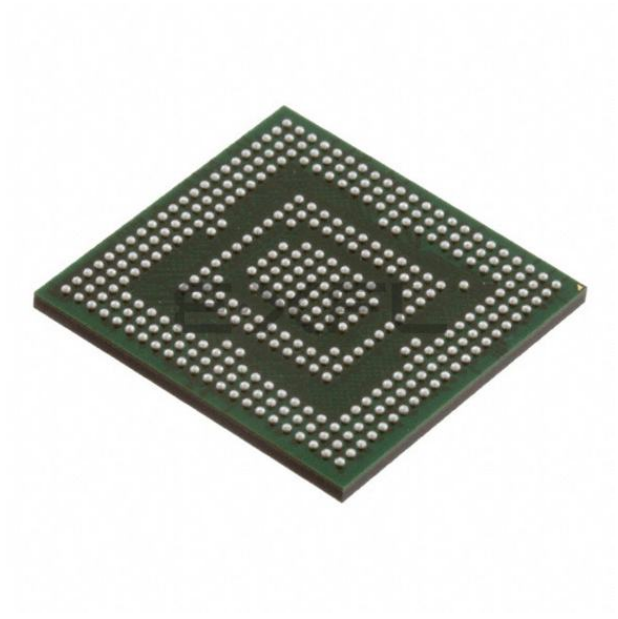


Welcome to E-XFL.COM

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, DDR2, DDR3, EBI/EMI, Ethernet, DAI, I²C, LPDDR1, MMC/SD/SDIO, SPI, SPORT, UART/USART, USB OTG
Clock Rate	450MHz
Non-Volatile Memory	ROM (512kB)
On-Chip RAM	640kB
Voltage - I/O	3.30V
Voltage - Core	1.10V
Operating Temperature	-40°C ~ 105°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/adsc582wcbcz4a10

TABLE OF CONTENTS

REVISION HISTORY

10/2016—Revision 0: Initial Version

The processing elements are referred to as PEx and PEy data registers and each contain an arithmetic logic unit (ALU), multiplier, shifter, and register file. PEx is always active and PEy is enabled by setting the PEYEN mode bit in the mode control register (MODE1).

Single instruction multiple data (SIMD) mode allows the processors to execute the same instruction in both processing elements, but each processing element operates on different data. This architecture efficiently executes math intensive DSP algorithms. In addition to all the features of previous generation SHARC cores, the SHARC+ core also provides a new and simpler way to execute an instruction only on the PEy data register.

SIMD mode also affects the way data transfers between memory and processing elements because to sustain computational operation in the processing elements requires twice the data bandwidth. Therefore, entering SIMD mode doubles the bandwidth between memory and the processing elements. When using the DAGs to transfer data in SIMD mode, two data values transfer with each memory or register file access.

Independent, Parallel Computation Units

Within each processing element is a set of pipelined computational units. The computational units consist of a multiplier, arithmetic/logic unit (ALU), and shifter. These units are arranged in parallel, maximizing computational throughput. These computational units support IEEE 32-bit single-precision floating-point, 40-bit extended-precision floating-point, IEEE 64-bit double-precision floating-point, and 32-bit fixed-point data formats.

A multifunction instruction set supports parallel execution of ALU and multiplier operations. In SIMD mode, the parallel ALU and multiplier operations occur in both processing elements per core.

All processing operations take one cycle to complete. For all floating-point operations, the processor takes two cycles to complete in case of data dependency. Double-precision floating-point data take two to six cycles to complete. The processor stalls for the appropriate number of cycles for an interlocked pipeline plus data dependency check.

Core Timer

Each SHARC+ processor core also has a timer. This extra timer is clocked by the internal processor clock and is typically used as a system tick clock for generating periodic operating system interrupts.

Data Register File

Each processing element contains a general-purpose data register file. The register files transfer data between the computation units and the data buses, and store intermediate results. These 10-port, 32-register register files (16 primary, 16 secondary), combined with the enhanced Harvard architecture of the processor, allow unconstrained data flow between computation units and internal memory. The registers in the PEx data register file are referred to as R0–R15 and in the PEy data register file as S0–S15.

Context Switch

Many of the registers of the processor have secondary registers that can activate during interrupt servicing for a fast context switch. The data, DAG, and multiplier result registers have secondary registers. The primary registers are active at reset, while control bits in MODE1 activate the secondary registers.

Universal Registers (USTAT)

General-purpose tasks use the universal registers. The four USTAT registers allow easy bit manipulations (set, clear, toggle, test, XOR) for all control and status peripheral registers.

The data bus exchange register (PX) permits data to pass between the 64-bit PM data bus and the 64-bit DM data bus or between the 40-bit register file and the PM or DM data bus. These registers contain hardware to handle the data width difference.

Data Address Generators With Zero-Overhead Hardware Circular Buffer Support

For indirect addressing and implementing circular data buffers in hardware, the ADSP-SC58x/ADSP-2158x processor uses the two data address generators (DAGs). Circular buffers allow efficient programming of delay lines and other data structures required in digital signal processing, and are commonly used in digital filters and Fourier transforms. The two DAGs of the processors contain sufficient registers to allow the creation of up to 32 circular buffers (16 primary register sets and 16 secondary sets). The DAGs automatically handle address pointer wraparound, reduce overhead, increase performance, and simplify implementation. Circular buffers can start and end at any memory location.

Flexible Instruction Set Architecture (ISA)

The ISA, a 48-bit instruction word, accommodates various parallel operations for concise programming. For example, the processors can conditionally execute a multiply, an add, and a subtract in both processing elements while branching and fetching up to four 32-bit values from memory—all in a single instruction. Additionally, the double-precision floating-point instruction set is an addition to the SHARC+ core.

Variable Instruction Set Architecture (VISA)

In addition to supporting the standard 48-bit instructions from previous SHARC processors, the SHARC+ core processors support 16-bit and 32-bit opcodes for many instructions, formerly 48-bit in the ISA. This feature, called variable instruction set architecture (VISA), drops redundant or unused bits within the 48-bit instruction to create more efficient and compact code. The program sequencer supports fetching these 16-bit and 32-bit instructions from both internal and external memories. VISA is not an operating mode; it is only address dependent (refer to memory map ISA/VISA address spaces in Table 7). Furthermore, it allows jumps between ISA and VISA instruction fetches.

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 15. Signal Multiplexing for Port C

Signal Name	Multiplexed Function 0	Multiplexed Function 1	Multiplexed Function 2	Multiplexed Function 3	Multiplexed Function Input Tap
PC_00	LP1_CLK	PWM0_BL	SPI0_SEL4	SMC0_ARE	
PC_01	SPI2_CLK				
PC_02	SPI2_MISO				
PC_03	SPI2_MOSI				
PC_04	SPI2_D2				
PC_05	SPI2_D3				
PC_06	SPI2_SEL1				SPI2_SS
PC_07	CAN0_RX	SPI0_SEL1		SMC0_AMS2	TM0_ACI3
PC_08	CAN0_TX			SMC0_AMS3	
PC_09	SPI0_CLK				
PC_10	SPI0_MISO				
PC_11	SPI0_MOSI				TM0_CLK
PC_12	SPI0_SEL3	SPI0_RDY	ACM0_T0	SMC0_A25	
PC_13	UART0_TX	SPI1_SEL1	ACM0_A0		
PC_14	UART0_RX		ACM0_A1		TM0_ACI0
PC_15	UART0_RTS	PPI0_FS3	ACM0_A2	SMC0_AMS0	

Table 16. Signal Multiplexing for Port D

Signal Name	Multiplexed Function 0	Multiplexed Function 1	Multiplexed Function 2	Multiplexed Function 3	Multiplexed Function Input Tap
PD_00	UART0_CTS	PPI0_D23	ACM0_A3	SMC0_D07	
PD_01	SPI0_SEL2		ACM0_A4	SMC0_AOE	SPI0_SS
PD_02	LP0_D0	PWM1_TRIP0	TRACE0_D00		
PD_03	LP0_D1	PWM1_AH	TRACE0_D01		
PD_04	LP0_D2	PWM1_AL	TRACE0_D02		
PD_05	LP0_D3	PWM1_BH	TRACE0_D03		
PD_06	LP0_D4	PWM1_BL	TRACE0_D04		
PD_07	LP0_D5	PWM1_CH	TRACE0_D05		
PD_08	LP0_D6	PWM1_CL	TRACE0_D06		TM0_ACLK1
PD_09	LP0_D7	PWM1_DH	TRACE0_D07		TM0_ACLK2
PD_10	LP0_CLK	PWM1_DL	TRACE0_CLK		
PD_11	LP0_ACK	PWM1_SYNC			
PD_12	UART2_TX		PPI0_D19	SMC0_A06	
PD_13	UART2_RX		PPI0_D18	SMC0_A05	TM0_ACI2
PD_14	PPI0_D11	PWM2_TRIP0	MLB0_CLKOUT	SMC0_D06	
PD_15	PPI0_D10	PWM2_CH		SMC0_D05	

529-BALL CSP_BGA SIGNAL DESCRIPTIONS

The processor pin definitions are shown Table 19 for the 529-ball CSP_BGA package. 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 description column provides a descriptive name for each signal.
- The port column shows whether or not a signal is multiplexed with other signals on a general-purpose I/O port pin.
- The pin name column identifies the name of the package pin (at power on reset) on which the signal is located (if a single function pin) or is multiplexed (if a general-purpose I/O pin).
- The DAI pins and their associated signal routing units (SRUs) connect inputs and outputs of the DAI peripherals (SPORT, ASRC, S/PDIF, and PCG). 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.

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

Signal Name	Description	Port	Pin Name
ACM0_A0	ACM0 ADC Control Signals	C	PC_13
ACM0_A1	ACM0 ADC Control Signals	C	PC_14
ACM0_A2	ACM0 ADC Control Signals	C	PC_15
ACM0_A3	ACM0 ADC Control Signals	D	PD_00
ACM0_A4	ACM0 ADC Control Signals	D	PD_01
ACM0_T0	ACM0 External Trigger n	C	PC_12
C1_FLG0	SHARC Core 1 Flag Pin	E	PE_01
C1_FLG1	SHARC Core 1 Flag Pin	E	PE_03
C1_FLG2	SHARC Core 1 Flag Pin	E	PE_05
C1_FLG3	SHARC Core 1 Flag Pin	E	PE_07
C2_FLG0	SHARC Core 2 Flag Pin	E	PE_02
C2_FLG1	SHARC Core 2 Flag Pin	E	PE_04
C2_FLG2	SHARC Core 2 Flag Pin	E	PE_06
C2_FLG3	SHARC Core 2 Flag Pin	E	PE_08
CAN0_RX	CAN0 Receive	C	PC_07
CAN0_TX	CAN0 Transmit	C	PC_08
CAN1_RX	CAN1 Receive	B	PB_10
CAN1_TX	CAN1 Transmit	B	PB_09
CNT0_DG	CNT0 Count Down and Gate	B	PB_14
CNT0_UD	CNT0 Count Up and Direction	B	PB_12
CNT0_ZM	CNT0 Count Zero Marker	B	PB_11
DAI0_PIN01	DAI0 Pin 1	Not Muxed	DAI0_PIN01
DAI0_PIN02	DAI0 Pin 2	Not Muxed	DAI0_PIN02
DAI0_PIN03	DAI0 Pin 3	Not Muxed	DAI0_PIN03
DAI0_PIN04	DAI0 Pin 4	Not Muxed	DAI0_PIN04
DAI0_PIN05	DAI0 Pin 5	Not Muxed	DAI0_PIN05
DAI0_PIN06	DAI0 Pin 6	Not Muxed	DAI0_PIN06
DAI0_PIN07	DAI0 Pin 7	Not Muxed	DAI0_PIN07
DAI0_PIN08	DAI0 Pin 8	Not Muxed	DAI0_PIN08
DAI0_PIN09	DAI0 Pin 9	Not Muxed	DAI0_PIN09
DAI0_PIN10	DAI0 Pin 10	Not Muxed	DAI0_PIN10
DAI0_PIN11	DAI0 Pin 11	Not Muxed	DAI0_PIN11
DAI0_PIN12	DAI0 Pin 12	Not Muxed	DAI0_PIN12
DAI0_PIN13	DAI0 Pin 13	Not Muxed	DAI0_PIN13
DAI0_PIN14	DAI0 Pin 14	Not Muxed	DAI0_PIN14

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

Signal Name	Description	Port	Pin Name
PWM2_AL	PWM2 Channel A Low Side	F	PF_06
PWM2_BH	PWM2 Channel B High Side	F	PF_09
PWM2_BL	PWM2 Channel B Low Side	F	PF_08
PWM2_CH	PWM2 Channel C High Side	D	PD_15
PWM2_CL	PWM2 Channel C Low Side	E	PE_00
PWM2_DH	PWM2 Channel D High Side	E	PE_04
PWM2_DL	PWM2 Channel D Low Side	E	PE_10
PWM2_SYNC	PWM2 PWMTMR Grouped	E	PE_05
$\overline{\text{PWM2\_TRIP0}}$	PWM2 Shutdown Input 0	D	PD_14
GND	Ground	Not Muxed	GND
VDD_EXT	External Voltage Domain	Not Muxed	VDD_EXT
VDD_INT	Internal Voltage Domain	Not Muxed	VDD_INT
RTC0_CLKIN	RTC0 Crystal input / external oscillator connection	Not Muxed	RTC0_CLKIN
RTC0_XTAL	RTC0 Crystal output	Not Muxed	RTC0_XTAL
SINC0_CLK0	SINC0 Clock 0	B	PB_01
SINC0_D0	SINC0 Data 0	A	PA_14
SINC0_D1	SINC0 Data 1	A	PA_15
SINC0_D2	SINC0 Data 2	B	PB_00
SINC0_D3	SINC0 Data 3	B	PB_04
SMC0_A01	SMC0 Address 1	B	PB_05
SMC0_A02	SMC0 Address 2	B	PB_06
SMC0_A03	SMC0 Address 3	B	PB_03
SMC0_A04	SMC0 Address 4	B	PB_02
SMC0_A05	SMC0 Address 5	D	PD_13
SMC0_A06	SMC0 Address 6	D	PD_12
SMC0_A07	SMC0 Address 7	B	PB_01
SMC0_A08	SMC0 Address 8	B	PB_00
SMC0_A09	SMC0 Address 9	A	PA_15
SMC0_A10	SMC0 Address 10	A	PA_14
SMC0_A11	SMC0 Address 11	A	PA_09
SMC0_A12	SMC0 Address 12	A	PA_08
SMC0_A13	SMC0 Address 13	A	PA_13
SMC0_A14	SMC0 Address 14	A	PA_12
SMC0_A15	SMC0 Address 15	A	PA_11
SMC0_A16	SMC0 Address 16	A	PA_07
SMC0_A17	SMC0 Address 17	A	PA_06
SMC0_A18	SMC0 Address 18	A	PA_05
SMC0_A19	SMC0 Address 19	A	PA_04
SMC0_A20	SMC0 Address 20	A	PA_01
SMC0_A21	SMC0 Address 21	A	PA_00
SMC0_A22	SMC0 Address 22	A	PA_10
SMC0_A23	SMC0 Address 23	A	PA_03
SMC0_A24	SMC0 Address 24	A	PA_02
SMC0_A25	SMC0 Address 25	C	PC_12
$\overline{\text{SMC0\_ABE0}}$	SMC0 Byte Enable 0	E	PE_14
$\overline{\text{SMC0\_ABE1}}$	SMC0 Byte Enable 1	E	PE_15
$\overline{\text{SMC0\_AMS0}}$	SMC0 Memory Select 0	C	PC_15
$\overline{\text{SMC0\_AMS1}}$	SMC0 Memory Select 1	E	PE_13

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
PB_09	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTB Position 9 | CAN1 Transmit | LP1 Data 2 | SMC0 Data 13 Notes: No notes
PB_10	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTB Position 10 | CAN1 Receive | LP1 Data 3 | SMC0 Data 12 | TIMER0 Timer 2 | TIMER0 Alternate Capture Input 4 Notes: No notes
PB_11	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTB Position 11 | LP1 Data 4 | PWM0 Channel D High Side | SMC0 Data 11 | CNT0 Count Zero Marker Notes: No notes
PB_12	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTB Position 12 | LP1 Data 5 | PWM0 Channel D Low Side | SMC0 Data 10 | CNT0 Count Up and Direction Notes: No notes
PB_13	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTB Position 13 | LP1 Data 6 | PWM0 Channel C High Side | SMC0 Data 9 Notes: No notes
PB_14	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTB Position 14 | LP1 Data 7 | PWM0 Channel C Low Side | SMC0 Data 8 | TIMER0 Timer 5 | CNT0 Count Down and Gate Notes: No notes
PB_15	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTB Position 15 | LP1 Acknowledge | PWM0 Shutdown Input 0 | SMC0 Write Enable | TIMER0 Timer 1 Notes: No notes
PCIE0_CLKM	Input	NA	PullDown	none	none	VDD_PCIE	Desc: PCIE0 CLK - Notes: No notes
PCIE0_CLKP	Input	NA	PullDown	none	none	VDD_PCIE	Desc: PCIE0 CLK + Notes: No notes
PCIE0_REF	a	NA	PullDown	none	none	VDD_PCIE	Desc: PCIE0 Reference Notes: No notes
PCIE0_RXM	Input	NA	PullDown	none	none	VDD_PCIE_RX	Desc: PCIE0 RX - Notes: No notes
PCIE0_RXP	Input	NA	PullDown	none	none	VDD_PCIE_RX	Desc: PCIE0 RX + Notes: No notes
PCIE0_TXM	InOut	J	PullDown	none	none	VDD_PCIE_TX	Desc: PCIE0 TX - Notes: No notes
PCIE0_TXP	InOut	J	PullDown	none	none	VDD_PCIE_TX	Desc: PCIE0 TX + 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
PG_04	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTG Position 4 | EMAC1 Receive Data 0 | TRACE0 Trace Data | TRACE0 Trace Data 14 Notes: No notes
PG_05	InOut	A	PullDown	none	none	VDD_EXT	Desc: PORTG Position 5 | EMAC1 Receive Data 1 | TRACE0 Trace Data | TRACE0 Trace Data 15 Notes: No notes
RTC0_CLKIN	a	NA	none	none	none	VDD_RTC	Desc: RTC0 Crystal input / external oscillator connection Notes: Connect to GND if not used
RTC0_XTAL	a	NA	none	none	none	VDD_RTC	Desc: RTC0 Crystal output Notes: No notes
SYS_BMODE0	Input	NA	PullDown	none	none	VDD_EXT	Desc: Boot Mode Control n Notes: No notes
SYS_BMODE1	Input	NA	PullDown	none	none	VDD_EXT	Desc: Boot Mode Control n Notes: No notes
SYS_BMODE2	Input	NA	PullDown	none	none	VDD_EXT	Desc: Boot Mode Control n Notes: No notes
SYS_CLKIN0	a	NA	none	none	none	VDD_EXT	Desc: Clock/Crystal Input Notes: No notes
SYS_CLKIN1	a	NA	none	none	none	VDD_EXT	Desc: Clock/Crystal Input Notes: Connect to GND if not used
SYS_CLKOUT	a	A	none	none	none		Desc: Processor Clock Output Notes: No notes
SYS_FAULT	InOut	A	none	none	none		Desc: Active-High Fault Output Notes: External pull-down required to keep signal in de-asserted state
$\overline{\text{SYS\_FAULT}}$	InOut	A	none	none	none		Desc: Active-Low Fault Output Notes: External pull-up required to keep signal in de-asserted state
$\overline{\text{SYS\_HWRST}}$	Input	NA	none	none	none	VDD_EXT	Desc: Processor Hardware Reset Control Notes: No notes
$\overline{\text{SYS\_RESOUT}}$	Output	A	none	none	L	VDD_EXT	Desc: Reset Output Notes: No notes
SYS_XTAL0	a	NA	none	none	none	VDD_EXT	Desc: Crystal Output Notes: No notes
SYS_XTAL1	a	NA	none	none	none	VDD_EXT	Desc: Crystal Output Notes: No notes
TWI0_SCL	InOut	D	none	none	none	VDD_EXT	Desc: TWI0 Serial Clock Notes: Add external pull-up if used. Can be pulled low when not used.

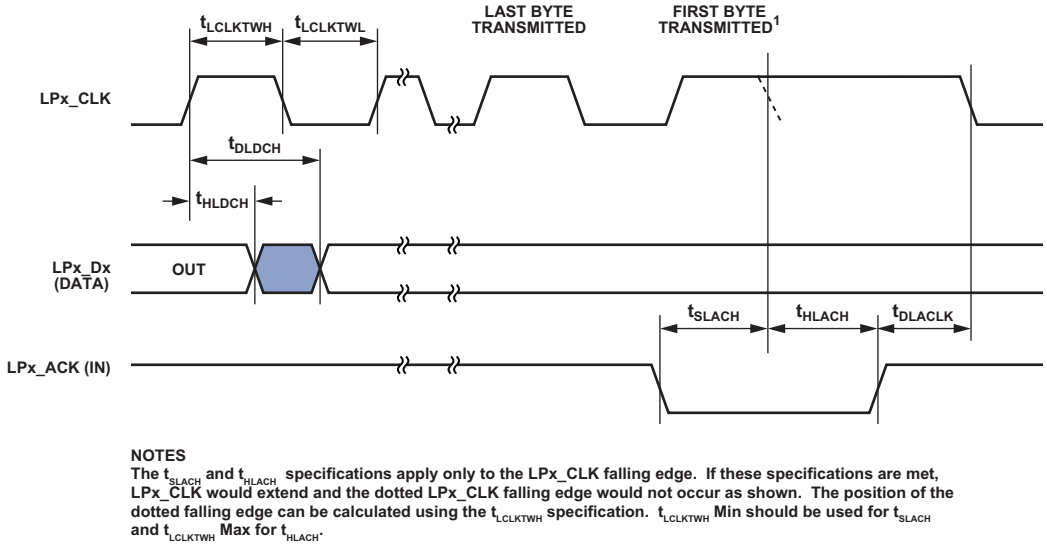
Table 27. ADSP-SC58x/ADSP-2158x Designer Quick Reference (Continued)

Signal Name	Type	Driver Type	Int Term	Reset Term	Reset Drive	Power Domain	Description and Notes
TWI0_SDA	InOut	D	none	none	none	VDD_EXT	Desc: TWI0 Serial Data Notes: Add external pull-up if used. Can be pulled low when not used.
TWI1_SCL	InOut	D	none	none	none	VDD_EXT	Desc: TWI1 Serial Clock Notes: Add external pull-up if used. Can be pulled low when not used.
TWI1_SDA	InOut	D	none	none	none	VDD_EXT	Desc: TWI1 Serial Data Notes: Add external pull-up if used. Can be pulled low when not used.
TWI2_SCL	InOut	D	none	none	none	VDD_EXT	Desc: TWI2 Serial Clock Notes: Add external pull-up if used. Can be pulled low when not used.
TWI2_SDA	InOut	D	none	none	none	VDD_EXT	Desc: TWI2 Serial Data Notes: Add external pull-up if used. Can be pulled low when not used.
USB0_DM	InOut	F	none	none	none	VDD_USB	Desc: USB0 Data - Notes: Add external pull-down if not used[1]
USB0_DP	InOut	F	none	none	none	VDD_USB	Desc: USB0 Data + Notes: Add external pull-down if not used[1]
USB0_ID	InOut		none	none	none	VDD_USB	Desc: USB0 OTG ID Notes: Connect to GND when USB is not used[1]
USB0_VBC	InOut	E	none	none	none	VDD_USB	Desc: USB0 VBUS Control Notes: Add external pull-down if not used[1]
USB0_VBUS	InOut	G	none	none	none	VDD_USB	Desc: USB0 Bus Voltage Notes: Connect to GND if not used[1]
USB1_DM	InOut	F	none	none	none	VDD_USB	Desc: USB1 Data - Notes: Add external pull-down if not used[1]
USB1_DP	InOut	F	none	none	none	VDD_USB	Desc: USB1 Data + Notes: Add external pull-down if not used[1]
USB1_VBUS	InOut	G	none	none	none	VDD_USB	Desc: USB1 Bus Voltage Notes: Connect to GND if not used[1]
USB_CLKIN	a		none	none	none		Desc: USB0/USB1 Clock/Crystal Input Notes: Services both USB0 and USB1. Connect to GND if not used.[1]

Table 63. Link Ports—Transmit[1]

Parameter		Min	Max	Unit
Timing Requirements				
t_{SLACH}	LPx_ACK Setup Before LPx_CLK Low	$2 \times t_{CLKO8} + 13.5$		ns
t_{HLACH}	LPx_ACK Hold After LPx_CLK Low	–5.5		ns
Switching Characteristics				
t_{DLDCH}	Data Delay After LPx_CLK High		1.6	ns
t_{HLDCH}	Data Hold After LPx_CLK High	–0.8		ns
$t_{LCLKTWL}$[2]	LPx_CLK Width Low	$0.33 \times t_{LCLKTPROG}$	$0.6 \times t_{LCLKTPROG}$	ns
$t_{LCLKTWH}$[2]	LPx_CLK Width High	$0.45 \times t_{LCLKTPROG}$	$0.66 \times t_{LCLKTPROG}$	ns
t_{LCLKTW}[2]	LPx_CLK Period	$N \times t_{LCLKTPROG} - 0.5$		ns
t_{DLACLK}	LPx_CLK Low Delay After LPx_ACK High	$t_{CLKO8} + 4$	$2 \times t_{CLKO8} + 1 \times t_{LPCLK} + 10$	ns

[1] Specifications apply to LP0 and LP1.
[2] See Table 29 for details on the minimum period that can be programmed for $t_{LCLKTPROG}$.



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

Figure 36. Link Ports—Transmit

ADC Controller Module (ACM) Timing

Table 86 and Figure 58 describe ACM operations.

When internally generated, the programmed ACM clock ($f_{ACLKPROG}$) frequency in MHz is set by the following equation where CKDIV is a field in the ACM_TC0 register and ranges from 1 to 255:

$$f_{ACLKPROG} = \frac{f_{SCLK1}}{CKDIV + 1}$$

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

Setup cycles (SC) in Table 86 is also a field in the ACM0_TC0 register and ranges from 0 to 4095. Hold cycles (HC) is a field in the ACM0_TC1 register that ranges from 0 to 15.

Table 86. ACM Timing

Parameter		Min	Max	Unit
Timing Requirements				
t_{SDR}	SPORT DRxPRI/DRxSEC Setup Before ACMx_CLK	3.5		ns
t_{HDR}	SPORT DRxPRI/DRxSEC Hold After ACMx_CLK	1.5		ns
Switching Characteristics				
t_{SCTLCS}	ACM Controls (ACMx_A[4:0]) Setup Before Assertion of $\overline{CS}$	$(SC + 1) \times t_{SCLK1} - 3$		ns
t_{HCTLCS}	ACM Control (ACMx_A[4:0]) Hold After Deassertion of $\overline{CS}$	$HC \times t_{ACLKPROG} - 1$		ns
t_{ACLKW}	ACM Clock Pulse Width[1]	$(0.5 \times t_{ACLKPROG}) - 1.5$		ns
t_{ACLK}	ACM Clock Period[1]	$t_{ACLKPROG} - 1.5$		ns
$t_{HCSACLK}$	$\overline{CS}$ Hold to ACMx_CLK Edge	–2.5		ns
$t_{SCSACLK}$	$\overline{CS}$ Setup to ACMx_CLK Edge	$t_{ACLKPROG} - 3.5$		ns

[1] See Table 29 for details on the minimum period that can be programmed for $t_{ACLKPROG}$.

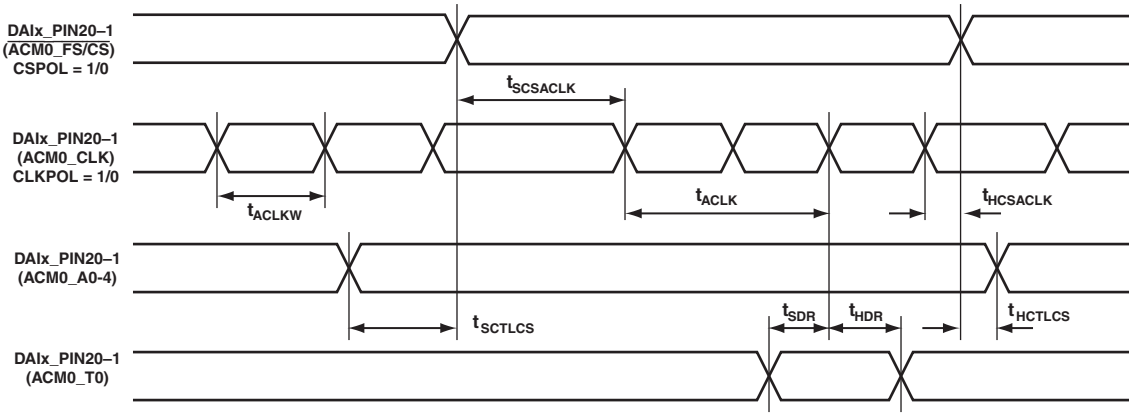


Figure 58. ACM Timing

Sinus Cardinalis (SINC) Filter Timing

The programmed SINC filter clock ($f_{SINCLKPROG}$) frequency in MHz is set by the following equation where MDIV is a field in the CLK control register that can be set from 4 to 63:

$$f_{SINCLKPROG} = \frac{f_{SCLK}}{MDIV}$$

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

Table 92. SINC Timing

Parameter		Min	Max	Unit
Timing Requirements				
t_{SSINC}	SINC0_Dx Setup Before SINC0_CLKx Rise	13.5		ns
t_{HSINC}	SINC0_Dx Hold After SINC0_CLKx Rise	0		ns
Switching Characteristics				
t_{SINCLK}	SINC0_CLKx Period[1]	$t_{SINCLKPROG} - 2.5$		ns
$t_{SINCLKW}$	SINC0_CLKx Width[1]	$0.5 \times t_{SINCLKPROG} - 2.5$		ns

[1] See Table 29 for details on the minimum period that may be programmed for $t_{SINCLKPROG}$.

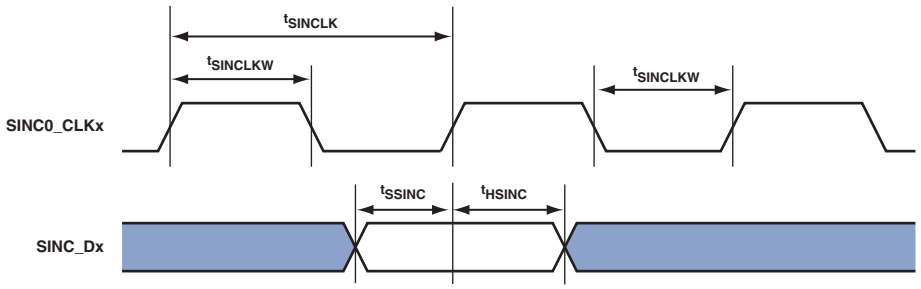


Figure 63. SINC Timing

Sony/Philips Digital Interface (S/PDIF) Transmitter

Serial data input to the S/PDIF transmitter can be formatted as left justified, I²S, or right justified with word widths of 16, 18, 20, or 24 bits. The following sections provide timing for the transmitter.

S/PDIF Transmitter Serial Input Waveforms

Figure 64 and Table 93 show the right justified mode. Frame sync is high for the left channel and low for the right channel. Data is valid on the rising edge of serial clock. The MSB is delayed the minimum in 24-bit output mode or the maximum in 16-bit output mode from a frame sync transition, so that when there are 64 serial clock periods per frame sync period, the LSB of the data is right justified to the next frame sync transition.

Table 93. S/PDIF Transmitter Right Justified Mode

Parameter		Conditions	Nominal	Unit
Timing Requirement				
t_{RJD}	Frame Sync to MSB Delay in Right Justified Mode	16-bit word mode	16	SCLK
		18-bit word mode	14	SCLK
		20-bit word mode	12	SCLK
		24-bit word mode	8	SCLK

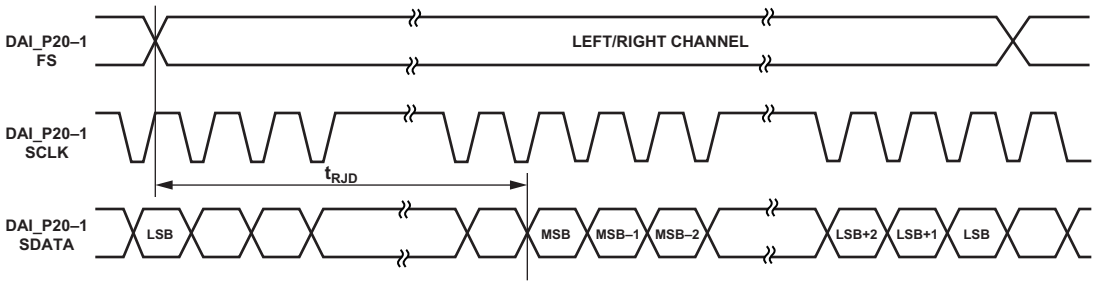


Figure 64. Right Justified Mode

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

Table 94. S/PDIF Transmitter I²S Mode

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

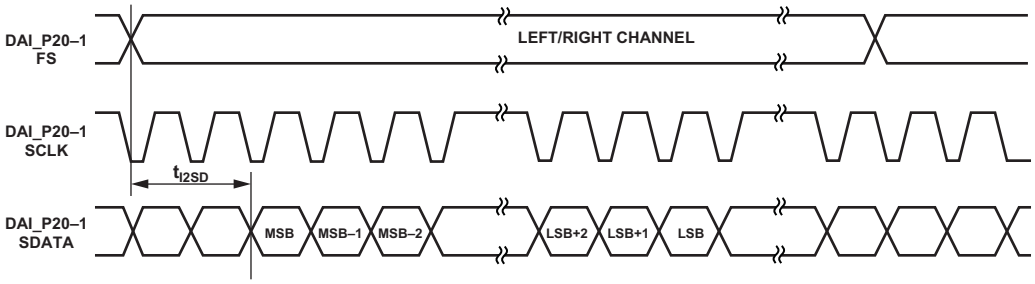


Figure 65. I²S Justified Mode

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

Table 95. S/PDIF Transmitter Left Justified Mode

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

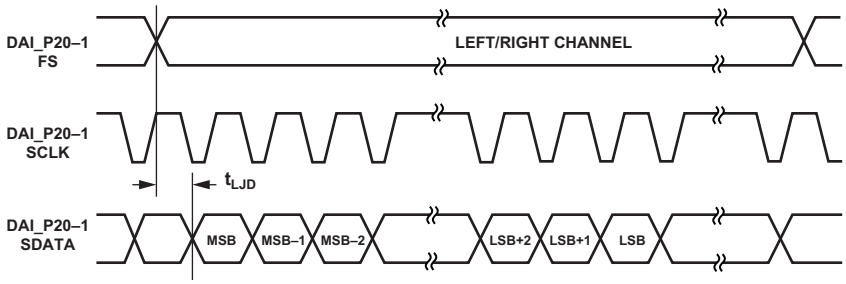


Figure 66. Left Justified Mode

S/PDIF Receiver

The following section describes timing as it relates to the S/PDIF receiver.

Internal Digital PLL Mode

In the internal digital PLL mode, the internal digital PLL generates the 512 × FS clock.

Table 98. S/PDIF Receiver Internal Digital PLL Mode Timing

Parameter		Min	Max	Unit
Switching Characteristics				
t_{DFSI}	Frame Sync Delay After Serial Clock		5	ns
t_{HOFSI}	Frame Sync Hold After Serial Clock	–2		ns
t_{DDTI}	Transmit Data Delay After Serial Clock		5	ns
t_{HDTI}	Transmit Data Hold After Serial Clock	–2		ns

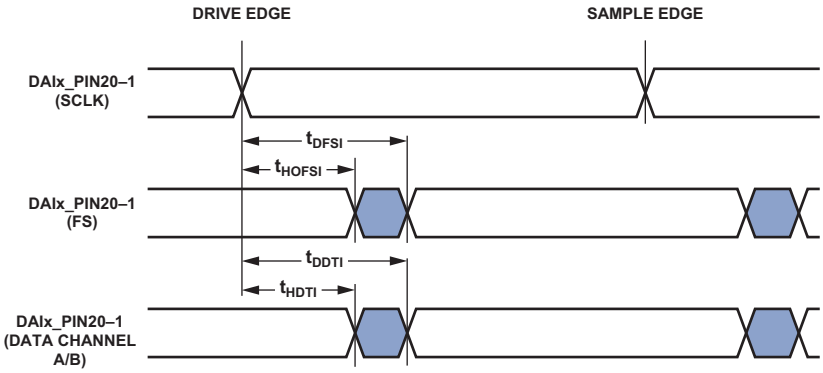


Figure 68. S/PDIF Receiver Internal Digital PLL Mode Timing

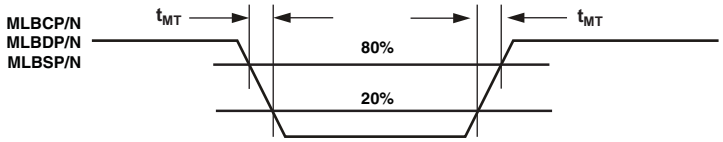


Figure 70. MLB 6-Pin Transition Time

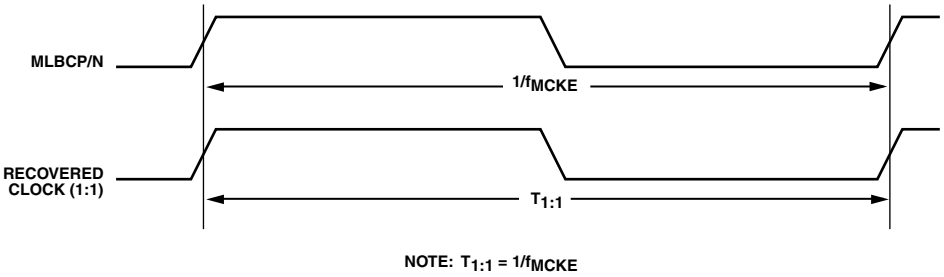


Figure 71. MLB 6-Pin Clock Definitions

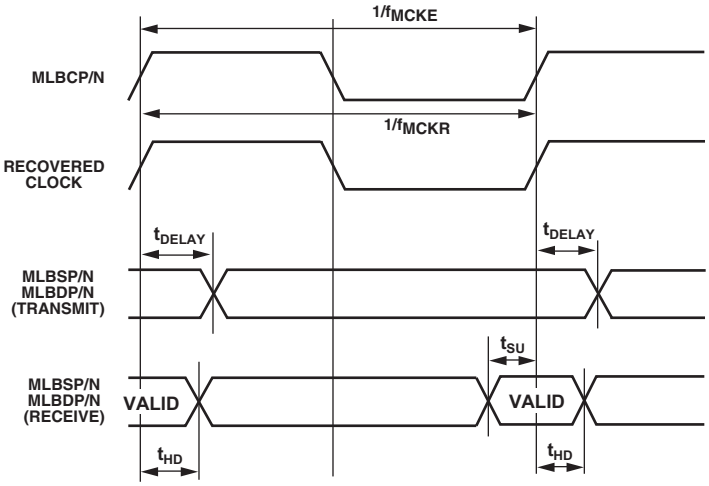


Figure 72. MLB 6-Pin Delay, Setup, and Hold Times

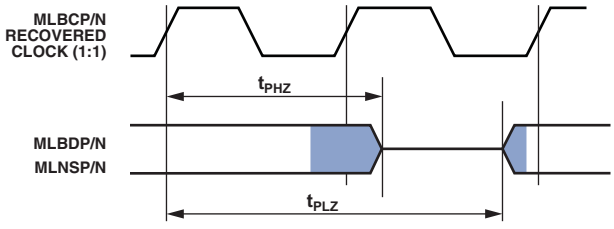


Figure 73. MLB 6-Pin Disable and Enable Turnaround Times

CONFIGURATION OF THE 349-BALL CSP_BGA

Figure 98 shows an overview of signal placement on the 349-ball CSP_BGA.

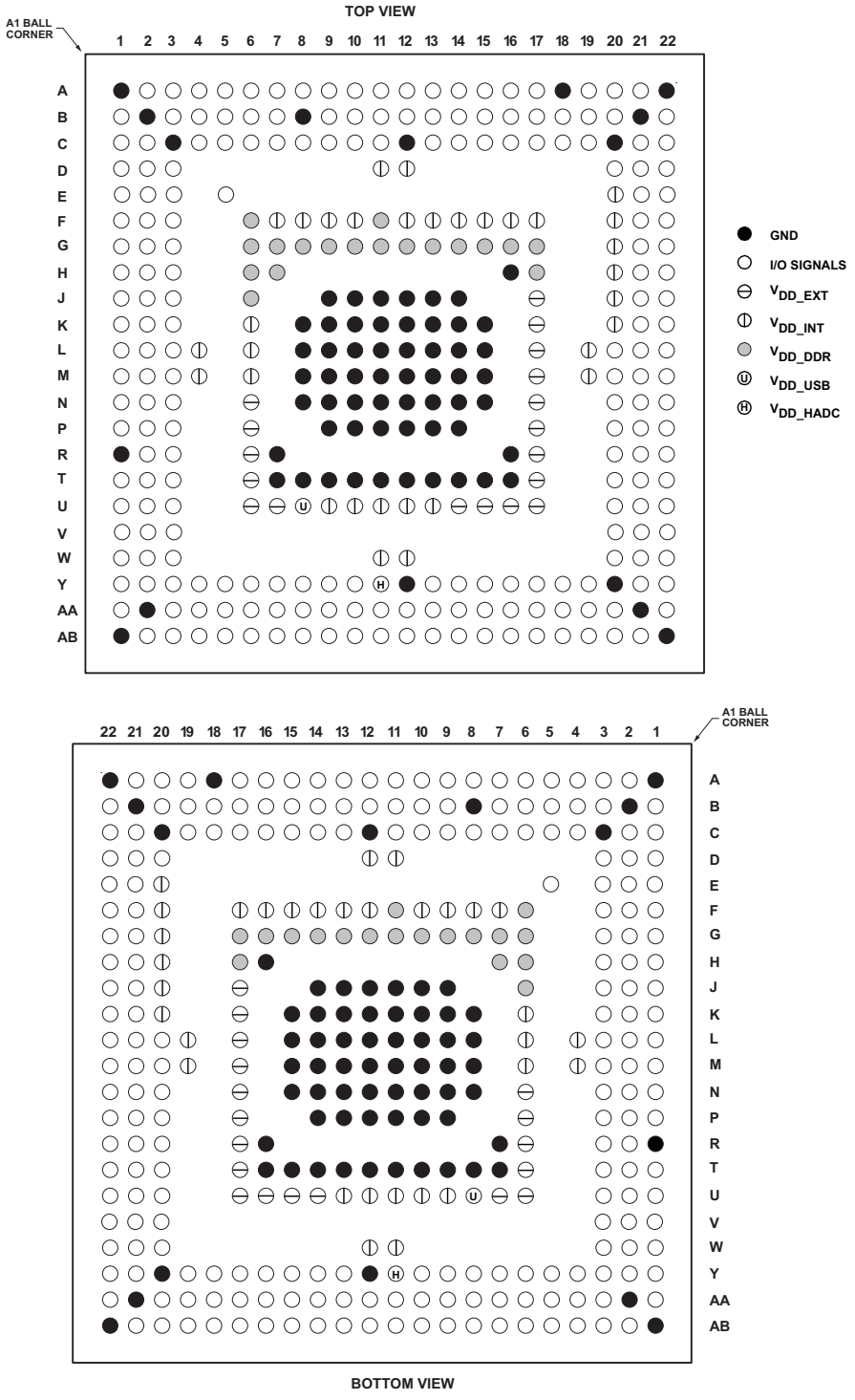


Figure 98. 349-Ball CSP_BGA Configuration

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.
GND	L16	GND	T08	PA_02	AB23	PC_11	K02
GND	L17	GND	T09	PA_03	AC22	PC_12	L02
GND	M07	GND	T10	PA_04	AB22	PC_13	C20
GND	M08	GND	T11	PA_05	AA20	PC_14	D21
GND	M09	GND	T12	PA_06	AB21	PC_15	E20
GND	M10	GND	T13	PA_07	AC20	PD_00	B22
GND	M11	GND	T14	PA_08	AC21	PD_01	C21
GND	M12	GND	T15	PA_09	AA19	PD_02	F21
GND	M13	GND	T16	PA_10	Y19	PD_03	J19
GND	M14	GND	T17	PA_11	AB20	PD_04	B23
GND	M15	GND	U07	PA_12	Y18	PD_05	C23
GND	M16	GND	U08	PA_13	Y17	PD_06	C22
GND	M17	GND	U09	PA_14	Y16	PD_07	J20
GND	N07	GND	U10	PA_15	AB19	PD_08	E21
GND	N08	GND	U11	PB_00	AA18	PD_09	D23
GND	N09	GND	U12	PB_01	AC19	PD_10	D22
GND	N10	GND	U13	PB_02	AA15	PD_11	E23
GND	N11	GND	U14	PB_03	AA17	PD_12	F23
GND	N12	GND	U15	PB_04	AA16	PD_13	F22
GND	N13	GND	U16	PB_05	Y15	PD_14	E22
GND	N14	GND	U17	PB_06	AA14	PD_15	K20
GND	N15	GND	Y14	PB_07	AA02	PE_00	G23
GND	N16	GND	AC01	PB_08	AA01	PE_01	G22
GND	N17	GND	AC14	PB_09	W02	PE_02	H23
GND	P07	GND	AC23	PB_10	Y02	PE_03	L20
GND	P08	HADC0_VIN0	Y12	PB_11	Y01	PE_04	G20
GND	P09	HADC0_VIN1	AA12	PB_12	W01	PE_05	H22
GND	P10	HADC0_VIN2	AB13	PB_13	V02	PE_06	F20
GND	P11	HADC0_VIN3	AB14	PB_14	T04	PE_07	J23
GND	P12	HADC0_VIN4	V12	PB_15	T02	PE_08	M19
GND	P13	HADC0_VIN5	AA13	PCIE0_CLKM	AC04	PE_09	L22
GND	P14	HADC0_VIN6	W12	PCIE0_CLKP	AC05	PE_10	K23
GND	P15	HADC0_VIN7	Y13	PCIE0_REF	AA07	PE_11	M20
GND	P16	HADC0_VREFN	AB12	PCIE0_RXM	AC03	PE_12	H21
GND	P17	HADC0_VREFP	AC12	PCIE0_RXP	AC02	PE_13	G21
GND	R07	JTG_TCK	P04	PCIE0_TXM	AC07	PE_14	L23
GND	R08	JTG_TDI	P02	PCIE0_TXP	AC06	PE_15	N20
GND	R09	JTG_TDO	P01	PC_00	U03	PF_00	M22
GND	R10	JTG_TMS	N01	PC_01	M01	PF_01	J22
GND	R11	$\overline{\text{JTG\_TRST}}$	N02	PC_02	M03	PF_02	M23
GND	R12	MLB0_CLKN	AB18	PC_03	N04	PF_03	M21
GND	R13	MLB0_CLKP	AC18	PC_04	L01	PF_04	N21
GND	R14	MLB0_DATN	AB17	PC_05	M02	PF_05	N22
GND	R15	MLB0_DATP	AC17	PC_06	K03	PF_06	K22
GND	R16	MLB0_SIGN	AB16	PC_07	L03	PF_07	N23
GND	R17	MLB0_SIGP	AC16	PC_08	J04	PF_08	P20
GND	T03	PA_00	Y20	PC_09	K04	PF_09	L21
GND	T07	PA_01	AA21	PC_10	L04	PF_10	P19

D13317-0-10/16(0)