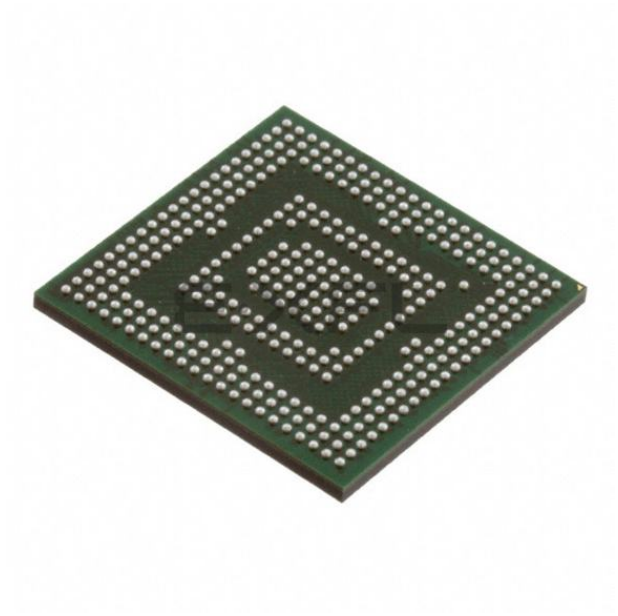


Welcome to E-XFL.COM

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	500MHz
Non-Volatile Memory	ROM (512kB)
On-Chip RAM	640kB
Voltage - I/O	3.30V
Voltage - Core	1.10V
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	349-LFBGA, CSPBGA
Supplier Device Package	349-CSPBGA (19x19)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/adsp-21584kbcz-5a

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

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.

3-Phase Pulse Width Modulator (PWM) Units

The pulse width modulator (PWM) module is a flexible and programmable waveform generator. With minimal CPU intervention, the PWM generates complex waveforms for motor control, pulse coded modulation (PCM), DAC conversions, power switching, and power conversion. The PWM module has four PWM pairs capable of 3-phase PWM generation for source inverters for ac induction and dc brushless motors.

Each of the three 3-phase PWM generation units features the following:

- 16-bit center-based PWM generation unit
- Programmable PWM pulse width
- Single update mode with an option for asymmetric duty
- Programmable dead time and switching frequency
- Programmable dead time per channel
- Twos complement implementation which permits smooth transition to full on and full off states
- Dedicated asynchronous PWM shutdown signal

Ethernet Media Access Controller (EMAC)

The processor features two ethernet media access controllers (EMACs): 10/100 Ethernet and 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 10-BaseT (10 Mb/sec), 100-BaseT (100 Mb/sec) and 1000-BaseT (1 Gb/sec) operations. The 10/100 EMAC peripheral on the processors is fully compliant to the IEEE 802.3-2002 standard. The peripheral provides programmable features designed to minimize supervision, bus use, or message processing by the rest of the processor system.

Some standard features of the EMAC are as follows:

- Support and RMII/RGMII protocols for external PHYs
- 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 are as follows:

- 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 (10/100/1000 EMAC Only)

The 10/100/1000 EMAC supports the following audio video (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 are as follows:

- 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

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.

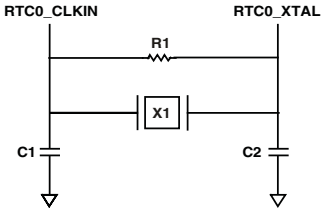
The reset source is defined as the following:

- 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).

Real-Time Clock (RTC)

The real-time clock (RTC) provides a robust set of digital watch features, including current time, stopwatch, and alarm. The RTC is clocked by a 32.768 kHz crystal external to the processor. Connect the RTC0_CLKIN and RTC0_XTAL pins with external components as shown in Figure 6.

The RTC peripheral has dedicated power supply pins so it can remain powered up and clocked even when the remainder of the processor is in a low power state. The RTC provides several programmable interrupt options, including interrupt per second, minute, hour, or day clock ticks; interrupt on programmable stopwatch countdown; or interrupt at a programmed alarm time.



NOTE: C1 AND C2 ARE SPECIFIC TO CRYSTAL SPECIFIED FOR X1. CONTACT CRYSTAL MANUFACTURER FOR DETAILS.

Figure 6. External Components for RTC

The 32.768 kHz input clock frequency is divided down to a 1 Hz signal by a prescaler. The counter function of the timer consists of four counters: a 60 second counter, a 60 minute counter, a 24 hour counter, and a 32,768 day counter. When the alarm interrupt is enabled, the alarm function generates an interrupt when the output of the timer matches the programmed value in the alarm control register (RTC_ALARM). There are two alarms: a time of day and a day and time of that day.

The stopwatch function counts down from a programmed value, with 1 sec resolution. When the stopwatch interrupt is enabled and the counter underflows, an interrupt is generated.

Clock Generation Unit (CGU)

The ADSP-SC58x/ADSP-2158x processors support two independent PLLs. Each PLL is part of a clock generation unit (CGU); see Figure 8. 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-SC58x/ADSP-2158x 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 7), a sine wave input, or a buffered, shaped clock derived from an external clock oscillator. If using an external clock, it should be a TTL-compatible signal and must not be halted, changed, or operated below the specified frequency during normal operation. This signal is connected to the SYS_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.

For fundamental frequency operation, use the circuit shown in Figure 7. 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 7, fine tune phase and amplitude of the sine frequency. The capacitor and resistor values shown in Figure 7 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.

Table 11. ADSP-SC58x/ADSP-2158x Detailed Signal Descriptions (Continued)

Signal Name	Direction	Description
DMC_UDQS	InOut	**Data Strobe for Upper Byte (Complement).** Complement of UDQS. Not used in single-ended mode.
DMC_VREF	Input	**Voltage Reference.** Externally driven to VDD_DMC/2.
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_CRS	Input	**Carrier Sense/RMII Receive Data Valid.** Multiplexed on alternate clock cycles. CRS— asserted by the PHY when either the transmit or receive medium is not idle. Deasserted when both are idle. RXDV—asserted by the PHY when the data on RXDn is valid.
ETH_MDC	Output	**Management Channel Clock.** Clocks the MDC input of the PHY.
ETH_MDIO	InOut	**Management Channel Serial Data.** Bidirectional data bus for PHY control.
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, PTPPPS is asserted every time the seconds counter is incremented.
ETH_REFCLK	Input	**Reference Clock.** Externally supplied Ethernet clock.
ETH_RXCLK_REFCLK	Input	**RXCLK (GigE) or REFCLK (10/100).**
ETH_RXCTL_CRS	Input	**RXCTL (GigE) or CRS (10/100).**
ETH_RXD[n]	Input	**Receive Data n.** Receive data bus.
ETH_TXCLK	Output	**Transmit Clock.**
ETH_TXCTL_TXEN	Output	**TXCTL (GigE) or TXEN (10/100).**
ETH_TXD[n]	Output	**Transmit Data n.** Transmits data bus.
ETH_TXEN	Output	**Transmit Enable.** When asserted, signal indicates the data on TXDn is valid.
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_MUX[n]	Input	**Controls to External Multiplexer.** Allows additional input channels when connected to an external multiplexer.
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_CLKN	Input	**Differential Clock (–).**
MLB_CLKP	Input	**Differential Clock (+).**
MLB_DATN	InOut	**Differential Data (–).**
MLB_DATP	InOut	**Differential Data (+).**
MLB_SIGN	InOut	**Differential Signal (–).**

Table 19. ADSP-SC58x/ADSP-2158x 529-Ball CSP_BGA Signal Descriptions (Continued)

Signal Name	Description	Port	Pin Name
VDD_PCIE	PCIE Supply Voltage	Not Muxed	VDD_PCIE
VDD_PCIE_RX	PCIE RX Supply Voltage	Not Muxed	VDD_PCIE_RX
VDD_PCIE_TX	PCIE TX Supply Voltage	Not Muxed	VDD_PCIE_TX
VDD_RTC	RTC VDD	Not Muxed	VDD_RTC
VDD_USB	USB VDD	Not Muxed	VDD_USB

ADSP-SC58X/ADSP-2158X DESIGNER QUICK REFERENCE

Table 27 provides a quick reference summary of pin related information for circuit board design. The columns in this table provide the following information:

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

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

Signal Name	Type	Driver Type	Int Term	Reset Term	Reset Drive	Power Domain	Description and Notes
DAI0_PIN01	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI0 Pin 1 Notes: No notes
DAI0_PIN02	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI0 Pin 2 Notes: No notes
DAI0_PIN03	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI0 Pin 3 Notes: No notes
DAI0_PIN04	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI0 Pin 4 Notes: No notes
DAI0_PIN05	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI0 Pin 5 Notes: No notes
DAI0_PIN06	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI0 Pin 6 Notes: No notes
DAI0_PIN07	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI0 Pin 7 Notes: No notes
DAI0_PIN08	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI0 Pin 8 Notes: No notes
DAI0_PIN09	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI0 Pin 9 Notes: No notes
DAI0_PIN10	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI0 Pin 10 Notes: No notes
DAI0_PIN11	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI0 Pin 11 Notes: No notes
DAI0_PIN12	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI0 Pin 12 Notes: No notes
DAI0_PIN13	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI0 Pin 13 Notes: No notes
DAI0_PIN14	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI0 Pin 14 Notes: No notes
DAI0_PIN15	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI0 Pin 15 Notes: No notes

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
DAI0_PIN16	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI0 Pin 16 Notes: No notes
DAI0_PIN17	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI0 Pin 17 Notes: No notes
DAI0_PIN18	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI0 Pin 18 Notes: No notes
DAI0_PIN19	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI0 Pin 19 Notes: No notes
DAI0_PIN20	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI0 Pin 20 Notes: No notes
DAI1_PIN01	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI1 Pin 1 Notes: No notes
DAI1_PIN02	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI1 Pin 2 Notes: No notes
DAI1_PIN03	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI1 Pin 3 Notes: No notes
DAI1_PIN04	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI1 Pin 4 Notes: No notes
DAI1_PIN05	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI1 Pin 5 Notes: No notes
DAI1_PIN06	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI1 Pin 6 Notes: No notes
DAI1_PIN07	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI1 Pin 7 Notes: No notes
DAI1_PIN08	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI1 Pin 8 Notes: No notes
DAI1_PIN09	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI1 Pin 9 Notes: No notes
DAI1_PIN10	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI1 Pin 10 Notes: No notes
DAI1_PIN11	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI1 Pin 11 Notes: No notes
DAI1_PIN12	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI1 Pin 12 Notes: No notes
DAI1_PIN13	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI1 Pin 13 Notes: No notes
DAI1_PIN14	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI1 Pin 14 Notes: No notes
DAI1_PIN15	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI1 Pin 15 Notes: No notes
DAI1_PIN16	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI1 Pin 16 Notes: No notes
DAI1_PIN17	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI1 Pin 17 Notes: No notes
DAI1_PIN18	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI1 Pin 18 Notes: No notes
DAI1_PIN19	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI1 Pin 19 Notes: No notes

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
DAI1_PIN20	InOut	A	PullDown	none	none	VDD_EXT	Desc: DAI1 Pin 20 Notes: No notes
DMC0_A00	Output	B	none	none	none	VDD_DMC	Desc: DMC0 Address 0 Notes: No notes
DMC0_A01	Output	B	none	none	none	VDD_DMC	Desc: DMC0 Address 1 Notes: No notes
DMC0_A02	Output	B	none	none	none	VDD_DMC	Desc: DMC0 Address 2 Notes: No notes
DMC0_A03	Output	B	none	none	none	VDD_DMC	Desc: DMC0 Address 3 Notes: No notes
DMC0_A04	Output	B	none	none	none	VDD_DMC	Desc: DMC0 Address 4 Notes: No notes
DMC0_A05	Output	B	none	none	none	VDD_DMC	Desc: DMC0 Address 5 Notes: No notes
DMC0_A06	Output	B	none	none	none	VDD_DMC	Desc: DMC0 Address 6 Notes: No notes
DMC0_A07	Output	B	none	none	none	VDD_DMC	Desc: DMC0 Address 7 Notes: No notes
DMC0_A08	Output	B	none	none	none	VDD_DMC	Desc: DMC0 Address 8 Notes: No notes
DMC0_A09	Output	B	none	none	none	VDD_DMC	Desc: DMC0 Address 9 Notes: No notes
DMC0_A10	Output	B	none	none	none	VDD_DMC	Desc: DMC0 Address 10 Notes: No notes
DMC0_A11	Output	B	none	none	none	VDD_DMC	Desc: DMC0 Address 11 Notes: No notes
DMC0_A12	Output	B	none	none	none	VDD_DMC	Desc: DMC0 Address 12 Notes: No notes
DMC0_A13	Output	B	none	none	none	VDD_DMC	Desc: DMC0 Address 13 Notes: No notes
DMC0_A14	Output	B	none	none	none	VDD_DMC	Desc: DMC0 Address 14 Notes: No notes
DMC0_A15	Output	B	none	none	none	VDD_DMC	Desc: DMC0 Address 15 Notes: No notes
DMC0_BA0	Output	B	none	none	none	VDD_DMC	Desc: DMC0 Bank Address Input 0 Notes: No notes
DMC0_BA1	Output	B	none	none	none	VDD_DMC	Desc: DMC0 Bank Address Input 1 Notes: No notes
DMC0_BA2	Output	B	none	none	none	VDD_DMC	Desc: DMC0 Bank Address Input 2 Notes: No notes
$\overline{\text{DMC0\_CAS}}$	Output	B	none	none	none	VDD_DMC	Desc: DMC0 Column Address Strobe Notes: No notes
DMC0_CK	Output	C	none	none	L	VDD_DMC	Desc: DMC0 Clock Notes: No notes

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
DMC0_UDQS	InOut	C	Internal logic ensures that input signal does not float	none	none	VDD_DMC	Desc: DMC0 Data Strobe for Upper Byte (complement) Notes: No notes
DMC0_VREF	a		none	none	none	VDD_DMC	Desc: DMC0 Voltage Reference Notes: No notes
DMC0_WE	Output	B	none	none	none	VDD_DMC	Desc: DMC0 Write Enable Notes: No notes
DMC1_A00	Output	B	none	none	none	VDD_DMC	Desc: DMC1 Address 0 Notes: No notes
DMC1_A01	Output	B	none	none	none	VDD_DMC	Desc: DMC1 Address 1 Notes: No notes
DMC1_A02	Output	B	none	none	none	VDD_DMC	Desc: DMC1 Address 2 Notes: No notes
DMC1_A03	Output	B	none	none	none	VDD_DMC	Desc: DMC1 Address 3 Notes: No notes
DMC1_A04	Output	B	none	none	none	VDD_DMC	Desc: DMC1 Address 4 Notes: No notes
DMC1_A05	Output	B	none	none	none	VDD_DMC	Desc: DMC1 Address 5 Notes: No notes
DMC1_A06	Output	B	none	none	none	VDD_DMC	Desc: DMC1 Address 6 Notes: No notes
DMC1_A07	Output	B	none	none	none	VDD_DMC	Desc: DMC1 Address 7 Notes: No notes
DMC1_A08	Output	B	none	none	none	VDD_DMC	Desc: DMC1 Address 8 Notes: No notes
DMC1_A09	Output	B	none	none	none	VDD_DMC	Desc: DMC1 Address 9 Notes: No notes
DMC1_A10	Output	B	none	none	none	VDD_DMC	Desc: DMC1 Address 10 Notes: No notes
DMC1_A11	Output	B	none	none	none	VDD_DMC	Desc: DMC1 Address 11 Notes: No notes
DMC1_A12	Output	B	none	none	none	VDD_DMC	Desc: DMC1 Address 12 Notes: No notes
DMC1_A13	Output	B	none	none	none	VDD_DMC	Desc: DMC1 Address 13 Notes: No notes
DMC1_A14	Output	B	none	none	none	VDD_DMC	Desc: DMC1 Address 14 Notes: No notes
DMC1_A15	Output	B	none	none	none	VDD_DMC	Desc: DMC1 Address 15 Notes: No notes
DMC1_BA0	Output	B	none	none	none	VDD_DMC	Desc: DMC1 Bank Address Input 0 Notes: No notes
DMC1_BA1	Output	B	none	none	none	VDD_DMC	Desc: DMC1 Bank Address Input 1 Notes: No notes

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
HADC0_VREFN	s	NA	none	none	none	VDD_HADC	Desc: HADC0 Ground Reference for ADC Notes: No notes
HADC0_VREFP	s	NA	none	none	none	VDD_HADC	Desc: HADC0 External Reference for ADC Notes: No notes
JTG_TCK	Input		PullUp	none	none	VDD_EXT	Desc: JTAG Clock Notes: No notes
JTG_TDI	Input		PullUp	none	none	VDD_EXT	Desc: JTAG Serial Data In Notes: No notes
JTG_TDO	Output	A	none	none	none	VDD_EXT	Desc: JTAG Serial Data Out Notes: No notes
JTG_TMS	InOut	A	PullUp	none	none	VDD_EXT	Desc: JTAG Mode Select Notes: No notes
$\overline{\text{JTG\_TRST}}$	Input		PullDown	none	none	VDD_EXT	Desc: JTAG Reset Notes: No notes
MLB0_CLKN	Input	NA	Internal logic ensures that input signal does not float	none	none	VDD_EXT	Desc: MLB0 Differential Clock (-) Notes: No notes
MLB0_CLKP	Input	NA	Internal logic ensures that input signal does not float	none	none	VDD_EXT	Desc: MLB0 Differential Clock (+) Notes: No notes
MLB0_DATN	InOut	I	Internal logic ensures that input signal does not float	none	none	VDD_EXT	Desc: MLB0 Differential Data (-) Notes: No notes
MLB0_DATP	InOut	I	Internal logic ensures that input signal does not float	none	none	VDD_EXT	Desc: MLB0 Differential Data (+) Notes: No notes
MLB0_SIGN	InOut	I	Internal logic ensures that input signal does not float	none	none	VDD_EXT	Desc: MLB0 Differential Signal (-) Notes: No notes
MLB0_SIGP	InOut	I	Internal logic ensures that input signal does not float	none	none	VDD_EXT	Desc: MLB0 Differential Signal (+) Notes: No notes
PA_00	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTA Position 0 | EMAC0 Transmit Data 0 | SMC0 Address 21 Notes: No notes
PA_01	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTA Position 1 | EMAC0 Transmit Data 1 | SMC0 Address 20 Notes: No notes

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.

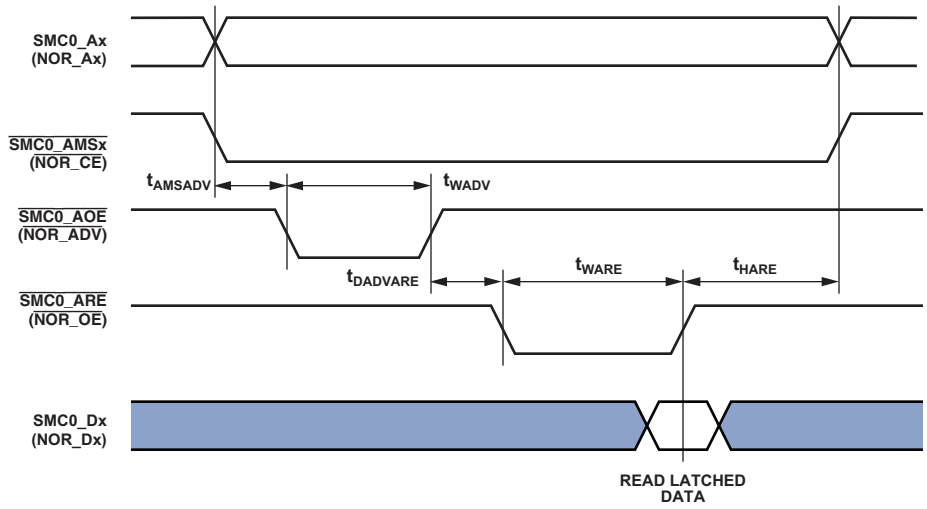


Figure 13. Asynchronous Flash Read

DDR2 SDRAM Read Cycle Timing

Table 52 and Figure 18 show DDR2 SDRAM read cycle timing, related to the DMC.

Table 52. DDR2 SDRAM Read Cycle Timing, V_{DD_DMCx} Nominal 1.8 V[1]

Parameter		400 MHz[2] Min	400 MHz[2] Max	Unit
Timing Requirements				
t_{DQSQ}	DMCx_DQS to DMCx_DQ Skew for DMCx_DQS and Associated DMCx_DQxx Signals		0.2	ns
t_{QH}	DMCx_DQxx, DMCx_DQS Output Hold Time From DMCx_DQS	0.9		ns
t_{RPRE}	Read Preamble	0.9		t_{CK}
t_{RPST}	Read Postamble	0.4		t_{CK}

[1] Specifications apply to both DMC0 and DMC1.
[2] In order to ensure proper operation of the DDR2, all the DDR2 guidelines must be strictly followed. See "Interfacing DDR3/DDR2/LPDDR Memory to ADSP-SC5xx/215xx Processors" (EE-387).

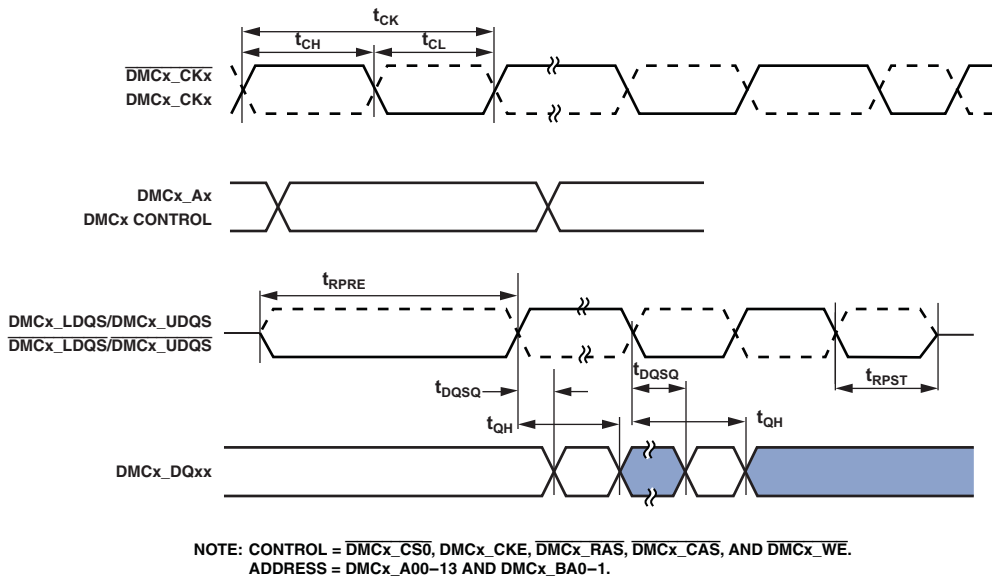


Figure 18. DDR2 SDRAM Controller Input AC Timing

Link Ports (LP)

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

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

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

$$f_{LCLKTPROG} = \frac{f_{CLKO8}}{(VALUE \times 2)}$$

In the case where VALUE = 0, $f_{LCLKTPROG} = f_{CLKO8}$. For all settings of VALUE, the following equation is true:

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

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

Table 62. Link Ports—Receive[1]

Parameter		Min	Max	Unit
Timing Requirements				
$f_{LCLKREXT}$	LPx_CLK Frequency		150	MHz
t_{SLDCL}	Data Setup Before LPx_CLK Low	0.9		ns
t_{HLDCL}	Data Hold After LPx_CLK Low	1.4		ns
t_{LCLKEW}	LPx_CLK Period[2]	$t_{LCLKREXT}$ – 0.42		ns
$t_{LCLKRWL}$	LPx_CLK Width Low[2]	0.5 × $t_{LCLKREXT}$		ns
$t_{LCLKRWH}$	LPx_CLK Width High[2]	0.5 × $t_{LCLKREXT}$		ns
Switching Characteristic				
t_{DLALC}	LPx_ACK Low Delay After LPx_CLK Low[3]	1.5 × t_{CLKO8} + 4	2.5 × t_{CLKO8} + 12	ns

[1] Specifications apply to LP0 and LP1.
[2] This specification indicates the minimum instantaneous width or period that can be tolerated due to duty cycle variation or jitter on the external LPx_CLK. For the external LPx_CLK ideal maximum frequency see the $f_{LCLKTEXT}$ specification in Table 29.
[3] LPx_ACK goes low with t_{DLALC} relative to rise of LPx_CLK after first byte, but does not go low if the link buffer of the receiver is not about to fill.

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 76. 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.

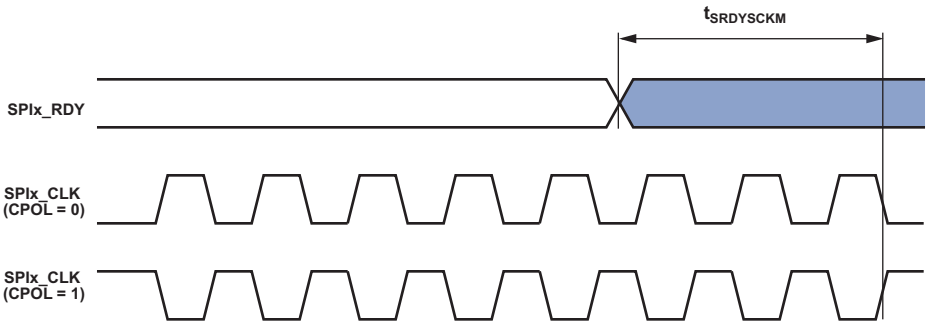


Figure 48. SPIx_RDY Setup Before SPIx_CLK

Precision Clock Generator (PCG) (Direct Pin Routing)

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

Table 77. Precision Clock Generator (Direct Pin Routing)

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

[1] D = FSxDIV, PH = FSxPHASE. For more information, see the ADSP-SC58x/ADSP-2158x SHARC+ Processor Hardware Reference.
[2] Normal mode of operation.

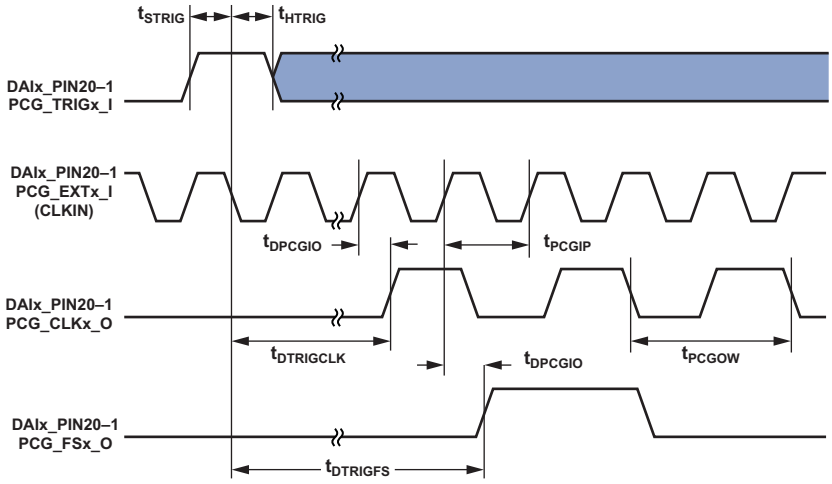


Figure 50. PCG (Direct Pin Routing)

Table 90. 10/100 EMAC Timing—RMII Station Management[1]

Parameter[2]		Min	Max	Unit
Timing Requirements				
t_{MDIOS}	ETHx_MDIO Input Valid to ETHx_MDC Rising Edge (Setup)	10.8		ns
t_{MDCIH}	ETHx_MDC Rising Edge to ETHx_MDIO Input Invalid (Hold)	0		ns
Switching Characteristics				
t_{MDCOV}	ETHx_MDC Falling Edge to ETHx_MDIO Output Valid		t_{SCLK0} + 2	ns
t_{MDCOH}	ETHx_MDC Falling Edge to ETHx_MDIO Output Invalid (Hold)	t_{SCLK0} –2.9		ns

[1] These specifications apply to ETH0 and ETH1.
[2] ETHx_MDC/ETHx_MDIO is a 2-wire serial bidirectional port for controlling one or more external PHYs. ETHx_MDC is an output clock with a minimum period that is programmable as a multiple of the system clock SCLK0. ETHx_MDIO is a bidirectional data line.

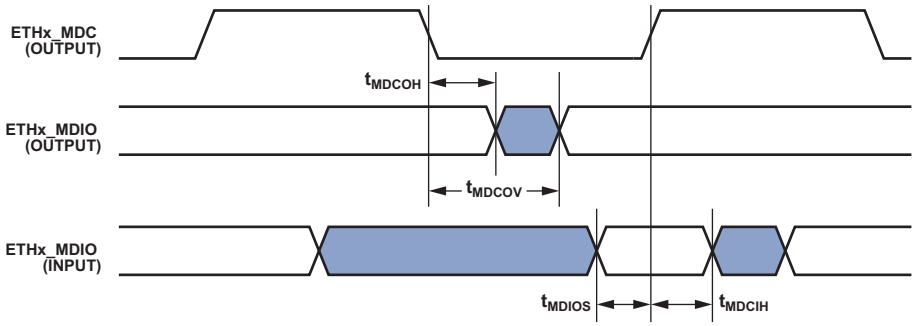


Figure 61. 10/100 Ethernet MAC Controller Timing—RMII Station Management

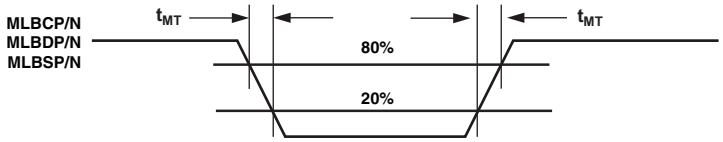


Figure 70. MLB 6-Pin Transition Time

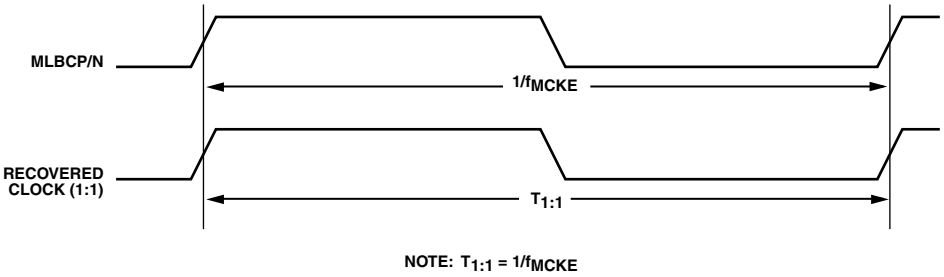


Figure 71. MLB 6-Pin Clock Definitions

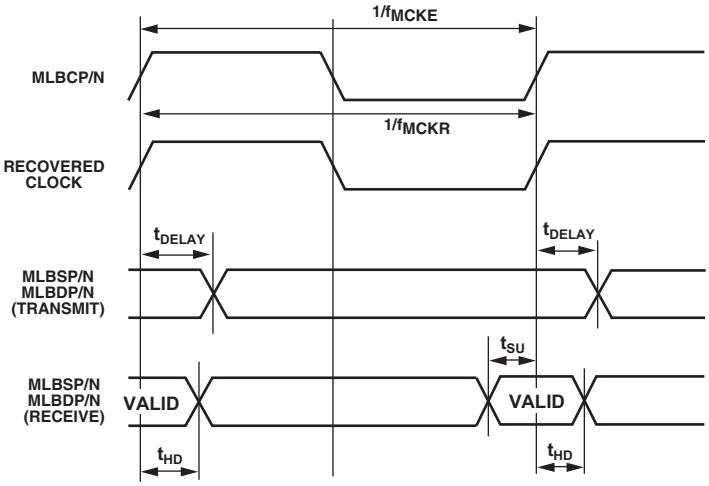


Figure 72. MLB 6-Pin Delay, Setup, and Hold Times

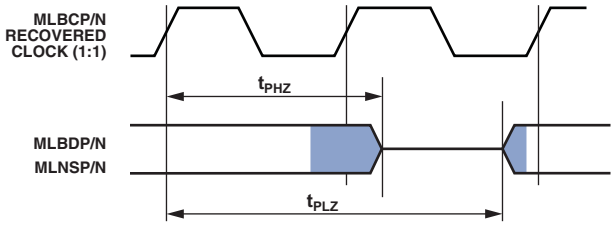


Figure 73. MLB 6-Pin Disable and Enable Turnaround Times

Mobile Storage Interface (MSI) Controller Timing

Table 101 and Figure 74 show I/O timing related to the MSI.

Table 101. 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]		50	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	ns
t_{OH}	Output Hold Time	–1.8		ns

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

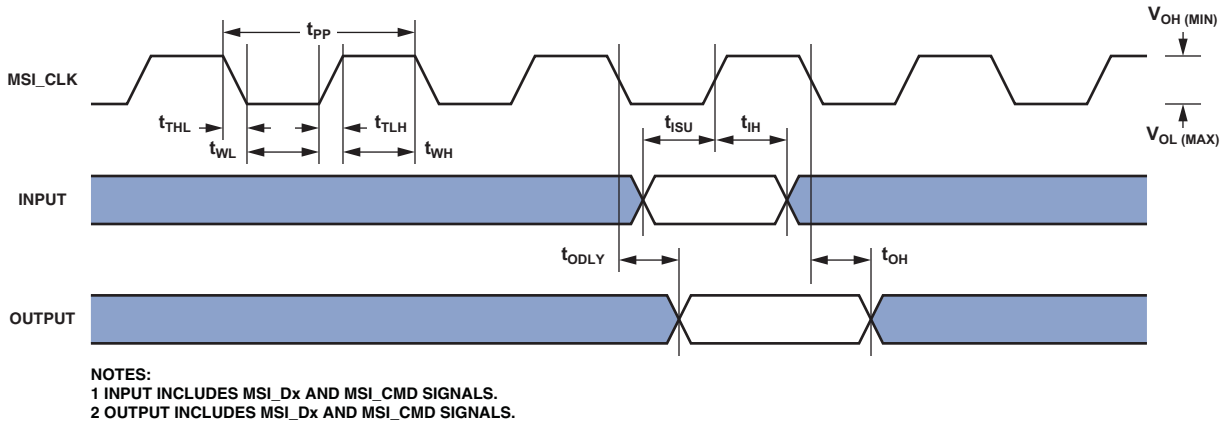


Figure 74. MSI Controller Timing

ADSP-SC58x/ADSP-2158x 529-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	AA06	DMC0_A06	F02	DMC1_A00	C08	DMC1_RZQ	E19
DAI0_PIN02	AB06	DMC0_A07	F05	DMC1_A01	C09	DMC1_UDM	C16
DAI0_PIN03	AB07	DMC0_A08	J03	DMC1_A02	D10	DMC1_UDQS	A20
DAI0_PIN04	AB05	DMC0_A09	G02	DMC1_A03	C07	$\overline{\text{DMC1\_UDQS}}$	A19
DAI0_PIN05	Y05	DMC0_A10	J02	DMC1_A04	C10	DMC1_VREF	A15
DAI0_PIN06	AA05	DMC0_A11	G04	DMC1_A05	D07	$\overline{\text{DMC1\_WE}}$	D08
DAI0_PIN07	AA04	DMC0_A12	H02	DMC1_A06	C11	GND	A01
DAI0_PIN08	Y04	DMC0_A13	G01	DMC1_A07	D09	GND	A23
DAI0_PIN09	AB03	DMC0_A14	H01	DMC1_A08	D12	GND	E06
DAI0_PIN10	Y06	DMC0_A15	J01	DMC1_A09	D17	GND	E07
DAI0_PIN11	W04	DMC0_BA0	D03	DMC1_A10	D19	GND	E08
DAI0_PIN12	V04	DMC0_BA1	E03	DMC1_A11	D20	GND	E09
DAI0_PIN13	AB04	DMC0_BA2	D04	DMC1_A12	C18	GND	F06
DAI0_PIN14	AB02	$\overline{\text{DMC0\_CAS}}$	F03	DMC1_A13	C19	GND	F09
DAI0_PIN15	AB01	DMC0_CK	A04	DMC1_A14	C17	GND	F16
DAI0_PIN16	AA03	DMC0_CKE	C04	DMC1_A15	D18	GND	J07
DAI0_PIN17	Y03	$\overline{\text{DMC0\_CK}}$	A03	DMC1_BA0	D11	GND	J08
DAI0_PIN18	W03	$\overline{\text{DMC0\_CS0}}$	C03	DMC1_BA1	C12	GND	J09
DAI0_PIN19	V03	DMC0_DQ00	B10	DMC1_BA2	D14	GND	J10
DAI0_PIN20	U04	DMC0_DQ01	A10	$\overline{\text{DMC1\_CAS}}$	D15	GND	J11
DAI1_PIN01	T23	DMC0_DQ02	B09	DMC1_CK	A16	GND	J12
DAI1_PIN02	U23	DMC0_DQ03	A09	DMC1_CKE	D13	GND	J13
DAI1_PIN03	T20	DMC0_DQ04	B08	$\overline{\text{DMC1\_CK}}$	A17	GND	J14
DAI1_PIN04	U21	DMC0_DQ05	A08	$\overline{\text{DMC1\_CS0}}$	C14	GND	J15
DAI1_PIN05	U22	DMC0_DQ06	B06	DMC1_DQ00	A12	GND	J16
DAI1_PIN06	V21	DMC0_DQ07	B07	DMC1_DQ01	B11	GND	J17
DAI1_PIN07	U20	DMC0_DQ08	B05	DMC1_DQ02	B12	GND	K07
DAI1_PIN08	U19	DMC0_DQ09	A05	DMC1_DQ03	A11	GND	K08
DAI1_PIN09	V23	DMC0_DQ10	B04	DMC1_DQ04	B13	GND	K09
DAI1_PIN10	W22	DMC0_DQ11	B03	DMC1_DQ05	B14	GND	K10
DAI1_PIN11	W21	DMC0_DQ12	B02	DMC1_DQ06	B15	GND	K11
DAI1_PIN12	V22	DMC0_DQ13	C02	DMC1_DQ07	B16	GND	K12
DAI1_PIN13	W23	DMC0_DQ14	C01	DMC1_DQ08	B17	GND	K13
DAI1_PIN14	Y21	DMC0_DQ15	D02	DMC1_DQ09	A18	GND	K14
DAI1_PIN15	Y23	DMC0_LDM	C05	DMC1_DQ10	B18	GND	K15
DAI1_PIN16	V20	DMC0_LDQS	A07	DMC1_DQ11	B19	GND	K16
DAI1_PIN17	Y22	$\overline{\text{DMC0\_LDQS}}$	A06	DMC1_DQ12	B20	GND	K17
DAI1_PIN18	AA23	DMC0_ODT	D05	DMC1_DQ13	A21	GND	L07
DAI1_PIN19	AA22	$\overline{\text{DMC0\_RAS}}$	E02	DMC1_DQ14	B21	GND	L08
DAI1_PIN20	W20	$\overline{\text{DMC0\_RESET}}$	K01	DMC1_DQ15	A22	GND	L09
DMC0_A00	H04	DMC0_RZQ	E05	DMC1_LDM	C15	GND	L10
DMC0_A01	F01	DMC0_UDM	D06	DMC1_LDQS	A13	GND	L11
DMC0_A02	F04	DMC0_UDQS	B01	$\overline{\text{DMC1\_LDQS}}$	A14	GND	L12
DMC0_A03	G03	$\overline{\text{DMC0\_UDQS}}$	A02	DMC1_ODT	C13	GND	L13
DMC0_A04	E01	DMC0_VREF	D01	$\overline{\text{DMC1\_RAS}}$	D16	GND	L14
DMC0_A05	H03	$\overline{\text{DMC0\_WE}}$	E04	$\overline{\text{DMC1\_RESET}}$	C06	GND	L15

Pin Name	Ball No.	Pin Name	Ball No.	Pin Name	Ball No.	Pin Name	Ball No.
PF_11	K21	VDD_DMC	G13	VDD_INT	E10	VDD_INT	W16
PF_12	P22	VDD_DMC	G14	VDD_INT	E11	VDD_INT	W17
PF_13	R23	VDD_DMC	G15	VDD_INT	E12	VDD_INT	W18
PF_14	J21	VDD_DMC	G16	VDD_INT	E13	VDD_INT	W19
PF_15	P21	VDD_DMC	G17	VDD_INT	E14	VDD_PCIE	W07
PG_00	P23	VDD_DMC	G18	VDD_INT	E15	VDD_PCIE_RX	V07
PG_01	R20	VDD_DMC	H06	VDD_INT	E16	VDD_PCIE_TX	V08
PG_02	T22	VDD_DMC	H07	VDD_INT	E17	VDD_RTC	W14
PG_03	T21	VDD_DMC	H08	VDD_INT	E18	VDD_USB	Y08
PG_04	R22	VDD_DMC	H09	VDD_INT	F07		
PG_05	R21	VDD_DMC	H10	VDD_INT	F08		
RTC0_CLKIN	AC15	VDD_DMC	H11	VDD_INT	F10		
RTC0_XTAL	AB15	VDD_DMC	H12	VDD_INT	F11		
SYS_BMODE0	R04	VDD_DMC	H13	VDD_INT	F12		
SYS_BMODE1	R02	VDD_DMC	H14	VDD_INT	F13		
SYS_BMODE2	R03	VDD_DMC	H15	VDD_INT	F14		
SYS_CLKIN0	V01	VDD_DMC	H16	VDD_INT	F15		
SYS_CLKIN1	T01	VDD_DMC	H17	VDD_INT	F17		
SYS_CLKOUT	H20	VDD_DMC	H18	VDD_INT	F18		
SYS_FAULT	P03	VDD_DMC	J06	VDD_INT	F19		
$\overline{\text{SYS\_FAULT}}$	M04	VDD_DMC	K06	VDD_INT	G05		
$\overline{\text{SYS\_HWRST}}$	N03	VDD_DMC	L06	VDD_INT	G19		
$\overline{\text{SYS\_RESOUT}}$	U02	VDD_DMC	M06	VDD_INT	H05		
SYS_XTAL0	U01	VDD_EXT	J18	VDD_INT	H19		
SYS_XTAL1	R01	VDD_EXT	K18	VDD_INT	J05		
TWI0_SCL	Y10	VDD_EXT	L18	VDD_INT	K05		
TWI0_SDA	AB11	VDD_EXT	M18	VDD_INT	K19		
TWI1_SCL	AA10	VDD_EXT	N06	VDD_INT	L05		
TWI1_SDA	AA11	VDD_EXT	N18	VDD_INT	L19		
TWI2_SCL	AB10	VDD_EXT	P06	VDD_INT	M05		
TWI2_SDA	Y11	VDD_EXT	P18	VDD_INT	N05		
USB0_DM	AC11	VDD_EXT	R06	VDD_INT	N19		
USB0_DP	AC10	VDD_EXT	R18	VDD_INT	P05		
USB0_ID	Y07	VDD_EXT	T06	VDD_INT	R05		
USB0_VBC	Y09	VDD_EXT	T18	VDD_INT	R19		
USB0_VBUS	AA09	VDD_EXT	U06	VDD_INT	T05		
USB1_DM	AC08	VDD_EXT	U18	VDD_INT	T19		
USB1_DP	AC09	VDD_EXT	V06	VDD_INT	U05		
USB1_VBUS	AA08	VDD_EXT	V09	VDD_INT	V05		
USB_CLKIN	AB09	VDD_EXT	V10	VDD_INT	V19		
USB_XTAL	AB08	VDD_EXT	V11	VDD_INT	W05		
VDD_DMC	G06	VDD_EXT	V13	VDD_INT	W06		
VDD_DMC	G07	VDD_EXT	V14	VDD_INT	W08		
VDD_DMC	G08	VDD_EXT	V15	VDD_INT	W09		
VDD_DMC	G09	VDD_EXT	V16	VDD_INT	W10		
VDD_DMC	G10	VDD_EXT	V17	VDD_INT	W11		
VDD_DMC	G11	VDD_EXT	V18	VDD_INT	W13		
VDD_DMC	G12	VDD_HADC	AC13	VDD_INT	W15		