPI1_SS | SPI1 Slave Select Input | A | PA_04 |
| SPI2_CLK | SPI2 Clock | B | PB_10 |

**Table 7. ADSP-BF70x 184-Ball CSP_BGA Signal Descriptions (Continued)**

| Signal Name | Description | Port | Pin Name |
|---|---|---|---|
| SPI2_D2 | SPI2 Data 2 | B | PB_13 |
| SPI2_D3 | SPI2 Data 3 | B | PB_14 |
| SPI2_MISO | SPI2 Master In, Slave Out | B | PB_11 |
| SPI2_MOSI | SPI2 Master Out, Slave In | B | PB_12 |
| SPI2_RDY | SPI2 Ready | A | PA_04 |
| $\overline{\text{SPI2_SEL1}}$ | SPI2 Slave Select Output 1 | B | PB_15 |
| $\overline{\text{SPI2_SEL2}}$ | SPI2 Slave Select Output 2 | B | PB_08 |
| $\overline{\text{SPI2_SEL3}}$ | SPI2 Slave Select Output 3 | B | PB_09 |
| $\overline{\text{SPI2_SS}}$ | SPI2 Slave Select Input | B | PB_15 |
| SPT0_ACLK | SPORT0 Channel A Clock | A | PA_13 |
| SPT0_ACLK | SPORT0 Channel A Clock | C | PC_09 |
| SPT0_AD0 | SPORT0 Channel A Data 0 | A | PA_14 |
| SPT0_AD0 | SPORT0 Channel A Data 0 | C | PC_08 |
| SPT0_AD1 | SPORT0 Channel A Data 1 | C | PC_00 |
| SPT0_AFS | SPORT0 Channel A Frame Sync | A | PA_12 |
| SPT0_AFS | SPORT0 Channel A Frame Sync | C | PC_05 |
| SPT0_ATDV | SPORT0 Channel A Transmit Data Valid | A | PA_15 |
| SPT0_BCLK | SPORT0 Channel B Clock | B | PB_04 |
| SPT0_BCLK | SPORT0 Channel B Clock | C | PC_04 |
| SPT0_BD0 | SPORT0 Channel B Data 0 | B | PB_05 |
| SPT0_BD0 | SPORT0 Channel B Data 0 | C | PC_06 |
| SPT0_BD1 | SPORT0 Channel B Data 1 | B | PB_07 |
| SPT0_BD1 | SPORT0 Channel B Data 1 | C | PC_01 |
| SPT0_BFS | SPORT0 Channel B Frame Sync | B | PB_06 |
| SPT0_BFS | SPORT0 Channel B Frame Sync | C | PC_07 |
| SPT0_BTDV | SPORT0 Channel B Transmit Data Valid | A | PA_15 |
| SPT1_ACLK | SPORT1 Channel A Clock | A | PA_08 |
| SPT1_AD0 | SPORT1 Channel A Data 0 | A | PA_10 |
| SPT1_AD1 | SPORT1 Channel A Data 1 | A | PA_11 |
| SPT1_AFS | SPORT1 Channel A Frame Sync | A | PA_09 |
| SPT1_ATDV | SPORT1 Channel A Transmit Data Valid | A | PA_07 |
| SPT1_BCLK | SPORT1 Channel B Clock | B | PB_00 |
| SPT1_BCLK | SPORT1 Channel B Clock | C | PC_10 |
| SPT1_BD0 | SPORT1 Channel B Data 0 | B | PB_02 |
| SPT1_BD0 | SPORT1 Channel B Data 0 | C | PC_12 |
| SPT1_BD1 | SPORT1 Channel B Data 1 | B | PB_03 |
| SPT1_BD1 | SPORT1 Channel B Data 1 | C | PC_13 |
| SPT1_BFS | SPORT1 Channel B Frame Sync | B | PB_01 |
| SPT1_BFS | SPORT1 Channel B Frame Sync | C | PC_11 |
| SPT1_BTDV | SPORT1 Channel B Transmit Data Valid | A | PA_07 |
| SPT1_BTDV | SPORT1 Channel B Transmit Data Valid | C | PC_14 |
| SYS_BMODE0 | Boot Mode Control 0 | Not Muxed | SYS_BMODE0 |
| SYS_BMODE1 | Boot Mode Control 1 | Not Muxed | SYS_BMODE1 |
| SYS_CLKIN | Clock/Crystal Input | Not Muxed | SYS_CLKIN |
| SYS_CLKOUT | Processor Clock Output | Not Muxed | SYS_CLKOUT |
| SYS_EXTWAKE | External Wake Control | Not Muxed | SYS_EXTWAKE |

**Table 7. ADSP-BF70x 184-Ball CSP_BGA Signal Descriptions (Continued)**

| Signal Name | Description | Port | Pin Name |
|---|---|---|---|
| SYS_FAULT | Active-Low Fault Output | Not Muxed | SYS_FAULT |
| SYS_HWRST | Processor Hardware Reset Control | Not Muxed | SYS_HWRST |
| SYS_NMI | Nonmaskable Interrupt | Not Muxed | SYS_NMI |
| SYS_RESOUT | Reset Output | Not Muxed | SYS_RESOUT |
| SYS_WAKE0 | Power Saving Mode Wake-up 0 | B | PB_07 |
| SYS_WAKE1 | Power Saving Mode Wake-up 1 | B | PB_08 |
| SYS_WAKE2 | Power Saving Mode Wake-up 2 | B | PB_12 |
| SYS_WAKE3 | Power Saving Mode Wake-up 3 | C | PC_02 |
| SYS_WAKE4 | Power Saving Mode Wake-up 4 | A | PA_12 |
| SYS_XTAL | Crystal Output | Not Muxed | SYS_XTAL |
| TM0_ACI0 | TIMER0 Alternate Capture Input 0 | C | PC_03 |
| TM0_ACI1 | TIMER0 Alternate Capture Input 1 | B | PB_01 |
| TM0_ACI2 | TIMER0 Alternate Capture Input 2 | C | PC_07 |
| TM0_ACI3 | TIMER0 Alternate Capture Input 3 | B | PB_09 |
| TM0_ACI4 | TIMER0 Alternate Capture Input 4 | C | PC_01 |
| TM0_ACI5 | TIMER0 Alternate Capture Input 5 | C | PC_02 |
| TM0_ACI6 | TIMER0 Alternate Capture Input 6 | A | PA_12 |
| TM0_ACLK0 | TIMER0 Alternate Clock 0 | C | PC_04 |
| TM0_ACLK1 | TIMER0 Alternate Clock 1 | C | PC_10 |
| TM0_ACLK2 | TIMER0 Alternate Clock 2 | C | PC_09 |
| TM0_ACLK3 | TIMER0 Alternate Clock 3 | B | PB_00 |
| TM0_ACLK4 | TIMER0 Alternate Clock 4 | B | PB_10 |
| TM0_ACLK5 | TIMER0 Alternate Clock 5 | A | PA_14 |
| TM0_ACLK6 | TIMER0 Alternate Clock 6 | B | PB_04 |
| TM0_CLK | TIMER0 Clock | B | PB_06 |
| TM0_TMR0 | TIMER0 Timer 0 | A | PA_05 |
| TM0_TMR1 | TIMER0 Timer 1 | A | PA_06 |
| TM0_TMR2 | TIMER0 Timer 2 | A | PA_07 |
| TM0_TMR3 | TIMER0 Timer 3 | C | PC_05 |
| TM0_TMR4 | TIMER0 Timer 4 | A | PA_09 |
| TM0_TMR5 | TIMER0 Timer 5 | A | PA_10 |
| TM0_TMR6 | TIMER0 Timer 6 | A | PA_11 |
| TM0_TMR7 | TIMER0 Timer 7 | A | PA_04 |
| TRACE0_CLK | TPIU0 Trace Clock | B | PB_10 |
| TRACE0_D00 | TPIU0 Trace Data 0 | B | PB_15 |
| TRACE0_D01 | TPIU0 Trace Data 1 | B | PB_14 |
| TRACE0_D02 | TPIU0 Trace Data 2 | B | PB_13 |
| TRACE0_D03 | TPIU0 Trace Data 3 | B | PB_12 |
| TRACE0_D04 | TPIU0 Trace Data 4 | B | PB_11 |
| TRACE0_D05 | TPIU0 Trace Data 5 | A | PA_02 |
| TRACE0_D06 | TPIU0 Trace Data 6 | A | PA_01 |
| TRACE0_D07 | TPIU0 Trace Data 7 | A | PA_00 |
| TWI0_SCL | TWI0 Serial Clock | Not Muxed | TWI0_SCL |
| TWI0_SDA | TWI0 Serial Data | Not Muxed | TWI0_SDA |
| UART0_CTS | UART0 Clear to Send | C | PC_03 |
| UART0_RTS | UART0 Request to Send | C | PC_02 |

**Table 7. ADSP-BF70x 184-Ball CSP_BGA Signal Descriptions (Continued)**

| Signal Name | Description | Port | Pin Name |
|---|---|---|---|
| UART0_RX | UART0 Receive | B | PB_09 |
| UART0_TX | UART0 Transmit | B | PB_08 |
| UART1_CTS | UART1 Clear to Send | B | PB_14 |
| UART1_RTS | UART1 Request to Send | B | PB_13 |
| UART1_RX | UART1 Receive | C | PC_01 |
| UART1_TX | UART1 Transmit | C | PC_00 |
| USB0_CLKIN | USB0 Clock/Crystal Input | Not Muxed | USB0_CLKIN |
| USB0_DM | USB0 Data – | Not Muxed | USB0_DM |
| USB0_DP | USB0 Data + | Not Muxed | USB0_DP |
| USB0_ID | USB0 OTG ID | Not Muxed | USB0_ID |
| USB0_VBC | USB0 VBUS Control | Not Muxed | USB0_VBC |
| USB0_VBUS | USB0 Bus Voltage | Not Muxed | USB0_VBUS |
| USB0_XTAL | USB0 Crystal | Not Muxed | USB0_XTAL |
| VDD_DMC | VDD for DMC | Not Muxed | VDD_DMC |
| VDD_EXT | External VDD | Not Muxed | VDD_EXT |
| VDD_HADC | VDD for HADC | Not Muxed | VDD_HADC |
| VDD_INT | Internal VDD | Not Muxed | VDD_INT |
| VDD_OTP | VDD for OTP | Not Muxed | VDD_OTP |
| VDD_RTC | VDD for RTC | Not Muxed | VDD_RTC |
| VDD_USB | VDD for USB | Not Muxed | VDD_USB |

# GPIO MULTIPLEXING FOR 184-BALL CSP_BGA

Table 8 through Table 10 identify the pin functions that are multiplexed on the general-purpose I/O pins of the 184-ball CSP_BGA package.

**Table 8. Signal Multiplexing for Port A**

| Signal Name | Multiplexed Function 0 | Multiplexed Function 1 | Multiplexed Function 2 | Multiplexed Function 3 | Multiplexed Function Input Tap |
|---|---|---|---|---|---|
| PA_00 | SPI1_CLK | | TRACE0_D07 | SMC0_ABE0 | |
| PA_01 | SPI1_MISO | | TRACE0_D06 | SMC0_ABE1 | |
| PA_02 | SPI1_MOSI | | TRACE0_D05 | SMC0_AMS1 | |
| PA_03 | SPI1_SEL2 | SPI1_RDY | | SMC0_ARDY | |
| PA_04 | SPI1_SEL1 | TM0_TMR7 | SPI2_RDY | SMC0_A08 | SPI1_SS |
| PA_05 | TM0_TMR0 | SPI0_SEL1 | | SMC0_A07 | SPI0_SS |
| PA_06 | TM0_TMR1 | SPI0_SEL2 | SPI0_RDY | SMC0_A06 | |
| PA_07 | TM0_TMR2 | SPT1_BTDV | SPT1_ATDV | SMC0_A05 | CNT0_DG |
| PA_08 | PPI0_D11 | MSI0_CD | SPT1_ACLK | SMC0_A01 | |
| PA_09 | PPI0_D10 | TM0_TMR4 | SPT1_AFS | SMC0_A02 | |
| PA_10 | PPI0_D09 | TM0_TMR5 | SPT1_AD0 | SMC0_A03 | |
| PA_11 | PPI0_D08 | TM0_TMR6 | SPT1_AD1 | SMC0_A04 | |
| PA_12 | PPI0_FS1 | CAN1_RX | SPT0_AFS | SMC0_AOE | TM0_ACI6/SYS_WAKE4 |
| PA_13 | PPI0_FS2 | CAN1_TX | SPT0_ACLK | SMC0_ARE | CNT0_ZM |
| PA_14 | PPI0_CLK | SPI1_SEL4 | SPT0_AD0 | SMC0_AWE | TM0_ACLK5 |
| PA_15 | PPI0_FS3 | SPT0_ATDV | SPT0_BTDV | SMC0_AMS0 | CNT0_UD |

**Table 9. Signal Multiplexing for Port B**

| Signal Name | Multiplexed Function 0 | Multiplexed Function 1 | Multiplexed Function 2 | Multiplexed Function 3 | Multiplexed Function Input Tap |
|---|---|---|---|---|---|
| PB_00 | PPI0_D07 | SPT1_BCLK | SPI0_CLK | SMC0_D07 | TM0_ACLK3 |
| PB_01 | PPI0_D06 | SPT1_BFS | SPI0_MISO | SMC0_D06 | TM0_ACI1 |
| PB_02 | PPI0_D05 | SPT1_BD0 | SPI0_MOSI | SMC0_D05 | |
| PB_03 | PPI0_D04 | SPT1_BD1 | SPI0_D2 | SMC0_D04 | |
| PB_04 | PPI0_D03 | SPT0_BCLK | SPI0_SEL4 | SMC0_D03 | TM0_ACLK6 |
| PB_05 | PPI0_D02 | SPT0_BD0 | SPI0_SEL5 | SMC0_D02 | |
| PB_06 | PPI0_D01 | SPT0_BFS | SPI0_SEL6 | SMC0_D01 | TM0_CLK |
| PB_07 | PPI0_D00 | SPT0_BD1 | SPI0_D3 | SMC0_D00 | SYS_WAKE0 |
| PB_08 | UART0_TX | PPI0_D16 | SPI2_SEL2 | SMC0_D08 | SYS_WAKE1 |
| PB_09 | UART0_RX | PPI0_D17 | SPI2_SEL3 | SMC0_D09 | TM0_ACI3 |
| PB_10 | SPI2_CLK | | TRACE0_CLK | SMC0_D10 | TM0_ACLK4 |
| PB_11 | SPI2_MISO | | TRACE0_D04 | SMC0_D11 | |
| PB_12 | SPI2_MOSI | | TRACE0_D03 | SMC0_D12 | SYS_WAKE2 |
| PB_13 | SPI2_D2 | UART1_RTS | TRACE0_D02 | SMC0_D13 | |
| PB_14 | SPI2_D3 | UART1_CTS | TRACE0_D01 | SMC0_D14 | |
| PB_15 | SPI2_SEL1 | | TRACE0_D00 | SMC0_D15 | SPI2_SS |

**Table 10. Signal Multiplexing for Port C**

| Signal Name | Multiplexed Function 0 | Multiplexed Function 1 | Multiplexed Function 2 | Multiplexed Function 3 | Multiplexed Function Input Tap |
|---|---|---|---|---|---|
| PC_00 | UART1_TX | SPT0_AD1 | PPI0_D15 | | |
| PC_01 | UART1_RX | SPT0_BD1 | PPI0_D14 | SMC0_A09 | TM0_ACI4 |
| PC_02 | UART0_RTS | CAN0_RX | PPI0_D13 | SMC0_A10 | TM0_ACI5/SYS_ WAKE3 |
| PC_03 | UART0_CTS | CAN0_TX | PPI0_D12 | SMC0_A11 | TM0_ACI0 |
| PC_04 | SPT0_BCLK | SPI0_CLK | MSI0_D1 | SMC0_A12 | TM0_ACLK0 |
| PC_05 | SPT0_AFS | TM0_TMR3 | MSI0_CMD | | |
| PC_06 | SPT0_BD0 | SPI0_MISO | MSI0_D3 | | |
| PC_07 | SPT0_BFS | SPI0_MOSI | MSI0_D2 | | TM0_ACI2 |
| PC_08 | SPT0_AD0 | SPI0_D2 | MSI0_D0 | | |
| PC_09 | SPT0_ACLK | SPI0_D3 | MSI0_CLK | | TM0_ACLK2 |
| PC_10 | SPT1_BCLK | MSI0_D4 | SPI1_SEL3 | | TM0_ACLK1 |
| PC_11 | SPT1_BFS | MSI0_D5 | SPI0_SEL3 | | |
| PC_12 | SPT1_BD0 | MSI0_D6 | | | |
| PC_13 | SPT1_BD1 | MSI0_D7 | | | |
| PC_14 | SPT1_BTDV | MSI0_INT | | | |

# 12 mm × 12 mm 88-LEAD LFCSP (QFN) SIGNAL DESCRIPTIONS

The processor's pin definitions are shown in Table 11. The columns in this table provide the following information:

- Signal Name: The Signal Name column in the table includes the signal name for every pin and (where applicable) the GPIO multiplexed pin function for every pin.
- Description: The Description column in the table provides a verbose (descriptive) name for the signal.
- General-Purpose Port: The Port column in the table shows whether or not the signal is multiplexed with other signals on a general-purpose I/O port pin.
- Pin Name: The Pin Name column in the table identifies the name of the package pin (at power on reset) on which the signal is located (if a single function pin) or is multiplexed (if a general-purpose I/O pin).

**Table 11. ADSP-BF70x 12 mm × 12 mm 88-Lead LFCSP (QFN) Signal Descriptions**

| Signal Name | Description | Port | Pin Name |
|---|---|---|---|
| CAN0_RX | CAN0 Receive | C | PC_02 |
| CAN0_TX | CAN0 Transmit | C | PC_03 |
| CAN1_RX | CAN1 Receive | A | PA_12 |
| CAN1_TX | CAN1 Transmit | A | PA_13 |
| CNT0_DG | CNT0 Count Down and Gate | A | PA_07 |
| CNT0_UD | CNT0 Count Up and Direction | A | PA_15 |
| CNT0_ZM | CNT0 Count Zero Marker | A | PA_13 |
| GND | Ground | Not Muxed | GND |
| JTG_SWCLK | TAPC0 Serial Wire Clock | Not Muxed | JTG_TCK_SWCLK |
| JTG_SWDIO | TAPC0 Serial Wire DIO | Not Muxed | JTG_TMS_SWDIO |
| JTG_SWO | TAPC0 Serial Wire Out | Not Muxed | JTG_TDO_SWO |
| JTG_TCK | TAPC0 JTAG Clock | Not Muxed | JTG_TCK_SWCLK |
| JTG_TDI | TAPC0 JTAG Serial Data In | Not Muxed | JTG_TDI |
| JTG_TDO | TAPC0 JTAG Serial Data Out | Not Muxed | JTG_TDO_SWO |
| JTG_TMS | TAPC0 JTAG Mode Select | Not Muxed | JTG_TMS_SWDIO |
| $\overline{\text{JTG_TRST}}$ | TAPC0 JTAG Reset | Not Muxed | $\overline{\text{JTG_TRST}}$ |
| $\overline{\text{MSI0_CD}}$ | MSI0 Card Detect | A | PA_08 |
| MSI0_CLK | MSI0 Clock | C | PC_09 |
| MSI0_CMD | MSI0 Command | C | PC_05 |
| MSI0_D0 | MSI0 Data 0 | C | PC_08 |
| MSI0_D1 | MSI0 Data 1 | C | PC_04 |
| MSI0_D2 | MSI0 Data 2 | C | PC_07 |
| MSI0_D3 | MSI0 Data 3 | C | PC_06 |
| MSI0_D4 | MSI0 Data 4 | C | PC_10 |
| PA_00-PA_15 | Position 00 through Position 15 | A | PA_00-PA_15 |
| PB_00-PB_15 | Position 00 through Position 15 | B | PB_00-PB_15 |
| PC_00-PC_10 | Position 00 through Position 10 | C | PC_00-PC_10 |
| PPI0_CLK | EPPI0 Clock | A | PA_14 |
| PPI0_D00 | EPPI0 Data 0 | B | PB_07 |
| PPI0_D01 | EPPI0 Data 1 | B | PB_06 |
| PPI0_D02 | EPPI0 Data 2 | B | PB_05 |
| PPI0_D03 | EPPI0 Data 3 | B | PB_04 |
| PPI0_D04 | EPPI0 Data 4 | B | PB_03 |
| PPI0_D05 | EPPI0 Data 5 | B | PB_02 |
| PPI0_D06 | EPPI0 Data 6 | B | PB_01 |
| PPI0_D07 | EPPI0 Data 7 | B | PB_00 |

**Table 11. ADSP-BF70x 12 mm × 12 mm 88-Lead LFCSP (QFN) Signal Descriptions (Continued)**

| Signal Name | Description | Port | Pin Name |
|---|---|---|---|
| PPI0_D08 | EPPI0 Data 8 | A | PA_11 |
| PPI0_D09 | EPPI0 Data 9 | A | PA_10 |
| PPI0_D10 | EPPI0 Data 10 | A | PA_09 |
| PPI0_D11 | EPPI0 Data 11 | A | PA_08 |
| PPI0_D12 | EPPI0 Data 12 | C | PC_03 |
| PPI0_D13 | EPPI0 Data 13 | C | PC_02 |
| PPI0_D14 | EPPI0 Data 14 | C | PC_01 |
| PPI0_D15 | EPPI0 Data 15 | C | PC_00 |
| PPI0_D16 | EPPI0 Data 16 | B | PB_08 |
| PPI0_D17 | EPPI0 Data 17 | B | PB_09 |
| PPI0_FS1 | EPPI0 Frame Sync 1 (HSYNC) | A | PA_12 |
| PPI0_FS2 | EPPI0 Frame Sync 2 (VSYNC) | A | PA_13 |
| PPI0_FS3 | EPPI0 Frame Sync 3 (FIELD) | A | PA_15 |
| RTC0_CLKIN | RTC0 Crystal input/external oscillator connection | Not Muxed | RTC0_CLKIN |
| RTC0_XTAL | RTC0 Crystal output | Not Muxed | RTC0_XTAL |
| SMC0_A01 | SMC0 Address 1 | A | PA_08 |
| SMC0_A02 | SMC0 Address 2 | A | PA_09 |
| SMC0_A03 | SMC0 Address 3 | A | PA_10 |
| SMC0_A04 | SMC0 Address 4 | A | PA_11 |
| SMC0_A05 | SMC0 Address 5 | A | PA_07 |
| SMC0_A06 | SMC0 Address 6 | A | PA_06 |
| SMC0_A07 | SMC0 Address 7 | A | PA_05 |
| SMC0_A08 | SMC0 Address 8 | A | PA_04 |
| SMC0_A09 | SMC0 Address 9 | C | PC_01 |
| SMC0_A10 | SMC0 Address 10 | C | PC_02 |
| SMC0_A11 | SMC0 Address 11 | C | PC_03 |
| SMC0_A12 | SMC0 Address 12 | C | PC_04 |
| $\overline{\text{SMC0_ABE0}}$ | SMC0 Byte Enable 0 | A | PA_00 |
| $\overline{\text{SMC0_ABE1}}$ | SMC0 Byte Enable 1 | A | PA_01 |
| $\overline{\text{SMC0_AMS0}}$ | SMC0 Memory Select 0 | A | PA_15 |
| $\overline{\text{SMC0_AMS1}}$ | SMC0 Memory Select 1 | A | PA_02 |
| $\overline{\text{SMC0_AOE}}$ | SMC0 Output Enable | A | PA_12 |
| SMC0_ARDY | SMC0 Asynchronous Ready | A | PA_03 |
| $\overline{\text{SMC0_ARE}}$ | SMC0 Read Enable | A | PA_13 |
| $\overline{\text{SMC0_AWE}}$ | SMC0 Write Enable | A | PA_14 |
| SMC0_D00 | SMC0 Data 0 | B | PB_07 |
| SMC0_D01 | SMC0 Data 1 | B | PB_06 |
| SMC0_D02 | SMC0 Data 2 | B | PB_05 |
| SMC0_D03 | SMC0 Data 3 | B | PB_04 |
| SMC0_D04 | SMC0 Data 4 | B | PB_03 |
| SMC0_D05 | SMC0 Data 5 | B | PB_02 |
| SMC0_D06 | SMC0 Data 6 | B | PB_01 |
| SMC0_D07 | SMC0 Data 7 | B | PB_00 |
| SMC0_D08 | SMC0 Data 8 | B | PB_08 |
| SMC0_D09 | SMC0 Data 9 | B | PB_09 |
| SMC0_D10 | SMC0 Data 10 | B | PB_10 |

**Table 11. ADSP-BF70x 12 mm × 12 mm 88-Lead LFCSP (QFN) Signal Descriptions (Continued)**

| Signal Name | Description | Port | Pin Name |
|---|---|---|---|
| SMC0_D11 | SMC0 Data 11 | B | PB_11 |
| SMC0_D12 | SMC0 Data 12 | B | PB_12 |
| SMC0_D13 | SMC0 Data 13 | B | PB_13 |
| SMC0_D14 | SMC0 Data 14 | B | PB_14 |
| SMC0_D15 | SMC0 Data 15 | B | PB_15 |
| SPI0_CLK | SPI0 Clock | B | PB_00 |
| SPI0_CLK | SPI0 Clock | C | PC_04 |
| SPI0_D2 | SPI0 Data 2 | B | PB_03 |
| SPI0_D2 | SPI0 Data 2 | C | PC_08 |
| SPI0_D3 | SPI0 Data 3 | B | PB_07 |
| SPI0_D3 | SPI0 Data 3 | C | PC_09 |
| SPI0_MISO | SPI0 Master In, Slave Out | B | PB_01 |
| SPI0_MISO | SPI0 Master In, Slave Out | C | PC_06 |
| SPI0_MOSI | SPI0 Master Out, Slave In | B | PB_02 |
| SPI0_MOSI | SPI0 Master Out, Slave In | C | PC_07 |
| SPI0_RDY | SPI0 Ready | A | PA_06 |
| $\overline{\text{SPI0_SEL1}}$ | SPI0 Slave Select Output 1 | A | PA_05 |
| $\overline{\text{SPI0_SEL2}}$ | SPI0 Slave Select Output 2 | A | PA_06 |
| $\overline{\text{SPI0_SEL4}}$ | SPI0 Slave Select Output 4 | B | PB_04 |
| $\overline{\text{SPI0_SEL5}}$ | SPI0 Slave Select Output 5 | B | PB_05 |
| $\overline{\text{SPI0_SEL6}}$ | SPI0 Slave Select Output 6 | B | PB_06 |
| $\overline{\text{SPI0_SS}}$ | SPI0 Slave Select Input | A | PA_05 |
| SPI1_CLK | SPI1 Clock | A | PA_00 |
| SPI1_MISO | SPI1 Master In, Slave Out | A | PA_01 |
| SPI1_MOSI | SPI1 Master Out, Slave In | A | PA_02 |
| SPI1_RDY | SPI1 Ready | A | PA_03 |
| $\overline{\text{SPI1_SEL1}}$ | SPI1 Slave Select Output 1 | A | PA_04 |
| $\overline{\text{SPI1_SEL2}}$ | SPI1 Slave Select Output 2 | A | PA_03 |
| $\overline{\text{SPI1_SEL3}}$ | SPI1 Slave Select Output 3 | C | PC_10 |
| $\overline{\text{SPI1_SEL4}}$ | SPI1 Slave Select Output 4 | A | PA_14 |
| $\overline{\text{SPI1_SS}}$ | SPI1 Slave Select Input | A | PA_04 |
| SPI2_CLK | SPI2 Clock | B | PB_10 |
| SPI2_D2 | SPI2 Data 2 | B | PB_13 |
| SPI2_D3 | SPI2 Data 3 | B | PB_14 |
| SPI2_MISO | SPI2 Master In, Slave Out | B | PB_11 |
| SPI2_MOSI | SPI2 Master Out, Slave In | B | PB_12 |
| SPI2_RDY | SPI2 Ready | A | PA_04 |
| $\overline{\text{SPI2_SEL1}}$ | SPI2 Slave Select Output 1 | B | PB_15 |
| $\overline{\text{SPI2_SEL2}}$ | SPI2 Slave Select Output 2 | B | PB_08 |
| $\overline{\text{SPI2_SEL3}}$ | SPI2 Slave Select Output 3 | B | PB_09 |
| $\overline{\text{SPI2_SS}}$ | SPI2 Slave Select Input | B | PB_15 |
| SPT0_ACLK | SPORT0 Channel A Clock | A | PA_13 |
| SPT0_ACLK | SPORT0 Channel A Clock | C | PC_09 |
| SPT0_AD0 | SPORT0 Channel A Data 0 | A | PA_14 |
| SPT0_AD0 | SPORT0 Channel A Data 0 | C | PC_08 |
| SPT0_AD1 | SPORT0 Channel A Data 1 | C | PC_00 |

**Table 11. ADSP-BF70x 12 mm × 12 mm 88-Lead LFCSP (QFN) Signal Descriptions (Continued)**

| Signal Name | Description | Port | Pin Name |
|---|---|---|---|
| SPT0_AFS | SPORT0 Channel A Frame Sync | A | PA_12 |
| SPT0_AFS | SPORT0 Channel A Frame Sync | C | PC_05 |
| SPT0_ATDV | SPORT0 Channel A Transmit Data Valid | A | PA_15 |
| SPT0_BCLK | SPORT0 Channel B Clock | B | PB_04 |
| SPT0_BCLK | SPORT0 Channel B Clock | C | PC_04 |
| SPT0_BD0 | SPORT0 Channel B Data 0 | B | PB_05 |
| SPT0_BD0 | SPORT0 Channel B Data 0 | C | PC_06 |
| SPT0_BD1 | SPORT0 Channel B Data 1 | B | PB_07 |
| SPT0_BD1 | SPORT0 Channel B Data 1 | C | PC_01 |
| SPT0_BFS | SPORT0 Channel B Frame Sync | B | PB_06 |
| SPT0_BFS | SPORT0 Channel B Frame Sync | C | PC_07 |
| SPT0_BTDV | SPORT0 Channel B Transmit Data Valid | A | PA_15 |
| SPT1_ACLK | SPORT1 Channel A Clock | A | PA_08 |
| SPT1_AD0 | SPORT1 Channel A Data 0 | A | PA_10 |
| SPT1_AD1 | SPORT1 Channel A Data 1 | A | PA_11 |
| SPT1_AFS | SPORT1 Channel A Frame Sync | A | PA_09 |
| SPT1_ATDV | SPORT1 Channel A Transmit Data Valid | A | PA_07 |
| SPT1_BCLK | SPORT1 Channel B Clock | B | PB_00 |
| SPT1_BCLK | SPORT1 Channel B Clock | C | PC_10 |
| SPT1_BD0 | SPORT1 Channel B Data 0 | B | PB_02 |
| SPT1_BD1 | SPORT1 Channel B Data 1 | B | PB_03 |
| SPT1_BFS | SPORT1 Channel B Frame Sync | B | PB_01 |
| SPT1_BTDV | SPORT1 Channel B Transmit Data Valid | A | PA_07 |
| SYS_BMODE0 | Boot Mode Control 0 | Not Muxed | SYS_BMODE0 |
| SYS_BMODE1 | Boot Mode Control 1 | Not Muxed | SYS_BMODE1 |
| SYS_CLKIN | Clock/Crystal Input | Not Muxed | SYS_CLKIN |
| SYS_CLKOUT | Processor Clock Output | Not Muxed | SYS_CLKOUT |
| SYS_EXTWAKE | External Wake Control | Not Muxed | SYS_EXTWAKE |
| $\overline{\text{SYS_FAULT}}$ | Active-Low Fault Output | Not Muxed | $\overline{\text{SYS_FAULT}}$ |
| $\overline{\text{SYS_HWRST}}$ | Processor Hardware Reset Control | Not Muxed | $\overline{\text{SYS_HWRST}}$ |
| $\overline{\text{SYS_NMI}}$ | Non-maskable Interrupt | Not Muxed | $\overline{\text{SYS_NMI}}$ |
| $\overline{\text{SYS_RESOUT}}$ | Reset Output | Not Muxed | $\overline{\text{SYS_RESOUT}}$ |
| SYS_WAKE0 | Power Saving Mode Wake-up 0 | B | PB_07 |
| SYS_WAKE1 | Power Saving Mode Wake-up 1 | B | PB_08 |
| SYS_WAKE2 | Power Saving Mode Wake-up 2 | B | PB_12 |
| SYS_WAKE3 | Power Saving Mode Wake-up 3 | C | PC_02 |
| SYS_WAKE4 | Power Saving Mode Wake-up 4 | A | PA_12 |
| SYS_XTAL | Crystal Output | Not Muxed | SYS_XTAL |
| TM0_ACI0 | TIMER0 Alternate Capture Input 0 | C | PC_03 |
| TM0_ACI1 | TIMER0 Alternate Capture Input 1 | B | PB_01 |
| TM0_ACI2 | TIMER0 Alternate Capture Input 2 | C | PC_07 |
| TM0_ACI3 | TIMER0 Alternate Capture Input 3 | B | PB_09 |
| TM0_ACI4 | TIMER0 Alternate Capture Input 4 | C | PC_01 |
| TM0_ACI5 | TIMER0 Alternate Capture Input 5 | C | PC_02 |
| TM0_ACI6 | TIMER0 Alternate Capture Input 6 | A | PA_12 |
| TM0_ACLK0 | TIMER0 Alternate Clock 0 | C | PC_04 |

**Table 11. ADSP-BF70x 12 mm × 12 mm 88-Lead LFCSP (QFN) Signal Descriptions (Continued)**

| Signal Name | Description | Port | Pin Name |
|---|---|---|---|
| TM0_ACLK1 | TIMER0 Alternate Clock 1 | C | PC_10 |
| TM0_ACLK2 | TIMER0 Alternate Clock 2 | C | PC_09 |
| TM0_ACLK3 | TIMER0 Alternate Clock 3 | B | PB_00 |
| TM0_ACLK4 | TIMER0 Alternate Clock 4 | B | PB_10 |
| TM0_ACLK5 | TIMER0 Alternate Clock 5 | A | PA_14 |
| TM0_ACLK6 | TIMER0 Alternate Clock 6 | B | PB_04 |
| TM0_CLK | TIMER0 Clock | B | PB_06 |
| TM0_TMR0 | TIMER0 Timer 0 | A | PA_05 |
| TM0_TMR1 | TIMER0 Timer 1 | A | PA_06 |
| TM0_TMR2 | TIMER0 Timer 2 | A | PA_07 |
| TM0_TMR3 | TIMER0 Timer 3 | C | PC_05 |
| TM0_TMR4 | TIMER0 Timer 4 | A | PA_09 |
| TM0_TMR5 | TIMER0 Timer 5 | A | PA_10 |
| TM0_TMR6 | TIMER0 Timer 6 | A | PA_11 |
| TM0_TMR7 | TIMER0 Timer 7 | A | PA_04 |
| TRACE0_CLK | TPIU0 Trace Clock | B | PB_10 |
| TRACE0_D00 | TPIU0 Trace Data 0 | B | PB_15 |
| TRACE0_D01 | TPIU0 Trace Data 1 | B | PB_14 |
| TRACE0_D02 | TPIU0 Trace Data 2 | B | PB_13 |
| TRACE0_D03 | TPIU0 Trace Data 3 | B | PB_12 |
| TRACE0_D04 | TPIU0 Trace Data 4 | B | PB_11 |
| TRACE0_D05 | TPIU0 Trace Data 5 | A | PA_02 |
| TRACE0_D06 | TPIU0 Trace Data 6 | A | PA_01 |
| TRACE0_D07 | TPIU0 Trace Data 7 | A | PA_00 |
| TWI0_SCL | TWI0 Serial Clock | Not Muxed | TWI0_SCL |
| TWI0_SDA | TWI0 Serial Data | Not Muxed | TWI0_SDA |
| $\overline{\text{UART0_CTS}}$ | UART0 Clear to Send | C | PC_03 |
| $\overline{\text{UART0_RTS}}$ | UART0 Request to Send | C | PC_02 |
| $\overline{\text{UART0_RX}}$ | UART0 Receive | B | PB_09 |
| $\overline{\text{UART0_TX}}$ | UART0 Transmit | B | PB_08 |
| $\overline{\text{UART1_CTS}}$ | UART1 Clear to Send | B | PB_14 |
| $\overline{\text{UART1_RTS}}$ | UART1 Request to Send | B | PB_13 |
| $\overline{\text{UART1_RX}}$ | UART1 Receive | C | PC_01 |
| $\overline{\text{UART1_TX}}$ | UART1 Transmit | C | PC_00 |
| USB0_CLKIN | USB0 Clock/Crystal Input | Not Muxed | USB0_CLKIN |
| USB0_DM | USB0 Data – | Not Muxed | USB0_DM |
| USB0_DP | USB0 Data + | Not Muxed | USB0_DP |
| USB0_ID | USB0 OTG ID | Not Muxed | USB0_ID |
| USB0_VBC | USB0 VBUS Control | Not Muxed | USB0_VBC |
| USB0_VBUS | USB0 Bus Voltage | Not Muxed | USB0_VBUS |
| USB0_XTAL | USB0 Crystal | Not Muxed | USB0_XTAL |
| VDD_EXT | External VDD | Not Muxed | VDD_EXT |
| VDD_INT | Internal VDD | Not Muxed | VDD_INT |
| VDD_OTP | VDD for OTP | Not Muxed | VDD_OTP |
| VDD_RTC | VDD for RTC | Not Muxed | VDD_RTC |
| VDD_USB | VDD for USB | Not Muxed | VDD_USB |

## GPIO MULTIPLEXING FOR 12 mm × 12 mm 88-LEAD LFCSP (QFN)

Table 12 through Table 14 identify the pin functions that are multiplexed on the general-purpose I/O pins of the 12 mm × 12 mm 88-Lead LFCSP (QFN) package.

**Table 12. Signal Multiplexing for Port A**

| Signal Name | Multiplexed Function 0 | Multiplexed Function 1 | Multiplexed Function 2 | Multiplexed Function 3 | Multiplexed Function Input Tap |
|---|---|---|---|---|---|
| PA_00 | SPI1_CLK | | TRACE0_D07 | SMC0_ABE0 | |
| PA_01 | SPI1_MISO | | TRACE0_D06 | SMC0_ABE1 | |
| PA_02 | SPI1_MOSI | | TRACE0_D05 | SMC0_AMS1 | |
| PA_03 | SPI1_SEL2 | SPI1_RDY | | SMC0_ARDY | |
| PA_04 | SPI1_SEL1 | TM0_TMR7 | SPI2_RDY | SMC0_A08 | SPI1_SS |
| PA_05 | TM0_TMR0 | SPI0_SEL1 | | SMC0_A07 | SPI0_SS |
| PA_06 | TM0_TMR1 | SPI0_SEL2 | SPI0_RDY | SMC0_A06 | |
| PA_07 | TM0_TMR2 | SPT1_BTDV | SPT1_ATDV | SMC0_A05 | CNT0_DG |
| PA_08 | PPI0_D11 | MSI0_CD | SPT1_ACLK | SMC0_A01 | |
| PA_09 | PPI0_D10 | TM0_TMR4 | SPT1_AFS | SMC0_A02 | |
| PA_10 | PPI0_D09 | TM0_TMR5 | SPT1_AD0 | SMC0_A03 | |
| PA_11 | PPI0_D08 | TM0_TMR6 | SPT1_AD1 | SMC0_A04 | |
| PA_12 | PPI0_FS1 | CAN1_RX | SPT0_AFS | SMC0_AOE | TM0_ACI6/SYS_WAKE4 |
| PA_13 | PPI0_FS2 | CAN1_TX | SPT0_ACLK | SMC0_ARE | CNT0_ZM |
| PA_14 | PPI0_CLK | SPI1_SEL4 | SPT0_AD0 | SMC0_AWE | TM0_ACLK5 |
| PA_15 | PPI0_FS3 | SPT0_ATDV | SPT0_BTDV | SMC0_AMS0 | CNT0_UD |

**Table 13. Signal Multiplexing for Port B**

| Signal Name | Multiplexed Function 0 | Multiplexed Function 1 | Multiplexed Function 2 | Multiplexed Function 3 | Multiplexed Function Input Tap |
|---|---|---|---|---|---|
| PB_00 | PPI0_D07 | SPT1_BCLK | SPI0_CLK | SMC0_D07 | TM0_ACLK3 |
| PB_01 | PPI0_D06 | SPT1_BFS | SPI0_MISO | SMC0_D06 | TM0_ACI1 |
| PB_02 | PPI0_D05 | SPT1_BD0 | SPI0_MOSI | SMC0_D05 | |
| PB_03 | PPI0_D04 | SPT1_BD1 | SPI0_D2 | SMC0_D04 | |
| PB_04 | PPI0_D03 | SPT0_BCLK | SPI0_SEL4 | SMC0_D03 | TM0_ACLK6 |
| PB_05 | PPI0_D02 | SPT0_BD0 | SPI0_SEL5 | SMC0_D02 | |
| PB_06 | PPI0_D01 | SPT0_BFS | SPI0_SEL6 | SMC0_D01 | TM0_CLK |
| PB_07 | PPI0_D00 | SPT0_BD1 | SPI0_D3 | SMC0_D00 | SYS_WAKE0 |
| PB_08 | UART0_TX | PPI0_D16 | SPI2_SEL2 | SMC0_D08 | SYS_WAKE1 |
| PB_09 | UART0_RX | PPI0_D17 | SPI2_SEL3 | SMC0_D09 | TM0_ACI3 |
| PB_10 | SPI2_CLK | | TRACE0_CLK | SMC0_D10 | TM0_ACLK4 |
| PB_11 | SPI2_MISO | | TRACE0_D04 | SMC0_D11 | |
| PB_12 | SPI2_MOSI | | TRACE0_D03 | SMC0_D12 | SYS_WAKE2 |
| PB_13 | SPI2_D2 | UART1_RTS | TRACE0_D02 | SMC0_D13 | |
| PB_14 | SPI2_D3 | UART1_CTS | TRACE0_D01 | SMC0_D14 | |
| PB_15 | SPI2_SEL1 | | TRACE0_D00 | SMC0_D15 | SPI2_SS |

**Table 14. Signal Multiplexing for Port C**

| Signal Name | Multiplexed Function 0 | Multiplexed Function 1 | Multiplexed Function 2 | Multiplexed Function 3 | Multiplexed Function Input Tap |
|---|---|---|---|---|---|
| PC_00 | UART1_TX | SPT0_AD1 | PPI0_D15 | | |
| PC_01 | UART1_RX | SPT0_BD1 | PPI0_D14 | SMC0_A09 | TM0_ACI4 |
| PC_02 | UART0_RTS | CAN0_RX | PPI0_D13 | SMC0_A10 | TM0_ACI5/SYS_ WAKE3 |
| PC_03 | UART0_CTS | CAN0_TX | PPI0_D12 | SMC0_A11 | TM0_ACI0 |
| PC_04 | SPT0_BCLK | SPI0_CLK | MSI0_D1 | SMC0_A12 | TM0_ACLK0 |
| PC_05 | SPT0_AFS | TM0_TMR3 | MSI0_CMD | | |
| PC_06 | SPT0_BD0 | SPI0_MISO | MSI0_D3 | | |
| PC_07 | SPT0_BFS | SPI0_MOSI | MSI0_D2 | | TM0_ACI2 |
| PC_08 | SPT0_AD0 | SPI0_D2 | MSI0_D0 | | |
| PC_09 | SPT0_ACLK | SPI0_D3 | MSI0_CLK | | TM0_ACLK2 |
| PC_10 | SPT1_BCLK | MSI0_D4 | SPI1_SEL3 | | TM0_ACLK1 |

# ADSP-BF70x DESIGNER QUICK REFERENCE

Table 15 provides a quick reference summary of pin related information for circuit board design. The columns in this table provide the following information:

- Signal Name: The Signal Name column in the table includes the signal name for every pin and (where applicable) the GPIO multiplexed pin function for every pin.
- Pin Type: The Type column in the table identifies the I/O type or supply type of the pin. The abbreviations used in this column are na (none), I/O (input/output), a (analog), s (supply), and g (ground).
- Driver Type: The Driver Type column in the table identifies the driver type used by the pin. The driver types are defined in the output drive currents section of this data sheet.
- Internal Termination: The Int Term column in the table specifies the termination present when the processor is not in the reset or hibernate state. The abbreviations used in this column are wk (weak keeper, weakly retains previous value driven on the pin), pu (pull-up), or pd (pull-down).
- Reset Termination: The Reset Term column in the table specifies the termination present when the processor is in the reset state. The abbreviations used in this column are wk (weak keeper, weakly retains previous value driven on the pin), pu (pull-up), or pd (pull-down).
- Reset Drive: The Reset Drive column in the table specifies the active drive on the signal when the processor is in the reset state.
- Hibernate Termination: The Hiber Term column in the table specifies the termination present when the processor is in the hibernate state. The abbreviations used in this column are wk (weak keeper, weakly retains previous value driven on the pin), pu (pull-up), or pd (pull-down).
- Hibernate Drive: The Hiber Drive column in the table specifies the active drive on the signal when the processor is in the hibernate state.
- Power Domain: The Power Domain column in the table specifies the power supply domain in which the signal resides.
- Description and Notes: The Description and Notes column in the table identifies any special requirements or characteristics for the signal. If no special requirements are listed the signal may be left unconnected if it is not used. Also, for multiplexed general-purpose I/O pins, this column identifies the functions available on the pin.

If an external pull-up or pull-down resistor is required for any signal, 100 kΩ is the maximum value that can be used unless otherwise noted.

Note that for Port A, Port B, and Port C (PA_00 to PC_14), when $\overline{\text{SYS_HWRST}}$ is low, these pads are three-state. After $\overline{\text{SYS_HWRST}}$ is released, but before code execution begins, these pins are internally pulled up. Subsequently, the state depends on the input enable and output enable which are controlled by software.

Software control of internal pull-ups works according to the following settings in the PADS_PCFG0 register. When PADS_PCFG0 = 0: For PA_15:PA_00, PB_15:PB_00, and PC_14:PC_00, the internal pull-up is enabled when both the input enable and output enable of a particular pin are deasserted. When PADS_PCFG0 = 1: For PA_15:PA_00, PB_15:PB_00, and PC_14:PC_00, the internal pull-up is enabled as long as the output enable of a particular pin is deasserted.

There are some exceptions to this scheme:

- Internal pull-ups are always disabled if MSI mode is selected for that signal.
- The following signals enabled the internal pull-down when the output enable is de-asserted: $\overline{\text{SMC0_AMS}}[1:0]$, $\overline{\text{SMC0_ARE}}$, $\overline{\text{SMC0_AWE}}$, $\overline{\text{SMC0_AOE}}$, SMC0_ARDY, $\overline{\text{SPI0_SEL}}[6:1]$, $\overline{\text{SPI1_SEL}}[4:1]$, and $\overline{\text{SPI2_SEL}}[3:1]$.

**Table 15. ADSP-BF70x Designer Quick Reference**

| Signal Name | Type | Driver Type | Int Term | Reset Term | Reset Drive | Hiber Term | Hiber Drive | Power Domain | Description and Notes |
|---|---|---|---|---|---|---|---|---|---|
| DMC0_A00 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Address 0<br>Notes: No notes. |
| DMC0_A01 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Address 1<br>Notes: No notes. |
| DMC0_A02 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Address 2<br>Notes: No notes. |
| DMC0_A03 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Address 3<br>Notes: No notes. |
| DMC0_A04 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Address 4<br>Notes: No notes. |
| DMC0_A05 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Address 5<br>Notes: No notes. |

**Table 15. ADSP-BF70x Designer Quick Reference (Continued)**

| Signal Name | Type | Driver Type | Int Term | Reset Term | Reset Drive | Hiber Term | Hiber Drive | Power Domain | Description and Notes |
|---|---|---|---|---|---|---|---|---|---|
| DMC0_A06 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Address 6<br>Notes: No notes. |
| DMC0_A07 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Address 7<br>Notes: No notes. |
| DMC0_A08 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Address 8<br>Notes: No notes. |
| DMC0_A09 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Address 9<br>Notes: No notes. |
| DMC0_A10 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Address 10<br>Notes: No notes. |
| DMC0_A11 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Address 11<br>Notes: No notes. |
| DMC0_A12 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Address 12<br>Notes: No notes. |
| DMC0_A13 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Address 13<br>Notes: No notes. |
| DMC0_BA0 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Bank Address Input 0<br>Notes: No notes. |
| DMC0_BA1 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Bank Address Input 1<br>Notes: No notes. |
| DMC0_BA2 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Bank Address Input 2<br>Notes: For LPDDR, leave unconnected. |
| $\overline{\text{DMC0_CAS}}$ | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Column Address Strobe<br>Notes: No notes. |
| DMC0_CK | I/O | C | none | none | L | none | L | VDD_DMC | Desc: DMC0 Clock<br>Notes: No notes. |
| $\overline{\text{DMC0_CK}}$ | I/O | C | none | none | L | none | L | VDD_DMC | Desc: DMC0 Clock (complement)<br>Notes: No notes. |
| DMC0_CKE | I/O | B | none | none | L | none | L | VDD_DMC | Desc: DMC0 Clock enable<br>Notes: No notes. |
| $\overline{\text{DMC0_CS0}}$ | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Chip Select 0<br>Notes: No notes. |
| DMC0_DQ00 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Data 0<br>Notes: No notes. |
| DMC0_DQ01 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Data 1<br>Notes: No notes. |
| DMC0_DQ02 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Data 2<br>Notes: No notes. |
| DMC0_DQ03 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Data 3<br>Notes: No notes. |
| DMC0_DQ04 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Data 4<br>Notes: No notes. |
| DMC0_DQ05 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Data 5<br>Notes: No notes. |
| DMC0_DQ06 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Data 6<br>Notes: No notes. |
| DMC0_DQ07 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Data 7<br>Notes: No notes. |

**Table 15. ADSP-BF70x Designer Quick Reference (Continued)**

| Signal Name | Type | Driver Type | Int Term | Reset Term | Reset Drive | Hiber Term | Hiber Drive | Power Domain | Description and Notes |
|---|---|---|---|---|---|---|---|---|---|
| DMC0_DQ08 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Data 8<br>Notes: No notes. |
| DMC0_DQ09 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Data 9<br>Notes: No notes. |
| DMC0_DQ10 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Data 10<br>Notes: No notes. |
| DMC0_DQ11 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Data 11<br>Notes: No notes. |
| DMC0_DQ12 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Data 12<br>Notes: No notes. |
| DMC0_DQ13 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Data 13<br>Notes: No notes. |
| DMC0_DQ14 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Data 14<br>Notes: No notes. |
| DMC0_DQ15 | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Data 15<br>Notes: No notes. |
| DMC0_LDM | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Data Mask for Lower Byte<br>Notes: No notes. |
| DMC0_LDQS | I/O | C | none | none | none | none | none | VDD_DMC | Desc: DMC0 Data Strobe for Lower Byte<br>Notes: For LPDDR, a pull-down is required. |
| $\overline{\text{DMC0_LDQS}}$ | I/O | C | none | none | none | none | none | VDD_DMC | Desc: DMC0 Data Strobe for Lower Byte (complement)<br>Notes: For single ended DDR2, connect to DMC0_VREF. For LPDDR, leave unconnected. |
| DMC0_ODT | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 On-die termination<br>Notes: For LPDDR, leave unconnected. |
| $\overline{\text{DMC0_RAS}}$ | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Row Address Strobe<br>Notes: No notes. |
| DMC0_UDM | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Data Mask for Upper Byte<br>Notes: No notes. |
| DMC0_UDQS | I/O | C | none | none | none | none | none | VDD_DMC | Desc: DMC0 Data Strobe for Upper Byte<br>Notes: For LPDDR, a pull-down is required. |
| $\overline{\text{DMC0_UDQS}}$ | I/O | C | none | none | none | none | none | VDD_DMC | Desc: DMC0 Data Strobe for Upper Byte (complement)<br>Notes: For single ended DDR2, connect to DMC0_VREF. For LPDDR, leave unconnected. |
| DMC0_VREF | a | na | none | none | none | none | none | VDD_DMC | Desc: DMC0 Voltage Reference<br>Notes: For LPDDR, leave unconnected. If the DMC is not used, connect to ground. |
| $\overline{\text{DMC0_WE}}$ | I/O | B | none | none | none | none | none | VDD_DMC | Desc: DMC0 Write Enable<br>Notes: No notes. |
| GND | g | na | none | none | none | none | none | na | Desc: Ground<br>Notes: No notes. |

**Table 15. ADSP-BF70x Designer Quick Reference (Continued)**

| Signal Name | Type | Driver Type | Int Term | Reset Term | Reset Drive | Hiber Term | Hiber Drive | Power Domain | Description and Notes |
|---|---|---|---|---|---|---|---|---|---|
| GND_HADC | g | na | none | none | none | none | none | na | Desc: Ground HADC<br>Notes: If HADC is not used, connect to ground. |
| HADC0_VIN0 | a | na | none | none | none | none | none | VDD_HADC | Desc: HADC0 Analog Input at channel 0<br>Notes: If HADC is not used, connect to ground. |
| HADC0_VIN1 | a | na | none | none | none | none | none | VDD_HADC | Desc: HADC0 Analog Input at channel 1<br>Notes: If HADC is not used, connect to ground. |
| HADC0_VIN2 | a | na | none | none | none | none | none | VDD_HADC | Desc: HADC0 Analog Input at channel 2<br>Notes: If HADC is not used, connect to ground. |
| HADC0_VIN3 | a | na | none | none | none | none | none | VDD_HADC | Desc: HADC0 Analog Input at channel 3<br>Notes: If HADC is not used, connect to ground. |
| HADC0_VREFN | a | na | none | none | none | none | none | VDD_HADC | Desc: HADC0 Ground Reference for ADC<br>Notes: If HADC is not used, connect to ground. |
| HADC0_VREFP | a | na | none | none | none | none | none | VDD_HADC | Desc: HADC0 External Reference for ADC<br>Notes: If HADC is not used, connect to ground. |
| JTG_TCK_ SWCLK | I/O | na | pd | none | none | none | none | VDD_EXT | Desc: JTAG Clock \| Serial Wire Clock<br>Notes: Functional during reset. |
| JTG_TDI | I/O | na | pu | none | none | none | none | VDD_EXT | Desc: JTAG Serial Data In<br>Notes: Functional during reset. |
| JTG_TDO_SWO | I/O | A | none | none | none | none | none | VDD_EXT | Desc: JTAG Serial Data Out \| Serial Wire Out<br>Notes: Functional during reset, three-state when $\overline{\text{JTG_TRST}}$ is asserted. |
| JTG_TMS_ SWDIO | I/O | A | pu | none | none | none | none | VDD_EXT | Desc: JTAG Mode Select \| Serial Wire DIO<br>Notes: Functional during reset. |
| $\overline{\text{JTG_TRST}}$ | I/O | na | pd | none | none | none | none | VDD_EXT | Desc: JTAG Reset<br>Notes: Functional during reset, a 10k external pull-down may be used to shorten the $t_{VDDEXT_RST}$ timing requirement. |
| PA_00 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SPI1 Clock \| TRACE0 Trace Data 7 \| SMC0 Byte Enable 0<br>Notes: SPI clock requires a pull-down when controlling most SPI flash devices. |
| PA_01 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SPI1 Master In, Slave Out \| TRACE0 Trace Data 6 \| SMC0 Byte Enable 1<br>Notes: Pull-up required for SPI_MISO if SPI master boot is used. |

**Table 15. ADSP-BF70x Designer Quick Reference (Continued)**

| Signal Name | Type | Driver Type | Int Term | Reset Term | Reset Drive | Hiber Term | Hiber Drive | Power Domain | Description and Notes |
|---|---|---|---|---|---|---|---|---|---|
| PA_02 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SPI1 Master Out, Slave In \| TRACE0 Trace Data 5 \| SMC0 Memory Select 1<br>Notes: May require a pull-up if used as an SMC memory select. Check the data sheet requirements of the IC it connects to. |
| PA_03 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SPI1 Slave Select Output 2 \| SPI1 Ready \| SMC0 Asynchronous Ready<br>Notes: May require a pull-up or pull-down if used as an SMC asynchronous ready. Check the data sheet requirements of the IC it connects to and the programmed polarity. |
| PA_04 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SPI1 Slave Select Output 1 \| TM0 Timer 7 \| SPI2 Ready \| SMC0 Address 8 \| SPI1 Slave Select Input<br>Notes: SPI slave select outputs require a pull-up when used. |
| PA_05 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: TM0 Timer 0 \| SPI0 Slave Select Output 1 \| SMC0 Address 7 \| SPI0 Slave Select Input<br>Notes: SPI slave select outputs require a pull-up when used. |
| PA_06 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: TM0 Timer 1 \| SPI0 Slave Select Output 2 \| SPI0 Ready \| SMC0 Address 6<br>Notes: SPI slave select outputs require a pull-up when used. |
| PA_07 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: TM0 Timer 2 \| SPT1 Channel B Transmit Data Valid \| SPT1 Channel A Transmit Data Valid \| SMC0 Address 5 \| CNT0 Count Down and Gate<br>Notes: No notes. |
| PA_08 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: PPI0 Data 11 \| MSI0 Card Detect \| SPT1 Channel A Clock \| SMC0 Address 1<br>Notes: An external pull-up may be required for MSI modes, see the MSI chapter in the hardware reference for details. |
| PA_09 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: PPI0 Data 10 \| TM0 Timer 4 \| SPT1 Channel A Frame Sync \| SMC0 Address 2<br>Notes: No notes. |
| PA_10 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: PPI0 Data 9 \| TM0 Timer 5 \| SPT1 Channel A Data 0 \| SMC0 Address 3<br>Notes: No notes. |
| PA_11 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: PPI0 Data 8 \| TM0 Timer 6 \| SPT1 Channel A Data 1 \| SMC0 Address 4<br>Notes: No notes. |

**Table 15. ADSP-BF70x Designer Quick Reference (Continued)**

| Signal Name | Type | Driver Type | Int Term | Reset Term | Reset Drive | Hiber Term | Hiber Drive | Power Domain | Description and Notes |
|---|---|---|---|---|---|---|---|---|---|
| PA_12 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: PPI0 Frame Sync 1 (HSYNC) \| CAN1 Receive \| SPORT0 Channel A Frame Sync \|SMC0 Output Enable \|SYS Power Saving Mode Wakeup 4 \| TM0 Alternate Capture Input 6<br>Notes: If hibernate mode is used one of the following must be true during hibernate. Either this pin must be actively driven by another IC, or it must have a pull-up or pull-down. |
| PA_13 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: PPI0 Frame Sync 2 (VSYNC) \| CAN1 Transmit \| SPORT0 Channel A Clock \| SMC0 Read Enable \| CNT0 Count Zero Marker<br>Notes: No notes. |
| PA_14 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: PPI0 Clock \| SPI1 Slave Select Output 4 \| SPORT0 Channel A Data 0 \| SMC0 Write Enable \| TM0 Alternate Clock 5<br>Notes: SPI slave select outputs require a pull-up when used. |
| PA_15 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: PPI0 Frame Sync 3 (FIELD) \| SPT0 Channel A Transmit Data Valid \| SPT0 Channel B Transmit Data Valid \| SMC0 Memory Select 0 \| CNT0 Count Up and Direction<br>Notes: May require a pull-up if used as an SMC memory select. Check the data sheet requirements of the IC it connects to. |
| PB_00 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: PPI0 Data 7 \| SPT1 Channel B Clock \| SPI0 Clock \| SMC0 Data 7 \| TM0 Alternate Clock 3<br>Notes: SPI clock requires a pull-down when controlling most SPI flash devices. |
| PB_01 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: PPI0 Data 6 \| SPT1 Channel B Frame Sync \| SPI0 Master In, Slave Out \| SMC0 Data 6 \| TM0 Alternate Capture Input 1<br>Notes: Pull-up required for SPI_MISO if SPI master boot is used. |
| PB_02 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: PPI0 Data 5 \| SPT1 Channel B Data 0 \| SPI0 Master Out, Slave In \| SMC0 Data 5<br>Notes: No notes. |
| PB_03 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: PPI0 Data 4 \| SPT1 Channel B Data 1 \| SPI0 Data 2 \| SMC0 Data 4<br>Notes: No notes. |

**Table 15. ADSP-BF70x Designer Quick Reference (Continued)**

| Signal Name | Type | Driver Type | Int Term | Reset Term | Reset Drive | Hiber Term | Hiber Drive | Power Domain | Description and Notes |
|---|---|---|---|---|---|---|---|---|---|
| PB_04 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: PPI0 Data 3 \| SPT0 Channel B Clock \| SPI0 Slave Select Output 4 \| SMC0 Data 3 \| TM0 Alternate Clock 6<br>Notes: SPI slave select outputs require a pull-up when used. |
| PB_05 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: PPI0 Data 2 \| SPT0 Channel B Data 0 \| SPI0 Slave Select Output 5 \| SMC0 Data 2<br>Notes: SPI slave select outputs require a pull-up when used. |
| PB_06 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: PPI0 Data 1 \| SPT0 Channel B Frame Sync \| SPI0 Slave Select Output 6 \| SMC0 Data 1 \| TM0 Clock<br>Notes: SPI slave select outputs require a pull-up when used. |
| PB_07 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: PPI0 Data 0 \| SPT0 Channel B Data 1 \| SPI0 Data 3 \| SMC0 Data 0 \| SYS Power Saving Mode Wakeup 0<br>Notes: If hibernate mode is used, one of the following must be true during hibernate. Either this pin must be actively driven by another IC, or it must have a pull-up or pull-down. |
| PB_08 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: UART0 Transmit \| PPI0 Data 16 \| SPI2 Slave Select Output 2 \| SMC0 Data 8 \| SYS Power Saving Mode Wakeup 1<br>Notes: SPI slave select outputs require a pull-up when used. If hibernate mode is used, one of the following must be true during hibernate. Either this pin must be actively driven by another IC, or it must have a pull-up or pull-down. |
| PB_09 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: UART0 Receive \| PPI0 Data 17 \| SPI2 Slave Select Output 3 \| SMC0 Data 9 \| TM0 Alternate Capture Input 3<br>Notes: SPI slave select outputs require a pull-up when used. |
| PB_10 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SPI2 Clock \| TRACE0 Trace Clock \| SMC0 Data 10 \| TM0 Alternate Clock 4<br>Notes: SPI clock requires a pull-down when controlling most SPI flash devices. |
| PB_11 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SPI2 Master In, Slave Out \| TRACE0 Trace Data 4 \| SMC0 Data 11<br>Notes: Pull-up required for SPI_MISO if SPI master boot is used. |

**Table 15. ADSP-BF70x Designer Quick Reference (Continued)**

| Signal Name | Type | Driver Type | Int Term | Reset Term | Reset Drive | Hiber Term | Hiber Drive | Power Domain | Description and Notes |
|---|---|---|---|---|---|---|---|---|---|
| PB_12 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SPI2 Master Out, Slave In \| TRACE0 Trace Data 3 \| SMC0 Data 12 \| SYS Power Saving Mode Wakeup 2<br>Notes: If hibernate mode is used, one of the following must be true during hibernate. Either this pin must be actively driven by another IC, or it must have a pull-up or pull-down. |
| PB_13 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SPI2 Data 2 \| UART1 Request to Send \| TRACE0 Trace Data 2 \| SMC0 Data 13<br>Notes: No notes. |
| PB_14 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SPI2 Data 3 \| UART1 Clear to Send \| TRACE0 Trace Data 1 \| SMC0 Data 14<br>Notes: No notes. |
| PB_15 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SPI2 Slave Select Output 1 \| TRACE0 Trace Data 0 \| SMC0 Data 15 \| SPI2 Slave Select Input<br>Notes: SPI slave select outputs require a pull-up when used. |
| PC_00 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: UART1 Transmit \| SPT0 Channel A Data 1 \| PPI0 Data 15<br>Notes: No notes. |
| PC_01 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: UART1 Receive \| SPT0 Channel B Data 1 \| PPI0 Data 14 \| SMC0 Address 9 \| TM0 Alternate Capture Input 4<br>Notes: No notes. |
| PC_02 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: UART0 Request to Send \| CAN0 Receive \| PPI0 Data 13 \| SMC0 Address 10 \| SYS Power Saving Mode Wakeup 3 \| TM0 Alternate Capture Input 5<br>Notes: If hibernate mode is used, one of the following must be true during hibernate. Either this pin must be actively driven by another IC, or it must have a pull-up or pull-down. |
| PC_03 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: UART0 Clear to Send \| CAN0 Transmit \| PPI0 Data 12 \| SMC0 Address 11 \| TM0 Alternate Capture Input 0<br>Notes: No notes. |
| PC_04 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SPT0 Channel B Clock \| SPI0 Clock \| MSI0 Data 1 \| SMC0 Address 12 \| TM0 Alternate Clock 0<br>Notes: An external pull-up may be required for MSI modes, see the MSI chapter in the hardware reference for details. |

**Table 15. ADSP-BF70x Designer Quick Reference (Continued)**

| Signal Name | Type | Driver Type | Int Term | Reset Term | Reset Drive | Hiber Term | Hiber Drive | Power Domain | Description and Notes |
|---|---|---|---|---|---|---|---|---|---|
| PC_05 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SPT0 Channel A Frame Sync \| TM0 Timer 3 \| MSI0 Command<br>Notes: An external pull-up may be required for MSI modes, see the MSI chapter in the hardware reference for details. |
| PC_06 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SPT0 Channel B Data 0 \| SPI0 Master In, Slave Out \| MSI0 Data 3<br>Notes: An external pull-up may be required for MSI modes, see the MSI chapter in the hardware reference for details. |
| PC_07 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SPT0 Channel B Frame Sync \| SPI0 Master Out, Slave In \| MSI0 Data 2 \| TM0 Alternate Capture Input 2<br>Notes: An external pull-up may be required for MSI modes, see the MSI chapter in the hardware reference for details. |
| PC_08 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SPT0 Channel A Data 0 \| SPI0 Data 2 \| MSI0 Data 0<br>Notes: An external pull-up may be required for MSI modes, see the MSI chapter in the hardware reference for details. |
| PC_09 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SPT0 Channel A Clock \| SPI0 Data 3 \| MSI0 Clock \| TM0 Alternate Clock 2<br>Notes: No notes. |
| PC_10 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SPT1 Channel B Clock \| MSI0 Data 4 \| SPI1 Slave Select Output 3 \| TM0 Alternate Clock 1<br>Notes: An external pull-up may be required for MSI modes, see the MSI chapter in the hardware reference for details. SPI slave select outputs require a pull-up when used. |
| PC_11 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SPT1 Channel B Frame Sync \| MSI0 Data 5 \| SPI0 Slave Select Output 3<br>Notes: An external pull-up may be required for MSI modes, see the MSI chapter in the hardware reference for details. SPI slave select outputs require a pull-up when used. |
| PC_12 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SPT1 Channel B Data 0 \| MSI0 Data 6<br>Notes: An external pull-up may be required for MSI modes, see the MSI chapter in the hardware reference for details. |

**Table 15. ADSP-BF70x Designer Quick Reference (Continued)**

| Signal Name | Type | Driver Type | Int Term | Reset Term | Reset Drive | Hiber Term | Hiber Drive | Power Domain | Description and Notes |
|---|---|---|---|---|---|---|---|---|---|
| PC_13 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SPT1 Channel B Data 1 \| MSI0 Data 7<br>Notes: An external pull-up may be required for MSI modes, see the MSI chapter in the hardware reference for details. |
| PC_14 | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SPT1 Channel B Transmit Data Valid \| MSI0 eSDIO Interrupt Input<br>Notes: No notes. |
| RTC0_CLKIN | a | na | none | none | none | none | none | VDD_RTC | Desc: RTC0 Crystal input / external oscillator connection<br>Notes: If RTC is not used, connect to ground. |
| RTC0_XTAL | a | na | none | none | none | none | none | VDD_RTC | Desc: RTC0 Crystal output<br>Notes: No notes. |
| SYS_BMODE0 | I/O | na | none | none | none | none | none | VDD_EXT | Desc: SYS Boot Mode Control 0<br>Notes: A pull-down is required for setting to 0 and a pull-up is required for setting to 1. |
| SYS_BMODE1 | I/O | na | none | none | none | none | none | VDD_EXT | Desc: SYS Boot Mode Control 1<br>Notes: A pull-down is required for setting to 0 and a pull-up is required for setting to 1. |
| SYS_CLKIN | a | na | none | none | none | none | none | VDD_EXT | Desc: SYS Clock/Crystal Input<br>Notes: No notes. |
| SYS_CLKOUT | I/O | A | none | none | L | none | none | VDD_EXT | Desc: SYS Processor Clock Output<br>Notes: During reset, SYS_CLKOUT drives out SYS_CLKIN Frequency. |
| SYS_EXTWAKE | I/O | A | none | none | H | none | L | VDD_EXT | Desc: SYS External Wake Control<br>Notes: Drives low during hibernate and high all other times including reset. |
| $\overline{\text{SYS_FAULT}}$ | I/O | A | none | none | none | none | none | VDD_EXT | Desc: SYS Complementary Fault Output<br>Notes: Open drain, requires an external pull-up resistor. |
| $\overline{\text{SYS_HWRST}}$ | I/O | na | none | none | none | none | none | VDD_EXT | Desc: SYS Processor Hardware Reset Control<br>Notes: Active during reset, must be externally driven. |
| $\overline{\text{SYS_NMI}}$ | I/O | na | none | none | none | none | none | VDD_EXT | Desc: SYS Non-maskable Interrupt<br>Notes: Requires an external pull-up resistor. |
| $\overline{\text{SYS_RESOUT}}$ | I/O | A | none | none | L | none | none | VDD_EXT | Desc: SYS Reset Output<br>Notes: Active during reset. |
| SYS_XTAL | a | na | none | none | none | none | none | VDD_EXT | Desc: SYS Crystal Output<br>Notes: Leave unconnected if an oscillator is used to provide SYS_CLKIN. Active during reset. State during hibernate is controlled by DPM_HIB_DIS. |

**Table 15. ADSP-BF70x Designer Quick Reference (Continued)**

| Signal Name | Type | Driver Type | Int Term | Reset Term | Reset Drive | Hiber Term | Hiber Drive | Power Domain | Description and Notes |
|---|---|---|---|---|---|---|---|---|---|
| TWI0_SCL | I/O | D | none | none | none | none | none | VDD_EXT | Desc: TWI0 Serial Clock<br>Notes: Open drain, requires external pull up. Consult version 2.1 of the I2C specification for the proper resistor value. If TWI is not used, connect to ground. |
| TWI0_SDA | I/O | D | none | none | none | none | none | VDD_EXT | Desc: TWI0 Serial Data<br>Notes: Open drain, requires external pull up. Consult version 2.1 of the I2C specification for the proper resistor value. If TWI is not used, connect to ground. |
| USB0_CLKIN | a | na | none | none | none | none | none | VDD_USB | Desc: USB0 Clock/Crystal Input<br>Notes: If USB is not used, connect to ground. Active during reset |
| USB0_DM | I/O | F | none | none | none | none | none | VDD_USB | Desc: USB0 Data –<br>Notes: Pull low if not using USB. For complete documentation of hibernate behavior when USB is used, see the USB chapter in the HRM. |
| USB0_DP | I/O | F | none | none | none | none | none | VDD_USB | Desc: USB0 Data +<br>Notes: Pull low if not using USB. For complete documentation of hibernate behavior when USB is used, see the USB chapter in the HRM. |
| USB0_ID | I/O | na | none | none | none | none | none | VDD_USB | Desc: USB0 OTG ID<br>Notes: If USB is not used connect to ground. When USB is being used, the internal pull-up that is present during hibernate is programmable. See the USB chapter in the HRM. Active during reset. |
| USB0_VBC | I/O | E | none | none | none | none | none | VDD_USB | Desc: USB0 VBUS Control<br>Notes: If USB is not, used pull low. |
| USB0_VBUS | I/O | G | none | none | none | none | none | VDD_USB | Desc: USB0 Bus Voltage<br>Notes: If USB is not used, connect to ground. |
| USB0_XTAL | a | na | none | none | none | none | none | VDD_USB | Desc: USB0 Crystal<br>Notes: No notes. |
| VDD_DMC | s | na | none | none | none | none | none | na | Desc: VDD for DMC<br>Notes: If the DMC is not used, connect to VDD_INT. |
| VDD_EXT | s | na | none | none | none | none | none | na | Desc: External VDD<br>Notes: Must be powered. |
| VDD_HADC | s | na | none | none | none | none | none | na | Desc: VDD for HADC<br>Notes: If HADC is not used, connect to ground. |
| VDD_INT | s | na | none | none | none | none | none | na | Desc: Internal VDD<br>Notes: Must be powered. |

**Table 15. ADSP-BF70x Designer Quick Reference (Continued)**

| Signal Name | Type | Driver Type | Int Term | Reset Term | Reset Drive | Hiber Term | Hiber Drive | Power Domain | Description and Notes |
|---|---|---|---|---|---|---|---|---|---|
| VDD_OTP | s | na | none | none | none | none | none | na | Desc: VDD for OTP<br>Notes: Must be powered. |
| VDD_RTC | s | na | none | none | none | none | none | na | Desc: VDD for RTC<br>Notes: If RTC is not used, connect to ground. |
| VDD_USB | s | na | none | none | none | none | none | na | Desc: VDD for USB<br>Notes: If USB is not used, connect to VDD_EXT. |

# SPECIFICATIONS

For information about product specifications, contact your Analog Devices, Inc. representative.

## OPERATING CONDITIONS

| Parameter | | Test Conditions/Comments | Min | Nominal | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{DD_INT}$ | Internal Supply Voltage | CCLK ≤ 400 MHz | 1.045 | 1.100 | 1.155 | V |
| $V_{DD_EXT}$[1] | External Supply Voltage | | 1.7 | 1.8 | 1.9 | V |
| $V_{DD_EXT}$[1] | External Supply Voltage | | 3.13 | 3.30 | 3.47 | V |
| $V_{DD_DMC}$ | DDR2/LPDDR Supply Voltage | | 1.7 | 1.8 | 1.9 | V |
| $V_{DD_USB}$[2] | USB Supply Voltage | | 3.13 | 3.30 | 3.47 | V |
| $V_{DD_RTC}$ | Real-Time Clock Supply Voltage | | 2.00 | 3.30 | 3.47 | V |
| $V_{DD_HADC}$ | Housekeeping ADC Supply Voltage | | 3.13 | 3.30 | 3.47 | V |
| $V_{DD_OTP}$[1] | OTP Supply Voltage | | | | | |
| | For Reads | | 2.25 | 3.30 | 3.47 | V |
| | For Writes | | 3.13 | 3.30 | 3.47 | V |
| $V_{DDR_VREF}$ | DDR2 Reference Voltage | | 0.49 × $V_{DD_DMC}$ | 0.50 × $V_{DD_DMC}$ | 0.51 × $V_{DD_DMC}$ | V |
| $V_{HADC_REF}$[3] | HADC Reference Voltage | | 2.5 | 3.30 | $V_{DD_HADC}$ | V |
| $V_{IH}$[4] | High Level Input Voltage | $V_{DD_EXT}$ = 3.47 V | 2.0 | | | V |
| $V_{IH}$[4] | High Level Input Voltage | $V_{DD_EXT}$ = 1.9 V | 0.7 × $V_{DD_EXT}$ | | | V |
| $V_{IHTWI}$[5, 6] | High Level Input Voltage | $V_{DD_EXT}$ = maximum | 0.7 × $V_{VBUSTWI}$ | | $V_{VBUSTWI}$ | V |
| $V_{IH_DDR2}$[7] | | $V_{DD_DMC}$ = 1.9 V | $V_{DDR_REF}$ + 0.25 | | | V |
| $V_{IH_LPDDR}$[8] | | $V_{DD_DMC}$ = 1.9 V | 0.8 × $V_{DD_DMC}$ | | | V |
| $V_{ID_DDR2}$[9] | Differential Input Voltage | $V_{IX}$ = 1.075 V | 0.50 | | | V |
| $V_{ID_DDR2}$[9] | Differential Input Voltage | $V_{IX}$ = 0.725 V | 0.55 | | | V |
| $V_{IL}$[4] | Low Level Input Voltage | $V_{DD_EXT}$ = 3.13 V | | | 0.8 | V |
| $V_{IL}$[4] | Low Level Input Voltage | $V_{DD_EXT}$ = 1.7 V | | | 0.3 × $V_{DD_EXT}$ | V |
| $V_{ILTWI}$[5, 6] | Low Level Input Voltage | $V_{DD_EXT}$ = minimum | | | 0.3 × $V_{VBUSTWI}$ | V |
| $V_{IL_DDR2}$[7] | | $V_{DD_DMC}$ = 1.7 V | | | $V_{DDR_REF}$ – 0.25 | V |
| $V_{IL_LPDDR}$[8] | | $V_{DD_DMC}$ = 1.7 V | | | 0.2 × $V_{DD_DMC}$ | V |
| $T_J$ | Junction Temperature | $T_{AMBIENT}$ = 0°C to +70°C | 0 | | 105 | °C |
| $T_J$ | Junction Temperature | $T_{AMBIENT}$ = –40°C to +85°C | –40 | | +105 | °C |
| $T_J$ | Junction Temperature | $T_{AMBIENT}$ = –40°C to +105°C | –40 | | +125 | °C |

[1] Must remain powered (even if the associated function is not used).
[2] If not used, connect to 1.8 V or 3.3 V.
[3] $V_{HADC_VREF}$ should always be less than $V_{DD_HADC}$.
[4] Parameter value applies to all input and bidirectional signals except RTC signals, TWI signals, DMC0 signals, and USB0 signals.
[5] Parameter applies to TWI signals.
[6] TWI signals are pulled up to $V_{BUSTWI}$. See Table 16.
[7] Parameter applies to DMC0 signals in DDR2 mode.
[8] Parameter applies to DMC0 signals in LPDDR mode.
[9] Parameter applies to signals DMC0_LDQS, $\overline{\text{DMC0_LDQS}}$, DMC0_UDQS, $\overline{\text{DMC0_UDQS}}$ when used in DDR2 differential input mode.

**Table 16. TWI_VSEL Selections and $V_{DD_EXT}/V_{BUSTWI}$**

| TWI_DT Setting | $V_{DD_EXT}$ Nominal | $V_{BUSTWI}$ Min | $V_{BUSTWI}$ Nominal | $V_{BUSTWI}$ Max | Unit |
|---|---|---|---|---|---|
| TWI000[1] | 3.30 | 3.13 | 3.30 | 3.47 | V |
| TWI001 | 1.80 | 1.70 | 1.80 | 1.90 | V |
| TWI011 | 1.80 | 3.13 | 3.30 | 3.47 | V |
| TWI100 | 3.30 | 4.75 | 5.00 | 5.25 | V |

[1] Designs must comply with the $V_{DD_EXT}$ and $V_{BUSTWI}$ voltages specified for the default TWI_DT setting for correct JTAG boundary scan operation during reset.

### *Clock Related Operating Conditions*

Table 17 and Table 18 describe the core clock, system clock, and peripheral clock timing requirements. The data presented in the tables applies to all speed grades (found in the Ordering Guide) except where expressly noted. Figure 6 provides a graphical representation of the various clocks and their available divider values.

**Table 17. Core and System Clock Operating Conditions**

| Parameter | | Ratio Restriction | PLLCLK Restriction | Min | Max | Unit |
|---|---|---|---|---|---|---|
| $f_{CCLK}$ | Core Clock Frequency | $f_{CCLK} \geq f_{SYSCLK}$ | PLLCLK = 800 | | 400 | MHz |
| $f_{CCLK}$ | Core Clock Frequency | $f_{CCLK} \geq f_{SYSCLK}$ | 600 ≤ PLLCLK < 800 | | 390 | MHz |
| $f_{CCLK}$ | Core Clock Frequency | $f_{CCLK} \geq f_{SYSCLK}$ | 380 ≤ PLLCLK < 600 | | 380 | MHz |
| $f_{CCLK}$ | Core Clock Frequency | $f_{CCLK} \geq f_{SYSCLK}$ | 230.2 ≤ PLLCLK < 380 | | PLLCLK | MHz |
| $f_{SYSCLK}$ | SYSCLK Frequency[1] | | PLLCLK = 800 | 60 | 200 | MHz |
| $f_{SYSCLK}$ | SYSCLK Frequency[1] | | 600 ≤ PLLCLK < 800 | 60 | 195 | MHz |
| $f_{SYSCLK}$ | SYSCLK Frequency[1] | | 380 ≤ PLLCLK < 600 | 60 | 190 | MHz |
| $f_{SYSCLK}$ | SYSCLK Frequency[1] | | 230.2 ≤ PLLCLK < 380 | 60 | PLLCLK ÷ 2 | MHz |
| $f_{SCLK0}$ | SCLK0 Frequency[1] | $f_{SYSCLK} \geq f_{SCLK0}$ | | 30 | 100 | MHz |
| $f_{SCLK1}$ | SCLK1 Frequency | $f_{SYSCLK} \geq f_{SCLK1}$ | | | 200 | MHz |
| $f_{DCLK}$ | DDR2 Clock Frequency | $f_{SYSCLK} \geq f_{DCLK}$ | | 125 | 200 | MHz |
| $f_{DCLK}$ | LPDDR Clock Frequency | $f_{SYSCLK} \geq f_{DCLK}$ | | 10 | 200 | MHz |

[1] The minimum frequency for SYSCLK and SCLK0 applies only when the USB is used.

**Table 18. Peripheral Clock Operating Conditions**

| Parameter | | Restriction | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $f_{OCLK}$ | Output Clock Frequency | | | | 50 | MHz |
| $f_{SYS_CLKOUTJ}$ | SYS_CLKOUT Period Jitter[1, 2] | | | ±2 | | % |
| $f_{PCLKPROG}$ | Programmed PPI Clock When Transmitting Data and Frame Sync | | | | 50 | MHz |
| $f_{PCLKPROG}$ | Programmed PPI Clock When Receiving Data or Frame Sync | | | | 50 | MHz |
| $f_{PCLKEXT}$ | External PPI Clock When Receiving Data and Frame Sync[3, 4] | $f_{PCLKEXT} \le f_{SCLK0}$ | | | 50 | MHz |
| $f_{PCLKEXT}$ | External PPI Clock Transmitting Data or Frame Sync[3, 4] | $f_{PCLKEXT} \le f_{SCLK0}$ | | | 50 | MHz |
| $f_{SPTCLKPROG}$ | Programmed SPT Clock When Transmitting Data and Frame Sync | | | | 50 | MHz |
| $f_{SPTCLKPROG}$ | Programmed SPT Clock When Receiving Data or Frame Sync | | | | 50 | MHz |
| $f_{SPTCLKEXT}$ | External SPT Clock When Receiving Data and Frame Sync[3, 4] | $f_{SPTCLKEXT} \le f_{SCLK0}$ | | | 50 | MHz |
| $f_{SPTCLKEXT}$ | External SPT Clock Transmitting Data or Frame Sync[3, 4] | $f_{SPTCLKEXT} \le f_{SCLK0}$ | | | 50 | MHz |
| $f_{SPICLKPROG}$ | Programmed SPI Clock When Transmitting Data | | | | 50 | MHz |
| $f_{SPICLKPROG}$ | Programmed SPI Clock When Receiving Data | | | | 50 | MHz |
| $f_{SPICLKEXT}$ | External SPI Clock When Receiving Data[3, 4] | $f_{SPICLKEXT} \le f_{SCLK0}$ | | | 50 | MHz |
| $f_{SPICLKEXT}$ | External SPI Clock When Transmitting Data[3, 4] | $f_{SPICLKEXT} \le f_{SCLK0}$ | | | 50 | MHz |
| $f_{MSICLKPROG}$ | Programmed MSI Clock | | | | 50 | MHz |

[1] SYS_CLKOUT jitter is dependent on the application system design including pin switching activity, board layout, and the jitter characteristics of the SYS_CLKIN source. Due to the dependency on these factors the measured jitter may be higher or lower than this typical specification for each end application.
[2] The value in the Typ field is the percentage of the SYS_CLKOUT period.
[3] The maximum achievable frequency for any peripheral in external clock mode is dependent on being able to meet the setup and hold times in the ac timing specifications section for that peripheral. Pay particular attention to setup and hold times for VDD_EXT = 1.8 V which may preclude the maximum frequency listed here.
[4] The peripheral external clock frequency must also be less than or equal to the $f_{SCLK}$ that clocks the peripheral.

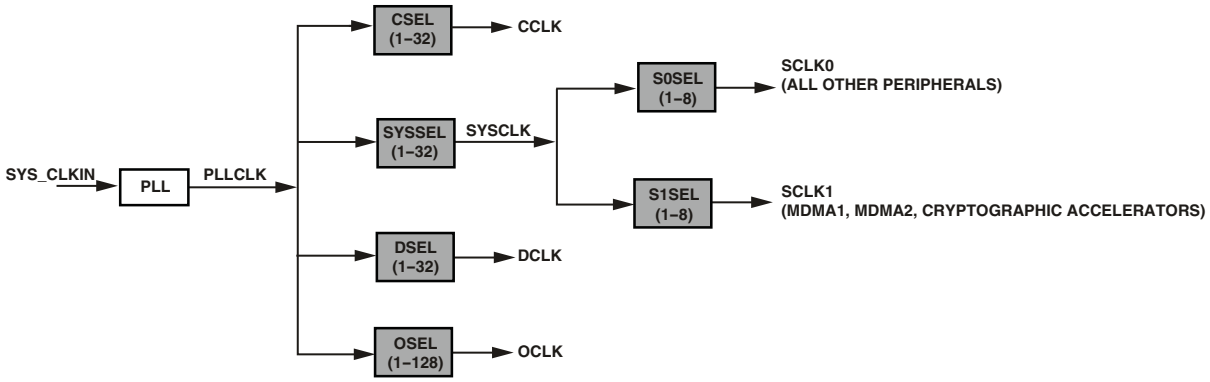



*Figure 6. Clock Relationships and Divider Values*

**Table 19. Phase-Locked Loop Operating Conditions**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| $f_{PLLCLK}$ | PLL Clock Frequency | 230.2 | 800 | MHz |
| CGU_CTL.MSEL[1] | PLL Multiplier | 8 | 41 | |

[1] The CGU_CTL.MSEL setting must also be chosen to ensure that the $f_{PLLCLK}$ specification is not violated.

## ELECTRICAL CHARACTERISTICS

| Parameter | | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $V_{OH}$[1] | High Level Output Voltage | $V_{DD_EXT}$ = 1.7 V, $I_{OH}$ = –1.0 mA | 0.8 × $V_{DD_EXT}$ | | | V |
| $V_{OH}$[1] | High Level Output Voltage | $V_{DD_EXT}$ = 3.13 V, $I_{OH}$ = –2.0 mA | 0.9 × $V_{DD_EXT}$ | | | V |
| $V_{OH_DDR2}$[2] | High Level Output Voltage, DDR2, Programmed Impedance = 34 Ω | $V_{DD_DMC}$ = 1.70 V, $I_{OH}$ = –7.1 mA | $V_{DD_DMC}$ – 0.320 | | | V |
| $V_{OH_DDR2}$[2] | High Level Output Voltage, DDR2, Programmed Impedance = 40 Ω | $V_{DD_DMC}$ = 1.70 V, $I_{OH}$ = –5.8 mA | $V_{DD_DMC}$ – 0.320 | | | V |
| $V_{OH_DDR2}$[2] | High Level Output Voltage, DDR2, Programmed Impedance = 50 Ω | $V_{DD_DMC}$ = 1.70 V, $I_{OH}$ = –4.1 mA | $V_{DD_DMC}$ – 0.320 | | | V |
| $V_{OH_DDR2}$[2] | High Level Output Voltage, DDR2, Programmed Impedance = 60 Ω | $V_{DD_DMC}$ = 1.70 V, $I_{OH}$ = –3.4 mA | $V_{DD_DMC}$ – 0.320 | | | V |
| $V_{OH_LPDDR}$[2] | High Level Output Voltage, LPDDR | $V_{DD_DMC}$ = 1.70 V, $I_{OH}$ = –2.0 mA | $V_{DD_DMC}$ – 0.320 | | | V |
| $V_{OL}$[3] | Low Level Output Voltage | $V_{DD_EXT}$ = 1.7 V, $I_{OL}$ = 1.0 mA | | | 0.400 | V |
| $V_{OL}$[3] | Low Level Output Voltage | $V_{DD_EXT}$ = 3.13 V, $I_{OL}$ = 2.0 mA | | | 0.400 | V |
| $V_{OL_DDR2}$[2] | Low Level Output Voltage, DDR2, Programmed Impedance = 34 Ω | $V_{DD_DMC}$ = 1.70 V, $I_{OL}$ = 7.1 mA | | | 0.320 | V |
| $V_{OL_DDR2}$[2] | Low Level Output Voltage, DDR2, Programmed Impedance = 40 Ω | $V_{DD_DMC}$ = 1.70 V, $I_{OL}$ = 5.8 mA | | | 0.320 | V |
| $V_{OL_DDR2}$[2] | Low Level Output Voltage, DDR2, Programmed Impedance = 50 Ω | $V_{DD_DMC}$ = 1.70 V, $I_{OL}$ = 4.1 mA | | | 0.320 | V |
| $V_{OL_DDR2}$[2] | Low Level Output Voltage, DDR2, Programmed Impedance = 60 Ω | $V_{DD_DMC}$ = 1.70 V, $I_{OL}$ = 3.4 mA | | | 0.320 | V |
| $V_{OL_LPDDR}$[2] | Low Level Output Voltage, LPDDR | $V_{DD_DMC}$ = 1.70 V, $I_{OL}$ = 2.0 mA | | | 0.320 | V |
| $I_{IH}$[4] | High Level Input Current | $V_{DD_EXT}$ = 3.47 V, $V_{DD_DMC}$ = 1.9 V, $V_{DD_USB}$ = 3.47 V, $V_{IN}$ = 3.47 V | | | 10 | µA |
| $I_{IH_DMC0_VREF}$[5] | High Level Input Current | $V_{DD_EXT}$ = 3.47 V, $V_{DD_DMC}$ = 1.9 V, $V_{DD_USB}$ = 3.47 V, $V_{IN}$ = 3.47 V | | | 1 | µA |
| $I_{IH_PD}$[6] | High Level Input Current with Pull-down Resistor | $V_{DD_EXT}$ = 3.47 V, $V_{DD_DMC}$ = 1.9 V, $V_{DD_USB}$ = 3.47 V, $V_{IN}$ = 3.47 V | | | 100 | µA |
| $R_{PD}$[6] | Internal Pull-down Resistance | $V_{DD_EXT}$ = 3.47 V, $V_{DD_DMC}$ = 1.9 V, $V_{DD_USB}$ = 3.47 V, $V_{IN}$ = 3.47 V | 57 | | 130 | kΩ |
| $I_{IL}$[7] | Low Level Input Current | $V_{DD_EXT}$ = 3.47 V, $V_{DD_DMC}$ = 1.9 V, $V_{DD_USB}$ = 3.47 V, $V_{IN}$ = 0 V | | | 10 | µA |
| $I_{IL_DMC0_VREF}$[5] | Low Level Input Current | $V_{DD_EXT}$ = 3.47 V, $V_{DD_DMC}$ = 1.9 V, $V_{DD_USB}$ = 3.47 V, $V_{IN}$ = 0 V | | | 1 | µA |
| $I_{IL_PU}$[8] | Low Level Input Current with Pull-up Resistor | $V_{DD_EXT}$ = 3.47 V, $V_{DD_DMC}$ = 1.9 V, $V_{DD_USB}$ = 3.47 V, $V_{IN}$ = 0 V | | | 100 | µA |
| $R_{PU}$[8] | Internal Pull-up Resistance | $V_{DD_EXT}$ = 3.47 V, $V_{DD_DMC}$ = 1.9 V, $V_{DD_USB}$ = 3.47 V, $V_{IN}$ = 0 V | 53 | | 129 | kΩ |
| $I_{IH_USB0}$[9] | High Level Input Current | $V_{DD_EXT}$ = 3.47 V, $V_{DD_DMC}$ = 1.9 V, $V_{DD_USB}$ = 3.47 V, $V_{IN}$ = 3.47 V | | | 10 | µA |
| $I_{IL_USB0}$[9] | Low Level Input Current | $V_{DD_EXT}$ = 3.47 V, $V_{DD_DMC}$ = 1.9 V, $V_{DD_USB}$ = 3.47 V, $V_{IN}$ = 0 V | | | 10 | µA |
| $I_{OZH}$[10] | Three-State Leakage Current | $V_{DD_EXT}$ = 3.47 V, $V_{DD_DMC}$ = 1.9 V, $V_{DD_USB}$ = 3.47 V, $V_{IN}$ = 3.47 V | | | 10 | µA |
| $I_{OZH}$[11] | Three-State Leakage Current | $V_{DD_EXT}$ = 3.47 V, $V_{DD_DMC}$ = 1.9 V, $V_{DD_USB}$ = 3.47 V, $V_{IN}$ = 1.9 V | | | 10 | µA |
| $I_{OZL}$[12] | Three-State Leakage Current | $V_{DD_EXT}$ = 3.47 V, $V_{DD_DMC}$ = 1.9 V, $V_{DD_USB}$ = 3.47 V, $V_{IN}$ = 0 V | | | 10 | µA |
| $I_{OZH_PD}$[13] | Three-State Leakage Current | $V_{DD_EXT}$ = 3.47 V, $V_{DD_DMC}$ = 1.9 V, $V_{DD_USB}$ = 3.47 V, $V_{IN}$ = 3.47 V | | | 100 | µA |

| Parameter | | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $I_{OZH_TWI}$[14] | Three-State Leakage Current | $V_{DD_EXT}$ = 3.47 V, $V_{DD_DMC}$ = 1.9 V, $V_{DD_USB}$ = 3.47 V, $V_{IN}$ = 5.5 V | | | 10 | µA |
| ADSP-BF701/703/705/707 Input Capacitance | | | | | | |
| $C_{IN}$ (GPIO)[15] | Input Capacitance | $T_{AMBIENT}$ = 25°C | | 5.2 | 6.0 | pF |
| $C_{IN_TWI}$[14] | Input Capacitance | $T_{AMBIENT}$ = 25°C | | 6.9 | 7.4 | pF |
| $C_{IN_DDR}$[16] | Input Capacitance | $T_{AMBIENT}$ = 25°C | | 6.1 | 6.9 | pF |
| ADSP-BF700/702/704/706 Input Capacitance | | | | | | |
| $C_{IN}$ (GPIO)[15] | Input Capacitance | $T_{AMBIENT}$ = 25°C | | 5.0 | 5.3 | pF |
| $C_{IN_TWI}$[14] | Input Capacitance | $T_{AMBIENT}$ = 25°C | | 6.8 | 7.4 | pF |
| $I_{DD_DEEPSLEEP}$[17, 18] | $V_{DD_INT}$ Current in Deep Sleep Mode | Clocks disabled<br>$T_J$ = 25°C | | 1.4 | | mA |
| $I_{DD_IDLE}$[18] | $V_{DD_INT}$ Current in Idle | $f_{PLLCLK}$ = 300 MHz<br>$f_{CCLK}$ = 100 MHz<br>ASF = 0.05 (idle)<br>$f_{SYSCLK}$ = $f_{SCLK0}$ = 25 MHz<br>USBCLK = DCLK = OUTCLK = SCLK1 = DISABLED<br>Peripherals disabled<br>$T_J$ = 25°C | | 13 | | mA |
| $I_{DD_TYP}$[18] | $V_{DD_INT}$ Current | $f_{PLLCLK}$ = 800 MHz<br>$f_{CCLK}$ = 400 MHz<br>ASF = 1.0 (full-on typical)<br>$f_{SYSCLK}$ = $f_{SCLK0}$ = 25 MHz<br>USBCLK = DCLK = OUTCLK = SCLK1 = DISABLED<br>Peripherals disabled<br>$T_J$ = 25°C | | 90 | | mA |
| $I_{DD_TYP}$[18] | $V_{DD_INT}$ Current | $f_{PLLCLK}$ = 300 MHz<br>$f_{CCLK}$ = 300 MHz<br>ASF = 1.0 (full-on typical)<br>$f_{SYSCLK}$ = $f_{SCLK0}$ = 25 MHz<br>USBCLK = DCLK = OUTCLK = SCLK1 = DISABLED<br>Peripherals disabled<br>$T_J$ = 25°C | | 66 | | mA |
| $I_{DD_TYP}$[18] | $V_{DD_INT}$ Current | $f_{PLLCLK}$ = 400 MHz<br>$f_{CCLK}$ = 200 MHz<br>ASF = 1.0 (full-on typical)<br>$f_{SYSCLK}$ = $f_{SCLK0}$ = 25 MHz<br>USBCLK = DCLK = OUTCLK = SCLK1 = DISABLED<br>Peripherals disabled<br>$T_J$ = 25°C | | 49 | | mA |
| $I_{DD_TYP}$[18] | $V_{DD_INT}$ Current | $f_{PLLCLK}$ = 300 MHz<br>$f_{CCLK}$ = 100 MHz<br>ASF = 1.0 (full-on typical)<br>$f_{SYSCLK}$ = $f_{SCLK0}$ = 25 MHz<br>USBCLK = DCLK = OUTCLK = SCLK1 = DISABLED<br>Peripherals disabled<br>$T_J$ = 25°C | | 30 | | mA |

| Parameter | | Test Conditions/Comments | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|
| $I_{DD_HIBERNATE}$[17, 19] | Hibernate State Current | $V_{DD_INT}$ = 0 V,<br>$V_{DD_DMC}$ = 1.8 V,<br>$V_{DD_EXT}$ = $V_{DD_HADC}$ = $V_{DD_OTP}$ =<br>$V_{DD_RTC}$ = $V_{DD_USB}$ = 3.3 V,<br>$T_J$ = 25°C,<br>$f_{CLKIN}$ = 0 | | 33 | | µA |
| $I_{DD_HIBERNATE}$[17, 19] | Hibernate State Current Without USB | $V_{DD_INT}$ = 0 V,<br>$V_{DD_DMC}$ = 1.8 V,<br>$V_{DD_EXT}$ = $V_{DD_HADC}$ = $V_{DD_OTP}$ =<br>$V_{DD_RTC}$ = $V_{DD_USB}$ = 3.3 V,<br>$T_J$ = 25°C,<br>$f_{CLKIN}$ = 0,<br>USB protection disabled<br>(USB_PHY_CTLDIS = 1) | | 15 | | µA |
| $I_{DD_INT}$[18] | $V_{DD_INT}$ Current | $V_{DD_INT}$ within operating conditions table specifications | | | See $I_{DDINT_TOT}$ equation on on Page 56 | mA |
| $I_{DD_RTC}$ | $I_{DD_RTC}$ Current | $V_{DD_RTC}$ = 3.3 V, $T_J$ = 125°C | | | 10 | µA |

[1] Applies to all output and bidirectional signals except DMC0 signals, TWI signals, and USB0 signals.

[2] Applies to DMC0_Axx, DMC0_CAS, DMC0_CKE, DMC0_CK, DMC0_CK, DMC0_CS, DMC0_DQxx, DMC0_LDM, DMC0_LDQS, DMC0_LDQS, DMC0_ODT, DMC0_RAS, DMC0_UDM, DMC0_UDQS, DMC0_UDQS, and DMC0_WE signals.

[3] Applies to all output and bidirectional signals except DMC0 signals and USB0 signals.

[4] Applies to SMC0_ARDY, SYS_BMODEx, SYS_CLKIN, SYS_HWRST, JTG_TDI, and JTG_TMS_SWDIO signals.

[5] Applies to DMC0_VREF signal.

[6] Applies to JTG_TCK_SWCLK and JTG_TRST signals.

[7] Applies to SMC0_ARDY, SYS_BMODEx, SYS_CLKIN, SYS_HWRST, JTG_TCK, and JTG_TRST signals.

[8] Applies to JTG_TDI, JTG_TMS_SWDIO, PA_xx, PB_xx, and PC_xx signals when internal GPIO pull-ups are enabled. For information on when internal pull-ups are enabled for GPIOs. See ADSP-BF70x Designer Quick Reference on Page 38.

[9] Applies to USB0_CLKIN signal.

[10] Applies to PA_xx, PB_xx, PC_xx, SMC0_AMS0, SMC0_ARE, SMC0_AWE, SMC0_A0E, SMC0_Axx, SMC0_Dxx, SYS_FAULT, JTG_TDO_SWO, USB0_DM, USB0_DP, USB0_ID, and USB0_VBC signals.

[11] Applies to DMC0_Axx, DMC0_BAxx, DMC0_CAS, DMC0_CS0, DMC0_DQxx, DMC0_LDQS, DMC0_LDQS, DMC0_UDQS, DMC0_UDQS, DMC0_LDM, DMC0_UDM, DMC0_ODT, DMC0_RAS, and DMC0_WE signals.

[12] Applies to PA_xx, PB_xx, PC_xx, SMC0_A0E, SMC0_Axx, SMC0_Dxx, SYS_FAULT, JTG_TDO_SWO, USB0_DM, USB0_DP, USB0_ID, USB0_VBC, USB0_VBUS, DMC0_Axx, DMC0_BAx, DMC0_CAS, DMC0_CS0, DMC0_DQxx, DMC0_LDQS, DMC0_LDQS, DMC0_UDQS, DMC0_UDQS, DMC0_LDM, DMC0_UDM, DMC0_ODT, DMC0_RAS, DMC0_WE, and TWI signals.

[13] Applies to USB0_VBUS signals.

[14] Applies to all TWI signals.

[15] Applies to all signals, except DMC0 and TWI signals.

[16] Applies to all DMC0 signals.

[17] See the *ADSP-BF70x Blackfin+ Processor Hardware Reference* for definition of deep sleep and hibernate operating modes.

[18] Additional information can be found at Total Internal Power Dissipation.

[19] Applies to VDD_EXT, VDD_DMC, and VDD_USB supply signals only. Clock inputs are tied high or low.

### *Total Internal Power Dissipation*

Total power dissipation has two components:

1. Static, including leakage current (deep sleep)
2. Dynamic, due to transistor switching characteristics for each clock domain

Many operating conditions can also affect power dissipation, including temperature, voltage, operating frequency, and processor activity. The following equation describes the internal current consumption.

$$I_{DDINT_TOT} = I_{DDINT_DEEPSLEEP} + I_{DDINT_CCLK_DYN} + I_{DDINT_PLLCLK_DYN} + I_{DDINT_SYSCLK_DYN} + I_{DDINT_SCLK0_DYN} + I_{DDINT_SCLK1_DYN} + I_{DDINT_DCLK_DYN} + I_{DDINT_DMA_DR_DYN} + I_{DDINT_USBCLK_DYN}$$

$I_{DDINT_DEEPSLEEP}$ is the only item present that is part of the static power dissipation component. $I_{DDINT_DEEPSLEEP}$ is specified as a function of voltage ($V_{DD_INT}$) and temperature (see Table 21).

There are eight different items that contribute to the dynamic power dissipation. These components fall into three broad categories: application-dependent currents, clock currents, and data transmission currents.

#### Application-Dependent Current

The application-dependent currents include the dynamic current in the core clock domain.

Core clock (CCLK) use is subject to an activity scaling factor (ASF) that represents application code running on the processor cores and L1/L2 memories (Table 22). The ASF is combined with the CCLK frequency and $V_{DD_INT}$ dependent data in Table 23 to calculate this portion.

$I_{DDINT_CCLK_DYN}$ (mA) = Table 23 × *ASF*

#### Clock Current

The dynamic clock currents provide the total power dissipated by all transistors switching in the clock paths. The power dissipated by each clock domain is dependent on voltage ($V_{DD_INT}$), operating frequency and a unique scaling factor.

$I_{DDINT_PLLCLK_DYN}$ (mA) = 0.012 × $f_{PLLCLK}$ (MHz) × $V_{DD_INT}$ (V)

$I_{DDINT_SYSCLK_DYN}$ (mA) = 0.120 × $f_{SYSCLK}$ (MHz) × $V_{DD_INT}$ (V)

$I_{DDINT_SCLK0_DYN}$ (mA) = 0.110 × $f_{SCLK0}$ (MHz) × $V_{DD_INT}$ (V)

$I_{DDINT_SCLK1_DYN}$ (mA) = 0.068 × $f_{SCLK1}$ (MHz) × $V_{DD_INT}$ (V)

$I_{DDINT_DCLK_DYN}$ (mA) = 0.055 × $f_{DCLK}$ (MHz) × $V_{DD_INT}$ (V)

The dynamic component of the USB clock is a unique case. The USB clock contributes a near constant current value when used.

**Table 20. $I_{DDINT_USBCLK_DYN}$ Current**

| Is USB Enabled? | $I_{DDINT_USBCLK_DYN}$ (mA) |
|---|---|
| Yes – High-Speed Mode | 13.94 |
| Yes – Full-Speed Mode | 10.83 |
| Yes – Suspend Mode | 5.2 |
| No | 0.34 |

#### Data Transmission Current

The data transmission current represents the power dissipated when transmitting data. This current is expressed in terms of data rate. The calculation is performed by adding the data rate (MB/s) of each DMA-driven access to peripherals, L1, L2, and external memory. This number is then multiplied by a weighted data-rate coefficient and $V_{DD_INT}$:

$I_{DDINT_DMADR_DYN}$ (mA) = *Weighted DRC* × *Total Data Rate* (MB/s) × $V_{DD_INT}$ (V)

A weighted data-rate coefficient is used because different coefficients exist depending on the source and destination of the transfer. For details on using this equation and calculating the weighted DRC, see the related Engineer Zone material. For a quick maximum calculation, the weighted DRC can be assumed to be 0.0497, which is the coefficient for L1 to L1 transfers.

**Table 21. Static Current—$I_{DD_DEEPSLEEP}$ (mA)**

| | Voltage ($V_{DD_INT}$) | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $T_J$ (°C) | 1.045 | 1.050 | 1.060 | 1.070 | 1.080 | 1.090 | 1.100 | 1.110 | 1.120 | 1.130 | 1.140 | 1.150 | 1.155 |
| –40 | 0.6 | 0.6 | 0.7 | 0.7 | 0.7 | 0.8 | 0.8 | 0.8 | 0.9 | 0.9 | 0.9 | 1.0 | 1.0 |
| –20 | 1.1 | 1.1 | 1.2 | 1.2 | 1.2 | 1.3 | 1.4 | 1.4 | 1.5 | 1.5 | 1.6 | 1.7 | 1.7 |
| 0 | 2.0 | 2.0 | 2.1 | 2.2 | 2.3 | 2.4 | 2.5 | 2.5 | 2.6 | 2.7 | 2.8 | 3.0 | 3.0 |
| 25 | 4.3 | 4.3 | 4.5 | 4.7 | 4.8 | 5.0 | 5.2 | 5.3 | 5.5 | 5.7 | 5.9 | 6.1 | 6.2 |
| 40 | 6.7 | 6.8 | 7.0 | 7.3 | 7.5 | 7.8 | 8.0 | 8.3 | 8.6 | 8.8 | 9.1 | 9.4 | 9.6 |
| 55 | 10.3 | 10.5 | 10.8 | 11.2 | 11.5 | 11.9 | 12.3 | 12.6 | 13.0 | 13.4 | 13.9 | 14.3 | 14.5 |
| 70 | 15.7 | 15.9 | 16.4 | 16.8 | 17.4 | 17.9 | 18.4 | 18.9 | 19.5 | 20.1 | 20.7 | 21.3 | 21.6 |
| 85 | 23.3 | 23.6 | 24.3 | 25.0 | 25.7 | 26.4 | 27.2 | 27.9 | 28.7 | 29.5 | 30.4 | 31.2 | 31.7 |
| 100 | 34.2 | 34.6 | 35.5 | 36.5 | 37.5 | 38.5 | 39.5 | 40.6 | 41.7 | 42.8 | 43.9 | 45.1 | 45.7 |
| 105 | 38.7 | 39.2 | 40.2 | 41.3 | 42.4 | 43.5 | 44.6 | 45.8 | 47.0 | 48.2 | 49.5 | 50.8 | 51.5 |
| 115 | 48.9 | 49.5 | 50.7 | 52.0 | 53.4 | 54.7 | 56.0 | 57.5 | 59.0 | 60.5 | 62.0 | 63.6 | 64.4 |
| 125 | 61.5 | 62.1 | 63.6 | 65.1 | 66.7 | 68.3 | 69.9 | 71.7 | 73.4 | 75.2 | 77.0 | 79.0 | 79.9 |

**Table 22. Activity Scaling Factors (ASF)**

| $I_{DDINT}$ Power Vector | ASF |
|---|---|
| $I_{DD-IDLE1}$ | 0.05 |
| $I_{DD-IDLE2}$ | 0.05 |
| $I_{DD-NOP1}$ | 0.56 |
| $I_{DD-NOP2}$ | 0.59 |
| $I_{DD-APP3}$ | 0.78 |
| $I_{DD-APP1}$ | 0.79 |
| $I_{DD-APP2}$ | 0.83 |
| $I_{DD-TYP1}$ | 1.00 |
| $I_{DD-TYP3}$ | 1.01 |
| $I_{DD-TYP2}$ | 1.03 |
| $I_{DD-HIGH1}$ | 1.39 |
| $I_{DD-HIGH3}$ | 1.39 |
| $I_{DD-HIGH2}$ | 1.54 |

**Table 23. CCLK Dynamic Current per core (mA, with ASF = 1)**

| | Voltage ($V_{DD_INT}$) | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $f_{CCLK}$ (MHz) | 1.045 | 1.050 | 1.060 | 1.070 | 1.080 | 1.090 | 1.100 | 1.110 | 1.120 | 1.130 | 1.140 | 1.150 | 1.155 |
| 400 | 66.7 | 67.2 | 67.9 | 68.7 | 69.4 | 70.2 | 71.1 | 71.8 | 72.6 | 73.4 | 74.2 | 74.9 | 75.4 |
| 350 | 58.6 | 59.0 | 59.6 | 60.3 | 61.0 | 61.7 | 62.4 | 63.0 | 63.7 | 64.4 | 65.1 | 65.8 | 66.1 |
| 300 | 50.2 | 50.5 | 51.1 | 51.7 | 52.3 | 52.9 | 53.5 | 54.1 | 54.7 | 55.3 | 55.9 | 56.4 | 56.8 |
| 250 | 42.1 | 42.3 | 42.8 | 43.3 | 43.8 | 44.3 | 44.7 | 45.3 | 45.8 | 46.3 | 46.8 | 47.4 | 47.6 |
| 200 | 33.7 | 33.9 | 34.3 | 34.7 | 35.1 | 35.5 | 35.9 | 36.3 | 36.7 | 37.1 | 37.5 | 37.9 | 38.0 |
| 150 | 25.4 | 25.5 | 25.8 | 26.1 | 26.4 | 26.7 | 27.0 | 27.3 | 27.6 | 27.9 | 28.2 | 28.5 | 28.8 |
| 100 | 17.0 | 17.1 | 17.3 | 17.5 | 17.7 | 17.9 | 18.1 | 18.3 | 18.5 | 18.6 | 18.8 | 19.0 | 19.1 |

## HADC

### *HADC Electrical Characteristics*

**Table 24. HADC Electrical Characteristics**

| Parameter | Test Conditions | Typ | Unit |
|---|---|---|---|
| $I_{DD_HADC_IDLE}$ | Current Consumption on $V_{DD_HADC}$. HADC is powered on, but not converting. | 2.0 | mA |
| $I_{DD_HADC_ACTIVE}$ | Current Consumption on $V_{DD_HADC}$ during a conversion. | 2.5 | mA |
| $I_{DD_HADC_POWERDOWN}$ | Current Consumption on $V_{DD_HADC}$. Analog circuitry of the HADC is powered down | 10 | µA |

### *HADC DC Accuracy*

**Table 25. HADC DC Accuracy**

| Parameter | Typ | Unit |
|---|---|---|
| Resolution | 12 | Bits |
| No Missing Codes (NMC) | 10 | Bits |
| Integral Nonlinearity (INL) | ±2 | LSB[1] |
| Differential Nonlinearity (DNL) | ±2 | LSB[1] |
| Offset Error | ±8 | LSB[1] |
| Offset Error Matching | ±10 | LSB[1] |
| Gain Error | ±4 | LSB[1] |
| Gain Error Matching | ±4 | LSB[1] |

[1] LSB = HADC0_VREFP ÷ 4096

### *HADC Timing Specifications*

**Table 26. HADC Timing Specifications**

| Parameter | Typ | Max | Unit |
|---|---|---|---|
| Conversion Time | $20 \times T_{SAMPLE}$ | | µs |
| Throughput Range | | 1 | MSPS |
| $T_{WAKEUP}$ | | 100 | µs |

## PACKAGE INFORMATION

The information presented in Figure 7 and Table 27 provides details about package branding. For a complete listing of product availability, see the Ordering Guide.

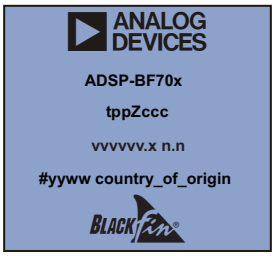



*Figure 7. Product Information on Package*[1]

[1] Exact brand may differ, depending on package type.

**Table 27. Package Brand Information**

| Brand Key | Field Description |
|---|---|
| ADSP-BF70x | Product model |
| t | Temperature range |
| pp | Package type |
| Z | RoHS compliant designation |
| ccc | See Ordering Guide |
| vvvvvv.x | Assembly lot code |
| n.n | Silicon revision |
| yyww | Date code |

## ABSOLUTE MAXIMUM RATINGS

Stresses at or above those listed in Table 28 may cause permanent damage to the product. This is a stress rating only; functional operation of the product at these or any other conditions above those indicated in the operational section of this specification is not implied. Operation beyond the maximum operating conditions for extended periods may affect product reliability.

**Table 28. Absolute Maximum Ratings**

| Parameter | Rating |
|---|---|
| Internal Supply Voltage ($V_{DD_INT}$) | –0.33 V to +1.20 V |
| External (I/O) Supply Voltage ($V_{DD_EXT}$) | –0.33 V to +3.60 V |
| DDR2 Controller Supply Voltage ($V_{DD_DMC}$) | –0.33 V to +1.90 V |
| USB PHY Supply Voltage ($V_{DD_USB}$) | –0.33 V to +3.60 V |
| Real-Time Clock Supply Voltage ($V_{DD_RTC}$) | –0.33 V to +3.60 V |
| Housekeeping ADC Supply Voltage ($V_{DD_HADC}$) | –0.33 V to +3.60 V |
| One-Time Programmable Memory Supply Voltage ($V_{DD_OTP}$) | –0.33 V to +3.60 V |
| HADC Reference Voltage ($V_{HADC_REF}$) | –0.33 V to +3.60 V |

**Table 28. Absolute Maximum Ratings (Continued)**

| Parameter | Rating |
|---|---|
| DDR2 Reference Voltage ($V_{DDR_REF}$) | –0.33 V to +1.90 V |
| Input Voltage[1, 2] | –0.33 V to +3.60 V |
| TWI Input Voltage[2, 3] | –0.33 V to +5.50 V |
| USB0_Dx Input Voltage[4] | –0.33 V to +5.25 V |
| USB0_VBUS Input Voltage[5] | –0.33 V to +6 V |
| DDR2 Input Voltage[5] | –0.33 V to +1.90 V |
| Output Voltage Swing | –0.33 V to $V_{DD_EXT}$ + 0.5 V |
| $I_{OH}/I_{OL}$ Current per Signal[1] | 4 mA (max) |
| Storage Temperature Range | –65°C to +150°C |
| Junction Temperature While Biased | +125°C |

[1] Applies to 100% transient duty cycle.
[2] Applies only when $V_{DD_EXT}$ is within specifications. When $V_{DD_EXT}$ is outside specifications, the range is $V_{DD_EXT}$ ± 0.2 V.
[3] Applies to balls TWI_SCL and TWI_SDA.
[4] If the USB is not used, connect USB0_Dx and USB0_VBUS according to Table 15 on Page 38.
[5] Applies only when $V_{DD_DMC}$ is within specifications. When $V_{DD_DMC}$ is outside specifications, the range is $V_{DD_DMC}$ ± 0.2 V.

## 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.

## TIMING SPECIFICATIONS

Specifications are subject to change without notice.

### *Clock and Reset Timing*

Table 29 and Figure 8 describe clock and reset operations related to the clock generation unit (CGU). Per the CCLK, SYSCLK, SCLK0, SCLK1, DCLK, and OCLK timing specifications in Table 17 on Page 51 and Table 18 on Page 52, combinations of SYS_CLKIN and clock multipliers must not select clock rates in excess of the processor's maximum instruction rate.

**Table 29. Clock and Reset Timing**

| Parameter | | $V_{DD_EXT}$ 1.8 V Nominal Min | $V_{DD_EXT}$ 1.8 V Nominal Max | $V_{DD_EXT}$ 3.3 V Nominal Min | $V_{DD_EXT}$ 3.3 V Nominal Max | Unit |
|---|---|---|---|---|---|---|
| *Timing Requirement* | | | | | | |
| $f_{CKIN}$ | SYS_CLKIN Crystal Frequency (CGU_CTL.DF = 0)[1, 2, 3] | 19.2 | 35 | 19.2 | 50 | MHz |
| $f_{CKIN}$ | SYS_CLKIN Crystal Frequency (CGU_CTL.DF = 1)[1, 2, 3] | N/A | N/A | 38.4 | 50 | MHz |
| $f_{CKIN}$ | SYS_CLKIN External Source Frequency (CGU_CTL.DF = 0)[1, 2, 3] | 19.2 | 60 | 19.2 | 60 | MHz |
| $f_{CKIN}$ | SYS_CLKIN External Source Frequency (CGU_CTL.DF = 1)[1, 2, 3] | 38.4 | 60 | 38.4 | 60 | MHz |
| $t_{CKINL}$ | SYS_CLKIN Low Pulse[1] | 8.33 | | 8.33 | | ns |
| $t_{CKINH}$ | SYS_CLKIN High Pulse[1] | 8.33 | | 8.33 | | ns |
| $t_{WRST}$ | $\overline{\text{SYS_HWRST}}$ Asserted Pulse Width Low[4] | 11 × $t_{CKIN}$ | | 11 × $t_{CKIN}$ | | ns |

[1] Applies to PLL bypass mode and PLL nonbypass mode.
[2] The $t_{CKIN}$ period (see Figure 8) equals $1/f_{CKIN}$.
[3] Combinations of the CLKIN frequency and the PLL clock multiplier must not exceed the allowed $f_{PLLCLK}$ setting discussed in Table 19.
[4] Applies after power-up sequence is complete. See Table 30 and Figure 9 for power-up reset timing.

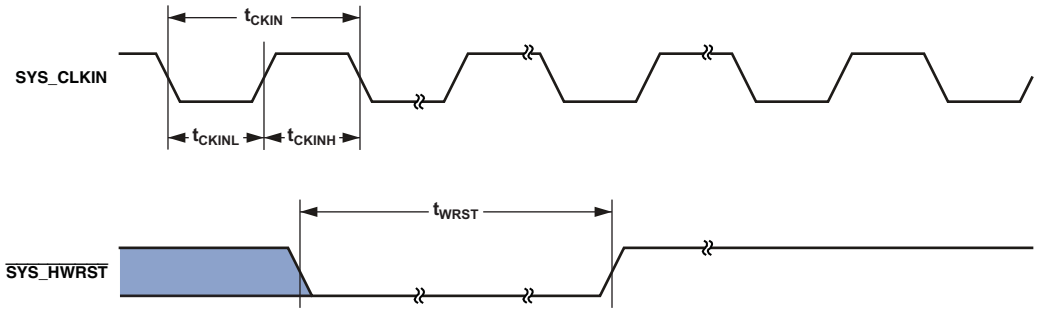



*Figure 8. Clock and Reset Timing*

### *Power-Up Reset Timing*

A power-up reset is required to place the processor in a known state after power-up. A power-up reset is initiated by asserting SYS_HWRST and JTG_TRST. During power-up reset, all pins are high impedance except for those noted in the ADSP-BF70x Designer Quick Reference on Page 38.

Both JTG_TRST and SYS_HWRST need to be asserted upon power-up, but only SYS_HWRST needs to be released for the device to boot properly. JTG_TRST may be asserted indefinitely for normal operation. JTG_TRST only needs to be released when using an emulator to connect to the DAP for debug or boundary scan. There is an internal pull-down on JTG_TRST to ensure internal emulation logic will always be properly initialized during power-up reset.

Table 30 and Figure 9 show the relationship between power supply startup and processor reset timing, related to the clock generation unit (CGU) and reset control unit (RCU). In Figure 9, $V_{DD_SUPPLIES}$ are $V_{DD_INT}$, $V_{DD_EXT}$, $V_{DD_DMC}$, $V_{DD_USB}$, $V_{DD_RTC}$, $V_{DD_OTP}$, and $V_{DD_HADC}$.

There is no power supply sequencing requirement for the ADSP-BF70x processor. However, if saving power during power-on is important, bringing up $V_{DD_INT}$ last is recommended. This avoids a small current drain in the $V_{DD_INT}$ domain during the transition period of I/O voltages from 0 V to within the voltage specification.

**Table 30. Power-Up Reset Timing**

| Parameter | | Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirement* | | | | |
| $t_{RST_IN_PWR}$ | SYS_HWRST and JTG_TRST Deasserted After $V_{DD_INT}$, $V_{DD_DMC}$, $V_{DD_USB}$, $V_{DD_RTC}$, $V_{DD_OTP}$, $V_{DD_HADC}$, and SYS_CLKIN are Stable and Within Specification | $11 \times t_{CKIN}$ | | ns |
| $t_{VDDEXT_RST}$ | SYS_HWRST Deasserted After $V_{DD_EXT}$ is Stable and Within Specifications (No External Pull-Down on JTG_TRST) | 10 | | µs |
| $t_{VDDEXT_RST}$ | SYS_HWRST Deasserted After $V_{DD_EXT}$ is Stable and Within Specifications (10k External Pull-Down on JTG_TRST) | 1 | | µs |

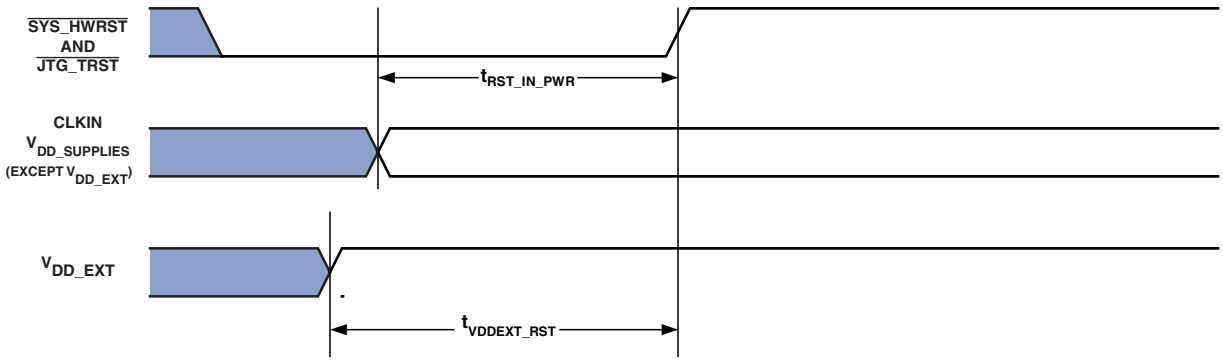



*Figure 9. Power-Up Reset Timing*

### *Asynchronous Read*

Table 31 and Figure 10 show asynchronous memory read timing, related to the static memory controller (SMC).

**Table 31. Asynchronous Memory Read (BxMODE = b#00)**

| Parameter | | $V_{DD_EXT}$ 1.8 V Nominal Min | $V_{DD_EXT}$ 1.8 V Nominal Max | $V_{DD_EXT}$ 3.3 V Nominal Min | $V_{DD_EXT}$ 3.3 V Nominal Max | Unit |
|---|---|---|---|---|---|---|
| *Timing Requirements* | | | | | | |
| $t_{SDATARE}$ | DATA in Setup Before $\overline{SMC0_ARE}$ High | 11.8 | | 10.8 | | ns |
| $t_{HDATARE}$ | DATA in Hold After $\overline{SMC0_ARE}$ High | 0 | | 0 | | ns |
| $t_{DARDYARE}$ | SMC0_ARDY Valid After $\overline{SMC0_ARE}$ Low[1, 2] | | (RAT – 2.5) × $t_{SCLK0}$ – 17.5 | | (RAT – 2.5) × $t_{SCLK0}$ – 17.5 | ns |
| *Switching Characteristics* | | | | | | |
| $t_{AMSARE}$ | SMC0_Ax/$\overline{SMC0_AMSx}$ Assertion Before $\overline{SMC0_ARE}$ Low[3] | (PREST + RST + PREAT) × $t_{SCLK0}$ – 2 | | (PREST + RST + PREAT) × $t_{SCLK0}$ – 2 | | ns |
| $t_{DADVARE}$ | SMC0_ARE Low Delay From ADV High | PREAT × $t_{SCLK0}$ – 2 | | PREAT × $t_{SCLK0}$ – 2 | | ns |
| $t_{AOEARE}$ | $\overline{SMC0_AOE}$ Assertion Before $\overline{SMC0_ARE}$ Low | (RST + PREAT) × $t_{SCLK0}$ – 2 | | (RST + PREAT) × $t_{SCLK0}$ – 2 | | ns |
| $t_{HARE}$ | Output[4] Hold After $\overline{SMC0_ARE}$ High[5] | RHT × $t_{SCLK0}$ – 2 | | RHT × $t_{SCLK0}$ – 2 | | ns |
| $t_{WARE}$ | $\overline{SMC0_ARE}$ Active Low Width[6] | RAT × $t_{SCLK0}$ – 2 | | RAT × $t_{SCLK0}$ – 2 | | ns |
| $t_{DAREARDY}$ | $\overline{SMC0_ARE}$ High Delay After SMC0_ARDY Assertion[1] | | 3.5 × $t_{SCLK0}$ + 17.5 | | 3.5 × $t_{SCLK0}$ + 17.5 | ns |

[1] SMC0_BxCTL.ARDYEN bit = 1.
[2] RAT value set using the SMC_BxTIM.RAT bits.
[3] PREST, RST, and PREAT values set using the SMC_BxETIM.PREST bits, SMC_BxTIM.RST bits, and the SMC_BxETIM.PREAT bits.
[4] Output signals are SMC0_Ax, $\overline{SMC0_AMSx}$, $\overline{SMC0_AOE}$, and $\overline{SMC0_ABEx}$.
[5] RHT value set using the SMC_BxTIM.RHT bits.
[6] SMC0_BxCTL.ARDYEN bit = 0.

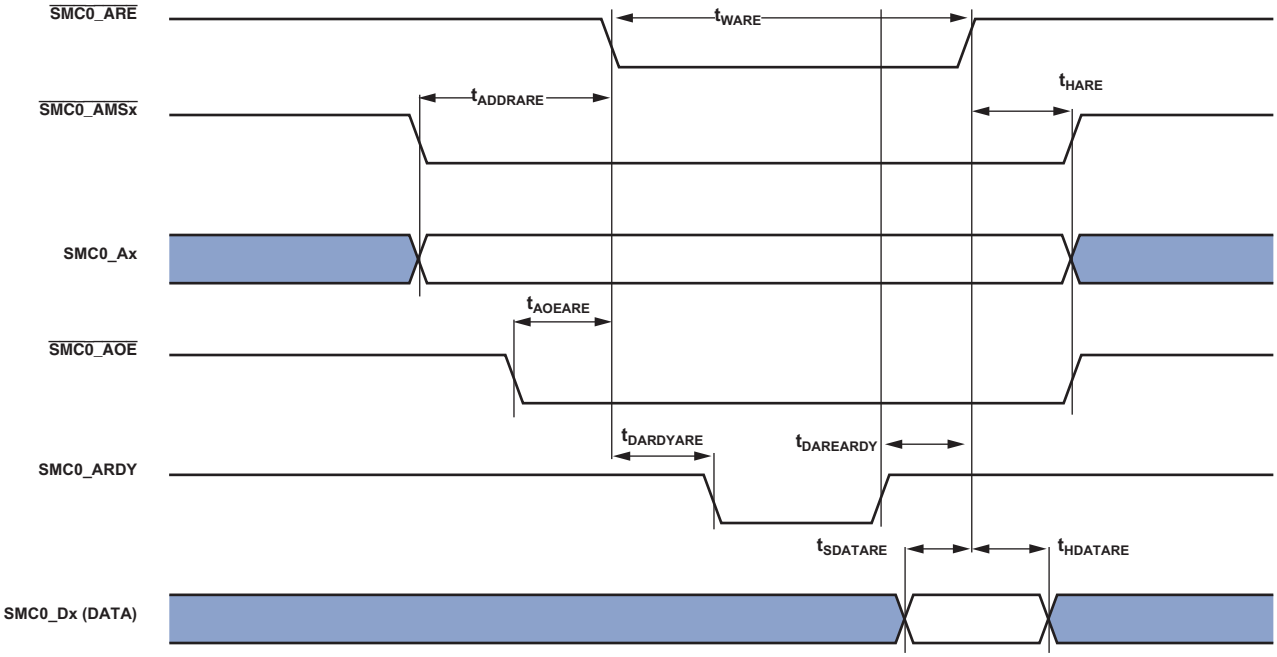



*Figure 10. Asynchronous Read*

### *SMC Read Cycle Timing With Reference to SYS_CLKOUT*

The following SMC specifications with respect to SYS_CLKOUT are given to accommodate the connection of the SMC to programmable logic devices. These specifications assume that SYS_CLKOUT is outputting a buffered version of SCLK0 by setting CGU_CLKOUTSEL.CLKOUTSEL = 0x3. However, SCLK0 must not run faster than the maximum $f_{OCLK}$ specification. For this example, RST = 0x2, RAT = 0x4, and RHT = 0x1.

**Table 32. SMC Read Cycle Timing With Reference to SYS_CLKOUT (BxMODE = b#00)**

| Parameter | | $V_{DD_EXT}$ 1.8 V Nominal Min | $V_{DD_EXT}$ 1.8 V Nominal Max | $V_{DD_EXT}$ 3.3 V Nominal Min | $V_{DD_EXT}$ 3.3 V Nominal Max | Unit |
|---|---|---|---|---|---|---|
| *Timing Requirements* | | | | | | |
| $t_{SDAT}$ | SMC0_Dx Setup Before SYS_CLKOUT | 5.3 | | 4.3 | | ns |
| $t_{HDAT}$ | SMC0_Dx Hold After SYS_CLKOUT | 1.5 | | 1.5 | | ns |
| $t_{SARDY}$ | SMC0_ARDY Setup Before SYS_CLKOUT | 16.6 | | 14.4 | | ns |
| $t_{HARDY}$ | SMC0_ARDY Hold After SYS_CLKOUT | 0.7 | | 0.7 | | ns |
| *Switching Characteristics* | | | | | | |
| $t_{DO}$ | Output Delay After SYS_CLKOUT[1] | | 7 | | 7 | ns |
| $t_{HO}$ | Output Hold After SYS_CLKOUT [1] | –2.5 | | –2.5 | | ns |

[1] Output signals are SMC0_Ax, $\overline{\text{SMC0_AMSx}}$, $\overline{\text{SMC0_AOE}}$, and $\overline{\text{SMC0_ABEx}}$.

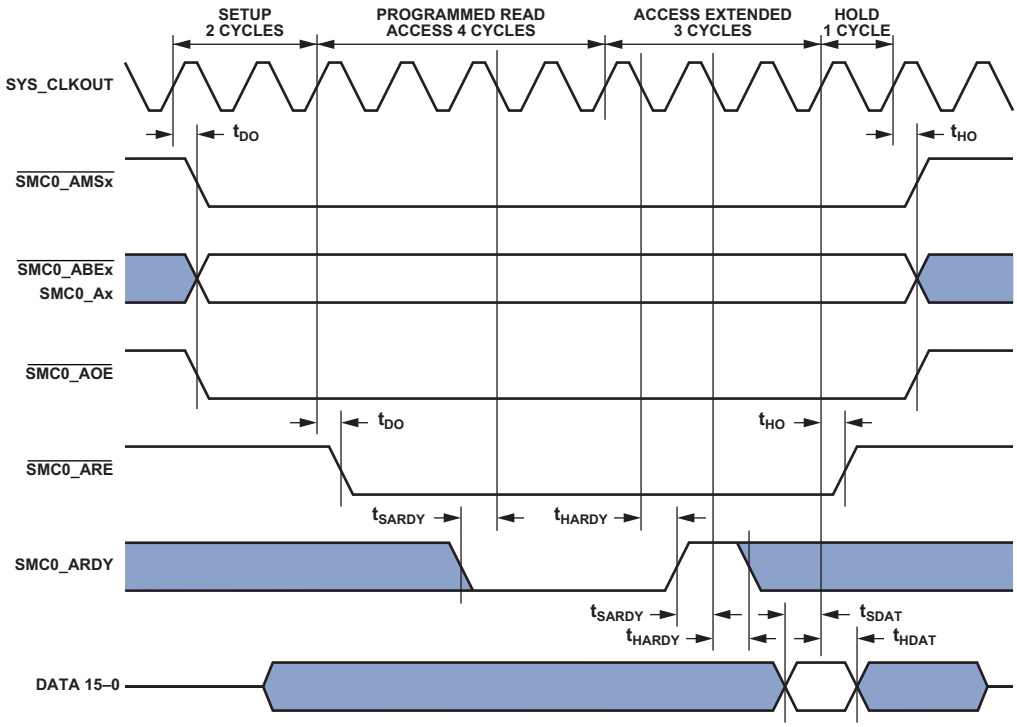



*Figure 11. Asynchronous Memory Read Cycle Timing*

### *Asynchronous Flash Read*

Table 33 and Figure 12 show asynchronous flash memory read timing, related to the static memory controller (SMC).

**Table 33. Asynchronous Flash Read**

| Parameter | | $V_{DD_EXT}$ 1.8 V/3.3 V Nominal Min | Max | Unit |
|---|---|---|---|---|
| *Switching Characteristics* | | | | |
| $t_{AMSADV}$ | SMC0_Ax (Address)/$\overline{\text{SMC0_AMSx}}$ Assertion Before SMC0_NORDV Low[1] | PREST × $t_{SCLK0}$ – 2 | | ns |
| $t_{WADV}$ | SMC0_NORDV Active Low Width[2] | RST × $t_{SCLK0}$ – 2 | | ns |
| $t_{DADVARE}$ | $\overline{\text{SMC0_ARE}}$ Low Delay From SMC0_NORDV High[3] | PREAT × $t_{SCLK0}$ – 2 | | ns |
| $t_{HARE}$ | Output[4] Hold After $\overline{\text{SMC0_ARE}}$ High[5] | RHT × $t_{SCLK0}$ – 2 | | ns |
| $t_{WARE}$[6] | $\overline{\text{SMC0_ARE}}$ Active Low Width[7] | RAT × $t_{SCLK0}$ – 2 | | ns |

[1] PREST value set using the SMC_BxETIM.PREST bits.
[2] RST value set using the SMC_BxTIM.RST bits.
[3] PREAT value set using the SMC_BxETIM.PREAT bits.
[4] Output signals are SMC0_Ax, $\overline{\text{SMC0_AMS}}$, $\overline{\text{SMC0_AOE}}$.
[5] RHT value set using the SMC_BxTIM.RHT bits.
[6] SMC0_BxCTL.ARDYEN bit = 0.
[7] RAT value set using the SMC_BxTIM.RAT bits.

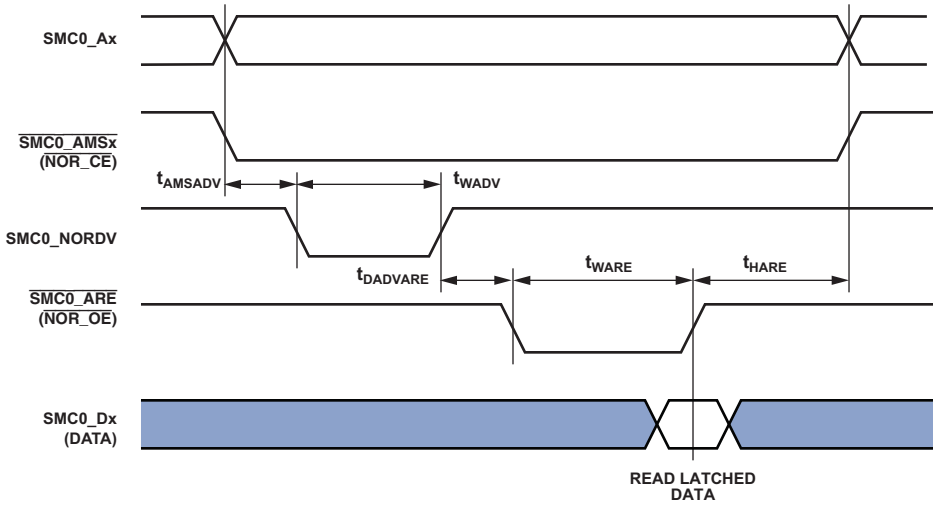



*Figure 12. Asynchronous Flash Read*

### *Asynchronous Page Mode Read*

Table 34 and Figure 13 show asynchronous memory page mode read timing, related to the static memory controller (SMC).

**Table 34. Asynchronous Page Mode Read**

| Parameter | | $V_{DD_EXT}$ 1.8 V /3.3 V Nominal Min | Max | Unit |
|---|---|---|---|---|
| *Switching Characteristics* | | | | |
| $t_{AV}$ | SMC0_Ax (Address) Valid for First Address Min Width[1] | (PREST + RST + PREAT + RAT) × $t_{SCLK0}$ – 2 | | ns |
| $t_{AV1}$ | SMC0_Ax (Address) Valid for Subsequent SMC0_Ax (Address) Min Width | PGWS × $t_{SCLK0}$ – 2 | | ns |
| $t_{WADV}$ | SMC0_NORDV Active Low Width[2] | RST × $t_{SCLK0}$ – 2 | | ns |
| $t_{HARE}$ | Output[3] Hold After $\overline{\text{SMC0_ARE}}$ High[4] | RHT × $t_{SCLK0}$ – 2 | | ns |
| $t_{WARE}$[5] | $\overline{\text{SMC0_ARE}}$ Active Low Width[6] | (RAT + (Nw – 1) × PGWS) × $t_{SCLK0}$ – 2 | | ns |

[1] PREST, RST, PREAT and RAT values set using the SMC_BxETIM.PREST bits, SMC_BxTIM.RST bits, SMC_BxETIM.PREAT bits, and the SMC_BxTIM.RAT bits.
[2] RST value set using the SMC_BxTIM.RST bits.
[3] Output signals are SMC0_Ax, $\overline{\text{SMC0_AMSx}}$, $\overline{\text{SMC0_AOE}}$.
[4] RHT value set using the SMC_BxTIM.RHT bits.
[5] SMC_BxCTL.ARDYEN bit = 0.
[6] RAT value set using the SMC_BxTIM.RAT bits.

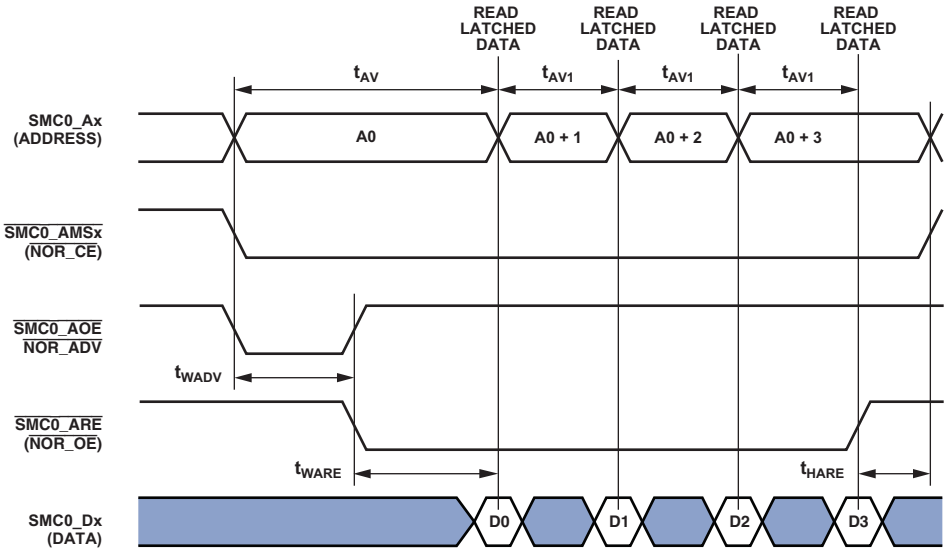



*Figure 13. Asynchronous Page Mode Read*

### *Asynchronous Write*

Table 35 and Figure 14 show asynchronous memory write timing, related to the static memory controller (SMC).

**Table 35. Asynchronous Memory Write (BxMODE = b#00)**

| Parameter | | $V_{DD_EXT}$ 1.8 V Nominal Min | $V_{DD_EXT}$ 1.8 V Nominal Max | $V_{DD_EXT}$ 3.3 V Nominal Min | $V_{DD_EXT}$ 3.3 V Nominal Max | Unit |
|---|---|---|---|---|---|---|
| *Timing Requirement* | | | | | | |
| $t_{DARDYAWE}$[1] | SMC0_ARDY Valid After $\overline{SMC0_AWE}$ Low[2] | | (WAT – 2.5) × $t_{SCLK0}$ – 17.5 | | (WAT – 2.5) × $t_{SCLK0}$ – 17.5 | ns |
| *Switching Characteristics* | | | | | | |
| $t_{ENDAT}$ | DATA Enable After $\overline{SMC0_AMSx}$ Assertion | –3 | | –2 | | ns |
| $t_{DDAT}$ | DATA Disable After $\overline{SMC0_AMSx}$ Deassertion | | 4.5 | | 4 | ns |
| $t_{AMSAWE}$ | SMC0_Ax/$\overline{SMC0_AMSx}$ Assertion Before $\overline{SMC0_AWE}$ Low[3] | (PREST + WST + PREAT) × $t_{SCLK0}$ – 2 | | (PREST + WST + PREAT) × $t_{SCLK0}$ – 4 | | ns |
| $t_{HAWE}$ | Output[4] Hold After $\overline{SMC0_AWE}$ High[5] | WHT × $t_{SCLK0}$ | | WHT × $t_{SCLK0}$ | | ns |
| $t_{WAWE}$[6] | $\overline{SMC0_AWE}$ Active Low Width[6] | WAT × $t_{SCLK0}$ – 2 | | WAT × $t_{SCLK0}$ – 2 | | ns |
| $t_{DAWEARDY}$[1] | $\overline{SMC0_AWE}$ High Delay After SMC0_ARDY Assertion | | 3.5 × $t_{SCLK0}$ + 17.5 | | 3.5 × $t_{SCLK0}$ + 17.5 | ns |

[1] SMC_BxCTL.ARDYEN bit = 1.
[2] WAT value set using the SMC_BxTIM.WAT bits.
[3] PREST, WST, PREAT values set using the SMC_BxETIM.PREST bits, SMC_BxTIM.WST bits, SMC_BxETIM.PREAT bits, and the SMC_BxTIM.RAT bits.
[4] Output signals are DATA, SMC0_Ax, $\overline{SMC0_AMSx}$, $\overline{SMC0_ABEx}$.
[5] WHT value set using the SMC_BxTIM.WHT bits.
[6] SMC_BxCTL.ARDYEN bit = 0.

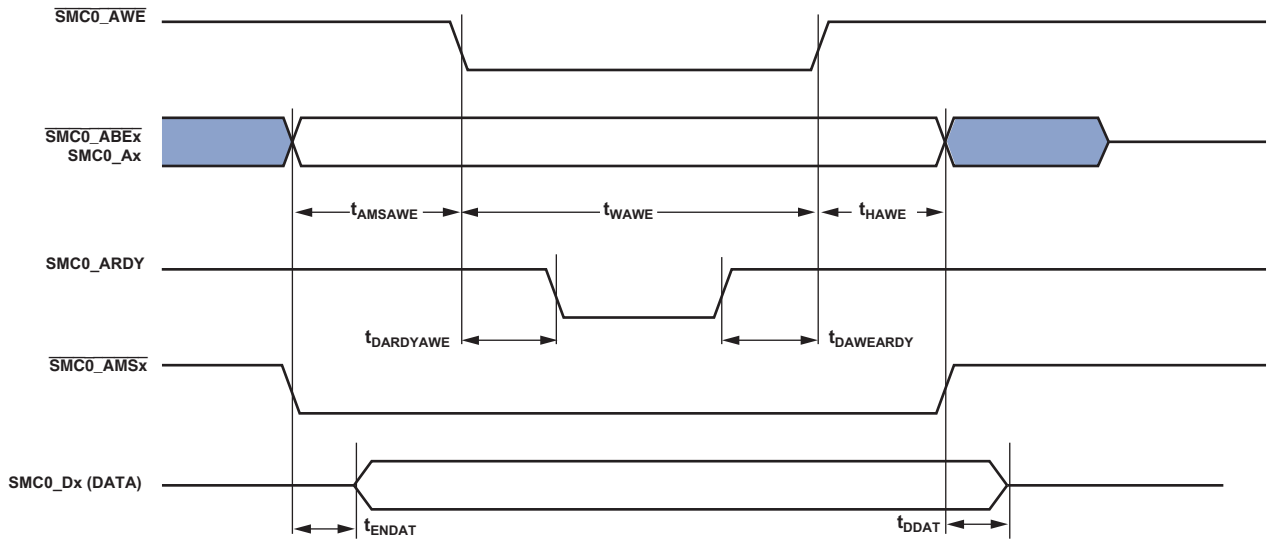



*Figure 14. Asynchronous Write*

### *SMC Write Cycle Timing With Reference to SYS_CLKOUT*

The following SMC specifications with respect to SYS_CLKOUT are given to accommodate the connection of the SMC to programmable logic devices. These specifications assume that SYS_CLKOUT is outputting a buffered version of SCLK0 by setting CGU_CLKOUTSEL.CLKOUTSEL = 0x3. However, SCLK0 must not run faster than the maximum $f_{OCLK}$ specification. For this example WST = 0x2, WAT = 0x2, and WHT = 0x1.

**Table 36. SMC Write Cycle Timing With Reference to SYS_CLKOUT (BxMODE = b#00)**

| Parameter | | $V_{DD_EXT}$ 1.8 V/3.3 V Nominal Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SARDY}$ | SMC0_ARDY Setup Before SYS_CLKOUT | 14.4 | | ns |
| $t_{HARDY}$ | SMC0_ARDY Hold After SYS_CLKOUT | 0.7 | | ns |
| *Switching Characteristics* | | | | |
| $t_{DDAT}$ | SMC0_Dx Disable After SYS_CLKOUT | | 7 | ns |
| $t_{ENDAT}$ | SMC0_Dx Enable After SYS_CLKOUT | –2.5 | | ns |
| $t_{DO}$ | Output Delay After SYS_CLKOUT[1] | | 7 | ns |
| $t_{HO}$ | Output Hold After SYS_CLKOUT [1] | –2.5 | | ns |

[1] Output pins/balls include $\overline{\text{SMC0_AMSx}}$, $\overline{\text{SMC0_ABEx}}$, SMC0_Ax, SMC0_Dx, $\overline{\text{SMC0_AOE}}$, and $\overline{\text{SMC0_AWE}}$.

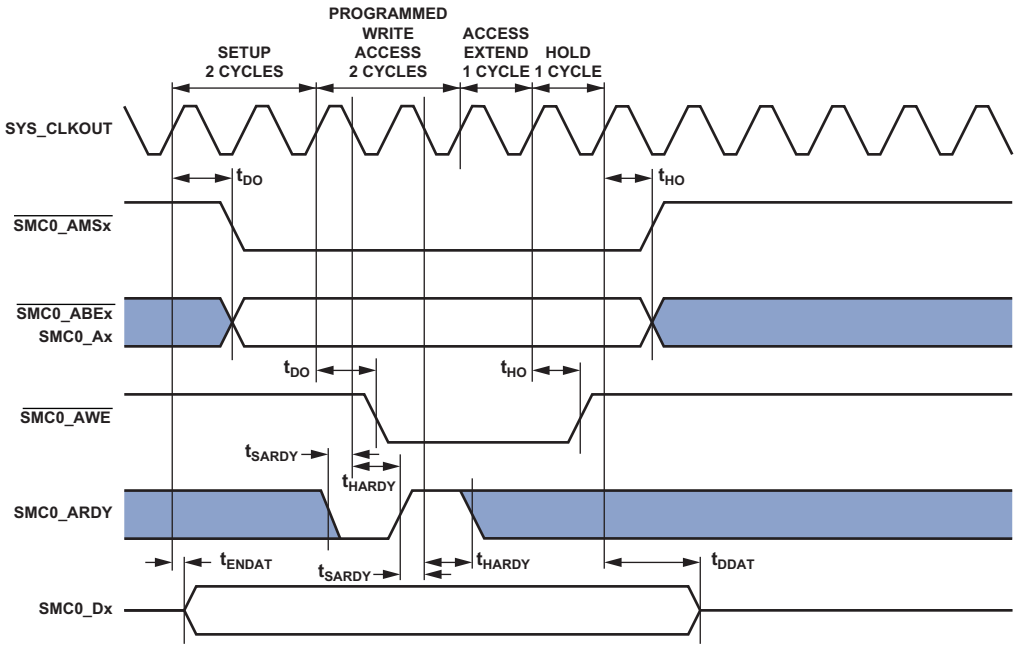



*Figure 15. SMC Write Cycle Timing With Reference to SYS_CLKOUT Timing*

### *Asynchronous Flash Write*

Table 37 and Figure 16 show asynchronous flash memory write timing, related to the static memory controller (SMC).

**Table 37. Asynchronous Flash Write**

| Parameter | | $V_{DD_EXT}$ 1.8 V/3.3 V Nominal Min | Max | Unit |
|---|---|---|---|---|
| *Switching Characteristics* | | | | |
| $t_{AMSADV}$ | SMC0_Ax/$\overline{\text{SMC0_AMSx}}$ Assertion Before ADV Low[1] | PREST × $t_{SCLK0}$ – 2 | | ns |
| $t_{DADVAWE}$ | $\overline{\text{SMC0_AWE}}$ Low Delay From ADV High[2] | PREAT × $t_{SCLK0}$ – 4 | | ns |
| $t_{WADV}$ | NR_ADV Active Low Width[3] | WST × $t_{SCLK0}$ – 2 | | ns |
| $t_{HAWE}$ | Output[4] Hold After $\overline{\text{SMC0_AWE}}$ High[5] | WHT × $t_{SCLK0}$ | | ns |
| $t_{WAWE}$[6] | $\overline{\text{SMC0_AWE}}$ Active Low Width[7] | WAT × $t_{SCLK0}$ – 2 | | ns |

[1] PREST value set using the SMC_BxETIM.PREST bits.
[2] PREAT value set using the SMC_BxETIM.PREAT bits.
[3] WST value set using the SMC_BxTIM.WST bits.
[4] Output signals are DATA, SMC0_Ax, $\overline{\text{SMC0_AMSx}}$, $\overline{\text{SMC0_ABEx}}$.
[5] WHT value set using the SMC_BxTIM.WHT bits.
[6] SMC_BxCTL.ARDYEN bit = 0.
[7] WAT value set using the SMC_BxTIM.WAT bits.

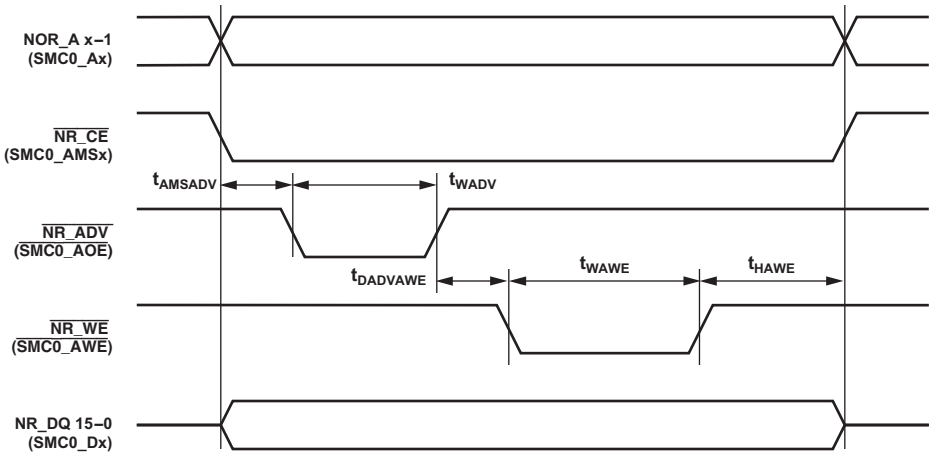



*Figure 16. Asynchronous Flash Write*

### *All Accesses*

Table 38 describes timing that applies to all memory accesses, related to the static memory controller (SMC).

**Table 38. All Accesses**

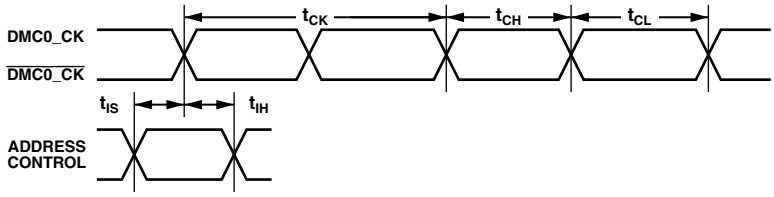
| Parameter | | $V_{DD_EXT}$ 1.8 V Nominal Min | Max | $V_{DD_EXT}$ 3.3 V Nominal Min | Max | Unit |
|---|---|---|---|---|---|---|
| *Switching Characteristic* | | | | | | |
| $t_{TURN}$ | $\overline{\text{SMC0_AMSx}}$ Inactive Width | (IT + TT) × $t_{SCLK0}$ – 2 | | (IT + TT) × $t_{SCLK0}$ – 2 | | ns |

### *DDR2 SDRAM Clock and Control Cycle Timing*

Table 39 and Figure 17 show DDR2 SDRAM clock and control cycle timing, related to the dynamic memory controller (DMC).

**Table 39. DDR2 SDRAM Read Cycle Timing, $V_{DD_DMC}$ Nominal 1.8 V**

| Parameter | | 200 MHz Min | 200 MHz Max | Unit |
|---|---|---|---|---|
| *Switching Characteristics* | | | | |
| $t_{CK}$ | Clock Cycle Time (CL = 2 Not Supported) | 5 | | ns |
| $t_{CH}$ | High Clock Pulse Width | 0.45 | 0.55 | $t_{CK}$ |
| $t_{CL}$ | Low Clock Pulse Width | 0.45 | 0.55 | $t_{CK}$ |
| $t_{IS}$ | Control/Address Setup Relative to DMC0_CK Rise | 350 | | ps |
| $t_{IH}$ | Control/Address Hold Relative to DMC0_CK Rise | 475 | | ps |




NOTE: CONTROL = $\overline{\text{DMC0_CS0}}$, DMC0_CKE, $\overline{\text{DMC0_RAS}}$, $\overline{\text{DMC0_CAS}}$, AND $\overline{\text{DMC0_WE}}$.
ADDRESS = DMC0_A00–13, AND DMC0_BA0–2.

*Figure 17. DDR2 SDRAM Clock and Control Cycle Timing*

### *DDR2 SDRAM Read Cycle Timing*

Table 40 and Figure 18 show DDR2 SDRAM read cycle timing, related to the dynamic memory controller (DMC).

**Table 40. DDR2 SDRAM Read Cycle Timing, $V_{DD_DMC}$ Nominal 1.8 V**

| Parameter | | 200 MHz[1] Min | 200 MHz[1] Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{DQSQ}$ | DMC0_DQS-DMC0_DQ Skew for DMC0_DQS and Associated DMC0_DQ Signals | | 0.35 | ns |
| $t_{QH}$ | DMC0_DQ, DMC0_DQS Output Hold Time From DMC0_DQS | 1.8 | | ns |
| $t_{RPRE}$ | Read Preamble | 0.9 | | $t_{CK}$ |
| $t_{RPST}$ | Read Postamble | 0.4 | | $t_{CK}$ |

[1] To ensure proper operation of the DDR2, all the DDR2 guidelines have to be strictly followed.

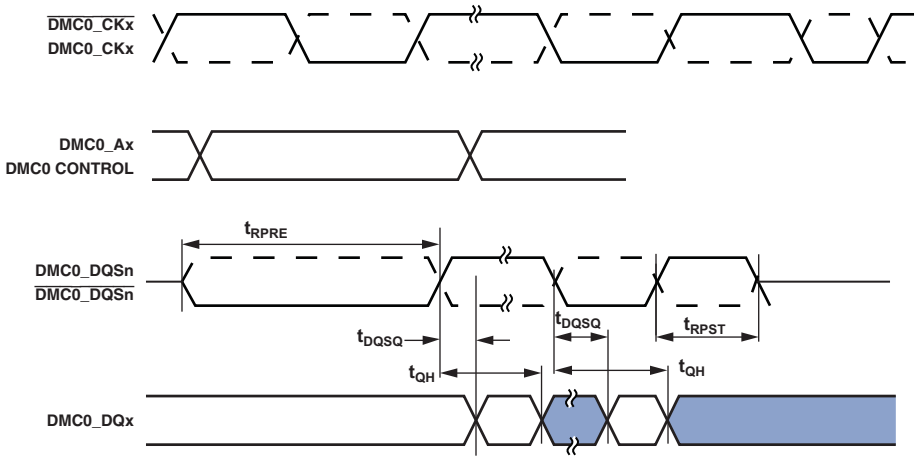



*Figure 18. DDR2 SDRAM Controller Input AC Timing*

### *DDR2 SDRAM Write Cycle Timing*

Table 41 and Figure 19 show DDR2 SDRAM write cycle timing, related to the dynamic memory controller (DMC).

**Table 41. DDR2 SDRAM Write Cycle Timing, $V_{DD_DMC}$ Nominal 1.8 V**

| Parameter | | 200 MHz[1] Min | 200 MHz[1] Max | Unit |
|---|---|---|---|---|
| *Switching Characteristics* | | | | |
| $t_{DQSS}$[2] | DMC0_DQS Latching Rising Transitions to Associated Clock Edges | –0.25 | +0.25 | $t_{CK}$ |
| $t_{DS}$ | Last Data Valid to DMC0_DQS Delay | 0.15 | | ns |
| $t_{DH}$ | DMC0_DQS to First Data Invalid Delay | 0.275 | | ns |
| $t_{DSS}$ | DMC0_DQS Falling Edge to Clock Setup Time | 0.2 | | $t_{CK}$ |
| $t_{DSH}$ | DMC0_DQS Falling Edge Hold Time From DMC0_CK | 0.2 | | $t_{CK}$ |
| $t_{DQSH}$ | DMC0_DQS Output High Pulse Width | 0.35 | | $t_{CK}$ |
| $t_{DQSL}$ | DMC0_DQS Output Low Pulse Width | 0.35 | | $t_{CK}$ |
| $t_{WPRE}$ | Write Preamble | 0.35 | | $t_{CK}$ |
| $t_{WPST}$ | Write Postamble | 0.4 | | $t_{CK}$ |
| $t_{IPW}$ | Address and Control Output Pulse Width | 0.6 | | $t_{CK}$ |
| $t_{DIPW}$ | DMC0_DQ and DMC0_DM Output Pulse Width | 0.35 | | $t_{CK}$ |

[1] To ensure proper operation of the DDR2, all the DDR2 guidelines have to be strictly followed.
[2] Write command to first DMC0_DQS delay = WL × $t_{CK}$ + $t_{DQSS}$.

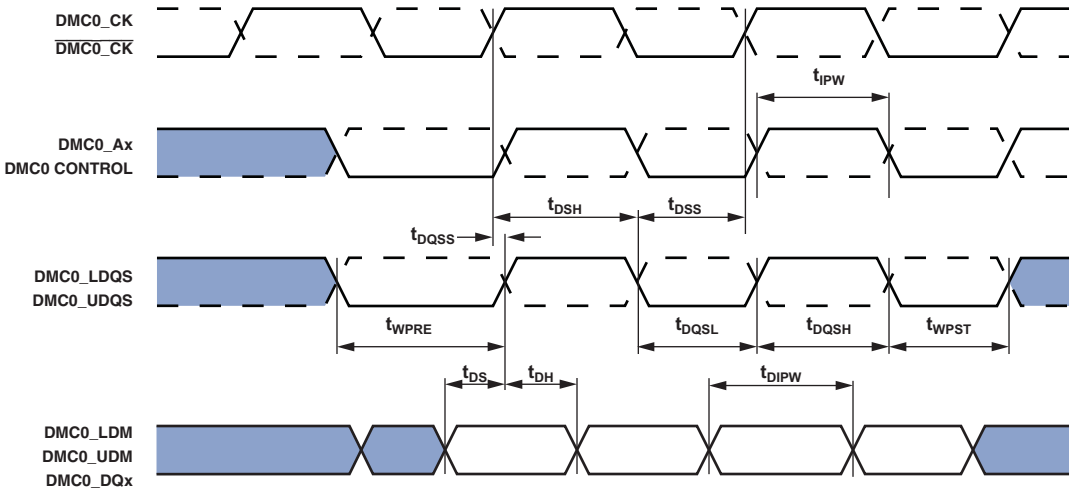


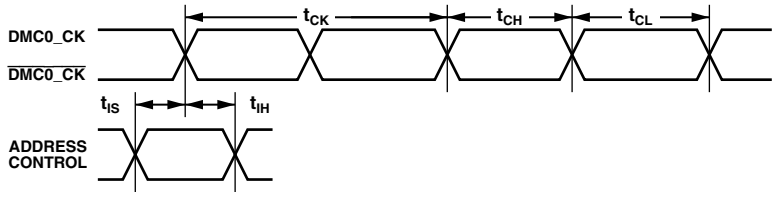

*Figure 19. DDR2 SDRAM Controller Output AC Timing*

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

Table 42 and Figure 20 show mobile DDR SDRAM clock and control cycle timing, related to the dynamic memory controller (DMC).

**Table 42. Mobile DDR SDRAM Clock and Control Cycle Timing, $V_{DD_DMC}$ Nominal 1.8 V**

| Parameter | | 200 MHz Min | 200 MHz Max | Unit |
|---|---|---|---|---|
| *Switching Characteristics* | | | | |
| $t_{CK}$ | Clock Cycle Time (CL = 2 Not Supported) | 5 | | ns |
| $t_{CH}$ | Minimum Clock Pulse Width | 0.45 | 0.55 | $t_{CK}$ |
| $t_{CL}$ | Maximum Clock Pulse Width | 0.45 | 0.55 | $t_{CK}$ |
| $t_{IS}$ | Control/Address Setup Relative to DMC0_CK Rise | 1.5 | | ns |
| $t_{IH}$ | Control/Address Hold Relative to DMC0_CK Rise | 1.5 | | ns |




NOTE: CONTROL = $\overline{\text{DMC0_CS0}}$, DMC0_CKE, $\overline{\text{DMC0_RAS}}$, $\overline{\text{DMC0_CAS}}$, AND $\overline{\text{DMC0_WE}}$.
ADDRESS = DMC0_A00–13, AND DMC0_BA0–2.

*Figure 20. Mobile DDR SDRAM Clock and Control Cycle Timing*

### *Mobile DDR SDRAM Read Cycle Timing*

Table 43 and Figure 21 show mobile DDR SDRAM read cycle timing, related to the dynamic memory controller (DMC).

**Table 43. Mobile DDR SDRAM Read Cycle Timing, $V_{DD_DMC}$ Nominal 1.8 V**

| Parameter | | 200 MHz | | Unit |
|---|---|---|---|---|
| | | Min | Max | |
| *Timing Requirements* | | | | |
| $t_{QH}$ | DMC0_DQ, DMC0_DQS Output Hold Time From DMC0_DQS | 1.5 | | ns |
| $t_{DQSQ}$ | DMC0_DQS-DMC0_DQ Skew for DMC0_DQS and Associated DMC0_DQ Signals | | 0.7 | ns |
| $t_{RPRE}$ | Read Preamble | 0.9 | 1.1 | $t_{CK}$ |
| $t_{RPST}$ | Read Postamble | 0.4 | 0.6 | $t_{CK}$ |

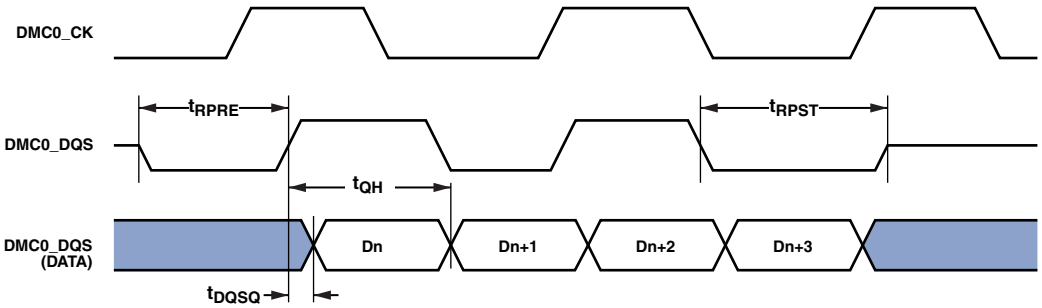



*Figure 21. Mobile DDR SDRAM Controller Input AC Timing*

### *Mobile DDR SDRAM Write Cycle Timing*

Table 44 and Figure 22 show mobile DDR SDRAM write cycle timing, related to the dynamic memory controller (DMC).

**Table 44. Mobile DDR SDRAM Write Cycle Timing, $V_{DD_DMC}$ Nominal 1.8 V**

| Parameter | | 200 MHz Min | 200 MHz Max | Unit |
|---|---|---|---|---|
| *Switching Characteristics* | | | | |
| $t_{DQSS}$[1] | DMC0_DQS Latching Rising Transitions to Associated Clock Edges | 0.75 | 1.25 | $t_{CK}$ |
| $t_{DS}$ | Last Data Valid to DMC0_DQS Delay (Slew > 1 V/ns) | 0.48 | | ns |
| $t_{DH}$ | DMC0_DQS to First Data Invalid Delay (Slew > 1 V/ns) | 0.48 | | ns |
| $t_{DSS}$ | DMC0_DQS Falling Edge to Clock Setup Time | 0.2 | | $t_{CK}$ |
| $t_{DSH}$ | DMC0_DQS Falling Edge Hold Time From DMC0_CK | 0.2 | | $t_{CK}$ |
| $t_{DQSH}$ | DMC0_DQS Input High Pulse Width | 0.4 | | $t_{CK}$ |
| $t_{DQSL}$ | DMC0_DQS Input Low Pulse Width | 0.4 | | $t_{CK}$ |
| $t_{WPRE}$ | Write Preamble | 0.25 | | $t_{CK}$ |
| $t_{WPST}$ | Write Postamble | 0.4 | | $t_{CK}$ |
| $t_{IPW}$ | Address and Control Output Pulse Width | 2.3 | | ns |
| $t_{DIPW}$ | DMC0_DQ and DMC0_DM Output Pulse Width | 1.8 | | ns |

[1] Write command to first DMC0_DQS delay = WL × $t_{CK}$ + $t_{DQSS}$.

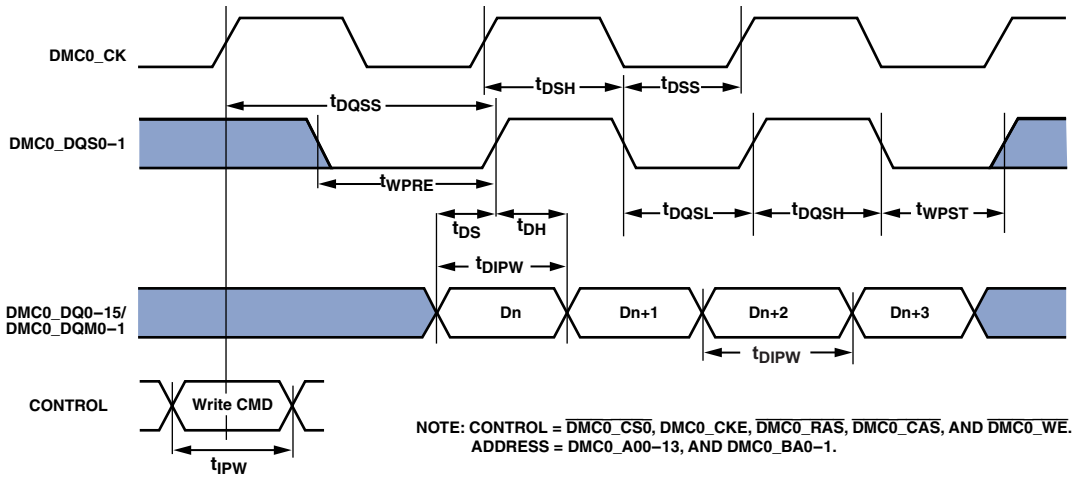



*Figure 22. Mobile DDR SDRAM Controller Output AC Timing*

### General-Purpose I/O Port Timing (GPIO)

Table 45 and Figure 23 describe I/O timing, related to the general-purpose ports (PORT).

**Table 45. General-Purpose I/O Port Timing**

| Parameter | | $V_{DD_EXT}$ 1.8 V/3.3 V Nominal Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirement* | | | | |
| $t_{WFI}$ | General-Purpose Port Pin Input Pulse Width | $2 \times t_{SCLK0} - 1.5$ | | ns |

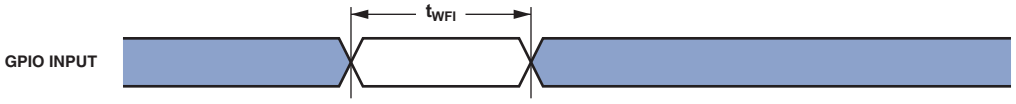



*Figure 23. General-Purpose I/O Port Timing*

### Timer Cycle Timing

Table 46 and Figure 24 describe timer expired operations, related to the general-purpose timer (TIMER). The input signal is asynchronous in width capture mode and external clock mode and has an ideal maximum input frequency of ($f_{SCLK0}$/4) MHz. The Period Value (VALUE) is the timer period assigned in the TMx_TMRn_PER register and can range from 2 to $2^{32} - 1$.

**Table 46. Timer Cycle Timing**

| Parameter | | $V_{DD_EXT}$ 1.8 V Nominal Min | Max | $V_{DD_EXT}$ 3.3 V Nominal Min | Max | Unit |
|---|---|---|---|---|---|---|
| *Timing Requirements* | | | | | | |
| $t_{WL}$ | Timer Pulse Width Input Low[1] | $2 \times t_{SCLK0} - 1.5$ | | $2 \times t_{SCLK0} - 1.5$ | | ns |
| $t_{WH}$ | Timer Pulse Width Input High[1] | $2 \times t_{SCLK0} - 1.5$ | | $2 \times t_{SCLK0} - 1.5$ | | ns |
| *Switching Characteristic* | | | | | | |
| $t_{HTO}$ | Timer Pulse Width Output | $t_{SCLK0} \times VALUE - 1$ | | $t_{SCLK0} \times VALUE - 1$ | | ns |

[1] This specification indicates the minimum instantaneous width that can be tolerated due to duty cycle variation or jitter for TMx signals in width capture and external clock modes. The ideal maximum frequency for TMx signals is listed in Timer Cycle Timing on this page.

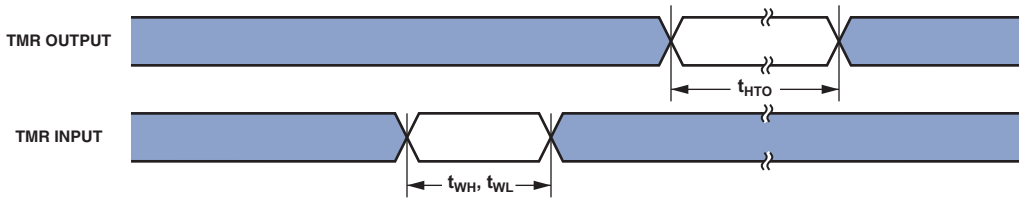



*Figure 24. Timer Cycle Timing*

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

Table 47 and Figure 25 describe timing, related to the general-purpose counter (CNT).

**Table 47. Up/Down Counter/Rotary Encoder Timing**

| Parameter | | $V_{DD_EXT}$ 1.8 V Nominal Min | $V_{DD_EXT}$ 1.8 V Nominal Max | $V_{DD_EXT}$ 3.3 V Nominal Min | $V_{DD_EXT}$ 3.3 V Nominal Max | Unit |
|---|---|---|---|---|---|---|
| *Timing Requirement* | | | | | | |
| $t_{WCOUNT}$ | Up/Down Counter/Rotary Encoder Input Pulse Width | $2 \times t_{SCLK0}$ | | $2 \times t_{SCLK0}$ | | ns |

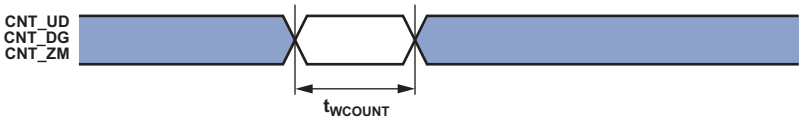



*Figure 25. Up/Down Counter/Rotary Encoder Timing*

### *Debug Interface (JTAG Emulation Port) Timing*

Table 48 and Figure 26 provide I/O timing, related to the debug interface (JTAG emulator port).

**Table 48. JTAG Port Timing**

| Parameter | | $V_{DD_EXT}$ 1.8 V Nominal Min | $V_{DD_EXT}$ 1.8 V Nominal Max | $V_{DD_EXT}$ 3.3 V Nominal Min | $V_{DD_EXT}$ 3.3 V Nominal Max | Unit |
|---|---|---|---|---|---|---|
| *Timing Requirements* | | | | | | |
| $t_{TCK}$ | JTG_TCK Period | 20 | | 20 | | ns |
| $t_{STAP}$ | JTG_TDI, JTG_TMS Setup Before JTG_TCK High | 5 | | 4 | | ns |
| $t_{HTAP}$ | JTG_TDI, JTG_TMS Hold After JTG_TCK High | 4 | | 4 | | ns |
| $t_{SSYS}$ | System Inputs Setup Before JTG_TCK High[1] | 4 | | 4 | | ns |
| $t_{HSYS}$ | System Inputs Hold After JTG_TCK High[1] | 4 | | 4 | | ns |
| $t_{TRSTW}$ | $\overline{\text{JTG_TRST}}$ Pulse Width (Measured in JTG_TCK Cycles)[2] | 4 | | 4 | | $t_{TCK}$ |
| *Switching Characteristics* | | | | | | |
| $t_{DTDO}$ | JTG_TDO Delay From JTG_TCK Low | | 16.5 | | 14.5 | ns |
| $t_{DSYS}$ | System Outputs Delay After JTG_TCK Low[3] | | 18 | | 16.5 | ns |
| $t_{DTMS}$ | TMS Delay After TCK High in SWD Mode | 3.5 | 16.5 | 3.5 | 14.5 | ns |

[1] System inputs = DMC0_DQxx, DMC0_LDQS, $\overline{\text{DMC0_LDQS}}$, DMC0_UDQS, $\overline{\text{DMC0_UDQS}}$, PA_xx, PB_xx, PC_xx, SYS_BMODEx, $\overline{\text{SYS_HWRST}}$, $\overline{\text{SYS_FAULT}}$, $\overline{\text{SYS_NMI}}$, TWI0_SCL, TWI0_SDA, and SYS_EXTWAKE.

[2] 50 MHz maximum.

[3] System outputs = DMC0_Axx, DMC0_BAx, $\overline{\text{DMC0_CAS}}$, DMC0_CK, $\overline{\text{DMC0_CK}}$, DMC0_CKE, $\overline{\text{DMC0_CS0}}$, DMC0_DQxx, DMC0_LDM, DMC0_LDQS, $\overline{\text{DMC0_LDQS}}$, DMC0_ODT, $\overline{\text{DMC0_RAS}}$, DMC0_UDM, DMC0_UDQS, $\overline{\text{DMC0_UDQS}}$, $\overline{\text{DMC0_WE}}$, PA_xx, PB_xx, PC_xx, SYS_CLKOUT, $\overline{\text{SYS_FAULT}}$, $\overline{\text{SYS_RESOUT}}$, and $\overline{\text{SYS_NMI}}$.

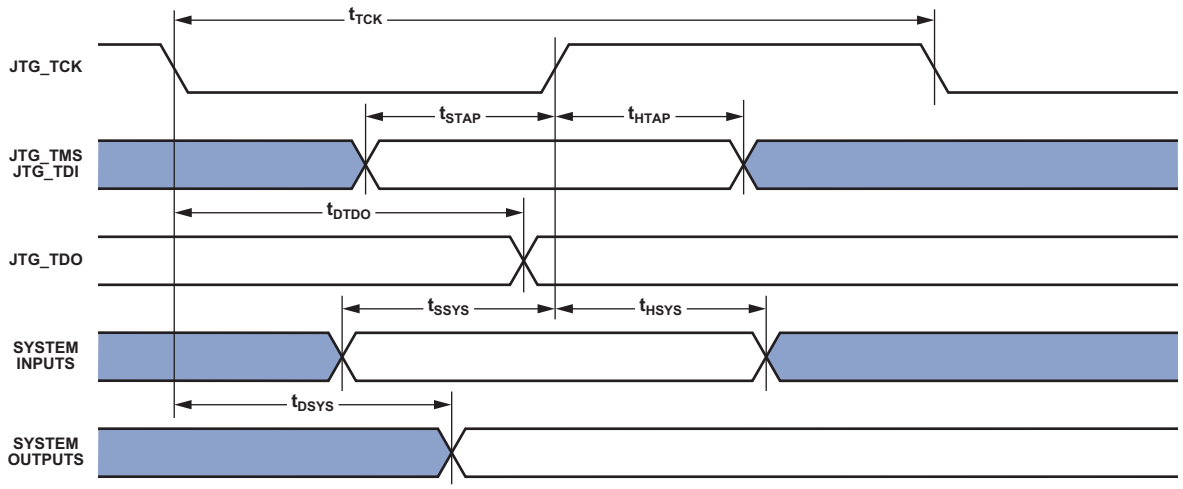



*Figure 26. JTAG Port Timing*

### Serial Ports

To determine whether serial port (SPORT) communication is possible between two devices at clock speed n, the following specifications must be confirmed: 1) frame sync delay and frame sync setup and hold, 2) data delay and data setup and hold, and 3) serial clock (SPT_CLK) width. In Figure 27 either the rising edge or the falling edge of SPT_CLK (external or internal) can be used as the active sampling edge.

When externally generated the SPORT clock is called $f_{SPTCLKEXT}$:

$$t_{SPTCLKEXT} = \frac{1}{f_{SPTCLKEXT}}$$

When internally generated, the programmed SPORT clock ($f_{SPTCLKPROG}$) frequency in MHz is set by the following equation where CLKDIV is a field in the SPORT_DIV register that can be set from 0 to 65,535:

$$f_{SPTCLKPROG} = \frac{f_{SCLK0}}{(CLKDIV + 1)}$$

$$t_{SPTCLKPROG} = \frac{1}{f_{SPTCLKPROG}}$$

**Table 49. Serial Ports—External Clock**

| Parameter | | $V_{DD_EXT}$ 1.8 V Nominal Min | $V_{DD_EXT}$ 1.8 V Nominal Max | $V_{DD_EXT}$ 3.3 V Nominal Min | $V_{DD_EXT}$ 3.3 V Nominal Max | Unit |
|---|---|---|---|---|---|---|
| *Timing Requirements* | | | | | | |
| $t_{SFSE}$ | Frame Sync Setup Before SPT_CLK (Externally Generated Frame Sync in Either Transmit or Receive Mode)[1] | 1.5 | | 1 | | ns |
| $t_{HFSE}$ | Frame Sync Hold After SPT_CLK (Externally Generated Frame Sync in Either Transmit or Receive Mode)[1] | 3 | | 3 | | ns |
| $t_{SDRE}$ | Receive Data Setup Before Receive SPT_CLK[1] | 1.5 | | 1 | | ns |
| $t_{HDRE}$ | Receive Data Hold After SPT_CLK[1] | 3 | | 3 | | ns |
| $t_{SCLKW}$ | SPT_CLK Width[2] | $(0.5 \times t_{SPTCLKEXT}) - 1$ | | $(0.5 \times t_{SPTCLKEXT}) - 1$ | | ns |
| $t_{SPTCLKE}$ | SPT_CLK Period[2] | $t_{SPTCLKEXT} - 1$ | | $t_{SPTCLKEXT} - 1$ | | ns |
| *Switching Characteristics* | | | | | | |
| $t_{DFSE}$ | Frame Sync Delay After SPT_CLK (Internally Generated Frame Sync in Either Transmit or Receive Mode)[3] | | 18 | | 15 | ns |
| $t_{HOFSE}$ | Frame Sync Hold After SPT_CLK (Internally Generated Frame Sync in Either Transmit or Receive Mode)[3] | 2.5 | | 2.5 | | ns |
| $t_{DDTE}$ | Transmit Data Delay After Transmit SPT_CLK[3] | | 18 | | 15 | ns |
| $t_{HDTE}$ | Transmit Data Hold After Transmit SPT_CLK[3] | 2.5 | | 2.5 | | ns |

[1] Referenced to sample edge.

[2] This specification indicates the minimum instantaneous width or period that can be tolerated due to duty cycle variation or jitter on the external SPT_CLK. For the external SPT_CLK ideal maximum frequency, see the $f_{SPTCLKEXT}$ specification in Table 18 on Page 52 in Clock Related Operating Conditions.

[3] Referenced to drive edge.

**Table 50. Serial Ports—Internal Clock**

| Parameter | | $V_{DD_EXT}$ 1.8 V Nominal Min | $V_{DD_EXT}$ 1.8 V Nominal Max | $V_{DD_EXT}$ 3.3 V Nominal Min | $V_{DD_EXT}$ 3.3 V Nominal Max | Unit |
|---|---|---|---|---|---|---|
| *Timing Requirements* | | | | | | |
| $t_{SFSI}$ | Frame Sync Setup Before SPT_CLK (Externally Generated Frame Sync in Either Transmit or Receive Mode)[1] | 17 | | 14.5 | | ns |
| $t_{HFSI}$ | Frame Sync Hold After SPT_CLK (Externally Generated Frame Sync in Either Transmit or Receive Mode)[1] | –0.5 | | –0.5 | | ns |
| $t_{SDRI}$ | Receive Data Setup Before SPT_CLK[1] | 6.5 | | 5 | | ns |
| $t_{HDRI}$ | Receive Data Hold After SPT_CLK[1] | 1.5 | | 1 | | ns |
| *Switching Characteristics* | | | | | | |
| $t_{DFSI}$ | Frame Sync Delay After SPT_CLK (Internally Generated Frame Sync in Transmit or Receive Mode)[2] | | 2 | | 2 | ns |
| $t_{HOFSI}$ | Frame Sync Hold After SPT_CLK (Internally Generated Frame Sync in Transmit or Receive Mode)[2] | –4.5 | | –3.5 | | ns |
| $t_{DDTI}$ | Transmit Data Delay After SPT_CLK[2] | | 2 | | 2 | ns |
| $t_{HDTI}$ | Transmit Data Hold After SPT_CLK[2] | –5 | | –3.5 | | ns |
| $t_{SCLKIW}$ | SPT_CLK Width[3] | $0.5 \times t_{SPTCLKPROG} - 1.5$ | | $0.5 \times t_{SPTCLKPROG} - 1.5$ | | ns |
| $t_{SPTCLKI}$ | SPT_CLK Period[3] | $t_{SPTCLKPROG} - 1.5$ | | $t_{SPTCLKPROG} - 1.5$ | | ns |

[1] Referenced to the sample edge.
[2] Referenced to drive edge.
[3] See Table 18 on Page 52 in Clock Related Operating Conditions for details on the minimum period that may be programmed for $t_{SPTCLKPROG}$.

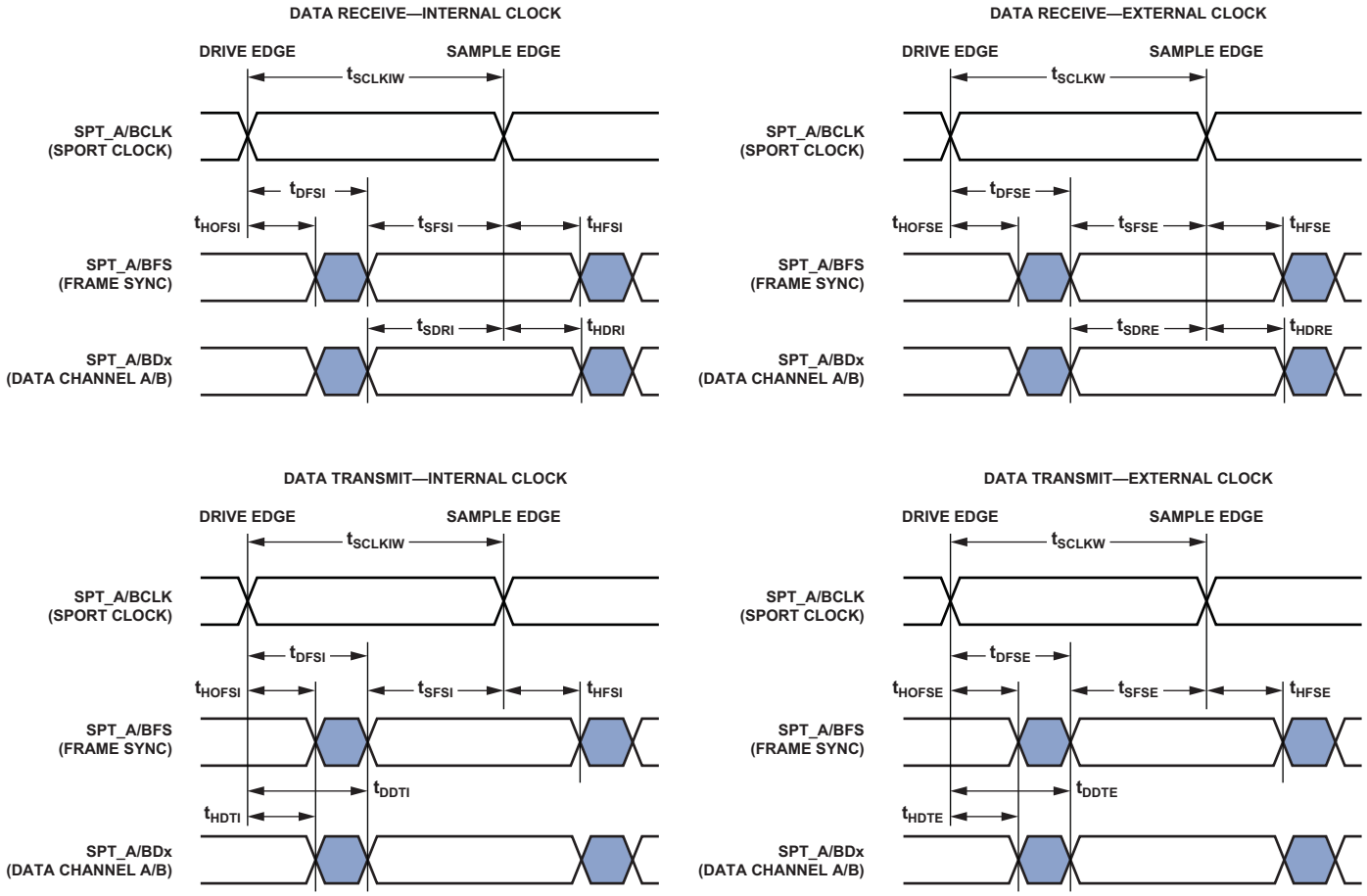



*Figure 27. Serial Ports*

**Table 51. Serial Ports—Enable and Three-State**

| Parameter | | $V_{DD_EXT}$ 1.8 V Nominal Min | $V_{DD_EXT}$ 1.8 V Nominal Max | $V_{DD_EXT}$ 3.3 V Nominal Min | $V_{DD_EXT}$ 3.3 V Nominal Max | Unit |
|---|---|---|---|---|---|---|
| *Switching Characteristics* | | | | | | |
| $t_{DDTEN}$ | Data Enable from External Transmit SPT_CLK[1] | 1 | | 1 | | ns |
| $t_{DDTTE}$ | Data Disable from External Transmit SPT_CLK[1] | | 14 | | 14 | ns |
| $t_{DDTIN}$ | Data Enable from Internal Transmit SPT_CLK[1] | –1.12 | | –1 | | ns |
| $t_{DDTTI}$ | Data Disable from Internal Transmit SPT_CLK[1] | | 2.8 | | 2.8 | ns |

[1] Referenced to drive edge.

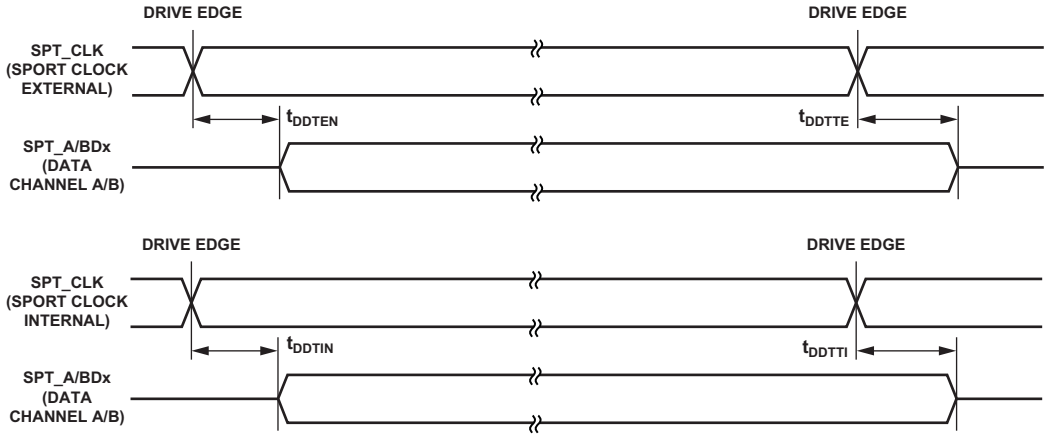



*Figure 28. Serial Ports—Enable and Three-State*

The SPT_TDV output signal becomes active in SPORT multichannel mode. During transmit slots (enabled with active channel selection registers) the SPT_TDV is asserted for communication with external devices.

**Table 52. Serial Ports—Transmit Data Valid (TDV)**

| Parameter | | $V_{DD_EXT}$ 1.8 V Nominal Min | $V_{DD_EXT}$ 1.8 V Nominal Max | $V_{DD_EXT}$ 3.3 V Nominal Min | $V_{DD_EXT}$ 3.3 V Nominal Max | Unit |
|---|---|---|---|---|---|---|
| *Switching Characteristics* | | | | | | |
| $t_{DRDVEN}$ | Data-Valid Enable Delay from Drive Edge of External Clock[1] | 2.5 | | 2.5 | | ns |
| $t_{DFDVEN}$ | Data-Valid Disable Delay from Drive Edge of External Clock[1] | | 17.5 | | 14.5 | ns |
| $t_{DRDVIN}$ | Data-Valid Enable Delay from Drive Edge of Internal Clock[1] | –4.5 | | –3.5 | | ns |
| $t_{DFDVIN}$ | Data-Valid Disable Delay from Drive Edge of Internal Clock[1] | | 2 | | 2 | ns |

[1] Referenced to drive edge.

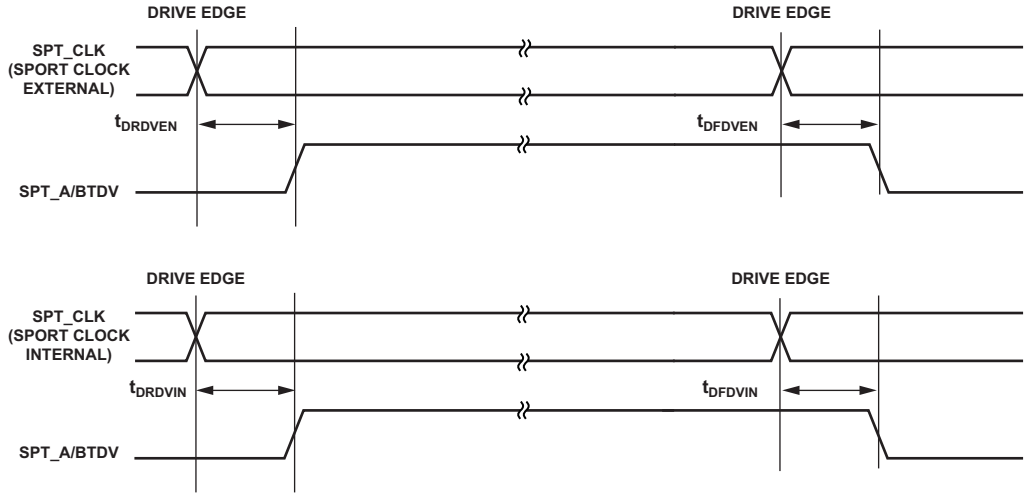



*Figure 29. Serial Ports—Transmit Data Valid Internal and External Clock*

**Table 53. Serial Ports—External Late Frame Sync**

| Parameter | | $V_{DD_EXT}$ 1.8 V Nominal Min | $V_{DD_EXT}$ 1.8 V Nominal Max | $V_{DD_EXT}$ 3.3 V Nominal Min | $V_{DD_EXT}$ 3.3 V Nominal Max | Unit |
|---|---|---|---|---|---|---|
| *Switching Characteristics* | | | | | | |
| $t_{DDTLFSE}$ | Data Delay from Late External Transmit Frame Sync or External Receive Frame Sync with MCE = 1, MFD = 0[1] | | 19 | | 15.5 | ns |
| $t_{DDTENFS}$ | Data Enable for MCE = 1, MFD = 0[1] | 0.5 | | 0.5 | | ns |

[1] The $t_{DDTLFSE}$ and $t_{DDTENFS}$ parameters apply to left-justified as well as standard serial mode, and MCE = 1, MFD = 0.

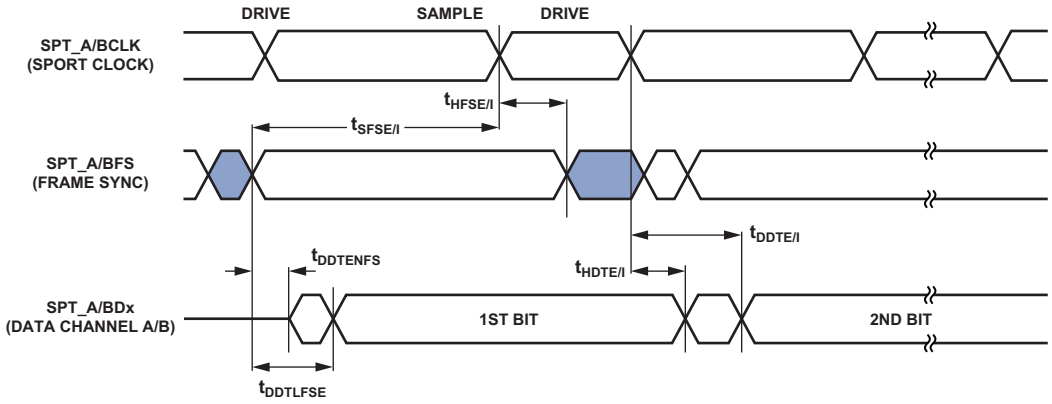



*Figure 30. External Late Frame Sync*

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

Table 54 and Figure 31 describe serial peripheral interface (SPI) port master operations.

When internally generated, the programmed SPI clock ($f_{SPICLKPROG}$) frequency in MHz is set by the following equation where BAUD is a field in the SPI_CLK register that can be set from 0 to 65,535:

$$f_{SPICLKPROG} = \frac{f_{SCLK0}}{(BAUD + 1)}$$

$$t_{SPICLKPROG} = \frac{1}{f_{SPICLKPROG}}$$

Note that:

- In dual mode data transmit, the SPI_MISO signal is also an output.
- In quad mode data transmit, the SPI_MISO, SPI_D2, and SPI_D3 signals are also outputs.
- In dual mode data receive, the SPI_MOSI signal is also an input.
- In quad mode data receive, the SPI_MOSI, SPI_D2, and SPI_D3 signals are also inputs.
- To add additional frame delays, see the documentation for the SPI_DLY register in the hardware reference manual.

**Table 54. Serial Peripheral Interface (SPI) Port—Master Timing**

| Parameter | | $V_{DD_EXT}$ 1.8 V Nominal Min | $V_{DD_EXT}$ 1.8 V Nominal Max | $V_{DD_EXT}$ 3.3 V Nominal Min | $V_{DD_EXT}$ 3.3 V Nominal Max | Unit |
|---|---|---|---|---|---|---|
| *Timing Requirements* | | | | | | |
| $t_{SSPIDM}$ | Data Input Valid to SPI_CLK Edge (Data Input Setup) | 6.5 | | 5.5 | | ns |
| $t_{HSPIDM}$ | SPI_CLK Sampling Edge to Data Input Invalid | 1 | | 1 | | ns |
| *Switching Characteristics* | | | | | | |
| $t_{SDSCIM}$ | $\overline{SPI_SEL}$ low to First SPI_CLK Edge | $0.5 \times t_{SCLK0} - 2.5$ | | $0.5 \times t_{SCLK0} - 1.5$ | | ns |
| $t_{SPICHM}$ | SPI_CLK High Period[1] | $0.5 \times t_{SPICLKPROG} - 1.5$ | | $0.5 \times t_{SPICLKPROG} - 1.5$ | | ns |
| $t_{SPICLM}$ | SPI_CLK Low Period[1] | $0.5 \times t_{SPICLKPROG} - 1.5$ | | $0.5 \times t_{SPICLKPROG} - 1.5$ | | ns |
| $t_{SPICLK}$ | SPI_CLK Period[1] | $t_{SPICLKPROG} - 1.5$ | | $t_{SPICLKPROG} - 1.5$ | | ns |
| $t_{HDSM}$ | Last SPI_CLK Edge to $\overline{SPI_SEL}$ High | $(0.5 \times t_{SCLK0}) - 2.5$ | | $(0.5 \times t_{SCLK0}) - 1.5$ | | ns |
| $t_{SPITDM}$ | Sequential Transfer Delay[2] | $(STOP \times t_{SPICLK}) - 1.5$ | | $(STOP \times t_{SPICLK}) - 1.5$ | | ns |
| $t_{DDSPIDM}$ | SPI_CLK Edge to Data Out Valid (Data Out Delay) | | 2.5 | | 2 | ns |
| $t_{HDSPIDM}$ | SPI_CLK Edge to Data Out Invalid (Data Out Hold) | –4.5 | | –3.5 | | ns |

[1] See Table 18 on Page 52 in Clock Related Operating Conditions for details on the minimum period that may be programmed for $t_{SPICLKPROG}$.
[2] STOP value set using the SPI_DLY.STOP bits.

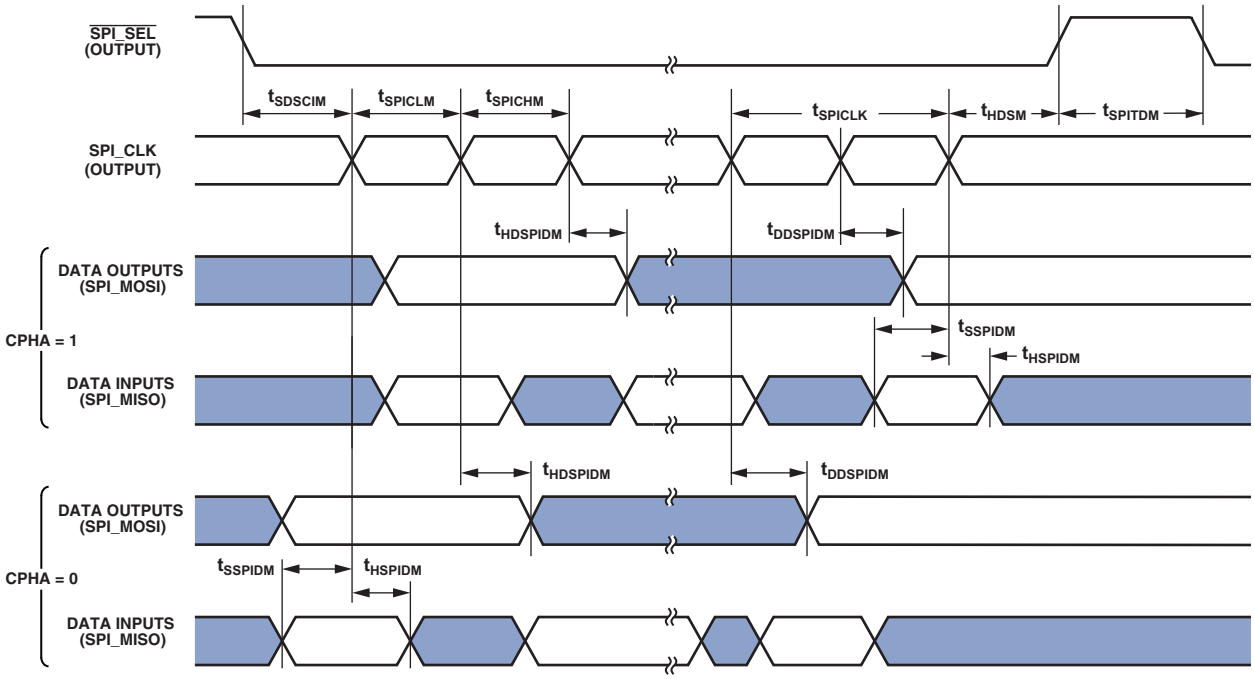



*Figure 31. Serial Peripheral Interface (SPI) Port—Master Timing*

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

Table 55 and Figure 32 describe serial peripheral interface (SPI) port slave operations. Note that:

- In dual mode data transmit, the SPI_MOSI signal is also an output.
- In quad mode data transmit, the SPI_MOSI, SPI_D2, and SPI_D3 signals are also outputs.
- In dual mode data receive, the SPI_MISO signal is also an input.
- In quad mode data receive, the SPI_MISO, SPI_D2, and SPI_D3 signals are also inputs.
- In SPI slave mode, the SPI clock is supplied externally and is called $f_{SPICLKEXT}$:

$$t_{SPICLKEXT} = \frac{1}{f_{SPICLKEXT}}$$

**Table 55. Serial Peripheral Interface (SPI) Port—Slave Timing**

| Parameter | | $V_{DD_EXT}$ 1.8 V Nominal Min | $V_{DD_EXT}$ 1.8 V Nominal Max | $V_{DD_EXT}$ 3.3 V Nominal Min | $V_{DD_EXT}$ 3.3 V Nominal Max | Unit |
|---|---|---|---|---|---|---|
| *Timing Requirements* | | | | | | |
| $t_{SPICHS}$ | SPI_CLK High Period[1] | $(0.5 \times t_{SPICLKEXT}) - 1.5$ | | $(0.5 \times t_{SPICLKEXT}) - 1.5$ | | ns |
| $t_{SPICLS}$ | SPI_CLK Low Period[1] | $(0.5 \times t_{SPICLKEXT}) - 1.5$ | | $(0.5 \times t_{SPICLKEXT}) - 1.5$ | | ns |
| $t_{SPICLK}$ | SPI_CLK Period[1] | $t_{SPICLKEXT} - 1.5$ | | $t_{SPICLKEXT} - 1.5$ | | ns |
| $t_{HDS}$ | Last SPI_CLK Edge to $\overline{\text{SPI_SS}}$ Not Asserted (NonSPIHP) | 5 | | 5 | | ns |
| $t_{HDS}$ | Last SPI_CLK Edge to $\overline{\text{SPI_SS}}$ Not Asserted (Using SPIHP) | $1.5 \times t_{SCLK0}$ | | $1.5 \times t_{SCLK0}$ | | ns |
| $t_{SPITDS}$ | Sequential Transfer Delay (NonSPIHP) | $0.5 \times t_{SPICLK} - 1.5$ | | $0.5 \times t_{SPICLK} - 1.5$ | | ns |
| $t_{SPITDS}$ | Sequential Transfer Delay (Using SPIHP) | $3 \times t_{SCLK0}$ | | $3 \times t_{SCLK0}$ | | ns |
| $t_{SDSCI}$ | $\overline{\text{SPI_SS}}$ Assertion to First SPI_CLK Edge | 11.5 | | 11.5 | | ns |
| $t_{SSPID}$ | Data Input Valid to SPI_CLK Edge (Data Input Setup) | 1.5 | | 1 | | ns |
| $t_{HSPID}$ | SPI_CLK Sampling Edge to Data Input Invalid | 3.3 | | 3 | | ns |
| *Switching Characteristics* | | | | | | |
| $t_{DSOE}$ | $\overline{\text{SPI_SS}}$ Assertion to Data Out Active | 0 | 17.5 | 0 | 14.5 | ns |
| $t_{DSDHI}$ | $\overline{\text{SPI_SS}}$ Deassertion to Data High Impedance | 0 | 13 | 0 | 11.5 | ns |
| $t_{DDSPID}$ | SPI_CLK Edge to Data Out Valid (Data Out Delay) | | 17.5 | | 14.5 | ns |
| $t_{HDSPID}$ | SPI_CLK Edge to Data Out Invalid (Data Out Hold) | 2.5 | | 2.5 | | ns |

[1] This specification indicates the minimum instantaneous width or period that can be tolerated due to duty cycle variation or jitter on the external SPI_CLK. For the external SPI_CLK ideal maximum frequency see the $f_{SPICLKTEXT}$ specification in Table 18 on Page 52 of Clock Related Operating Conditions.

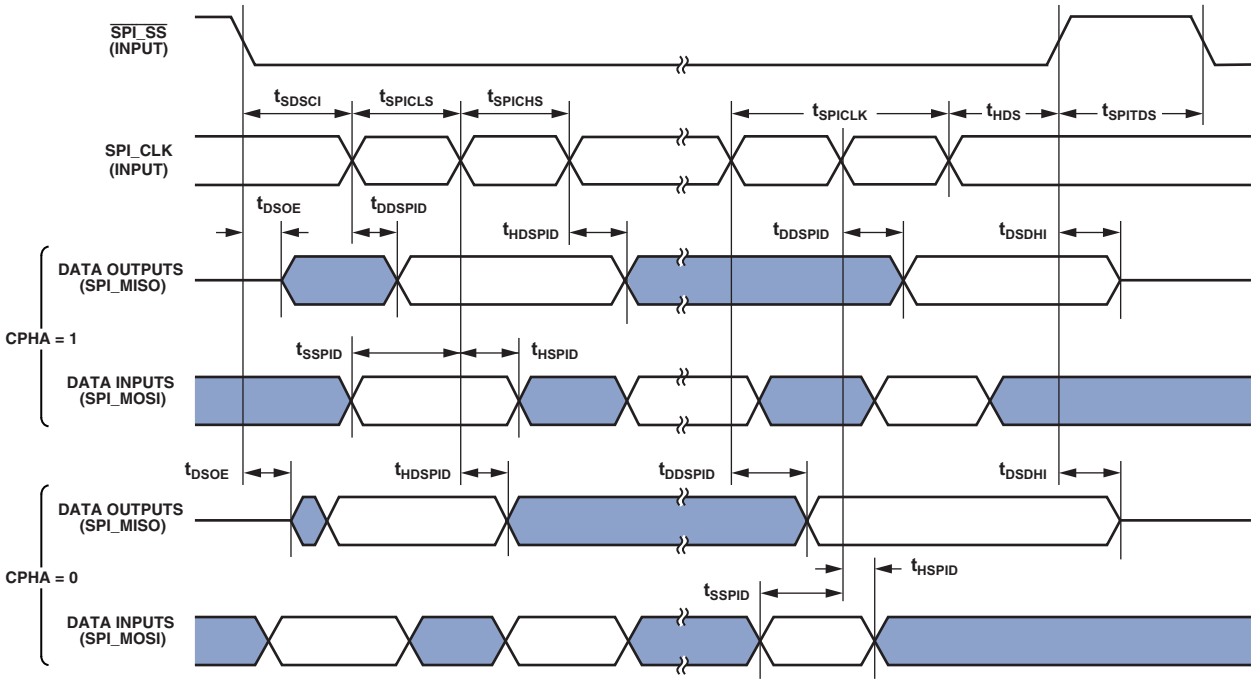



*Figure 32. Serial Peripheral Interface (SPI) Port—Slave Timing*

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

**Table 56. SPI Port—SPI_RDY Slave Timing**

| Parameter | | $V_{DD_EXT}$ 1.8 V/3.3 V Nominal Min | Max | Unit |
|---|---|---|---|---|
| *Switching Characteristics* | | | | |
| $t_{DSPISCKRDYSR}$ | SPI_RDY De-assertion from Valid Input SPI_CLK Edge in Slave Mode Receive | $2.5 \times t_{SCLK0} + t_{HDSPID}$ | $3.5 \times t_{SCLK0} + t_{DDSPID}$ | ns |
| $t_{DSPISCKRDYST}$ | SPI_RDY De-assertion from Valid Input SPI_CLK Edge in Slave Mode Transmit | $3.5 \times t_{SCLK0} + t_{HDSPID}$ | $4.5 \times t_{SCLK0} + t_{DDSPID}$ | ns |

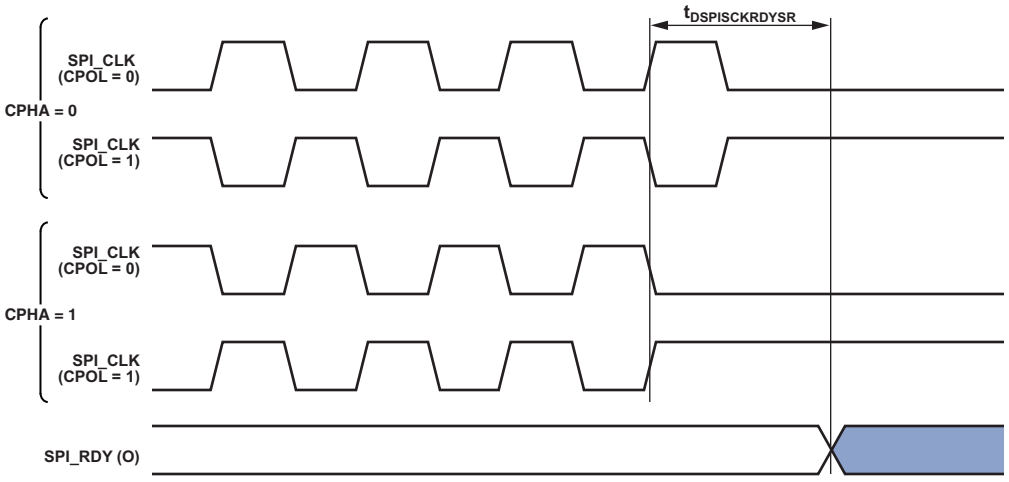



*Figure 33. SPI_RDY De-assertion from Valid Input SPI_CLK Edge in Slave Mode Receive (FCCH = 0)*

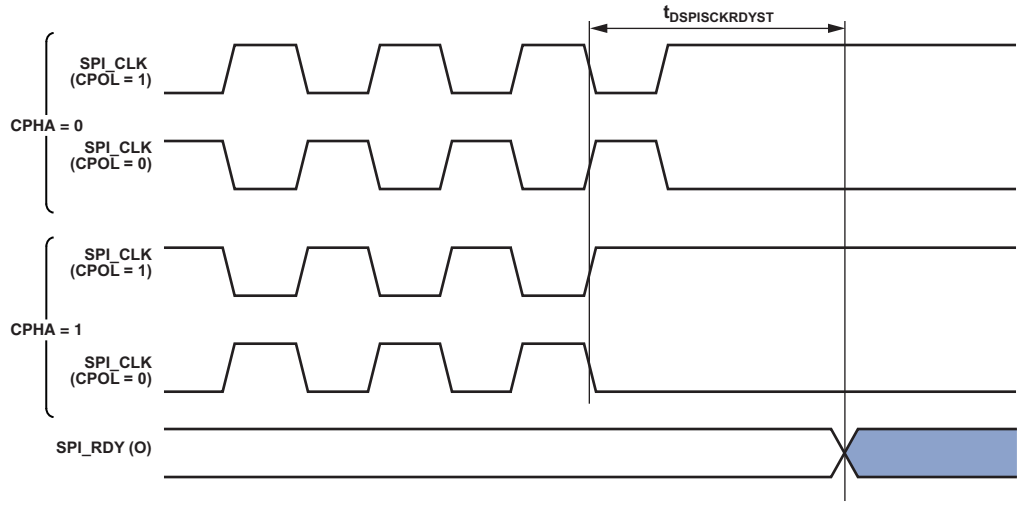



*Figure 34. SPI_RDY De-assertion from Valid Input SPI_CLK Edge in Slave Mode Transmit (FCCH = 1)*

### *Serial Peripheral Interface (SPI) Port—Open Drain Mode (ODM) Timing*

In Figure 35 and Figure 36, the outputs can be SPI_MOSI SPI_MISO, SPI_D2, and/or SPI_D3 depending on the mode of operation.

**Table 57. SPI Port ODM Master Mode Timing**

| Parameter | | $V_{DD_EXT}$ 1.8 V Nominal Min | $V_{DD_EXT}$ 1.8 V Nominal Max | $V_{DD_EXT}$ 3.3 V Nominal Min | $V_{DD_EXT}$ 3.3 V Nominal Max | Unit |
|---|---|---|---|---|---|---|
| *Switching Characteristics* | | | | | | |
| $t_{HDSPIODMM}$ | SPI_CLK Edge to High Impedance from Data Out Valid | –4.5 | | –3.5 | | ns |
| $t_{DDSPIODMM}$ | SPI_CLK Edge to Data Out Valid from High Impedance | | 2.5 | | 2 | ns |

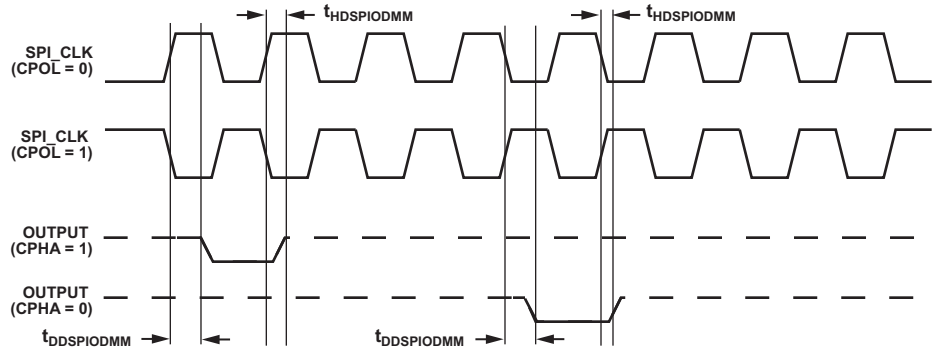



*Figure 35. ODM Master*

**Table 58. SPI Port—ODM Slave Mode**

| Parameter | | $V_{DD_EXT}$ 1.8 V Nominal Min | $V_{DD_EXT}$ 1.8 V Nominal Max | $V_{DD_EXT}$ 3.3 V Nominal Min | $V_{DD_EXT}$ 3.3 V Nominal Max | Unit |
|---|---|---|---|---|---|---|
| *Switching Characteristics* | | | | | | |
| $t_{HDSPIODMS}$ | SPI_CLK Edge to High Impedance from Data Out Valid | 2.5 | | 2.5 | | ns |
| $t_{DDSPIODMS}$ | SPI_CLK Edge to Data Out Valid from High Impedance | | 17.5 | | 14.5 | ns |

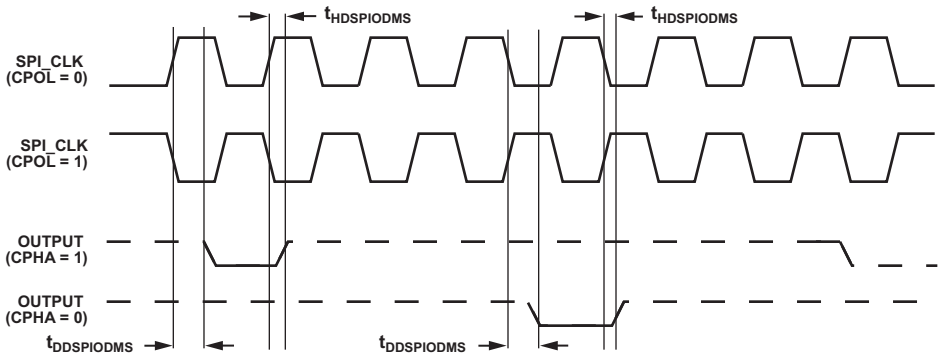



*Figure 36. ODM Slave*

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

SPI_RDY is used to provide flow control. The CPOL and CPHA bits are set in SPI_CTL, while LEADX, LAGX, and STOP are in SPI_DLY.

**Table 59. SPI Port—SPI_RDY Timing**

| Parameter | | $V_{DD_EXT}$ 1.8 V/3.3 V Nominal Min | Max | Unit |
|---|---|---|---|---|
| *Timing Requirements* | | | | |
| $t_{SRDYSCKM0}$ | Minimum Setup Time for SPI_RDY De-assertion in Master Mode Before Last SPI_CLK Edge of Valid Data Transfer to Block Subsequent Transfer with CPHA = 0 | $(2.5 + 1.5 \times BAUD^{1}) \times t_{SCLK0} + 14.5$ | | ns |
| $t_{SRDYSCKM1}$ | Minimum Setup Time for SPI_RDY De-assertion in Master Mode Before Last SPI_CLK Edge of Valid Data Transfer to Block Subsequent Transfer with CPHA = 1 | $(2.5 + BAUD^{1}) \times t_{SCLK0} + 14.5$ | | ns |
| *Switching Characteristic* | | | | |
| $t_{SRDYSCKM}$ | Time Between Assertion of SPI_RDY by Slave and First Edge of SPI_CLK for New SPI Transfer with CPHA = 0 and BAUD = 0 (STOP, LEADX, LAGX = 0) | $3 \times t_{SCLK0}$ | $4 \times t_{SCLK0} + 17.5$ | ns |
| | Time Between Assertion of SPI_RDY by Slave and First Edge of SPI_CLK for New SPI Transfer with CPHA = 0 and BAUD ≥ 1 (STOP, LEADX, LAGX = 0) | $(4 + 1.5 \times BAUD^{1}) \times t_{SCLK0}$ | $(5 + 1.5 \times BAUD^{1}) \times t_{SCLK0} + 17.5$ | ns |
| | Time Between Assertion of SPI_RDY by Slave and First Edge of SPI_CLK for New SPI Transfer with CPHA = 1 (STOP, LEADX, LAGX = 0) | $(3 + 0.5 \times BAUD^{1}) \times t_{SCLK0}$ | $(4 + 0.5 \times BAUD^{1}) \times t_{SCLK0} + 17.5$ | ns |

[1] BAUD value set using the SPI_CLK.BAUD bits.

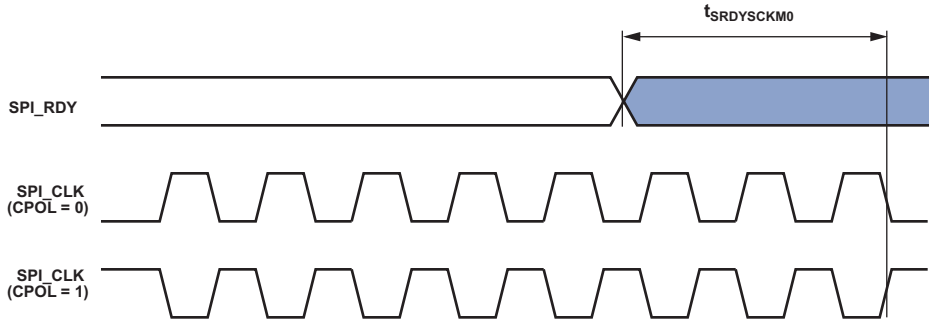



*Figure 37. SPI_RDY Setup Before SPI_CLK with CPHA = 0*

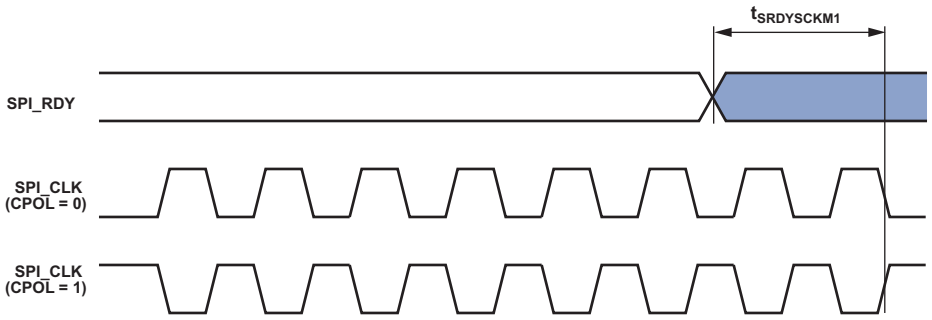



*Figure 38. SPI_RDY Setup Before SPI_CLK with CPHA = 1*

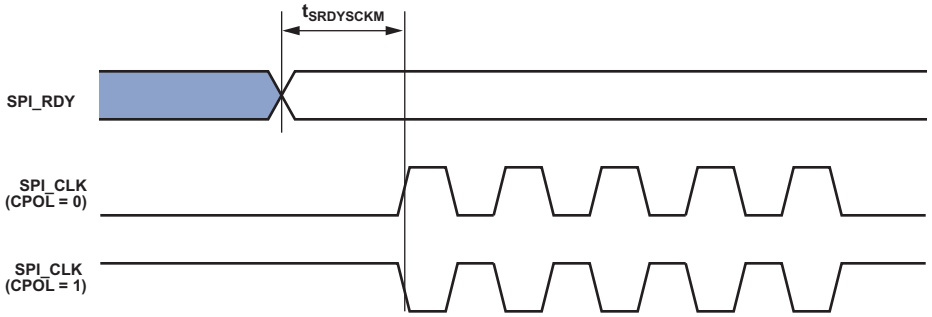



*Figure 39. SPI_CLK Switching Diagram after SPI_RDY Assertion, CPHA = x*

### *Enhanced Parallel Peripheral Interface Timing*

The following tables and figures describe enhanced parallel peripheral interface timing operations. The POLC bits in the EPPI_CTL register may be used to set the sampling/driving edges of the EPPI clock.

When internally generated, the programmed PPI clock ($f_{PCLKPROG}$) frequency in MHz is set by the following equation where VALUE is a field in the EPPI_CLKDIV register that can be set from 0 to 65,535:

$$f_{PCLKPROG} = \frac{f_{SCLK0}}{(VALUE + 1)}$$

$$t_{PCLKPROG} = \frac{1}{f_{PCLKPROG}}$$

When externally generated the EPPI_CLK is called $f_{PCLKEXT}$:

$$t_{PCLKEXT} = \frac{1}{f_{PCLKEXT}}$$

**Table 60. Enhanced Parallel Peripheral Interface—Internal Clock**

| Parameter | | $V_{DD_EXT}$ 1.8 V Nominal Min | $V_{DD_EXT}$ 1.8 V Nominal Max | $V_{DD_EXT}$ 3.3 V Nominal Min | $V_{DD_EXT}$ 3.3 V Nominal Max | Unit |
|---|---|---|---|---|---|---|
| *Timing Requirements* | | | | | | |
| $t_{SFSPI}$ | External FS Setup Before EPPI_CLK | 6.5 | | 5 | | ns |
| $t_{HFSPI}$ | External FS Hold After EPPI_CLK | 1.5 | | 1 | | ns |
| $t_{SDRPI}$ | Receive Data Setup Before EPPI_CLK | 6.4 | | 5 | | ns |
| $t_{HDRPI}$ | Receive Data Hold After EPPI_CLK | 1 | | 1 | | ns |
| $t_{SFS3GI}$ | External FS3 Input Setup Before EPPI_CLK Fall Edge in Clock Gating Mode | 16.5 | | 14 | | ns |
| $t_{HFS3GI}$ | External FS3 Input Hold Before EPPI_CLK Fall Edge in Clock Gating Mode | 1.5 | | 0 | | ns |
| *Switching Characteristics* | | | | | | |
| $t_{PCLKW}$ | EPPI_CLK Width[1] | $0.5 \times t_{PCLKPROG} - 2$ | | $0.5 \times t_{PCLKPROG} - 2$ | | ns |
| $t_{PCLK}$ | EPPI_CLK Period[1] | $t_{PCLKPROG} - 2$ | | $t_{PCLKPROG} - 2$ | | ns |
| $t_{DFSPI}$ | Internal FS Delay After EPPI_CLK | | 2 | | 2 | ns |
| $t_{HOFSPI}$ | Internal FS Hold After EPPI_CLK | –4 | | –3 | | ns |
| $t_{DDTPI}$ | Transmit Data Delay After EPPI_CLK | | 2 | | 2 | ns |
| $t_{HDTPI}$ | Transmit Data Hold After EPPI_CLK | –4 | | –3 | | ns |

[1] See Table 18 on Page 52 in Clock Related Operating Conditions for details on the minimum period that may be programmed for $t_{PCLKPROG}$.

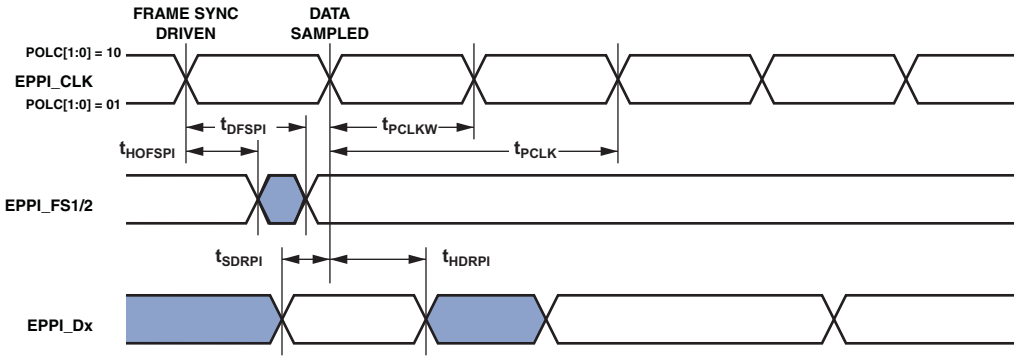



*Figure 40. PPI Internal Clock GP Receive Mode with Internal Frame Sync Timing*

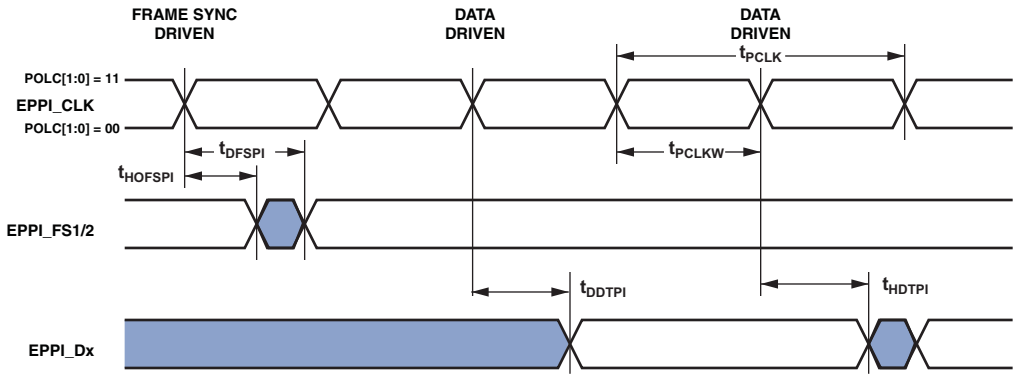



*Figure 41. PPI Internal Clock GP Transmit Mode with Internal Frame Sync Timing*

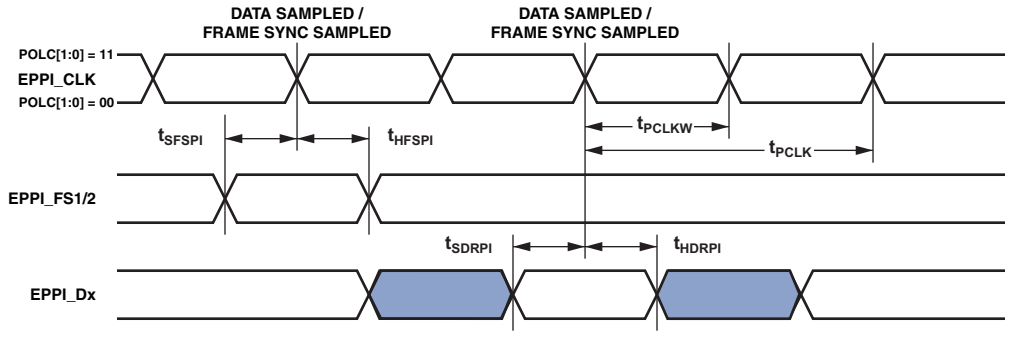



*Figure 42. PPI Internal Clock GP Receive Mode with External Frame Sync Timing*

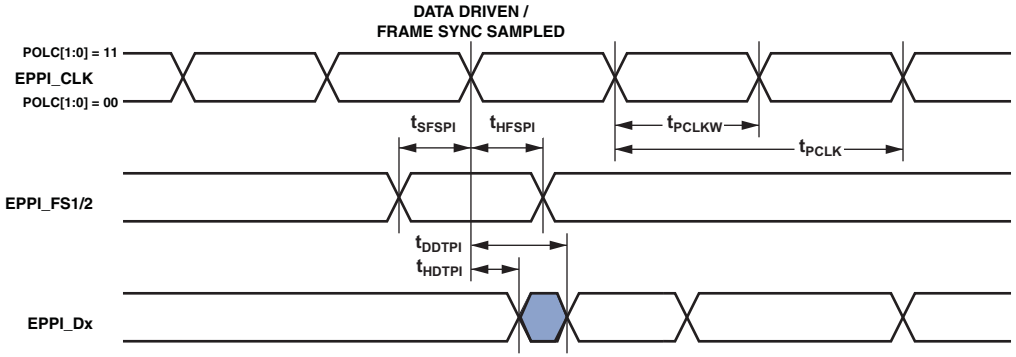



*Figure 43. PPI Internal Clock GP Transmit Mode with External Frame Sync Timing*

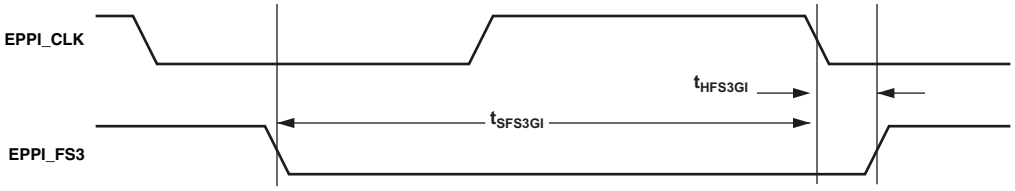



*Figure 44. Clock Gating Mode with Internal Clock and External Frame Sync Timing*

**Table 61. Enhanced Parallel Peripheral Interface—External Clock**

| Parameter | | $V_{DD_EXT}$ 1.8 V Nominal Min | $V_{DD_EXT}$ 1.8 V Nominal Max | $V_{DD_EXT}$ 3.3 V Nominal Min | $V_{DD_EXT}$ 3.3 V Nominal Max | Unit |
|---|---|---|---|---|---|---|
| *Timing Requirements* | | | | | | |
| $t_{PCLKW}$ | EPPI_CLK Width[1] | $(0.5 \times t_{PCLKEXT}) - 1$ | | $(0.5 \times t_{PCLKEXT}) - 1$ | | ns |
| $t_{PCLK}$ | EPPI_CLK Period[1] | $t_{PCLKEXT} - 1$ | | $t_{PCLKEXT} - 1$ | | ns |
| $t_{SFSPE}$ | External FS Setup Before EPPI_CLK | 1.5 | | 1 | | ns |
| $t_{HFSPE}$ | External FS Hold After EPPI_CLK | 3.3 | | 3 | | ns |
| $t_{SDRPE}$ | Receive Data Setup Before EPPI_CLK | 1 | | 1 | | ns |
| $t_{HDRPE}$ | Receive Data Hold After EPPI_CLK | 3 | | 3 | | ns |
| *Switching Characteristics* | | | | | | |
| $t_{DFSPE}$ | Internal FS Delay After EPPI_CLK | | 17.5 | | 14.5 | ns |
| $t_{HOFSPE}$ | Internal FS Hold After EPPI_CLK | 2.5 | | 2.5 | | ns |
| $t_{DDTPE}$ | Transmit Data Delay After EPPI_CLK | | 17.5 | | 14.5 | ns |
| $t_{HDTPE}$ | Transmit Data Hold After EPPI_CLK | 2.5 | | 2.5 | | ns |

[1] This specification indicates the minimum instantaneous width or period that can be tolerated due to duty cycle variation or jitter on the external EPPI_CLK. For the external EPPI_CLK ideal maximum frequency, see the $f_{PCLKEXT}$ specification in Table 18 on Page 52 in Clock Related Operating Conditions.

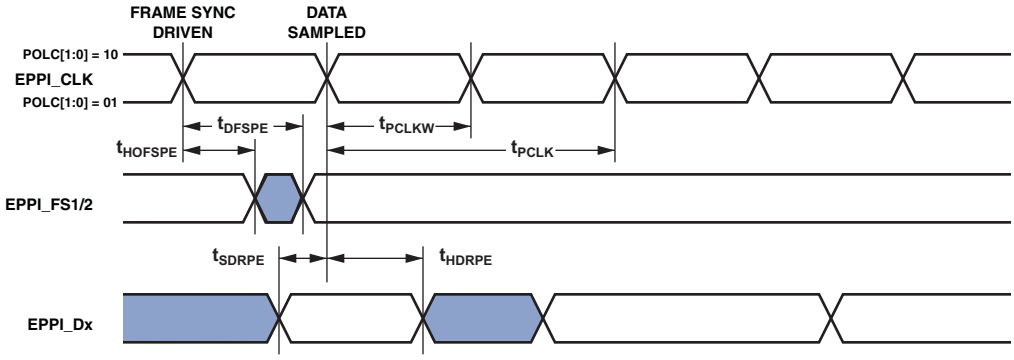



*Figure 45. PPI External Clock GP Receive Mode with Internal Frame Sync Timing*

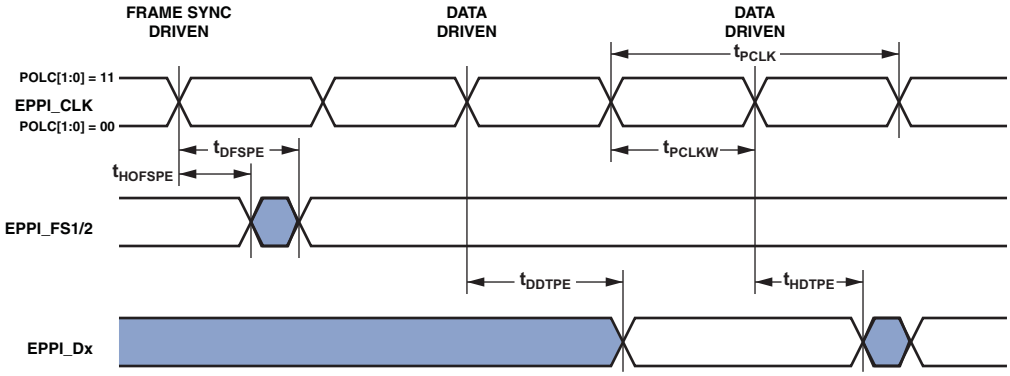



*Figure 46. PPI External Clock GP Transmit Mode with Internal Frame Sync Timing*

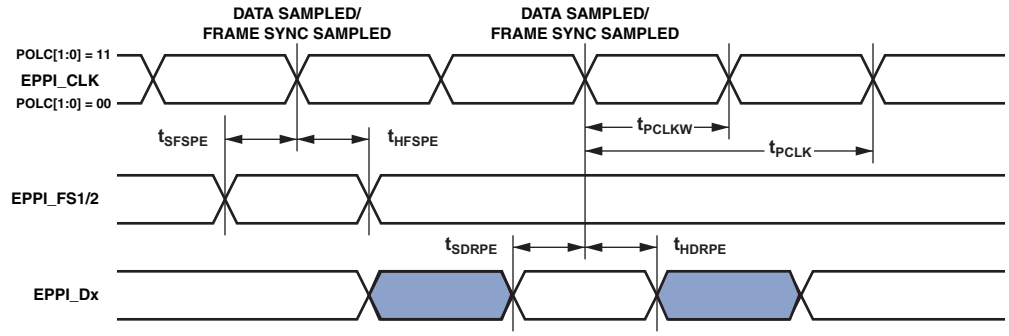



*Figure 47. PPI External Clock GP Receive Mode with External Frame Sync Timing*

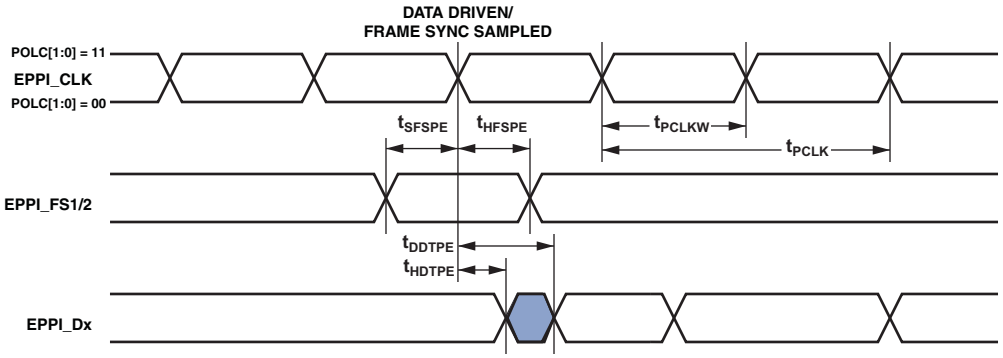



*Figure 48. PPI External Clock GP Transmit Mode with External Frame Sync Timing*

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

The universal asynchronous receiver-transmitter (UART) ports receive and transmit operations are described in the *ADSP-BF70x Blackfin+ Processor Hardware Reference*.

### *Controller Area Network (CAN) Interface*

The controller area network (CAN) interface timing is described in the *ADSP-BF70x Blackfin+ Processor Hardware Reference*.

### *Universal Serial Bus (USB) On-The-Go—Receive and Transmit Timing*

Table 62 describes the universal serial bus (USB) on-the-go receive and transmit operations.

**Table 62. USB On-The-Go—Receive and Transmit Timing**

| | | $V_{DD_USB}$ 3.3 V Nominal | | |
|---|---|---|---|---|
| **Parameter** | | **Min** | **Max** | **Unit** |
| *Timing Requirements* | | | | |
| $f_{USBS}$ | USB_XI Frequency | 24 | 24 | MHz |
| $fs_{USB}$ | USB_XI Clock Frequency Stability | –50 | +50 | ppm |

### *Mobile Storage Interface (MSI) Controller Timing*

Table 64 and Figure 49 show I/O timing, related to the mobile storage interface (MSI).

The MSI timing depends on the period of the input clock that has been routed to the MSI peripheral ($t_{MSICLKIN}$) by setting the MSI0_UHS_EXT register. See Table 63 for this information.

**Table 63. $t_{MSICLKIN}$ Settings**

| EXT_CLK_MUX_CTRL[31:30] | $t_{MSICLKIN}$ |
|---|---|
| 00 | $t_{SCLK0} \times 2$ |
| 01 | $t_{SCLK0}$ |
| 10 | $t_{SCLK1} \times 3$ |

$$t_{MSICLKIN} = \frac{1}{f_{MSICLKIN}}$$

($f_{MSICLKPROG}$) frequency in MHz is set by the following equation where DIV0 is a field in the MSI_CLKDIV register that can be set from 0 to 255. When DIV0 is set between 1 and 255, the following equation is used to determine $f_{MSICLKPROG}$:

$$f_{MSICLKPROG} = \frac{f_{MSICLKIN}}{DIV0 \times 2}$$

When DIV0 = 0,

$$f_{MSICLKPROG} = f_{MSICLKIN}$$

Also note the following:

$$t_{MSICLKPROG} = \frac{1}{f_{MSICLKPROG}}$$

**Table 64. MSI Controller Timing**

| Parameter | | $V_{DD_EXT}$ 1.8 V Nominal Min | $V_{DD_EXT}$ 1.8 V Nominal Max | $V_{DD_EXT}$ 3.3 V Nominal Min | $V_{DD_EXT}$ 3.3 V Nominal Max | Unit |
|---|---|---|---|---|---|---|
| *Timing Requirements* | | | | | | |
| $t_{ISU}$ | Input Setup Time | 5.5 | | 4.7 | | ns |
| $t_{IH}$ | Input Hold Time | 2 | | 0.5 | | ns |
| *Switching Characteristics* | | | | | | |
| $t_{MSICLK}$ | Clock Period Data Transfer Mode[1] | $t_{MSICLKPROG} - 1.5$ | | $t_{MSICLKPROG} - 1.5$ | | ns |
| $t_{WL}$ | Clock Low Time | 7 | | 7 | | ns |
| $t_{WH}$ | Clock High Time | 7 | | 7 | | ns |
| $t_{TLH}$ | Clock Rise Time | | 3 | | 3 | ns |
| $t_{THL}$ | Clock Fall Time | | 3 | | 3 | ns |
| $t_{ODLY}$ | Output Delay Time During Data Transfer Mode | | $(0.5 \times t_{MSICLKIN}) + 3.2$ | | $(0.5 \times t_{MSICLKIN}) + 3$ | ns |
| $t_{OH}$ | Output Hold Time | $(0.5 \times t_{MSICLKIN}) - 4$ | | $(0.5 \times t_{MSICLKIN}) - 3$ | | ns |

[1] See Table 18 on Page 52 in Clock Related Operating Conditions for details on the minimum period that may be programmed for $t_{MSICLKPROG}$.

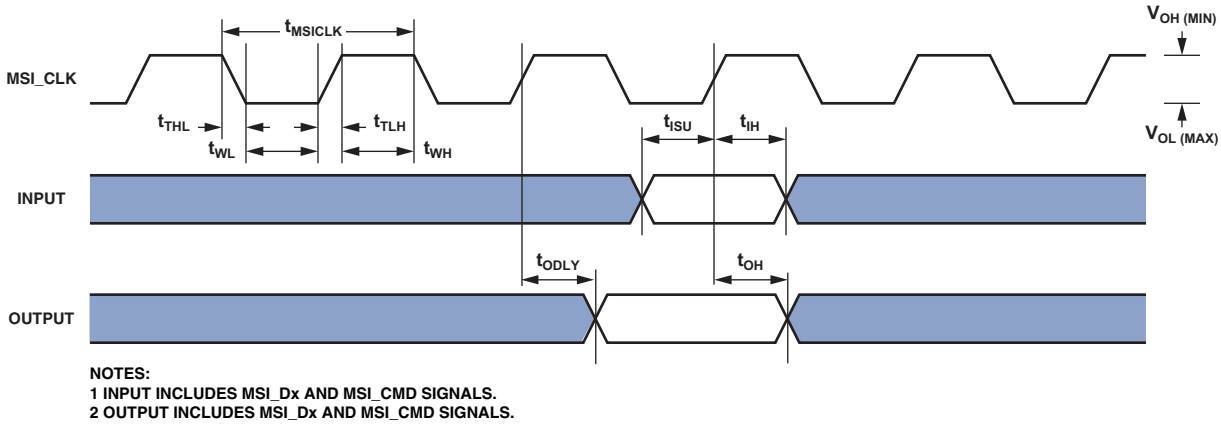



*Figure 49. MSI Controller Timing*

## OUTPUT DRIVE CURRENTS

Figure 50 through Figure 61 show typical current-voltage characteristics for the output drivers of the ADSP-BF70x Blackfin processors. The curves represent the current drive capability of the output drivers as a function of output voltage.

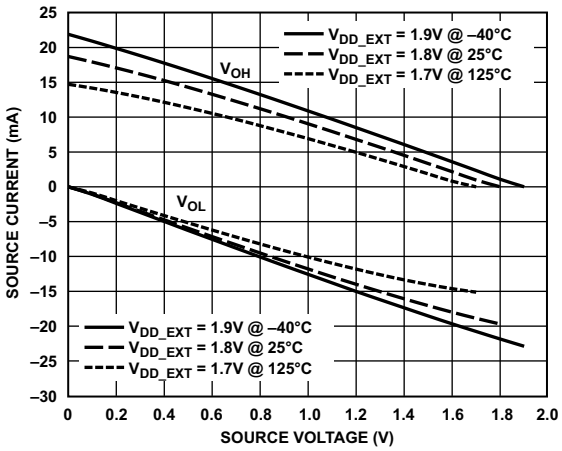



*Figure 50. Driver Type A Current (1.8 V $V_{DD_EXT}$)*

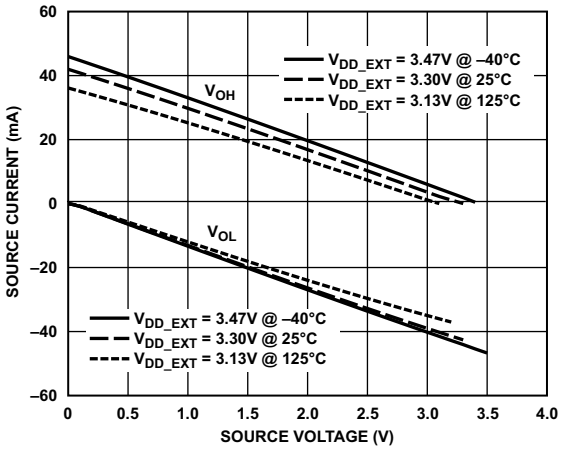



*Figure 51. Driver Type A Current (3.3 V $V_{DD_EXT}$)*

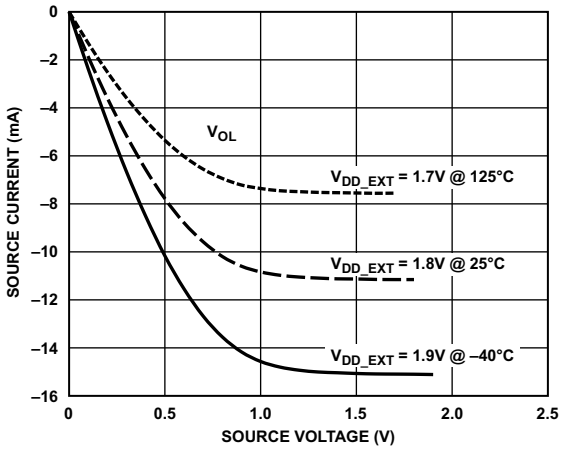



*Figure 52. Driver Type D Current (1.8 V $V_{DD_EXT}$)*

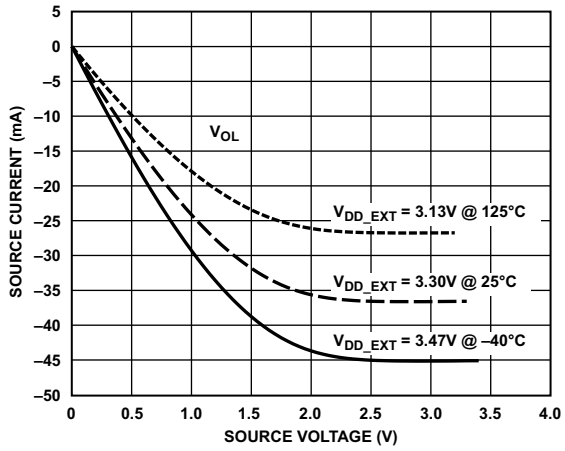



*Figure 53. Driver Type D Current (3.3 V $V_{DD_EXT}$)*

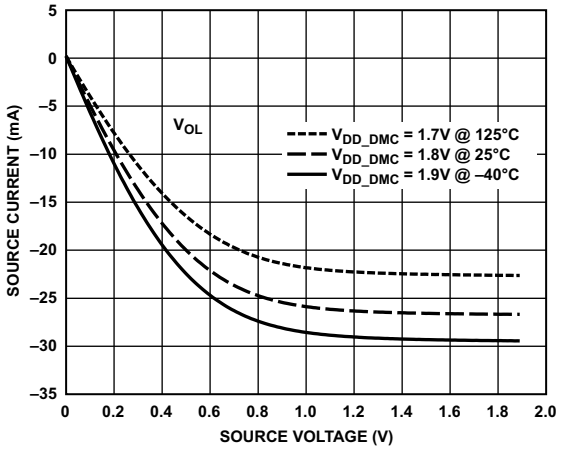



*Figure 54. Driver Type B and Driver Type C (DDR Drive Strength 34 Ω)*

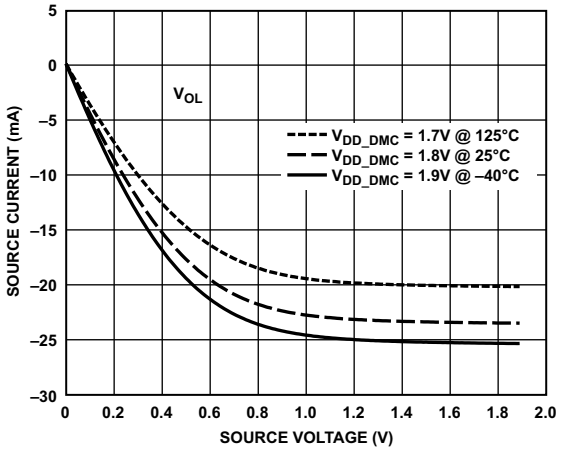



*Figure 55. Driver Type B and Driver Type C (DDR Drive Strength 40 Ω)*

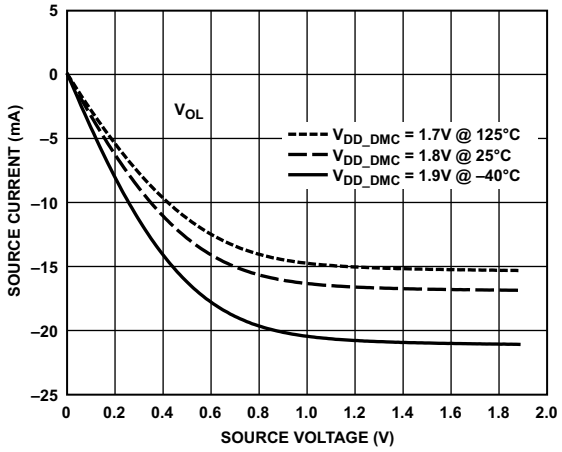



*Figure 56. Driver Type B and Driver Type C (DDR Drive Strength 50 Ω)*

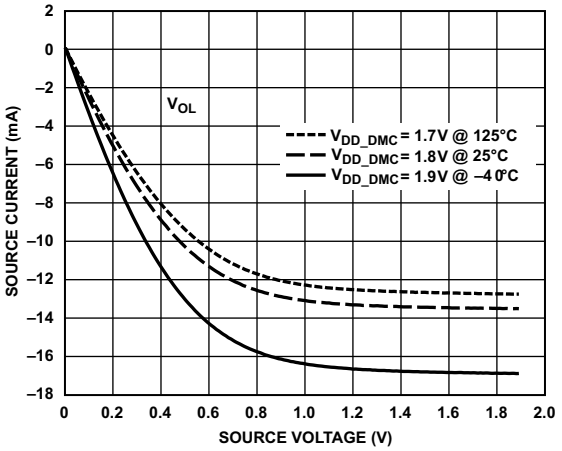



*Figure 57. Driver Type B and Driver Type C (DDR Drive Strength 60 Ω)*

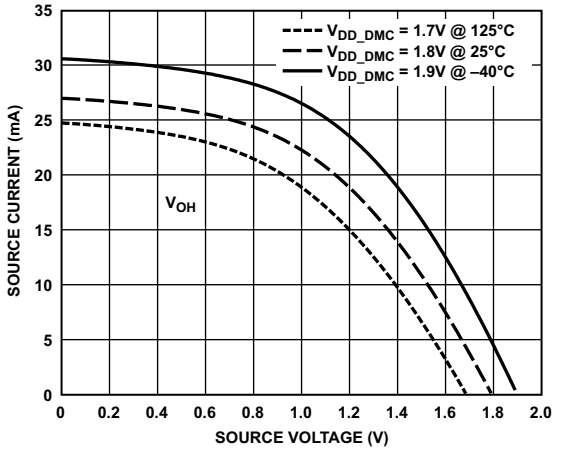



*Figure 58. Driver Type B and Driver Type C (DDR Drive Strength 34 Ω)*

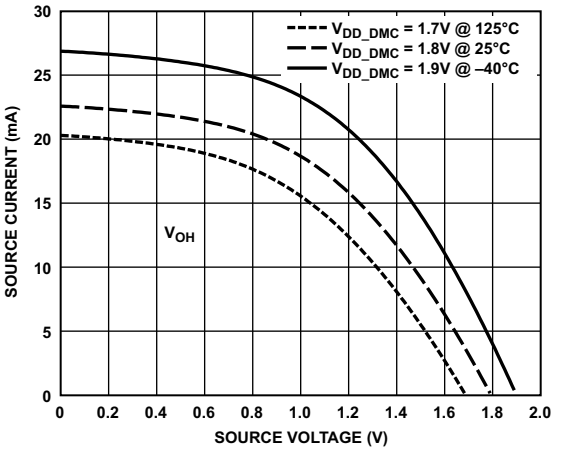



*Figure 59. Driver Type B and Driver Type C (DDR Drive Strength 40 Ω)*

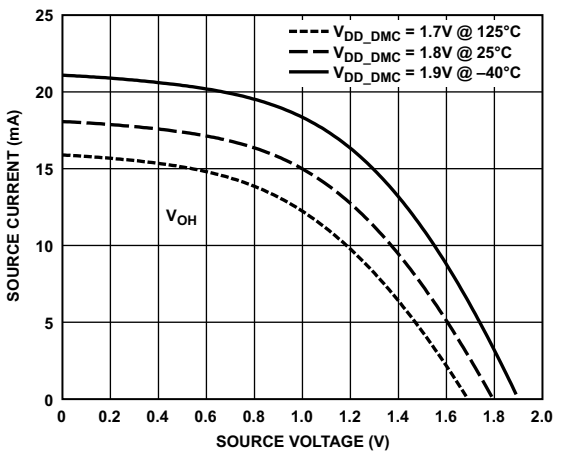



*Figure 60. Driver Type B and Driver Type C (DDR Drive Strength 50 Ω)*

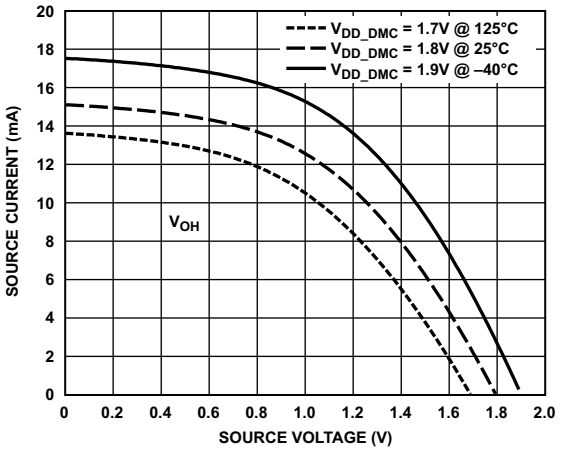



*Figure 61. Driver Type B and Device Driver C (DDR Drive Strength 60 Ω)*

## TEST CONDITIONS

All timing requirements appearing in this data sheet were measured under the conditions described in this section. Figure 62 shows the measurement point for ac measurements (except output enable/disable). The measurement point $V_{MEAS}$ is $V_{DD_EXT}/2$ for $V_{DD_EXT}$ (nominal) = 1.8 V/3.3 V.

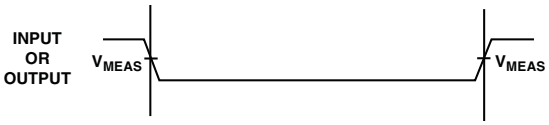



*Figure 62. Voltage Reference Levels for AC Measurements (Except Output Enable/Disable)*

### Output Enable Time Measurement

Output balls 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 on the right side of Figure 63.

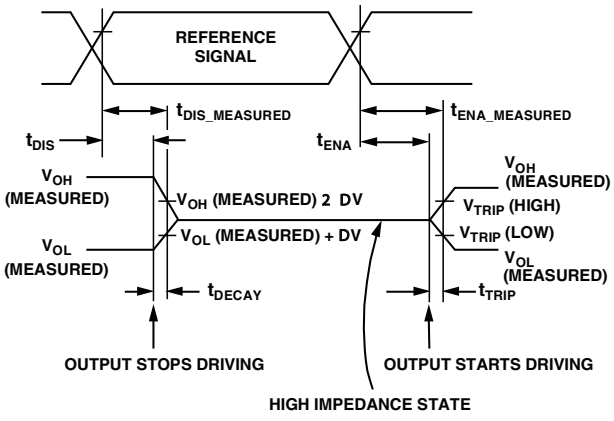



*Figure 63. Output Enable/Disable*

The time $t_{ENA_MEASURED}$ is the interval from when the reference signal switches to when the output voltage reaches $V_{TRIP}$ (high) or $V_{TRIP}$ (low). For $V_{DD_EXT}$ (nominal) = 1.8 V, $V_{TRIP}$ (high) is 1.05 V, and $V_{TRIP}$ (low) is 0.75 V. For $V_{DD_EXT}$ (nominal) = 3.3 V, $V_{TRIP}$ (high) is 1.9 V, and $V_{TRIP}$ (low) is 1.4 V. Time $t_{TRIP}$ is the interval from when the output starts driving to when the output reaches the $V_{TRIP}$ (high) or $V_{TRIP}$ (low) trip voltage.

Time $t_{ENA}$ is calculated as shown in the equation:

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

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

### Output Disable Time Measurement

Output balls 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 output disable time $t_{DIS}$ is the difference between $t_{DIS_MEASURED}$ and $t_{DECAY}$ as shown on the left side of Figure 63.

$$t_{DIS} = t_{DIS_MEASURED} - t_{DECAY}$$

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 time $t_{DECAY}$ is calculated with test loads $C_L$ and $I_L$, and with ΔV equal to 0.25 V for $V_{DD_EXT}$ (nominal) = 3.3 V and 0.15 V for $V_{DD_EXT}$ (nominal) = 1.8V.

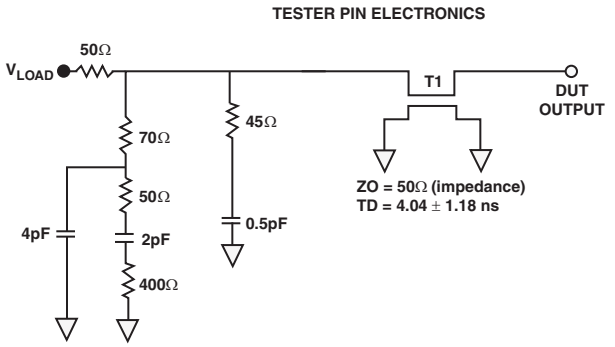
The time $t_{DIS_MEASURED}$ is the interval from when the reference signal switches, to when the output voltage decays ΔV from the measured output high or output low voltage.

### Example System Hold Time Calculation

To determine the data output hold time in a particular system, first calculate $t_{DECAY}$ using the previous equation. Choose ΔV to be the difference between the processor's output voltage and the input threshold for the device requiring the hold time. $C_L$ 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 various output disable times as specified in the Timing Specifications on Page 60.

### *Capacitive Loading*

Output delays and holds are based on standard capacitive loads of an average of 6 pF on all balls (see Figure 64). $V_{LOAD}$ is equal to $V_{DD_EXT}/2$. The graphs of Figure 65 through Figure 68 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.




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

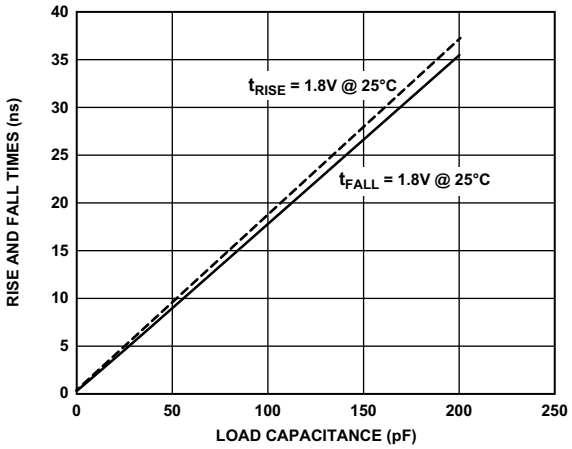



*Figure 65. Driver Type A Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance ($V_{DD_EXT}$ = 1.8 V)*

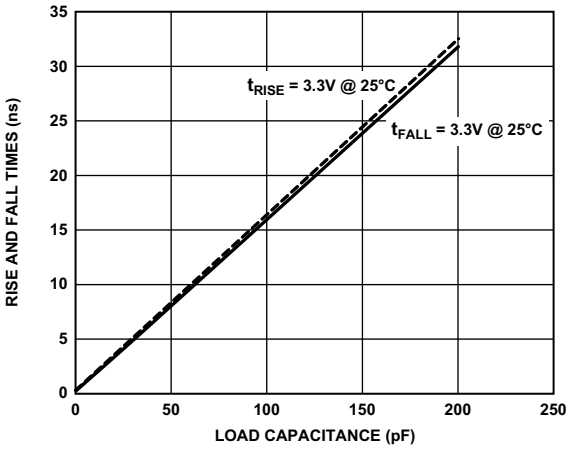



*Figure 66. Driver Type A Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance ($V_{DD_EXT}$ = 3.3 V)*

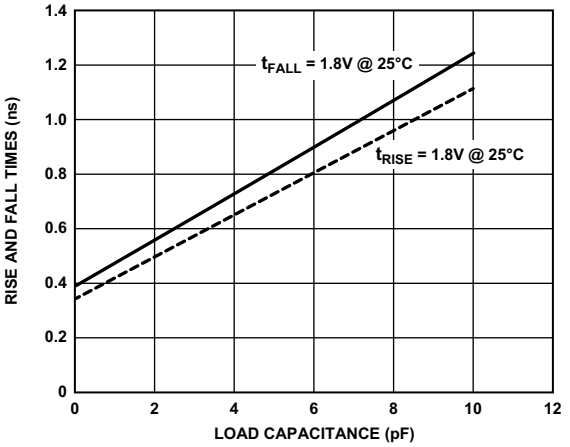



*Figure 67. Driver Type B & C Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance ($V_{DD_DMC}$ = 1.8 V)*

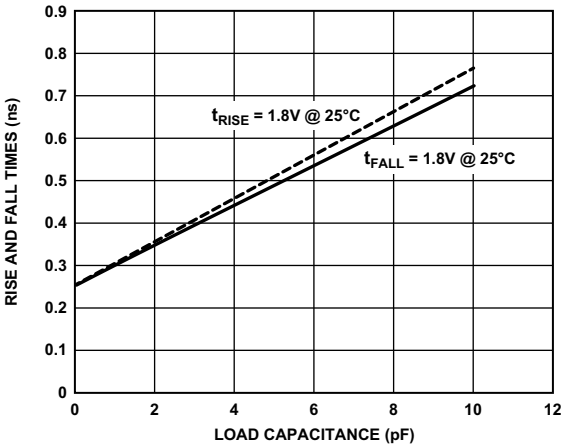



*Figure 68. Driver Type B and Driver Type C Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance ($V_{DD_DMC}$ = 1.8 V) for LPDDR*

## ENVIRONMENTAL CONDITIONS

To determine the junction temperature on the application printed circuit board, use the following equation:

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

where:

$T_J$ = Junction temperature (°C).

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

$\Psi_{JT}$ = From Table 65 and Table 66.

$P_D$ = Power dissipation (see Total Internal Power Dissipation on Page 56 for the 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).

Values of $\theta_{JC}$ are provided for package comparison and printed circuit board design considerations when an external heat sink is required.

In Table 65 and Table 66, airflow measurements comply with JEDEC standards JESD51-2 and JESD51-6. The junction-to-case measurement complies with MIL-STD-883 (Method 1012.1). All measurements use a 2S2P JEDEC test board.

**Table 65. Thermal Characteristics for CSP_BGA**

| Parameter | Condition | Typical | Unit |
|---|---|---|---|
| $\theta_{JA}$ | 0 linear m/s air flow | 28.7 | °C/W |
| $\theta_{JMA}$ | 1 linear m/s air flow | 26.2 | °C/W |
| $\theta_{JMA}$ | 2 linear m/s air flow | 25.2 | °C/W |
| $\theta_{JC}$ | | 10.1 | °C/W |
| $\Psi_{JT}$ | 0 linear m/s air flow | 0.24 | °C/W |
| $\Psi_{JT}$ | 1 linear m/s air flow | 0.40 | °C/W |
| $\Psi_{JT}$ | 2 linear m/s air flow | 0.51 | °C/W |

**Table 66. Thermal Characteristics for LFCSP (QFN)**

| Parameter | Condition | Typical | Unit |
|---|---|---|---|
| $\theta_{JA}$ | 0 linear m/s air flow | 22.9 | °C/W |
| $\theta_{JMA}$ | 1 linear m/s air flow | 17.9 | °C/W |
| $\theta_{JMA}$ | 2 linear m/s air flow | 16.4 | °C/W |
| $\theta_{JC}$ | | 2.26 | °C/W |
| $\Psi_{JT}$ | 0 linear m/s air flow | 0.14 | °C/W |
| $\Psi_{JT}$ | 1 linear m/s air flow | 0.27 | °C/W |
| $\Psi_{JT}$ | 2 linear m/s air flow | 0.30 | °C/W |

# ADSP-BF70x 184-BALL CSP_BGA BALL ASSIGNMENTS (NUMERICAL BY BALL NUMBER)

Figure 69 shows an overview of signal placement on the 184-ball CSP_BGA.

Table 67 lists the 184-ball CSP_BGA package by ball number for the ADSP-BF70x. Table 68 lists the 184-ball CSP_BGA package by signal.

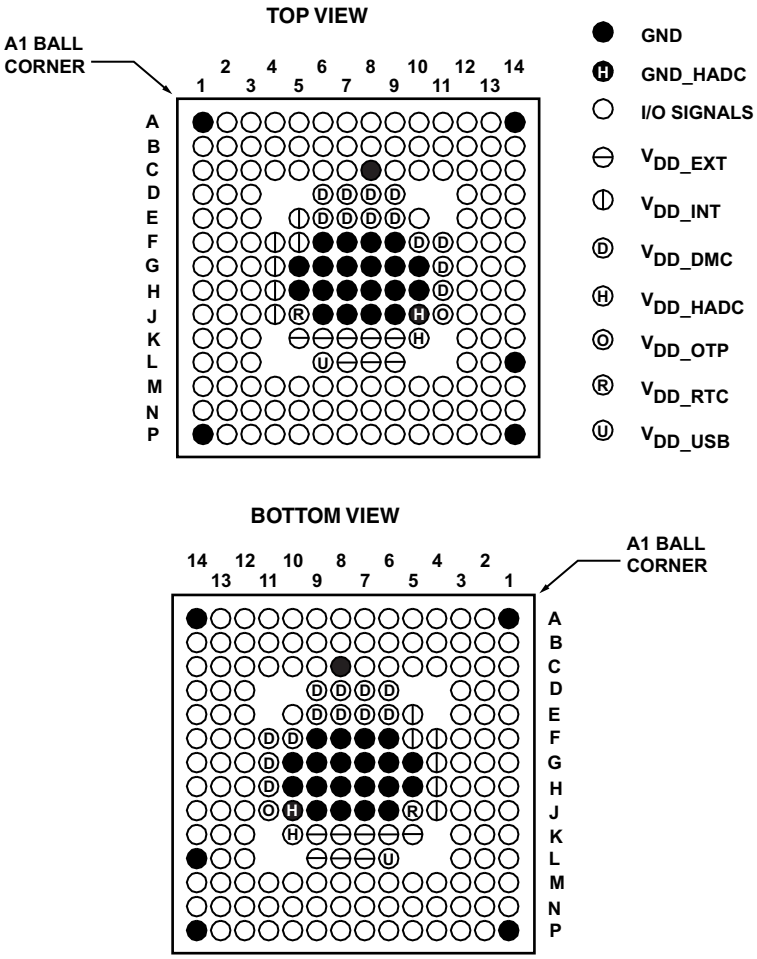



*Figure 69. 184-Ball CSP_BGA Configuration*

**Table 67. 184-Ball CSP_BGA Ball Assignment (Numerical by Ball Number)**

| Ball No. | Signal Name | Ball No. | Signal Name | Ball No. | Signal Name | Ball No. | Signal Name |
|---|---|---|---|---|---|---|---|
| A01 | GND | D08 | VDD_DMC | H03 | SYS_CLKOUT | L14 | GND |
| A02 | DMC0_A09 | D09 | VDD_DMC | H04 | VDD_INT | M01 | PC_00 |
| A03 | DMC0_BA0 | D12 | PA_08 | H05 | GND | M02 | RTC0_CLKIN |
| A04 | DMC0_BA1 | D13 | DMC0_DQ06 | H06 | GND | M03 | PB_15 |
| A05 | DMC0_BA2 | D14 | DMC0_DQ05 | H07 | GND | M04 | PB_12 |
| A06 | $\overline{\text{DMC0_CAS}}$ | E01 | DMC0_A06 | H08 | GND | M05 | PC_12 |
| A07 | $\overline{\text{DMC0_RAS}}$ | E02 | DMC0_A05 | H09 | GND | M06 | USB0_VBUS |
| A08 | DMC0_A13 | E03 | JTG_TDI | H10 | GND | M07 | USB0_VBC |
| A09 | PA_03 | E05 | VDD_INT | H11 | VDD_DMC | M08 | PB_09 |
| A10 | DMC0_CK | E06 | VDD_DMC | H12 | PA_10 | M09 | PB_05 |
| A11 | $\overline{\text{DMC0_CK}}$ | E07 | VDD_DMC | H13 | PA_11 | M10 | PB_04 |
| A12 | DMC0_LDQS | E08 | VDD_DMC | H14 | $\overline{\text{DMC0_UDQS}}$ | M11 | PB_01 |
| A13 | $\overline{\text{DMC0_LDQS}}$ | E09 | VDD_DMC | J01 | PC_05 | M12 | PB_03 |
| A14 | GND | E10 | DMC0_VREF | J02 | PC_06 | M13 | DMC0_LDM |
| B01 | DMC0_A07 | E12 | SYS_BMODE0 | J03 | $\overline{\text{SYS_RESOUT}}$ | M14 | SYS_CLKIN |
| B02 | DMC0_A08 | E13 | DMC0_DQ08 | J04 | VDD_INT | N01 | RTC0_XTAL |
| B03 | DMC0_A11 | E14 | DMC0_DQ07 | J05 | VDD_RTC | N02 | PB_14 |
| B04 | DMC0_A10 | F01 | DMC0_A01 | J06 | GND | N03 | PB_11 |
| B05 | DMC0_A12 | F02 | DMC0_A02 | J07 | GND | N04 | PC_14 |
| B06 | $\overline{\text{DMC0_WE}}$ | F03 | PC_09 | J08 | GND | N05 | PC_11 |
| B07 | $\overline{\text{DMC0_CS0}}$ | F04 | VDD_INT | J09 | GND | N06 | USB0_ID |
| B08 | DMC0_ODT | F05 | VDD_INT | J10 | GND_HADC | N07 | USB0_DP |
| B09 | DMC0_CKE | F06 | GND | J11 | VDD_OTP | N08 | PB_08 |
| B10 | DMC0_DQ00 | F07 | GND | J12 | PA_13 | N09 | PB_06 |
| B11 | DMC0_DQ02 | F08 | GND | J13 | DMC0_DQ13 | N10 | PB_00 |
| B12 | DMC0_DQ01 | F09 | GND | J14 | DMC0_UDQS | N11 | HADC0_VIN2 |
| B13 | DMC0_DQ04 | F10 | VDD_DMC | K01 | PC_04 | N12 | HADC0_VIN1 |
| B14 | DMC0_DQ03 | F11 | VDD_DMC | K02 | PC_01 | N13 | PA_15 |
| C01 | JTG_TDO_SWO | F12 | $\overline{\text{SYS_FAULT}}$ | K03 | PC_02 | N14 | SYS_XTAL |
| C02 | JTG_TMS_SWDIO | F13 | DMC0_DQ10 | K05 | VDD_EXT | P01 | GND |
| C03 | JTG_TCK_SWCLK | F14 | DMC0_DQ09 | K06 | VDD_EXT | P02 | PB_13 |
| C04 | PA_01 | G01 | DMC0_A03 | K07 | VDD_EXT | P03 | PB_10 |
| C05 | SYS_EXTWAKE | G02 | PA_00 | K08 | VDD_EXT | P04 | PC_13 |
| C06 | PA_02 | G03 | PC_08 | K09 | VDD_EXT | P05 | USB0_XTAL |
| C07 | $\overline{\text{SYS_NMI}}$ | G04 | VDD_INT | K10 | VDD_HADC | P06 | USB0_CLKIN |
| C08 | GND | G05 | GND | K12 | PA_12 | P07 | USB0_DM |
| C09 | PA_04 | G06 | GND | K13 | DMC0_DQ15 | P08 | PB_07 |
| C10 | PA_05 | G07 | GND | K14 | DMC0_DQ14 | P09 | HADC0_VREFN |
| C11 | PA_06 | G08 | GND | L01 | PC_03 | P10 | HADC0_VREFP |
| C12 | PA_07 | G09 | GND | L02 | TWI0_SDA | P11 | HADC0_VIN3 |
| C13 | $\overline{\text{SYS_HWRST}}$ | G10 | GND | L03 | TWI0_SCL | P12 | HADC0_VIN0 |
| C14 | SYS_BMODE1 | G11 | VDD_DMC | L06 | VDD_USB | P13 | PA_14 |
| D01 | DMC0_A00 | G12 | PA_09 | L07 | VDD_EXT | P14 | GND |
| D02 | DMC0_A04 | G13 | DMC0_DQ11 | L08 | VDD_EXT | | |
| D03 | $\overline{\text{JTG_TRST}}$ | G14 | DMC0_DQ12 | L09 | VDD_EXT | | |
| D06 | VDD_DMC | H01 | PC_07 | L12 | PB_02 | | |
| D07 | VDD_DMC | H02 | PC_10 | L13 | DMC0_UDM | | |

**Table 68. ADSP-BF70x 184-Ball CSP_BGA Ball Assignments (Alphabetical by Signal Name)**

| Signal Name | Ball No. | Signal Name | Ball No. | Signal Name | Ball No. | Signal Name | Ball No. |
|---|---|---|---|---|---|---|---|
| DMC0_A00 | D01 | DMC0_WE | B06 | PA_08 | D12 | SYS_HWRST | C13 |
| DMC0_A01 | F01 | GND | C08 | PA_09 | G12 | SYS_NMI | C07 |
| DMC0_A02 | F02 | GND | A01 | PA_10 | H12 | SYS_RESOUT | J03 |
| DMC0_A03 | G01 | GND | A14 | PA_11 | H13 | SYS_XTAL | N14 |
| DMC0_A04 | D02 | GND | F06 | PA_12 | K12 | TWI0_SCL | L03 |
| DMC0_A05 | E02 | GND | F07 | PA_13 | J12 | TWI0_SDA | L02 |
| DMC0_A06 | E01 | GND | F08 | PA_14 | P13 | USB0_CLKIN | P06 |
| DMC0_A07 | B01 | GND | F09 | PA_15 | N13 | USB0_DM | P07 |
| DMC0_A08 | B02 | GND | G05 | PB_00 | N10 | USB0_DP | N07 |
| DMC0_A09 | A02 | GND | G06 | PB_01 | M11 | USB0_ID | N06 |
| DMC0_A10 | B04 | GND | G07 | PB_02 | L12 | USB0_VBC | M07 |
| DMC0_A11 | B03 | GND | G08 | PB_03 | M12 | USB0_VBUS | M06 |
| DMC0_A12 | B05 | GND | G09 | PB_04 | M10 | USB0_XTAL | P05 |
| DMC0_A13 | A08 | GND | G10 | PB_05 | M09 | VDD_DMC | D06 |
| DMC0_BA0 | A03 | GND | H05 | PB_06 | N09 | VDD_DMC | D07 |
| DMC0_BA1 | A04 | GND | H06 | PB_07 | P08 | VDD_DMC | D08 |
| DMC0_BA2 | A05 | GND | H07 | PB_08 | N08 | VDD_DMC | D09 |
| DMC0_CAS | A06 | GND | H08 | PB_09 | M08 | VDD_DMC | E06 |
| DMC0_CK | A10 | GND | H09 | PB_10 | P03 | VDD_DMC | E07 |
| DMC0_CKE | B09 | GND | H10 | PB_11 | N03 | VDD_DMC | E08 |
| DMC0_CK | A11 | GND | J06 | PB_12 | M04 | VDD_DMC | E09 |
| DMC0_CS0 | B07 | GND | J07 | PB_13 | P02 | VDD_DMC | F10 |
| DMC0_DQ00 | B10 | GND | J08 | PB_14 | N02 | VDD_DMC | F11 |
| DMC0_DQ01 | B12 | GND | J09 | PB_15 | M03 | VDD_DMC | G11 |
| DMC0_DQ02 | B11 | GND | L14 | PC_00 | M01 | VDD_DMC | H11 |
| DMC0_DQ03 | B14 | GND | P01 | PC_01 | K02 | VDD_EXT | K05 |
| DMC0_DQ04 | B13 | GND | P14 | PC_02 | K03 | VDD_EXT | K06 |
| DMC0_DQ05 | D14 | GND_HADC | J10 | PC_03 | L01 | VDD_EXT | K07 |
| DMC0_DQ06 | D13 | HADC0_VIN0 | P12 | PC_04 | K01 | VDD_EXT | K08 |
| DMC0_DQ07 | E14 | HADC0_VIN1 | N12 | PC_05 | J01 | VDD_EXT | K09 |
| DMC0_DQ08 | E13 | HADC0_VIN2 | N11 | PC_06 | J02 | VDD_EXT | L07 |
| DMC0_DQ09 | F14 | HADC0_VIN3 | P11 | PC_07 | H01 | VDD_EXT | L08 |
| DMC0_DQ10 | F13 | HADC0_VREFN | P09 | PC_08 | G03 | VDD_EXT | L09 |
| DMC0_DQ11 | G13 | HADC0_VREFP | P10 | PC_09 | F03 | VDD_HADC | K10 |
| DMC0_DQ12 | G14 | JTG_TCK_SWCLK | C03 | PC_10 | H02 | VDD_INT | E05 |
| DMC0_DQ13 | J13 | JTG_TDI | E03 | PC_11 | N05 | VDD_INT | F04 |
| DMC0_DQ14 | K14 | JTG_TDO_SWO | C01 | PC_12 | M05 | VDD_INT | F05 |
| DMC0_DQ15 | K13 | JTG_TMS_SWDIO | C02 | PC_13 | P04 | VDD_INT | G04 |
| DMC0_LDM | M13 | JTG_TRST | D03 | PC_14 | N04 | VDD_INT | H04 |
| DMC0_LDQS | A12 | PA_00 | G02 | RTC0_CLKIN | M02 | VDD_INT | J04 |
| DMC0_LDQS | A13 | PA_01 | C04 | RTC0_XTAL | N01 | VDD_OTP | J11 |
| DMC0_ODT | B08 | PA_02 | C06 | SYS_BMODE0 | E12 | VDD_RTC | J05 |
| DMC0_RAS | A07 | PA_03 | A09 | SYS_BMODE1 | C14 | VDD_USB | L06 |
| DMC0_UDM | L13 | PA_04 | C09 | SYS_CLKIN | M14 | | |
| DMC0_UDQS | J14 | PA_05 | C10 | SYS_CLKOUT | H03 | | |
| DMC0_UDQS | H14 | PA_06 | C11 | SYS_EXTWAKE | C05 | | |
| DMC0_VREF | E10 | PA_07 | C12 | SYS_FAULT | F12 | | |

## ADSP-BF70x 12 mm × 12 mm 88-LEAD LFCSP (QFN) LEAD ASSIGNMENTS (NUMERICAL BY LEAD NUMBER)

Figure 70 shows an overview of signal placement on the 12 mm × 12 mm 88-lead LFCSP (QFN).

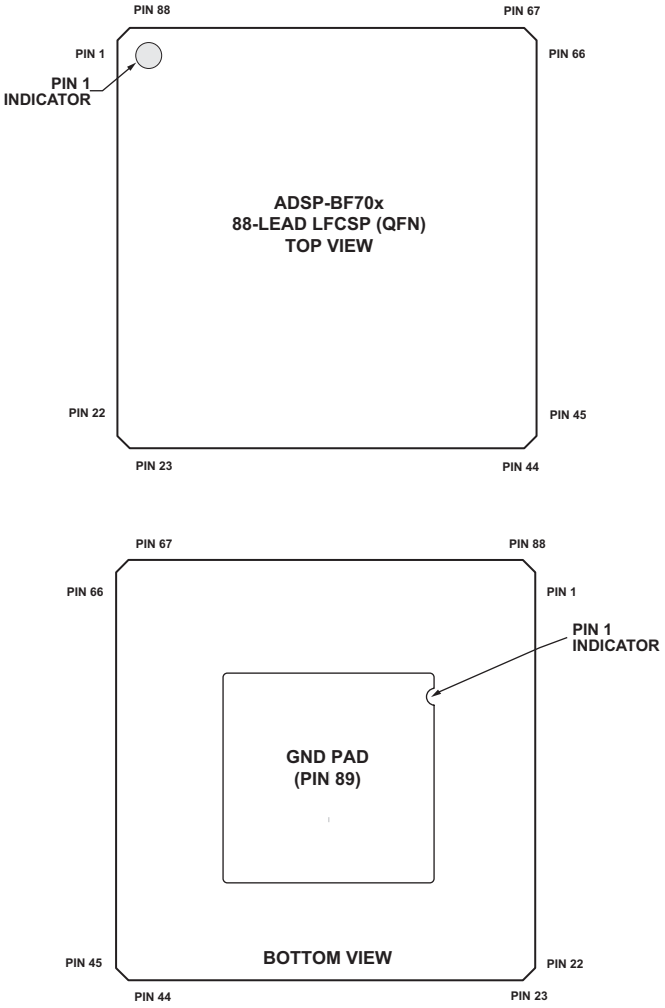



*Figure 70. 12 mm × 12 mm 88-Lead LFCSP (QFN) Configuration*

Table 69 lists the 12 mm × 12 mm 88-Lead LFCSP (QFN) package by lead number for the ADSP-BF70x. Table 70 lists the 12 mm × 12 mm 88-Lead LFCSP (QFN) package by signal.

**Table 69. 12 mm × 12 mm 88-Lead LFCSP (QFN) Lead Assignment (Numerical by Lead Number)**

| Lead No. | Signal Name | Lead No. | Signal Name | Lead No. | Signal Name | Lead No. | Signal Name |
|---|---|---|---|---|---|---|---|
| 1 | PC_10 | 24 | PB_14 | 47 | PB_02 | 70 | PA_07 |
| 2 | PC_09 | 25 | PB_13 | 48 | PB_01 | 71 | PA_06 |
| 3 | PC_08 | 26 | VDD_EXT | 49 | VDD_OTP | 72 | VDD_EXT |
| 4 | VDD_EXT | 27 | PB_12 | 50 | VDD_EXT | 73 | PA_05 |
| 5 | PC_07 | 28 | PB_11 | 51 | VDD_INT | 74 | PA_04 |
| 6 | PC_06 | 29 | PB_10 | 52 | PB_00 | 75 | PA_03 |
| 7 | PC_05 | 30 | VDD_INT | 53 | PA_15 | 76 | GND |
| 8 | PC_04 | 31 | USB0_XTAL | 54 | PA_14 | 77 | $\overline{\text{SYS_NMI}}$ |
| 9 | PC_03 | 32 | USB0_CLKIN | 55 | VDD_EXT | 78 | PA_02 |
| 10 | PC_02 | 33 | USB0_ID | 56 | SYS_XTAL | 79 | SYS_EXTWAKE |
| 11 | VDD_EXT | 34 | USB0_VBUS | 57 | SYS_CLKIN | 80 | PA_01 |
| 12 | SYS_CLKOUT | 35 | USB0_DP | 58 | PA_13 | 81 | VDD_INT |
| 13 | PC_01 | 36 | VDD_USB | 59 | PA_12 | 82 | VDD_EXT |
| 14 | VDD_INT | 37 | USB0_DM | 60 | PA_11 | 83 | JTG_TDO_SWO |
| 15 | $\overline{\text{SYS_RESOUT}}$ | 38 | USB0_VBC | 61 | VDD_INT | 84 | JTG_TMS_SWDIO |
| 16 | PC_00 | 39 | PB_09 | 62 | VDD_EXT | 85 | JTG_TCK_SWCLK |
| 17 | VDD_EXT | 40 | PB_08 | 63 | PA_10 | 86 | JTG_TDI |
| 18 | TWI0_SDA | 41 | VDD_EXT | 64 | PA_09 | 87 | $\overline{\text{JTG_TRST}}$ |
| 19 | TWI0_SCL | 42 | PB_07 | 65 | $\overline{\text{SYS_FAULT}}$ | 88 | PA_00 |
| 20 | RTC0_XTAL | 43 | PB_06 | 66 | SYS_BMODE0 | 89* | GND |
| 21 | RTC0_CLKIN | 44 | PB_05 | 67 | SYS_BMODE1 | | |
| 22 | VDD_RTC | 45 | PB_04 | 68 | $\overline{\text{SYS_HWRST}}$ | | |
| 23 | PB_15 | 46 | PB_03 | 69 | PA_08 | | |

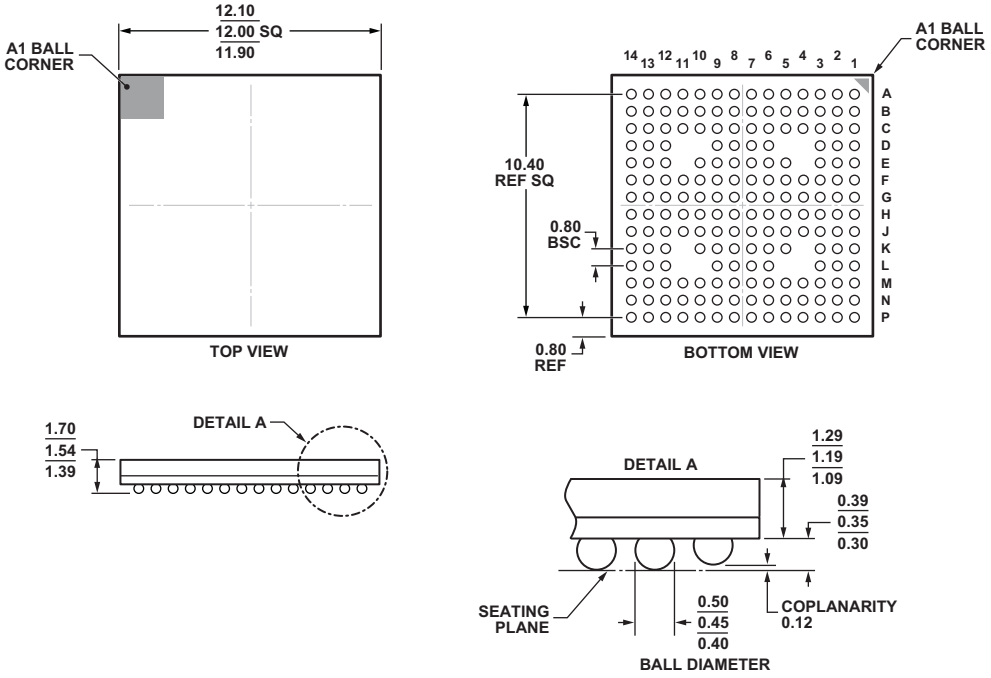
*Pin no. 89 is the GND supply (see Figure 70) for the processor; this pad must connect to GND.

**Table 70. ADSP-BF70x 12 mm × 12 mm 88 -Lead LFCSP (QFN) Lead Assignments (Alphabetical by Signal Name)**

| Signal Name | Lead No. | Signal Name | Lead No. | Signal Name | Lead No. | Signal Name | Lead No. |
|---|---|---|---|---|---|---|---|
| GND | 76 | PB_00 | 52 | PC_07 | 5 | USB0_VBUS | 34 |
| GND | 89 | PB_01 | 48 | PC_08 | 3 | USB0_XTAL | 31 |
| JTG_TCK_SWCLK | 85 | PB_02 | 47 | PC_09 | 2 | VDD_EXT | 4 |
| JTG_TDI | 86 | PB_03 | 46 | PC_10 | 1 | VDD_EXT | 11 |
| JTG_TDO_SWO | 83 | PB_04 | 45 | RTC0_CLKIN | 21 | VDD_EXT | 17 |
| JTG_TMS_SWDIO | 84 | PB_05 | 44 | RTC0_XTAL | 20 | VDD_EXT | 26 |
| $\overline{\text{JTG_TRST}}$ | 87 | PB_06 | 43 | SYS_BMODE0 | 66 | VDD_EXT | 41 |
| PA_00 | 88 | PB_07 | 42 | SYS_BMODE1 | 67 | VDD_EXT | 50 |
| PA_01 | 80 | PB_08 | 40 | SYS_CLKIN | 57 | VDD_EXT | 55 |
| PA_02 | 78 | PB_09 | 39 | SYS_CLKOUT | 12 | VDD_EXT | 62 |
| PA_03 | 75 | PB_10 | 29 | SYS_EXTWAKE | 79 | VDD_EXT | 72 |
| PA_04 | 74 | PB_11 | 28 | $\overline{\text{SYS_FAULT}}$ | 65 | VDD_EXT | 82 |
| PA_05 | 73 | PB_12 | 27 | $\overline{\text{SYS_HWRST}}$ | 68 | VDD_INT | 14 |
| PA_06 | 71 | PB_13 | 25 | $\overline{\text{SYS_NMI}}$ | 77 | VDD_INT | 30 |
| PA_07 | 70 | PB_14 | 24 | $\overline{\text{SYS_RESOUT}}$ | 15 | VDD_INT | 51 |
| PA_08 | 69 | PB_15 | 23 | SYS_XTAL | 56 | VDD_INT | 61 |
| PA_09 | 64 | PC_00 | 16 | TWI0_SCL | 19 | VDD_INT | 81 |
| PA_10 | 63 | PC_01 | 13 | TWI0_SDA | 18 | VDD_OTP | 49 |
| PA_11 | 60 | PC_02 | 10 | USB0_CLKIN | 32 | VDD_RTC | 22 |
| PA_12 | 59 | PC_03 | 9 | USB0_DM | 37 | VDD_USB | 36 |
| PA_13 | 58 | PC_04 | 8 | USB0_DP | 35 | | |
| PA_14 | 54 | PC_05 | 7 | USB0_ID | 33 | | |
| PA_15 | 53 | PC_06 | 6 | USB0_VBC | 38 | | |

# OUTLINE DIMENSIONS

Dimensions for the 12 mm × 12 mm CSP_BGA package in Figure 71 are shown in millimeters.




COMPLIANT TO JEDEC STANDARDS MO-275-GGAA-1

*Figure 71. 184-Ball Chip Scale Package Ball Grid Array [CSP_BGA]*
*(BC-184-1)*
*Dimensions shown in millimeters*

Dimensions for the 12 mm × 12 mm LFCSP_VQ package in Figure 72 are shown in millimeters.

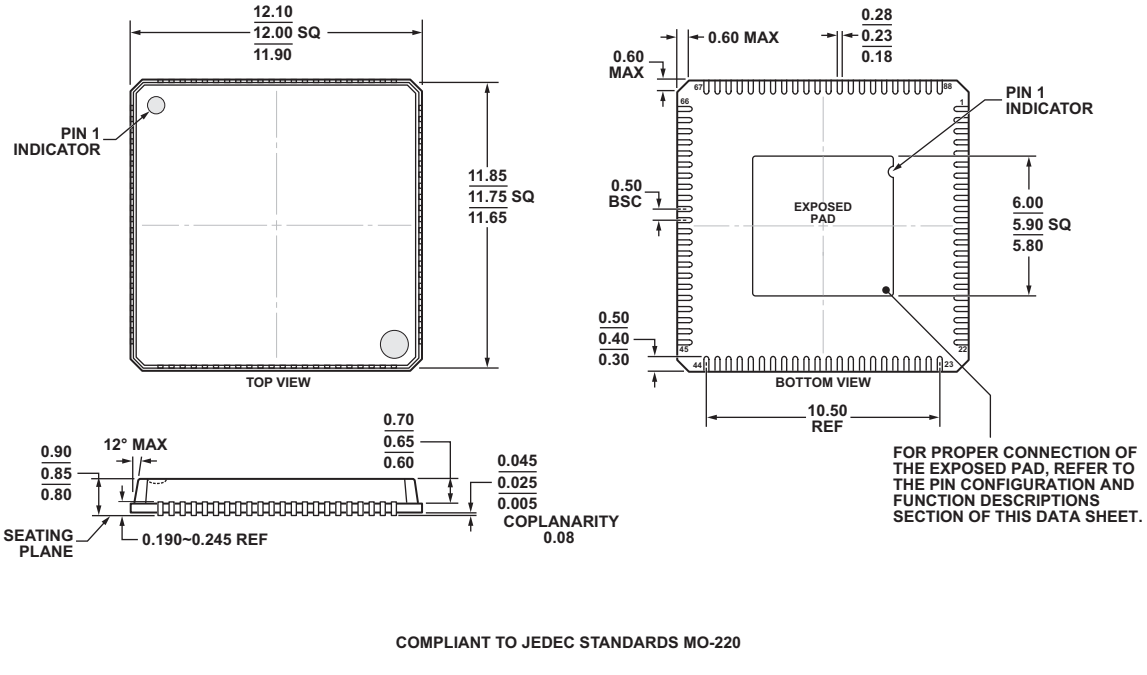



*Figure 72. 88-Lead Lead Frame Chip Scale Package [LFCSP_VQ]*
*(CP-88-8)*
*Dimensions shown in millimeters*

## SURFACE-MOUNT DESIGN

Table 71 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 71. CSP_BGA Data for Use with Surface-Mount Design**

| Package | Package Ball Attach Type | Package Solder Mask Opening | Package Ball Pad Size |
|---|---|---|---|
| BC-184-1 | Solder Mask Defined | 0.4 mm Diameter | 0.5 mm Diameter |

## PLANNED AUTOMOTIVE PRODUCTION PRODUCTS

| Model [1, 2, 3] | Max. Core Clock | L2 SRAM | Temperature Grade[4] | Package Description | Package Option |
|---|---|---|---|---|---|
| ADBF702WCCPZ3xx | 300 MHz | 256K bytes | –40°C to +105°C | 88-Lead LFCSP_VQ | CP-88-8 |
| ADBF702WCCPZ4xx | 400 MHz | 256K bytes | –40°C to +105°C | 88-Lead LFCSP_VQ | CP-88-8 |
| ADBF703WCBCZ3xx | 300 MHz | 256K bytes | –40°C to +105°C | 184-Ball CSP_BGA | BC-184-1 |
| ADBF703WCBCZ4xx | 400 MHz | 256K bytes | –40°C to +105°C | 184-Ball CSP_BGA | BC-184-1 |
| ADBF704WCCPZ3xx | 300 MHz | 512K bytes | –40°C to +105°C | 88-Lead LFCSP_VQ | CP-88-8 |
| ADBF704WCCPZ4xx | 400 MHz | 512K bytes | –40°C to +105°C | 88-Lead LFCSP_VQ | CP-88-8 |
| ADBF705WCBCZ3xx | 300 MHz | 512K bytes | –40°C to +105°C | 184-Ball CSP_BGA | BC-184-1 |
| ADBF705WCBCZ4xx | 400 MHz | 512K bytes | –40°C to +105°C | 184-Ball CSP_BGA | BC-184-1 |
| ADBF706WCCPZ3xx | 300 MHz | 1024K bytes | –40°C to +105°C | 88-Lead LFCSP_VQ | CP-88-8 |
| ADBF706WCCPZ4xx | 400 MHz | 1024K bytes | –40°C to +105°C | 88-Lead LFCSP_VQ | CP-88-8 |
| ADBF707WCBCZ3xx | 300 MHz | 1024K bytes | –40°C to +105°C | 184-Ball CSP_BGA | BC-184-1 |
| ADBF707WCBCZ4xx | 400 MHz | 1024K bytes | –40°C to +105°C | 184-Ball CSP_BGA | BC-184-1 |

[1] Select Automotive grade products, supporting –40°C to +105°C $T_{AMBIENT}$ condition, will be available when they appear in the Automotive Products table.
[2] Z = RoHS Compliant Part.
[3] xx denotes the current die revision.
[4] Referenced temperature is ambient temperature. The ambient temperature is not a specification. See Operating Conditions on Page 50 for the junction temperature ($T_J$) specification which is the only temperature specification.

## ORDERING GUIDE

| Model[1] | Max. Core Clock | L2 SRAM | Temperature Grade[2] | Package Description | Package Option |
|---|---|---|---|---|---|
| ADSP-BF700KCPZ-1 | 100 MHz | 128K bytes | 0°C to +70°C | 88-Lead LFCSP_VQ | CP-88-8 |
| ADSP-BF700KCPZ-2 | 200 MHz | 128K bytes | 0°C to +70°C | 88-Lead LFCSP_VQ | CP-88-8 |
| ADSP-BF700BCPZ-2 | 200 MHz | 128K bytes | –40°C to +85°C | 88-Lead LFCSP_VQ | CP-88-8 |
| ADSP-BF701KBCZ-1 | 100 MHz | 128K bytes | 0°C to +70°C | 184-Ball CSP_BGA | BC-184-1 |
| ADSP-BF701KBCZ-2 | 200 MHz | 128K bytes | 0°C to +70°C | 184-Ball CSP_BGA | BC-184-1 |
| ADSP-BF701BBCZ-2 | 200 MHz | 128K bytes | –40°C to +85°C | 184-Ball CSP_BGA | BC-184-1 |
| ADSP-BF702KCPZ-3 | 300 MHz | 256K bytes | 0°C to +70°C | 88-Lead LFCSP_VQ | CP-88-8 |
| ADSP-BF702BCPZ-3 | 300 MHz | 256K bytes | –40°C to +85°C | 88-Lead LFCSP_VQ | CP-88-8 |
| ADSP-BF702KCPZ-4 | 400 MHz | 256K bytes | 0°C to +70°C | 88-Lead LFCSP_VQ | CP-88-8 |
| ADSP-BF702BCPZ-4 | 400 MHz | 256K bytes | –40°C to +85°C | 88-Lead LFCSP_VQ | CP-88-8 |
| ADSP-BF703KBCZ-3 | 300 MHz | 256K bytes | 0°C to +70°C | 184-Ball CSP_BGA | BC-184-1 |
| ADSP-BF703BBCZ-3 | 300 MHz | 256K bytes | –40°C to +85°C | 184-Ball CSP_BGA | BC-184-1 |
| ADSP-BF703KBCZ-4 | 400 MHz | 256K bytes | 0°C to +70°C | 184-Ball CSP_BGA | BC-184-1 |
| ADSP-BF703BBCZ-4 | 400 MHz | 256K bytes | –40°C to +85°C | 184-Ball CSP_BGA | BC-184-1 |
| ADSP-BF704KCPZ-3 | 300 MHz | 512K bytes | 0°C to +70°C | 88-Lead LFCSP_VQ | CP-88-8 |
| ADSP-BF704BCPZ-3 | 300 MHz | 512K bytes | –40°C to +85°C | 88-Lead LFCSP_VQ | CP-88-8 |
| ADSP-BF704KCPZ-4 | 400 MHz | 512K bytes | 0C to +70C | 88-Lead LFCSP_VQ | CP-88-8 |
| ADSP-BF704BCPZ-4 | 400 MHz | 512K bytes | –40°C to +85°C | 88-Lead LFCSP_VQ | CP-88-8 |
| ADSP-BF705KBCZ-3 | 300 MHz | 512K bytes | 0°C to +70°C | 184-Ball CSP_BGA | BC-184-1 |
| ADSP-BF705BBCZ-3 | 300 MHz | 512K bytes | –40°C to +85°C | 184-Ball CSP_BGA | BC-184-1 |
| ADSP-BF705KBCZ-4 | 400 MHz | 512K bytes | 0C to +70C | 184-Ball CSP_BGA | BC-184-1 |
| ADSP-BF705BBCZ-4 | 400 MHz | 512K bytes | –40°C to +85°C | 184-Ball CSP_BGA | BC-184-1 |
| ADSP-BF706KCPZ-3 | 300 MHz | 1024K bytes | 0°C to +70°C | 88-Lead LFCSP_VQ | CP-88-8 |
| ADSP-BF706BCPZ-3 | 300 MHz | 1024K bytes | –40°C to +85°C | 88-Lead LFCSP_VQ | CP-88-8 |
| ADSP-BF706KCPZ-4 | 400 MHz | 1024K bytes | 0°C to +70°C | 88-Lead LFCSP_VQ | CP-88-8 |
| ADSP-BF706BCPZ-4 | 400 MHz | 1024K bytes | –40°C to +85°C | 88-Lead LFCSP_VQ | CP-88-8 |
| ADSP-BF707KBCZ-3 | 300 MHz | 1024K bytes | 0°C to +70°C | 184-Ball CSP_BGA | BC-184-1 |
| ADSP-BF707BBCZ-3 | 300 MHz | 1024K bytes | –40°C to +85°C | 184-Ball CSP_BGA | BC-184-1 |
| ADSP-BF707KBCZ-4 | 400 MHz | 1024K bytes | 0°C to +70°C | 184-Ball CSP_BGA | BC-184-1 |
| ADSP-BF707BBCZ-4 | 400 MHz | 1024K bytes | –40°C to +85°C | 184-Ball CSP_BGA | BC-184-1 |

[1] Z = RoHS Compliant Part.

[2] Referenced temperature is ambient temperature. The ambient temperature is not a specification. See Operating Conditions on Page 50 for the junction temperature ($T_J$) specification which is the only temperature specification.

©2015 Analog Devices, Inc. All rights reserved. Trademarks and registered trademarks are the property of their respective owners.
D12396-0-9/15(A)

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