Welcome to E-XFL.COM

Understanding Embedded - DSP (Digital Signal Processors)

Embedded - DSP (Digital Signal Processors) are specialized microprocessors designed to perform complex mathematical computations on digital signals in real-time. Unlike general-purpose processors, DSPs are optimized for high-speed numeric processing tasks, making them ideal for applications that require efficient and precise manipulation of digital data. These processors are fundamental in converting and processing signals in various forms, including audio, video, and communication signals, ensuring that data is accurately interpreted and utilized in embedded systems.

Applications of Embedded - DSP (Digital Signal Processors)

Details	
Product Status	Active
Type	Floating Point
Interface	CAN, EBI/EMI, Ethernet, DAI, I²C, MMC/SD/SDIO, SPI, SPORT, UART/USART, USB OTG
Clock Rate	450MHz
Non-Volatile Memory	ROM (512kB)
On-Chip RAM	640kB
Voltage - I/O	3.30V
Voltage - Core	1.10V
Operating Temperature	-40°C ~ 90°C (TA)
Mounting Type	Surface Mount
Package / Case	529-LFBGA, CSPBGA
Supplier Device Package	529-CSPBGA (19x19)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/adsc587wcbcz4b10

GENERAL DESCRIPTION

The ADSP-SC58x/ADSP-2158x processors are members of the SHARC® family of products. The ADSP-SC58x processor is based on the SHARC+ dual core and the ARM® Cortex®-A5 core. The ADSP-SC58x/ADSP-2158x SHARC processors are members of the 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 RISC (reduced instruction set computing), multimedia support, and leading edge signal processing in one integrated package. These applications span a wide array of markets, including automotive, pro audio, and industrial-based applications that require high floating-point performance.

Table 2 provides comparison information for features that vary across the standard processors. (N/A in the table means not applicable.)

Table 3 provides comparison information for features that vary across the automotive processors. (N/A in the table means not applicable.)

Table 1. Common Product Features

Product Features	ADSP-SC58x/ADSP-2158x
DAI (includes SRU)	2
Full SPORTs	4 per DAI
S/PDIF receive/transmit	1per DAI
ASRCs	4 pair per DAI
PCGs	2 per DAI
I^2C (TWI)	3
Quad-data bit SPI	1
Dual-data bit SPI	2
CAN2.0	2
UARTs	3
Link ports	2
Enhanced PPI	1
GP timer[1]	8
GP counter	1
Enhanced PWMs[2]	3
Watchdog timers	2
ADC control module	Yes
Static memory controller	Yes
Hardware accelerators	
High performance FFT/IFFT	Yes
FIR/IIR	Yes
Harmonic analysis engine	Yes
SINC filter	Yes
Security cryptographic engine	Yes
Multichannel 12-bit ADC	8-channel

[1] Eight timers are available in the 529-BGA package only. The 349-BGA package does not include Timer 6 and 7.

[2] Three 3ePWMs are available in the 529-BGA package only. The 349-BGA package does not include PWM 2.

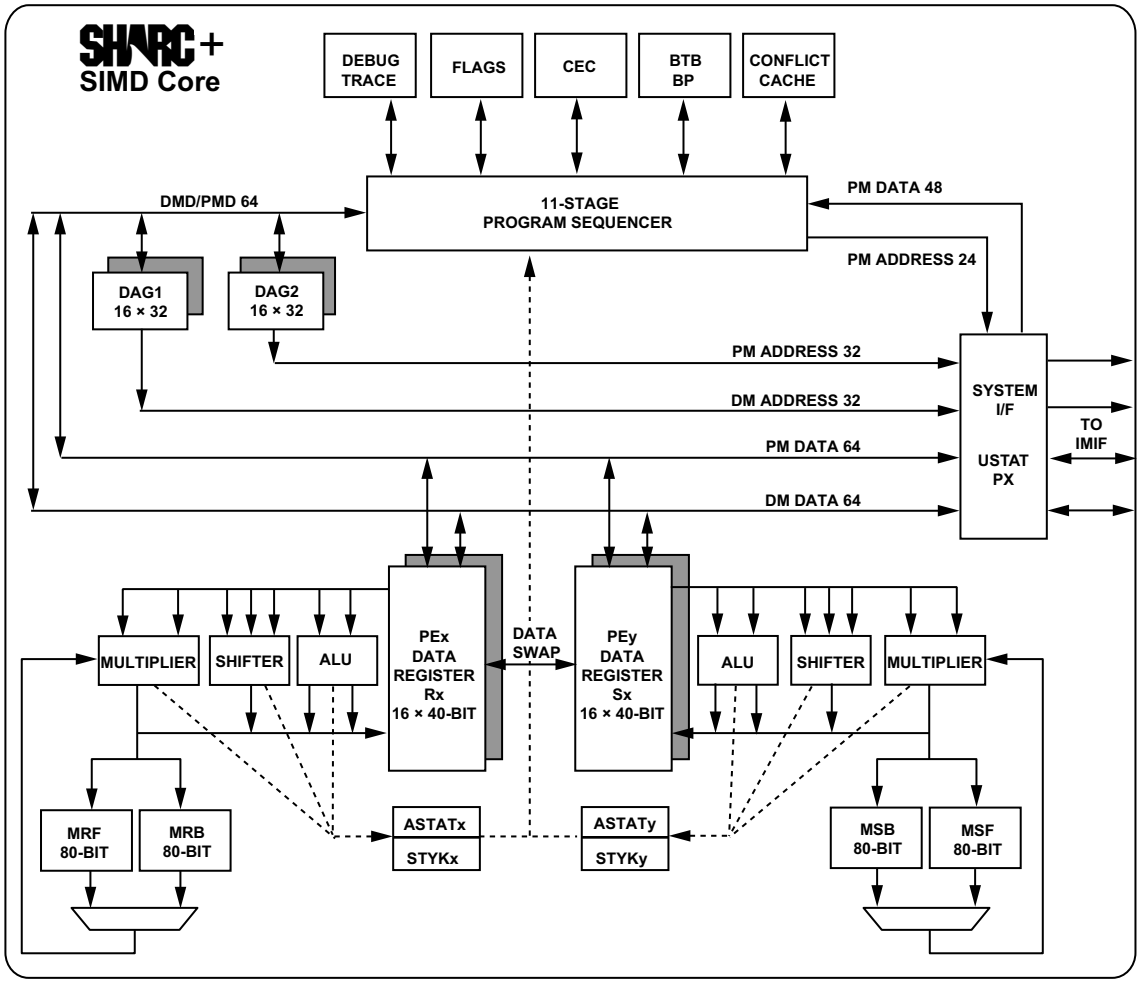


Figure 4. SHARC+ SIMD Core Block Diagram

L1 Memory

Figure 5 shows the ADSP-SC58x/ADSP-2158x memory map. Each SHARC+ core has a tightly coupled L1 SRAM of up to 5 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 0x 0000 0000 through 0x 0003 FFFF in Normal Word (32-bit). Each block can be configured for different combinations of code and data storage. Of the 5 Mb SRAM, up to 1024 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 DMA engine in a single cycle. The SRAM of the processor can be configured as a maximum of 160k words of 32-bit data, 320k words of 16-bit data, 106.7k words of 48-bit instructions (or 40-bit data), or combinations of different word sizes up to 5 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 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. Both slave ports allow conflict free core/direct memory access (DMA) streams to the individual memory blocks. For slave port addresses, 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.

Instruction and Data Cache

The ADSP-SC58x/ADSP-2158x processors also include a traditional instruction cache (I-cache) and two data caches (D-cache) (PM and DM 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. For all SEC channels, the processor automatically stacks the arithmetic status (ASTATx and ASTATy) registers and mode (MODE1) register in parallel with the interrupt servicing.

Core Memory-Mapped Registers (CMMR)

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

SHARC+ CORE ARCHITECTURE

The ADSP-SC58x/ADSP-2158x 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.

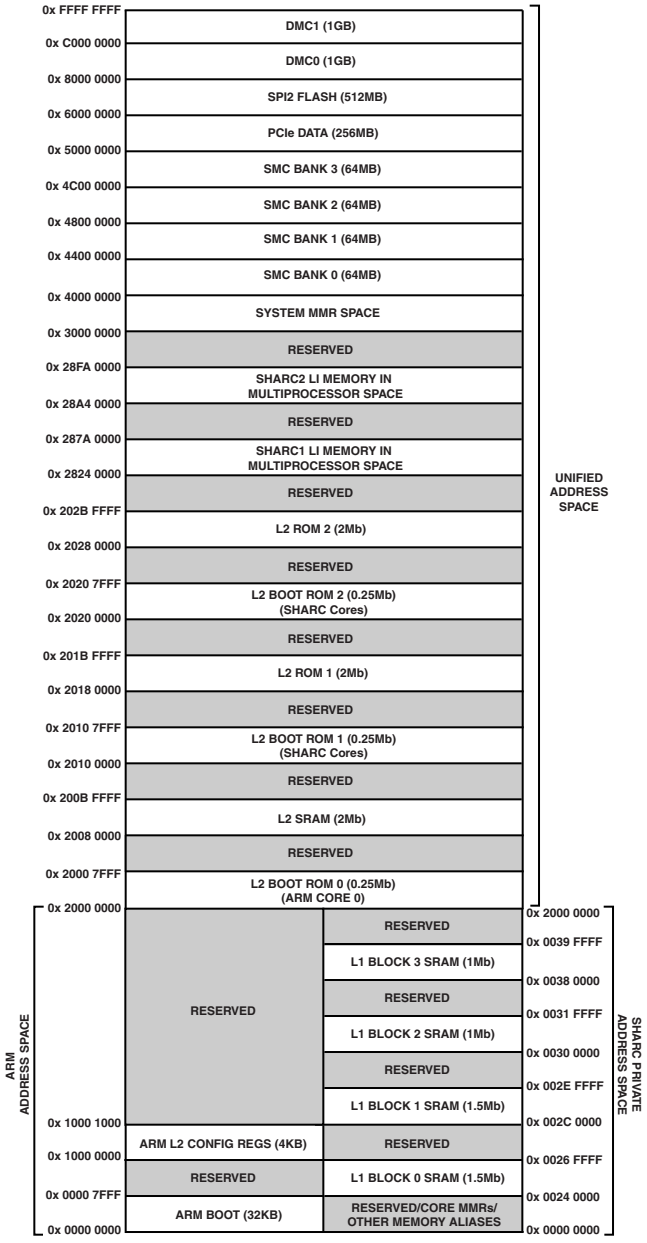


Figure 5. ADSP-SC58x/ADSP-2158x Memory Map

The ADSP-SC58x/ADSP-2158x 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.

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).

Single instruction multiple data (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 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 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 (USTAT)

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 With Zero-Overhead Hardware Circular Buffer Support

For indirect addressing and implementing circular data buffers in hardware, the ADSP-SC58x/ADSP-2158x 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 Fourier transforms. 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 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.

Table 7. Memory Map of Mapped I/Os

	Byte Address Space ARM Cortex-A5 – Data Access and Instruction Fetch SHARC+ – Data Access	Normal Word Address Space for Data Access SHARC+	SHARC+ Core Instruction Fetch	
			VISA Space	ISA Space
SMC Bank 0 (64 MB)	0x40000000–0x43FFFFFF	0x01000000–0x01FFFFFF	0x00F00000–0x00F3FFFF	0x00700000–0x0073FFFF
SMC Bank 1 (64 MB)	0x44000000–0x47FFFFFF	Not applicable	Not applicable	Not applicable
SMC Bank 2 (64 MB)	0x48000000–0x4BFFFFFF	Not applicable	Not applicable	Not applicable
SMC Bank 3 (64 MB)	0x4C000000–0x4FFFFFFF	Not applicable	Not applicable	Not applicable
PCIe Data (256 MB)	0x50000000–0x5FFFFFFF	0x02000000–0x03FFFFFF	0x00F40000–0x00F7FFFF	0x00740000–0x0077FFFF
SPI2 Memory (512 MB)	0x60000000–0x7FFFFFFF	0x04000000–0x07FFFFFF	0x00F80000–0x00FFFFFF	0x00780000–0x007FFFFF

Table 8. DMC Memory Map

	Byte Address Space ARM Cortex-A5 – Data Access and Instruction Fetch SHARC+ – Data Access	Normal Word Address Space for Data Access SHARC+	SHARC+ Core Instruction Fetch	
			VISA Space	ISA Space
DMC0 (1 GB)	0x80000000–0xBFFFFFFF	0x10000000–0x17FFFFFF	0x00800000–0x00AFFFFF	0x00400000–0x004FFFFF
DMC1 (1 GB)	0xC0000000–0xFFFFFFFF	0x18000000–0x1FFFFFFF	0x00C00000–0x00EFFFFF	0x00600000–0x006FFFFF

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: one channel is the source channel and the second is 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

SECURITY FEATURES

The following sections describe the security features of the ADSP-SC58x/ADSP-2158x processors.

ARM TrustZone

The ADSP-SC58x 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-SC58x/ADSP-2158x 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 and to block access to secure resources from nonsecure masters.

System Memory Protection Unit (SMPU)

Synonymously, 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-SC58x/ADSP-2158x 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.

SAFETY FEATURES

The ADSP-SC58x/ADSP-2158x 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 does not protect 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.

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 L3 memories (DDR2, LPDDR). The processors feature two CRC engines that are embedded in the memory to memory DMA controllers.

Table 12. ADSP-SC58x/ADSP-2158x 349-Ball CSP_BGA Signal Descriptions (Continued)

Signal Name	Description	Port	Pin Name
SYS_BMODE0	Boot Mode Control n	Not Muxed	SYS_BMODE0
SYS_BMODE1	Boot Mode Control n	Not Muxed	SYS_BMODE1
SYS_BMODE2	Boot Mode Control n	Not Muxed	SYS_BMODE2
SYS_CLKIN0	Clock/Crystal Input	Not Muxed	SYS_CLKIN0
SYS_CLKIN1	Clock/Crystal Input	Not Muxed	SYS_CLKIN1
SYS_CLKOUT	Processor Clock Output	Not Muxed	SYS_CLKOUT
SYS_FAULT	Active High Fault Output	Not Muxed	SYS_FAULT
$\overline{\text{SYS\_FAULT}}$	Active Low Fault Output	Not Muxed	$\overline{\text{SYS\_FAULT}}$
$\overline{\text{SYS\_HWRST}}$	Processor Hardware Reset Control	Not Muxed	$\overline{\text{SYS\_HWRST}}$
$\overline{\text{SYS\_RESOUT}}$	Reset Output	Not Muxed	$\overline{\text{SYS\_RESOUT}}$
SYS_XTAL0	Crystal Output	Not Muxed	SYS_XTAL0
SYS_XTAL1	Crystal Output	Not Muxed	SYS_XTAL1
TM0_ACI0	TIMER0 Alternate Capture Input 0	C	PC_14
TM0_ACI1	TIMER0 Alternate Capture Input 1	B	PB_03
TM0_ACI2	TIMER0 Alternate Capture Input 2	D	PD_13
TM0_ACI3	TIMER0 Alternate Capture Input 3	C	PC_07
TM0_ACI4	TIMER0 Alternate Capture Input 4	B	PB_10
TM0_ACLK1	TIMER0 Alternate Clock 1	D	PD_08
TM0_ACLK2	TIMER0 Alternate Clock 2	D	PD_09
TM0_ACLK3	TIMER0 Alternate Clock 3	B	PB_00
TM0_ACLK4	TIMER0 Alternate Clock 4	B	PB_01
TM0_CLK	TIMER0 Clock	C	PC_11
TM0_TMR0	TIMER0 Timer 0	E	PE_09
TM0_TMR1	TIMER0 Timer 1	B	PB_15
TM0_TMR2	TIMER0 Timer 2	B	PB_10
TM0_TMR3	TIMER0 Timer 3	B	PB_07
TM0_TMR4	TIMER0 Timer 4	B	PB_08
TM0_TMR5	TIMER0 Timer 5	B	PB_14
TRACE0_CLK	TRACE0 Trace Clock	D	PD_10
TRACE0_D00	TRACE0 Trace Data 0	D	PD_02
TRACE0_D01	TRACE0 Trace Data 1	D	PD_03
TRACE0_D02	TRACE0 Trace Data 2	D	PD_04
TRACE0_D03	TRACE0 Trace Data 3	D	PD_05
TRACE0_D04	TRACE0 Trace Data 4	D	PD_06
TRACE0_D05	TRACE0 Trace Data 5	D	PD_07
TRACE0_D06	TRACE0 Trace Data 6	D	PD_08
TRACE0_D07	TRACE0 Trace Data 7	D	PD_09
TWI0_SCL	TWI0 Serial Clock	Not Muxed	TWI0_SCL
TWI0_SDA	TWI0 Serial Data	Not Muxed	TWI0_SDA
TWI1_SCL	TWI1 Serial Clock	Not Muxed	TWI1_SCL
TWI1_SDA	TWI1 Serial Data	Not Muxed	TWI1_SDA
TWI2_SCL	TWI2 Serial Clock	Not Muxed	TWI2_SCL
TWI2_SDA	TWI2 Serial Data	Not Muxed	TWI2_SDA
$\overline{\text{UART0\_CTS}}$	UART0 Clear to Send	D	PD_00
$\overline{\text{UART0\_RTS}}$	UART0 Request to Send	C	PC_15
$\overline{\text{UART0\_RX}}$	UART0 Receive	C	PC_14
$\overline{\text{UART0\_TX}}$	UART0 Transmit	C	PC_13
$\overline{\text{UART1\_CTS}}$	UART1 Clear to Send	E	PE_01

Table 12. ADSP-SC58x/ADSP-2158x 349-Ball CSP_BGA Signal Descriptions (Continued)

Signal Name	Description	Port	Pin Name
UART1_RTS	UART1 Request to Send	E	PE_02
UART1_RX	UART1 Receive	B	PB_03
UART1_TX	UART1 Transmit	B	PB_02
UART2_CTS	UART2 Clear to Send	E	PE_11
UART2_RTS	UART2 Request to Send	E	PE_10
UART2_RX	UART2 Receive	D	PD_13
UART2_TX	UART2 Transmit	D	PD_12
USB0_CLKIN	USB0 Clock/Crystal Input	Not Muxed	USB_CLKIN
USB0_DM	USB0 Negative Data (–)	Not Muxed	USB0_DM
USB0_DP	USB0 Positive Data (+)	Not Muxed	USB0_DP
USB0_ID	USB0 OTG ID	Not Muxed	USB0_ID
USB0_VBC	USB0 VBUS Control	Not Muxed	USB0_VBC
USB0_VBUS	USB0 Bus Voltage	Not Muxed	USB0_VBUS
USB0_XTAL	USB0 Crystal	Not Muxed	USB_XTAL
VDD_DMC	DMC VDD	Not Muxed	VDD_DMC
VDD_HADC	HADC VDD	Not Muxed	VDD_HADC
VDD_USB	USB VDD	Not Muxed	VDD_USB

Table 27. ADSP-SC58x/ADSP-2158x Designer Quick Reference (Continued)

Signal Name	Type	Driver Type	Int Term	Reset Term	Reset Drive	Power Domain	Description and Notes
PA_02	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTA Position 2 | EMAC0 Management Channel Clock | SMC0 Address 24 Notes: No notes
PA_03	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTA Position 3 | EMAC0 Management Channel Serial Data | SMC0 Address 23 Notes: No notes
PA_04	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTA Position 4 | EMAC0 Receive Data 0 | SMC0 Address 19 Notes: No notes
PA_05	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTA Position 5 | EMAC0 Receive Data 1 | SMC0 Address 18 Notes: No notes
PA_06	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTA Position 6 | EMAC0 RXCLK (GigE) or REFCLK (10/100) | SMC0 Address 17 Notes: No notes
PA_07	InOut	A	PullDown	none	none	VDD_EXT	Desc: EMAC0 RXCTL (GigE) or CRS (10/100) | PORTA Position 7 | EMAC0 Carrier Sense/RMII Receive Data Valid | SMC0 Address 16 Notes: No notes
PA_08	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTA Position 8 | EMAC0 Receive Data 2 | SMC0 Address 12 Notes: No notes
PA_09	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTA Position 9 | EMAC0 Receive Data 3 | SMC0 Address 11 Notes: No notes
PA_10	InOut	A	PullDown	none	none	VDD_EXT	Desc: EMAC0 TXCTL (GigE) or TXEN (10/100) | PORTA Position 10 | EMAC0 Transmit Enable | SMC0 Address 22 Notes: No notes
PA_11	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTA Position 11 | EMAC0 Transmit Clock | SMC0 Address 15 Notes: No notes
PA_12	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTA Position 12 | EMAC0 Transmit Data 2 | SMC0 Address 14 Notes: No notes
PA_13	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTA Position 13 | EMAC0 Transmit Data 3 | SMC0 Address 13 Notes: No notes
PA_14	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTA Position 14 | EMAC0 PTP Pulse-Per-Second Output 3 | SINC0 Data 0 | SMC0 Address 10 Notes: No notes

Table 34. Dynamic Current for Each SHARC+ Core (mA, with ASF = 1.00)

	Voltage (V_{DD_INT})		
f_{CCLK} (MHz)	1.05	1.10	1.15
450	321.3	336.6	351.9
400	285.6	299.2	312.8
350	249.9	261.8	273.7
300	214.2	224.4	234.6
250	178.5	187.0	195.5
200	142.8	149.6	156.4
150	107.1	112.2	117.3
100	71.4	74.8	78.2

Table 35. Dynamic Current for the ARM Cortex-A5 Core (mA, with ASF = 1.00)

	Voltage (V_{DD_INT})		
f_{CCLK} (MHz)	1.05	1.10	1.15
450	70.88	74.25	77.63
400	63.00	66.00	69.00
350	55.13	57.75	60.38
300	47.25	49.50	51.75
250	39.38	41.25	43.13
200	31.50	33.00	34.50
150	23.63	24.75	25.88
100	15.75	16.50	17.25

The following equation is used to compute the power dissipation when the FFT accelerator is used:

$I_{DD\_INT\_ACCL\_DYN}$ (mA) = $ASF_{ACCL} \times f_{SYSCLK}$ (MHz) × V_{DD_INT} (V)

Table 36. Activity Scaling Factors for the FFT Accelerator (ASF_{ACCL})

I_{DD_INT} Power Vector	ASF_{ACCL}
Unused	0.0
I_{DD-TYP}	0.32

Clock Current

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

$I_{DD\_INT\_SYSCLK\_DYN}$ (mA) = 0.78 × f_{SYSCLK} (MHz) × V_{DD_INT} (V)

$I_{DD\_INT\_SCLK0\_DYN}$ (mA) = 0.44 × f_{SCLK0} (MHz) × V_{DD_INT} (V)

$I_{DD\_INT\_SCLK1\_DYN}$ (mA) = 0.06 × f_{SCLK1} (MHz) × V_{DD_INT} (V)

$I_{DD\_INT\_DCLK\_DYN}$ (mA) = 0.14 × f_{DCLK} (MHz) × V_{DD_INT} (V)

$I_{DD\_INT\_OCLK\_DYN}$ (mA) = 0.02 × f_{OCLK} (MHz) × V_{DD_INT} (V)

Current from High-Speed Peripheral Operation

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

$I_{DD\_INT\_USB\_DYN}$ = 20 mA (if both USBs are enabled in HS mode)

$I_{DD\_INT\_MLB\_DYN}$ = 10 mA (if MLB 6-pin interface is enabled)

$I_{DD\_INT\_GIGE\_DYN}$ = 10 mA (if gigabit EMAC is enabled)

$I_{DD\_INT\_PCIE\_DYN}$ = 240 mA (if PCIe is enabled in 5 Gbps mode)

Data Transmission Current

The data transmission current represents the power dissipated when moving data throughout the system via direct memory access (DMA). This current is proportional to the data rate. Refer to the power calculator available with "Estimating Power for ADSP-SC58x/2158x SHARC+ Processors" (EE-392) to estimate $I_{DD\_INT\_DMA\_DR\_DYN}$ based on the bandwidth of the data transfer.

TIMING SPECIFICATIONS

Specifications are subject to change without notice.

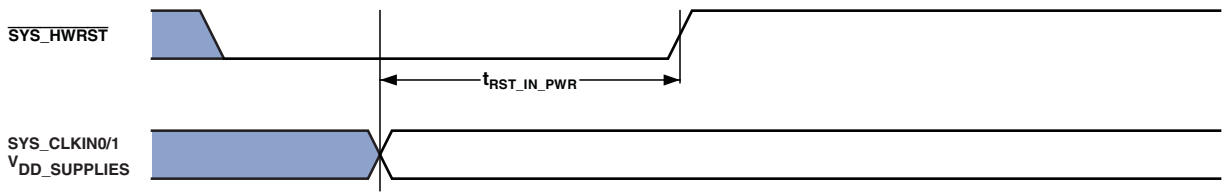
Power-Up Reset Timing

Table 43 and Figure 10 show the relationship between power supply startup and processor reset timing, related to the clock generation unit (CGU) and reset control unit (RCU).

In Figure 10, $V_{DD\_SUPPLIES}$ are V_{DD_INT}, V_{DD_EXT}, V_{DD_DMC}, V_{DD_USB}, V_{DD_HADC}, V_{DD_RTC}, $V_{DD\_PCI\_TX}$, $V_{DD\_PCI\_RX}$, and $V_{DD\_PCI\_CORE}$.

Table 43. Power-Up Reset Timing

Parameter		Min	Max	Unit
Timing Requirement				
$t_{RST\_IN\_PWR}$	$\overline{\text{SYS\_HWRST}}$ Deasserted after $V_{DD\_SUPPLIES}$ (V_{DD_INT}, V_{DD_EXT}, V_{DD_DMC}, V_{DD_USB}, V_{DD_HADC}, V_{DD_RTC}, $V_{DD\_PCI\_TX}$, $V_{DD\_PCI\_RX}$, $V_{DD\_PCI\_CORE}$) and SYS_CLKINx are Stable and Within Specification	$11 \times t_{CKIN}$		ns



NOTE: $V_{DD\_SUPPLIES}$ REFER TO V_{DD_INT}, V_{DD_EXT}, V_{DD_DMC}, V_{DD_USB}, V_{DD_HADC}, V_{DD_RTC}, $V_{DD\_PCI\_TX}$, $V_{DD\_PCI\_RX}$, AND $V_{DD\_PCI\_CORE}$.

Figure 10. Power-Up Reset Timing

Asynchronous Flash Read

Table 46 and Figure 13 show asynchronous flash memory read timing, related to the SMC.

Table 46. Asynchronous Flash Read

Parameter		Min	Max	Unit
Switching Characteristics				
t_{AMSADV}	SMC0_Ax (Address)/$\overline{\text{SMC0\_AMSx}}$ Assertion Before SMC0_NORDV Low[1]	PREST × t_{SCLK0} – 2		ns
t_{WADV}	SMC0_NORDV Active Low Width[2]	RST × t_{SCLK0} – 2		ns
$t_{DADVARE}$	$\overline{\text{SMC0\_ARE}}$ Low Delay From SMC0_NORDV High[3]	PREAT × t_{SCLK0} – 2		ns
t_{HARE}	Output[4] Hold After $\overline{\text{SMC0\_ARE}}$ High[5]	RHT × t_{SCLK0} – 2		ns
t_{WARE}[6]	$\overline{\text{SMC0\_ARE}}$ Active Low Width[7]	RAT × t_{SCLK0} – 2		ns

[1] PREST value set using the SMC_BxETIM.PREST bits.
[2] RST value set using the SMC_BxTIM.RST bits.
[3] PREAT value set using the SMC_BxETIM.PREAT bits.
[4] Output signals are SMC0_Ax, $\overline{\text{SMC0\_AMS}}$, $\overline{\text{SMC0\_AOE}}$.
[5] RHT value set using the SMC_BxTIM.RHT bits.
[6] SMC0_BxCTL.ARDYEN bit = 0.
[7] RAT value set using the SMC_BxTIM.RAT bits.

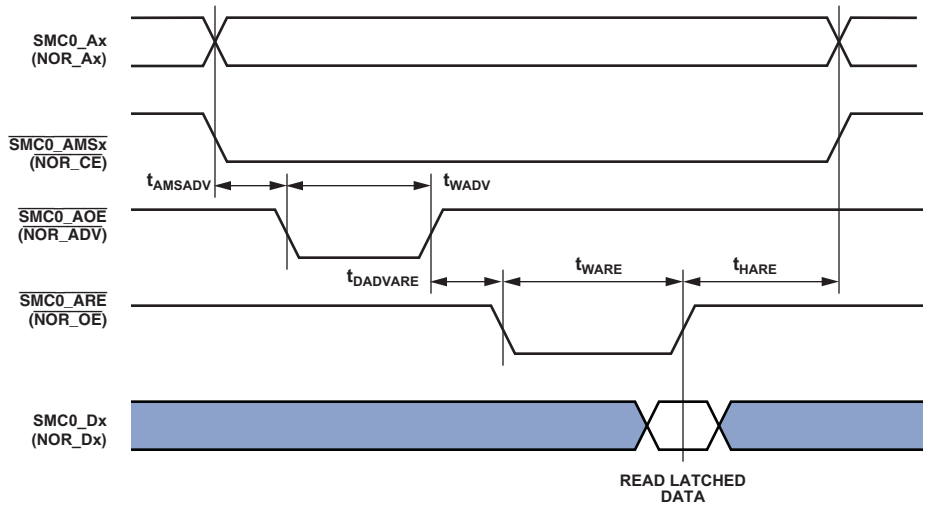


Figure 13. Asynchronous Flash Read

Asynchronous Page Mode Read

Table 47 and Figure 14 show asynchronous memory page mode read timing, related to the SMC.

Table 47. Asynchronous Page Mode Read

Parameter		Min	Max	Unit
Switching Characteristics				
t_{AV}	SMC0_Axx (Address) Valid for First Address Minimum Width[1]	(PREST + RST + PREAT + RAT) × t_{SCLK0} – 2		ns
t_{AV1}	SMC0_Axx (Address) Valid for Subsequent SMC0_Ax (Address) Minimum Width	PGWS × t_{SCLK0} – 2		ns
t_{WADV}	SMC0_NORDV Active Low Width[2]	RST × t_{SCLK0} – 2		ns
t_{HARE}	Output[3] Hold After $\overline{\text{SMC0\_ARE}}$ High[4]	RHT × t_{SCLK0} – 2		ns
t_{WARE}[5]	$\overline{\text{SMC0\_ARE}}$ Active Low Width[6, 7]	(RAT + (Nw – 1) × PGWS) × t_{SCLK0} – 2		ns

[1] PREST, RST, PREAT and RAT values set using the SMC_BxETIM.PREST bits, SMC_BxTIM.RST bits, SMC_BxETIM.PREAT bits, and the SMC_BxTIM.RAT bits.
[2] RST value set using the SMC_BxTIM.RST bits.
[3] Output signals are SMC0_Ax, $\overline{\text{SMC0\_AMSx}}$, $\overline{\text{SMC0\_AOE}}$.
[4] RHT value set using the SMC_BxTIM.RHT bits.
[5] SMC_BxCTL.ARDYEN bit = 0.
[6] RAT value set using the SMC_BxTIM.RAT bits.
[7] Nw = Number of 16-bit data words read.

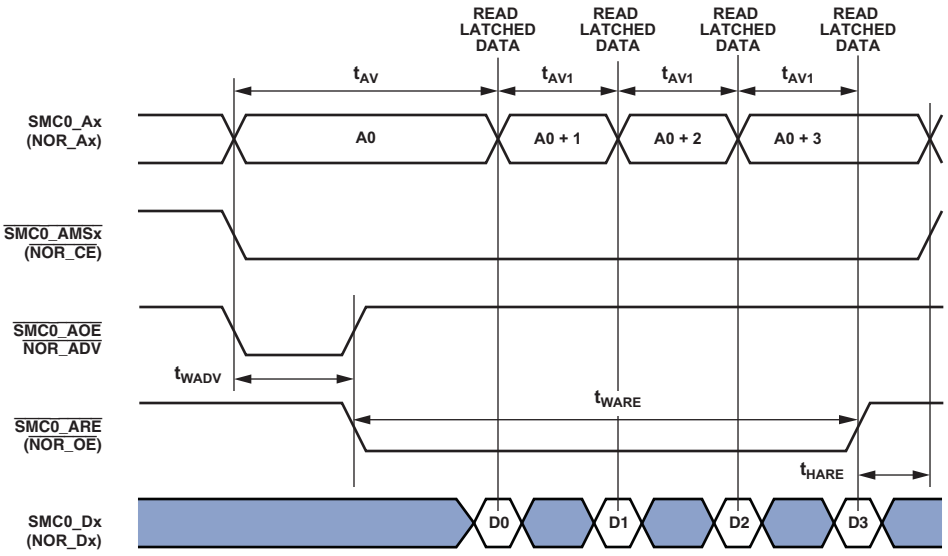


Figure 14. Asynchronous Page Mode Read

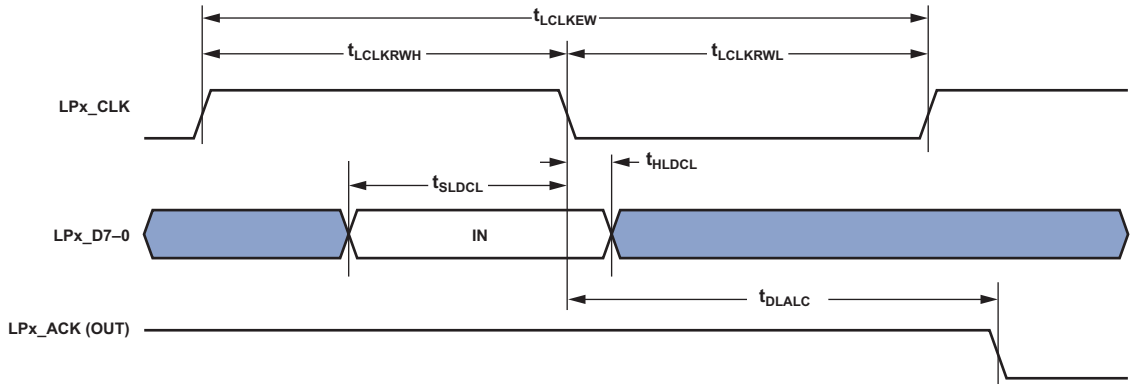


Figure 35. Link Ports—Receive

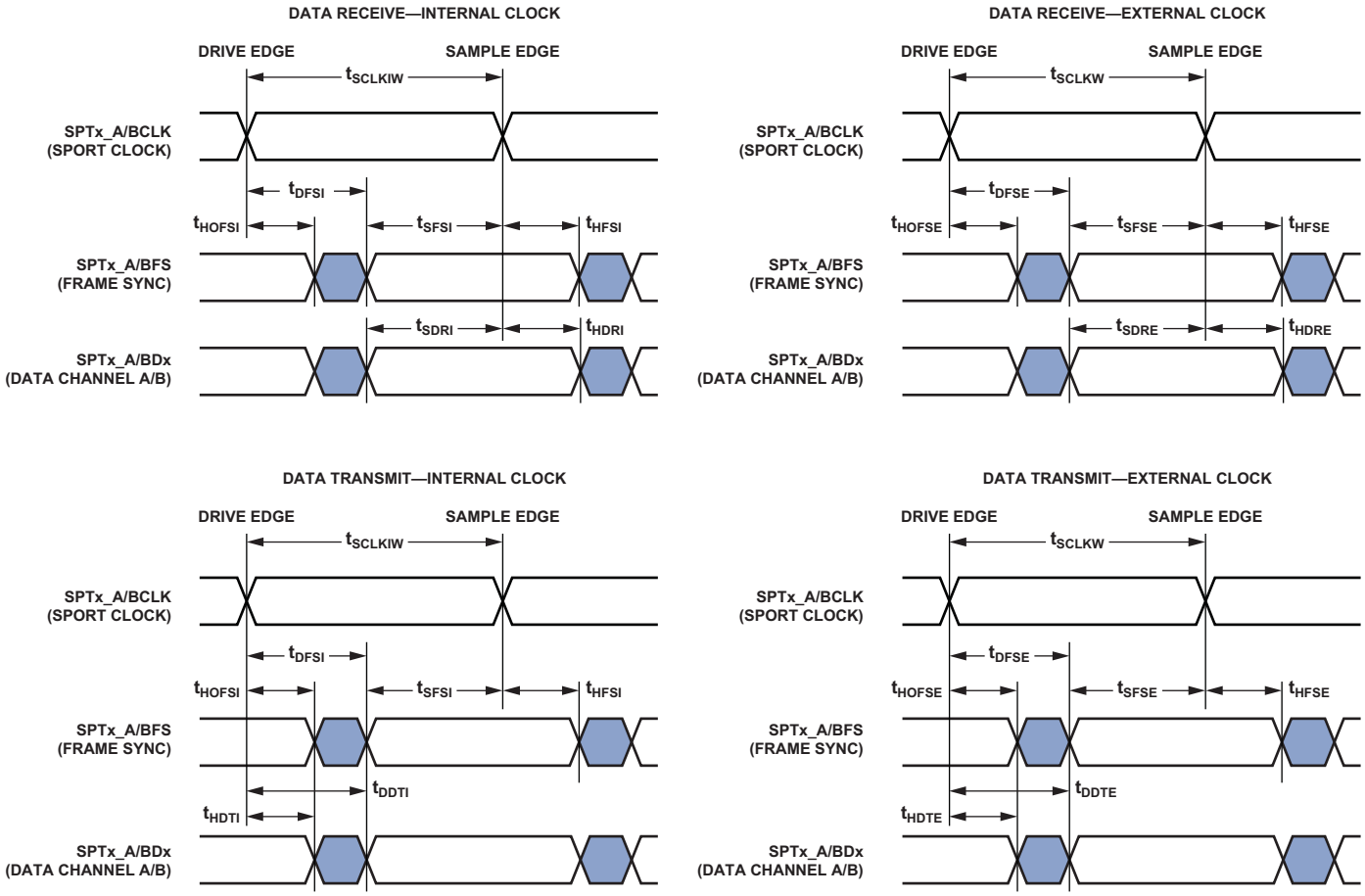


Figure 37. Serial Ports

Figure 65 and Table 94 show the default I²S justified mode. The frame sync is low for the left channel and high for the right channel. Data is valid on the rising edge of serial clock. The MSB is left justified to the frame sync transition but with a delay.

Table 94. S/PDIF Transmitter I²S Mode

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

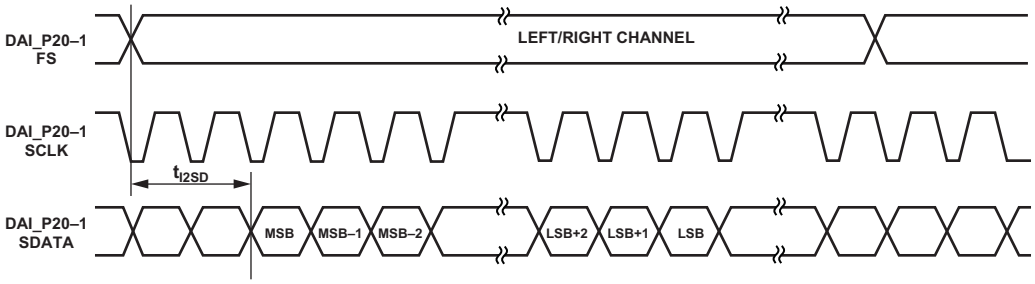


Figure 65. I²S Justified Mode

Figure 66 and Table 95 show the left justified mode. The frame sync is high for the left channel and low for the right channel. Data is valid on the rising edge of serial clock. The MSB is left justified to the frame sync transition with no delay.

Table 95. S/PDIF Transmitter Left Justified Mode

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

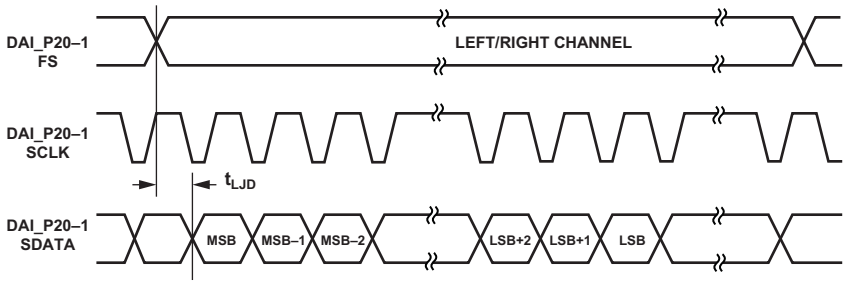


Figure 66. Left Justified Mode

Media Local Bus (MLB)

All the numbers shown in Table 99 are applicable for all MLB speed modes (1024 FS, 512 FS, and 256 FS) for the 3-pin protocol, unless otherwise specified. Refer to the *Media Local Bus Specification version 4.2* for more details.

Table 99. 3-Pin MLB Interface Specifications

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

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

Program Trace Macrocell (PTM) Timing

Table 102 and Figure 75 provide I/O timing related to the PTM.

Table 102. Trace Timing

Parameter		Min	Max	Unit
Switching Characteristics				
t_{DTRD}	Trace Data Delay From Trace Clock Maximum		5	ns
t_{HTRD}	Trace Data Hold From Trace Clock Minimum	2		ns
t_{PTRCK}	Trace Clock Period Minimum	12.32		ns

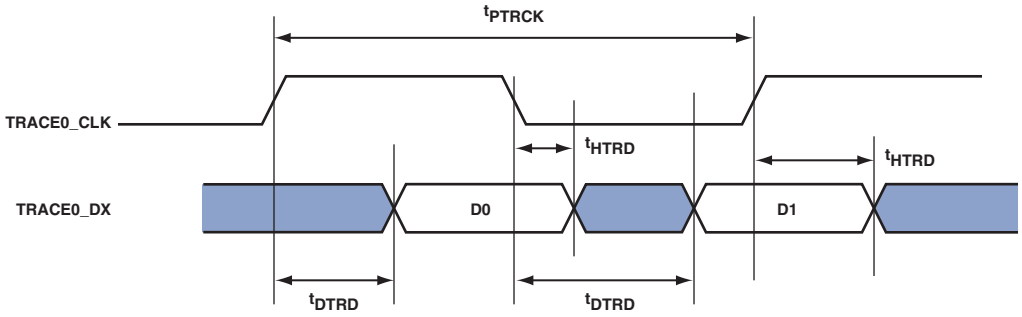


Figure 75. Trace Timing

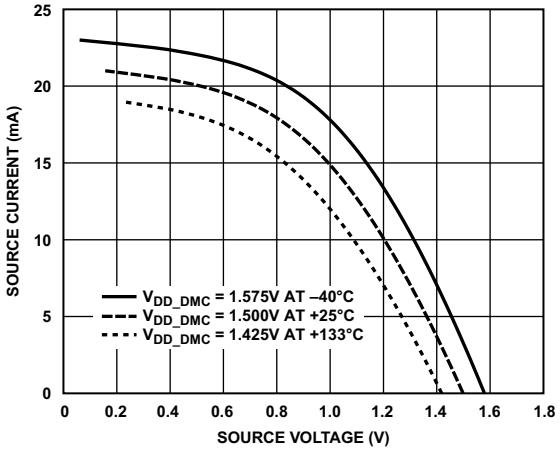


Figure 82. Driver Type B and Driver Type C (DDR3 Drive Strength 40 Ω)

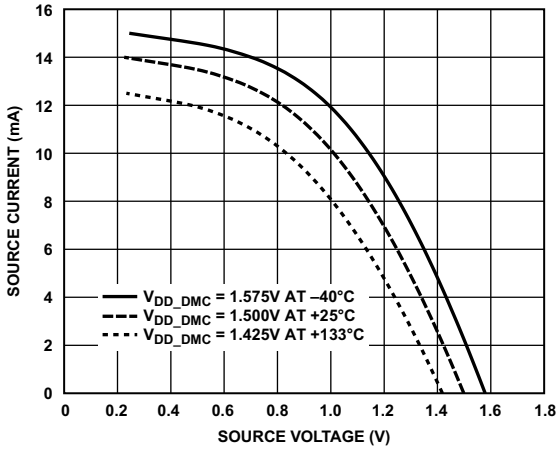


Figure 83. Driver Type B and Driver Type C (DDR3 Drive Strength 60 Ω)

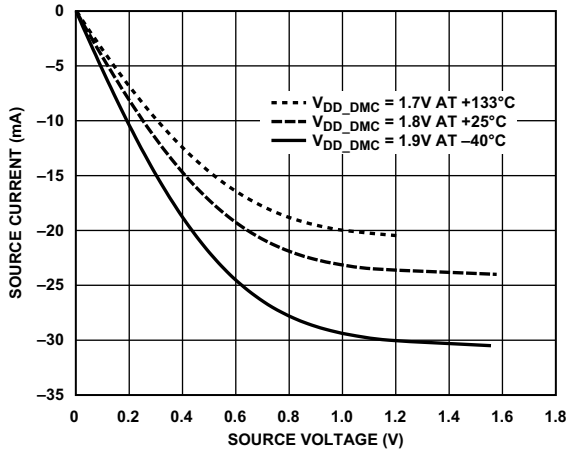


Figure 84. Driver Type B and Driver Type C (DDR2 Drive Strength 40 Ω)

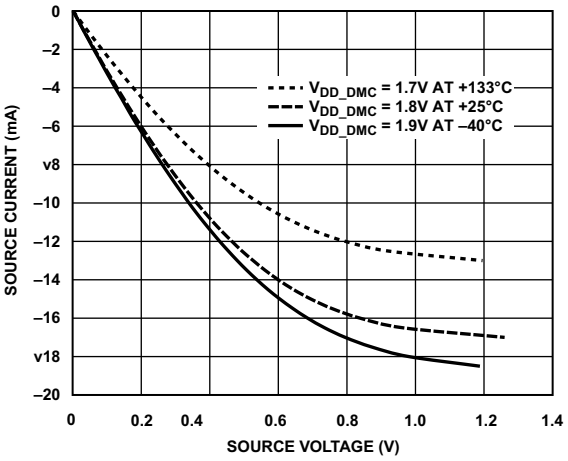


Figure 85. Driver Type B and Driver Type C (DDR2 Drive Strength 60Ω)

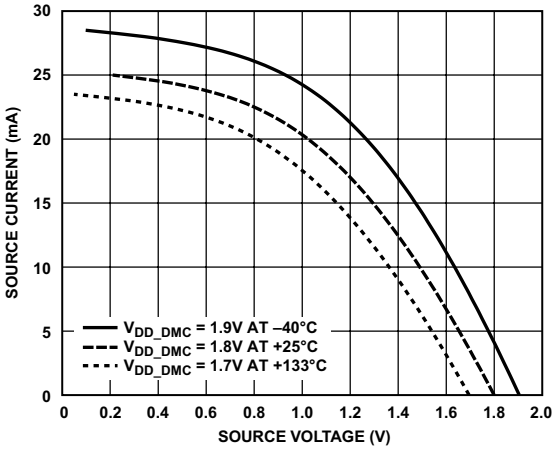


Figure 86. Driver Type B and Driver Type C (DDR2 Drive Strength 40Ω)

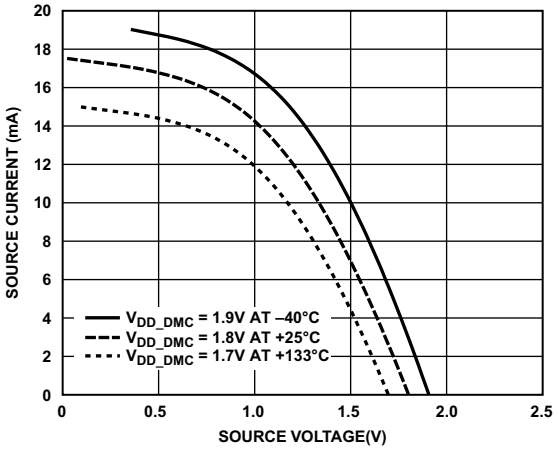


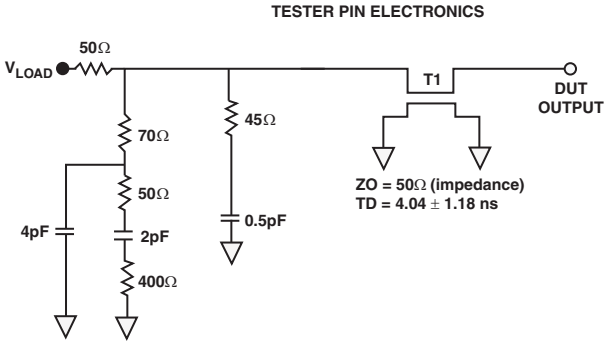
Figure 87. Driver Type B and Driver Type C (DDR2 Drive Strength 60 Ω)

Example System Hold Time Calculation

To determine the data output hold time in a particular system, first calculate t_{DECAY} using the previous equation. Choose ΔV to be the difference between the output voltage of the processor and the input threshold for the device requiring the hold time. C_L is the total bus capacitance (per data line) and I_L is the total leakage or three-state current (per data line). The hold time will be t_{DECAY} plus the various output disable times as specified in the Timing Specifications section.

Capacitive Loading

Output delays and holds are based on standard capacitive loads of an average of 6 pF on all balls (see Figure 92). V_{LOAD} is equal to $V_{DD\_EXT}/2$. Figure 93 through Figure 97 show how output rise time varies with capacitance. The delay and hold specifications given must be derated by a factor derived from these figures. The graphs in these figures may not be linear outside the ranges shown.



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

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

Figure 92. Equivalent Device Loading for AC Measurements (Includes All Fixtures)

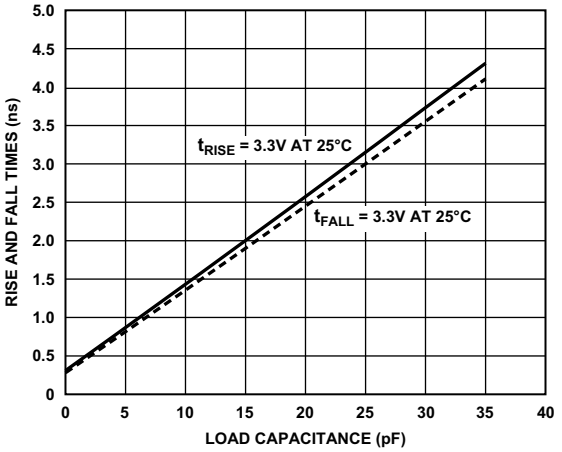


Figure 93. Driver Type A Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance (V_{DD_EXT} = 3.3 V)

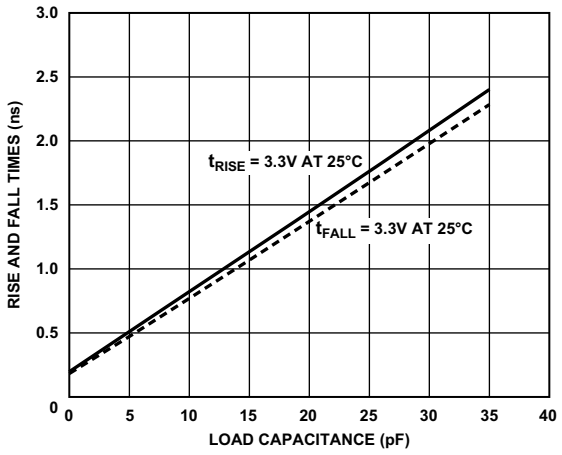


Figure 94. Driver Type H Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance (V_{DD_EXT} = 3.3 V)

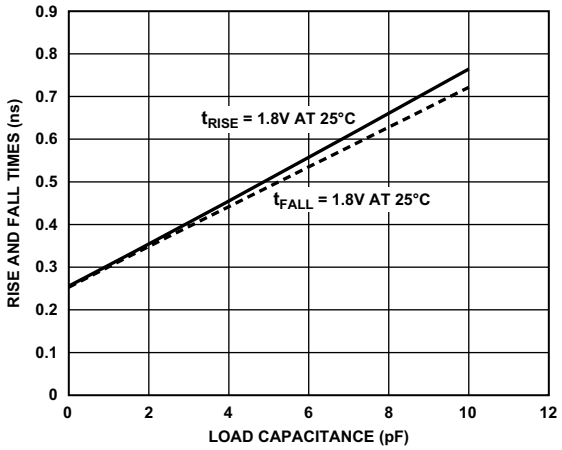


Figure 95. Driver Type B and Driver Type C Typical Rise and Fall Times (10% to 90%) vs. Load Capacitance (V_{DD_DMC} = 1.8 V) for LPDDR