Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

SHARC+ Dual-Core DSP with ARM Cortex-A5

ADSP-SC570/SC571/SC572/SC573/ADSP-21571/21573

SYSTEM FEATURES

Dual-enhanced SHARC+ high performance floating-point cores
- **Up to 500 MHz per SHARC+ core**
- **Up to 3 Mb (384 kB) L1 SRAM memory per core with parity (optional ability to configure as cache)**
- **32-bit, 40-bit, and 64-bit floating-point support**
- **32-bit fixed point**
- **Byte, short word, word, long word addressed**

ARM Cortex-A5 core
- **500 MHz/800 DMIPS with NEON/VFPv4-D16/Jazelle**
- **32 kB L1 instruction cache with parity/32 kB L1 data cache with parity**
- **256 kB L2 cache with parity**

Powerful DMA system

On-chip memory protection

Integrated safety features

17 mm × 17 mm 400-ball CSP_BGA and 176-lead LQFP_EP, RoHS compliant

Low system power across automotive temperature range

MEMORY

Large on-chip L2 SRAM with ECC protection, up to 1 MB

One L3 interface optimized for low system power, providing 16-bit interface to DDR3 (supporting 1.5 V capable DDR3L devices), DDR2, or LPDDR1 SDRAM devices

ADDITIONAL FEATURES

Security and Protection
- **Cryptographic hardware accelerators**
- **Fast secure boot with IP protection**
- **Support for ARM TrustZone**

Accelerators
- **FIR, IIR offload engines**

Qualified for automotive applications

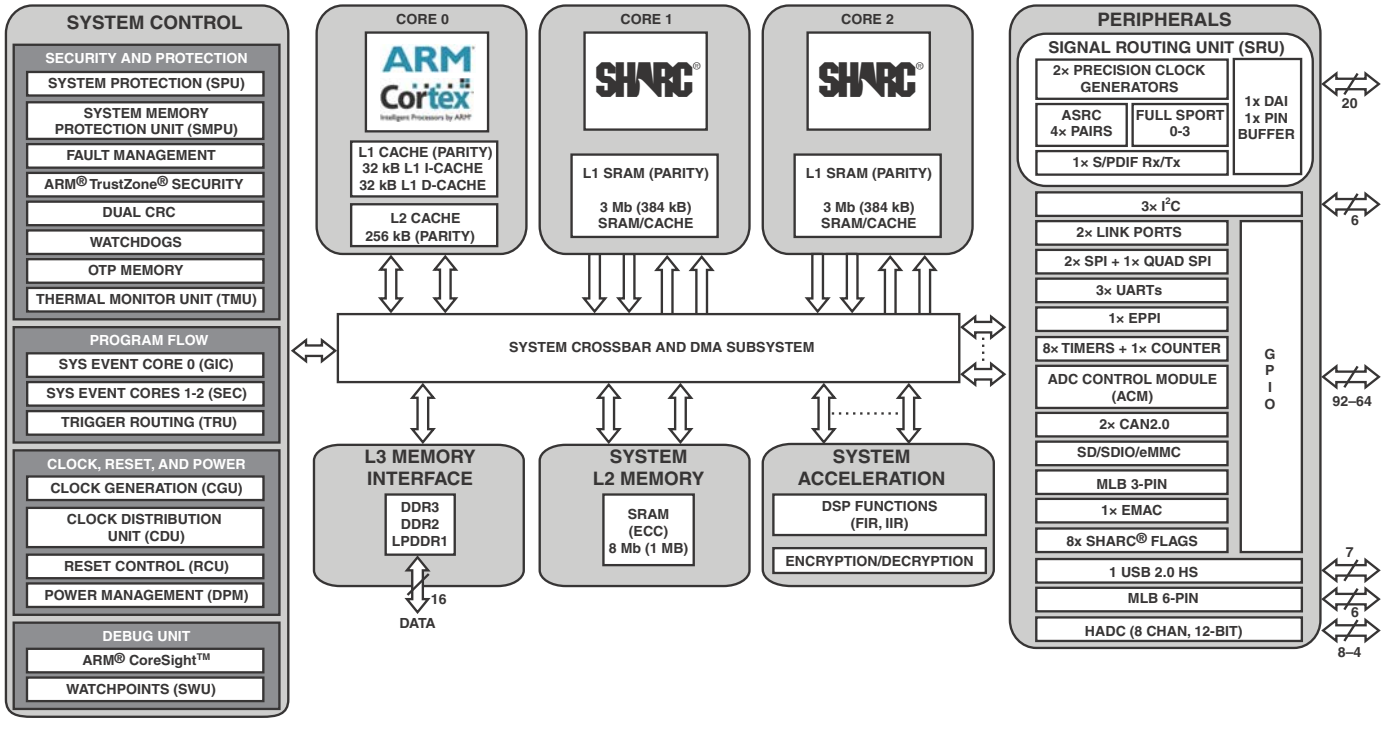

Figure 1. Processor Block Diagram

Rev. B Document Feedback

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106 U.S.A.
Tel: 781.329.4700 **Technical Support** **www.analog.com**

TABLE OF CONTENTS

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

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

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
	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
	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.

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

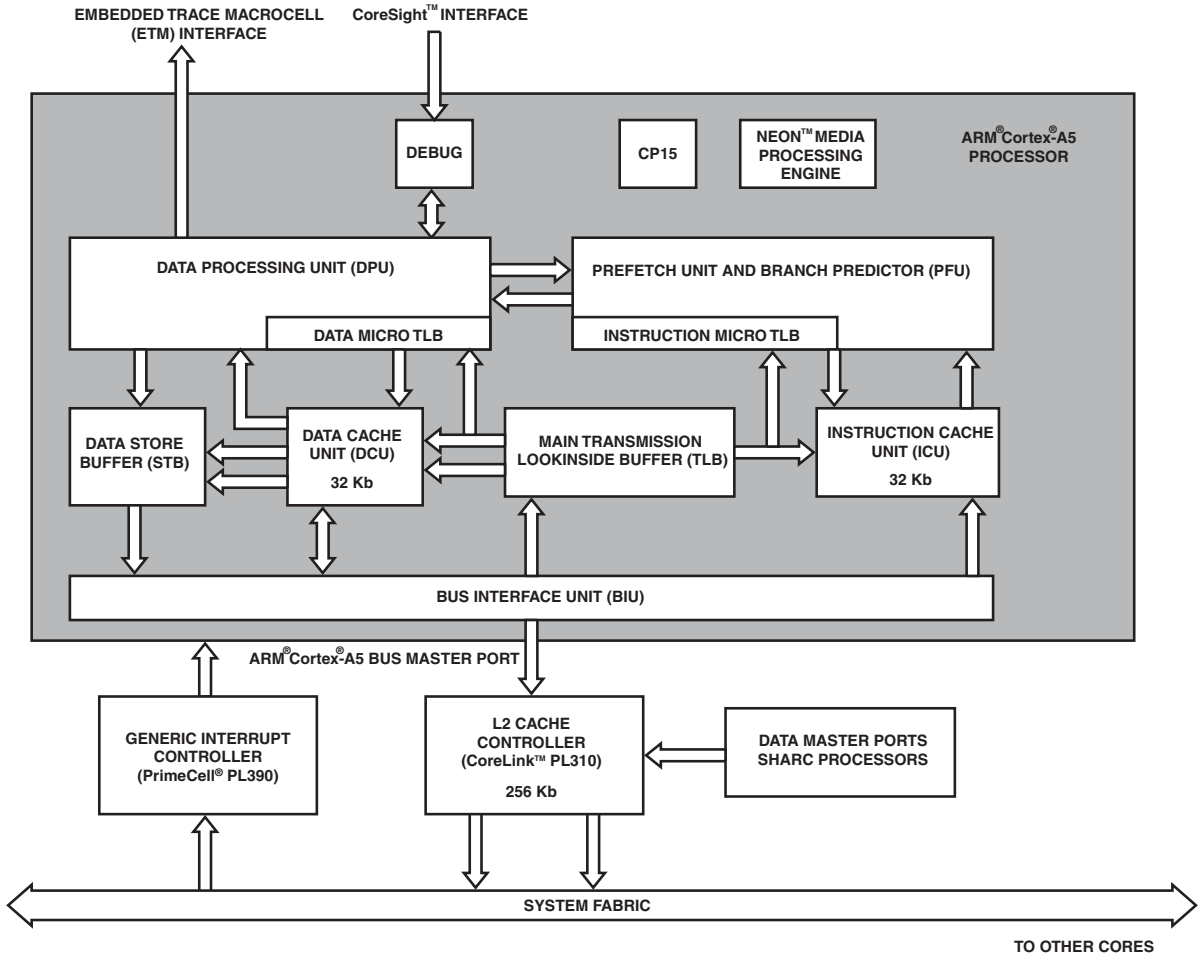

Figure 2. ARM Cortex-A5 Processor Block Diagram

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.

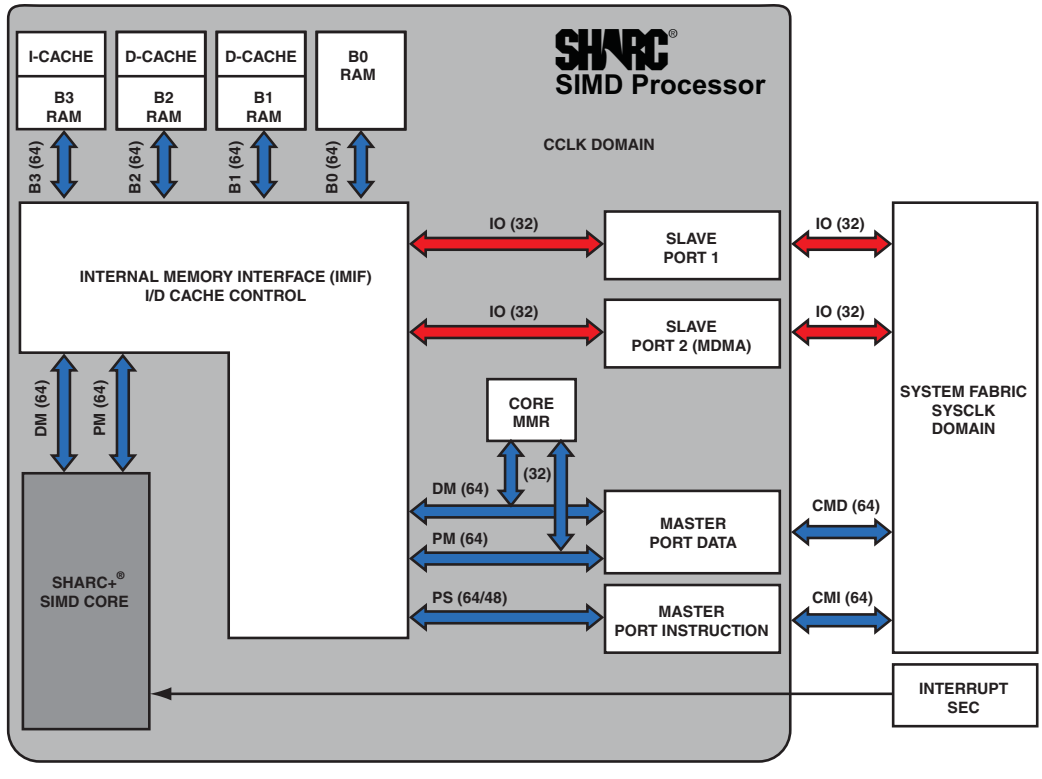

Figure 3. SHARC Processor Block Diagram

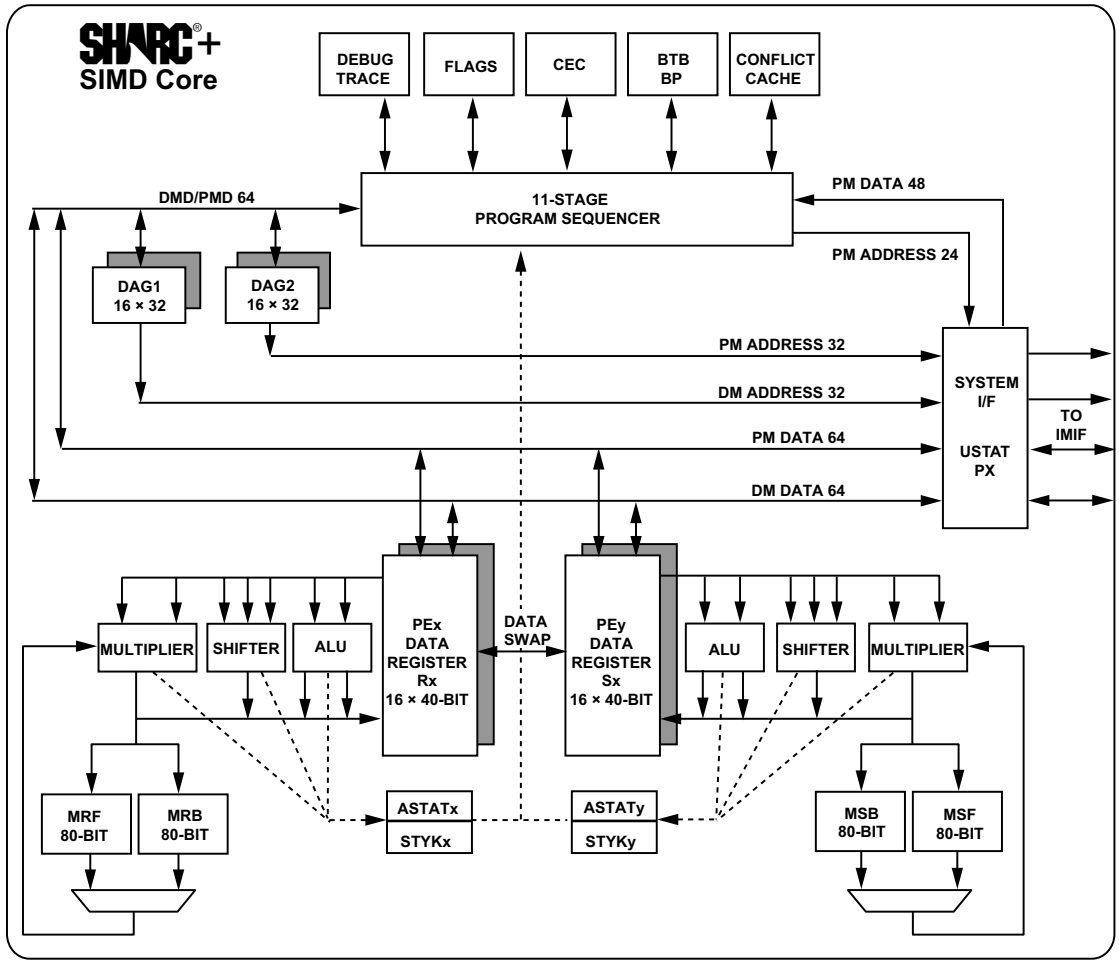

Figure 4. SHARC+ SIMD Core Block Diagram

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.

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.

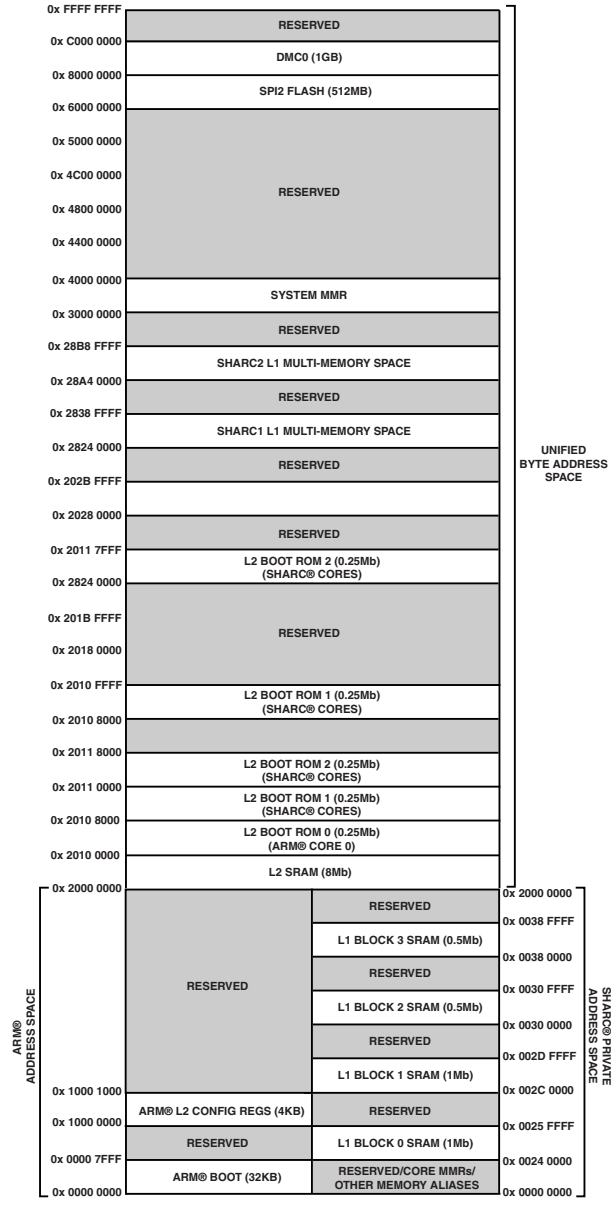

Figure 5. ADSP-SC57x/ADSP-2157x Memory Map

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.

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.

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).

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 SHARC/DMA: 0x20100000–0x20107FFF	0x08040000–0x08041FFF	0x00B20000–0x00B23FFF	0x00580000–0x00581555
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.

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

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

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

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

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,

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

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.

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.

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.

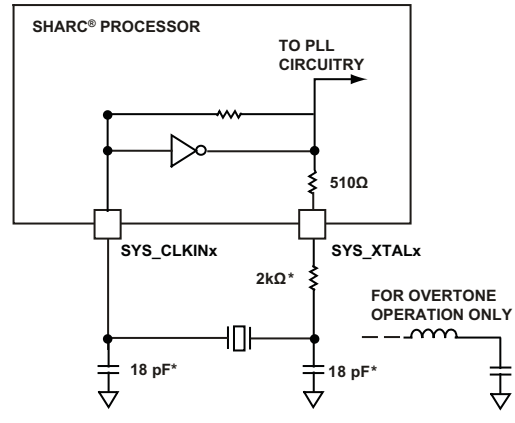

Figure 6. External Crystal Connection

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.

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.

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.

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

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.

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

Signal Name	Direction	Description
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.
DMC_WE	Output	**Write Enable.** Defines the operation for external dynamic memory to perform in conjunction with other DMC command signals. Connect to the 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.
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.**