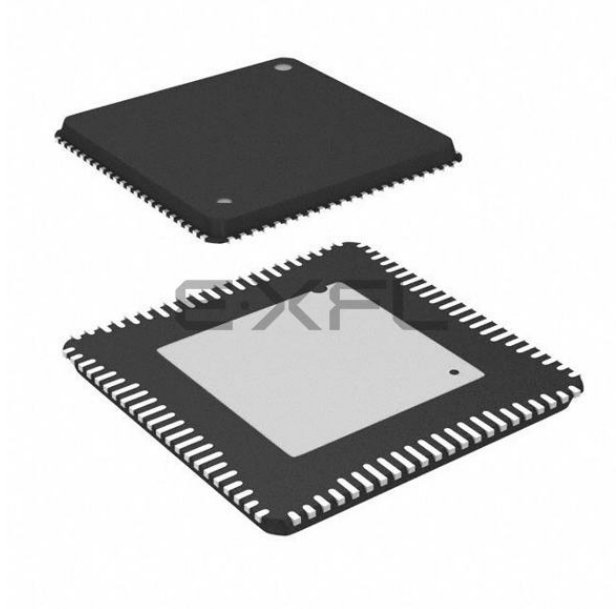


Welcome to E-XFL.COM

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	Sigma
Interface	I²C, SPI
Clock Rate	294.912MHz
Non-Volatile Memory	ROM (64kB)
On-Chip RAM	192kB
Voltage - I/O	3.30V
Voltage - Core	1.20V
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	88-VFQFN Exposed Pad, CSP
Supplier Device Package	88-LFCSP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/adau1463wbcpz300

ELECTRICAL CHARACTERISTICS

Digital Input/Output

Table 4.

Parameter	Min	Typ	Max	Unit	Test Conditions/Comments
DIGITAL INPUT					
Input Voltage					Excluding SPDIFIN, which is not a standard digital input
IOVDD = 3.3 V					
High Level (V_{IH})	1.71		3.3	V	
Low Level (V_{IL})	0		1.71	V	
IOVDD = 1.8 V					
High Level (V_{IH})	0.92		1.8	V	
Low Level (V_{IL})	0		0.89	V	
Input Leakage					
High Level (I_{IH})			2	µA	Digital input pins with pull-up resistor
			14	µA	Digital input pins with pull-down resistor
			2	µA	Digital input pins with no pull resistor
			8	µA	MCLK
			120	µA	SPDIFIN
Low Level (I_{IL}) at 0 V	−14			µA	Digital input pins with pull-up resistor
	−2			µA	Digital input pins with pull-down resistor
	−2			µA	Digital input pins with no pull resistor
	−8			µA	MCLK
	−120			µA	SPDIFIN
Input Capacitance (C_I)		2		pF	
DIGITAL OUTPUT					
Output Voltage					
IOVDD = 3.3 V					
High Level (V_{OH})	3.09		3.3	V	I_{OH} = 1 mA
Low Level (V_{OL})	0		0.26	V	I_{OL} = 1 mA
IOVDD = 1.8 V					
High Level (V_{OH})	1.45		1.8	V	
Low Level (V_{OL})	0		0.33	V	
Digital Output Pins, Output Drive					The digital output pins are driving low impedance PCB traces to a high impedance digital input buffer
IOVDD = 1.8 V					
Drive Strength Setting					
Lowest			1	mA	The digital output pins are not designed for static current draw; do not use these pins to drive LEDs directly
Low			2	mA	The digital output pins are not designed for static current draw; do not use these pins to drive LEDs directly
High			3	mA	The digital output pins are not designed for static current draw; do not use these pins to drive LEDs directly
Highest			5	mA	The digital output pins are not designed for static current draw; do not use these pins to drive LEDs directly
IOVDD = 3.3 V					
Drive Strength Setting					
Lowest			2	mA	The digital output pins are not designed for static current draw; do not use these pins to drive LEDs directly
Low			5	mA	The digital output pins are not designed for static current draw; do not use these pins to drive LEDs directly
High			10	mA	The digital output pins are not designed for static current draw; do not use these pins to drive LEDs directly
Highest			15	mA	The digital output pins are not designed for static current draw; do not use these pins to drive LEDs directly

I²C Interface—Slave

T_A = −40°C to +105°C, DVDD = 1.2 V ± 5%, IOVDD = 1.8 V − 5% to 3.3 V + 10%.

Table 11.

Parameter	Min	Typ	Max	Unit	Description
f_{SCL}			1000	kHz	SCL clock frequency
t_{SCLH}	0.26			µs	SCL pulse width high
t_{SCLL}	0.5			µs	SCL pulse width low
t_{SCS}	0.26			µs	Start and repeated start condition setup time
t_{SCH}	0.26			µs	Start condition hold time
t_{DS}	50			ns	Data setup time
t_{DH}			0.45	µs	Data hold time
t_{SCLR}			120	ns	SCL rise time
t_{SCLF}			120	ns	SCL fall time
t_{SDR}			120	ns	SDA rise time
t_{SDF}			120	ns	SDA fall time
t_{BFT}	0.5			µs	Bus free time between stop and start
t_{SUSTO}	0.26			µs	Stop condition setup time

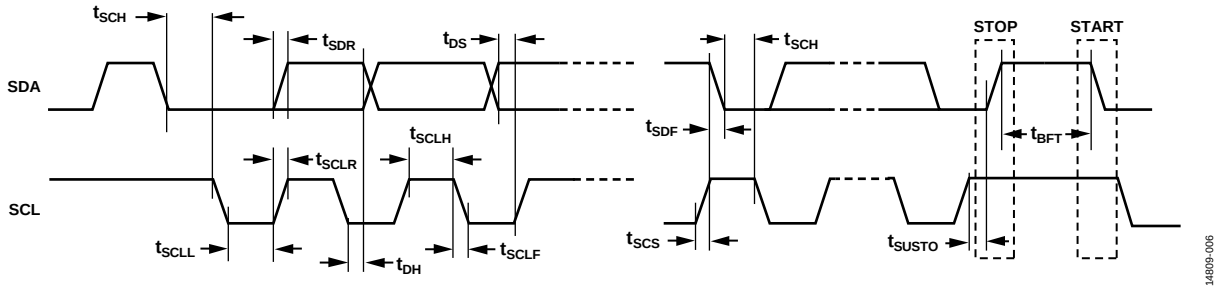


Figure 6. I²C Slave Port Timing Specifications

Example PLL Settings

Depending on the input clock frequency, there are several possible configurations for the PLL. Setting the PLL to generate the highest possible system clock, without exceeding the maximum, allows the execution of more DSP program instructions for each audio frame. Alternatively, setting the PLL to generate a lower frequency system clock allows fewer instructions to be executed and lowers overall power consumption of the device. Table 20 shows several example MCLK frequencies and the corresponding PLL settings that allow the highest number of program instructions to be executed for each audio frame. The settings provide the highest possible system clock without exceeding the 294.912 MHz upper limit.

Table 20. Optimal Predivider and Feedback Divider Settings for Varying Input MCLK Frequencies

Input MCLK Frequency (MHz)	Predivider Setting	PLL Input Clock (MHz)	Feedback Divider Setting	ADAU1463/ADAU1467 Fast Grade System Clock (MHz)	ADAU1463 Slow Grade System Clock (MHz)
2.8224	1	2.8224	104	293.5296	146.7648
3	1	3	98	294	147
3.072	1	3.072	96	294.912	147.456
3.5	1	3.5	84	294	147
4	1	4	73	292	146
4.5	1	4.5	65	292.5	146.25
5	2	2.5	117	292.5	146.25
5.5	2	2.75	107	294.25	147.125
5.6448	2	2.8224	104	293.5296	146.7648
6	2	3	98	294	147
6.144	2	3.072	96	294.912	147.456
6.5	2	3.25	90	292.5	146.25
7	2	3.5	84	294	147
7.5	2	3.75	78	292.5	146.25
8	2	4	73	292	146
8.5	2	4.25	69	293.25	146.625
9	2	4.5	65	292.5	146.25
9.5	4	2.375	124	294.5	147.25
10	4	2.5	117	292.5	146.25
10.5	4	2.625	112	294	147
11	4	2.75	107	294.25	147.125
11.2896	4	2.8224	104	293.5296	146.7648
11.5	4	2.875	102	293.25	146.625
12	4	3	98	294	147
12.288	4	3.072	96	294.912	147.456
12.5	4	3.125	94	293.75	146.875
13	4	3.25	90	292.5	146.25
13.5	4	3.375	87	293.625	146.8125
14	4	3.5	84	294	147
14.5	4	3.625	81	293.625	146.8125
15	4	3.75	78	292.5	146.25
15.5	4	3.875	76	294.5	147.25
16	4	4	73	292	146
16.5	4	4.125	71	292.875	146.4375
17	4	4.25	69	293.25	146.625
17.5	4	4.375	67	293.125	146.5625
18	4	4.5	65	292.5	146.25
18.5	8	2.3125	127	293.6875	146.84375
19	8	2.375	124	294.5	147.25
19.5	8	2.4375	120	292.5	146.25
20	8	2.5	117	292.5	146.25
20.5	8	2.5625	115	294.6875	147.34375
21	8	2.625	112	294	147
21.5	8	2.6875	109	292.9375	146.46875

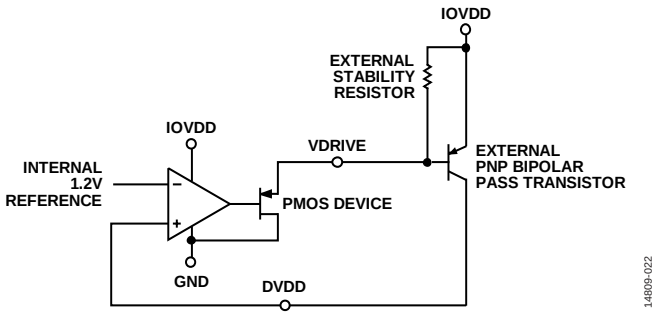


Figure 22. Simplified Block Diagram of Regulator Internal Structure, Including External Components

Power Reduction Modes

All sections of the IC have clock gating functionality that allows individual functional blocks to be disabled for power savings. Functional blocks that can optionally be powered down include the following:

- Clock Generator 1, Clock Generator 2, and Clock Generator 3
- S/PDIF receiver
- S/PDIF transmitter
- Serial data input and output ports
- Auxiliary ADC
- ASRCs (in two banks of eight channels each)
- PDM microphone inputs and decimation filters

Overview of Power Reduction Registers

An overview of the registers related to power reduction is shown in Table 24. For a more detailed description, see the Power Reduction Registers section.

Table 24. Power Reduction Registers

Address	Register	Description
0xF050	POWER_ENABLE0	Disables clock generators, serial ports, and ASRCs
0xF051	POWER_ENABLE1	Disables PDM microphone inputs, S/PDIF interfaces, and auxiliary ADCs

Hardware Reset

An active low hardware reset pin ($\overline{\text{RESET}}$) is available for externally triggering a reset of the device. When this pin is tied to ground, all functional blocks in the device are disabled, and the current consumption decreases dramatically. The amount of current drawn depends on the leakage current of the silicon, which depends greatly on the ambient temperature and the properties of the die. When the $\overline{\text{RESET}}$ pin is connected to IOVDD, all control registers are reset to their power-on default values. The state of the RAM is not guaranteed to be cleared after a reset; therefore, the memory must be manually cleared by the DSP program.

The default program generated by SigmaStudio includes code that automatically clears the memory. To ensure that no chatter exists on the $\overline{\text{RESET}}$ signal line, implement an external reset generation circuit in the system hardware design. Figure 23 shows an example of the ADM811 microprocessor supervisory circuit with a push-button connected, providing a method for manually generating a clean $\overline{\text{RESET}}$ signal. For reliability purposes on the application level, place a weak pull-down resistor on the $\overline{\text{RESET}}$ line to guarantee that the device is held in reset in the event that the reset supervisory circuitry fails.

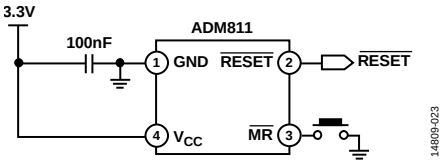


Figure 23. Example Manual Reset Generation Circuit

If the hardware reset function is not required in a system, pull the $\overline{\text{RESET}}$ pin high to the IOVDD supply using a weak pull-up resistor (in the range of several kΩ). The device is designed to boot properly even when the $\overline{\text{RESET}}$ pin is permanently pulled high.

TEMPERATURE SENSOR DIODE

The chip includes an on-board temperature sensor diode with an approximate range of 0°C to 120°C. The temperature sensor function is enabled by the two sides of a diode connected to the THD_P and THD_M pins. Value processing (calculating the actual temperature based on the current through the diode) is handled off chip by an external controller IC. The temperature value is not stored in an internal register; it is available only in the external controller IC. The temperature sensor requires an external IC to operate properly. See the Engineer-to-Engineer Note EE-346 for more information and instructions for using the temperature sensor diode.

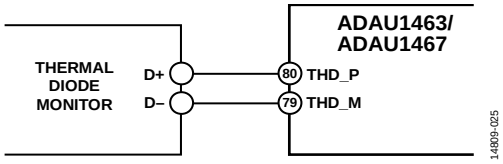


Figure 24. Example External Temperature Sensor Circuit

SLAVE CONTROL PORTS

A total of four control ports are available: two slave ports and two master ports. The slave I²C port and slave SPI port allow an external master device to modify the contents of the memory and registers. The master I²C port and master SPI port allow the device to self boot and to send control messages to slave devices on the same bus.

Table 36. Serial Output Pin Mapping from SigmaStudio Channels[1]

Output Channel in SigmaStudio	Serial Output Pin	Position in I²S Stream (2-Channel)	Position in TDM4 Stream	Position in TDM8 Stream	Position in TDM16 Stream
0	SDATA_OUT0	Left	0	0	0
1	SDATA_OUT0	Right	1	1	1
2	SDATA_OUT0	Not applicable	2	2	2
3	SDATA_OUT0	Not applicable	3	3	3
4	SDATA_OUT0	First SDATAIOx left	First SDATAIOx	4	4
5	SDATA_OUT0	First SDATAIOx right	First SDATAIOx	5	5
6	SDATA_OUT0	Not applicable	First SDATAIOx	6	6
7	SDATA_OUT0	Not applicable	First SDATAIOx	7	7
8	SDATA_OUT0	Second SDATAIOx left	Second SDATAIOx	First SDATAIOx	8
9	SDATA_OUT0	Second SDATAIOx right	Second SDATAIOx	First SDATAIOx	9
10	SDATA_OUT0	Not applicable	Second SDATAIOx	First SDATAIOx	10
11	SDATA_OUT0	Not applicable	Second SDATAIOx	First SDATAIOx	11
12	SDATA_OUT0	Third SDATAIOx left	Third SDATAIOx	First SDATAIOx	12
13	SDATA_OUT0	Third SDATAIOx right	Third SDATAIOx	First SDATAIOx	13
14	SDATA_OUT0	Not applicable	Third SDATAIOx	First SDATAIOx	14
15	SDATA_OUT0	Not applicable	Third SDATAIOx	First SDATAIOx	15
16	SDATA_OUT1	Left	0	0	0
17	SDATA_OUT1	Right	1	1	1
18	SDATA_OUT1	Not applicable	2	2	2
19	SDATA_OUT1	Not applicable	3	3	3
20	SDATA_OUT1	First SDATAIOx left	First SDATAIOx	4	4
21	SDATA_OUT1	First SDATAIOx right	First SDATAIOx	5	5
22	SDATA_OUT1	Not applicable	First SDATAIOx	6	6
23	SDATA_OUT1	Not applicable	First SDATAIOx	7	7
24	SDATA_OUT1	Second SDATAIOx left	Second SDATAIOx	First SDATAIOx	8
25	SDATA_OUT1	Second SDATAIOx right	Second SDATAIOx	First SDATAIOx	9
26	SDATA_OUT1	Not applicable	Second SDATAIOx	First SDATAIOx	10
27	SDATA_OUT1	Not applicable	Second SDATAIOx	First SDATAIOx	11
28	SDATA_OUT1	Third SDATAIOx left	Third SDATAIOx	First SDATAIOx	12
29	SDATA_OUT1	Third SDATAIOx right	Third SDATAIOx	First SDATAIOx	13
30	SDATA_OUT1	Not applicable	Third SDATAIOx	First SDATAIOx	14
31	SDATA_OUT1	Not applicable	Third SDATAIOx	First SDATAIOx	15
32	SDATA_OUT2	Left	0	0	Not applicable
33	SDATA_OUT2	Right	1	1	Not applicable
34	SDATA_OUT2	Not applicable	2	2	Not applicable
35	SDATA_OUT2	Not applicable	3	3	Not applicable
36	SDATA_OUT2	SDATAIOx left	First SDATAIOx	4	Not applicable
37	SDATA_OUT2	SDATAIOx right	First SDATAIOx	5	Not applicable
38	SDATA_OUT2	Not applicable	First SDATAIOx	6	Not applicable
39	SDATA_OUT2	Not applicable	First SDATAIOx	7	Not applicable
40	SDATA_OUT3	Left	0	0	Not applicable
41	SDATA_OUT3	Right	1	1	Not applicable
42	SDATA_OUT3	Not applicable	2	2	Not applicable
43	SDATA_OUT3	Not applicable	3	3	Not applicable
44	SDATA_OUT3	SDATAIOx left	First SDATAIOx	4	Not applicable
45	SDATA_OUT3	SDATAIOx right	First SDATAIOx	5	Not applicable
46	SDATA_OUT3	Not applicable	First SDATAIOx	6	Not applicable
47	SDATA_OUT3	Not applicable	First SDATAIOx	7	Not applicable

[1] Any of the eight SDATAIOx pins can be assigned to any output.

Figure 53 shows timing diagrams for possible serial port configurations in 4-channel mode, with 16 bit clock cycles per channel, for a total of 64 bit clock cycles per frame (refer to the SERIAL_BYTE_x_0 registers, Bits[2:0] (TDM_MODE) = 0b100). Different bit clock polarities are shown (refer to the SERIAL_BYTE_x_0 registers, Bit 7 (BCLK_POL)). The audio word length is fixed at 16 bits (refer to the SERIAL_BYTE_x_0 registers, Bits[6:5] (WORD_LEN) = 0b01), and there are four possible configurations for MSB position (SERIAL_BYTE_x_0, Bits[4:3] (DATA_FMT)), all of which are shown in Figure 53.

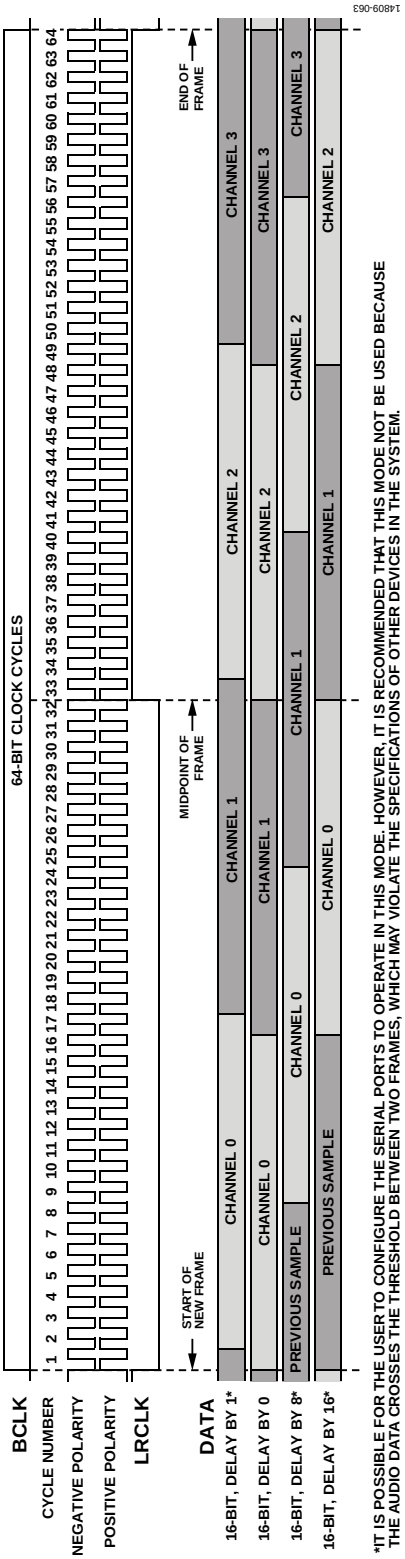


Figure 53. Serial Audio Data Formats; Four Channels, 16 Bits per Channel

Address	Register	Description
0xF3A5	FTDM_OUT37	FTDM mapping for the serial outputs (Port 3, Channel 1, Bits[23:16])
0xF3A6	FTDM_OUT38	FTDM mapping for the serial outputs (Port 3, Channel 1, Bits[15:8])
0xF3A7	FTDM_OUT39	FTDM mapping for the serial outputs (Port 3, Channel 1, Bits[7:0])
0xF3A8	FTDM_OUT40	FTDM mapping for the serial outputs (Port 3, Channel 2, Bits[31:24])
0xF3A9	FTDM_OUT41	FTDM mapping for the serial outputs (Port 3, Channel 2, Bits[23:16])
0xF3AA	FTDM_OUT42	FTDM mapping for the serial outputs (Port 3, Channel 2, Bits[15:8])
0xF3AB	FTDM_OUT43	FTDM mapping for the serial outputs (Port 3, Channel 2, Bits[7:0])
0xF3AC	FTDM_OUT44	FTDM mapping for the serial outputs (Port 3, Channel 3, Bits[31:24])
0xF3AD	FTDM_OUT45	FTDM mapping for the serial outputs (Port 3, Channel 3, Bits[23:16])
0xF3AE	FTDM_OUT46	FTDM mapping for the serial outputs (Port 3, Channel 3, Bits[15:8])
0xF3AF	FTDM_OUT47	FTDM mapping for the serial outputs (Port 3, Channel 3, Bits[7:0])
0xF3B0	FTDM_OUT48	FTDM mapping for the serial outputs (Port 3, Channel 4, Bits[31:24])
0xF3B1	FTDM_OUT49	FTDM mapping for the serial outputs (Port 3, Channel 4, Bits[23:16])
0xF3B2	FTDM_OUT50	FTDM mapping for the serial outputs (Port 3, Channel 4, Bits[15:8])
0xF3B3	FTDM_OUT51	FTDM mapping for the serial outputs (Port 3, Channel 4, Bits[7:0])
0xF3B4	FTDM_OUT52	FTDM mapping for the serial outputs (Port 3, Channel 5, Bits[31:24])
0xF3B5	FTDM_OUT53	FTDM mapping for the serial outputs (Port 3, Channel 5, Bits[23:16])
0xF3B6	FTDM_OUT54	FTDM mapping for the serial outputs (Port 3, Channel 5, Bits[15:8])
0xF3B7	FTDM_OUT55	FTDM mapping for the serial outputs (Port 3, Channel 5, Bits[7:0])
0xF3B8	FTDM_OUT56	FTDM mapping for the serial outputs (Port 3, Channel 6, Bits[31:24])
0xF3B9	FTDM_OUT57	FTDM mapping for the serial outputs (Port 3, Channel 6, Bits[23:16])
0xF3BA	FTDM_OUT58	FTDM mapping for the serial outputs (Port 3, Channel 6, Bits[15:8])
0xF3BB	FTDM_OUT59	FTDM mapping for the serial outputs (Port 3, Channel 6, Bits[7:0])
0xF3BC	FTDM_OUT60	FTDM mapping for the serial outputs (Port 3, Channel 7, Bits[31:24])
0xF3BD	FTDM_OUT61	FTDM mapping for the serial outputs (Port 3, Channel 7, Bits[23:16])
0xF3BE	FTDM_OUT62	FTDM mapping for the serial outputs (Port 3, Channel 7, Bits[15:8])
0xF3BF	FTDM_OUT63	FTDM mapping for the serial outputs (Port 3, Channel 7, Bits[7:0])

S/PDIF INTERFACE

To simplify interfacing at the system level, wire the on-chip S/PDIF receiver and transmitter data ports directly to other S/PDIF-compatible equipment. The S/PDIF receiver consists of two audio channels input on one hardware pin (SPDIFIN). The clock signal is embedded in the data using biphase mark code. The S/PDIF transmitter consists of two audio channels output on one hardware pin (SPDIFOUT). The clock signal is embedded in the data using biphase mark code. The S/PDIF input and output word lengths can be independently set to 16, 20, or 24 bits.

The S/PDIF interface meets the S/PDIF consumer performance specification. It does not meet the AES3 professional specification.

S/PDIF Receiver

The S/PDIF input port is designed to accept both transistor to transistor logic (TTL) and bipolar signals, provided there is an ac coupling capacitor on the input pin of the chip. Because the S/PDIF input data is most likely asynchronous to the DSP core, it must be routed through an ASRC.

The S/PDIF receiver works over a wide range of sampling frequencies between 18 kHz and 192 kHz. Note that the RX_MCLKSPEED bit must be set in the SPDIF_RX_MCLKSPEED register and the TX_MCLKSPEED bit must be set in the SPDIF_TX_MCLKSPEED register for receive and transmit rates greater than 96 kHz, respectively.

The S/PDIF receiver input is a comparator that is centered at IOVDD/2 and requires an input signal level of at least 200 mV p-p to operate properly.

In addition to audio data, S/PDIF streams contain user data, channel status, validity bit, virtual LRCLK, and block start information. The receiver decodes audio data and sends it to the corresponding registers in the control register map, where the information can be read over the I^2C or SPI slave port.

For improved jitter performance, the S/PDIF clock recovery implementation is completely digital. The S/PDIF ports are designed to meet the following Audio Engineering Society (AES) and European Broadcasting Union (EBU) specifications: a jitter of 0.25 UI p-p at 8 kHz and above, a jitter of 10 UI p-p below 200 Hz, and a minimum signal voltage of 200 mV.

S/PDIF Transmitter

The S/PDIF transmitter outputs two channels of audio data directly from the DSP core at the core rate. The extra nonaudio data bits on the transmitted signal can be copied directly from the S/PDIF receiver or programmed manually, using the corresponding registers in the control register map.

Auxiliary Output Mode

The received data on the S/PDIF receiver can be converted to a TDM8 stream, bypass the SigmaDSP core, and be output directly on a serial data output pin. This mode of operation is called auxiliary output mode. Configure this mode using Register 0xF608 (SPDIF_AUX_EN). The TDM8 output from the S/PDIF receiver regroups the recovered data in a TDM like format, as shown in Table 46.

The S/PDIF receiver, when operating in auxiliary output mode, also recovers the embedded BCLK_OUTx and LRCLK_OUTx signals in the S/PDIF stream and outputs them on the corresponding BCLK_OUTx and LRCLK_OUTx pins in master mode when Register 0xF608 (SPDIF_AUX_EN), Bits[3:0] (TDMOUT) are configured to enable auxiliary output mode. The selected BCLK_OUTx signal has a frequency of 256× the recovered sample rate, and the LRCLK_OUTx signal is a 50% duty cycle square wave that has the same frequency as the audio sample rate (see Table 144).

Table 46. S/PDIF Auxiliary Output Mode, TDM8 Data Format

TDM8 Channel	Description of Data Format
0	8 zero bits followed by 24 audio bits, recovered from the left audio channel of the S/PDIF stream
1	28 zero bits followed by the left parity bit, left validity bit, left user data, and left channel status
2	30 zero bits followed by the compression type bit (COMPR_TYPE) (0b0 = AC3, 0b1 = DTS) and the audio type bit (AUDIO_TYPE) (0 = PCM, 1 = compressed)
3	No data
4	8 zero bits followed by 24 audio bits, recovered from the right audio channel of the S/PDIF stream
5	28 zero bits followed by the right parity bit, right validity bit, right user data, and right channel status
6	No data
7	31 zero bits followed by the block start signal

S/PDIF Receiver Inputs to DSP Core

The S/PDIF receiver input must pass through an ASRC to guarantee that it is synchronous to the DSP core. The two channels from the S/PDIF receiver can be selected as the audio source to ASRCs in the routing matrix. When the source is the S/PDIF receiver, the serial input channel that is specified is ignored.

Table 47. S/PDIF Input Mapping to SigmaStudio Channels

Channel in S/PDIF Receiver Data Stream	S/PDIF Input Channels in SigmaStudio
Left	0
Right	1

S/PDIF Audio Outputs from DSP Core to S/PDIF Transmitter

The output signal of the S/PDIF transmitter can come from the DSP core or directly from the S/PDIF receiver. The selection is controlled by Register 0xF1C0 (SPDIFTX_INPUT). When the signal comes from the DSP core, use the S/PDIF output cells in SigmaStudio.

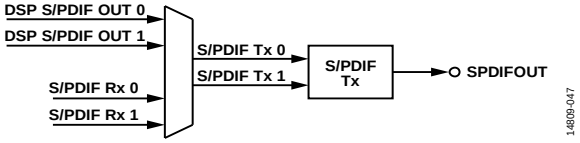


Figure 69. S/PDIF Transmitter Source Selection

Table 48. S/PDIF Output Mapping from SigmaStudio Channels

Channel in S/PDIF Transmitter Data Stream	S/PDIF Output Channel in SigmaStudio
Left	0
Right	1

S/PDIF Interface Registers

An overview of the registers related to the S/PDIF interface is shown in Table 49. For a more detailed description, refer to the S/PDIF Interface Registers section.

Hardware Accelerators

The core includes accelerators like division, square root, barrel shifters, Base 2 logarithm, Base 2 exponential, slew, and a pseudorandom number generator. These hardware accelerators reduce the number of instructions required for complex audio processing algorithms.

The division accelerator enables efficient processing for audio algorithms like compression and limiting. The square root accelerator enables efficient processing for audio algorithms such as loudness, rms envelopes, and filter coefficient calculations. The logarithm and exponent accelerators enable efficient processing for audio algorithms involving decibel conversion. The slew accelerators provide click free updates of parameters that must change slowly over time, allowing audio processing algorithms such as mixers, crossfaders, dynamic filters, and dynamic volume controls. The pseudorandom number generator can efficiently produce white noise, pink noise, and dither.

Programming the SigmaDSP Core

The SigmaDSP is programmable via the SigmaStudio graphical development tools.

When the SigmaDSP core is running a program and the user needs to reprogram the program and data memories during operation of the device, the core must be stopped while the memory is being updated to avoid undesired noises on the DSP outputs.

The following sequence of steps is appropriate for programming the memories at boot time, or reprogramming the memories during operation:

1. Enable soft reset (Register 0xF890 (SOFT_RESET), Bit 0 (SOFT_RESET) = 0b0), then disable soft reset (Register 0xF890 (SOFT_RESET), Bit 0 (SOFT_RESET) = 0b1).
2. If the DSP is in the process of executing a program, wait for the current sample or block to finish processing. For programs with no block processing elements in the signal flow, use the length of one sample. For example, at a sample rate of 48 kHz, one sample is 1/48000 sec, or 20.83 µs. For programs with block processing elements in the signal flow, use the length of one block. For example, at a sample rate of 48 kHz, with a block size of 256 samples, one block is 256/48,000 sec, or 53.3 ms.
3. After waiting the appropriate amount of time, as defined in Step 2, download the new program and data memory contents to the corresponding memory locations using the I^2C/SPI slave control port.
4. Start the DSP core (Register 0xF402 (START_CORE), Bit 0 (START_CORE) = 0b1).
5. Wait at least two audio samples for the DSP initialization to execute. For example, at a sample rate of 48 kHz, two samples are equal to 2/48,000 sec, or 41.66 µs.

Reliability Features

Several reliability features are controlled by a panic manager subsystem that monitors the state of the SigmaDSP core and memories and generates alerts if error conditions are encountered. The panic manager indicates error conditions to the user via register flags and GPIO outputs. The origin of the error can be traced to different functional blocks such as the watchdog, memory, stack, software program, and core op codes.

Although designed mostly as an aid for software development, the panic manager is also useful in monitoring the state of the memories over long periods of time, such as in applications where the system operates unattended for an extended period, and resets are infrequent. The memories in the device have a built in self test feature that runs automatically while the device is in operation. If a memory corruption is detected, the appropriate flag is signaled in the panic manager. The program running in the DSP core can monitor the state of the panic manager and can mute the audio outputs if an error is encountered, and external devices, such as microcontrollers, can poll the panic manager registers or monitor the multipurpose pins to perform some preprogrammed action, if necessary.

DSP Core and Reliability Registers

An overview of the registers related to the DSP core is shown in Table 54. For a more detailed description, see the DSP Core Control Registers section and Debug and Reliability Registers section.

SOFTWARE FEATURES

Software Safeload

To prevent making the filter unstable during coefficient transitions, the SigmaStudio compiler implements a software safeload mechanism that is enabled by default. The safeload mechanism is also helpful for reducing pops and clicks during parameter updates. SigmaStudio automatically sets up the necessary code and parameters for all new projects. The safeload code, together with other initialization code, fills the beginning section of program RAM. Several data memory locations are reserved by the compiler for use with the software safeload feature. The exact parameter addresses are not fixed; therefore, the addresses must be obtained by reading the log file generated by the compiler. In most cases, the addresses for software safeload parameters match the defaults shown in Table 55.

Table 55. Default Software Safeload Memory Addresses

Address (Hex)	Parameter	Function
0x6000	data_SafeLoad[0]	Safeload Data Slot 0
0x6001	data_SafeLoad[1]	Safeload Data Slot 1
0x6002	data_SafeLoad[2]	Safeload Data Slot 2
0x6003	data_SafeLoad[3]	Safeload Data Slot 3
0x6004	data_SafeLoad[4]	Safeload Data Slot 4
0x6005	address_SafeLoad	Target address for safeload transfer
0x6006	num_SafeLoad_Lower	Number of words to write/safeload trigger if on Page 1 lower memory
0x6007	num_SafeLoad_Upper	Number of words to write/safeload trigger if on Page 2 upper memory

The first five addresses in Table 55 are the five data_SafeLoad[x] parameters, which are slots for storing the data to be transferred into another target memory location. The safeload parameter space contains five data slots, by default, because most standard signal processing algorithms have five parameters or fewer.

The address_SafeLoad parameter is the target address in parameter RAM. This target address designates the first address to be written in the safeload transfer. If more than one word is written, the address increments automatically for each data-word.

The num_SafeLoad_Lower and num_SafeLoad_Upper parameters designate the number of words to be written. For a biquad filter algorithm, the number of words to be written is five because there are five coefficients in a biquad IIR filter. For a simple, single-gain algorithm, the number of words to be written is one. This parameter also serves as the trigger; when it is written, a safeload write is triggered on the next frame.

Because the slave port cannot access all of the core data memory from a single 16-bit address space, the safeload subroutine needs to know whether to write to the lower (Page 1) or upper (Page 2) section of memory. If the first parameter is to be place on Page 1 (lower memory), write the number of parameters to be automatically written (1 to 5) to num_SafeLoad_Lower and write 0 to num_SafeLoad_Upper. Conversely, if the first parameter is to be placed on Page 2 (upper memory), write 0 to num_SafeLoad_Lower and write the number of parameters to be automatically written (1 to 5) to num_SafeLoad_Upper. One of these values passed must always be a number between one and five inclusive, and the other value must be zero. The second write triggers the safeload operation.

The safeload mechanism is software based and executes once per audio frame. Therefore, system designers must take care when designing the communication protocol. A delay that is equal to or greater than the sampling period (the inverse of the sampling frequency) is required between each safeload write. At a sample rate of 48 kHz, the delay is equal to ≥20.83 µs. Not observing this delay corrupts the downloaded data.

Because the compiler has control over the addresses used for software safeload, the addresses assigned to each parameter may differ from the default values in Table 55. The compiler generates a file named compiler_output.log in the project folder where the SigmaStudio project is stored on the hard drive. In this file, the addresses assigned to the software safeload parameters can be confirmed.

Figure 80 shows an example of the software safeload parameter definitions in an excerpt from the compiler_output.log file.

The following steps are necessary for executing a software safeload:

1. Confirm that no safeload operation has been executed in the span of the last audio sample.
2. Write the desired data to the data_SafeLoad[x], Bit x parameters, starting at data_SafeLoad[x], Bit 0, and incrementing, as needed, up to a maximum of five parameters.
3. Write the desired starting target address to the address_SafeLoad parameter.
4. Write the number of words to be transferred to the num_SafeLoad_Lower and num_SafeLoad_Upper parameters. The minimum write length is one word, and the maximum write length is five words.
5. Wait one audio frame for the safeload operation to complete.

CONTROL REGISTERS

All control registers store 16 bits (two bytes) of data. The register map is defined in Table 60.

Table 60. Control Register Summary

Reg	Name	Bits	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset	RW
0xF000	PLL_CTRL0	[15:8]	RESERVED								0x0060	RW
		[7:0]	RESERVED	PLL_FBDIVIDER								
0xF001	PLL_CTRL1	[15:8]	RESERVED								0x0000	RW
		[7:0]	RESERVED						PLL_DIV			
0xF002	PLL_CLK_SRC	[15:8]	RESERVED								0x0000	RW
		[7:0]	RESERVED							CLKSRC		
0xF003	PLL_ENABLE	[15:8]	RESERVED								0x0000	RW
		[7:0]	RESERVED							PLL_ENABLE		
0xF004	PLL_LOCK	[15:8]	RESERVED								0x0000	R
		[7:0]	RESERVED							PLL_LOCK		
0xF005	MCLK_OUT	[15:8]	RESERVED								0x0000	R
		[7:0]	RESERVED					CLKOUT_RATE		CLKOUT_ENABLE		
0xF006	PLL_WATCHDOG	[15:8]	RESERVED								0x0001	R
		[7:0]	RESERVED							PLL_WATCHDOG		
0xF020	CLK_GEN1_M	[15:8]	RESERVED							CLOCKGEN1_M[8]	0x0006	RW
		[7:0]	CLOCKGEN1_M[7:0]									
0xF021	CLK_GEN1_N	[15:8]	RESERVED							CLOCKGEN1_N[8]	0x0001	RW
		[7:0]	CLOCKGEN1_N[7:0]									
0xF022	CLK_GEN2_M	[15:8]	RESERVED							CLOCKGEN2_M[8]	0x0009	RW
		[7:0]	CLOCKGEN2_M[7:0]									
0xF023	CLK_GEN2_N	[15:8]	RESERVED							CLOCKGEN2_N[8]	0x0001	RW
		[7:0]	CLOCKGEN2_N[7:0]									
0xF024	CLK_GEN3_M	[15:8]	CLOCKGEN3_M[15:8]								0x0000	RW
		[7:0]	CLOCKGEN3_M[7:0]									
0xF025	CLK_GEN3_N	[15:8]	CLOCKGEN3_N[15:8]								0x0000	RW
		[7:0]	CLOCKGEN3_N[7:0]									
0xF026	CLK_GEN3_SRC	[15:8]	RESERVED								0x000E	RW
		[7:0]	RESERVED			CLK_GEN3_SRC	FREF_PIN					
0xF027	CLK_GEN3_LOCK	[15:8]	RESERVED								0x0000	R
		[7:0]	RESERVED							GEN3_LOCK		
0xF050	POWER_ENABLE0	[15:8]	RESERVED			CLK_GEN3_PWR	CLK_GEN2_PWR	CLK_GEN1_PWR	ASRCBANK1_PWR	ASRCBANK0_PWR	0x0000	RW
		[7:0]	SOUT3_PWR	SOUT2_PWR	SOUT1_PWR	SOUT0_PWR	SIN3_PWR	SIN2_PWR	SIN1_PWR	SIN0_PWR		
0xF051	POWER_ENABLE1	[15:8]	RESERVED								0x0000	RW
		[7:0]	RESERVED			PDM1_PWR	PDM0_PWR	TX_PWR	RX_PWR	ADC_PWR		
0xF100 to 0xF107	ASRC_INPUTx	[15:8]	RESERVED								0x0000	RW
		[7:0]	ASRC_SIN_CHANNEL					ASRC_SOURCE				
0xF140 to 0xF147	ASRC_OUT_RATEx	[15:8]	RESERVED								0x0000	RW
		[7:0]	RESERVED				ASRC_RATE					
0xF180 to 0xF197	SOUT_SOURCEx	[15:8]	RESERVED								0x0000	RW
		[7:0]	RESERVED		SOUT_ASRC_SELECT			SOUT_SOURCE				
0xF1C0	SPDIFTX_INPUT	[15:8]	RESERVED								0x0000	RW
		[7:0]	RESERVED						SPDIFTX_SOURCE			
0xF200 to 0xF21C	SERIAL_BYTE_x_0	[15:8]	LRCLK_SRC			BCLK_SRC			LRCLK_MODE	LRCLK_POL	0x0000	RW
		[7:0]	BCLK_POL	WORD_LEN		DATA_FMT		TDM_MODE				

Reg	Name	Bits	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset	RW
0xF630 to 0xF63B	SPDIF_RX_UD_LEFT_x	[15:8]	SPDIF_RX_UD_LEFT[15:8]								0x0000	R
		[7:0]	SPDIF_RX_UD_LEFT[7:0]									
0xF640 to 0xF64B	SPDIF_RX_UD_RIGHT_x	[15:8]	SPDIF_RX_UD_RIGHT[15:8]								0x0000	R
		[7:0]	SPDIF_RX_UD_RIGHT[7:0]									
0xF650 to 0xF65B	SPDIF_RX_VB_LEFT_x	[15:8]	SPDIF_RX_VB_LEFT[15:8]								0x0000	R
		[7:0]	SPDIF_RX_VB_LEFT[7:0]									
0xF660 to 0xF66B	SPDIF_RX_VB_RIGHT_x	[15:8]	SPDIF_RX_VB_RIGHT[15:8]								0x0000	R
		[7:0]	SPDIF_RX_VB_RIGHT[7:0]									
0xF670 to 0xF67B	SPDIF_RX_PB_LEFT_x	[15:8]	SPDIF_RX_PB_LEFT[15:8]								0x0000	R
		[7:0]	SPDIF_RX_PB_LEFT[7:0]									
0xF680 to 0xF68B	SPDIF_RX_PB_RIGHT_x	[15:8]	SPDIF_RX_PB_RIGHT[15:8]								0x0000	R
		[7:0]	SPDIF_RX_PB_RIGHT[7:0]									
0xF690	SPDIF_TX_EN	[15:8]	RESERVED								0x0000	RW
		[7:0]	RESERVED							TXEN		
0xF691	SPDIF_TX_CTRL	[15:8]	RESERVED								0x0000	RW
		[7:0]	RESERVED						TX_LENGTHCTRL			
0xF69F	SPDIF_TX_AUXBIT_SOURCE	[15:8]	RESERVED								0x0000	RW
		[7:0]	RESERVED							TX_AUXBITS_SOURCE		
0xF6A0 to 0xF6AB	SPDIF_TX_CS_LEFT_x	[15:8]	SPDIF_TX_CS_LEFT[15:8]								0x0000	RW
		[7:0]	SPDIF_TX_CS_LEFT[7:0]									
0xF6B0 to 0xF6BB	SPDIF_TX_CS_RIGHT_x	[15:8]	SPDIF_TX_CS_RIGHT[15:8]								0x0000	RW
		[7:0]	SPDIF_TX_CS_RIGHT[7:0]									
0xF6C0 to 0xF6CB	SPDIF_TX_UD_LEFT_x	[15:8]	SPDIF_TX_UD_LEFT[15:8]								0x0000	RW
		[7:0]	SPDIF_TX_UD_LEFT[7:0]									
0xF6D0 to 0xF6DB	SPDIF_TX_UD_RIGHT_x	[15:8]	SPDIF_TX_UD_RIGHT[15:8]								0x0000	RW
		[7:0]	SPDIF_TX_UD_RIGHT[7:0]									
0xF6E0 to 0xF6EB	SPDIF_TX_VB_LEFT_x	[15:8]	SPDIF_TX_VB_LEFT[15:8]								0x0000	RW
		[7:0]	SPDIF_TX_VB_LEFT[7:0]									
0xF6F0 to 0xF6FB	SPDIF_TX_VB_RIGHT_x	[15:8]	SPDIF_TX_VB_RIGHT[15:8]								0x0000	RW
		[7:0]	SPDIF_TX_VB_RIGHT[7:0]									
0xF700 to 0xF70B	SPDIF_TX_PB_LEFT_x	[15:8]	SPDIF_TX_PB_LEFT[15:8]								0x0000	RW
		[7:0]	SPDIF_TX_PB_LEFT[7:0]									
0xF710 to 0xF71B	SPDIF_TX_PB_RIGHT_x	[15:8]	SPDIF_TX_PB_RIGHT[15:8]								0x0000	RW
		[7:0]	SPDIF_TX_PB_RIGHT[7:0]									
0xF780 to 0xF783	BCLK_INx_PIN	[15:8]	RESERVED								0x0018	RW
		[7:0]	RESERVED			BCLK_IN_PULL		BCLK_IN_SLEW		BCLK_IN_DRIVE		
0xF784 to 0xF787	BCLK_OUTx_PIN	[15:8]	RESERVED								0x0018	RW
		[7:0]	RESERVED			BCLK_OUT_PULL		BCLK_OUT_SLEW		BCLK_OUT_DRIVE		
0xF788 to 0xF78B	LRCLK_INx_PIN	[15:8]	RESERVED								0x0018	RW
		[7:0]	RESERVED			LRCLK_IN_PULL		LRCLK_IN_SLEW		LRCLK_IN_DRIVE		
0xF78C to 0xF78F	LRCLK_OUTx_PIN	[15:8]	RESERVED								0x0018	RW
		[7:0]	RESERVED			LRCLK_OUT_PULL		LRCLK_OUT_SLEW		LRCLK_OUT_DRIVE		
0xF790 to 0xF793	SDATA_INx_PIN	[15:8]	RESERVED								0x0018	RW
		[7:0]	RESERVED			SDATA_IN_PULL		SDATA_IN_SLEW		SDATA_IN_DRIVE		
0xF794 to 0xF797	SDATA_OUTx_PIN	[15:8]	RESERVED								0x0008	RW
		[7:0]	RESERVED			SDATA_OUT_PULL		SDATA_OUT_SLEW		SDATA_OUT_DRIVE		

Bits	Bit Name	Settings	Description	Reset	Access
		00	Port 0.		
		01	Port 1.		
		10	Port 2.		
		11	Port 3.		
[1:0]	CHAN		For Serial Port 0 in 2-channel mode:	0x00	RW
		00	Channel 7 to Channel 4.		
		01	Channel 11 to Channel 8.		
		10	Channel 15 to Channel 12.		
			For Serial Port 0 in TDM mode:	0x00	
		00	Channel 7 to Channel 4 (TDM-4).		
		01	Channel 11 to Channel 8 (TDM-4).		
		10	Channel 15 to Channel 12 (TDM-4).		
		11	Channel 15 to Channel 8 (TDM-8).		
			For Serial Port 1 in 2-channel mode:	0x00	
		00	Channel 7 and Channel 6.		
		01	Channel 11 and Channel 10.		
		10	Channel 15 and Channel 14.		
			For Serial Port 1 in TDM mode:	0x00	
		00	Channel 23 to Channel 20 (TDM-4).		
		01	Channel 27 to Channel 24 (TDM-4).		
		10	Channel 31 to Channel 28 (TDM-4).		
		11	Channel 31 to Channel 27 (TDM-8).		
			For Serial Port 2 in 2-channel mode:	0x00	
		00	Channel 35 and Channel 34.		
			For Serial Port 2 in TDM4 mode:	0x00	
		00	Channel 39 to Channel 36.		
			For Serial Port 3 in 2-channel mode:	0x00	
		00	Channel 47 to Channel 46.		
			For Serial Port 3 in TDM4 mode:		
		00	Channel 47 to Channel 44.	0x00	

DEBUG AND RELIABILITY REGISTERS

Clear the Panic Manager Register

Address: 0xF421, Reset: 0x0000, Name: PANIC_CLEAR

When Register 0xF427 (PANIC_FLAG) signals that an error has occurred, use Register 0xF421 (PANIC_CLEAR) to reset it. Toggle Bit 0 (PANIC_CLEAR) of this register from 0b0 to 0b1 and then back to 0b0 again to clear the flag and reset the state of the panic manager.

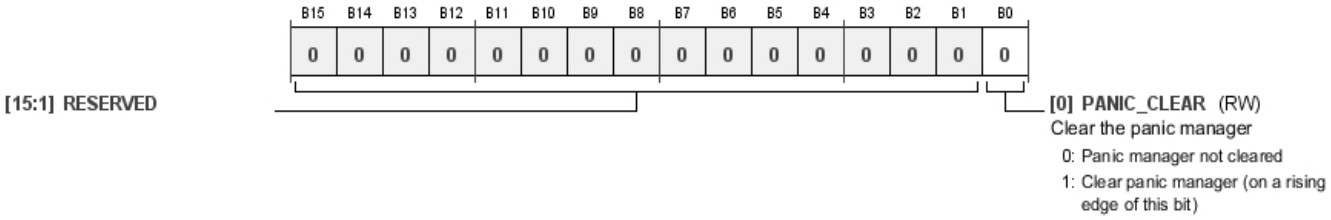


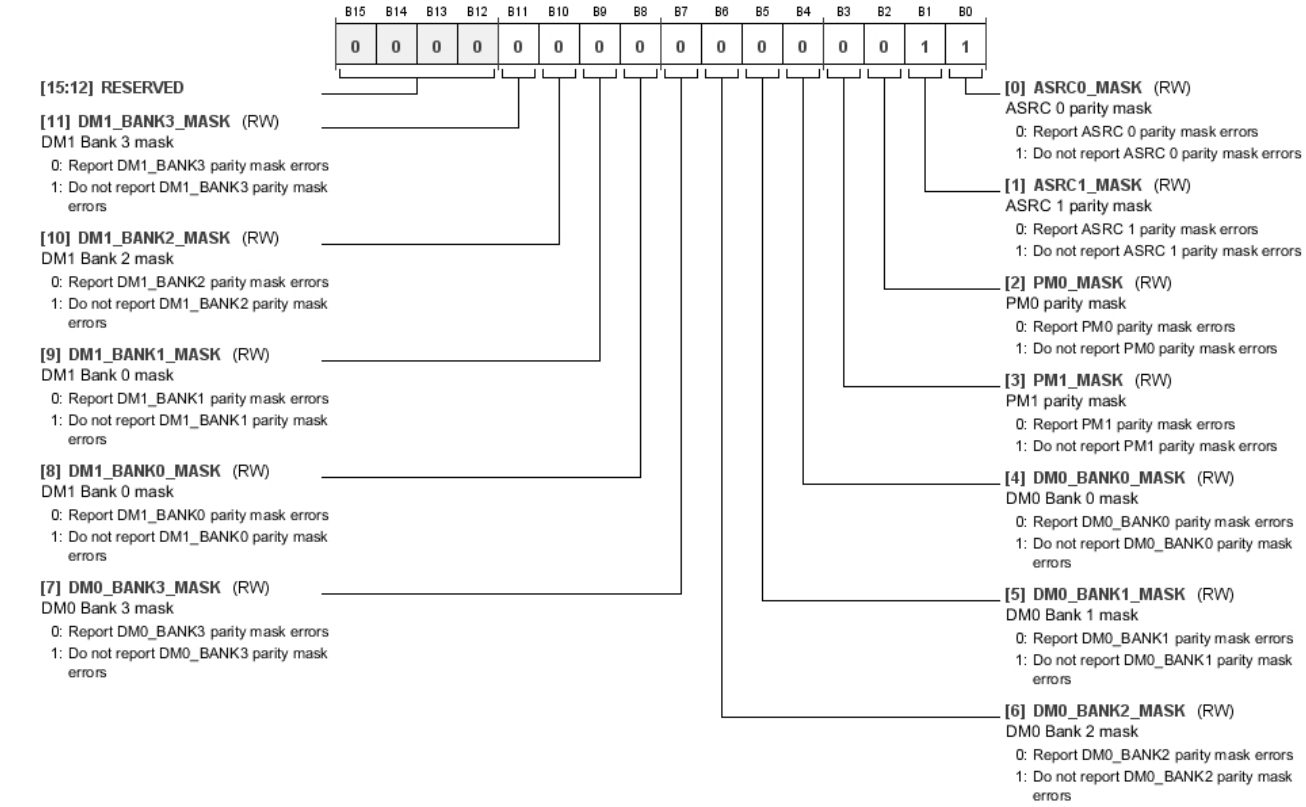
Table 95. Bit Descriptions for PANIC_CLEAR

Bits	Bit Name	Settings	Description	Reset	Access
[15:1]	RESERVED			0x0	RW
0	PANIC_CLEAR		Clear the panic manager. To reset the PANIC_FLAG register (Register 0xF427), toggle this bit on and then off again.	0x0	RW
		0	Panic manager is not cleared.		
		1	Clear panic manager (on a rising edge of this bit).		

Panic Parity Register

Address: 0xF422, Reset: 0x0003, Name: PANIC_PARITY_MASK

The panic manager checks and reports memory parity mask errors. Register 0xF422 (PANIC_PARITY_MASK) allows the user to configure which memories, if any, are subject to error reporting.



Panic Mask 0 Register

Address: 0xF423, Reset: 0x0000, Name: PANIC_SOFTWARE_MASK

The panic manager checks and reports software errors. Register 0xF423 (PANIC_SOFTWARE_MASK) allows the user to configure whether software errors are reported to the panic manager or ignored.

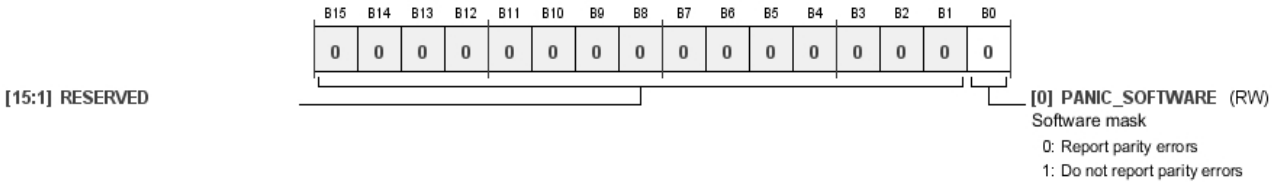


Table 97. Bit Descriptions for PANIC_SOFTWARE_MASK

Bits	Bit Name	Settings	Description	Reset	Access
[15:1]	RESERVED			0x0	RW
0	PANIC_SOFTWARE		Software mask.	0x0	RW
		0	Report parity errors.		
		1	Do not report parity errors.		

Digital PDM Microphone Control Register

Address: 0xF560 to 0xF561 (Increments of 0x1), Reset: 0x4000, Name: DMIC_CTRLx

These registers configure the digital PDM microphone interface. Two registers are used to control up to four PDM microphones: Register 0xF560 (DMIC_CTRL0) configures PDM Microphone Channel 0 and PDM Microphone Channel 1, and Register 0xF561 (DMIC_CTRL1) configures PDM Microphone Channel 2 and PDM Microphone Channel 3.

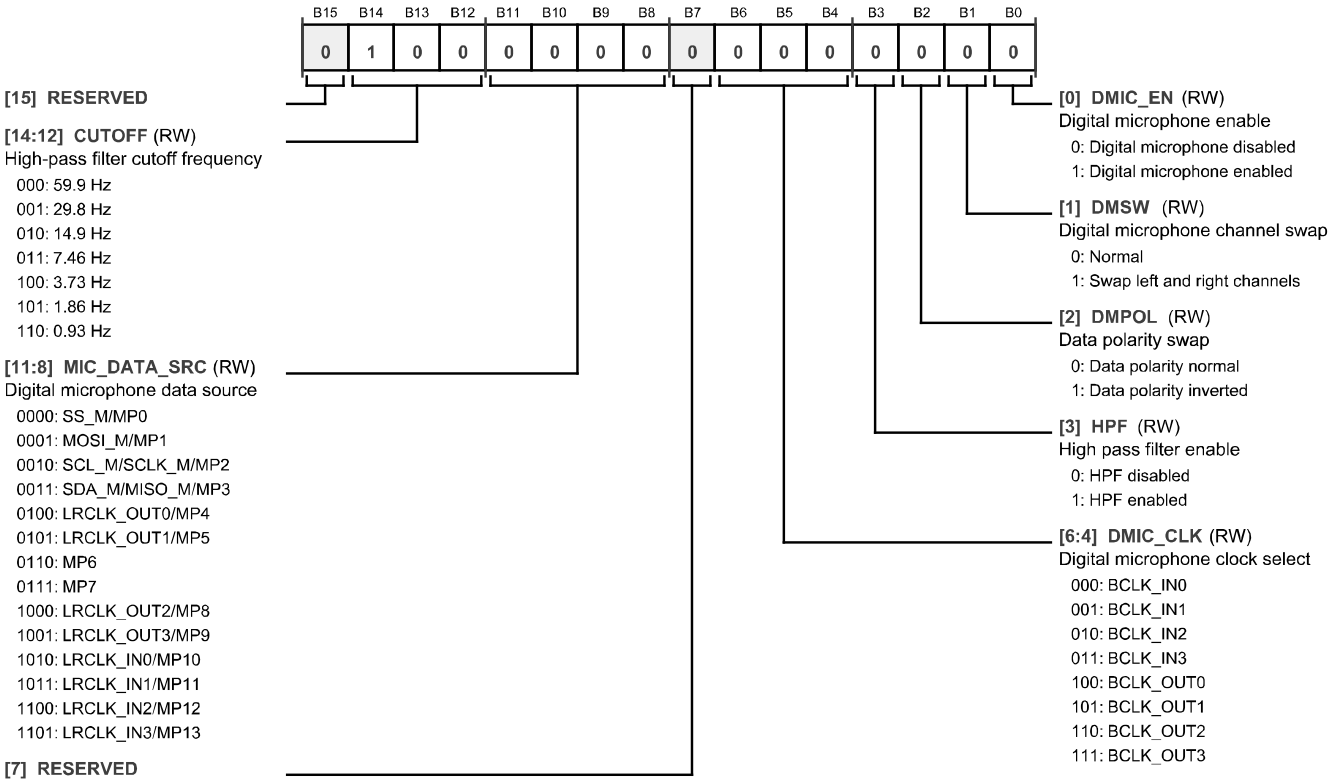


Table 128. Bit Descriptions for DMIC_CTRLx

Bits	Bit Name	Settings	Description	Reset	Access
15	RESERVED			0x0	RW
[14:12]	CUTOFF		High-pass filter cutoff frequency. These bits configure the cutoff frequency of an optional high-pass filter designed to remove dc components from the microphone data signal(s). To use these bits, Bit 3 (HPF), must be enabled.	0x4	RW
		000	59.9 Hz.		
		001	29.8 Hz.		
		010	14.9 Hz.		
		011	7.46 Hz.		
		100	3.73 Hz.		
		101	1.86 Hz.		
		110	0.93 Hz.		
[11:8]	MIC_DATA_SRC		Digital PDM microphone data source pin. These bits configure which hardware pin acts as a data input from the PDM microphone(s). Up to two microphones can be connected to a single pin.	0x0	RW
		0000	SS_M/MP0.		
		0001	MOSI_M/MP1.		
		0010	SCL_M/SCLK_M/MP2.		
		0011	SDA_M/MISO_M/MP3.		
		0100	LRCLK_OUT0/MP4.		
		0101	LRCLK_OUT1/MP5.		
		0110	MP6.		

S/PDIF Receiver Validity Bits (Left) Register

Address: 0xF650 to 0xF65B (Increments of 0x1), Reset: 0x0000, Name: SPDIF_RX_VB_LEFT_x

These 12 registers store the 192 validity bits decoded from the left channel of the S/PDIF input stream on the ADAU1463 and ADAU1467.

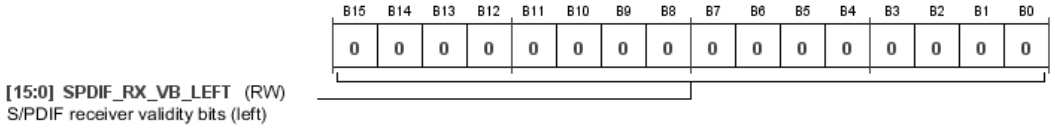


Table 150. Bit Descriptions for SPDIF_RX_VB_LEFT_x

Bits	Bit Name	Settings	Description	Reset	Access
[15:0]	SPDIF_RX_VB_LEFT		S/PDIF receiver validity bits (left).	0x0000	R

S/PDIF Receiver Validity Bits (Right) Register

Address: 0xF660 to 0xF66B (Increments of 0x1), Reset: 0x0000, Name: SPDIF_RX_VB_RIGHT_x

These 12 registers store the 192 validity bits decoded from the left channel of the S/PDIF input stream on the ADAU1463 and ADAU1467.

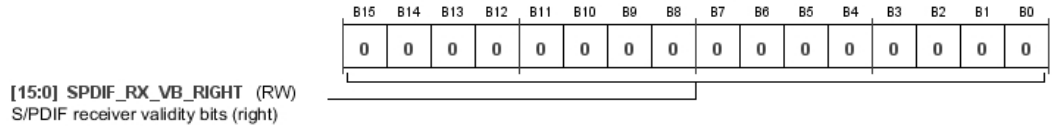


Table 151. Bit Descriptions for SPDIF_RX_VB_RIGHT_x

Bits	Bit Name	Settings	Description	Reset	Access
[15:0]	SPDIF_RX_VB_RIGHT		S/PDIF receiver validity bits (right).	0x0000	R

S/PDIF Receiver Parity Bits (Left) Register

Address: 0xF670 to 0xF67B (Increments of 0x1), Reset: 0x0000, Name: SPDIF_RX_PB_LEFT_x

These 12 registers store the 192 parity bits decoded from the left channel of the S/PDIF input stream on the ADAU1463 and ADAU1467.

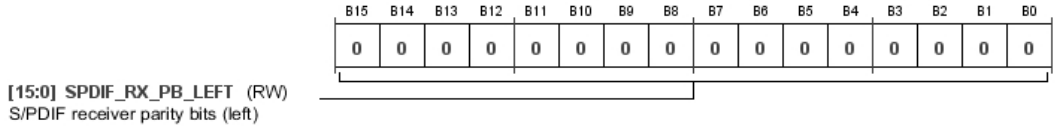


Table 152. Bit Descriptions for SPDIF_RX_PB_LEFT_x

Bits	Bit Name	Settings	Description	Reset	Access
[15:0]	SPDIF_RX_PB_LEFT		S/PDIF receiver parity bits (left).	0x0000	R

S/PDIF Receiver Parity Bits (Right) Register

Address: 0xF680 to 0xF68B (Increments of 0x1), Reset: 0x0000, Name: SPDIF_RX_PB_RIGHT_x

These 12 registers store the 192 parity bits decoded from the right channel of the S/PDIF input stream on the ADAU1463 and ADAU1467.

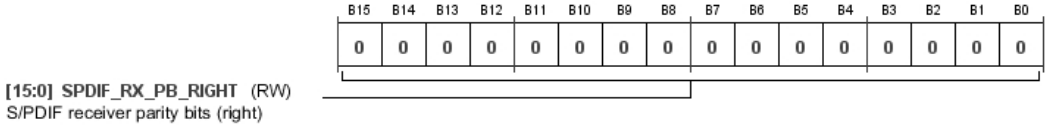


Table 153. Bit Descriptions for SPDIF_RX_PB_RIGHT_x

Bits	Bit Name	Settings	Description	Reset	Access
[15:0]	SPDIF_RX_PB_RIGHT		S/PDIF receiver parity bits (right).	0x0000	R

MOSI/ADDR1 Pin Drive Strength and Slew Rate Register

Address: 0xF79C, Reset: 0x0018, Name: MOSI_ADDR1_PIN

This register configures the drive strength, slew rate, and pull resistors for the MOSI/ADDR1 pin.

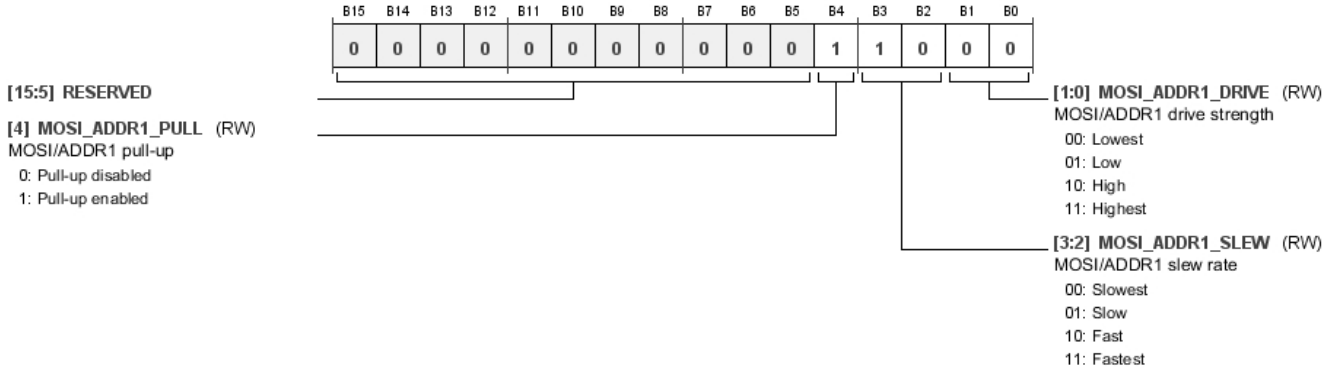


Table 175. Bit Descriptions for MOSI_ADDR1_PIN

Bits	Bit Name	Settings	Description	Reset	Access
[15:5]	RESERVED		Reserved.	0x0	RW
4	MOSI_ADDR1_PULL		MOSI/ADDR1 pull-up.	0x1	RW
		0	Pull-up disabled.		
		1	Pull-up enabled.		
[3:2]	MOSI_ADDR1_SLEW		MOSI/ADDR1 slew rate.	0x2	RW
		00	Slowest.		
		01	Slow.		
		10	Fast.		
		11	Fastest.		
[1:0]	MOSI_ADDR1_DRIVE		MOSI/ADDR1 drive strength.	0x0	RW
		00	Lowest.		
		01	Low.		
		10	High.		
		11	Highest.		

MP6 Pin Drive Strength and Slew Rate Register

Address: 0xF7A1, Reset: 0x0018, Name: MP6_PIN

This register configures the drive strength, slew rate, and pull resistors for the MP6 pin.

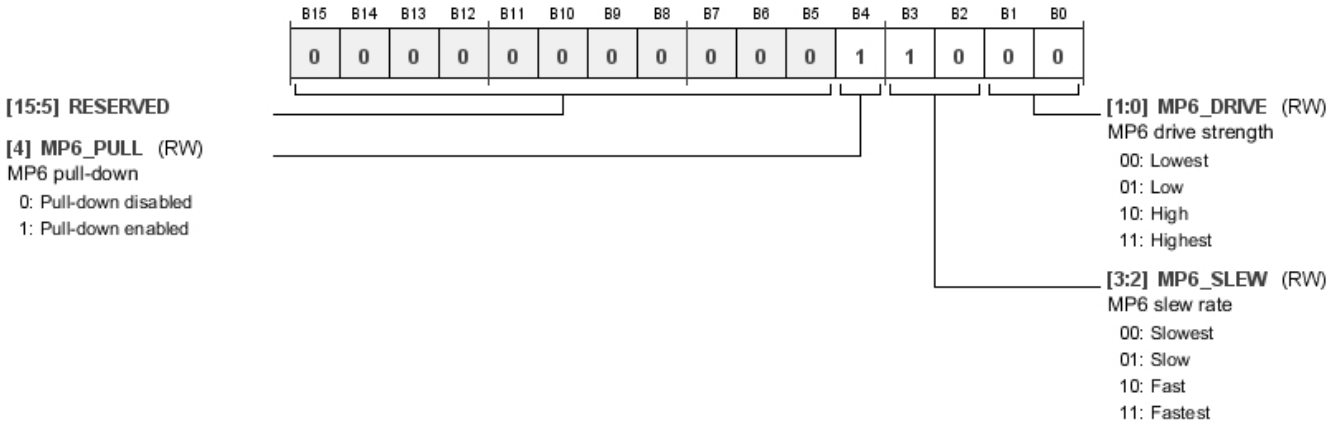


Table 180. Bit Descriptions for MP6_PIN

Bits	Bit Name	Settings	Description	Reset	Access
[15:5]	RESERVED		Reserved.	0x0	RW
4	MP6_PULL		MP6 pull-down.	0x1	RW
		0	Pull-down disabled.		
		1	Pull-down enabled.		
[3:2]	MP6_SLEW		MP6 slew rate.	0x2	RW
		00	Slowest.		
		01	Slow.		
		10	Fast.		
		11	Fastest.		
[1:0]	MP6_DRIVE		MP6 drive strength.	0x0	RW
		00	Lowest.		
		01	Low.		
		10	High.		
		11	Highest.		

PCB MANUFACTURING GUIDELINES

The soldering profile in Figure 91 is recommended for the LFCSP package. See the AN-772 Application Note for more information about PCB manufacturing guidelines.

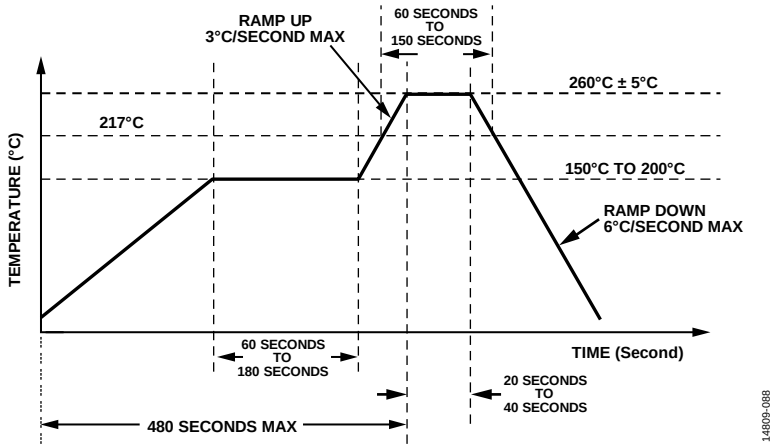


Figure 91. Soldering Profile

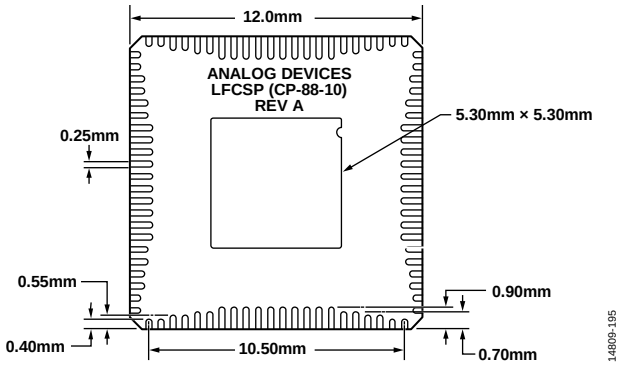


Figure 92. PCB Decal Dimensions