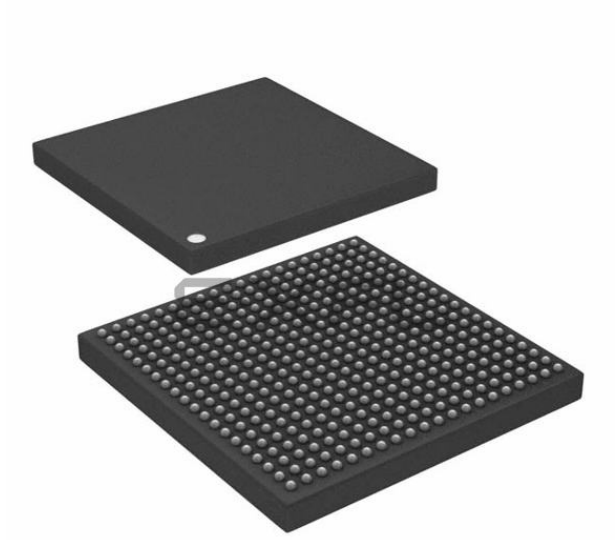


Welcome to E-XFL.COM

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
Non-Volatile Memory	External
On-Chip RAM	1.768MB
Voltage - I/O	3.30V
Voltage - Core	1.10V
Operating Temperature	-40°C ~ 100°C (TA)
Mounting Type	Surface Mount
Package / Case	400-LFBGA, CSPBGA
Supplier Device Package	400-CSPBGA (17x17)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/adsp-21573cbcz-4

Table 8. DMC Memory Map[1]

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

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

System Crossbars (SCBs)

The system crossbars (SCBs) are the fundamental building blocks of a switch fabric style for on-chip system bus interconnection. The SCBs connect system bus masters to system bus slaves, providing concurrent data transfer between multiple bus masters and multiple bus slaves. A hierarchical model—built from multiple SCBs—provides a power and area efficient system interconnection.

The SCBs provide the following features:

- Highly efficient, pipelined bus transfer protocol for sustained throughput
- Full-duplex bus operation for flexibility and reduced latency
- Concurrent bus transfer support to allow multiple bus masters to access bus slaves simultaneously
- Protection model (privileged/secure) support for selective bus interconnect protection

Direct Memory Access (DMA)

The processors use direct memory access (DMA) to transfer data within memory spaces or between a memory space and a peripheral. The processors can specify data transfer operations and return to normal processing while the fully integrated DMA controller carries out the data transfers independent of processor activity.

DMA transfers can occur between memory and a peripheral or between one memory and another memory. Each memory to memory DMA stream uses two channels: the source channel and the destination channel.

All DMA channels can transport data to and from all on-chip and off-chip memories. Programs can use two types of DMA transfers: descriptor-based or register-based. Register-based DMA allows the processors to program DMA control registers directly to initiate a DMA transfer. On completion, the DMA control registers automatically update with original setup values for continuous transfer. Descriptor-based DMA transfers require a set of parameters stored within memory to initiate a DMA sequence. Descriptor-based DMA transfers allow multiple DMA sequences to be chained together. Program a DMA channel to set up and start another DMA transfer automatically after the current sequence completes.

The DMA engine supports the following DMA operations:

- A single linear buffer that stops on completion
- A linear buffer with negative, positive, or zero stride length
- A circular autorefreshing buffer that interrupts when each buffer becomes full
- A similar circular buffer that interrupts on fractional buffers, such as at the halfway point
- The 1D DMA uses a set of identical ping pong buffers defined by a linked ring of two-word descriptor sets, each containing a link pointer and an address
- The 1D DMA uses a linked list of four-word descriptor sets containing a link pointer, an address, a length, and a configuration
- The 2D DMA uses an array of one-word descriptor sets, specifying only the base DMA address
- The 2D DMA uses a linked list of multiword descriptor sets, specifying all configurable parameters

Memory Direct Memory Access (MDMA)

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

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

Extended Memory DMA

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

Cyclic Redundant Code (CRC) Protection

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

Controller Area Network (CAN)

There are two controller area network (CAN) modules. A CAN controller implements the CAN 2.0B (active) protocol. This protocol is an asynchronous communications protocol used in both industrial and automotive control systems. The CAN protocol is well suited for control applications due to the capability to communicate reliably over a network. This is because the protocol incorporates CRC checking, message error tracking, and fault node confinement.

The CAN controller offers the following features:

- 32 mailboxes (8 receive only, 8 transmit only, 16 configurable for receive or transmit)
- Dedicated acceptance masks for each mailbox
- Additional data filtering on the first two bytes
- Support for both the standard (11-bit) and extended (29-bit) identifier (ID) message formats
- Support for remote frames
- Active or passive network support
- Interrupts, including transmit and receive complete, error, and global

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

Timers

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

General-Purpose (GP) Timers (TIMER)

There is one general-purpose (GP) timer unit, providing eight GP programmable timers. Each timer has an external pin that can be configured either as PWM or timer output, as an input to clock the timer, or as a mechanism for measuring pulse widths and periods of external events. These timers can be synchronized to an external clock input on the TM_TMR[n] pins, an external TM_CLK input pin, or to the internal SCLK0.

These timer units can be used in conjunction with the UARTs and the CAN controller to measure the width of the pulses in the data stream to provide a software autobaud detect function for the respective serial channels.

The GP timers can generate interrupts to the processor core, providing periodic events for synchronization to either the system clock or to external signals. Timer events can also trigger other peripherals via the TRU (for instance, to signal a fault). Each timer can also be started and/or stopped by any TRU master without core intervention.

Watchdog Timer (WDT)

Three on-chip software watchdog timers (WDT) can be used by the ARM Cortex-A5 and/or SHARC+ cores. A software watchdog can improve system availability by forcing the processors to a known state, via a general-purpose interrupt, or a fault, if the timer expires before being reset by software.

The programmer initializes the count value of the timer, enables the appropriate interrupt, then enables the timer. Thereafter, the software must reload the counter before it counts down to zero from the programmed value, protecting the system from remaining in an unknown state where software that normally resets the timer stops running due to an external noise condition or software error.

General-Purpose Counters (CNT)

A 32-bit counter (CNT) is provided that can operate in general-purpose up/down count modes and can sense 2-bit quadrature or binary codes as typically emitted by industrial drives or manual thumbwheels. Count direction is either controlled by a level-sensitive input pin or by two edge detectors.

A third counter input can provide flexible zero marker support and can input the push button signal of thumbwheel devices. All three CNT0 pins have a programmable debouncing circuit.

Internal signals forwarded to a GP timer enable the timer to measure the intervals between count events. Boundary registers enable auto-zero operation or simple system warning by interrupts when programmed count values are exceeded.

Housekeeping Analog-to-Digital Converter (HADC)

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

- 12-bit ADC core with built in sample and hold.
- Eight single-ended input channels for the BGA package; four single-ended input channels for the LQFP package.
- Throughput rates up to 1 MSPS.
- Single external reference with analog inputs between 0 V and 3.3 V.
- Selectable ADC clock frequency including the ability to program a prescaler.
- Adaptable conversion type; allows single or continuous conversion with option of autoscan.
- Autosequencing capability with up to eight autoconversions in a single session. Each conversion can be programmed to select one to eight input channels.
- Six data registers (individually addressable) to store conversion values

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

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

The USB 2.0 OTG dual-role device controller provides a low cost connectivity solution in industrial applications, as well as consumer mobile devices such as cell phones, digital still cameras, and MP3 players. The USB 2.0 controller allows these devices to transfer data using a point to point USB connection without the need for a PC host. The module can operate in a traditional USB peripheral only mode as well as the host mode presented in the OTG supplement to the USB 2.0 specification.

Table 17. Signal Multiplexing for Port E (Continued)

Signal Name	Multiplexed Function 0	Multiplexed Function 1	Multiplexed Function 2	Multiplexed Function 3	Multiplexed Function Input Tap
PE_03	LP0_ACK				
PE_04	LP0_D0				
PE_05	LP0_D1				
PE_06	LP0_D2				
PE_07	LP0_D3				
PE_08	LP0_D4				
PE_09	LP0_D5				
PE_10	LP0_D6				
PE_11	LP0_D7				
PE_12	MSI0_D0		TM0_TMR0		
PE_13	MSI0_D1	C1_FLG0	CNT0_UD		
PE_14	MSI0_D2	$\overline{\text{UART1\_CTS}}$	TM0_TMR6		
PE_15	MSI0_D3	C2_FLG3			

Table 18. Signal Multiplexing for Port F

Signal Name	Multiplexed Function 0	Multiplexed Function 1	Multiplexed Function 2	Multiplexed Function 3	Multiplexed Function Input Tap
PF_00	MSI0_D4	TRACE0_D00			
PF_01	MSI0_D5	TRACE0_D01			
PF_02	MSI0_D6	TRACE0_D02			
PF_03	MSI0_D7	TRACE0_D03			
PF_04	MSI0_CLK	C1_FLG2	$\overline{\text{SPI0\_SEL6}}$		
PF_05	ETH0_PTPCLKIN0	TM0_TMR1	$\overline{\text{SPI0\_SEL5}}$		
PF_06	ETH0_PTPAUXIN2	TRACE0_CLK			TM0_ACLK1
PF_07	ETH0_PTPAUXIN3	TM0_TMR2	MSI0_CMD		
PF_08	$\overline{\text{UART0\_TX}}$				
PF_09	$\overline{\text{UART0\_RX}}$				TM0_ACI0
PF_10	$\overline{\text{UART1\_TX}}$	$\overline{\text{SPI2\_SEL2}}$			
PF_11	$\overline{\text{UART1\_RX}}$	ACM0_A0	$\overline{\text{SPI1\_SEL3}}$	C2_FLG2	TM0_ACI1

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

Table 19. Internal Timer Signal Routing

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

[1] Not applicable for LQFP package.

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

Signal Name	Description	Port	Pin Name
DAI0_PIN15	DAI0 Pin 15	Not Muxed	DAI0_PIN15
DAI0_PIN16	DAI0 Pin 16	Not Muxed	DAI0_PIN16
DAI0_PIN17	DAI0 Pin 17	Not Muxed	DAI0_PIN17
DAI0_PIN18	DAI0 Pin 18	Not Muxed	DAI0_PIN18
DAI0_PIN19	DAI0 Pin 19	Not Muxed	DAI0_PIN19
DAI0_PIN20	DAI0 Pin 20	Not Muxed	DAI0_PIN20
ETH0_COL	EMAC0 MII Collision detect	C	PC_06
ETH0_CRS	EMAC0 Carrier Sense/RMII Receive Data Valid	B	PB_01
ETH0_MDC	EMAC0 Management Channel Clock	A	PA_11
ETH0_MDIO	EMAC0 Management Channel Serial Data	A	PA_10
ETH0_PTPAUXIN0	EMAC0 PTP Auxiliary Trigger Input 0	D	PD_14
ETH0_PTPAUXIN1	EMAC0 PTP Auxiliary Trigger Input 1	D	PD_15
ETH0_PTPPPS0	EMAC0 PTP Pulse Per Second Output 0	A	PA_09
ETH0_PTPPPS1	EMAC0 PTP Pulse Per Second Output 1	D	PD_08
ETH0_RXCLK_REFCLK	EMAC0 RXCLK (10/100/1000) or REFCLK (10/100)	B	PB_00
ETH0_RXCTL_RXDV	EMAC0 RXCTL (10/100/1000) or CRS (10/100)	B	PB_01
ETH0_RXD0	EMAC0 Receive Data 0	A	PA_13
ETH0_RXD1	EMAC0 Receive Data 1	A	PA_12
ETH0_RXD2	EMAC0 Receive Data 2	A	PA_14
ETH0_RXD3	EMAC0 Receive Data 3	A	PA_15
ETH0_RXERR	EMAC0 Receive Error	B	PB_03
ETH0_TXCLK	EMAC0 Transmit Clock	B	PB_04
ETH0_TXCTL_TXEN	EMAC0 TXCTL (10/100/1000) or TXEN (10/100)	B	PB_09
ETH0_TXD0	EMAC0 Transmit Data 0	B	PB_07
ETH0_TXD1	EMAC0 Transmit Data 1	B	PB_08
ETH0_TXD2	EMAC0 Transmit Data 2	B	PB_06
ETH0_TXD3	EMAC0 Transmit Data 3	B	PB_05
HADC0_EOC_DOUT	HADC0 End of Conversion/Serial Data Out	D	PD_09
HADC0_VIN0	HADC0 Analog Input at channel 0	Not Muxed	HADC0_VIN0
HADC0_VIN1	HADC0 Analog Input at channel 1	Not Muxed	HADC0_VIN1
HADC0_VIN2	HADC0 Analog Input at channel 2	Not Muxed	HADC0_VIN2
HADC0_VIN3	HADC0 Analog Input at channel 3	Not Muxed	HADC0_VIN3
HADC0_VREFN	HADC0 Ground Reference for ADC	Not Muxed	HADC0_VREFN
HADC0_VREFP	HADC0 External Reference for ADC	Not Muxed	HADC0_VREFP
JTG_TCK	JTAG Clock	Not Muxed	JTG_TCK
JTG_TDI	JTAG Serial Data In	Not Muxed	JTG_TDI
JTG_TDO	JTAG Serial Data Out	Not Muxed	JTG_TDO
JTG_TMS	JTAG Mode Select	Not Muxed	JTG_TMS
$\overline{\text{JTG\_TRST}}$	JTAG Reset	Not Muxed	$\overline{\text{JTG\_TRST}}$
LP1_ACK	LP1 Acknowledge	B	PB_01
LP1_CLK	LP1 Clock	B	PB_03
LP1_D0	LP1 Data 0	D	PD_10
LP1_D1	LP1 Data 1	D	PD_11
LP1_D2	LP1 Data 2	D	PD_12
LP1_D3	LP1 Data 3	D	PD_13
LP1_D4	LP1 Data 4	D	PD_14
LP1_D5	LP1 Data 5	D	PD_15
LP1_D6	LP1 Data 6	A	PA_09

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

Signal Name	Description	Port	Pin Name
LP1_D7	LP1 Data 7	D	PD_09
MLB0_CLK	MLB0 Single-Ended Clock	B	PB_06
MLB0_CLKOUT	MLB0 Single-Ended Clock Out	B	PB_03
MLB0_DAT	MLB0 Single-Ended Data	B	PB_04
MLB0_SIG	MLB0 Single-Ended Signal	B	PB_05
PPI0_CLK	EPPI0 Clock	C	PC_11
PPI0_D00	EPPI0 Data 0	D	PD_10
PPI0_D01	EPPI0 Data 1	D	PD_11
PPI0_D02	EPPI0 Data 2	D	PD_12
PPI0_D03	EPPI0 Data 3	D	PD_13
PPI0_D04	EPPI0 Data 4	D	PD_14
PPI0_D05	EPPI0 Data 5	D	PD_15
PPI0_D06	EPPI0 Data 6	C	PC_05
PPI0_D07	EPPI0 Data 7	D	PD_09
PPI0_D08	EPPI0 Data 8	C	PC_01
PPI0_D09	EPPI0 Data 9	C	PC_02
PPI0_D10	EPPI0 Data 10	C	PC_03
PPI0_D11	EPPI0 Data 11	C	PC_04
PPI0_FS1	EPPI0 Frame Sync 1 (HSYNC)	C	PC_14
PPI0_FS2	EPPI0 Frame Sync 2 (VSYNC)	C	PC_15
PPI0_FS3	EPPI0 Frame Sync 3 (FIELD)	C	PC_06
SPI0_CLK	SPI0 Clock	C	PC_01
SPI0_MISO	SPI0 Master In, Slave Out	C	PC_02
SPI0_MOSI	SPI0 Master Out, Slave In	C	PC_03
SPI0_RDY	SPI0 Ready	C	PC_05
$\overline{\text{SPI0\_SEL1}}$	SPI0 Slave Select Output 1	C	PC_04
$\overline{\text{SPI0\_SEL2}}$	SPI0 Slave Select Output 2	C	PC_05
$\overline{\text{SPI0\_SEL3}}$	SPI0 Slave Select Output 3	C	PC_06
$\overline{\text{SPI0\_SEL4}}$	SPI0 Slave Select Output 4	A	PA_09
$\overline{\text{SPI0\_SEL5}}$	SPI0 Slave Select Output 5	D	PD_03
$\overline{\text{SPI0\_SEL6}}$	SPI0 Slave Select Output 6	D	PD_04
$\overline{\text{SPI0\_SEL7}}$	SPI0 Slave Select Output 7	D	PD_05
$\overline{\text{SPI0\_SS}}$	SPI0 Slave Select Input	C	PC_04
SPI1_CLK	SPI1 Clock	C	PC_07
SPI1_MISO	SPI1 Master In, Slave Out	C	PC_08
SPI1_MOSI	SPI1 Master Out, Slave In	C	PC_09
SPI1_RDY	SPI1 Ready	C	PC_11
$\overline{\text{SPI1\_SEL1}}$	SPI1 Slave Select Output 1	C	PC_10
$\overline{\text{SPI1\_SEL2}}$	SPI1 Slave Select Output 2	C	PC_11
$\overline{\text{SPI1\_SEL3}}$	SPI1 Slave Select Output 3	A	PA_08
$\overline{\text{SPI1\_SEL4}}$	SPI1 Slave Select Output 4	A	PA_14
$\overline{\text{SPI1\_SEL5}}$	SPI1 Slave Select Output 5	B	PB_02
$\overline{\text{SPI1\_SEL6}}$	SPI1 Slave Select Output 6	D	PD_07
$\overline{\text{SPI1\_SEL7}}$	SPI1 Slave Select Output 7	D	PD_06
$\overline{\text{SPI1\_SS}}$	SPI1 Slave Select Input	C	PC_10
SPI2_CLK	SPI2 Clock	B	PB_14
SPI2_D2	SPI2 Data 2	B	PB_12
SPI2_D3	SPI2 Data 3	B	PB_13

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

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

DDR2 SDRAM Read Cycle Timing

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

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

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

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

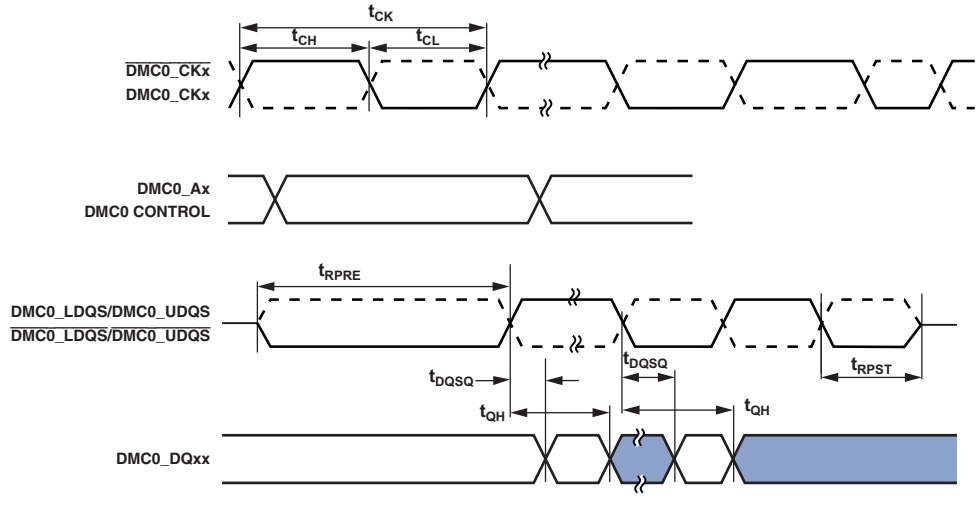


Figure 11. DDR2 SDRAM Controller Input AC Timing

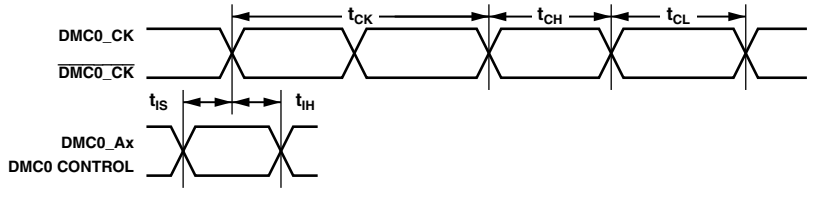
Mobile DDR (LPDDR) SDRAM Clock and Control Cycle Timing

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

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

Parameter		200 MHz[1]		Unit
		Min	Max	
Switching Characteristics				
t_{CK}	Clock Cycle Time (CL = 2 Not Supported)	5		ns
t_{CH}	Minimum Clock Pulse Width	0.45	0.55	t_{CK}
t_{CL}	Maximum Clock Pulse Width	0.45	0.55	t_{CK}
t_{IS}	Control/Address Setup Relative to DMC0_CK Rise	1		ns
t_{IH}	Control/Address Hold Relative to DMC0_CK Rise	1		ns

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



NOTE: CONTROL = DMC0_CS0, DMC0_CKE, DMC0_RAS, DMC0_CAS, AND DMC0_WE.
ADDRESS = DMC0_A0–A15 AND DMC0_BA0–BA2.

Figure 13. Mobile DDR SDRAM Clock and Control Cycle Timing

DDR3 SDRAM Read Cycle Timing

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

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

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

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

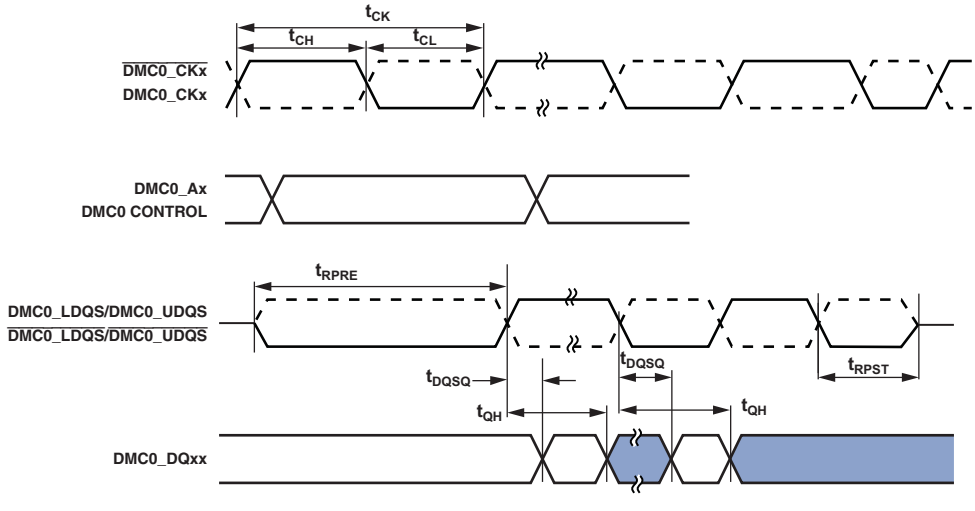


Figure 17. DDR3 SDRAM Controller Input AC Timing

Enhanced Parallel Peripheral Interface (EPPI) Timing

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

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

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

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

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

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

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

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

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

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

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

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

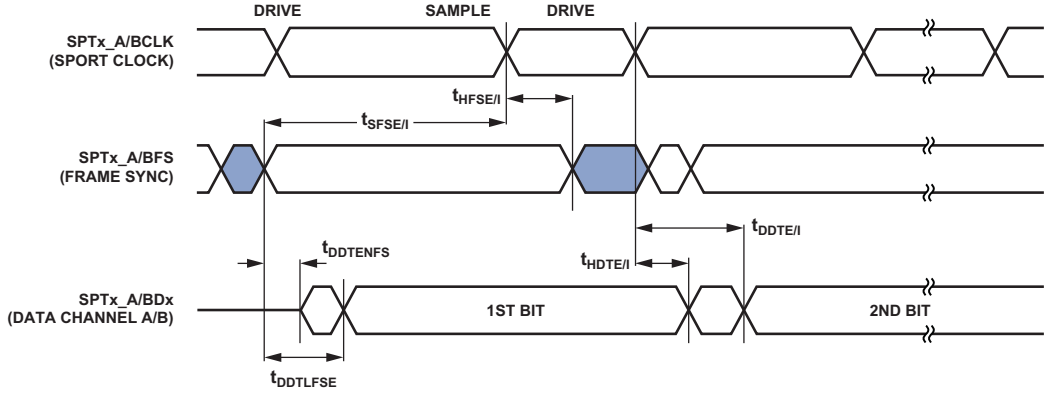


Figure 33. External Late Frame Sync

SPI Port—Master Timing

SPI0, SPI1, and SPI2

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

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

For SPI0, SPI1,

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

For SPI2,

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

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

Note that

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

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

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

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

SPI Port—SPIx_RDY Slave Timing

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

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

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

[1] All specifications apply to all three SPIs.

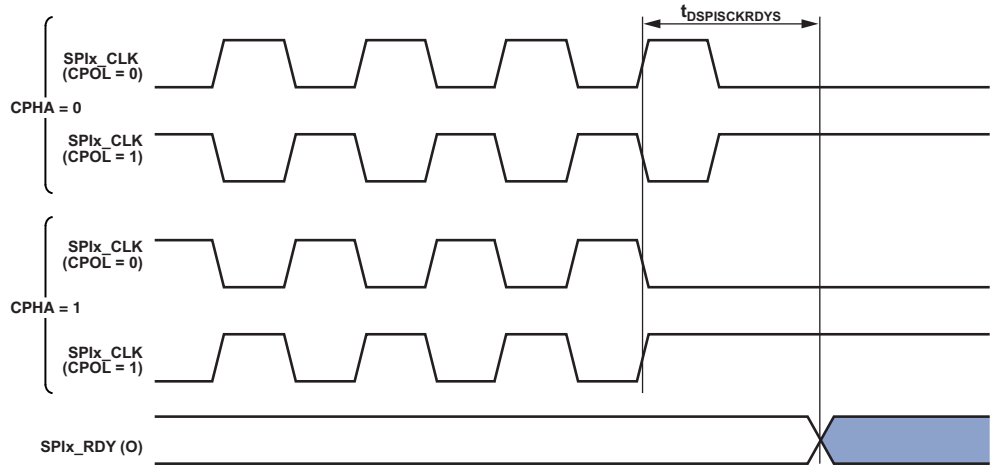


Figure 38. SPIx_RDY Deassertion from Valid Input SPIx_CLK Edge in Slave Mode

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

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

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

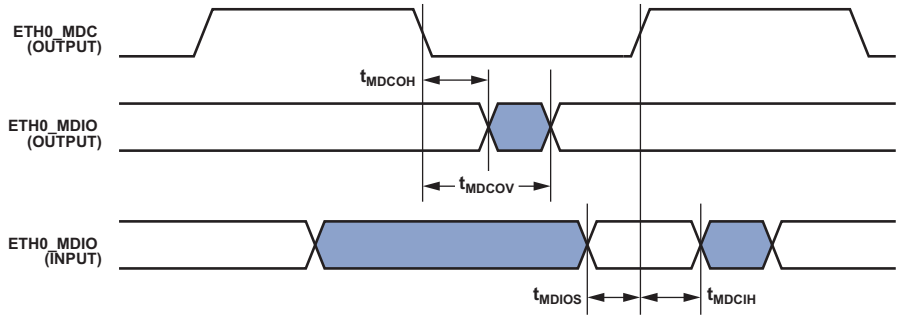


Figure 53. 10/100 /1000 Ethernet MAC Controller Timing—RMII and RGMII Station Management

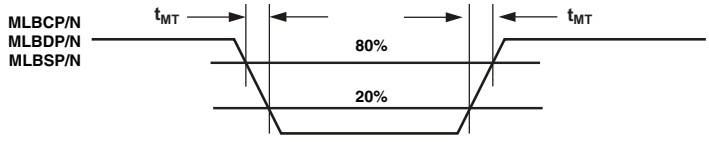


Figure 61. MLB 6-Pin Transition Time

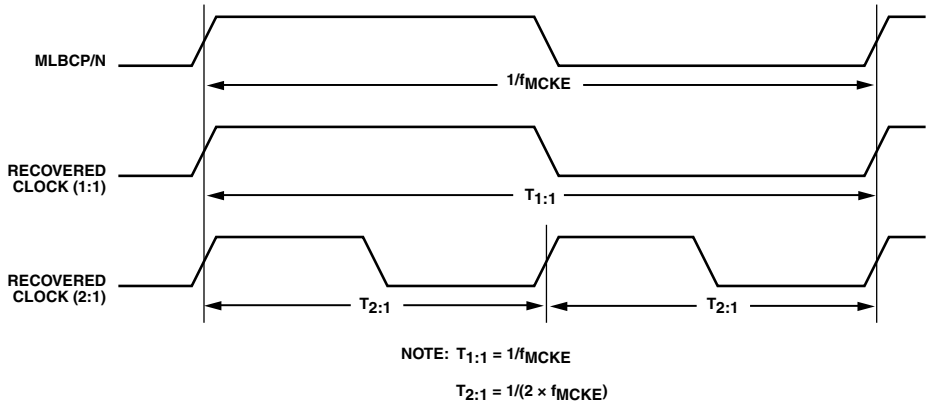


Figure 62. MLB 6-Pin Clock Definitions

Debug Interface (JTAG Emulation Port) Timing

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

Table 95. JTAG Emulation Port Timing

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

[1] System Inputs = MLB0_CLKP, MLB0_DATP, MLB0_SIGP, DAI0_PIN20-01, DMC0_A15-0, DMC0_DQ15-0, $\overline{\text{DMC0\_RESET}}$, PA_15-0, PB_15-0, PC_15-0, PD_15-0, PE_15-0, PF_11-0, SYS_BMODE2-0, SYS_FAULT, $\overline{\text{SYS\_FAULT}}$, $\overline{\text{SYS\_RESOUT}}$, TWI2-0_SCL, TWI2-0_SDA2.

[2] 50 MHz maximum.

[3] System Outputs = DMC0_A15-0, DMC0_BA2-0, $\overline{\text{DMC0\_CAS}}$, DMC0_CK, DMC0_CKE, $\overline{\text{DMC0\_CS0}}$, DMC0_DQ15-0, DMC0_LDM, DMC0_LDQS, DMC0_ODT, $\overline{\text{DMC0\_RAS}}$, $\overline{\text{DMC0\_RESET}}$, DMC0_UDM, DMC0_UDQS, $\overline{\text{DMC0\_WE}}$, MLB0_DATP, MLB0_SIGP, PA_15-0, PB_15-0, PC_15-0, PD_15-0, PE_15-0, PF_11-0, SYS_BMODE2-0, SYS_CLKOUT, SYS_FAULT, $\overline{\text{SYS\_FAULT}}$, $\overline{\text{SYS\_RESOUT}}$.

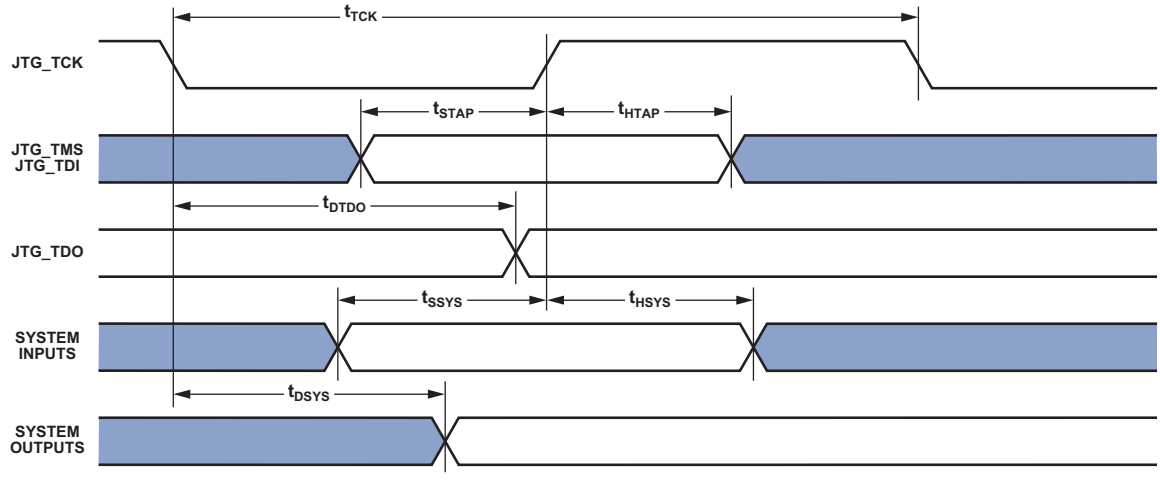


Figure 67. JTAG Port Timing

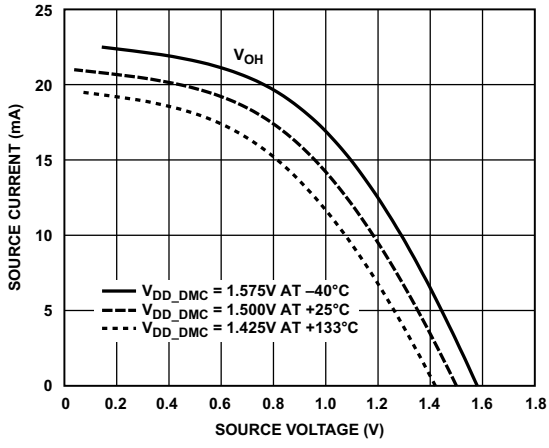


Figure 73. Driver Type B and Driver Type C (DDR3 Drive Strength 40 Ω)

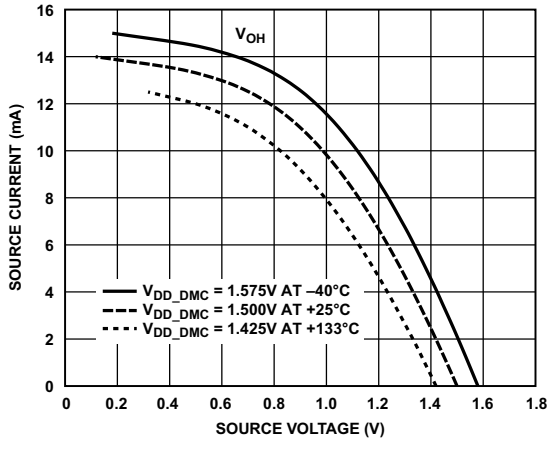


Figure 74. Driver Type B and Driver Type C (DDR3 Drive Strength 60 Ω)

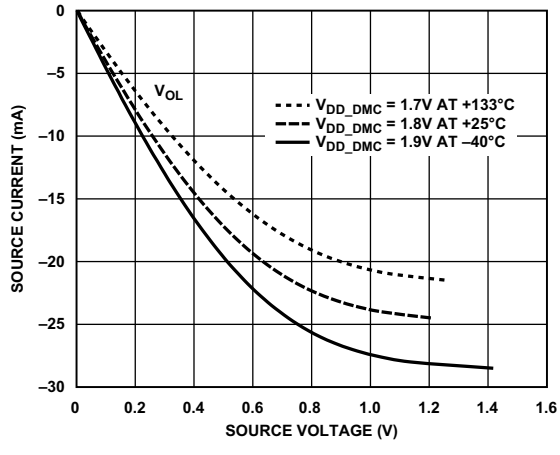


Figure 75. Driver Type B and Driver Type C (DDR2 Drive Strength 40 Ω)

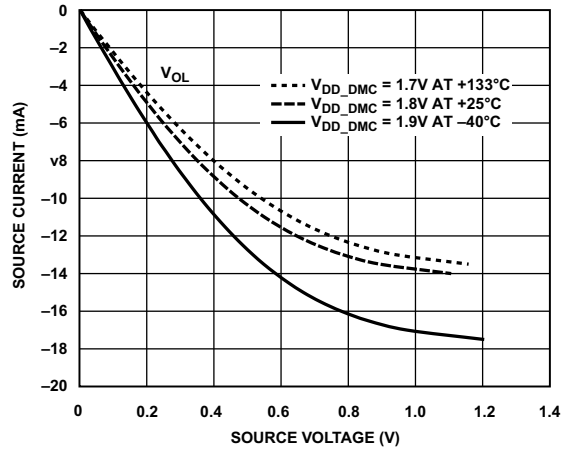


Figure 76. Driver Type B and Driver Type C (DDR2 Drive Strength 60 Ω)

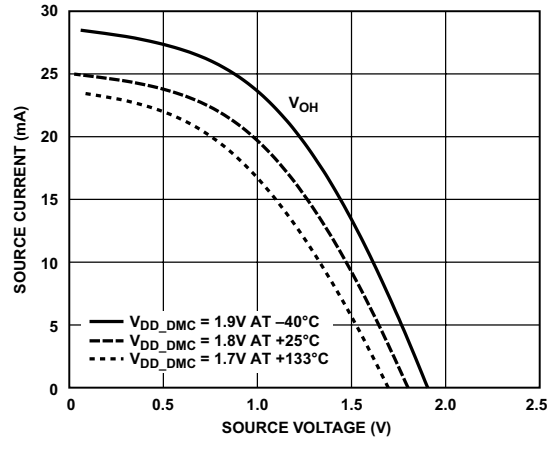


Figure 77. Driver Type B and Driver Type C (DDR2 Drive Strength 40 Ω)

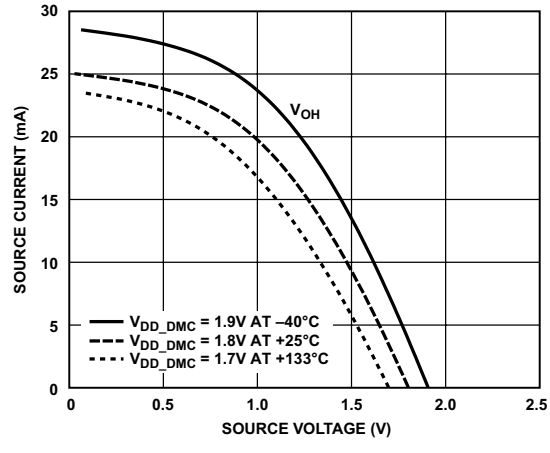


Figure 78. Driver Type B and Driver Type C (DDR2 Drive Strength 60 Ω)

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

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

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

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

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

D16121-0-6/18(B)