Welcome to E-XFL.COM

Understanding Embedded - DSP (Digital Signal Processors)

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

Applications of Embedded - DSP (Digital Signal Processors)

Details	
Product Status	Active
Type	Floating Point
Interface	CAN, EBI/EMI, Ethernet, DAI, I²C, MMC/SD/SDIO, SPI, SPORT, UART/USART, USB OTG
Clock Rate	450MHz
Non-Volatile Memory	ROM (512kB)
On-Chip RAM	640kB
Voltage - I/O	3.30V
Voltage - Core	1.10V
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	529-LFBGA, CSPBGA
Supplier Device Package	529-CSPBGA (19x19)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/adsp-sc587kbcz-4b

TABLE OF CONTENTS

REVISION HISTORY

10/2016—Revision 0: Initial Version

Memory Direct Memory Access (MDMA)

The processor supports various MDMA operations, including,

- Standard bandwidth MDMA channels with CRC protection (32-bit bus width, runs on SCLK0)
- Enhanced bandwidth MDMA channel (32-bit bus width, runs on SYSCLK)
- Maximum bandwidth MDMA channels (64-bit bus width, run on SYCLK, one channel can be assigned to the FFT accelerator)

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 C ode (CRC) Protection

The cyclic redundant codes (CRC) protection modules allow system software to calculate the signature of code, data, or both in memory, the content of memory-mapped registers, or periodic communication message objects. Dedicated hardware circuitry compares the signature with precalculated values and triggers appropriate fault events.

For example, every 100 ms the system software initiates the signature calculation of the entire memory contents and compares these contents with expected, precalculated values. If a mismatch occurs, a fault condition is generated through the processor core or the trigger routing unit.

The CRC is a hardware module based on a CRC32 engine that computes the CRC value of the 32-bit data-words presented to it. The source channel of the memory to memory DMA (in memory scan mode) provides data. The data can be optionally forwarded to the destination channel (memory transfer mode). The main features of the CRC peripheral are as follows:

- Memory scan mode
- Memory transfer mode
- Data verify mode
- Data fill mode
- User-programmable CRC32 polynomial
- Bit/byte mirroring option (endianness)
- Fault/error interrupt mechanisms
- 1D and 2D fill block to initialize an array with constants
- 32-bit CRC signature of a block of a memory or an MMR block

Event Handling

The processors provide event handling that supports both nesting and prioritization. Nesting allows multiple event service routines to be active simultaneously. Prioritization ensures that servicing a higher priority event takes precedence over servicing a lower priority event.

The processors provide support for five different types of events:

- An emulation event causes the processors to enter emulation mode, allowing command and control of the processors through the JTAG interface.
- A reset event resets the processors.
- An exceptions event occur synchronously to program flow (in other words, the exception is taken before the instruction is allowed to complete). Conditions triggered on the one side by the SHARC+ core, such as data alignment (SIMD/long word) or compute violations (fixed or floating point), and illegal instructions cause core exceptions. Conditions triggered on the other side by the SEC, such as error correcting codes (ECC)/parity/watchdog/system clock, cause system exceptions.
- An interrupts event occurs asynchronously to program flow. They are caused by input signals, timers, and other peripherals, as well as by an explicit software instruction.

System Event Controller (SEC)

Both SHARC+ cores feature a system event controller. The SEC features include the following:

- Comprehensive system event source management including interrupt enable, fault enable, priority, core mapping, and source grouping
- A distributed programming model where each system event source control and all status fields are independent of each other
- Determinism where all system events have the same propagation delay and provide unique identification of a specific system event source
- A slave control port that provides access to all SEC registers for configuration, status, and interrupt/fault services
- Global locking that supports a register level protection model to prevent writes to locked registers
- Fault management including fault action configuration, time out, external indication, and system reset

Trigger Routing Unit (TRU)

The trigger routing unit (TRU) provides system-level sequence control without core intervention. The TRU maps trigger masters (generators of triggers) to trigger slaves (receivers of triggers). Slave endpoints can be configured to respond to triggers in various ways. Common applications enabled by the TRU include,

- Automatically triggering the start of a DMA sequence after a sequence from another DMA channel completes
- Software triggering
- Synchronization of concurrent activities

CRC checksums can be calculated or compared automatically during memory transfers, or one or multiple memory regions can be continuously scrubbed by a single DMA work unit as per DMA descriptor chain instructions. The CRC engine also protects data loaded during the boot process.

Signal Watchdogs

The eight general-purpose timers feature modes to monitor off-chip signals. The watchdog period mode monitors whether external signals toggle with a period within an expected range. The watchdog width mode monitors whether the pulse widths of external signals are within an expected range. Both modes help to detect undesired toggling or lack of toggling of system level signals.

System Event Controller (SEC)

Besides system events, the system event controller (SEC) further supports fault management including fault action configuration as timeout, internal indication by system interrupt, or external indication through the $\overline{\text{SYS\_FAULT}}$ pin and system reset.

PROCESSOR PERIPHERALS

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

Dynamic Memory Controller (DMC)

The 16-bit dynamic memory controller (DMC) interfaces to:

- LPDDR1 (JESD209A) maximum frequency 200 MHz, DDRCLK (64 Mb to 2 Gb)
- DDR2 (JESD79-2E) maximum frequency 400 MHz, DDRCLK (256 Mb to 4 Gb)
- DDR3 (JESD79-3E) maximum frequency 450 MHz, DDRCLK (512 Mb to 8 Gb)
- DDR3L (1.5 V compatible only) maximum frequency 450 MHz, DDRCLK (512 Mb to 8 Gb)

See Table 8 for the DMC memory map.

Digital Audio Interface (DAI)

The processors support two mirrored digital audio interface (DAI) units. Each DAI can connect various peripherals to any of the DAI pins (DAI_PIN20–DAI_PIN01).

The application code makes these connections using the signal routing unit (SRU), shown in Figure 1.

The SRU is a matrix routing unit (or group of multiplexers) that enables the peripherals provided by the DAI to interconnect under software control. This functionality allows easy use of the DAI associated peripherals for a wider variety of applications by using a larger set of algorithms than is possible with nonconfigurable signal paths.

The DAI includes the peripherals described in the following sections (SPORTs, ASRC, S/PDIF, and PCG). DAI pin buffers 20 and 19 can change the polarity of the input signals. Most signals of the peripherals belonging to different DAIs cannot be interconnected, with few exceptions.

The DAI_PINx pin buffers may also be used as GPIO pins. DAI input signals allow the triggering of interrupts on the rising edge, the falling edge, or both edges.

See the Digital Audio Interface (DAI) chapter of the ADSP-SC58x/ADSP-2158x SHARC+ Processor Hardware Reference for complete information on the use of the DAIs and SRUs.

Serial Ports (SPORTs)

The processors feature eight synchronous full serial ports. These ports provide an inexpensive interface to a wide variety of digital and mixed-signal peripheral devices. These devices include Analog Devices AD19xx/ADAU19xx family of audio codecs, analog-to-digital converters (ADCs) and digital-to-analog converters (DACs). Two data lines, a clock, and frame sync make up the serial ports. The data lines can be programmed to either transmit or receive data and each data line has a dedicated DMA channel.

An individual full SPORT module consists of two independently configurable SPORT halves with identical functionality. Two bidirectional data lines—primary (0) and secondary (1)—are available per SPORT half and are configurable as either transmitters or receivers. Therefore, each SPORT half permits two unidirectional streams into or out of the same SPORT. This bidirectional functionality provides greater flexibility for serial communications. For full-duplex configuration, one half SPORT provides two transmit signals, while the other half SPORT provides the two receive signals. The frame sync and clock are shared.

Serial ports operate in the following six modes:

- Standard DSP serial mode
- Multichannel time division multiplexing (TDM) mode
- I^2S mode
- Packed I^2S mode
- Left justified mode
- Right justified mode

Asynchronous Sample Rate Converter (ASRC)

The asynchronous sample rate converter (ASRC) contains eight ASRC blocks. It is the same core in the AD1896 192 kHz stereo asynchronous sample rate converter. The ASRC provides up to 140 dB signal-to-noise ratio (SNR). The ASRC block performs synchronous or asynchronous sample rate conversion across independent stereo channels, without using internal processor resources. The ASRC blocks can also be configured to operate together to convert multichannel audio data without phase mismatches. Finally, the ASRC can clean up audio data from jittery clock sources such as the S/PDIF receiver.

S/PDIF-Compatible Digital Audio Receiver/Transmitter

The Sony/Philips Digital Interface Format (S/PDIF) is a standard audio data transfer format that allows the transfer of digital audio signals from one device to another without converting them to an analog signal. There are two S/PDIF transmit/receive

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.

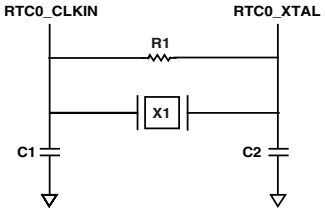
The reset source is defined as the following:

- 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.

This permits the customer to download, execute, and debug programs for the EZ-KIT Lite system. It also supports in circuit programming of the on-board Flash device to store user specific boot code, enabling standalone operation. With the full version of CrossCore Embedded Studio installed (sold separately), engineers can develop software for supported EZ-KITs or any custom system utilizing supported Analog Devices processors.

Software Add-Ins for CrossCore Embedded Studio

Analog Devices offers software add-ins which seamlessly integrate with CrossCore Embedded Studio to extend the capabilities and reduce development time. Add-ins include board support packages for evaluation hardware, various middleware packages, and algorithmic modules. Documentation, help, configuration dialogs, and coding examples present in these add-ins are viewable through the CrossCore Embedded Studio IDE once the add-in is installed.

Board Support Packages for Evaluation Hardware

Software support for the EZ-KIT Lite evaluation boards and EZ-Extender daughter cards is provided by software add-ins called board support packages (BSPs). The BSPs contain the required drivers, pertinent release notes, and select example code for the given evaluation hardware. A download link for a specific BSP is located on the web page for the associated EZ-KIT or EZ-Extender product.

Middleware Packages

Analog Devices offers middleware add-ins such as real time operating systems, file systems, USB stacks, and TCP/IP stacks. For more information, see the following web pages:

- www.analog.com/ucos2
- www.analog.com/ucos3
- www.analog.com/ucfs
- www.analog.com/ucusbd
- www.analog.com/ucusbh
- www.analog.com/lwip

Algorithmic Modules

To speed development, Analog Devices offers add-ins that perform popular audio and video processing algorithms. These are available for use with CrossCore Embedded Studio. For more information visit www.analog.com.

Designing an Emulator-Compatible DSP Board (Target)

For embedded system test and debug, Analog Devices provides a family of emulators. On each JTAG DSP, Analog Devices supplies an IEEE 1149.1 JTAG test access port (TAP). In-circuit emulation is facilitated by use of this JTAG interface. The emulator accesses the internal features of the processor via the TAP, allowing the developer to load code, set breakpoints, and view variables, memory, and registers.

The processor must be halted to send data and commands, but once an operation is completed by the emulator, the DSP system is set to run at full speed with no impact on system timing. The emulators require the target board to include a header that supports connection of the JTAG port of the DSP to the emulator.

For details on target board design issues including mechanical layout, single processor connections, signal buffering, signal termination, and emulator pod logic, see "Analog Devices JTAG Emulation Technical Reference" (EE-68).

ADDITIONAL INFORMATION

This data sheet provides a general overview of the ADSP-SC58x/ADSP-2158x architecture and functionality. For detailed information on the core architecture and instruction set, refer to the SHARC+ Core Programming Reference.

RELATED SIGNAL CHAINS

A signal chain is a series of signal-conditioning electronic components that receive input (data acquired from sampling either real-time phenomena or from stored data) in tandem, with the output of one portion of the chain supplying input to the next. Signal chains are often used in signal processing applications to gather and process data or to apply system controls based on analysis of real-time phenomena.

Analog Devices eases signal processing system development by providing signal processing components that are designed to work together well. A tool for viewing relationships between specific applications and related components is available on the www.analog.com website.

The application signal chains page in the Circuits from the Lab® site (http:\\www.analog.com\circuits) provides the following:

- Graphical circuit block diagram presentation of signal chains for a variety of circuit types and applications
- Drill down links for components in each chain to selection guides and application information
- Reference designs applying best practice design techniques

SECURITY FEATURES DISCLAIMER

To our knowledge, the Security Features, when used in accordance with the data sheet and hardware reference manual specifications, provide a secure method of implementing code and data safeguards. However, Analog Devices does not guarantee that this technology provides absolute security. ACCORDINGLY, ANALOG DEVICES HEREBY DISCLAIMS ANY AND ALL EXPRESS AND IMPLIED WARRANTIES THAT THE SECURITY FEATURES CANNOT BE BREACHED, COMPROMISED, OR OTHERWISE CIRCUMVENTED AND IN NO EVENT SHALL ANALOG DEVICES BE LIABLE FOR ANY LOSS, DAMAGE, DESTRUCTION, OR RELEASE OF DATA, INFORMATION, PHYSICAL PROPERTY, OR INTELLECTUAL PROPERTY.

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

Signal Name	Direction	Description
MLB_SIGP	InOut	**Differential Signal (+).**
MLB_CLK	Input	**Single-Ended Clock.**
MLB_DAT	InOut	**Single-Ended Data.**
MLB_SIG	InOut	**Single-Ended Signal.**
MLB_CLKOUT	Output	**Single-Ended Clock Out.**
$\overline{\text{MSI\_CD}}$	Input	**Card Detect.** Connects to a pull-up resistor and to the card detect output of an SD socket.
MSI_CLK	Output	**Clock.** The clock signal applied to the connected device from the MSI.
MSI_CMD	InOut	**Command.** Sends commands to and receives responses from the connected device.
MSI_D[n]	InOut	**Data n.** Bidirectional data bus.
$\overline{\text{MSI\_INT}}$	Input	**eSDIO Interrupt Input.** Used only for eSDIO. Connects to an eSDIO card interrupt output. An interrupt may be sampled even when the MSI clock to the card is switched off.
PCIE_CLKM	Input	**CLK –.**
PCIE_CLKP	Input	**CLK +.**
PCIE_REF	InOut	**Reference Resistor.** Attach a 200 Ω, 1%, 100-ppm/C precision resistor to ground on the board.
PCIE_RXM	Input	**RX –.**
PCIE_RXP	Input	**RX +.**
PCIE_TXM	Output	**TX –.**
PCIE_TXP	Output	**TX +.**
PPI_CLK	InOut	**Clock.** Input in external clock mode, output in internal clock mode.
PPI_D[nn]	InOut	**Data n.** Bidirectional data bus.
PPI_FS1	InOut	**Frame Sync 1 (HSYNC).** Behavior depends on EPPI mode. See the EPPI chapter of the ADSP-SC58x/ADSP-2158x SHARC+ Processor Hardware Reference for more details.
PPI_FS2	InOut	**Frame Sync 2 (VSYNC).** Behavior depends on EPPI mode. See the EPPI chapter of the ADSP-SC58x/ADSP-2158x SHARC+ Processor Hardware Reference for more details.
PPI_FS3	InOut	**Frame Sync 3 (FIELD).** Behavior depends on EPPI mode. See the EPPI chapter of the ADSP-SC58x/ADSP-2158x SHARC+ Processor Hardware Reference for more details.
PWM_AH	Output	**Channel A High Side.** High side drive signal.
PWM_AL	Output	**Channel A Low Side.** Low side drive signal.
PWM_BH	Output	**Channel B High Side.** High side drive signal.
PWM_BL	Output	**Channel B Low Side.** Low side drive signal.
PWM_CH	Output	**Channel C High Side.** High side drive signal.
PWM_CL	Output	**Channel C Low Side.** Low side drive signal.
PWM_DH	Output	**Channel D High Side.** High side drive signal.
PWM_DL	Output	**Channel D Low Side.** Low side drive signal.
PWM_SYNC	Input	**PWMTMR Grouped.** This input is for an externally generated sync signal. If the sync signal is internally generated, no connection is necessary.
$\overline{\text{PWM\_TRIP[n]}}$	Input	**Shutdown Input n.** When asserted, the selected PWM channel outputs are shut down immediately.
P_[nn]	InOut	**Position n.** General-purpose input/output. See the GP Ports chapter of the ADSP-SC58x/ADSP-2158x SHARC+ Processor Hardware Reference for more details.
RTC_CLKIN	Input	**Crystal Input/External Oscillator Connection.** Connect to an external clock source or crystal.
RTC_XTAL	Output	**Crystal Output.** Drives an external crystal. Must be left unconnected if an external clock is driving RTC_CLKIN.
SINC_CLK0	Output	**Clock 0.**
SINC_D0	Input	**Data 0.**
SINC_D1	Input	**Data 1.**
SINC_D2	Input	**Data 2.**
SINC_D3	Input	**Data 3.**

GPIO MULTIPLEXING FOR THE 349-BALL CSP_BGA PACKAGE

Table 13 through Table 17 identify the pin functions that are multiplexed on the general-purpose I/O pins of the 349-ball CSP_BGA package.

Table 13. Signal Multiplexing for Port A

Signal Name	Multiplexed Function 0	Multiplexed Function 1	Multiplexed Function 2	Multiplexed Function 3	Multiplexed Function Input Tap
PA_00	ETH0_TXD0			SMC0_A21	
PA_01	ETH0_TXD1			SMC0_A20	
PA_02	ETH0_MDC			SMC0_A24	
PA_03	ETH0_MDIO			SMC0_A23	
PA_04	ETH0_RXD0			SMC0_A19	
PA_05	ETH0_RXD1			SMC0_A18	
PA_06	ETH0_RXCLK_REFCLK			SMC0_A17	
PA_07	ETH0_CRS			SMC0_A16	
PA_08	ETH0_RXD2			SMC0_A12	
PA_09	ETH0_RXD3			SMC0_A11	
PA_10	ETH0_TXEN			SMC0_A22	
PA_11	ETH0_TXCLK			SMC0_A15	
PA_12	ETH0_TXD2			SMC0_A14	
PA_13	ETH0_TXD3			SMC0_A13	
PA_14	ETH0_PTPPPS3	SINC0_D0		SMC0_A10	
PA_15	ETH0_PTPPPS2	SINC0_D1		SMC0_A09	

Table 14. Signal Multiplexing for Port B

Signal Name	Multiplexed Function 0	Multiplexed Function 1	Multiplexed Function 2	Multiplexed Function 3	Multiplexed Function Input Tap
PB_00	ETH0_PTPPPS1	SINC0_D2	PPI0_D14	SMC0_A08	TM0_ACLK3
PB_01	ETH0_PTPPPS0	SINC0_CLK0	PPI0_D15	SMC0_A07	TM0_ACLK4
PB_02	ETH0_PTPCLKIN0	$\overline{\text{UART1\_TX}}$	PPI0_D16	SMC0_A04	
PB_03	ETH0_PTPAUXIN0	$\overline{\text{UART1\_RX}}$	PPI0_D17	SMC0_A03	TM0_ACI1
PB_04	MLB0_CLK	SINC0_D3	PPI0_D12	SMC0_ARDY	ETH0_PTPAUXIN1
PB_05	MLB0_SIG		PPI0_D13	SMC0_A01	ETH0_PTPAUXIN2
PB_06	MLB0_DAT		PWM0_BH	SMC0_A02	ETH0_PTPAUXIN3
PB_07	LP1_D0	PWM0_AH	TM0_TMR3	SMC0_D15	
PB_08	LP1_D1	PWM0_AL	TM0_TMR4	SMC0_D14	
PB_09	LP1_D2		CAN1_TX	SMC0_D13	
PB_10	LP1_D3	TM0_TMR2	CAN1_RX	SMC0_D12	TM0_ACI4
PB_11	LP1_D4		PWM0_DH	SMC0_D11	CNT0_ZM
PB_12	LP1_D5		PWM0_DL	SMC0_D10	CNT0_UD
PB_13	LP1_D6		PWM0_CH	SMC0_D09	
PB_14	LP1_D7	TM0_TMR5	PWM0_CL	SMC0_D08	CNT0_DG
PB_15	LP1_ACK	$\overline{\text{PWM0\_TRIP0}}$	TM0_TMR1	$\overline{\text{SMC0\_AWE}}$	

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

Signal Name	Description	Port	Pin Name
TRACE0_D03	TRACE0 Trace Data (first instance)	G	PG_01
TRACE0_D03	TRACE0 Trace Data 3 (second instance)	D	PD_05
TRACE0_D04	TRACE0 Trace Data (first instance)	G	PG_02
TRACE0_D04	TRACE0 Trace Data 4 (second instance)	D	PD_06
TRACE0_D05	TRACE0 Trace Data 5 (first instance)	D	PD_07
TRACE0_D05	TRACE0 Trace Data (second instance)	G	PG_03
TRACE0_D06	TRACE0 Trace Data (first instance)	G	PG_04
TRACE0_D06	TRACE0 Trace Data 6 (second instance)	D	PD_08
TRACE0_D07	TRACE0 Trace Data (first instance)	G	PG_05
TRACE0_D07	TRACE0 Trace Data 7 (second instance)	D	PD_09
TRACE0_D08	TRACE0 Trace Data 8	F	PF_13
TRACE0_D09	TRACE0 Trace Data 9	F	PF_14
TRACE0_D10	TRACE0 Trace Data 10	F	PF_15
TRACE0_D11	TRACE0 Trace Data 11	G	PG_01
TRACE0_D12	TRACE0 Trace Data 12	G	PG_02
TRACE0_D13	TRACE0 Trace Data 13	G	PG_03
TRACE0_D14	TRACE0 Trace Data 14	G	PG_04
TRACE0_D15	TRACE0 Trace Data 15	G	PG_05
TWI0_SCL	TWI0 Serial Clock	Not Muxed	TWI0_SCL
TWI0_SDA	TWI0 Serial Data	Not Muxed	TWI0_SDA
TWI1_SCL	TWI1 Serial Clock	Not Muxed	TWI1_SCL
TWI1_SDA	TWI1 Serial Data	Not Muxed	TWI1_SDA
TWI2_SCL	TWI2 Serial Clock	Not Muxed	TWI2_SCL
TWI2_SDA	TWI2 Serial Data	Not Muxed	TWI2_SDA
$\overline{\text{UART0\_CTS}}$	UART0 Clear to Send	D	PD_00
$\overline{\text{UART0\_RTS}}$	UART0 Request to Send	C	PC_15
$\overline{\text{UART0\_RX}}$	UART0 Receive	C	PC_14
$\overline{\text{UART0\_TX}}$	UART0 Transmit	C	PC_13
$\overline{\text{UART1\_CTS}}$	UART1 Clear to Send	E	PE_01
$\overline{\text{UART1\_RTS}}$	UART1 Request to Send	E	PE_02
$\overline{\text{UART1\_RX}}$	UART1 Receive	B	PB_03
$\overline{\text{UART1\_TX}}$	UART1 Transmit	B	PB_02
$\overline{\text{UART2\_CTS}}$	UART2 Clear to Send	E	PE_11
$\overline{\text{UART2\_RTS}}$	UART2 Request to Send	E	PE_10
$\overline{\text{UART2\_RX}}$	UART2 Receive	D	PD_13
$\overline{\text{UART2\_TX}}$	UART2 Transmit	D	PD_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
USB1_DM	USB1 Data -	Not Muxed	USB1_DM
USB1_DP	USB1 Data +	Not Muxed	USB1_DP
USB1_VBUS	USB1 Bus Voltage	Not Muxed	USB1_VBUS
VDD_DMC	DMC VDD	Not Muxed	VDD_DMC
VDD_HADC	HADC VDD	Not Muxed	VDD_HADC

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

Table 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
PC_00	InOut	H	PullDown	none	none	VDD_EXT	Desc: PORTC Position 0 | LP1 Clock | PWM0 Channel B Low Side | SMC0 Read Enable | SPI0 Slave Select Output 4 Notes: No notes
PC_01	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTC Position 1 | SPI2 Clock Notes: No notes
PC_02	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTC Position 2 | SPI2 Master In, Slave Out Notes: No notes
PC_03	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTC Position 3 | SPI2 Master Out, Slave In Notes: No notes
PC_04	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTC Position 4 | SPI2 Data 2 Notes: No notes
PC_05	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTC Position 5 | SPI2 Data 3 Notes: No notes
PC_06	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTC Position 6 | SPI2 Slave Select Output 1 | SPI2 Slave Select Input Notes: No notes
PC_07	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTC Position 7 | CAN0 Receive | SMC0 Memory Select 2 | SPI0 Slave Select Output 1 | TIMER0 Alternate Capture Input 3 Notes: No notes
PC_08	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTC Position 8 | CAN0 Transmit | SMC0 Memory Select 3 Notes: No notes
PC_09	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTC Position 9 | SPI0 Clock Notes: No notes
PC_10	InOut	H	PullDown	none	none	VDD_EXT	Desc: PORTC Position 10 | SPI0 Master In, Slave Out Notes: No notes
PC_11	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTC Position 11 | SPI0 Master Out, Slave In | TIMER0 Clock Notes: No notes
PC_12	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTC Position 12 | ACM0 External Trigger n | SMC0 Address 25 | SPI0 Ready | SPI0 Slave Select Output 3 Notes: No notes

ELECTRICAL CHARACTERISTICS

Parameter		Conditions	450 MHz Min	450 MHz Typ	450 MHz Max	Unit
V_{OH}[1]	High Level Output Voltage	At V_{DD_EXT} = minimum, I_{OH} = –1.0 mA[2]	2.4			V
V_{OL}[1]	Low Level Output Voltage	At V_{DD_EXT} = minimum, I_{OL} = 1.0 mA[2]			0.4	V
V_{OH_DDR2}[3]	High Level Output Voltage for DDR2 DS = 40 Ω	At V_{DD_DDR} = minimum, I_{OH} = –5.8 mA	1.38			V
V_{OL_DDR2}[3]	Low Level Output Voltage for DDR2 DS = 40 Ω	At V_{DD_DDR} = minimum, I_{OL} = 5.8 mA			0.32	V
V_{OH_DDR2}[3]	High Level Output Voltage for DDR2 DS = 60 Ω	At V_{DD_DDR} = minimum, I_{OH} = –3.4 mA	1.38			V
V_{OL_DDR2}[3]	Low Level Output Voltage for DDR2 DS = 60 Ω	At V_{DD_DDR} = minimum, I_{OL} = 3.4 mA			0.32	V
V_{OH_DDR3}[4]	High Level Output Voltage for DDR3 DS = 40 Ω	At V_{DD_DDR} = minimum, I_{OH} = –5.8 mA	1.105			V
V_{OL_DDR3}[4]	Low Level Output Voltage for DDR3 DS = 40 Ω	At V_{DD_DDR} = minimum, I_{OL} = 5.8 mA			0.32	V
V_{OH_DDR3}[4]	High Level Output Voltage for DDR3 DS = 60 Ω	At V_{DD_DDR} = minimum, I_{OH} = –3.4 mA	1.105			V
V_{OL_DDR3}[4]	Low Level Output Voltage for DDR3 DS = 60 Ω	At V_{DD_DDR} = minimum, I_{OL} = 3.4 mA			0.32	V
V_{OH_LPDDR}[5]	High Level Output Voltage for LPDDR	At V_{DD_DDR} = minimum, I_{OH} = –6.0 mA	1.38			V
V_{OL_LPDDR}[5]	Low Level Output Voltage for LPDDR	At V_{DD_DDR} = minimum, I_{OL} = 6.0 mA			0.32	V
I_{IH}[6, 7]	High Level Input Current	At V_{DD_EXT} = maximum, V_{IN} = V_{DD_EXT} maximum			10	µA
I_{IL}[6]	Low Level Input Current	At V_{DD_EXT} = maximum, V_{IN} = 0 V			10	µA
I_{IL_PU}[7]	Low Level Input Current Pull-up	At V_{DD_EXT} = maximum, V_{IN} = 0 V			200	µA
I_{IH_PD}[8]	High Level Input Current Pull-down	At V_{DD_EXT} = maximum, V_{IN} = 0 V			200	µA
I_{OZH}[9]	Three-State Leakage Current	At V_{DD_EXT}/V_{DD_DDR} = maximum, V_{IN} = V_{DD_EXT}/V_{DD_DDR} maximum			10	µA
I_{OZL}[9]	Three-State Leakage Current	at V_{DD_EXT}/V_{DD_DDR} = maximum, V_{IN} = 0 V			10	µA
C_{IN}[10]	Input Capacitance	T_{CASE} = 25°C			5	pF

Parameter		Conditions	450 MHz Min	Typ	Max	Unit
I_{DD_IDLE}	V_{DD_INT} Current in Idle	f_{CCLK} = 450 MHz ASF_{SHARC1} = 0.31 ASF_{SHARC2} = 0.31 ASF_{A5} = 0.29 f_{SYSCLK} = 225 MHz $f_{SCLK0/1}$ = 112.5 MHz (Other clocks are disabled) No peripheral or DMA activity T_J = 25°C V_{DD_INT} = 1.1 V		495		mA
I_{DD_TYP}	V_{DD_INT} Current	f_{CCLK} = 450 MHz ASF_{SHARC1} = 1.0 ASF_{SHARC2} = 1.0 ASF_{A5} = 0.73 f_{SYSCLK} = 225 MHz $f_{SCLK0/1}$ = 112.5 MHz (Other clocks are disabled) FFT accelerator operating at $f_{SYSCLK/4}$ DMA data rate = 600 MB/s T_J = 25°C V_{DD_INT} = 1.1 V		1112		mA
I_{DD_INT}[11]	V_{DD_INT} Current	f_{CCLK} > 0 MHz $f_{SCLK0/1} \geq 0$ MHz			See $I_{DD\_INT\_TOT}$ equation in the Total Internal Power Dissipation section.	mA

[1] Applies to all output and bidirectional pins except TWI, DMC, USB, PCIe, and MLB.
[2] See the Output Drive Currents section for typical drive current capabilities.
[3] Applies to all DMC output and bidirectional signals in DDR2 mode.
[4] Applies to all DMC output and bidirectional signals in DDR3 mode.
[5] Applies to all DMC output and bidirectional signals in LPDDR mode.
[6] Applies to input pins SYS_BMODE0-2, SYS_CLKIN0, SYS_CLKIN1, $\overline{\text{SYS\_HWRST}}$, JTG_TDI, JTG_TMS, and USB0_CLKIN.
[7] Applies to input pins with internal pull-ups including JTG_TDI, JTG_TMS, and JTG_TCK.
[8] Applies to signals JTAG_TRST, USB0_VBUS, USB1_VBUS.
[9] Applies to signals PA0-15, PB0-15, PC0-15, PD0-15, PE0-15, PF0-15, PG0-5, DAI0_PINx, DAI1_PINx, DMC0_DQx, DMC0_LDQS, DMC0_UDQS, $\overline{\text{DMC0\_LDQS}}$, $\overline{\text{DMC0\_UDQS}}$, SYS_FAULT, $\overline{\text{SYS\_FAULT}}$, JTG_TDO, USB0_ID, USBx_DM, USBx_DP, and USBx_VBC.
[10] Applies to all signal pins.
[11] See "Estimating Power for ADSP-SC58x/2158x SHARC+ Processors" (EE-392) for further information.

Clock and Reset Timing

Table 44 and Figure 11 describe clock and reset operations related to the CGU and RCU. Per the CCLK, SYSCLK, SCLK, DCLK, and OCLK timing specifications in Table 29, combinations of SYS_CLKIN and clock multipliers must not select clock rates in excess of the maximum instruction rate of the processor.

Table 44. Clock and Reset Timing

Parameter		Min	Max	Unit
Timing Requirements				
f_{CKIN}	SYS_CLKINx Frequency (Crystal)[1, 2, 3]	20	50	MHz
	SYS_CLKINx Frequency (External CLKIN)[1, 2, 3]	20	50	MHz
t_{CKINL}	CLKIN Low Pulse[1]	10		ns
t_{CKINH}	CLKIN High Pulse[1]	10		ns
t_{WRST}	$\overline{\text{RESET}}$ Asserted Pulse Width Low[4]	11 × t_{CKIN}		ns

[1] Applies to PLL bypass mode and PLL nonbypass mode.
[2] The t_{CKIN} period (see Figure 11) equals $1/f_{CKIN}$.
[3] If the CGU_CTL.DF bit is set, the minimum f_{CKIN} specification is 40 MHz.
[4] Applies after power-up sequence is complete. See Table 43 and Figure 10 for power-up reset timing.

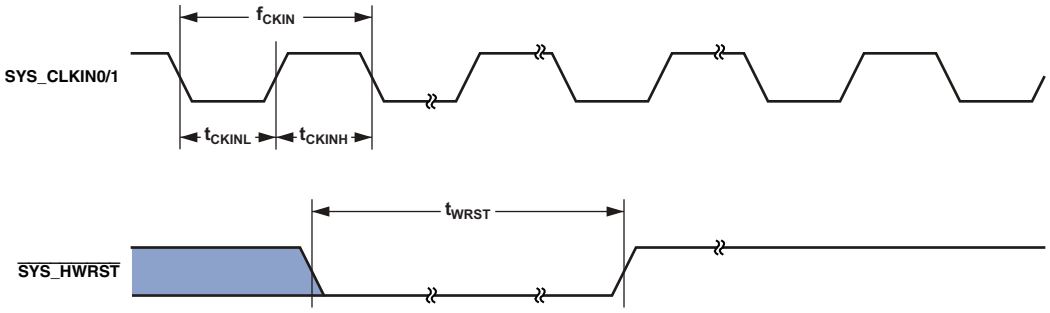


Figure 11. Clock and Reset Timing

Serial Ports (SPORT)

To determine whether a device is compatible with the SPORT at clock speed n, the following specifications must be confirmed: frame sync delay and frame sync setup and hold; data delay and data setup and hold; and serial clock (SPTx_CLK) width. In Figure 37, either the rising edge or the falling edge of SPTx_CLK (external or internal) can be used as the active sampling edge.

When externally generated, the SPORT clock is called $f_{SPTCLKEXT}$:

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

When internally generated, the programmed SPORT clock ($f_{SPTCLKPROG}$) frequency in MHz is set by the following equation where CLKDIV is a field in the SPORT_DIV register that can be set from 0 to 65535:

$$f_{SPTCLKPROG} = \frac{f_{SCLK0}}{(CLKDIV + 1)}$$

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

Table 64. Serial Ports—External Clock[1]

Parameter		Min	Max	Unit
Timing Requirements				
t_{SFSE}	Frame Sync Setup Before SPTx_CLK (Externally Generated Frame Sync in either Transmit or Receive Mode)[2]	2		ns
t_{HFSE}	Frame Sync Hold After SPTx_CLK (Externally Generated Frame Sync in either Transmit or Receive Mode)[2]	2.7		ns
t_{SDRE}	Receive Data Setup Before Receive SPTx_CLK[2]	2		ns
t_{HDRE}	Receive Data Hold After SPTx_CLK[2]	2.7		ns
$t_{SPTCLKW}$	SPTx_CLK Width[3]	$0.5 \times t_{SPTCLKEXT} - 1.5$		ns
t_{SPTCLK}	SPTx_CLK Period[3]	$t_{SPTCLKEXT} - 1.5$		ns
Switching Characteristics				
t_{DFSE}	Frame Sync Delay After SPTx_CLK (Internally Generated Frame Sync in either Transmit or Receive Mode)[4]		14.5	ns
t_{HOFSE}	Frame Sync Hold After SPTx_CLK (Internally Generated Frame Sync in either Transmit or Receive Mode)[4]	2		ns
t_{DDTE}	Transmit Data Delay After Transmit SPTx_CLK[4]		14	ns
t_{HDTE}	Transmit Data Hold After Transmit SPTx_CLK[4]	2		ns

[1] Specifications apply to all eight SPORTs.
[2] Referenced to sample edge.
[3] This specification indicates the minimum instantaneous width or period that can be tolerated due to duty cycle variation or jitter on the external SPTx_CLK. For the external SPTx_CLK ideal maximum frequency see the $f_{SPTCLKEXT}$ specification in Table 29.
[4] Referenced to drive edge.

PWM — Heightened Precision (HP) Mode Timing

Table 84 and Table 85 and Figure 56 and Figure 57 describe heightened precision (HP) PWM operations.

Table 84. PWM—HP Mode, Output Pulse

Parameter		Min	Max	Unit
Switching Characteristics				
t_{HPWMW}	HP PWM Output Pulse Width[1, 2]	$(N + m \times 0.25) \times t_{SCLK} - 0.5$	$(N + m \times 0.25) \times t_{SCLK} + 0.5$	ns

[1] N is the DUTY bit field (coarse duty) from the duty register. m is the ENHDIV (Enhanced Precision Divider bits) value from the HP duty register.
[2] Applies to individual PWM channel with 50% duty cycle. Other PWM channels within the same unit are toggling at the same time. No other GPIO pins toggle.

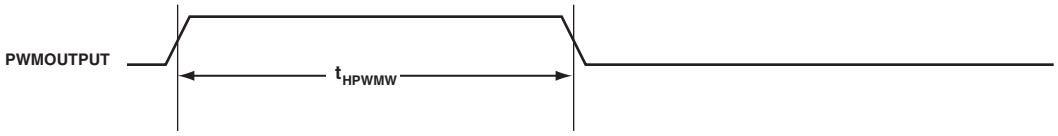


Figure 56. PWM HP Mode Timing, Output Pulse

Table 85. PWM—HP Mode, Output Skew

Parameter		Min	Max	Unit
Switching Characteristics				
t_{HPWMS}	HP PWM Output Skew [1]		1.0	ns

[1] Output edge difference between any two PWM channels (AH, AL, BH, BL, CH, CL, DH and DL) in the same PWM unit (a unit is PWMx where x = 0, 1, 2), with the same HP edge placement.

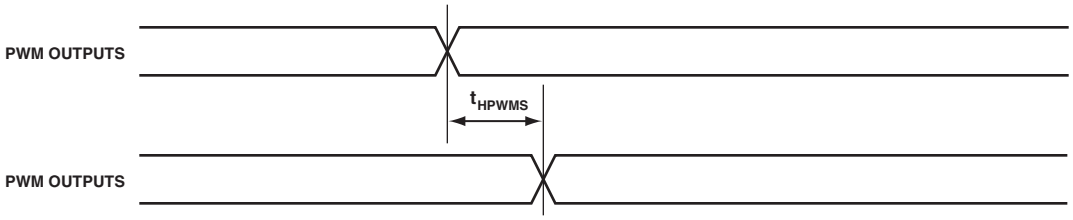


Figure 57. PWM HP Mode Timing, Output Skew

Universal Asynchronous Receiver-Transmitter (UART) Ports—Receive and Transmit Timing

The UART ports receive and transmit operations are described in the ADSP-SC58x/ADSP-2158x SHARC+ Processor Hardware Reference.

Controller Area Network (CAN) Interface

The CAN interface timing is described in the ADSP-SC58x/ADSP-2158x SHARC+ Processor Hardware Reference.

Universal Serial Bus (USB) OTG—Receive and Transmit Timing

Table 87 describes the USB OTG receive and transmit operations.

Table 87. USB OTG—Receive and Transmit Timing[1]

Parameter		Min	Max	Unit
Timing Requirements				
f_{USBS}	USB_XI Frequency	24	24	MHz
fs_{USB}	USB_XI Clock Frequency Stability	–50	+50	ppm

[1] This specification is supported by USB0.

PCI Express (PCIe)

The PCIe interface complies with the Gen1 and Gen2 x1 lane data rate specification and supports up to 3.0 PCIe base functionality.

For more information about PCIe, see the following standards:

- *PCI Express Base 3.0 Specification*, Revision 1.0, PCI-SIG
- *PCI Express 2.0 Card Electromechanical Specification*, Revision 2.0, PCI-SIG
- *PHY Interface for the PCI Express Architecture*, Revision 2.0, Intel Corporation
- *PCI-SIG Engineering Change Request: L1 Substates*, February 1, 2012, PCI-SIG
- *IEEE Standard 1149.1-2001*, IEEE
- *IEEE Standard 1149.6-2003*, IEEE

ADSP-SC58x/ADSP-2158x 529-BALL BGA BALL ASSIGNMENTS

The ADSP-SC58x/ADSP-2158x 529-Ball BGA Ball Assignments (Numerical by Ball Number) table lists the 529-ball BGA package by ball number.

The ADSP-SC58x/ADSP-2158x 529-Ball BGA Ball Assignments (Alphabetical by Pin Name) table lists the 529-ball BGA package by pin name.

ADSP-SC58x/ADSP-2158x 529-BALL BGA BALL ASSIGNMENTS (NUMERICAL BY BALL NUMBER)

Ball No.	Pin Name	Ball No.	Pin Name	Ball No.	Pin Name	Ball No.	Pin Name
A01	GND	B19	DMC1_DQ11	D14	DMC1_BA2	F09	GND
A02	DMC0_UDQS	B20	DMC1_DQ12	D15	DMC1_CAS	F10	VDD_INT
A03	DMC0_CK	B21	DMC1_DQ14	D16	DMC1_RAS	F11	VDD_INT
A04	DMC0_CK	B22	PD_00	D17	DMC1_A09	F12	VDD_INT
A05	DMC0_DQ09	B23	PD_04	D18	DMC1_A15	F13	VDD_INT
A06	DMC0_LDQS	C01	DMC0_DQ14	D19	DMC1_A10	F14	VDD_INT
A07	DMC0_LDQS	C02	DMC0_DQ13	D20	DMC1_A11	F15	VDD_INT
A08	DMC0_DQ05	C03	DMC0_CS0	D21	PC_14	F16	GND
A09	DMC0_DQ03	C04	DMC0_CKE	D22	PD_10	F17	VDD_INT
A10	DMC0_DQ01	C05	DMC0_LDM	D23	PD_09	F18	VDD_INT
A11	DMC1_DQ03	C06	DMC1_RESET	E01	DMC0_A04	F19	VDD_INT
A12	DMC1_DQ00	C07	DMC1_A03	E02	DMC0_RAS	F20	PE_06
A13	DMC1_LDQS	C08	DMC1_A00	E03	DMC0_BA1	F21	PD_02
A14	DMC1_LDQS	C09	DMC1_A01	E04	DMC0_WE	F22	PD_13
A15	DMC1_VREF	C10	DMC1_A04	E05	DMC0_RZQ	F23	PD_12
A16	DMC1_CK	C11	DMC1_A06	E06	GND	G01	DMC0_A13
A17	DMC1_CK	C12	DMC1_BA1	E07	GND	G02	DMC0_A09
A18	DMC1_DQ09	C13	DMC1_ODT	E08	GND	G03	DMC0_A03
A19	DMC1_UDQS	C14	DMC1_CS0	E09	GND	G04	DMC0_A11
A20	DMC1_UDQS	C15	DMC1_LDM	E10	VDD_INT	G05	VDD_INT
A21	DMC1_DQ13	C16	DMC1_UDM	E11	VDD_INT	G06	VDD_DMC
A22	DMC1_DQ15	C17	DMC1_A14	E12	VDD_INT	G07	VDD_DMC
A23	GND	C18	DMC1_A12	E13	VDD_INT	G08	VDD_DMC
B01	DMC0_UDQS	C19	DMC1_A13	E14	VDD_INT	G09	VDD_DMC
B02	DMC0_DQ12	C20	PC_13	E15	VDD_INT	G10	VDD_DMC
B03	DMC0_DQ11	C21	PD_01	E16	VDD_INT	G11	VDD_DMC
B04	DMC0_DQ10	C22	PD_06	E17	VDD_INT	G12	VDD_DMC
B05	DMC0_DQ08	C23	PD_05	E18	VDD_INT	G13	VDD_DMC
B06	DMC0_DQ06	D01	DMC0_VREF	E19	DMC1_RZQ	G14	VDD_DMC
B07	DMC0_DQ07	D02	DMC0_DQ15	E20	PC_15	G15	VDD_DMC
B08	DMC0_DQ04	D03	DMC0_BA0	E21	PD_08	G16	VDD_DMC
B09	DMC0_DQ02	D04	DMC0_BA2	E22	PD_14	G17	VDD_DMC
B10	DMC0_DQ00	D05	DMC0_ODT	E23	PD_11	G18	VDD_DMC
B11	DMC1_DQ01	D06	DMC0_UDM	F01	DMC0_A01	G19	VDD_INT
B12	DMC1_DQ02	D07	DMC1_A05	F02	DMC0_A06	G20	PE_04
B13	DMC1_DQ04	D08	DMC1_WE	F03	DMC0_CAS	G21	PE_13
B14	DMC1_DQ05	D09	DMC1_A07	F04	DMC0_A02	G22	PE_01
B15	DMC1_DQ06	D10	DMC1_A02	F05	DMC0_A07	G23	PE_00
B16	DMC1_DQ07	D11	DMC1_BA0	F06	GND	H01	DMC0_A14
B17	DMC1_DQ08	D12	DMC1_A08	F07	VDD_INT	H02	DMC0_A12
B18	DMC1_DQ10	D13	DMC1_CKE	F08	VDD_INT	H03	DMC0_A05

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		