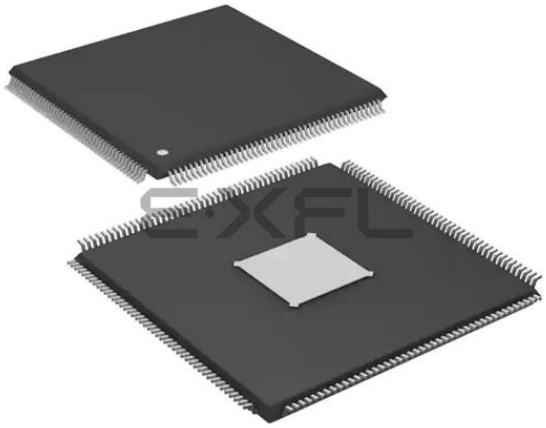


Welcome to E-XFL.COM

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	Fixed/Floating Point
Interface	CAN, EBI/EMI, Ethernet, DAI, I²C, MMC/SD/SDIO, SPI, SPORT, UART/USART, USB OTG
Clock Rate	450MHz, 450MHz
Non-Volatile Memory	External
On-Chip RAM	1.768MB
Voltage - I/O	3.30V
Voltage - Core	1.10V
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	176-LQFP Exposed Pad
Supplier Device Package	176-LQFP-EP (24x24)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/adsp-sc571bswz-4

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.

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.

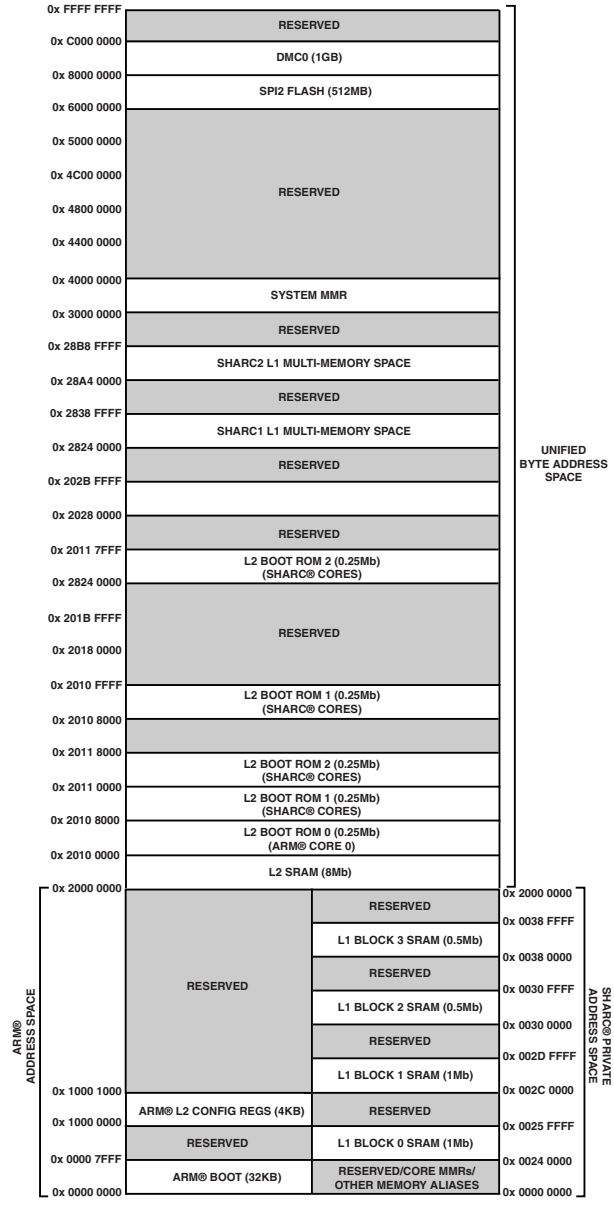


Figure 5. ADSP-SC57x/ADSP-2157x Memory Map

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.

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.

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

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

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

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

Signal Name	Description	Port	Pin Name
TRACE0_D03	TRACE0 Trace Data 3	F	PF_03
TRACE0_D04	TRACE0 Trace Data 4	D	PD_10
TRACE0_D05	TRACE0 Trace Data 5	D	PD_11
TRACE0_D06	TRACE0 Trace Data 6	D	PD_12
TRACE0_D07	TRACE0 Trace Data 7	D	PD_13
TWI0_SCL	TWI0 Serial Clock	Not Muxed	TWI0_SCL
TWI0_SDA	TWI0 Serial Data	Not Muxed	TWI0_SDA
TWI1_SCL	TWI1 Serial Clock	Not Muxed	TWI1_SCL
TWI1_SDA	TWI1 Serial Data	Not Muxed	TWI1_SDA
TWI2_SCL	TWI2 Serial Clock	Not Muxed	TWI2_SCL
TWI2_SDA	TWI2 Serial Data	Not Muxed	TWI2_SDA
$\overline{\text{UART0\_CTS}}$	UART0 Clear to Send	D	PD_06
$\overline{\text{UART0\_RTS}}$	UART0 Request to Send	D	PD_05
$\overline{\text{UART0\_RX}}$	UART0 Receive	F	PF_09
$\overline{\text{UART0\_TX}}$	UART0 Transmit	F	PF_08
$\overline{\text{UART1\_CTS}}$	UART1 Clear to Send	E	PE_14
$\overline{\text{UART1\_RTS}}$	UART1 Request to Send	E	PE_00
$\overline{\text{UART1\_RX}}$	UART1 Receive	F	PF_11
$\overline{\text{UART1\_TX}}$	UART1 Transmit	F	PF_10
$\overline{\text{UART2\_CTS}}$	UART2 Clear to Send	A	PA_11
$\overline{\text{UART2\_RTS}}$	UART2 Request to Send	A	PA_10
$\overline{\text{UART2\_RX}}$	UART2 Receive	C	PC_13
$\overline{\text{UART2\_TX}}$	UART2 Transmit	C	PC_12
USB0_CLKIN	USB0 Clock/Crystal Input	Not Muxed	USB_CLKIN
USB0_DM	USB0 Data –	Not Muxed	USB0_DM
USB0_DP	USB0 Data +	Not Muxed	USB0_DP
USB0_ID	USB0 OTG ID	Not Muxed	USB0_ID
USB0_VBC	USB0 VBUS Control	Not Muxed	USB0_VBC
USB0_VBUS	USB0 Bus Voltage	Not Muxed	USB0_VBUS
USB0_XTAL	USB0 Crystal	Not Muxed	USB_XTAL
VDD_EXT	External Voltage Domain	Not Muxed	VDD_EXT
VDD_INT	Internal Voltage Domain	Not Muxed	VDD_INT
VDD_DMC	DMC VDD	Not Muxed	VDD_DMC
VDD_HADC	HADC/TMU VDD	Not Muxed	VDD_HADC
VDD_USB	USB VDD	Not Muxed	VDD_USB

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

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

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

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

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

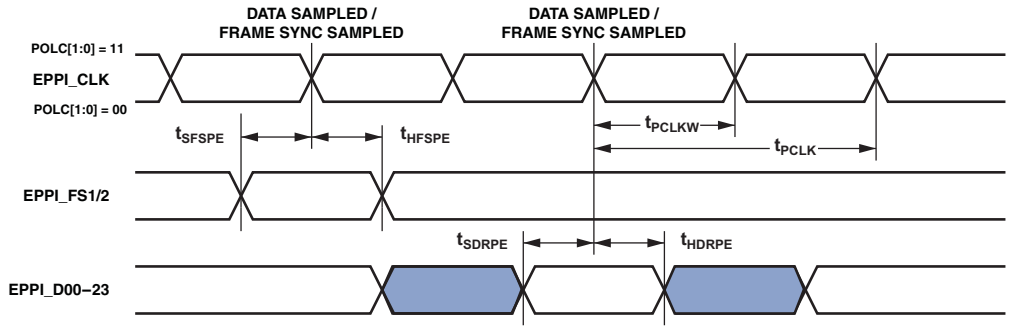


Figure 26. EPPI External Clock GP Receive Mode with External Frame Sync Timing

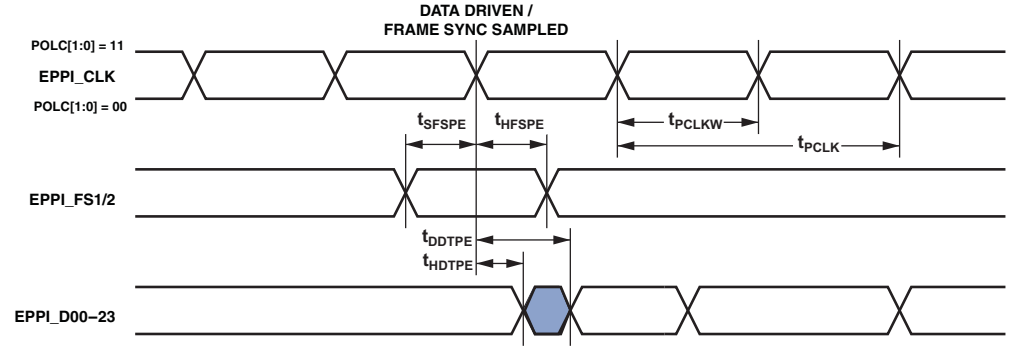
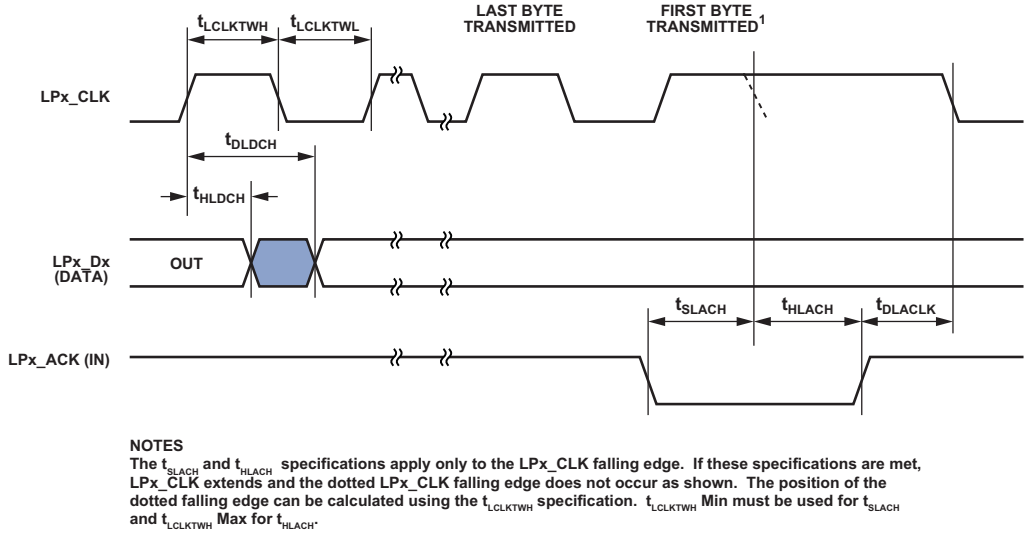


Figure 27. EPPI External Clock GP Transmit Mode with External Frame Sync Timing

Table 55. LPs—Transmit[1]

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

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



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

Figure 29. LPs—Transmit

SPI Port—SPIx_RDY Master Timing

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

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

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

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

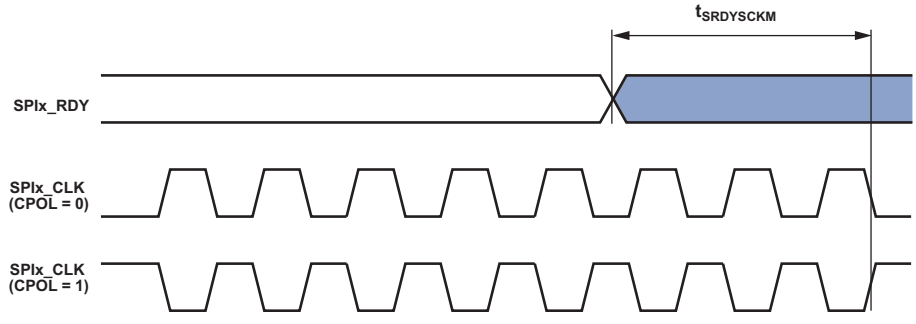


Figure 41. SPIx_RDY Setup Before SPIx_CLK

10/100 EMAC Timing

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

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

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

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

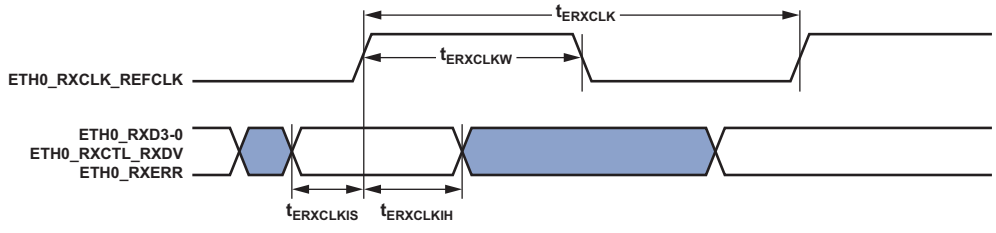


Figure 49. 10/100 EMAC Timing: MII Receive Signal

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

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

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

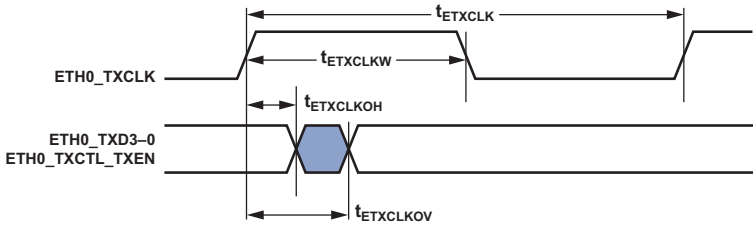


Figure 50. 10/100 EMAC Timing: MII Transmit Signal

10/100/1000 EMAC Timing

Table 84 and Figure 54 describe the RGMII EMAC timing.

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

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

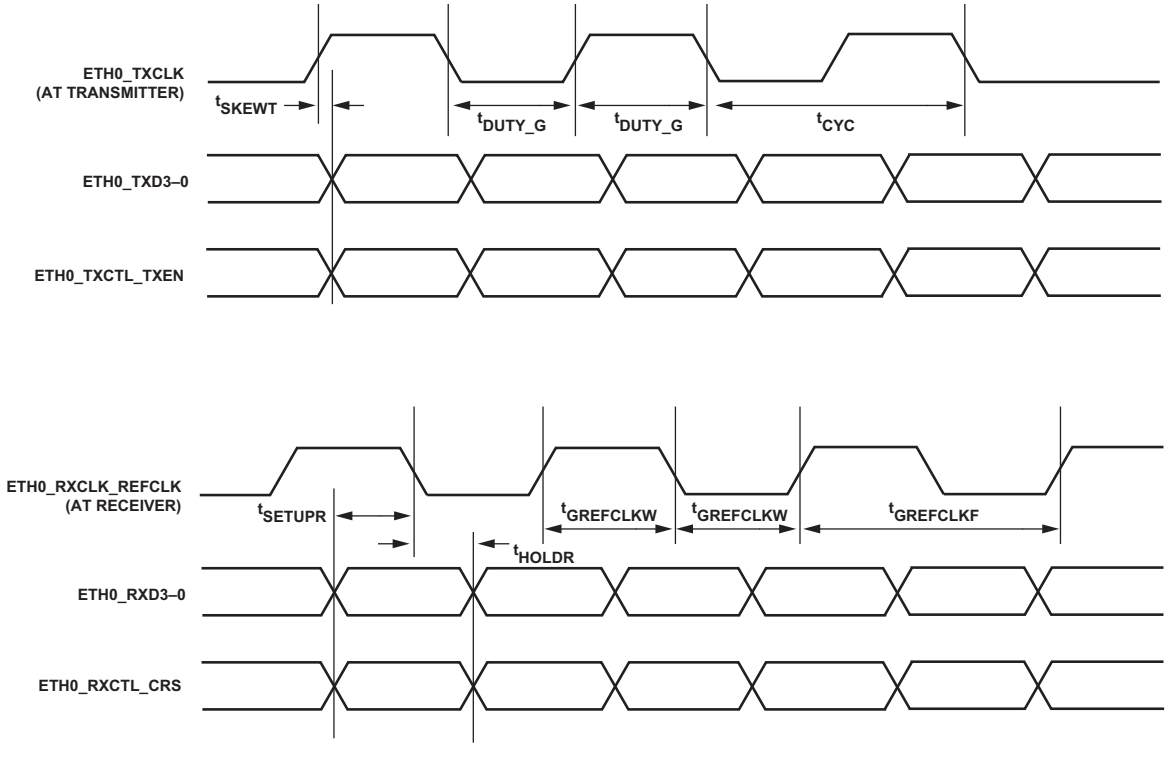


Figure 54. EMAC Timing—RGMII

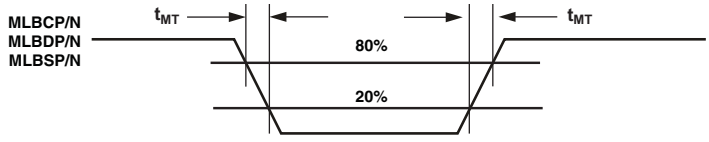


Figure 61. MLB 6-Pin Transition Time

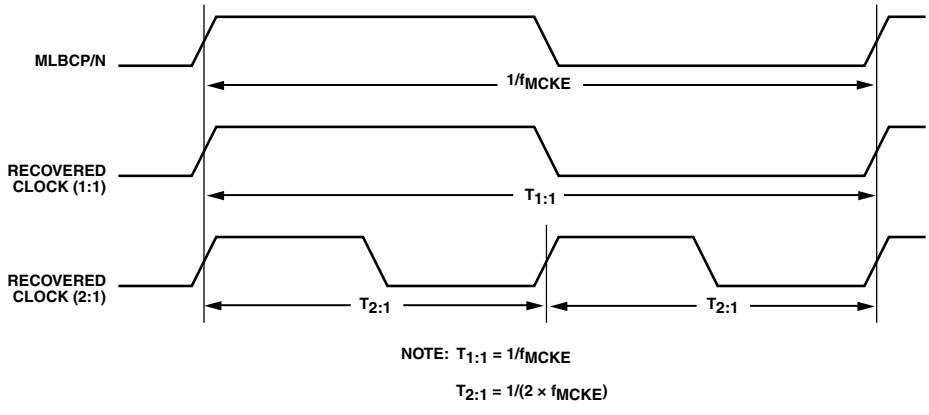


Figure 62. MLB 6-Pin Clock Definitions

Mobile Storage Interface (MSI) Controller Timing

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

Table 93. MSI Controller Timing

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

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

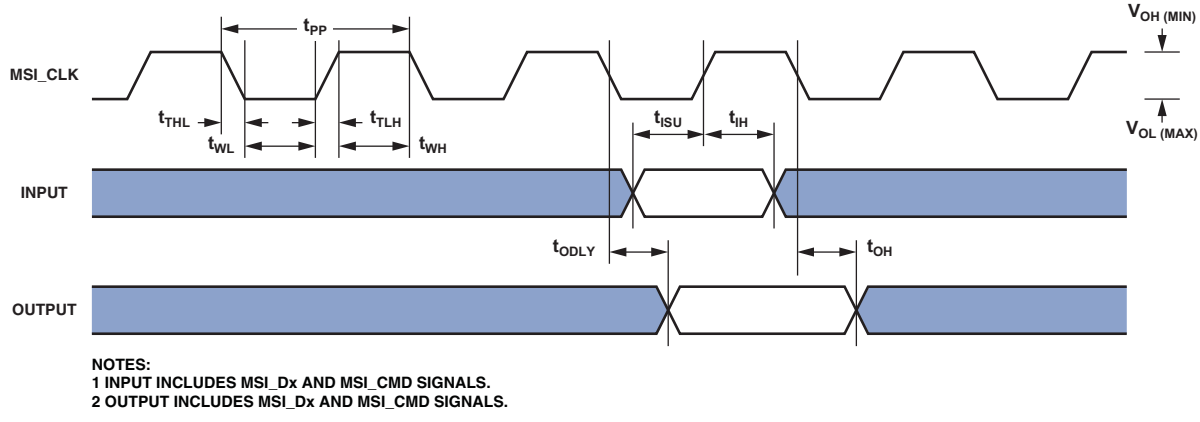


Figure 65. MSI Controller Timing

OUTPUT DRIVE CURRENTS

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

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

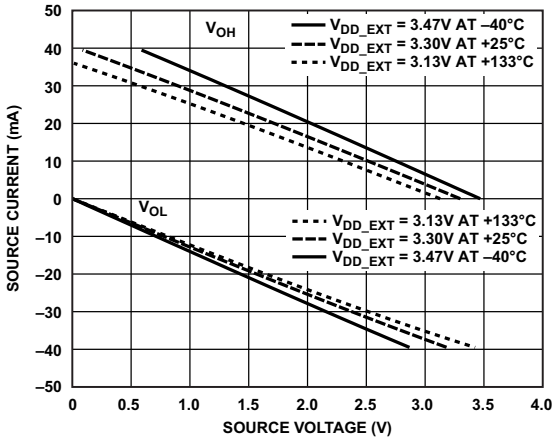


Figure 68. Driver Type A Current (3.3 V V_{DD_EXT})

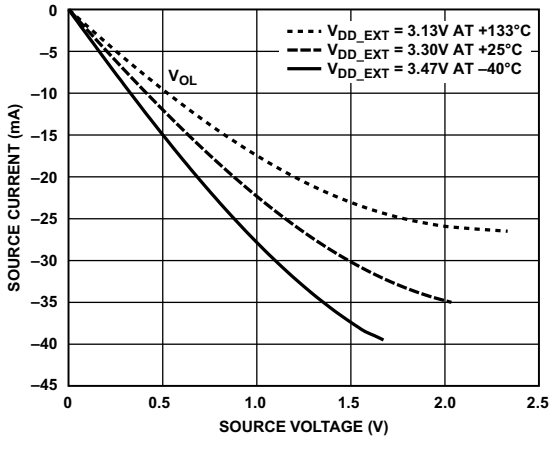


Figure 69. Driver Type D Current (3.3 V V_{DD_EXT})

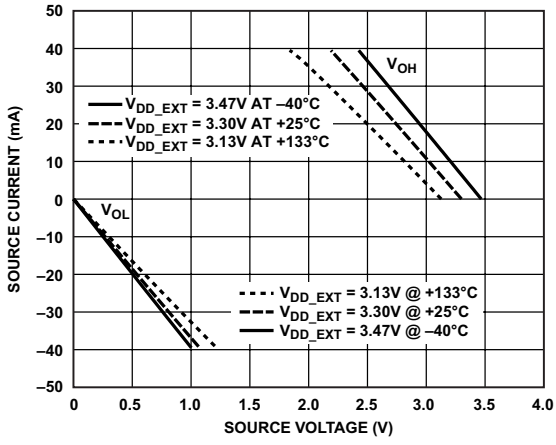


Figure 70. Driver Type H Current (3.3 V V_{DD_EXT})

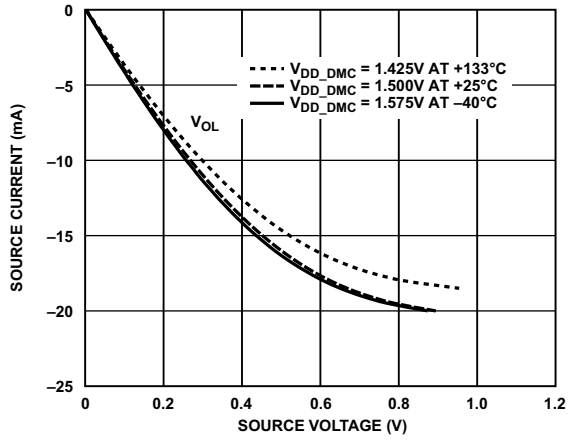


Figure 71. Driver Type B and Driver Type C (DDR3 Drive Strength 40 Ω)

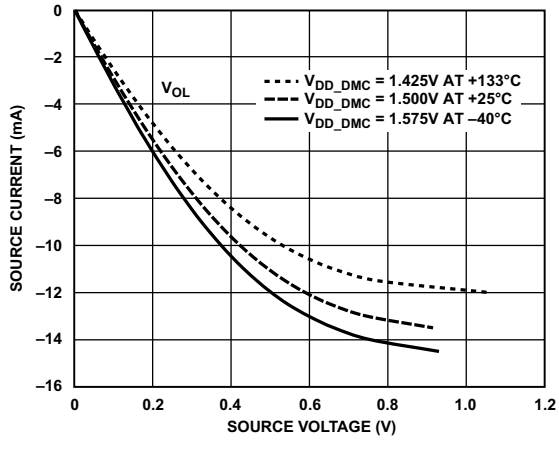


Figure 72. Driver Type B and Driver Type C (DDR3 Drive Strength 60 Ω)

Ball No.	Pin Name
V17	DMC0_BA1
V18	GND
V19	DMC0_A04
V20	DMC0_A05
W01	TWI2_SCL
W02	GND
W03	DMC0_DQ12
W04	DMC0_DQ11
W05	DMC0_DQ09
W06	PD_02
W07	PD_00
W08	PA_07
W09	PA_06
W10	PA_04
W11	DMC0_DQ05
W12	DMC0_DQ04
W13	DMC0_DQ03
W14	DMC0_DQ02
W15	$\overline{\text{SYS\_FAULT}}$
W16	DMC0_ODT
W17	DMC0_A08
W18	SYS_BMODE1
W19	GND
W20	DMC0_A07
Y01	GND
Y02	$\overline{\text{DMC0\_UDQS}}$
Y03	DMC0_UDQS
Y04	DMC0_DQ10
Y05	DMC0_DQ08
Y06	DMC0_UDM
Y07	DMC0_LDM
Y08	DMC0_CK
Y09	$\overline{\text{DMC0\_CK}}$
Y10	DMC0_DQ07
Y11	DMC0_DQ06
Y12	DMC0_LDQS
Y13	$\overline{\text{DMC0\_LDQS}}$
Y14	DMC0_DQ01
Y15	DMC0_DQ00
Y16	DMC0_CKE
Y17	$\overline{\text{DMC0\_CS0}}$
Y18	SYS_BMODE0
Y19	SYS_BMODE2
Y20	GND

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

Pin Name	Ball No.	Pin Name	Ball No.	Pin Name	Ball No.	Pin Name	Ball No.
DAI0_PIN01	D19	DMC0_DQ02	W14	GND	G11	GND	M06
DAI0_PIN02	F19	DMC0_DQ03	W13	GND	G12	GND	M07
DAI0_PIN03	E20	DMC0_DQ04	W12	GND	G13	GND	M08
DAI0_PIN04	D20	DMC0_DQ05	W11	GND	G14	GND	M09
DAI0_PIN05	H18	DMC0_DQ06	Y11	GND	G15	GND	M10
DAI0_PIN06	F20	DMC0_DQ07	Y10	GND	H06	GND	M11
DAI0_PIN07	E19	DMC0_DQ08	Y05	GND	H07	GND	M12
DAI0_PIN08	G18	DMC0_DQ09	W05	GND	H08	GND	M13
DAI0_PIN09	G20	DMC0_DQ10	Y04	GND	H09	GND	M14
DAI0_PIN10	G19	DMC0_DQ11	W04	GND	H10	GND	M15
DAI0_PIN11	H20	DMC0_DQ12	W03	GND	H11	GND	N06
DAI0_PIN12	J18	DMC0_DQ13	V02	GND	H12	GND	N07
DAI0_PIN13	J19	DMC0_DQ14	U02	GND	H13	GND	N08
DAI0_PIN14	H19	DMC0_DQ15	U01	GND	H14	GND	N09
DAI0_PIN15	K18	DMC0_LDM	Y07	GND	H15	GND	N10
DAI0_PIN16	J20	DMC0_LDQS	Y12	GND	J06	GND	N11
DAI0_PIN17	L18	$\overline{\text{DMC0\_LDQS}}$	Y13	GND	J07	GND	N12
DAI0_PIN18	K20	DMC0_ODT	W16	GND	J08	GND	N13
DAI0_PIN19	K19	$\overline{\text{DMC0\_RAS}}$	V16	GND	J09	GND	N14
DAI0_PIN20	L20	$\overline{\text{DMC0\_RESET}}$	T20	GND	J10	GND	N15
DMC0_A00	U20	DMC0_RZQ	P02	GND	J11	GND	N20
DMC0_A01	T19	DMC0_UDM	Y06	GND	J12	GND	P07
DMC0_A02	T18	DMC0_UDQS	Y03	GND	J13	GND	P14
DMC0_A03	U19	$\overline{\text{DMC0\_UDQS}}$	Y02	GND	J14	GND	R06
DMC0_A04	V19	DMC0_VREF	P01	GND	J15	GND	R15
DMC0_A05	V20	$\overline{\text{DMC0\_WE}}$	U16	GND	K06	GND	T05
DMC0_A06	U18	GND	A01	GND	K07	GND	T16
DMC0_A07	W20	GND	A09	GND	K08	GND	U04
DMC0_A08	W17	GND	A12	GND	K09	GND	U17
DMC0_A09	P03	GND	A15	GND	K10	GND	V03
DMC0_A10	P04	GND	A20	GND	K11	GND	V18
DMC0_A11	N01	GND	B02	GND	K12	GND	W02
DMC0_A12	N03	GND	B19	GND	K13	GND	W19
DMC0_A13	N02	GND	C03	GND	K14	GND	Y01
DMC0_A14	M01	GND	C18	GND	K15	GND	Y20
DMC0_A15	M02	GND	D04	GND	L06	HADC0_VIN0	P18
DMC0_BA0	R18	GND	D17	GND	L07	HADC0_VIN1	P17
DMC0_BA1	V17	GND	E05	GND	L08	HADC0_VIN2	R19
DMC0_BA2	U15	GND	E16	GND	L09	HADC0_VIN3	N19
$\overline{\text{DMC0\_CAS}}$	V15	GND	F06	GND	L10	HADC0_VIN4	N18
DMC0_CK	Y08	GND	F15	GND	L11	HADC0_VIN5	M19
DMC0_CKE	Y16	GND	G06	GND	L12	HADC0_VIN6	M20
$\overline{\text{DMC0\_CK}}$	Y09	GND	G07	GND	L13	HADC0_VIN7	M18
$\overline{\text{DMC0\_CS0}}$	Y17	GND	G08	GND	L14	HADC0_VREFN	P20
DMC0_DQ00	Y15	GND	G09	GND	L15	HADC0_VREFP	P19
DMC0_DQ01	Y14	GND	G10	GND	L19	JTG_TCK	E14