TABLE OF CONTENTS

System Features 1
Memory 1
Additional Features 1
Table Of Contents 2
Revision History 2
General Description 3
ARM Cortex-A5 Processor 5
SHARC Processor 6
SHARC+ Core Architecture 8
System Infrastructure 10
System Memory Map 11
Security Features 13
Security Features Disclaimer 14
Safety Features 14
Processor Peripherals 15
System Acceleration 19
System Design 20
System Debug 22
Development Tools 22
Additional Information 23
Related Signal Chains 23
ADSP-SC57x/ADSP-2157x Detailed Signal Descriptions 24
400-Ball CSP_BGA Signal Descriptions 28
GPIO Multiplexing for 400-Ball CSP_BGA Package 35
176-Lead LQFP Signal Descriptions 38
GPIO Multiplexing for 176-Lead LQFP Package 43
ADSP-SC57x/ADSP-2157x Designer Quick Reference 45
Specifications 56
Operating Conditions 56
Electrical Characteristics 60
HADC 64
TMU 64
Absolute Maximum Ratings 65
ESD Caution 65
Timing Specifications 66
Output Drive Currents 122
Test Conditions 124
Environmental Conditions 126
ADSP-SC57x/ADSP-2157x 400-Ball BGA Ball Assignments 127
Numerical by Ball Number 127
Alphabetical by Pin Name 130
Configuration of the 400-Ball CSP_BGA 133
ADSP-SC57x/ADSP-2157x 176-Lead LQFP Lead Assignments 134
Numerical by Lead Number 134
Alphabetical by Pin Name 136
Configuration of the 176-Lead LQFP Lead Configuration 137
Outline Dimensions 138
Surface-Mount Design 139
Automotive Products 140
Ordering Guide 141

REVISION HISTORY

6/2018—Rev. A to Rev. B

Changes to System Features 1
Changes to Additional Features 1
Changes to Table 2 and Table 3, General Description 3
Changes to Operating Conditions 56
Deleted Package Information from Specifications 56
Changes to Table 27 and Table 28, Clock Related Operating Conditions 58
Changes to Electrical Characteristics 60
Changes to Table 29, Table 32, and Table 33, Total Internal Power Dissipation 62
Changes to Table 37, HADC Timing Specifications 64
Changes to Program Trace Macrocell (PTM) Timing 120
Changes to Test Conditions 124
Changes to Automotive Products 140
Changes to Ordering Guide 141

Downloaded from Arrow.com.

GENERAL DESCRIPTION

The ADSP-SC57x/ADSP-2157x processors are members of the SHARC® family of products. The ADSP-SC57x processor is based on the SHARC+® dual-core and the ARM® Cortex®-A5 core. The ADSP-SC57x/ADSP-2157x SHARC processors are members of the single-instruction, multiple data (SIMD) SHARC family of digital signal processors (DSPs) that feature Analog Devices Super Harvard Architecture. These 32-bit/40-bit/64-bit floating-point processors are optimized for high performance audio/floating-point applications with large on-chip static random-access memory (SRAM), multiple internal buses that eliminate input/output (I/O) bottlenecks, and innovative digital audio interfaces (DAI). New additions to the SHARC+ core include cache enhancements and branch prediction, while maintaining instruction set compatibility to previous SHARC products.

By integrating a set of industry leading system peripherals and memory (see Table 1, Table 2, and Table 3), the ARM Cortex-A5 and SHARC processor is the platform of choice for applications that require programmability similar to reduced instruction set computing (RISC), multimedia support, and leading edge signal processing in one integrated package. These applications span a wide array of markets, including automotive, professional audio, and industrial-based applications that require high floating-point performance.

Table 2 provides comparison information for features that vary across the standard processors.

Table 3 provides comparison information for features that vary across the automotive processors.

Table 1. Common Product Features

Product Features	ADSP-SC57x/ADSP-2157x
DAI (includes SRU)	1
Full SPORTs	4
S/PDIF receive/transmit	1
ASRCs	4
PCGs	2
Pin buffers	20
I^2C (TWI)	3
Quad-data bit SPI	1
Dual-data bit SPI	2
CAN2.0	2
UARTs	3
Enhanced PPI Up to 16-bit on BGA 12-bit on LQFP	1
GP timer	8
GP counter	1
Watchdog timers	3
ADC control module	Yes
Hardware accelerators	
FIR/IIR	Yes
Security cryptographic engine	Yes
Multichannel 12-bit ADC	8-channel BGA; 4-channel LQFP

Downloaded from Arrow.com.

Table 2. Comparison of ADSP-SC57x/ADSP-2157x Processor Features[1]

Processor Feature	ADSP-SC570	ADSP-SC571	ADSP-SC572	ADSP-SC573	ADSP-21571	ADSP-21573
ARM Cortex-A5 (MHz, Max)	450	500	450	500	N/A	N/A
ARM Core L1 Cache (I, D kB)	32, 32	32, 32	32, 32	32, 32	N/A	N/A
ARM Core L2 Cache (kB)	256	256	256	256	N/A	N/A
SHARC+ Core1 (MHz, Max)	450	500	450	500	500	500
SHARC+ Core2 (MHz, Max)	N/A	500	N/A	500	500	500
SHARC L1 SRAM (kB)	1 × 384	2 × 384	1 × 384	2 × 384	2 × 384	2 × 384
System Memory: L2 SRAM (Shared) (MB)	1	1	1	1	1	1
System Memory: DDR3/DDR2/LPDDR1 Controller (16-bit)	N/A	N/A	1	1	N/A	1
USB 2.0 HS + PHY (Host/Device/OTG)	N/A	N/A	1	1	N/A	N/A
EMAC Std/AVB + Timer IEEE 1588	10/100	10/100	10/100/1000	10/100/1000	N/A	N/A
SDIO/eMMC	N/A	N/A	1	1	N/A	N/A
Link Ports	1	1	2	2	1	2
GPIO Ports	Port A to D	Port A to D	Port A to F	Port A to F	Port A to D	Port A to F
GPIO + DAI Pins	64 + 20	64 + 20	92 + 20	92 + 20	64 + 20	92 + 20
Package Options	176-LQFP	176-LQFP	400-BGA	400-BGA	176-LQFP	400-BGA

[1] N/A means not applicable.

Table 3. Comparison of ADSP-SC57x/ADSP-2157x Processor Features for Automotive [1]

Processor Feature	ADSP-SC570W	ADSP-SC571W	ADSP-SC572W	ADSP-SC573W	ADSP-21571W	ADSP-21573W
ARM Cortex-A5 (MHz, Max)	450	500	450	500	N/A	N/A
ARM Core L1 Cache (I, D kB)	32, 32	32, 32	32, 32	32, 32	N/A	N/A
ARM Core L2 Cache (kB)	256	256	256	256	N/A	N/A
SHARC+ Core1 (MHz, Max)	450	500	450	500	500	500
SHARC+ Core2 (MHz, Max)	N/A	500	N/A	500	500	500
SHARC L1 SRAM (kB)	1 × 384	2 × 384	1 × 384	2 × 384	2 × 384	2 × 384
System Memory: L2 SRAM (Shared) (MB)	1	1	1	1	1	1
System Memory: DDR3/DDR2/LPDDR1 Controller (16-bit)	N/A	N/A	1	1	N/A	1
USB 2.0 HS + PHY (Host/Device/OTG)	N/A	N/A	1	1	N/A	N/A
EMAC Std/AVB + Timer IEEE 1588	10/100	10/100	10/100/1000	10/100/1000	N/A	N/A
SDIO/eMMC	N/A	N/A	1	1	N/A	N/A
MLB 3-Pin/6-Pin	3-pin	3-pin	6-pin/3-pin	6-pin/3-pin	3-pin	6-pin/3-pin
Link Ports	1	1	2	2	1	2
GPIO Ports	Port A to D	Port A to D	Port A to F	Port A to F	Port A to D	Port A to F
GPIO + DAI Pins	64 + 20	64 + 20	92 + 20	92 + 20	64 + 20	92 + 20
Package Options	176-LQFP	176-LQFP	400-BGA	400-BGA	176-LQFP	400-BGA

[1] N/A means not applicable.

Downloaded from Arrow.com.

ARM CORTEX-A5 PROCESSOR

The ARM Cortex-A5 processor (see Figure 2) is a high performance processor with the following features:

- Instruction cache unit (32 Kb) and data Level 1 (L1) cache unit (32 Kb)
- In order pipeline with dynamic branch prediction
- ARM, Thumb, and ThumbEE instruction set support
- ARM TrustZone® security extensions
- Harvard L1 memory system with a memory management unit (MMU)
- ARM v7 debug architecture
- Trace support through an embedded trace macrocell (ETM) interface
- Extension—vector floating-point unit (IEEE754) with trapless execution
- Extension—media processing engine (MPE) with NEON™ technology
- Extension—Jazelle® hardware acceleration

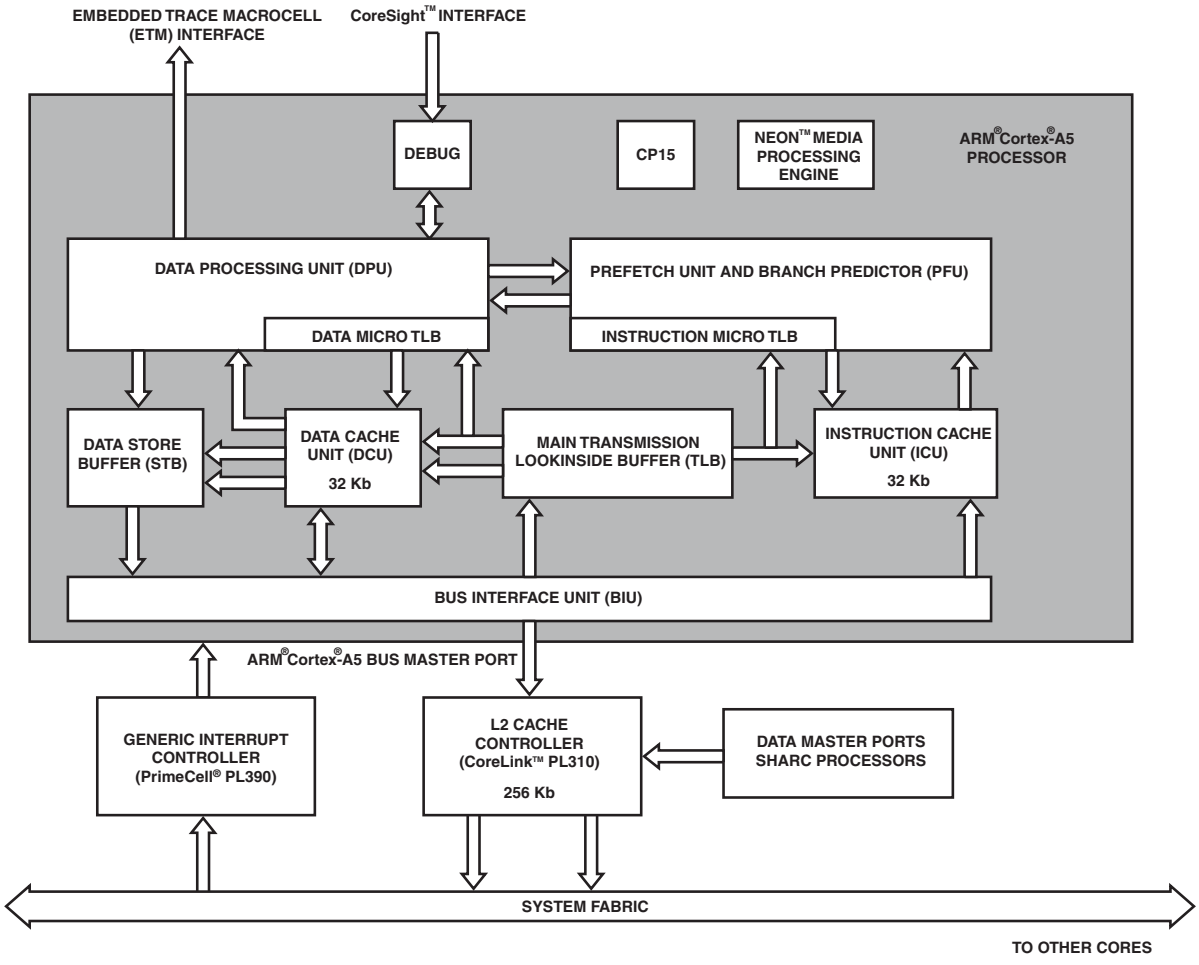


Figure 2. ARM Cortex-A5 Processor Block Diagram

Downloaded from Arrow.com.

Generic Interrupt Controller (GIC), PL390 (ADSP-SC57x Only)

The generic interrupt controller (GIC) is a centralized resource for supporting and managing interrupts. The GIC splits into the distributor block (GICPORT0) and the central processing unit (CPU) interface block (GICPORT1).

Generic Interrupt Controller Port0 (GICPORT0)

The GICPORT0 distributor block performs interrupt prioritization and distribution to the GICPORT1 CPU interface blocks that connect to the processors in the system. It centralizes all interrupt sources, determines the priority of each interrupt, and forwards the interrupt with the highest priority to the interface, for priority masking and preemption handling.

Generic Interrupt Controller Port1 (GICPORT1)

The GICPORT1 CPU interface block performs priority masking and preemption handling for a connected processor in the system. GICPORT1 supports 8 software generated interrupts (SGIs) and 212 shared peripheral interrupts (SPIs).

L2 Cache Controller, PL310 (ADSP-SC57x Only)

The Level 2 (L2) cache controller, PL310 (see Figure 2), works efficiently with the ARM Cortex-A5 processors that implement system fabric. The cache controller directly interfaces on the data and instruction interface. The internal pipelining of the cache controller is optimized to enable the processors to operate at the same clock frequency. The cache controller supports the following:

- Two read/write 64-bit slave ports, one connected to the ARM Cortex-A5 instruction and data interfaces, and one connecting the ARM Cortex-A5 and SHARC+ cores for data coherency.
- Two read/write 64-bit master ports for interfacing with the system fabric.

SHARC PROCESSOR

Figure 3 shows the SHARC processor integrates a SHARC+ SIMD core, L1 memory crossbar, I/D cache controller, L1 memory blocks, and the master/slave ports. Figure 4 shows the SHARC+ SIMD core block diagram.

The SHARC processor supports a modified Harvard architecture in combination with a hierarchical memory structure. L1 memories typically operate at the full processor speed with little or no latency.

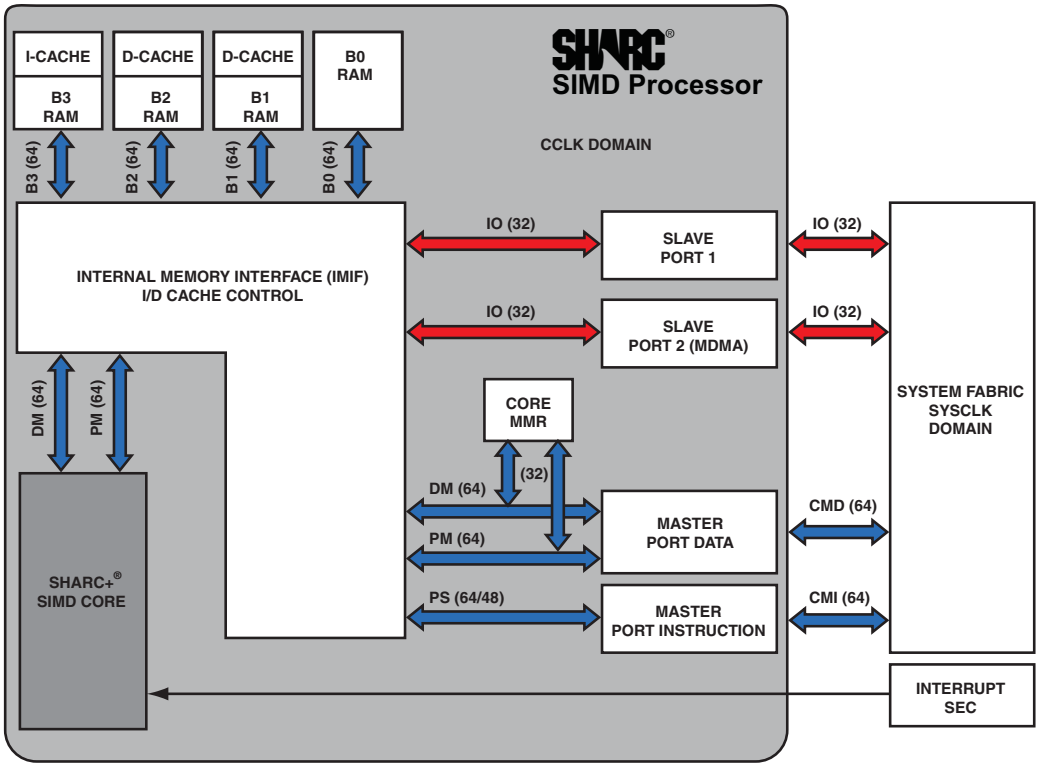


Figure 3. SHARC Processor Block Diagram

Downloaded from Arrow.com.

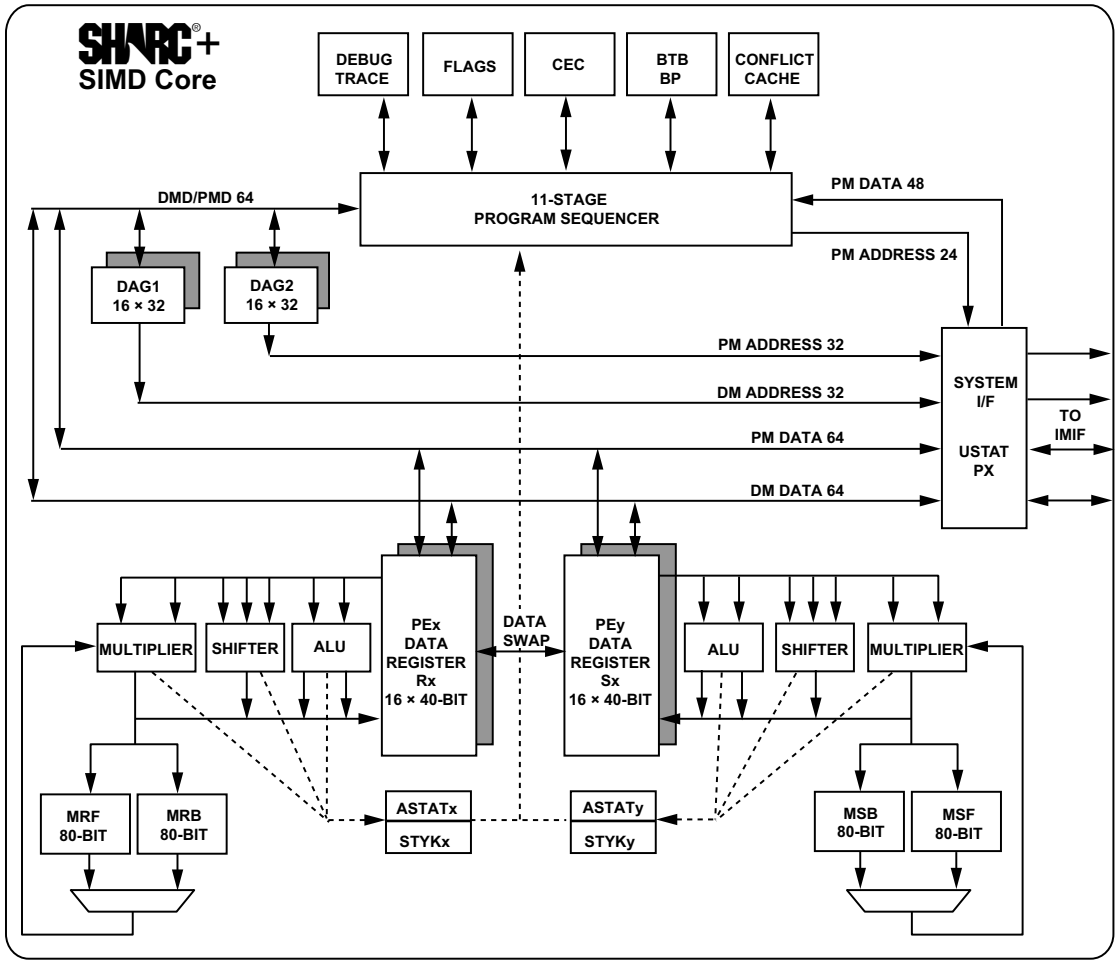


Figure 4. SHARC+ SIMD Core Block Diagram

L1 Memory

Figure 5 shows the ADSP-SC57x/ADSP-2157x memory map. Each SHARC+ core has a tightly coupled L1 SRAM of up to 3 Mb. Each SHARC+ core can access code and data in a single cycle from this memory space. The ARM Cortex-A5 core can also access this memory space with multicycle accesses.

In the SHARC+ core private address space, both cores have L1 memory.

SHARC+ core memory-mapped register (CMMR) address space is 0x00000000 through 0x0003FFFF in normal word (32-bit). Each block can be configured for different combinations of code and data storage. Of the 3 Mb SRAM, up to 1024 Kb/512 Kb can be configured for data memory (DM), program memory (PM), and instruction cache. Each memory block supports single-cycle, independent accesses by the core processor and I/O processor. The memory architecture, in combination with its separate on-chip buses, allows two data transfers from the core and one from the direct memory access (DMA) engine in a single cycle.

The SRAM of the processor can be configured as a maximum of 96k words of 32-bit data, 192k words of 16-bit data, 64k words of 48-bit instructions (or 40-bit data), or combinations of different word sizes up to 3 Mb. All of the memory can be accessed as 8-bit, 16-bit, 32-bit, 48-bit, or 64-bit words. Support of a 16-bit floating-point storage format doubles the amount of data that can be stored on chip.

Conversion between the 32-bit floating-point and 16-bit floating-point formats is performed in a single instruction. While each memory block can store combinations of code and data, accesses are most efficient when one block stores data using the DM bus for transfers, and the other block stores instructions and data using the PM bus for transfers.

Using the DM and PM buses, with each bus dedicated to a memory block, assures single-cycle execution with two data transfers. In this case, the instruction must be available in the cache.

Downloaded from Arrow.com.

The system configuration is flexible, but a typical configuration is 512 Kb DM, 128 Kb PM, and 128 Kb of instruction cache, with the remaining L1 memory configured as SRAM. Each addressable memory space outside the L1 memory can be accessed either directly or via cache.

The memory map in Table 4 gives the L1 memory address space and shows multiple L1 memory blocks offering a configurable mix of SRAM and cache.

L1 Master and Slave Ports

Each SHARC+ core has two master ports and two slave ports to and from the system fabric. One master port fetches instructions. The second master port drives data to the system world. Slave port 1 together with slave port 2 (MDMA) run conflict free access to the individual memory blocks. For the slave port address, refer to the L1 memory address map in Table 4.

L1 On-Chip Memory Bandwidth

The internal memory architecture allows programs to have four accesses at the same time to any of the four blocks, assuming no block conflicts. The total bandwidth is realized using both the DMD and PMD buses (2 × 64-bits CCLK speed and 2 × 32-bit SYSCLK speed).

Instruction and Data Cache

The ADSP-SC57x/ADSP-2157x processors also include a traditional instruction cache (I-cache) and two data caches (D-cache) (PM/DM caches) with parity support for all caches. These caches support one instruction access and two data accesses over the DM and PM buses, per CCLK cycle. The cache controllers automatically manage the configured L1 memory. The system can configure part of the L1 memory for automatic management by the cache controllers. The sizes of these caches are independently configurable from 0 kB to a maximum of 128 kB each. The memory not managed by the cache controllers is directly addressable by the processors. The controllers ensure the data coherence between the two data caches. The caches provide user-controllable features such as full and partial locking, range bound invalidation, and flushing.

System Event Controller (SEC) Input

The output of the system event controller (SEC) controller is forwarded to the core event controller (CEC) to respond directly to all unmasked system-based interrupts. The SEC also supports nesting including various SEC interrupt channel arbitration options. The processor automatically stacks the arithmetic status (ASTATx and ASTATy) registers and mode (MODE1) register in parallel with the interrupt servicing for all SEC channels.

Core Memory-Mapped Registers (CMMR)

The core memory-mapped registers (CMMR) control the L1 instruction and data cache, BTB, L2 cache, parity error, system control, debug, and monitor functions.

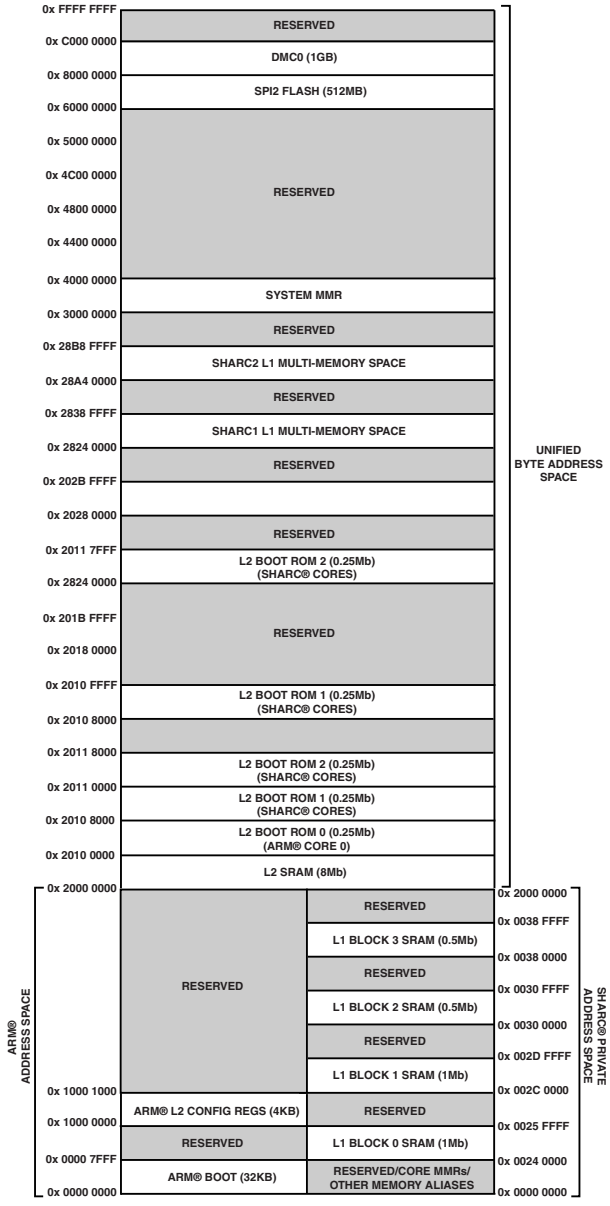


Figure 5. ADSP-SC57x/ADSP-2157x Memory Map

SHARC+ CORE ARCHITECTURE

The ADSP-SC57x/ADSP-2157x processors are code compatible at the assembly level with the ADSP-2148x, ADSP-2147x, ADSP-2146x, ADSP-2137x, ADSP-2136x, ADSP-2126x, ADSP-2116x, and with the first-generation ADSP-2106x SHARC processors.

The ADSP-SC57x/ADSP-2157x processors share architectural features with the ADSP-2126x, ADSP-2136x, ADSP-2137x, ADSP-214xx, and ADSP-2116x SIMD SHARC processors, shown in Figure 4 and detailed in the following sections.

Downloaded from Arrow.com.

Single-Instruction, Multiple Data (SIMD) Computational Engine

The SHARC+ core contains two computational processing elements that operate as a single-instruction, multiple data (SIMD) engine.

The processing elements are referred to as PEx and PEy data registers and each contain an arithmetic logic unit (ALU), multiplier, shifter, and register file. PEx is always active and PEy is enabled by setting the PEYEN mode bit in the mode control register (MODE1).

SIMD mode allows the processors to execute the same instruction in both processing elements, but each processing element operates on different data. This architecture efficiently executes math intensive DSP algorithms. In addition to all the features of previous generation SHARC cores, the SHARC+ core also provides a new and simpler way to execute an instruction only on the PEy data register.

SIMD mode also affects the way data transfers between memory and the processing elements because to sustain computational operation in the processing elements requires twice the data bandwidth. Therefore, entering SIMD mode doubles the bandwidth between memory and the processing elements. When using the DAGs to transfer data in SIMD mode, two data values transfer with each memory or register file access.

Independent Parallel Computation Units

Within each processing element is a set of pipelined computational units. The computational units consist of a multiplier, arithmetic/logic unit (ALU), and shifter. These units are arranged in parallel, maximizing computational throughput. These computational units support IEEE 32-bit single-precision floating-point, 40-bit extended-precision floating-point, IEEE 64-bit double-precision floating-point, and 32-bit fixed-point data formats.

A multifunction instruction set supports parallel execution of the ALU and multiplier operations. In SIMD mode, the parallel ALU and multiplier operations occur in both processing elements per core.

All processing operations take one cycle to complete. For all floating-point operations, the processor takes two cycles to complete in case of data dependency. Double-precision floating-point data take two to six cycles to complete. The processor stalls for the appropriate number of cycles for an interlocked pipeline plus data dependency check.

Core Timer

Each SHARC+ processor core also has a 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.

Data Register File

Each processing element contains a general-purpose data register file. The register files transfer data between the computation units and the data buses, and store intermediate results. These 10-port, 32-register register files (16 primary, 16 secondary), combined with the enhanced Harvard architecture of the processor, allow unconstrained data flow between computation units and internal memory. The registers in the PEx data register file are referred to as R0–R15 and in the PEy data register file as S0–S15.

Context Switch

Many of the registers of the processor have secondary registers that can activate during interrupt servicing for a fast context switch. The data, DAG, and multiplier result registers have secondary registers. The primary registers are active at reset, while control bits in MODE1 activate the secondary registers.

Universal Registers

General-purpose tasks use the universal registers. The four USTAT registers allow easy bit manipulations (set, clear, toggle, test, XOR) for all control and status peripheral registers.

The data bus exchange register (PX) permits data to pass between the 64-bit PM data bus and the 64-bit DM data bus or between the 40-bit register file and the PM or DM data bus. These registers contain hardware to handle the data width difference.

Data Address Generators (DAG) With Zero-Overhead Hardware Circular Buffer Support

For indirect addressing and implementing circular data buffers in hardware, the ADSP-SC57x/ADSP-2157x processor uses the two data address generators (DAGs). Circular buffers allow efficient programming of delay lines and other data structures required in digital signal processing, and are commonly used in digital filters and fast Fourier transforms (FFT). The two DAGs of the processors contain sufficient registers to allow the creation of up to 32 circular buffers (16 primary register sets and 16 secondary sets). The DAGs automatically handle address pointer wraparound, reduce overhead, increase performance, and simplify implementation. Circular buffers can start and end at any memory location.

Flexible Instruction Set Architecture (ISA)

The flexible instruction set architecture (ISA), a 48-bit instruction word, accommodates various parallel operations for concise programming. For example, the processors can conditionally execute a multiply, an add, and a subtract in both processing elements while branching and fetching up to four 32-bit values from memory—all in a single instruction. Additionally, the double-precision floating-point instruction set is an addition to the SHARC+ core.

Downloaded from Arrow.com.

Variable Instruction Set Architecture (VISA)

In addition to supporting the standard 48-bit instructions from previous SHARC processors, the SHARC+ core processors support 16-bit and 32-bit opcodes for many instructions, formerly 48-bit in the ISA. This feature, called variable instruction set architecture (VISA), drops redundant or unused bits within the 48-bit instruction to create more efficient and compact code. The program sequencer supports fetching these 16-bit and 32-bit instructions from both internal and external memories. VISA is not an operating mode; it is only address dependent (refer to memory map ISA/VISA address spaces in Table 7). Furthermore, it allows jumps between ISA and VISA instruction fetches.

Single-Cycle Fetch of Instructional Four Operands

The ADSP-SC57x/ADSP-2157x processors feature an enhanced Harvard architecture in which the DM bus transfers data and PM bus transfers both instructions and data.

With the separate program memory bus, data memory buses, and on-chip instruction conflict cache, the processor can simultaneously fetch four operands (two over each data bus) and one instruction from the conflict cache, in a single cycle.

Core Event Controller (CEC)

The SHARC+ core generates various core interrupts (including arithmetic and circular buffer instruction flow exceptions) and SEC events (debug or monitor and software). The core event controller (CEC) is used to unmask interrupts for core processing (enabled in the IMASK register).

Instruction Conflict Cache

The processors include a 32-entry instruction cache that enables three-bus operation for fetching an instruction and four data values. The cache is selective—only the instructions that require fetches conflict with the PM bus data accesses cache. This cache allows full speed execution of core, looped operations, such as digital filter multiply accumulates, and FFT butterfly processing. The conflict cache serves for on-chip bus conflicts only.

Branch Target Buffer (BTB)/Branch Predictor (BP)

Implementation of a hardware-based branch predictor (BP) and branch target buffer (BTB) reduce branch delay. The program sequencer supports efficient branching using the BTB for conditional and unconditional instructions.

Addressing Spaces

In addition to traditionally supported long word, normal word, extended precision word, and short word addressing aliases, the processors support byte addressing for the data and instruction accesses. The enhanced ISA/VISA provides new instructions for accessing all sizes of data from byte space as well as converting word addresses to byte and byte to word addresses.

Additional Features

The enhanced ISA/VISA of the ADSP-SC57x/ADSP-2157x processors provides a memory barrier instruction for data synchronization, exclusive data access support for multicore data sharing, and exclusive data access to enable multiprocessor programming. To enhance the reliability of the application, L1 data RAMs support parity error detection logic for every byte. Additionally, the processors detect illegal opcodes. Core interrupts flag both errors. Master ports of the core also detect for failed external accesses.

SYSTEM INFRASTRUCTURE

The following sections describe the system infrastructure of the ADSP-SC57x/ADSP-2157x processors.

System L2 Memory

A system L2 SRAM memory of 8 Mb (1 MB) is available to both SHARC+ cores, the ARM Cortex-A5 core, and the system DMA channels (see Table 5). The L2 SRAM block is subdivided into eight banks to support concurrent access to the L2 memory ports. Memory accesses to the L2 memory space are multicycle accesses by both the ARM Cortex-A5 and SHARC+ cores.

The memory space is used for various situations including

- ARM Cortex-A5 to SHARC+ core data sharing and intercore communications
- Accelerator and peripheral sources and destination memory to avoid accessing data in the external memory
- A location for DMA descriptors
- Storage for additional data for either the ARM Cortex-A5 or SHARC+ cores to avoid external memory latencies and reduce external memory bandwidth
- Storage for incoming Ethernet traffic to improve performance
- Storage for data coefficient tables cached by the SHARC+ core

See System Memory Protection Unit (SMPU) section for options in limiting access by specific cores and DMA masters.

The ARM Cortex-A5 core has an L1 instruction and data cache, each of which is 32 kB in size. The core also has an L2 cache controller of 256 kB. When enabling the caches, accesses to all other memory spaces (internal and external) go through the cache.

SHARC+ Core L1 Memory in Multiprocessor Space

The ARM Cortex-A5 core can access the L1 memory of the SHARC+ core. See Table 6 for the L1 memory address in multiprocessor space. The SHARC+ core can access the L1 memory of the other SHARC+ core in the multiprocessor space.

One Time Programmable Memory (OTP)

The processors feature 7 Kb of one time programmable (OTP) memory which is memory map accessible. This memory can be programmed with custom keys and it supports secure boot and secure operation.

I/O Memory Space

Mapped I/Os include SPI2 memory address space (see Table 7).

Downloaded from Arrow.com.

SYSTEM MEMORY MAP

Table 4. L1 Block 0, Block 1, Block 2, and Block 3 SHARC+® Addressing Memory Map (Private Address Space)

Memory	Long Word (64 Bits)	Extended Precision/ ISA Code (48 Bits)	Normal Word (32 Bits)	Short Word/ VISA Code (16 Bits)	Byte Access (8 Bits)
L1 Block 0 SRAM (1 Mb)	0x00048000–0x0004BFFF	0x00090000–0x00095554	0x00090000–0x00097FFF	0x00120000–0x0012FFFF	0x00240000–0x0025FFFF
L1 Block 1 SRAM (1 Mb)	0x00058000–0x0005BFFF	0x000B0000–0x000B5554	0x000B0000–0x000B7FFF	0x00160000–0x0016FFFF	0x002C0000–0x002DFFFF
L1 Block 2 SRAM (0.5 Mb)	0x00060000–0x00061FFF	0x000C0000–0x000C2AA9	0x000C0000–0x000C3FFF	0x00180000–0x00187FFF	0x00300000–0x0030FFFF
L1 Block 3 SRAM (0.5 Mb)	0x00070000–0x00071FFF	0x000E0000–0x000E2AA9	0x000E0000–0x000E3FFF	0x001C0000–0x001C7FFF	0x00380000–0x0038FFFF

Table 5. L2 Memory Addressing Map

Memory[1]	Byte Address Space ARM Cortex-A5—Data Access and Instruction Fetch SHARC+—Data Access	Normal Word Address Space SHARC+ Data Access	VISA Address Space SHARC+ Instruction Fetch	ISA Address Space SHARC+ Instruction Fetch
L2 Boot ROM0[2]	ARM: 0x00000000–0x00007FFF	0x08040000–0x08041FFF	0x00B20000–0x00B23FFF	0x00580000–0x00581555
	SHARC/DMA: 0x20100000–0x20107FFF			
L2 RAM (8 Mb)	0x20000000–0x200FFFFF	0x08000000–0x0803FFFF	0x00B80000–0x00BFFFFF	0x005C0000–0x005EAAAA
L2 Boot ROM1	0x20108000–0x2010FFFF	0x08042000–0x08043FFF	0x00B00000–0x00B03FFF	0x00500000–0x00501555
L2 Boot ROM2[3]	0x20110000–0x20117FFF	0x08044000–0x08045FFF	0x00B40000–0x00B43FFF	0x00540000–0x00541555

[1] All L2 RAM blocks are subdivided into eight banks.
[2] For ADSP-SC57x products, the L2 Boot ROM0 byte address space is 0x00000000–0x00007FFF.
[3] L2 Boot ROM address for ADSP-2157x products.

Table 6. SHARC+® L1 Memory in Multiprocessor Space

		Memory Block	Byte Address Space ARM Cortex-A5 and SHARC+	Normal Word Address Space SHARC+
L1 memory of SHARC1 in multiprocessor space	Address via Slave1 Port	Block 0	0x28240000–0x2825FFFF	0x0A090000–0x0A097FFF
		Block 1	0x282C0000–0x282DFFFF	0x0A0B0000–0x0A0B7FFF
		Block 2	0x28300000–0x2830FFFF	0x0A0C0000–0x0A0C3FFF
		Block 3	0x28380000–0x2838FFFF	0x0A0E0000–0x0A0E3FFF
L1 memory of SHARC2 in multiprocessor space	Address via Slave1 Port	Block 0	0x28A40000–0x28A5FFFF	0x0A290000–0x0A297FFF
		Block 1	0x28AC0000–0x28ADFFFF	0x0A2B0000–0x0A2B7FFF
		Block 2	0x28B00000–0x28B0FFFF	0x0A2C0000–0x0A2C3FFF
		Block 3	0x28B80000–0x28B8FFFF	0x0A2E0000–0x0A2E3FFF

Table 7. Memory Map of Mapped I/Os[1]

	Byte Address Space ARM Cortex-A5—Data Access and Instruction Fetch SHARC+—Data Access	Normal Word Address Space SHARC+ Data Access	VISA Address Space SHARC+ Instruction Fetch	ISA Address Space SHARC+ Instruction Fetch
SPI2 Memory (512 MB)	0x60000000–0x600FFFFF	0x04000000–0x07FFFFFF	0x00F80000–0x00FFFFFF	0x00780000–0x007FFFFF
	0x60100000–0x602FFFFF		Not applicable	
	0x60300000–0x6FFFFFFF		Not applicable	Not applicable
	0x70000000–0x7FFFFFFF	Not applicable	Not applicable	Not applicable

[1] The ARM Cortex-A5 can access the entire byte address space. The SHARC+ VISA/ISA address space for instruction fetch and the normal word address space for data access do not cover the entire byte address space.

Downloaded from Arrow.com.

Table 8. DMC Memory Map[1]

	Byte Address Space ARM Cortex-A5—Data Access and Instruction Fetch SHARC+—Data Access	Normal Word Address Space SHARC+ Data Access	VISA Address Space SHARC+ Instruction Fetch	ISA Address Space SHARC+ Instruction Fetch
DMC0 (1 GB)	0x80000000–0x805FFFFF	0x10000000–0x17FFFFFF	Not applicable	0x00400000–0x004FFFFF
	0x80600000–0x809FFFFF		Not applicable	Not applicable
	0x80A00000–0x80FFFFFF		0x00800000–0x00AFFFFF	Not applicable
	0x81000000–0x9FFFFFFF		Not applicable	Not applicable
	0xA0000000–0xBFFFFFFF	Not applicable	Not applicable	Not applicable

[1] The ARM Cortex-A5 can access the entire byte address space. The SHARC+ VISA/ISA address space for instruction fetch and the normal word address space for data access do not cover the entire byte address space.

System Crossbars (SCBs)

The system crossbars (SCBs) 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 interconnection.

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

Direct Memory Access (DMA)

The processors use direct memory access (DMA) to transfer data within memory spaces or between a memory space and a peripheral. The processors 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: the source channel and the destination channel.

All DMA channels 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 processors to program DMA control registers directly to initiate a DMA transfer. On completion, the DMA control registers automatically update with 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. Program a DMA channel to set up and start another DMA transfer automatically after the current sequence completes.

The DMA engine 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 autorefreshing buffer that interrupts when each buffer becomes full
- A similar circular buffer that interrupts on fractional buffers, such as at the halfway point
- The 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
- The 1D DMA uses a linked list of four-word descriptor sets containing a link pointer, an address, a length, and a configuration
- The 2D DMA uses an array of one-word descriptor sets, specifying only the base DMA address
- The 2D DMA uses a linked list of multiword descriptor sets, specifying all configurable parameters

Memory Direct Memory Access (MDMA)

The processor supports various memory direct memory access (MDMA) operations, including,

- Enhanced bandwidth MDMA channels with CRC protection (32-bit bus width, run on SYSCLK)
- Enhanced bandwidth MDMA channel (32-bit bus width, runs on SYSCLK)
- Maximum bandwidth MDMA channel (64-bit bus width, runs on SYCLK)

Extended Memory DMA

Extended memory DMA supports various operating modes, such as delay line (which allows processor reads and writes to external delay line buffers and to the external memory), with limited core interaction and scatter/gather DMA (writes to and from noncontiguous memory blocks).

Cyclic Redundant Code (CRC) Protection

The cyclic redundant codes (CRC) protection modules allow system software to calculate the signature of code, data, or both in memory, the content of memory-mapped registers, or

Downloaded from Arrow.com.

periodic communication message objects. Dedicated hardware circuitry compares the signature with precalculated values and triggers appropriate fault events.

For example, every 100 ms the system software initiates the signature calculation of the entire memory contents and compares these contents with expected, precalculated values. If a mismatch occurs, a fault condition is generated through the processor core or the trigger routing unit.

The CRC is a hardware module based on a CRC32 engine that computes the CRC value of the 32-bit data-words presented to it. The source channel of the memory to memory DMA (in memory scan mode) provides data. The data can be optionally forwarded to the destination channel (memory transfer mode). The main features of the CRC peripheral are as follows:

- Memory scan mode
- Memory transfer mode
- Data verify mode
- Data fill mode
- User-programmable CRC32 polynomial
- Bit and byte mirroring option (endianness)
- Fault and error interrupt mechanisms
- 1D and 2D fill block to initialize an array with constants
- 32-bit CRC signature of a block of a memory or an MMR block

Event Handling

The processors provide event handling that supports both nesting and prioritization. Nesting allows multiple event service routines to be active simultaneously. Prioritization ensures that servicing a higher priority event takes precedence over servicing a lower priority event.

The processors provide support for four different types of events:

- An emulation event causes the processors to enter emulation mode, allowing command and control of the processors through the JTAG interface.
- A reset event resets the processors.
- An exceptions event occurs synchronously to program flow (in other words, the exception is taken before the instruction is allowed to complete). Conditions triggered on the one side by the SHARC+ core, such as data alignment (SIMD or long word) or compute violations (fixed or floating point), and illegal instructions cause core exceptions. Conditions triggered on the other side by the SEC, such as error correcting codes (ECC), parity, watchdog, or system clock, cause system exceptions.
- An interrupts event occurs 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)

Both SHARC+ cores feature a system event controller. The SEC features include the following:

- Comprehensive system event source management, including interrupt enable, fault enable, priority, core mapping, and source grouping
- A distributed programming model where each system event source control and all status fields are independent of each other
- Determinism where all system events have the same propagation delay and provide unique identification of a specific system event source
- A slave control port that provides access to all SEC registers for configuration, status, and interrupt and fault services
- Global locking that supports a register level protection model to prevent writes to locked registers
- Fault management including fault action configuration, time out, external indication, and system reset

Trigger Routing Unit (TRU)

The trigger routing unit (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

SECURITY FEATURES

The following sections describe the security features of the ADSP-SC57x/ADSP-2157x processors.

ARM TrustZone

The ADSP-SC57x processors provide TrustZone technology that is integrated into the ARM Cortex-A5 processors. The TrustZone technology enables a secure state that is extended throughout the system fabric.

Cryptographic Hardware Accelerators

The ADSP-SC57x/ADSP-2157x processors support standards-based hardware accelerated encryption, decryption, authentication, and true random number generation.

Support for the hardware accelerated cryptographic ciphers includes the following:

- AES in ECB, CBC, ICM, and CTR modes with 128-bit, 192-bit, 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
- ARC4 in stateful, stateless mode, up to 128-bit key

Downloaded from Arrow.com.

Support for the hardware accelerated hash functions includes the following:

- SHA-1
- SHA-2 with 224-bit and 256-bit digests
- HMAC transforms for SHA-1 and SHA-2
- MD5

Public key accelerator (PKA) is available to offload computation intensive public key cryptography operations.

Both a hardware-based nondeterministic random number generator and pseudorandom number generator are available.

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

Employ secure debug to allow only trusted users to access the system with debug tools.

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.

System Protection Unit (SPU)

The system protection unit (SPU) guards against accidental or unwanted access to an MMR space of the peripheral by providing a write protection mechanism. The user can choose and configure the protected peripherals as well as configure which of the four system MMR masters (two SHARC+ cores, memory DMA, 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 nonsecure as well as block access to secure resources from nonsecure masters.

System Memory Protection Unit (SMPU)

The system memory protection unit (SMPU) provides memory protection against read and/or write transactions to defined regions of memory. There are SMPU units in the ADSP-SC57x/ADSP-2157x processors for each memory space, except for SHARC L1 and SPI direct memory slave.

The SMPU is also part of the security infrastructure. It allows the user to protect against arbitrary read and/or write transactions and allows regions of memory to be defined as secure and prevent nonsecure masters from accessing those memory regions.

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.

SAFETY FEATURES

The ADSP-SC57x/ADSP-2157x processors are designed to support functional safety applications. While the level of safety is mainly dominated by the system concept, the following primitives are provided by the processors to build a robust safety concept.

Multiparity Bit Protected SHARC+ Core L1 Memories

In the SHARC+ core L1 memory space, whether SRAM or cache, multiple parity bits protect each word to detect the single event upsets that occur in all RAMs. Parity also protects the cache tags and BTB.

Parity Protected ARM L1 Cache

In the ARM Cortex-A5 L1 cache space, each word is protected by multiple parity bits to detect the single event upsets that occur in all RAMs. Parity also protects the cache tags.

Error Correcting Codes (ECC) Protected L2 Memories

Error correcting codes (ECC) correct single event upsets. A single error correct/double error detect (SEC/DED) code protects the L2 memory. By default, ECC is enabled, but it can be disabled on a per bank basis. Single-bit errors correct transparently. If enabled, dual-bit errors can issue a system event or fault. ECC protection is fully transparent to the user, even if L2 memory is read or written by 8-bit or 16-bit entities.

Parity-Protected Peripheral Memories

Parity protection is added to all peripheral memories:

- ASRC
- IIR
- FIR
- USB
- CAN
- CRYPTO
- EMAC
- SDIO
- MLB
- TRACE

Downloaded from Arrow.com.

Cyclic Redundant Code (CRC) Protected Memories

While parity bit and ECC protection mainly protect against random soft errors in L1 and L2 memory cells, the cyclic redundant code (CRC) engines can protect against systematic errors (pointer errors) and static content (instruction code) of L1, L2, and even Level 3 (L3) memories (DDR2, LPDDR). The processors feature two CRC engines that are embedded in the memory to memory DMA controllers.

CRC checksums can be calculated or compared automatically 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.

Signal Watchdogs

The eight general-purpose (GP) 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 detect undesired toggling or lack of toggling of system level signals.

System Event Controller (SEC)

Besides system events, the system event controller (SEC) further supports fault management including fault action configuration as timeout, internal indication by system interrupt, or external indication through the $\overline{\text{SYS_FAULT}}$ pin and system reset.

Memory Error Controller (MEC)

The memory error controller (MEC) manages memory parity/ECC errors and warnings from the cores and peripherals and sends out interrupts and triggers.

PROCESSOR PERIPHERALS

The following sections describe the peripherals of the ADSP-SC57x/ADSP-2157x processors.

Dynamic Memory Controller (DMC)

The 16-bit dynamic memory controller (DMC) interfaces to

- LPDDR1 (JESD209A) maximum frequency 200 MHz, DDRCLK (64 Mb to 2 Gb)
- DDR2 (JESD79-2E) maximum frequency 400 MHz, DDRCLK (256 Mb to 4 Gb)
- DDR3 (JESD79-3E) maximum frequency 450 MHz, DDRCLK (512 Mb to 8 Gb)
- DDR3L (1.5 V compatible only) maximum frequency 450 MHz, DDRCLK (512 Mb to 8 Gb)

See Table 8 for the DMC memory map.

Digital Audio Interface (DAI)

The processors support one mirrored digital audio interface (DAI) unit. The DAI can connect various peripherals to any of the DAI pins (DAI_PIN20–DAI_PIN01).

The application code makes these connections using the signal routing unit (SRU), shown in Figure 1.

The SRU is a matrix routing unit (or group of multiplexers) that enables the peripherals provided by the DAI to interconnect under software control. This functionality allows easy use of the DAI associated peripherals for a wider variety of applications by using a larger set of algorithms than is possible with nonconfigurable signal paths.

The DAI includes the peripherals described in the following sections (SPORTs, ASRC, S/PDIF, and PCG). DAI Pin Buffers 20 and 19 can change the polarity of the input signals. Most signals of the peripherals belonging to different DAIs cannot be interconnected, with few exceptions.

The DAI_PINx pin buffers can also be used as GPIO pins. DAI input signals allow the triggering of interrupts on the rising edge, falling edge, or both.

See the Digital Audio Interface (DAI) chapter of the ADSP-SC57x/ADSP-2157x SHARC+ Processor Hardware Reference for complete information on the use of the DAIs and SRUs.

Serial Port (SPORT)

The processors feature four synchronous full serial ports (SPORTs). These ports provide an inexpensive interface to a wide variety of digital and mixed-signal peripheral devices. These devices include Analog Devices AD19xx and ADAU19xx family of audio codecs, analog-to-digital converters (ADCs) and digital-to-analog converters (DACs). Two data lines, a clock, and frame sync make up the serial ports. The data lines can be programmed to either transmit or receive data and each data line has a dedicated DMA channel.

An individual full SPORT module consists of two independently configurable SPORT halves with identical functionality. Two bidirectional data lines—primary (0) and secondary (1)—are available per SPORT half and are configurable as either transmitters or receivers. Therefore, each SPORT half permits two unidirectional streams into or out of the same SPORT. This bidirectional functionality provides greater flexibility for serial communications. For full-duplex configuration, one half SPORT provides two transmit signals, while the other half SPORT provides the two receive signals. The frame sync and clock are shared.

Serial ports operate in the following six modes:

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

Asynchronous Sample Rate Converter (ASRC)

The asynchronous sample rate converter (ASRC) contains four ASRC blocks. It is the same core in the AD1896 192 kHz stereo asynchronous sample rate converter. The ASRC provides up to 140 dB signal-to-noise ratio (SNR). The ASRC block performs

Downloaded from Arrow.com.

synchronous or asynchronous sample rate conversion across independent stereo channels, without using internal processor resources. The ASRC blocks can also be configured to operate together to convert multichannel audio data without phase mismatches. Finally, the ASRC can clean up audio data from jittery clock sources such as the S/PDIF receiver.

S/PDIF-Compatible Digital Audio Receiver/Transmitter

The Sony/Philips Digital Interface Format (S/PDIF) is a standard audio data transfer format that allows the transfer of digital audio signals from one device to another without converting them to an analog signal. There is one S/PDIF transmit/receive block on the processor. The digital audio interface carries three types of information: audio data, nonaudio data (compressed data), and timing information.

The S/PDIF interface supports one stereo channel or compressed audio streams. The S/PDIF transmitter and receiver are AES3 compliant and support the sample rate from 24 KHz to 192 KHz. The S/PDIF receiver supports professional jitter standards.

The S/PDIF receiver/transmitter has no separate DMA channels. It receives audio data in serial format and converts it into a biphase encoded signal. The serial data input to the receiver/transmitter can be formatted as left justified, I^2S, or right justified with word widths of 16, 18, 20, or 24 bits. The serial data, clock, and frame sync inputs to the S/PDIF receiver/transmitter are routed through the signal routing unit (SRU). They can come from various sources, such as the SPORTs, external pins, and the precision clock generators (PCGs), and are controlled by the SRU control registers.

Precision Clock Generators (PCG)

The precision clock generators (PCG) consist of two units located in the DAI block. The PCG can generate a pair of signals (clock and frame sync) derived from a clock input signal (CLKIN, SCLK0, or DAI pin buffer). Both units are identical in functionality and operate independently of each other. The two signals generated by each unit are normally used as a serial bit clock/frame sync pair.

Enhanced Parallel Peripheral Interface (EPPI)

The processors provide an enhanced parallel peripheral interface (EPPI) that supports data widths up to 16 bits for the BGA package and 12 bits for the LQFP package. The EPPI supports direct connection to thin film transistor (TFT) LCD panels, parallel ADCs and DACs, video encoders and decoders, image sensor modules, and other general-purpose peripherals.

The features supported in the EPPI module include the following:

- Programmable data length of 8 bits, 10 bits, 12 bits, 14 bits, and 16 bits per clock.
- Various framed, nonframed, 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 decoding.
- Optional packing and unpacking of data to/from 32 bits from/to 8 bits and 16 bits. If packing/unpacking is enabled, configure endianness to change the order of packing/unpacking of bytes or words.
- RGB888 can be converted to RGB666 or RGB565 for transmit modes.
- Various deinterleaving/interleaving modes for receiving or transmitting 4:2:2 YCrCb data.
- Configurable LCD data enable output available on Frame Sync 3.

Universal Asynchronous Receiver/Transmitter (UART) Ports

The processors provide three full-duplex universal asynchronous receiver/transmitter (UART) ports, 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 as well as no parity, even parity, or odd parity.

Optionally, an additional address bit can be transferred to interrupt only addressed nodes in multidrop 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 first in, first out (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 interframe space.

Serial Peripheral Interface (SPI) Ports

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

The baseline SPI peripheral is a synchronous, 4-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 extra 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. DMA mode allows for transferring several words with minimal central processing unit (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,

Downloaded from Arrow.com.

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 (SPI_RDY) which flexibly controls the transfers.

The baud rate and clock phase and polarities of the SPI port are programmable. The port has integrated DMA channels for both transmit and receive data streams.

Link Port (LP)

Two 8-bit wide link ports (LPs) for the BGA package (one link port for the LQFP package) can connect to the link ports of other DSPs or peripherals. Link ports are bidirectional and have eight data lines, an acknowledge line, and a clock line.

ADC Control Module (ACM) Interface

The ADC control module (ACM) provides an interface that synchronizes the controls between the processors and an ADC. The analog-to-digital conversions are initiated by the processors, based on external or internal events.

The ACM allows for flexible scheduling of sampling instants and provides precise sampling signals to the ADC.

The ACM synchronizes the ADC conversion process, generating the ADC controls, the ADC conversion start signal, and other signals. The actual data acquisition from the ADC is done by an internal DAI routing of the ACM with the SPORT0 block.

The processors interface directly to many ADCs without any glue logic required.

Ethernet Media Access Controller (EMAC)

The processor features an ethernet media access controller (EMAC): 10/100/1000 AVB Ethernet with precision time protocol (IEEE 1588).

The processors can directly connect to a network through embedded fast EMAC that supports 10Base-T (10 Mb/sec), 100Base-T (100 Mb/sec) and 1000Base-T (1 Gb/sec) operations.

Some standard features of the EMAC are as follows:

- Support and MII/RMII/RGMII protocols for external PHYs.
- RGMII support for the BGA package only
- Full-duplex and half-duplex modes
- Media access management (in half-duplex operation)
- Flow control
- Station management, including the generation of MDC/MDIO frames for read/write access to PHY registers

Some advanced features of the EMAC include the following:

- Automatic checksum computation of IP header and IP payload fields of receive frames
- Independent 32-bit descriptor driven receive and transmit DMA channels
- Frame status delivery to memory through DMA, including frame completion semaphores for efficient buffer queue management in software
- Transmit DMA support for separate descriptors for MAC header and payload fields to eliminate buffer copy operations
- Convenient frame alignment modes
- 47 MAC management statistics counters with selectable clear on read behavior and programmable interrupts on half maximum value
- Advanced power management
- Magic packet detection and wakeup frame filtering
- Support for 802.3Q tagged VLAN frames
- Programmable MDC clock rate and preamble suppression

Audio Video Bridging (AVB) Support

The 10/100/1000 EMAC supports the following audio video bridging (AVB) features:

- Separate channels or queues for AV data transfer in 100 Mbps and 1000 Mbps modes)
- IEEE 802.1-Qav specified credit-based shaper (CBS) algorithm for the additional transmit channels
- Configuring up to two additional channels (Channel 1 and Channel 2) on the transmit and receive paths for AV traffic. Channel 0 is available by default and carries the legacy best effort Ethernet traffic on the transmit side.
- Separate DMA, transmit and receive FIFO for AVB latency class
- Programmable control to route received VLAN tagged non AV packets to channels or queues

Precision Time Protocol (PTP) IEEE 1588 Support

The IEEE 1588 standard is a precision clock synchronization protocol for networked measurement and control systems. The processors include hardware support for IEEE 1588 with an integrated precision time protocol synchronization engine (PTP_TSYNC).

This engine provides hardware assisted time stamping to improve the accuracy of clock synchronization between PTP nodes. The main features of the engine include the following:

- Support for both IEEE 1588-2002 and IEEE 1588-2008 protocol standards
- Hardware assisted time stamping capable of up to 12.5 ns resolution
- Lock adjustment
- Automatic detection of IPv4 and IPv6 packets, as well as PTP messages
- Multiple input clock sources (SCLK0, RGMII, RMII, MII clock, and external clock)
- Programmable pulse per second (PPS) output
- Auxiliary snapshot to time stamp external events

Downloaded from Arrow.com.

Controller Area Network (CAN)

There are two controller area network (CAN) modules. 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 the 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 the 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
- Interrupts, including transmit and receive complete, error, and global

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

Timers

The processors include several timers that are described in the following sections.

General-Purpose (GP) Timers (TIMER)

There is one general-purpose (GP) timer unit, providing eight GP programmable timers. Each timer has an external pin that can be configured either as 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 TM_TMR[n] pins, an external TM_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 autobaud 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 via the TRU (for instance, to signal a fault). Each timer can also be started and/or stopped by any TRU master without core intervention.

Watchdog Timer (WDT)

Three on-chip software watchdog timers (WDT) can be used by the ARM Cortex-A5 and/or SHARC+ cores. A software watchdog can improve system availability by forcing the processors to a known state, via a general-purpose interrupt, or a fault, 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, protecting the system from remaining in an unknown state where software that normally resets the timer stops running due to an external noise condition or software error.

General-Purpose Counters (CNT)

A 32-bit counter (CNT) 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 input the push button signal of thumbwheel devices. All three CNT0 pins have a programmable debouncing circuit.

Internal signals forwarded to a GP timer enable the 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.

Housekeeping Analog-to-Digital Converter (HADC)

The housekeeping analog-to-digital converter (HADC) provides a general-purpose, multichannel successive approximation ADC. It supports the following set of features:

- 12-bit ADC core with built in sample and hold.
- Eight single-ended input channels for the BGA package; four single-ended input channels for the LQFP package.
- 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.
- Autosequencing capability with up to eight autoconversions in a single session. Each conversion can be programmed to select one to eight input channels.
- Six data registers (individually addressable) to store conversion values

USB 2.0 On the Go (OTG) Dual-Role Device Controller (BGA Only)

The USB supports high speed/full speed/low speed (HS/FS/LS) USB2.0 on the go (OTG).

The USB 2.0 OTG dual-role device controller provides a low cost connectivity solution 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.

Downloaded from Arrow.com.

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 (PLL) with programmable multipliers to generate the necessary internal clocking frequency for the USB.

Media Local Bus (MediaLB)

The automotive model has a Microchip MediaLB (MLB) slave interface that allows the processors to function as a media local bus device. It includes support for both 3-pin and 6-pin media local bus protocols. The MLB 3-pin configuration supports speeds up to 1024 × FS. The MLB 6-pin configuration supports speed of 2048 × FS. The MLB also supports up to 64 logical channels with up to 468 bytes of data per MLB frame.

The MLB interface supports MOST25, MOST50, and MOST150 data rates and operates in slave mode only.

2-Wire Controller Interface (TWI)

The processors include three 2-wire interface (TWI) modules that provide 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 400 kb/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.

General-Purpose I/O (GPIO)

Each general-purpose port pin can be individually controlled by manipulating 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 have a write one to modify mechanism that 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 processors. 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 the rising edge or both the rising and falling edges of the signal are significant.

Pin Interrupts

Every port pin on the processors can request interrupts in either an edge sensitive or a level sensitive manner with programmable polarity. Interrupt functionality is decoupled from GPIO operation. Five system level interrupt channels (PINT0–PINT4) 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.

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 MSI controller has the following features:

- 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 4.3 embedded NAND flash devices
- An 11-signal external interface with clock, command, optional interrupt, and up to eight data lines
- Integrated DMA controller
- Card interface clock generation in the clock distribution unit (CDU)
- SDIO interrupt and read wait features

SYSTEM ACCELERATION

The following sections describe the system acceleration blocks of the ADSP-SC57x/ADSP-2157x processors.

Finite Impulse Response (FIR) Accelerator

The finite impulse response (FIR) accelerator consists of a 1024 word coefficient memory, a 1024 word deep delay line for the data, and four MAC units. A controller manages the accelerator. The FIR accelerator runs at the peripheral clock frequency. The FIR accelerator can access all memory spaces and can run concurrently with the other accelerators on the processor.

Infinite Impulse Response (IIR) Accelerator

The infinite impulse response (IIR) accelerator consists of a 1440 word coefficient memory for storage of biquad coefficients, a data memory for storing the intermediate data, and one MAC unit. A controller manages the accelerator. The IIR accelerator runs at the peripheral clock frequency. The IIR accelerator can access all memory spaces and run concurrently with the other accelerators on the processor.

Downloaded from Arrow.com.

SYSTEM DESIGN

The following sections provide an introduction to system design features and power supply issues.

Clock Management

The processors provide three operating modes, each with a different performance and power profile. Control of clocking to each of the processor peripherals reduces power consumption. The processors do not support any low power operation modes. Control of clocking to each of the processor peripherals can reduce the power consumption.

Reset Control Unit (RCU)

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 default values and functional units are idle. Exiting a full system reset starts with the core 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 put the system into an undefined state or causes resources to stall. This is particularly important when the core resets (programs must ensure that there is no pending system activity involving the core when it is reset).

From a system perspective, reset is defined by both the reset target and the reset source.

The reset target is defined as the following:

- System reset—all functional units except the RCU are set to default states.
- Hardware reset—all functional units are set to default states without exception. History is lost.
- Core only reset— affects the core only. When in reset state, the core is not accessed by any bus master.

The reset source is defined as the following:

- System reset—can be triggered by software (writing to the RCU_CTL register) or by another functional unit such as the dynamic power management (DPM) unit or any of the SEC, TRU, or emulator inputs.
- Hardware reset—the $\overline{\text{SYS_HWRST}}$ input signal asserts active (pulled down).
- Core only reset—affects only the core. The core is not accessed by any bus master when in reset state.
- Trigger request (peripheral).

Clock Generation Unit (CGU)

The ADSP-SC57x/ADSP-2157x processors support two independent PLLs. Each PLL is part of a clock generation unit (CGU); see Figure 7. Each CGU can be either driven externally by the same clock source or each can be driven by separate sources. This provides flexibility in determining the internal clocking frequencies for each clock domain.

Frequencies generated by each CGU are derived from a common multiplier with different divider values available for each output.

The 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, the DDR1/DDR2/DDR3 clock (DCLK), and the output clock (OCLK). For more information on clocking, see the ADSP-SC57x/ADSP-2157x SHARC+ Processor Hardware Reference.

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 it transitions smoothly from the current conditions to the new conditions.

System Crystal Oscillator and USB Crystal Oscillator

The processor can be clocked by an external crystal (see Figure 6), a sine wave input, or a buffered, shaped clock derived from an external clock oscillator. If using an external clock, it must 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_CLKINx pin and the USB_CLKIN pin of the processor. When using an external clock, the SYS_XTALx pin and the USB_XTAL pin must be left unconnected. Alternatively, because the processor includes an on-chip oscillator circuit, an external crystal can be used.

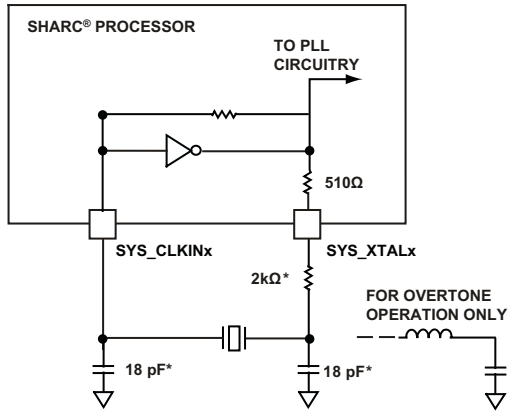


Figure 6. External Crystal Connection

For fundamental frequency operation, use the circuit shown in Figure 6. A parallel resonant, fundamental frequency, microprocessor grade crystal is connected across the SYS_CLKINx pin and the SYS_XTALx pin. The on-chip resistance between the SYS_CLKINx pin and the SYS_XTALx pin is in the 500 kΩ range. Further parallel resistors are typically not recommended.

Downloaded from Arrow.com.

The two capacitors and the series resistor, shown in Figure 6, fine tune phase and amplitude of the sine frequency. The capacitor and resistor values shown in Figure 6 are typical values only. The capacitor values are dependent upon the load capacitance recommendations of the crystal manufacturer and the physical layout of the printed circuit board (PCB). The resistor value depends on the drive level specified by the crystal manufacturer. The user must verify the customized values based on careful investigations on multiple devices over the required temperature range.

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, shown in Figure 6. A design procedure for third overtone operation is discussed in detail in "Using Third Overtone Crystals with the ADSP-218x DSP" (EE-168). The same recommendations can be used for the USB crystal oscillator.

Clock Distribution Unit (CDU)

The two CGUs each provide outputs which feed a clock distribution unit (CDU). The clock outputs CLKO0–CLKO9 are connected to various targets. For more information, refer to the ADSP-SC57x/ADSP-2157x SHARC+ Processor Hardware Reference.

Power-Up

SYS_XTALx oscillations (SYS_CLKINx) start when power is applied to the VDD_EXT pins. The rising edge of $\overline{\text{SYS_HWRST}}$ starts on-chip PLL locking (PLL lock counter). The deassertion must apply only if all voltage supplies and SYS_CLKINx oscillations are valid (refer to the Power-Up Reset Timing section).

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_ CLKIN0 input. Refer to the ADSP-SC57x/ADSP-2157x SHARC+ Processor Hardware Reference to change the default mapping of clocks.

Booting

The processors have several mechanisms for automatically loading internal and external memory after a reset. The boot mode is defined by the SYS_BMODE[n] input pins. There are two categories of boot modes. In master boot mode, the processors actively load data from serial memories. In slave boot modes, the processors receive data from external host devices.

The boot modes are shown in Table 9. These modes are implemented by the SYS_BMODE[n] bits of the reset configuration register and are sampled during power-on resets and software initiated resets.

In the ADSP-SC57x processors, the ARM Cortex-A5 (Core 0) controls the boot process, including loading all internal and external memory. Likewise, in the ADSP-2157x processors, the SHARC+ (Core 1) controls the boot function. The option for secure boot is available on all models.

Table 9. Boot Modes

SYS_BMODE[n] Setting[1, 2]	Boot Mode
000	No boot
001	SPI2 master
010	SPI2 slave
011	UART0 slave
100	Reserved
101	Reserved
110	Link0 slave

[1] SYS_BMODE2 pin is applicable only for the BGA package.
[2] Link0 slave boot is supported only on the BGA package.

Thermal Monitoring Unit (TMU)

The thermal monitoring unit (TMU) provides on-chip temperature measurement for applications that require substantial power consumption. The TMU is integrated into the processor die and digital infrastructure using an MMR-based system access to measure the die temperature variations in real-time.

TMU features include the following:

- On-chip temperature sensing
- Programmable over temperature and under temperature limits
- Programmable conversion rate
- Programmable clock source selection to run the sensor off an independent local clock
- Averaging feature available

Power Supplies

The processors have separate power supply connections for

- Internal (VDD_INT)
- External (VDD_EXT)
- USB (VDD_USB)
- HADC/TMU (VDD_HADC)
- DMC (VDD_DMC)

All power supplies must meet the specifications provided in Operating Conditions section. All external supply pins must be connected to the same power supply.

Power Management

As shown in Table 10, the processors support four different 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 (see the Specifications section for processor operating conditions). If the feature or the peripheral is not used, refer to Table 25.

Downloaded from Arrow.com.

Table 10. Power Domains

Power Domain	V_{DD} Range
All internal logic	V_{DD_INT}
DDR3/DDR2/LPDDR	V_{DD_DMC}
USB	V_{DD_USB}
HADC/TMU	V_{DD_HADC}
All other I/O (includes SYS, JTAG, and ports pins)	V_{DD_EXT}

The power dissipated by a processor is largely a function of the 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.

Target Board JTAG Emulator Connector

The Analog Devices DSP tools product line of JTAG emulators uses the IEEE 1149.1 JTAG test access port of the processors to monitor and control the target board processor during emulation. The Analog Devices DSP tools product line of JTAG emulators provides emulation at full processor speed, allowing inspection and modification of memory, registers, and processor stacks. The processor JTAG interface ensures the emulator does not affect target system loading or timing.

For information on JTAG emulator operation, see the appropriate emulator hardware user's guide at SHARC Processors Software and Tools.

SYSTEM DEBUG

The processors include various features that allow easy system debug. These are described in the following sections.

System Watchpoint Unit (SWU)

The system watchpoint unit (SWU) is a single module that connects to a single system bus and provides 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 (for example, interrupt and trigger) outputs.

Debug Access Port (DAP)

Debug access port (DAP) provides IEEE 1149.1 JTAG interface support through the JTAG debug. 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 an integrated development environment (CrossCore® Embedded Studio), evaluation products, emulators, and a variety of software add ins.

Integrated Development Environments (IDEs)

For C/C++ software writing and editing, code generation, and debug support, Analog Devices offers the CrossCore Embedded Studio integrated development environment (IDE).

CrossCore Embedded Studio is based on the Eclipse framework. Supporting most Analog Devices processor families, it is the IDE of choice for 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. Various EZ-Extenders® are also available, which are daughter cards that deliver additional specialized functionality, including audio and video processing. For more information visit www.analog.com.

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(s), a USB cable, and a power supply. The USB controller on the EZ-KIT Lite board connects to the USB port of the user PC, enabling the chosen IDE evaluation suite to emulate the on-board processor in circuit. This permits users 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.

Software Add Ins for CrossCore Embedded Studio

Analog Devices offers software add ins which seamlessly integrate with CrossCore Embedded Studio to extend the 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.

Downloaded from Arrow.com.

Board Support Packages (BSPs) 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.

Middleware Packages

Analog Devices 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/ucos2
- www.analog.com/ucos3
- www.analog.com/ucfs
- www.analog.com/ucusbd
- www.analog.com/ucusbh
- 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.

Designing an Emulator-Compatible DSP Board (Target)

For embedded system test and debug, Analog Devices provides a family of emulators. On each JTAG DSP, Analog Devices supplies an IEEE 1149.1 JTAG test access port (TAP). In circuit emulation is facilitated by use of this JTAG interface. The emulator accesses the internal features of the processor via the TAP, allowing the developer to load code, set breakpoints, and view variables, memory, and registers.

The processor must be halted to send data and commands, but once an operation is completed by the emulator, the DSP system is set to run at full speed with no impact on system timing. The emulators require the target board to include a header that supports connection of the JTAG port of the DSP to the emulator.

For details on target board design issues including mechanical layout, single processor connections, signal buffering, signal termination, and emulator pod logic, see "Analog Devices JTAG Emulation Technical Reference" (EE-68).

ADDITIONAL INFORMATION

This data sheet provides a general overview of the ADSP-SC57x/ADSP-2157x architecture and functionality. For detailed information on the core architecture and instruction set, refer to the SHARC+ Core Programming Reference.

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 (www.analog.com\circuits) provides the following:

- 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

Downloaded from Arrow.com.

ADSP-SC57x/ADSP-2157x DETAILED SIGNAL DESCRIPTIONS

Table 11 provides a detailed description of each pin.

Table 11. ADSP-SC57x/ADSP-2157x Detailed Signal Descriptions

Signal Name	Direction	Description
ACM_A[n]	Output	**ADC Control Signals.** Function varies by mode.
ACM_T[n]	Input	**External Trigger n.** Input for external trigger events.
C1_FLG[n]	Output	**SHARC Core 1 Flag Pin.**
C2_FLG[n]	Output	**SHARC Core 2 Flag Pin.**
CAN_RX	Input	**Receive.** Typically an external CAN transceiver RX output.
CAN_TX	Output	**Transmit.** Typically an external CAN transceiver 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.
DAI_PIN[nn]	InOut	**Pin n.** The digital applications interface (DAI0) connects various peripherals to any of the DAI0_PINxx pins. Programs make these connections using the signal routing unit (SRU).
DMC_A[nn]	Output	**Address n.** Address bus.
DMC_BA[n]	Output	**Bank Address n.** Defines which internal bank an activate, read, write or precharge command is applied to on the dynamic memory. Bank Address n also defines which mode registers (MR, EMR, EMR2, and/or EMR3) load 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 CKE input of the dynamic memory.
$\overline{\text{DMC_CS[n]}}$	Output	**Chip Select n.** Commands are recognized by the memory only when this signal is asserted.
DMC_DQ[nn]	InOut	**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	InOut	**Data Strobe for Lower Byte.** DMC_DQ07:DMC_DQ00 data strobe. Output with write data. Input with read data. Can be single-ended or differential depending on register settings.
$\overline{\text{DMC_LDQS}}$	InOut	**Data Strobe for Lower Byte (Complement).** Complement of DMC_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 or 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.
$\overline{\text{DMC_RESET}}$	Output	**Reset (DDR3 Only).**
DMC_RZQ	InOut	**External Calibration Resistor Connection.**
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	InOut	**Data Strobe for Upper Byte.** DMC_DQ15:DMC_DQ08 data strobe. Output with write data. Input with read data. Can be single-ended or differential depending on register settings.

Downloaded from Arrow.com.

Table 11. ADSP-SC57x/ADSP-2157x Detailed Signal Descriptions (Continued)

Signal Name	Direction	Description
$\overline{\text{DMC_UDQS}}$	InOut	**Data Strobe for Upper Byte (Complement).** Complement of DMC_UDQS. Not used in single-ended mode.
DMC_VREF	Input	**Voltage Reference.** Connects to half of the VDD_DMC voltage. Applies to the DMC0_VREF pin.
$\overline{\text{DMC_WE}}$	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.
ETH_COL	Input	**MII Collision Detect.** Collision detect input signal valid only in MII.
ETH_CRS	Input	**MII Carrier Sense.** Asserted by the PHY when either the transmit or receive medium is not idle. Deasserted when both are idle. This signal is not used in RMII/RGMII modes.
ETH_MDC	Output	**Management Channel Clock.** Clocks the MDC input of the PHY for RMII/RGMII.
ETH_MDIO	InOut	**Management Channel Serial Data.** Bidirectional data bus for PHY control for RMII/RGMII.
ETH_PTPAUXIN[n]	Input	**PTP Auxiliary Trigger Input.** Assert this signal to take an auxiliary snapshot of the time and store it in the auxiliary time stamp FIFO.
ETH_PTPCLKIN[n]	Input	**PTP Clock Input.** Optional external PTP clock input.
ETH_PTPPPS[n]	Output	**PTP Pulse Per Second Output.** When the advanced time stamp feature enables, this signal is asserted based on the PPS mode selected. Otherwise, this signal is asserted every time the seconds counter is incremented.
ETH_RXCLK_REFCLK	InOut	**RXCLK (10/100/1000) or REFCLK (10/100).**
ETH_RXCTL_RXDV	InOut	**RXCTL (10/100/1000) or RXDV (10/100).** In RGMII mode, RX_CTL multiplexes receive data valid and receiver error. In RMII mode, RXDV is carrier sense and receive data valid (CRS_DV), multiplexed on alternating clock cycles. In MII mode, RXDV is receive data valid (RX_DV), asserted by the PHY when the data on ETH_RXD[n] is valid.
ETH_RXD[n]	Input	**Receive Data n.** Receive data bus.
ETH_RXERR	Input	**Receive Error.**
ETH_TXCLK	Input	**Reference Clock.** Externally supplied Ethernet clock
ETH_TXCTL_TXEN	InOut	**TXCTL (10/100/1000) or TXEN (10/100).**
ETH_TXD[n]	Output	**Transmit Data n.** Transmit data bus.
HADC_EOC_DOUT	Output	**End of Conversion/Serial Data Out.** Transitions high for one cycle of the HADC internal clock at the end of every conversion. Alternatively, HADC serial data out can be seen by setting the appropriate bit in HADC_CTL.
HADC_VIN[n]	Input	**Analog Input at Channel n.** Analog voltage inputs for digital conversion.
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.
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.
LP_ACK	InOut	**Acknowledge.** Provides handshaking. When the link port is configured as a receiver, ACK is an output. When the link port is configured as a transmitter, ACK is an input.
LP_CLK	InOut	**Clock.** When the link port is configured as a receiver, CLK is an input. When the link port is configured as a transmitter, CLK is an output.
LP_D[n]	InOut	**Data n.** Data bus. Input when receiving, output when transmitting.
MLB_CLK	InOut	**Single Ended Clock.**
MLB_CLKN	InOut	**Differential Clock (–).**
MLB_CLKOUT	InOut	**Single Ended Clock Out.**
MLB_CLKP	InOut	**Differential Clock (+).**
MLB_DAT	InOut	**Single Ended Data.**

Downloaded from Arrow.com.

Table 11. ADSP-SC57x/ADSP-2157x Detailed Signal Descriptions (Continued)

Signal Name	Direction	Description
MLB_DATN	InOut	**Differential Data (–).**
MLB_DATP	InOut	**Differential Data (+).**
MLB_SIG	InOut	**Single Ended Signal.**
MLB_SIGN	InOut	**Differential Signal (–).**
MLB_SIGP	InOut	**Differential Signal (+).**
$\overline{\text{MSI_CD}}$	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	InOut	**Command.** Sends commands to and receive responses from the connected device.
MSI_D[n]	InOut	**Data n.** Bidirectional data bus.
$\overline{\text{MSI_INT}}$	Input	**eSDIO Interrupt Input.** Used only for eSDIO. Connects to an eSDIO card interrupt output. An interrupt can be sampled even when the MSI clock to the card is switched off.
PPI_CLK	InOut	**Clock.** Input in external clock mode, output in internal clock mode.
PPI_D[nn]	InOut	**Data n.** Bidirectional data bus.
PPI_FS1	InOut	**Frame Sync 1 (HSYNC).** Behavior depends on EPPI mode. See the EPPI chapter of the ADSP-SC57x/ADSP-2157x SHARC+ Processor Hardware Reference for more details.
PPI_FS2	InOut	**Frame Sync 2 (VSYNC).** Behavior depends on EPPI mode. See the EPPI chapter of the ADSP-SC57x/ADSP-2157x SHARC+ Processor Hardware Reference for more details.
PPI_FS3	InOut	**Frame Sync 3 (FIELD).** Behavior depends on EPPI mode. See the EPPI chapter of the ADSP-SC57x/ADSP-2157x SHARC+ Processor Hardware Reference for more details.
P_[nn]	InOut	**Position n.** General-purpose input/output. See the GP Ports chapter of the ADSP-SC57x/ADSP-2157x SHARC+ Processor Hardware Reference for more details.
SPI_CLK	InOut	**Clock.** Input in slave mode, output in master mode.
SPI_D2	InOut	**Data 2.** Transfers serial data in quad mode. Open-drain when ODM mode is enabled.
SPI_D3	InOut	**Data 3.** Transfers serial data in quad mode. Open-drain when ODM mode is enabled.
SPI_MISO	InOut	**Master In, Slave Out.** Transfers serial data. Operates in the same direction as SPI_MOSI in dual and quad modes. Open-drain when ODM mode is enabled.
SPI_MOSI	InOut	**Master Out, Slave In.** Transfers serial data. Operates in the same direction as SPI_MISO in dual and quad modes. Open-drain when ODM mode is enabled.
SPI_RDY	InOut	**Ready.** Optional flow signal. Output in slave mode, input in master mode.
$\overline{\text{SPI_SEL[n]}}$	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	InOut	**Channel A Clock.** Data and frame sync are driven or sampled with respect to this clock. This signal can be either internally or externally generated.
SPT_AD0	InOut	**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	InOut	**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	InOut	**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.
SPT_BCLK	InOut	**Channel B Clock.** Data and frame sync are driven or sampled with respect to this clock. This signal can be either internally or externally generated.
SPT_BD0	InOut	**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.

Downloaded from Arrow.com.

Table 11. ADSP-SC57x/ADSP-2157x Detailed Signal Descriptions (Continued)

Signal Name	Direction	Description
SPT_BD1	InOut	**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	InOut	**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 multichannel transmit mode. It is asserted during enabled slots.
SYS_BMODE[n]	Input	**Boot Mode Control n.** Selects the boot mode of the processor.
SYS_CLKIN0	Input	**Clock/Crystal Input.**
SYS_CLKIN1	Input	**Clock/Crystal Input**.
SYS_CLKOUT	Output	**Processor Clock Output.** Outputs internal clocks. Clocks may be divided down. See the CGU chapter of the ADSP-SC57x/ADSP-2157x SHARC+ Processor Hardware Reference for more details.
SYS_FAULT	InOut	**Active-High Fault Output.** Indicates internal faults or senses external faults depending on the operating mode.
$\overline{\text{SYS_FAULT}}$	InOut	**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.
$\overline{\text{SYS_RESOUT}}$	Output	**Reset Output.** Indicates the device is in the reset state.
SYS_XTAL0	Output	**Crystal Output.**
SYS_XTAL1	Output	**Crystal Output.**
TM_ACI[n]	Input	**Alternate Capture Input n.** Provides an additional input for WIDCAP, WATCHDOG, and PININT modes.
TM_ACLK[n]	Input	**Alternate Clock n.** Provides an additional time base for an individual timer.
TM_CLK	Input	**Clock.** Provides an additional global time base for all GP timers.
TM_TMR[n]	InOut	**Timer n.** The main input/output signal for each timer.
TRACE_CLK	Output	**Trace Clock.** Clock output.
TRACE_D[nn]	Output	**Trace Data n.** Unidirectional data bus.
TWI_SCL	InOut	**Serial Clock.** Clock output when master, clock input when slave.
TWI_SDA	InOut	**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.** Receives input. Typically connects to a transceiver that meets the electrical requirements of the device being communicated with.
$\overline{\text{UART_TX}}$	Output	**Transmit.** Transmits 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.
USB_DM	InOut	**Data –.** Bidirectional differential data line.
USB_DP	InOut	**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). 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. Can be configured as open-drain. Polarity is configurable as well.
USB_VBUS	InOut	**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.

Downloaded from Arrow.com.

400-BALL CSP_BGA SIGNAL DESCRIPTIONS

The processor pin definitions are shown in Table 12 for the 400-ball CSP_BGA package. The columns in this table provide the following information:

- The signal name column includes the signal name for every pin and the GPIO multiplexed pin function, where applicable.
- The description column provides a descriptive name for each signal.
- The port column shows whether or not a signal is multiplexed with other signals on a GPIO port pin.
- The pin name column 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 GPIO pin).
- The DAI pins and their associated signal routing units (SRUs) connect inputs and outputs of the DAI peripherals (SPORT, ASRC, S/PDIF, and PCG). See the Digital Audio Interface (DAI) chapter of the ADSP-SC57x/ADSP-2157x SHARC+ Processor Hardware Reference for complete information on the use of the DAI and SRUs.

Table 12. ADSP-SC57x/ADSP-2157x 400-Ball CSP_BGA Signal Descriptions

Signal Name	Description	Port	Pin Name
ACM0_A0	ACM0 ADC Control Signals	F	PF_11
ACM0_A1	ACM0 ADC Control Signals	C	PC_14
ACM0_A2	ACM0 ADC Control Signals	C	PC_15
ACM0_A3	ACM0 ADC Control Signals	A	PA_14
ACM0_A4	ACM0 ADC Control Signals	B	PB_01
ACM0_T0	ACM0 External Trigger n	A	PA_15
C1_FLG0	SHARC Core 1 Flag Pin	E	PE_13
C1_FLG1	SHARC Core 1 Flag Pin	E	PE_01
C1_FLG2	SHARC Core 1 Flag Pin	F	PF_04
C1_FLG3	SHARC Core 1 Flag Pin	D	PD_06
C2_FLG0	SHARC Core 2 Flag Pin	B	PB_00
C2_FLG1	SHARC Core 2 Flag Pin	C	PC_14
C2_FLG2	SHARC Core 2 Flag Pin	F	PF_11
C2_FLG3	SHARC Core 2 Flag Pin	E	PE_15
CAN0_RX	CAN0 Receive	C	PC_12
CAN0_TX	CAN0 Transmit	C	PC_13
CAN1_RX	CAN1 Receive	C	PC_14
CAN1_TX	CAN1 Transmit	C	PC_15
CNT0_DG	CNT0 Count Down and Gate	D	PD_08
CNT0_UD	CNT0 Count Up and Direction	E	PE_13
CNT0_ZM	CNT0 Count Zero Marker	D	PD_07
DAI0_PIN01	DAI0 Pin 1	Not Muxed	DAI0_PIN01
DAI0_PIN02	DAI0 Pin 2	Not Muxed	DAI0_PIN02
DAI0_PIN03	DAI0 Pin 3	Not Muxed	DAI0_PIN03
DAI0_PIN04	DAI0 Pin 4	Not Muxed	DAI0_PIN04
DAI0_PIN05	DAI0 Pin 5	Not Muxed	DAI0_PIN05
DAI0_PIN06	DAI0 Pin 6	Not Muxed	DAI0_PIN06
DAI0_PIN07	DAI0 Pin 7	Not Muxed	DAI0_PIN07
DAI0_PIN08	DAI0 Pin 8	Not Muxed	DAI0_PIN08
DAI0_PIN09	DAI0 Pin 9	Not Muxed	DAI0_PIN09
DAI0_PIN10	DAI0 Pin 10	Not Muxed	DAI0_PIN10
DAI0_PIN11	DAI0 Pin 11	Not Muxed	DAI0_PIN11
DAI0_PIN12	DAI0 Pin 12	Not Muxed	DAI0_PIN12
DAI0_PIN13	DAI0 Pin 13	Not Muxed	DAI0_PIN13
DAI0_PIN14	DAI0 Pin 14	Not Muxed	DAI0_PIN14
DAI0_PIN15	DAI0 Pin 15	Not Muxed	DAI0_PIN15

Downloaded from Arrow.com.

Table 12. ADSP-SC57x/ADSP-2157x 400-Ball CSP_BGA Signal Descriptions (Continued)

Signal Name	Description	Port	Pin Name
DAI0_PIN16	DAI0 Pin 16	Not Muxed	DAI0_PIN16
DAI0_PIN17	DAI0 Pin 17	Not Muxed	DAI0_PIN17
DAI0_PIN18	DAI0 Pin 18	Not Muxed	DAI0_PIN18
DAI0_PIN19	DAI0 Pin 19	Not Muxed	DAI0_PIN19
DAI0_PIN20	DAI0 Pin 20	Not Muxed	DAI0_PIN20
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_A14	DMC0 Address 14	Not Muxed	DMC0_A14
DMC0_A15	DMC0 Address 15	Not Muxed	DMC0_A15
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
$\overline{\text{DMC0_CK}}$	DMC0 Clock (complement)	Not Muxed	$\overline{\text{DMC0_CK}}$
DMC0_CKE	DMC0 Clock enable	Not Muxed	DMC0_CKE
$\overline{\text{DMC0_CS0}}$	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
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	$\overline{\text{DMC0_LDQS}}$

Downloaded from Arrow.com.

Table 12. ADSP-SC57x/ADSP-2157x 400-Ball CSP_BGA Signal Descriptions (Continued)

Signal Name	Description	Port	Pin Name
DMC0_ODT	DMC0 On die termination	Not Muxed	DMC0_ODT
$\overline{\text{DMC0_RAS}}$	DMC0 Row Address Strobe	Not Muxed	$\overline{\text{DMC0_RAS}}$
$\overline{\text{DMC0_RESET}}$	DMC0 Reset (DDR3 only)	Not Muxed	$\overline{\text{DMC0_RESET}}$
DMC0_RZQ	DMC0 External calibration resistor connection	Not Muxed	DMC0_RZQ
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	$\overline{\text{DMC0_WE}}$
ETH0_COL	EMAC0 MII Collision detect	C	PC_06
ETH0_CRS	EMAC0 Carrier Sense/RMII Receive Data Valid	B	PB_01
ETH0_MDC	EMAC0 Management Channel Clock	A	PA_11
ETH0_MDIO	EMAC0 Management Channel Serial Data	A	PA_10
ETH0_PTPAUXIN0	EMAC0 PTP Auxiliary Trigger Input 0	D	PD_14
ETH0_PTPAUXIN1	EMAC0 PTP Auxiliary Trigger Input 1	D	PD_15
ETH0_PTPAUXIN2	EMAC0 PTP Auxiliary Trigger Input 2	F	PF_06
ETH0_PTPAUXIN3	EMAC0 PTP Auxiliary Trigger Input 3	F	PF_07
ETH0_PTPCLKIN0	EMAC0 PTP Clock Input 0	F	PF_05
ETH0_PTPPPS0	EMAC0 PTP Pulse Per Second Output 0	A	PA_09
ETH0_PTPPPS1	EMAC0 PTP Pulse Per Second Output 1	D	PD_08
ETH0_PTPPPS2	EMAC0 PTP Pulse Per Second Output 2	E	PE_00
ETH0_PTPPPS3	EMAC0 PTP Pulse Per Second Output 3	E	PE_01
ETH0_RXCLK_REFCLK	EMAC0 RXCLK (10/100/1000) or REFCLK (10/100)	B	PB_00
ETH0_RXCTL_RXDV	EMAC0 RXCTL (10/100/1000) or CRS (10/100)	B	PB_01
ETH0_RXD0	EMAC0 Receive Data 0	A	PA_13
ETH0_RXD1	EMAC0 Receive Data 1	A	PA_12
ETH0_RXD2	EMAC0 Receive Data 2	A	PA_14
ETH0_RXD3	EMAC0 Receive Data 3	A	PA_15
ETH0_RXERR	EMAC0 Receive Error	B	PB_03
ETH0_TXCLK	EMAC0 Transmit Clock	B	PB_04
ETH0_TXCTL_TXEN	EMAC0 TXCTL (10/100/1000) or TXEN (10/100)	B	PB_09
ETH0_TXD0	EMAC0 Transmit Data 0	B	PB_07
ETH0_TXD1	EMAC0 Transmit Data 1	B	PB_08
ETH0_TXD2	EMAC0 Transmit Data 2	B	PB_06
ETH0_TXD3	EMAC0 Transmit Data 3	B	PB_05
HADC0_EOC_DOUT	HADC0 End of Conversion/Serial Data Out	D	PD_09
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_VIN4	HADC0 Analog Input at channel 4	Not Muxed	HADC0_VIN4
HADC0_VIN5	HADC0 Analog Input at channel 5	Not Muxed	HADC0_VIN5
HADC0_VIN6	HADC0 Analog Input at channel 6	Not Muxed	HADC0_VIN6
HADC0_VIN7	HADC0 Analog Input at channel 7	Not Muxed	HADC0_VIN7
HADC0_VREFN	HADC0 Ground Reference for ADC	Not Muxed	HADC0_VREFN
HADC0_VREFP	HADC0 External Reference for ADC	Not Muxed	HADC0_VREFP
JTG_TCK	JTAG Clock	Not Muxed	JTG_TCK
JTG_TDI	JTAG Serial Data In	Not Muxed	JTG_TDI

Downloaded from Arrow.com.

Table 12. ADSP-SC57x/ADSP-2157x 400-Ball CSP_BGA Signal Descriptions (Continued)

Signal Name	Description	Port	Pin Name
JTG_TDO	JTAG Serial Data Out	Not Muxed	JTG_TDO
JTG_TMS	JTAG Mode Select	Not Muxed	JTG_TMS
$\overline{\text{JTG_TRST}}$	JTAG Reset	Not Muxed	$\overline{\text{JTG_TRST}}$
LP0_ACK	LP0 Acknowledge	E	PE_03
LP0_CLK	LP0 Clock	E	PE_02
LP0_D0	LP0 Data 0	E	PE_04
LP0_D1	LP0 Data 1	E	PE_05
LP0_D2	LP0 Data 2	E	PE_06
LP0_D3	LP0 Data 3	E	PE_07
LP0_D4	LP0 Data 4	E	PE_08
LP0_D5	LP0 Data 5	E	PE_09
LP0_D6	LP0 Data 6	E	PE_10
LP0_D7	LP0 Data 7	E	PE_11
LP1_ACK	LP1 Acknowledge	B	PB_01
LP1_CLK	LP1 Clock	B	PB_03
LP1_D0	LP1 Data 0	D	PD_10
LP1_D1	LP1 Data 1	D	PD_11
LP1_D2	LP1 Data 2	D	PD_12
LP1_D3	LP1 Data 3	D	PD_13
LP1_D4	LP1 Data 4	D	PD_14
LP1_D5	LP1 Data 5	D	PD_15
LP1_D6	LP1 Data 6	A	PA_09
LP1_D7	LP1 Data 7	D	PD_09
MLB0_CLK	MLB0 Single-Ended Clock	B	PB_06
MLB0_CLKN	MLB0 Differential Clock (–)	Not Muxed	MLB0_CLKN
MLB0_CLKOUT	MLB0 Single-Ended Clock Out	B	PB_03
MLB0_CLKP	MLB0 Differential Clock (+)	Not Muxed	MLB0_CLKP
MLB0_DAT	MLB0 Single-Ended Data	B	PB_04
MLB0_DATN	MLB0 Differential Data (–)	Not Muxed	MLB0_DATN
MLB0_DATP	MLB0 Differential Data (+)	Not Muxed	MLB0_DATP
MLB0_SIG	MLB0 Single-Ended Signal	B	PB_05
MLB0_SIGN	MLB0 Differential Signal (–)	Not Muxed	MLB0_SIGN
MLB0_SIGP	MLB0 Differential Signal (+)	Not Muxed	MLB0_SIGP
$\overline{\text{MSI0_CD}}$	MSI0 Card Detect	C	PC_12
MSI0_CLK	MSI0 Clock	F	PF_04
MSI0_CMD	MSI0 Command	F	PF_07
MSI0_D0	MSI0 Data 0	E	PE_12
MSI0_D1	MSI0 Data 1	E	PE_13
MSI0_D2	MSI0 Data 2	E	PE_14
MSI0_D3	MSI0 Data 3	E	PE_15
MSI0_D4	MSI0 Data 4	F	PF_00
MSI0_D5	MSI0 Data 5	F	PF_01
MSI0_D6	MSI0 Data 6	F	PF_02
MSI0_D7	MSI0 Data 7	F	PF_03
$\overline{\text{MSI0_INT}}$	MSI0 eSDIO Interrupt Input	C	PC_13
PPI0_CLK	EPPI0 Clock	C	PC_11
PPI0_D00	EPPI0 Data 0	D	PD_10
PPI0_D01	EPPI0 Data 1	D	PD_11

Downloaded from Arrow.com.

Table 12. ADSP-SC57x/ADSP-2157x 400-Ball CSP_BGA Signal Descriptions (Continued)

Signal Name	Description	Port	Pin Name
PPI0_D02	EPPI0 Data 2	D	PD_12
PPI0_D03	EPPI0 Data 3	D	PD_13
PPI0_D04	EPPI0 Data 4	D	PD_14
PPI0_D05	EPPI0 Data 5	D	PD_15
PPI0_D06	EPPI0 Data 6	C	PC_05
PPI0_D07	EPPI0 Data 7	D	PD_09
PPI0_D08	EPPI0 Data 8	C	PC_01
PPI0_D09	EPPI0 Data 9	C	PC_02
PPI0_D10	EPPI0 Data 10	C	PC_03
PPI0_D11	EPPI0 Data 11	C	PC_04
PPI0_D12	EPPI0 Data 12	E	PE_00
PPI0_D13	EPPI0 Data 13	C	PC_07
PPI0_D14	EPPI0 Data 14	C	PC_08
PPI0_D15	EPPI0 Data 15	E	PE_01
PPI0_FS1	EPPI0 Frame Sync 1 (HSYNC)	C	PC_14
PPI0_FS2	EPPI0 Frame Sync 2 (VSYNC)	C	PC_15
PPI0_FS3	EPPI0 Frame Sync 3 (FIELD)	C	PC_06
SPI0_CLK	SPI0 Clock	C	PC_01
SPI0_MISO	SPI0 Master In, Slave Out	C	PC_02
SPI0_MOSI	SPI0 Master Out, Slave In	C	PC_03
SPI0_RDY	SPI0 Ready	C	PC_05
$\overline{\text{SPI0_SEL1}}$	SPI0 Slave Select Output 1	C	PC_04
$\overline{\text{SPI0_SEL2}}$	SPI0 Slave Select Output 2	C	PC_05
$\overline{\text{SPI0_SEL3}}$	SPI0 Slave Select Output 3	C	PC_06
$\overline{\text{SPI0_SEL4}}$	SPI0 Slave Select Output 4	A	PA_09
$\overline{\text{SPI0_SEL5}}$	SPI0 Slave Select Output 5	F	PF_05
$\overline{\text{SPI0_SEL6}}$	SPI0 Slave Select Output 6	F	PF_04
$\overline{\text{SPI0_SEL7}}$	SPI0 Slave Select Output 7	D	PD_05
$\overline{\text{SPI0_SS}}$	SPI0 Slave Select Input	C	PC_04
SPI1_CLK	SPI1 Clock	C	PC_07
SPI1_MISO	SPI1 Master In, Slave Out	C	PC_08
SPI1_MOSI	SPI1 Master Out, Slave In	C	PC_09
SPI1_RDY	SPI1 Ready	C	PC_11
$\overline{\text{SPI1_SEL1}}$	SPI1 Slave Select Output 1	C	PC_10
$\overline{\text{SPI1_SEL2}}$	SPI1 Slave Select Output 2	C	PC_11
$\overline{\text{SPI1_SEL3}}$	SPI1 Slave Select Output 3	F	PF_11
$\overline{\text{SPI1_SEL4}}$	SPI1 Slave Select Output 4	A	PA_14
$\overline{\text{SPI1_SEL5}}$	SPI1 Slave Select Output 5	B	PB_02
$\overline{\text{SPI1_SEL6}}$	SPI1 Slave Select Output 6	D	PD_07
$\overline{\text{SPI1_SEL7}}$	SPI1 Slave Select Output 7	D	PD_06
$\overline{\text{SPI1_SS}}$	SPI1 Slave Select Input	C	PC_10
SPI2_CLK	SPI2 Clock	B	PB_14
SPI2_D2	SPI2 Data 2	B	PB_12
SPI2_D3	SPI2 Data 3	B	PB_13
SPI2_MISO	SPI2 Master In, Slave Out	B	PB_10
SPI2_MOSI	SPI2 Master Out, Slave In	B	PB_11
SPI2_RDY	SPI2 Ready	C	PC_00
$\overline{\text{SPI2_SEL1}}$	SPI2 Slave Select Output 1	B	PB_15

Downloaded from Arrow.com.

Table 12. ADSP-SC57x/ADSP-2157x 400-Ball CSP_BGA Signal Descriptions (Continued)

Signal Name	Description	Port	Pin Name
$\overline{\text{SPI2_SEL2}}$	SPI2 Slave Select Output 2	F	PF_10
$\overline{\text{SPI2_SEL3}}$	SPI2 Slave Select Output 3	C	PC_00
$\overline{\text{SPI2_SEL4}}$	SPI2 Slave Select Output 4	D	PD_08
$\overline{\text{SPI2_SEL5}}$	SPI2 Slave Select Output 5	A	PA_15
$\overline{\text{SPI2_SEL6}}$	SPI2 Slave Select Output n	A	PA_10
$\overline{\text{SPI2_SEL7}}$	SPI2 Slave Select Output n	B	PB_07
$\overline{\text{SPI2_SS}}$	SPI2 Slave Select Input	B	PB_15
SYS_BMODE0	Boot Mode Control n	Not Muxed	SYS_BMODE0
SYS_BMODE1	Boot Mode Control n	Not Muxed	SYS_BMODE1
SYS_BMODE2	Boot Mode Control n	Not Muxed	SYS_BMODE2
SYS_CLKIN0	Clock/Crystal Input	Not Muxed	SYS_CLKIN0
SYS_CLKIN1	Clock/Crystal Input	Not Muxed	SYS_CLKIN1
SYS_CLKOUT	Processor Clock Output	Not Muxed	SYS_CLKOUT
SYS_FAULT	Active-High Fault Output	Not Muxed	SYS_FAULT
$\overline{\text{SYS_FAULT}}$	Active-Low Fault Output	Not Muxed	$\overline{\text{SYS_FAULT}}$
$\overline{\text{SYS_HWRST}}$	Processor Hardware Reset Control	Not Muxed	$\overline{\text{SYS_HWRST}}$
$\overline{\text{SYS_RESOUT}}$	Reset Output	Not Muxed	$\overline{\text{SYS_RESOUT}}$
SYS_XTAL0	Crystal Output	Not Muxed	SYS_XTAL0
SYS_XTAL1	Crystal Output	Not Muxed	SYS_XTAL1
TM0_ACI0	TIMER0 Alternate Capture Input 0	F	PF_09
TM0_ACI1	TIMER0 Alternate Capture Input 1	F	PF_11
TM0_ACI2	TIMER0 Alternate Capture Input 2	C	PC_12
TM0_ACI3	TIMER0 Alternate Capture Input 3	C	PC_14
TM0_ACI4	TIMER0 Alternate Capture Input 4	C	PC_13
TM0_ACI5	TIMER0 Alternate Capture Input 5	Not Applicable	DAI0_PIN04[1]
TM0_ACI6	TIMER0 Alternate Capture Input 6	Not Applicable	DAI0_PIN19[1]
TM0_ACI7	TIMER0 Alternate Capture Input 7	Not Applicable	CNT0_TO
TM0_ACLK0	TIMER0 Alternate Clock 0	Not Applicable	SYS_CLKIN1
TM0_ACLK1	TIMER0 Alternate Clock 1	F	PF_06
TM0_ACLK2	TIMER0 Alternate Clock 2	C	PC_01
TM0_ACLK3	TIMER0 Alternate Clock 3	D	PD_09
TM0_ACLK4	TIMER0 Alternate Clock 4	E	PE_02
TM0_ACLK5	TIMER0 Alternate Clock 5	Not Applicable	DAI0_PIN03[1]
TM0_ACLK6	TIMER0 Alternate Clock 6	Not Applicable	DAI0_PIN20[1]
TM0_ACLK7	TIMER0 Alternate Clock 7	Not Applicable	SYS_CLKIN0
TM0_CLK	TIMER0 Clock	C	PC_03
TM0_TMR0	TIMER0 Timer 0	E	PE_12
TM0_TMR1	TIMER0 Timer 1	F	PF_05
TM0_TMR2	TIMER0 Timer 2	F	PF_07
TM0_TMR3	TIMER0 Timer 3	B	PB_01
TM0_TMR4	TIMER0 Timer 4	B	PB_03
TM0_TMR5	TIMER0 Timer 5	C	PC_15
TM0_TMR6	TIMER0 Timer 6	E	PE_14
TM0_TMR7	TIMER0 Timer 7	D	PD_07
TRACE0_CLK	TRACE0 Trace Clock	F	PF_06
TRACE0_D00	TRACE0 Trace Data 0	F	PF_00
TRACE0_D01	TRACE0 Trace Data 1	F	PF_01
TRACE0_D02	TRACE0 Trace Data 2	F	PF_02

Downloaded from Arrow.com.

Table 12. ADSP-SC57x/ADSP-2157x 400-Ball CSP_BGA Signal Descriptions (Continued)

Signal Name	Description	Port	Pin Name
TRACE0_D03	TRACE0 Trace Data 3	F	PF_03
TRACE0_D04	TRACE0 Trace Data 4	D	PD_10
TRACE0_D05	TRACE0 Trace Data 5	D	PD_11
TRACE0_D06	TRACE0 Trace Data 6	D	PD_12
TRACE0_D07	TRACE0 Trace Data 7	D	PD_13
TWI0_SCL	TWI0 Serial Clock	Not Muxed	TWI0_SCL
TWI0_SDA	TWI0 Serial Data	Not Muxed	TWI0_SDA
TWI1_SCL	TWI1 Serial Clock	Not Muxed	TWI1_SCL
TWI1_SDA	TWI1 Serial Data	Not Muxed	TWI1_SDA
TWI2_SCL	TWI2 Serial Clock	Not Muxed	TWI2_SCL
TWI2_SDA	TWI2 Serial Data	Not Muxed	TWI2_SDA
$\overline{\text{UART0_CTS}}$	UART0 Clear to Send	D	PD_06
$\overline{\text{UART0_RTS}}$	UART0 Request to Send	D	PD_05
$\overline{\text{UART0_RX}}$	UART0 Receive	F	PF_09
$\overline{\text{UART0_TX}}$	UART0 Transmit	F	PF_08
$\overline{\text{UART1_CTS}}$	UART1 Clear to Send	E	PE_14
$\overline{\text{UART1_RTS}}$	UART1 Request to Send	E	PE_00
$\overline{\text{UART1_RX}}$	UART1 Receive	F	PF_11
$\overline{\text{UART1_TX}}$	UART1 Transmit	F	PF_10
$\overline{\text{UART2_CTS}}$	UART2 Clear to Send	A	PA_11
$\overline{\text{UART2_RTS}}$	UART2 Request to Send	A	PA_10
$\overline{\text{UART2_RX}}$	UART2 Receive	C	PC_13
$\overline{\text{UART2_TX}}$	UART2 Transmit	C	PC_12
USB0_CLKIN	USB0 Clock/Crystal Input	Not Muxed	USB_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	USB_XTAL
VDD_EXT	External Voltage Domain	Not Muxed	VDD_EXT
VDD_INT	Internal Voltage Domain	Not Muxed	VDD_INT
VDD_DMC	DMC VDD	Not Muxed	VDD_DMC
VDD_HADC	HADC/TMU VDD	Not Muxed	VDD_HADC
VDD_USB	USB VDD	Not Muxed	VDD_USB

[1] Signal is routed to the DAI0_PINnn pin through the DAI0_PBnn pin buffers using the SRU.

Downloaded from Arrow.com.

GPIO MULTIPLEXING FOR 400-BALL CSP_BGA PACKAGE

Table 13 through Table 18 identify the pin functions that are multiplexed on the GPIO pins of the 400-ball CSP_BGA package.

Table 13. 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					
PA_01					
PA_02					
PA_03					
PA_04					
PA_05					
PA_06					
PA_07					
PA_08					
PA_09	ETH0_PTPPPS0	LP1_D6	$\overline{\text{SPI0_SEL4}}$		
PA_10	ETH0_MDIO	$\overline{\text{UART2_RTS}}$	$\overline{\text{SPI2_SEL6}}$		
PA_11	ETH0_MDC	$\overline{\text{UART2_CTS}}$			
PA_12	ETH0_RXD1				
PA_13	ETH0_RXD0				
PA_14	ETH0_RXD2	ACM0_A3	$\overline{\text{SPI1_SEL4}}$		
PA_15	ETH0_RXD3	ACM0_T0	$\overline{\text{SPI2_SEL5}}$		

Table 14. 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	ETH0_RXCLK_REFCLK	C2_FLG0			
PB_01	ETH0_CRS	ACM0_A4	LP1_ACK	TM0_TMR3	
PB_02	ETH0_RXCTL_RXDV		$\overline{\text{SPI1_SEL5}}$		
PB_03	ETH0_RXERR	MLB0_CLKOUT	LP1_CLK	TM0_TMR4	
PB_04	ETH0_TXCLK	MLB0_DAT			
PB_05	ETH0_TXD3	MLB0_SIG			
PB_06	ETH0_TXD2	MLB0_CLK			
PB_07	ETH0_TXD0		$\overline{\text{SPI2_SEL7}}$		
PB_08	ETH0_TXD1				
PB_09	ETH0_TXCTL_TXEN				
PB_10	SPI2_MISO				
PB_11	SPI2_MOSI				
PB_12	SPI2_D2				
PB_13	SPI2_D3				
PB_14	SPI2_CLK				
PB_15	$\overline{\text{SPI2_SEL1}}$				$\overline{\text{SPI2_SS}}$

Downloaded from Arrow.com.

Table 15. 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	$\overline{\text{SPI2_SEL3}}$	SPI2_RDY			
PC_01	SPI0_CLK	PPI0_D08			TM0_ACLK2
PC_02	SPI0_MISO	PPI0_D09			
PC_03	SPI0_MOSI	PPI0_D10			TM0_CLK
PC_04	$\overline{\text{SPI0_SEL1}}$	PPI0_D11			$\overline{\text{SPI0_SS}}$
PC_05	$\overline{\text{SPI0_SEL2}}$	PPI0_D06	SPI0_RDY		
PC_06	$\overline{\text{SPI0_SEL3}}$	ETH0_COL	PPI0_FS3		
PC_07	SPI1_CLK	PPI0_D13			
PC_08	SPI1_MISO	PPI0_D14			
PC_09	SPI1_MOSI				
PC_10	$\overline{\text{SPI1_SEL1}}$				$\overline{\text{SPI1_SS}}$
PC_11	$\overline{\text{SPI1_SEL2}}$	PPI0_CLK	SPI1_RDY		TM0_ACLK4
PC_12	CAN0_RX	$\overline{\text{MSI0_CD}}$	$\overline{\text{UART2_TX}}$		TM0_ACI2
PC_13	CAN0_TX	$\overline{\text{MSI0_INT}}$	$\overline{\text{UART2_RX}}$		TM0_ACI4
PC_14	CAN1_RX	PPI0_FS1	ACM0_A1	C2_FLG1	TM0_ACI3
PC_15	CAN1_TX	PPI0_FS2	ACM0_A2	TM0_TMR5	

Table 16. Signal Multiplexing for Port D

Signal Name	Multiplexed Function 0	Multiplexed Function 1	Multiplexed Function 2	Multiplexed Function 3	Multiplexed Function Input Tap
PD_00					
PD_01					
PD_02					
PD_03					
PD_04					
PD_05	$\overline{\text{SPI0_SEL7}}$		$\overline{\text{UART0_RTS}}$		
PD_06	$\overline{\text{SPI1_SEL7}}$	C1_FLG3	$\overline{\text{UART0_CTS}}$		
PD_07	$\overline{\text{SPI1_SEL6}}$	CNT0_ZM	TM0_TMR7		
PD_08	ETH0_PTPPPS1	CNT0_DG	$\overline{\text{SPI2_SEL4}}$		
PD_09	LP1_D7	PPI0_D07	HADC0_EOC_DOUT		TM0_ACLK3
PD_10	LP1_D0	PPI0_D00	TRACE0_D04		
PD_11	LP1_D1	PPI0_D01	TRACE0_D05		
PD_12	LP1_D2	PPI0_D02	TRACE0_D06		
PD_13	LP1_D3	PPI0_D03	TRACE0_D07		
PD_14	LP1_D4	PPI0_D04	ETH0_PTPAUXIN0		
PD_15	LP1_D5	PPI0_D05	ETH0_PTPAUXIN1		

Table 17. Signal Multiplexing for Port E

Signal Name	Multiplexed Function 0	Multiplexed Function 1	Multiplexed Function 2	Multiplexed Function 3	Multiplexed Function Input Tap
PE_00	ETH0_PTPPPS2	PPI0_D12	$\overline{\text{UART1_RTS}}$		
PE_01	ETH0_PTPPPS3	PPI0_D15	C1_FLG1		
PE_02	LP0_CLK				

Downloaded from Arrow.com.

Table 17. Signal Multiplexing for Port E (Continued)

Signal Name	Multiplexed Function 0	Multiplexed Function 1	Multiplexed Function 2	Multiplexed Function 3	Multiplexed Function Input Tap
PE_03	LP0_ACK				
PE_04	LP0_D0				
PE_05	LP0_D1				
PE_06	LP0_D2				
PE_07	LP0_D3				
PE_08	LP0_D4				
PE_09	LP0_D5				
PE_10	LP0_D6				
PE_11	LP0_D7				
PE_12	MSI0_D0		TM0_TMR0		
PE_13	MSI0_D1	C1_FLG0	CNT0_UD		
PE_14	MSI0_D2	$\overline{\text{UART1_CTS}}$	TM0_TMR6		
PE_15	MSI0_D3	C2_FLG3			

Table 18. Signal Multiplexing for Port F

Signal Name	Multiplexed Function 0	Multiplexed Function 1	Multiplexed Function 2	Multiplexed Function 3	Multiplexed Function Input Tap
PF_00	MSI0_D4	TRACE0_D00			
PF_01	MSI0_D5	TRACE0_D01			
PF_02	MSI0_D6	TRACE0_D02			
PF_03	MSI0_D7	TRACE0_D03			
PF_04	MSI0_CLK	C1_FLG2	$\overline{\text{SPI0_SEL6}}$		
PF_05	ETH0_PTPCLKIN0	TM0_TMR1	$\overline{\text{SPI0_SEL5}}$		
PF_06	ETH0_PTPAUXIN2	TRACE0_CLK			TM0_ACLK1
PF_07	ETH0_PTPAUXIN3	TM0_TMR2	MSI0_CMD		
PF_08	$\overline{\text{UART0_TX}}$				
PF_09	$\overline{\text{UART0_RX}}$				TM0_ACI0
PF_10	$\overline{\text{UART1_TX}}$	$\overline{\text{SPI2_SEL2}}$			
PF_11	$\overline{\text{UART1_RX}}$	ACM0_A0	$\overline{\text{SPI1_SEL3}}$	C2_FLG2	TM0_ACI1

Table 19 shows the internal timer signal routing. This table applies to both the 400-ball CSP_BGA and 176-lead LQFP packages.

Table 19. Internal Timer Signal Routing

Timer Input Signal	Internal Source
TM0_ACLK0[1]	SYS_CLKIN1
TM0_ACI5	DAI0_PB04_O
TM0_ACLK5	DAI0_PB03_O
TM0_ACI6	DAI0_PB20_O
TM0_ACLK6	DAI0_PB19_O
TM0_ACI7	CNT0_TO
TM0_ACLK7	SYS_CLKIN0

[1] Not applicable for LQFP package.

Downloaded from Arrow.com.

176-LEAD LQFP SIGNAL DESCRIPTIONS

The processor pin definitions are shown Table 20 for the 176-lead LQFP package. The columns in this table provide the following information:

- The signal name column includes the signal name for every pin and the GPIO multiplexed pin function, where applicable.
- The description column provides a descriptive name for each signal.
- The port column shows whether or not a signal is multiplexed with other signals on a GPIO port pin.
- The pin name column 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 GPIO pin).
- The DAI pins and their associated signal routing units (SRUs) connect inputs and outputs of the DAI peripherals (SPORT, ASRC, S/PDIF, and PCG). See the Digital Audio Interface (DAI) chapter of the ADSP-SC57x/ADSP-2157x SHARC+ Processor Hardware Reference for complete information on the use of the DAIs and SRUs.

Table 20. ADSP-SC57x/ADSP-2157x 176-Lead LQFP Signal Descriptions

Signal Name	Description	Port	Pin Name
ACM0_A0	ACM0 ADC Control Signals	A	PA_08
ACM0_A1	ACM0 ADC Control Signals	C	PC_14
ACM0_A2	ACM0 ADC Control Signals	C	PC_15
ACM0_A3	ACM0 ADC Control Signals	A	PA_14
ACM0_A4	ACM0 ADC Control Signals	B	PB_01
ACM0_T0	ACM0 External Trigger n	A	PA_15
C1_FLG0	SHARC Core 1 Flag Pin	D	PD_00
C1_FLG1	SHARC Core 1 Flag Pin	D	PD_01
C1_FLG2	SHARC Core 1 Flag Pin	C	PC_09
C1_FLG3	SHARC Core 1 Flag Pin	D	PD_06
C2_FLG0	SHARC Core 2 Flag Pin	B	PB_00
C2_FLG1	SHARC Core 2 Flag Pin	C	PC_14
C2_FLG2	SHARC Core 2 Flag Pin	C	PC_15
C2_FLG3	SHARC Core 2 Flag Pin	D	PD_05
CAN0_RX	CAN0 Receive	C	PC_12
CAN0_TX	CAN0 Transmit	C	PC_13
CAN1_RX	CAN1 Receive	C	PC_14
CAN1_TX	CAN1 Transmit	C	PC_15
CNT0_DG	CNT0 Count Down and Gate	D	PD_08
CNT0_UD	CNT0 Count Up and Direction	D	PD_00
CNT0_ZM	CNT0 Count Zero Marker	D	PD_07
DAI0_PIN01	DAI0 Pin 1	Not Muxed	DAI0_PIN01
DAI0_PIN02	DAI0 Pin 2	Not Muxed	DAI0_PIN02
DAI0_PIN03	DAI0 Pin 3	Not Muxed	DAI0_PIN03
DAI0_PIN04	DAI0 Pin 4	Not Muxed	DAI0_PIN04
DAI0_PIN05	DAI0 Pin 5	Not Muxed	DAI0_PIN05
DAI0_PIN06	DAI0 Pin 6	Not Muxed	DAI0_PIN06
DAI0_PIN07	DAI0 Pin 7	Not Muxed	DAI0_PIN07
DAI0_PIN08	DAI0 Pin 8	Not Muxed	DAI0_PIN08
DAI0_PIN09	DAI0 Pin 9	Not Muxed	DAI0_PIN09
DAI0_PIN10	DAI0 Pin 10	Not Muxed	DAI0_PIN10
DAI0_PIN11	DAI0 Pin 11	Not Muxed	DAI0_PIN11
DAI0_PIN12	DAI0 Pin 12	Not Muxed	DAI0_PIN12
DAI0_PIN13	DAI0 Pin 13	Not Muxed	DAI0_PIN13
DAI0_PIN14	DAI0 Pin 14	Not Muxed	DAI0_PIN14

Downloaded from Arrow.com.

Table 20. ADSP-SC57x/ADSP-2157x 176-Lead LQFP Signal Descriptions (Continued)

Signal Name	Description	Port	Pin Name
DAI0_PIN15	DAI0 Pin 15	Not Muxed	DAI0_PIN15
DAI0_PIN16	DAI0 Pin 16	Not Muxed	DAI0_PIN16
DAI0_PIN17	DAI0 Pin 17	Not Muxed	DAI0_PIN17
DAI0_PIN18	DAI0 Pin 18	Not Muxed	DAI0_PIN18
DAI0_PIN19	DAI0 Pin 19	Not Muxed	DAI0_PIN19
DAI0_PIN20	DAI0 Pin 20	Not Muxed	DAI0_PIN20
ETH0_COL	EMAC0 MII Collision detect	C	PC_06
ETH0_CRS	EMAC0 Carrier Sense/RMII Receive Data Valid	B	PB_01
ETH0_MDC	EMAC0 Management Channel Clock	A	PA_11
ETH0_MDIO	EMAC0 Management Channel Serial Data	A	PA_10
ETH0_PTPAUXIN0	EMAC0 PTP Auxiliary Trigger Input 0	D	PD_14
ETH0_PTPAUXIN1	EMAC0 PTP Auxiliary Trigger Input 1	D	PD_15
ETH0_PTPPPS0	EMAC0 PTP Pulse Per Second Output 0	A	PA_09
ETH0_PTPPPS1	EMAC0 PTP Pulse Per Second Output 1	D	PD_08
ETH0_RXCLK_REFCLK	EMAC0 RXCLK (10/100/1000) or REFCLK (10/100)	B	PB_00
ETH0_RXCTL_RXDV	EMAC0 RXCTL (10/100/1000) or CRS (10/100)	B	PB_01
ETH0_RXD0	EMAC0 Receive Data 0	A	PA_13
ETH0_RXD1	EMAC0 Receive Data 1	A	PA_12
ETH0_RXD2	EMAC0 Receive Data 2	A	PA_14
ETH0_RXD3	EMAC0 Receive Data 3	A	PA_15
ETH0_RXERR	EMAC0 Receive Error	B	PB_03
ETH0_TXCLK	EMAC0 Transmit Clock	B	PB_04
ETH0_TXCTL_TXEN	EMAC0 TXCTL (10/100/1000) or TXEN (10/100)	B	PB_09
ETH0_TXD0	EMAC0 Transmit Data 0	B	PB_07
ETH0_TXD1	EMAC0 Transmit Data 1	B	PB_08
ETH0_TXD2	EMAC0 Transmit Data 2	B	PB_06
ETH0_TXD3	EMAC0 Transmit Data 3	B	PB_05
HADC0_EOC_DOUT	HADC0 End of Conversion/Serial Data Out	D	PD_09
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_TCK	JTAG Clock	Not Muxed	JTG_TCK
JTG_TDI	JTAG Serial Data In	Not Muxed	JTG_TDI
JTG_TDO	JTAG Serial Data Out	Not Muxed	JTG_TDO
JTG_TMS	JTAG Mode Select	Not Muxed	JTG_TMS
$\overline{\text{JTG_TRST}}$	JTAG Reset	Not Muxed	$\overline{\text{JTG_TRST}}$
LP1_ACK	LP1 Acknowledge	B	PB_01
LP1_CLK	LP1 Clock	B	PB_03
LP1_D0	LP1 Data 0	D	PD_10
LP1_D1	LP1 Data 1	D	PD_11
LP1_D2	LP1 Data 2	D	PD_12
LP1_D3	LP1 Data 3	D	PD_13
LP1_D4	LP1 Data 4	D	PD_14
LP1_D5	LP1 Data 5	D	PD_15
LP1_D6	LP1 Data 6	A	PA_09

Downloaded from Arrow.com.

Table 20. ADSP-SC57x/ADSP-2157x 176-Lead LQFP Signal Descriptions (Continued)

Signal Name	Description	Port	Pin Name
LP1_D7	LP1 Data 7	D	PD_09
MLB0_CLK	MLB0 Single-Ended Clock	B	PB_06
MLB0_CLKOUT	MLB0 Single-Ended Clock Out	B	PB_03
MLB0_DAT	MLB0 Single-Ended Data	B	PB_04
MLB0_SIG	MLB0 Single-Ended Signal	B	PB_05
PPI0_CLK	EPPI0 Clock	C	PC_11
PPI0_D00	EPPI0 Data 0	D	PD_10
PPI0_D01	EPPI0 Data 1	D	PD_11
PPI0_D02	EPPI0 Data 2	D	PD_12
PPI0_D03	EPPI0 Data 3	D	PD_13
PPI0_D04	EPPI0 Data 4	D	PD_14
PPI0_D05	EPPI0 Data 5	D	PD_15
PPI0_D06	EPPI0 Data 6	C	PC_05
PPI0_D07	EPPI0 Data 7	D	PD_09
PPI0_D08	EPPI0 Data 8	C	PC_01
PPI0_D09	EPPI0 Data 9	C	PC_02
PPI0_D10	EPPI0 Data 10	C	PC_03
PPI0_D11	EPPI0 Data 11	C	PC_04
PPI0_FS1	EPPI0 Frame Sync 1 (HSYNC)	C	PC_14
PPI0_FS2	EPPI0 Frame Sync 2 (VSYNC)	C	PC_15
PPI0_FS3	EPPI0 Frame Sync 3 (FIELD)	C	PC_06
SPI0_CLK	SPI0 Clock	C	PC_01
SPI0_MISO	SPI0 Master In, Slave Out	C	PC_02
SPI0_MOSI	SPI0 Master Out, Slave In	C	PC_03
SPI0_RDY	SPI0 Ready	C	PC_05
$\overline{\text{SPI0_SEL1}}$	SPI0 Slave Select Output 1	C	PC_04
$\overline{\text{SPI0_SEL2}}$	SPI0 Slave Select Output 2	C	PC_05
$\overline{\text{SPI0_SEL3}}$	SPI0 Slave Select Output 3	C	PC_06
$\overline{\text{SPI0_SEL4}}$	SPI0 Slave Select Output 4	A	PA_09
$\overline{\text{SPI0_SEL5}}$	SPI0 Slave Select Output 5	D	PD_03
$\overline{\text{SPI0_SEL6}}$	SPI0 Slave Select Output 6	D	PD_04
$\overline{\text{SPI0_SEL7}}$	SPI0 Slave Select Output 7	D	PD_05
$\overline{\text{SPI0_SS}}$	SPI0 Slave Select Input	C	PC_04
SPI1_CLK	SPI1 Clock	C	PC_07
SPI1_MISO	SPI1 Master In, Slave Out	C	PC_08
SPI1_MOSI	SPI1 Master Out, Slave In	C	PC_09
SPI1_RDY	SPI1 Ready	C	PC_11
$\overline{\text{SPI1_SEL1}}$	SPI1 Slave Select Output 1	C	PC_10
$\overline{\text{SPI1_SEL2}}$	SPI1 Slave Select Output 2	C	PC_11
$\overline{\text{SPI1_SEL3}}$	SPI1 Slave Select Output 3	A	PA_08
$\overline{\text{SPI1_SEL4}}$	SPI1 Slave Select Output 4	A	PA_14
$\overline{\text{SPI1_SEL5}}$	SPI1 Slave Select Output 5	B	PB_02
$\overline{\text{SPI1_SEL6}}$	SPI1 Slave Select Output 6	D	PD_07
$\overline{\text{SPI1_SEL7}}$	SPI1 Slave Select Output 7	D	PD_06
$\overline{\text{SPI1_SS}}$	SPI1 Slave Select Input	C	PC_10
SPI2_CLK	SPI2 Clock	B	PB_14
SPI2_D2	SPI2 Data 2	B	PB_12
SPI2_D3	SPI2 Data 3	B	PB_13

Downloaded from Arrow.com.

Table 20. ADSP-SC57x/ADSP-2157x 176-Lead LQFP Signal Descriptions (Continued)

Signal Name	Description	Port	Pin Name
SPI2_MISO	SPI2 Master In, Slave Out	B	PB_10
SPI2_MOSI	SPI2 Master Out, Slave In	B	PB_11
SPI2_RDY	SPI2 Ready	C	PC_00
$\overline{\text{SPI2_SEL1}}$	SPI2 Slave Select Output 1	B	PB_15
$\overline{\text{SPI2_SEL2}}$	SPI2 Slave Select Output 2	A	PA_07
$\overline{\text{SPI2_SEL3}}$	SPI2 Slave Select Output 3	C	PC_00
$\overline{\text{SPI2_SEL4}}$	SPI2 Slave Select Output 4	D	PD_08
$\overline{\text{SPI2_SEL5}}$	SPI2 Slave Select Output 5	A	PA_15
$\overline{\text{SPI2_SEL6}}$	SPI2 Slave Select Output n	A	PA_10
$\overline{\text{SPI2_SEL7}}$	SPI2 Slave Select Output n	B	PB_07
$\overline{\text{SPI2_SS}}$	SPI2 Slave Select Input	B	PB_15
SYS_BMODE0	Boot Mode Control n	Not Muxed	SYS_BMODE0
SYS_BMODE1	Boot Mode Control n	Not Muxed	SYS_BMODE1
SYS_CLKIN0	Clock/Crystal Input	Not Muxed	SYS_CLKIN0
SYS_CLKOUT	Processor Clock Output	Not Muxed	SYS_CLKOUT
$\overline{\text{SYS_FAULT}}$	Active-High Fault Output	Not Muxed	$\overline{\text{SYS_FAULT}}$
$\overline{\text{SYS_HWRST}}$	Processor Hardware Reset Control	Not Muxed	$\overline{\text{SYS_HWRST}}$
$\overline{\text{SYS_RESOUT}}$	Reset Output	Not Muxed	$\overline{\text{SYS_RESOUT}}$
SYS_XTAL0	Crystal Output	Not Muxed	SYS_XTAL0
TM0_ACI0	TIMER0 Alternate Capture Input 0	A	PA_06
TM0_ACI1	TIMER0 Alternate Capture Input 1	A	PA_08
TM0_ACI2	TIMER0 Alternate Capture Input 2	C	PC_12
TM0_ACI3	TIMER0 Alternate Capture Input 3	C	PC_14
TM0_ACI4	TIMER0 Alternate Capture Input 4	C	PC_13
TM0_ACI5	TIMER0 Alternate Capture Input 5	Not Applicable	DAI_PB04_O
TM0_ACI6	TIMER0 Alternate Capture Input 6	Not Applicable	DAI_PB19_O
TM0_ACI7	TIMER0 Alternate Capture Input 7	Not Applicable	CNT0_TO
TM0_ACLK1	TIMER0 Alternate Clock 1	A	PA_00
TM0_ACLK2	TIMER0 Alternate Clock 2	C	PC_01
TM0_ACLK3	TIMER0 Alternate Clock 3	D	PD_09
TM0_ACLK4	TIMER0 Alternate Clock 4	C	PC_11
TM0_ACLK5	TIMER0 Alternate Clock 5	Not Applicable	DAI_PB03_O
TM0_ACLK6	TIMER0 Alternate Clock 6	Not Applicable	DAI_PB20_O
TM0_ACLK7	TIMER0 Alternate Clock 7	Not Applicable	SYS_CLKIN0
TM0_CLK	TIMER0 Clock	C	PC_03
TM0_TMR0	TIMER0 Timer 0	D	PD_02
TM0_TMR1	TIMER0 Timer 1	D	PD_03
TM0_TMR2	TIMER0 Timer 2	D	PD_04
TM0_TMR3	TIMER0 Timer 3	B	PB_01
TM0_TMR4	TIMER0 Timer 4	B	PB_03
TM0_TMR5	TIMER0 Timer 5	C	PC_15
TM0_TMR7	TIMER0 Timer 7	D	PD_07
TRACE0_CLK	TRACE0 Trace Clock	A	PA_00
TRACE0_D00	TRACE0 Trace Data	A	PA_01
TRACE0_D01	TRACE0 Trace Data	A	PA_02
TRACE0_D02	TRACE0 Trace Data	A	PA_03
TRACE0_D03	TRACE0 Trace Data	A	PA_04
TRACE0_D04	TRACE0 Trace Data	D	PD_10

Downloaded from Arrow.com.

Table 20. ADSP-SC57x/ADSP-2157x 176-Lead LQFP Signal Descriptions (Continued)

Signal Name	Description	Port	Pin Name
TRACE0_D05	TRACE0 Trace Data	D	PD_11
TRACE0_D06	TRACE0 Trace Data	D	PD_12
TRACE0_D07	TRACE0 Trace Data 7	D	PD_13
TWI0_SCL	TWI0 Serial Clock	Not Muxed	TWI0_SCL
TWI0_SDA	TWI0 Serial Data	Not Muxed	TWI0_SDA
TWI1_SCL	TWI1 Serial Clock	Not Muxed	TWI1_SCL
TWI1_SDA	TWI1 Serial Data	Not Muxed	TWI1_SDA
TWI2_SCL	TWI2 Serial Clock	Not Muxed	TWI2_SCL
TWI2_SDA	TWI2 Serial Data	Not Muxed	TWI2_SDA
$\overline{\text{UART0_CTS}}$	UART0 Clear to Send	D	PD_06
$\overline{\text{UART0_RTS}}$	UART0 Request to Send	D	PD_05
$\overline{\text{UART0_RX}}$	UART0 Receive	A	PA_06
$\overline{\text{UART0_TX}}$	UART0 Transmit	A	PA_05
$\overline{\text{UART1_CTS}}$	UART1 Clear to Send	D	PD_01
$\overline{\text{UART1_RTS}}$	UART1 Request to Send	D	PD_00
$\overline{\text{UART1_RX}}$	UART1 Receive	A	PA_08
$\overline{\text{UART1_TX}}$	UART1 Transmit	A	PA_07
$\overline{\text{UART2_CTS}}$	UART2 Clear to Send	A	PA_11
$\overline{\text{UART2_RTS}}$	UART2 Request to Send	A	PA_10
$\overline{\text{UART2_RX}}$	UART2 Receive	C	PC_13
$\overline{\text{UART2_TX}}$	UART2 Transmit	C	PC_12

Downloaded from Arrow.com.

GPIO MULTIPLEXING FOR 176-LEAD LQFP PACKAGE

Table 21 through Table 24 identify the pin functions that are multiplexed on the GPIO pins of the 176-lead LQFP package.

Table 21. 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	TRACE0_CLK				TM0_ACLK1
PA_01	TRACE0_D00				
PA_02	TRACE0_D01				
PA_03	TRACE0_D02				
PA_04	TRACE0_D03				
PA_05	$\overline{\text{UART0_TX}}$				
PA_06	$\overline{\text{UART0_RX}}$				TM0_ACI0
PA_07	$\overline{\text{UART1_TX}}$	$\overline{\text{SPI2_SEL2}}$			
PA_08	$\overline{\text{UART1_RX}}$	ACM0_A0	$\overline{\text{SPI1_SEL3}}$		TM0_ACI1
PA_09	ETH0_PTPPPS0	LP1_D6	$\overline{\text{SPI0_SEL4}}$		
PA_10	ETH0_MDIO	$\overline{\text{UART2_RTS}}$	$\overline{\text{SPI2_SEL6}}$		
PA_11	ETH0_MDC	$\overline{\text{UART2_CTS}}$			
PA_12	ETH0_RXD1				
PA_13	ETH0_RXD0				
PA_14	ETH0_RXD2	ACM0_A3	$\overline{\text{SPI1_SEL4}}$		
PA_15	ETH0_RXD3	ACM0_T0	$\overline{\text{SPI2_SEL5}}$		

Table 22. 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	ETH0_RXCLK_REFCLK	C2_FLG0			
PB_01	ETH0_CRS	ACM0_A4	LP1_ACK	TM0_TMR3	
PB_02	ETH0_RXCTL_RXDV		$\overline{\text{SPI1_SEL5}}$		
PB_03	ETH0_RXERR	MLB0_CLKOUT	LP1_CLK	TM0_TMR4	
PB_04	ETH0_TXCLK	MLB0_DAT			
PB_05	ETH0_TXD3	MLB0_SIG			
PB_06	ETH0_TXD2	MLB0_CLK			
PB_07	ETH0_TXD0		$\overline{\text{SPI2_SEL7}}$		
PB_08	ETH0_TXD1				
PB_09	ETH0_TXCTL_TXEN				
PB_10	SPI2_MISO				
PB_11	SPI2_MOSI				
PB_12	SPI2_D2				
PB_13	SPI2_D3				
PB_14	SPI2_CLK				
PB_15	$\overline{\text{SPI2_SEL1}}$				$\overline{\text{SPI2_SS}}$

Downloaded from Arrow.com.

Table 23. 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	$\overline{\text{SPI2_SEL3}}$	SPI2_RDY			
PC_01	SPI0_CLK	PPI0_D08			TM0_ACLK2
PC_02	SPI0_MISO	PPI0_D09			
PC_03	SPI0_MOSI	PPI0_D10			TM0_CLK
PC_04	$\overline{\text{SPI0_SEL1}}$	PPI0_D11			$\overline{\text{SPI0_SS}}$
PC_05	$\overline{\text{SPI0_SEL2}}$	PPI0_D06	SPI0_RDY		
PC_06	$\overline{\text{SPI0_SEL3}}$	ETH0_COL	PPI0_FS3		
PC_07	SPI1_CLK				
PC_08	SPI1_MISO				
PC_09	SPI1_MOSI	C1_FLG2			
PC_10	$\overline{\text{SPI1_SEL1}}$	C2_FLG2			$\overline{\text{SPI1_SS}}$
PC_11	$\overline{\text{SPI1_SEL2}}$	PPI0_CLK	SPI1_RDY		TM0_ACLK4
PC_12	CAN0_RX		$\overline{\text{UART2_TX}}$		TM0_ACI2
PC_13	CAN0_TX		$\overline{\text{UART2_RX}}$		TM0_ACI4
PC_14	CAN1_RX	PPI0_FS1	ACM0_A1	C2_FLG1	TM0_ACI3
PC_15	CAN1_TX	PPI0_FS2	ACM0_A2	TM0_TMR5	

Table 24. Signal Multiplexing for Port D

Signal Name	Multiplexed Function 0	Multiplexed Function 1	Multiplexed Function 2	Multiplexed Function 3	Multiplexed Function Input Tap
PD_00	C1_FLG0	$\overline{\text{UART1_RTS}}$	CNT0_UD		
PD_01	C1_FLG1	$\overline{\text{UART1_CTS}}$	TM0_TMR6		
PD_02	TM0_TMR0				
PD_03	TM0_TMR1	$\overline{\text{SPI0_SEL5}}$			
PD_04	TM0_TMR2		$\overline{\text{SPI0_SEL6}}$		
PD_05	$\overline{\text{SPI0_SEL7}}$	C2_FLG3	$\overline{\text{UART0_RTS}}$		
PD_06	$\overline{\text{SPI1_SEL7}}$	C1_FLG3	$\overline{\text{UART0_CTS}}$		
PD_07	$\overline{\text{SPI1_SEL6}}$	CNT0_ZM	TM0_TMR7		
PD_08	ETH0_PTPPPS1	CNT0_DG	$\overline{\text{SPI2_SEL4}}$		
PD_09	LP1_D7	PPI0_D07	HADC0_EOC_DOUT		TM0_ACLK3
PD_10	LP1_D0	PPI0_D00	TRACE0_D04		
PD_11	LP1_D1	PPI0_D01	TRACE0_D05		
PD_12	LP1_D2	PPI0_D02	TRACE0_D06		
PD_13	LP1_D3	PPI0_D03	TRACE0_D07		
PD_14	LP1_D4	PPI0_D04	ETH0_PTPAUXIN0		
PD_15	LP1_D5	PPI0_D05	ETH0_PTPAUXIN1		

Downloaded from Arrow.com.

ADSP-SC57x/ADSP-2157x DESIGNER QUICK REFERENCE

Table 25 provides a quick reference summary of pin related information for circuit board design. The columns in this table provide the following information:

- The signal name column includes the signal name for every pin and the GPIO multiplexed pin function, where applicable.
- The type column identifies the I/O type or supply type of the pin. The abbreviations used in this column are analog (a), supply (s), ground (g) and Input, Output, and InOut.
- The driver type column identifies the driver type used by the corresponding pin. The driver types are defined in the Output Drive Currents section of this data sheet.
- The internal termination column specifies the termination present after the processor is powered up (both during reset and after reset).
- The reset drive column specifies the active drive on the signal when the processor is in the reset state.
- The power domain column specifies the power supply domain in which the signal resides.
- The description and notes column identifies any special requirements or characteristics for a signal. These recommendations apply whether or not the hardware block associated with the signal is featured on the product. If no special requirements are listed, the signal can be left unconnected if it is not used. For multiplexed GPIO pins, this column identifies the functions available on the pin.

Table 25. ADSP-SC57x/ADSP-2157x Designer Quick Reference

Signal Name	Type	Driver Type	Internal Termination	Reset Drive	Power Domain	Description and Notes
DAI0_PIN01	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: DAI0 Pin 1 Notes: See note[2]
DAI0_PIN02	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: DAI0 Pin 2 Notes: See note[2]
DAI0_PIN03	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: DAI0 Pin 3 Notes: See note[2]
DAI0_PIN04	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: DAI0 Pin 4 Notes: See note[2]
DAI0_PIN05	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: DAI0 Pin 5 Notes: See note[2]
DAI0_PIN06	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: DAI0 Pin 6 Notes: See note[2]
DAI0_PIN07	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: DAI0 Pin 7 Notes: See note[2]
DAI0_PIN08	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: DAI0 Pin 8 Notes: See note[2]
DAI0_PIN09	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: DAI0 Pin 9 Notes: See note[2]
DAI0_PIN10	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: DAI0 Pin 10 Notes: See note[2]
DAI0_PIN11	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: DAI0 Pin 11 Notes: See note[2]
DAI0_PIN12	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: DAI0 Pin 12 Notes: See note[2]
DAI0_PIN13	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: DAI0 Pin 13 Notes: See note[2]
DAI0_PIN14	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: DAI0 Pin 14 Notes: See note[2]
DAI0_PIN15	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: DAI0 Pin 15 Notes: See note[2]

Downloaded from Arrow.com.

Table 25. ADSP-SC57x/ADSP-2157x Designer Quick Reference (Continued)

Signal Name	Type	Driver Type	Internal Termination	Reset Drive	Power Domain	Description and Notes
DAI0_PIN16	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: DAI0 Pin 16 Notes: See note[2]
DAI0_PIN17	InOut	A	Programmable PullUp[3]	none	VDD_EXT	Desc: DAI0 Pin 17 Notes: See note[2]
DAI0_PIN18	InOut	A	Programmable PullUp[3]	none	VDD_EXT	Desc: DAI0 Pin 18 Notes: See note[2]
DAI0_PIN19	InOut	A	Programmable PullUp[3]	none	VDD_EXT	Desc: DAI0 Pin 19 Notes: See note[2]
DAI0_PIN20	InOut	A	Programmable PullUp[3]	none	VDD_EXT	Desc: DAI0 Pin 20 Notes: See note[2]
DMC0_A00	Output	B	none	L	VDD_DMC	Desc: DMC0 Address 0 Notes: No notes
DMC0_A01	Output	B	none	L	VDD_DMC	Desc: DMC0 Address 1 Notes: No notes
DMC0_A02	Output	B	none	L	VDD_DMC	Desc: DMC0 Address 2 Notes: No notes
DMC0_A03	Output	B	none	L	VDD_DMC	Desc: DMC0 Address 3 Notes: No notes
DMC0_A04	Output	B	none	L	VDD_DMC	Desc: DMC0 Address 4 Notes: No notes
DMC0_A05	Output	B	none	L	VDD_DMC	Desc: DMC0 Address 5 Notes: No notes
DMC0_A06	Output	B	none	L	VDD_DMC	Desc: DMC0 Address 6 Notes: No notes
DMC0_A07	Output	B	none	L	VDD_DMC	Desc: DMC0 Address 7 Notes: No notes
DMC0_A08	Output	B	none	L	VDD_DMC	Desc: DMC0 Address 8 Notes: No notes
DMC0_A09	Output	B	none	L	VDD_DMC	Desc: DMC0 Address 9 Notes: No notes
DMC0_A10	Output	B	none	L	VDD_DMC	Desc: DMC0 Address 10 Notes: No notes
DMC0_A11	Output	B	none	L	VDD_DMC	Desc: DMC0 Address 11 Notes: No notes
DMC0_A12	Output	B	none	L	VDD_DMC	Desc: DMC0 Address 12 Notes: No notes
DMC0_A13	Output	B	none	L	VDD_DMC	Desc: DMC0 Address 13 Notes: No notes
DMC0_A14	Output	B	none	L	VDD_DMC	Desc: DMC0 Address 14 Notes: No notes
DMC0_A15	Output	B	none	L	VDD_DMC	Desc: DMC0 Address 15 Notes: No notes
DMC0_BA0	Output	B	none	L	VDD_DMC	Desc: DMC0 Bank Address Input 0 Notes: No notes
DMC0_BA1	Output	B	none	L	VDD_DMC	Desc: DMC0 Bank Address Input 1 Notes: No notes
DMC0_BA2	Output	B	none	L	VDD_DMC	Desc: DMC0 Bank Address Input 2 Notes: No notes

Downloaded from Arrow.com.

Table 25. ADSP-SC57x/ADSP-2157x Designer Quick Reference (Continued)

Signal Name	Type	Driver Type	Internal Termination	Reset Drive	Power Domain	Description and Notes
$\overline{\text{DMC0_CAS}}$	Output	B	none	L	VDD_DMC	Desc: DMC0 Column Address Strobe Notes: No notes
DMC0_CK	Output	C	none	L	VDD_DMC	Desc: DMC0 Clock Notes: No notes
DMC0_CKE	Output	B	none	L	VDD_DMC	Desc: DMC0 Clock enable Notes: No notes
$\overline{\text{DMC0_CK}}$	Output	C	none	L	VDD_DMC	Desc: DMC0 Clock (complement) Notes: No notes
$\overline{\text{DMC0_CS0}}$	Output	B	none	L	VDD_DMC	Desc: DMC0 Chip Select 0 Notes: No notes
DMC0_DQ00	InOut	B	none	none	VDD_DMC	Desc: DMC0 Data 0 Notes: No notes
DMC0_DQ01	InOut	B	none	none	VDD_DMC	Desc: DMC0 Data 1 Notes: No notes
DMC0_DQ02	InOut	B	none	none	VDD_DMC	Desc: DMC0 Data 2 Notes: No notes
DMC0_DQ03	InOut	B	none	none	VDD_DMC	Desc: DMC0 Data 3 Notes: No notes
DMC0_DQ04	InOut	B	none	none	VDD_DMC	Desc: DMC0 Data 4 Notes: No notes
DMC0_DQ05	InOut	B	none	none	VDD_DMC	Desc: DMC0 Data 5 Notes: No notes
DMC0_DQ06	InOut	B	none	none	VDD_DMC	Desc: DMC0 Data 6 Notes: No notes
DMC0_DQ07	InOut	B	none	none	VDD_DMC	Desc: DMC0 Data 7 Notes: No notes
DMC0_DQ08	InOut	B	none	none	VDD_DMC	Desc: DMC0 Data 8 Notes: No notes
DMC0_DQ09	InOut	B	none	none	VDD_DMC	Desc: DMC0 Data 9 Notes: No notes
DMC0_DQ10	InOut	B	none	none	VDD_DMC	Desc: DMC0 Data 10 Notes: No notes
DMC0_DQ11	InOut	B	none	none	VDD_DMC	Desc: DMC0 Data 11 Notes: No notes
DMC0_DQ12	InOut	B	none	none	VDD_DMC	Desc: DMC0 Data 12 Notes: No notes
DMC0_DQ13	InOut	B	none	none	VDD_DMC	Desc: DMC0 Data 13 Notes: No notes
DMC0_DQ14	InOut	B	none	none	VDD_DMC	Desc: DMC0 Data 14 Notes: No notes
DMC0_DQ15	InOut	B	none	none	VDD_DMC	Desc: DMC0 Data 15 Notes: No notes
DMC0_LDM	Output	B	none	L	VDD_DMC	Desc: DMC0 Data Mask for Lower Byte Notes: No notes
DMC0_LDQS	InOut	C	none	none	VDD_DMC	Desc: DMC0 Data Strobe for Lower Byte Notes: External weak pull-down required in LPDDR mode

Downloaded from Arrow.com.

Table 25. ADSP-SC57x/ADSP-2157x Designer Quick Reference (Continued)

Signal Name	Type	Driver Type	Internal Termination	Reset Drive	Power Domain	Description and Notes
$\overline{\text{DMC0_LDQS}}$	InOut	C	none	none	VDD_DMC	Desc: DMC0 Data Strobe for Lower Byte (complement) Notes: No notes
DMC0_ODT	Output	B	none	L	VDD_DMC	Desc: DMC0 On-die termination Notes: No notes
$\overline{\text{DMC0_RAS}}$	Output	B	none	L	VDD_DMC	Desc: DMC0 Row Address Strobe Notes: No notes
$\overline{\text{DMC0_RESET}}$	Output	B	none	L	VDD_DMC	Desc: DMC0 Reset (DDR3 only) Notes: No notes
DMC0_RZQ	a	B	none	none	VDD_DMC	Desc: DMC0 External calibration resistor connection Notes: Applicable for DDR2 and DDR3 only. Pull down using a 34 Ohm resistor.
DMC0_UDM	Output	B	none	L	VDD_DMC	Desc: DMC0 Data Mask for Upper Byte Notes: No notes
DMC0_UDQS	InOut	C	none	none	VDD_DMC	Desc: DMC0 Data Strobe for Upper Byte Notes: External weak pull-down required in LPDDR mode
$\overline{\text{DMC0_UDQS}}$	InOut	C	none	none	VDD_DMC	Desc: DMC0 Data Strobe for Upper Byte (complement) Notes: No notes
DMC0_VREF	a		none	none	VDD_DMC	Desc: DMC0 Voltage Reference Notes: No notes
$\overline{\text{DMC0_WE}}$	Output	B	none	L	VDD_DMC	Desc: DMC0 Write Enable Notes: No notes
GND	g		none	none		Desc: Ground Notes: No notes
HADC0_VIN0	a	NA	none	none	VDD_HADC	Desc: HADC0 Analog Input at channel 0 Notes: Connect to GND through a resistor if not used[4]
HADC0_VIN1	a	NA	none	none	VDD_HADC	Desc: HADC0 Analog Input at channel 1 Notes: Connect to GND through a resistor if not used[4]
HADC0_VIN2	a	NA	none	none	VDD_HADC	Desc: HADC0 Analog Input at channel 2 Notes: Connect to GND through a resistor if not used[4]
HADC0_VIN3	a	NA	none	none	VDD_HADC	Desc: HADC0 Analog Input at channel 3 Notes: Connect to GND through a resistor if not used[4]
HADC0_VIN4	a	NA	none	none	VDD_HADC	Desc: HADC0 Analog Input at channel 4 Notes: Connect to GND through a resistor if not used[4]
HADC0_VIN5	a	NA	none	none	VDD_HADC	Desc: HADC0 Analog Input at channel 5 Notes: Connect to GND through a resistor if not used[4]
HADC0_VIN6	a	NA	none	none	VDD_HADC	Desc: HADC0 Analog Input at channel 6 Notes: Connect to GND through a resistor if not used[4]

Downloaded from Arrow.com.

Table 25. ADSP-SC57x/ADSP-2157x Designer Quick Reference (Continued)

Signal Name	Type	Driver Type	Internal Termination	Reset Drive	Power Domain	Description and Notes
HADC0_VIN7	a	NA	none	none	VDD_HADC	Desc: HADC0 Analog Input at channel 7 Notes: Connect to GND through a resistor if not used[4]
HADC0_VREFN	s	NA	none	none	VDD_HADC	Desc: HADC0 Ground Reference for ADC Notes: Connect to GND if HADC and TMU are not used
HADC0_VREFP	s	NA	none	none	VDD_HADC	Desc: HADC0 External Reference for ADC Notes: No notes
JTG_TCK	Input		PullUp	none	VDD_EXT	Desc: JTAG Clock Notes: No notes
JTG_TDI	Input		PullUp	none	VDD_EXT	Desc: JTAG Serial Data In Notes: No notes
JTG_TDO	Output	A	none	High-Z when $\overline{\text{JTG_TRST}}$ is low, not affected by $\overline{\text{SYS_HWRST}}$	VDD_EXT	Desc: JTAG Serial Data Out Notes: No notes
JTG_TMS	InOut	A	PullUp	none	VDD_EXT	Desc: JTAG Mode Select Notes: No notes
$\overline{\text{JTG_TRST}}$	Input		PullDown	none	VDD_EXT	Desc: JTAG Reset Notes: No notes
MLB0_CLKN	Input	NA	Internal logic ensures that input signal does not float	none	VDD_EXT	Desc: MLB0 Differential Clock (–) Notes: No notes
MLB0_CLKP	Input	NA	Internal logic ensures that input signal does not float	none	VDD_EXT	Desc: MLB0 Differential Clock (+) Notes: No notes
MLB0_DATN	InOut	I	Internal logic ensures that input signal does not float	none	VDD_EXT	Desc: MLB0 Differential Data (–) Notes: No notes
MLB0_DATP	InOut	I	Internal logic ensures that input signal does not float	none	VDD_EXT	Desc: MLB0 Differential Data (+) Notes: No notes
MLB0_SIGN	InOut	I	Internal logic ensures that input signal does not float	none	VDD_EXT	Desc: MLB0 Differential Signal (–) Notes: No notes
MLB0_SIGP	InOut	I	Internal logic ensures that input signal does not float	none	VDD_EXT	Desc: MLB0 Differential Signal (+) Notes: No notes
PA_00	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTA Position 0 Notes: See note[2]
PA_01	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTA Position 1 Notes: See note[2]
PA_02	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTA Position 2 Notes: See note[2]
PA_03	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTA Position 3 Notes: See note[2]
PA_04	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTA Position 4 Notes: See note[2]

Downloaded from Arrow.com.

Table 25. ADSP-SC57x/ADSP-2157x Designer Quick Reference (Continued)

Signal Name	Type	Driver Type	Internal Termination	Reset Drive	Power Domain	Description and Notes
PA_05	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTA Position 5 Notes: See note[2]
PA_06	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTA Position 6 Notes: See note[2]
PA_07	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTA Position 7 Notes: See note[2]
PA_08	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTA Position 8 Notes: See note[2]
PA_09	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTA Position 9 Notes: See note[2]
PA_10	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTA Position 10 Notes: See note[2]
PA_11	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTA Position 11 Notes: See note[2]
PA_12	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTA Position 12 Notes: See note[2]
PA_13	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTA Position 13 Notes: See note[2]
PA_14	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTA Position 14 Notes: See note[2]
PA_15	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTA Position 15 Notes: See note[2]
PB_00	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTB Position 0 Notes: See note[2]
PB_01	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTB Position 1 Notes: See note[2]
PB_02	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTB Position 2 Notes: See note[2]
PB_03	InOut	H	none	none	VDD_EXT	Desc: PORTB Position 3 Notes: Connect to VDD_EXT or GND if not used
PB_04	InOut	H	none	none	VDD_EXT	Desc: PORTB Position 4 Notes: Connect to VDD_EXT or GND if not used
PB_05	InOut	H	none	none	VDD_EXT	Desc: PORTB Position 5 Notes: Connect to VDD_EXT or GND if not used
PB_06	InOut	H	none	none	VDD_EXT	Desc: PORTB Position 6 Notes: Connect to VDD_EXT or GND if not used
PB_07	InOut	H	none	none	VDD_EXT	Desc: PORTB Position 7 Notes: Connect to VDD_EXT or GND if not used
PB_08	InOut	H	none	none	VDD_EXT	Desc: PORTB Position 8 Notes: Connect to VDD_EXT or GND if not used
PB_09	InOut	H	none	none	VDD_EXT	Desc: PORTB Position 9 Notes: Connect to VDD_EXT or GND if not used

Downloaded from Arrow.com.

Table 25. ADSP-SC57x/ADSP-2157x Designer Quick Reference (Continued)

Signal Name	Type	Driver Type	Internal Termination	Reset Drive	Power Domain	Description and Notes
PB_10	InOut	H	none	none	VDD_EXT	Desc: PORTB Position 10 Notes: Connect to VDD_EXT or GND if not used
PB_11	InOut	H	none	none	VDD_EXT	Desc: PORTB Position 11 Notes: Connect to VDD_EXT or GND if not used
PB_12	InOut	H	none	none	VDD_EXT	Desc: PORTB Position 12 Notes: Connect to VDD_EXT or GND if not used
PB_13	InOut	H	none	none	VDD_EXT	Desc: PORTB Position 13 Notes: Connect to VDD_EXT or GND if not used
PB_14	InOut	H	none	none	VDD_EXT	Desc: PORTB Position 14 Notes: Connect to VDD_EXT or GND if not used
PB_15	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTB Position 15 Notes: See note[2]
PC_00	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTC Position 0 Notes: See note[2]
PC_01	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTC Position 1 Notes: See note[2]
PC_02	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTC Position 2 Notes: See note[2]
PC_03	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTC Position 3 Notes: See note[2]
PC_04	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTC Position 4 Notes: See note[2]
PC_05	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTC Position 5 Notes: See note[2]
PC_06	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTC Position 6 Notes: See note[2]
PC_07	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTC Position 7 Notes: See note[2]
PC_08	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTC Position 8 Notes: See note[2]
PC_09	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTC Position 9 Notes: See note[2]
PC_10	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTC Position 10 Notes: See note[2]
PC_11	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTC Position 11 Notes: See note[2]
PC_12	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTC Position 12 Notes: See note[2]
PC_13	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTC Position 13 Notes: See note[2]
PC_14	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTC Position 14 Notes: See note[2]
PC_15	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTC Position 15 Notes: See note[2]

Downloaded from Arrow.com.

Table 25. ADSP-SC57x/ADSP-2157x Designer Quick Reference (Continued)

Signal Name	Type	Driver Type	Internal Termination	Reset Drive	Power Domain	Description and Notes
PD_00	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTD Position 0 Notes: See note[2]
PD_01	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTD Position 1 Notes: See note[2]
PD_02	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTD Position 2 Notes: See note[2]
PD_03	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTD Position 3 Notes: See note[2]
PD_04	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTD Position 4 Notes: See note[2]
PD_05	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTD Position 5 Notes: See note[2]
PD_06	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTD Position 6 Notes: See note[2]
PD_07	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTD Position 7 Notes: See note[2]
PD_08	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTD Position 8 Notes: See note[2]
PD_09	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTD Position 9 Notes: See note[2]
PD_10	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTD Position 10 Notes: See note[2]
PD_11	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTD Position 11 Notes: See note[2]
PD_12	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTD Position 12 Notes: See note[2]
PD_13	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTD Position 13 Notes: See note[2]
PD_14	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTD Position 14 Notes: See note[2]
PD_15	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTD Position 15 Notes: See note[2]
PE_00	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTE Position 0 Notes: See note[2]
PE_01	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTE Position 1 Notes: See note[2]
PE_02	InOut	H	none	none	VDD_EXT	Desc: PORTE Position 2 Notes: Connect to VDD_EXT or GND if not used
PE_03	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTE Position 3 Notes: See note[2]
PE_04	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTE Position 4 Notes: See note[2]
PE_05	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTE Position 5 Notes: See note[2]
PE_06	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTE Position 6 Notes: See note[2]
PE_07	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTE Position 7 Notes: See note[2]

Downloaded from Arrow.com.

Table 25. ADSP-SC57x/ADSP-2157x Designer Quick Reference (Continued)

Signal Name	Type	Driver Type	Internal Termination	Reset Drive	Power Domain	Description and Notes
PE_08	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTE Position 8 Notes: See note[2]
PE_09	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTE Position 9 Notes: See note[2]
PE_10	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTE Position 10 Notes: See note[2]
PE_11	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTE Position 11 Notes: See note[2]
PE_12	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTE Position 12 Notes: See note[2]
PE_13	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTE Position 13 Notes: See note[2]
PE_14	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTE Position 14 Notes: See note[2]
PE_15	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTE Position 15 Notes: See note[2]
PF_00	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTF Position 0 Notes: See note[2]
PF_01	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTF Position 1 Notes: See note[2]
PF_02	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTF Position 2 Notes: See note[2]
PF_03	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTF Position 3 Notes: See note[2]
PF_04	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTF Position 4 Notes: See note[2]
PF_05	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTF Position 5 Notes: See note[2]
PF_06	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTF Position 6 Notes: See note[2]
PF_07	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTF Position 7 Notes: See note[2]
PF_08	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTF Position 8 Notes: See note[2]
PF_09	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTF Position 9 Notes: See note[2]
PF_10	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTF Position 10 Notes: See note[2]
PF_11	InOut	A	Programmable PullUp[1]	none	VDD_EXT	Desc: PORTF Position 11 Notes: See note[2]
SYS_BMODE0	Input	NA	none	none	VDD_EXT	Desc: Boot Mode Control n Notes: No connection not allowed
SYS_BMODE1	Input	NA	none	none	VDD_EXT	Desc: Boot Mode Control n Notes: No connection not allowed
SYS_BMODE2	Input	NA	none	none	VDD_EXT	Desc: Boot Mode Control n Notes: No connection not allowed
SYS_CLKIN0	a	NA	none	none	VDD_EXT	Desc: Clock/Crystal Input Notes: No connection not allowed

Downloaded from Arrow.com.

Table 25. ADSP-SC57x/ADSP-2157x Designer Quick Reference (Continued)

Signal Name	Type	Driver Type	Internal Termination	Reset Drive	Power Domain	Description and Notes
SYS_CLKIN1	a	NA	none	none	VDD_EXT	Desc: Clock/Crystal Input Notes: Connect to GND if not used
SYS_CLKOUT	a	H	none	High-Z when $\overline{\text{SYS_HWRST}}$ and $\overline{\text{JTG_TRST}}$ are both active[5]	VDD_EXT	Desc: Processor Clock Output Notes: No notes
SYS_FAULT	InOut	A	none	none	VDD_EXT	Desc: Active-High Fault Output Notes: Pull down if not used
$\overline{\text{SYS_FAULT}}$	InOut	A	none	none	VDD_EXT	Desc: Active-Low Fault Output Notes: Pull up if not used
$\overline{\text{SYS_HWRST}}$	Input	NA	none	none	VDD_EXT	Desc: Processor Hardware Reset Control Notes: No connection not allowed
$\overline{\text{SYS_RESOUT}}$	Output	A	none	High-Z when $\overline{\text{SYS_HWRST}}$ and $\overline{\text{JTG_TRST}}$ are both active[5]	VDD_EXT	Desc: Reset Output Notes: No notes
SYS_XTAL0	a	NA	none	none	VDD_EXT	Desc: Crystal Output Notes: No notes
SYS_XTAL1	a	NA	none	none	VDD_EXT	Desc: Crystal Output Notes: No notes
TWI0_SCL	InOut	D	none	none	VDD_EXT	Desc: TWI0 Serial Clock Notes: Add external pull-up if used. Connect to GND if not used.
TWI0_SDA	InOut	D	none	none	VDD_EXT	Desc: TWI0 Serial Data Notes: Add external pull-up if used. Connect to GND if not used.
TWI1_SCL	InOut	D	none	none	VDD_EXT	Desc: TWI1 Serial Clock Notes: Add external pull-up if used. Connect to GND if not used.
TWI1_SDA	InOut	D	none	none	VDD_EXT	Desc: TWI1 Serial Data Notes: Add external pull-up if used. Connect to GND if not used.
TWI2_SCL	InOut	D	none	none	VDD_EXT	Desc: TWI2 Serial Clock Notes: Add external pull-up if used. Connect to GND if not used.
TWI2_SDA	InOut	D	none	none	VDD_EXT	Desc: TWI2 Serial Data Notes: Add external pull-up if used. Connect to GND if not used.
USB0_DM	InOut	F	none	none	VDD_USB	Desc: USB0 Data– Notes: Add external pull-down if not used[6]
USB0_DP	InOut	F	none	none	VDD_USB	Desc: USB0 Data + Notes: Add external pull-down if not used[6]
USB0_ID	InOut		none	none	VDD_USB	Desc: USB0 OTG ID Notes: Connect to GND when USB is not used[6]

Downloaded from Arrow.com.

Table 25. ADSP-SC57x/ADSP-2157x Designer Quick Reference (Continued)

Signal Name	Type	Driver Type	Internal Termination	Reset Drive	Power Domain	Description and Notes
USB0_VBC	InOut	E	none	none	VDD_USB	Desc: USB0 VBUS Control Notes: Add external pull-down if not used[6]
USB0_VBUS	InOut	G	none	none	VDD_USB	Desc: USB0 Bus Voltage Notes: Connect to GND when USB is not used[6]
USB0_CLKIN	a		none	none	VDD_USB	Desc: USB0/USB1 Clock/Crystal Input Notes: Connect to GND when USB is not used[6]
USB0_XTAL	a		none	none	VDD_USB	Desc: USB0/USB1 Crystal Notes: No notes
VDD_DMC	s		none	none		Desc: DMC VDD Notes: No notes
VDD_EXT	s		none	none		Desc: External Voltage Domain Notes: No notes
VDD_HADC	s		none	none		Desc: HADC/TMU VDD Notes: Can be left floating if HADC and TMU are not used
VDD_INT	s		none	none		Desc: Internal Voltage Domain Notes: No notes
VDD_USB	s		none	none		Desc: USB VDD Notes: Connect to VDD_EXT when USB is not used

[1] Disabled by default.
[2] Input by default. When unused, terminate externally in hardware or enable the internal pull-up resistor (when applicable) in software. When present, the internal pull-up design holds the internal path from the pins at the expected logic levels. To pull up the external pads to the expected logic levels, use external resistors..
[3] Enabled by default.
[4] All HADC0_VINx pins can be connected directly to GND if HADC and TMU are not used.
[5] Actively driven by processor otherwise.
[6] Guidance also applies to models that do not feature the associated hardware block. See Table 2 or Table 3 for further information.

Downloaded from Arrow.com.

SPECIFICATIONS

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

OPERATING CONDITIONS

Parameter		Conditions	Min	Nominal	Max	Unit
V_{DD_INT}	Internal (Core) Supply Voltage	CCLK ≤ 450 MHz	1.05	1.10	1.15	V
		CCLK ≤ 500 MHz	1.10	1.15	1.20	V
V_{DD_EXT}	External (I/O) Supply Voltage		3.13	3.3	3.47	V
V_{DD_HADC}	Analog Power Supply Voltage		3.13	3.3	3.47	V
V_{DD_DMC}[1]	DDR2/LPDDR Controller Supply Voltage		1.7	1.8	1.9	V
	DDR3 Controller Supply Voltage		1.425	1.5	1.575	V
V_{DD_USB}[2]	USB Supply Voltage		3.13	3.3	3.47	V
V_{DDR_VREF}	DDR2 Reference Voltage Applies to the DMC0_VREF pin		$0.49 \times V_{DD_DMC}$	$0.50 \times V_{DD_DMC}$	$0.51 \times V_{DD_DMC}$	V
V_{HADC_REF}[3]	HADC Reference Voltage		2.5	3.30	V_{DD_HADC}	V
V_{HADC0_VINx}	HADC Input Voltage		0		$V_{HADC_REF} + 0.2$	V
V_{IH}[4]	High Level Input Voltage	$V_{DD_EXT} = 3.47$ V	2.0			V
V_{IHTWI}[5, 6]	High Level Input Voltage	$V_{DD_EXT} = 3.47$ V	$0.7 \times V_{VBUSTWI}$		$V_{VBUSTWI}$	V
V_{IL}[4]	Low Level Input Voltage	$V_{DD_EXT} = 3.13$ V			0.8	V
V_{ILTWI}[5, 6]	Low Level Input Voltage	$V_{DD_EXT} = 3.13$ V			$0.3 \times V_{VBUSTWI}$	V
V_{IL_DDR2}[7]	Low Level Input Voltage	$V_{DD_DMC} = 1.7$ V			$V_{REF} - 0.25$	V
V_{IL_DDR3}[7]	Low Level Input Voltage	$V_{DD_DMC} = 1.425$ V			$V_{REF} - 0.175$	V
V_{IH_DDR2}[7]	High Level Input Voltage	$V_{DD_DMC} = 1.9$ V	$V_{REF} + 0.25$			V
V_{IH_DDR3}[7]	High Level Input Voltage	$V_{DD_DMC} = 1.575$ V	$V_{REF} + 0.175$			V
V_{IL_LPDDR}[8]	Low Level Input Voltage	$V_{DD_DMC} = 1.7$ V			$0.2 \times V_{DD_DMC}$	V
V_{IH_LPDDR}[8]	High Level Input Voltage	$V_{DD_DMC} = 1.9$ V	$0.8 \times V_{DD_DMC}$			V
T_J	Junction Temperature 400-Ball CSP_BGA	$T_{AMBIENT}$ = 0°C to +70°C CCLK ≤ 450 MHz	0		95	°C
T_J	Junction Temperature 400-Ball CSP_BGA	$T_{AMBIENT}$ = –40°C to +100°C CCLK ≤ 450 MHz	–40		+125	°C
T_J	Junction Temperature 176-Lead LQFP-EP	$T_{AMBIENT}$ = 0°C to +70°C CCLK ≤ 450 MHz	0		90	°C
T_J	Junction Temperature 176-Lead LQFP-EP	$T_{AMBIENT}$ = –40°C to +105°C CCLK ≤ 450 MHz	–40		+125	°C
T_J	Junction Temperature 400-Ball CSP_BGA	$T_{AMBIENT}$ = 0°C to +70°C CCLK ≤ 500 MHz	0		100	°C
T_J	Junction Temperature 400-Ball CSP_BGA	$T_{AMBIENT}$ = –40°C to +95°C CCLK ≤ 500 MHz	–40		+125	°C
T_J	Junction Temperature 176-Lead LQFP-EP	$T_{AMBIENT}$ = 0°C to +70°C CCLK ≤ 500 MHz	0		95	°C
T_J	Junction Temperature 176-Lead LQFP-EP	$T_{AMBIENT}$ = –40°C to +100°C CCLK ≤ 500 MHz	–40		+125	°C

Downloaded from Arrow.com.

Parameter		Conditions	Min	Nominal	Max	Unit
AUTOMOTIVE USE ONLY						
T_J	Junction Temperature 400-Ball CSP_BGA (Automotive Grade)	$T_{AMBIENT}$ = –40°C to +105°C CCLK ≤ 450 MHz	–40		+130[9]	°C
T_J	Junction Temperature 176-Lead LQFP-EP (Automotive Grade)	$T_{AMBIENT}$ = –40°C to +105°C CCLK ≤ 450 MHz	–40		+125[9]	°C
T_J	Junction Temperature 400-Ball CSP_BGA (Automotive Grade)	$T_{AMBIENT}$ = –40°C to +105°C CCLK ≤ 500 MHz	–40		+133[9]	°C
T_J	Junction Temperature 176-Lead LQFP-EP (Automotive Grade)	$T_{AMBIENT}$ = –40°C to +105°C CCLK ≤ 500 MHz	–40		+130[9]	°C

[1] Applies to DDR2/DDR3/LPDDR signals.
[2] If not used, V_{DD_USB} must be connected to 3.3 V.
[3] V_{HADC_VREF} must always be less than V_{DD_HADC}.
[4] Parameter value applies to all input and bidirectional pins except the TWI, DMC, USB, and MLB pins.
[5] Parameter applies to TWI signals.
[6] TWI signals are pulled up to V_{BUSTWI}. See Table 26.
[7] This parameter applies to all DMC0 signals in DDR2/DDR3 mode. V_{REF} is the voltage applied to the V_{REF_DMC} pin, nominally V_{DD_DMC}/2.
[8] This parameter applies to DMC0 signals in LPDDR mode.
[9] Automotive application use profile only. Not supported for nonautomotive use. Contact Analog Devices for more information.

Table 26. TWI_VSEL Selections and V_{DD_EXT}/V_{BUSTWI}

TWI_VSEL Selections	V_{DD_EXT} Nominal	V_{BUSTWI}			Unit
		Min	Nominal	Max	
TWI000[1]	3.30	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.

Downloaded from Arrow.com.

Clock Related Operating Conditions

Table 27 describes the core clock, system clock, and peripheral clock timing requirements. The data presented in the table applies to all speed grades except where expressly noted.

Table 27. Clock Operating Conditions

Parameter		Restriction	Min	Typ	Max	Unit
f_{CCLK}	Core Clock Frequency	$f_{CCLK} \geq f_{SYSCLK}$	100		500	MHz
f_{SYSCLK}	SYSCLK Frequency[1]				250	MHz
f_{SCLK0}	SCLK0 Frequency[2]	$f_{SYSCLK} \geq f_{SCLK0}$	30		125	MHz
f_{SCLK1}	SCLK1 Frequency	$f_{SYSCLK} \geq f_{SCLK1}$			125	MHz
f_{DCLK}	LPDDR Clock Frequency				200	MHz
f_{DCLK}	DDR2 Clock Frequency				400	MHz
f_{DCLK}	DDR3 Clock Frequency				450	MHz
f_{OCLK}	Output Clock Frequency[3]				250	MHz
$f_{SYS_CLKOUTJ}$	SYS_CLKOUT Period Jitter[4, 5]			±1		%
$f_{PCLKPROG}$	Programmed PPI Clock When Transmitting Data and Frame Sync				62.5	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[6, 7]	$f_{PCLKEXT} \leq f_{SCLK0}$			62.5	MHz
$f_{PCLKEXT}$	External PPI Clock Transmitting Data or Frame Sync[6, 7]	$f_{PCLKEXT} \leq f_{SCLK0}$			50	MHz
$f_{LCLKTPROG}$	Programmed Link Port Transmit Clock				125	MHz
$f_{LCLKREXT}$	External Link Port Receive Clock[6, 7]	$f_{LCLKEXT} \leq f_{SCLK0}$			125	MHz
$f_{SPTCLKPROG}$	Programmed SPT Clock When Transmitting Data and Frame Sync				62.5	MHz
$f_{SPTCLKPROG}$	Programmed SPT Clock When Receiving Data or Frame Sync				31.25	MHz
$f_{SPTCLKEXT}$	External SPT Clock When Receiving Data and Frame Sync[6, 7]	$f_{SPTCLKEXT} \leq f_{SCLK0}$			62.5	MHz
$f_{SPTCLKEXT}$	External SPT Clock Transmitting Data or Frame Sync[6, 7]	$f_{SPTCLKEXT} \leq f_{SCLK0}$			31.25	MHz
$f_{SPICLKPROG}$	Programmed SPI2 Clock When Transmitting Data				75	MHz
	Programmed SPI0, SPI1 Clock When Transmitting Data				62.5	MHz
$f_{SPICLKPROG}$	Programmed SPI2 Clock When Receiving Data				75	MHz
	Programmed SPI0, SPI1 Clock When Receiving Data				62.5	MHz
$f_{SPICLKEXT}$	External SPI2 Clock When Receiving Data[6, 7]	$f_{SPICLKEXT} \leq f_{SCLK1}$			75	MHz
	External SPI0, SPI1 Clock When Receiving Data[6, 7]	$f_{SPICLKEXT} \leq f_{SCLK0}$			62.5	MHz
$f_{SPICLKEXT}$	External SPI2 Clock When Transmitting Data[6, 7]	$f_{SPICLKEXT} \leq f_{SCLK1}$			45	MHz
	External SPI0, SPI1 Clock When Transmitting Data[6,7]	$f_{SPICLKEXT} \leq f_{SCLK0}$			62.5	MHz
$f_{ACLKPROG}$	Programmed ACM Clock				56.25	MHz

[1] When using MLB, there is a requirement that the f_{SYSCLK} value must be a minimum of 100 MHz for both 3-pin and 6-pin modes and for all supported speeds.
[2] The minimum frequency for SCLK0 applies only when using the USB.
[3] f_{OCLK} must not exceed f_{SCLK0} when selected as SYS_CLKOUT.
[4] 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 can be higher or lower than this typical specification for each end application.
[5] The value in the Typ field is the percentage of the SYS_CLKOUT period.
[6] 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.
[7] The peripheral external clock frequency must also be less than or equal to the f_{SCLK} (f_{SCLK0} or f_{SCLK1}) that clocks the peripheral.

Downloaded from Arrow.com.

Table 28. PLL Operating Conditions

Parameter		Min	Max	Unit
f_{PLLCLK}	PLL Clock Frequency	200	1000	MHz

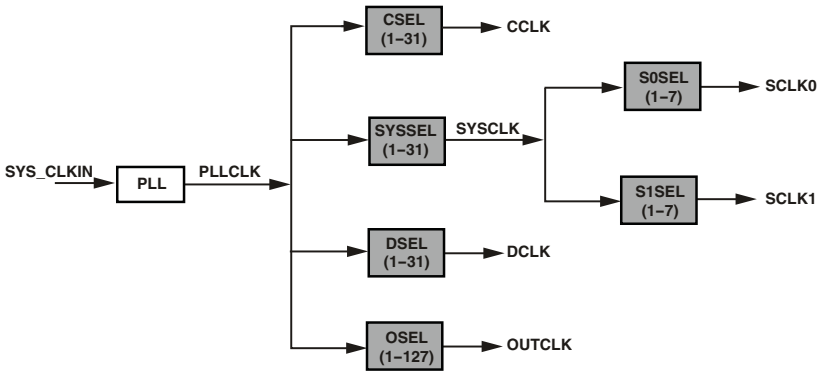


Figure 7. Clock Relationships and Divider Values

Downloaded from Arrow.com.

ELECTRICAL CHARACTERISTICS

Parameter		Conditions	Min	Typ	Max	Unit
V_{OH}[1]	High Level Output Voltage	At V_{DD_EXT} = minimum, I_{OH} = –1.0 mA[2]	2.4			V
V_{OL}[1]	Low Level Output Voltage	At V_{DD_EXT} = minimum, I_{OL} = 1.0 mA[2]			0.4	V
V_{OH_DDR2}[3]	High Level Output Voltage for DDR2 DS = 40 Ω	At V_{DD_DDR} = minimum, I_{OH} = –5.8 mA	1.38			V
V_{OL_DDR2}[3]	Low Level Output Voltage for DDR2 DS = 40 Ω	At V_{DD_DDR} = minimum, I_{OL} = 5.8 mA			0.32	V
V_{OH_DDR2}[3]	High Level Output Voltage for DDR2 DS = 60 Ω	At V_{DD_DDR} = minimum, I_{OH} = –3.4 mA	1.38			V
V_{OL_DDR2}[3]	Low Level Output Voltage for DDR2 DS = 60 Ω	At V_{DD_DDR} = minimum, I_{OL} = 3.4 mA			0.32	V
V_{OH_DDR3}[4]	High Level Output Voltage for DDR3 DS = 40 Ω	At V_{DD_DDR} = minimum, I_{OH} = –5.8 mA	1.105			V
V_{OL_DDR3}[4]	Low Level Output Voltage for DDR3 DS = 40 Ω	At V_{DD_DDR} = minimum, I_{OL} = 5.8 mA			0.32	V
V_{OH_DDR3}[4]	High Level Output Voltage for DDR3 DS = 60 Ω	At V_{DD_DDR} = minimum, I_{OH} = –3.4 mA	1.105			V
V_{OL_DDR3}[4]	Low Level Output Voltage for DDR3 DS = 60 Ω	At V_{DD_DDR} = minimum, I_{OL} = 3.4 mA			0.32	V
V_{OH_LPDDR}[5]	High Level Output Voltage for LPDDR	At V_{DD_DDR} = minimum, I_{OH} = –6.0 mA	1.38			V
V_{OL_LPDDR}[5]	Low Level Output Voltage for LPDDR	At V_{DD_DDR} = minimum, I_{OL} = 6.0 mA			0.32	V
I_{IH}[6, 7]	High Level Input Current	At V_{DD_EXT} = maximum, V_{IN} = V_{DD_EXT} maximum			10	µA
I_{IL}[6]	Low Level Input Current	At V_{DD_EXT} = maximum, V_{IN} = 0 V			10	µA
I_{IL_PU}[7]	Low Level Input Current Pull-Up	At V_{DD_EXT} = maximum, V_{IN} = 0 V			200	µA
I_{IH_PD}[8]	High Level Input Current Pull-Down	At V_{DD_EXT} = maximum, V_{IN} = V_{DD_EXT} maximum			200	µA
I_{OZH}[9]	Three-State Leakage Current	At V_{DD_EXT}/V_{DD_DDR} = maximum, V_{IN} = V_{DD_EXT}/V_{DD_DDR} maximum			10	µA
I_{OZL}[9]	Three-State Leakage Current	At V_{DD_EXT}/V_{DD_DDR} = maximum, V_{IN} = 0 V			10	µA
C_{IN}[10]	Input Capacitance	T_{CASE} = 25°C			5	pF
I_{DD_IDLE}	V_{DD_INT} Current in Idle	f_{CCLK} = 450 MHz ASF_{SHARC1} = 0.32 ASF_{SHARC2} = 0.32 ASF_{A5} = 0.25 f_{SYSCLK} = 225 MHz $f_{SCLK0/1}$ = 112.5 MHz (Other clocks are disabled) No Peripheral or DMA activity T_J = 25°C V_{DD_INT} = 1.1 V		410		mA

Downloaded from Arrow.com.

Parameter		Conditions	Min	Typ	Max	Unit
I_{DD_IDLE}	V_{DD_INT} Current in Idle	f_{CCLK} = 500 MHz ASF_{SHARC1} = 0.32 ASF_{SHARC2} = 0.32 ASF_{A5} = 0.25 f_{SYSCLK} = 250 MHz $f_{SCLK0/1}$ = 125 MHz (Other clocks are disabled) No Peripheral or DMA activity T_J = 25°C V_{DD_INT} = 1.15 V		477		mA
I_{DD_TYP}	V_{DD_INT} Current	f_{CCLK} = 450 MHz ASF_{SHARC1} = 1.0 ASF_{SHARC2} = 1.0 ASF_{A5} = 0.67 f_{SYSCLK} = 225 MHz $f_{SCLK0/1}$ = 112.5 MHz (Other clocks are disabled) DMA data rate = 600 MB/s T_J = 25°C V_{DD_INT} = 1.1 V		890		mA
I_{DD_TYP}	V_{DD_INT} Current	f_{CCLK} = 500 MHz ASF_{SHARC1} = 1.0 ASF_{SHARC2} = 1.0 ASF_{A5} = 0.67 f_{SYSCLK} = 250 MHz $f_{SCLK0/1}$ = 125 MHz (Other clocks are disabled) DMA data rate = 600 MB/s T_J = 25°C V_{DD_INT} = 1.15 V		1031		mA
I_{DD_INT}[11]	V_{DD_INT} Current	f_{CCLK} > 0 MHz $f_{SCLK0/1}$ ≥ 0 MHz			See $I_{DD_INT_TOT}$ equation in the Total Internal Power Dissipation section.	mA

[1] Applies to all output and bidirectional pins except TWI, DMC, USB, and MLB.
[2] See the Output Drive Currents section for typical drive current capabilities.
[3] Applies to all DMC output and bidirectional signals in DDR2 mode.
[4] Applies to all DMC output and bidirectional signals in DDR3 mode.
[5] Applies to all DMC output and bidirectional signals in LPDDR mode.
[6] Applies to input pins: SYS_BMODE0-2, SYS_CLKIN0, SYS_CLKIN1, $\overline{\text{SYS_HWRST}}$, JTG_TDI, JTG_TMS, and USB0_CLKIN.
[7] Applies to input pins with internal pull-ups: JTG_TDI, JTG_TMS, and JTG_TCK.
[8] Applies to signals: JTAG_TRST, USB0_VBUS.
[9] Applies to signals: PA0-15, PB0-15, PC0-15, PD0-15, PE0-15, PF0-11, DAI0_PINx, DMC0_DQx, DMC0_LDQS, DMC0_UDQS, $\overline{\text{DMC0_LDQS}}$, $\overline{\text{DMC0_UDQS}}$, SYS_FAULT, $\overline{\text{SYS_FAULT}}$, JTG_TDO, USB0_ID, USB0_DM, USB0_DP, and USB0_VBC.
[10] Applies to all signal pins.
[11] See "Estimating Power for ADSP-SC57x/2157x SHARC+ Processors" (EE-397) for further information.

Downloaded from Arrow.com.

Total Internal Power Dissipation

Total power dissipation has two components:

1. Static, including leakage current
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.

$$I_{DD_INT_TOT} = I_{DD_INT_STATIC} + I_{DD_INT_CCLK_SHARC1_DYN} + I_{DD_INT_CCLK_SHARC2_DYN} + I_{DD_INT_CCLK_A5_DYN} + I_{DD_INT_DCLK_DYN} + I_{DD_INT_SYSCLK_DYN} + I_{DD_INT_SCLK0_DYN} + I_{DD_INT_SCLK1_DYN} + I_{DD_INT_OCLK_DYN} + I_{DD_INT_ACCL_DYN} + I_{DD_INT_USB_DYN} + I_{DD_INT_MLB_DYN} + I_{DD_INT_EMAC_DYN} + I_{DD_INT_DMA_DR_DYN}$$

$I_{DD_INT_STATIC}$ is the sole contributor to the static power dissipation component and is specified as a function of voltage (V_{DD_INT}) and junction temperature (T_J) in Table 29.

Table 29. Static Current—$I_{DD_INT_STATIC}$ (mA)

T_J (°C)	Voltage (V_{DD_INT})			
	1.05	1.10	1.15	1.20
–40	4	5	6	7
–20	6	8	9	11
–10	8	10	12	14
0	11	13	16	18
+10	15	17	20	24
+25	22	26	30	35
+40	34	39	45	52
+55	50	57	66	76
+70	74	84	95	109
+85	107	121	137	155
+100	153	172	194	218
+105	173	195	219	246
+115	217	243	273	305
+125	271	302	338	377
+133	323	359	400	446

The other 13 addends in the $I_{DD_INT_TOT}$ equation comprise the dynamic power dissipation component and fall into four broad categories: application-dependent currents, clock currents, currents from high speed peripheral operation, and data transmission currents.

Application Dependent Current

The application dependent currents include the dynamic current in the core clock domain of the two SHARC+ cores and the ARM Cortex-A5 core, as well as the dynamic current in the accelerator block.

Dynamic current consumed by the core is subject to an activity scaling factor (ASF) that represents application code running on the processor cores (see Table 30 and Table 31). The ASF is combined with the CCLK frequency and V_{DD_INT} dependent dynamic current data in Table 32 and Table 33, respectively, to calculate this portion of the total dynamic power dissipation component.

$I_{DD_INT_CCLK_SHARC1_DYN}$ = Table 32 × ASF_{SHARC1}
$I_{DD_INT_CCLK_SHARC2_DYN}$ = Table 32 × ASF_{SHARC2}
$I_{DD_INT_CCLK_A5_DYN}$ = Table 33 × ASF_{A5}

Table 30. Activity Scaling Factors for the SHARC+® Core 1 and Core 2 (ASF_{SHARC1} and ASF_{SHARC2})

I_{DD_INT} Power Vector	ASF
$I_{DD\text{-}IDLE}$	0.32
$I_{DD\text{-}NOP}$	0.55
$I_{DD\text{-}TYP_3070}$	0.75
$I_{DD\text{-}TYP_5050}$	0.88
$I_{DD\text{-}TYP_7030}$	1.00
$I_{DD\text{-}PEAK_100}$	1.13

Table 31. Activity Scaling Factors for the ARM® Cortex®-A5 Core (ASF_{A5})

I_{DD_INT} Power Vector	ASF
$I_{DD\text{-}IDLE}$	0.25
$I_{DD\text{-}DHRYSTONE}$	0.67
$I_{DD\text{-}TYP_2575}$	0.53
$I_{DD\text{-}TYP_5050}$	0.75
$I_{DD\text{-}TYP_7525}$	1.00
$I_{DD\text{-}PEAK_100}$	1.27

Downloaded from Arrow.com.

Table 32. Dynamic Current for Each SHARC+®Core (mA, with ASF = 1.00)[1]

f_{CCLK} (MHz)	Voltage (V_{DD_INT})			
	1.05	1.10	1.15	1.20
500	N/A	347	362	378
450	298	312	326	340
400	265	277	290	302
350	232	243	254	265
300	198	208	217	227
250	165	173	181	189
200	132	139	145	151
150	99	104	109	113
100	66	69	72	76

[1] N/A means not applicable.

Table 33. Dynamic Current for the ARM® Cortex®-A5 Core (mA, with ASF = 1.00)[1]

f_{CCLK} (MHz)	Voltage (V_{DD_INT})			
	1.05	1.10	1.15	1.20
500	N/A	88	92	96
450	76	79	83	86
400	67	70	74	77
350	59	62	64	67
300	50	53	55	58
250	42	44	46	48
200	34	35	37	39
150	25	26	28	29
100	17	18	18	19

[1] N/A means not applicable.

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.

$I_{DD_INT_SYSCLK_DYN}$ (mA) = 0.52 × f_{SYSCLK} (MHz) × V_{DD_INT} (V)

$I_{DD_INT_SCLK0_DYN}$ (mA) = 0.28 × f_{SCLK0} (MHz) × V_{DD_INT} (V)

$I_{DD_INT_SCLK1_DYN}$ (mA) = 0.013 × f_{SCLK1} (MHz) × V_{DD_INT} (V)

$I_{DD_INT_DCLK_DYN}$ (mA) = 0.08 × f_{DCLK} (MHz) × V_{DD_INT} (V)

$I_{DD_INT_OCLK_DYN}$ (mA) = 0.015 × f_{OCLK} (MHz) × V_{DD_INT} (V)

Current from High Speed Peripheral Operation

The following modules contribute significantly to power dissipation, and a single term is added when they are used.

$I_{DD_INT_USB_DYN}$ = 9.6 mA (if USB is enabled in HS mode)

$I_{DD_INT_MLB_DYN}$ = 10 mA (if MLB 6-pin interface is enabled)

$I_{DD_INT_EMAC_DYN}$ = 10 mA (if EMAC is enabled)

Data Transmission Current

The data transmission current represents the power dissipated when moving data throughout the system via DMA. This current is proportional to the data rate. Refer to the power calculator available with "Estimating Power for ADSP-SC57x/2157x SHARC+ Processors" (EE-397) to estimate $I_{DD_INT_DMA_DR_DYN}$ based on the bandwidth of the data transfer.

Downloaded from Arrow.com.

HADC

HADC Electrical Characteristics

Table 34. HADC Electrical Characteristics

Parameter	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	60	µA

HADC DC Accuracy

Table 35. HADC DC Accuracy for CSP_BGA[1]

Parameter	Typ	Unit[2]
Resolution	9	Bits
No Missing Codes (NMC) – Unrestricted	9	Bits
No Missing Codes (NMC) – Pin Restrictions[3]	10	Bits
Integral Nonlinearity (INL)	±2	LSB
Differential Nonlinearity (DNL)	±2	LSB
Offset Error	±5	LSB
Offset Error Matching	±6	LSB
Gain Error	±4	LSB
Gain Error Matching	±4	LSB

[1] See the Operating Conditions section for the HADC0_VINx specification.
[2] LSB = HADC0_VREFP ÷ 512.
[3] Pin restrictions required: pins DAI18, DAI19, and DAI20 must be programmed to inputs and a static (non-switching) signal applied to the pins.

Table 36. HADC DC Accuracy for LQFP_EP[1]

Parameter	Typ	Unit[2]
Resolution	7	Bits
No Missing Codes (NMC) – Unrestricted	7	Bits
No Missing Codes (NMC) – Pin Restrictions[3]	9	Bits
Integral Nonlinearity (INL)	±2	LSB
Differential Nonlinearity (DNL)	±2	LSB
Offset Error	±5	LSB
Offset Error Matching	±6	LSB
Gain Error	±4	LSB
Gain Error Matching	±4	LSB

[1] See the Operating Conditions section for the HADC0_VINx specification.
[2] LSB = HADC0_VREFP ÷ 128.
[3] Pin restrictions required: pins DAI18, DAI19, and DAI20 must be programmed to inputs and a static (non-switching) signal applied to the pins.

HADC Timing Specifications

Table 37. HADC Timing Specifications

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

[1] Refer to the ADSP-SC57x/ADSP-2157x SHARC+ Processor Hardware Reference for additional information about T_{SAMPLE}.

TMU

TMU Characteristics

Table 38. TMU Characteristics

Parameter	Typ	Unit
Resolution	1	°C
Accuracy	±8	°C

Table 39. TMU Gain and Offset

Junction Temperature Range	TMU_GAIN	TMU_OFFSET
−40°C to +40°C	Contact Analog Devices, Inc.	
40°C to 85°C	Contact Analog Devices, Inc.	
85°C to 133°C	Contact Analog Devices, Inc.	

Downloaded from Arrow.com.

ABSOLUTE MAXIMUM RATINGS

Stresses at or above those listed in Table 40 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 40. Absolute Maximum Ratings

Parameter	Rating
Internal (Core) Supply Voltage (V_{DD_INT})	–0.33 V to +1.26 V
External (I/O) Supply Voltage (V_{DD_EXT})	–0.33 V to +3.60 V
DDR2/LPDDR Controller Supply Voltage (V_{DD_DMC})	–0.33 V to +1.90 V
DDR3 Controller Supply Voltage (V_{DD_DMC})	–0.33 V to +1.60 V
DDR2 Reference Voltage (V_{DDR_VREF})	–0.33 V to +1.90 V
USB PHY Supply Voltage (V_{DD_USB})	–0.33 V to +3.60 V
HADC Supply Voltage (V_{DD_HADC})	–0.33 V to +3.60 V
HADC Reference Voltage (V_{HADC_REF})	–0.33 V to +3.60 V
DDR2/LPDDR Input Voltage[1]	–0.33 V to +1.90 V
DDR3 Input Voltage[1]	–0.33 V to +1.60 V
Digital Input Voltage[1, 2]	–0.33 V to +3.60 V
TWI Input Voltage[1, 3]	–0.33 V to +5.50 V
USB0_Dx Input Voltage[1, 4]	–0.33 V to +5.25 V
USB0_VBUS Input Voltage[1, 4]	–0.33 V to +6 V
Output Voltage Swing	–0.33 V to V_{DD_EXT} +0.5 V
Analog Input Voltage[5]	–0.2 V to V_{DD_HADC} +0.2 V
I_{OH}/I_{OL} Current per Signal[2]	6 mA (maximum)
Storage Temperature Range	–65°C to +150°C
Junction Temperature While Biased	133°C

[1] Applies only when the related power supply (V_{DD_DMC}, V_{DD_EXT}, or V_{DD_USB}) is within specification. When the power supply is below specification, the range is the voltage being applied to that power domain ± 0.2 V.

[2] Applies to 100% transient duty cycle.

[3] Applies to TWI_SCL and TWI_SDA.

[4] If the USB is not used, connect these pins according to Table 25.

[5] Applies only when V_{DD_HADC} is within specifications and ≤ 3.4 V. When V_{DD_HADC} is within specifications and > 3.4 V, the maximum rating is 3.6 V. When V_{DD_HADC} is below specifications, the range is V_{DD_HADC} ± 0.2 V.

ESD CAUTION

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.

Downloaded from Arrow.com.

TIMING SPECIFICATIONS

Specifications are subject to change without notice.

Power-Up Reset Timing

Table 41 and Figure 8 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 8, $V_{DD_SUPPLIES}$ are V_{DD_INT}, V_{DD_EXT}, V_{DD_DMC}, V_{DD_USB}, and V_{DD_HADC}.

Table 41. Power-Up Reset Timing

Parameter		Min	Max	Unit
Timing Requirement				
$t_{RST_IN_PWR}$	$\overline{SYS_HWRST}$ Deasserted after $V_{DD_SUPPLIES}$ (V_{DD_INT}, V_{DD_EXT}, V_{DD_DMC}, V_{DD_USB}, V_{DD_HADC}) and SYS_CLKINx are Stable and within Specification	11 × t_{CKIN}		ns

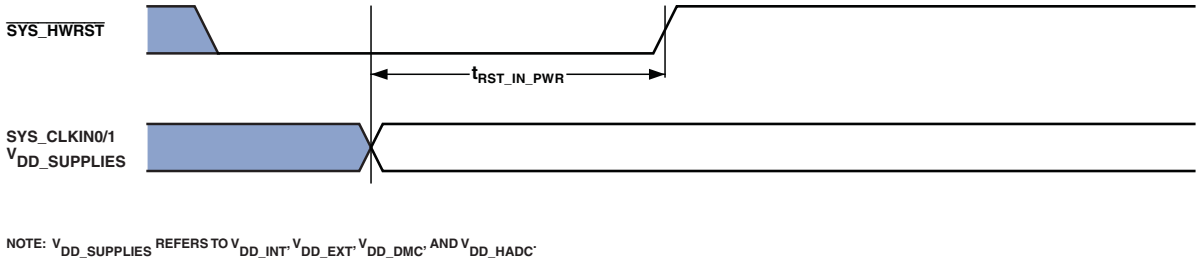


Figure 8. Power-Up Reset Timing

Downloaded from Arrow.com.

Clock and Reset Timing

Table 42 and Figure 9 describe clock and reset operations related to the CGU and RCU. Per the CCLK, SYSCLK, SCLK, DCLK, and OCLK timing specifications in Table 27, combinations of SYS_CLKIN and clock multipliers must not select clock rates in excess of the maximum instruction rate of the processor.

Table 42. Clock and Reset Timing

Parameter		Min	Max	Unit
Timing Requirements				
f_{CKIN}	SYS_CLKINx Frequency (Crystal)[1, 2, 3]	20	50	MHz
	SYS_CLKINx Frequency (External CLKIN)[1, 2, 3]	20	50	MHz
t_{CKINL}	CLKIN Low Pulse[1]	10		ns
t_{CKINH}	CLKIN High Pulse[1]	10		ns
t_{WRST}	$\overline{\text{RESET}}$ Asserted Pulse Width Low[4]	11 × t_{CKIN}		ns

[1] Applies to PLL bypass mode and PLL nonbypass mode.
[2] The t_{CKIN} period (see Figure 9) equals $1/f_{CKIN}$.
[3] If the CGU_CTL.DF bit is set, the minimum f_{CKIN} specification is 40 MHz.
[4] Applies after power-up sequence is complete. See Table 41 and Figure 8 for power-up reset timing.

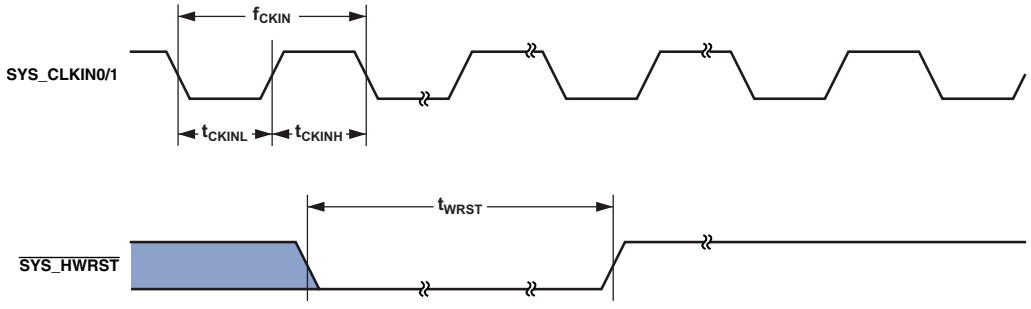


Figure 9. Clock and Reset Timing

Downloaded from Arrow.com.

DDR2 SDRAM Clock and Control Cycle Timing

Table 43 and Figure 10 show DDR2 SDRAM clock and control cycle timing, related to the DMC.

Table 43. DDR2 SDRAM Clock and Control Cycle Timing, V_{DD_DMC} Nominal 1.8 V

Parameter		400 MHz[1] Min	400 MHz[1] Max	Unit
Switching Characteristics				
t_{CK}	Clock Cycle Time (CL = 2 Not Supported)	2.5		ns
$t_{CH(abs)}$[2]	Minimum Clock Pulse Width	0.48	0.52	t_{CK}
$t_{CL(abs)}$[2]	Maximum Clock Pulse Width	0.48	0.52	t_{CK}
t_{IS}	Control/Address Setup Relative to DMC0_CK Rise	175		ps
t_{IH}	Control/Address Hold Relative to DMC0_CK Rise	250		ps

[1] To ensure proper operation of DDR2, all the DDR2 requirements must be strictly followed. See "Interfacing DDR3/DDR2/LPDDR Memory to ADSP-SC5xx/215xx Processors" (EE-387).
[2] As per JESD79-2E definition.

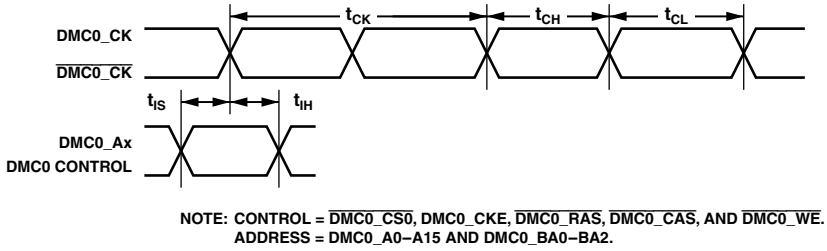


Figure 10. DDR2 SDRAM Clock and Control Cycle Timing

Downloaded from Arrow.com.

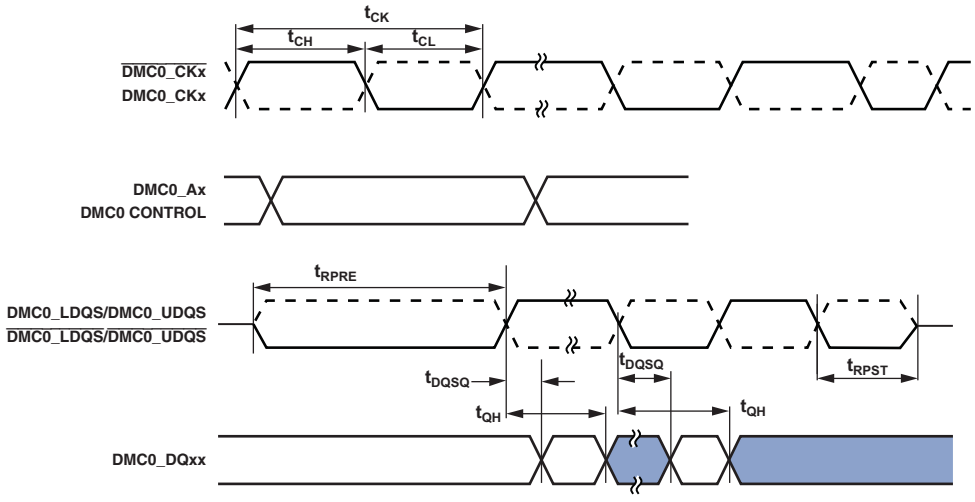
DDR2 SDRAM Read Cycle Timing

Table 44 and Figure 11 show DDR2 SDRAM read cycle timing, related to the DMC.

Table 44. DDR2 SDRAM Read Cycle Timing, V_{DD_DMC} Nominal 1.8 V

Parameter		400 MHz[1] Min	400 MHz[1] Max	Unit
Timing Requirements				
t_{DQSQ}	DMC0_DQS to DMC0_DQ Skew for DMC0_DQS and Associated DMC0_DQxx Signals		0.2	ns
t_{QH}	DMC0_DQxx, DMC0_DQS Output Hold Time From DMC0_DQS	0.8		ns
t_{RPRE}	Read Preamble	0.9		t_{CK}
t_{RPST}	Read Postamble	0.4		t_{CK}

[1] To ensure proper operation of DDR2, all the DDR2 requirements must be strictly followed. See "Interfacing DDR3/DDR2/LPDDR Memory to ADSP-SC5xx/215xx Processors" (EE-387).



NOTE: CONTROL = DMC0_CS0, DMC0_CKE, DMC0_RAS, DMC0_CAS, AND DMC0_WE.
ADDRESS = DMC0_A00–13 AND DMC0_BA0–1.

Figure 11. DDR2 SDRAM Controller Input AC Timing

Downloaded from Arrow.com.

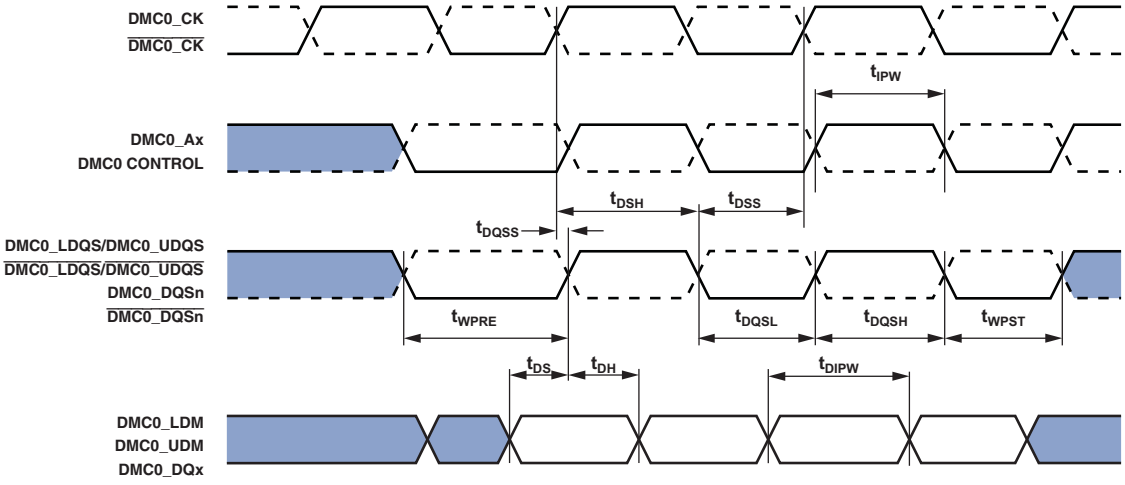
DDR2 SDRAM Write Cycle Timing

Table 45 and Figure 12 show DDR2 SDRAM write cycle timing, related to the DMC.

Table 45. DDR2 SDRAM Write Cycle Timing, V_{DD_DMC} Nominal 1.8 V

Parameter		400 MHz[1] Min	400 MHz[1] Max	Unit
Switching Characteristics				
t_{DQSS}	DMC0_DQS Latching Rising Transitions to Associated Clock Edges[2]	–0.15	+0.15	t_{CK}
t_{DS}	Last Data Valid to DMC0_DQS Delay	0.1		ns
t_{DH}	DMC0_DQS to First Data Invalid Delay	0.15		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	0.35		t_{CK}
t_{DQSL}	DMC0_DQS Input 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 requirements must be strictly followed. See "Interfacing DDR3/DDR2/LPDDR Memory to ADSP-SC5xx/215xx Processors" (EE-387).
[2] Write command to first DMC0_DQS delay = WL × t_{CK} + t_{DQSS}.



NOTE: CONTROL = DMC0_CS0, DMC0_CKE, DMC0_RAS, DMC0_CAS, AND DMC0_WE.
ADDRESS = DMC0_A00–13 AND DMC0_BA0–1.

Figure 12. DDR2 SDRAM Controller Output AC Timing

Downloaded from Arrow.com.

Mobile DDR (LPDDR) SDRAM Clock and Control Cycle Timing

Table 46 and Figure 13 show mobile DDR SDRAM clock and control cycle timing, related to the DMC.

Table 46. Mobile DDR SDRAM Clock and Control Cycle Timing, V_{DD_DMC} Nominal 1.8 V

Parameter		200 MHz[1] Min	200 MHz[1] 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		ns
t_{IH}	Control/Address Hold Relative to DMC0_CK Rise	1		ns

[1] To ensure proper operation of LPDDR, all the LPDDR requirements must be strictly followed. See "Interfacing DDR3/DDR2/LPDDR Memory to ADSP-SC5xx/215xx Processors" (EE-387).

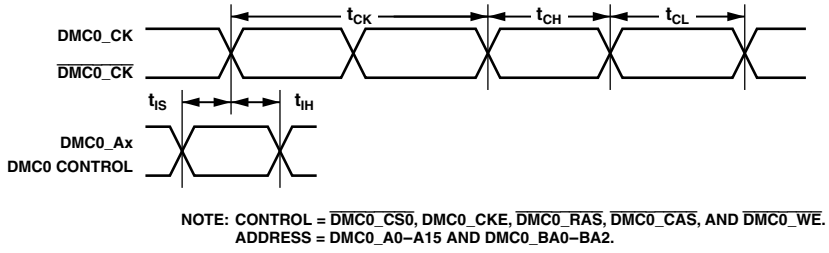


Figure 13. Mobile DDR SDRAM Clock and Control Cycle Timing

Downloaded from Arrow.com.

Mobile DDR SDRAM Read Cycle Timing

Table 47 and Figure 14 show mobile DDR SDRAM read cycle timing, related to the DMC.

Table 47. Mobile DDR SDRAM Read Cycle Timing, V_{DD_DMC} Nominal 1.8 V

Parameter		200 MHz[1] Min	200 MHz[1] Max	Unit
Timing Requirements				
t_{QH}	DMC0_DQ, DMC0_DQS Output Hold Time From DMC0_DQS	1.75		ns
t_{DQSQ}	DMC0_DQS to DMC0_DQ Skew for DMC0_DQS and Associated DMC0_DQ Signals		0.4	ns
t_{RPRE}	Read Preamble	0.9	1.1	t_{CK}
t_{RPST}	Read Postamble	0.4	0.6	t_{CK}

[1] To ensure proper operation of LPDDR, all the LPDDR requirements must be strictly followed. See "Interfacing DDR3/DDR2/LPDDR Memory to ADSP-SC5xx/215xx Processors" (EE-387).

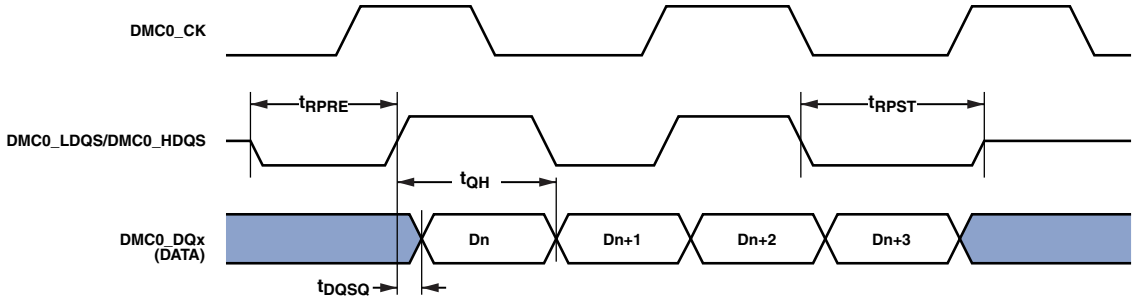


Figure 14. Mobile DDR SDRAM Controller Input AC Timing

Downloaded from Arrow.com.

Mobile DDR SDRAM Write Cycle Timing

Table 48 and Figure 15 show mobile DDR SDRAM write cycle timing, related to the DMC.

Table 48. Mobile DDR 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	0.75	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] To ensure proper operation of LPDDR, all the LPDDR requirements must be strictly followed. See "Interfacing DDR3/DDR2/LPDDR Memory to ADSP-SC5xx/215xx Processors" (EE-387).
[2] Write command to first DMC0_DQS delay = WL × t_{CK} + t_{DQSS}.

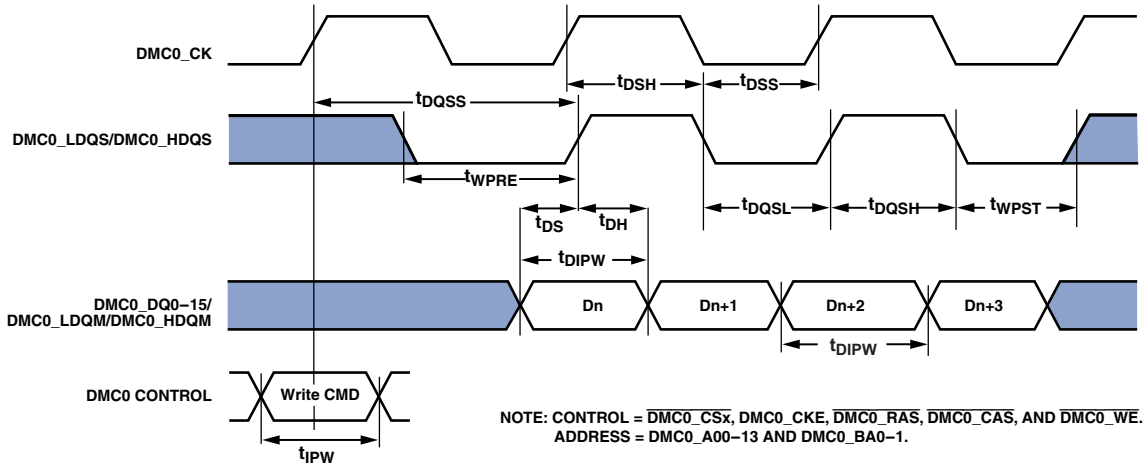


Figure 15. Mobile DDR SDRAM Controller Output AC Timing

Downloaded from Arrow.com.

DDR3 SDRAM Clock and Control Cycle Timing

Table 49 and Figure 16 show mobile DDR3 SDRAM clock and control cycle timing, related to the DMC.

Table 49. DDR3 SDRAM Clock and Control Cycle Timing, V_{DD_DMC} Nominal 1.5 V

Parameter		450 MHz[1] Min	450 MHz[1] Max	Unit
Switching Characteristics				
t_{CK}	Clock Cycle Time (CL = 2 Not Supported)	2.22		ns
$t_{CH(abs)}$[2]	Minimum Clock Pulse Width	0.47	0.53	t_{CK}
$t_{CL(abs)}$[2]	Maximum Clock Pulse Width	0.47	0.53	t_{CK}
t_{IS}	Control/Address Setup Relative to DMC0_CK Rise	0.2		ns
t_{IH}	Control/Address Hold Relative to DMC0_CK Rise	0.275		ns

[1] To ensure proper operation of the DDR3, all the DDR3 requirements must be strictly followed. See "Interfacing DDR3/DDR2/LPDDR Memory to ADSP-SC5xx/215xx Processors" (EE-387).
[2] As per JESD79-3F definition.

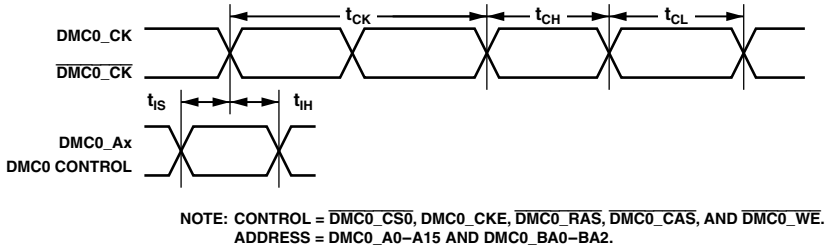


Figure 16. DDR3 SDRAM Clock and Control Cycle Timing

Downloaded from Arrow.com.

DDR3 SDRAM Read Cycle Timing

Table 50 and Figure 17 show mobile DDR3 SDRAM read cycle timing, related to the DMC.

Table 50. DDR3 SDRAM Read Cycle Timing, V_{DD_DMC} Nominal 1.5 V

Parameter		450 MHz[1] Min	450 MHz[1] Max	Unit
Timing Requirements				
t_{DQSQ}	DMC0_DQS to DMC0_DQ Skew for DMC0_DQS and Associated DMC0_DQ Signals		0.15	ns
t_{QH}	DMC0_DQ, DMC0_DQS Output Hold Time From DMC0_DQS	0.38		t_{CK}
t_{RPRE}	Read Preamble	0.9		t_{CK}
t_{RPST}	Read Postamble	0.3		t_{CK}

[1] To ensure proper operation of the DDR3, all the DDR3 requirements must be strictly followed. See "Interfacing DDR3/DDR2/LPDDR Memory to ADSP-SC5xx/215xx Processors" (EE-387).

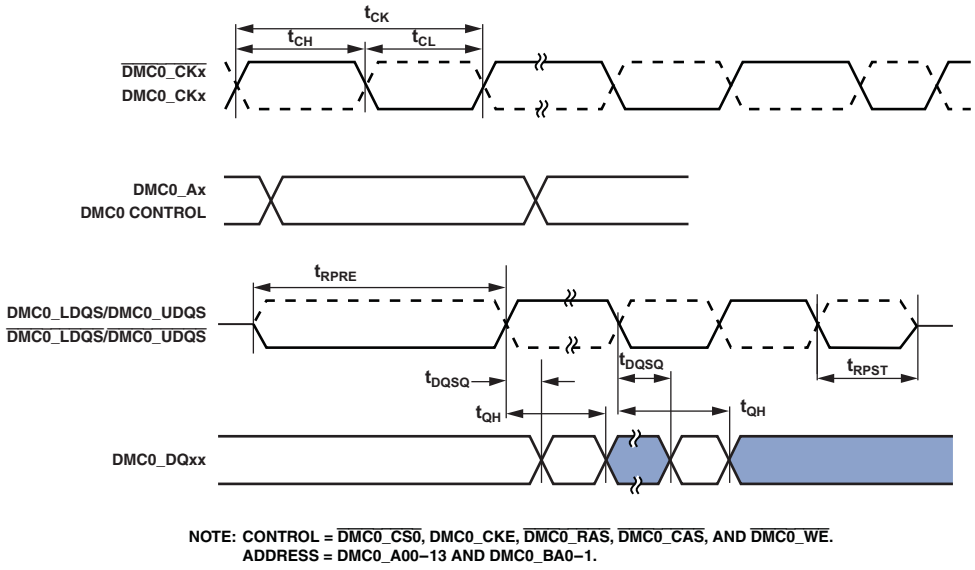


Figure 17. DDR3 SDRAM Controller Input AC Timing

Downloaded from Arrow.com.

DDR3 SDRAM Write Cycle Timing

Table 51 and Figure 18 show mobile DDR3 SDRAM output ac timing, related to the DMC.

Table 51. DDR3 SDRAM Write Cycle Timing, V_{DD_DMC} Nominal 1.5 V

Parameter		450 MHz[1] Min	450 MHz[1] Max	Unit
Switching Characteristics				
t_{DQSS}	DMC0_DQS Latching Rising Transitions to Associated Clock Edges[2]	–0.25	+0.25	t_{CK}
t_{DS}	Last Data Valid to DMC0_DQS Delay (Slew > 1 V/ns)	0.125		ns
t_{DH}	DMC0_DQS to First Data Invalid Delay (Slew > 1 V/ns)	0.150		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	0.45	0.55	t_{CK}
t_{DQSL}	DMC0_DQS Input Low Pulse Width	0.45	0.55	t_{CK}
t_{WPRE}	Write Preamble	0.9		t_{CK}
t_{WPST}	Write Postamble	0.3		t_{CK}
t_{IPW}	Address and Control Output Pulse Width	0.840		ns
t_{DIPW}	DMC0_DQ and DMC0_DM Output Pulse Width	0.550		ns

[1] To ensure proper operation of the DDR3, all the DDR3 requirements must be strictly followed. See "Interfacing DDR3/DDR2/LPDDR Memory to ADSP-SC5xx/215xx Processors" (EE-387).
[2] Write command to first DMC0_DQS delay = WL × t_{CK} + t_{DQSS}.

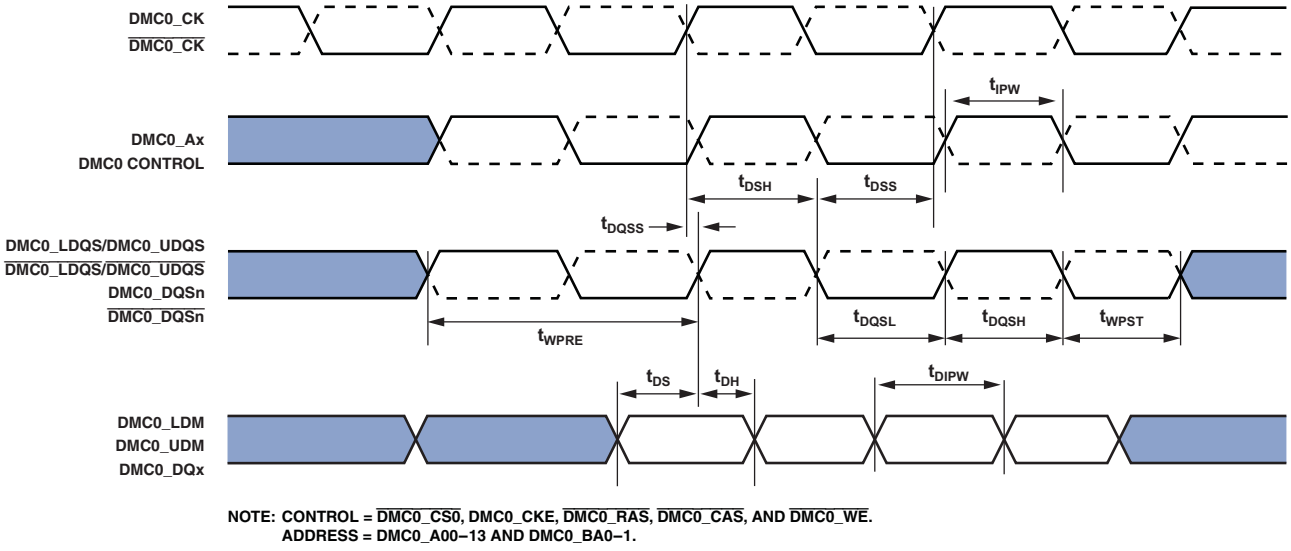


Figure 18. DDR3 SDRAM Controller Output AC Timing

Downloaded from Arrow.com.

Enhanced Parallel Peripheral Interface (EPPI) Timing

Table 52 and Table 53 and Figure 19 through Figure 27 describe enhanced parallel peripheral interface (EPPI) timing operations. In Figure 19 through Figure 27, POLC[1:0] represents the setting of the EPPI_CTL register, which sets the sampling/driving edges of the EPPI clock.

When internally generated, the programmed PPI clock ($f_{PCLKPROG}$) frequency in megahertz is set by the following equation where VALUE is a field in the EPPI_CLKDIV register that can be set from 0 to 65535:

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

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

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

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

Table 52. Enhanced Parallel Peripheral Interface (EPPI)—Internal Clock

Parameter		Min	Max	Unit
Timing Requirements				
t_{SFSPI}	External FS Setup Before EPPI_CLK	6.5		ns
t_{HFSPI}	External FS Hold After EPPI_CLK	0		ns
t_{SDRPI}	Receive Data Setup Before EPPI_CLK	6.5		ns
t_{HDRPI}	Receive Data Hold After EPPI_CLK	0		ns
t_{SFS3GI}	External FS3 Input Setup Before EPPI_CLK Fall Edge in Clock Gating Mode	14		ns
t_{HFS3GI}	External FS3 Input Hold Before EPPI_CLK Fall Edge in Clock Gating Mode	0		ns
Switching Characteristics				
t_{PCLKW}	EPPI_CLK Width[1]	$0.5 \times t_{PCLKPROG} - 1.5$		ns
t_{PCLK}	EPPI_CLK Period[1]	$t_{PCLKPROG} - 1.5$		ns
t_{DFSPI}	Internal FS Delay After EPPI_CLK		3.6	ns
t_{HOFSPI}	Internal FS Hold After EPPI_CLK	–0.72		ns
t_{DDTPI}	Transmit Data Delay After EPPI_CLK		3.5	ns
t_{HDTPI}	Transmit Data Hold After EPPI_CLK	–0.5		ns

[1] See Table 27 for details on the minimum period that can be programmed for $t_{PCLKPROG}$.

Downloaded from Arrow.com.

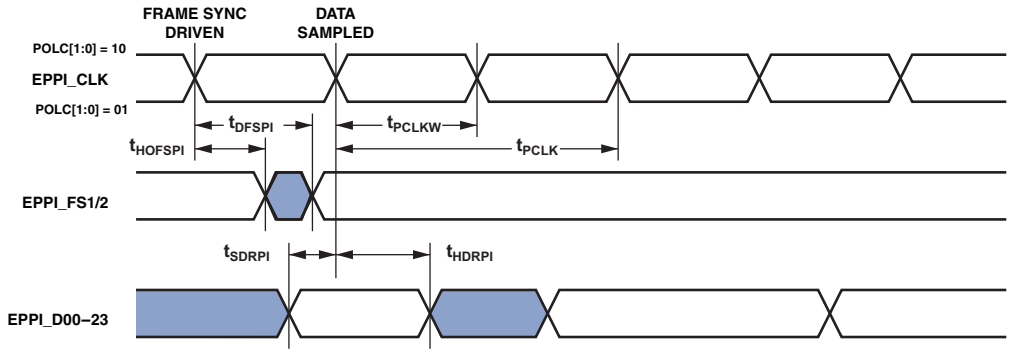


Figure 19. EPPI Internal Clock GP Receive Mode with Internal Frame Sync Timing

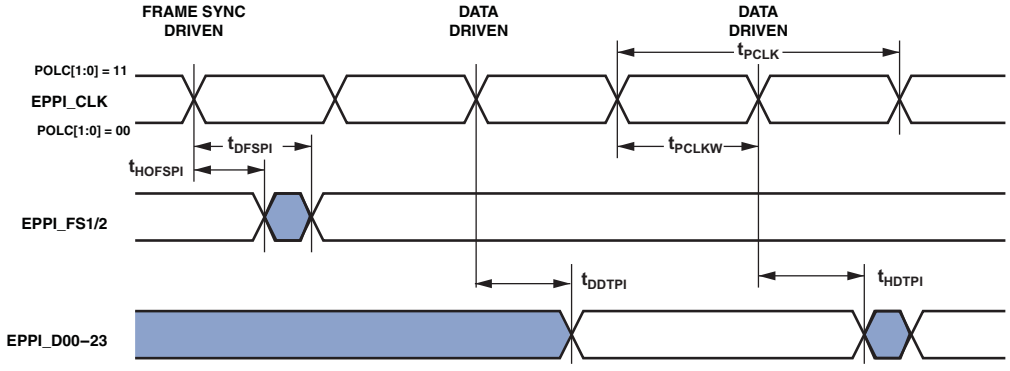


Figure 20. EPPI Internal Clock GP Transmit Mode with Internal Frame Sync Timing

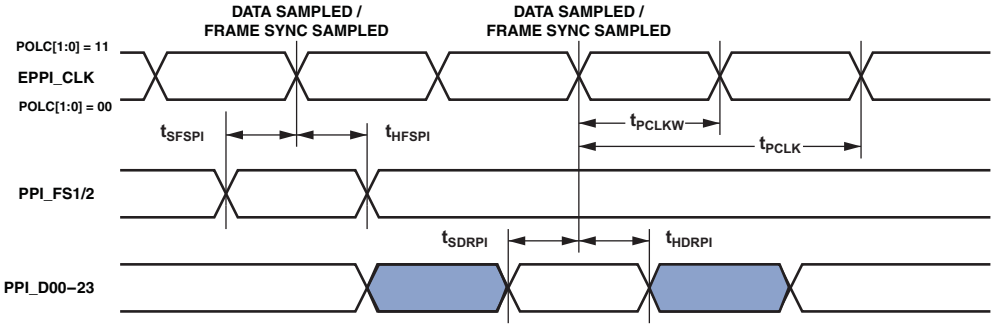


Figure 21. EPPI Internal Clock GP Receive Mode with External Frame Sync Timing

Downloaded from Arrow.com.

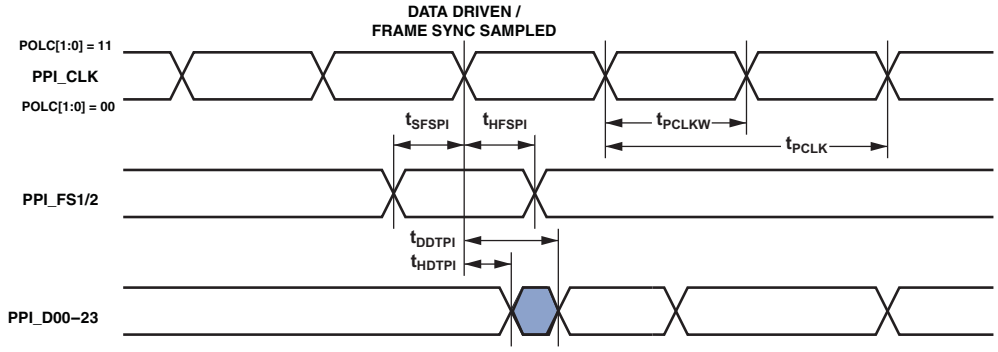


Figure 22. EPPI Internal Clock GP Transmit Mode with External Frame Sync Timing

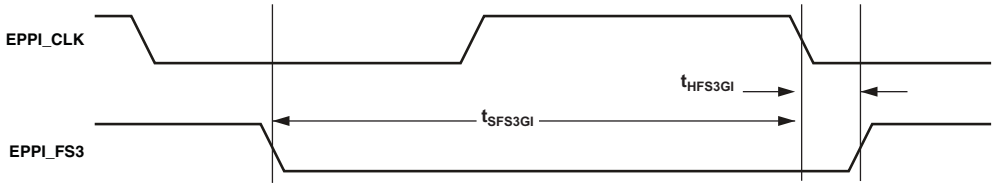


Figure 23. Clock Gating Mode with Internal Clock and External Frame Sync Timing

Downloaded from Arrow.com.

Table 53. Enhanced Parallel Peripheral Interface (EPPI)—External Clock

Parameter		Min	Max	Unit
Timing Requirements				
t_{PCLKW}	EPPI_CLK Width[1]	$0.5 \times t_{PCLKEXT} - 0.5$		ns
t_{PCLK}	EPPI_CLK Period[1]	$t_{PCLKEXT} - 1$		ns
t_{SFSPE}	External FS Setup Before EPPI_CLK	2		ns
t_{HFSPE}	External FS Hold After EPPI_CLK	3.7		ns
t_{SDRPE}	Receive Data Setup Before EPPI_CLK	2		ns
t_{HDRPE}	Receive Data Hold After EPPI_CLK	3.7		ns
Switching Characteristics				
t_{DFSPE}	Internal FS Delay After EPPI_CLK		15.3	ns
t_{HOFSPE}	Internal FS Hold After EPPI_CLK	2.4		ns
t_{DDTPE}	Transmit Data Delay After EPPI_CLK		15.3	ns
t_{HDTPE}	Transmit Data Hold After EPPI_CLK	2.4		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 27.

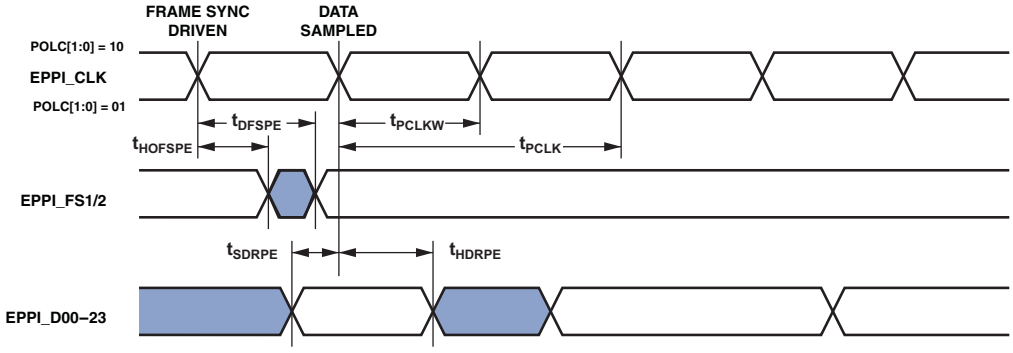


Figure 24. EPPI External Clock GP Receive Mode with Internal Frame Sync Timing

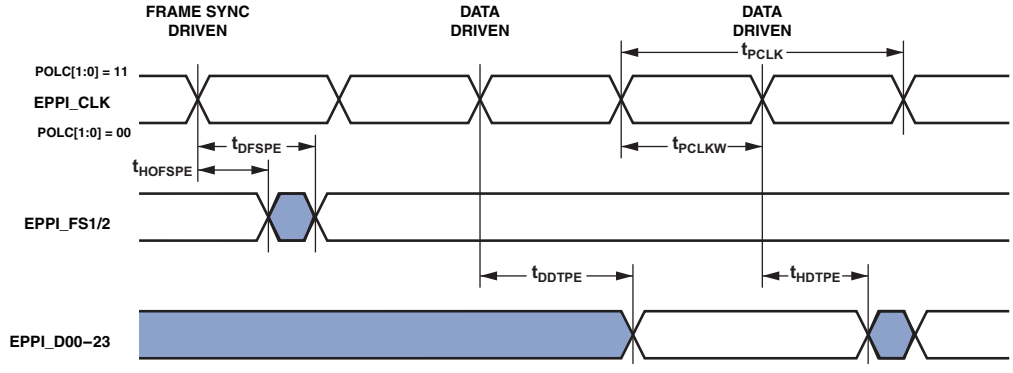


Figure 25. EPPI External Clock GP Transmit Mode with Internal Frame Sync Timing

Downloaded from Arrow.com.

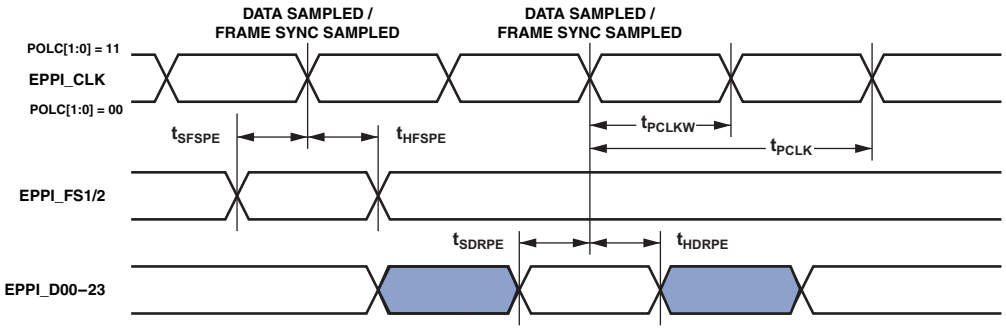


Figure 26. EPPI External Clock GP Receive Mode with External Frame Sync Timing

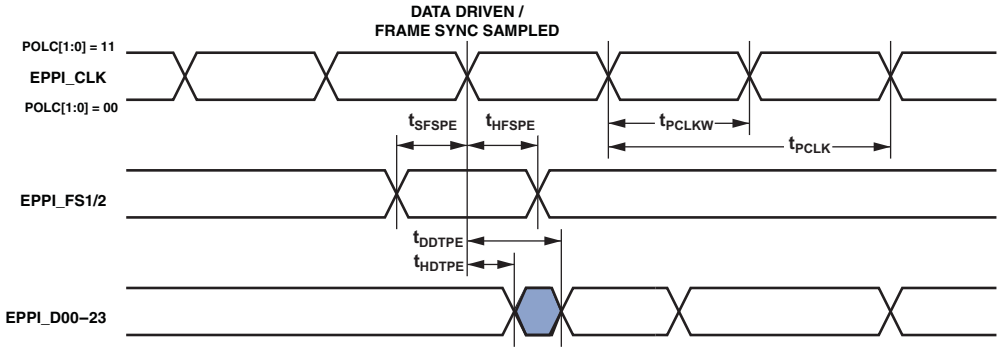


Figure 27. EPPI External Clock GP Transmit Mode with External Frame Sync Timing

Downloaded from Arrow.com.

Link Ports (LPs)

In LP receive mode, the LP clock is supplied externally and is called $f_{LCLKREXT}$, therefore the period can be represented by

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

In LP transmit mode, the programmed LP clock ($f_{LCLKTPROG}$) frequency in megahertz is set by the following equation where VALUE is a field in the LP_DIV register that can be set from 1 to 255:

$$f_{LCLKTPROG} = \frac{f_{SCLK0}}{(VALUE \times 2)}$$

In the case where VALUE = 0, $f_{LCLKTPROG} = f_{SCLK0}$. For all settings of VALUE, the following equation is true:

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

Calculation of the link receiver data setup and hold relative to the link clock is required to determine the maximum allowable skew that can be introduced in the transmission path length difference between LPx_Dx and LPx_CLK. Setup skew is the maximum delay that can be introduced in LPx_Dx relative to LPx_CLK (setup skew = $t_{LCLKTWH}$ minimum − t_{DLDCH} − t_{SLDCL}). Hold skew is the maximum delay that can be introduced in LPx_CLK relative to LPx_Dx (hold skew = $t_{LCLKTWL}$ minimum − t_{HLDCH} − t_{HLDCL}).

Table 54. LPs—Receive[1]

Parameter		Min	Max	Unit
Timing Requirements				
$f_{LCLKREXT}$	LPx_CLK Frequency		112.5	MHz
t_{SLDCL}	Data Setup Before LPx_CLK Low	0.9		ns
t_{HLDCL}	Data Hold After LPx_CLK Low	1.4		ns
t_{LCLKEW}	LPx_CLK Period[2]	$t_{LCLKREXT}$ − 0.8		ns
$t_{LCLKRWL}$	LPx_CLK Width Low[2]	0.5 × $t_{LCLKREXT}$		ns
$t_{LCLKRWH}$	LPx_CLK Width High[2]	0.5 × $t_{LCLKREXT}$		ns
Switching Characteristic				
t_{DLALC}	LPx_ACK Low Delay After LPx_CLK Low[3]	1.5 × t_{SCLK0} + 4	2.5 × t_{SCLK0} + 12	ns

[1] Specifications apply to LP0 and LP1.
[2] This specification indicates the minimum instantaneous width or period that can be tolerated due to duty cycle variation or jitter on the external LPx_CLK. For the external LPx_CLK ideal maximum frequency, see the $f_{LCLKTEXT}$ specification in Table 27.
[3] LPx_ACK goes low with t_{DLALC} relative to rise of LPx_CLK after first byte, but does not go low if the link buffer of the receiver is not about to fill.

Downloaded from Arrow.com.

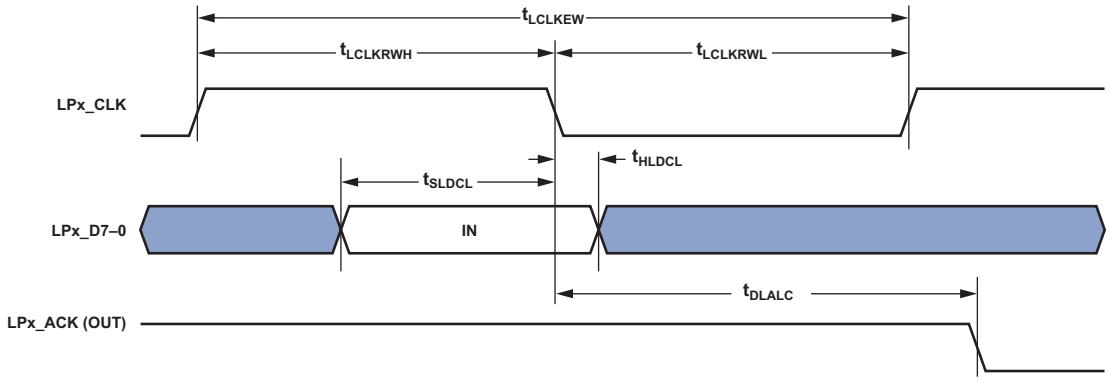


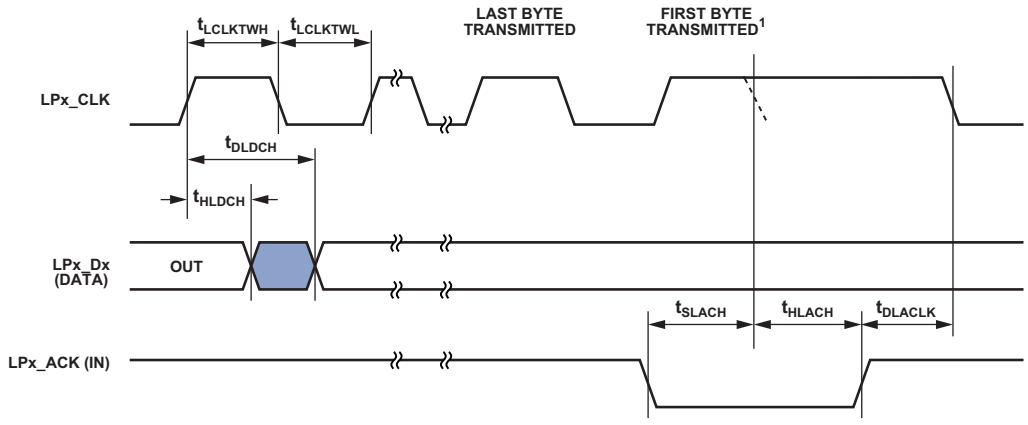
Figure 28. LPs—Receive

Downloaded from Arrow.com.

Table 55. LPs—Transmit[1]

Parameter		Min	Max	Unit
Timing Requirements				
t_{SLACH}	LPx_ACK Setup Before LPx_CLK Low	$2 \times t_{SCLK0} + 13.5$		ns
t_{HLACH}	LPx_ACK Hold After LPx_CLK Low	−5.5		ns
Switching Characteristics				
t_{DLDCH}	Data Delay After LPx_CLK High		2.23	ns
t_{HLDCH}	Data Hold After LPx_CLK High	−2.3		ns
$t_{LCLKTWL}$[2]	LPx_CLK Width Low	$0.4 \times t_{LCLKTPROG}$	$0.6 \times t_{LCLKTPROG}$	ns
$t_{LCLKTWH}$[2]	LPx_CLK Width High	$0.4 \times t_{LCLKTPROG}$	$0.6 \times t_{LCLKTPROG}$	ns
t_{LCLKTW}[2]	LPx_CLK Period	$N \times t_{LCLKTPROG} - 0.6$		ns
t_{DLACLK}	LPx_CLK Low Delay After LPx_ACK High	$t_{SCLK0} + 4$	$2 \times t_{SCLK0} + 1 \times t_{LPCLK} + 10$	ns

[1] Specifications apply to LP0 and LP1.
[2] See Table 27 for details on the minimum period that can be programmed for $t_{LCLKTPROG}$.



NOTES
The t_{SLACH} and t_{HLACH} specifications apply only to the LPx_CLK falling edge. If these specifications are met, LPx_CLK extends and the dotted LPx_CLK falling edge does not occur as shown. The position of the dotted falling edge can be calculated using the $t_{LCLKTWH}$ specification. $t_{LCLKTWH}$ Min must be used for t_{SLACH} and $t_{LCLKTWH}$ Max for t_{HLACH}.

Figure 29. LPs—Transmit

Downloaded from Arrow.com.

Serial Ports (SPORTs)

To determine whether a device is compatible with the SPORT at clock speed n, the following specifications must be confirmed: frame sync delay and frame sync setup and hold; data delay and data setup and hold; and serial clock (SPTx_CLK) width. In Figure 30, either the rising edge or the falling edge of SPTx_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 megahertz is set by the following equation where CLKDIV is a field in the SPORT_DIV register that can be set from 0 to 65535:

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

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

Table 56. SPORTs—External Clock[1]

Parameter		Min	Max	Unit
Timing Requirements				
t_{SFSE}	Frame Sync Setup Before SPTx_CLK (Externally Generated Frame Sync in either Transmit or Receive Mode)[2]	2		ns
t_{HFSE}	Frame Sync Hold After SPTx_CLK (Externally Generated Frame Sync in either Transmit or Receive Mode)[2]	2.7		ns
t_{SDRE}	Receive Data Setup Before Receive SPTx_CLK[2]	2		ns
t_{HDRE}	Receive Data Hold After SPTx_CLK[2]	2.7		ns
$t_{SPTCLKW}$	SPTx_CLK Width[3]	$0.5 \times t_{SPTCLKEXT} - 1.5$		ns
t_{SPTCLK}	SPTx_CLK Period[3]	$t_{SPTCLKEXT} - 1.5$		ns
Switching Characteristics				
t_{DFSE}	Frame Sync Delay After SPTx_CLK (Internally Generated Frame Sync in either Transmit or Receive Mode)[4]		14.5	ns
t_{HOFSE}	Frame Sync Hold After SPTx_CLK (Internally Generated Frame Sync in either Transmit or Receive Mode)[4]	2		ns
t_{DDTE}	Transmit Data Delay After Transmit SPTx_CLK[4]		14	ns
t_{HDTE}	Transmit Data Hold After Transmit SPTx_CLK[4]	2		ns

[1] Specifications apply to all four SPORTs.
[2] Referenced to sample edge.
[3] This specification indicates the minimum instantaneous width or period that can be tolerated due to duty cycle variation or jitter on the external SPTx_CLK. For the external SPTx_CLK ideal maximum frequency see the $f_{SPTCLKEXT}$ specification in Table 27.
[4] Referenced to drive edge.

Downloaded from Arrow.com.

Table 57. SPORTs—Internal Clock[1]

Parameter		Min	Max	Unit
Timing Requirements				
t_{SFSI}	Frame Sync Setup Before SPTx_CLK (Externally Generated Frame Sync in either Transmit or Receive Mode)[2]	12		ns
t_{HFSI}	Frame Sync Hold After SPTx_CLK (Externally Generated Frame Sync in either Transmit or Receive Mode)[2]	–0.5		ns
t_{SDRI}	Receive Data Setup Before SPTx_CLK[2]	3.4		ns
t_{HDRI}	Receive Data Hold After SPTx_CLK[2]	1.5		ns
Switching Characteristics				
t_{DFSI}	Frame Sync Delay After SPTx_CLK (Internally Generated Frame Sync in Transmit or Receive Mode)[3]		3.5	ns
t_{HOFSI}	Frame Sync Hold After SPTx_CLK (Internally Generated Frame Sync in Transmit or Receive Mode)[3]	–2.5		ns
t_{DDTI}	Transmit Data Delay After SPTx_CLK[3]		3.5	ns
t_{HDTI}	Transmit Data Hold After SPTx_CLK[3]	–2.5		ns
$t_{SPTCLKIW}$	SPTx_CLK Width[4]	$0.5 \times t_{SPTCLKPROG} - 2$		ns
t_{SPTCLK}	SPTx_CLK Period[4]	$t_{SPTCLKPROG} - 1.5$		ns

[1] Specifications apply to all four SPORTs.
[2] Referenced to the sample edge.
[3] Referenced to drive edge.
[4] See Table 27 for details on the minimum period that can be programmed for $t_{SPTCLKPROG}$.

Downloaded from Arrow.com.

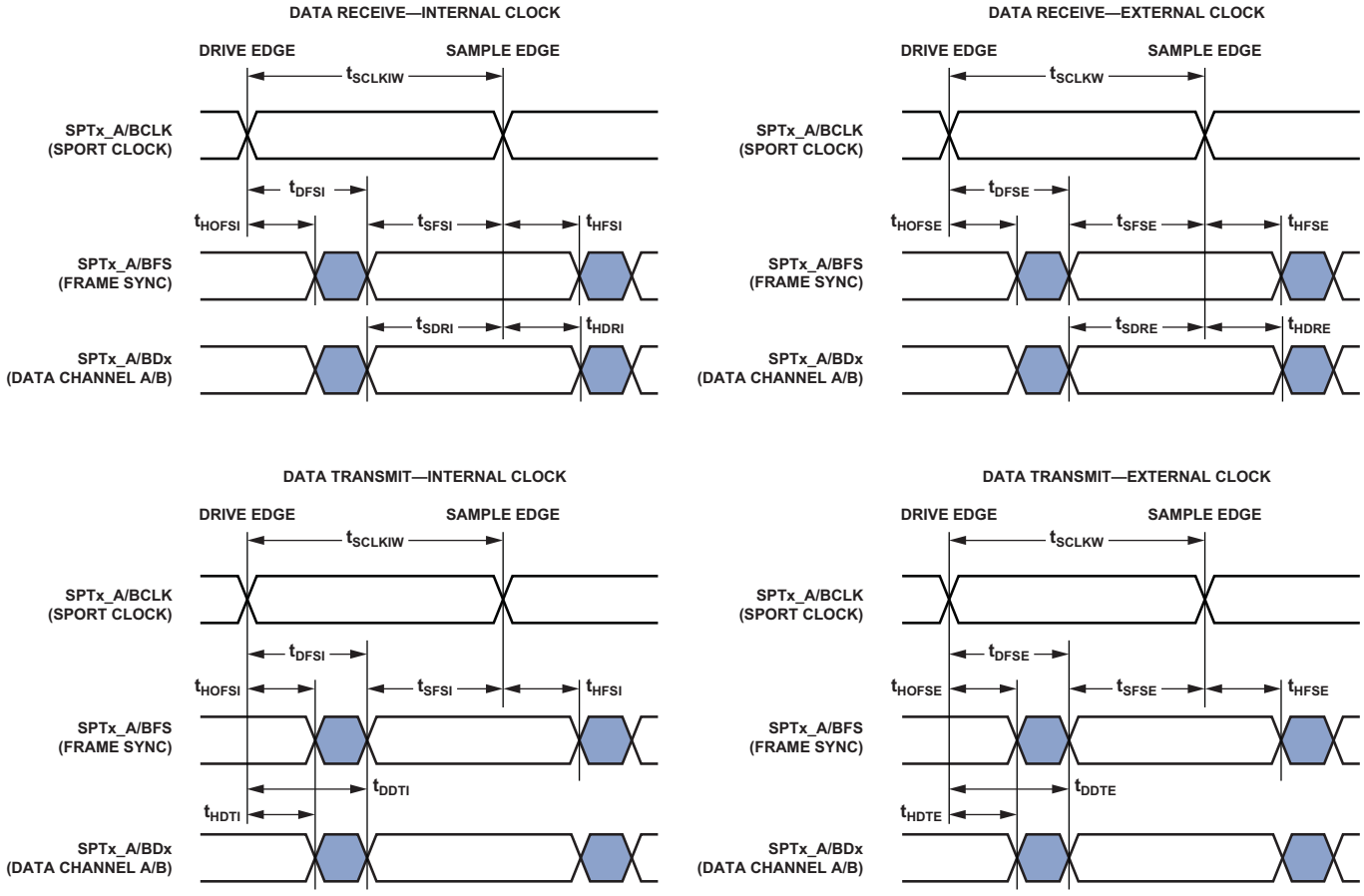


Figure 30. SPORTs

Downloaded from Arrow.com.

Table 58. SPORTs—Enable and Three-State[1]

Parameter		Min	Max	Unit
Switching Characteristics				
t_{DDTEN}	Data Enable from External Transmit SPTx_CLK[2]	1		ns
t_{DDTTE}	Data Disable from External Transmit SPTx_CLK[2]		14	ns
t_{DDTIN}	Data Enable from Internal Transmit SPTx_CLK[2]	–2.5		ns
t_{DDTTI}	Data Disable from Internal Transmit SPTx_CLK[2]		2.8	ns

[1] Specifications apply to all four SPORTs.
[2] Referenced to drive edge.

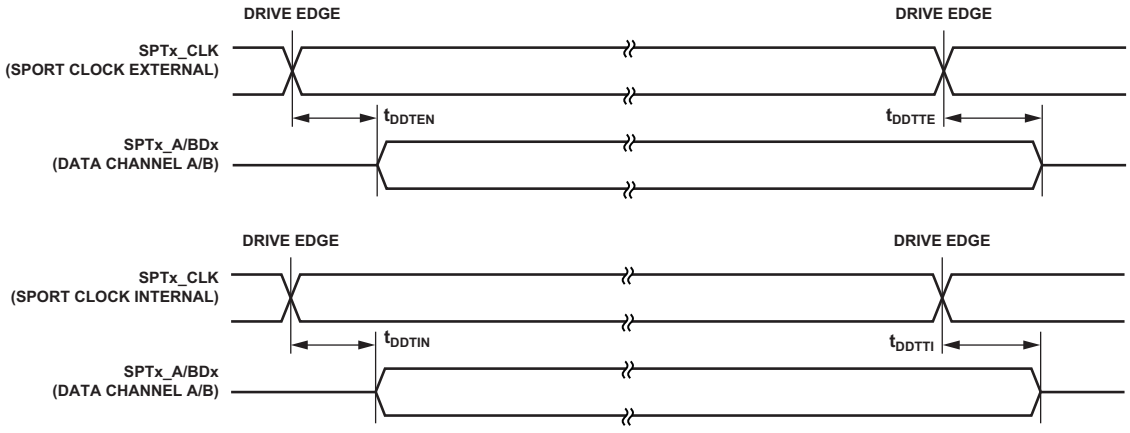


Figure 31. SPORTs—Enable and Three-State

Downloaded from Arrow.com.

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

Table 59. SPORTs—Transmit Data Valid (TDV)[1]

Parameter		Min	Max	Unit
Switching Characteristics				
t_{DRDVEN}	Data Valid Enable Delay from Drive Edge of External Clock[2]	2		ns
t_{DFDVEN}	Data Valid Disable Delay from Drive Edge of External Clock[2]		14	ns
t_{DRDVIN}	Data Valid Enable Delay from Drive Edge of Internal Clock[2]	–2.5		ns
t_{DFDVIN}	Data Valid Disable Delay from Drive Edge of Internal Clock[2]		3.5	ns

[1] Specifications apply to all four SPORTs.
[2] Referenced to drive edge.

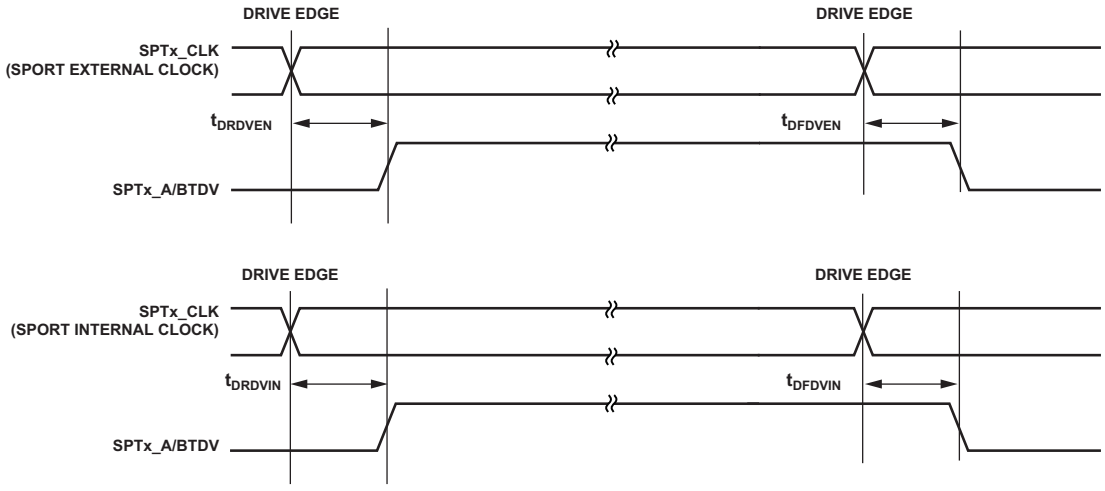


Figure 32. SPORTs—Transmit Data Valid Internal and External Clock

Downloaded from Arrow.com.

Table 60. SPORTs—External Late Frame Sync[1]

Parameter		Min	Max	Unit
Switching Characteristics				
$t_{DDTLFSE}$	Data Delay from Late External Transmit Frame Sync or External Receive Frame Sync with MCE = 1, MFD = 0[2]		14	ns
$t_{DDTENFS}$	Data Enable for MCE = 1, MFD = 0[2]	0.5		ns

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

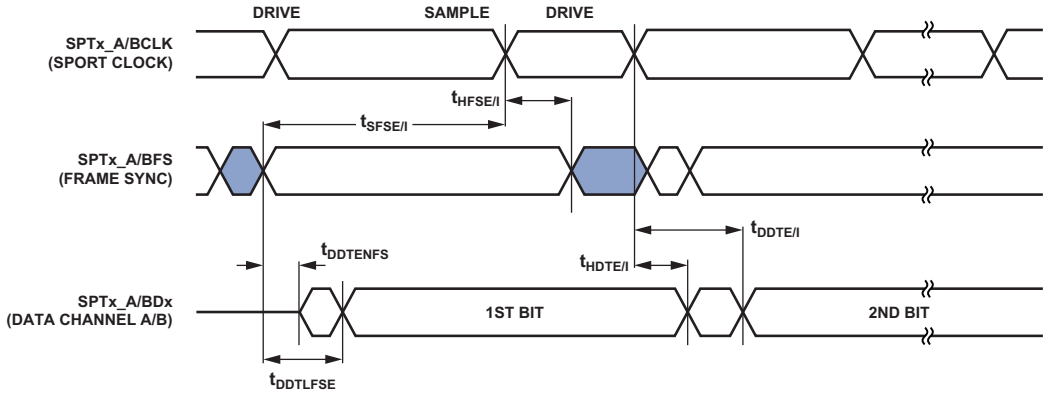


Figure 33. External Late Frame Sync

Downloaded from Arrow.com.

Asynchronous Sample Rate Converter (ASRC)—Serial Input Port

The ASRC input signals are routed from the DAI0_PINx pins using the SRU. Therefore, the timing specifications provided in Table 61 are valid at the DAI0_PINx pins.

Table 61. ASRC, Serial Input Port

Parameter		Min	Max	Unit
Timing Requirements				
t_{SRCSFS}[1]	Frame Sync Setup Before Serial Clock Rising Edge	4		ns
t_{SRCHFS}[1]	Frame Sync Hold After Serial Clock Rising Edge	5.5		ns
t_{SRCSD}[1]	Data Setup Before Serial Clock Rising Edge	4		ns
t_{SRCHD}[1]	Data Hold After Serial Clock Rising Edge	5.5		ns
$t_{SRCCLKW}$	Clock Width	$t_{SCLK0} - 1$		ns
t_{SRCCLK}	Clock Period	$2 \times t_{SCLK0}$		ns

[1] The serial clock, data, and frame sync signals can come from any of the DAI pins. The serial clock and frame sync signals can also come via PCG or SPORTs. The input of the PCG can be either CLKIN or any of the DAI pins.

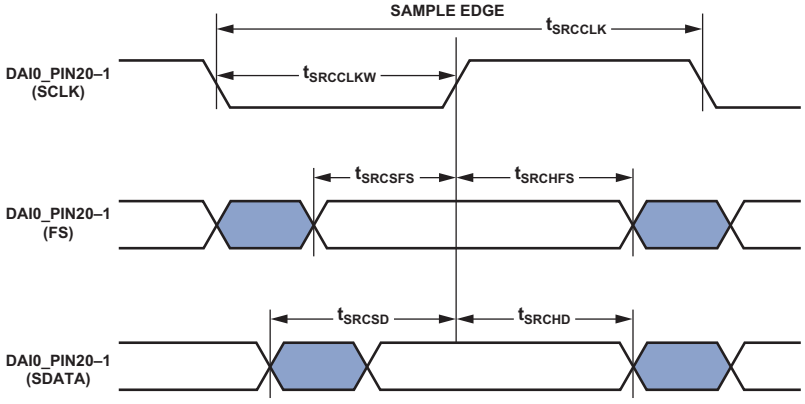


Figure 34. ASRC Serial Input Port Timing

Downloaded from Arrow.com.

Asynchronous Sample Rate Converter (ASRC)—Serial Output Port

For the serial output port, the frame sync is an input and it must meet setup and hold times with regard to SCLK on the output port. The serial data output has a hold time and delay specification with regard to serial clock. The serial clock rising edge is the sampling edge, and the falling edge is the drive edge.

Table 62. ASRC, Serial Output Port

Parameter		Min	Max	Unit
Timing Requirements				
t_{SRCSFS}[1]	Frame Sync Setup Before Serial Clock Rising Edge	4		ns
t_{SRCHFS}[1]	Frame Sync Hold After Serial Clock Rising Edge	5.5		ns
$t_{SRCCLKW}$	Clock Width	$t_{SCLK0} - 1$		ns
t_{SRCCLK}	Clock Period	$2 \times t_{SCLK0}$		ns
Switching Characteristics				
t_{SRCTDD}[1]	Transmit Data Delay After Serial Clock Falling Edge		13	ns
t_{SRCTDH}[1]	Transmit Data Hold After Serial Clock Falling Edge	1		ns

[1] The serial clock, data, and frame sync signals can come from any of the DAI pins. The serial clock and frame sync signals can also come via PCG or SPORTs. The input of the PCG can be either CLKIN, SCLK0, or any of the DAI pins.

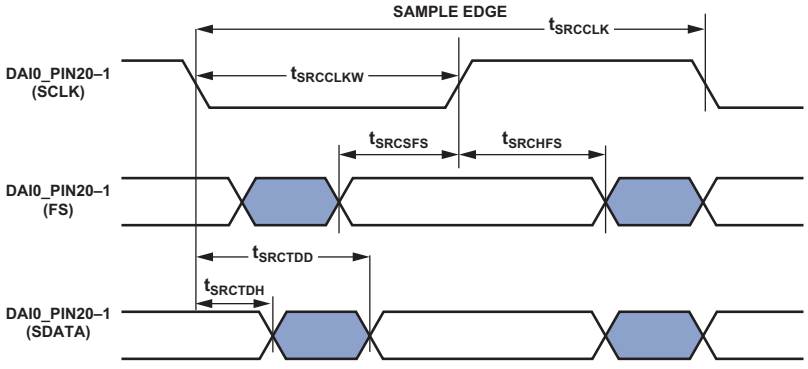


Figure 35. ASRC Serial Output Port Timing

Downloaded from Arrow.com.

SPI Port—Master Timing

SPI0, SPI1, and SPI2

Table 63, Table 64, and Figure 36 describe the SPI port master operations.

When internally generated, the programmed SPI clock ($f_{SPICLKPROG}$) frequency in megahertz is set by the following equation where BAUD is a field in the SPIx_CLK register that can be set from 0 to 65535.

For SPI0, SPI1,

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

For SPI2,

$$f_{SPICLKPROG} = \frac{f_{SCLK1}}{(BAUD + 1)}$$

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

Note that

- In dual-mode data transmit, the SPIx_MISO signal is also an output.
- In quad-mode data transmit, the SPIx_MISO, SPIx_D2, and SPIx_D3 signals are also outputs.
- In dual-mode data receive, the SPIx_MOSI signal is also an input.
- In quad-mode data receive, the SPIx_MOSI, SPIx_D2, and SPIx_D3 signals are also inputs.
- Quad-mode is supported by SPI2 only.
- CPHA is a configuration bit in the SPI_CTL register.

Table 63. SPI0, SPI1 Port—Master Timing[1]

Parameter		Min	Max	Unit
Timing Requirements				
t_{SSPIDM}	Data Input Valid to SPIx_CLK Edge (Data Input Setup)	3		ns
t_{HSPIDM}	SPIx_CLK Sampling Edge to Data Input Invalid	1.2		ns
Switching Characteristics				
t_{SDSCIM}	$\overline{\text{SPIx_SEL}}$ low to First SPI_CLK Edge for CPHA = 1[2]	$t_{SPICLKPROG} - 5$		ns
	$\overline{\text{SPIx_SEL}}$ low to First SPI_CLK Edge for CPHA = 0[2]	$1.5 \times t_{SPICLKPROG} - 5$		ns
t_{SPICHM}	SPIx_CLK High Period[3]	$0.5 \times t_{SPICLKPROG} - 1.5$		ns
t_{SPICLM}	SPIx_CLK Low Period[3]	$0.5 \times t_{SPICLKPROG} - 1.8$		ns
t_{SPICLK}	SPIx_CLK Period[3]	$t_{SPICLKPROG} - 1.5$		ns
t_{HDSM}	Last SPIx_CLK Edge to $\overline{\text{SPIx_SEL}}$ High for CPHA = 1[2]	$1.5 \times t_{SPICLKPROG} - 5$		ns
	Last SPIx_CLK Edge to $\overline{\text{SPIx_SEL}}$ High for CPHA = 0[2]	$t_{SPICLKPROG} - 5$		ns
t_{SPITDM}	Sequential Transfer Delay[2, 4]	$t_{SPICLKPROG} - 1.5$		ns
$t_{DDSPIDM}$	SPIx_CLK Edge to Data Out Valid (Data Out Delay)		2.7	ns
$t_{HDSPIDM}$	SPIx_CLK Edge to Data Out Invalid (Data Out Hold)	–3.75		ns

[1] All specifications apply to SPI0 and SPI1 only.
[2] Specification assumes the LEADX and LAGX bits in the SPI_DLY register are 1.
[3] See Table 27 for details on the minimum period that can be programmed for $t_{SPICLKPROG}$.
[4] Applies to sequential mode with STOP ≥ 1.

Downloaded from Arrow.com.

Table 64. SPI2 Port—Master Timing[1]

Parameter		Min	Max	Unit
Timing Requirements				
t_{SSPIDM}	Data Input Valid to SPIx_CLK Edge (Data Input Setup)	2.7		ns
t_{HSPIDM}	SPIx_CLK Sampling Edge to Data Input Invalid	0.75		ns
Switching Characteristics				
t_{SDSCIM}	$\overline{\text{SPIx_SEL}}$ low to First SPI_CLK Edge for CPHA = 1[2]	$t_{SPICLKPROG} - 5$		ns
	$\overline{\text{SPIx_SEL}}$ low to First SPI_CLK Edge for CPHA = 0[2]	$1.5 \times t_{SPICLKPROG} - 5$		ns
t_{SPICHM}	SPIx_CLK High Period[3]	$0.5 \times t_{SPICLKPROG} - 1.5$		ns
t_{SPICLM}	SPIx_CLK Low Period[3]	$0.5 \times t_{SPICLKPROG} - 1.5$		ns
t_{SPICLK}	SPIx_CLK Period[3]	$t_{SPICLKPROG} - 1.5$		ns
t_{HDSM}	Last SPIx_CLK Edge to $\overline{\text{SPIx_SEL}}$ High for CPHA = 1[2]	$1.5 \times t_{SPICLKPROG} - 5$		ns
	Last SPIx_CLK Edge to $\overline{\text{SPIx_SEL}}$ High for CPHA = 0[2]	$t_{SPICLKPROG} - 5$		ns
t_{SPITDM}	Sequential Transfer Delay[2, 4]	$t_{SPICLKPROG} - 1.5$		ns
$t_{DDSPIDM}$	SPIx_CLK Edge to Data Out Valid (Data Out Delay)		3.17	ns
$t_{HDSPIDM}$	SPIx_CLK Edge to Data Out Invalid (Data Out Hold)	–2.4		ns

[1] All specifications apply to SPI2 only.
[2] Specification assumes the LEADX and LAGX bits in the SPI_DLY register are 1.
[3] See Table 27 for details on the minimum period that may be programmed for $t_{SPICLKPROG}$.
[4] Applies to sequential mode with STOP ≥ 1.

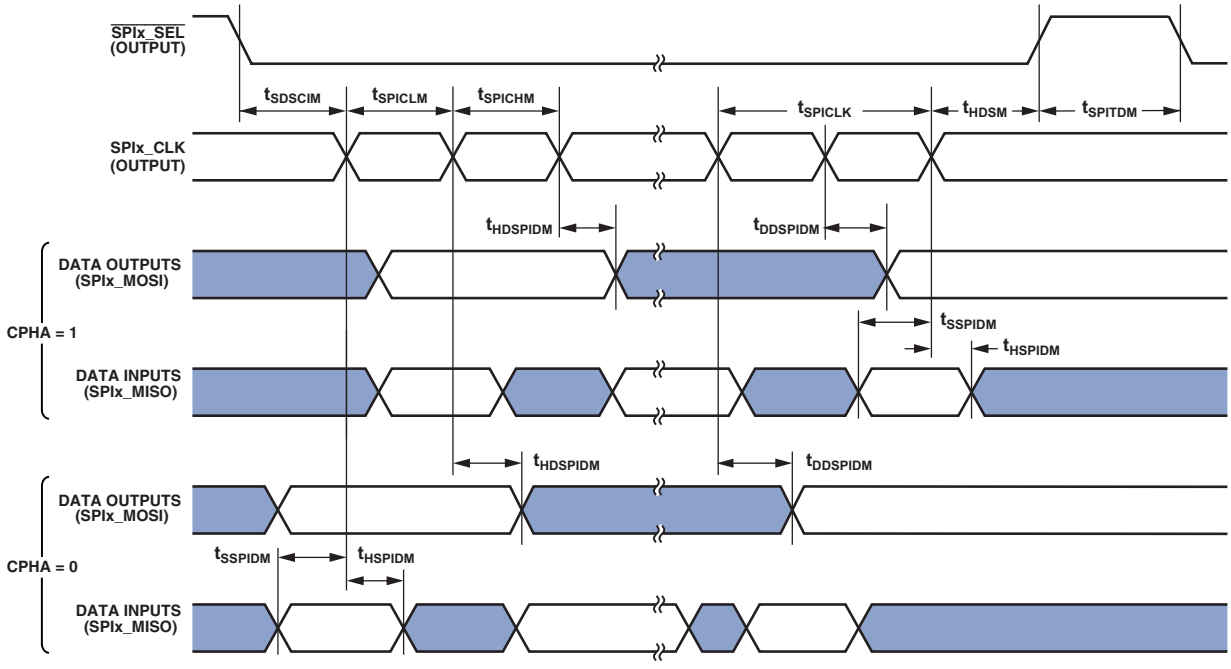


Figure 36. SPI Port—Master Timing

Downloaded from Arrow.com.

SPI Port—Slave Timing

SPI0, SPI1, and SPI2

Table 65, Table 66, and Figure 37 describe SPI port slave operations. Note that

- In dual-mode data transmit, the SPIx_MOSI signal is also an output.
- In quad-mode data transmit, the SPIx_MOSI, SPIx_D2, and SPIx_D3 signals are also outputs.
- In dual-mode data receive, the SPIx_MISO signal is also an input.
- In quad-mode data receive, the SPIx_MISO, SPIx_D2, and SPIx_D3 signals are also inputs.
- In SPI slave mode, the SPI clock is supplied externally and is called $f_{SPICLKEXT}$:

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

- Quad mode is supported by SPI2 only.
- CPHA is a configuration bit in the SPI_CTL register.

Table 65. SPI0, SPI1 Port—Slave Timing[1]

Parameter		Min	Max	Unit
Timing Requirements				
t_{SPICHS}	SPIx_CLK High Period[2]	$0.5 \times t_{SPICLKEXT} - 1.5$		ns
t_{SPICLS}	SPIx_CLK Low Period[2]	$0.5 \times t_{SPICLKEXT} - 1.5$		ns
t_{SPICLK}	SPIx_CLK Period[2]	$t_{SPICLKEXT} - 1.5$		ns
t_{HDS}	Last SPIx_CLK Edge to $\overline{\text{SPIx_SS}}$ Not Asserted	5		ns
t_{SPITDS}	Sequential Transfer Delay	$t_{SPICLKEXT} - 1.5$		ns
t_{SDSCI}	$\overline{\text{SPIx_SS}}$ Assertion to First SPIx_CLK Edge	11.7		ns
t_{SSPID}	Data Input Valid to SPIx_CLK Edge (Data Input Setup)	2		ns
t_{HSPID}	SPIx_CLK Sampling Edge to Data Input Invalid	1.6		ns
Switching Characteristics				
t_{DSOE}	$\overline{\text{SPIx_SS}}$ Assertion to Data Out Active	0	14.12	ns
t_{DSDHI}	$\overline{\text{SPIx_SS}}$ Deassertion to Data High Impedance	0	12.6	ns
t_{DDSPID}	SPIx_CLK Edge to Data Out Valid (Data Out Delay)		14.16	ns
t_{HDSPID}	SPIx_CLK Edge to Data Out Invalid (Data Out Hold)	1.5		ns

[1] All specifications apply to SPI0 and SPI1.
[2] This specification indicates the minimum instantaneous width or period that can be tolerated due to duty cycle variation or jitter on the external SPIx_CLK. For the external SPIx_CLK ideal maximum frequency, see the $f_{SPICLKTEXT}$ specification in Table 27.

Downloaded from Arrow.com.

Table 66. SPI2 Port—Slave Timing[1]

Parameter		Min	Max	Unit
Timing Requirements				
t_{SPICHS}	SPIx_CLK High Period[2]	$0.5 \times t_{SPICLKEXT} - 1.5$		ns
t_{SPICLS}	SPIx_CLK Low Period[2]	$0.5 \times t_{SPICLKEXT} - 1.5$		ns
t_{SPICLK}	SPIx_CLK Period[2]	$t_{SPICLKEXT} - 1.5$		ns
t_{HDS}	Last SPIx_CLK Edge to $\overline{SPIx_SS}$ Not Asserted	5		ns
t_{SPITDS}	Sequential Transfer Delay	$t_{SPICLKEXT} - 1.5$		ns
t_{SDSCI}	$\overline{SPIx_SS}$ Assertion to First SPIx_CLK Edge	10.5		ns
t_{SSPID}	Data Input Valid to SPIx_CLK Edge (Data Input Setup)	2		ns
t_{HSPID}	SPIx_CLK Sampling Edge to Data Input Invalid	1.6		ns
Switching Characteristics				
t_{DSOE}	$\overline{SPIx_SS}$ Assertion to Data Out Active	0	14	ns
t_{DSDHI}	$\overline{SPIx_SS}$ Deassertion to Data High Impedance	0	11.5	ns
t_{DDSPID}	SPIx_CLK Edge to Data Out Valid (Data Out Delay)		14	ns
t_{HDSPID}	SPIx_CLK Edge to Data Out Invalid (Data Out Hold)	1.5		ns

[1] All specifications apply to SPI2 only.
[2] This specification indicates the minimum instantaneous width or period that can be tolerated due to duty cycle variation or jitter on the external SPIx_CLK. For the external SPIx_CLK ideal maximum frequency, see the $f_{SPICLKTEXT}$ specification in Table 27.

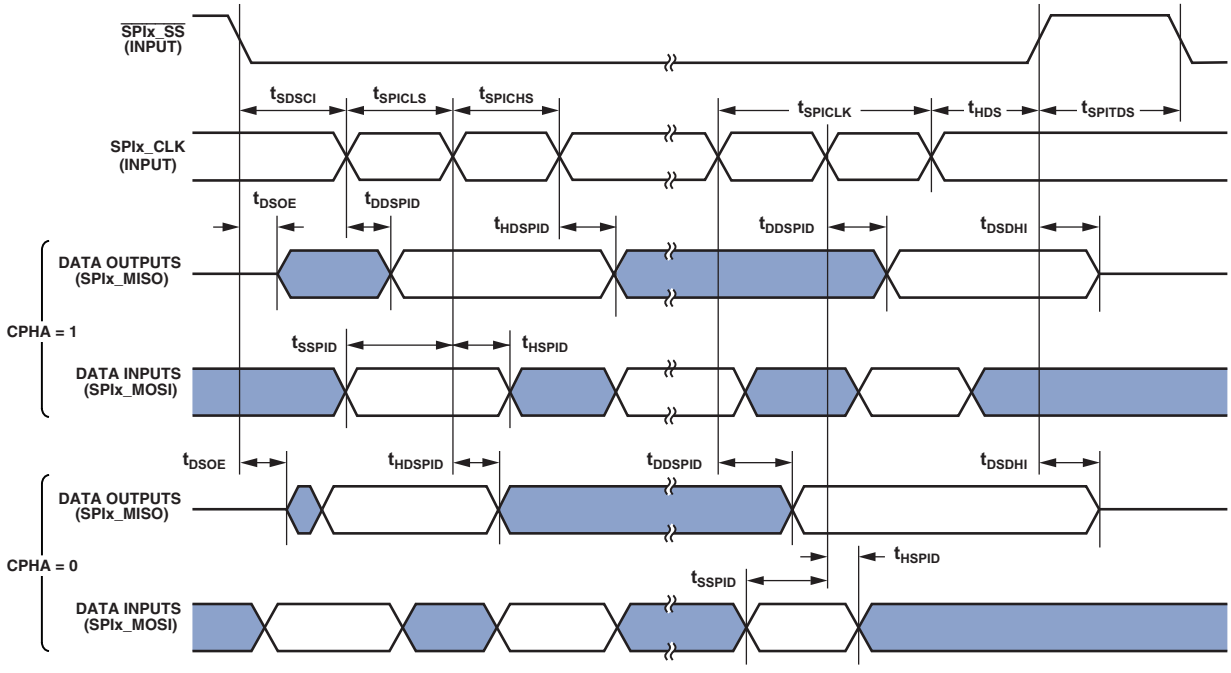


Figure 37. SPI Port—Slave Timing

Downloaded from Arrow.com.

SPI Port—SPIx_RDY Slave Timing

SPIx_RDY provides flow control. CPOL, CPHA, and FCCH are configuration bits in the SPIx_CTL register.

Table 67. SPI Port—SPIx_RDY Slave Timing[1]

Parameter		Conditions	Min	Max	Unit
Switching Characteristic					
$t_{DSPISCKRDYS}$	SPIx_RDY Deassertion from Last Valid Input SPIx_CLK Edge	FCCH = 0	$3 \times t_{SCLK1}$	$4 \times t_{SCLK1} + 10$	ns
		FCCH = 1	$4 \times t_{SCLK1}$	$5 \times t_{SCLK1} + 10$	ns

[1] All specifications apply to all three SPIs.

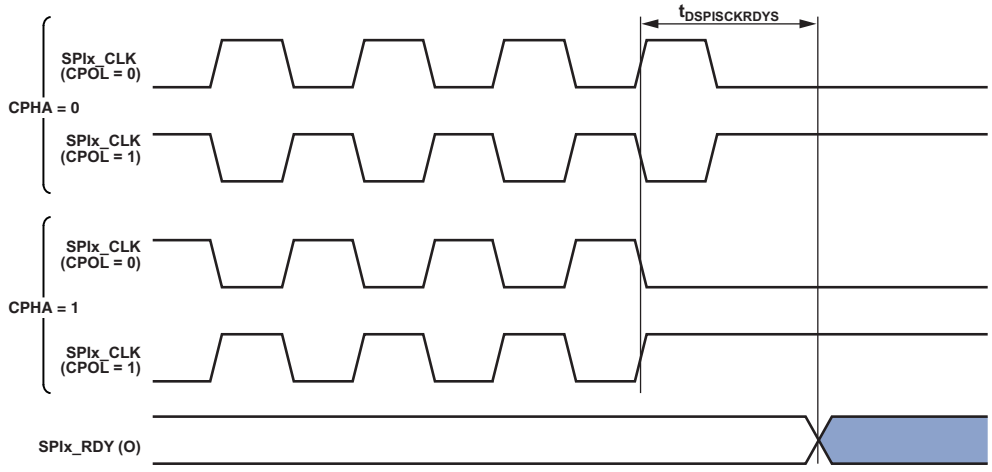


Figure 38. SPIx_RDY Deassertion from Valid Input SPIx_CLK Edge in Slave Mode

Downloaded from Arrow.com.

SPI Port—Open Drain Mode (ODM) Timing

In Figure 39 and Figure 40, the outputs can be SPIx_MOSI, SPIx_MISO, SPIx_D2, and/or SPIx_D3, depending on the mode of operation. CPOL and CPHA are configuration bits in the SPI_CTL register.

Table 68. SPI Port—ODM Master Mode Timing[1]

Parameter		Min	Max	Unit
Switching Characteristics				
$t_{HDSPIODMM}$	SPIx_CLK Edge to High Impedance from Data Out Valid	–1.1		ns
$t_{DDSPIODMM}$	SPIx_CLK Edge to Data Out Valid from High Impedance	–1	6	ns

[1] All specifications apply to all three SPIs.

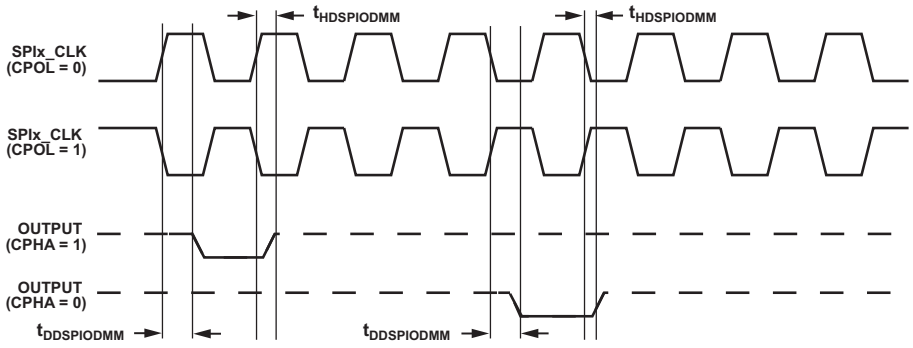


Figure 39. ODM Master Mode

Table 69. SPI Port—ODM Slave Mode[1]

Parameter		Min	Max	Unit
Timing Requirements				
$t_{HDSPIODMS}$	SPIx_CLK Edge to High Impedance from Data Out Valid	0		ns
$t_{DDSPIODMS}$	SPIx_CLK Edge to Data Out Valid from High Impedance		11	ns

[1] All specifications apply to all three SPIs.

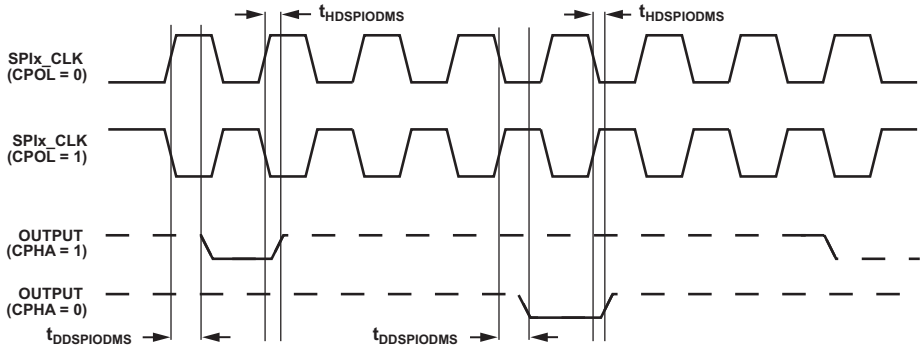


Figure 40. ODM Slave Mode

Downloaded from Arrow.com.

SPI Port—SPIx_RDY Master Timing

SPIx_RDY is used to provide flow control. CPOL and CPHA are configuration bits in the SPIx_CTL register, while LEADX, LAGX, and STOP are configuration bits in the SPIx_DLY register.

Table 70. SPI Port—SPIx_RDY Master Timing[1]

Parameter		Conditions	Min	Max	Unit
Timing Requirement					
$t_{SRDYSCKM}$	Setup Time for SPIx_RDY Deassertion Before Last Valid Data SPIx_CLK Edge		$(2 + 2 \times BAUD^2) \times t_{SCLK1} + 10$		ns
Switching Characteristic					
$t_{DRDYSCKM}$[3]	Assertion of SPIx_RDY to First SPIx_CLK Edge of Next Transfer	BAUD = 0, CPHA = 0	$4.5 \times t_{SCLK1}$	$5.5 \times t_{SCLK1} + 10$	ns
		BAUD = 0, CPHA = 1	$4 \times t_{SCLK1}$	$5 \times t_{SCLK1} + 10$	ns
		BAUD > 0, CPHA = 0	$(1 + 1.5 \times BAUD^2) \times t_{SCLK1}$	$(2 + 2.5 \times BAUD^2) \times t_{SCLK1} + 10$	ns
		BAUD > 0, CPHA = 1	$(1 + 1 \times BAUD^2) \times t_{SCLK1}$	$(2 + 2 \times BAUD^2) \times t_{SCLK1} + 10$	ns

[1] All specifications apply to all three SPIs.
[2] BAUD value is set using the SPIx_CLK.BAUD bits. BAUD value = SPIx_CLK.BAUD bits + 1.
[3] Specification assumes the LEADX, LAGX, and STOP bits in the SPI_DLY register are zero.

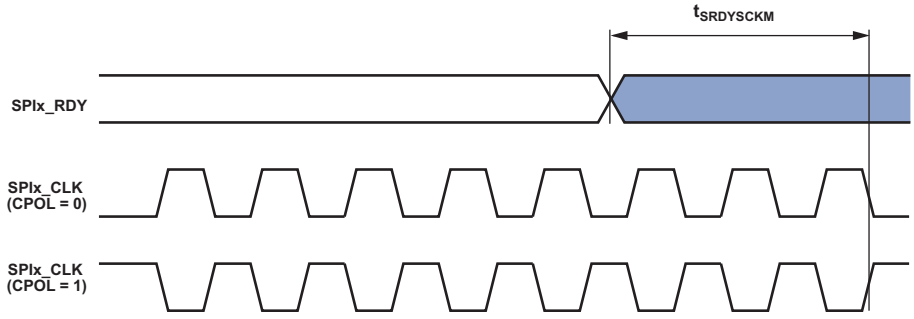


Figure 41. SPIx_RDY Setup Before SPIx_CLK

Downloaded from Arrow.com.

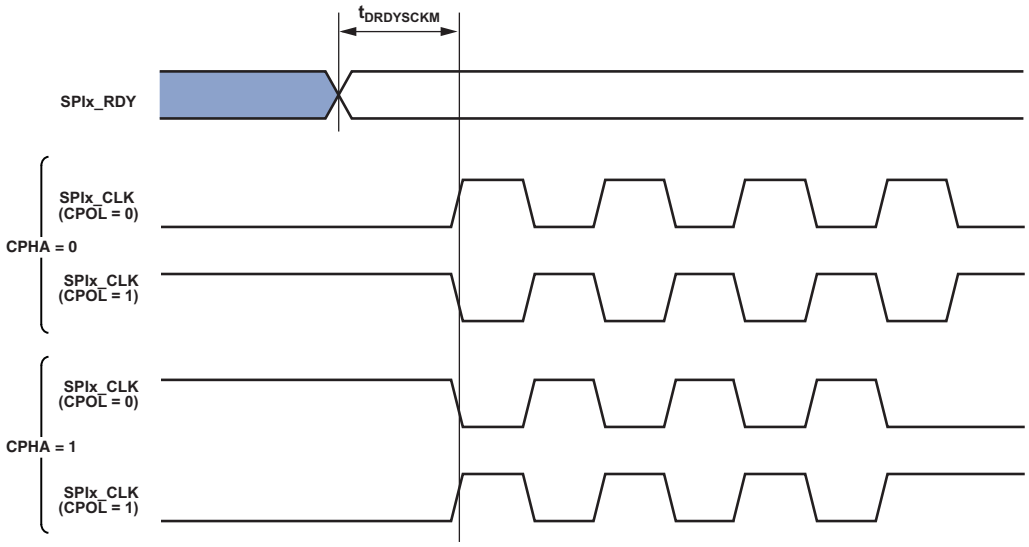


Figure 42. SPIx_CLK Switching Diagram after SPIx_RDY Assertion

Downloaded from Arrow.com.

Precision Clock Generator (PCG) (Direct Pin Routing)

This timing is only valid when the SRU is configured such that the precision clock generator (PCG) takes inputs directly from the DAI pins (via pin buffers) and sends outputs directly to the DAI pins. For the other cases, where the PCG inputs and outputs are not directly routed to/from DAI pins (via pin buffers), there is no timing data available. All timing parameters and switching characteristics apply to external DAI pins (DAI0_PINx).

Table 71. PCG (Direct Pin Routing)

Parameter		Min	Max	Unit
Timing Requirements				
t_{PCGIP}	Input Clock Period	$t_{SCLK} \times 2$		ns
t_{STRIG}	PCG Trigger Setup Before Falling Edge of PCG Input Clock	4.5		ns
t_{HTRIG}	PCG Trigger Hold After Falling Edge of PCG Input Clock	3		ns
Switching Characteristics				
t_{DPCGIO}	PCG Output Clock and Frame Sync Active Edge Delay After PCG Input Clock	2.5	13.5	ns
$t_{DTRIGCLK}$	PCG Output Clock Delay After PCG Trigger	$2.5 + (2.5 \times t_{PCGIP})$	$13.5 + (2.5 \times t_{PCGIP})$	ns
$t_{DTRIGFS}$[1]	PCG Frame Sync Delay After PCG Trigger	$2.5 + ((2.5 + D - PH) \times t_{PCGIP})$	$13.5 + ((2.5 + D - PH) \times t_{PCGIP})$	ns
t_{PCGOW}[2]	Output Clock Period	$2 \times t_{PCGIP} - 1$		ns

[1] D = FSxDIV, PH = FSxPHASE. For more information, see the ADSP-SC57x/ADSP-2157x SHARC+ Processor Hardware Reference.
[2] Normal mode of operation.

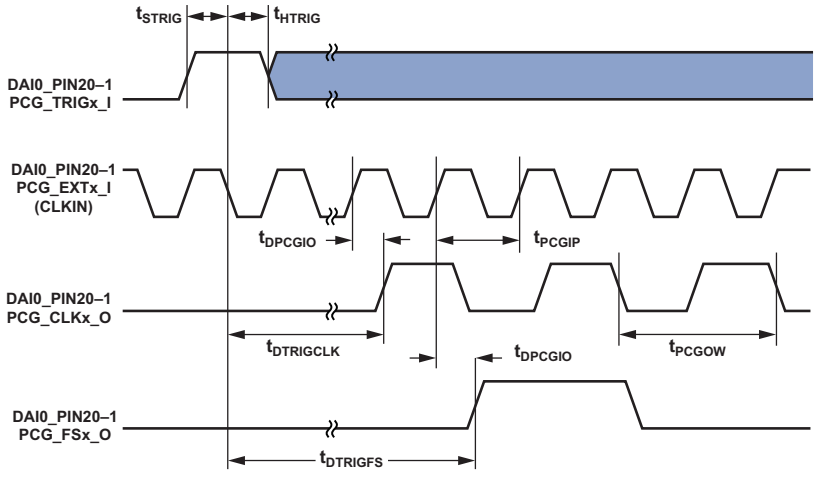


Figure 43. PCG (Direct Pin Routing)

Downloaded from Arrow.com.

General-Purpose IO Port Timing

Table 72 and Figure 44 describe I/O timing, related to the general-purpose ports (PORT).

Table 72. General-Purpose Port Timing

Parameter		Min	Max	Unit
Timing Requirement				
t_{WFI}	General-Purpose Port Pin Input Pulse Width	$2 \times t_{SCLK0} - 1.5$		ns

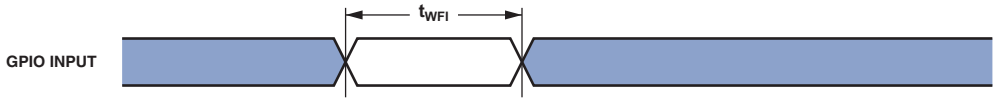


Figure 44. General-Purpose Port Timing

General-Purpose I/O Timer Cycle Timing

Table 73, Table 74, and Figure 45 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 absolute maximum input frequency of $f_{SCLK}/4$ MHz. The Width Value value is the timer period assigned in the TMx_TMRn_WIDTH register and can range from 1 to $2^{32} - 1$. When externally generated, the TMx_CLK clock is called $f_{TMRCLKEXT}$:

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

Table 73. Timer Cycle Timing—Internal Mode

Parameter		Min	Max	Unit
Timing Requirements				
t_{WL}	Timer Pulse Width Input Low (Measured In SCLK Cycles)[1]	$2 \times t_{SCLK}$		ns
t_{WH}	Timer Pulse Width Input High (Measured In SCLK Cycles)[1]	$2 \times t_{SCLK}$		ns
Switching Characteristic				
t_{HTO}	Timer Pulse Width Output (Measured In SCLK Cycles)[2]	$t_{SCLK} \times$ WIDTH – 1.5	$t_{SCLK} \times$ WIDTH + 1.5	ns

[1] The minimum pulse width applies for timer signals in width capture and external clock modes.
[2] WIDTH refers to the value in the TMRx_WIDTH register (it can vary from 2 to $2^{32} - 1$).

Table 74. Timer Cycle Timing—External Mode

Parameter		Min	Max	Unit
Timing Requirements				
t_{WL}	Timer Pulse Width Input Low (Measured In EXT_CLK Cycles)[1]	$2 \times t_{EXT_CLK}$		ns
t_{WH}	Timer Pulse Width Input High (Measured In EXT_CLK Cycles)[1]	$2 \times t_{EXT_CLK}$		ns
t_{EXT_CLK}	Timer External Clock Period[2]	$t_{TMRCLKEXT}$		ns
Switching Characteristic				
t_{HTO}	Timer Pulse Width Output (Measured In EXT_CLK Cycles)[3]	$t_{EXT_CLK} \times$ WIDTH – 1.5	$t_{EXT_CLK} \times$ WIDTH + 1.5	ns

[1] The minimum pulse width applies for timer signals in width capture and external clock modes.
[2] This specification indicates the minimum instantaneous width or period that can be tolerated due to duty cycle variation or jitter on the external TMR_CLK. For the external TMR_CLK maximum frequency, see the $f_{TMRCLKEXT}$ specification in Table 27.
[3] WIDTH refers to the value in the TMRx_WIDTH register (it can vary from 1 to $2^{32} - 1$).

Downloaded from Arrow.com.

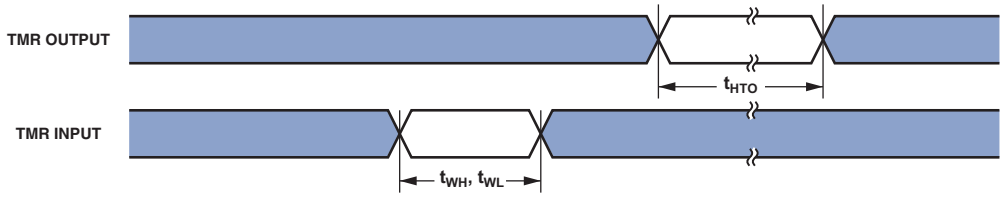


Figure 45. Timer Cycle Timing

DAI0 Pin to DAI0 Pin Direct Routing

Table 75 and Figure 46 describe I/O timing related to the DAI for direct pin connections only (for example, DAI0_PB01_I to DAI0_PB02_O).

Table 75. DAI Pin to DAI Pin Routing

Parameter		Min	Max	Unit
Switching Characteristic				
t_{DPIO}	Delay DAI Pin Input Valid to DAI Output Valid	1.5	12	ns

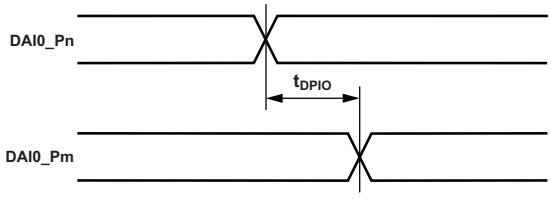


Figure 46. DAI Pin to DAI Pin Direct Routing

Up/Down Counter/Rotary Encoder Timing

Table 76 and Figure 47 describe timing, related to the general-purpose counter (CNT).

Table 76. Up/Down Counter/Rotary Encoder Timing

Parameter		Min	Max	Unit
Timing Requirement				
t_{WCOUNT}	Up/Down Counter/Rotary Encoder Input Pulse Width	$2 \times t_{SCLK0}$		ns

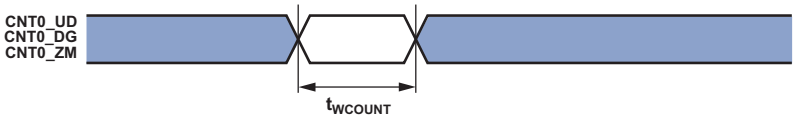


Figure 47. Up/Down Counter/Rotary Encoder Timing

Downloaded from Arrow.com.

ADC Controller Module (ACM) Timing

Table 77 and Figure 48 describe ACM operations.

When internally generated, the programmed ACM clock ($f_{ACLKPROG}$) frequency in megahertz is set by the following equation where CKDIV is a field in the ACM_TC0 register and ranges from 1 to 255:

$$f_{ACLKPROG} = \frac{f_{SCLK1}}{CKDIV + 1}$$

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

Setup cycles (SC) in Table 77 is also a field in the ACM_TC0 register and ranges from 0 to 4095. Hold Cycles (HC) is a field in the ACM_TC1 register that ranges from 0 to 15.

Table 77. ACM Timing

Parameter		Min	Max	Unit
Timing Requirements				
t_{SDR}	SPORT DRxPRI/DRxSEC Setup Before ACMx_CLK	3.4		ns
t_{HDR}	SPORT DRxPRI/DRxSEC Hold After ACMx_CLK	1.5		ns
Switching Characteristics				
t_{SCTLCS}	ACM Controls (ACMx_A[4:0]) Setup Before Assertion of $\overline{CS}$	$(SC + 1) \times t_{SCLK1} - 4.88$		ns
t_{HCTLCS}	ACM Control (ACMx_A[4:0]) Hold After Deassertion of $\overline{CS}$	$HC \times t_{ACLKPROG} - 1$		ns
t_{ACLKW}	ACM Clock Pulse Width[1]	$(0.5 \times t_{ACLKPROG}) - 1.6$		ns
t_{ACLK}	ACM Clock Period[1]	$t_{ACLKPROG} - 1.5$		ns
$t_{HCSACLK}$	$\overline{CS}$ Hold to ACMx_CLK Edge	–2.5		ns
$t_{SCSACLK}$	$\overline{CS}$ Setup to ACMx_CLK Edge	$t_{ACLKPROG} - 3.5$		ns

[1] See Table 27 for details on the minimum period that can be programmed for $t_{ACLKPROG}$.

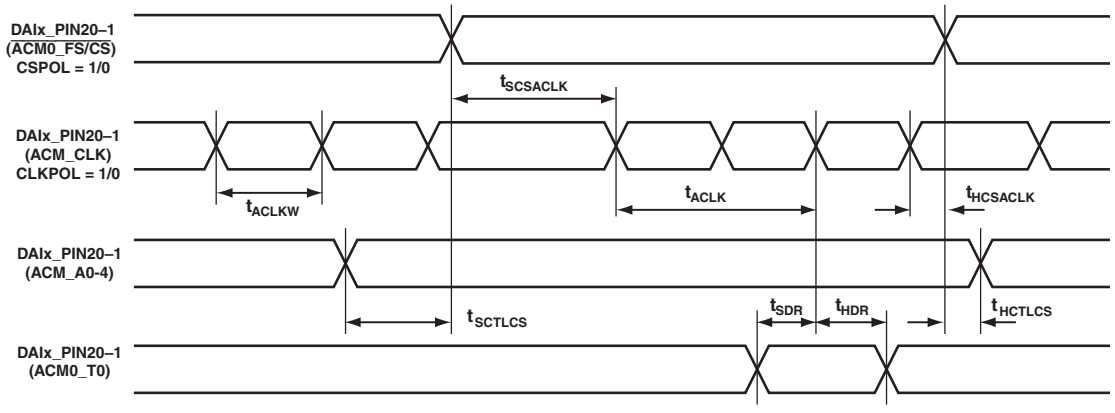


Figure 48. ACM Timing

Downloaded from Arrow.com.

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

The UART ports receive and transmit operations are described in the ADSP-SC57x/ADSP-2157x SHARC+ Processor Hardware Reference.

Controller Area Network (CAN) Interface

The CAN interface timing is described in the ADSP-SC57x/ADSP-2157x SHARC+ Processor Hardware Reference.

Universal Serial Bus (USB)

Table 78 describes the universal serial bus (USB) clock timing. Refer to the *USB 2.0 Specification* for timing and dc specifications for USB pins (including output characteristics for driver types E, F, and G listed in the ADSP-SC57x/ADSP-2157x Designer Quick Reference).

Table 78. USB Clock Timing[1]

Parameter		Min	Max	Unit
Timing Requirements				
f_{USBS}	USB_CLKIN Frequency	24	24	MHz
fs_{USB}	USB_CLKIN Clock Frequency Stability	–50	+50	ppm

[1] This specification is supported by USB0.

Downloaded from Arrow.com.

10/100 EMAC Timing

Table 79 through Table 83 and Figure 49 through Figure 53 describe the MII and RMII EMAC operations.

Table 79. 10/100 EMAC Timing: MII Receive Signal

Parameter[1]		V_{DDEXT} 3.3 V Nominal Min	Max	Unit
Timing Requirements				
$t_{ERXCLKF}$	ETH0_RXCLK_REFCLK Frequency (f_{SCLK} = SCLK Frequency)	None	25 + 1%	MHz
$t_{ERXCLKW}$	ETH0_RXCLK_REFCLK Width (t_{ERxCLK} = ETH0_RXCLK_REFCLK Period)	t_{ERxCLK} × 35%	t_{ERxCLK} × 65%	ns
$t_{ERXCLKIS}$	Rx Input Valid to ETH0_RXCLK_REFCLK Rising Edge (Data In Setup)	1.75		ns
$t_{ERXCLKIH}$	ETH0_RXCLK_REFCLK Rising Edge to Rx Input Invalid (Data In Hold)	1.5		ns

[1] MII inputs synchronous to ETH0_RXCLK_REFCLK are ETH0_RXD3-0, ETH0_RXCTL_RXDV, and ETH0_RXERR.

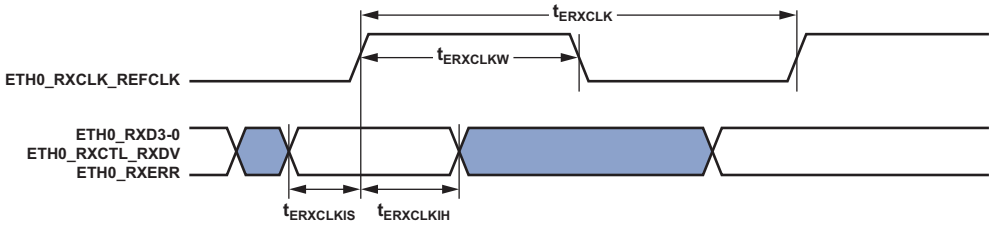


Figure 49. 10/100 EMAC Timing: MII Receive Signal

Table 80. 10/100 EMAC Timing: MII Transmit Signal

Parameter[1]		V_{DDEXT} 3.3 V Nominal Min	Max	Unit
Timing Requirements				
$t_{ETXCLKF}$	ETH0_TXCLK Frequency (f_{SCLK} = SCLK Frequency)	None	25 + 1%	MHz
$t_{ETXCLKW}$	ETH0_TXCLK Width (t_{ETxCLK} = ETH0_TXCLK Period)	t_{ETxCLK} × 35%	t_{ETxCLK} × 65%	ns
Switching Characteristics				
$t_{ETXCLKOV}$	ETH0_TXCLK Rising Edge to Tx Output Valid (Data Out Valid)		11.4	ns
$t_{ETXCLKOH}$	ETH0_TXCLK Rising Edge to Tx Output Invalid (Data Out Hold)	2		ns

[1] MII outputs synchronous to ETH0_TXCLK are ETH0_TXD3-0.

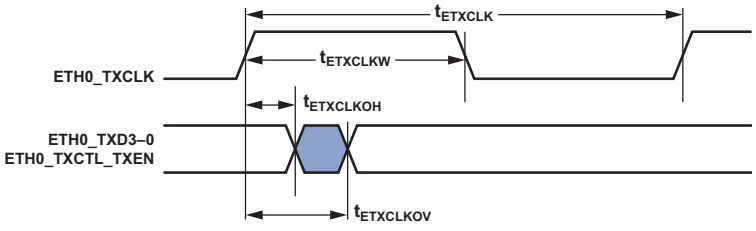


Figure 50. 10/100 EMAC Timing: MII Transmit Signal

Downloaded from Arrow.com.

Table 81. 10/100 EMAC Timing—RMII Receive Signal

Parameter[1]		Min	Max	Unit
Timing Requirements				
$t_{REFCLKF}$	ETH0_RXCLK_REFCLK Frequency (f_{SCLK0} = SCLK0 Frequency)	None	50 + 1%	MHz
$t_{REFCLKW}$	ETH0_RXCLK_REFCLK Width ($t_{REFCLKF}$ = ETH0_RXCLK_REFCLK Period)	t_{REFCLK} × 35%	t_{REFCLK} × 65%	ns
$t_{REFCLKIS}$	Rx Input Valid to RMII ETH0_RXCLK_REFCLK Rising Edge (Data In Setup)	1.75		ns
$t_{REFCLKIH}$	RMII ETH0_RXCLK_REFCLK Rising Edge to Rx Input Invalid (Data In Hold)	1.6		ns

[1] RMII inputs synchronous to RMII ETH0_RXCLK_REFCLK are ETH0_RXD1–0, RMII ETH0_RXCTL_RXDV, and ETH0_RXERR.

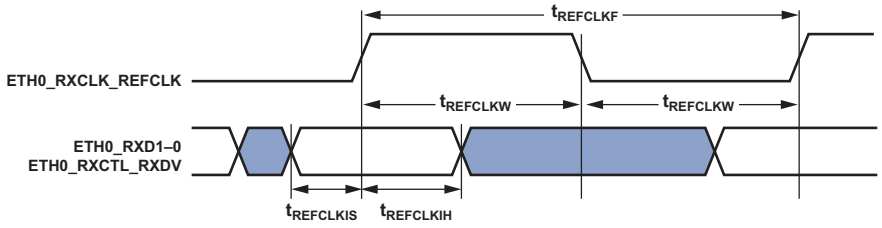


Figure 51. 10/100 EMAC Controller Timing—RMII Receive Signal

Table 82. 10/100 EMAC Timing—RMII Transmit Signal

Parameter[1]		Min	Max	Unit
Switching Characteristics				
$t_{REFCLKOV}$	RMII ETH0_RXCLK_REFCLK Rising Edge to Transmit Output Valid (Data Out Valid)		11.9	ns
$t_{REFCLKOH}$	RMII ETH0_RXCLK_REFCLK Rising Edge to Transmit Output Invalid (Data Out Hold)	2		ns

[1] RMII outputs synchronous to RMII ETH0_RXCLK_REFCLK are ETH0_TXD1–0.

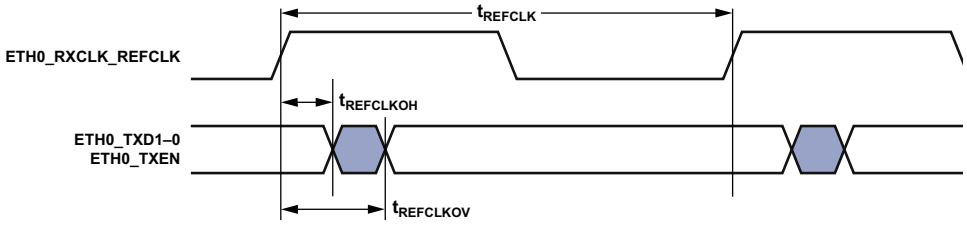


Figure 52. 10/100 EMAC Controller Timing—RMII Transmit Signal

Downloaded from Arrow.com.

Table 83. 10/100/1000 EMAC Timing—RMII and RGMII Station Management

Parameter[1]		Min	Max	Unit
Timing Requirements				
t_{MDIOS}	ETH0_MDIO Input Valid to ETH0_MDC Rising Edge (Setup)	12.6		ns
t_{MDCIH}	ETH0_MDC Rising Edge to ETH0_MDIO Input Invalid (Hold)	0		ns
Switching Characteristics				
t_{MDCOV}	ETH0_MDC Falling Edge to ETH0_MDIO Output Valid		t_{SCLK0} + 2	ns
t_{MDCOH}	ETH0_MDC Falling Edge to ETH0_MDIO Output Invalid (Hold)	t_{SCLK0} −2.9		ns

[1] ETH0_MDC/ETH0_MDIO is a 2-wire serial bidirectional port for controlling one or more external PHYs. ETH0_MDC is an output clock with a minimum period that is programmable as a multiple of the system clock SCLK0. ETH0_MDIO is a bidirectional data line.

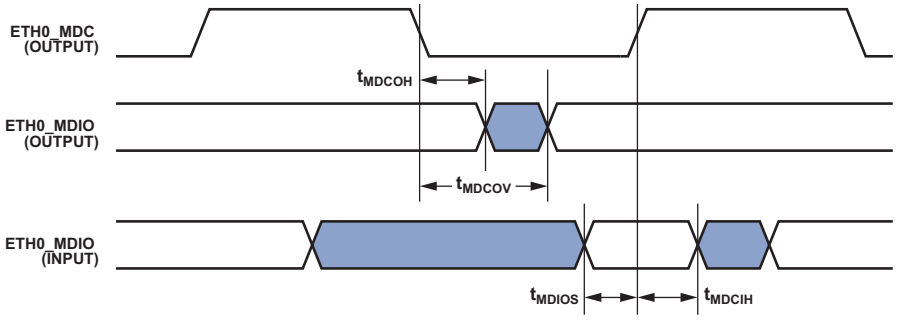


Figure 53. 10/100 /1000 Ethernet MAC Controller Timing—RMII and RGMII Station Management

Downloaded from Arrow.com.

10/100/1000 EMAC Timing

Table 84 and Figure 54 describe the RGMII EMAC timing.

Table 84. 10/100/1000 EMAC Timing—RGMII Receive and Transmit Signals

Parameter		Min	Max	Unit
Timing Requirements				
t_{SETUPR}	Data to Clock Input Setup at Receiver	1		ns
t_{HOLDR}	Data to Clock Input Hold at Receiver	1		ns
$t_{GREFCLKF}$	RGMII Receive Clock Period	8		ns
$t_{GREFCLKW}$	RGMII Receive Clock Pulse Width	4		ns
Switching Characteristics				
t_{SKEWT}	Data to Clock Output Skew at Transmitter	–0.5	+0.5	ns
t_{CYC}	Clock Cycle Duration	7.2	8.8	ns
t_{DUTY_G}	Duty Cycle for RGMII Minimum	$t_{GREFCLKF} \times 45\%$	$t_{GREFCLKF} \times 55\%$	ns

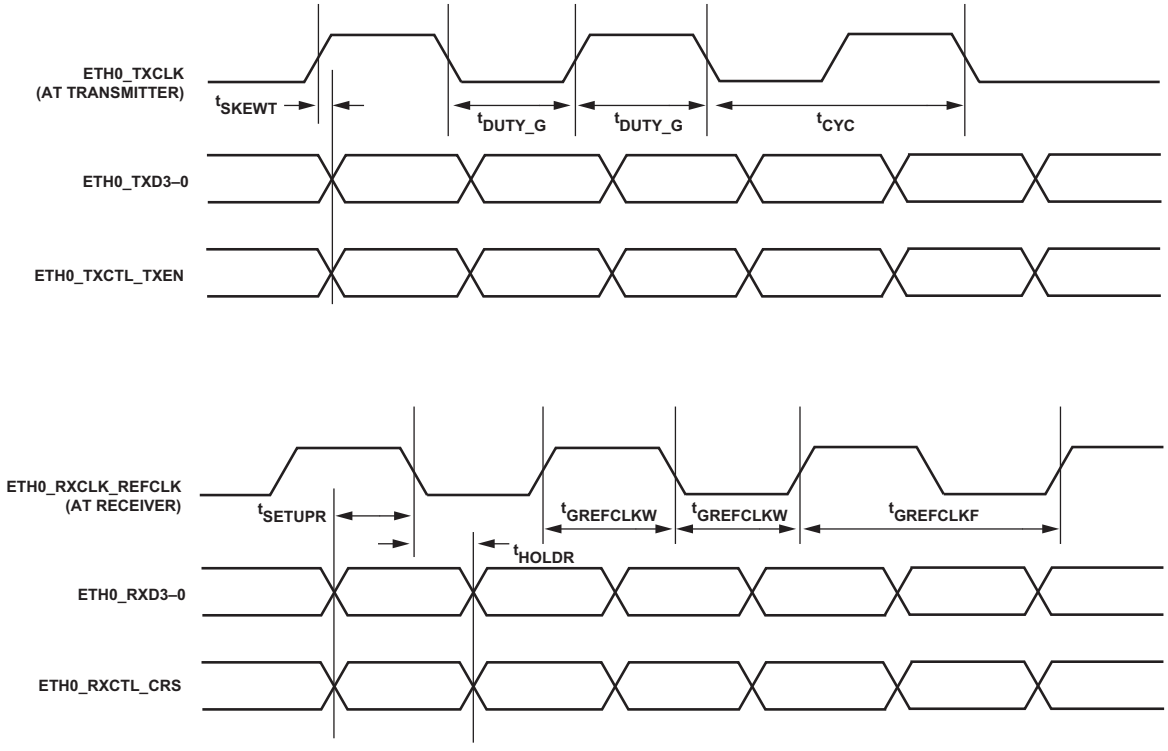


Figure 54. EMAC Timing—RGMII

Downloaded from Arrow.com.

Sony/Philips Digital Interface (S/PDIF) Transmitter

Serial data input to the S/PDIF transmitter can be formatted as left justified, I^2S, or right justified with word widths of 16, 18, 20, or 24 bits. The following sections provide timing for the transmitter.

S/PDIF Transmitter Serial Input Waveforms

Figure 55 and Table 85 show the right justified mode. Frame sync is high for the left channel and low for the right channel. Data is valid on the rising edge of serial clock. The MSB is delayed the minimum in 24-bit output mode or the maximum in 16-bit output mode from a frame sync transition, so that when there are 64 serial clock periods per frame sync period, the LSB of the data is right justified to the next frame sync transition.

Table 85. S/PDIF Transmitter Right Justified Mode

Parameter		Conditions	Nominal	Unit
Timing Requirement				
t_{RJD}	Frame Sync to MSB Delay in Right Justified Mode	16-bit word mode	16	SCLK
		18-bit word mode	14	SCLK
		20-bit word mode	12	SCLK
		24-bit word mode	8	SCLK

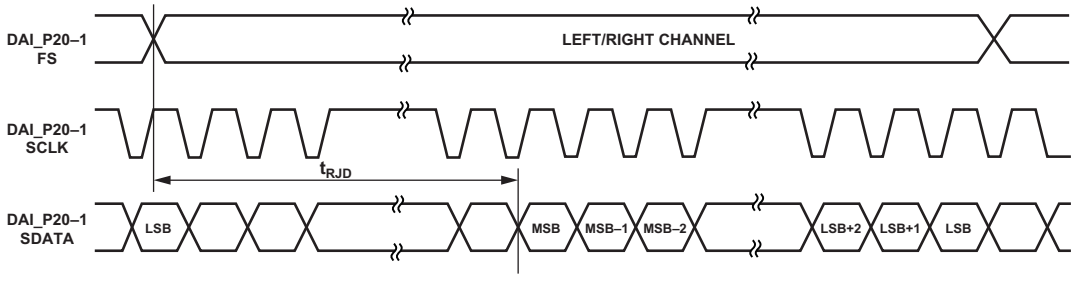


Figure 55. Right Justified Mode

Downloaded from Arrow.com.

Figure 56 and Table 86 show the default I²S justified mode. The frame sync is low for the left channel and high for the right channel. Data is valid on the rising edge of serial clock. The MSB is left justified to the frame sync transition but with a delay.

Table 86. S/PDIF Transmitter I²S Mode

Parameter		Nominal	Unit
Timing Requirement			
t_{I2SD}	Frame Sync to MSB Delay in I²S Mode	1	SCLK

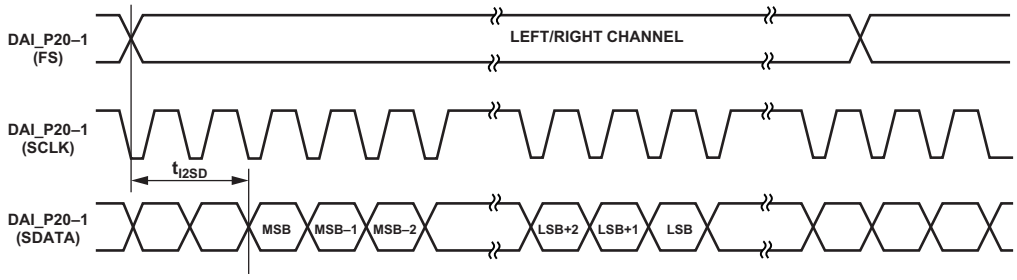


Figure 56. I²S Justified Mode

Figure 57 and Table 87 show the left justified mode. The frame sync is high for the left channel and low for the right channel. Data is valid on the rising edge of serial clock. The MSB is left justified to the frame sync transition with no delay.

Table 87. S/PDIF Transmitter Left Justified Mode

Parameter		Nominal	Unit
Timing Requirement			
t_{LJD}	Frame Sync to MSB Delay in Left Justified Mode	0	SCLK

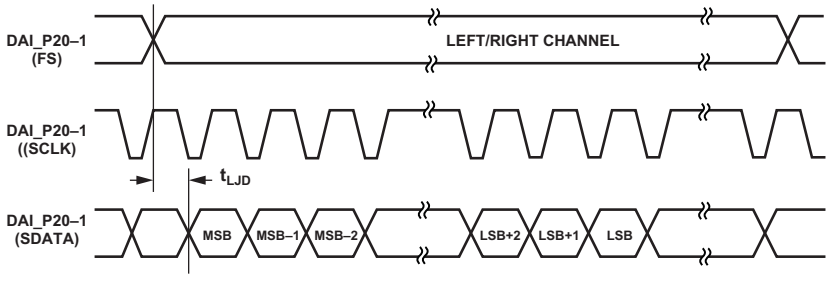


Figure 57. Left Justified Mode

Downloaded from Arrow.com.

S/PDIF Transmitter Input Data Timing

The timing requirements for the S/PDIF transmitter are given in Table 88. Input signals are routed to the DAI0_PINx pins using the SRU. Therefore, the timing specifications provided below are valid at the DAI0_PINx pins.

Table 88. S/PDIF Transmitter Input Data Timing

Parameter		Min	Max	Unit
Timing Requirements				
t_{SISFS}[1]	Frame Sync Setup Before Serial Clock Rising Edge	3		ns
t_{SIHFS}[1]	Frame Sync Hold After Serial Clock Rising Edge	3		ns
t_{SISD}[1]	Data Setup Before Serial Clock Rising Edge	3		ns
t_{SIHD}[1]	Data Hold After Serial Clock Rising Edge	3		ns
$t_{SITXCLKW}$	Transmit Clock Width	9		ns
$t_{SITXCLK}$	Transmit Clock Period	20		ns
$t_{SISCLKW}$	Clock Width	36		ns
t_{SISCLK}	Clock Period	80		ns

[1] The serial clock, data, and frame sync signals can come from any of the DAI pins. The serial clock and frame sync signals can also come via PCG or SPORTs. The input of the PCG can be either CLKIN or any of the DAI pins.

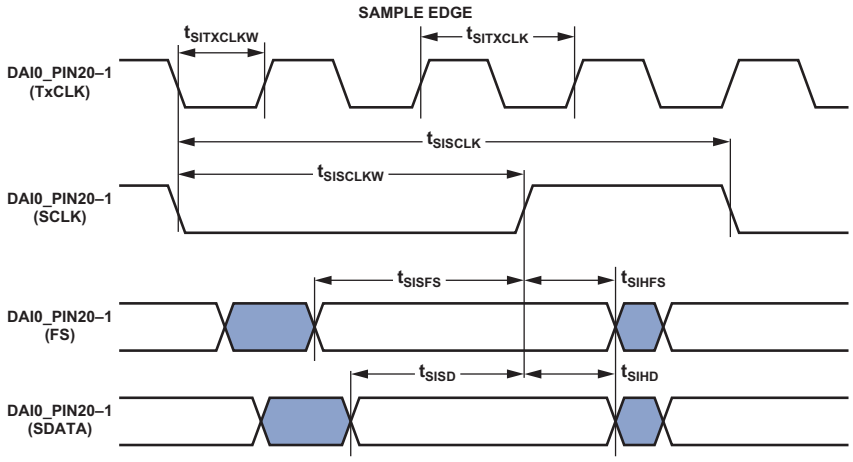


Figure 58. S/PDIF Transmitter Input Timing

Oversampling Clock (TxCLK) Switching Characteristics

The S/PDIF transmitter requires an oversampling clock input. This high frequency clock (TxCLK) input is divided down to generate the internal biphase clock.

Table 89. Oversampling Clock (TxCLK) Switching Characteristics

Parameter		Max	Unit
Switching Characteristics			
f_{TXCLK_384}	Frequency for TxCLK = 384 × Frame Sync	Oversampling ratio × frame sync ≤ $1/t_{SITXCLK}$	MHz
f_{TXCLK_256}	Frequency for TxCLK = 256 × Frame Sync	49.2	MHz
f_{FS}	Frame Rate (FS)	192.0	kHz

Downloaded from Arrow.com.

S/PDIF Receiver

The following section describes timing as it relates to the S/PDIF receiver.

Internal Digital PLL Mode

In the internal digital PLL mode, the internal digital PLL generates the 512 × FS clock.

Table 90. S/PDIF Receiver Internal Digital PLL Mode Timing

Parameter		Min	Max	Unit
Switching Characteristics				
t_{DFSI}	Frame Sync Delay After Serial Clock		5	ns
t_{HOFSI}	Frame Sync Hold After Serial Clock	–2		ns
t_{DDTI}	Transmit Data Delay After Serial Clock		5	ns
t_{HDTI}	Transmit Data Hold After Serial Clock	–2		ns

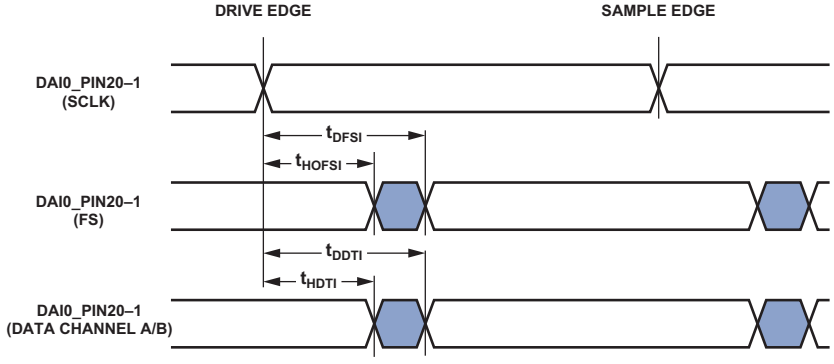


Figure 59. S/PDIF Receiver Internal Digital PLL Mode Timing

Downloaded from Arrow.com.

MediaLB (MLB)

All the numbers shown in Table 91 are applicable for all MLB speed modes (1024 FS, 512 FS, and 256 FS) for the 3-pin protocol, unless otherwise specified. Refer to the *Media Local Bus Specification version 4.2* for more details.

Table 91. 3-Pin MLB Interface Specifications

Parameter		Min	Typ	Max	Unit
t_{MLBCLK}	MLB Clock Period				
	1024 FS		20.3		ns
	512 FS		40		ns
	256 FS		81		ns
t_{MCKL}	MLBCLK Low Time				
	1024 FS	6.1			ns
	512 FS	14			ns
	256 FS	30			ns
t_{MCKH}	MLBCLK High Time				
	1024 FS	9.3			ns
	512 FS	14			ns
	256 FS	30			ns
t_{MCKR}	MLBCLK Rise Time (V_{IL} to V_{IH})				
	1024 FS			1	ns
	512 FS/256 FS			3	ns
t_{MCKF}	MLBCLK Fall Time (V_{IH} to V_{IL})				
	1024 FS			1	ns
	512 FS/256 FS			3	ns
t_{MPWV}[1]	MLBCLK Pulse Width Variation				
	1024 FS			0.7	nspp
	512 FS/256			2.0	nspp
t_{DSMCF}	DAT/SIG Input Setup Time	1			ns
t_{DHMCF}	DAT/SIG Input Hold Time	2			ns
t_{MCFDZ}	DAT/SIG Output Time to Three-State	0		15	ns
t_{MCDRV}	DAT/SIG Output Data Delay From MLBCLK Rising Edge			8	ns
t_{MDZH}[2]	Bus Hold Time				
	1024 FS	2			ns
	512 FS/256	4			ns
C_{MLB}	DAT/SIG Pin Load				
	1024 FS			40	pf
	512 FS/256			60	pf

[1] Pulse width variation is measured at 1.25 V by triggering on one edge of MLBCLK and measuring the spread on the other edge, measured in nanoseconds peak-to-peak.
[2] Board designs must ensure the high impedance bus does not leave the logic state of the final driven bit for this time period. Therefore, coupling must be minimized while meeting the maximum capacitive load listed.

Downloaded from Arrow.com.

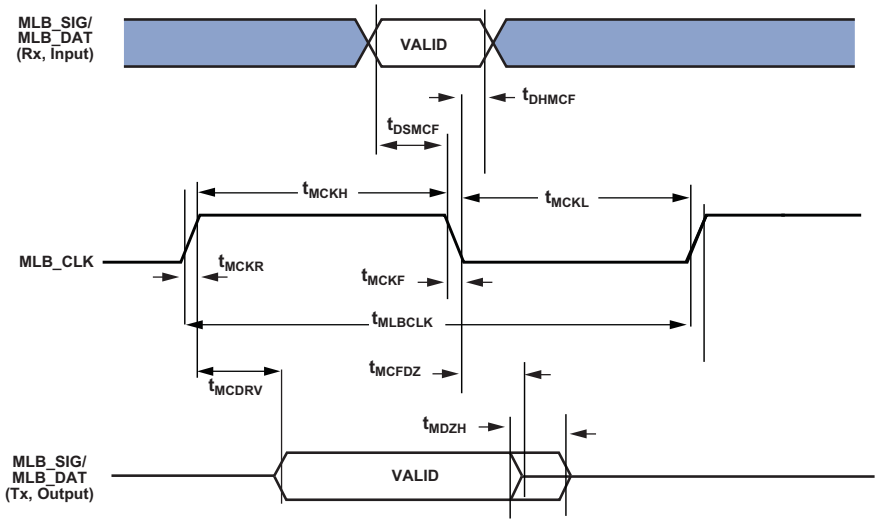


Figure 60. MLB Timing (3-Pin Interface)

The ac timing specifications of the 6-pin MLB interface is detailed in Table 92. Refer to the *Media Local Bus Specification version 4.2* for more details.

Table 92. 6-Pin MLB Interface Specifications

Parameter		Conditions	Min	Typ	Max	Unit
t_{MT}	Differential Transition Time at the Input Pin (See Figure 61)	20% to 80% $V_{IN}+/V_{IN}-$ 80% to 20% $V_{IN}+/V_{IN}-$			1	ns
f_{MCKE}	MLBCP/N External Clock Operating Frequency (See Figure 62)[1]	2048 × FS at 44.0 kHz	90.112			MHz
		2048 × FS at 50.0 kHz			102.4	MHz
f_{MCKR}	Recovered Clock Operating Frequency (Internal, Not Observable at Pins, Only for Timing References) (See Figure 62)	2048 × FS at 44.0 kHz	90.112			MHz
		2048 × FS at 50.0 kHz			102.4	MHz
t_{DELAY}	Transmitter MLBSP/N (MLBDP/N) Output Valid From Transition of MLBCP/N (Low to High) (See Figure 63)	f_{MCKR} = 2048 × FS	0.6		5	ns
t_{PHZ}	Disable Turnaround Time From Transition of MLBCP/N (Low to High) (See Figure 64)	f_{MCKR} = 2048 × FS	0.6		7	ns
t_{PLZ}	Enable Turnaround Time From Transition of MLBCP/N (Low to High) (See Figure 64)	f_{MCKR} = 2048 × FS	0.6		11.2	ns
t_{SU}	MLBSP/N (MLBDP/N) Valid to Transition of MLBCP/N (Low to High) (See Figure 63)	f_{MCKR} = 2048 × FS	1			ns
t_{HD}	MLBSP/N (MLBDP/N) Hold From Transition of MLBCP/N (Low to High) (See Figure 63)[2]		0.6			ns

[1] f_{MCKE} (maximum) and f_{MCKR} (maximum) include maximum cycle to cycle system jitter (t_{JITTER}) of 600 ps for a bit error rate of 10E-9.
[2] Receivers must latch MLBSP/N (MLBDP/N) data within t_{HD} (minimum) of the rising edge of MLBCP/N.

Downloaded from Arrow.com.

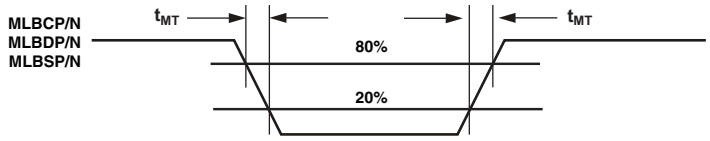


Figure 61. MLB 6-Pin Transition Time

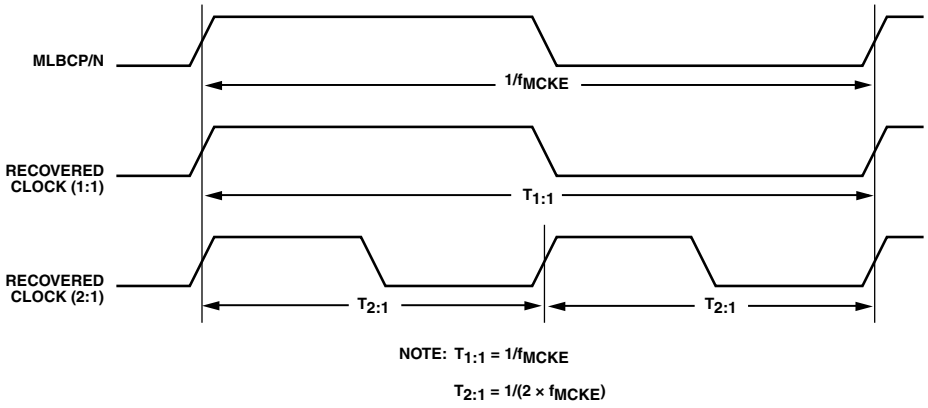


Figure 62. MLB 6-Pin Clock Definitions

Downloaded from Arrow.com.

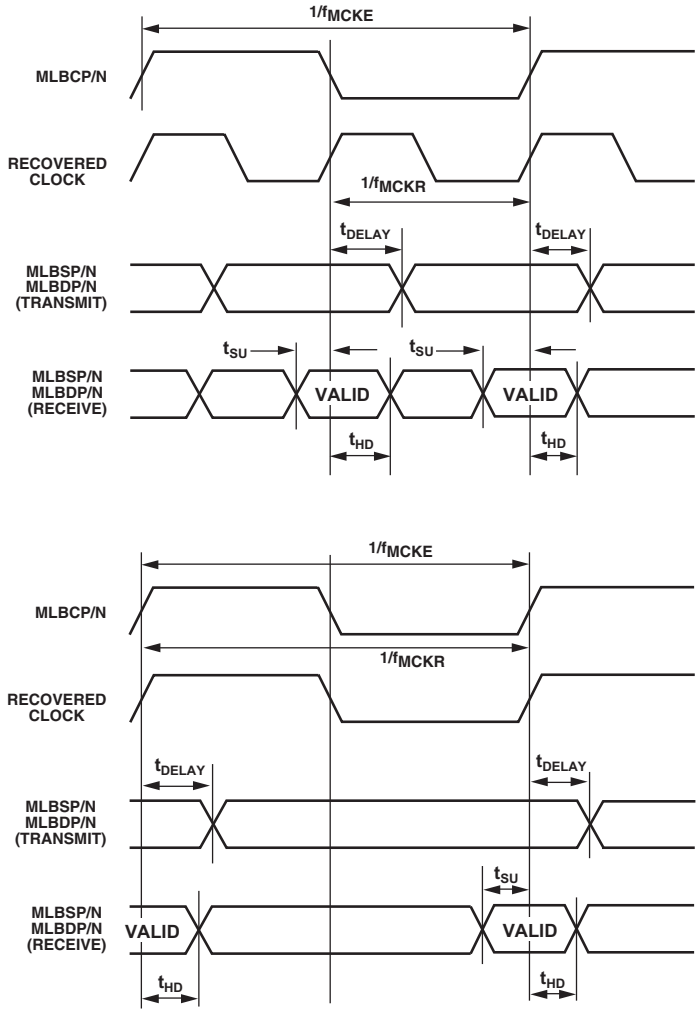


Figure 63. MLB 6-Pin Delay, Setup, and Hold Times

Downloaded from Arrow.com.

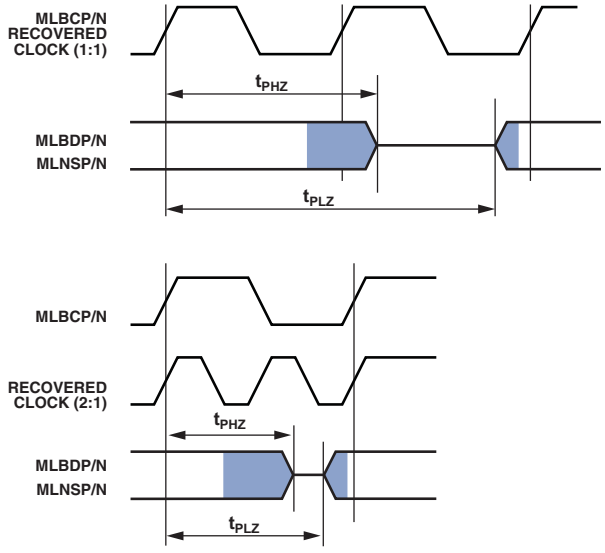


Figure 64. MLB 6-Pin Disable and Enable Turnaround Times

Downloaded from Arrow.com.

Mobile Storage Interface (MSI) Controller Timing

Table 93 and Figure 65 show I/O timing related to the MSI.

Table 93. MSI Controller Timing

Parameter		Min	Max	Unit
Timing Requirements				
t_{ISU}	Input Setup Time	4.8		ns
t_{IH}	Input Hold Time	–0.5		ns
Switching Characteristics				
f_{PP}	Clock Frequency Data Transfer Mode[1]		45	MHz
t_{WL}	Clock Low Time	8		ns
t_{WH}	Clock High Time	8		ns
t_{TLH}	Clock Rise Time		3	ns
t_{THL}	Clock Fall Time		3	ns
t_{ODLY}	Output Delay Time During Data Transfer Mode		2.1	ns
t_{OH}	Output Hold Time	–1.8		ns

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

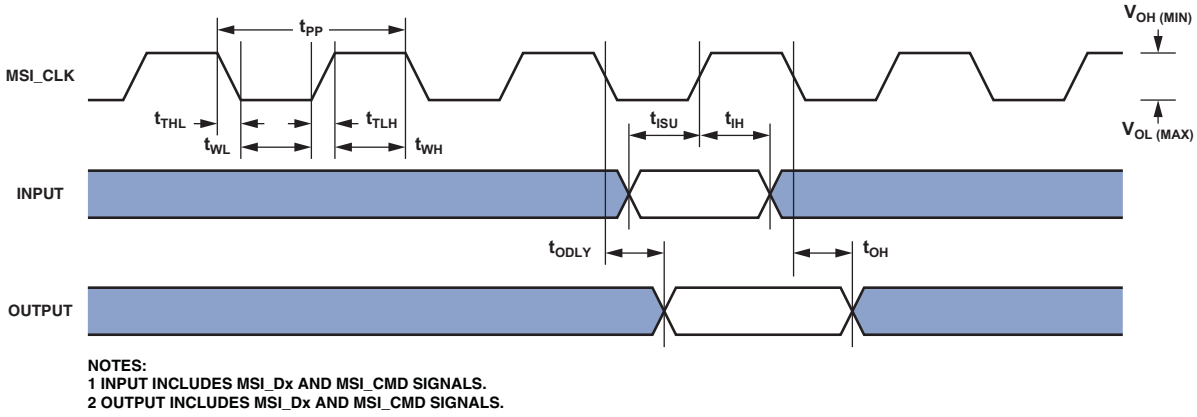


Figure 65. MSI Controller Timing

Downloaded from Arrow.com.

Program Trace Macrocell (PTM) Timing

Table 94 and Figure 66 provide I/O timing related to the PTM.

Table 94. Trace Timing

Parameter		Min	Max	Unit
Switching Characteristics				
t_{DTRD}	TRACE Data Delay From Trace Clock Maximum		$0.5 \times t_{SCLK0} + 4$	ns
t_{HTRD}	TRACE Data Hold From Trace Clock Minimum	$0.5 \times t_{SCLK0} - 2.2$		ns
t_{PTRCK}	TRACE Clock Period Minimum	$2 \times t_{SCLK0} - 1$		ns

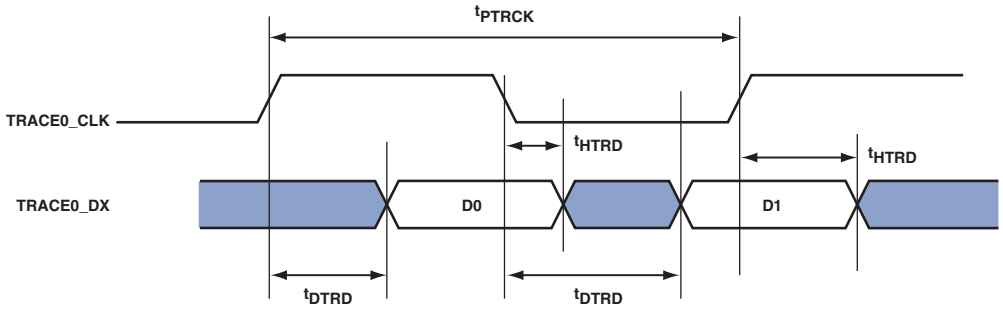


Figure 66. Trace Timing

Downloaded from Arrow.com.

Debug Interface (JTAG Emulation Port) Timing

Table 95 and Figure 67 provide I/O timing related to the debug interface (JTAG Emulator Port).

Table 95. JTAG Emulation Port Timing

Parameter		Min	Max	Unit
Timing Requirements				
t_{TCK}	JTG_TCK Period	20		ns
t_{STAP}	JTG_TDI, JTG_TMS Setup Before JTG_TCK High	4		ns
t_{HTAP}	JTG_TDI, JTG_TMS Hold After JTG_TCK High	4		ns
t_{SSYS}	System Inputs Setup Before JTG_TCK High[1]	12		ns
t_{HSYS}	System Inputs Hold After JTG_TCK High[1]	5		ns
t_{TRSTW}	$\overline{\text{JTG_TRST}}$ Pulse Width (measured in JTG_TCK cycles)[2]	4		T_{CK}
Switching Characteristics				
t_{DTDO}	JTG_TDO Delay From JTG_TCK Low		13.5	ns
t_{DSYS}	System Outputs Delay After JTG_TCK Low[3]		17	ns

[1] System Inputs = MLB0_CLKP, MLB0_DATP, MLB0_SIGP, DAI0_PIN20-01, DMC0_A15-0, DMC0_DQ15-0, $\overline{\text{DMC0_RESET}}$, PA_15-0, PB_15-0, PC_15-0, PD_15-0, PE_15-0, PF_11-0, SYS_BMODE2-0, SYS_FAULT, $\overline{\text{SYS_FAULT}}$, $\overline{\text{SYS_RESOUT}}$, TWI2-0_SCL, TWI2-0_SDA2.

[2] 50 MHz maximum.

[3] System Outputs = DMC0_A15-0, DMC0_BA2-0, $\overline{\text{DMC0_CAS}}$, DMC0_CK, DMC0_CKE, $\overline{\text{DMC0_CS0}}$, DMC0_DQ15-0, DMC0_LDM, DMC0_LDQS, DMC0_ODT, $\overline{\text{DMC0_RAS}}$, $\overline{\text{DMC0_RESET}}$, DMC0_UDM, DMC0_UDQS, $\overline{\text{DMC0_WE}}$, MLB0_DATP, MLB0_SIGP, PA_15-0, PB_15-0, PC_15-0, PD_15-0, PE_15-0, PF_11-0, SYS_BMODE2-0, SYS_CLKOUT, SYS_FAULT, $\overline{\text{SYS_FAULT}}$, $\overline{\text{SYS_RESOUT}}$.

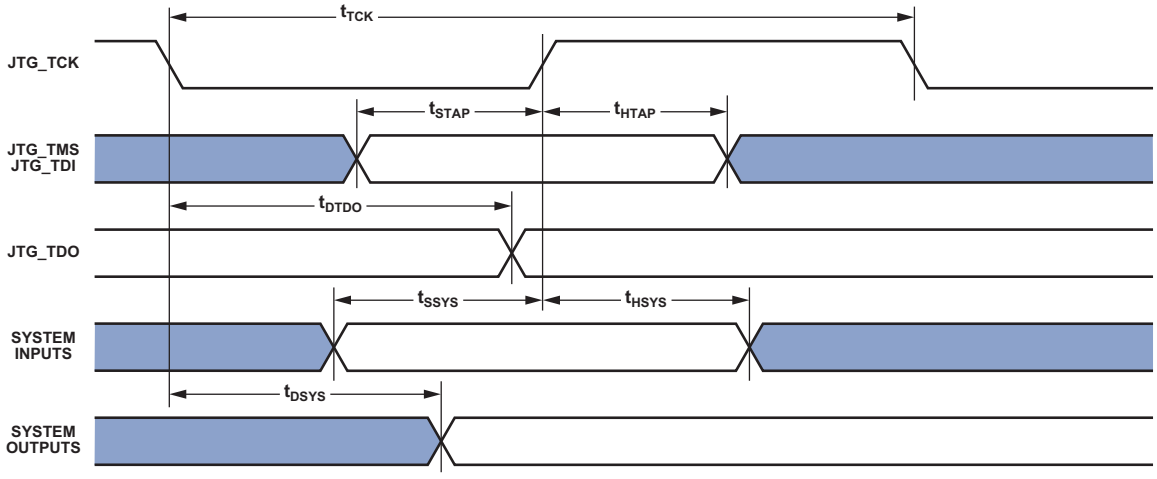


Figure 67. JTAG Port Timing

Downloaded from Arrow.com.

OUTPUT DRIVE CURRENTS

Figure 68 through Figure 80 show typical current-voltage characteristics for the output drivers of the ADSP-SC57x and ADSP-2157x processors. The curves represent the current drive capability of the output drivers as a function of output voltage.

Output drive currents for MLB pins are compliant with MOST150 LVDS specifications. Output drive currents for USB pins are compliant with the USB 2.0 specifications.

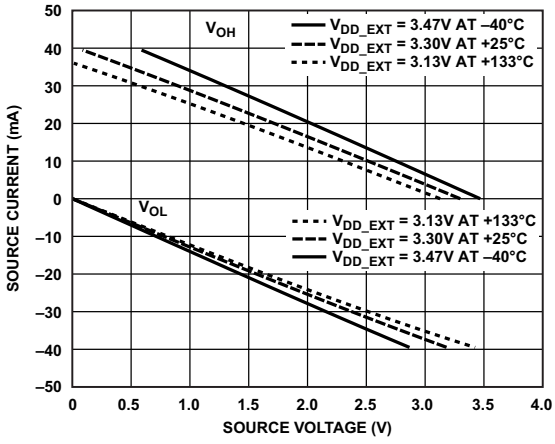


Figure 68. Driver Type A Current (3.3 V V_{DD_EXT})

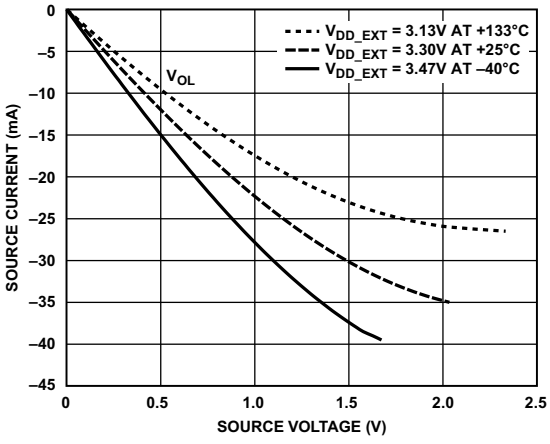


Figure 69. Driver Type D Current (3.3 V V_{DD_EXT})

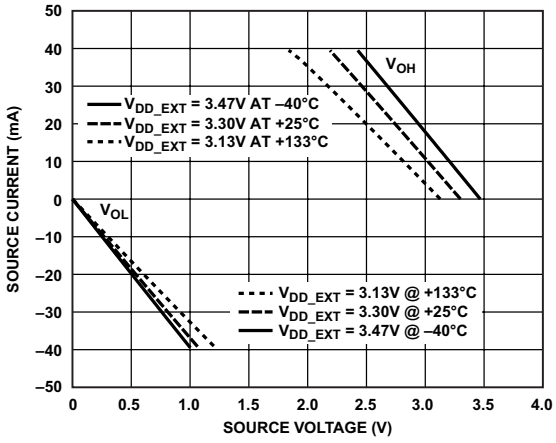


Figure 70. Driver Type H Current (3.3 V V_{DD_EXT})

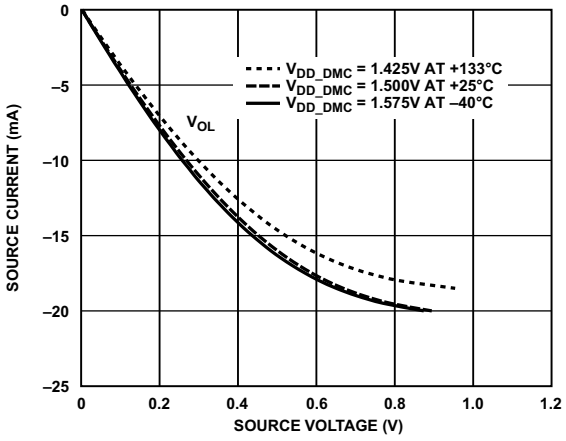


Figure 71. Driver Type B and Driver Type C (DDR3 Drive Strength 40 Ω)

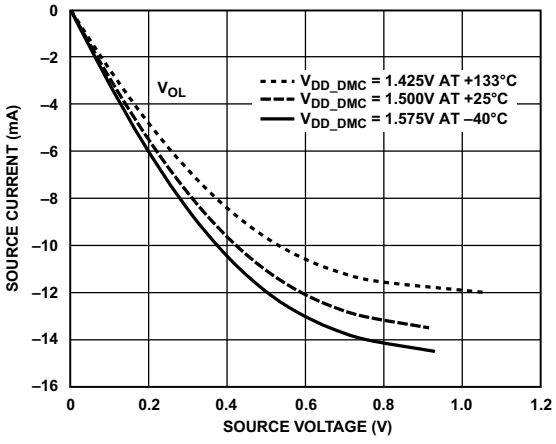


Figure 72. Driver Type B and Driver Type C (DDR3 Drive Strength 60 Ω)

Downloaded from Arrow.com.

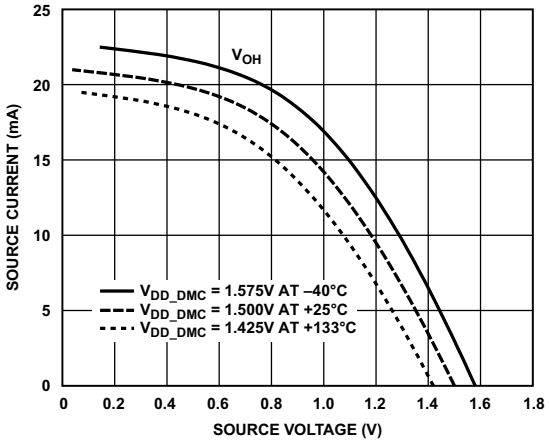


Figure 73. Driver Type B and Driver Type C (DDR3 Drive Strength 40 Ω)

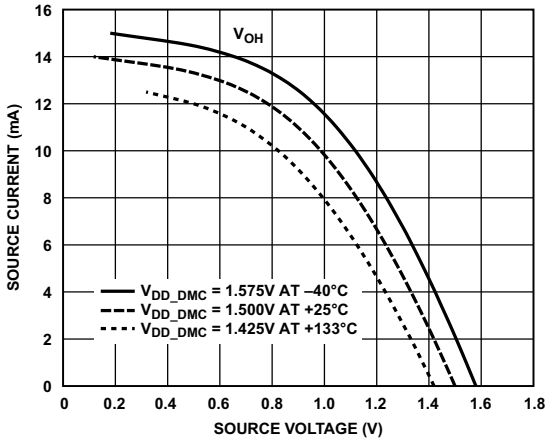


Figure 74. Driver Type B and Driver Type C (DDR3 Drive Strength 60 Ω)

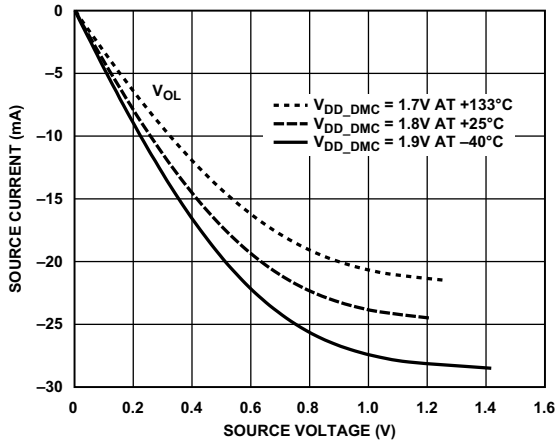


Figure 75. Driver Type B and Driver Type C (DDR2 Drive Strength 40 Ω)

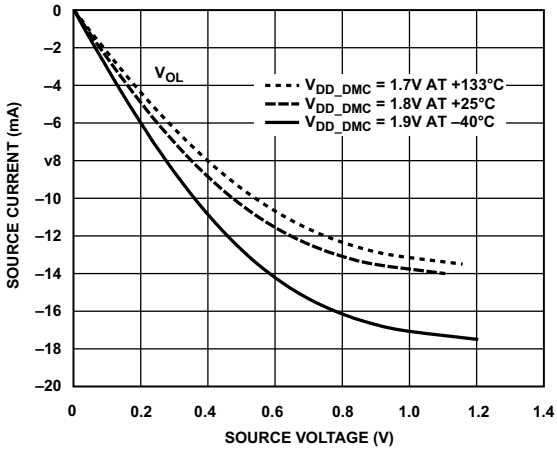


Figure 76. Driver Type B and Driver Type C (DDR2 Drive Strength 60 Ω)

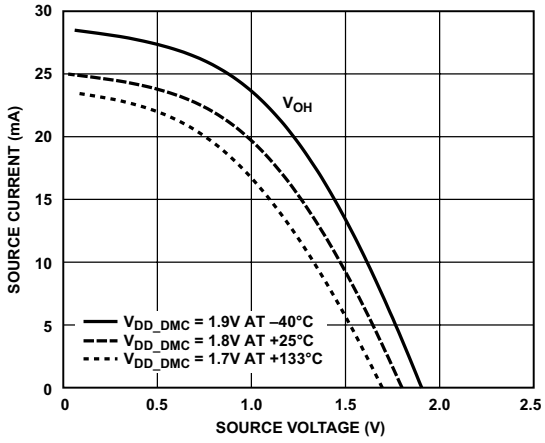


Figure 77. Driver Type B and Driver Type C (DDR2 Drive Strength 40 Ω)

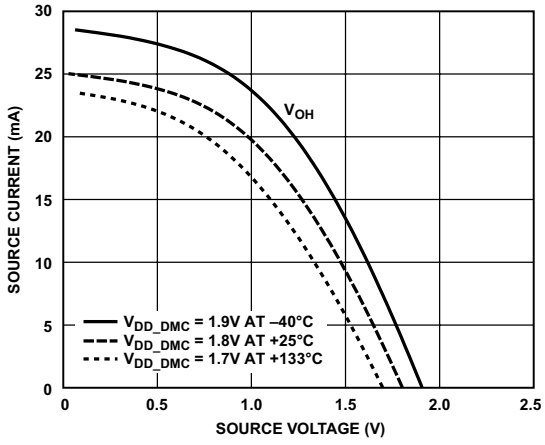


Figure 78. Driver Type B and Driver Type C (DDR2 Drive Strength 60 Ω)

Downloaded from Arrow.com.

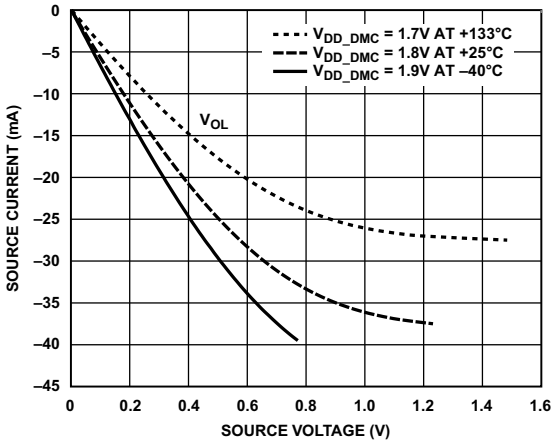


Figure 79. Driver Type B and Device Driver C (LPDDR)

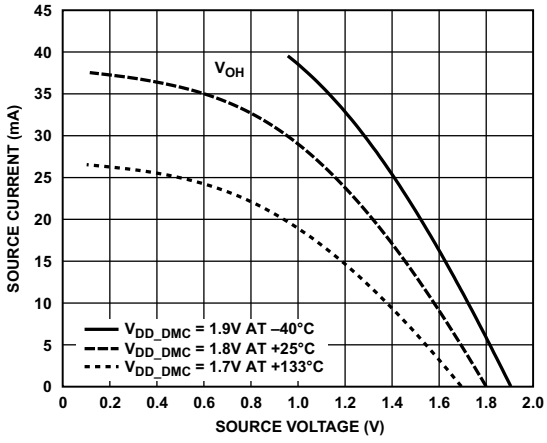


Figure 80. Driver Type B and Device Driver C (LPDDR)

TEST CONDITIONS

All timing parameters appearing in this data sheet were measured under the conditions described in this section. Figure 81 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) = 3.3 V.

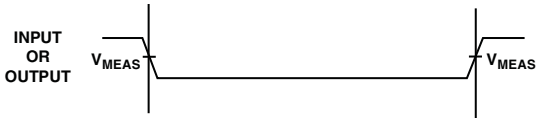


Figure 81. Voltage Reference Levels for AC Measurements (Except Output Enable/Disable)

Output Enable Time Measurement

Output pins are considered enabled when they make 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 82. If multiple pins are enabled, the measurement value is that of the first pin to start driving.

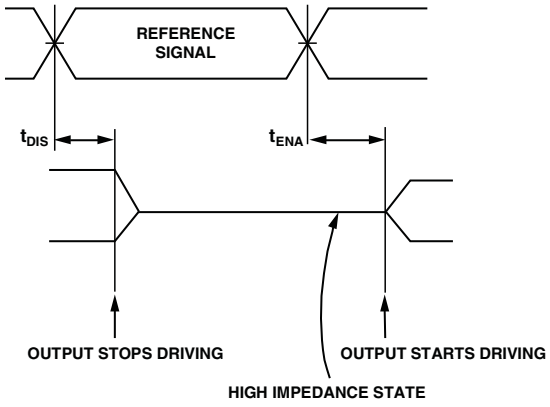


Figure 82. Output Enable/Disable

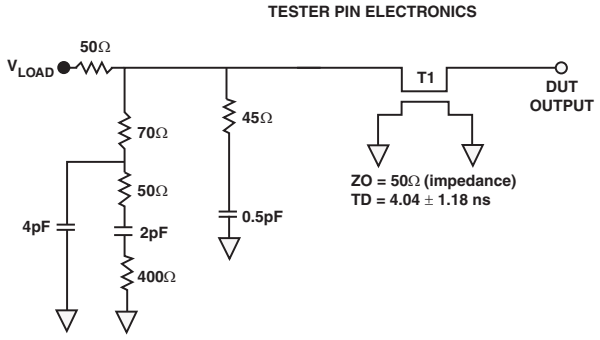
Output Disable Time Measurement

Output pins are considered disabled when they stop driving, enter a high impedance state, and start to decay from the output high or low voltage. The output disable time, t_{DIS}, is the interval from when a reference signal reaches a high or low voltage level to the point when the output stops driving, as shown on the left side of Figure 82).

Capacitive Loading

Output delays and holds are based on standard capacitive loads of an average of 6 pF on all pins (see Figure 83). V_{LOAD} is equal to $V_{DD_EXT}/2$. Figure 84 through Figure 88 show how output rise time varies with capacitance. The delay and hold specifications given must be derated by a factor derived from these figures. The graphs in Figure 84 through Figure 88 cannot be linear outside the ranges shown.

Downloaded from Arrow.com.



NOTES:
THE WORST CASE TRANSMISSION LINE DELAY IS SHOWN AND CAN BE USED FOR THE OUTPUT TIMING ANALYSIS TO REFLECT 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, THE SYSTEM CAN INCORPORATE EXTERNAL DRIVERS TO COMPENSATE FOR ANY TIMING DIFFERENCES.

Figure 83. Equivalent Device Loading for AC Measurements (Includes All Fixtures)

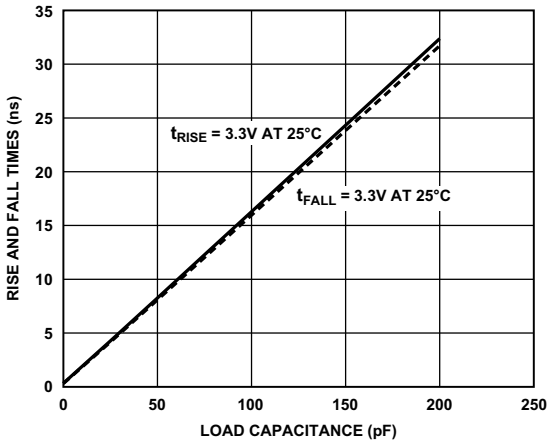


Figure 84. Driver Type A Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance (V_{DD_EXT} = 3.3 V)

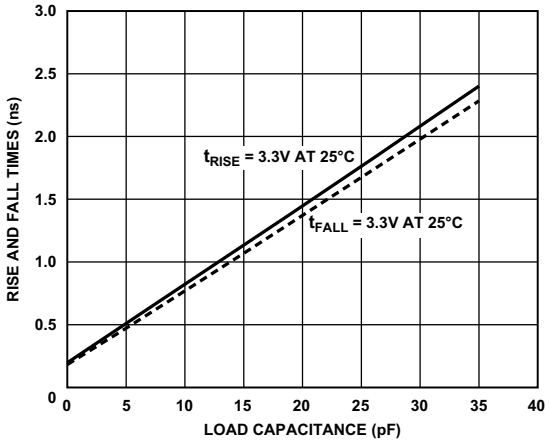


Figure 85. Driver Type H Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance (V_{DD_EXT} = 3.3 V)

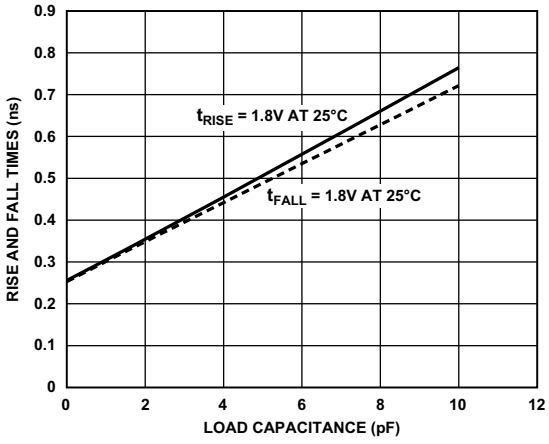


Figure 86. Driver Type B and Driver Type C Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance (V_{DD_DMC} = 1.8 V) for LPDDR

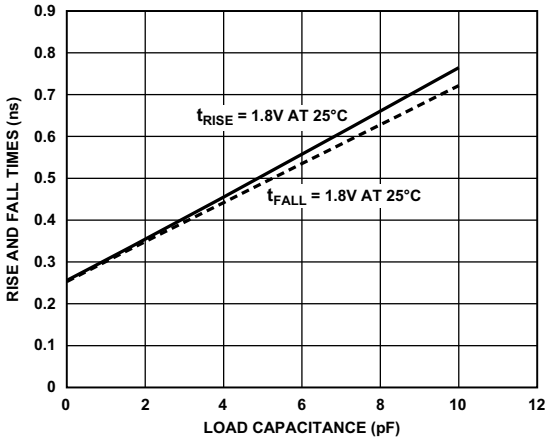


Figure 87. Driver Type B and Driver Type C Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance (V_{DD_DMC} = 1.8 V) for DDR2

Downloaded from Arrow.com.

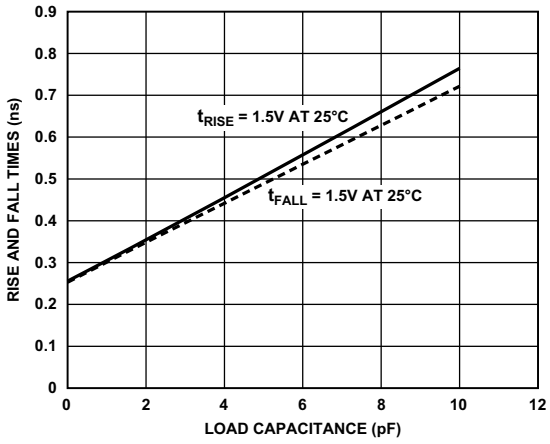


Figure 88. Driver Type B and Driver Type C Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance (V_{DD_DMC} = 1.5 V) for DDR3

ENVIRONMENTAL CONDITIONS

To determine the junction temperature on the application PCB, 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 at the top center of the package.
Ψ_{JT} = from Table 96 and Table 97.
P_D = power dissipation (see the Total Internal Power Dissipation section for the method to calculate P_D).

Values of θ_{JA} are provided for package comparison and PCB design considerations. θ_{JA} can be used for a first order approximation of T_J by the following equation:

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

where T_A = ambient temperature (°C).

Values of θ_{JC} are provided for package comparison and PCB design considerations when an external heat sink is required.

In Table 96 and Table 97, 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 6 layer PCB with 101.6 mm × 152.4 mm dimensions.

Table 96. Thermal Characteristics for 400 CSP_BGA

Parameter	Conditions	Typical	Unit
θ_{JA}	0 linear m/s air flow	14.24	°C/W
θ_{JA}	1 linear m/s air flow	12.61	°C/W
θ_{JA}	2 linear m/s air flow	12.09	°C/W
θ_{JC}		5.71	°C/W
Ψ_{JT}	0 linear m/s air flow	0.08	°C/W
Ψ_{JT}	1 linear m/s air flow	0.14	°C/W
Ψ_{JT}	2 linear m/s air flow	0.17	°C/W

Table 97. Thermal Characteristics for 176 LQFP_EP

Parameter	Conditions	Typical	Unit
θ_{JA}	0 linear m/s air flow	11.95	°C/W
θ_{JA}	1 linear m/s air flow	10.43	°C/W
θ_{JA}	2 linear m/s air flow	9.98	°C/W
θ_{JC}		11.10	°C/W
Ψ_{JT}	0 linear m/s air flow	0.15	°C/W
Ψ_{JT}	1 linear m/s air flow	0.24	°C/W
Ψ_{JT}	2 linear m/s air flow	0.29	°C/W

Downloaded from Arrow.com.

ADSP-SC57x/ADSP-2157x 400-BALL BGA BALL ASSIGNMENTS

The ADSP-SC57x/ADSP-2157x 400-Ball BGA Ball Assignments (Numerical by Ball Number) table lists the 400-ball BGA package by ball number.

The ADSP-SC57x/ADSP-2157x 400-Ball BGA Ball Assignments (Alphabetical by Pin Name) table lists the 400-ball BGA package by pin name.

ADSP-SC57x/ADSP-2157x 400-BALL BGA BALL ASSIGNMENTS (NUMERICAL BY BALL NUMBER)

Ball No.	Pin Name	Ball No.	Pin Name	Ball No.	Pin Name	Ball No.	Pin Name
A01	GND	C02	PC_13	E03	PE_03	G04	VDD_EXT
A02	PA_10	C03	GND	E04	PE_02	G05	VDD_INT
A03	PA_09	C04	PA_12	E05	GND	G06	GND
A04	PA_11	C05	PA_14	E06	PB_00	G07	GND
A05	PE_07	C06	PB_03	E07	VDD_EXT	G08	GND
A06	MLB0_CLKN	C07	PB_02	E08	VDD_EXT	G09	GND
A07	MLB0_CLKP	C08	PE_10	E09	VDD_EXT	G10	GND
A08	MLB0_SIGN	C09	PB_06	E10	VDD_EXT	G11	GND
A09	GND	C10	PB_05	E11	VDD_EXT	G12	GND
A10	SYS_XTAL0	C11	$\overline{\text{SYS_HWRST}}$	E12	VDD_EXT	G13	GND
A11	SYS_CLKIN0	C12	USB0_ID	E13	VDD_USB	G14	GND
A12	GND	C13	USB0_CLKIN	E14	JTG_TCK	G15	GND
A13	SYS_XTAL1	C14	PB_12	E15	PE_15	G16	VDD_INT
A14	SYS_CLKIN1	C15	PB_13	E16	GND	G17	PB_15
A15	GND	C16	JTG_TDI	E17	VDD_EXT	G18	DAI0_PIN08
A16	USB0_DP	C17	PE_14	E18	PF_04	G19	DAI0_PIN10
A17	USB0_DM	C18	GND	E19	DAI0_PIN07	G20	DAI0_PIN09
A18	PF_03	C19	PF_08	E20	DAI0_PIN03	H01	PE_01
A19	PF_05	C20	PF_11	F01	PC_02	H02	PC_09
A20	GND	D01	PC_06	F02	PC_03	H03	PC_15
B01	PC_12	D02	PC_08	F03	PC_04	H04	VDD_EXT
B02	GND	D03	PE_04	F04	PE_06	H05	VDD_INT
B03	PA_13	D04	GND	F05	VDD_INT	H06	GND
B04	PA_15	D05	PE_08	F06	GND	H07	GND
B05	PB_01	D06	PE_11	F07	VDD_INT	H08	GND
B06	PB_04	D07	PE_09	F08	VDD_INT	H09	GND
B07	MLB0_DATN	D08	PB_08	F09	VDD_INT	H10	GND
B08	MLB0_DATP	D09	PB_07	F10	VDD_INT	H11	GND
B09	MLB0_SIGP	D10	PB_09	F11	VDD_INT	H12	GND
B10	$\overline{\text{JTG_TRST}}$	D11	SYS_CLKOUT	F12	VDD_INT	H13	GND
B11	USB0_VBUS	D12	PB_11	F13	VDD_INT	H14	GND
B12	USB0_XTAL	D13	USB0_VBC	F14	VDD_INT	H15	GND
B13	PB_10	D14	PB_14	F15	GND	H16	VDD_INT
B14	JTG_TDO	D15	PE_13	F16	VDD_INT	H17	VDD_EXT
B15	JTG_TMS	D16	PE_12	F17	PF_02	H18	DAI0_PIN05
B16	PF_00	D17	GND	F18	PF_09	H19	DAI0_PIN14
B17	PF_01	D18	PF_10	F19	DAI0_PIN02	H20	DAI0_PIN11
B18	PF_06	D19	DAI0_PIN01	F20	DAI0_PIN06	J01	PE_00
B19	GND	D20	DAI0_PIN04	G01	PC_00	J02	PC_07
B20	PF_07	E01	PC_05	G02	PC_14	J03	PC_10
C01	PC_11	E02	PE_05	G03	PC_01	J04	VDD_EXT

Downloaded from Arrow.com.

Ball No.	Pin Name	Ball No.	Pin Name	Ball No.	Pin Name	Ball No.	Pin Name
J05	VDD_INT	L13	GND	P01	DMC0_VREF	T09	VDD_DMC
J06	GND	L14	GND	P02	DMC0_RZQ	T10	VDD_DMC
J07	GND	L15	GND	P03	DMC0_A09	T11	VDD_DMC
J08	GND	L16	VDD_INT	P04	DMC0_A10	T12	VDD_DMC
J09	GND	L17	VDD_EXT	P05	VDD_INT	T13	VDD_DMC
J10	GND	L18	DAI0_PIN17	P06	VDD_INT	T14	VDD_DMC
J11	GND	L19	GND	P07	GND	T15	VDD_DMC
J12	GND	L20	DAI0_PIN20	P08	VDD_INT	T16	GND
J13	GND	M01	DMC0_A14	P09	VDD_INT	T17	VDD_DMC
J14	GND	M02	DMC0_A15	P10	VDD_INT	T18	DMC0_A02
J15	GND	M03	PD_09	P11	VDD_INT	T19	DMC0_A01
J16	VDD_INT	M04	VDD_EXT	P12	VDD_INT	T20	$\overline{\text{DMC0_RESET}}$
J17	VDD_EXT	M05	VDD_INT	P13	VDD_INT	U01	DMC0_DQ15
J18	DAI0_PIN12	M06	GND	P14	GND	U02	DMC0_DQ14
J19	DAI0_PIN13	M07	GND	P15	VDD_INT	U03	TWI1_SDA
J20	DAI0_PIN16	M08	GND	P16	VDD_INT	U04	GND
K01	PD_14	M09	GND	P17	HADC0_VIN1	U05	TWI1_SCL
K02	PD_13	M10	GND	P18	HADC0_VIN0	U06	VDD_EXT
K03	PD_15	M11	GND	P19	HADC0_VREFP	U07	VDD_DMC
K04	VDD_EXT	M12	GND	P20	HADC0_VREFN	U08	VDD_DMC
K05	VDD_INT	M13	GND	R01	PD_06	U09	VDD_DMC
K06	GND	M14	GND	R02	PD_07	U10	VDD_DMC
K07	GND	M15	GND	R03	PD_08	U11	VDD_DMC
K08	GND	M16	VDD_INT	R04	VDD_INT	U12	VDD_DMC
K09	GND	M17	VDD_EXT	R05	VDD_INT	U13	VDD_DMC
K10	GND	M18	HADC0_VIN7	R06	GND	U14	VDD_DMC
K11	GND	M19	HADC0_VIN5	R07	VDD_INT	U15	DMC0_BA2
K12	GND	M20	HADC0_VIN6	R08	VDD_INT	U16	$\overline{\text{DMC0_WE}}$
K13	GND	N01	DMC0_A11	R09	VDD_INT	U17	GND
K14	GND	N02	DMC0_A13	R10	VDD_INT	U18	DMC0_A06
K15	GND	N03	DMC0_A12	R11	VDD_INT	U19	DMC0_A03
K16	VDD_INT	N04	VDD_DMC	R12	VDD_INT	U20	DMC0_A00
K17	VDD_EXT	N05	VDD_INT	R13	VDD_INT	V01	TWI2_SDA
K18	DAI0_PIN15	N06	GND	R14	VDD_INT	V02	DMC0_DQ13
K19	DAI0_PIN19	N07	GND	R15	GND	V03	GND
K20	DAI0_PIN18	N08	GND	R16	VDD_INT	V04	PD_03
L01	PD_11	N09	GND	R17	VDD_INT	V05	PD_04
L02	PD_10	N10	GND	R18	DMC0_BA0	V06	PD_01
L03	PD_12	N11	GND	R19	HADC0_VIN2	V07	PA_08
L04	VDD_EXT	N12	GND	R20	VDD_HADC	V08	PA_05
L05	VDD_INT	N13	GND	T01	PD_05	V09	PA_03
L06	GND	N14	GND	T02	TWI0_SDA	V10	PA_02
L07	GND	N15	GND	T03	TWI0_SCL	V11	PA_01
L08	GND	N16	VDD_INT	T04	VDD_EXT	V12	PA_00
L09	GND	N17	VDD_EXT	T05	GND	V13	$\overline{\text{SYS_RESOUT}}$
L10	GND	N18	HADC0_VIN4	T06	VDD_DMC	V14	SYS_FAULT
L11	GND	N19	HADC0_VIN3	T07	VDD_DMC	V15	$\overline{\text{DMC0_CAS}}$
L12	GND	N20	GND	T08	VDD_DMC	V16	$\overline{\text{DMC0_RAS}}$

Downloaded from Arrow.com.

Ball No.	Pin Name
V17	DMC0_BA1
V18	GND
V19	DMC0_A04
V20	DMC0_A05
W01	TWI2_SCL
W02	GND
W03	DMC0_DQ12
W04	DMC0_DQ11
W05	DMC0_DQ09
W06	PD_02
W07	PD_00
W08	PA_07
W09	PA_06
W10	PA_04
W11	DMC0_DQ05
W12	DMC0_DQ04
W13	DMC0_DQ03
W14	DMC0_DQ02
W15	$\overline{\text{SYS_FAULT}}$
W16	DMC0_ODT
W17	DMC0_A08
W18	SYS_BMODE1
W19	GND
W20	DMC0_A07
Y01	GND
Y02	$\overline{\text{DMC0_UDQS}}$
Y03	DMC0_UDQS
Y04	DMC0_DQ10
Y05	DMC0_DQ08
Y06	DMC0_UDM
Y07	DMC0_LDM
Y08	DMC0_CK
Y09	$\overline{\text{DMC0_CK}}$
Y10	DMC0_DQ07
Y11	DMC0_DQ06
Y12	DMC0_LDQS
Y13	$\overline{\text{DMC0_LDQS}}$
Y14	DMC0_DQ01
Y15	DMC0_DQ00
Y16	DMC0_CKE
Y17	$\overline{\text{DMC0_CS0}}$
Y18	SYS_BMODE0
Y19	SYS_BMODE2
Y20	GND

Downloaded from Arrow.com.

ADSP-SC57x/ADSP-2157x 400-BALL BGA BALL ASSIGNMENTS (ALPHABETICAL BY PIN NAME)

Pin Name	Ball No.	Pin Name	Ball No.	Pin Name	Ball No.	Pin Name	Ball No.
DAI0_PIN01	D19	DMC0_DQ02	W14	GND	G11	GND	M06
DAI0_PIN02	F19	DMC0_DQ03	W13	GND	G12	GND	M07
DAI0_PIN03	E20	DMC0_DQ04	W12	GND	G13	GND	M08
DAI0_PIN04	D20	DMC0_DQ05	W11	GND	G14	GND	M09
DAI0_PIN05	H18	DMC0_DQ06	Y11	GND	G15	GND	M10
DAI0_PIN06	F20	DMC0_DQ07	Y10	GND	H06	GND	M11
DAI0_PIN07	E19	DMC0_DQ08	Y05	GND	H07	GND	M12
DAI0_PIN08	G18	DMC0_DQ09	W05	GND	H08	GND	M13
DAI0_PIN09	G20	DMC0_DQ10	Y04	GND	H09	GND	M14
DAI0_PIN10	G19	DMC0_DQ11	W04	GND	H10	GND	M15
DAI0_PIN11	H20	DMC0_DQ12	W03	GND	H11	GND	N06
DAI0_PIN12	J18	DMC0_DQ13	V02	GND	H12	GND	N07
DAI0_PIN13	J19	DMC0_DQ14	U02	GND	H13	GND	N08
DAI0_PIN14	H19	DMC0_DQ15	U01	GND	H14	GND	N09
DAI0_PIN15	K18	DMC0_LDM	Y07	GND	H15	GND	N10
DAI0_PIN16	J20	DMC0_LDQS	Y12	GND	J06	GND	N11
DAI0_PIN17	L18	$\overline{\text{DMC0_LDQS}}$	Y13	GND	J07	GND	N12
DAI0_PIN18	K20	DMC0_ODT	W16	GND	J08	GND	N13
DAI0_PIN19	K19	$\overline{\text{DMC0_RAS}}$	V16	GND	J09	GND	N14
DAI0_PIN20	L20	$\overline{\text{DMC0_RESET}}$	T20	GND	J10	GND	N15
DMC0_A00	U20	DMC0_RZQ	P02	GND	J11	GND	N20
DMC0_A01	T19	DMC0_UDM	Y06	GND	J12	GND	P07
DMC0_A02	T18	DMC0_UDQS	Y03	GND	J13	GND	P14
DMC0_A03	U19	$\overline{\text{DMC0_UDQS}}$	Y02	GND	J14	GND	R06
DMC0_A04	V19	DMC0_VREF	P01	GND	J15	GND	R15
DMC0_A05	V20	$\overline{\text{DMC0_WE}}$	U16	GND	K06	GND	T05
DMC0_A06	U18	GND	A01	GND	K07	GND	T16
DMC0_A07	W20	GND	A09	GND	K08	GND	U04
DMC0_A08	W17	GND	A12	GND	K09	GND	U17
DMC0_A09	P03	GND	A15	GND	K10	GND	V03
DMC0_A10	P04	GND	A20	GND	K11	GND	V18
DMC0_A11	N01	GND	B02	GND	K12	GND	W02
DMC0_A12	N03	GND	B19	GND	K13	GND	W19
DMC0_A13	N02	GND	C03	GND	K14	GND	Y01
DMC0_A14	M01	GND	C18	GND	K15	GND	Y20
DMC0_A15	M02	GND	D04	GND	L06	HADC0_VIN0	P18
DMC0_BA0	R18	GND	D17	GND	L07	HADC0_VIN1	P17
DMC0_BA1	V17	GND	E05	GND	L08	HADC0_VIN2	R19
DMC0_BA2	U15	GND	E16	GND	L09	HADC0_VIN3	N19
$\overline{\text{DMC0_CAS}}$	V15	GND	F06	GND	L10	HADC0_VIN4	N18
DMC0_CK	Y08	GND	F15	GND	L11	HADC0_VIN5	M19
DMC0_CKE	Y16	GND	G06	GND	L12	HADC0_VIN6	M20
$\overline{\text{DMC0_CK}}$	Y09	GND	G07	GND	L13	HADC0_VIN7	M18
$\overline{\text{DMC0_CS0}}$	Y17	GND	G08	GND	L14	HADC0_VREFN	P20
DMC0_DQ00	Y15	GND	G09	GND	L15	HADC0_VREFP	P19
DMC0_DQ01	Y14	GND	G10	GND	L19	JTG_TCK	E14

Downloaded from Arrow.com.

Pin Name	Ball No.	Pin Name	Ball No.	Pin Name	Ball No.	Pin Name	Ball No.
JTG_TDI	C16	PC_05	E01	PF_04	E18	VDD_DMC	U09
JTG_TDO	B14	PC_06	D01	PF_05	A19	VDD_DMC	U10
JTG_TMS	B15	PC_07	J02	PF_06	B18	VDD_DMC	U11
$\overline{\text{JTG_TRST}}$	B10	PC_08	D02	PF_07	B20	VDD_DMC	U12
MLB0_CLKN	A06	PC_09	H02	PF_08	C19	VDD_DMC	U13
MLB0_CLKP	A07	PC_10	J03	PF_09	F18	VDD_DMC	U14
MLB0_DATN	B07	PC_11	C01	PF_10	D18	VDD_EXT	E07
MLB0_DATP	B08	PC_12	B01	PF_11	C20	VDD_EXT	E08
MLB0_SIGN	A08	PC_13	C02	SYS_BMODE0	Y18	VDD_EXT	E09
MLB0_SIGP	B09	PC_14	G02	SYS_BMODE1	W18	VDD_EXT	E10
PA_00	V12	PC_15	H03	SYS_BMODE2	Y19	VDD_EXT	E11
PA_01	V11	PD_00	W07	SYS_CLKIN0	A11	VDD_EXT	E12
PA_02	V10	PD_01	V06	SYS_CLKIN1	A14	VDD_EXT	E17
PA_03	V09	PD_02	W06	SYS_CLKOUT	D11	VDD_EXT	G04
PA_04	W10	PD_03	V04	SYS_FAULT	V14	VDD_EXT	H04
PA_05	V08	PD_04	V05	$\overline{\text{SYS_FAULT}}$	W15	VDD_EXT	H17
PA_06	W09	PD_05	T01	$\overline{\text{SYS_HWRST}}$	C11	VDD_EXT	J04
PA_07	W08	PD_06	R01	$\overline{\text{SYS_RESOUT}}$	V13	VDD_EXT	J17
PA_08	V07	PD_07	R02	SYS_XTAL0	A10	VDD_EXT	K04
PA_09	A03	PD_08	R03	SYS_XTAL1	A13	VDD_EXT	K17
PA_10	A02	PD_09	M03	TWI0_SCL	T03	VDD_EXT	L04
PA_11	A04	PD_10	L02	TWI0_SDA	T02	VDD_EXT	L17
PA_12	C04	PD_11	L01	TWI1_SCL	U05	VDD_EXT	M04
PA_13	B03	PD_12	L03	TWI1_SDA	U03	VDD_EXT	M17
PA_14	C05	PD_13	K02	TWI2_SCL	W01	VDD_EXT	N17
PA_15	B04	PD_14	K01	TWI2_SDA	V01	VDD_EXT	T04
PB_00	E06	PD_15	K03	USB0_DM	A17	VDD_EXT	U06
PB_01	B05	PE_00	J01	USB0_DP	A16	VDD_HADC	R20
PB_02	C07	PE_01	H01	USB0_ID	C12	VDD_INT	F05
PB_03	C06	PE_02	E04	USB0_VBC	D13	VDD_INT	F07
PB_04	B06	PE_03	E03	USB0_VBUS	B11	VDD_INT	F08
PB_05	C10	PE_04	D03	USB0_CLKIN	C13	VDD_INT	F09
PB_06	C09	PE_05	E02	USB0_XTAL	B12	VDD_INT	F10
PB_07	D09	PE_06	F04	VDD_DMC	N04	VDD_INT	F11
PB_08	D08	PE_07	A05	VDD_DMC	T06	VDD_INT	F12
PB_09	D10	PE_08	D05	VDD_DMC	T07	VDD_INT	F13
PB_10	B13	PE_09	D07	VDD_DMC	T08	VDD_INT	F14
PB_11	D12	PE_10	C08	VDD_DMC	T09	VDD_INT	F16
PB_12	C14	PE_11	D06	VDD_DMC	T10	VDD_INT	G05
PB_13	C15	PE_12	D16	VDD_DMC	T11	VDD_INT	G16
PB_14	D14	PE_13	D15	VDD_DMC	T12	VDD_INT	H05
PB_15	G17	PE_14	C17	VDD_DMC	T13	VDD_INT	H16
PC_00	G01	PE_15	E15	VDD_DMC	T14	VDD_INT	J05
PC_01	G03	PF_00	B16	VDD_DMC	T15	VDD_INT	J16
PC_02	F01	PF_01	B17	VDD_DMC	T17	VDD_INT	K05
PC_03	F02	PF_02	F17	VDD_DMC	U07	VDD_INT	K16
PC_04	F03	PF_03	A18	VDD_DMC	U08	VDD_INT	L05

Downloaded from Arrow.com.

Pin Name	Ball No.
VDD_INT	L16
VDD_INT	M05
VDD_INT	M16
VDD_INT	N05
VDD_INT	N16
VDD_INT	P05
VDD_INT	P06
VDD_INT	P08
VDD_INT	P09
VDD_INT	P10
VDD_INT	P11
VDD_INT	P12
VDD_INT	P13
VDD_INT	P15
VDD_INT	P16
VDD_INT	R04
VDD_INT	R05
VDD_INT	R07
VDD_INT	R08
VDD_INT	R09
VDD_INT	R10
VDD_INT	R11
VDD_INT	R12
VDD_INT	R13
VDD_INT	R14
VDD_INT	R16
VDD_INT	R17
VDD_USB	E13

Downloaded from Arrow.com.

CONFIGURATION OF THE 400-BALL CSP_BGA

Figure 89 shows an overview of signal placement on the 400-ball CSP_BGA.

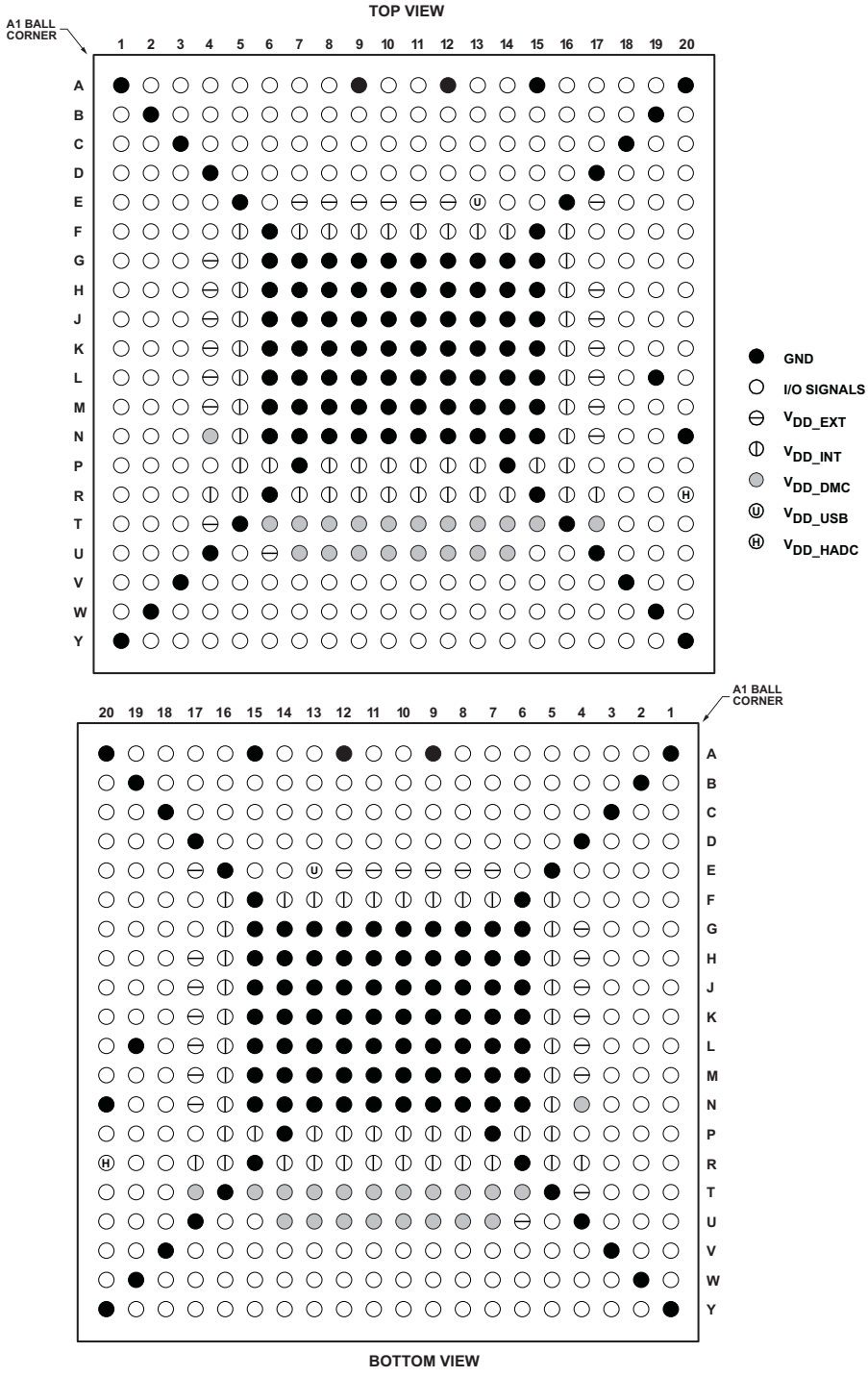


Figure 89. 400-Ball CSP_BGA Configuration

Downloaded from Arrow.com.

ADSP-SC57x/ADSP-2157x 176-LEAD LQFP LEAD ASSIGNMENTS

The ADSP-SC57x/ADSP-2157x 176-Lead LQFP Lead Assignments (Numerical by Lead Number) table lists the 176-lead LQFP package by lead number.

The ADSP-SC57x/ADSP-2157x 176-Lead LQFP Lead Assignments (Alphabetical by Pin Name) table lists the 176-lead LQFP package by pin name.

ADSP-SC57x/ADSP-2157x 176-LEAD LQFP LEAD ASSIGNMENTS (NUMERICAL BY LEAD NUMBER)

Lead No.	Pin Name	Lead No.	Pin Name	Lead No.	Pin Name	Lead No.	Pin Name
01	VDD_INT	41	PD_11	81	$\overline{\text{SYS_RESOUT}}$	121	DAI0_PIN03
02	GND	42	PD_10	82	VDD_INT	122	DAI0_PIN04
03	VDD_INT	43	PD_09	83	GND	123	DAI0_PIN01
04	PA_15	44	GND	84	VDD_EXT	124	VDD_INT
05	PA_14	45	GND	85	$\overline{\text{SYS_FAULT}}$	125	GND
06	PA_13	46	VDD_EXT	86	SYS_BMODE0	126	VDD_EXT
07	VDD_INT	47	VDD_INT	87	SYS_BMODE1	127	DAI0_PIN02
08	PA_12	48	PD_08	88	VDD_INT	128	PB_15
09	VDD_EXT	49	PD_07	89	GND	129	VDD_INT
10	PA_10	50	PD_06	90	VDD_HADC	130	VDD_INT
11	PA_11	51	PD_05	91	HADC0_VIN0	131	GND
12	PC_15	52	VDD_INT	92	HADC0_VIN1	132	VDD_INT
13	PA_09	53	TWI0_SDA	93	HADC0_VREFN	133	GND
14	VDD_INT	54	TWI0_SCL	94	HADC0_VIN2	134	VDD_INT
15	GND	55	TWI1_SDA	95	HADC0_VIN3	135	JTG_TCK
16	VDD_INT	56	TWI1_SCL	96	HADC0_VREFP	136	JTG_TDO
17	PC_14	57	TWI2_SDA	97	GND	137	JTG_TDI
18	PC_13	58	TWI2_SCL	98	VDD_INT	138	JTG_TMS
19	PC_12	59	VDD_INT	99	GND	139	VDD_INT
20	PC_11	60	VDD_EXT	100	DAI0_PIN20	140	VDD_EXT
21	VDD_EXT	61	PD_04	101	DAI0_PIN19	141	PB_14
22	PC_10	62	PD_03	102	DAI0_PIN18	142	PB_13
23	PC_09	63	PD_02	103	VDD_INT	143	VDD_EXT
24	PC_08	64	PD_01	104	VDD_EXT	144	PB_12
25	PC_07	65	GND	105	DAI0_PIN17	145	VDD_INT
26	PC_06	66	VDD_INT	106	DAI0_PIN16	146	PB_11
27	PC_05	67	PD_00	107	DAI0_PIN15	147	VDD_EXT
28	PC_04	68	PA_08	108	DAI0_PIN14	148	PB_10
29	PC_03	69	PA_07	109	VDD_INT	149	VDD_EXT
30	VDD_INT	70	PA_06	110	DAI0_PIN13	150	VDD_INT
31	VDD_EXT	71	VDD_EXT	111	DAI0_PIN12	151	$\overline{\text{SYS_HWRST}}$
32	PC_02	72	VDD_INT	112	DAI0_PIN11	152	VDD_EXT
33	PC_01	73	VDD_INT	113	DAI0_PIN10	153	$\overline{\text{JTG_TRST}}$
34	PC_00	74	PA_05	114	VDD_INT	154	SYS_CLKIN0
35	PD_15	75	PA_04	115	VDD_EXT	155	SYS_XTAL0
36	PD_14	76	PA_03	116	DAI0_PIN09	156	VDD_INT
37	PD_13	77	PA_02	117	DAI0_PIN08	157	SYS_CLKOUT
38	VDD_EXT	78	VDD_EXT	118	DAI0_PIN06	158	VDD_EXT
39	VDD_INT	79	PA_01	119	DAI0_PIN07	159	PB_09
40	PD_12	80	PA_00	120	DAI0_PIN05	160	VDD_EXT

Downloaded from Arrow.com.

Lead No.	Pin Name
161	PB_08
162	PB_07
163	VDD_INT
164	VDD_EXT
165	PB_06
166	PB_05
167	VDD_EXT
168	PB_04
169	PB_03
170	VDD_INT
171	VDD_EXT
172	PB_02
173	PB_01
174	PB_00
175	VDD_INT
176	GND
177[1]	GND

[1] Pin177 is the GND supply (see Figure 91) for the processor; this pad must connect to GND.

Downloaded from Arrow.com.

ADSP-SC57X/ADSP-2157X 176-LEAD LQFP LEAD ASSIGNMENTS (ALPHABETICAL BY PIN NAME)

Pin Name	Lead No.	Pin Name	Lead No.	Pin Name	Lead No.	Pin Name	Lead No.
DAI0_PIN01	123	PA_01	79	PC_15	12	VDD_EXT	149
DAI0_PIN02	127	PA_02	77	PD_00	67	VDD_EXT	152
DAI0_PIN03	121	PA_03	76	PD_01	64	VDD_EXT	158
DAI0_PIN04	122	PA_04	75	PD_02	63	VDD_EXT	160
DAI0_PIN05	120	PA_05	74	PD_03	62	VDD_EXT	164
DAI0_PIN06	118	PA_06	70	PD_04	61	VDD_EXT	167
DAI0_PIN07	119	PA_07	69	PD_05	51	VDD_EXT	171
DAI0_PIN08	117	PA_08	68	PD_06	50	VDD_HADC	90
DAI0_PIN09	116	PA_09	13	PD_07	49	VDD_INT	01
DAI0_PIN10	113	PA_10	10	PD_08	48	VDD_INT	03
DAI0_PIN11	112	PA_11	11	PD_09	43	VDD_INT	07
DAI0_PIN12	111	PA_12	08	PD_10	42	VDD_INT	14
DAI0_PIN13	110	PA_13	06	PD_11	41	VDD_INT	16
DAI0_PIN14	108	PA_14	05	PD_12	40	VDD_INT	30
DAI0_PIN15	107	PA_15	04	PD_13	37	VDD_INT	39
DAI0_PIN16	106	PB_00	174	PD_14	36	VDD_INT	47
DAI0_PIN17	105	PB_01	173	PD_15	35	VDD_INT	52
DAI0_PIN18	102	PB_02	172	SYS_BMODE0	86	VDD_INT	59
DAI0_PIN19	101	PB_03	169	SYS_BMODE1	87	VDD_INT	66
DAI0_PIN20	100	PB_04	168	SYS_CLKIN0	154	VDD_INT	72
GND	02	PB_05	166	SYS_CLKOUT	157	VDD_INT	73
GND	15	PB_06	165	$\overline{\text{SYS_FAULT}}$	85	VDD_INT	82
GND	44	PB_07	162	$\overline{\text{SYS_HWRST}}$	151	VDD_INT	88
GND	45	PB_08	161	$\overline{\text{SYS_RESOUT}}$	81	VDD_INT	98
GND	65	PB_09	159	SYS_XTAL0	155	VDD_INT	103
GND	83	PB_10	148	TWI0_SCL	54	VDD_INT	109
GND	89	PB_11	146	TWI0_SDA	53	VDD_INT	114
GND	97	PB_12	144	TWI1_SCL	56	VDD_INT	124
GND	99	PB_13	142	TWI1_SDA	55	VDD_INT	129
GND	125	PB_14	141	TWI2_SCL	58	VDD_INT	130
GND	131	PB_15	128	TWI2_SDA	57	VDD_INT	132
GND	133	PC_00	34	VDD_EXT	09	VDD_INT	134
GND	176	PC_01	33	VDD_EXT	21	VDD_INT	139
GND	177[1]	PC_02	32	VDD_EXT	31	VDD_INT	145
HADC0_VIN0	91	PC_03	29	VDD_EXT	38	VDD_INT	150
HADC0_VIN1	92	PC_04	28	VDD_EXT	46	VDD_INT	156
HADC0_VIN2	94	PC_05	27	VDD_EXT	60	VDD_INT	163
HADC0_VIN3	95	PC_06	26	VDD_EXT	71	VDD_INT	170
HADC0_VREFN	93	PC_07	25	VDD_EXT	78	VDD_INT	175
HADC0_VREFP	96	PC_08	24	VDD_EXT	84		
JTG_TCK	135	PC_09	23	VDD_EXT	104		
JTG_TDI	137	PC_10	22	VDD_EXT	115		
JTG_TDO	136	PC_11	20	VDD_EXT	126		
JTG_TMS	138	PC_12	19	VDD_EXT	140		
$\overline{\text{JTG_TRST}}$	153	PC_13	18	VDD_EXT	143		
PA_00	80	PC_14	17	VDD_EXT	147		

[1] Pin 177 is the GND supply (see Figure 91) for the processor; this pad must connect to GND.

Downloaded from Arrow.com.

CONFIGURATION OF THE 176-LEAD LQFP LEAD CONFIGURATION

Figure 90 shows the top view of the 176-lead LQFP lead configuration and Figure 91 shows the bottom view of the 176-lead LQFP lead configuration.

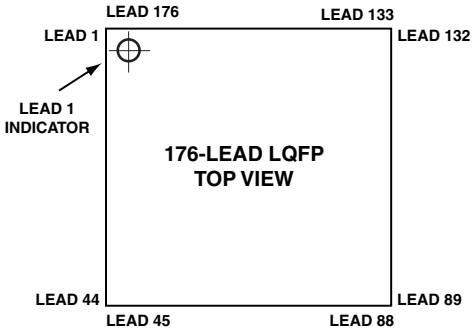


Figure 90. 176-Lead LQFP Lead Configuration (Top View)

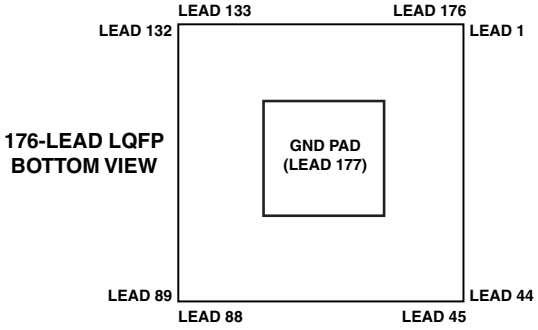


Figure 91. 176-Lead LQFP Lead Configuration (Bottom View)

Downloaded from Arrow.com.

OUTLINE DIMENSIONS

Dimensions in Figure 92 (for the 400-ball BGA) and Figure 93 (for the 176-lead LQFP) are shown in millimeters.

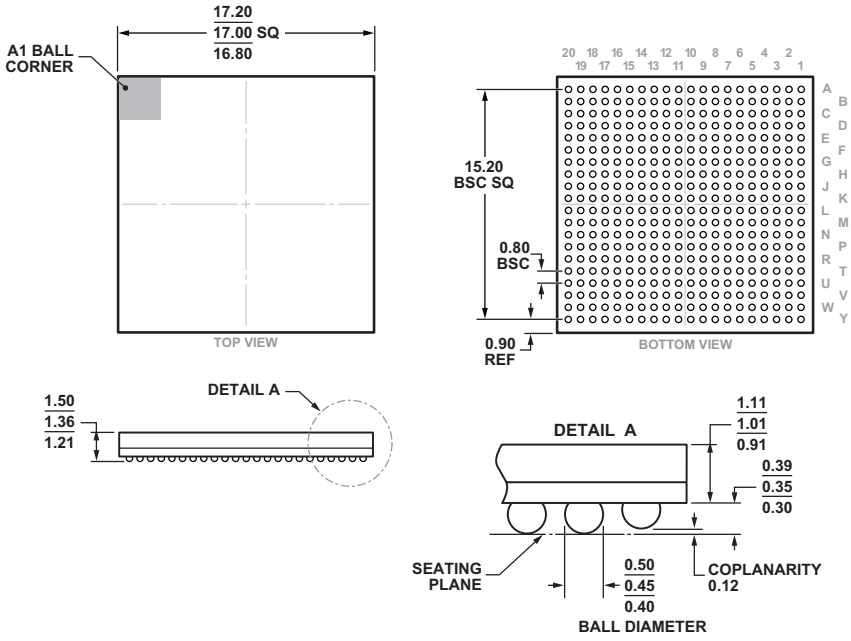


Figure 92. 400-Ball Chip Scale Package Ball Grid Array [CSP_BGA] (BC-400-2) Dimensions shown in millimeters

Downloaded from Arrow.com.

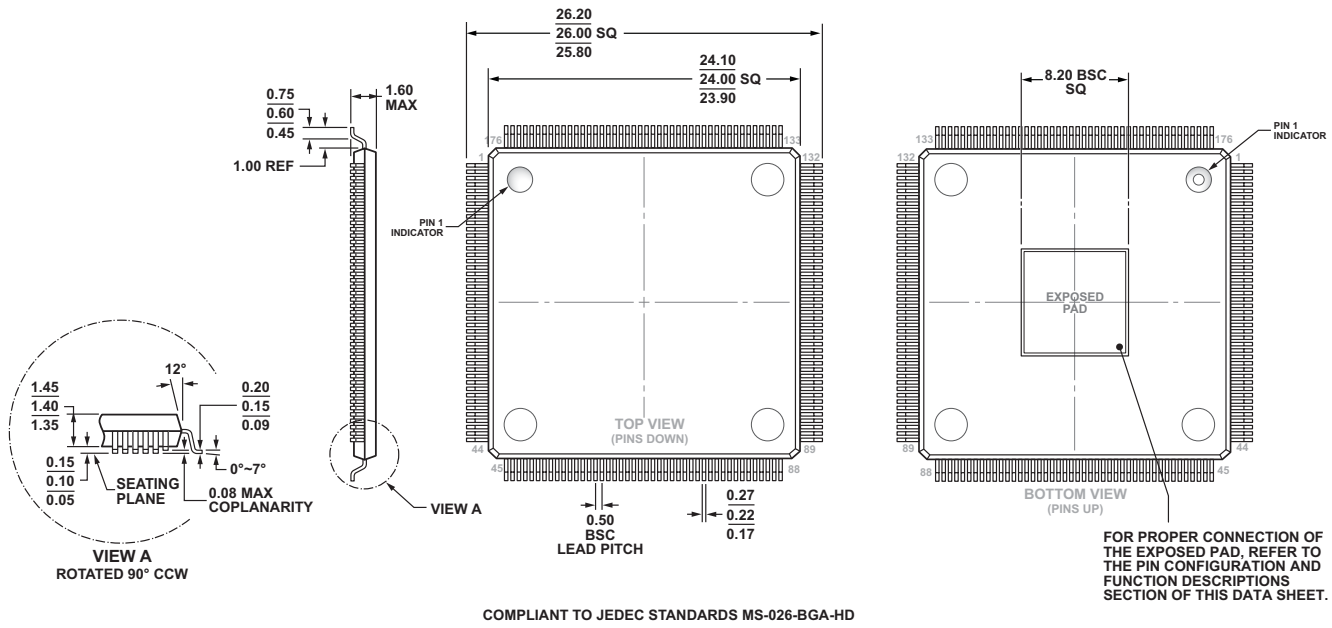


Figure 93. 176-Lead Low Profile Quad Flat Package, Exposed Pad [LQFP_EP]
(SW-176-5)
Dimensions shown in millimeters

SURFACE-MOUNT DESIGN

Table 98 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 98. CSP_BGA Data for Use with Surface-Mount Design

Package	Package Ball Attach Type	Package Solder Mask Opening	Package Ball Pad Size
BC-400-2	Solder Mask Defined	0.4 mm Diameter	0.5 mm Diameter

Downloaded from Arrow.com.

AUTOMOTIVE PRODUCTS

The following models are available with controlled manufacturing to support the quality and reliability requirements of automotive applications. Note that these automotive models may have specifications that differ from the commercial models; therefore, designers should review the Specifications section of this data sheet carefully. Only the automotive grade products shown in Table 99 are available for use in automotive applications. Contact your local Analog Devices account representative for specific product ordering information and to obtain the specific Automotive Reliability reports for these models.

Table 99. Automotive Products

Model [1, 2, 3]	Processor Instruction Rate (Max)	ARM Instruction Rate (Max)[4]	Temperature Range[5]	ARM Cores[4]	SHARC+ Cores	External Memory Ports	Package Description	Package Option
AD21571WCSWZ4xx	450 MHz	N/A	–40°C to +105°C	N/A	2	0	176-Lead LQFP_EP	SW-176-5
AD21571WCSWZ5xx	500 MHz	N/A	–40°C to +105°C	N/A	2	0	176-Lead LQFP_EP	SW-176-5
AD21573WCBCZ4xx	450 MHz	N/A	–40°C to +105°C	N/A	2	1	Pad 400-Ball CSP_BGA	BC-400-2
AD21573WCBCZ5xx	500 MHz	N/A	–40°C to +105°C	N/A	2	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSC570WCSWZ42xx	450 MHz	225 MHz	–40°C to +105°C	1	1	0	176-Lead LQFP_EP	SW-176-5
ADSC570WCSWZ4xx	450 MHz	450 MHz	–40°C to +105°C	1	1	0	176-Lead LQFP_EP	SW-176-5
ADSC571WCSWZ3xx	300 MHz	300 MHz	–40°C to +105°C	1	2	0	176-Lead LQFP_EP	SW-176-5
ADSC571WCSWZ4xx	450 MHz	450 MHz	–40°C to +105°C	1	2	0	176-Lead LQFP_EP	SW-176-5
ADSC571WCSWZ5xx	500 MHz	500 MHz	–40°C to +105°C	1	2	0	176-Lead LQFP_EP	SW-176-5
ADSC572WCBCZ42xx	450 MHz	225 MHz	–40°C to +105°C	1	1	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSC572WCBCZ4xx	450 MHz	450 MHz	–40°C to +105°C	1	1	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSC573WCBCZ3xx	300 MHz	300 MHz	–40°C to +105°C	1	2	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSC573WCBCZ4xx	450 MHz	450 MHz	–40°C to +105°C	1	2	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSC573WCBCZ5xx	500 MHz	500 MHz	–40°C to +105°C	1	2	1	Pad 400-Ball CSP_BGA	BC-400-2

[1] Z = RoHS Compliant Part.
[2] xx denotes the current die revision.
[3] For evaluation of all models, order the ADZS-SC573-EZLITE evaluation board.
[4] N/A means not applicable.
[5] Referenced temperature is ambient temperature. The ambient temperature is not a specification. Please see the Operating Conditions section for the junction temperature (T_J) specification which is the only temperature specification.

Downloaded from Arrow.com.

ORDERING GUIDE

Model[1, 2]	Processor Instruction Rate (Max)	ARM Instruction Rate (Max)[3]	Temperature Range[4]	ARM Cores[3]	SHARC+ Cores	External Memory Ports	Package Description	Package Option
ADSP-21571KSWZ-4	450 MHz	N/A	0°C to +70°C	N/A	2	0	176-Lead LQFP_EP	SW-176-5
ADSP-21571BSWZ-4	450 MHz	N/A	–40°C to +85°C	N/A	2	0	176-Lead LQFP_EP	SW-176-5
ADSP-21571CSWZ-4	450 MHz	N/A	–40°C to +105°C	N/A	2	0	176-Lead LQFP_EP	SW-176-5
ADSP-21571KSWZ-5	500 MHz	N/A	0°C to +70°C	N/A	2	0	176-Lead LQFP_EP	SW-176-5
ADSP-21571BSWZ-5	500 MHz	N/A	–40°C to +85°C	N/A	2	0	176-Lead LQFP_EP	SW-176-5
ADSP-21571CSWZ-5	500 MHz	N/A	–40°C to +100°C	N/A	2	0	176-Lead LQFP_EP	SW-176-5
ADSP-21573KBCZ-4	450 MHz	N/A	0°C to +70°C	N/A	2	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSP-21573BBCZ-4	450 MHz	N/A	–40°C to +85°C	N/A	2	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSP-21573CBCZ-4	450 MHz	N/A	–40°C to +100°C	N/A	2	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSP-21573KBCZ-5	500 MHz	N/A	0°C to +70°C	N/A	2	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSP-21573BBCZ-5	500 MHz	N/A	–40°C to +85°C	N/A	2	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSP-21573CBCZ-5	500 MHz	N/A	–40°C to +95°C	N/A	2	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSP-SC570KSWZ-42	450 MHz	225 MHz	0°C to +70°C	1	1	0	176-Lead LQFP_EP	SW-176-5
ADSP-SC570BSWZ-42	450 MHz	225 MHz	–40°C to +85°C	1	1	0	176-Lead LQFP_EP	SW-176-5
ADSP-SC570CSWZ-42	450 MHz	225 MHz	–40°C to +105°C	1	1	0	176-Lead LQFP_EP	SW-176-5
ADSP-SC570KSWZ-4	450 MHz	450 MHz	0°C to +70°C	1	1	0	176-Lead LQFP_EP	SW-176-5
ADSP-SC570BSWZ-4	450 MHz	450 MHz	–40°C to +85°C	1	1	0	176-Lead LQFP_EP	SW-176-5
ADSP-SC570CSWZ-4	450 MHz	450 MHz	–40°C to +105°C	1	1	0	176-Lead LQFP_EP	SW-176-5
ADSP-SC571KSWZ-3	300 MHz	300 MHz	0°C to +70°C	1	2	0	176-Lead LQFP_EP	SW-176-5
ADSP-SC571BSWZ-3	300 MHz	300 MHz	–40°C to +85°C	1	2	0	176-Lead LQFP_EP	SW-176-5
ADSP-SC571CSWZ-3	300 MHz	300 MHz	–40°C to +105°C	1	2	0	176-Lead LQFP_EP	SW-176-5
ADSP-SC571KSWZ-4	450 MHz	450 MHz	0°C to +70°C	1	2	0	176-Lead LQFP_EP	SW-176-5
ADSP-SC571BSWZ-4	450 MHz	450 MHz	–40°C to +85°C	1	2	0	176-Lead LQFP_EP	SW-176-5
ADSP-SC571CSWZ-4	450 MHz	450 MHz	–40°C to +105°C	1	2	0	176-Lead LQFP_EP	SW-176-5
ADSP-SC571KSWZ-5	500 MHz	500 MHz	0°C to +70°C	1	2	0	176-Lead LQFP_EP	SW-176-5
ADSP-SC571BSWZ-5	500 MHz	500 MHz	–40°C to +85°C	1	2	0	176-Lead LQFP_EP	SW-176-5
ADSP-SC571CSWZ-5	500 MHz	500 MHz	–40°C to +100°C	1	2	0	176-Lead LQFP_EP	SW-176-5
ADSP-SC572KBCZ-42	450 MHz	225 MHz	0°C to +70°C	1	1	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSP-SC572BBCZ-42	450 MHz	225 MHz	–40°C to +85°C	1	1	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSP-SC572CBCZ-42	450 MHz	225 MHz	–40°C to +100°C	1	1	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSP-SC572KBCZ-4	450 MHz	450 MHz	0°C to +70°C	1	1	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSP-SC572BBCZ-4	450 MHz	450 MHz	–40°C to +85°C	1	1	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSP-SC572CBCZ-4	450 MHz	450 MHz	–40°C to +100°C	1	1	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSP-SC573KBCZ-3	300 MHz	300 MHz	0°C to +70°C	1	2	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSP-SC573BBCZ-3	300 MHz	300 MHz	–40°C to +85°C	1	2	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSP-SC573CBCZ-3	300 MHz	300 MHz	–40°C to +100°C	1	2	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSP-SC573KBCZ-4	450 MHz	450 MHz	0°C to +70°C	1	2	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSP-SC573BBCZ-4	450 MHz	450 MHz	–40°C to +85°C	1	2	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSP-SC573CBCZ-4	450 MHz	450 MHz	–40°C to +100°C	1	2	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSP-SC573KBCZ-5	500 MHz	500 MHz	0°C to +70°C	1	2	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSP-SC573BBCZ-5	500 MHz	500 MHz	–40°C to +85°C	1	2	1	Pad 400-Ball CSP_BGA	BC-400-2
ADSP-SC573CBCZ-5	500 MHz	500 MHz	–40°C to +95°C	1	2	1	Pad 400-Ball CSP_BGA	BC-400-2

[1] Z =RoHS Compliant Part.
[2] For evaluation of all models, order the ADZS-SC573-EZLITE evaluation board.
[3] N/A means not applicable.
[4] Referenced temperature is ambient temperature. The ambient temperature is not a specification. Please see the Operating Conditions section for the junction temperature (T_J) specification which is the only temperature specification.

Downloaded from Arrow.com.

I^2C refers to a communications protocol originally developed by Philips Semiconductors (now NXP Semiconductors).

©2018 Analog Devices, Inc. All rights reserved. Trademarks and registered trademarks are the property of their respective owners.
D16121-0-6/18(B)

www.analog.com

Downloaded from Arrow.com.