

FEATURES

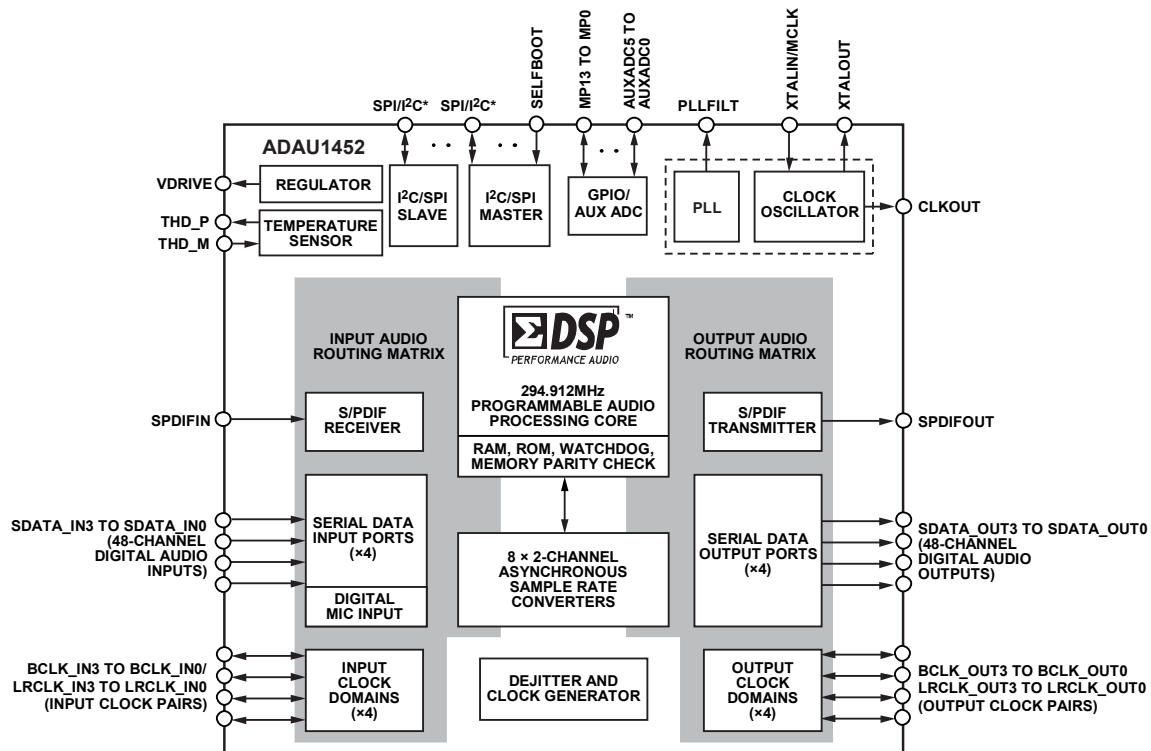
- Qualified for automotive applications
- Fully programmable audio DSP for enhanced sound processing
- Features SigmaStudio, a proprietary graphical programming tool for the development of custom signal flows
- 294.912 MHz, 32-bit SigmaDSP core at 1.2 V
 - Up to 6144 SIMD instructions per sample at 48 kHz
 - 40 kWords of parameter/data RAM
 - Up to 800 ms digital audio delay pool at 48 kHz
- Audio I/O and routing
 - 4 serial input ports, 4 serial output ports
 - 48-channel, 32-bit digital I/O up to a sample rate of 192 kHz
 - Flexible configuration for TDM, I²S, left and right justified formats, and PCM
 - Up to 8 stereo ASRCs from 1:8 up to 7.75:1 ratio and 139 dB DNR
 - Stereo S/PDIF input and output
 - Four PDM microphone input channels
 - Multichannel, byte addressable TDM serial ports

- Clock oscillator for generating master clock from crystal
- Integer PLL and flexible clock generators
- Integrated die temperature sensor
- I²C and SPI control interfaces (both slave and master)
- Standalone operation
 - Self boot from serial EEPROM
 - 6-channel, 10-bit SAR auxiliary control ADC
 - 14 multipurpose pins for digital controls and outputs
- On-chip regulator for generating 1.2 V from 3.3 V supply
- 72-lead, 10 mm × 10 mm LFCSP package with 5.3 mm exposed pad
- Temperature range: -40°C to +105°C

APPLICATIONS

- Automotive audio processing
 - Head units
 - Navigation systems
 - Rear seat entertainment systems
 - DSP amplifiers (sound system amplifiers)
- Commercial and professional audio processing

FUNCTIONAL BLOCK DIAGRAM



*SPI/I²C INCLUDES THE FOLLOWING PIN FUNCTIONS: SS_M, MOSI_M, SCL_M, SCLK_M, SDA_M, MISO_M, MISO, SDA, SCLK, SCL, MOSI, ADDR1, SS, AND ADDR0 PINS.

Figure 1.

11485-001

Rev. A

Document Feedback

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

TABLE OF CONTENTS

| | | | |
|--|----|--|-----|
| Features | 1 | SigmaDSP Core | 72 |
| Applications..... | 1 | Software Features | 75 |
| Functional Block Diagram | 1 | Pin Drive Strength, Slew Rate, and Pull Configuration..... | 77 |
| Revision History | 2 | Global RAM and Control Register Map..... | 78 |
| General Description | 3 | Random Access Memory | 78 |
| Specifications..... | 4 | Control Registers..... | 78 |
| Electrical Characteristics | 6 | Control Register Details | 89 |
| Timing Specifications | 7 | PLL Configuration Registers | 89 |
| Absolute Maximum Ratings..... | 15 | Clock Generator Registers | 93 |
| Thermal Characteristics | 15 | Power Reduction Registers | 97 |
| Maximum Power Dissipation | 15 | Audio Signal Routing Registers..... | 100 |
| ESD Caution..... | 15 | Serial Port Configuration Registers | 106 |
| Pin Configuration and Function Descriptions..... | 16 | Flexible TDM Interface Registers..... | 109 |
| Theory of Operation | 20 | DSP Core Control Registers..... | 113 |
| System Block Diagram..... | 20 | Debug and Reliability Registers..... | 118 |
| Overview..... | 20 | DSP Program Execution Registers..... | 126 |
| Initialization | 22 | Multipurpose Pin Configuration Registers..... | 130 |
| Master Clock, PLL, and Clock Generators..... | 25 | ASRC Status and Control Registers | 135 |
| Power Supplies, Voltage Regulator, and Hardware Reset..... | 29 | Auxiliary ADC Registers..... | 138 |
| Temperature Sensor Diode..... | 31 | S/PDIF Interface Registers | 139 |
| Slave Control Ports..... | 31 | Hardware Interfacing Registers | 152 |
| Master Control Ports..... | 37 | Soft Reset Register..... | 170 |
| Self Boot..... | 38 | Applications Information | 171 |
| Audio Signal Routing..... | 40 | PCB Design Considerations | 171 |
| Serial Data Input/Output..... | 50 | Typical Applications Block Diagram | 172 |
| Flexible TDM Interface..... | 60 | Example PCB Layout | 173 |
| Asynchronous Sample Rate Converters | 65 | PCB Manufacturing Guidelines | 174 |
| S/PDIF Interface | 66 | Outline Dimensions | 175 |
| Digital PDM Microphone Interface..... | 68 | Ordering Guide | 175 |
| Multipurpose Pins | 69 | Automotive Products..... | 175 |
| Auxiliary ADC..... | 72 | | |

REVISION HISTORY

1/14—Rev. 0 to Rev. A

Changed S/PDIF Transceiver and Receiver Maximum Audio Sample Rate from 192 kHz to 96 kHz; Table 9 and Table 10 9

10/13—Revision 0: Initial Version

GENERAL DESCRIPTION

The [ADAU1452](#) is an automotive-qualified audio processor that far exceeds the digital signal processing capabilities of earlier SigmaDSP® devices. Its restructured hardware architecture is optimized for efficient audio processing. The audio processing algorithms are realized in sample-by-sample and block-by-block paradigms, which can both be executed simultaneously in a signal processing flow created using the graphical programming tool, SigmaStudio™. The new digital signal processor (DSP) core architecture enables some types of audio processing algorithms to be executed using significantly fewer instructions than were required on previous SigmaDSP generations, leading to vastly improved code efficiency.

The 1.2 V, 32-bit DSP core can run at frequencies of up to 294.912 MHz and execute up to 6144 instructions per sample at the standard sample rate of 48 kHz. However, a wide range of other sample rates are available, in addition to industry standard rates. The integer PLL and flexible clock generator hardware can generate up to 15 audio sample rates simultaneously. These clock generators, along with the on-board asynchronous sample rate converters (ASRCs) and flexible hardware audio routing matrix, make the [ADAU1452](#) an ideal audio hub that greatly simplifies the design of complex multirate audio systems.

The [ADAU1452](#) interfaces with a wide range of ADCs, DACs, digital audio devices, amplifiers, and control circuitry, due to its highly configurable serial ports, S/PDIF interfaces, and multipurpose input/output pins. It can also directly interface with PDM-output MEMS microphones, thanks to integrated decimation filters specifically designed for that purpose.

Independent slave and master I²C/SPI control ports allow the [ADAU1452](#) not only to be programmed and configured by an external master device, but also to program and configure external slave devices directly. This, combined with self boot functionality, enables the design of standalone systems that do not require any external input to operate.

The power efficient DSP core executes full programs while consuming only a few hundred milliwatts (mW) of power and can run at a maximum program load while consuming less than a watt, even in worst case temperatures exceeding 100°C. This relatively low power consumption and small footprint make the [ADAU1452](#) an ideal replacement for large, general-purpose DSPs that consume more power at the same processing load.

SPECIFICATIONS

AVDD = 3.3 V ± 10%, DVDD = 1.2 V ± 5%, PVDD = 3.3 V ± 10%, IOVDD = 1.8 V – 10% to 3.3 V + 10%, T_A = 25°C, master clock input = 12.288 MHz, core clock (f_{CORE}) = 294.912 MHz, I/O pins set to low drive setting, unless otherwise noted.

Table 1.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|--|------|------|--------|------|--|
| POWER | | | | | |
| Supply Voltage | | | | | |
| Analog Voltage (AVDD) | 2.97 | 3.3 | 3.63 | V | Supply for analog circuitry, including auxiliary ADC |
| Digital Voltage (DVDD) | 1.14 | 1.2 | 1.26 | V | Supply for digital circuitry, including the DSP core, ASRCs, and signal routing |
| PLL Voltage (PVDD) | 2.97 | 3.3 | 3.63 | V | Supply for phase-locked loop (PLL) circuitry |
| I/O Supply Voltage (IOVDD) | 1.71 | 3.3 | 3.63 | V | Supply for input/output circuitry, including pads and level shifters |
| Supply Current | | | | | |
| Analog Current (AVDD) | | | | | |
| Idle State | 0 | 5 | 40 | µA | Power applied, chip not programmed |
| Reset State | 1.9 | 6.5 | 40 | µA | Power applied, RESET held low |
| PLL Current (PVDD) | | | | | |
| Idle State | 0 | 7.3 | 40 | µA | 12.288 MHz MCLK with default PLL settings |
| Reset State | 3.9 | 8.5 | 40 | µA | Power applied, RESET held low |
| I/O Current (IOVDD) | | | | | |
| Operation State | | | | | |
| | | 53 | | mA | IOVDD = 3.3 V; all serial ports are clock masters |
| | | 22 | | mA | IOVDD = 1.8 V; all serial ports are clock masters |
| Power-Down State | | | | | |
| | | 0.3 | 2.5 | mA | IOVDD = 1.8 V – 10% to 3.3 V + 10% |
| Digital Current (DVDD) | | | | | |
| Operation State | | | | | |
| Maximum Program | | 350 | 415 | mA | Test program includes 16-channel I/O, 10-band EQ per channel, all ASRCs active |
| Typical Program | | 100 | | mA | |
| Minimal Program | | | | | |
| Idle State | | 85 | | mA | Test program includes 2-channel I/O, 10-band EQ per channel |
| Reset State | | 20 | 95 | mA | Power applied, DSP not enabled |
| | | 20 | 95 | mA | Power applied, RESET held low |
| ASYNCHRONOUS SAMPLE RATE CONVERTERS | | | | | |
| Dynamic Range | | | | | |
| | | 139 | | dB | A-weighted, 20 Hz to 20 kHz |
| I/O Sample Rate | | | | | |
| | 6 | | 192 | kHz | |
| I/O Sample Rate Ratio | | | | | |
| | 1:8 | | 7.75:1 | | |
| THD + N | | | | | |
| | | | -120 | dB | |
| CRYSTAL OSCILLATOR | | | | | |
| Transconductance | | | | | |
| | 8.3 | 10.6 | 13.4 | mS | |
| REGULATOR | | | | | |
| DVDD Voltage | | | | | |
| | 1.14 | 1.2 | | V | Regulator maintains typical output voltage up to a maximum 800 mA load; IOVDD = 1.8 V – 10% to 3.3 V + 10% |

AVDD = 3.3 V ± 10%, DVDD = 1.2 V ± 5%, PVDD = 3.3 V ± 10%, IOVDD = 1.8 V – 10% to 3.3 V + 10%, T_A = –40°C to +105°C, master clock input = 12.288 MHz, core clock (f_{CORE}) = 294.912 MHz, I/O pins set to low drive setting, unless otherwise noted.

Table 2.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|------|------|--------|------|--|
| POWER | | | | | |
| Supply Voltage | | | | | |
| Analog Voltage (AVDD) | 2.97 | 3.3 | 3.63 | V | Supply for analog circuitry, including auxiliary ADC |
| Digital Voltage (DVDD) | 1.14 | 1.2 | 1.26 | V | Supply for digital circuitry, including the DSP core, ASRCs, and signal routing |
| PLL Voltage (PVDD) | 2.97 | 3.3 | 3.63 | V | Supply for PLL circuitry |
| IOVDD Voltage (IOVDD) | 1.71 | 3.3 | 3.63 | V | Supply for input/output circuitry, including pads and level shifters |
| Supply Current | | | | | |
| Analog Current (AVDD) | | | | | |
| Idle State | 0 | 6.3 | 40 | µA | |
| Reset State | 0.26 | 7.1 | 40 | µA | |
| PLL Current (PVDD) | | | | | |
| Idle State | 0 | 7.8 | 40 | µA | 12.288 MHz master clock; default PLL settings |
| Reset State | 1.2 | 9.3 | 40 | µA | Power applied, PLL not configured |
| I/O Current (IOVDD) | | | | | |
| Dependent on the number of active serial ports, clock pins, and characteristics of external loads | | | | | |
| Operation State | | | | | |
| | | 47 | | mA | IOVDD = 3.3 V; all serial ports are clock masters |
| | | 15 | | mA | IOVDD = 1.8 V; all serial ports are clock masters |
| Power-Down State | | | | | |
| | | 1.3 | 2.2 | mA | IOVDD = 1.8 V – 10% to 3.3 V + 10% |
| Digital Current (DVDD) | | | | | |
| Operation State | | | | | |
| Maximum Program | | | | | |
| | | 500 | 690 | mA | |
| Typical Program | | | | | |
| | | 200 | | mA | Test program includes 16-channel I/O, 10-band EQ per channel, all ASRCs active |
| Minimal Program | | | | | |
| | | 160 | | mA | Test program includes 2-channel I/O, 10-band EQ per channel |
| Idle State | | | | | |
| | | 315 | 635 | mA | |
| Reset State | | | | | |
| | | 315 | 635 | mA | |
| ASYNCHRONOUS SAMPLE RATE CONVERTERS | | | | | |
| Dynamic Range | | | | | |
| | | 139 | | dB | A-weighted, 20 Hz to 20 kHz |
| I/O Sample Rate | | | | | |
| | 6 | | 192 | kHz | |
| I/O Sample Rate Ratio | | | | | |
| | 1:8 | | 7.75:1 | | |
| THD + N | | | | | |
| | | | –120 | dB | |
| CRYSTAL OSCILLATOR | | | | | |
| Transconductance | | | | | |
| | 8.1 | 10.6 | 14.6 | mS | |
| REGULATOR | | | | | |
| DVDD Voltage | | | | | |
| | 1.14 | 1.2 | | V | Regulator maintains typical output voltage up to a maximum 800 mA load; IOVDD = 1.8 V – 10% to 3.3 V + 10% |

ELECTRICAL CHARACTERISTICS

Digital Input/Output

Table 3.

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|--|------|-----|------|------|--|
| DIGITAL INPUT | | | | | |
| Input Voltage | | | | | |
| IOVDD = 3.3 V | | | | | |
| High Level (V _{IH}) ¹ | 1.71 | | 3.3 | V | |
| Low Level (V _{IL}) ¹ | 0 | | 1.71 | V | |
| IOVDD = 1.8 V | | | | | |
| High Level (V _{IH}) ¹ | 0.92 | | 1.8 | V | |
| Low Level (V _{IL}) ¹ | 0 | | 0.89 | V | |
| Input Leakage | | | | | |
| High Level (I _{IH}) | -2 | | +2 | μA | Digital input pins with pull-up resistor |
| | 2 | | 12 | μA | Digital input pins with pull-down resistor |
| | -2 | | +2 | μA | Digital input pins with no pull resistor |
| | 0 | | 8 | μA | MCLK |
| | 80 | | 120 | μA | SPDIFIN |
| Low Level (I _{IL}) at 0 V | -12 | | -2 | μA | Digital input pins with pull-up resistor |
| | -2 | | +2 | μA | Digital input pins with pull-down resistor |
| | -2 | | +2 | μA | Digital input pins with no pull resistor |
| | -8 | | 0 | μA | MCLK |
| | -120 | | -77 | μA | SPDIFIN |
| Input Capacitance (C _I) | | 2 | | pF | Guaranteed by design |
| 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 | | |
| Low Level (V _{OL}) | 0 | | 0.33 | | |
| 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 | | | | | |
| Lowest Drive Strength Setting | | | 1 | mA | The digital output pins are not designed for static current draw; do not use these pins to drive LEDs directly |
| Low Drive Strength Setting | | | 2 | mA | The digital output pins are not designed for static current draw; do not use these pins to drive LEDs directly |
| High Drive Strength Setting | | | 3 | mA | The digital output pins are not designed for static current draw; do not use these pins to drive LEDs directly |
| Highest Drive Strength Setting | | | 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 | | | | | |
| Lowest Drive Strength Setting | | | 2 | mA | The digital output pins are not designed for static current draw; do not use these pins to drive LEDs directly |
| Low Drive Strength Setting | | | 5 | mA | The digital output pins are not designed for static current draw; do not use these pins to drive LEDs directly |
| High Drive Strength Setting | | | 10 | mA | The digital output pins are not designed for static current draw; do not use these pins to drive LEDs directly |
| Highest Drive Strength Setting | | | 15 | mA | The digital output pins are not designed for static current draw; do not use these pins to drive LEDs directly |

¹ Digital input pins except SPDIFIN, which is not a standard digital input.

Auxiliary ADC

$T_A = -40^{\circ}\text{C}$ to $+105^{\circ}\text{C}$, $DVDD = 1.2\text{ V} \pm 5\%$, $AVDD = 3.3\text{ V} \pm 10\%$, $IOVDD = 1.8\text{ V} - 10\%$ to $3.3\text{ V} + 10\%$, unless otherwise noted.

Table 4.

| Parameter | Min | Typ | Max | Unit |
|---------------------------------|-----|------------------------|-----|------------|
| RESOLUTION | | 10 | | Bits |
| FULL-SCALE ANALOG INPUT | | AVDD | | V |
| NONLINEARITY | | | | |
| Integrated Nonlinearity (INL) | -2 | | +2 | LSB |
| Differential Nonlinearity (DNL) | -2 | | +2 | LSB |
| GAIN ERROR | -2 | | +2 | LSB |
| INPUT IMPEDANCE | | 200 | | k Ω |
| SAMPLE RATE | | $f_{\text{CORE}}/6144$ | | Hz |

TIMING SPECIFICATIONS

Master Clock Input

$T_A = -40^{\circ}\text{C}$ to $+105^{\circ}\text{C}$, $DVDD = 1.2\text{ V} \pm 5\%$, $IOVDD = 1.8\text{ V} - 10\%$ to $3.3\text{ V} + 10\%$, unless otherwise noted.

Table 5.

| Parameter | Min | Max | Unit | Description |
|---------------------------|-------------------------------|-------------------------------|------|---|
| MASTER CLOCK INPUT (MCLK) | | | | |
| f_{MCLK} | 2.375 | 36 | MHz | MCLK frequency |
| t_{MCLK} | 27.8 | 421 | ns | MCLK period |
| t_{MCLKD} | 25 | 75 | % | MCLK duty cycle |
| t_{MCLKH} | $0.25 \times t_{\text{MCLK}}$ | $0.75 \times t_{\text{MCLK}}$ | ns | MCLK width high |
| t_{MCLKL} | $0.25 \times t_{\text{MCLK}}$ | $0.75 \times t_{\text{MCLK}}$ | ns | MCLK width low |
| CLKOUT Jitter | 12 | 106 | ps | Cycle-to-cycle rms average |
| CORE CLOCK | | | | |
| f_{CORE} | 152 | 294.912 | MHz | System (DSP core) clock frequency; PLL feedback divider ranges from 64 to 108 |
| t_{CORE} | 3.39 | | ns | System (DSP core) clock period |

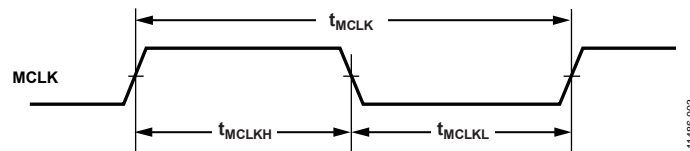


Figure 2. Master Clock Input Timing Specifications

Reset

$T_A = -40^{\circ}\text{C}$ to $+105^{\circ}\text{C}$, $DVDD = 1.2\text{ V} \pm 5\%$, $IOVDD = 1.8\text{ V} - 10\%$ to $3.3\text{ V} + 10\%$.

Table 6.

| Parameter | Min | Max | Unit | Description |
|-------------------|-----|-----|------|-----------------------|
| RESET | | | | |
| t_{WRST} | 10 | | ns | Reset pulse width low |

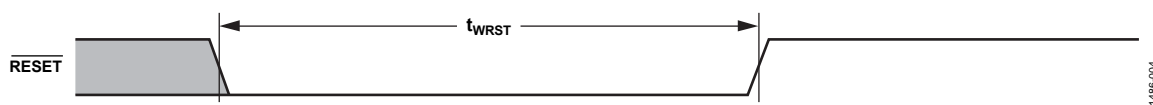


Figure 3. Reset Timing Specification

Serial Ports

TA = -40°C to +105°C, DVDD = 1.2 V ± 5%, IOVDD = 1.8 V - 10% to 3.3 V + 10%, unless otherwise noted. BCLK in Table 7 refers to BCLK_OUT3 to BCLK_OUT0 and BCLK_IN3 to BCLK_IN0. LRCLK refers to LRCLK_OUT3 to LRCLK_OUT0 and LRCLK_IN3 to LRCLK_IN0.

Table 7.

| Parameter | Min | Max | Unit | Description |
|--------------------|------|--------|------|---|
| SERIAL PORT | | | | |
| f _{LRCLK} | | 192 | kHz | LRCLK frequency |
| t _{LRCLK} | 5.21 | | µs | LRCLK period |
| f _{BCLK} | | 24.576 | MHz | BCLK frequency, sample rate ranging from 6 kHz to 192 kHz |
| t _{BCLK} | 40.7 | | ns | BCLK period |
| t _{BIL} | 10 | | ns | BCLK low pulse width, slave mode; BCLK frequency = 24.576 MHz; BCLK period = 40.6 ns |
| t _{BIH} | 14.5 | | ns | BCLK high pulse width, slave mode; BCLK frequency = 24.576 MHz; BCLK period = 40.6 ns |
| t _{LIS} | 20 | | ns | LRCLK setup to BCLK_INx input rising edge, slave mode; LRCLK frequency = 192 kHz |
| t _{LIH} | 5 | | ns | LRCLK hold from BCLK_INx input rising edge, slave mode; LRCLK frequency = 192 kHz |
| t _{SIS} | 5 | | ns | SDATA_INx setup to BCLK_INx input rising edge |
| t _{SIH} | 5 | | ns | SDATA_INx hold from BCLK_INx input rising edge |
| t _{TS} | | 10 | ns | BCLK_OUTx output falling edge to LRCLK_OUTx output timing skew, slave |
| t _{SODS} | | 35 | ns | SDATA_OUTx delay in slave mode from BCLK_OUTx output falling edge; serial outputs function in slave mode at all valid sample rates, provided that the external circuit design provides sufficient electrical signal integrity |
| t _{SODM} | | 10 | ns | SDATA_OUTx delay in master mode from BCLK_OUTx output falling edge |
| t _{TM} | | 5 | ns | BCLK falling edge to LRCLK timing skew, master |

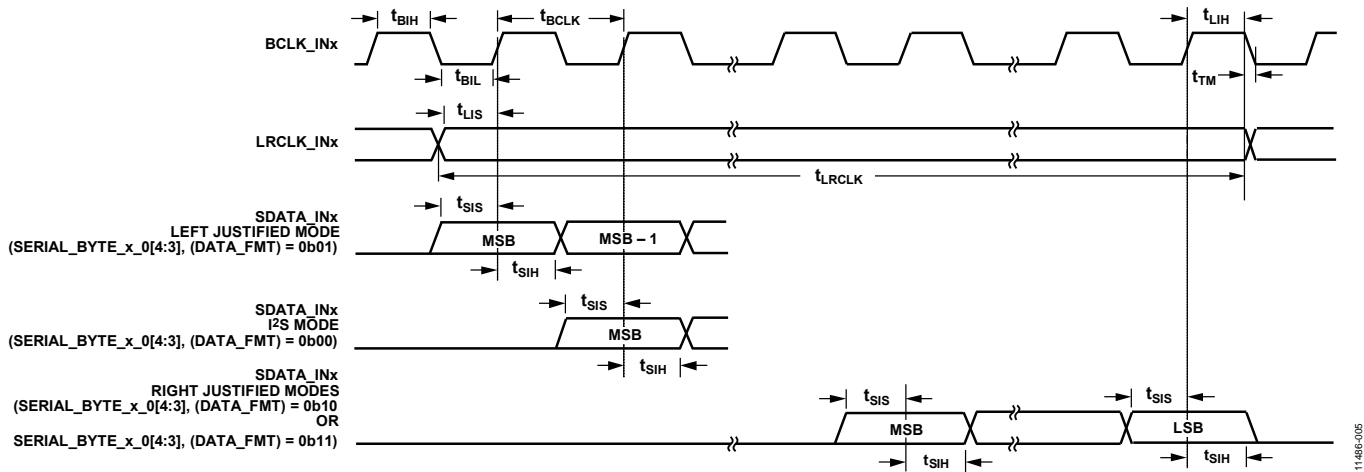


Figure 4. Serial Input Port Timing Specifications

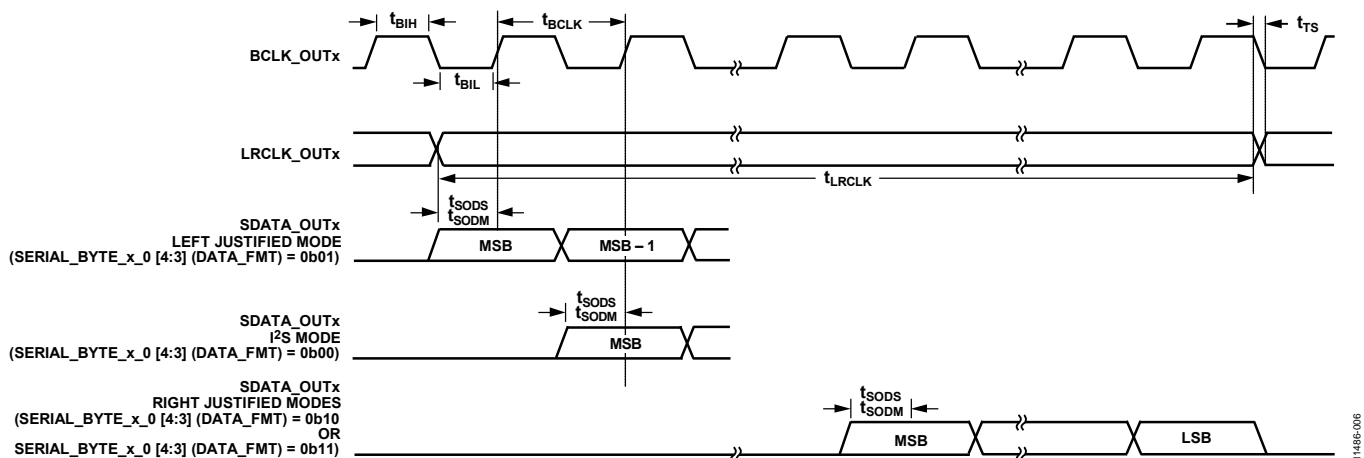


Figure 5. Serial Output Port Timing Specifications

Multipurpose Pins

$T_A = -40^{\circ}\text{C}$ to $+105^{\circ}\text{C}$, $DVDD = 1.2\text{ V} \pm 5\%$, $IOVDD = 1.8\text{ V} - 10\%$ to $3.3\text{ V} + 10\%$.

Table 8.

| Parameter | Min | Max | Unit | Description |
|-------------------------|----------------------|------------------------|------|---|
| MULTIPURPOSE PINS (MPx) | | | | |
| f_{MP}^1 | | 24.576 | MHz | MPx maximum switching rate when pin is configured as a general-purpose input or general-purpose output |
| t_{MPIL}^1 | $10 \times t_{CORE}$ | $6144 \times t_{CORE}$ | sec | MPx pin input latency until high/low value is read by core; the duration in the Max column is equal to the period of one audio sample when the DSP is processing 6144 instructions per sample |

¹ Guaranteed by design.

S/PDIF Transmitter

$T_A = -40^{\circ}\text{C}$ to $+105^{\circ}\text{C}$, $DVDD = 1.2\text{ V} \pm 5\%$, $IOVDD = 1.8\text{ V} - 10\%$ to $3.3\text{ V} + 10\%$.

Table 9.

| Parameter | Min | Max | Unit | Description |
|--------------------|-----|-----|------|--|
| S/PDIF Transmitter | | | | |
| Audio Sample Rate | 18 | 96 | kHz | Audio sample rate of data output from S/PDIF transmitter |

S/PDIF Receiver

$T_A = -40^{\circ}\text{C}$ to $+105^{\circ}\text{C}$, $DVDD = 1.2\text{ V} \pm 5\%$, $IOVDD = 1.8\text{ V} - 10\%$ to $3.3\text{ V} + 10\%$.

Table 10.

| Parameter | Min | Max | Unit | Description |
|-------------------|-----|-----|------|--|
| S/PDIF Receiver | | | | |
| Audio Sample Rate | 18 | 96 | kHz | Audio sample rate of data input to S/PDIF receiver |

I²C Interface—Slave

T_A = -40°C to +105°C, DVDD = 1.2 V ± 5%, IOVDD = 1.8 V - 10% to 3.3 V + 10%, default drive strength (f_{SCL}) = 400 kHz.

Table 11.

| Parameter | Min | Max | Unit | Description |
|----------------------------------|-----|-----|------|---|
| I²C SLAVE PORT | | | | |
| f _{SCL} | | 400 | kHz | SCL clock frequency |
| t _{SCLH} | 0.6 | | μs | SCL pulse width high |
| t _{SCLL} | 1.3 | | μs | SCL pulse width low |
| t _{SCS} | 0.6 | | μs | Start and repeated start condition setup time |
| t _{SCH} | 0.6 | | μs | Start condition hold time |
| t _{DS} | 100 | | ns | Data setup time |
| t _{DH} | 0.9 | | μs | Data hold time |
| t _{SCLR} | | 300 | ns | SCL rise time |
| t _{SCLF} | | 300 | ns | SCL fall time |
| t _{SDR} | | 300 | ns | SDA rise time |
| t _{SDF} | | 300 | ns | SDA fall time |
| t _{BFT} | 1.3 | | μs | Bus-free time between stop and start |
| t _{SUSTO} | 0.6 | | μs | Stop condition setup time |

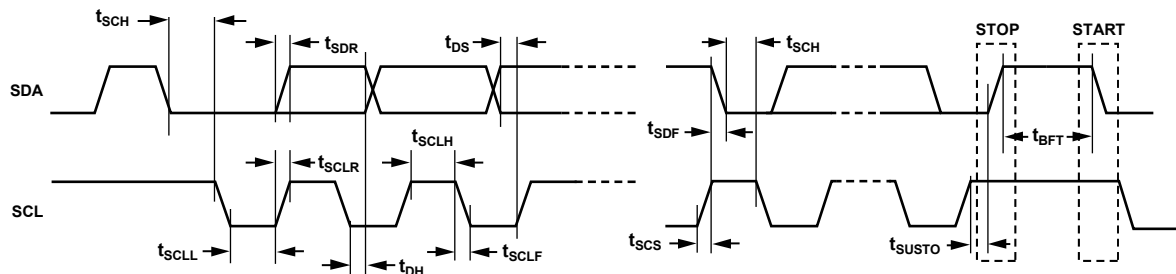


Figure 6. I²C Slave Port Timing Specifications

11486-007

I²C Interface—Master

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

Table 12.

| Parameter | Min | Max | Unit | Description |
|-----------------------------------|-----|-----|------|---|
| I²C MASTER PORT | | | | |
| f _{SCL} | | 400 | kHz | SCL clock frequency |
| t _{SCLH} | 0.6 | | μs | SCL pulse width high |
| t _{SCLL} | 1.3 | | μs | SCL pulse width low |
| t _{SCS} | 0.6 | | μs | Start and repeated start condition setup time |
| t _{SCH} | 0.6 | | μs | Start condition hold time |
| t _{DS} | 100 | | ns | Data setup time |
| t _{DH} | 0.9 | | μs | Data hold time |
| t _{SCLR} | | 300 | ns | SCL rise time |
| t _{SCLF} | | 300 | ns | SCL fall time |
| t _{SDR} | | 300 | ns | SDA rise time |
| t _{SDF} | | 300 | ns | SDA fall time |
| t _{BFT} | 1.3 | | μs | Bus-free time between stop and start |
| t _{SUSTO} | 0.6 | | μs | Stop condition setup time |

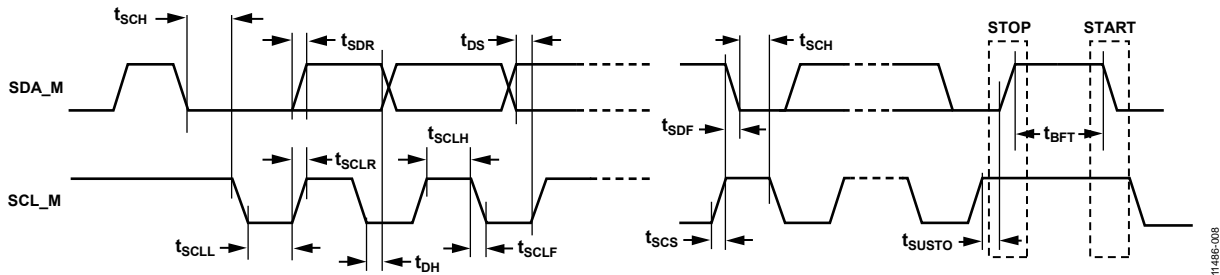


Figure 7. I²C Master Port Timing Specifications

11485-008

SPI Interface—Slave

$T_A = -40^{\circ}\text{C}$ to $+105^{\circ}\text{C}$, DVDD = 1.2 V \pm 5%, IOVDD = 1.8 V - 10% to 3.3 V + 10%.

Table 13.

| Parameter | Min | Max | Unit | Description |
|------------------------|-----|-----|------|--|
| SPI SLAVE PORT | | | | |
| $f_{\text{SCLKWRITE}}$ | | 22 | MHz | SCLK write frequency |
| f_{SCLKREAD} | | 22 | MHz | SCLK read frequency |
| t_{SCLKPWL} | 6 | | ns | SCLK pulse width low, SCLK = 22 MHz |
| t_{SCLKPWH} | 21 | | ns | SCLK pulse width high, SCLK = 22 MHz |
| t_{SSS} | 1 | | ns | SS setup to SCLK rising edge |
| t_{SSH} | 2 | | ns | SS hold from SCLK rising edge |
| t_{SSPWH} | 10 | | ns | SS pulse width high |
| t_{SSPWL} | 10 | | ns | SS pulse width low; minimum low pulse width for SS when entering SPI mode by toggling the SS pin three times |
| t_{MOSIS} | 1 | | ns | MOSI setup to SCLK rising edge |
| t_{MOSIH} | 2 | | ns | MOSI hold from SCLK rising edge |
| t_{MISOD} | | 39 | ns | MISO valid output delay from SCLK falling edge |

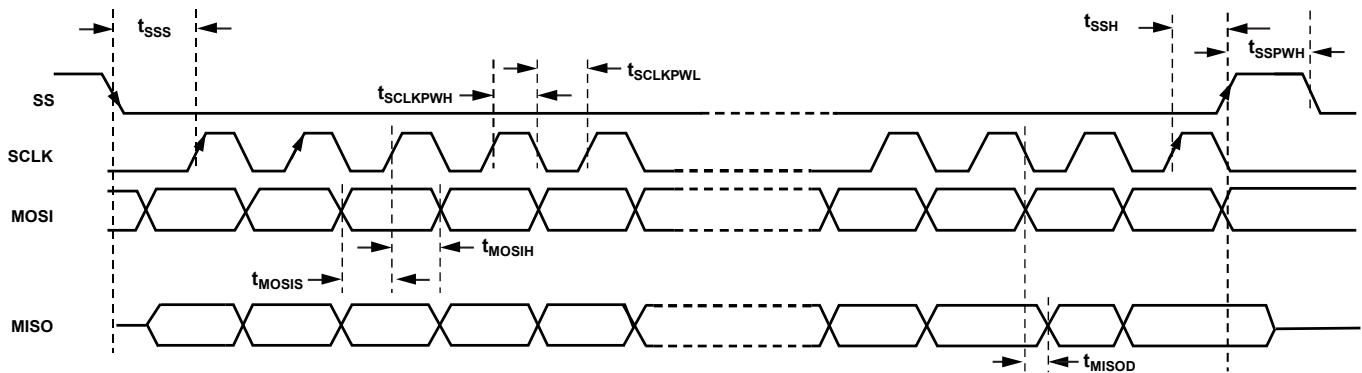


Figure 8. SPI Slave Port Timing Specifications

11486-019

SPI Interface—Master

$T_A = -40^{\circ}\text{C}$ to $+105^{\circ}\text{C}$, $\text{DVDD} = 1.2\text{ V} \pm 5\%$, $\text{IOVDD} = 1.8\text{ V} - 10\%$ to $3.3\text{ V} + 10\%$.

Table 14.

| Parameter | Min | Max | Unit | Description |
|----------------------------------|------|------|------|---|
| SPI MASTER PORT | | | | |
| Timing Requirements | | | | |
| t_{SSPIDM} | 15 | | ns | MISO_M data input valid to SCLK_M edge (data input setup time) |
| t_{HSPIDM} | 5 | | ns | SCLK_M last sampling edge to data input not valid (data input hold time) |
| Switching Characteristics | | | | |
| t_{SPICLK} | 41.7 | | ns | SPI master clock cycle period |
| $f_{\text{SCLK_M}}$ | | 24 | MHz | SPI master clock frequency |
| t_{SPICHM} | 17 | | ns | SCLK_M high period ($f_{\text{SCLK_M}} = 24\text{ MHz}$) |
| t_{SPICLM} | 17 | | ns | SCLK_M low period ($f_{\text{SCLK_M}} = 24\text{ MHz}$) |
| t_{DDSPIDM} | | 16.9 | ns | SCLK_M edge to data out valid (data out delay time) ($f_{\text{SCLK_M}} = 24\text{ MHz}$) |
| t_{HDSPIDM} | 21 | | ns | SCLK_M edge to data out not valid (data out hold time) ($f_{\text{SCLK_M}} = 24\text{ MHz}$) |
| t_{SDSCIM} | 36 | | ns | SS_M (SPI device select) low to first SCLK_M edge ($f_{\text{SCLK_M}} = 24\text{ MHz}$) |
| t_{HDSM} | 95 | | ns | Last SCLK_M edge to SS_M high ($f_{\text{SCLK_M}} = 24\text{ MHz}$) |

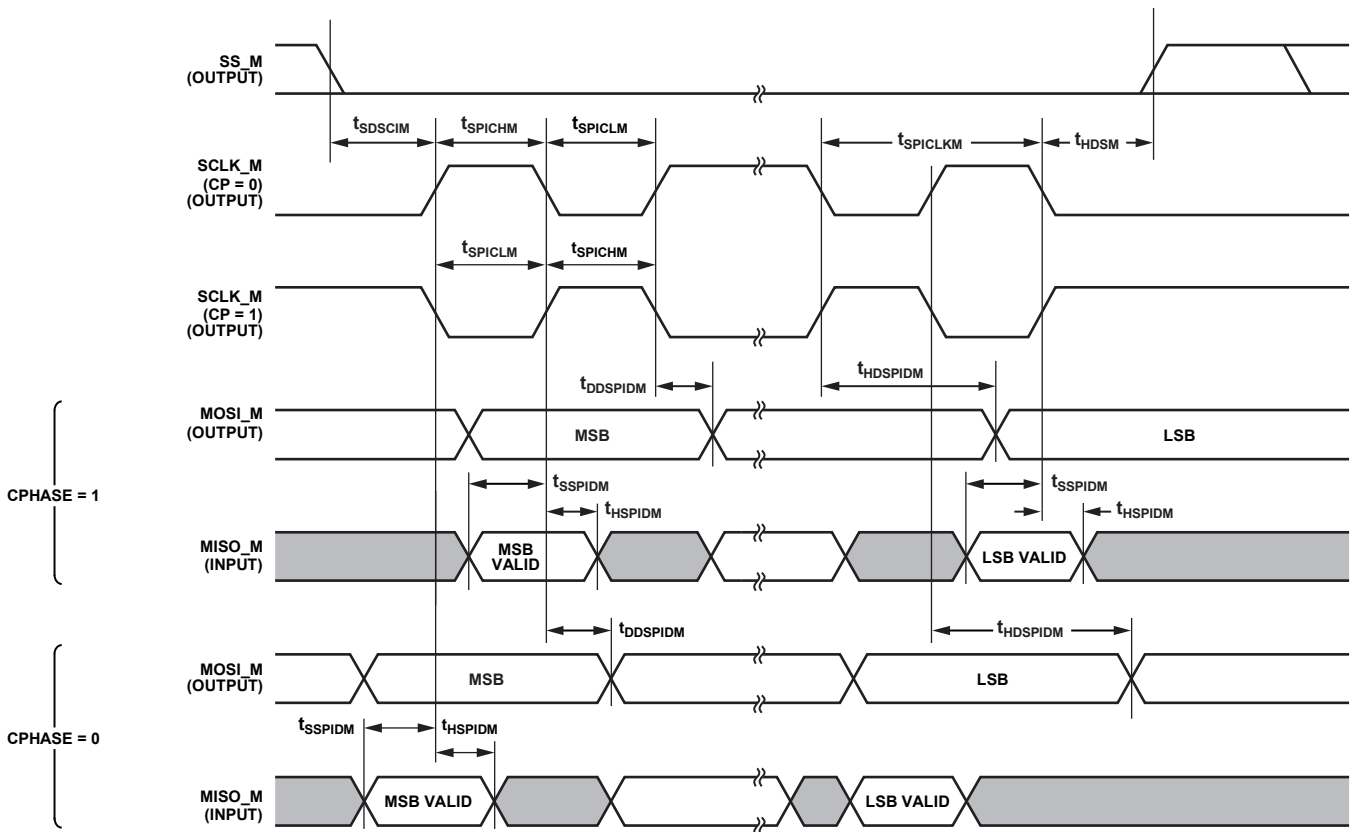


Figure 9. SPI Master Port Timing Specifications

1145B-010

PDM Inputs

$T_A = -40^{\circ}\text{C}$ to $+105^{\circ}\text{C}$, $DVDD = 1.2\text{ V} \pm 5\%$, $IOVDD = 1.8\text{ V} - 10\%$ to $3.3\text{ V} + 10\%$. PDM data is latched on both edges of the clock (see Figure 10).

Table 15.

| Parameter | t_{MIN} | t_{MAX} | Unit | Description |
|---------------------|------------------|------------------|------|-----------------|
| Timing Requirements | | | | |
| t_{SETUP} | 10 | | ns | Data setup time |
| t_{HOLD} | 5 | | ns | Data hold time |

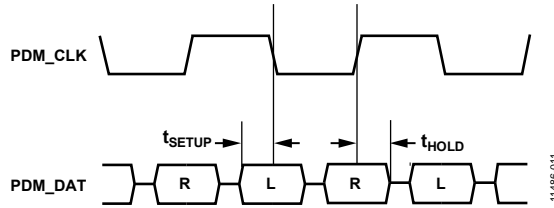


Figure 10. PDM Timing Diagram

11498-011

ABSOLUTE MAXIMUM RATINGS

Table 16.

| Parameter | Rating |
|-----------------------------------|-------------------------------|
| DVDD to Ground | 0 V to 1.4 V |
| AVDD to Ground | 0 V to 4.0 V |
| IOVDD to Ground | 0 V to 4.0 V |
| PVDD to Ground | 0 V to 4.0 V |
| Digital Inputs | DGND – 0.3 V to IOVDD + 0.3 V |
| Maximum Ambient Temperature Range | –40°C to +105°C |
| Maximum Junction Temperature | 125°C |
| Storage Temperature Range | –65°C to +150°C |
| Soldering (10 sec) | 300°C |

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

THERMAL CHARACTERISTICS

θ_{JA} represents the junction-to-ambient thermal resistance; θ_{JC} represents the junction-to-case thermal resistance. All characteristics are for a 4-layer JEDEC board. The exposed pad has 49 vias that are arranged in a 7 × 7 grid.

Table 17. Thermal Resistance

| Package Type | θ_{JA} | θ_{JC} | Unit |
|---------------|---------------|---------------|------|
| 72-Lead LFCSP | 23.38 | 3.3 | °C/W |

MAXIMUM POWER DISSIPATION

The specifications listed in Table 18 show the absolute worst case power dissipation for the ADAU1452. These tests were conducted at an ambient temperature of 105°C, with a completely full DSP program that executes an endless loop of the most power-intensive core calculations and with all power supplies at their maximum values.

In other words, the conditions described in Table 18 are intended only as a stress test and are not representative of realistic device operation in a real-world application. In a system where the operating conditions and limits outlined in the Specifications section of this document are not exceeded, and where the device is mounted to a printed circuit board (PCB) that follows the design recommendations in the PCB Design Considerations section of this document, the values that are listed represent the total power consumption of the device. In actual applications, the power consumption of the device is far lower. Table 19 shows a more realistic estimate for power consumption in a typical use case.

Table 18. Worst Case Maximum Power Dissipation

| Parameter | Value | Unit | Test Conditions/Comments |
|--|-------|------|--|
| AVDD, DVDD, PVDD During ADAU1452 Operation | 960 | mW | Ambient temperature = 105°C, all supplies at maximum, full DSP program using most power intensive calculations; measurement does not include IOVDD |
| Reset All Supplies | 570 | mW | Ambient temperature = 105°C, all supplies at maximum, reset mode enabled; measurement does not include IOVDD |

Table 19. Typical Power Dissipation Estimates

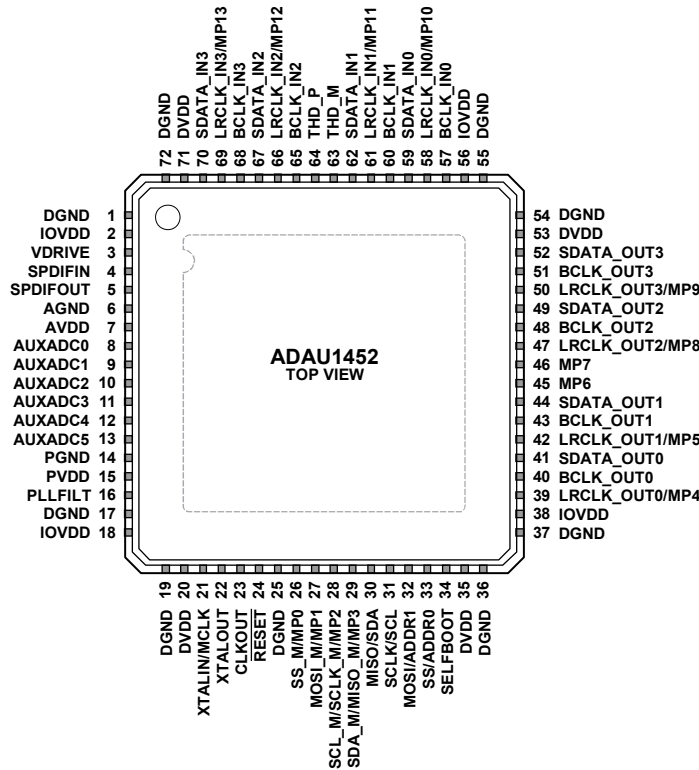
| Ambient Temperature (°C) | Full Program (mW) | Typical (mW) |
|--------------------------|-------------------|--------------|
| 25 | 420 | 250 |
| 85 | 700 | 420 |
| 105 | 885 | 530 |

ESD CAUTION



ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS



NOTES
 1. THE EXPOSED PAD MUST BE GROUNDED BY SOLDERING IT TO A COPPER SQUARE OF EQUIVALENT SIZE ON THE PCB. IDENTICAL COPPER SQUARES MUST EXIST ON ALL LAYERS OF THE BOARD, CONNECTED BY VIAS, AND THEY MUST BE CONNECTED TO A DEDICATED COPPER GROUND LAYER WITHIN THE PCB.

1148B-002

Figure 11. Pin Configuration

Table 20. Pin Function Descriptions

| Pin No. | Mnemonic | Internal Pull Resistor | Description |
|---------|----------|------------------------|--|
| 1 | DGND | None | Digital and I/O Ground Reference. Tie all DGND, AGND, and PGND pins directly together in a common ground plane. |
| 2 | IOVDD | None | Input/Output Supply, 1.8 V – 10% to 3.3 V + 10%. Bypass this pin with decoupling capacitors to Pin 1 (DGND). |
| 3 | VDRIVE | None | PNP Bipolar Junction Transistor-Base Drive Bias Pin for the Digital Supply Regulator. Connect to the base of an external PNP pass transistor (STD2805 is recommended). If an external supply is provided directly to DVDD, connect the VDRIVE pin to ground. |
| 4 | SPDIFIN | None | Input to the Integrated Sony/Philips Digital Interface Format Receiver. Leave this pin disconnected when not in use. This pin is internally biased to IOVDD/2. |
| 5 | SPDIFOUT | Configurable | Output from the Integrated Sony/Philips Digital Interface Format Transmitter. Leave this pin disconnected when not in use. |
| 6 | AGND | None | Analog Ground Reference. Tie all DGND, AGND, and PGND pins directly together in a common ground plane. |
| 7 | AVDD | None | Analog (Auxiliary ADC) Supply. Must be 3.3 V ± 10%. Bypass this pin with decoupling capacitors to Pin 6 (AGND). |
| 8 | AUXADC0 | None | Auxiliary ADC Input Channel 0. This pin reads an analog input signal and uses its value in the DSP program. Leave this pin disconnected when not in use. |
| 9 | AUXADC1 | None | Auxiliary ADC Input Channel 1. This pin reads an analog input signal and uses its value in the DSP program. Leave this pin disconnected when not in use. |
| 10 | AUXADC2 | None | Auxiliary ADC Input Channel 2. This pin reads an analog input signal and uses its value in the DSP program. Leave this pin disconnected when not in use. |

| Pin No. | Mnemonic | Internal Pull Resistor | Description |
|---------|---------------------------|--|--|
| 11 | AUXADC3 | None | Auxiliary ADC Input Channel 3. This pin reads an analog input signal and uses its value in the DSP program. Leave this pin disconnected when not in use. |
| 12 | AUXADC4 | None | Auxiliary ADC Input Channel 4. This pin reads an analog input signal and uses its value in the DSP program. Leave this pin disconnected when not in use. |
| 13 | AUXADC5 | None | Auxiliary ADC Input Channel 5. This pin reads an analog input signal and uses its value in the DSP program. Leave this pin disconnected when not in use. |
| 14 | PGND | None | PLL Ground Reference. Tie all DGND, AGND, and PGND pins directly together in a common ground plane. |
| 15 | PVDD | None | PLL Supply. Must be $3.3\text{ V} \pm 10\%$. Bypass this pin with decoupling capacitors to Pin 14 (PGND). |
| 16 | PLLFILT | None | PLL Filter. The voltage on the PLLFILT pin, which is internally generated, is typically between 1.65 V and 2.10 V. |
| 17 | DGND | None | Digital and I/O Ground Reference. Tie all DGND, AGND, and PGND pins directly together in a common ground plane. |
| 18 | IOVDD | None | Input/Output Supply, $1.8\text{ V} - 10\%$ to $3.3\text{ V} + 10\%$. Bypass this pin to Pin 17 (DGND) with decoupling capacitors. |
| 19 | DGND | None | Digital and I/O Ground Reference. Tie all DGND, AGND, and PGND pins directly together in a common ground plane. |
| 20 | DVDD | None | Digital Supply. Must be $1.2\text{ V} \pm 5\%$. This pin can be supplied externally or by using the internal regulator and external pass transistor. Bypass this pin to Pin 19 (DGND) with decoupling capacitors. |
| 21 | XTALIN/MCLK | None | Crystal Oscillator Input (XTALIN)/Master Clock Input to the PLL (MCLK). This pin can be supplied directly or generated by driving a crystal with the internal crystal oscillator via Pin 22 (XTALOUT). If a crystal is used, refer to the circuit shown in Figure 14. |
| 22 | XTALOUT | None | Crystal Oscillator Output for Driving an External Crystal. If a crystal is used, refer to the circuit shown in Figure 14. Leave this pin disconnected when not in use. |
| 23 | CLKOUT | Configurable | Master Clock Output. This pin drives a master clock signal to other ICs in the system. CLKOUT can be configured to output a clock signal with a frequency of 1x, 2x, 4x, or 8x the frequency of the divided clock signal being input to the PLL. Leave this pin disconnected when not in use. |
| 24 | $\overline{\text{RESET}}$ | Pull-down | Active Low Reset Input. A reset is triggered on a high-to-low edge and exited on a low-to-high edge. A reset event sets all RAMs and registers to their default values. |
| 25 | DGND | None | Digital and I/O Ground Reference. Tie all DGND, AGND, and PGND pins directly together in a common ground plane. |
| 26 | SS_M/MP0 | Pull-up; nominally 250 k Ω ; can be disabled by a write to control register | SPI Master/Slave Select Port (SS_M)/Multipurpose, General-Purpose Input/Output (MP0). When in SPI master mode, this pin acts as the slave select signal to slave devices on the SPI bus. The pin must go low at the beginning of a master SPI transaction and high at the end of a transaction. This pin has an internal pull-up resistor that is nominally 250 k Ω . When the SELFBOOT pin is held high and the $\overline{\text{RESET}}$ pin has a transition from low to high, Pin 26 sets the communications protocol for self boot operation. If this pin is left floating, the SPI communications protocol is used for self boot operation. If this pin has a 10 k Ω pull-down resistor to GND, the I ² C communications protocol is used for self boot operation. When self boot operation is not used and this pin is not needed as a general-purpose input or output, leave it disconnected. |
| 27 | MOSI_M/MP1 | Pull-up; can be disabled by a write to control register | SPI Master Data Output Port (MOSI_M)/Multipurpose, General-Purpose Input/Output (MP1). When in SPI master mode, this pin sends data from the SPI master port to slave devices on the SPI bus. Leave this pin disconnected when not in use. |
| 28 | SCL_M/ SCLK_M/MP2 | Pull-up; can be disabled by a write to control register | I ² C Master Serial Clock Port (SCL_M)/SPI Master Mode Slave Select (SCLK_M)/Multipurpose, General-Purpose Input/Output (MP2). When in I ² C master mode, this pin functions as an open collector output and drives a serial clock to slave devices on the I ² C bus; use a 2.0 k Ω pull-up resistor to IOVDD on the line connected to this pin. When in SPI master mode, this pin drives the clock signal to slave devices on the SPI bus. Leave this pin disconnected when not in use. |
| 29 | SDA_M/ MISO_M/MP3 | Pull-up; can be disabled by a write to control register | I ² C Master Port Serial Data (SDA_M)/SPI Master Mode Data Input (MISO_M)/Multipurpose, General-Purpose Input/Output (MP3). When in I ² C master mode, this pin functions as a bi-directional open collector data line between the I ² C master port and slave devices on the I ² C bus; use a 2.0 k Ω pull-up resistor to IOVDD on the line connected to this pin. When in SPI master mode, this pin receives data from slave devices on the SPI bus. Leave this pin disconnected when not in use. |

| Pin No. | Mnemonic | Internal Pull Resistor | Description |
|---------|--------------------|---|---|
| 30 | MISO/SDA | Pull-up; can be disabled by a write to control register | SPI Slave Data Output Port (MISO)/I ² C Slave Serial Data Port (SDA). In SPI slave mode, this pin outputs data to the master device on the SPI bus. In I ² C slave mode, this pin functions as a bi-directional open collector data line between the I ² C slave port and the master device on the I ² C bus; use a 2.0 kΩ pull-up resistor to IOVDD on the line connected to this pin. When this pin is not in use, connect it to IOVDD with a 10.0 kΩ pull-up resistor. |
| 31 | SCLK/SCL | Pull-up; can be disabled by a write to control register | SPI Slave Port Serial Clock (SCLK)/I ² C Slave Port Serial Clock (SCL). In SPI slave mode, this pin receives the serial clock signal from the master device on the SPI bus. In I ² C slave mode, this pin receives the serial clock signal from the master device on the I ² C bus; use a 2.0 kΩ pull-up resistor to IOVDD on the line connected to this pin. When this pin is not in use, connect it to IOVDD with a 10.0 kΩ pull-up resistor. |
| 32 | MOSI/ADDR1 | Pull-up; can be disabled by a write to control register | SPI Slave Port Data Input (MOSI)/I ² C Slave Port Address MSB (ADDR1). In SPI slave mode, this pin receives a data signal from the master device on the SPI bus. In I ² C slave mode, this pin acts as an input and sets the chip address of the I ² C slave port, in conjunction with Pin 33 (SS/ADDR0). |
| 33 | SS/ADDR0 | Pull-up, nominally 250 kΩ; can be disabled by a write to control register | SPI Slave Port Slave Select (SS)/I ² C Slave Port Address LSB (ADDR0). In SPI slave mode, this pin receives the slave select signal from the master device on the SPI bus. In I ² C slave mode, this pin acts as an input and sets the chip address of the I ² C slave port in conjunction with Pin 32 (MOSI/ADDR1). |
| 34 | SELFBOOT | Pull-up | Self Boot Select. This pin allows the device to perform a self boot, in which it loads its RAM and register settings from an external EEPROM. Connecting Pin 34 to logic high (IOVDD) initiates a self boot operation the next time there is a rising edge on Pin 24 (RESET). When this pin is connected to ground, no self boot operation is initiated. This pin can be connected to IOVDD or to ground either directly or pulled up or down with a 1.0 kΩ or larger resistor. |
| 35 | DVDD | None | Digital Supply. Must be 1.2 V ± 5%. This pin can be supplied externally or by using the internal regulator and external pass transistor. Bypass this pin to Pin 36 (DGND) with decoupling capacitors. |
| 36 | DGND | None | Digital and I/O Ground Reference. Tie all DGND, AGND, and PGND pins directly together in a common ground plane. |
| 37 | DGND | None | Digital and I/O Ground Reference. Tie all DGND, AGND, and PGND pins directly together in a common ground plane. |
| 38 | IOVDD | None | Input/Output Supply, 1.8 V – 10% to 3.3 V + 10%. Bypass this pin with decoupling capacitors to Pin 37 (DGND). |
| 39 | LRCLK_OUT0/ MP4 | Configurable | Frame Clock, Serial Output Port 0 (LRCLK_OUT0)/Multipurpose, General-Purpose Input/Output (MP4). This pin is bidirectional, with the direction depending on whether Serial Output Port 0 is set up as a master or slave. Leave this pin disconnected when not in use. |
| 40 | BCLK_OUT0 | Configurable | Bit Clock, Serial Output Port 0. This pin is bidirectional, with the direction depending on whether the Serial Output Port 0 is set up as a master or slave. Leave this pin disconnected when not in use. |
| 41 | SDATA_OUT0 | Configurable | Serial Data Output Port 0 (Channel 0 to Channel 15). Capable of 2-channel, 4-channel, 8-channel, and 16-channel modes. Leave this pin disconnected when not in use. |
| 42 | LRCLK_OUT1/ MP5 | Configurable | Frame Clock, Serial Output Port 1 (LRCLK_OUT1)/Multipurpose, General-Purpose Input/Output (MP5). This pin is bidirectional, with the direction depending on whether Serial Output Port 1 is set up as a master or slave. Leave this pin disconnected when not in use. |
| 43 | BCLK_OUT1 | Configurable | Bit Clock, Serial Output Port 1. This pin is bidirectional, with the direction depending on whether Output Serial Port 1 is set up as a master or slave. Leave this pin disconnected when not in use. |
| 44 | SDATA_OUT1 | Configurable | Serial Data Output Port 1 (Channel 16 to Channel 31). Capable of 2-channel, 4-channel, 8-channel, and 16-channel modes. Leave this pin disconnected when not in use. |
| 45 | MP6 | Configurable | Multipurpose, General-Purpose Input/Output 6. Leave this pin disconnected when not in use. |
| 46 | MP7 | Configurable | Multipurpose, General-Purpose Input/Output 7. Leave this pin disconnected when not in use. |
| 47 | LRCLK_OUT2/ MP8 | Configurable | Frame Clock, Serial Output Port 2 (LRCLK_OUT2)/Multipurpose, General-Purpose Input/Output (MP8). This pin is bidirectional, with the direction depending on whether Serial Output Port 2 is set up as a master or slave. Leave this pin disconnected when not in use. |
| 48 | BCLK_OUT2 | Configurable | Bit Clock, Serial Output Port 2. This pin is bidirectional, with the direction depending on whether Serial Output Port 2 is set up as a master or slave. Leave this pin disconnected when not in use. |
| 49 | SDATA_OUT2 | Configurable | Serial Data Output Port 2 (Channel 32 to Channel 39). Capable of 2-channel, 4-channel, 8-channel, or flexible TDM mode. Leave this pin disconnected when not in use. |
| 50 | LRCLK_OUT3/ MP9 | Configurable | Frame Clock, Serial Output Port 3 (LRCLK_OUT3)/Multipurpose, General-Purpose Input/Output (MP9). This pin is bidirectional, with the direction depending on whether Serial Output Port 3 is set up as a master or slave. Leave this pin disconnected when not in use. |

| Pin No. | Mnemonic | Internal Pull Resistor | Description |
|---------|--------------------|------------------------|---|
| 51 | BCLK_OUT3 | Configurable | Bit Clock, Serial Output Port 3. This pin is bidirectional, with the direction depending on whether Serial Output Port 3 is set up as a master or slave. Leave this pin disconnected when not in use. |
| 52 | SDATA_OUT3 | Configurable | Serial Data Output Port 3 (Channel 40 to Channel 47). Capable of 2-channel, 4-channel, 8-channel, and flexible TDM modes. Leave this pin disconnected when not in use. |
| 53 | DVDD | None | Digital Supply. Must be $1.2\text{ V} \pm 5\%$. This pin can be supplied externally or by using the internal regulator and external pass transistor. Bypass Pin 53 with decoupling capacitors to Pin 54 (DGND). |
| 54 | DGND | None | Digital and I/O Ground Reference. Tie all DGND, AGND, and PGND pins directly together in a common ground plane. |
| 55 | DGND | None | Digital and I/O Ground Reference. Tie all DGND, AGND, and PGND pins directly together in a common ground plane. |
| 56 | IOVDD | None | Input/Output Supply, $1.8\text{ V} - 10\%$ to $3.3\text{ V} + 10\%$. Bypass this pin with decoupling capacitors to Pin 55 (DGND). |
| 57 | BCLK_IN0 | Configurable | Bit Clock, Serial Input Port 0. This pin is bidirectional, with the direction depending on whether Serial Input Port 0 is set up as a master or slave. Leave this pin disconnected when not in use. |
| 58 | LRCLK_IN0/ MP10 | Configurable | Frame Clock, Serial Input Port 0 (LRCLK_IN0)/Multipurpose, General-Purpose Input/Output (MP10). This pin is bidirectional, with the direction depending on whether Serial Input Port 0 is set up as a master or slave. Leave this pin disconnected when not in use. |
| 59 | SDATA_IN0 | Configurable | Serial Data Input Port 0 (Channel 0 to Channel 15). Capable of 2-channel, 4-channel, 8-channel, or 16-channel mode. When not used, this pin can be left disconnected. |
| 60 | BCLK_IN1 | Configurable | Bit Clock, Serial Input Port 1. This pin is bidirectional, with the direction depending on whether the Serial Input Port 1 is set up as a master or slave. Leave this pin disconnected when not in use. |
| 61 | LRCLK_IN1/ MP11 | Configurable | Frame Clock, Serial Input Port 1 (LRCLK_IN1)/Multipurpose, General-Purpose Input/Output (MP11). This pin is bidirectional, with the direction depending on whether the Serial Input Port 1 is set up as a master or slave. Leave this pin disconnected when not in use. |
| 62 | SDATA_IN1 | Configurable | Serial Data Input Port 1 (Channels 16 to Channel 31). Capable of 2-channel, 4-channel, 8-channel, or 16-channel mode. Leave this pin disconnected when not in use. |
| 63 | THD_M | None | Thermal Diode Negative (-) Input. Connect this pin to the D- pin of an external temperature sensor IC. Leave this pin disconnected when not in use. |
| 64 | THD_P | None | Thermal Diode Positive (+) Input. Connect this pin to the D+ pin of an external temperature sensor IC. Leave this pin disconnected when not in use. |
| 65 | BCLK_IN2 | Configurable | Bit Clock, Serial Input Port 2. This pin is bidirectional, with the direction depending on whether the Serial Input Port 2 is set up as a master or slave. Leave this pin disconnected when not in use. |
| 66 | LRCLK_IN2/ MP12 | Configurable | Frame Clock, Input Serial Port 2 (LRCLK_IN2)/Multipurpose, General-Purpose Input/Output (MP12). This pin is bidirectional, with the direction depending on whether Serial Input Port 2 is set up as a master or slave. Leave this pin disconnected when not in use. |
| 67 | SDATA_IN2 | Configurable | Serial Data Input Port 2 (Channel 32 to Channel 39). Capable of 2-channel, 4-channel, 8-channel, or flexible TDM mode. Leave this pin disconnected when not in use. |
| 68 | BCLK_IN3 | Configurable | Bit Clock, Input Serial Port 3. This pin is bidirectional, with the direction depending on whether Input Serial Port 3 is set up as a master or slave. Leave this pin disconnected when not in use. |
| 69 | LRCLK_IN3/ MP13 | Configurable | Frame Clock, Serial Input Port 3 (LRCLK_IN3)/Multipurpose, General-Purpose Input/Output (MP13). This pin is bidirectional, with the direction depending on whether Serial Input Port 3 is set up as a master or slave. Leave this pin disconnected when not in use. |
| 70 | SDATA_IN3 | Configurable | Serial Data Input Port 3 (Channel 40 to Channel 47). Capable of 2-channel, 4-channel, 8-channel, or flexible TDM mode. Leave this pin disconnected when not in use. |
| 71 | DVDD | None | Digital Supply. Must be $1.2\text{ V} \pm 5\%$. This pin can be supplied externally or by using the internal regulator and external pass transistor. Bypass with decoupling capacitors to Pin 72 (DGND). |
| 72 | DGND | None | Digital and I/O Ground Reference. Tie all DGND, AGND, and PGND pins directly together in a common ground plane. |
| EP | Exposed Pad | None | The exposed pad must be grounded by soldering it to a copper square of equivalent size on the PCB. Identical copper squares must exist on all layers of the board, connected by vias, and they must be connected to a dedicated copper ground layer within the PCB. For more detailed information, see Figure 83 and Figure 84. |

THEORY OF OPERATION
SYSTEM BLOCK DIAGRAM

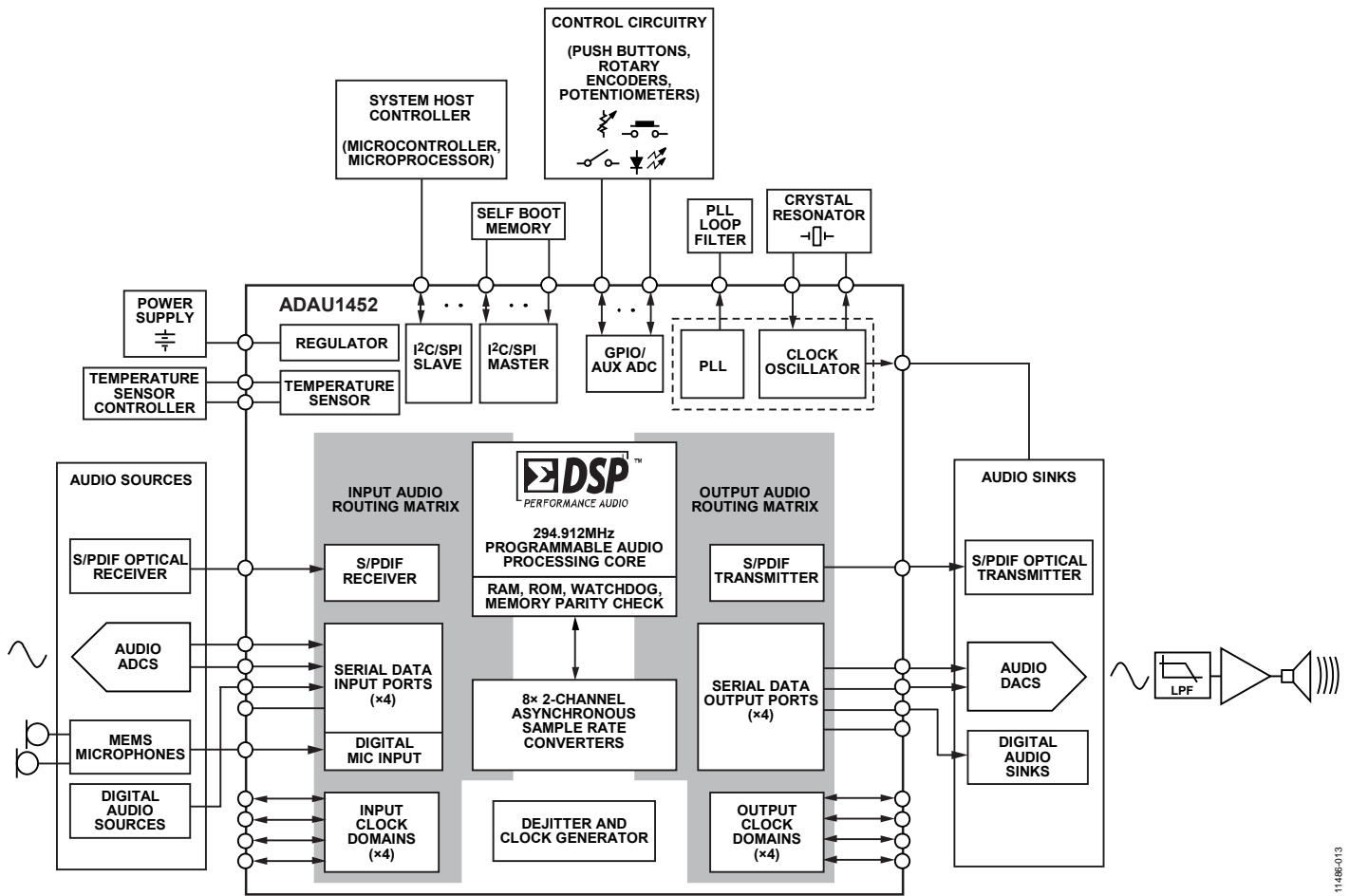


Figure 12. System Block Diagram with Example Connections to External Components

OVERVIEW

The ADAU1452 is an enhanced audio processor with 48 channels of input and output. It includes options for the hardware routing of audio signals between the various inputs, outputs, SigmaDSP core, and integrated sample rate converters. The SigmaDSP core features full 32-bit processing (that is, 64-bit processing in double precision mode) with an 80-bit arithmetic logic unit (ALU). By using a quadruple multiply accumulator (MAC) data path, the ADAU1452 can execute more than 1.2 billion MAC operations per second, which allows for processing power that far exceeds the predecessors in the SigmaDSP family of products. The powerful DSP core can process over 3000 double-precision biquad filters or 24,000 FIR filter taps per sample at the standard 48 kHz audio sampling rate. Other features, including synchronous parameter loading for ensuring filter stability and 100% code efficiency with the SigmaStudio tools, reduce complexity in audio system development. The SigmaStudio library of audio processing algorithms allows system designers to compensate for

real-world limitations of speakers, amplifiers, and listening environments, through speaker equalization, multiband compression, limiting, and third-party branded algorithms.

The input audio routing matrix and output audio routing matrix allow the user to multiplex inputs from multiple sources that are running at various sample rates to or from the SigmaDSP core, and then to pass them on to the desired hardware outputs. This drastically reduces the complexity of signal routing and clocking issues in the audio system. The audio subsystem includes up to eight stereo asynchronous sample rate converters (ASRCs), depending on the device model; Sony/Philips Digital Interconnect Format (S/PDIF) input and output; and serial (I²S) and time division multiplexing (TDM) I/Os. Any of these inputs can be routed to the SigmaDSP core or to any of the ASRCs. Similarly, the output signals can be taken from the SigmaDSP core, any of the ASRC outputs, the serial inputs, the PDM microphones, or the S/PDIF receiver. This routing scheme, which can be modified at any time using control registers, allows for maximum system flexibility without requiring hardware design changes.

Two serial input ports and two serial output ports can operate as pairs in a special flexible TDM mode, allowing the user to independently assign byte-specific locations to audio streams at varying bit depths. This mode ensures compatibility with codecs that use similar flexible TDM streams.

The DSP core is optimized for audio processing, and it can process audio at sample rates of up to 192 kHz. The program and parameter/data RAMs can be loaded with a custom audio processing signal flow built with the SigmaStudio graphical programming software from Analog Devices, Inc. The values that are stored in the parameter RAM can control individual signal processing blocks, such as IIR and FIR equalization filters, dynamics processors, audio delays, and mixer levels. A software safeload feature allows for transparent parameter updates and prevents clicks on the output signals.

Reliability features, such as memory parity checking and a program counter watchdog, help ensure that the system can detect and recover from any errors related to memory corruption.

S/PDIF signals can be routed through an ASRC for processing in the DSP or can be sent directly to output on the multipurpose pins (MPx) for recovery of the embedded audio signal. Other components of the embedded signal, including status and user bits, are not lost and can be output on the MPx pins as well. The user can also independently program the nonaudio data that is embedded in the output signal of the S/PDIF transmitter.

The fourteen MPx pins are available for providing a simple user interface without the need for an external microcontroller. These multipurpose pins are available to input external control signals and output flags or controls to other devices in the system. As inputs, the MPx pins can be connected to push buttons, switches, rotary encoders, or other external control circuitry to control the internal signal processing program. When configured as outputs, these pins can be used to drive LEDs (with a buffer), output flags to a microcontroller, control other ICs, or connect to other external circuitry in an application. In addition to the multipurpose pins, six dedicated input pins (AUXADC5 to AUXADC0) are connected to an auxiliary ADC for use with analog controls such as potentiometers or system voltages.

The SigmaStudio software programs and controls the device through the control port. Along with designing and tuning a signal flow, the software can configure all of the DSP registers in real time and download a new program and parameters into the external self boot EEPROM. The easy to use SigmaStudio graphical interface allows anyone with audio processing knowledge to easily design a DSP signal flow and port it to a target application without

the need for writing line level code. At the same time, the software provides enough flexibility and programmability to allow an experienced DSP programmer to have in-depth control of the design. In SigmaStudio, the user can add signal processing cells from the library by dragging and dropping cells, connect them together in a flow, compile the design, and load the program and parameter files into memory through the control port. The complicated tasks of linking, compiling, and downloading the project are all handled automatically by the software.

Signal processing algorithms that are available in the provided libraries include the following:

- Single and double precision biquad filter
- Mono and multichannel dynamics processors with peak or rms detection
- Mixer and splitter
- Tone and noise generator
- Fixed and variable gain
- Loudness
- Delay
- Stereo enhancement
- Dynamic bass boost
- Noise and tone source
- Level detector
- MPx pin control and conditioning
- FFT and frequency domain processing algorithms

New processing algorithms are always being developed. Analog Devices also provides proprietary and third-party algorithms for applications such as matrix decoding, bass enhancement, and surround virtualizers.

Several power-saving mechanisms are available, including programmable pad strength for digital I/O pins and the ability to power down unused subsystems.

The device is fabricated on a single monolithic integrated circuit for operation over the -40°C to $+105^{\circ}\text{C}$ temperature range and housed in a 72-lead LFCSP package with an exposed pad to assist in heat dissipation.

The device can be controlled in one of two operational modes, as follows:

- The settings of the chip can be loaded and dynamically updated through the SPI/I²C port.
- The DSP can self boot from an external EEPROM in a system with no microcontroller.

The [ADAU1452WBCPZ](#) and the [ADAU1452WBCPZ-RL](#) models are qualified for use in automotive applications.

INITIALIZATION

Power-Up Sequence

The first step in the initialization sequence is to power up the device. First, apply voltage to the power pins. All power pins can be supplied simultaneously. If the power pins are not supplied simultaneously, then supply IOVDD first because the internal ESD protection diodes are referenced to the IOVDD voltage. AVDD, DVDD, and PVDD can be supplied at the same time as IOVDD or after, but they must not be supplied prior to IOVDD. The order in which AVDD, DVDD, and PVDD are supplied does not matter.

If the internal regulator is not used and DVDD is directly supplied, no special sequence is required when providing the proper voltages to AVDD, DVDD, and PVDD.

If the internal regulator is used, DVDD is generated by the regulator, in combination with an external pass transistor, after AVDD, IOVDD, and PVDD are supplied. See the Power Supplies section for more information.

Each power supply domain has its own power-on reset (POR) circuits (also known as power OK circuits) to ensure that the level shifters attached to each power domain can be initialized properly. AVDD and PVDD must reach their nominal level before the auxiliary ADC and PLL can be used, respectively.

However, the AVDD and PVDD supplies do not play a part in the rest of the power-up sequence. After AVDD power reaches its nominal threshold, the regulator becomes active and begins to charge up the DVDD supply. The DVDD also has a POR circuit to ensure that the level shifters are initialized during power-up.

The POR signals are combined into three global level shifter resets that properly initialize the signal crossings between each separate power domain and DVDD.

The digital circuits are kept in reset until the IOVDD to DVDD level shifter reset is released. At that point, the digital circuits exit reset.

If a crystal is in use, the crystal oscillator circuit must provide a stable master clock to the XTALIN/MCLK pin by the time the PVDD supply reaches its nominal level. The XTALIN/MCLK pin is restricted from passing into the PLL circuitry until the DVDD POR signal becomes active and the PVDD to DVDD level shifter is initialized.

When all four POR circuits signal that the power-on conditions are met, a reset synchronizer circuit releases the internal digital circuitry from reset, provided the following conditions are met:

- A valid MCLK signal is provided to the digital circuitry and the PLL.
- The RESET pin is high.

When the internal digital circuitry becomes active, the DSP core runs eight lines of initialization code stored in ROM, requiring eight cycles of the MCLK signal. For a 12.288 MHz MCLK input, this process takes 650 ns.

After the ROM program completes its execution, the PLL is ready to be configured using register writes to Register 0xF000 (PLL_CTRL0), Register 0xF001 (PLL_CTRL1), Register 0xF002 (PLL_CLK_SRC), and Register 0xF003 (PLL_ENABLE).

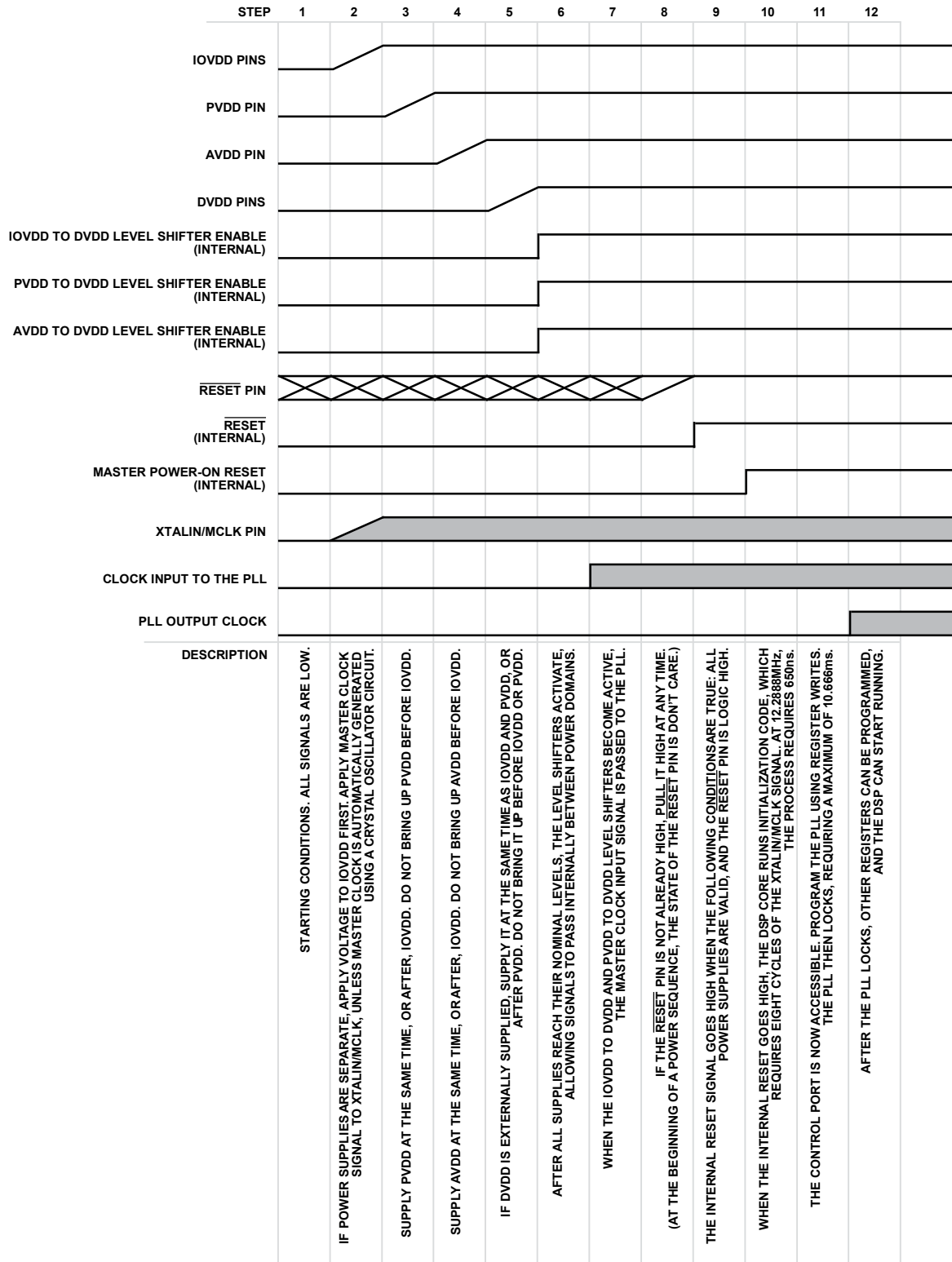
When the PLL is configured and enabled, the PLL starts to lock to the incoming master clock signal. The absolute maximum PLL lock time is $32 \times 1024 = 32,768$ clock cycles on the clock signal (after the input prescaler), which is fed to the input of the PLL. In a standard 48 kHz use case, the PLL input clock frequency after the prescaler is 3.072 MHz; therefore, the maximum PLL lock time is 10.666 ms.

Typically, the PLL locks much faster than 10.666 ms. In most systems, the PLL locks within about 3.5 ms. The PLL_LOCK register (Address 0xF004) can be polled via the control port until Bit 0 (PLL_LOCK) goes high, signifying that the PLL lock has completed successfully.

While the PLL is attempting to lock to the input clock, the I²C slave and SPI slave control ports are inactive; therefore, no other registers are accessible over the control port. While the PLL is attempting to lock, all attempts to write to the control port fail.

Figure 13 shows an example power-up sequence with all relevant signals labeled. If possible, apply the required voltage to all four power supply domains (IOVDD, AVDD, PVDD, and DVDD) simultaneously. If the power supplies are separate, IOVDD, which is the reference for the ESD protection diodes that are situated

inside the input and output pins, must be applied first to avoid stressing these diodes. PVDD, AVDD, and DVDD can then be supplied in any order (see the System Initialization Sequence section for more information). Note that the gray areas in this figure represent clock signals.



1148B-018

Figure 13. Power Sequencing and POR Timing Diagram for a System with Separate Power Supplies

System Initialization Sequence

Before the IC can process the audio in the DSP, the following initialization sequence must be completed.

1. If possible, apply the required voltage to all four power supply domains (IOVDD, AVDD, PVDD, and DVDD) simultaneously. If simultaneous application is not possible, supply IOVDD first to prevent damage or reduced operating lifetime. If using the on-board regulator, AVDD and PVDD can be supplied in any order, and DVDD is then generated automatically. If not using the on-board regulator, AVDD, PVDD, and DVDD can be supplied in any order following IOVDD.
2. Start providing a master clock signal to the XTALIN/MCLK pin, or, if using the crystal oscillator, let the crystal oscillator start generating a master clock signal. The master clock signal must be valid when the DVDD supply stabilizes.
3. If the SELFB00T pin is pulled high, a self boot sequence initiates on the master control port. Wait until the self boot operation is complete.
4. If SPI slave control mode is desired, toggle the SS/ADDR0 pin three times. Ensure that each toggle lasts at least the duration of one cycle of the master clock being input to the

- XTALIN/MCLK pin. When the SS/ADDR0 line rises for the third time, the slave control port is then in SPI mode.
5. Execute the register and memory write sequence that is required to configure the device in the proper operating mode.

Table 21 contains an example series of register writes used to configure the system at startup. The contents of the data column may vary depending on the desired system configuration. The configuration that is listed in Table 21 represents the default initialization sequence for project files generated in SigmaStudio.

Recommended Program/Parameter Loading Procedure

When writing large amounts of data to the program or parameter RAM in direct write mode (such as when downloading the initial contents of the RAMs from an external memory), use the hibernate register (Address 0xF400) to disable the processor core, thus preventing unpleasant noises from appearing at the audio output. When small amounts of data are transmitted during real-time operation of the DSP (such as when updating individual parameters), the software safeload mechanism can be used (see the Software Safeload section).

Table 21. Example System Initialization Register Write Sequence¹

| Address | Data | Register/Memory | Description |
|---------|-------------------------------|------------------|--|
| N/A | N/A | N/A | Toggle SS/ADDR0 three times to enable SPI slave mode, if necessary. |
| 0xF890 | 0x00, 0x00 | SOFT_RESET | Enter soft reset. |
| 0xF890 | 0x00, 0x01 | SOFT_RESET | Exit soft reset. |
| 0xF000 | 0x00, 0x60 | PLL_CTRL0 | Set feedback divider to 96 (this is the default power-on setting). |
| 0xF001 | 0x00, 0x02 | PLL_CTRL1 | Set PLL input clock divider to 4. |
| 0xF002 | 0x00, 0x01 | PLL_CLK_SRC | Set clock source to PLL clock. |
| 0xF005 | 0x00, 0x05 | MCLK_OUT | Enable MCLK output (12.288 MHz). |
| 0xF003 | 0x00, 0x01 | PLL_ENABLE | Enable PLL. |
| N/A | N/A | N/A | Wait for PLL lock (see the Power-Up Sequence section); the maximum PLL lock time is 10.666 ms. |
| 0xF050 | 0x4F, 0xFF | POWER_ENABLE0 | Enable power for all major systems except Clock Generator 3 (Clock Generator 3 is rarely used in most systems). |
| 0xF051 | 0x00, 0x00 | POWER_ENABLE1 | Disable power for subsystems like PDM microphones, S/PDIF, and the ADC if they are not being used in the system. |
| 0xC000 | Data generated by SigmaStudio | Program RAM data | Download the entire program RAM contents using a block write (data provided by SigmaStudio compiler). |
| 0x0000 | Data generated by SigmaStudio | DM0 RAM data | Download Data Memory DM0 using a block write (data provided by SigmaStudio compiler). |
| 0x6000 | Data generated by SigmaStudio | DM1 RAM data | Download Data Memory DM1 using a block write (data provided by SigmaStudio compiler); the start address of DM1 may vary, depending on the SigmaStudio compilation. |
| 0xF404 | 0x00, 0x00 | START_ADDRESS | Set program start address as defined by the SigmaStudio compiler. |
| 0xF401 | 0x00, 0x02 | START_PULSE | Set DSP core start pulse to internally generated pulse. |
| N/A | N/A | N/A | Configure any other registers that require nondefault values. |
| 0xF402 | 0x00, 0x00 | START_CORE | Stop the core. |
| 0xF402 | 0x00, 0x01 | START_CORE | Start the core. |
| N/A | N/A | N/A | Wait 50 μs for initialization program to execute. |

¹ N/A means not applicable.

MASTER CLOCK, PLL, AND CLOCK GENERATORS

Clocking Overview

Connect the clock source directly to the XTALIN/MCLK pin to externally supply the master clock. Alternatively, use the internal clock oscillator to drive an external crystal.

Using the Oscillator

The ADAU1452 can use an on-board oscillator to generate its master clock. However, to complete the oscillator circuit, an external crystal must be attached. The on-board oscillator is designed to work with a crystal that is tuned to resonate at a frequency of the nominal system clock divided by 24. For a normal system, where the nominal system clock is 294.912 MHz, this frequency is 12.288 MHz.

The fundamental frequency of the crystal can be up to 30 MHz. Practically speaking, in most systems the fundamental frequency of the crystal should be in a range from 3.072 MHz to 24.576 MHz.

For the external crystal in the circuit, use an AT-cut parallel resonance device operating at its fundamental frequency. Do not use ceramic resonators, due to their poor jitter performance. Quartz crystals are ideal for audio applications. Figure 14 shows the crystal oscillator circuit that is recommended for proper operation.

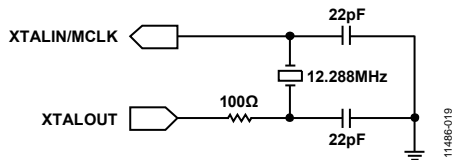


Figure 14. Crystal Resonator Circuit

The 100 Ω damping resistor on XTALOUT provides the oscillator with a voltage swing of approximately 3.1 V at the XTALIN/MCLK pin. The optimal crystal shunt capacitance is 7 pF. Its optimal load capacitance, specified by the manufacturer, should be about 20 pF, although the circuit supports values of up to 25 pF. Ensure that the equivalent series resistance is as small as possible. The necessary values of the two load capacitors in the circuit can be calculated from the crystal load capacitance, using the following equation:

$$C_L = \frac{C1 \times C2}{C1 + C2} + C_{STRAY}$$

where:

$C1$ and $C2$ are the load capacitors.

C_{STRAY} is the stray capacitance in the circuit. C_{STRAY} is usually assumed to be approximately 2 pF to 5 pF, but it varies depending on the PCB design.

Short trace lengths in the oscillator circuit decrease stray capacitance, thereby increasing the loop gain of the circuit and helping to avoid crystal start-up problems. Therefore, place the crystal as near to the XTALOUT pin as possible, and on the same side of the PCB.

On the EVAL-ADAU1452MINIZ evaluation board, the $C1$ and $C2$ load capacitors are 22 pF.

Do not use XTALOUT to directly drive the crystal signal to another IC. This signal is an analog sine wave with low drive capability and, therefore, is not appropriate to drive an external digital input. A separate pin, CLKOUT, is provided for this purpose. The CLKOUT pin is set up using the MCLK_OUT register (Address 0xF005). For a more detailed explanation of CLKOUT, refer to the Master Clock Output section or the register map description of the MCLK_OUT register (see the CLKOUT Control Register section).

If a clock signal is provided from elsewhere in the system directly to the XTALIN/MCLK pin, the crystal resonator circuit is not necessary, and the XTALOUT pin can be left disconnected.

Setting the Master Clock and PLL Mode

An integer PLL is available to generate the core system clock from the master clock input signal. The PLL generates the nominal 294.912 MHz core system clock to run the DSP core. This nominal core clock frequency can be used for a variety of audio sample rates, thanks to the flexible clock generator circuitry. An integer prescaler takes the clock signal from the MCLK pin and divides its frequency by 1, 2, 4, or 8 to meet the appropriate frequency range requirements for the PLL itself. The nominal input frequency to the PLL is 3.072 MHz. For systems with an 11.2896 MHz input master clock, the input to the PLL is 2.8224 MHz.

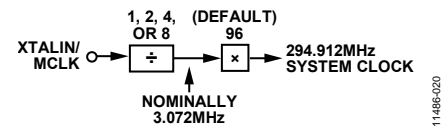


Figure 15. PLL Functional Block Diagram

The master clock input signal ranges in frequency from 2.375 MHz to 36 MHz. For systems that are intended to operate at a 48 kHz, 96 kHz, or 192 kHz audio sample rate, the typical master clock input frequencies are 3.072 MHz, 6.144 MHz, 12.288 MHz, and 24.576 MHz. Note that the flexibility of the PLL allows for a large range of other clock frequencies, as well.

The nominal (and maximum) output frequency of the PLL is 294.912 MHz.

The PLL is configured by setting Registers 0xF000 (PLL_CTRL0), Register 0xF001 (PLL_CTRL1), and Register 0xF002 (PLL_CLK_SRC). After these registers are modified, set Register 0xF003, Bit 0 (PLL_ENABLE), forcing the PLL to reset itself and attempt to relock to the incoming clock signal. Typically, the PLL locks within 3.5 ms. When the PLL locks to an input clock and creates a stable output clock, a lock flag is set in Register 0xF004, Bit 0 (PLL_LOCK).

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 for more DSP program instructions to be executed for each audio frame. Alternatively, setting the PLL to generate a lower frequency system clock allows for fewer instructions to be executed and also lowers overall power consumption of the device.

Table 22 shows several example MCLK frequencies and the corresponding PLL settings that allow for 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 22. Optimal Predivider and Feedback Divider Settings for Varying Input MCLK Frequencies

| Input MCLK Frequency (MHz) | Pre-divider Setting | PLL Input Clock (MHz) | Feedback Divider Setting | System Clock (MHz) |
|----------------------------|---------------------|-----------------------|--------------------------|--------------------|
| 2.8224 | 1 | 2.8224 | 104 | 293.5296 |
| 3 | 1 | 3 | 98 | 294 |
| 3.072 | 1 | 3.072 | 96 | 294.912 |
| 3.5 | 1 | 3.5 | 84 | 294 |
| 4 | 1 | 4 | 73 | 292 |
| 4.5 | 1 | 4.5 | 65 | 292.5 |
| 5 | 2 | 2.5 | 117 | 292.5 |
| 5.5 | 2 | 2.75 | 107 | 294.25 |
| 5.6448 | 2 | 2.8224 | 104 | 293.5296 |
| 6 | 2 | 3 | 98 | 294 |
| 6.144 | 2 | 3.072 | 96 | 294.912 |
| 6.5 | 2 | 3.25 | 90 | 292.5 |
| 7 | 2 | 3.5 | 84 | 294 |
| 7.5 | 2 | 3.75 | 78 | 292.5 |
| 8 | 2 | 4 | 73 | 292 |
| 8.5 | 2 | 4.25 | 69 | 293.25 |
| 9 | 2 | 4.5 | 65 | 292.5 |
| 9.5 | 4 | 2.375 | 124 | 294.5 |
| 10 | 4 | 2.5 | 117 | 292.5 |
| 10.5 | 4 | 2.625 | 112 | 294 |
| 11 | 4 | 2.75 | 107 | 294.25 |
| 11.2896 | 4 | 2.8224 | 104 | 293.5296 |
| 11.5 | 4 | 2.875 | 102 | 293.25 |

| Input MCLK Frequency (MHz) | Pre-Divider Setting | PLL Input Clock (MHz) | Feedback Divider Setting | System Clock (MHz) |
|----------------------------|---------------------|-----------------------|--------------------------|--------------------|
| 12 | 4 | 3 | 98 | 294 |
| 12.288 | 4 | 3.072 | 96 | 294.912 |
| 12.5 | 4 | 3.125 | 94 | 293.75 |
| 13 | 4 | 3.25 | 90 | 292.5 |
| 13.5 | 4 | 3.375 | 87 | 293.625 |
| 14 | 4 | 3.5 | 84 | 294 |
| 14.5 | 4 | 3.625 | 81 | 293.625 |
| 15 | 4 | 3.75 | 78 | 292.5 |
| 15.5 | 4 | 3.875 | 76 | 294.5 |
| 16 | 4 | 4 | 73 | 292 |
| 16.5 | 4 | 4.125 | 71 | 292.875 |
| 17 | 4 | 4.25 | 69 | 293.25 |
| 17.5 | 4 | 4.375 | 67 | 293.125 |
| 18 | 4 | 4.5 | 65 | 292.5 |
| 18.5 | 8 | 2.3125 | 127 | 293.6875 |
| 19 | 8 | 2.375 | 124 | 294.5 |
| 19.5 | 8 | 2.4375 | 120 | 292.5 |
| 20 | 8 | 2.5 | 117 | 292.5 |
| 20.5 | 8 | 2.5625 | 115 | 294.6875 |
| 21 | 8 | 2.625 | 112 | 294 |
| 21.5 | 8 | 2.6875 | 109 | 292.9375 |
| 22 | 8 | 2.75 | 107 | 294.25 |
| 22.5 | 8 | 2.8125 | 104 | 292.5 |
| 22.5792 | 8 | 2.8224 | 104 | 293.5296 |
| 23 | 8 | 2.875 | 102 | 293.25 |
| 23.5 | 8 | 2.9375 | 100 | 293.75 |
| 24 | 8 | 3 | 98 | 294 |
| 24.5 | 8 | 3.0625 | 96 | 294 |
| 24.576 | 8 | 3.072 | 96 | 294.912 |
| 25 | 8 | 3.125 | 94 | 293.75 |

Relationship Between System Clock and Instructions per Sample

The DSP core can execute only a limited number of instructions within the span of each audio sample. The number of instructions that can be executed is a function of the system clock and the DSP core sample rate. The core sample rate is set by Register 0xF401 (START_PULSE), Bits[4:0] (START_PULSE).

The number of instructions that can be executed per sample is equal to the system clock frequency divided by the DSP core sample rate. However, the program RAM size is 8192 words; therefore, in cases where the maximum instructions per sample exceeds 8192, subroutines and loops must be utilized to make use of all available instructions (see Table 23).

Table 23. Maximum Instructions Per Sample, Depending on System Clock and DSP Core Sample Rate

| System Clock (MHz) | DSP Core Sample Rate (kHz) | Maximum Instructions per Sample |
|--------------------|----------------------------|---------------------------------|
| 294.912 | 8 | 36,864 ¹ |
| 294.912 | 12 | 24,576 ¹ |
| 294.912 | 16 | 18,432 ¹ |
| 294.912 | 24 | 12,288 ¹ |
| 294.912 | 32 | 9216 ¹ |
| 294.912 | 48 | 6144 |
| 294.912 | 64 | 4608 |
| 294.912 | 96 | 3072 |
| 294.912 | 128 | 2304 |
| 294.912 | 192 | 1536 |
| 293.5296 | 11.025 | 26,624 ¹ |
| 293.5296 | 22.05 | 13,312 ¹ |
| 293.5296 | 44.1 | 6656 |
| 293.5296 | 88.2 | 3328 |
| 293.5296 | 176.4 | 1664 |

¹ The instructions per sample in these cases exceed the program memory size of 8192 words; therefore, to utilize the full number of instructions, subroutines or branches are required in the SigmaStudio program.

PLL Loop Filter

An external PLL loop filter is required to help the PLL maintain stability and to limit the amount of ripple appearing on the phase detector output of the PLL. For a nominal 3.072 MHz PLL input and a 294.912 MHz system clock output, the recommended filter configuration is shown in Figure 16. This filter works for the full frequency range of the PLL.

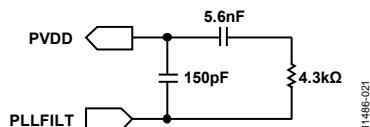


Figure 16. PLL Loop Filter

Because the center frequency and bandwidth of the loop filter is determined by the values of the included components, use high accuracy (low tolerance) components. Components that are valued within 10% of the recommended component values and with a 15% or lower tolerance are suitable for use in the loop filter circuit.

The voltage on the PLLFILT pin, which is internally generated, is typically between 1.65 V and 2.10 V.

Clock Generators

Three clock generators are available to generate audio clocks for the serial ports, DSP, ASRCs, and other audio related functional blocks in the system. Each clock generator can be configured to generate a base frequency and several fractions or multiples of that base frequency, creating a total of 15 clock domains available for use in the system. Each of the 15 clock domains can create the appropriate LRCLK (frame clock) and BCLK (bit clock) signals for the serial ports. Five BCLK signals are generated at frequencies of 32 BCLK/sample, 64 BCLK/sample, 128 BCLK/sample, 256 BCLK/sample, and 512 BCLK/sample to deal with TDM data. Thus, with a single master clock input frequency, 15 different frame clock frequencies and 75 different bit clock frequencies can be generated for use in the system.

The nominal output of each clock generator is determined by the following formula:

$$\text{Output_Frequency} = (\text{Input_Frequency} \times N) / (1024 \times M)$$

where:

Input_Frequency is the PLL output (nominally 294.912 MHz).

Output_Frequency is the frame clock output frequency.

N and *M* are integers that are configured by writing to the clock generator configuration registers.

In addition to the nominal output, four additional output signals are generated at double, quadruple, half, and a quarter of the frequency of the nominal output frequency.

For Clock Generator 1 and Clock Generator 2, the integer numerator (*N*) and the integer denominator (*M*) are each nine bits long. For Clock Generator 3, *N* and *M* are each 16 bits long, allowing for a higher precision when generating arbitrary clock frequencies.

Figure 17 shows a basic block diagram of the PLL and clock generators. Each division operator symbolizes that the frequency of the clock is divided when passing through that block. Each multiplication operator symbolizes that the frequency of the clock is multiplied when passing through that block.

Figure 18 shows an example where the master clock input has a frequency of 12.288 MHz, and the default settings are used for the PLL predivider, feedback divider, and Clock Generator 1 and Clock Generator 2. The resulting system clock is $12.288 \text{ MHz} \div 4 \times 96 = 294.912 \text{ MHz}$. The base output of Clock Generator 1 is $294.912 \text{ MHz} \div 1024 \times 1 \div 6 = 48 \text{ kHz}$. The base output of Clock Generator 2 is $294.912 \text{ MHz} \div 1024 \times 1 \div 9 = 32 \text{ kHz}$. In this

example, Clock Generator 3 is configured with $N = 49$ and $M = 320$; therefore, the resulting base output of Clock Generator 3 is $294.912 \text{ MHz} \div 1024 \times 49 \div 320 = 44.1 \text{ kHz}$.

Figure 19 shows an example where the master clock input has a frequency of 11.2896 MHz, and the default settings are used for the PLL predivider, feedback divider, and Clock Generator 1 and Clock Generator 2. The resulting system clock is $11.2896 \text{ MHz} \div 4 \times 96 = 270.9504 \text{ MHz}$. The base output of Clock Generator 1 is $270.9504 \text{ MHz} \div 1024 \times 1 \div 6 = 44.1 \text{ kHz}$. The base output of Clock Generator 2 is $270.9504 \text{ MHz} \div 1024 \times 1 \div 9 = 29.4 \text{ kHz}$. In this example, Clock Generator 3 is configured with $N = 80$ and $M = 441$; therefore, the resulting base output of Clock Generator 3 is $270.9504 \text{ MHz} \div 1024 \times 80 \div 441 = 48 \text{ kHz}$.

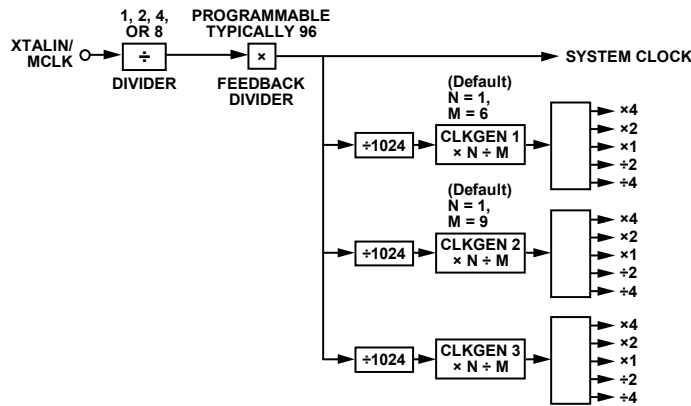


Figure 17. PLL and Clock Generators Block Diagram

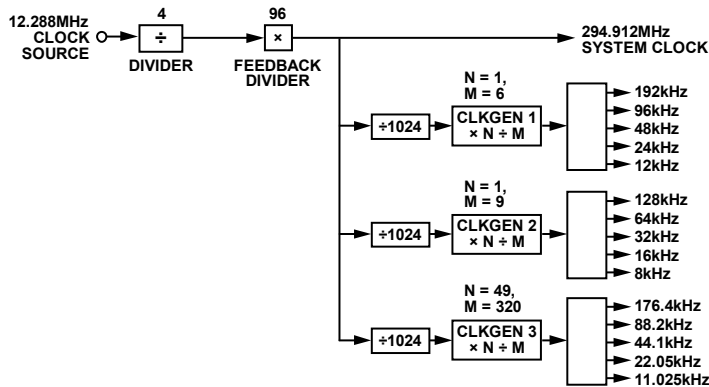


Figure 18. PLL and Audio Clock Generators with Default Settings and Resulting Clock Frequencies Labeled, XTALIN/MCLK = 12.288 MHz

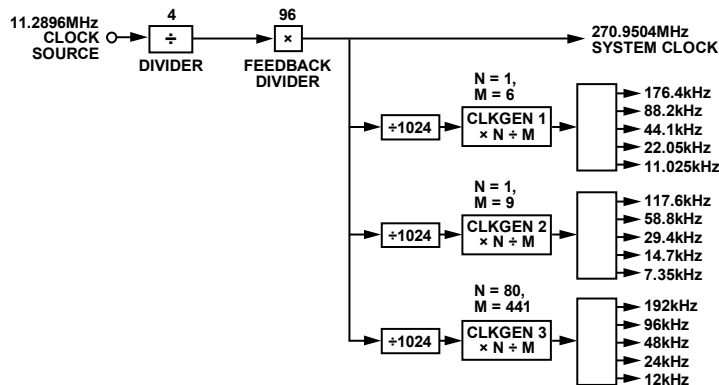


Figure 19. PLL and Audio Clock Generators with Default Settings and Resulting Clock Frequencies Labeled, XTALIN/MCLK = 11.2896 MHz

Master Clock Output

The master clock output pin (CLKOUT) is useful in cases where a master clock must be fed to other ICs in the system, such as audio codecs. The master clock output frequency is determined by the setting of the MCLK_OUT register (Address 0xF005). Four frequencies are possible: 1x, 2x, 4x, or 8x the frequency of the predivider output. The predivider output $\times 1$ generates a 3.072 MHz output for a nominal system clock of 294.912 MHz. The predivider output $\times 2$ generates a 6.144 MHz output for a nominal system clock of 294.912 MHz. The predivider output $\times 4$ generates a 12.288 MHz output for a nominal system clock of 294.912 MHz. The predivider output $\times 8$ generates a 24.576 MHz output for a nominal system clock of 294.912 MHz.

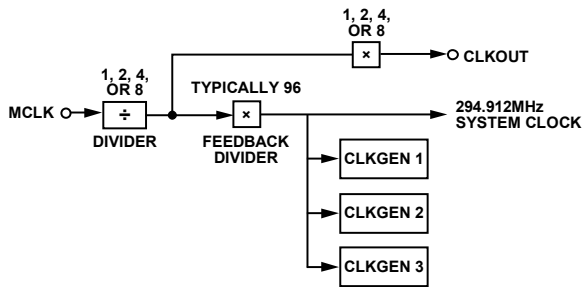


Figure 20. Clock Output Generator

The CLKOUT pin can drive more than one external slave IC, provided that the drive strength is sufficient to drive the traces and external receiver circuitry. The ability to drive external ICs varies greatly, depending on the application and the characteristics of the PCB and the slave ICs. The drive strength and slew rate of the CLKOUT pin is configurable in the CLKOUT_PIN register (Address 0xF7A3); thus, its performance can be tuned to match the specific application. The CLKOUT pin is not designed to drive long cables or other high impedance transmission lines. Use the CLKOUT pin only to drive signals to other integrated circuits on the same PCB. When changing the settings for the predivider, disable and then reenable the PLL using Register 0xF003 (PLL_ENABLE), allowing the frequency of the CLKOUT signal to update.

Dejitter Circuitry

To account for jitter between ICs in the system and to safely handle interfacing between internal and external clocks, dejitter circuits are included to guarantee that jitter related clocking errors are avoided. The dejitter circuitry is automated and does not require interaction or control from the user.

Master Clock, PLL, and Clock Generators Registers

An overview of the registers related to the master clock, PLL, and clock generators is listed in Table 24. For a more detailed description, see the PLL Configuration Registers section and the Clock Generator Registers section.

Table 24. Master Clock, PLL, and Clock Generator Registers

| Address | Register | Description |
|---------|---------------|--|
| 0xF000 | PLL_CTRL0 | PLL feedback divider |
| 0xF001 | PLL_CTRL1 | PLL prescale divider |
| 0xF002 | PLL_CLK_SRC | PLL clock source |
| 0xF003 | PLL_ENABLE | PLL enable |
| 0xF004 | PLL_LOCK | PLL lock |
| 0xF005 | MCLK_OUT | CLKOUT control |
| 0xF006 | PLL_WATCHDOG | Analog PLL watchdog control |
| 0xF020 | CLK_GEN1_M | Denominator (M) for Clock Generator 1 |
| 0xF021 | CLK_GEN1_N | Numerator (N) for Clock Generator 1 |
| 0xF022 | CLK_GEN2_M | Denominator (M) for Clock Generator 2 |
| 0xF023 | CLK_GEN2_N | Numerator (N) for Clock Generator 2 |
| 0xF024 | CLK_GEN3_M | Denominator (M) for Clock Generator 3 |
| 0xF025 | CLK_GEN3_N | Numerator (N) for Clock Generator 3 |
| 0xF026 | CLK_GEN3_SRC | Input reference for Clock Generator 3 |
| 0xF027 | CLK_GEN3_LOCK | Lock bit for Clock Generator 3 input reference |

POWER SUPPLIES, VOLTAGE REGULATOR, AND HARDWARE RESET

Power Supplies

The ADAU1452 is supplied by four power supplies: IOVDD, DVDD, AVDD, and PVDD.

- IOVDD (input/output supply) sets the reference voltage for all digital input and output pins. It can be any value ranging from 1.8 V – 10% to 3.3 V + 10%. To use the I²C/SPI control ports or any of the digital input or output pins, the IOVDD supply must be present.
- DVDD (digital supply) powers the DSP core and supporting digital logic circuitry. It must be 1.2 V \pm 5%.
- AVDD (analog supply) powers the analog auxiliary ADC circuitry. It must be supplied even if the auxiliary ADCs are not in use.
- PVDD (PLL supply) powers the PLL and acts as a reference for the voltage controlled oscillator (VCO). It must be supplied even if the PLL is not in use.

Table 25. Power Supply Details

| Supply | Voltage | Externally Supplied? | Description |
|----------------------|----------------------------|----------------------|---|
| IOVDD (Input/Output) | 1.8 V – 10% to 3.3 V + 10% | Yes | Can be derived from IOVDD using an internal LDO regulator |
| DVDD (Digital) | 1.2 V \pm 5% | Optional | |
| AVDD (Analog) | 3.3 V \pm 10% | Yes | |
| PVDD (PLL) | 3.3 V \pm 10% | Yes | |

Voltage Regulator

The ADAU1452 includes a linear regulator that can generate the 1.2 V supply required by the DSP core and other internal digital circuitry from an external supply. Source the linear regulator from the I/O supply (IOVDD), which can range from 1.8 V – 10% to 3.3 V + 10%. A simplified block diagram of the internal structure of the regulator is shown in Figure 22.

For proper operation, the linear regulator requires several external components. A PNP bipolar junction transistor acts as an external pass device to bring the higher IOVDD voltage down to the lower DVDD voltage, thus externally dissipating the power of the IC package. Ensure that the current gain of the transistor (β) is 200 or greater and the transistor is able to dissipate at least 1 W in the worst case. Place a 1 k Ω resistor between the transistor emitter and base to help stabilize the regulator for varying loads. This resistor placement also guarantees that current is always flowing into the VDRIVE pin, even for minimal regulator loads. Figure 21 shows the connection of the external components.

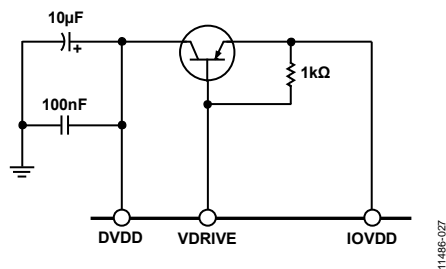


Figure 21. External Components Required for Voltage Regulator Circuit

If an external supply is provided to DVDD, ground the VDRIVE pin. The regulator continues to draw a small amount of current (around 100 μ A) from the IOVDD supply. Do not use the regulator to provide a voltage supply to external ICs. There are no control registers associated with the regulator.

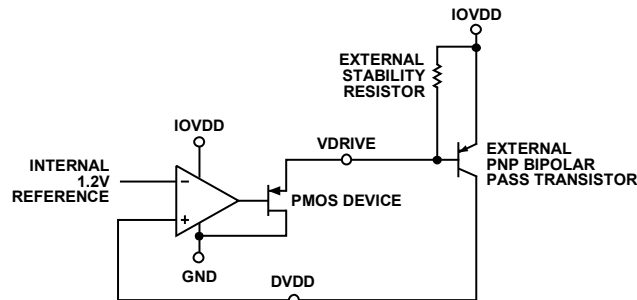


Figure 22. Simplified Block Diagram of Regulator Internal Structure, Including External Components

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 26. For a more detailed description, refer to the Power Reduction Registers section.

Table 26. 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, so 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 **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 **RESET** signal. For reliability purposes on the application level, place a weak pull-down resistor (in the range of several kΩ) on the **RESET** line to guarantee that the device is held in reset in the event that the reset supervisory circuitry fails.

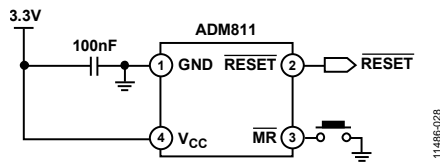


Figure 23. Example Manual Reset Generation Circuit

If the hardware reset function is not required in a system, pull the **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 **RESET** pin is permanently pulled high.

DSP Core Current Consumption

The DSP core draws varying amounts of current, depending on the processing load required by the program it is running. Figure 24 shows the relationship between program size and digital (DVDD) current draw. The minimum of 0 MIPS signifies the case where no program is running in the DSP core, and the maximum of 294 MIPS signifies that the DSP core is at full utilization, executing a typical audio processing program.

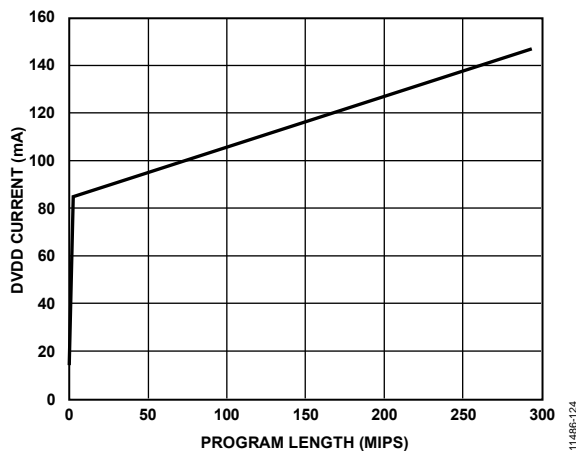


Figure 24. Typical DVDD Current Draw vs. Program MIPS at an Ambient Temperature of 25°C and a Sample Rate of 48 kHz

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. The 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. The temperature value cannot be read by the on-board auxiliary ADC.

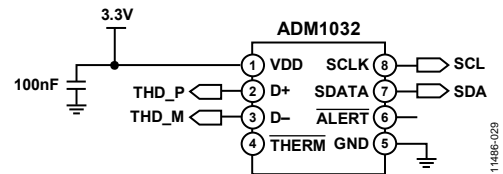


Figure 25. Example External Temperature Sensor Circuit

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.

Slave Control Port Overview

To program the DSP and configure the control registers, a slave port is available that can communicate using either the I²C or SPI protocols. A separate master communications port can be used to self boot the chip by reading from an external EEPROM, or to boot or control external ICs by addressing them directly using I²C or SPI. The slave communications port defaults to I²C mode; however, it can be put into SPI mode by toggling **SS** (**SS/ADDR0**), the slave select pin, low three times. Each toggle should last at least the duration of one clock period of the clock on **MCLK** (**XTALIN/MCLK**), the master clock input pin. Until the PLL locks, only the PLL configuration registers (Address 0xF000 to Address 0xF004) are accessible. For this reason, always write to the PLL registers first after the chip powers up. After the PLL locks, the remaining registers and the RAM become accessible. See the System Initialization Sequence section for more information.

The control port is capable of full read/write operation for all addressable registers. The **ADAU1452** must have a valid master clock to write to all registers, with the exception of Register 0xF000 to Register 0xF004. All addresses can be accessed in both single address mode and burst mode. The first byte (Byte 0) of a control port write contains the 7-bit chip address plus the R/W bit. The next two bytes (Byte 1 and Byte 2) together form the subaddress of the register location within the **ADAU1452** memory map. This subaddress must be two bytes long because the memory locations within the **ADAU1452** are directly addressable, and their sizes exceed the range of single byte addressing. All subsequent bytes (starting with Byte 3) contain the data, such as control port data, program data, or parameter data. The number of bytes per word depends on the type of data that is being written.

The **ADAU1452** has several mechanisms for updating signal processing parameters in real time without causing pops or clicks.

If large blocks of data must be downloaded, halt the output of the DSP core (using Register 0xF400, HIBERNATE), load new data, and then restart the device (using Register 0xF402, START_CORE). This process is typically performed during the booting sequence at startup or when loading a new program into RAM.

When updating a signal processing parameter while the DSP core is running, use the software safeload function to avoid a situation where a parameter is updated over the boundary of an audio frame, which can lead to an audio artifact such as a click or pop sound. For more information, see the Software Safeload section.

The slave control port pins are multifunctional, depending on the mode in which the device is operating. Table 27 describes these multiple functions.

Burst Mode Writing and Reading

Burst write and read modes are available for convenience when writing large amounts of data to contiguous registers. In these modes, the chip and memory addresses are written once, and then a large amount of data can follow uninterrupted. The subaddresses are automatically incremented at the word boundaries. This increment happens automatically after a single word write or read unless a stop condition is encountered (I²C mode) or the slave select is disabled and brought high (SPI mode). A burst write starts like a single word write, but, following the first data-word, the data-word for the next address can be written immediately without sending its 2-byte address. The control registers in the ADAU1452 are two bytes wide, and the memories are four bytes wide. The autoincrement feature knows the word length at each subaddress; therefore, it is not necessary to manually specify the subaddress for each address in a burst write.

Table 27. Control Port Pin Functions

| Pin Name | I ² C Slave Mode | SPI Slave Mode |
|------------|--|---|
| SS/ADDR0 | Address 0 (Bit 1 of the address word, input to the ADAU1452) | Slave select (input to the ADAU1452) |
| CCLK/SCL | Clock (input to the ADAU1452) | Clock (input to the ADAU1452) |
| MOSI/ADDR1 | Address 1 (Bit 2 of the address word, input to the ADAU1452) | Data; master out, slave in (input to the ADAU1452) |
| MISO/SDA | Data (bidirectional, open collector) | Data; master in, slave out (output from the ADAU1452) |

Table 28. Address Bit Sequence

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|-------|-------|-------|-------|-------|-----------------------------------|---------------------------------|-------|
| 0 | 1 | 1 | 1 | 0 | ADDR1 (set by the MOSI/ADDR1 pin) | ADDR0 (set by the SS/ADDR0 pin) | R/W |

Table 29. I²C Slave Addresses

| MOSI/ADDR1 | SS/ADDR0 | Read/Write ¹ | Slave Address (Eight Bits, Including R/W Bit) | Slave Address (Seven Bits, Excluding R/W Bit) |
|------------|----------|-------------------------|---|---|
| 0 | 0 | 0 | 0x70 | 0x38 |
| 0 | 0 | 1 | 0x71 | 0x38 |
| 0 | 1 | 0 | 0x72 | 0x39 |
| 0 | 1 | 1 | 0x73 | 0x39 |
| 1 | 0 | 0 | 0x74 | 0x3A |
| 1 | 0 | 1 | 0x75 | 0x3A |
| 1 | 1 | 0 | 0x76 | 0x3B |
| 1 | 1 | 1 | 0x77 | 0x3B |

¹ 0 = write, 1 = read.

The subaddresses are automatically incremented by one address, following each read or write of a data-word, regardless of whether there is a valid register or RAM word at that address.

I²C Slave Port

The ADAU1452 supports a 2-wire serial (I²C-compatible) micro-processor bus driving multiple peripherals. The maximum clock frequency on the I²C slave port is 400 kHz. Two pins, serial data (SDA) and serial clock (SCL), carry information between the ADAU1452 and the system I²C master controller. In I²C mode, the ADAU1452 is always a slave on the bus, meaning that it cannot initiate a data transfer. Each slave device is recognized by a unique address. The address bit sequence and the format of the read/write byte is shown in Table 28. The address resides in the first seven bits of the I²C write. The two address bits that follow can be set to assign the I²C slave address of the device, as follows: Bit 1 can be set by pulling the SS/ADDR0 pin either to IOVDD (by setting it to 1) or to GND (by setting it to 0); and Bit 2 can be set by pulling the MOSI/ADDR1 pin either to IOVDD (by setting it to 1) or to GND (by setting it to 0). The LSB of the address (the R/W bit) specifies either a read or write operation. Logic Level 1 corresponds to a read operation; Logic Level 0 corresponds to a write operation.

Table 28 describes the sequence of eight bits that define the I²C device address byte.

Table 29 describes the relationship between the state of the address pins (0 represents logic low and 1 represents logic high) and the I²C slave address. Ensure that the address pins (SS/ADDR0 and MOSI/ADDR1) are hardwired in the design. Do not allow them to change states while the device is operating.

Place a 2 kΩ pull-up resistor on each line connected to the SDA and SCL pins. Ensure that the voltage on these signal lines does not exceed IOVDD (1.8 V – 10% to 3.3 V + 10%).

Addressing

Initially, each device on the I²C bus is in an idle state and monitors the SDA and SCL lines for a start condition and the proper address. The I²C master initiates a data transfer by establishing a start condition, defined by a high-to-low transition on SDA while SCL remains high. This indicates that an address/data stream follows. All devices on the bus respond to the start condition and shift the next eight bits (the 7-bit address plus the R/W bit), MSB first. The device that recognizes the transmitted address responds by pulling the data line low during the ninth clock pulse. This ninth bit is known as an acknowledge bit. All other devices withdraw from the bus at this point and return to the idle condition.

The R/W bit determines the direction of the data. A Logic 0 on the LSB of the first byte means that the master writes information to the peripheral, whereas a Logic 1 means that the master reads information from the peripheral after writing the subaddress and repeating the start address. A data transfer occurs until a stop condition is encountered. A stop condition occurs when SDA transitions from low to high while SCL is held high.

Figure 26 shows the timing of an I²C single word write operation, Figure 27 shows the timing of an I²C burst mode write operation, and Figure 28 shows an I²C burst mode read operation.

Stop and start conditions can be detected at any stage during the data transfer. If these conditions are asserted out of sequence with normal read and write operations, the ADAU1452 slave I²C port immediately jumps to the idle condition. During a given SCL high period, issue only one start condition and one stop condition, or a single stop condition followed by a single start condition. If the user issues an invalid subaddress, the ADAU1452 does not issue an acknowledge and returns to the idle condition.

Note the following conditions:

- Do not issue an autoincrement (burst) write command that exceeds the highest subaddress in the memory.
- Do not issue an autoincrement (burst) write command that writes to subaddresses that are not defined in the Global RAM and Control Register Map section.

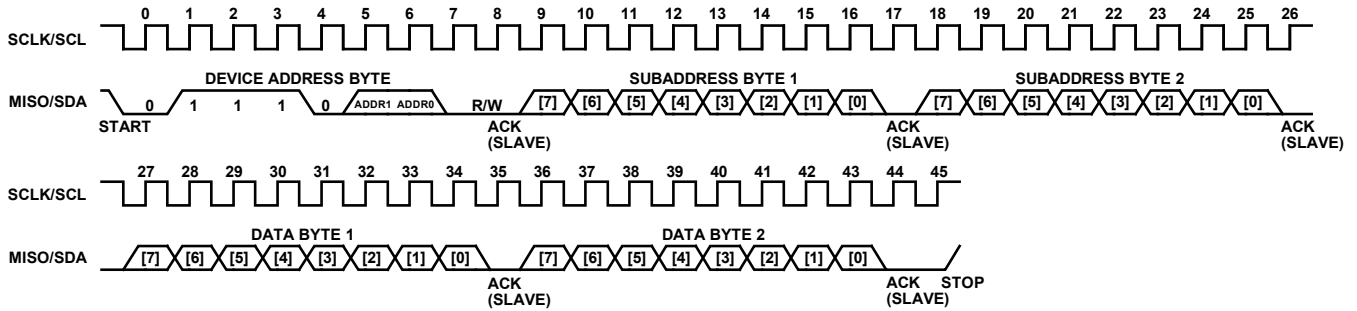


Figure 26. I²C Slave Single Word Write Operation (Two Bytes)

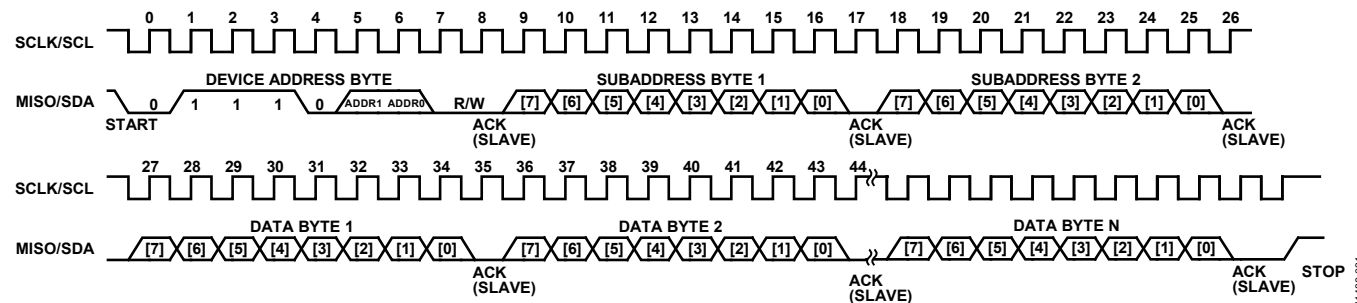


Figure 27. I²C Slave Burst Mode Write Operation (N Bytes)

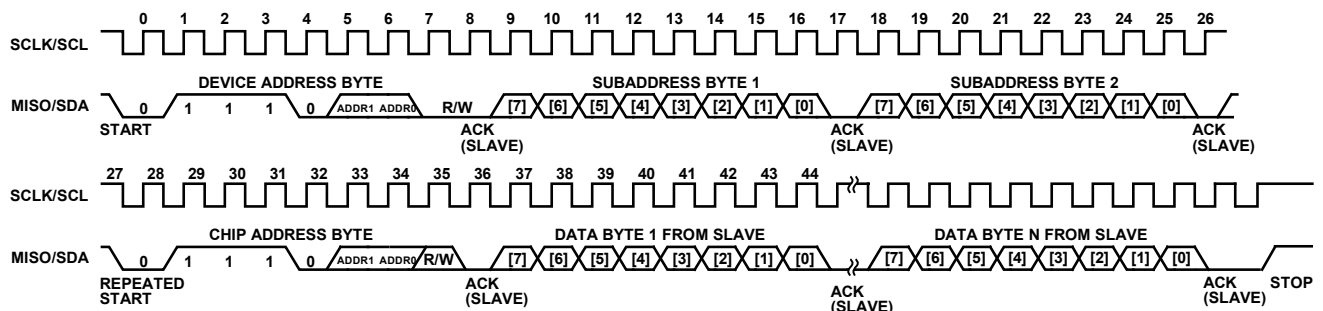


Figure 28. I²C Slave Burst Mode Read Operation (N Bytes)

I²C Read and Write Operations

Figure 29 shows the format of a single word write operation. Every ninth clock pulse, the ADAU1452 issues an acknowledge by pulling SDA low.

Figure 30 shows the simplified format of a burst mode write sequence. This figure shows an example of a write to sequential single byte registers. The ADAU1452 increments its subaddress register after every byte because the requested subaddress corresponds to a register or memory area with a 1-byte word length.

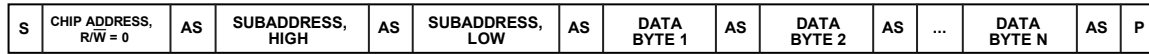
Figure 31 shows the format of a single word read operation. Note that the first R/W bit is 0, indicating a write operation. This is because the subaddress still needs to be written to set up the internal address. After the ADAU1452 acknowledges the receipt of the subaddress, the master must issue a repeated start command followed by the chip address byte with the R/W bit set to 1 (read).

This causes the ADAU1452 SDA to reverse and begin driving data back to the master. The master then responds every ninth pulse with an acknowledge pulse to the ADAU1452.

Figure 32 shows the format of a burst mode read sequence. This figure shows an example of a read from sequential single byte registers. The ADAU1452 increments its subaddress register after every byte because the requested subaddress corresponds to a register or memory area with a 1-byte word length. The ADAU1452 always decodes the subaddress and sets the auto-increment circuit so that the address increments after the appropriate number of bytes.

Figure 29 to Figure 32 use the following abbreviations:

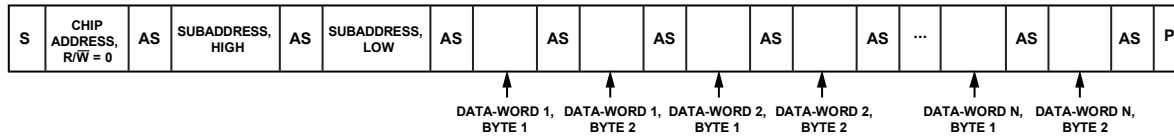
- S = start bit
- P = stop bit
- AM = acknowledge by master
- AS = acknowledge by slave



S = START BIT, P = STOP BIT, AM = ACKNOWLEDGE BY MASTER, AS = ACKNOWLEDGE BY SLAVE. SHOWS A ONE-WORD WRITE, WHERE EACH WORD HAS N BYTES.

Figure 29. Simplified Single Word I²C Write Sequence

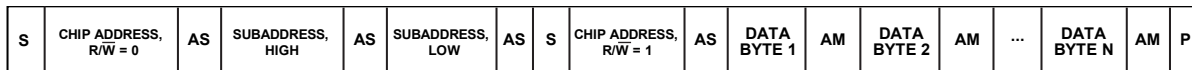
11486-033



S = START BIT, P = STOP BIT, AM = ACKNOWLEDGE BY MASTER, AS = ACKNOWLEDGE BY SLAVE. SHOWS AN N-WORD WRITE, WHERE EACH WORD HAS TWO BYTES. (OTHER WORD LENGTHS ARE POSSIBLE, RANGING FROM ONE TO FIVE BYTES.)

Figure 30. Simplified Burst Mode I²C Write Sequence

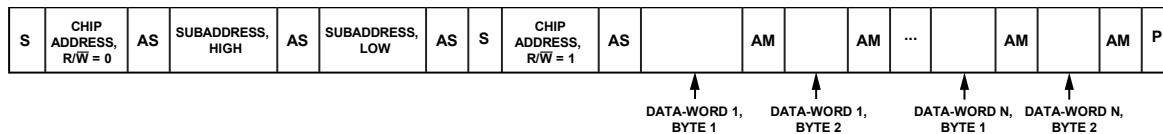
11486-034



S = START BIT, P = STOP BIT, AM = ACKNOWLEDGE BY MASTER, AS = ACKNOWLEDGE BY SLAVE. SHOWS A ONE-WORD READ, WHERE EACH WORD HAS N BYTES.

Figure 31. Simplified Single Word I²C Read Sequence

11486-035



S = START BIT, P = STOP BIT, AM = ACKNOWLEDGE BY MASTER, AS = ACKNOWLEDGE BY SLAVE. SHOWS AN N-WORD READ, WHERE EACH WORD HAS TWO BYTES. (OTHER WORD LENGTHS ARE POSSIBLE, RANGING FROM ONE TO FIVE BYTES.)

Figure 32. Simplified Burst Mode I²C Read Sequence

11486-036

SPI Slave Port

By default, the slave port is in I²C mode, but it can be put into SPI control mode by pulling SS/ADDR0 low three times. This can be done either by toggling the SS/ADDR0 successively between logic high and logic low states, or by performing three dummy writes to the SPI port, writing any arbitrary data to any arbitrary subaddress (the slave port does not acknowledge these three writes). After the SS/ADDR0 is toggled three times, data can be written to or read from the IC. An example of dummy writing is shown in Figure 33. Once set in SPI slave mode, the only way to revert back to I²C slave mode is by executing a full hardware reset using the RESET pin or by power-cycling the power supplies.

The SPI port uses a 4-wire interface, consisting of the SS, MOSI, MISO, and SCLK signals, and it is always a slave port. The SS signal goes low at the beginning of a transaction and high at the end of a transaction. The SCLK signal latches MOSI on a low-to-high transition. MISO data is shifted out of the device on the falling edge of SCLK and must be clocked into a receiving device, such as a microcontroller, on the SCLK rising edge. The MOSI signal carries the serial input data, and the MISO signal carries the serial output data. The MISO signal remains three-state until a read operation is requested. This allows other SPI-compatible peripherals to share the same MISO line. All SPI transactions have the same basic format shown in Table 31. A timing diagram is shown in Figure 8. Write all data MSB first.

There is only one chip address available in SPI mode. The 7-bit chip address is 0b0000000. The LSB of the first byte of an SPI transaction is an R/W bit. This bit determines whether the communication is a read (Logic Level 1) or a write (Logic Level 0). This format is shown in Table 30.

Table 30. SPI Address and Read/Write Byte Format

| Bit 0 | Bit 1 | Bit 2 | Bit 3 | Bit 4 | Bit 5 | Bit 6 | Bit 7 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | R/W |

The 16-bit subaddress word is decoded into a location in one of the registers. This subaddress is the location of the appropriate register. The MSBs of the subaddress are zero padded to bring the word to a full 2-byte length.

The format for the SPI communications slave port is commonly known as SPI Mode 3, where clock polarity (CPOL) = 1 and clock phase (CPHA) = 1 (see Figure 34). The base value of the clock is 1. Data is captured on the rising edge of the clock, and data is propagated on the falling edge.

The maximum read and write speed for the SPI slave port is 22 MHz, but this speed is valid only after the PLL is locked. Before the PLL locks, the maximum clock rate in the chip is limited to the frequency of the input clock to the PLL. Nominally, this is 3.072 MHz. Therefore, the SPI clock must not exceed 3.072 MHz until the PLL lock completes.

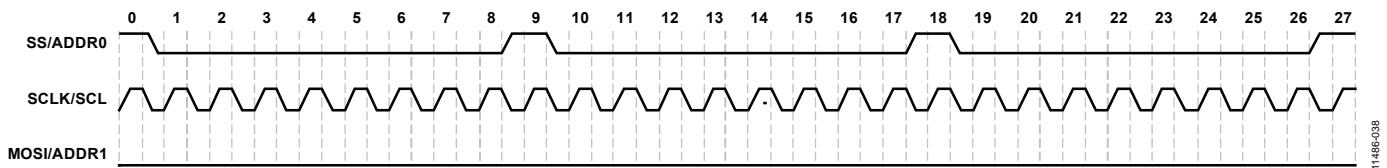


Figure 33. Example of SPI Slave Mode Initialization Sequence Using Dummy Writes

Table 31. Generic Control Word Sequence

| Byte 0 | Byte 1 | Byte 2 | Byte 3 | Byte 4 and Subsequent Bytes |
|------------------------|------------------|-----------------|--------|-----------------------------|
| Chip Address[6:0], R/W | Subaddress[15:8] | Subaddress[7:0] | Data | Data |

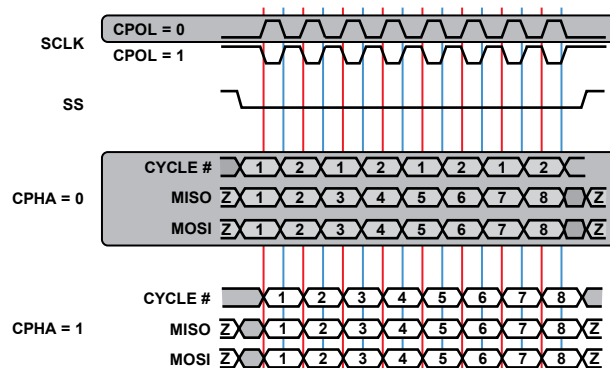


Figure 34. Clock Polarity and Phase for SPI Slave Port

A sample timing diagram for a multiple word SPI write operation to a register is shown in Figure 35. A sample timing diagram of a single word SPI read operation is shown in Figure 36. The MISO/SDA pin transitions from being three-state to being driven at the beginning of Byte 3. In this example, Byte 0 to Byte 2 contain

the addresses and $\overline{R/\overline{W}}$ bit, and subsequent bytes carry the data. A sample timing diagram of a multiple word SPI read operation is shown in Figure 37. In Figure 35 to Figure 37, rising edges on SCLK/SCL are indicated with an arrow, signifying that the data lines are sampled on the rising edge.

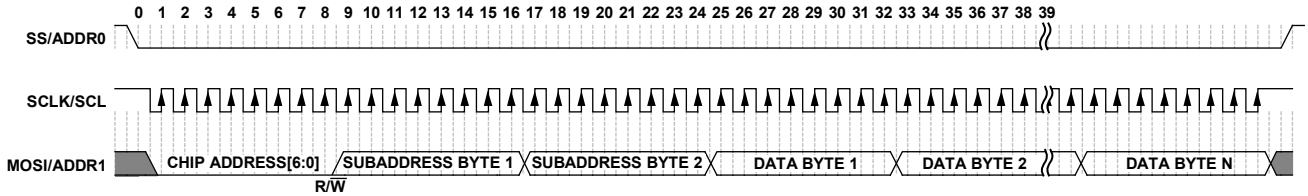


Figure 35. SPI Slave Write Clocking (Burst Write Mode, N Bytes)

11486-039

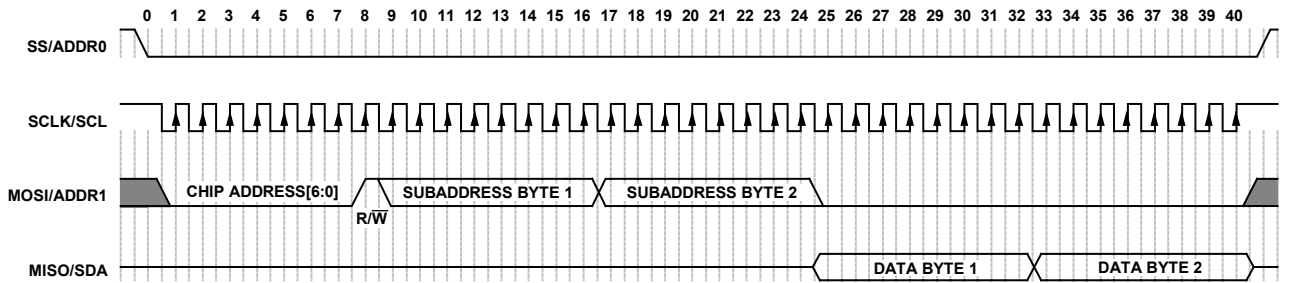


Figure 36. SPI Slave Read Clocking (Single Word Mode, Two Bytes)

11486-040

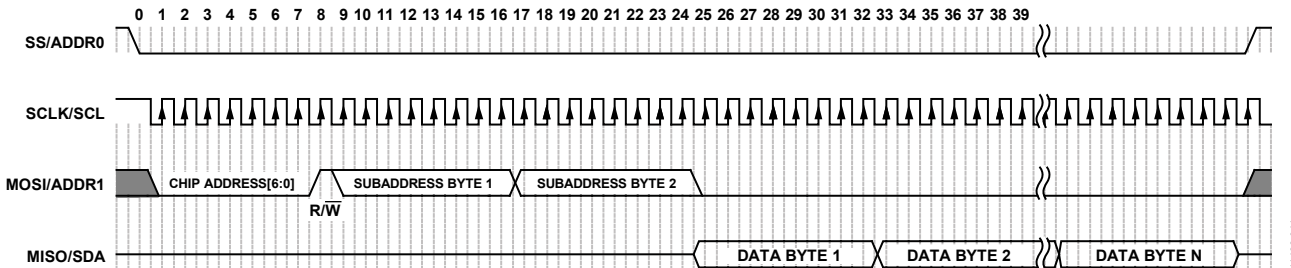


Figure 37. SPI Slave Read Clocking (Burst Read Mode, N Bytes)

11486-041

MASTER CONTROL PORTS

The device contains a combined I²C and SPI master control port that is accessible through a common interface. The master port can be enabled through a self boot operation or directly from the DSP core. The master control port can buffer up to 128 bits of data per single interrupt period. The smallest data transfer unit for both bus interfaces is one byte, and all transfers are 8-bit aligned. No error detection is supported, and single master operation is assumed. Only one bus interface protocol (I²C or SPI) can be used at a time.

The master control port can be used for several purposes, as follows:

- Self boot the [ADAU1452](#) from an external serial EEPROM.
- Boot and control external slave devices such as codecs and amplifiers.
- Read from and write to an external SPI RAM or flash memory.

SPI Master Interface

The SPI master supports up to seven slave devices (via the MPx pins) and speeds between 2.3 kHz and 20 MHz. SPI Mode 0 (CPOL = 0, CPHA = 0) and SPI Mode 3 (CPOL = 1, CPHA = 1) are supported. Communication is assumed to be half duplex, and the SPI master does not support a 3-wire interface. There is no JTAG or SGPIO support. The SPI interface uses a minimum of four general-purpose input/output pins of the processor and up to six additional MPx pins for additional slave select signals (SS). See Table 32 for more information.

The SPI master clock frequency can range between 2.3 kHz and 20 MHz. JTAG and SGPIO are not supported. Data transfers are

8-bit aligned. By default, the SPI master port is in Mode 3 (CPOL = 1, CPHA = 1), which matches the mode of the SPI slave port. The SPI master port can be configured to operate in Mode 0 (CPOL = 0, CPHA = 0) in the DSP program. No error detection or handling is implemented. Single master operation is assumed; therefore, no other master devices can exist on the same SPI bus.

The SPI master interface has been tested with EEPROM, flash, and serial RAM devices and has been confirmed to work in all cases.

When the data rate is very high on the SPI master interface (at 10 MHz or higher), a condition may arise where there is a high level of current draw on the IOVDD supply, which can lead to sagging of the internal IOVDD supply. To avoid potential issues, design the PCB such that the traces connecting the SPI master interface to external devices are kept as short as possible, and the slew rate and drive strength for SPI master interface pins are kept to a minimum to keep current draw as low as possible. Keeping IOVDD low (2.5 V or 1.8 V) also reduces the IOVDD current draw.

SigmaStudio generates EEPROM images for self boot systems, requiring no manual SPI master port configuration or programming on the part of the user.

I²C Master Interface

The I²C master is 7-bit addressable and supports standard and fast mode operation with speeds between 20 kHz and 400 kHz. The serial camera control bus (SCCB) and power management bus (PMBus) protocols are not supported. Data transfers are 8-bit aligned. No error detection or correction is implemented. The I²C master interface uses two general-purpose input/output pins, MP2 and MP3. See Table 33 for more information.

Table 32. SPI Master Interface Pin Functionality

| Pin Name | SPI Master Function | Description |
|------------------|---------------------|---|
| MOSI_M/MP1 | MOSI | SPI master port data output. Sends data from the SPI master port to slave devices on the SPI master bus. |
| SCL_M/SCLK_M/MP2 | SCLK | SPI master port serial clock. Drives the clock signal to slave devices on the SPI master bus. |
| SDA_M/MISO_M/MP3 | MISO | SPI master port data input. Receives data from slave devices on the SPI master bus. |
| SS_M/MP0 | SS | SPI master port slave select. Acts as the primary slave select signal to slave device on the SPI master bus. |
| MP4 to MP13 | SS | SPI master port slave select. These additional multipurpose pins can be configured to act as secondary slave select signals to additional slave devices on the SPI master bus. Up to seven slave devices, one per pin, are supported. |

Table 33. I²C Master Interface Pin Functionality

| Pin Name | I ² C Master Function | Description |
|------------------|----------------------------------|---|
| SCL_M/SCLK_M/MP2 | SCL | I ² C master port serial clock. This pin functions as an open collector output and drives a serial clock to slave devices on the I ² C bus. The line connected to this pin must have a 2.0 kΩ pull-up resistor to IOVDD. |
| SDA_M/MISO_M/MP3 | SDA | I ² C master port serial data. This pin functions as a bidirectional open collector data line between the I ² C master port and slave devices on the I ² C bus. The line connected to this pin must have a 2.0 kΩ pull-up resistor to IOVDD. |

SELF BOOT

The master control port is capable of booting the device from a single EEPROM by connecting the SELFBOOT pin to logic high (IOVDD) and powering up the power supplies while the RESET pin is pulled high. This initiates a self boot operation, in which the master control port downloads all required memory and register settings and automatically starts executing the DSP program without requiring external intervention or supervision. A self boot operation can also be triggered while the device is already in operation by initiating a rising edge of the RESET pin while the SELFBOOT pin is held high. When the self boot operation begins, the state of the SS_M/MP0 pin determines whether the SPI master or the I²C master carries out the self boot operation. If the SS_M/MP0 pin is connected to logic low, the I²C master port carries out the self boot operation. Otherwise, connect this pin to the slave select pin of the external slave device. The SPI master port then carries out the self boot operation.

When self booting from SPI, the chip assumes the following:

- The slave EEPROM is selected via the SS_M/MP0 pin.
- The slave EEPROM has 16- or 24-bit addressing, giving it a total memory size of between 4 kb and 64 Mb.
- The slave EEPROM supports serial clock frequencies down to 1 MHz or lower (a majority of the self boot operation uses a much higher clock frequency, but the initial transactions are performed at a slower frequency).
- The data stored in the slave EEPROM follows the format described in the EEPROM Self Boot Data Format section.
- The data is stored in the slave EEPROM with the MSB first.
- The slave EEPROM supports SPI Mode 3.
- The slave EEPROM sequential read operation has the command of 0x03.
- The slave EEPROM can be accessed immediately after it is powered up, with no manual configuration required.

When self booting from I²C, the chip assumes the following:

- The slave EEPROM has I²C Address 0x50.
- The slave EEPROM has 16-bit addressing, giving it a size of between 16 kb and 512 kb.
- The slave EEPROM supports standard mode clock frequencies of 100 kHz and lower (a majority of the self boot operation uses a much higher clock frequency, but the initial transactions are performed at a slower frequency).
- The data stored in the slave EEPROM follows the format described in the EEPROM Self Boot Data Format section.
- The slave EEPROM can be accessed immediately after it is powered, with no manual configuration required.

Self Boot Failure

The SPI or I²C master port attempts to self boot from the EEPROM three times. If all three self boot attempts fail, the SigmaDSP core issues a software panic and then enters a sleep state. During a self boot operation, the panic manager is unable to output a panic flag on a multipurpose pin. Therefore, the only way to debug a self boot failure is by reading back the status of Register 0xF427 (PANIC_FLAG) and Register 0xF428 (PANIC_CODE). The contents of Register 0xF428 indicate the nature of the failure.

EEPROM Self Boot Data Format

The self boot EEPROM image is generated using the SigmaStudio software; thus, the user does not need to manually create the data that is stored in the EEPROM. However, for reference, the details of the data format are described in this section.

The EEPROM self boot format consists of a fixed header, an arbitrary number of variable length blocks, and a fixed footer. The blocks themselves consist of a fixed header and a block of data with a variable length. Each data block can be placed anywhere in the DSP memory through configuration of the block header.

Header Format

The self boot EEPROM header consists of 16 bytes of data, starting at the beginning of the internal memory of the slave EEPROM (Address 0). The header format (see Figure 38) consists of the following:

- 8-bit Sentinel 0xAA (shown in Figure 38 as 0b10101010)
- 24-bit address indicating the byte address of the header of the first block (normally this is 0x000010, which is the address immediately following the header)
- 64-bit PLL configuration (PLL_CHECKSUM = PLL_FB_DIV + MCLK_OUT + PLL_DIV)

Data Block Format

Following the header, several data blocks are stored in the EEPROM memory (see Figure 39).

Each data block consists of eight bytes that configure the length and address of the data, followed by a series of 4-byte data packets.

Each block consists of the following:

- One LST bit, which signals the last block before the footer. LST = 0b1 indicates the last block; LST = 0b0 indicates that additional blocks are still to follow.
- 13 bits that are reserved for future use. Set these bits to 0b0.

- Two MEM bits that select the target data memory bank (0x0 = Data Memory 0, 0x1 = Data Memory 1, 0x2 = program memory).
- 16-bit base address that sets the memory address at which the master port starts writing when loading data from the block into memory.
- 16-bit data length that defines the number of 4-byte data-words to be written.
- 16-bit jump address that tells the DSP core at which address in program memory it should begin execution when the self boot operation is complete. The jump address bits are ignored unless the LST bit is set to 0b1.
- Arbitrary number of packets of 32-bit data. The number of packets is defined by the 16-bit data length.

Footer Format

After all the data blocks, a footer signifies the end of the self boot EEPROM memory (see Figure 40). The footer consists of a 64-bit checksum, which is the sum of the header and all blocks and all data as 32-bit words.

After the self boot operation completes, the checksum of the downloaded data is calculated and the panic manager signals if it does not match the checksum in the EEPROM. If the checksum is set to 0 dec, the checksum checking is disabled.

| BYTE 0 | | BYTE 1 | | BYTE 2 | | BYTE 3 | |
|-----------------------------|---|--------------|---|---------|---|------------|---|
| 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| ADDRESS OF FIRST BOOT BLOCK | | | | | | | |
| BYTE 4 | | BYTE 5 | | BYTE 6 | | BYTE 7 | |
| 0x00 | | PLL_DIV | | 0x00 | | PLL_FB_DIV | |
| BYTE 8 | | BYTE 9 | | BYTE 10 | | BYTE 11 | |
| 0x00 | | PLL_CHECKSUM | | 0x00 | | MCLK_OUT | |
| BYTE 12 | | BYTE 13 | | BYTE 14 | | BYTE 15 | |
| EEPROM SPEED CONFIGURATION | | | | | | | |

Figure 38. Self Boot EEPROM Header Format

11486-042

| BYTE 0 | | BYTE 1 | | BYTE 2 | | BYTE 3 | |
|-------------|----------|---------|--|--------------|--------------|---------|--|
| LST | RESERVED | | | MEM | BASE ADDRESS | | |
| BYTE 4 | | BYTE 5 | | BYTE 6 | | BYTE 7 | |
| DATA LENGTH | | | | JUMP ADDRESS | | | |
| BYTE 8 | | BYTE 9 | | BYTE 10 | | BYTE 11 | |
| DATA-WORD 1 | | | | | | | |
| BYTE 12 | | BYTE 13 | | BYTE 14 | | BYTE 15 | |
| DATA-WORD 2 | | | | | | | |

CONTINUED UNTIL LAST WORD IS REACHED...

| FOURTH TO LAST BYTE | | THIRD TO LAST BYTE | | SECOND TO LAST BYTE | | LAST BYTE | |
|---------------------|--|--------------------|--|---------------------|--|-----------|--|
| DATA-WORD N | | | | | | | |

Figure 39. Self Boot EEPROM Data Block Format

11486-043

| BYTE 0 | | BYTE 1 | | BYTE 2 | | BYTE 3 | |
|------------------------------|--|--------|--|--------|--|--------|--|
| FIRST FOUR BYTES OF CHECKSUM | | | | | | | |
| BYTE 4 | | BYTE 5 | | BYTE 6 | | BYTE 7 | |
| LAST FOUR BYTES OF CHECKSUM | | | | | | | |

Figure 40. Self Boot EEPROM Footer Format

11486-044

Considerations When Using a 1 Mb I²C Self Boot EEPROM

Because of the way I²C addressing works, 1 Mb of I²C EEPROM memory can be divided, with part of its address space at Chip Address 0x50; another portion of the memory can be located at a different address (for example, Chip Address 0x51). The memory allocation varies, depending on the EEPROM design. In cases when the EEPROM memory is divided, the memory portion that resides at a different chip address must be handled as though it exists in a separate EEPROM.

Considerations When Using Multiple EEPROMs on the SPI Master Bus

When multiple EEPROMs are connected on the same SPI master bus, the self boot mechanism works only with the first EEPROM.

AUDIO SIGNAL ROUTING

A large number of audio inputs and outputs are available in the device, and control registers are available for configuring the way in which the audio is routed between different functional blocks.

All input channels are accessible by both the DSP core and the ASRCs. Each ASRC can connect to a pair of audio channels from any of the input sources or from the DSP_TO_ASRC channels of the DSP core. The serial outputs can obtain their data from a number of sources, including the DSP core, ASRCs, PDM microphones, S/PDIF receiver, or directly from the serial inputs.

See Figure 41 for an overview of the audio routing matrix with its available audio data connections.

To route audio to and from the DSP core, select the appropriate input and output cells in SigmaStudio. These cells can be found in the IO folder of the SigmaStudio algorithm toolbox.

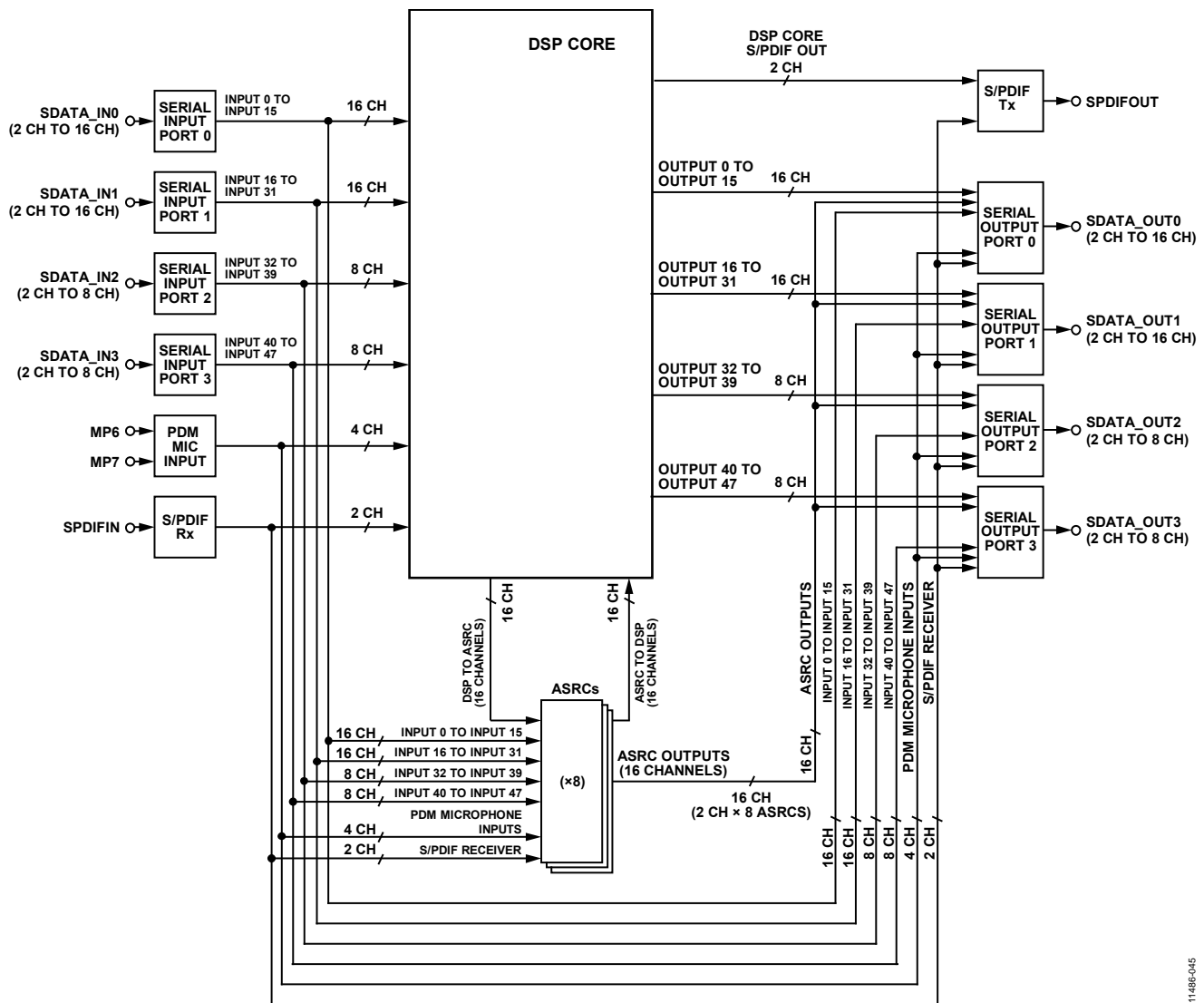


Figure 41. Audio Routing Overview

11465-045

Serial Audio Inputs to DSP Core

The 48 serial input channels are mapped to four audio input cells in SigmaStudio. Each input cell corresponds to one of the serial input pins (see Table 34).

Depending on whether the serial port is configured as 2-channel, 4-channel, 8-channel, or 16-channel mode, the available channels in SigmaStudio change. The channel count for each serial port is configured in the SERIAL_BYTE_x_0 registers, Bits[2:0] (TDM_MODE), at Address 0xF200 to Address 0xF21C (in increments of 0x4).

Table 34. Serial Input Pin Mapping to SigmaStudio Input Cells

| Serial Input Pin | Channels in SigmaStudio |
|------------------|-------------------------|
| SDATA_IN0 | 0 to 15 |
| SDATA_IN1 | 16 to 31 |
| SDATA_IN2 | 32 to 39 |
| SDATA_IN3 | 40 to 47 |

Table 35. Detailed Serial Input Mapping to SigmaStudio Input Channels¹

| Serial Input Pin | Position in I ² S Stream (2-Channel) | Position in TDM4 Stream | Position in TDM8 Stream | Position in TDM16 Stream | Input Channel in SigmaStudio |
|------------------|---|-------------------------|-------------------------|--------------------------|------------------------------|
| SDATA_IN0 | Left | 0 | 0 | 0 | 0 |
| SDATA_IN0 | Right | 1 | 1 | 1 | 1 |
| SDATA_IN0 | N/A | 2 | 2 | 2 | 2 |
| SDATA_IN0 | N/A | 3 | 3 | 3 | 3 |
| SDATA_IN0 | N/A | N/A | 4 | 4 | 4 |
| SDATA_IN0 | N/A | N/A | 5 | 5 | 5 |
| SDATA_IN0 | N/A | N/A | 6 | 6 | 6 |
| SDATA_IN0 | N/A | N/A | 7 | 7 | 7 |
| SDATA_IN0 | N/A | N/A | N/A | 8 | 8 |
| SDATA_IN0 | N/A | N/A | N/A | 9 | 9 |
| SDATA_IN0 | N/A | N/A | N/A | 10 | 10 |
| SDATA_IN0 | N/A | N/A | N/A | 11 | 11 |
| SDATA_IN0 | N/A | N/A | N/A | 12 | 12 |
| SDATA_IN0 | N/A | N/A | N/A | 13 | 13 |
| SDATA_IN0 | N/A | N/A | N/A | 14 | 14 |
| SDATA_IN0 | N/A | N/A | N/A | 15 | 15 |
| SDATA_IN1 | Left | 0 | 0 | 0 | 16 |
| SDATA_IN1 | Right | 1 | 1 | 1 | 17 |
| SDATA_IN1 | N/A | 2 | 2 | 2 | 18 |
| SDATA_IN1 | N/A | 3 | 3 | 3 | 19 |
| SDATA_IN1 | N/A | N/A | 4 | 4 | 20 |
| SDATA_IN1 | N/A | N/A | 5 | 5 | 21 |
| SDATA_IN1 | N/A | N/A | 6 | 6 | 22 |
| SDATA_IN1 | N/A | N/A | 7 | 7 | 23 |
| SDATA_IN1 | N/A | N/A | N/A | 8 | 24 |
| SDATA_IN1 | N/A | N/A | N/A | 9 | 25 |
| SDATA_IN1 | N/A | N/A | N/A | 10 | 26 |
| SDATA_IN1 | N/A | N/A | N/A | 11 | 27 |
| SDATA_IN1 | N/A | N/A | N/A | 12 | 28 |
| SDATA_IN1 | N/A | N/A | N/A | 13 | 29 |
| SDATA_IN1 | N/A | N/A | N/A | 14 | 30 |
| SDATA_IN1 | N/A | N/A | N/A | 15 | 31 |
| SDATA_IN2 | Left | 0 | 0 | 0 | 32 |
| SDATA_IN2 | Right | 1 | 1 | 1 | 33 |
| SDATA_IN2 | N/A | 2 | 2 | 2 | 34 |
| SDATA_IN2 | N/A | 3 | 3 | 3 | 35 |
| SDATA_IN2 | N/A | N/A | 4 | 4 | 36 |
| SDATA_IN2 | N/A | N/A | 5 | 5 | 37 |
| SDATA_IN2 | N/A | N/A | 6 | 6 | 38 |
| SDATA_IN2 | N/A | N/A | 7 | 7 | 39 |
| SDATA_IN3 | Left | 0 | 0 | 0 | 40 |
| SDATA_IN3 | Right | 1 | 1 | 1 | 41 |

| Serial Input Pin | Position in I ² S Stream (2-Channel) | Position in TDM4 Stream | Position in TDM8 Stream | Position in TDM16 Stream | Input Channel in SigmaStudio |
|------------------|---|-------------------------|-------------------------|--------------------------|------------------------------|
| SDATA_IN3 | N/A | 2 | 2 | 2 | 42 |
| SDATA_IN3 | N/A | 3 | 3 | 3 | 43 |
| SDATA_IN3 | N/A | N/A | 4 | 4 | 44 |
| SDATA_IN3 | N/A | N/A | 5 | 5 | 45 |
| SDATA_IN3 | N/A | N/A | 6 | 6 | 46 |
| SDATA_IN3 | N/A | N/A | 7 | 7 | 47 |

¹ N/A = not applicable.

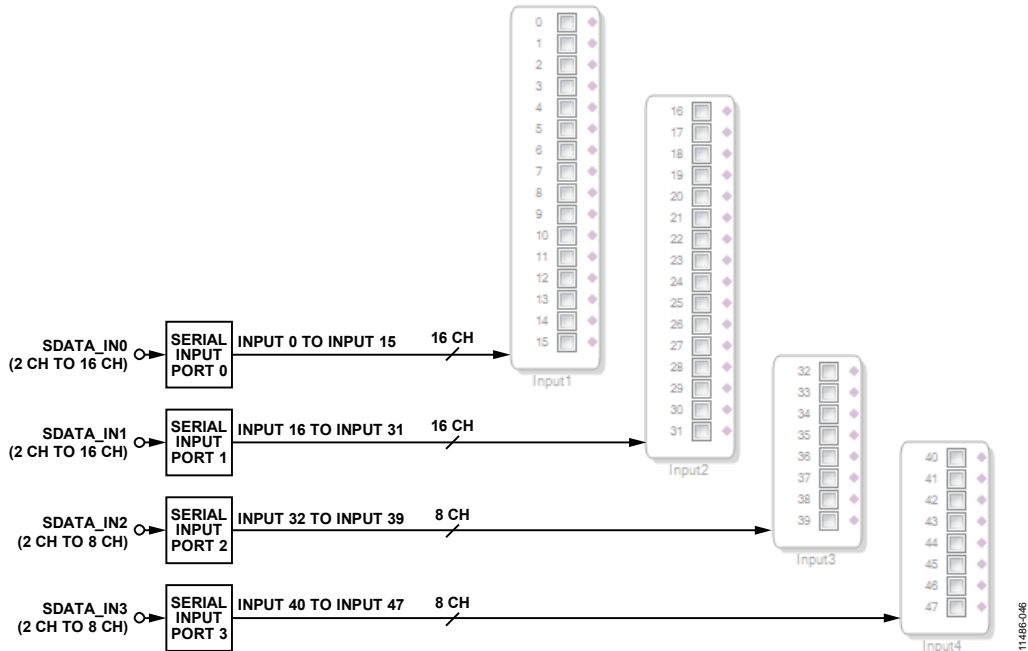


Figure 42. Serial Port Audio Input Mapping to DSP in SigmaStudio

Figure 42 shows how the input pins map to the input cells in SigmaStudio, including their graphical appearance in the software.

PDM Microphone Inputs to DSP Core

The PDM microphone inputs are mapped to a single digital microphone input cell in SigmaStudio (see Table 36 and Figure 43). The corresponding hardware pins are configured in Register 0xF560 (DMIC_CTRL0) and Register 0xF561 (DMIC_CTRL1).

Table 36. PDM Microphone Input Mapping to SigmaStudio Channels

| PDM Data Channel | PDM Microphone Input Channel in SigmaStudio |
|--------------------|---|
| Left (DMIC_CTRL0) | 0 |
| Right (DMIC_CTRL0) | 1 |
| Left (DMIC_CTRL1) | 2 |
| Right (DMIC_CTRL1) | 3 |

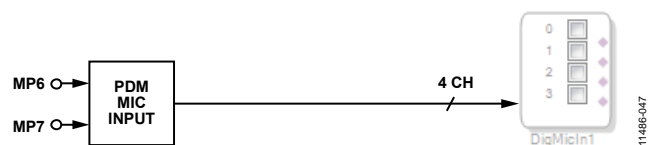


Figure 43. PDM Microphone Input Mapping to DSP in SigmaStudio

S/PDIF Receiver Inputs to DSP Core

The S/PDIF receiver can be accessed directly in the DSP core by using the S/PDIF input cell. However, in most applications, the S/PDIF receiver input is asynchronous to the DSP core, so an ASRC is typically required; in such cases, the S/PDIF input cell must not be used.

Table 37. S/PDIF Input Mapping to SigmaStudio Channels

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

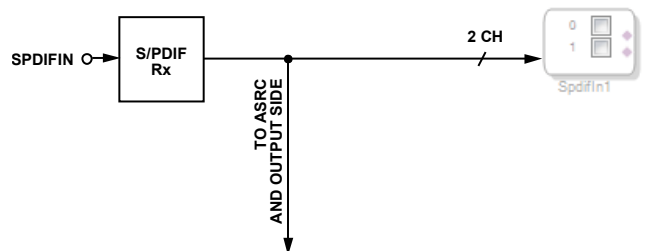


Figure 44. S/PDIF Receiver Direct Input Mapping to DSP in SigmaStudio

Serial Audio Outputs from DSP Core

The 48 serial output channels are mapped to 48 separate audio output cells in SigmaStudio. Each audio output cell corresponds to a single output channel. The first 16 channels are mapped to

the SDATA_OUT0 pin. The next 16 channels are mapped to the SDATA_OUT1 pin. The following eight channels are mapped to the SDATA_OUT2 pin. The last eight channels are mapped to the SDATA_OUT3 pin (see Table 38 and Figure 45).

Table 38. Serial Output Pin Mapping from SigmaStudio Channels¹

| Channel in SigmaStudio | Serial Output Pin | Position in I ² S 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 | N/A | 2 | 2 | 2 |
| 3 | SDATA_OUT0 | N/A | 3 | 3 | 3 |
| 4 | SDATA_OUT0 | N/A | N/A | 4 | 4 |
| 5 | SDATA_OUT0 | N/A | N/A | 5 | 5 |
| 6 | SDATA_OUT0 | N/A | N/A | 6 | 6 |
| 7 | SDATA_OUT0 | N/A | N/A | 7 | 7 |
| 8 | SDATA_OUT0 | N/A | N/A | N/A | 8 |
| 9 | SDATA_OUT0 | N/A | N/A | N/A | 9 |
| 10 | SDATA_OUT0 | N/A | N/A | N/A | 10 |
| 11 | SDATA_OUT0 | N/A | N/A | N/A | 11 |
| 12 | SDATA_OUT0 | N/A | N/A | N/A | 12 |
| 13 | SDATA_OUT0 | N/A | N/A | N/A | 13 |
| 14 | SDATA_OUT0 | N/A | N/A | N/A | 14 |
| 15 | SDATA_OUT0 | N/A | N/A | N/A | 15 |
| 16 | SDATA_OUT1 | Left | 0 | 0 | 0 |
| 17 | SDATA_OUT1 | Right | 1 | 1 | 1 |
| 18 | SDATA_OUT1 | N/A | 2 | 2 | 2 |
| 19 | SDATA_OUT1 | N/A | 3 | 3 | 3 |
| 20 | SDATA_OUT1 | N/A | N/A | 4 | 4 |
| 21 | SDATA_OUT1 | N/A | N/A | 5 | 5 |
| 22 | SDATA_OUT1 | N/A | N/A | 6 | 6 |
| 23 | SDATA_OUT1 | N/A | N/A | 7 | 7 |
| 24 | SDATA_OUT1 | N/A | N/A | N/A | 8 |
| 25 | SDATA_OUT1 | N/A | N/A | N/A | 9 |
| 26 | SDATA_OUT1 | N/A | N/A | N/A | 10 |
| 27 | SDATA_OUT1 | N/A | N/A | N/A | 11 |
| 28 | SDATA_OUT1 | N/A | N/A | N/A | 12 |
| 29 | SDATA_OUT1 | N/A | N/A | N/A | 13 |
| 30 | SDATA_OUT1 | N/A | N/A | N/A | 14 |
| 31 | SDATA_OUT1 | N/A | N/A | N/A | 15 |
| 32 | SDATA_OUT2 | Left | 0 | 0 | 0 |
| 33 | SDATA_OUT2 | Right | 1 | 1 | 1 |
| 34 | SDATA_OUT2 | N/A | 2 | 2 | 2 |
| 35 | SDATA_OUT2 | N/A | 3 | 3 | 3 |
| 36 | SDATA_OUT2 | N/A | N/A | 4 | 4 |
| 37 | SDATA_OUT2 | N/A | N/A | 5 | 5 |
| 38 | SDATA_OUT2 | N/A | N/A | 6 | 6 |
| 39 | SDATA_OUT2 | N/A | N/A | 7 | 7 |
| 40 | SDATA_OUT3 | Left | 0 | 0 | 0 |
| 41 | SDATA_OUT3 | Right | 1 | 1 | 1 |
| 42 | SDATA_OUT3 | N/A | 2 | 2 | 2 |
| 43 | SDATA_OUT3 | N/A | 3 | 3 | 3 |
| 44 | SDATA_OUT3 | N/A | N/A | 4 | 4 |
| 45 | SDATA_OUT3 | N/A | N/A | 5 | 5 |
| 46 | SDATA_OUT3 | N/A | N/A | 6 | 6 |
| 47 | SDATA_OUT3 | N/A | N/A | 7 | 7 |

¹ N/A = not applicable.

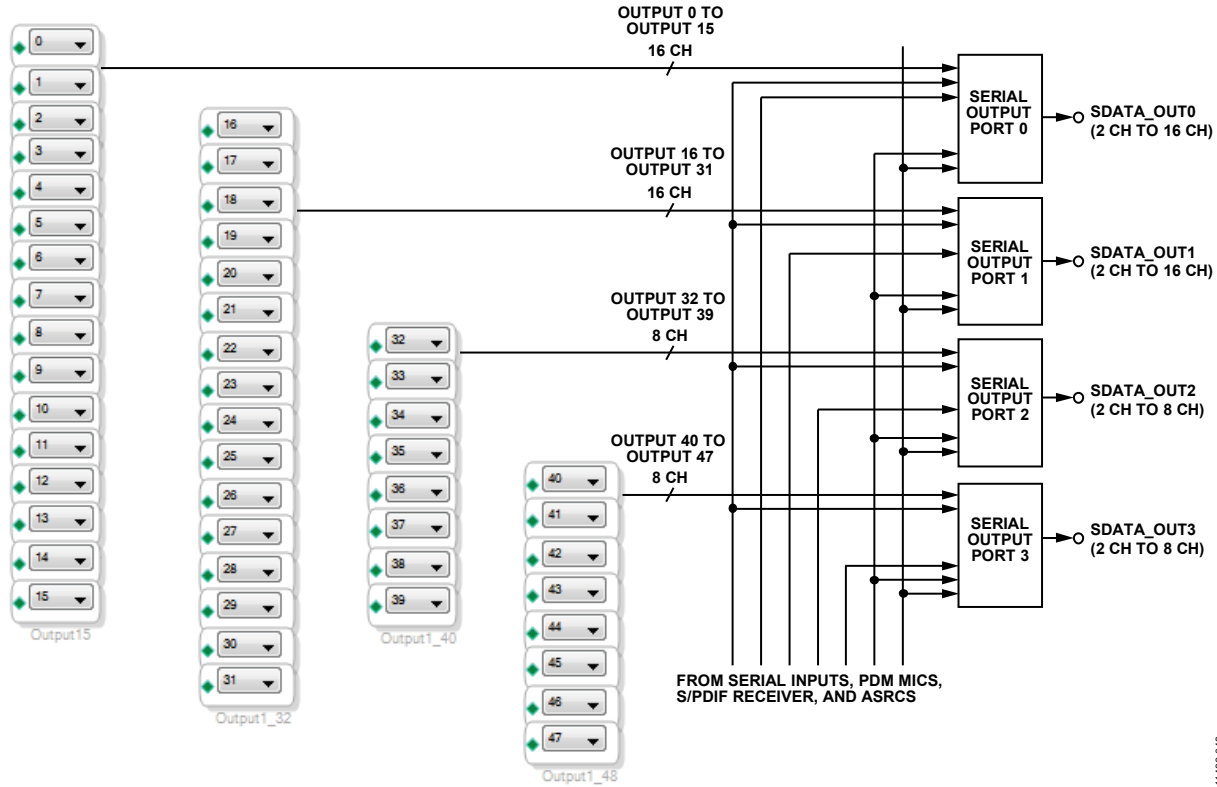


Figure 45. DSP to Serial Output Mapping in SigmaStudio

11486-049

The data that is output from each serial output pin can also be configured, via the SOUT_SOURCEx registers, to originate from one of the following sources: the DSP, the serial inputs, the PDM microphone inputs, the S/PDIF receiver, or the ASRCs. These registers can be configured graphically in SigmaStudio, as shown in Figure 46.

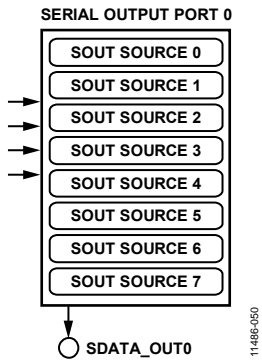


Figure 46. Configuring the Serial Output Data Channels (SOUT_SOURCEx Registers) Graphically in SigmaStudio

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).

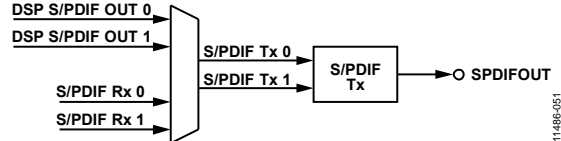


Figure 47. S/PDIF Transmitter Source Selection

11486-051

When the signal comes from the DSP core, use the S/PDIF output cells in SigmaStudio.

Table 39. S/PDIF Output Mapping from SigmaStudio Channels

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

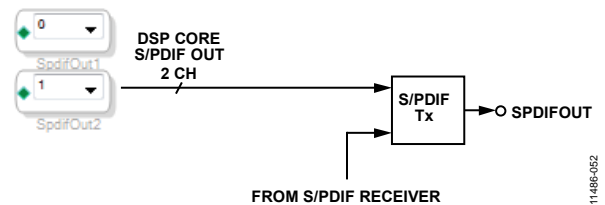


Figure 48. DSP to S/PDIF Transmitter Output Mapping in SigmaStudio

11486-052

Asynchronous Sample Rate Converter Input Routing

Any asynchronous input can be routed to the ASRCs to be resynchronized to a desired target sample rate (see Figure 49). The source signals for any ASRC can come from any of the serial inputs, any of the DSP-to-ASRC channels, the S/PDIF receiver, or the digital PDM microphone inputs. There are eight ASRCs, each with two input channels and two output channels. This means a total of 16 channels can pass through the ASRCs. Asynchronous input signals (either serial inputs, PDM microphone inputs, or the S/PDIF input) typically need to be routed to an ASRC and then synchronized to the DSP core rate. They are then available for input to the DSP core for processing.

In the example shown in Figure 50, the two channels from the S/PDIF receiver are routed to one of the ASRCs and then to the DSP core. For this example, the corresponding ASRC input selector register (Register 0xF100 to Register 0xF107, ASRC_INPUTx), Bits[2:0] (ASRC_SOURCE) is set to 0b011 to take the input from the S/PDIF receiver, and the corresponding ASRC output rate selector register (Register 0xF140 to Register 0xF147, ASRC_OUT_RATEx, Bits[3:0] (ASRC_RATE)) is set to 0b0101 to synchronize the ASRC output data to the DSP core sample rate.

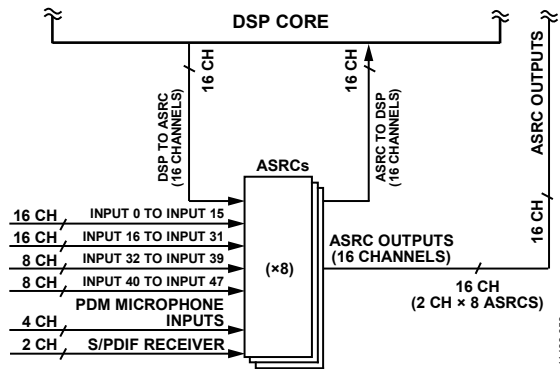


Figure 49. Channel Routing to ASRC Inputs

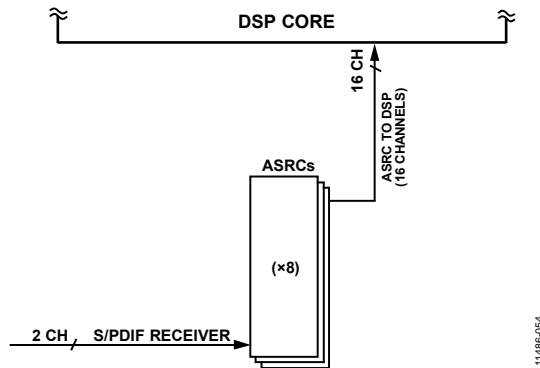


Figure 50. Example ASRC Routing for Asynchronous Input to the DSP Core

When the outputs of the ASRCs are required for processing in the SigmaDSP core, the ASRC input block must be selected in SigmaStudio (see Figure 51 and Figure 52).

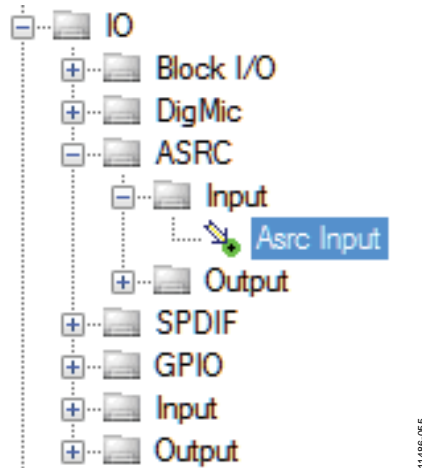


Figure 51. Location of ASRC-to-DSP Input Cell in SigmaStudio Toolbox

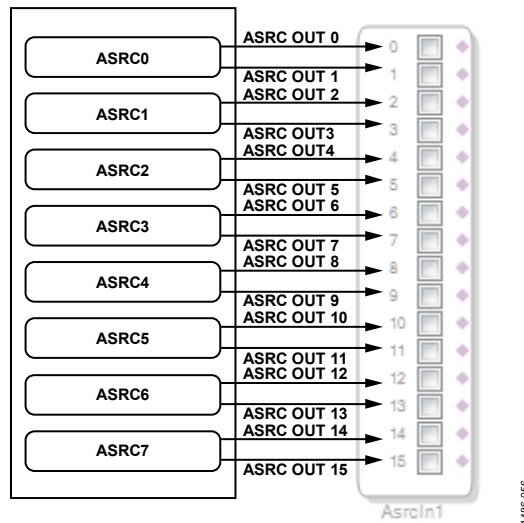


Figure 52. Routing of ASRC Outputs to ASRC-to-DSP Input Cell in SigmaStudio

Asynchronous output signals (for example, serial outputs that are slaves to an external, asynchronous device) typically are routed from the DSP core into the ASRCs, where they are synchronized to the serial output port that is acting as a slave to the external asynchronous master device.

In the example shown in Figure 53, two (or more) audio channels from the DSP core are routed to one (or more) of the ASRCs and then to the serial outputs. For this example, the corresponding ASRC input selector register (Address 0xF100 to Address 0xF107 (ASRC_INPUTx), Bits[2:0] (ASRC_SOURCE)) is set to 0b010 to take the data from the DSP core, and the corresponding ASRC output rate selector register (Address 0xF140 to Address 0xF147 (ASRC_OUT_RATEx), Bits[3:0] (ASRC_RATE)) is set to one of the following: 0b0001 to synchronize the ASRC output data to SDATA_OUT0; 0b0010 to synchronize the ASRC output data to SDATA_OUT1; 0b0011 to synchronize the ASRC output data to SDATA_OUT2; or 0b0100 to synchronize the ASRC output data to SDATA_OUT3. Next, the corresponding serial output port data source register (Address 0xF180 to Address 0xF197 (SOUT_SOURCEx), Bits[2:0] (SOUT_SOURCE)) must be set to 0b011 to receive the data from the ASRC outputs, and Bits[5:3] (SOUT_ASRC_SELECT) must be configured to select the correct ASRC from which to receive the output data.

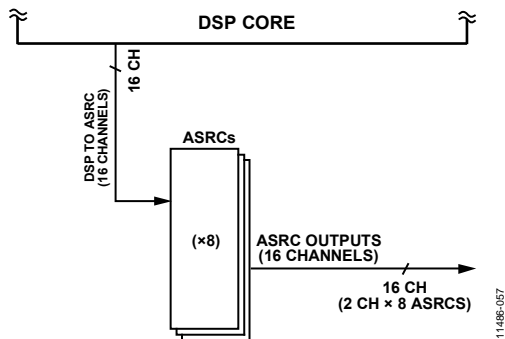


Figure 53. Example ASRC Routing for Asynchronous Serial Output from the DSP Core

When signals must route from the DSP core to the ASRCs, use the DSP-to-ASRC output cell in SigmaStudio (see Figure 54).

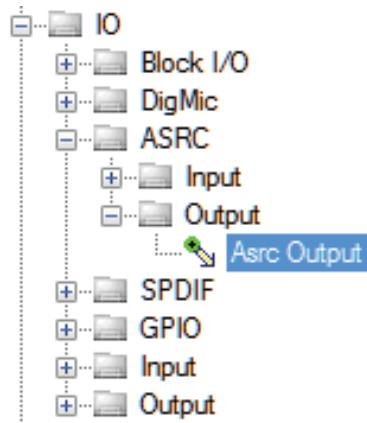


Figure 54. Location of DSP-to-ASRC Output Cell in SigmaStudio Toolbox

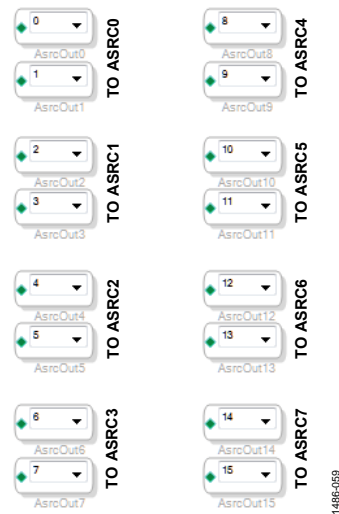


Figure 55. Routing of DSP-to-ASRC Output Cells in SigmaStudio to ASRC Inputs

The ASRCs can also be used to take asynchronous inputs and convert them to a different sample rate without doing any processing in the DSP core.

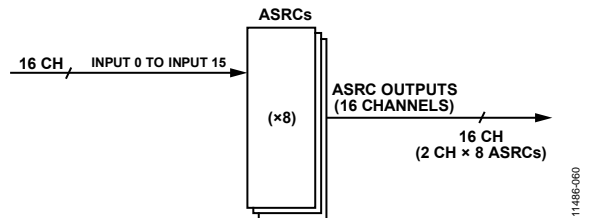


Figure 56. Example ASRC Routing, Bypassing DSP Core

Configure the ASRC routing registers using a simple graphical interface in the SigmaStudio software (see Figure 58).

Asynchronous Sample Rate Converter Output Routing

The outputs of the ASRCs are always available at both the DSP core and the serial outputs. No manual routing is necessary. To route ASRC output data to serial output channels, configure Register 0xF180 to Register 0xF197 (SOUT_SOURCEx) accordingly. For more information, see Figure 57 and Table 40.

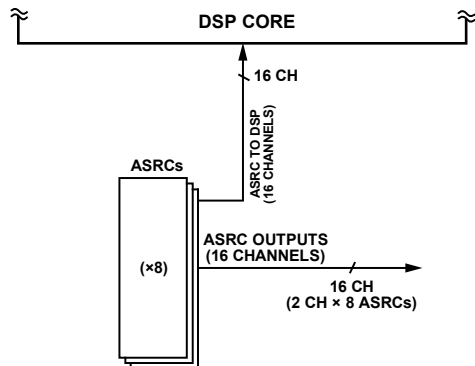


Figure 57. ASRC Outputs

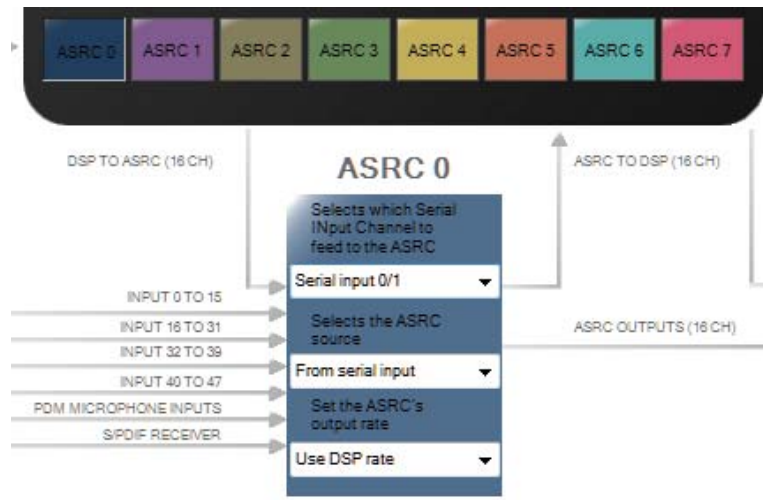


Figure 58. Configuring the ASRC Input Source and Target Rate in SigmaStudio

Audio Signal Routing Registers

An overview of the registers related to audio routing is listed in Table 40. For more detailed information, refer to the Audio Signal Routing Registers section.

Table 40. Audio Routing Matrix Registers

| Address | Register | Description |
|---------|----------------|---|
| 0xF100 | ASRC_INPUT0 | ASRC input selector (ASRC 0, Channel 0 and Channel 1) |
| 0xF101 | ASRC_INPUT1 | ASRC input selector (ASRC 1, Channel 2 and Channel 3) |
| 0xF102 | ASRC_INPUT2 | ASRC input selector (ASRC 2, Channel 4 and Channel 5) |
| 0xF103 | ASRC_INPUT3 | ASRC input selector (ASRC 3, Channel 6 and Channel 7) |
| 0xF104 | ASRC_INPUT4 | ASRC input selector (ASRC 4, Channel 8 and Channel 9) |
| 0xF105 | ASRC_INPUT5 | ASRC input selector (ASRC 5, Channel 10 and Channel 11) |
| 0xF106 | ASRC_INPUT6 | ASRC input selector (ASRC 6, Channel 12 and Channel 13) |
| 0xF107 | ASRC_INPUT7 | ASRC input selector (ASRC 7, Channel 14 and Channel 15) |
| 0xF140 | ASRC_OUT_RATE0 | ASRC output rate (ASRC 0, Channel 0 and Channel 1) |
| 0xF141 | ASRC_OUT_RATE1 | ASRC output rate (ASRC 1, Channel 2 and Channel 3) |
| 0xF142 | ASRC_OUT_RATE2 | ASRC output rate (ASRC 2, Channel 4 and Channel 5) |
| 0xF143 | ASRC_OUT_RATE3 | ASRC output rate (ASRC 3, Channel 6 and Channel 7) |
| 0xF144 | ASRC_OUT_RATE4 | ASRC output rate (ASRC 4, Channel 8 and Channel 9) |
| 0xF145 | ASRC_OUT_RATE5 | ASRC output rate (ASRC 5, Channel 10 and Channel 11) |
| 0xF146 | ASRC_OUT_RATE6 | ASRC output rate (ASRC 6, Channel 12 and Channel 13) |
| 0xF147 | ASRC_OUT_RATE7 | ASRC output rate (ASRC 7, Channel 14 and Channel 15) |
| 0xF180 | SOUT_SOURCE0 | Source of data for serial output port (Channel 0 and Channel 1) |
| 0xF181 | SOUT_SOURCE1 | Source of data for serial output port (Channel 2 and Channel 3) |
| 0xF182 | SOUT_SOURCE2 | Source of data for serial output port (Channel 4 and Channel 5) |
| 0xF183 | SOUT_SOURCE3 | Source of data for serial output port (Channel 6 and Channel 7) |
| 0xF184 | SOUT_SOURCE4 | Source of data for serial output port (Channel 8 and Channel 9) |
| 0xF185 | SOUT_SOURCE5 | Source of data for serial output port (Channel 10 and Channel 11) |
| 0xF186 | SOUT_SOURCE6 | Source of data for serial output port (Channel 12 and Channel 13) |
| 0xF187 | SOUT_SOURCE7 | Source of data for serial output port (Channel 14 and Channel 15) |
| 0xF188 | SOUT_SOURCE8 | Source of data for serial output port (Channel 16 and Channel 17) |
| 0xF189 | SOUT_SOURCE9 | Source of data for serial output port (Channel 18 and Channel 19) |
| 0xF18A | SOUT_SOURCE10 | Source of data for serial output port (Channel 20 and Channel 21) |
| 0xF18B | SOUT_SOURCE11 | Source of data for serial output port (Channel 22 and Channel 23) |
| 0xF18C | SOUT_SOURCE12 | Source of data for serial output port (Channel 24 and Channel 25) |
| 0xF18D | SOUT_SOURCE13 | Source of data for serial output port (Channel 26 and Channel 27) |
| 0xF18E | SOUT_SOURCE14 | Source of data for serial output port (Channel 28 and Channel 29) |
| 0xF18F | SOUT_SOURCE15 | Source of data for serial output port (Channel 30 and Channel 31) |
| 0xF190 | SOUT_SOURCE16 | Source of data for serial output port (Channel 32 and Channel 33) |
| 0xF191 | SOUT_SOURCE17 | Source of data for serial output port (Channel 34 and Channel 35) |
| 0xF192 | SOUT_SOURCE18 | Source of data for serial output port (Channel 36 and Channel 37) |
| 0xF193 | SOUT_SOURCE19 | Source of data for serial output port (Channel 38 and Channel 39) |
| 0xF194 | SOUT_SOURCE20 | Source of data for serial output port (Channel 40 and Channel 41) |
| 0xF195 | SOUT_SOURCE21 | Source of data for serial output port (Channel 42 and Channel 43) |
| 0xF196 | SOUT_SOURCE22 | Source of data for serial output port (Channel 44 and Channel 45) |
| 0xF197 | SOUT_SOURCE23 | Source of data for serial output port (Channel 46 and Channel 47) |
| 0xF1C0 | SPDIFTX_INPUT | S/PDIF transmitter data selector |

SERIAL DATA INPUT/OUTPUT

There are four serial data input pins (SDATA_IN3 to SDATA_IN0) and four serial data output pins (SDATA_OUT3 to SDATA_OUT0). Each pin is capable of 2-channel, 4-channel, or 8-channel mode. In addition, SDATA_IN0, SDATA_IN1, SDATA_OUT0, and SDATA_OUT1 are capable of 16-channel mode.

The serial ports have a very flexible configuration scheme that allows for completely independent and orthogonal configuration of clock pin assignment, clock waveform type, clock polarity, channel count, position of the data bits within the stream, audio word length, slave or master operation, and sample rate. A detailed description of all possible serial port settings is included in the Serial Port Configuration Registers.

The physical serial data input and output pins are connected to functional blocks called serial ports, which deal with handling the audio data and clocks as they pass in and out of the device. Table 41 describes this relationship.

Table 41. Relationship Between Hardware Serial Data Pins and Serial Input/Output Ports

| Serial Data Pin | Serial Port |
|-----------------|----------------------|
| SDATA_IN0 | Serial Input Port 0 |
| SDATA_IN1 | Serial Input Port 1 |
| SDATA_IN2 | Serial Input Port 2 |
| SDATA_IN3 | Serial Input Port 3 |
| SDATA_OUT0 | Serial Output Port 0 |
| SDATA_OUT1 | Serial Output Port 1 |
| SDATA_OUT2 | Serial Output Port 2 |
| SDATA_OUT3 | Serial Output Port 3 |

There are 48 channels of serial audio data inputs and 48 channels of serial audio data outputs. The 48 audio input channels and 48 audio output channels are distributed among the four serial data input pins and the four serial data output pins. This distribution is described in Table 42.

The maximum sample rate for the serial audio data on the serial ports is 192 kHz. The minimum sample rate is 6 kHz.

SDATA_IN2, SDATA_IN3, SDATA_OUT2, and SDATA_OUT3 are capable of operating in a special mode called flexible TDM mode, which allows for custom byte addressable configuration, where the data for each channel is located in the serial data stream. Flexible TDM mode is not a standard audio interface. Use it only in cases where a customized serial data format is desired. See the Flexible TDM Interface section for more information.

Serial Audio Data Format

The serial data input and output ports are designed to work with audio data that is encoded in a linear pulse code modulation (PCM) format, based on the common I²S standard. Audio data-words can be 16, 24, or 32 bits in length. The serial ports can handle TDM formats with channel counts ranging from two channels to 16 channels on a single data line.

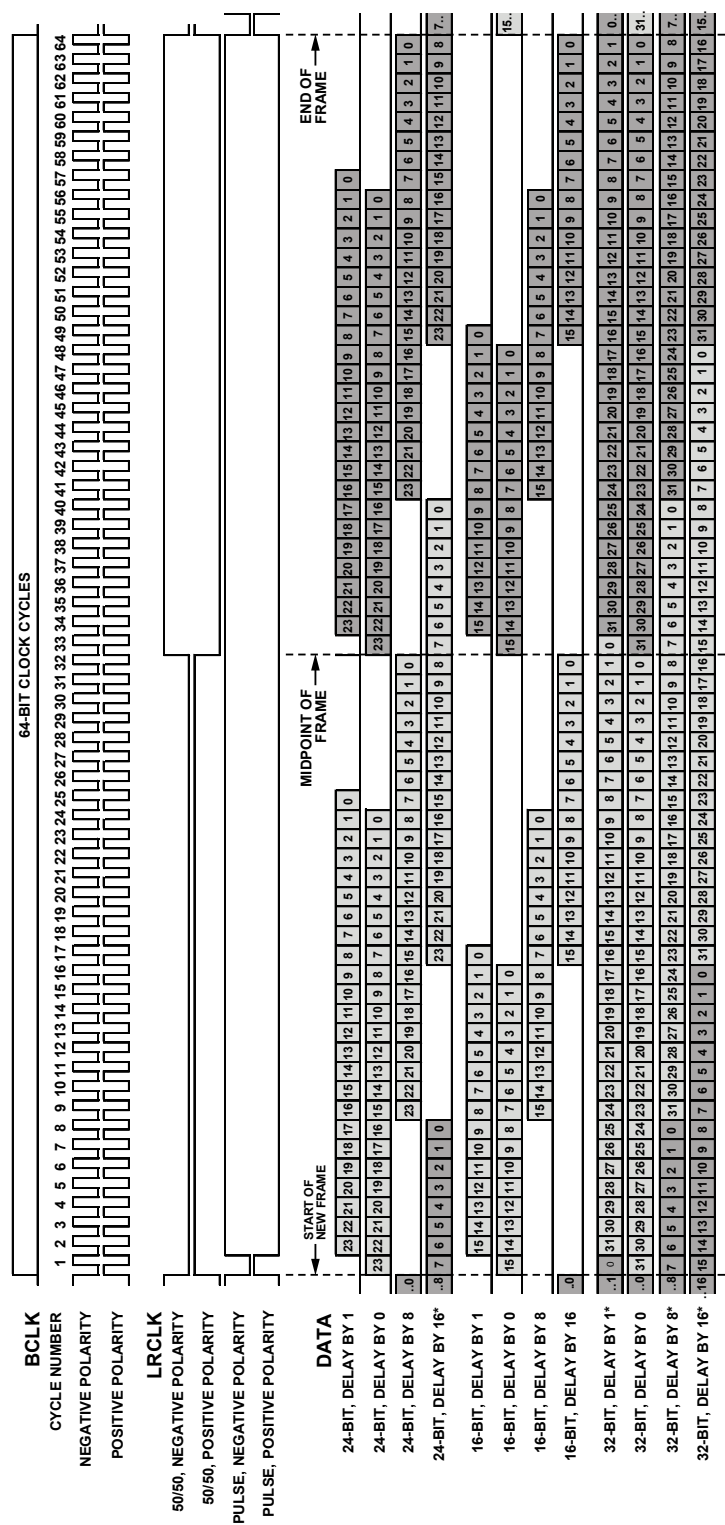
Almost every aspect of the serial audio data format can be configured using the SERIAL_BYTE_x_0 and SERIAL_BYTE_x_1 registers, and every setting can be configured independently. As a result, there are more than 70,000 valid configurations for each serial audio port.

Serial Audio Data Timing Diagrams

Because it is impractical to show timing diagrams for each possible combination, timing diagrams for the more common configurations are shown in Figure 59 to Figure 64. Explanatory text accompanies each figure.

Table 42. Relationship Between Data Pin, Audio Channels, Clock Pins, and TDM Options

| Serial Data Pin | Channel Numbering | Corresponding Clock Pins in Master Mode | Maximum TDM Channels | Flexible TDM Mode |
|-----------------|--------------------------|---|----------------------|-------------------|
| SDATA_IN0 | Channel 0 to Channel 15 | BCLK_IN0, LRCLK_IN0 | 16 channels | No |
| SDATA_IN1 | Channel 16 to Channel 31 | BCLK_IN1, LRCLK_IN1 | 16 channels | No |
| SDATA_IN2 | Channel 32 to Channel 39 | BCLK_IN2, LRCLK_IN2 | 8 channels | Yes |
| SDATA_IN3 | Channel 40 to Channel 47 | BCLK_IN3, LRCLK_IN3 | 8 channels | Yes |
| SDATA_OUT0 | Channel 0 to Channel 15 | BCLK_OUT0, LRCLK_OUT0 | 16 channels | No |
| SDATA_OUT1 | Channel 16 to Channel 31 | BCLK_OUT1, LRCLK_OUT1 | 16 channels | No |
| SDATA_OUT2 | Channel 32 to Channel 39 | BCLK_OUT2, LRCLK_OUT2 | 8 channels | Yes |
| SDATA_OUT3 | Channel 40 to Channel 47 | BCLK_OUT3, LRCLK_OUT3 | 8 channels | Yes |



*IT IS POSSIBLE FOR THE USER TO CONFIGURE THE SERIAL PORTS TO OPERATE IN THIS MODE. HOWEVER, IT IS RECOMMENDED THAT THIS MODE NOT BE USED BECAUSE THE AUDIO DATA CROSSES THE THRESHOLD BETWEEN TWO FRAMES, WHICH MAY VIOLATE THE SPECIFICATIONS OF OTHER DEVICES IN THE SYSTEM.

Figure 59. Serial Audio Formats; Two Channels, 32 Bits per Channel

Figure 59 shows timing diagrams for possible serial port configurations in 2-channel mode, with 32 cycles of the bit clock signal per channel, for a total of 64 bit clock cycles per frame (see the SERIAL_BYTE_x_0 registers, Bits[2:0] (TDM_MODE) = 0b000). Different bit clock polarities are illustrated in Figure 59 (SERIAL_BYTE_x_0, Bit 7 (BCLK_POL)) as well as different

frame clock waveforms and polarities (SERIAL_BYTE_x_0, Bit 9 (LRCLK_MODE) and Bit 8 (LRCLK_POL)). Excluding flexible TDM mode, there are 12 possible combinations of settings for the audio word length (SERIAL_BYTE_x_0, Bits[6:5] (WORD_LEN)) and MSB position (SERIAL_BYTE_x_0, Bits[4:3] (DATA_FMT)), all of which are shown in Figure 59.

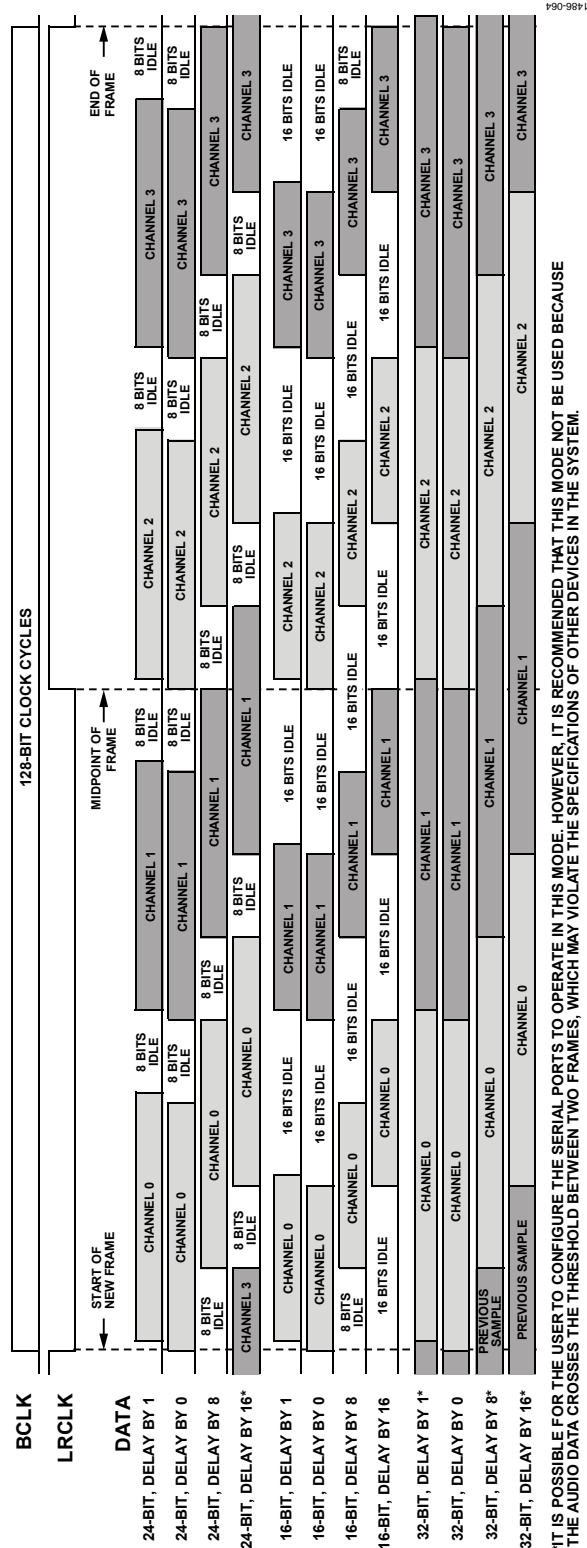


Figure 60. Serial Audio Data Formats; Four Channels, 32 Bits per Channel

Figure 60 shows timing diagrams for possible serial port configurations in 4-channel mode, with 32 bit clock cycles per channel, for a total of 128 bit clock cycles per frame (refer to the SERIAL_BYTE_x_0 registers, Bits[2:0] (TDM_MODE) = 0b001). The bit clock signal is omitted from the figure.

Excluding flexible TDM mode, there are 12 possible combinations of settings for the audio word length (SERIAL_BYTE_x_0, Bits[6:5] (WORD_LEN)) and MSB position (SERIAL_BYTE_x_0, Bits[4:3] (DATA_FMT)), all of which are shown in Figure 60.

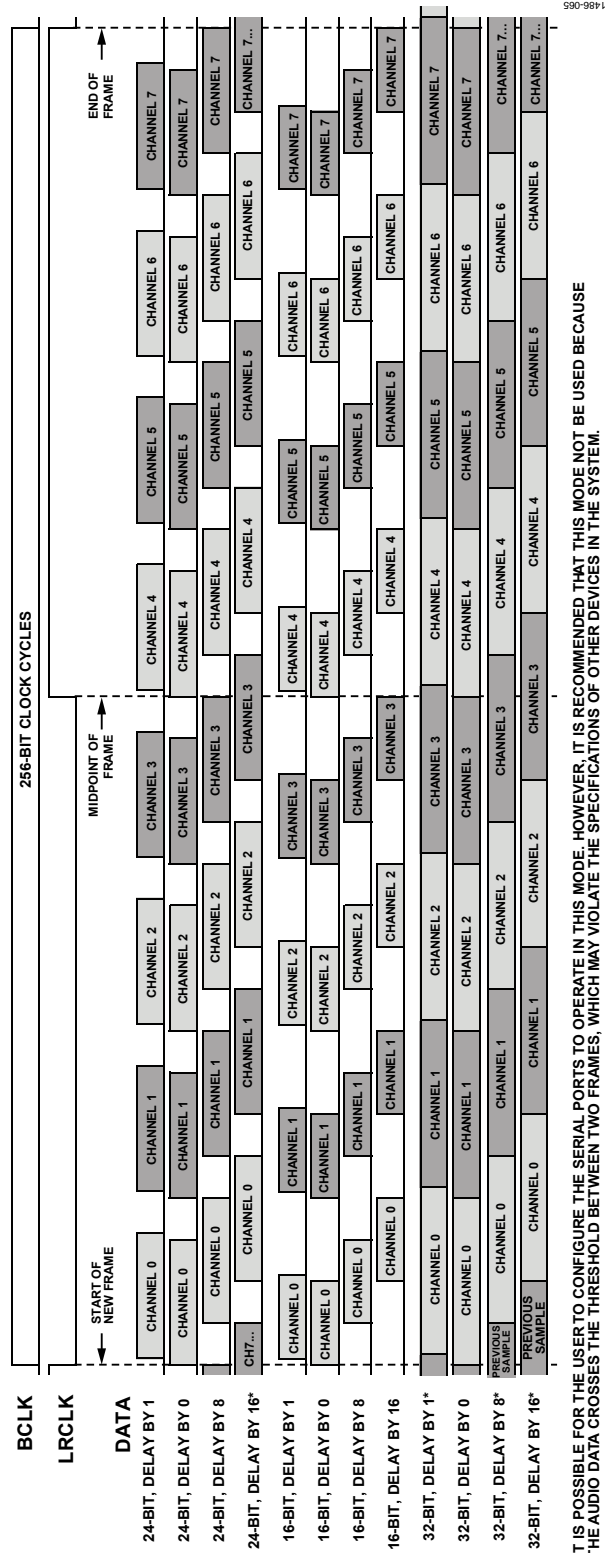


Figure 61. Serial Audio Data Formats; Eight Channels, 32 Bits per Channel

Figure 61 shows timing diagrams for possible serial port configurations in 8-channel mode, with 32 bit clock cycles per channel, for a total of 256 bit clock cycles per frame (refer to the SERIAL_BYTE_x_0 registers, Bits[2:0] (TDM_MODE) = 0b010). The bit clock signal is omitted from the figure.

Excluding flexible TDM mode, there are 12 possible combinations of settings for the audio word length (SERIAL_BYTE_x_0, Bits[6:5] (WORD_LEN)) and MSB position (SERIAL_BYTE_x_0, Bits[4:3] (DATA_FMT)), all of which are shown in Figure 61.

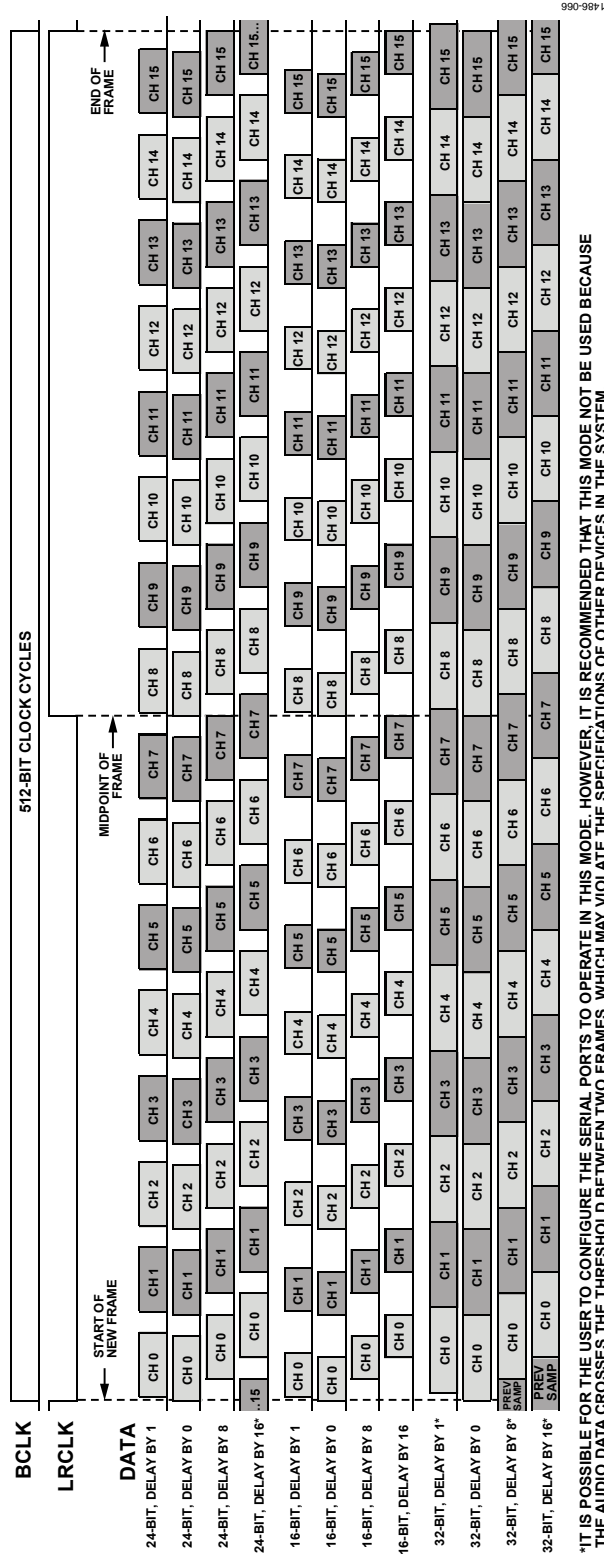


Figure 62. Serial Audio Data Formats; 16 Channels, 32 Bits per Channel

Figure 62 shows some timing diagrams for possible serial port configurations in 16-channel mode, with 32 bit clock cycles per channel, for a total of 512 bit clock cycles per frame (refer to the SERIAL_BYTE_x_0 registers, Bits[2:0] (TDM_MODE) = 0b011). The bit clock signal is omitted from the figure.

Excluding flexible TDM mode, there are 12 possible combinations of settings for the audio word length (SERIAL_BYTE_x_0, Bits[6:5] (WORD_LEN)) and MSB position (SERIAL_BYTE_x_0, Bits[4:3] (DATA_FMT)), all of which are shown in Figure 62.

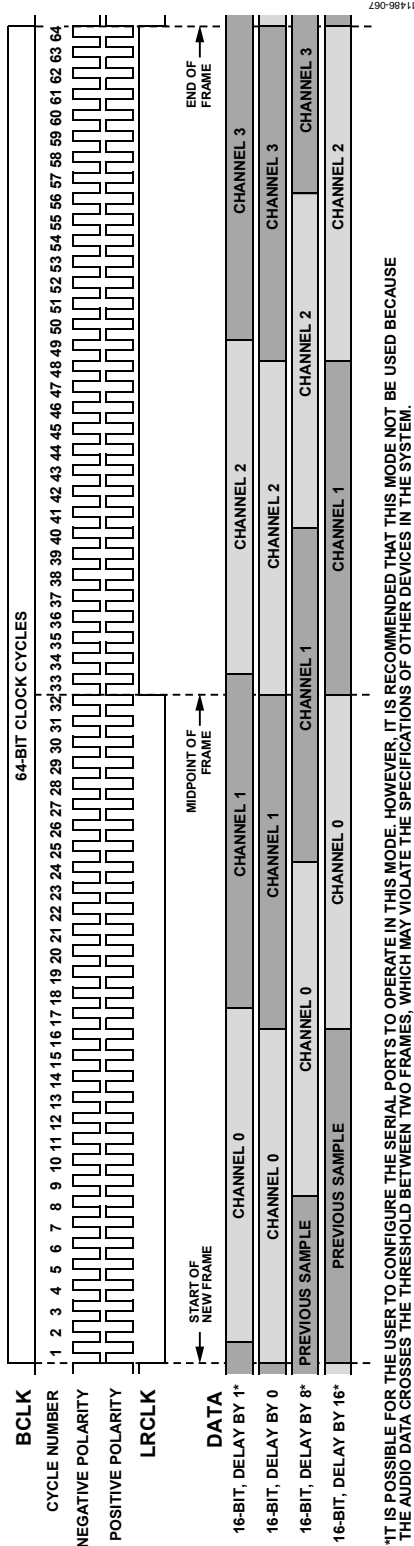
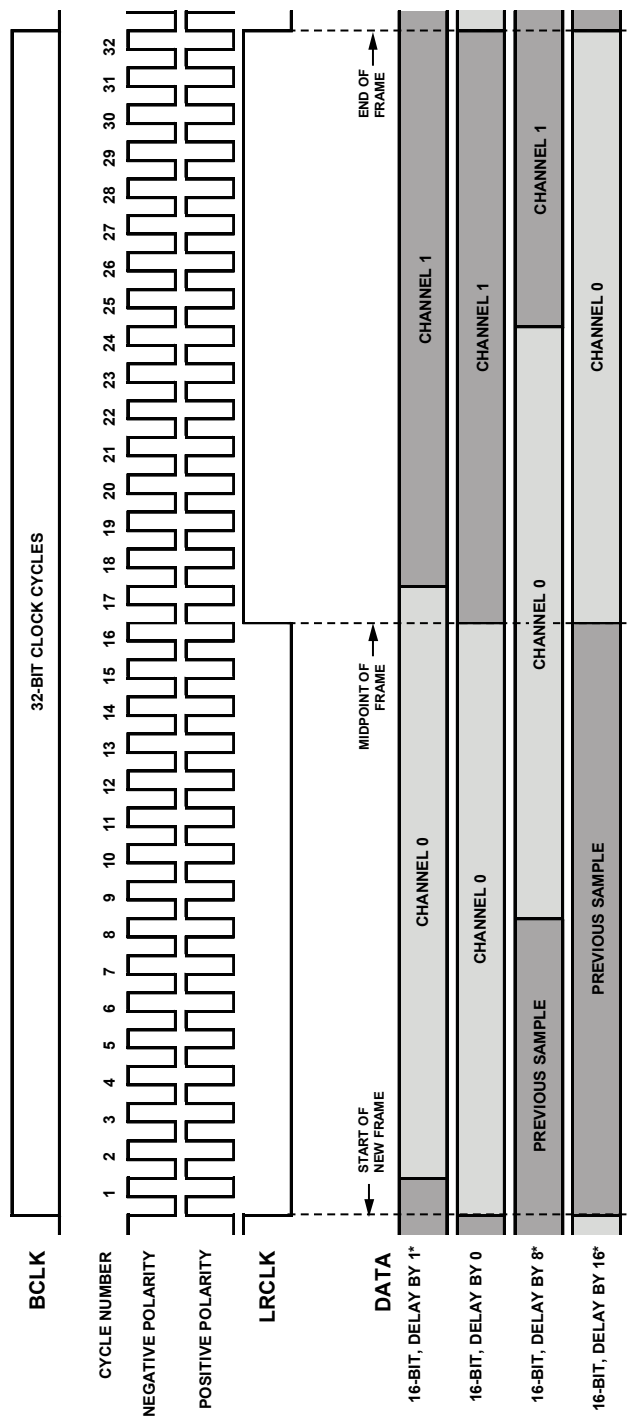


Figure 63. Serial Audio Data Formats; Four Channels, 16 Bits per Channel

Figure 63 shows some 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 63.



*IT IS POSSIBLE FOR THE USER TO CONFIGURE THE SERIAL PORTS TO OPERATE IN THIS MODE. HOWEVER, IT IS RECOMMENDED THAT THIS MODE NOT BE USED BECAUSE THE AUDIO DATA CROSSES THE THRESHOLD BETWEEN TWO FRAMES, WHICH MAY VIOLATE THE SPECIFICATIONS OF OTHER DEVICES IN THE SYSTEM.

Figure 64. Serial Audio Data Formats; Two Channels, 16 Bits per Channel

Figure 64 shows some timing diagrams for possible serial port configurations in two channel mode, with 16 bit clock cycles per channel, for a total of 32 bit clock cycles per frame (refer to the SERIAL_BYTE_x_0 registers, Register 0xF200 to Register 0xF21C, Bits[2:0] (TDM_MODE) = 0b101).

Different bit clock polarities are illustrated (SERIAL_BYTE_x_0, Bit 7 (BCLK_POL)). The audio word length is fixed at 16 bits (SERIAL_BYTE_x_0, 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 64.

Serial Clock Domains

There are four input clock domains and four output clock domains. A clock domain consists of a pair of LRCLK_OUT_x and LRCLK_IN_x (frame clock) and BCLK_OUT_x and BCLK_IN_x (bit clock) pins, which are used to synchronize the transmission of audio data to and from the device. There are eight total clock domains. Four of them are input domains and four of them are output domains. In master mode (refer to the SERIAL_BYTE_x_0 registers, Register 0xF200 to Register 0xF21C, Bits[15:13] (LRCLK_SRC) = 0b100 and Bits[12:10] (BCLK_SRC) = 0b100), each clock domain corresponds to exactly one serial data pin, one frame clock

pin, and one bit clock pin. Any serial data input can be clocked by any input clock domains when it is configured in slave mode (refer to the SERIAL_BYTE_x_0 registers, Bits[15:13] (LRCLK_SRC), which can be set to 0b000, 0b001, 0b010, or 0b011; and Bits[12:10] (BCLK_SRC), which can be set to 0b000, 0b001, 0b010, or 0b011). Any serial data output can be clocked by any output clock domain when it is configured in slave mode (see the SERIAL_BYTE_x_0 registers, Bits[15:13] (LRCLK_SRC), which can be set to 0b000, 0b001, 0b010, or 0b011; and Bits[12:10] (BCLK_SRC), which can be set to 0b000, 0b001, 0b010, or 0b011).

Table 43. Relationship Between Serial Data Pins and Clock Pins in Master or Slave Mode

| Serial Data Pin | Corresponding Clock Pins in Master Mode | Corresponding Clock Pins in Slave Mode |
|-----------------|---|---|
| SDATA_IN0 | BCLK_IN0, LRCLK_IN0 (LRCLK_IN0/MP10) | BCLK_IN0, LRCLK_IN0 or BCLK_IN1, LRCLK_IN1 or BCLK_IN2, LRCLK_IN2 or BCLK_IN3, LRCLK_IN3 |
| SDATA_IN1 | BCLK_IN1, LRCLK_IN1 (LRCLK_IN1/MP11) | BCLK_IN0, LRCLK_IN0 or BCLK_IN1, LRCLK_IN1 or BCLK_IN2, LRCLK_IN2 or BCLK_IN3, LRCLK_IN3 |
| SDATA_IN2 | BCLK_IN2, LRCLK_IN2 (LRCLK_IN2/MP12) | BCLK_IN0, LRCLK_IN0 or BCLK_IN1, LRCLK_IN1 or BCLK_IN2, LRCLK_IN2 or BCLK_IN3, LRCLK_IN3 |
| SDATA_IN3 | BCLK_IN3, LRCLK_IN3 (LRCLK_IN3/MP13) | BCLK_IN0, LRCLK_IN0 or BCLK_IN1, LRCLK_IN1 or BCLK_IN2, LRCLK_IN2 or BCLK_IN3, LRCLK_IN3 |
| SDATA_OUT0 | BCLK_OUT0, LRCLK_OUT0 (LRCLK_OUT0/MP4) | BCLK_OUT0, LRCLK_OUT0 or BCLK_OUT1, LRCLK_OUT1 or BCLK_OUT2, LRCLK_OUT2 or BCLK_OUT3, LRCLK_OUT3 |
| SDATA_OUT1 | BCLK_OUT1, LRCLK_OUT1 (LRCLK_OUT1/MP5) | BCLK_OUT0, LRCLK_OUT0 or BCLK_OUT1, LRCLK_OUT1 or BCLK_OUT2, LRCLK_OUT2 or BCLK_OUT3, LRCLK_OUT3 |
| SDATA_OUT2 | BCLK_OUT2, LRCLK_OUT2 (LRCLK_OUT2/MP8) | BCLK_OUT0, LRCLK_OUT0 or BCLK_OUT1, LRCLK_OUT1 or BCLK_OUT2, LRCLK_OUT2 or BCLK_OUT3, LRCLK_OUT3 |
| SDATA_OUT3 | BCLK_OUT3, LRCLK_OUT3 (LRCLK_OUT3/MP9) | BCLK_OUT0, LRCLK_OUT0 or BCLK_OUT1, LRCLK_OUT1 or BCLK_OUT2, LRCLK_OUT2 or BCLK_OUT3, LRCLK_OUT3 |

Serial Input Ports

There is a one-to-one mapping between the serial input ports and the audio input channels in the DSP and the ASRC input selectors, which is described in Table 44.

Table 44. Relationship Between Serial Input Port and Corresponding Channel Numbers on the DSP and ASRC Inputs

| Serial Port | Audio Input Channels in the DSP and ASRC |
|----------------|--|
| Serial Input 0 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 |
| Serial Input 1 | 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 |
| Serial Input 2 | 32, 33, 34, 35, 36, 37, 38, 39 |
| Serial Input 3 | 40, 41, 42, 43, 44, 45, 46, 47 |

If a serial input port is configured using the SERIAL_BYTE_x_0 registers, Bits[2:0] (TDM_MODE) for a number of channels that is less than its maximum channel count, the unused channels carry zero data. For example, if Serial Input 0 is set in 8-channel (TDM8) mode, the first eight channels (Channel 0 to Channel 7) carry data; and the unused channels (Channel 8 to Channel 15) carry no data.

There are four options for the word length of each serial input port: 24 bits, 16 bits, 32 bits, or flexible TDM. The flexible TDM option is described in the Flexible TDM Input section.

In 32-bit mode (see Figure 65), the 32 bits received on the serial input are mapped directly to a 32-bit word in the DSP core. To use 32-bit mode, the special 32-bit input cells must be used in SigmaStudio.

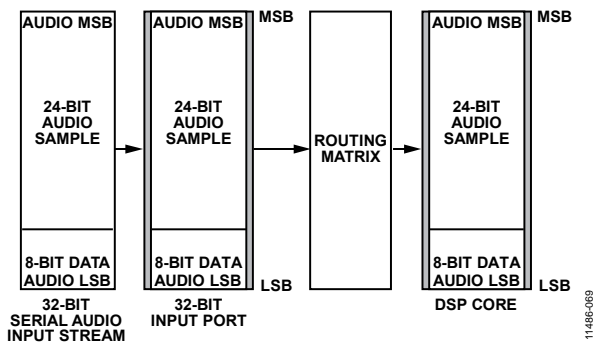


Figure 65. 32-Bit Serial Input Example

In 24-bit mode (see Figure 67), the 24-bit audio sample (in 1.23 format) is padded with eight zeros below its LSB (in 1.31 format) as it is input to the routing matrix. Then, the audio data is shifted such that the audio sample has seven sign-extended zeros on top, one padded zero on the bottom, and 24 bits of data in the middle (8.24 format).

Sixteen-bit mode is similar to 24-bit mode, but the 16-bit audio data has 16 zeros below its LSB instead of just eight zeros (in the 24-bit case). The resulting 8.24 sample, therefore, has seven sign-extended zeros on top, nine padded zeros on the bottom, and 16 bits of data in the middle (8.24 format).

Serial Output Ports

There is a one-to-one mapping between the serial output ports and the output audio channels in the DSP (see Table 45).

Table 45. Relationship Between Serial Input Port and Corresponding DSP Output Channel Numbers

| Serial Input Port | Audio Output Channels from the DSP |
|-------------------|--|
| Serial Output 0 | 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 |
| Serial Output 1 | 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 |
| Serial Output 2 | 32, 33, 34, 35, 36, 37, 38, 39 |
| Serial Output 3 | 40, 41, 42, 43, 44, 45, 46, 47 |

If a serial output port is configured using the SERIAL_BYTE_x_0 registers, Bits[2:0] (TDM_MODE), for a number of channels that is less than its maximum channel count, the unused channels are ignored. For example, if Serial Output Port 0 is set in 8-channel (TDM8) mode, and data is routed to it from the DSP, the first eight DSP output channels (Channel 0 through Channel 7) are output on SDATA_OUT0, but the remaining channels (Channel 8 through Channel 15) are not output from the device.

There are four options for the word length of each serial output port: 24 bits, 16 bits, 32 bits, or flexible TDM. See the Flexible TDM Output section for more information.

In 32-bit mode (see Figure 66), all 32 bits from the 8.24 word in the DSP core are copied directly to the serial output. To use 32-bit mode, the special 32-bit output cells must be used in SigmaStudio.

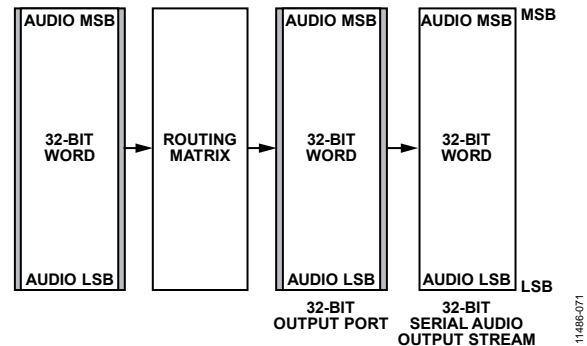


Figure 66. 32-Bit Serial Output Example

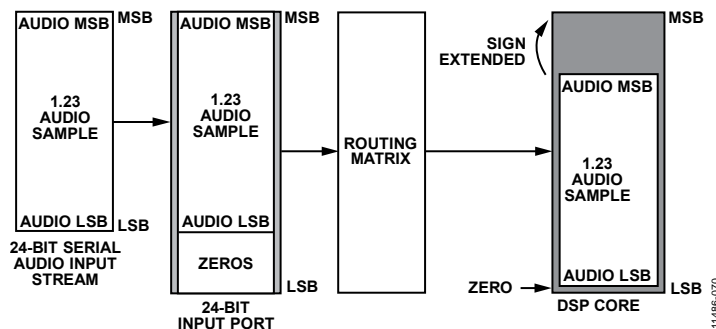


Figure 67. 24-Bit Serial Input Example

In 24-bit mode, the top seven MSBs of the 8.24 audio word in the DSP core are saturated, and the resulting 1.23 word is output from the serial port, with eight zeros padded under the LSB (see Figure 68).

In 16-bit mode, the top seven MSBs of the 8.24 audio word in the DSP core are saturated, and the resulting 1.23 word is then truncated to a 1.15 word by removing the eight LSBs. The

resulting 1.15 word is then zero padded with 16 zeros under the LSB and output from the serial port.

Serial Port Registers

An overview of the registers related to the serial ports is shown in Table 46. For a more detailed description, see the Serial Port Configuration Registers section.

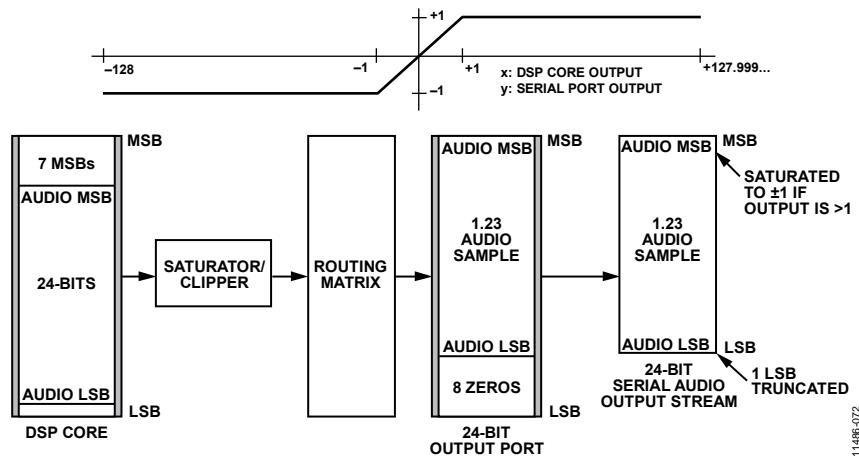


Figure 68. 24-Bit Serial Output Example

Table 46. Serial Port Registers

| Address | Register | Description |
|---------|-----------------|--|
| 0xF200 | SERIAL_BYTE_0_0 | Serial Port Control 0 (SDATA_IN0 pin) |
| 0xF201 | SERIAL_BYTE_0_1 | Serial Port Control 1 (SDATA_IN0 pin) |
| 0xF204 | SERIAL_BYTE_1_0 | Serial Port Control 0 (SDATA_IN1 pin) |
| 0xF205 | SERIAL_BYTE_1_1 | Serial Port Control 1 (SDATA_IN1 pin) |
| 0xF208 | SERIAL_BYTE_2_0 | Serial Port Control 0 (SDATA_IN2 pin) |
| 0xF209 | SERIAL_BYTE_2_1 | Serial Port Control 1 (SDATA_IN2 pin) |
| 0xF20C | SERIAL_BYTE_3_0 | Serial Port Control 0 (SDATA_IN3 pin) |
| 0xF20D | SERIAL_BYTE_3_1 | Serial Port Control 1 (SDATA_IN3 pin) |
| 0xF210 | SERIAL_BYTE_4_0 | Serial Port Control 0 (SDATA_OUT0 pin) |
| 0xF211 | SERIAL_BYTE_4_1 | Serial Port Control 1 (SDATA_OUT0 pin) |
| 0xF214 | SERIAL_BYTE_5_0 | Serial Port Control 0 (SDATA_OUT1 pin) |
| 0xF215 | SERIAL_BYTE_5_1 | Serial Port Control 1 (SDATA_OUT1 pin) |
| 0xF218 | SERIAL_BYTE_6_0 | Serial Port Control 0 (SDATA_OUT2 pin) |
| 0xF219 | SERIAL_BYTE_6_1 | Serial Port Control 1 (SDATA_OUT2 pin) |
| 0xF21C | SERIAL_BYTE_7_0 | Serial Port Control 0 (SDATA_OUT3 pin) |
| 0xF21D | SERIAL_BYTE_7_1 | Serial Port Control 1 (SDATA_OUT3 pin) |

FLEXIBLE TDM INTERFACE

The flexible TDM interface is available as an optional mode of operation on the SDATA_IN2 and SDATA_IN3 serial input ports, as well as on the SDATA_OUT2 and SDATA_OUT3 serial output ports. To use flexible TDM mode, the corresponding serial ports must be set in flexible TDM mode (SERIAL_BYTE_x_0 register, Bits[6:5] (WORD_LEN) = 0b11 and SERIAL_BYTE_x_0 register, Bits[2:0] = 0b010). Flexible TDM input mode requires that both SDATA_IN2 and SDATA_IN3 be configured for flexible TDM mode. Likewise, flexible TDM output mode requires that both SDATA_OUT2 and SDATA_OUT3 pins be configured for flexible TDM mode.

The flexible TDM interface provides byte addressable data placement in the input and output data streams on the corresponding serial data input/output pins. Each data stream is configured like a standard 8-channel TDM interface, with a total of 256 data bits (or 32 bytes) in the span of an audio frame. Because flexible TDM mode runs on two pins simultaneously, and each pin has 32 bytes of data, this means that there are a total of 64 data bytes. In flexible TDM input mode, each input channel inside the device can select its source data from any of the 64 input data bytes. In flexible TDM output mode, any serial output channel can be routed to any of the 64 output data bytes.

Flexible TDM Input

In flexible TDM input mode, two 256-bit data streams are input to the SDATA_IN2 and SDATA_IN3 pins. These 256 bits of data compose eight channels of four bytes each, for a total of 32 bytes on each pin, and a total of 64 bytes when both input pins are combined. The flexible TDM input functional block routes the desired input byte to a given byte in the serial input channels. Those serial input channels are then available as normal audio data in the audio routing matrix. The data can be passed to the DSP core, the ASRC inputs, or the serial outputs as needed.

There are a total of 64 control registers (FTDM_INx) that can be configured to set up the mapping of input data bytes to the corresponding bytes in the serial input channels. Each byte in each serial input channel has a corresponding control register, which selects the incoming data byte on the serial input pins that must be mapped to it. Figure 69 shows, from left to right, the data streams entering the serial input pins, the serial input channels, and the registers (see FTDM_INx, Register 0xF300 to Register 0xF33F) that correspond to each byte in the serial input channels.

Flexible TDM Output

In flexible TDM output mode, two 256-bit data streams are output from the SDATA_OUT2 and SDATA_OUT3 pins. These 256 bits of data compose eight channels of four bytes each, for a total of 32 bytes on each pin, and a total of 64 bytes when both input pins are combined. The flexible TDM output functional block routes the desired byte from the desired serial output channel to a given byte in the output streams. The serial output channels originate from the audio routing matrix, which is configured using the SOUT_SOURCEx control registers.

There are a total of 64 control registers (see FTDM_OUTx, Register 0xF3880 to Register 0xF3BF) that can be configured to set up the mapping of the bytes in the serial output channels and the bytes in the data streams exiting the serial output pins. Each byte in the data streams being output from the serial output pins has a corresponding control register, which selects the desired byte from the desired serial output channel. Figure 70 shows, from left to right, the serial output channels originating from the routing matrix, the serial output pins and data streams, and the control registers (FTDM_OUTx) that correspond to each byte in the serial output data streams.

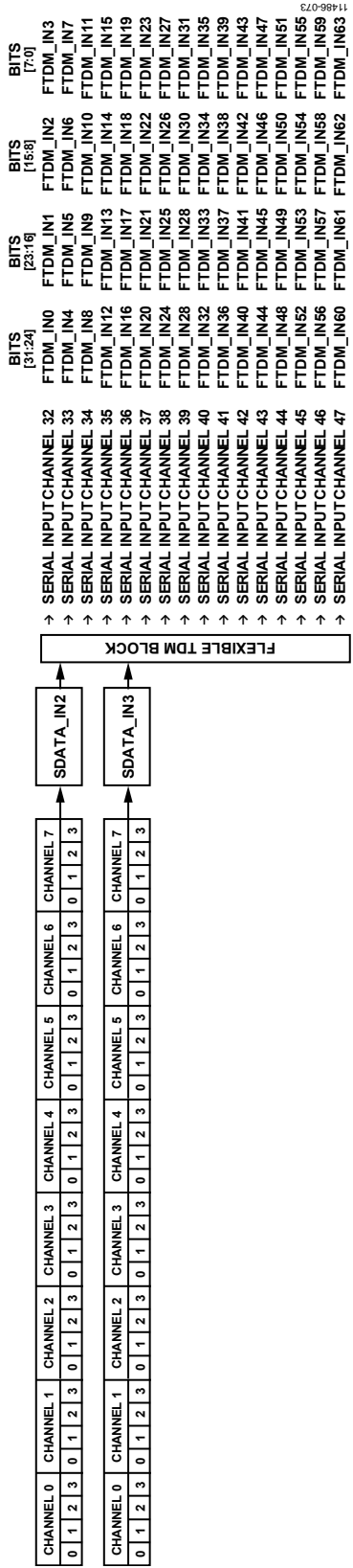


Figure 69. Flexible TDM Input Mapping

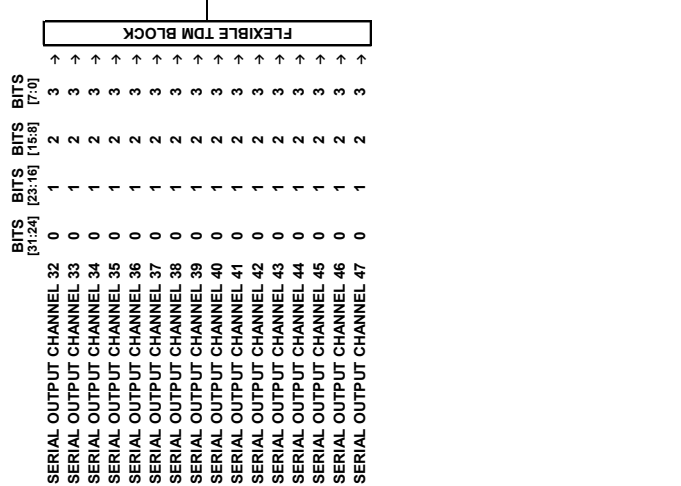


Figure 70. Flexible TDM Output Mapping

11486-074

Flexible TDM Registers

An overview of the registers related to the flexible TDM interface is shown in Table 47. For a more detailed description, see the Flexible TDM Interface Registers section.

Table 47. Flexible TDM Registers

| Address | Register | Description |
|---------|-----------|--|
| 0xF300 | FTDM_IN0 | FTDM mapping for the serial inputs (Channel 32, Bits[31:24]) |
| 0xF301 | FTDM_IN1 | FTDM mapping for the serial inputs (Channel 32, Bits[23:16]) |
| 0xF302 | FTDM_IN2 | FTDM mapping for the serial inputs (Channel 32, Bits[15:8]) |
| 0xF303 | FTDM_IN3 | FTDM mapping for the serial inputs (Channel 32, Bits[7:0]) |
| 0xF304 | FTDM_IN4 | FTDM mapping for the serial inputs (Channel 33, Bits[31:24]) |
| 0xF305 | FTDM_IN5 | FTDM mapping for the serial inputs (Channel 33, Bits[23:16]) |
| 0xF306 | FTDM_IN6 | FTDM mapping for the serial inputs (Channel 33, Bits[15:8]) |
| 0xF307 | FTDM_IN7 | FTDM mapping for the serial inputs Channel 33, Bits[7:0]) |
| 0xF308 | FTDM_IN8 | FTDM mapping for the serial inputs (Channel 34, Bits[31:24]) |
| 0xF309 | FTDM_IN9 | FTDM mapping for the serial inputs (Channel 34, Bits[23:16]) |
| 0xF30A | FTDM_IN10 | FTDM mapping for the serial inputs (Channel 34, Bits[15:8]) |
| 0xF30B | FTDM_IN11 | FTDM mapping for the serial inputs (Channel 34, Bits[7:0]) |
| 0xF30C | FTDM_IN12 | FTDM mapping for the serial inputs (Channel 35, Bits[31:24]) |
| 0xF30D | FTDM_IN13 | FTDM mapping for the serial inputs (Channel 35, Bits[23:16]) |
| 0xF30E | FTDM_IN14 | FTDM mapping for the serial inputs (Channel 35, Bits[15:8]) |
| 0xF30F | FTDM_IN15 | FTDM mapping for the serial inputs (Channel 35, Bits[7:0]) |
| 0xF310 | FTDM_IN16 | FTDM mapping for the serial inputs (Channel 36, Bits[31:24]) |
| 0xF311 | FTDM_IN17 | FTDM mapping for the serial inputs (Channel 36, Bits[23:16]) |
| 0xF312 | FTDM_IN18 | FTDM mapping for the serial inputs (Channel 36, Bits[15:8]) |
| 0xF313 | FTDM_IN19 | FTDM mapping for the serial inputs (Channel 36, Bits[7:0]) |
| 0xF314 | FTDM_IN20 | FTDM mapping for the serial inputs (Channel 37, Bits[31:24]) |
| 0xF315 | FTDM_IN21 | FTDM mapping for the serial inputs (Channel 37, Bits[23:16]) |
| 0xF316 | FTDM_IN22 | FTDM mapping for the serial inputs (Channel 37, Bits[15:8]) |
| 0xF317 | FTDM_IN23 | FTDM mapping for the serial inputs (Channel 37, Bits[7:0]) |
| 0xF318 | FTDM_IN24 | FTDM mapping for the serial inputs (Channel 38, Bits[31:24]) |
| 0xF319 | FTDM_IN25 | FTDM mapping for the serial inputs (Channel 38, Bits[23:16]) |
| 0xF31A | FTDM_IN26 | FTDM mapping for the serial inputs (Channel 38, Bits[15:8]) |
| 0xF31B | FTDM_IN27 | FTDM mapping for the serial inputs (Channel 38, Bits[7:0]) |
| 0xF31C | FTDM_IN28 | FTDM mapping for the serial inputs (Channel 39, Bits[31:24]) |
| 0xF31D | FTDM_IN29 | FTDM mapping for the serial inputs (Channel 39, Bits[23:16]) |
| 0xF31E | FTDM_IN30 | FTDM mapping for the serial inputs (Channel 39, Bits[15:8]) |
| 0xF31F | FTDM_IN31 | FTDM mapping for the serial inputs (Channel 39, Bits[7:0]) |
| 0xF320 | FTDM_IN32 | FTDM mapping for the serial inputs (Channel 40, Bits[31:24]) |
| 0xF321 | FTDM_IN33 | FTDM mapping for the serial inputs (Channel 40, Bits[23:16]) |
| 0xF322 | FTDM_IN34 | FTDM mapping for the serial inputs (Channel 40, Bits[15:8]) |
| 0xF323 | FTDM_IN35 | FTDM mapping for the serial inputs (Channel 40, Bits[7:0]) |
| 0xF324 | FTDM_IN36 | FTDM mapping for the serial inputs (Channel 41, Bits[31:24]) |
| 0xF325 | FTDM_IN37 | FTDM mapping for the serial inputs (Channel 41, Bits[23:16]) |
| 0xF326 | FTDM_IN38 | FTDM mapping for the serial inputs (Channel 41, Bits[15:8]) |
| 0xF327 | FTDM_IN39 | FTDM mapping for the serial inputs (Channel 41, Bits[7:0]) |
| 0xF328 | FTDM_IN40 | FTDM mapping for the serial inputs (Channel 42, Bits[31:24]) |
| 0xF329 | FTDM_IN41 | FTDM mapping for the serial inputs (Channel 42, Bits[23:16]) |
| 0xF32A | FTDM_IN42 | FTDM mapping for the serial inputs (Channel 42, Bits[15:8]) |
| 0xF32B | FTDM_IN43 | FTDM mapping for the serial inputs (Channel 42, Bits[7:0]) |
| 0xF32C | FTDM_IN44 | FTDM mapping for the serial inputs (Channel 43, Bits[31:24]) |
| 0xF32D | FTDM_IN45 | FTDM mapping for the serial inputs (Channel 43, Bits[23:16]) |

| Address | Register | Description |
|---------|------------|--|
| 0xF32E | FTDM_IN46 | FTDM mapping for the serial inputs (Channel 43, Bits[15:8]) |
| 0xF32F | FTDM_IN47 | FTDM mapping for the serial inputs (Channel 43, Bits[7:0]) |
| 0xF330 | FTDM_IN48 | FTDM mapping for the serial inputs (Channel 44, Bits[31:24]) |
| 0xF331 | FTDM_IN49 | FTDM mapping for the serial inputs (Channel 44, Bits[23:16]) |
| 0xF332 | FTDM_IN50 | FTDM mapping for the serial inputs (Channel 44, Bits[15:8]) |
| 0xF333 | FTDM_IN51 | FTDM mapping for the serial inputs (Channel 44, Bits[7:0]) |
| 0xF334 | FTDM_IN52 | FTDM mapping for the serial inputs (Channel 45, Bits[31:24]) |
| 0xF335 | FTDM_IN53 | FTDM mapping for the serial inputs (Channel 45, Bits[23:16]) |
| 0xF336 | FTDM_IN54 | FTDM mapping for the serial inputs (Channel 45, Bits[15:8]) |
| 0xF337 | FTDM_IN55 | FTDM mapping for the serial inputs (Channel 45, Bits[7:0]) |
| 0xF338 | FTDM_IN56 | FTDM mapping for the serial inputs (Channel 46, Bits[31:24]) |
| 0xF339 | FTDM_IN57 | FTDM mapping for the serial inputs (Channel 46, Bits[23:16]) |
| 0xF33A | FTDM_IN58 | FTDM mapping for the serial inputs (Channel 46, Bits[15:8]) |
| 0xF33B | FTDM_IN59 | FTDM mapping for the serial inputs (Channel 46, Bits[7:0]) |
| 0xF33C | FTDM_IN60 | FTDM mapping for the serial inputs (Channel 47, Bits[31:24]) |
| 0xF33D | FTDM_IN61 | FTDM mapping for the serial inputs (Channel 47, Bits[23:16]) |
| 0xF33E | FTDM_IN62 | FTDM mapping for the serial inputs (Channel 47, Bits[15:8]) |
| 0xF33F | FTDM_IN63 | FTDM mapping for the serial inputs (Channel 47, Bits[7:0]) |
| 0xF380 | FTDM_OUT0 | FTDM mapping for the serial outputs (Port 2, Channel 0, Bits[31:24]) |
| 0xF381 | FTDM_OUT1 | FTDM mapping for the serial outputs (Port 2, Channel 0, Bits[23:16]) |
| 0xF382 | FTDM_OUT2 | FTDM mapping for the serial outputs (Port 2, Channel 0, Bits[15:8]) |
| 0xF383 | FTDM_OUT3 | FTDM mapping for the serial outputs (Port 2, Channel 0, Bits[7:0]) |
| 0xF384 | FTDM_OUT4 | FTDM mapping for the serial outputs (Port 2, Channel 1, Bits[31:24]) |
| 0xF385 | FTDM_OUT5 | FTDM mapping for the serial outputs (Port 2, Channel 1, Bits[23:16]) |
| 0xF386 | FTDM_OUT6 | FTDM mapping for the serial outputs (Port 2, Channel 1, Bits[15:8]) |
| 0xF387 | FTDM_OUT7 | FTDM mapping for the serial outputs (Port 2, Channel 1, Bits[7:0]) |
| 0xF388 | FTDM_OUT8 | FTDM mapping for the serial outputs (Port 2, Channel 2, Bits[31:24]) |
| 0xF389 | FTDM_OUT9 | FTDM mapping for the serial outputs (Port 2, Channel 2, Bits[23:16]) |
| 0xF38A | FTDM_OUT10 | FTDM mapping for the serial outputs (Port 2, Channel 2, Bits[15:8]) |
| 0xF38B | FTDM_OUT11 | FTDM mapping for the serial outputs (Port 2, Channel 2, Bits[7:0]) |
| 0xF38C | FTDM_OUT12 | FTDM mapping for the serial outputs (Port 2, Channel 3, Bits[31:24]) |
| 0xF38D | FTDM_OUT13 | FTDM mapping for the serial outputs (Port 2, Channel 3, Bits[23:16]) |
| 0xF38E | FTDM_OUT14 | FTDM mapping for the serial outputs (Port 2, Channel 3, Bits[15:8]) |
| 0xF38F | FTDM_OUT15 | FTDM mapping for the serial outputs (Port 2, Channel 3, Bits[7:0]) |
| 0xF390 | FTDM_OUT16 | FTDM mapping for the serial outputs (Port 2, Channel 4, Bits[31:24]) |
| 0xF391 | FTDM_OUT17 | FTDM mapping for the serial outputs (Port 2, Channel 4, Bits[23:16]) |
| 0xF392 | FTDM_OUT18 | FTDM mapping for the serial outputs (Port 2, Channel 4, Bits[15:8]) |
| 0xF393 | FTDM_OUT19 | FTDM mapping for the serial outputs (Port 2, Channel 4, Bits[7:0]) |
| 0xF394 | FTDM_OUT20 | FTDM mapping for the serial outputs (Port 2, Channel 5, Bits[31:24]) |
| 0xF395 | FTDM_OUT21 | FTDM mapping for the serial outputs (Port 2, Channel 5, Bits[23:16]) |
| 0xF396 | FTDM_OUT22 | FTDM mapping for the serial outputs (Port 2, Channel 5, Bits[15:8]) |
| 0xF397 | FTDM_OUT23 | FTDM mapping for the serial outputs (Port 2, Channel 5, Bits[7:0]) |
| 0xF398 | FTDM_OUT24 | FTDM mapping for the serial outputs (Port 2, Channel 6, Bits[31:24]) |
| 0xF399 | FTDM_OUT25 | FTDM mapping for the serial outputs (Port 2, Channel 6, Bits[23:16]) |
| 0xF39A | FTDM_OUT26 | FTDM mapping for the serial outputs (Port 2, Channel 6, Bits[15:8]) |
| 0xF39B | FTDM_OUT27 | FTDM mapping for the serial outputs (Port 2, Channel 6, Bits[7:0]) |
| 0xF39C | FTDM_OUT28 | FTDM mapping for the serial outputs (Port 2, Channel 7, Bits[31:24]) |
| 0xF39D | FTDM_OUT29 | FTDM mapping for the serial outputs (Port 2, Channel 7, Bits[23:16]) |
| 0xF39E | FTDM_OUT30 | FTDM mapping for the serial outputs (Port 2, Channel 7, Bits[15:8]) |
| 0xF39F | FTDM_OUT31 | FTDM mapping for the serial outputs (Port 2, Channel 7, Bits[7:0]) |

| Address | Register | Description |
|---------|------------|--|
| 0xF3A0 | FTDM_OUT32 | FTDM mapping for the serial outputs (Port 3, Channel 0, Bits[31:24]) |
| 0xF3A1 | FTDM_OUT33 | FTDM mapping for the serial outputs (Port 3, Channel 0, Bits[23:16]) |
| 0xF3A2 | FTDM_OUT34 | FTDM mapping for the serial outputs (Port 3, Channel 0, Bits[15:8]) |
| 0xF3A3 | FTDM_OUT35 | FTDM mapping for the serial outputs (Port 3, Channel 0, Bits[7:0]) |
| 0xF3A4 | FTDM_OUT36 | FTDM mapping for the serial outputs (Port 3, Channel 1, Bits[31:24]) |
| 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]) |

ASYNCHRONOUS SAMPLE RATE CONVERTERS

Sixteen channels of integrated asynchronous sample rate converters are available in the ADAU1452. These sample rate converters are capable of receiving audio data input signals, along with their corresponding clocks, and resynchronizing the data stream to an arbitrary target sample rate. The sample rate converters use some filtering to accomplish this task; therefore, the data output from the sample rate converter is not a bit-accurate representation of the data input.

The 16 channels of sample rate converters are grouped into eight stereo sets. These eight stereo sample rate converters are individually configurable and are referred to as ASRC 0 through ASRC 7. Channel 0 and Channel 1 belong to ASRC 0, Channel 2 and Channel 3 belong to ASRC 1, Channel 4 and Channel 5 belong to ASRC 2, Channel 6 and Channel 7 belong to ASRC 3, Channel 8 and Channel 9 belong to ASRC 4, Channel 10 and Channel 11 belong to ASRC 5, Channel 12 and Channel 13 belong to ASRC 6, and Channel 14 and Channel 15 belong to ASRC 7.

Audio is routed to the sample rate converters using the ASRC_INPUTx registers, and the target sample rate of each ASRC is configured using the ASRC_OUT_RATEx registers. A complete description of audio routing is included in the Audio Signal Routing section.

Asynchronous Sample Rate Converter Group Delay

The group delay of the sample rate converter is dependent on the input and output sampling frequencies as described in the following equations:

For $f_{S_OUT} > f_{S_IN}$,

$$GDS = \frac{16}{f_{S_IN}} + \frac{32}{f_{S_IN}}$$

For $f_{S_OUT} < f_{S_IN}$,

$$GDS = \frac{16}{f_{S_IN}} + \left(\frac{32}{f_{S_IN}} \right) \times \left(\frac{f_{S_IN}}{f_{S_OUT}} \right)$$

where GDS is the group delay in seconds.

ASRC Lock

Each ASRC monitors the incoming signal and attempts to lock on to the clock and data signals. When a valid signal is detected and several consecutive valid samples are received, and there is a valid output target sample rate, the corresponding bit in Register 0xF580 (ASRC_LOCK) signifies that the ASRC has successfully locked to the incoming signal.

ASRC Muting

The ASRC outputs can be manually muted at any time using the corresponding bits in Register 0xF581 (ASRC_MUTE). However, for creating a smooth volume ramp when muting audio signals, more options are available in the DSP core; therefore, in most cases, using the DSP program to manually mute signals is preferable to using Register 0xF581.

Asynchronous Sample Rate Converters Registers

An overview of the registers related to the ASRCs is shown in Table 48. For a more detailed description, refer to the ASRC Status and Control Registers section.

Table 48. Asynchronous Sample Rate Converters Registers

| Address | Register | Description |
|---------|-------------|--|
| 0xF580 | ASRC_LOCK | ASRC lock status |
| 0xF581 | ASRC_MUTE | ASRC mute |
| 0xF582 | ASRC0_RATIO | ASRC ratio (ASRC 0, Channel 0 and Channel 1) |
| 0xF583 | ASRC1_RATIO | ASRC ratio (ASRC 1, Channel 2 and Channel 3) |
| 0xF584 | ASRC2_RATIO | ASRC ratio (ASRC 2, Channel 4 and Channel 5) |
| 0xF585 | ASRC3_RATIO | ASRC ratio (ASRC 3, Channel 6 and Channel 7) |
| 0xF586 | ASRC4_RATIO | ASRC ratio (ASRC 4, Channel 8 and Channel 9) |
| 0xF587 | ASRC5_RATIO | ASRC ratio (ASRC 5, Channel 10 and Channel 11) |
| 0xF588 | ASRC6_RATIO | ASRC ratio (ASRC 6, Channel 12 and Channel 13) |
| 0xF589 | ASRC7_RATIO | ASRC ratio (ASRC 7, Channel 14 and Channel 15) |

S/PDIF INTERFACE

For easy interfacing on 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 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 at a wide range of sampling frequencies between 18 kHz and 192 kHz.

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 I²C 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 AES and 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 49.

Table 49. 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 (0b0 = AC3, 0b1 = DTS) and the audio type bit (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 |

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 50-50 duty cycle square wave that has the same frequency as the audio sample rate (see Table 125).

S/PDIF Interface Registers

An overview of the registers related to the S/PDIF interface is shown in Table 50. For a more detailed description, refer to the S/PDIF Interface Registers section.

Table 50. S/PDIF Interface Registers

| Address | Register | Description |
|------------------|------------------------|--|
| 0xF600 | SPDIF_LOCK_DET | S/PDIF receiver lock bit detection |
| 0xF601 | SPDIF_RX_CTRL | S/PDIF receiver control |
| 0xF602 | SPDIF_RX_DECODE | Decoded signals from the S/PDIF receiver |
| 0xF603 | SPDIF_RX_COMPRMODE | Compression mode from the S/PDIF receiver |
| 0xF604 | SPDIF_RESTART | Automatically resume S/PDIF receiver audio input |
| 0xF605 | SPDIF_LOSS_OF_LOCK | S/PDIF receiver loss of lock detection |
| 0xF608 | SPDIF_AUX_EN | S/PDIF receiver auxiliary outputs enable |
| 0xF60F | SPDIF_RX_AUXBIT_READY | S/PDIF receiver auxiliary bits ready flag |
| 0xF610 to 0xF61B | SPDIF_RX_CS_LEFT_x | S/PDIF receiver channel status bits (left) |
| 0xF620 to 0xF62B | SPDIF_RX_CS_RIGHT_x | S/PDIF receiver channel status bits (right) |
| 0xF630 to 0xF63B | SPDIF_RX_UD_LEFT_x | S/PDIF receiver user data bits (left) |
| 0xF640 to 0xF64B | SPDIF_RX_UD_RIGHT_x | S/PDIF receiver user data bits (right) |
| 0xF650 to 0xF65B | SPDIF_RX_VB_LEFT_x | S/PDIF receiver validity bits (left) |
| 0xF660 to 0xF66B | SPDIF_RX_VB_RIGHT_x | S/PDIF receiver validity bits (right) |
| 0xF670 to 0xF67B | SPDIF_RX_PB_LEFT_x | S/PDIF receiver parity bits (left) |
| 0xF680 to 0xF68B | SPDIF_RX_PB_RIGHT_x | S/PDIF receiver parity bits (right) |
| 0xF690 | SPDIF_TX_EN | S/PDIF transmitter enable |
| 0xF691 | SPDIF_TX_CTRL | S/PDIF transmitter control |
| 0xF69F | SPDIF_TX_AUXBIT_SOURCE | S/PDIF transmitter auxiliary bits source select |
| 0xF6A0 to 0xF6AB | SPDIF_TX_CS_LEFT_x | S/PDIF transmitter channel status bits (left) |
| 0xF6B0 to 0xF6BB | SPDIF_TX_CS_RIGHT_x | S/PDIF transmitter channel status bits (right) |
| 0xF6C0 to 0xF6CB | SPDIF_TX_UD_LEFT_x | S/PDIF transmitter user data bits (left) |
| 0xF6D0 to 0xF6DB | SPDIF_TX_UD_RIGHT_x | S/PDIF transmitter user data bits (right) |
| 0xF6E0 to 0xF6EB | SPDIF_TX_VB_LEFT_x | S/PDIF transmitter validity bits (left) |
| 0xF6F0 to 0xF6FB | SPDIF_TX_VB_RIGHT_x | S/PDIF transmitter validity bits (right) |
| 0xF700 to 0xF70B | SPDIF_TX_PB_LEFT_x | S/PDIF transmitter parity bits (left) |
| 0xF710 to 0xF71B | SPDIF_TX_PB_RIGHT_x | S/PDIF transmitter parity bits (right) |

DIGITAL PDM MICROPHONE INTERFACE

Up to four pulse density modulation (PDM) microphones can be connected as audio inputs. Each pair of microphones can share a single data line; therefore, using four PDM microphones requires two GPIO pins. Any multipurpose pin can be used as a microphone data input, with up to two microphones connected to each pin. This configuration is set up using the corresponding MPx_MODE and DMIC_CTRLx registers.

A bit clock pin from one of the serial input clock domains (BCLK_INx) or one of the serial output clock domains (BCLK_OUTx) must be set up as a master clock source, and its output signal must be connected to the PDM microphones to provide them with a clock.

PDM microphones, such as the ADMP522, typically require a bit clock frequency in the range of 1 MHz to 3.3 MHz, corresponding to audio sample rates of 15.625 kHz to 51.5625 kHz. This means that the serial port corresponding to the BCLK_INx pin or BCLK_OUTx pin driving the PDM microphones should operate in 2-channel mode at a sample rate between 16 kHz and 48 kHz.

PDM microphone inputs are automatically routed through decimation filters and then are available for use at the DSP core, the ASRCs, and the serial output ports.

Figure 71 shows an example circuit with two ADMP522 PDM output MEMS microphones connected to the ADAU1452. Any of the BCLK_INx pins or BCLK_OUTx pins can be used to

provide a clock signal to the microphones, and the data output of the microphones can be connected to any MPx pin that has been configured as a PDM microphone data input.

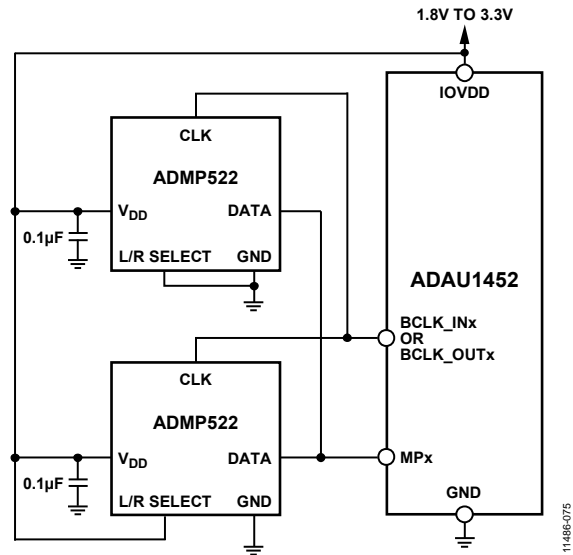


Figure 71. Example Stereo PDM Microphone Input Circuit

Digital PDM Microphone Interface Registers

An overview of the registers related to the digital microphone interface is shown in Table 51. For a more detailed description, see the Digital PDM Microphone Control Register.

Table 51. Digital PDM Microphone Interface Registers

| Address | Register | Description |
|---------|------------|--|
| 0xF560 | DMIC_CTRL0 | Digital PDM microphone control (Channel 0 and Channel 1) |
| 0xF561 | DMIC_CTRL1 | Digital PDM microphone control (Channel 2 and Channel 3) |

MULTIPURPOSE PINS

A total of 14 pins are available for use as general-purpose inputs or outputs (GPIO) that are multiplexed with other functions, such as clock inputs/outputs. Because these pins have multiple functions, they are referred to as multipurpose pins, or MPx pins.

Multipurpose pins can be configured in several modes using the MPx_MODE registers:

- Hardware input from pin
- Software input (written via I²C or SPI slave control port)
- Hardware output with internal pull-up
- Hardware output without internal pull-up
- PDM microphone data input
- Flag output from panic manager
- Slave select line for master SPI port

When configured in hardware input mode, a debounce circuit is available to avoid data glitches.

When operating in GPIO mode, pin status is updated once per sample. This means that the state of a GPIO (MPx pin) cannot change more than once in a sample period.

General-Purpose Inputs to the DSP Core

When a multipurpose pin is configured as a general-purpose input, its value can be used as a control logic signal in the DSP program, which is configured using SigmaStudio. Figure 72 shows the location of the general-purpose input cell within the SigmaStudio toolbox.

The 14 available general-purpose inputs in SigmaStudio map to the corresponding 14 multipurpose pins, but their data is valid only if the corresponding multipurpose pin has been configured as an input using the MPx_MODE registers. Figure 74 shows all of the general-purpose inputs as they appear in the SigmaStudio signal flow.

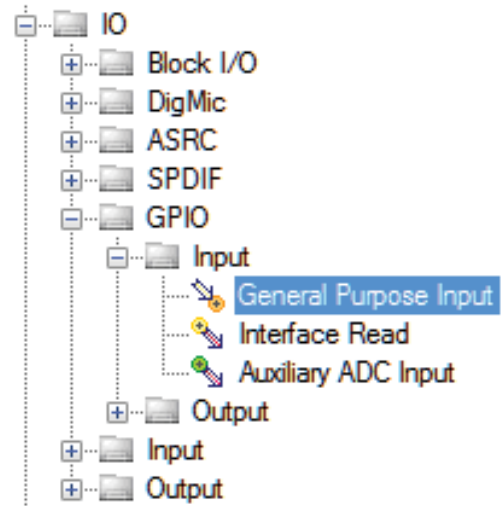


Figure 72. General-Purpose Input in the SigmaStudio Toolbox

General-Purpose Outputs from the DSP Core

When a multipurpose pin is configured as a general-purpose output, a Boolean value is output from the DSP program to the corresponding multipurpose pin. Figure 73 shows the location of the general-purpose output cell within the SigmaStudio toolbox.

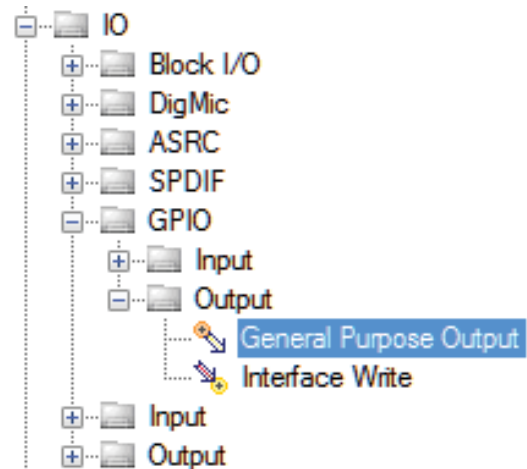


Figure 73. General-Purpose Output in the SigmaStudio Toolbox

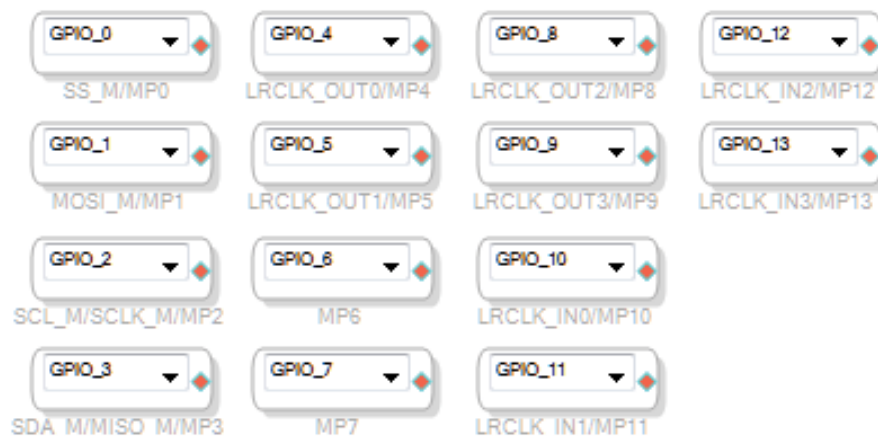


Figure 74. Complete Set of General-Purpose Inputs in SigmaStudio

The 14 available general-purpose outputs in SigmaStudio map to the corresponding 14 multipurpose pins, but their data is output to the pin only if the corresponding multipurpose pin is

configured as an output using the MPx_MODE registers. Figure 75 shows all of the general-purpose inputs as they appear in the SigmaStudio signal flow.

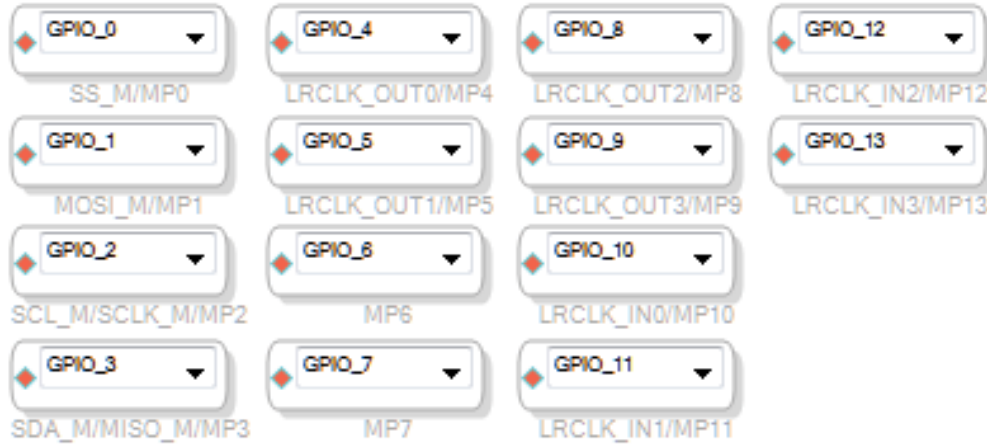


Figure 75. Complete Set of General-Purpose Outputs in SigmaStudio

1148E-079

Multipurpose Pin Registers

An overview of the registers related to GPIO is shown in Table 52. For a more detailed description, refer to the Multipurpose Pin Configuration Registers section.

Table 52. Multipurpose Pins Registers

| Address | Register | Description |
|---------|------------|---|
| 0xF510 | MP0_MODE | Multipurpose pin mode (SS_M/MP0) |
| 0xF511 | MP1_MODE | Multipurpose pin mode (MOSI_M/MP1) |
| 0xF512 | MP2_MODE | Multipurpose pin mode (SCL_M/SCLK_M/MP2) |
| 0xF513 | MP3_MODE | Multipurpose pin mode (SDA_M/MISO_M/MP3) |
| 0xF514 | MP4_MODE | Multipurpose pin mode (LRCLK_OUT0/MP4) |
| 0xF515 | MP5_MODE | Multipurpose pin mode (LRCLK_OUT1/MP5) |
| 0xF516 | MP6_MODE | Multipurpose pin mode (MP6) |
| 0xF517 | MP7_MODE | Multipurpose pin mode (MP7) |
| 0xF518 | MP8_MODE | Multipurpose pin mode (LRCLK_OUT2/MP8) |
| 0xF519 | MP9_MODE | Multipurpose pin mode (LRCLK_OUT3/MP9) |
| 0xF51A | MP10_MODE | Multipurpose pin mode (LRCLK_IN0/MP10) |
| 0xF51B | MP11_MODE | Multipurpose pin mode (LRCLK_IN1/MP11) |
| 0xF51C | MP12_MODE | Multipurpose pin mode (LRCLK_IN2/MP12) |
| 0xF51D | MP13_MODE | Multipurpose pin mode (LRCLK_IN3/MP13) |
| 0xF520 | MP0_WRITE | Multipurpose pin write value (SS_M/MP0) |
| 0xF521 | MP1_WRITE | Multipurpose pin write value (MOSI_M/MP1) |
| 0xF522 | MP2_WRITE | Multipurpose pin write value (SCL_M/SCLK_M/MP2) |
| 0xF523 | MP3_WRITE | Multipurpose pin write value (SDA_M/MISO_M/MP3) |
| 0xF524 | MP4_WRITE | Multipurpose pin write value (LRCLK_OUT0/MP4) |
| 0xF525 | MP5_WRITE | Multipurpose pin write value (LRCLK_OUT1/MP5) |
| 0xF526 | MP6_WRITE | Multipurpose pin write value (MP6) |
| 0xF527 | MP7_WRITE | Multipurpose pin write value (MP7) |
| 0xF528 | MP8_WRITE | Multipurpose pin write value (LRCLK_OUT2/MP8) |
| 0xF529 | MP9_WRITE | Multipurpose pin write value (LRCLK_OUT3/MP9) |
| 0xF52A | MP10_WRITE | Multipurpose pin write value (LRCLK_IN0/MP10) |
| 0xF52B | MP11_WRITE | Multipurpose pin write value (LRCLK_IN1/MP11) |
| 0xF52C | MP12_WRITE | Multipurpose pin write value (LRCLK_IN2/MP12) |
| 0xF52D | MP13_WRITE | Multipurpose pin write value (LRCLK_IN3/MP13) |
| 0xF530 | MP0_READ | Multipurpose pin read value (SS_M/MP0) |
| 0xF531 | MP1_READ | Multipurpose pin read value (MOSI_M/MP1) |
| 0xF532 | MP2_READ | Multipurpose pin read value (SCL_M/SCLK_M/MP2) |
| 0xF533 | MP3_READ | Multipurpose pin read value (SDA_M/MISO_M/MP3) |
| 0xF534 | MP4_READ | Multipurpose pin read value (LRCLK_OUT0/MP4) |
| 0xF535 | MP5_READ | Multipurpose pin read value (LRCLK_OUT1/MP5) |
| 0xF536 | MP6_READ | Multipurpose pin read value (MP6) |
| 0xF537 | MP7_READ | Multipurpose pin read value (MP7) |
| 0xF538 | MP8_READ | Multipurpose pin read value (LRCLK_OUT2/MP8) |
| 0xF539 | MP9_READ | Multipurpose pin read value (LRCLK_OUT3/MP9) |
| 0xF53A | MP10_READ | Multipurpose pin read value (LRCLK_IN0/MP10) |
| 0xF53B | MP11_READ | Multipurpose pin read value (LRCLK_IN1/MP11) |
| 0xF53C | MP12_READ | Multipurpose pin read value (LRCLK_IN2/MP12) |
| 0xF53D | MP13_READ | Multipurpose pin read value (LRCLK_IN3/MP13) |

AUXILIARY ADC

There are six auxiliary ADC inputs with 10 bits of accuracy. They are intended to be used as control signal inputs, such as potentiometer outputs or battery monitor signals.

The auxiliary ADC samples each channel at a frequency of the core system clock divided by 6144. In the case of a default clocking scheme, the system clock is 294.912 MHz; thus, the auxiliary ADC sample rate is 48 kHz. If the system clock is scaled down by configuring the PLL to generate a lower output frequency, the auxiliary ADC sample rate is scaled down proportionately.

The auxiliary ADC is referenced so that a full-scale input is achieved when the input voltage is equal to AVDD, and an input of zero is achieved when the input is connected to ground.

The input impedance of the auxiliary ADC is approximately 200 kΩ at dc (zero hertz).

Auxiliary ADC inputs can be used directly in the DSP program (as configured in the SigmaStudio software). The instantaneous value of each ADC is also available in the ADC_READx registers, which are accessible via the I²C or SPI slave control port.

Auxiliary ADC Inputs to the DSP Core

Auxiliary ADC inputs can be used as control signals in the DSP program as configured by SigmaStudio. Figure 76 shows the location of the auxiliary ADC input cell in the SigmaStudio toolbox.

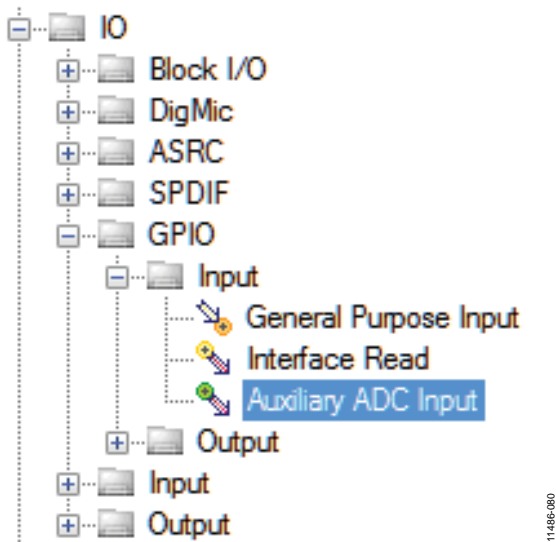


Figure 76. Auxiliary ADC Input Cell in the SigmaStudio Toolbox

The six auxiliary input pins map to the corresponding six auxiliary ADC input cells. Figure 77 shows the complete set of auxiliary ADC input cells in SigmaStudio.



Figure 77. Complete Set of Auxiliary ADC Inputs in SigmaStudio

Auxiliary ADC Registers

An overview of the registers related to the auxiliary ADC is shown in Table 53. For a more detailed description, see the Auxiliary ADC Registers section.

Table 53. Auxiliary ADC Registers

| Address | Register | Description |
|---------|-----------|------------------------------------|
| 0xF5A0 | ADC_READ0 | Auxiliary ADC read value (AUXADC0) |
| 0xF5A1 | ADC_READ1 | Auxiliary ADC read value (AUXADC1) |
| 0xF5A2 | ADC_READ2 | Auxiliary ADC read value (AUXADC2) |
| 0xF5A3 | ADC_READ3 | Auxiliary ADC read value (AUXADC3) |
| 0xF5A4 | ADC_READ4 | Auxiliary ADC read value (AUXADC4) |
| 0xF5A5 | ADC_READ5 | Auxiliary ADC read value (AUXADC5) |

SigmaDSP CORE

The SigmaDSP core operates at a maximum frequency of 294.912 MHz, which is equivalent to 6144 clock cycles per sample at a sample rate of 48 kHz. For a sample rate of 48 kHz, the largest program possible consists of 6144 program instructions per sample. If the system clock remains at 294.912 MHz but the audio frame rate of the DSP core is decreased, programs consisting of more than 6144 instructions per sample are possible. The program RAM is 8192 words long, so the largest program possible (but only at lower sample rates) is 8192 instructions per frame.

The core consists of four multipliers and two accumulators. At an operating frequency of 294.912 MHz, the core performs 1.2 billion MAC operations per second. At maximum efficiency, the core processes 3072 IIR biquad filters (single or double precision) per sample at a sample rate of 48 kHz. At maximum efficiency, the core processes approximately 24,000 FIR filter taps per sample at a sample rate of 48 kHz. The instruction set is a single instruction, multiple data (SIMD) computing model. The DSP core is 32-bit fixed-point, with an 8.24 data format for audio. The four multipliers are 64-bit double precision, capable of multiplying an 8.56 format number by an 8.24 number. The multiply accumulators consist of 16 registers, with a depth of 80 bits. The core can access RAM with a load/store width of 256 bits (eight 32-bit words per frame). The two ALUs have an 80-bit width and operate on numbers in 24.56 format. The 24.56 bit format provides more than 42 dB of headroom.

It is possible to create combinations of time domain and frequency domain processing, using block and sample frame interrupts. Sixteen data address generator (DAG) registers are available, and circular buffer addressing is possible.

Many of the signal processing functions are coded using full, 64-bit, double precision arithmetic. The serial port input and output word lengths are 24 bits, but eight extra headroom bits are used in the processor to allow internal gains of up to 48 dB without clipping. Additional gains can be achieved by initially scaling down the input signal in the DSP signal flow.

Numeric Formats

DSP systems commonly use a standard numeric format. Fractional number systems are specified by an A.B format, where A is the number of bits to the left of the decimal point and B is the number of bits to the right of the decimal point.

The same numeric format is used for both the parameter and data values.

A digital clipper circuit is used within the DSP core before outputting to the serial port outputs, ASRCs, and S/PDIF. This clips the top seven bits (and the least significant bit) of the signal to produce a 24-bit output with a range of +1.0 (minus 1 LSB) to -1.0. Figure 78 shows the maximum signal levels at each point in the data flow in both binary and decibel levels.

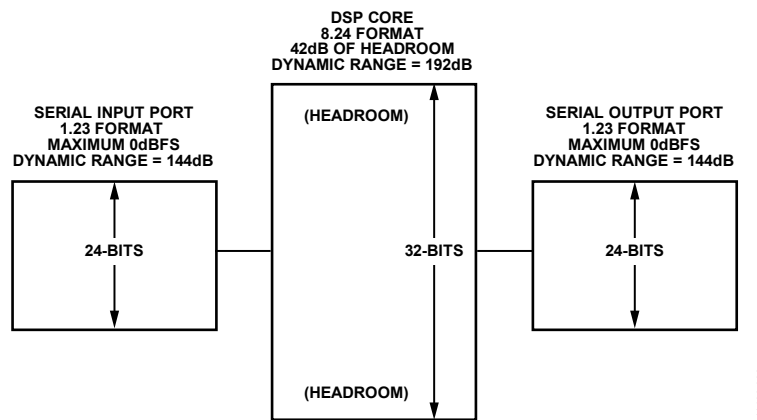


Figure 78. Signal Range for 1.23 Format (Serial Ports, ASRCs) and 8.24 Format (DSP Core)

Numerical Format: 8.24

Linear range: -128.0 to (+128.0 - 1 LSB)

Dynamic range (ratio of the largest possible signal level to the smallest possible non-zero signal level): 192 dB

Examples:

- 0b 1000 0000 0000 0000 0000 0000 0000 0000 = 0x80000000 = -128.0
- 0b 1110 0000 0000 0000 0000 0000 0000 0000 = 0xE0000000 = -32.0
- 0b 1111 1000 0000 0000 0000 0000 0000 0000 = 0xF8000000 = -8.0
- 0b 1111 1110 0000 0000 0000 0000 0000 0000 = 0xFE000000 = -2
- 0b 1111 1111 0000 0000 0000 0000 0000 0000 = 0xFF000000 = -1
- 0b 1111 1111 1000 0000 0000 0000 0000 0000 = 0xFF800000 = -0.5
- 0b 1111 1111 1110 0110 0110 0110 0110 0110 = 0xFFE66666 = -0.1
- 0b 1111 1111 1111 1111 1111 1111 1111 1111 = 0xFFFFFFFF = -0.00000005 (1 LSB below 0.0)
- 0b 0000 0000 0000 0000 0000 0000 0000 0000 = 0x00000000 = 0.0
- 0b 0000 0000 0000 0000 0000 0000 0000 0001 = 0x00000001 = 0.00000005 (1 LSB above 0.0)
- 0b 0000 0000 0001 1001 1001 1001 1001 1001 = 0x00199999 = 0.1
- 0b 0000 0000 0100 0000 0000 0000 0000 0000 = 0x00400000 = 0.25
- 0b 0000 0000 1000 0000 0000 0000 0000 0000 = 0x00800000 = 0.5
- 0b 0000 0001 0000 0000 0000 0000 0000 0000 = 0x01000000 = 1.0
- 0b 0000 0010 0000 0000 0000 0000 0000 0000 = 0x02000000 = 2.0
- 0b 0111 1111 1111 1111 1111 1111 1111 1111 = 0x7FFFFFFF = 127.99999994 (1 LSB below 128.0)

Numerical Format: 32.0

The 32.0 format is used for logic signals in the DSP program flow that are integers.

Linear range: -2,147,483,648 to +2,147,483,647

Dynamic range (ratio of the largest possible signal level to the smallest possible non-zero signal level): 192 dB

Examples:

```

0b 1000 0000 0000 0000 0000 0000 0000 0000 = 0x80000000 = -2147483648
0b 1000 0000 0000 0000 0000 0000 0000 0001 = 0x80000001 = -2147483647
0b 1000 0000 0000 0000 0000 0000 0000 0010 = 0x80000002 = -2147483646
0b 1100 0000 0000 0000 0000 0000 0000 0000 = 0xC0000000 = -1073741824
0b 1110 0000 0000 0000 0000 0000 0000 0000 = 0xE0000000 = -536870912
0b 1111 1111 1111 1111 1111 1111 1111 1100 = 0xFFFFFFF0 = -4
0b 1111 1111 1111 1111 1111 1111 1111 1110 = 0xFFFFFFF2 = -2
0b 1111 1111 1111 1111 1111 1111 1111 1111 = 0xFFFFFFF8 = -1
0b 0000 0000 0000 0000 0000 0000 0000 0000 = 0x00000000 = 0
0b 0000 0000 0000 0000 0000 0000 0000 0001 = 0x00000001 = 1
0b 0000 0000 0000 0000 0000 0000 0000 0010 = 0x00000002 = 2
0b 0000 0000 0000 0000 0000 0000 0000 0011 = 0x00000003 = 3
0b 0000 0000 0000 0000 0000 0000 0000 0100 = 0x00000004 = 4
0b 0111 1111 1111 1111 1111 1111 1111 1110 = 0x7FFFFFFE = 2147483646
0b 0111 1111 1111 1111 1111 1111 1111 1111 = 0x7FFFFFFF = 2147483647
    
```

Hardware Accelerators

The core includes accelerators like division, square root, barrel shifters, base-2 logarithm, base-2 exponential, slew, and a pseudo-random number generator. This reduces 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 for click-free updates of parameters that must change slowly over time, allowing for audio processing algorithms such as mixers, crossfaders, dynamic filters, and dynamic volume controls. The pseudo-random 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 the previous step, download the new program and data memory contents to the corresponding memory locations using the I²C/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 55. For a more detailed description, see the DSP Core Control Registers section and Debug and Reliability Registers section.

Table 55. DSP Core and Reliability Registers

| Address | Register | Description |
|---------|----------------------|---|
| 0xF400 | HIBERNATE | Hibernate setting |
| 0xF401 | START_PULSE | Start pulse selection |
| 0xF402 | START_CORE | Instruction to start the core |
| 0xF403 | KILL_CORE | Instruction to stop the core |
| 0xF404 | START_ADDRESS | Start address of the program |
| 0xF405 | CORE_STATUS | Core status |
| 0xF421 | PANIC_CLEAR | Clear the panic manager |
| 0xF422 | PANIC_PARITY_MASK | Panic parity |
| 0xF423 | PANIC_SOFTWARE_MASK | Panic Mask 0 |
| 0xF424 | PANIC_WD_MASK | Panic Mask 1 |
| 0xF425 | PANIC_STACK_MASK | Panic Mask 2 |
| 0xF426 | PANIC_LOOP_MASK | Panic Mask 3 |
| 0xF427 | PANIC_FLAG | Panic flag |
| 0xF428 | PANIC_CODE | Panic code |
| 0xF432 | EXECUTE_COUNT | Execute stage error program count |
| 0xF443 | WATCHDOG_MAXCOUNT | Watchdog maximum count |
| 0xF444 | WATCHDOG_PRESCALE | Watchdog prescale |
| 0xF450 | BLOCKINT_EN | Enable block interrupts |
| 0xF451 | BLOCKINT_VALUE | Value for the block interrupt counter |
| 0xF460 | PROG_CNTR0 | Program counter, Bits[23:16] |
| 0xF461 | PROG_CNTR1 | Program counter, Bits[15:0] |
| 0xF462 | PROG_CNTR_CLEAR | Program counter clear |
| 0xF463 | PROG_CNTR_LENGTH0 | Program counter length, Bits[23:16] |
| 0xF464 | PROG_CNTR_LENGTH1 | Program counter length, Bits[15:0] |
| 0xF465 | PROG_CNTR_MAXLENGTH0 | Program counter maximum length, Bits[23:16] |

SOFTWARE FEATURES

Software Safeload

To update parameters in real time while avoiding pop and click noises on the output, a software safeload mechanism has been implemented by default in the SigmaStudio compiler. SigmaStudio automatically sets up the necessary code and parameters for all new projects. The safeload code, along 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, so 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 54.

Table 54. Software Safeload Memory Address Defaults

| Address (Hex) | Parameter | Function |
|---------------|------------------|---|
| 0x0014 | data_SafeLoad[0] | Safeload Data Slot 0 |
| 0x0015 | data_SafeLoad[1] | Safeload Data Slot 1 |
| 0x0016 | data_SafeLoad[2] | Safeload Data Slot 2 |
| 0x0017 | data_SafeLoad[3] | Safeload Data Slot 3 |
| 0x0018 | data_SafeLoad[4] | Safeload Data Slot 4 |
| 0x0019 | address_SafeLoad | Target address for safeload transfer |
| 0x001A | num_SafeLoad | Number of words to write/safeload trigger |

The first five addresses in Table 54 are the five data_SafeLoad parameters, which are slots for storing the data that is going 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 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 parameter designates 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 mono 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.

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, this equates to a delay of $\geq 20.83 \mu s$. 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 54. 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 79 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, Bit x parameters, starting at data_SafeLoad, 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 parameter. 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.

Soft Reset Function

The soft reset function allows the device to enter a state similar to when the hardware RESET pin is connected to ground. All control registers are reset to their default values, except the PLL registers, as follows: Register 0xF000 (PLL_CTRL0), Register 0xF001 (PLL_CTRL1), Register 0xF002 (PLL_CLK_SRC), Register 0xF003 (PLL_ENABLE), Register 0xF004 (PLL_LOCK), Register 0xF005 (MCLK_OUT), and Register 0xF006 (PLL_WATCHDOG), as well as the registers related to the panic manager.

An overview of the register related to the soft reset function is shown in Table 56. For a more detailed description, see the Soft Reset Register section.

Table 56. Soft Reset Register

| Address | Name | Description |
|---------|------------|----------------|
| 0xF890 | SOFT_RESET | Software reset |

```

##### DMO Allocation Summary #####
ModuleMemoryRegions:   Addr:   Modulo: Length:
__STACK_MODULO__      0x0     0       16

Address:  Module:
0x0       __STACK__
          __STACK__ [16] =0x00000000,0x00000000,0x00000000,0x00000000,
          0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,
          0x00000000,0x00000000,0x00000000,0x00000000,0x00000000,0x00000000
0x10      MP1_DM1 [4] =0x00000000,0x00000000,0x00000008,0x00000000
0x14      __SafeLoad_Module__
          data_SafeLoad [5] =0x00000000,0x00000000,0x00000000,0x00000000,0x00000000
0x19      __SafeLoad_Module__
          address_SafeLoad =0x00000000
0x1A      __SafeLoad_Module__
          num_SafeLoad =0x00000000
0x1B      EQS300MultiS1
          B2_1 =0x00000000
0x1C      EQS300MultiS1
          B1_1 =0x00000000
0x1D      EQS300MultiS1
          B0_1 =0x00000000
0x1E      EQS300MultiS1
          A2_1 =0x00000000
0x1F      EQS300MultiS1
          A1_1 =0x00000000
0x20      __DMO_PADDING__
          __DMO_PADDING__ [4] =0x00000000,0x00000000,0x00000000,0x00000000
    
```

Figure 79. Compiler Log Output Excerpt with Safeload Module Definitions

1149c-083

PIN DRIVE STRENGTH, SLEW RATE, AND PULL CONFIGURATION

Every digital output pin has configurable drive strength and slew rate. This allows the current sourcing ability of the driver to be modified to fit the application circuit. In general, higher drive strength is needed to improve signal integrity when driving high frequency clocks over long distances. Lower drive strength can be used for lower frequency clock signals, shorter traces, or in cases where reduced system EMI (electromagnetic interference) is desired. Slew rate can be increased if the edges

of the clock signal have rise or fall times that are too long. To achieve adequate signal integrity and minimize electromagnetic emissions, use the drive strength and slew rate settings in combination with good mixed-signal PCB design practices.

Pin Drive Strength, Slew Rate, and Pull Configuration Registers

An overview of the registers related to pin drive strength, slew rate, and pull configuration is shown in Table 57. For a more detailed description, see the Hardware Interfacing Registers section.

Table 57. Pin Drive Strength, Slew Rate, and Pull Configuration Registers

| Address | Register | Description |
|---------|----------------|--|
| 0xF780 | BCLK_IN0_PIN | BCLK input pin drive strength and slew rate (BCLK_IN0) |
| 0xF781 | BCLK_IN1_PIN | BCLK input pin drive strength and slew rate (BCLK_IN1) |
| 0xF782 | BCLK_IN2_PIN | BCLK input pin drive strength and slew rate (BCLK_IN2) |
| 0xF783 | BCLK_IN3_PIN | BCLK input pin drive strength and slew rate (BCLK_IN3) |
| 0xF784 | BCLK_OUT0_PIN | BCLK output pin drive strength and slew rate (BCLK_OUT0) |
| 0xF785 | BCLK_OUT1_PIN | BCLK output pin drive strength and slew rate (BCLK_OUT1) |
| 0xF786 | BCLK_OUT2_PIN | BCLK output pin drive strength and slew rate (BCLK_OUT2) |
| 0xF787 | BCLK_OUT3_PIN | BCLK output pin drive strength and slew rate (BCLK_OUT3) |
| 0xF788 | LRCLK_IN0_PIN | LRCLK input pin drive strength and slew rate (LRCLK_IN0) |
| 0xF789 | LRCLK_IN1_PIN | LRCLK input pin drive strength and slew rate (LRCLK_IN1) |
| 0xF78A | LRCLK_IN2_PIN | LRCLK input pin drive strength and slew rate (LRCLK_IN2) |
| 0xF78B | LRCLK_IN3_PIN | LRCLK input pin drive strength and slew rate (LRCLK_IN3) |
| 0xF78C | LRCLK_OUT0_PIN | LRCLK output pin drive strength and slew rate (LRCLK_OUT0) |
| 0xF78D | LRCLK_OUT1_PIN | LRCLK output pin drive strength and slew rate (LRCLK_OUT1) |
| 0xF78E | LRCLK_OUT2_PIN | LRCLK output pin drive strength and slew rate (LRCLK_OUT2) |
| 0xF78F | LRCLK_OUT3_PIN | LRCLK output pin drive strength and slew rate (LRCLK_OUT3) |
| 0xF790 | SDATA_IN0_PIN | SDATA input pin drive strength and slew rate (SDATA_IN0) |
| 0xF791 | SDATA_IN1_PIN | SDATA input pin drive strength and slew rate (SDATA_IN1) |
| 0xF792 | SDATA_IN2_PIN | SDATA input pin drive strength and slew rate (SDATA_IN2) |
| 0xF793 | SDATA_IN3_PIN | SDATA input pin drive strength and slew rate (SDATA_IN3) |
| 0xF794 | SDATA_OUT0_PIN | SDATA output pin drive strength and slew rate (SDATA_OUT0) |
| 0xF795 | SDATA_OUT1_PIN | SDATA output pin drive strength and slew rate (SDATA_OUT1) |
| 0xF796 | SDATA_OUT2_PIN | SDATA output pin drive strength and slew rate (SDATA_OUT2) |
| 0xF797 | SDATA_OUT3_PIN | SDATA output pin drive strength and slew rate (SDATA_OUT3) |
| 0xF798 | SPDIF_TX_PIN | S/PDIF transmitter pin drive strength and slew rate |
| 0xF799 | SCLK_SCL_PIN | SCLK/SCL pin drive strength and slew rate |
| 0xF79A | MISO_SDA_PIN | MISO/SDA pin drive strength and slew rate |
| 0xF79B | SS_PIN | SS/ADDR0 pin drive strength and slew rate |
| 0xF79C | MOSI_ADDR1_PIN | MOSI/ADDR1 pin drive strength and slew rate |
| 0xF79D | SCLK_SCL_M_PIN | SCL_M/SCLK_M/MP2 pin drive strength and slew rate |
| 0xF79E | MISO_SDA_M_PIN | SDA_M/MISO_M/MP3 pin drive strength and slew rate |
| 0xF79F | SS_M_PIN | SS_M/MP0 pin drive strength and slew rate |
| 0xF7A0 | MOSI_M_PIN | MOSI_M/MP1 pin drive strength and slew rate |
| 0xF7A1 | MP6_PIN | MP6 pin drive strength and slew rate |
| 0xF7A2 | MP7_PIN | MP7 pin drive strength and slew rate |
| 0xF7A3 | CLKOUT_PIN | CLKOUT pin drive strength and slew rate |

GLOBAL RAM AND CONTROL REGISTER MAP

The complete set of addresses accessible via the slave I²C/SPI control port is described in this section. The addresses are divided into two main parts: memory and registers.

RANDOM ACCESS MEMORY

The ADAU1452 has 1.28 Mb of data (40 kWords storing 32-bit data).

The ADAU1452 has eight kWords of program memory. Program memory consists of 32 bit words. Op codes for the DSP core are either 32 or 64 bits; therefore, program instructions can take up one or two addresses in memory. The program memory has parity bit protection. The panic manager flags parity errors when they are detected.

Program memory can only be written or read when the core is stopped. The program memory is hardware protected so it cannot be accidentally overwritten or corrupted at run time.

The DSP core is able to directly access all memory and registers.

Data memory acts as a storage area for both audio data and signal processing parameters, such as filter coefficients. The data memory has parity bit protection. The panic manager flags parity errors when they are detected. Modulo memory addressing is used in several audio processing algorithms. The boundaries between the fixed and rotating memories are set in SigmaStudio by the compiler, and they require no action on the part of the user.

CONTROL REGISTERS

All control registers store 16 bits (two bytes) of data. The register map for the ADAU1452 is defined in Table 59.

Table 59. Control Register Summary

| Address | Register Name | Description | Reset | RW |
|---------|---------------|---|--------|----|
| 0xF000 | PLL_CTRL0 | PLL feedback divider | 0x0060 | RW |
| 0xF001 | PLL_CTRL1 | PLL prescale divider | 0x0000 | RW |
| 0xF002 | PLL_CLK_SRC | PLL clock source | 0x0000 | RW |
| 0xF003 | PLL_ENABLE | PLL enable | 0x0000 | RW |
| 0xF004 | PLL_LOCK | PLL lock | 0x0000 | R |
| 0xF005 | MCLK_OUT | CLKOUT control | 0x0000 | RW |
| 0xF006 | PLL_WATCHDOG | Analog PLL watchdog control | 0x0001 | RW |
| 0xF020 | CLK_GEN1_M | Denominator (M) for Clock Generator 1 | 0x0006 | RW |
| 0xF021 | CLK_GEN1_N | Numerator (N) for Clock Generator 1 | 0x0001 | RW |
| 0xF022 | CLK_GEN2_M | Denominator (M) for Clock Generator 2 | 0x0009 | RW |
| 0xF023 | CLK_GEN2_N | Numerator (N) for Clock Generator 2 | 0x0001 | RW |
| 0xF024 | CLK_GEN3_M | Denominator (M) for Clock Generator 3 | 0x0000 | RW |
| 0xF025 | CLK_GEN3_N | Numerator for (N) Clock Generator 3 | 0x0000 | RW |
| 0xF026 | CLK_GEN3_SRC | Input Reference for Clock Generator 3 | 0x000E | RW |
| 0xF027 | CLK_GEN3_LOCK | Lock Bit for Clock Generator 3 input reference | 0x0000 | R |
| 0xF050 | POWER_ENABLE0 | Power Enable 0 | 0x0000 | RW |
| 0xF051 | POWER_ENABLE1 | Power Enable 1 | 0x0000 | RW |
| 0xF100 | ASRC_INPUT0 | ASRC input selector (ASRC 0, Channel 0 and Channel 1) | 0x0000 | RW |
| 0xF101 | ASRC_INPUT1 | ASRC input selector (ASRC 1, Channel 2 and Channel 3) | 0x0000 | RW |
| 0xF102 | ASRC_INPUT2 | ASRC input selector (ASRC 2, Channel 4 and Channel 5) | 0x0000 | RW |
| 0xF103 | ASRC_INPUT3 | ASRC input selector (ASRC 3, Channel 6 and Channel 7) | 0x0000 | RW |
| 0xF104 | ASRC_INPUT4 | ASRC input selector (ASRC 4, Channel 8 and Channel 9) | 0x0000 | RW |

Data and parameters assignment to the different memory spaces are handled in software. The modulo boundary locations are flexible.

A ROM table (of over seven kWords), containing a set of commonly used constants, can be accessed by the DSP core. This memory is used to increase the efficiency of audio processing algorithm development. The table includes information such as trigonometric tables, including sine, cosine, tangent, and hyperbolic tangent, twiddle factors for frequency domain processing, real mathematical constants, such as pi and factors of 2, and complex constants. The ROM table is not accessible from the I²C or SPI slave control port.

All memory addresses store 32 bits (four bytes) of data. The memory spaces for the ADAU1452 are defined in Table 58.

Table 58. Memory Map

| Address Range | Length | Memory | Data-Word Size |
|------------------|-------------|---------------------|----------------|
| 0x0000 to 0x4FFF | 20480 words | DM0 (Data Memory 0) | 32 bits |
| 0x6000 to 0xAFFF | 20480 words | DM1 (Data Memory 1) | 32 bits |
| 0xC000 to 0xDFFF | 8192 words | Program memory | 32 bits |

| Address | Register Name | Description | Reset | RW |
|---------|-----------------|--|--------|----|
| 0xF105 | ASRC_INPUT5 | ASRC input selector (ASRC 5, Channel 10 and Channel 11) | 0x0000 | RW |
| 0xF106 | ASRC_INPUT6 | ASRC input selector (ASRC 6, Channel 12 and Channel 13) | 0x0000 | RW |
| 0xF107 | ASRC_INPUT7 | ASRC input selector (ASRC 7, Channel 14 and Channel 15) | 0x0000 | RW |
| 0xF140 | ASRC_OUT_RATE0 | ASRC output rate (ASRC 0, Channel 0 and Channel 1) | 0x0000 | RW |
| 0xF141 | ASRC_OUT_RATE1 | ASRC output rate (ASRC 1, Channel 2 and Channel 3) | 0x0000 | RW |
| 0xF142 | ASRC_OUT_RATE2 | ASRC output rate (ASRC 2, Channel 4 and Channel 5) | 0x0000 | RW |
| 0xF143 | ASRC_OUT_RATE3 | ASRC output rate (ASRC 3, Channel 6 and Channel 7) | 0x0000 | RW |
| 0xF144 | ASRC_OUT_RATE4 | ASRC output rate (ASRC 4, Channel 8 and Channel 9) | 0x0000 | RW |
| 0xF145 | ASRC_OUT_RATE5 | ASRC output rate (ASRC 5, Channel 10 and Channel 11) | 0x0000 | RW |
| 0xF146 | ASRC_OUT_RATE6 | ASRC output rate (ASRC 6, Channel 12 and Channel 13) | 0x0000 | RW |
| 0xF147 | ASRC_OUT_RATE7 | ASRC output rate (ASRC 7, Channel 14 and Channel 15) | 0x0000 | RW |
| 0xF180 | SOUT_SOURCE0 | Source of data for serial output ports (Channel 0 and Channel 1) | 0x0000 | RW |
| 0xF181 | SOUT_SOURCE1 | Source of data for serial output ports (Channel 2 and Channel 3) | 0x0000 | RW |
| 0xF182 | SOUT_SOURCE2 | Source of data for serial output ports (Channel 4 and Channel 5) | 0x0000 | RW |
| 0xF183 | SOUT_SOURCE3 | Source of data for serial output ports (Channel 6 and Channel 7) | 0x0000 | RW |
| 0xF184 | SOUT_SOURCE4 | Source of data for serial output ports (Channel 8 and Channel 9) | 0x0000 | RW |
| 0xF185 | SOUT_SOURCE5 | Source of data for serial output ports (Channel 10 and Channel 11) | 0x0000 | RW |
| 0xF186 | SOUT_SOURCE6 | Source of data for serial output ports (Channel 12 and Channel 13) | 0x0000 | RW |
| 0xF187 | SOUT_SOURCE7 | Source of data for serial output ports (Channel 14 and Channel 15) | 0x0000 | RW |
| 0xF188 | SOUT_SOURCE8 | Source of data for serial output ports (Channel 16 and Channel 17) | 0x0000 | RW |
| 0xF189 | SOUT_SOURCE9 | Source of data for serial output ports (Channel 18 and Channel 19) | 0x0000 | RW |
| 0xF18A | SOUT_SOURCE10 | Source of data for serial output ports (Channel 20 and Channel 21) | 0x0000 | RW |
| 0xF18B | SOUT_SOURCE11 | Source of data for serial output ports (Channel 22 and Channel 23) | 0x0000 | RW |
| 0xF18C | SOUT_SOURCE12 | Source of data for serial output ports (Channel 24 and Channel 25) | 0x0000 | RW |
| 0xF18D | SOUT_SOURCE13 | Source of data for serial output ports (Channel 26 and Channel 27) | 0x0000 | RW |
| 0xF18E | SOUT_SOURCE14 | Source of data for serial output ports (Channel 28 and Channel 29) | 0x0000 | RW |
| 0xF18F | SOUT_SOURCE15 | Source of data for serial output ports (Channel 30 and Channel 31) | 0x0000 | RW |
| 0xF190 | SOUT_SOURCE16 | Source of data for serial output ports (Channel 32 and Channel 33) | 0x0000 | RW |
| 0xF191 | SOUT_SOURCE17 | Source of data for serial output ports (Channel 34 and Channel 35) | 0x0000 | RW |
| 0xF192 | SOUT_SOURCE18 | Source of data for serial output ports (Channel 36 and Channel 37) | 0x0000 | RW |
| 0xF193 | SOUT_SOURCE19 | Source of data for serial output ports (Channel 38 and Channel 39) | 0x0000 | RW |
| 0xF194 | SOUT_SOURCE20 | Source of data for serial output ports (Channel 40 and Channel 41) | 0x0000 | RW |
| 0xF195 | SOUT_SOURCE21 | Source of data for serial output ports (Channel 42 and Channel 43) | 0x0000 | RW |
| 0xF196 | SOUT_SOURCE22 | Source of data for serial output ports (Channel 44 and Channel 45) | 0x0000 | RW |
| 0xF197 | SOUT_SOURCE23 | Source of data for serial output ports (Channel 46 and Channel 47) | 0x0000 | RW |
| 0xF1C0 | SPDIFTX_INPUT | S/PDIF transmitter data selector | 0x0000 | RW |
| 0xF200 | SERIAL_BYTE_0_0 | Serial Port Control 0 (SDATA_IN0) | 0x0000 | RW |
| 0xF201 | SERIAL_BYTE_0_1 | Serial Port Control 1 (SDATA_IN0) | 0x0002 | RW |
| 0xF204 | SERIAL_BYTE_1_0 | Serial Port Control 0 (SDATA_IN1) | 0x0000 | RW |
| 0xF205 | SERIAL_BYTE_1_1 | Serial Port Control 1 (SDATA_IN1) | 0x0002 | RW |
| 0xF208 | SERIAL_BYTE_2_0 | Serial Port Control 0 (SDATA_IN2) | 0x0000 | RW |
| 0xF209 | SERIAL_BYTE_2_1 | Serial Port Control 1 (SDATA_IN2) | 0x0002 | RW |
| 0xF20C | SERIAL_BYTE_3_0 | Serial Port Control 0 (SDATA_IN3) | 0x0000 | RW |
| 0xF20D | SERIAL_BYTE_3_1 | Serial Port Control 1 (SDATA_IN3) | 0x0002 | RW |
| 0xF210 | SERIAL_BYTE_4_0 | Serial Port Control 0 (SDATA_OUT0) | 0x0000 | RW |
| 0xF211 | SERIAL_BYTE_4_1 | Serial Port Control 1 (SDATA_OUT0) | 0x0002 | RW |
| 0xF214 | SERIAL_BYTE_5_0 | Serial Port Control 0 (SDATA_OUT1) | 0x0000 | RW |
| 0xF215 | SERIAL_BYTE_5_1 | Serial Port Control 1 (SDATA_OUT1) | 0x0002 | RW |
| 0xF218 | SERIAL_BYTE_6_0 | Serial Port Control 0 (SDATA_OUT2) | 0x0000 | RW |
| 0xF219 | SERIAL_BYTE_6_1 | Serial Port Control 1 (SDATA_OUT2) | 0x0002 | RW |
| 0xF21C | SERIAL_BYTE_7_0 | Serial Port Control 0 (SDATA_OUT3) | 0x0000 | RW |
| 0xF21D | SERIAL_BYTE_7_1 | Serial Port Control 1 (SDATA_OUT3) | 0x0002 | RW |

| Address | Register Name | Description | Reset | RW |
|---------|---------------|--|--------|----|
| 0xF300 | FTDM_IN0 | FTDM mapping for the serial inputs (Channel 32, Bits[31:24]) | 0x0000 | RW |
| 0xF301 | FTDM_IN1 | FTDM mapping for the serial inputs (Channel 32, Bits[23:16]) | 0x0000 | RW |
| 0xF302 | FTDM_IN2 | FTDM mapping for the serial inputs (Channel 32, Bits[15:8]) | 0x0000 | RW |
| 0xF303 | FTDM_IN3 | FTDM mapping for the serial inputs (Channel 32, Bits[7:0]) | 0x0000 | RW |
| 0xF304 | FTDM_IN4 | FTDM mapping for the serial inputs (Channel 33, Bits[31:24]) | 0x0000 | RW |
| 0xF305 | FTDM_IN5 | FTDM mapping for the serial inputs (Channel 33, Bits[23:16]) | 0x0000 | RW |
| 0xF306 | FTDM_IN6 | FTDM mapping for the serial inputs (Channel 33, Bits[15:8]) | 0x0000 | RW |
| 0xF307 | FTDM_IN7 | FTDM mapping for the serial inputs (Channel 33, Bits[7:0]) | 0x0000 | RW |
| 0xF308 | FTDM_IN8 | FTDM mapping for the serial inputs (Channel 34, Bits[31:24]) | 0x0000 | RW |
| 0xF309 | FTDM_IN9 | FTDM mapping for the serial inputs (Channel 34, Bits[23:16]) | 0x0000 | RW |
| 0xF30A | FTDM_IN10 | FTDM mapping for the serial inputs (Channel 34, Bits[15:8]) | 0x0000 | RW |
| 0xF30B | FTDM_IN11 | FTDM mapping for the serial inputs (Channel 34, Bits[7:0]) | 0x0000 | RW |
| 0xF30C | FTDM_IN12 | FTDM mapping for the serial inputs (Channel 35, Bits[31:24]) | 0x0000 | RW |
| 0xF30D | FTDM_IN13 | FTDM mapping for the serial inputs (Channel 35, Bits[23:16]) | 0x0000 | RW |
| 0xF30E | FTDM_IN14 | FTDM mapping for the serial inputs (Channel 35, Bits[15:8]) | 0x0000 | RW |
| 0xF30F | FTDM_IN15 | FTDM mapping for the serial inputs (Channel 35, Bits[7:0]) | 0x0000 | RW |
| 0xF310 | FTDM_IN16 | FTDM mapping for the serial inputs (Channel 36, Bits[31:24]) | 0x0000 | RW |
| 0xF311 | FTDM_IN17 | FTDM mapping for the serial inputs (Channel 36, Bits[23:16]) | 0x0000 | RW |
| 0xF312 | FTDM_IN18 | FTDM mapping for the serial inputs (Channel 36, Bits[15:8]) | 0x0000 | RW |
| 0xF313 | FTDM_IN19 | FTDM mapping for the serial inputs (Channel 36, Bits[7:0]) | 0x0000 | RW |
| 0xF314 | FTDM_IN20 | FTDM mapping for the serial inputs (Channel 37, Bits[31:24]) | 0x0000 | RW |
| 0xF315 | FTDM_IN21 | FTDM mapping for the serial inputs (Channel 37, Bits[23:16]) | 0x0000 | RW |
| 0xF316 | FTDM_IN22 | FTDM mapping for the serial inputs (Channel 37, Bits[15:8]) | 0x0000 | RW |
| 0xF317 | FTDM_IN23 | FTDM mapping for the serial inputs (Channel 37, Bits[7:0]) | 0x0000 | RW |
| 0xF318 | FTDM_IN24 | FTDM mapping for the serial inputs (Channel 38, Bits[31:24]) | 0x0000 | RW |
| 0xF319 | FTDM_IN25 | FTDM mapping for the serial inputs (Channel 38, Bits[23:16]) | 0x0000 | RW |
| 0xF31A | FTDM_IN26 | FTDM mapping for the serial inputs (Channel 38, Bits[15:8]) | 0x0000 | RW |
| 0xF31B | FTDM_IN27 | FTDM mapping for the serial inputs (Channel 38, Bits[7:0]) | 0x0000 | RW |
| 0xF31C | FTDM_IN28 | FTDM mapping for the serial inputs (Channel 39, Bits[31:24]) | 0x0000 | RW |
| 0xF31D | FTDM_IN29 | FTDM mapping for the serial inputs (Channel 39, Bits[23:16]) | 0x0000 | RW |
| 0xF31E | FTDM_IN30 | FTDM mapping for the serial inputs (Channel 39, Bits[15:8]) | 0x0000 | RW |
| 0xF31F | FTDM_IN31 | FTDM mapping for the serial inputs (Channel 39, Bits[7:0]) | 0x0000 | RW |
| 0xF320 | FTDM_IN32 | FTDM mapping for the serial inputs (Channel 40, Bits[31:24]) | 0x0000 | RW |
| 0xF321 | FTDM_IN33 | FTDM mapping for the serial inputs (Channel 40, Bits[23:16]) | 0x0000 | RW |
| 0xF322 | FTDM_IN34 | FTDM mapping for the serial inputs (Channel 40, Bits[15:8]) | 0x0000 | RW |
| 0xF323 | FTDM_IN35 | FTDM mapping for the serial inputs (Channel 40, Bits[7:0]) | 0x0000 | RW |
| 0xF324 | FTDM_IN36 | FTDM mapping for the serial inputs (Channel 41, Bits[31:24]) | 0x0000 | RW |
| 0xF325 | FTDM_IN37 | FTDM mapping for the serial inputs (Channel 41, Bits[23:16]) | 0x0000 | RW |
| 0xF326 | FTDM_IN38 | FTDM mapping for the serial inputs (Channel 41, Bits[15:8]) | 0x0000 | RW |
| 0xF327 | FTDM_IN39 | FTDM mapping for the serial inputs (Channel 41, Bits[7:0]) | 0x0000 | RW |
| 0xF328 | FTDM_IN40 | FTDM mapping for the serial inputs (Channel 42, Bits[31:24]) | 0x0000 | RW |
| 0xF329 | FTDM_IN41 | FTDM mapping for the serial inputs (Channel 42, Bits[23:16]) | 0x0000 | RW |
| 0xF32A | FTDM_IN42 | FTDM mapping for the serial inputs (Channel 42, Bits[15:8]) | 0x0000 | RW |
| 0xF32B | FTDM_IN43 | FTDM mapping for the serial inputs (Channel 42, Bits[7:0]) | 0x0000 | RW |
| 0xF32C | FTDM_IN44 | FTDM mapping for the serial inputs (Channel 43, Bits[31:24]) | 0x0000 | RW |
| 0xF32D | FTDM_IN45 | FTDM mapping for the serial inputs (Channel 43, Bits[23:16]) | 0x0000 | RW |
| 0xF32E | FTDM_IN46 | FTDM mapping for the serial inputs (Channel 43, Bits[15:8]) | 0x0000 | RW |
| 0xF32F | FTDM_IN47 | FTDM mapping for the serial inputs (Channel 43, Bits[7:0]) | 0x0000 | RW |
| 0xF330 | FTDM_IN48 | FTDM mapping for the serial inputs (Channel 44, Bits[31:24]) | 0x0000 | RW |
| 0xF331 | FTDM_IN49 | FTDM mapping for the serial inputs (Channel 44, Bits[23:16]) | 0x0000 | RW |
| 0xF332 | FTDM_IN50 | FTDM mapping for the serial inputs (Channel 44, Bits[15:8]) | 0x0000 | RW |
| 0xF333 | FTDM_IN51 | FTDM mapping for the serial inputs (Channel 44, Bits[7:0]) | 0x0000 | RW |
| 0xF334 | FTDM_IN52 | FTDM mapping for the serial inputs (Channel 45, Bits[31:24]) | 0x0000 | RW |
| 0xF335 | FTDM_IN53 | FTDM mapping for the serial inputs (Channel 45, Bits[23:16]) | 0x0000 | RW |

| Address | Register Name | Description | Reset | RW |
|---------|---------------|--|--------|----|
| 0xF336 | FTDM_IN54 | FTDM mapping for the serial inputs (Channel 45, Bits[15:8]) | 0x0000 | RW |
| 0xF337 | FTDM_IN55 | FTDM mapping for the serial inputs (Channel 45, Bits[7:0]) | 0x0000 | RW |
| 0xF338 | FTDM_IN56 | FTDM mapping for the serial inputs (Channel 46, Bits[31:24]) | 0x0000 | RW |
| 0xF339 | FTDM_IN57 | FTDM mapping for the serial inputs ts (Channel 46, Bits[23:16]) | 0x0000 | RW |
| 0xF33A | FTDM_IN58 | FTDM mapping for the serial inputs (Channel 46, Bits[15:8]) | 0x0000 | RW |
| 0xF33B | FTDM_IN59 | FTDM mapping for the serial inputs (Channel 46, Bits[7:0]) | 0x0000 | RW |
| 0xF33C | FTDM_IN60 | FTDM mapping for the serial inputs (Channel 47, Bits[31:24]) | 0x0000 | RW |
| 0xF33D | FTDM_IN61 | FTDM mapping for the serial inputs (Channel 47, Bits[23:16]) | 0x0000 | RW |
| 0xF33E | FTDM_IN62 | FTDM mapping for the serial inputs (Channel 47, Bits[15:8]) | 0x0000 | RW |
| 0xF33F | FTDM_IN63 | FTDM mapping for the serial inputs (Channel 47, Bits[7:0]) | 0x0000 | RW |
| 0xF380 | FTDM_OUT0 | FTDM mapping for the serial outputs (Port 2, Channel 0, Bits[31:24]) | 0x0000 | RW |
| 0xF381 | FTDM_OUT1 | FTDM mapping for the serial outputs (Port 2, Channel 0, Bits[23:16]) | 0x0000 | RW |
| 0xF382 | FTDM_OUT2 | FTDM mapping for the serial outputs (Port 2, Channel 0, Bits[15:8]) | 0x0000 | RW |
| 0xF383 | FTDM_OUT3 | FTDM mapping for the serial outputs (Port 2, Channel 0, Bits[7:0]) | 0x0000 | RW |
| 0xF384 | FTDM_OUT4 | FTDM mapping for the serial outputs (Port 2, Channel 1, Bits[31:24]) | 0x0000 | RW |
| 0xF385 | FTDM_OUT5 | FTDM mapping for the serial outputs (Port 2, Channel 1, Bits[23:16]) | 0x0000 | RW |
| 0xF386 | FTDM_OUT6 | FTDM mapping for the serial outputs (Port 2, Channel 1, Bits[15:8]) | 0x0000 | RW |
| 0xF387 | FTDM_OUT7 | FTDM mapping for the serial outputs (Port 2, Channel 1, Bits[7:0]) | 0x0000 | RW |
| 0xF388 | FTDM_OUT8 | FTDM mapping for the serial outputs (Port 2, Channel 2, Bits[31:24]) | 0x0000 | RW |
| 0xF389 | FTDM_OUT9 | FTDM mapping for the serial outputs (Port 2, Channel 2, Bits[23:16]) | 0x0000 | RW |
| 0xF38A | FTDM_OUT10 | FTDM mapping for the serial outputs (Port 2, Channel 2, Bits[15:8]) | 0x0000 | RW |
| 0xF38B | FTDM_OUT11 | FTDM mapping for the serial outputs (Port 2, Channel 2, Bits[7:0]) | 0x0000 | RW |
| 0xF38C | FTDM_OUT12 | FTDM mapping for the serial outputs (Port 2, Channel 3, Bits[31:24]) | 0x0000 | RW |
| 0xF38D | FTDM_OUT13 | FTDM mapping for the serial outputs (Port 2, Channel 3, Bits[23:16]) | 0x0000 | RW |
| 0xF38E | FTDM_OUT14 | FTDM mapping for the serial outputs (Port 2, Channel 3, Bits[15:8]) | 0x0000 | RW |
| 0xF38F | FTDM_OUT15 | FTDM mapping for the serial outputs (Port 2, Channel 3, Bits[7:0]) | 0x0000 | RW |
| 0xF390 | FTDM_OUT16 | FTDM mapping for the serial outputs (Port 2, Channel 4, Bits[31:24]) | 0x0000 | RW |
| 0xF391 | FTDM_OUT17 | FTDM mapping for the serial outputs (Port 2, Channel 4, Bits[23:16]) | 0x0000 | RW |
| 0xF392 | FTDM_OUT18 | FTDM mapping for the serial outputs (Port 2, Channel 4, Bits[15:8]) | 0x0000 | RW |
| 0xF393 | FTDM_OUT19 | FTDM mapping for the serial outputs (Port 2, Channel 4, Bits[7:0]) | 0x0000 | RW |
| 0xF394 | FTDM_OUT20 | FTDM mapping for the serial outputs (Port 2, Channel 5, Bits[31:24]) | 0x0000 | RW |
| 0xF395 | FTDM_OUT21 | FTDM mapping for the serial outputs (Port 2, Channel 5, Bits[23:16]) | 0x0000 | RW |
| 0xF396 | FTDM_OUT22 | FTDM mapping for the serial outputs (Port 2, Channel 5, Bits[15:8]) | 0x0000 | RW |
| 0xF397 | FTDM_OUT23 | FTDM mapping for the serial outputs (Port 2, Channel 5, Bits[7:0]) | 0x0000 | RW |
| 0xF398 | FTDM_OUT24 | FTDM mapping for the serial outputs (Port 2, Channel 6, Bits[31:24]) | 0x0000 | RW |
| 0xF399 | FTDM_OUT25 | FTDM mapping for the serial outputs (Port 2, Channel 6, Bits[23:16]) | 0x0000 | RW |
| 0xF39A | FTDM_OUT26 | FTDM mapping for the serial outputs (Port 2, Channel 6, Bits[15:8]) | 0x0000 | RW |
| 0xF39B | FTDM_OUT27 | FTDM mapping for the serial outputs (Port 2, Channel 6, Bits[7:0]) | 0x0000 | RW |
| 0xF39C | FTDM_OUT28 | FTDM mapping for the serial outputs (Port 2, Channel 7, Bits[31:24]) | 0x0000 | RW |
| 0xF39D | FTDM_OUT29 | FTDM mapping for the serial outputs (Port 2, Channel 7, Bits[23:16]) | 0x0000 | RW |
| 0xF39E | FTDM_OUT30 | FTDM mapping for the serial outputs (Port 2, Channel 7, Bits[15:8]) | 0x0000 | RW |
| 0xF39F | FTDM_OUT31 | FTDM mapping for the serial outputs (Port 2, Channel 7, Bits[7:0]) | 0x0000 | RW |
| 0xF3A0 | FTDM_OUT32 | FTDM mapping for the serial outputs (Port 3, Channel 0, Bits[31:24]) | 0x0000 | RW |
| 0xF3A1 | FTDM_OUT33 | FTDM mapping for the serial outputs (Port 3, Channel 0, Bits[23:16]) | 0x0000 | RW |
| 0xF3A2 | FTDM_OUT34 | FTDM mapping for the serial outputs (Port 3, Channel 0, Bits[15:8]) | 0x0000 | RW |
| 0xF3A3 | FTDM_OUT35 | FTDM mapping for the serial outputs (Port 3, Channel 0, Bits[7:0]) | 0x0000 | RW |
| 0xF3A4 | FTDM_OUT36 | FTDM mapping for the serial outputs (Port 3, Channel 1, Bits[31:24]) | 0x0000 | RW |
| 0xF3A5 | FTDM_OUT37 | FTDM mapping for the serial outputs (Port 3, Channel 1, Bits[23:16]) | 0x0000 | RW |
| 0xF3A6 | FTDM_OUT38 | FTDM mapping for the serial outputs (Port 3, Channel 1, Bits[15:8]) | 0x0000 | RW |
| 0xF3A7 | FTDM_OUT39 | FTDM mapping for the serial outputs (Port 3, Channel 1, Bits[7:0]) | 0x0000 | RW |
| 0xF3A8 | FTDM_OUT40 | FTDM mapping for the serial outputs (Port 3, Channel 2, Bits[31:24]) | 0x0000 | RW |
| 0xF3A9 | FTDM_OUT41 | FTDM mapping for the serial outputs (Port 3, Channel 2, Bits[23:16]) | 0x0000 | RW |
| 0xF3AA | FTDM_OUT42 | FTDM mapping for the serial outputs (Port 3, Channel 2, Bits[15:8]) | 0x0000 | RW |
| 0xF3AB | FTDM_OUT43 | FTDM mapping for the serial outputs (Port 3, Channel 2, Bits[7:0]) | 0x0000 | RW |

| Address | Register Name | Description | Reset | RW |
|---------|----------------------|--|--------|----|
| 0xF3AC | FTDM_OUT44 | FTDM mapping for the serial outputs (Port 3, Channel 3, Bits[31:24]) | 0x0000 | RW |
| 0xF3AD | FTDM_OUT45 | FTDM mapping for the serial outputs (Port 3, Channel 3, Bits[23:16]) | 0x0000 | RW |
| 0xF3AE | FTDM_OUT46 | FTDM mapping for the serial outputs (Port 3, Channel 3, Bits[15:8]) | 0x0000 | RW |
| 0xF3AF | FTDM_OUT47 | FTDM mapping for the serial outputs (Port 3, Channel 3, Bits[7:0]) | 0x0000 | RW |
| 0xF3B0 | FTDM_OUT48 | FTDM mapping for the serial outputs (Port 3, Channel 4, Bits[31:24]) | 0x0000 | RW |
| 0xF3B1 | FTDM_OUT49 | FTDM mapping for the serial outputs (Port 3, Channel 4, Bits[23:16]) | 0x0000 | RW |
| 0xF3B2 | FTDM_OUT50 | FTDM mapping for the serial outputs (Port 3, Channel 4, Bits[15:8]) | 0x0000 | RW |
| 0xF3B3 | FTDM_OUT51 | FTDM mapping for the serial outputs (Port 3, Channel 4, Bits[7:0]) | 0x0000 | RW |
| 0xF3B4 | FTDM_OUT52 | FTDM mapping for the serial outputs (Port 3, Channel 5, Bits[31:24]) | 0x0000 | RW |
| 0xF3B5 | FTDM_OUT53 | FTDM mapping for the serial outputs (Port 3, Channel 5, Bits[23:16]) | 0x0000 | RW |
| 0xF3B6 | FTDM_OUT54 | FTDM mapping for the serial outputs (Port 3, Channel 5, Bits[15:8]) | 0x0000 | RW |
| 0xF3B7 | FTDM_OUT55 | FTDM mapping for the serial outputs (Port 3, Channel 5, Bits[7:0]) | 0x0000 | RW |
| 0xF3B8 | FTDM_OUT56 | FTDM mapping for the serial outputs (Port 3, Channel 6, Bits[31:24]) | 0x0000 | RW |
| 0xF3B9 | FTDM_OUT57 | FTDM mapping for the serial outputs (Port 3, Channel 6, Bits[23:16]) | 0x0000 | RW |
| 0xF3BA | FTDM_OUT58 | FTDM mapping for the serial outputs (Port 3, Channel 6, Bits[15:8]) | 0x0000 | RW |
| 0xF3BB | FTDM_OUT59 | FTDM mapping for the serial outputs (Port 3, Channel 6, Bits[7:0]) | 0x0000 | RW |
| 0xF3BC | FTDM_OUT60 | FTDM mapping for the serial outputs (Port 3, Channel 7, Bits[31:24]) | 0x0000 | RW |
| 0xF3BD | FTDM_OUT61 | FTDM mapping for the serial outputs (Port 3, Channel 7, Bits[23:16]) | 0x0000 | RW |
| 0xF3BE | FTDM_OUT62 | FTDM mapping for the serial outputs (Port 3, Channel 7, Bits[15:8]) | 0x0000 | RW |
| 0xF3BF | FTDM_OUT63 | FTDM mapping for the serial outputs (Port 3, Channel 7, Bits[7:0]) | 0x0000 | RW |
| 0xF400 | HIBERNATE | Hibernate setting | 0x0000 | RW |
| 0xF401 | START_PULSE | Start pulse selection | 0x0002 | RW |
| 0xF402 | START_CORE | Instruction to start the core | 0x0000 | RW |
| 0xF403 | KILL_CORE | Instruction to stop the core | 0x0000 | RW |
| 0xF404 | START_ADDRESS | Start address of the program | 0x0000 | RW |
| 0xF405 | CORE_STATUS | Core status | 0x0000 | R |
| 0xF421 | PANIC_CLEAR | Clear the panic manager | 0x0000 | RW |
| 0xF422 | PANIC_PARITY_MASK | Panic parity | 0x0003 | RW |
| 0xF423 | PANIC_SOFTWARE_MASK | Panic Mask 0 | 0x0000 | RW |
| 0xF424 | PANIC_WD_MASK | Panic Mask 1 | 0x0000 | RW |
| 0xF425 | PANIC_STACK_MASK | Panic Mask 2 | 0x0000 | RW |
| 0xF426 | PANIC_LOOP_MASK | Panic Mask 3 | 0x0000 | RW |
| 0xF427 | PANIC_FLAG | Panic flag | 0x0000 | R |
| 0xF428 | PANIC_CODE | Panic code | 0x0000 | R |
| 0xF432 | EXECUTE_COUNT | Execute stage error program count | 0x0000 | R |
| 0xF443 | WATCHDOG_MAXCOUNT | Watchdog maximum count | 0x0000 | RW |
| 0xF444 | WATCHDOG_PRESCALE | Watchdog prescale | 0x0000 | RW |
| 0xF450 | BLOCKINT_EN | Enable block interrupts | 0x0000 | RW |
| 0xF451 | BLOCKINT_VALUE | Value for the block interrupt counter | 0x0000 | RW |
| 0xF460 | PROG_CNTR0 | Program counter, Bits[23:16] | 0x0000 | R |
| 0xF461 | PROG_CNTR1 | Program counter, Bits[15:0] | 0x0000 | R |
| 0xF462 | PROG_CNTR_CLEAR | Program counter clear | 0x0000 | RW |
| 0xF463 | PROG_CNTR_LENGTH0 | Program counter length, Bits[23:16] | 0x0000 | R |
| 0xF464 | PROG_CNTR_LENGTH1 | Program counter length, Bits[15:0] | 0x0000 | R |
| 0xF465 | PROG_CNTR_MAXLENGTH0 | Program counter max length, Bits[23:16] | 0x0000 | R |
| 0xF466 | PROG_CNTR_MAXLENGTH1 | Program counter max length, Bits[15:0] | 0x0000 | R |
| 0xF510 | MP0_MODE | Multipurpose pin mode (SS_M/MP0) | 0x0000 | RW |
| 0xF511 | MP1_MODE | Multipurpose pin mode (MOSI_M/MP1) | 0x0000 | RW |
| 0xF512 | MP2_MODE | Multipurpose pin mode (SCL_M/SCLK_M/MP2) | 0x0000 | RW |
| 0xF513 | MP3_MODE | Multipurpose pin mode (SDA_M/MISO_M/MP3) | 0x0000 | RW |
| 0xF514 | MP4_MODE | Multipurpose pin mode (LRCLK_OUT0/MP4) | 0x0000 | RW |
| 0xF515 | MP5_MODE | Multipurpose pin mode (LRCLK_OUT1/MP5) | 0x0000 | RW |
| 0xF516 | MP6_MODE | Multipurpose pin mode (MP6) | 0x0000 | RW |
| 0xF517 | MP7_MODE | Multipurpose pin mode (MP7) | 0x0000 | RW |

| Address | Register Name | Description | Reset | RW |
|---------|----------------|--|--------|----|
| 0xF518 | MP8_MODE | Multipurpose pin mode (LRCLK_OUT2/MP8) | 0x0000 | RW |
| 0xF519 | MP9_MODE | Multipurpose pin mode (LRCLK_OUT3/MP9) | 0x0000 | RW |
| 0xF51A | MP10_MODE | Multipurpose pin mode (LRCLK_IN0/MP10) | 0x0000 | RW |
| 0xF51B | MP11_MODE | Multipurpose pin mode (LRCLK_IN1/MP11) | 0x0000 | RW |
| 0xF51C | MP12_MODE | Multipurpose pin mode (LRCLK_IN2/MP12) | 0x0000 | RW |
| 0xF51D | MP13_MODE | Multipurpose pin mode (LRCLK_IN3/MP13) | 0x0000 | RW |
| 0xF520 | MP0_WRITE | Multipurpose pin write value (SS_M/MP0) | 0x0000 | RW |
| 0xF521 | MP1_WRITE | Multipurpose pin write value (MOSI_M/MP1) | 0x0000 | RW |
| 0xF522 | MP2_WRITE | Multipurpose pin write value (SCL_M/SCLK_M/MP2) | 0x0000 | RW |
| 0xF523 | MP3_WRITE | Multipurpose pin write value (SDA_M,MISO_M/MP3) | 0x0000 | RW |
| 0xF524 | MP4_WRITE | Multipurpose pin write value (LRCLK_OUT0/MP4) | 0x0000 | RW |
| 0xF525 | MP5_WRITE | Multipurpose pin write value (LRCLK_OUT1/MP5) | 0x0000 | RW |
| 0xF526 | MP6_WRITE | Multipurpose pin write value (MP6) | 0x0000 | RW |
| 0xF527 | MP7_WRITE | Multipurpose pin write value (MP7) | 0x0000 | RW |
| 0xF528 | MP8_WRITE | Multipurpose pin write value (LRCLK_OUT2/MP8) | 0x0000 | RW |
| 0xF529 | MP9_WRITE | Multipurpose pin write value (LRCLK_OUT3/MP9) | 0x0000 | RW |
| 0xF52A | MP10_WRITE | Multipurpose pin write value (LRCLK_IN0/MP10) | 0x0000 | RW |
| 0xF52B | MP11_WRITE | Multipurpose pin write value (LRCLK_IN1/MP11) | 0x0000 | RW |
| 0xF52C | MP12_WRITE | Multipurpose pin write value (LRCLK_IN2/MP12) | 0x0000 | RW |
| 0xF52D | MP13_WRITE | Multipurpose pin write value (LRCLK_IN3/MP13) | 0x0000 | RW |
| 0xF530 | MP0_READ | Multipurpose pin read value (SS_M/MP0) | 0x0000 | R |
| 0xF531 | MP1_READ | Multipurpose pin read value (MOSI_M/MP1) | 0x0000 | R |
| 0xF532 | MP2_READ | Multipurpose pin read value (SCL_M/SCLK_M/MP2) | 0x0000 | R |
| 0xF533 | MP3_READ | Multipurpose pin read value (SDA_M,MISO_M/MP3) | 0x0000 | R |
| 0xF534 | MP4_READ | Multipurpose pin read value (LRCLK_OUT0/MP4) | 0x0000 | R |
| 0xF535 | MP5_READ | Multipurpose pin read value (LRCLK_OUT1/MP5) | 0x0000 | R |
| 0xF536 | MP6_READ | Multipurpose pin read value (MP6) | 0x0000 | R |
| 0xF537 | MP7_READ | Multipurpose pin read value (MP7) | 0x0000 | R |
| 0xF538 | MP8_READ | Multipurpose pin read value (LRCLK_OUT2/MP8) | 0x0000 | R |
| 0xF539 | MP9_READ | Multipurpose pin read value (LRCLK_OUT3/MP9) | 0x0000 | R |
| 0xF53A | MP10_READ | Multipurpose pin read value (LRCLK_IN0/MP10) | 0x0000 | R |
| 0xF53B | MP11_READ | Multipurpose pin read value (LRCLK_IN1/MP11) | 0x0000 | R |
| 0xF53C | MP12_READ | Multipurpose pin read value (LRCLK_IN2/MP12) | 0x0000 | R |
| 0xF53D | MP13_READ | Multipurpose pin read value (LRCLK_IN3/MP13) | 0x0000 | R |
| 0xF560 | DMIC_CTRL0 | Digital PDM microphone control (Channel 0 and Channel 1) | 0x4000 | RW |
| 0xF561 | DMIC_CTRL1 | Digital PDM microphone control (Channel 2 and Channel 3) | 0x4000 | RW |
| 0xF580 | ASRC_LOCK | ASRC lock status | 0x0000 | R |
| 0xF581 | ASRC_MUTE | ASRC mute | 0x0000 | RW |
| 0xF582 | ASRC0_RATIO | ASRC ratio (ASRC 0, Channel 0 and Channel 1) | 0x0000 | R |
| 0xF583 | ASRC1_RATIO | ASRC ratio (ASRC 1, Channel 2 and Channel 3) | 0x0000 | R |
| 0xF584 | ASRC2_RATIO | ASRC ratio (ASRC 2, Channel 4 and Channel 5) | 0x0000 | R |
| 0xF585 | ASRC3_RATIO | ASRC ratio (ASRC 3, Channel 6 and Channel 7) | 0x0000 | R |
| 0xF586 | ASRC4_RATIO | ASRC ratio (ASRC 4, Channel 8 and Channel 9) | 0x0000 | R |
| 0xF587 | ASRC5_RATIO | ASRC ratio (ASRC 5, Channel 10 and Channel 11) | 0x0000 | R |
| 0xF588 | ASRC6_RATIO | ASRC ratio (ASRC 6, Channel 12 and Channel 13) | 0x0000 | R |
| 0xF589 | ASRC7_RATIO | ASRC ratio (ASRC 7, Channel 14 and Channel 15) | 0x0000 | R |
| 0xF5A0 | ADC_READ0 | Auxiliary ADC read value (AUXADC0) | 0x0000 | R |
| 0xF5A1 | ADC_READ1 | Auxiliary ADC read value (AUXADC1) | 0x0000 | R |
| 0xF5A2 | ADC_READ2 | Auxiliary ADC read value (AUXADC2) | 0x0000 | R |
| 0xF5A3 | ADC_READ3 | Auxiliary ADC read value (AUXADC3) | 0x0000 | R |
| 0xF5A4 | ADC_READ4 | Auxiliary ADC read value (AUXADC4) | 0x0000 | R |
| 0xF5A5 | ADC_READ5 | Auxiliary ADC read value (AUXADC5) | 0x0000 | R |
| 0xF600 | SPDIF_LOCK_DET | S/PDIF receiver lock bit detection | 0x0000 | R |
| 0xF601 | SPDIF_RX_CTRL | S/PDIF receiver control | 0x0000 | RW |

| Address | Register Name | Description | Reset | RW |
|---------|-----------------------|--|--------|----|
| 0xF602 | SPDIF_RX_DECODE | Decoded signals from the S/PDIF receiver | 0x0000 | R |
| 0xF603 | SPDIF_RX_COMPRMODE | Compression mode from the S/PDIF receiver | 0x0000 | R |
| 0xF604 | SPDIF_RESTART | Automatically resume S/PDIF receiver audio input | 0x0000 | RW |
| 0xF605 | SPDIF_LOSS_OF_LOCK | S/PDIF receiver loss of lock detection | 0x0000 | R |
| 0xF608 | SPDIF_AUX_EN | S/PDIF receiver auxiliary outputs enable | 0x0000 | RW |
| 0xF60F | SPDIF_RX_AUXBIT_READY | S/PDIF receiver auxiliary bits ready flag | 0x0000 | R |
| 0xF610 | SPDIF_RX_CS_LEFT_0 | S/PDIF receiver channel status bits (left) | 0x0000 | R |
| 0xF611 | SPDIF_RX_CS_LEFT_1 | S/PDIF receiver channel status bits (left) | 0x0000 | R |
| 0xF612 | SPDIF_RX_CS_LEFT_2 | S/PDIF receiver channel status bits (left) | 0x0000 | R |
| 0xF613 | SPDIF_RX_CS_LEFT_3 | S/PDIF receiver channel status bits (left) | 0x0000 | R |
| 0xF614 | SPDIF_RX_CS_LEFT_4 | S/PDIF receiver channel status bits (left) | 0x0000 | R |
| 0xF615 | SPDIF_RX_CS_LEFT_5 | S/PDIF receiver channel status bits (left) | 0x0000 | R |
| 0xF616 | SPDIF_RX_CS_LEFT_6 | S/PDIF receiver channel status bits (left) | 0x0000 | R |
| 0xF617 | SPDIF_RX_CS_LEFT_7 | S/PDIF receiver channel status bits (left) | 0x0000 | R |
| 0xF618 | SPDIF_RX_CS_LEFT_8 | S/PDIF receiver channel status bits (left) | 0x0000 | R |
| 0xF619 | SPDIF_RX_CS_LEFT_9 | S/PDIF receiver channel status bits (left) | 0x0000 | R |
| 0xF61A | SPDIF_RX_CS_LEFT_10 | S/PDIF receiver channel status bits (left) | 0x0000 | R |
| 0xF61B | SPDIF_RX_CS_LEFT_11 | S/PDIF receiver channel status bits (left) | 0x0000 | R |
| 0xF620 | SPDIF_RX_CS_RIGHT_0 | S/PDIF receiver channel status bits (right) | 0x0000 | R |
| 0xF621 | SPDIF_RX_CS_RIGHT_1 | S/PDIF receiver channel status bits (right) | 0x0000 | R |
| 0xF622 | SPDIF_RX_CS_RIGHT_2 | S/PDIF receiver channel status bits (right) | 0x0000 | R |
| 0xF623 | SPDIF_RX_CS_RIGHT_3 | S/PDIF receiver channel status bits (right) | 0x0000 | R |
| 0xF624 | SPDIF_RX_CS_RIGHT_4 | S/PDIF receiver channel status bits (right) | 0x0000 | R |
| 0xF625 | SPDIF_RX_CS_RIGHT_5 | S/PDIF receiver channel status bits (right) | 0x0000 | R |
| 0xF626 | SPDIF_RX_CS_RIGHT_6 | S/PDIF receiver channel status bits (right) | 0x0000 | R |
| 0xF627 | SPDIF_RX_CS_RIGHT_7 | S/PDIF receiver channel status bits (right) | 0x0000 | R |
| 0xF628 | SPDIF_RX_CS_RIGHT_8 | S/PDIF receiver channel status bits (right) | 0x0000 | R |
| 0xF629 | SPDIF_RX_CS_RIGHT_9 | S/PDIF receiver channel status bits (right) | 0x0000 | R |
| 0xF62A | SPDIF_RX_CS_RIGHT_10 | S/PDIF receiver channel status bits (right) | 0x0000 | R |
| 0xF62B | SPDIF_RX_CS_RIGHT_11 | S/PDIF receiver channel status bits (right) | 0x0000 | R |
| 0xF630 | SPDIF_RX_UD_LEFT_0 | S/PDIF receiver user data bits (left) | 0x0000 | R |
| 0xF631 | SPDIF_RX_UD_LEFT_1 | S/PDIF receiver user data bits (left) | 0x0000 | R |
| 0xF632 | SPDIF_RX_UD_LEFT_2 | S/PDIF receiver user data bits (left) | 0x0000 | R |
| 0xF633 | SPDIF_RX_UD_LEFT_3 | S/PDIF receiver user data bits (left) | 0x0000 | R |
| 0xF634 | SPDIF_RX_UD_LEFT_4 | S/PDIF receiver user data bits (left) | 0x0000 | R |
| 0xF635 | SPDIF_RX_UD_LEFT_5 | S/PDIF receiver user data bits (left) | 0x0000 | R |
| 0xF636 | SPDIF_RX_UD_LEFT_6 | S/PDIF receiver user data bits (left) | 0x0000 | R |
| 0xF637 | SPDIF_RX_UD_LEFT_7 | S/PDIF receiver user data bits (left) | 0x0000 | R |
| 0xF638 | SPDIF_RX_UD_LEFT_8 | S/PDIF receiver user data bits (left) | 0x0000 | R |
| 0xF639 | SPDIF_RX_UD_LEFT_9 | S/PDIF receiver user data bits (left) | 0x0000 | R |
| 0xF63A | SPDIF_RX_UD_LEFT_10 | S/PDIF receiver user data bits (left) | 0x0000 | R |
| 0xF63B | SPDIF_RX_UD_LEFT_11 | S/PDIF receiver user data bits (left) | 0x0000 | R |
| 0xF640 | SPDIF_RX_UD_RIGHT_0 | S/PDIF receiver user data bits (right) | 0x0000 | R |
| 0xF641 | SPDIF_RX_UD_RIGHT_1 | S/PDIF receiver user data bits (right) | 0x0000 | R |
| 0xF642 | SPDIF_RX_UD_RIGHT_2 | S/PDIF receiver user data bits (right) | 0x0000 | R |
| 0xF643 | SPDIF_RX_UD_RIGHT_3 | S/PDIF receiver user data bits (right) | 0x0000 | R |
| 0xF644 | SPDIF_RX_UD_RIGHT_4 | S/PDIF receiver user data bits (right) | 0x0000 | R |
| 0xF645 | SPDIF_RX_UD_RIGHT_5 | S/PDIF receiver user data bits (right) | 0x0000 | R |
| 0xF646 | SPDIF_RX_UD_RIGHT_6 | S/PDIF receiver user data bits (right) | 0x0000 | R |
| 0xF647 | SPDIF_RX_UD_RIGHT_7 | S/PDIF receiver user data bits (right) | 0x0000 | R |
| 0xF648 | SPDIF_RX_UD_RIGHT_8 | S/PDIF receiver user data bits (right) | 0x0000 | R |
| 0xF649 | SPDIF_RX_UD_RIGHT_9 | S/PDIF receiver user data bits (right) | 0x0000 | R |
| 0xF64A | SPDIF_RX_UD_RIGHT_10 | S/PDIF receiver user data bits (right) | 0x0000 | R |
| 0xF64B | SPDIF_RX_UD_RIGHT_11 | S/PDIF receiver user data bits (right) | 0x0000 | R |

| Address | Register Name | Description | Reset | RW |
|---------|------------------------|---|--------|----|
| 0xF650 | SPDIF_RX_VB_LEFT_0 | S/PDIF receiver validity bits (left) | 0x0000 | R |
| 0xF651 | SPDIF_RX_VB_LEFT_1 | S/PDIF receiver validity bits (left) | 0x0000 | R |
| 0xF652 | SPDIF_RX_VB_LEFT_2 | S/PDIF receiver validity bits (left) | 0x0000 | R |
| 0xF653 | SPDIF_RX_VB_LEFT_3 | S/PDIF receiver validity bits (left) | 0x0000 | R |
| 0xF654 | SPDIF_RX_VB_LEFT_4 | S/PDIF receiver validity bits (left) | 0x0000 | R |
| 0xF655 | SPDIF_RX_VB_LEFT_5 | S/PDIF receiver validity bits (left) | 0x0000 | R |
| 0xF656 | SPDIF_RX_VB_LEFT_6 | S/PDIF receiver validity bits (left) | 0x0000 | R |
| 0xF657 | SPDIF_RX_VB_LEFT_7 | S/PDIF receiver validity bits (left) | 0x0000 | R |
| 0xF658 | SPDIF_RX_VB_LEFT_8 | S/PDIF receiver validity bits (left) | 0x0000 | R |
| 0xF659 | SPDIF_RX_VB_LEFT_9 | S/PDIF receiver validity bits (left) | 0x0000 | R |
| 0xF65A | SPDIF_RX_VB_LEFT_10 | S/PDIF receiver validity bits (left) | 0x0000 | R |
| 0xF65B | SPDIF_RX_VB_LEFT_11 | S/PDIF receiver validity bits (left) | 0x0000 | R |
| 0xF660 | SPDIF_RX_VB_RIGHT_0 | S/PDIF receiver validity bits (right) | 0x0000 | R |
| 0xF661 | SPDIF_RX_VB_RIGHT_1 | S/PDIF receiver validity bits (right) | 0x0000 | R |
| 0xF662 | SPDIF_RX_VB_RIGHT_2 | S/PDIF receiver validity bits (right) | 0x0000 | R |
| 0xF663 | SPDIF_RX_VB_RIGHT_3 | S/PDIF receiver validity bits (right) | 0x0000 | R |
| 0xF664 | SPDIF_RX_VB_RIGHT_4 | S/PDIF receiver validity bits (right) | 0x0000 | R |
| 0xF665 | SPDIF_RX_VB_RIGHT_5 | S/PDIF receiver validity bits (right) | 0x0000 | R |
| 0xF666 | SPDIF_RX_VB_RIGHT_6 | S/PDIF receiver validity bits (right) | 0x0000 | R |
| 0xF667 | SPDIF_RX_VB_RIGHT_7 | S/PDIF receiver validity bits (right) | 0x0000 | R |
| 0xF668 | SPDIF_RX_VB_RIGHT_8 | S/PDIF receiver validity bits (right) | 0x0000 | R |
| 0xF669 | SPDIF_RX_VB_RIGHT_9 | S/PDIF receiver validity bits (right) | 0x0000 | R |
| 0xF66A | SPDIF_RX_VB_RIGHT_10 | S/PDIF receiver validity bits (right) | 0x0000 | R |
| 0xF66B | SPDIF_RX_VB_RIGHT_11 | S/PDIF receiver validity bits (right) | 0x0000 | R |
| 0xF670 | SPDIF_RX_PB_LEFT_0 | S/PDIF receiver parity bits (left) | 0x0000 | R |
| 0xF671 | SPDIF_RX_PB_LEFT_1 | S/PDIF receiver parity bits (left) | 0x0000 | R |
| 0xF672 | SPDIF_RX_PB_LEFT_2 | S/PDIF receiver parity bits (left) | 0x0000 | R |
| 0xF673 | SPDIF_RX_PB_LEFT_3 | S/PDIF receiver parity bits (left) | 0x0000 | R |
| 0xF674 | SPDIF_RX_PB_LEFT_4 | S/PDIF receiver parity bits (left) | 0x0000 | R |
| 0xF675 | SPDIF_RX_PB_LEFT_5 | S/PDIF receiver parity bits (left) | 0x0000 | R |
| 0xF676 | SPDIF_RX_PB_LEFT_6 | S/PDIF receiver parity bits (left) | 0x0000 | R |
| 0xF677 | SPDIF_RX_PB_LEFT_7 | S/PDIF receiver parity bits (left) | 0x0000 | R |
| 0xF678 | SPDIF_RX_PB_LEFT_8 | S/PDIF receiver parity bits (left) | 0x0000 | R |
| 0xF679 | SPDIF_RX_PB_LEFT_9 | S/PDIF receiver parity bits (left) | 0x0000 | R |
| 0xF67A | SPDIF_RX_PB_LEFT_10 | S/PDIF receiver parity bits (left) | 0x0000 | R |
| 0xF67B | SPDIF_RX_PB_LEFT_11 | S/PDIF receiver parity bits (left) | 0x0000 | R |
| 0xF680 | SPDIF_RX_PB_RIGHT_0 | S/PDIF receiver parity bits (right) | 0x0000 | R |
| 0xF681 | SPDIF_RX_PB_RIGHT_1 | S/PDIF receiver parity bits (right) | 0x0000 | R |
| 0xF682 | SPDIF_RX_PB_RIGHT_2 | S/PDIF receiver parity bits (right) | 0x0000 | R |
| 0xF683 | SPDIF_RX_PB_RIGHT_3 | S/PDIF receiver parity bits (right) | 0x0000 | R |
| 0xF684 | SPDIF_RX_PB_RIGHT_4 | S/PDIF receiver parity bits (right) | 0x0000 | R |
| 0xF685 | SPDIF_RX_PB_RIGHT_5 | S/PDIF receiver parity bits (right) | 0x0000 | R |
| 0xF686 | SPDIF_RX_PB_RIGHT_6 | S/PDIF receiver parity bits (right) | 0x0000 | R |
| 0xF687 | SPDIF_RX_PB_RIGHT_7 | S/PDIF receiver parity bits (right) | 0x0000 | R |
| 0xF688 | SPDIF_RX_PB_RIGHT_8 | S/PDIF receiver parity bits (right) | 0x0000 | R |
| 0xF689 | SPDIF_RX_PB_RIGHT_9 | S/PDIF receiver parity bits (right) | 0x0000 | R |
| 0xF68A | SPDIF_RX_PB_RIGHT_10 | S/PDIF receiver parity bits (right) | 0x0000 | R |
| 0xF68B | SPDIF_RX_PB_RIGHT_11 | S/PDIF receiver parity bits (right) | 0x0000 | R |
| 0xF690 | SPDIF_TX_EN | S/PDIF transmitter enable | 0x0000 | RW |
| 0xF691 | SPDIF_TX_CTRL | S/PDIF transmitter control | 0x0000 | RW |
| 0xF69F | SPDIF_TX_AUXBIT_SOURCE | S/PDIF transmitter auxiliary bits source select | 0x0000 | RW |
| 0xF6A0 | SPDIF_TX_CS_LEFT_0 | S/PDIF transmitter channel status bits (left) | 0x0000 | RW |
| 0xF6A1 | SPDIF_TX_CS_LEFT_1 | S/PDIF transmitter channel status bits (left) | 0x0000 | RW |
| 0xF6A2 | SPDIF_TX_CS_LEFT_2 | S/PDIF transmitter channel status bits (left) | 0x0000 | RW |

| Address | Register Name | Description | Reset | RW |
|---------|----------------------|--|--------|----|
| 0xF6A3 | SPDIF_TX_CS_LEFT_3 | S/PDIF transmitter channel status bits (left) | 0x0000 | RW |
| 0xF6A4 | SPDIF_TX_CS_LEFT_4 | S/PDIF transmitter channel status bits (left) | 0x0000 | RW |
| 0xF6A5 | SPDIF_TX_CS_LEFT_5 | S/PDIF transmitter channel status bits (left) | 0x0000 | RW |
| 0xF6A6 | SPDIF_TX_CS_LEFT_6 | S/PDIF transmitter channel status bits (left) | 0x0000 | RW |
| 0xF6A7 | SPDIF_TX_CS_LEFT_7 | S/PDIF transmitter channel status bits (left) | 0x0000 | RW |
| 0xF6A8 | SPDIF_TX_CS_LEFT_8 | S/PDIF transmitter channel status bits (left) | 0x0000 | RW |
| 0xF6A9 | SPDIF_TX_CS_LEFT_9 | S/PDIF transmitter channel status bits (left) | 0x0000 | RW |
| 0xF6AA | SPDIF_TX_CS_LEFT_10 | S/PDIF transmitter channel status bits (left) | 0x0000 | RW |
| 0xF6AB | SPDIF_TX_CS_LEFT_11 | S/PDIF transmitter channel status bits (left) | 0x0000 | RW |
| 0xF6B0 | SPDIF_TX_CS_RIGHT_0 | S/PDIF transmitter channel status bits (right) | 0x0000 | RW |
| 0xF6B1 | SPDIF_TX_CS_RIGHT_1 | S/PDIF transmitter channel status bits (right) | 0x0000 | RW |
| 0xF6B2 | SPDIF_TX_CS_RIGHT_2 | S/PDIF transmitter channel status bits (right) | 0x0000 | RW |
| 0xF6B3 | SPDIF_TX_CS_RIGHT_3 | S/PDIF transmitter channel status bits (right) | 0x0000 | RW |
| 0xF6B4 | SPDIF_TX_CS_RIGHT_4 | S/PDIF transmitter channel status bits (right) | 0x0000 | RW |
| 0xF6B5 | SPDIF_TX_CS_RIGHT_5 | S/PDIF transmitter channel status bits (right) | 0x0000 | RW |
| 0xF6B6 | SPDIF_TX_CS_RIGHT_6 | S/PDIF transmitter channel status bits (right) | 0x0000 | RW |
| 0xF6B7 | SPDIF_TX_CS_RIGHT_7 | S/PDIF transmitter channel status bits (right) | 0x0000 | RW |
| 0xF6B8 | SPDIF_TX_CS_RIGHT_8 | S/PDIF transmitter channel status bits (right) | 0x0000 | RW |
| 0xF6B9 | SPDIF_TX_CS_RIGHT_9 | S/PDIF transmitter channel status bits (right) | 0x0000 | RW |
| 0xF6BA | SPDIF_TX_CS_RIGHT_10 | S/PDIF transmitter channel status bits (right) | 0x0000 | RW |
| 0xF6BB | SPDIF_TX_CS_RIGHT_11 | S/PDIF transmitter channel status bits (right) | 0x0000 | RW |
| 0xF6C0 | SPDIF_TX_UD_LEFT_0 | S/PDIF transmitter user data bits (left) | 0x0000 | RW |
| 0xF6C1 | SPDIF_TX_UD_LEFT_1 | S/PDIF transmitter user data bits (left) | 0x0000 | RW |
| 0xF6C2 | SPDIF_TX_UD_LEFT_2 | S/PDIF transmitter user data bits (left) | 0x0000 | RW |
| 0xF6C3 | SPDIF_TX_UD_LEFT_3 | S/PDIF transmitter user data bits (left) | 0x0000 | RW |
| 0xF6C4 | SPDIF_TX_UD_LEFT_4 | S/PDIF transmitter user data bits (left) | 0x0000 | RW |
| 0xF6C5 | SPDIF_TX_UD_LEFT_5 | S/PDIF transmitter user data bits (left) | 0x0000 | RW |
| 0xF6C6 | SPDIF_TX_UD_LEFT_6 | S/PDIF transmitter user data bits (left) | 0x0000 | RW |
| 0xF6C7 | SPDIF_TX_UD_LEFT_7 | S/PDIF transmitter user data bits (left) | 0x0000 | RW |
| 0xF6C8 | SPDIF_TX_UD_LEFT_8 | S/PDIF transmitter user data bits (left) | 0x0000 | RW |
| 0xF6C9 | SPDIF_TX_UD_LEFT_9 | S/PDIF transmitter user data bits (left) | 0x0000 | RW |
| 0xF6CA | SPDIF_TX_UD_LEFT_10 | S/PDIF transmitter user data bits (left) | 0x0000 | RW |
| 0xF6CB | SPDIF_TX_UD_LEFT_11 | S/PDIF transmitter user data bits (left) | 0x0000 | RW |
| 0xF6D0 | SPDIF_TX_UD_RIGHT_0 | S/PDIF transmitter user data bits (right) | 0x0000 | RW |
| 0xF6D1 | SPDIF_TX_UD_RIGHT_1 | S/PDIF transmitter user data bits (right) | 0x0000 | RW |
| 0xF6D2 | SPDIF_TX_UD_RIGHT_2 | S/PDIF transmitter user data bits (right) | 0x0000 | RW |
| 0xF6D3 | SPDIF_TX_UD_RIGHT_3 | S/PDIF transmitter user data bits (right) | 0x0000 | RW |
| 0xF6D4 | SPDIF_TX_UD_RIGHT_4 | S/PDIF transmitter user data bits (right) | 0x0000 | RW |
| 0xF6D5 | SPDIF_TX_UD_RIGHT_5 | S/PDIF transmitter user data bits (right) | 0x0000 | RW |
| 0xF6D6 | SPDIF_TX_UD_RIGHT_6 | S/PDIF transmitter user data bits (right) | 0x0000 | RW |
| 0xF6D7 | SPDIF_TX_UD_RIGHT_7 | S/PDIF transmitter user data bits (right) | 0x0000 | RW |
| 0xF6D8 | SPDIF_TX_UD_RIGHT_8 | S/PDIF transmitter user data bits (right) | 0x0000 | RW |
| 0xF6D9 | SPDIF_TX_UD_RIGHT_9 | S/PDIF transmitter user data bits (right) | 0x0000 | RW |
| 0xF6DA | SPDIF_TX_UD_RIGHT_10 | S/PDIF transmitter user data bits (right) | 0x0000 | RW |
| 0xF6DB | SPDIF_TX_UD_RIGHT_11 | S/PDIF transmitter user data bits (right) | 0x0000 | RW |
| 0xF6E0 | SPDIF_TX_VB_LEFT_0 | S/PDIF transmitter validity bits (left) | 0x0000 | RW |
| 0xF6E1 | SPDIF_TX_VB_LEFT_1 | S/PDIF transmitter validity bits (left) | 0x0000 | RW |
| 0xF6E2 | SPDIF_TX_VB_LEFT_2 | S/PDIF transmitter validity bits (left) | 0x0000 | RW |
| 0xF6E3 | SPDIF_TX_VB_LEFT_3 | S/PDIF transmitter validity bits (left) | 0x0000 | RW |
| 0xF6E4 | SPDIF_TX_VB_LEFT_4 | S/PDIF transmitter validity bits (left) | 0x0000 | RW |
| 0xF6E5 | SPDIF_TX_VB_LEFT_5 | S/PDIF transmitter validity bits (left) | 0x0000 | RW |
| 0xF6E6 | SPDIF_TX_VB_LEFT_6 | S/PDIF transmitter validity bits (left) | 0x0000 | RW |
| 0xF6E7 | SPDIF_TX_VB_LEFT_7 | S/PDIF transmitter validity bits (left) | 0x0000 | RW |
| 0xF6E8 | SPDIF_TX_VB_LEFT_8 | S/PDIF transmitter validity bits (left) | 0x0000 | RW |

| Address | Register Name | Description | Reset | RW |
|---------|----------------------|---|--------|----|
| 0xF6E9 | SPDIF_TX_VB_LEFT_9 | S/PDIF transmitter validity bits (left) | 0x0000 | RW |
| 0xF6EA | SPDIF_TX_VB_LEFT_10 | S/PDIF transmitter validity bits (left) | 0x0000 | RW |
| 0xF6EB | SPDIF_TX_VB_LEFT_11 | S/PDIF transmitter validity bits (left) | 0x0000 | RW |
| 0xF6F0 | SPDIF_TX_VB_RIGHT_0 | S/PDIF transmitter validity bits (right) | 0x0000 | RW |
| 0xF6F1 | SPDIF_TX_VB_RIGHT_1 | S/PDIF transmitter validity bits (right) | 0x0000 | RW |
| 0xF6F2 | SPDIF_TX_VB_RIGHT_2 | S/PDIF transmitter validity bits (right) | 0x0000 | RW |
| 0xF6F3 | SPDIF_TX_VB_RIGHT_3 | S/PDIF transmitter validity bits (right) | 0x0000 | RW |
| 0xF6F4 | SPDIF_TX_VB_RIGHT_4 | S/PDIF transmitter validity bits (right) | 0x0000 | RW |
| 0xF6F5 | SPDIF_TX_VB_RIGHT_5 | S/PDIF transmitter validity bits (right) | 0x0000 | RW |
| 0xF6F6 | SPDIF_TX_VB_RIGHT_6 | S/PDIF transmitter validity bits (right) | 0x0000 | RW |
| 0xF6F7 | SPDIF_TX_VB_RIGHT_7 | S/PDIF transmitter validity bits (right) | 0x0000 | RW |
| 0xF6F8 | SPDIF_TX_VB_RIGHT_8 | S/PDIF transmitter validity bits (right) | 0x0000 | RW |
| 0xF6F9 | SPDIF_TX_VB_RIGHT_9 | S/PDIF transmitter validity bits (right) | 0x0000 | RW |
| 0xF6FA | SPDIF_TX_VB_RIGHT_10 | S/PDIF transmitter validity bits (right) | 0x0000 | RW |
| 0xF6FB | SPDIF_TX_VB_RIGHT_11 | S/PDIF transmitter validity bits (right) | 0x0000 | RW |
| 0xF700 | SPDIF_TX_PB_LEFT_0 | S/PDIF transmitter parity bits (left) | 0x0000 | RW |
| 0xF701 | SPDIF_TX_PB_LEFT_1 | S/PDIF transmitter parity bits (left) | 0x0000 | RW |
| 0xF702 | SPDIF_TX_PB_LEFT_2 | S/PDIF transmitter parity bits (left) | 0x0000 | RW |
| 0xF703 | SPDIF_TX_PB_LEFT_3 | S/PDIF transmitter parity bits (left) | 0x0000 | RW |
| 0xF704 | SPDIF_TX_PB_LEFT_4 | S/PDIF transmitter parity bits (left) | 0x0000 | RW |
| 0xF705 | SPDIF_TX_PB_LEFT_5 | S/PDIF transmitter parity bits (left) | 0x0000 | RW |
| 0xF706 | SPDIF_TX_PB_LEFT_6 | S/PDIF transmitter parity bits (left) | 0x0000 | RW |
| 0xF707 | SPDIF_TX_PB_LEFT_7 | S/PDIF transmitter parity bits (left) | 0x0000 | RW |
| 0xF708 | SPDIF_TX_PB_LEFT_8 | S/PDIF transmitter parity bits (left) | 0x0000 | RW |
| 0xF709 | SPDIF_TX_PB_LEFT_9 | S/PDIF transmitter parity bits (left) | 0x0000 | RW |
| 0xF70A | SPDIF_TX_PB_LEFT_10 | S/PDIF transmitter parity bits (left) | 0x0000 | RW |
| 0xF70B | SPDIF_TX_PB_LEFT_11 | S/PDIF transmitter parity bits (left) | 0x0000 | RW |
| 0xF710 | SPDIF_TX_PB_RIGHT_0 | S/PDIF transmitter parity bits (right) | 0x0000 | RW |
| 0xF711 | SPDIF_TX_PB_RIGHT_1 | S/PDIF transmitter parity bits (right) | 0x0000 | RW |
| 0xF712 | SPDIF_TX_PB_RIGHT_2 | S/PDIF transmitter parity bits (right) | 0x0000 | RW |
| 0xF713 | SPDIF_TX_PB_RIGHT_3 | S/PDIF transmitter parity bits (right) | 0x0000 | RW |
| 0xF714 | SPDIF_TX_PB_RIGHT_4 | S/PDIF transmitter parity bits (right) | 0x0000 | RW |
| 0xF715 | SPDIF_TX_PB_RIGHT_5 | S/PDIF transmitter parity bits (right) | 0x0000 | RW |
| 0xF716 | SPDIF_TX_PB_RIGHT_6 | S/PDIF transmitter parity bits (right) | 0x0000 | RW |
| 0xF717 | SPDIF_TX_PB_RIGHT_7 | S/PDIF transmitter parity bits (right) | 0x0000 | RW |
| 0xF718 | SPDIF_TX_PB_RIGHT_8 | S/PDIF transmitter parity bits (right) | 0x0000 | RW |
| 0xF719 | SPDIF_TX_PB_RIGHT_9 | S/PDIF transmitter parity bits (right) | 0x0000 | RW |
| 0xF71A | SPDIF_TX_PB_RIGHT_10 | S/PDIF transmitter parity bits (right) | 0x0000 | RW |
| 0xF71B | SPDIF_TX_PB_RIGHT_11 | S/PDIF transmitter parity bits (right) | 0x0000 | RW |
| 0xF780 | BCLK_IN0_PIN | BCLK input pins drive strength and slew rate (BCLK_IN0) | 0x0018 | RW |
| 0xF781 | BCLK_IN1_PIN | BCLK input pins drive strength and slew rate (BCLK_IN1) | 0x0018 | RW |
| 0xF782 | BCLK_IN2_PIN | BCLK input pins drive strength and slew rate (BCLK_IN2) | 0x0018 | RW |
| 0xF783 | BCLK_IN3_PIN | BCLK input pins drive strength and slew rate (BCLK_IN3) | 0x0018 | RW |
| 0xF784 | BCLK_OUT0_PIN | BCLK output pins drive strength and slew rate (BCLK_OUT0) | 0x0018 | RW |
| 0xF785 | BCLK_OUT1_PIN | BCLK output pins drive strength and slew rate (BCLK_OUT1) | 0x0018 | RW |
| 0xF786 | BCLK_OUT2_PIN | BCLK output pins drive strength and slew rate (BCLK_OUT2) | 0x0018 | RW |
| 0xF787 | BCLK_OUT3_PIN | BCLK output pins drive strength and slew rate (BCLK_OUT3) | 0x0018 | RW |
| 0xF788 | LRCLK_IN0_PIN | LRCLK input pins drive strength and slew rate (LRCLK_IN0) | 0x0018 | RW |
| 0xF789 | LRCLK_IN1_PIN | LRCLK input pins drive strength and slew rate (LRCLK_IN1) | 0x0018 | RW |
| 0xF78A | LRCLK_IN2_PIN | LRCLK input pins drive strength and slew rate (LRCLK_IN2) | 0x0018 | RW |
| 0xF78B | LRCLK_IN3_PIN | LRCLK input pins drive strength and slew rate (LRCLK_IN3) | 0x0018 | RW |
| 0xF78C | LRCLK_OUT0_PIN | LRCLK output pins drive strength and slew rate (LRCLK_OUT0) | 0x0018 | RW |
| 0xF78D | LRCLK_OUT1_PIN | LRCLK output pins drive strength and slew rate (LRCLK_OUT1) | 0x0018 | RW |
| 0xF78E | LRCLK_OUT2_PIN | LRCLK output pins drive strength and slew rate (LRCLK_OUT2) | 0x0018 | RW |

| Address | Register Name | Description | Reset | RW |
|---------|----------------|---|--------|----|
| 0xF78F | LRCLK_OUT3_PIN | LRCLK output pins drive strength and slew rate (LRCLK_OUT3) | 0x0018 | RW |
| 0xF790 | SDATA_IN0_PIN | SDATA input pins drive strength and slew rate (SDATA_IN0) | 0x0018 | RW |
| 0xF791 | SDATA_IN1_PIN | SDATA input pins drive strength and slew rate (SDATA_IN1) | 0x0018 | RW |
| 0xF792 | SDATA_IN2_PIN | SDATA input pins drive strength and slew rate (SDATA_IN2) | 0x0018 | RW |
| 0xF793 | SDATA_IN3_PIN | SDATA input pins drive strength and slew rate (SDATA_IN3) | 0x0018 | RW |
| 0xF794 | SDATA_OUT0_PIN | SDATA output pins drive strength and slew rate (SDATA_OUT0) | 0x0008 | RW |
| 0xF795 | SDATA_OUT1_PIN | SDATA output pins drive strength and slew rate (SDATA_OUT1) | 0x0008 | RW |
| 0xF796 | SDATA_OUT2_PIN | SDATA output pins drive strength and slew rate (SDATA_OUT2) | 0x0008 | RW |
| 0xF797 | SDATA_OUT3_PIN | SDATA output pins drive strength and slew rate (SDATA_OUT3) | 0x0008 | RW |
| 0xF798 | SPDIF_TX_PIN | S/PDIF transmitter pin drive strength and slew rate | 0x0008 | RW |
| 0xF799 | SCLK_SCL_PIN | SCLK/SCL pin drive strength and slew rate | 0x0008 | RW |
| 0xF79A | MISO_SDA_PIN | MISO/SDA pin drive strength and slew rate | 0x0008 | RW |
| 0xF79B | SS_PIN | SS/ADDR0 pin drive strength and slew rate | 0x0018 | RW |
| 0xF79C | MOSI_ADDR1_PIN | MOSI/ADDR1 pin drive strength and slew rate | 0x0018 | RW |
| 0xF79D | SCLK_SCL_M_PIN | SCL_M/SCLK_M/MP2 pin drive strength and slew rate | 0x0008 | RW |
| 0xF79E | MISO_SDA_M_PIN | SDA_M/MISO_M/MP3 pin drive strength and slew rate | 0x0008 | RW |
| 0xF79F | SS_M_PIN | SS_M/MP0 pin drive strength and slew rate | 0x0018 | RW |
| 0xF7A0 | MOSI_M_PIN | MOSI_M/MP1 pin drive strength and slew rate | 0x0018 | RW |
| 0xF7A1 | MP6_PIN | MP6 pin drive strength and slew rate | 0x0018 | RW |
| 0xF7A2 | MP7_PIN | MP7 pin drive strength and slew rate | 0x0018 | RW |
| 0xF7A3 | CLKOUT_PIN | CLKOUT pin drive strength and slew rate | 0x0008 | RW |
| 0xF890 | SOFT_RESET | Soft reset | 0x0001 | RW |

CONTROL REGISTER DETAILS

PLL CONFIGURATION REGISTERS

PLL Feedback Divider Register

Address: 0xF000, Reset: 0x0060, Name: PLL_CTRL0

This register is the value of the feedback divider in the PLL. This value effectively multiplies the frequency of the input clock to the PLL, creating the output system clock, which clocks the DSP core and other digital circuit blocks. The format of the value stored in this register is binary integer in 7.0 format. For example, the default feedback divider value of 96 is stored as 0x60. The value written to this register does not take effect until Register 0xF003 (PLL_ENABLE), Bit 0 (PLL_ENABLE) changes state from 0b0 to 0b1.

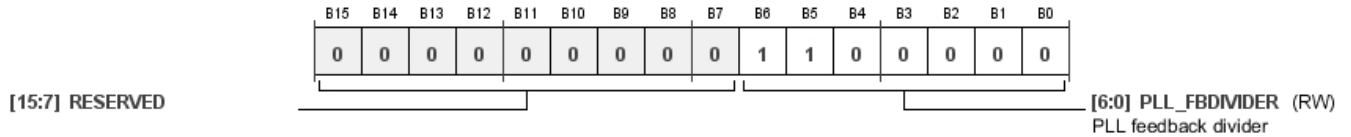


Table 60. Bit Descriptions for PLL_CTRL0

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|----------------|----------|---|-------|--------|
| [15:7] | RESERVED | | | 0x0 | RW |
| [6:0] | PLL_FBDDIVIDER | | PLL feedback divider. This is the value of the feedback divider in the PLL, which effectively multiplies the frequency of the input clock to the PLL, creating the output system clock, which clocks the DSP core and other digital circuit blocks. The format of the value stored in this register is binary integer in 7.0 format. For example, the default feedback divider value of 96 is stored as 0x60. | 0x60 | RW |

PLL Prescale Divider Register

Address: 0xF001, Reset: 0x0000, Name: PLL_CTRL1

This register sets the input prescale divider for the PLL. The value written to this register does not take effect until Register 0xF003 (PLL_ENABLE), Bit 0 (PLL_ENABLE) changes state from 0b0 to 0b1.

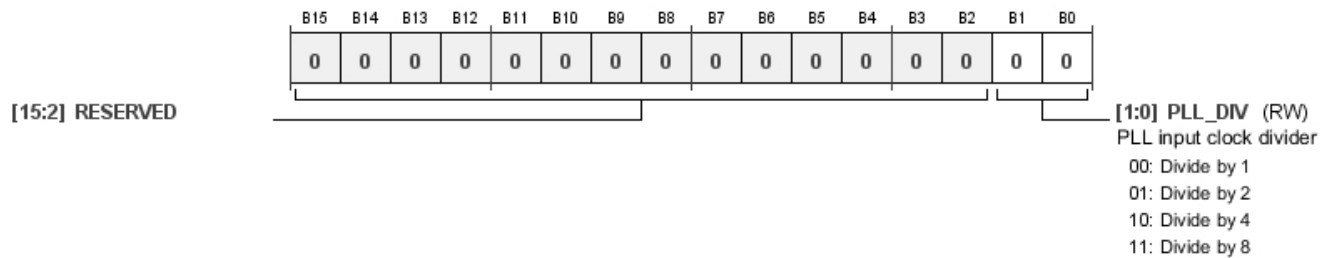


Table 61. Bit Descriptions for PLL_CTRL1

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|----------|----------|--|-------|--------|
| [15:2] | RESERVED | | | 0x0 | RW |
| [1:0] | PLL_DIV | | PLL input clock divider. This prescale clock divider creates the PLL input clock from the externally input master clock. The nominal frequency of the PLL input is 3.072 MHz. Therefore, if the input master clock frequency is 3.072 MHz, set the prescale clock divider to divide by 1. If the input clock is 12.288 MHz, set the prescale clock divider to divide by 4. The goal is to make the input to the PLL as close to 3.072 MHz as possible. | 0x0 | RW |
| | | 00 | Divide by 1 | | |
| | | 01 | Divide by 2 | | |
| | | 10 | Divide by 4 | | |
| | | 11 | Divide by 8 | | |

PLL Clock Source Register

Address: 0xF002, Reset: 0x0000, Name: PLL_CLK_SRC

This register selects the source of the clock used for input to the core and the clock generators. The clock can either be taken directly from the signal on the XTALIN/MCLK pin or from the output of the PLL. In almost every case, the PLL clock should be used. The value written to this register does not take effect until Register 0xF003 (PLL_ENABLE), Bit 0 (PLL_ENABLE) changes state from 0b0 to 0b1.

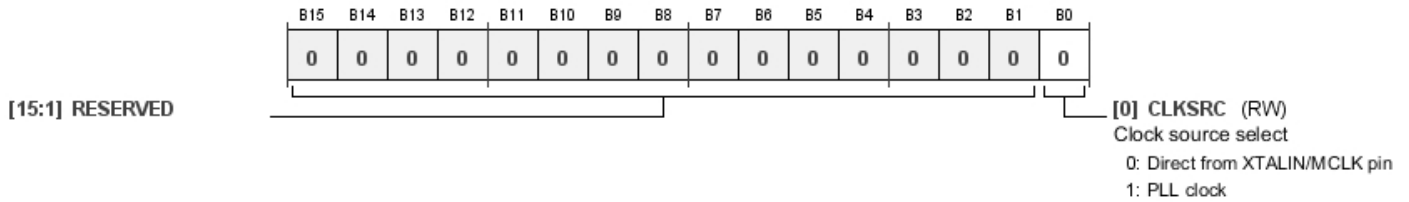


Table 62. Bit Descriptions for PLL_CLK_SRC

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|----------|----------|--|-------|--------|
| [15:1] | RESERVED | | | 0x0 | RW |
| 0 | CLKSRC | 0 1 | Clock source select. The PLL output is nominally 294.912 MHz, which is the nominal operating frequency of the core and the clock generator inputs. In most use cases, do not use the direct XTALIN/MCLK input option because the range of allowable frequencies on the XTALIN/MCLK pin is has an upper limit that is significantly lower in frequency than the nominal system clock frequency. 0 Direct from XTALIN/MCLK pin 1 PLL clock | 0x0 | RW |

PLL Enable Register

Address: 0xF003, Reset: 0x0000, Name: PLL_ENABLE

This register enables or disables the PLL. The PLL does not attempt to lock to an incoming clock until Bit 0 (PLL_ENABLE) is enabled. When Bit 0 (PLL_ENABLE) is set to 0b0, the PLL does not output a clock signal, causing all other clock circuits in the device that rely on the PLL to become idle. When Bit 0 (PLL_ENABLE) transitions from 0b0 to 0b1, the settings in Register 0xF000 (PLL_CTRL0), Register 0xF001 (PLL_CTRL1), Register 0xF002 (PLL_CLK_SRC), and Register 0xF005 (MCLK_OUT) are activated.

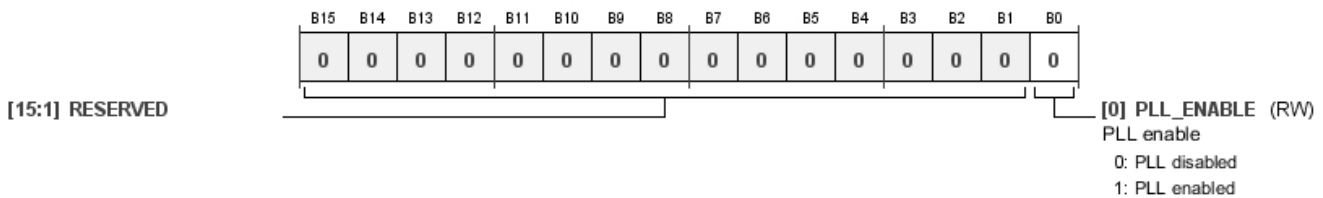


Table 63. Bit Descriptions for PLL_ENABLE

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|------------|----------|---|-------|--------|
| [15:1] | RESERVED | | | 0x0 | RW |
| 0 | PLL_ENABLE | 0 1 | PLL enable. Load the values of Register 0xF000, Register 0xF001, Register 0xF002, and Register 0xF005 when this bit transitions from 0b0 to 0b1. 0 PLL disabled 1 PLL enabled | 0x0 | RW |

PLL Lock Register

Address: 0xF004, Reset: 0x0000, Name: PLL_LOCK

This register contains a flag that represents the lock status of the PLL. Lock status has two prerequisites: a stable input clock is being routed to the PLL, the related PLL registers (Register 0xF000 (PLL_CTRL0), Register 0xF001 (PLL_CTRL1), and Register 0xF002 (PLL_CLK_SRC)) are set appropriately, the PLL is enabled (Register 0xF003 (PLL_ENABLE), Bit 0 (PLL_ENABLE) = 0b1), and the PLL has had adequate time to adjust its feedback path and provide a stable output clock to the rest of the device. The amount of time required to achieve lock to a new input clock signal varies based on system conditions, so Bit 0 (PLL_LOCK) provides a clear indication of when lock has been achieved.

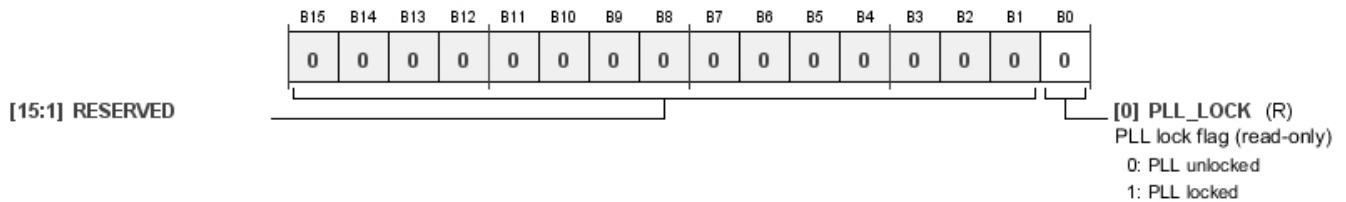


Table 64. Bit Descriptions for PLL_LOCK

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|----------|----------|--|-------|--------|
| [15:1] | RESERVED | | | 0x0 | RW |
| 0 | PLL_LOCK | 0 1 | PLL lock flag (read only). PLL unlocked PLL locked | 0x0 | R |

CLKOUT Control Register

Address: 0xF005, Reset: 0x0000, Name: MCLK_OUT

This register enables and configures the signal output from the CLKOUT pin. The value written to this register does not take effect until Register 0xF003 (PLL_ENABLE), Bit 0 (PLL_ENABLE), changes state from 0b0 to 0b1.

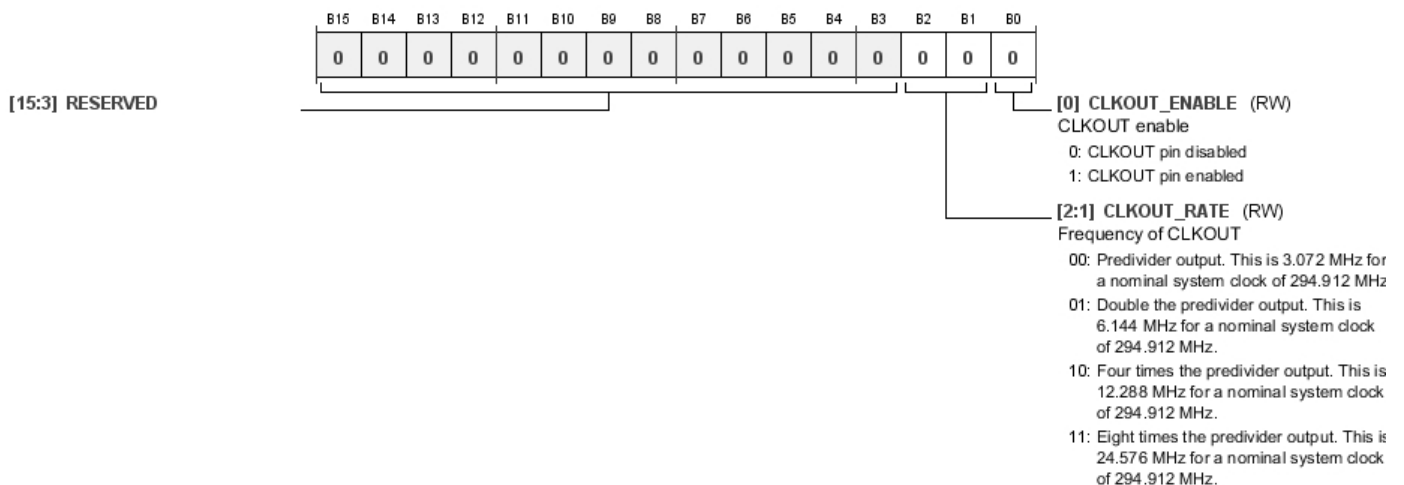


Table 65. Bit Descriptions for MCLK_OUT

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|---------------|--|--|-------|--------|
| [15:3] | RESERVED | | | 0x0 | RW |
| [2:1] | CLKOUT_RATE | 00 Predivider output. This is 3.072 MHz for a nominal system clock of 294.912 MHz. 01 Double the predivider output. This is 6.144 MHz for a nominal system clock of 294.912 MHz. 10 Four times the predivider output. This is 12.288 MHz for a nominal system clock of 294.912 MHz. 11 Eight times the predivider output. This is 24.576 MHz for a nominal system clock of 294.912 MHz. | Frequency of CLKOUT. Frequency of the signal output from the CLKOUT pin. These bits set the frequency of the signal on the CLKOUT pin. The frequencies documented in Table 65 are examples that are valid for a master clock input that is a binary multiple of 3.072 MHz. In this case, the options for output rates are 3.072 MHz, 6.144 MHz, 12.288 MHz, or 24.576 MHz. If the input master clock is scaled down (for example, to a binary multiple of 2.8224 MHz), the possible output rates are 2.8224 MHz, 5.6448 MHz, 11.2896 MHz, or 22.5792 MHz). | 0x0 | RW |
| 0 | CLKOUT_ENABLE | 0 CLKOUT pin disabled 1 CLKOUT pin enabled | CLKOUT enable. When this bit is enabled, a clock signal is output from the CLKOUT pin of the device. When disabled, the CLKOUT pin is high impedance. | 0x0 | RW |

Analog PLL Watchdog Control Register

Address: 0xF006, Reset: 0x0001, Name: PLL_WATCHDOG

The PLL watchdog is a feature that monitors the PLL and automatically resets it in the event that it reaches an unstable condition. The PLL resets itself and automatically attempts to lock to the incoming clock signal again, with the same settings as before. This functionality requires no interaction on the part of the user. Ensure that the PLL watchdog is enabled at all times.

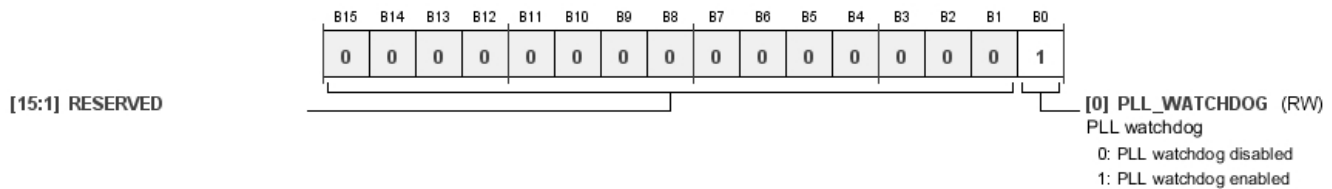


Table 66. Bit Descriptions for PLL_WATCHDOG

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|--------------|---|---------------|-------|--------|
| [15:1] | RESERVED | | | 0x0 | RW |
| 0 | PLL_WATCHDOG | 0 PLL watchdog disabled 1 PLL watchdog enabled | PLL watchdog. | 0x1 | RW |

CLOCK GENERATOR REGISTERS

Denominator (M) for Clock Generator 1 Register

Address: 0xF020, Reset: 0x0006, Name: CLK_GEN1_M

The denominator (M) for Clock Generator 1.

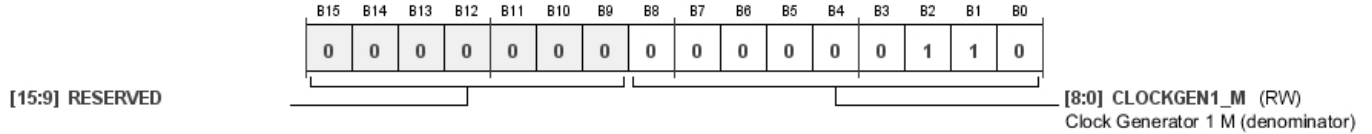


Table 67. Bit Descriptions for CLK_GEN1_M

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-------------|----------|--|-------|--------|
| [15:9] | RESERVED | | | 0x0 | RW |
| [8:0] | CLOCKGEN1_M | | Clock Generator 1 M (denominator). Format is binary integer. | 0x006 | RW |

Numerator (N) for Clock Generator 1 Register

Address: 0xF021, Reset: 0x0001, Name: CLK_GEN1_N

The numerator (N) for Clock Generator 1.

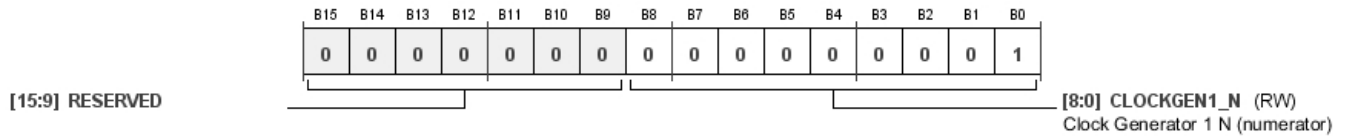


Table 68. Bit Descriptions for CLK_GEN1_N

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-------------|----------|--|-------|--------|
| [15:9] | RESERVED | | | 0x0 | RW |
| [8:0] | CLOCKGEN1_N | | Clock Generator 1 N (numerator). Format is binary integer. | 0x001 | RW |

Denominator (M) for Clock Generator 2 Register

Address: 0xF022, Reset: 0x0009, Name: CLK_GEN2_M

The denominator (M) for Clock Generator 2.

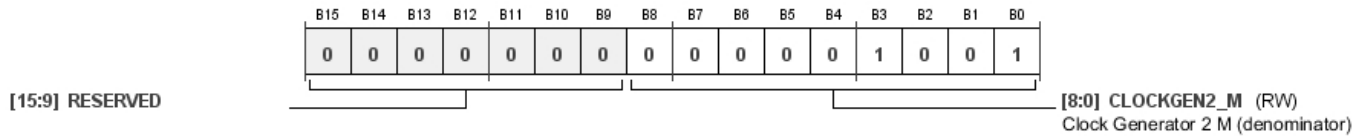


Table 69. Bit Descriptions for CLK_GEN2_M

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-------------|----------|--|-------|--------|
| [15:9] | RESERVED | | | 0x0 | RW |
| [8:0] | CLOCKGEN2_M | | Clock Generator 2 M (denominator). Format is binary integer. | 0x009 | RW |

Numerator (N) for Clock Generator 2 Register

Address: 0xF023, Reset: 0x0001, Name: CLK_GEN2_N

The numerator (N) for Clock Generator 2.

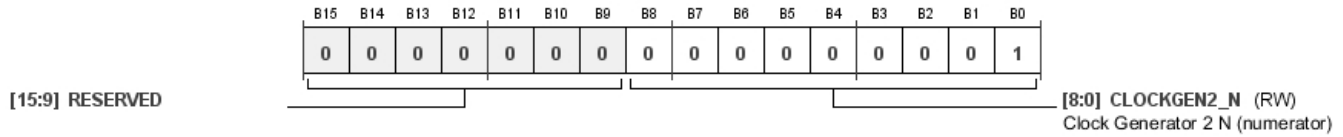


Table 70. Bit Descriptions for CLK_GEN2_N

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-------------|----------|--|-------|--------|
| [15:9] | RESERVED | | | 0x0 | RW |
| [8:0] | CLOCKGEN2_N | | Clock Generator 2 N (numerator). Format is binary integer. | 0x001 | RW |

Denominator (M) for Clock Generator 3 Register

Address: 0xF024, Reset: 0x0000, Name: CLK_GEN3_M

The denominator (M) for Clock Generator 3.

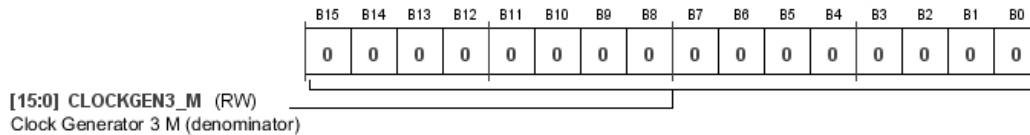


Table 71. Bit Descriptions for CLK_GEN3_M

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-------------|----------|--|--------|--------|
| [15:0] | CLOCKGEN3_M | | Clock Generator 3 M (denominator). Format is binary integer. | 0x0000 | RW |

Numerator for (N) Clock Generator 3 Register

Address: 0xF025, Reset: 0x0000, Name: CLK_GEN3_N

The numerator (N) for Clock Generator 3.

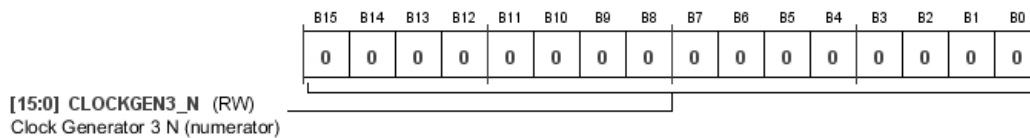


Table 72. Bit Descriptions for CLK_GEN3_N

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-------------|----------|--|--------|--------|
| [15:0] | CLOCKGEN3_N | | Clock Generator 3 N (numerator). Format is binary integer. | 0x0000 | RW |

Input Reference for Clock Generator 3 Register

Address: 0xF026, Reset: 0x000E, Name: CLK_GEN3_SRC

Clock Generator 3 can generate audio clocks using the PLL output (system clock) as a reference, or it can optionally use a reference clock entering the device from an external source either on a multipurpose pin (MPx) or the S/PDIF receiver. This register determines the source of the reference signal.

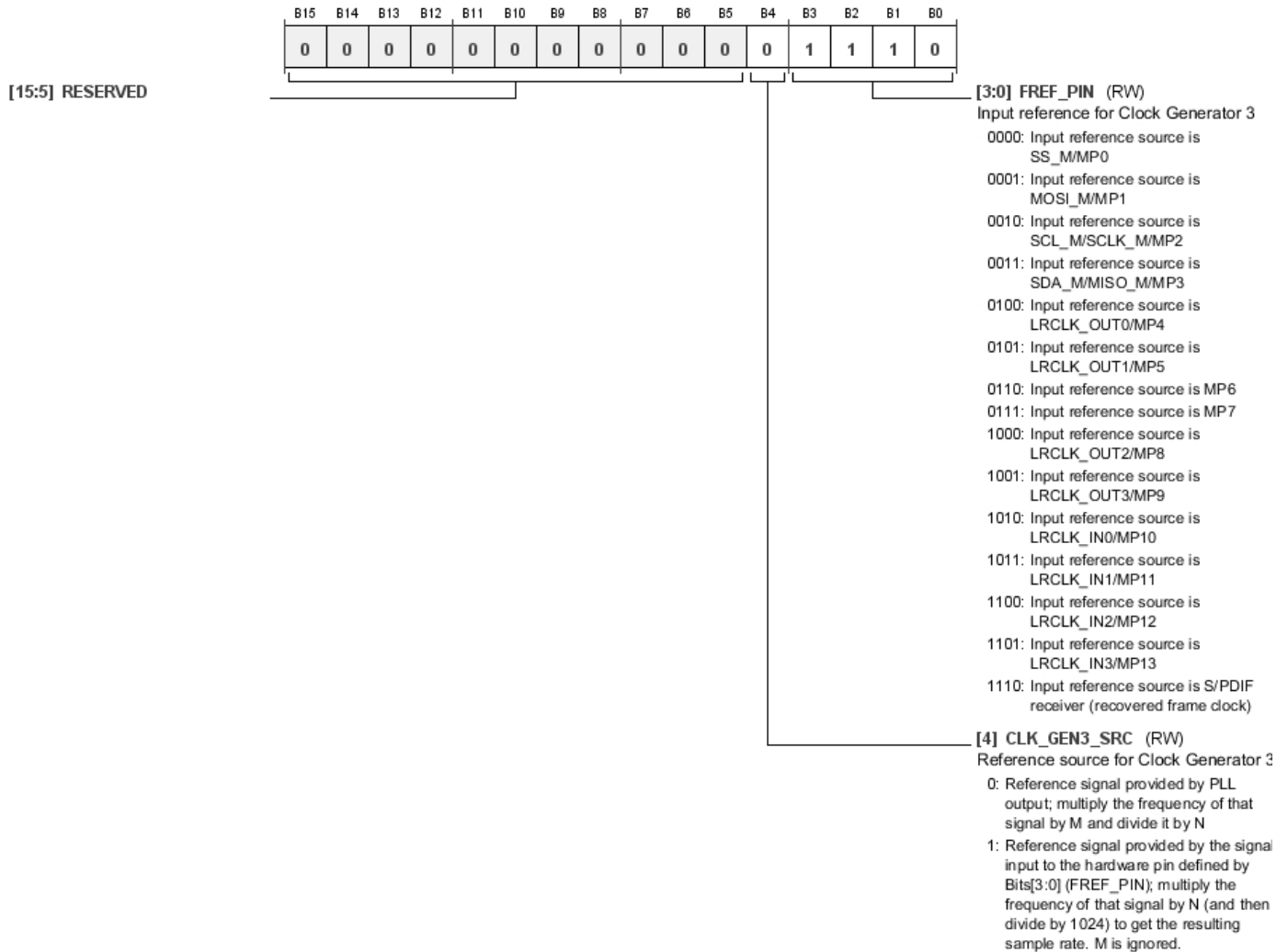


Table 73. Bit Descriptions for CLK_GEN3_SRC

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|----------|----------|-------------|-------|--------|
| [15:5] | RESERVED | | | 0x0 | RW |

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|--------------|----------|---|-------|--------|
| 4 | CLK_GEN3_SRC | | Reference source for Clock Generator 3. This bit selects the reference of Clock Generator 3. If set to use an external reference clock, Bits[3:0] define the source pin. Otherwise, the PLL output is used as the reference clock. When an external reference clock is used for Clock Generator 3, the resulting base output frequency of Clock Generator 3 is the frequency of the input reference clock multiplied by the Clock Generator 3 numerator, divided by 1024. For example: if Bit 4 (CLK_GEN3_SRC) = 0b1 (an external reference clock is used); Bits[3:0] (FREF_PIN) = 0b1110 (the input signal of the S/PDIF receiver is used as the reference source); the sample rate of the S/PDIF input signal = 48 kHz; and the numerator of Clock Generator 3 = 2048; the resulting base output sample rate of Clock Generator 3 is 48 kHz × 2048/1024 = 96 kHz. 0 Reference signal provided by PLL output; multiply the frequency of that signal by M and divide it by N 1 Reference signal provided by the signal input to the hardware pin defined by Bits[3:0] (FREF_PIN); multiply the frequency of that signal by N (and then divide by 1024) to get the resulting sample rate; M is ignored | 0x0 | RW |
| [3:0] | FREF_PIN | | Input reference for Clock Generator 3. If Clock Generator 3 is set up to lock to an external reference clock (Bit 4 (CLK_GEN3_SRC) = 0b1), these bits allow the user to specify which pin is receiving the reference clock. The signal input to the corresponding pin should be a 50% duty cycle square wave clock representing the reference sample rate. 0000 Input reference source is SS_M/MP0 0001 Input reference source is MOSI_M/MP1 0010 Input reference source is SCL_M/SCLK_M/MP2 0011 Input reference source is SDA_M/MISO_M/MP3 0100 Input reference source is LRCLK_OUT0/MP4 0101 Input reference source is LRCLK_OUT1/MP5 0110 Input reference source is MP6 0111 Input reference source is MP7 1000 Input reference source is LRCLK_OUT2/MP8 1001 Input reference source is LRCLK_OUT3/MP9 1010 Input reference source is LRCLK_IN0/MP10 1011 Input reference source is LRCLK_IN1/MP11 1100 Input reference source is LRCLK_IN2/MP12 1101 Input reference source is LRCLK_IN3/MP13 1110 Input reference source is S/PDIF receiver (recovered frame clock) | 0xE | RW |

Lock Bit for Clock Generator 3 Input Reference Register

Address: 0xF027, Reset: 0x0000, Name: CLK_GEN3_LOCK

This register monitors whether or not Clock Generator 3 has locked to its reference clock source, regardless of whether it is coming from the PLL output or from an external reference signal, which is configured in Register 0xF026, Bit 4 (CLK_GEN3_SRC).

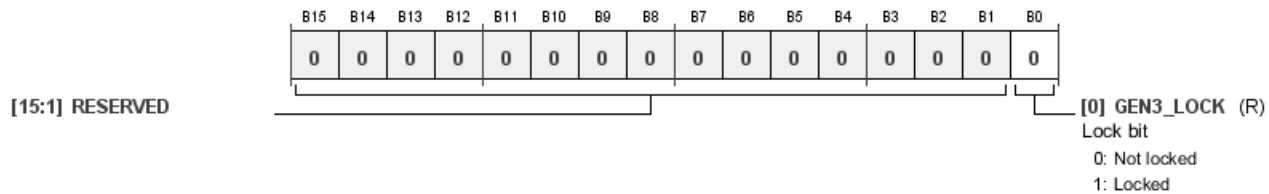


Table 74. Bit Descriptions for CLK_GEN3_LOCK

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-----------|----------|---------------------------------------|-------|--------|
| [15:1] | RESERVED | | | 0x0 | RW |
| 0 | GEN3_LOCK | | Lock bit. 0 Not locked 1 Locked | 0x0 | R |

POWER REDUCTION REGISTERS

Power Enable 0 Register

Address: 0xF050, Reset: 0x0000, Name: POWER_ENABLE0

For the purpose of power savings, this register allows the clock generators, ASRCs, and serial ports to be disabled when not in use. When these functional blocks are disabled, the current draw on the corresponding supply pins decreases.

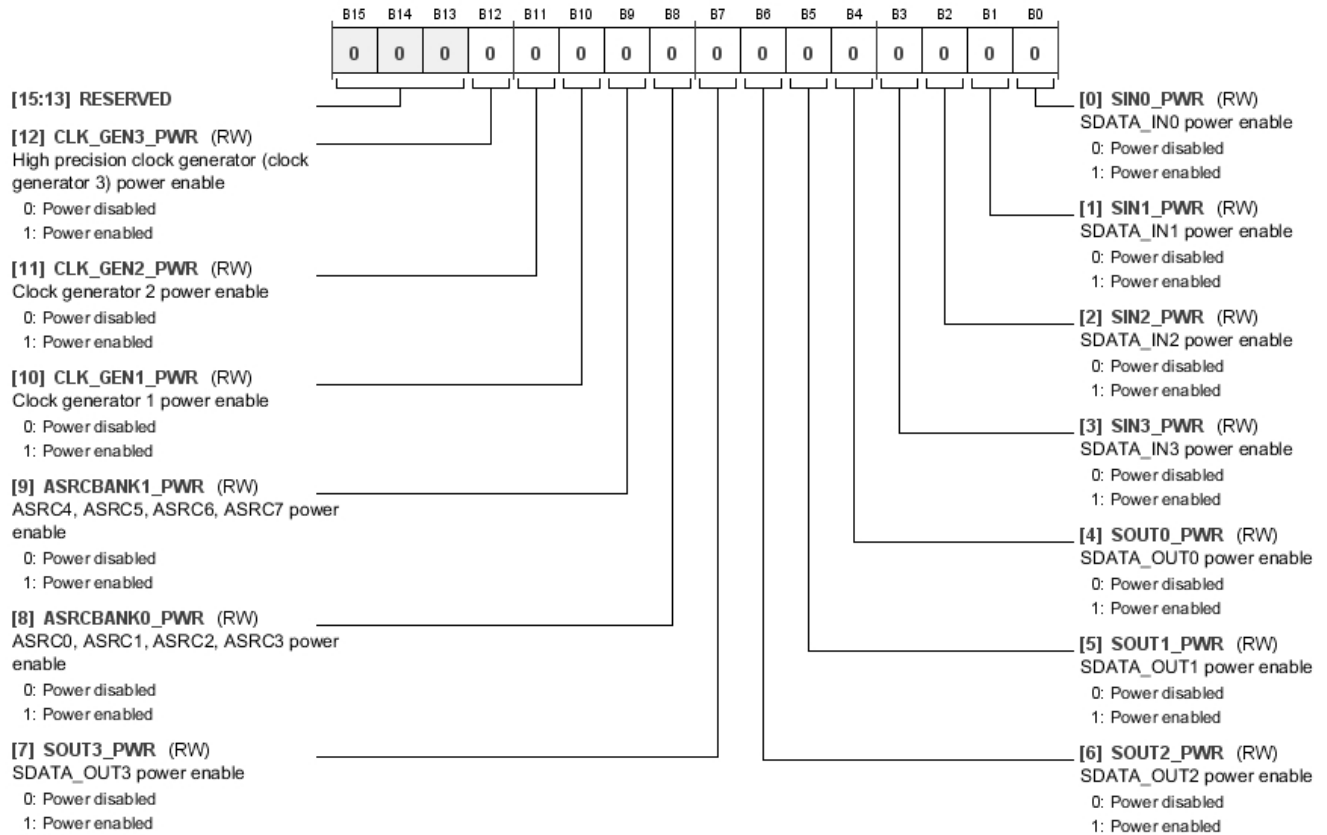


Table 75. Bit Descriptions for POWER_ENABLE0

| Bits | Bit Name | Settings | Description | Reset | Access |
|---------|--------------|----------|--|-------|--------|
| [15:13] | RESERVED | | | 0x0 | RW |
| 12 | CLK_GEN3_PWR | 0 1 | High precision clock generator (Clock Generator 3) power enable. When this bit is disabled, Clock Generator 3 is disabled and ceases to output audio clocks. Any functional block in hardware, including the DSP core, that has been configured to be clocked by Clock Generator 3 ceases to function while this bit is disabled. Power disabled Power enabled | 0x0 | RW |
| 11 | CLK_GEN2_PWR | 0 1 | Clock Generator 2 power enable. When this bit is disabled, Clock Generator 2 is disabled and ceases to output audio clocks. Any LRCLK_OUTx, LRCLK_INx or BCLK_OUTx, BCLK_INx pins that have been configured to output clocks generated by Clock Generator 2 output a logic low signal while Clock Generator 2 is disabled. Any functional block in hardware, including the DSP core, that has been configured to be clocked by Clock Generator 2 ceases to function while this bit is disabled. Power disabled Power enabled | 0x0 | RW |

| Bits | Bit Name | Settings | Description | Reset | Access |
|------|---------------|----------|---|-------|--------|
| 10 | CLK_GEN1_PWR | 0 1 | Clock Generator 1 power enable. When this bit is disabled, Clock Generator 1 is disabled and ceases to output audio clocks. Any LRCLK_OUTx, LRCLK_INx or BCLK_OUTx, BCLK_INx pins that are configured to output clocks generated by Clock Generator 1 output a logic low signal while Clock Generator 1 is disabled. Any functional block in hardware, including the DSP core, that is configured to be clocked by Clock Generator 1 ceases to function when this bit is disabled. Power disabled Power enabled | 0x0 | RW |
| 9 | ASRCBANK1_PWR | 0 1 | ASRC 4, ASRC 5, ASRC 6, ASRC 7 power enable. When this bit is disabled, ASRC Channel 8 to Channel 15 are disabled, and their output data streams cease. Power disabled Power enabled | 0x0 | RW |
| 8 | ASRCBANK0_PWR | 0 1 | ASRC 0, ASRC 1, ASRC 2, ASRC 3 power enable. When this bit is disabled, ASRC Channel 0 to Channel 7 are disabled, and their output data streams cease. Power disabled Power enabled | 0x0 | RW |
| 7 | SOUT3_PWR | 0 1 | SDATA_OUT3 power enable. When this bit is disabled, the SDATA_OUT3 pin and associated serial port circuitry are also disabled. LRCLK_OUT3 and BCLK_OUT3 are not affected. Power disabled Power enabled | 0x0 | RW |
| 6 | SOUT2_PWR | 0 1 | SDATA_OUT2 power enable. When this bit is disabled, the SDATA_OUT2 pin and associated serial port circuitry is disabled. LRCLK_OUT2 and BCLK_OUT2 are not affected. Power disabled Power enabled | 0x0 | RW |
| 5 | SOUT1_PWR | 0 1 | SDATA_OUT1 power enable. When this bit is disabled, the SDATA_OUT1 pin and associated serial port circuitry are also disabled. LRCLK_OUT1 and BCLK_OUT1 are not affected. Power disabled Power enabled | 0x0 | RW |
| 4 | SOUT0_PWR | 0 1 | SDATA_OUT0 power enable. When this bit is disabled, the SDATA_OUT0 pin and associated serial port circuitry are disabled. LRCLK_OUT0 and BCLK_OUT0 are not affected. Power disabled Power enabled | 0x0 | RW |
| 3 | SIN3_PWR | 0 1 | SDATA_IN3 power enable. When this bit is disabled, the SDATA_IN3 pin and associated serial port circuitry are disabled. LRCLK_IN3 and BCLK_IN3 are not affected. Power disabled Power enabled | 0x0 | RW |
| 2 | SIN2_PWR | 0 1 | SDATA_IN2 power enable. When this bit is disabled, the SDATA_IN2 pin and associated serial port circuitry are disabled. LRCLK_IN2 and BCLK_IN2 are not affected. Power disabled Power enabled | 0x0 | RW |
| 1 | SIN1_PWR | 0 1 | SDATA_IN1 power enable. When this bit is disabled, the SDATA_IN1 pin and associated serial port circuitry are disabled. The LRCLK_IN1 and BCLK_IN1 pins are not affected. Power disabled Power enabled | 0x0 | RW |
| 0 | SIN0_PWR | 0 1 | SDATA_IN0 power enable. When this bit is disabled, the SDATA_IN0 pin and associated serial port circuitry are disabled. The LRCLK_IN0 and BCLK_IN0 pins are not affected. Power disabled Power enabled | 0x0 | RW |

Power Enable 1 Register

Address: 0xF051, Reset: 0x0000, Name: POWER_ENABLE1

For the purpose of power savings, this register allows the PDM microphone interfaces, S/PDIF interfaces, and auxiliary ADCs to be disabled when not in use. When these functional blocks are disabled, the current draw on the corresponding supply pins decreases.

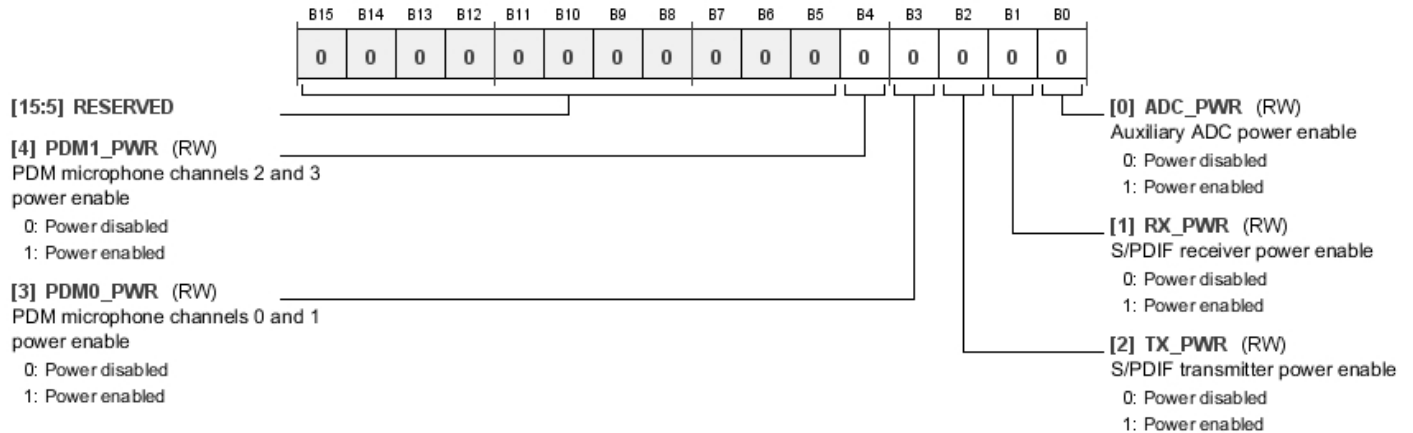


Table 76. Bit Descriptions for POWER_ENABLE1

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|----------|----------|---|-------|--------|
| [15:5] | RESERVED | | | 0x0 | RW |
| 4 | PDM1_PWR | 0 1 | PDM Microphone Channel 2 and PDM Microphone Channel 3 power enable. When this bit is disabled, PDM Microphone Channel 2 and PDM Microphone Channel 3 and their associated circuitry are disabled, and their data values cease to update. 0 Power disabled 1 Power enabled | 0x0 | RW |
| 3 | PDM0_PWR | 0 1 | PDM Microphone Channel 0 and PDM Microphone Channel 1 power enable. When this bit is disabled, PDM Microphone Channel 0 and PDM Microphone Channel 1 and their associated circuitry are disabled, and their data values cease to update. 0 Power disabled 1 Power enabled | 0x0 | RW |
| 2 | TX_PWR | 0 1 | S/PDIF transmitter power enable. This bit disables the S/PDIF transmitter circuit. Clock and data ceases to output from the S/PDIF transmitter pin, and the output is held at logic low as long as this bit is disabled. 0 Power disabled 1 Power enabled | 0x0 | RW |
| 1 | RX_PWR | 0 1 | S/PDIF receiver power enable. This bit disables the S/PDIF receiver circuit. Clock and data recovery from the S/PDIF input stream ceases until this bit is reenabled. 0 Power disabled 1 Power enabled | 0x0 | RW |
| 0 | ADC_PWR | 0 1 | Auxiliary ADC power enable. When this bit is disabled, the auxiliary ADCs are powered down, their outputs cease to update, and they hold their last value. 0 Power disabled 1 Power enabled | 0x0 | RW |

AUDIO SIGNAL ROUTING REGISTERS

ASRC Input Selector Register

Address: 0xF100 to Address 0xF107 (Increments of 0x1), Reset: 0x0000, Name: ASRC_INPUTx

These eight registers configure the input signal to the corresponding eight stereo ASRCs on the ADAU1452. ASRC_INPUT0 configures ASRC Channel 0 and ASRC Channel 1, ASRC_INPUT1 configures ASRC Channel 2 and ASRC Channel 3, and so on. Valid input signals to the ASRCs include Serial Input Channel 0 to Serial Input Channel 47, the PDM Microphone Input Channel 0 to PDM Microphone Input Channel 3, and the S/PDIF Receiver Channel 0 to S/PDIF Receiver Channel 1.

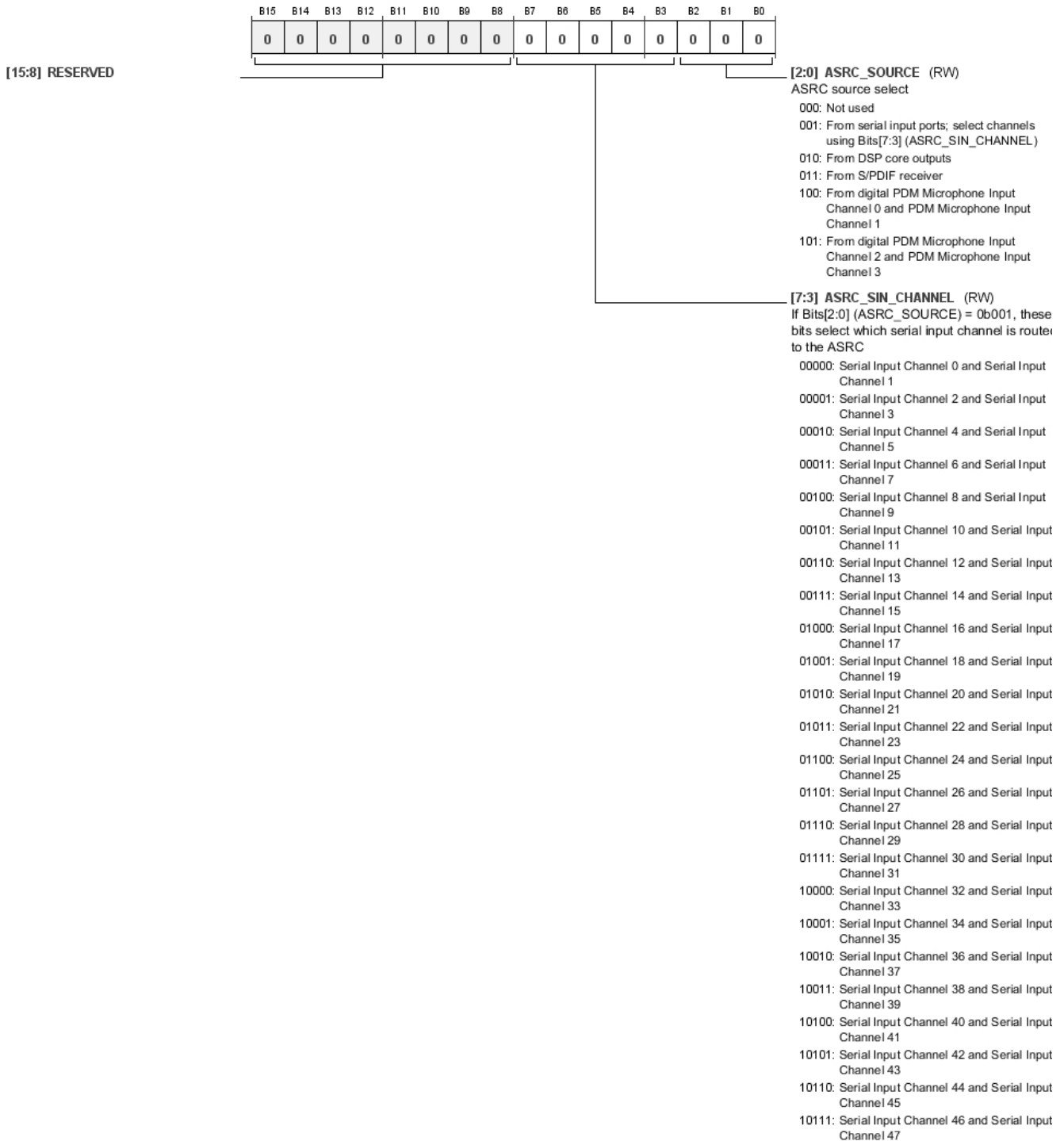


Table 77. Bit Descriptions for ASRC_INPUTx

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|------------------|--|---|-------|--------|
| [15:8] | RESERVED | | | 0x0 | RW |
| [7:3] | ASRC_SIN_CHANNEL | 00000 Serial Input Channel 0 and Serial Input Channel 1 00001 Serial Input Channel 2 and Serial Input Channel 3 00010 Serial Input Channel 4 and Serial Input Channel 5 00011 Serial Input Channel 6 and Serial Input Channel 7 00100 Serial Input Channel 8 and Serial Input Channel 9 00101 Serial Input Channel 10 and Serial Input Channel 11 00110 Serial Input Channel 12 and Serial Input Channel 13 00111 Serial Input Channel 14 and Serial Input Channel 15 01000 Serial Input Channel 16 and Serial Input Channel 17 01001 Serial Input Channel 18 and Serial Input Channel 19 01010 Serial Input Channel 20 and Serial Input Channel 21 01011 Serial Input Channel 22 and Serial Input Channel 23 01100 Serial Input Channel 24 and Serial Input Channel 25 01101 Serial Input Channel 26 and Serial Input Channel 27 01110 Serial Input Channel 28 and Serial Input Channel 29 01111 Serial Input Channel 30 and Serial Input Channel 31 10000 Serial Input Channel 32 and Serial Input Channel 33 10001 Serial Input Channel 34 and Serial Input Channel 35 10010 Serial Input Channel 36 and Serial Input Channel 37 10011 Serial Input Channel 38 and Serial Input Channel 39 10100 Serial Input Channel 40 and Serial Input Channel 41 10101 Serial Input Channel 42 and Serial Input Channel 43 10110 Serial Input Channel 44 and Serial Input Channel 45 10111 Serial Input Channel 46 and Serial Input Channel 47 | If Bits[2:0] (ASRC_SOURCE) = 0b001, these bits select which serial input channel is routed to the ASRC. | 0x00 | RW |
| [2:0] | ASRC_SOURCE | 000 Not used 001 From serial input ports; select channels using Bits[7:3] (ASRC_SIN_CHANNEL) 010 From DSP core outputs 011 From S/PDIF receiver 100 From digital PDM Microphone Input Channel 0 and PDM Microphone Input Channel 1 101 From digital PDM Microphone Input Channel 2 and PDM Microphone Input Channel 3 | ASRC source select. | 0x0 | RW |

ASRC Output Rate Selector Register

Address: 0xF140 to Address 0xF147 (Increments of 0x1), Reset: 0x0000, Name: ASRC_OUT_RATEx

These eight registers configure the target output sample rates of the corresponding eight stereo ASRCs on the ADAU1452. The ASRC takes any arbitrary input sample rate and automatically attempts to resample the data in that signal and output it at the target sample rate as configured by these registers. Each of the eight registers corresponds to one of the eight stereo ASRCs. ASRC_OUT_RATE0 configures ASRC Channel 0 and ASRC Channel 1, ASRC_INPUT1 configures ASRC Channel 2 and ASRC Channel 3, ASRC_OUT_RATE2 configures ASRC Channel 4 and ASRC Channel 5, ASRC_OUT_RATE3 configures ASRC Channel 6 and ASRC Channel 7, ASRC_OUT_RATE4 configures ASRC Channel 8 and ASRC Channel 9, ASRC_OUT_RATE5 configures ASRC Channel 10 and ASRC Channel 11, ASRC_OUT_RATE6 configures ASRC Channel 12 and ASRC Channel 13, and ASRC_OUT_RATE7 configures ASRC Channel 14 and ASRC Channel 15. The ASRCs lock their output frequencies to the audio sample rates of any of the serial output ports, the DSP start pulse rate of the core, or one of several internally generated sample rates coming from the clock generators.

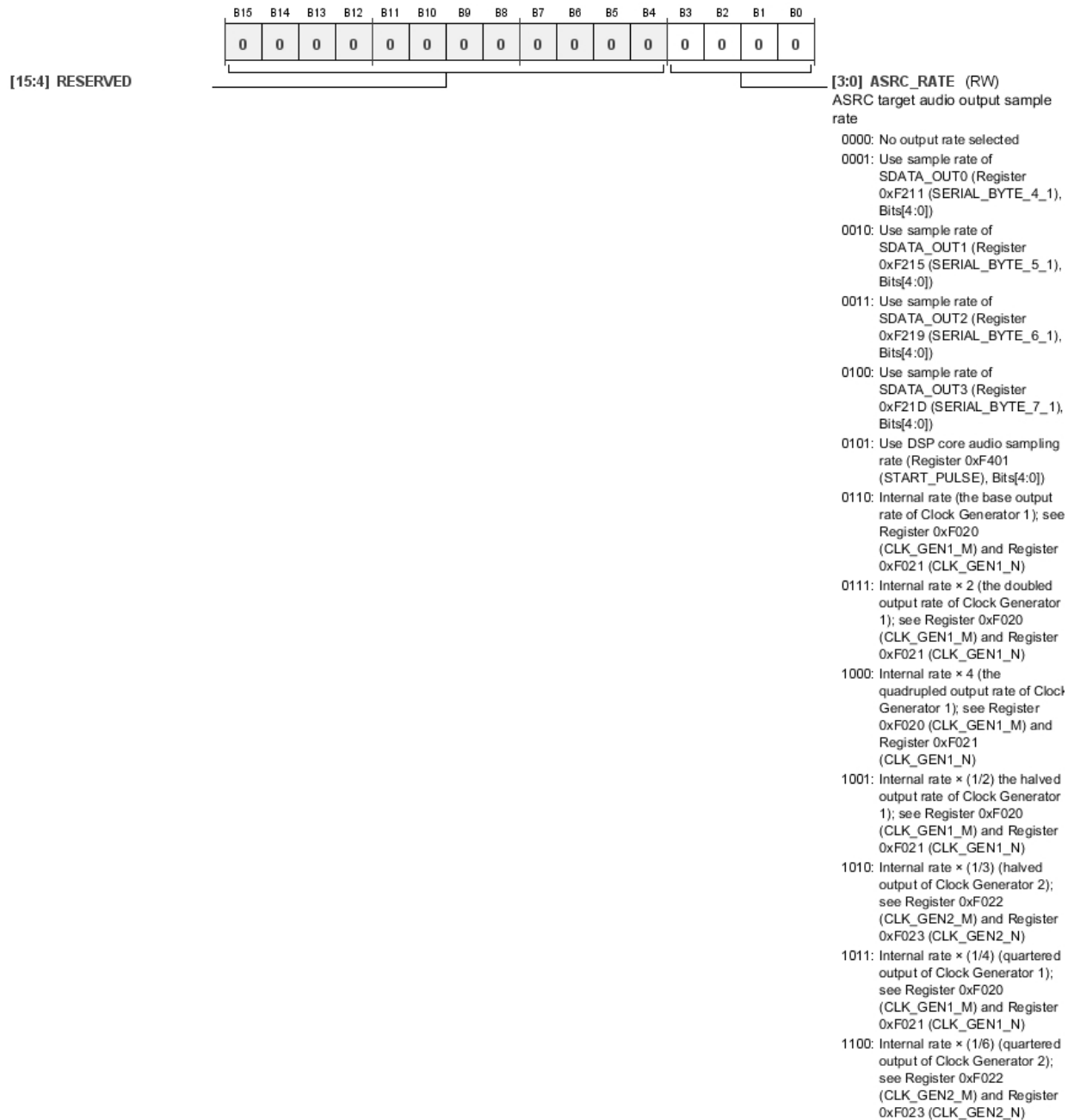


Table 78. Bit Descriptions for ASRC_OUT_RATEx

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-----------|----------|--|-------|--------|
| [15:4] | RESERVED | | | 0x0 | RW |
| [3:0] | ASRC_RATE | | ASRC target audio output sample rate. The corresponding ASRC can lock its output to a serial output port, the DSP core, or an internally generated rate. | 0x0 | RW |
| | | 0000 | No output rate selected | | |
| | | 0001 | Use sample rate of SDATA_OUT0 (Register 0xF211 (SERIAL_BYTE_4_1), Bits[4:0]) | | |
| | | 0010 | Use sample rate of SDATA_OUT1 (Register 0xF215 (SERIAL_BYTE_5_1), Bits[4:0]) | | |
| | | 0011 | Use sample rate of SDATA_OUT2 (Register 0xF219 (SERIAL_BYTE_6_1), Bits[4:0]) | | |
| | | 0100 | Use sample rate of SDATA_OUT3 (Register 0xF21D (SERIAL_BYTE_7_1), Bits[4:0]) | | |
| | | 0101 | Use DSP core audio sampling rate (Register 0xF401 (START_PULSE), Bits[4:0]) | | |
| | | 0110 | Internal rate (the base output rate of Clock Generator 1); see Register 0xF020 (CLK_GEN1_M) and Register 0xF021 (CLK_GEN1_N) | | |
| | | 0111 | Internal rate $\times 2$ (the doubled output rate of Clock Generator 1); see Register 0xF020 (CLK_GEN1_M) and Register 0xF021 (CLK_GEN1_N) | | |
| | | 1000 | Internal rate $\times 4$ (the quadrupled output rate of Clock Generator 1); see Register 0xF020 (CLK_GEN1_M) and Register 0xF021 (CLK_GEN1_N) | | |
| | | 1001 | Internal rate $\times (1/2)$ the halved output rate of Clock Generator 1; see Register 0xF020 (CLK_GEN1_M) and Register 0xF021 (CLK_GEN1_N) | | |
| | | 1010 | Internal rate $\times (1/3)$ (halved output of Clock Generator 2); see Register 0xF022 (CLK_GEN2_M) and Register 0xF023 (CLK_GEN2_N) | | |
| | | 1011 | Internal rate $\times (1/4)$ (quartered output of Clock Generator 1); see Register 0xF020 (CLK_GEN1_M) and Register 0xF021 (CLK_GEN1_N) | | |
| | | 1100 | Internal rate $\times (1/6)$ (quartered output of Clock Generator 2); see Register 0xF022 (CLK_GEN2_M) and Register 0xF023 (CLK_GEN2_N) | | |

Source of Data for Serial Output Ports Register

Address: 0xF180 to 0xF197 (Increments of 0x1), Reset: 0x0000, Name: SOUT_SOURCEx

These 24 registers correspond to the 24 pairs of output channels used by the serial output ports. Each register corresponds to two audio channels. SOUT_SOURCE0 corresponds to Channel 0 and Channel 1, SOUT_SOURCE1 corresponds to Channel 2 and Channel 3, and so on. SOUT_SOURCE0 to SOUT_SOURCE7 map to the 16 total channels (Channel 0 to Channel 15) that are fed to SDATA_OUT0. SOUT_SOURCE8 to SOUT_SOURCE15 map to the 16 total channels (Channel 16 to Channel 31) that are fed to SDATA_OUT1. SOUT_SOURCE16 to SOUT_SOURCE19 map to the eight total channels (Channel 32 to Channel 39) that are fed to SDATA_OUT2. SOUT_SOURCE20 to SOUT_SOURCE23 map to the eight total channels (Channel 40 to Channel 47) that are fed to SDATA_OUT3. Data originates from several places, including directly from the corresponding input audio channels from the serial input ports, from the corresponding audio output channels of the DSP core, from an ASRC output pair, or directly from the PDM microphone inputs.

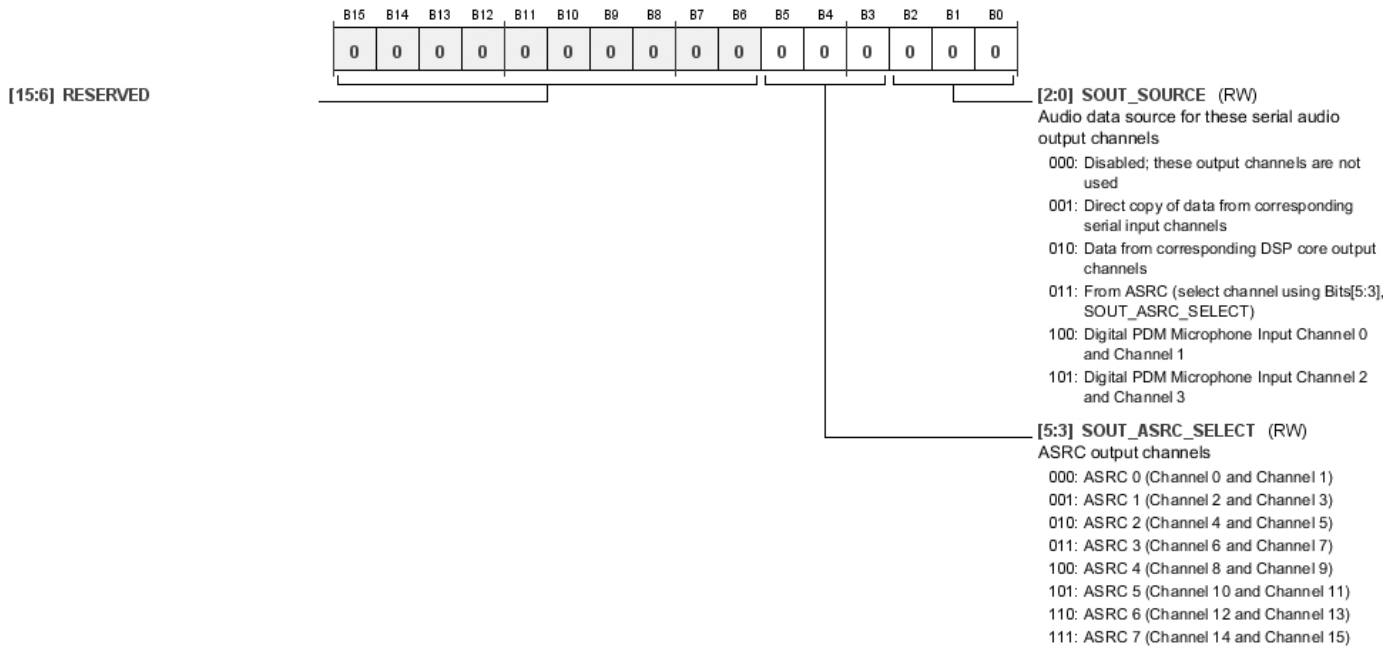


Table 79. Bit Descriptions for SOUT_SOURCEx

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|------------------|--|--|-------|--------|
| [15:6] | RESERVED | | | 0x000 | RW |
| [5:3] | SOUT_ASRC_SELECT | 000 001 010 011 100 101 110 111 | ASRC output channels. If Bits[2:0] (SOUT_SOURCE) are set to 0b011, these bits select which ASRC channels are routed to the serial output channels. | 0x0 | RW |

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|-------------|--|---|-------|--------|
| [2:0] | SOUT_SOURCE | 000 Disabled; these output channels are not used 001 Direct copy of data from corresponding serial input channels 010 Data from corresponding DSP core output channels 011 From ASRC (select channel using Bits[5:3], SOUT_ASRC_SELECT) 100 Digital PDM Microphone Input Channel 0 and Digital PDM Microphone Input Channel 1 101 Digital PDM Microphone Input Channel 2 and Digital PDM Microphone Input Channel 3 | Audio data source for these serial audio output channels. If these bits are set to 0b001, the corresponding output channels output a copy of the data from the corresponding input channels. For example, if Address 0xF180, Bits[2:0] are set to 0b001, Serial Input Channel 0 and Serial Input Channel 1 copy to Serial Output Channel 0 and Serial Output Channel 1, respectively. If these bits are set to 0b010, DSP Output Channel 0 and DSP Output Channel 1 copy to Serial Output Channel 0 and Serial Output Channel 1, respectively. If these bits are set to 0b011, Bits[5:3] (SOUT_ASRC_SELECT) must be configured to select the desired ASRC output. | 0x0 | RW |

S/PDIF Transmitter Data Selector Register

Address: 0xF1C0, Reset: 0x0000, Name: SPDIFTX_INPUT

This register configures which data source feeds the S/PDIF transmitter. Data can originate from the S/PDIF outputs of the DSP core or directly from the S/PDIF receiver.

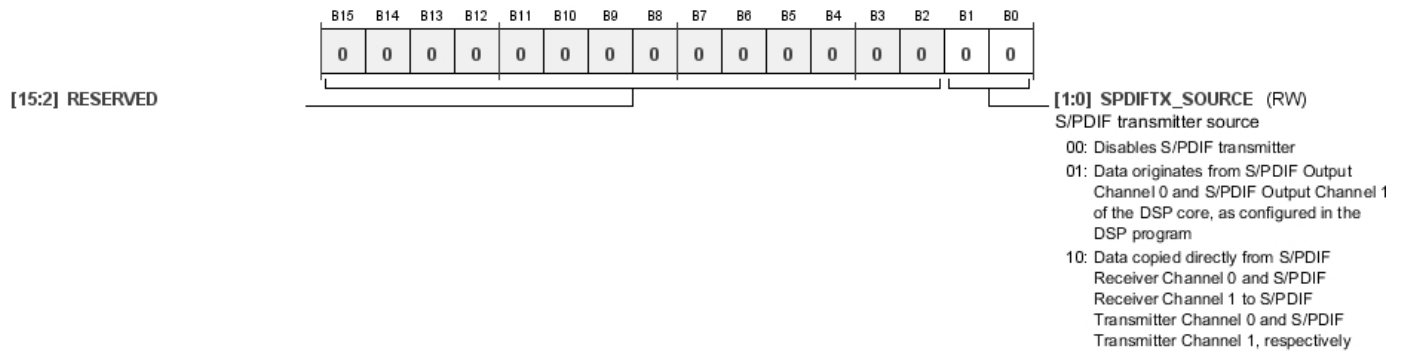


Table 80. Bit Descriptions for SPDIFTX_INPUT

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|----------------|--|----------------------------|-------|--------|
| [15:2] | RESERVED | | | 0x0 | RW |
| [1:0] | SPDIFTX_SOURCE | 00 Disables S/PDIF transmitter 01 Data originates from S/PDIF Output Channel 0 and S/PDIF Output Channel 1 of the DSP core, as configured in the DSP program 10 Data copied directly from S/PDIF Receiver Channel 0 and S/PDIF Receiver Channel 1 to S/PDIF Transmitter Channel 0 and S/PDIF Transmitter Channel 1, respectively | S/PDIF transmitter source. | 0x0 | RW |

SERIAL PORT CONFIGURATION REGISTERS

Serial Port Control 0 Register

Address: 0xF200 to 0xF21C (Increments of 0x4), Reset: 0x0000, Name: SERIAL_BYTE_x_0

These eight registers configure several settings for the corresponding serial input and serial output ports. Channel count, MSB position, data-word length, clock polarity, clock sources, and clock type are configured using these registers. On the input side, Register 0xF200 (SERIAL_BYTE_0_0) corresponds to SDATA_IN0; Register 0xF204 (SERIAL_BYTE_1_0) corresponds to SDATA_IN1; Register 0xF208 (SERIAL_BYTE_2_0) corresponds to SDATA_IN2; and Register 0xF20C (SERIAL_BYTE_3_0) corresponds to SDATA_IN3. On the output side, Register 0xF210 (SERIAL_BYTE_4_0) corresponds to SDATA_OUT0; Register 0xF214 (SERIAL_BYTE_5_0) corresponds to SDATA_OUT1; Register 0xF218 (SERIAL_BYTE_6_0) corresponds to SDATA_OUT2; and Register 0xF21C (SERIAL_BYTE_7_0) corresponds to SDATA_OUT3.

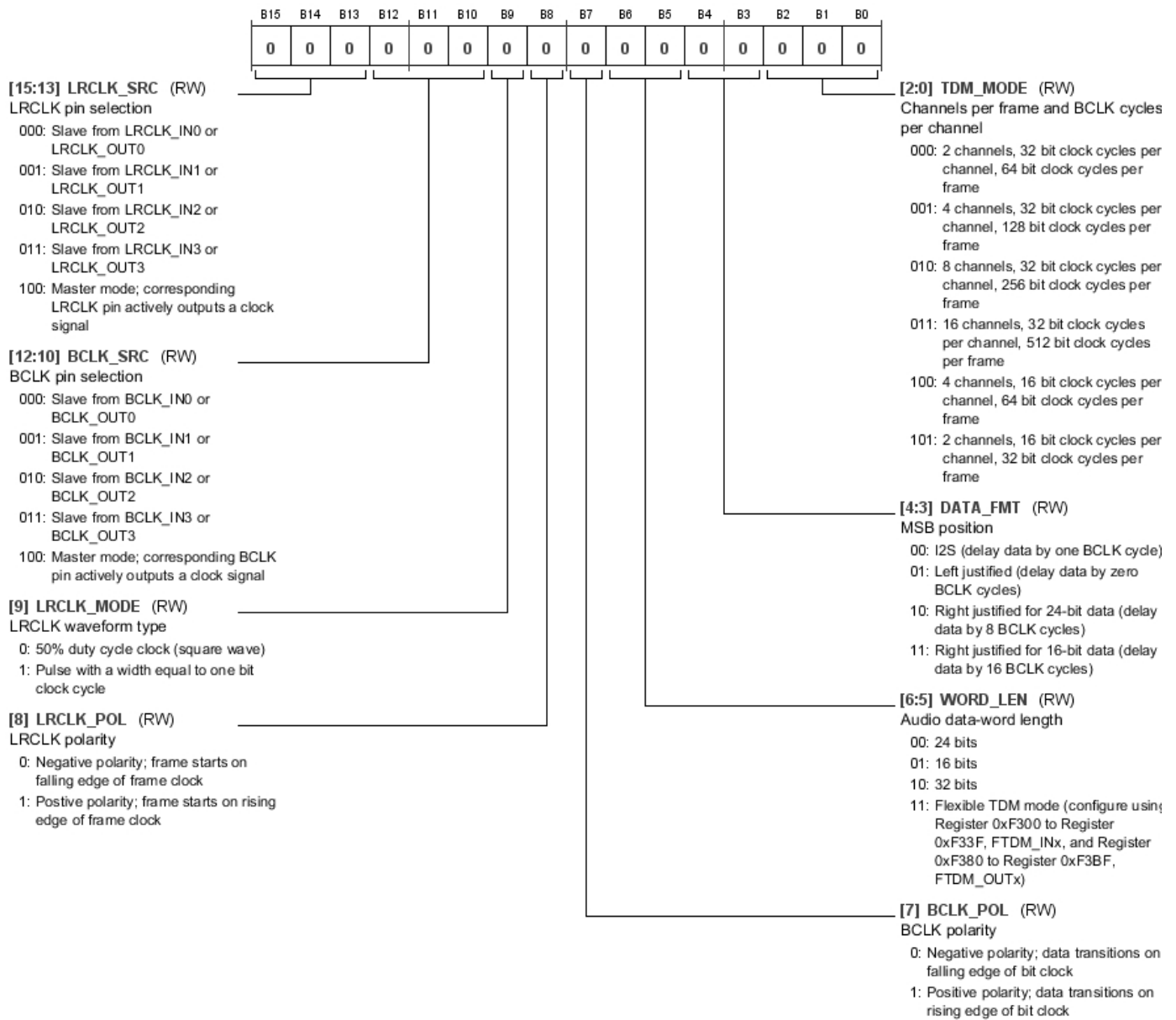


Table 81. Bit Descriptions for SERIAL_BYTE_x_0

| Bits | Bit Name | Settings | Description | Reset | Access |
|---------|------------|--|--|-------|--------|
| [15:13] | LRCLK_SRC | 000 Slave from LRCLK_IN0 or LRCLK_OUT0 001 Slave from LRCLK_IN1 or LRCLK_OUT1 010 Slave from LRCLK_IN2 or LRCLK_OUT2 011 Slave from LRCLK_IN3 or LRCLK_OUT3 100 Master mode; corresponding LRCLK pin actively outputs a clock signal | LRCLK pin selection. These bits configure whether the corresponding serial port is a frame clock master or slave. When configured as a master, the corresponding LRCLK pin (LRCLK_INx for SDATA_IN pins and LRCLK_OUTx for SDATA_OUT pins) with the same number as the serial port (for example, LRCLK_OUT0 for SDATA_OUT0) actively drives out a clock signal. When configured as a slave, the serial port can receive its clock signal from any of the four corresponding LRCLK pins (LRCLK_INx pins for SDATA_INx pins or LRCLK_OUTx pins for SDATA_OUTx pins). | 0x0 | RW |
| [12:10] | BCLK_SRC | 000 Slave from BCLK_IN0 or BCLK_OUT0 001 Slave from BCLK_IN1 or BCLK_OUT1 010 Slave from BCLK_IN2 or BCLK_OUT2 011 Slave from BCLK_IN3 or BCLK_OUT3 100 Master mode; corresponding BCLK pin actively outputs a clock signal | BCLK pin selection. These bits configure whether the corresponding serial port is a bit clock master or slave. When configured as a master, the corresponding BCLK pin (BCLK_INx for SDATA_INx pins and BCLK_OUTx for SDATA_OUTx pins) with the same number as the serial port (for example, BCLK_OUT0 for SDATA_OUT0) actively drives out a clock signal. When configured as a slave, the serial port can receive its clock signal from any of the four corresponding BCLK pins (BCLK_INx pins for SDATA_INx pins or BCLK_OUTx pins for SDATA_OUTx pins). | 0x0 | RW |
| 9 | LRCLK_MODE | 0 50% duty cycle clock (square wave) 1 Pulse with a width equal to one bit clock cycle | LRCLK waveform type. The frame clock can be a 50/50 duty cycle square wave or a short pulse. | 0x0 | RW |
| 8 | LRCLK_POL | 0 Negative polarity; frame starts on falling edge of frame clock 1 Postive polarity; frame starts on rising edge of frame clock | LRCLK polarity. This bit sets the frame clock polarity on the corresponding serial port. Negative polarity means that the frame starts on the falling edge of the frame clock. This conforms to the I ² S standard audio format. | 0x0 | RW |
| 7 | BCLK_POL | 0 Negative polarity; data transitions on falling edge of bit clock 1 Positive polarity; data transitions on rising edge of bit clock | BCLK polarity. This bit sets the bit clock polarity on the corresponding serial port. Negative polarity means that the data signal transitions on the falling edge of the bit clock. This conforms to the I ² S standard audio format. | 0x0 | RW |
| [6:5] | WORD_LEN | 00 24 bits 01 16 bits 10 32 bits 11 Flexible TDM mode (configure using Register 0xF300 to Register 0xF33F, FTDM_INx, and Register 0xF380 to Register 0xF3BF, FTDM_OUTx) | Audio data-word length. These bits set the word length of the audio data channels on the corresponding serial port. For serial input ports, if the input data has more words than the length as configured by these bits, the extra data bits are ignored. For output serial ports, if the word length, as configured by these bits, is shorter than the data length coming from the data source (the DSP, ASRCs, S/PDIF receiver, PDM inputs, or serial inputs), the extra data bits are truncated and output as 0s. If Bits[6:5] (WORD_LEN) are set to 0b10 for 32-bit mode, the corresponding 32-bit input or output cells are required in SigmaStudio. | 0x0 | RW |

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|----------|--|--|-------|--------|
| [4:3] | DATA_FMT | 00 01 10 11 | MSB position. These bits set the positioning of the data in the frame on the corresponding serial port. l ² S (delay data by one BCLK cycle) Left justified (delay data by zero BCLK cycles) Right justified for 24-bit data (delay data by 8 BCLK cycles) Right justified for 16-bit data (delay data by 16 BCLK cycles) | 0x0 | RW |
| [2:0] | TDM_MODE | 000 001 010 011 100 101 | Channels per frame and BCLK cycles per channel. These bits set the number of channels per frame and the number of bit clock cycles per frame on the corresponding serial port. 2 channels, 32 bit clock cycles per channel, 64 bit clock cycles per frame 4 channels, 32 bit clock cycles per channel, 128 bit clock cycles per frame 8 channels, 32 bit clock cycles per channel, 256 bit clock cycles per frame 16 channels, 32 bit clock cycles per channel, 512 bit clock cycles per frame 4 channels, 16 bit clock cycles per channel, 64 bit clock cycles per frame 2 channels, 16 bit clock cycles per channel, 32 bit clock cycles per frame | 0x0 | RW |

Serial Port Control 1 Register

Address: 0xF201 to 0xF21D (Increments of 0x4), Reset: 0x0002, Name: SERIAL_BYTE_x_1

These eight registers configure several settings for the corresponding serial input and serial output ports. Clock generator, sample rate, and behavior during inactive channels are configured with these registers. On the input side, Register 0xF201 (SERIAL_BYTE_0_1) corresponds to SDATA_IN0; Register 0xF205 (SERIAL_BYTE_1_1) corresponds to SDATA_IN1; Register 0xF209 (SERIAL_BYTE_2_1) corresponds to SDATA_IN2; and Register 0xF20D (SERIAL_BYTE_3_1) corresponds to SDATA_IN3. On the output side, Register 0xF211 (SERIAL_BYTE_4_1) corresponds to SDATA_OUT0; Register 0xF215 (SERIAL_BYTE_5_1) corresponds to SDATA_OUT1; Register 0xF219 (SERIAL_BYTE_6_1) corresponds to SDATA_OUT2; and Register 0xF21D (SERIAL_BYTE_7_1) corresponds to SDATA_OUT3.

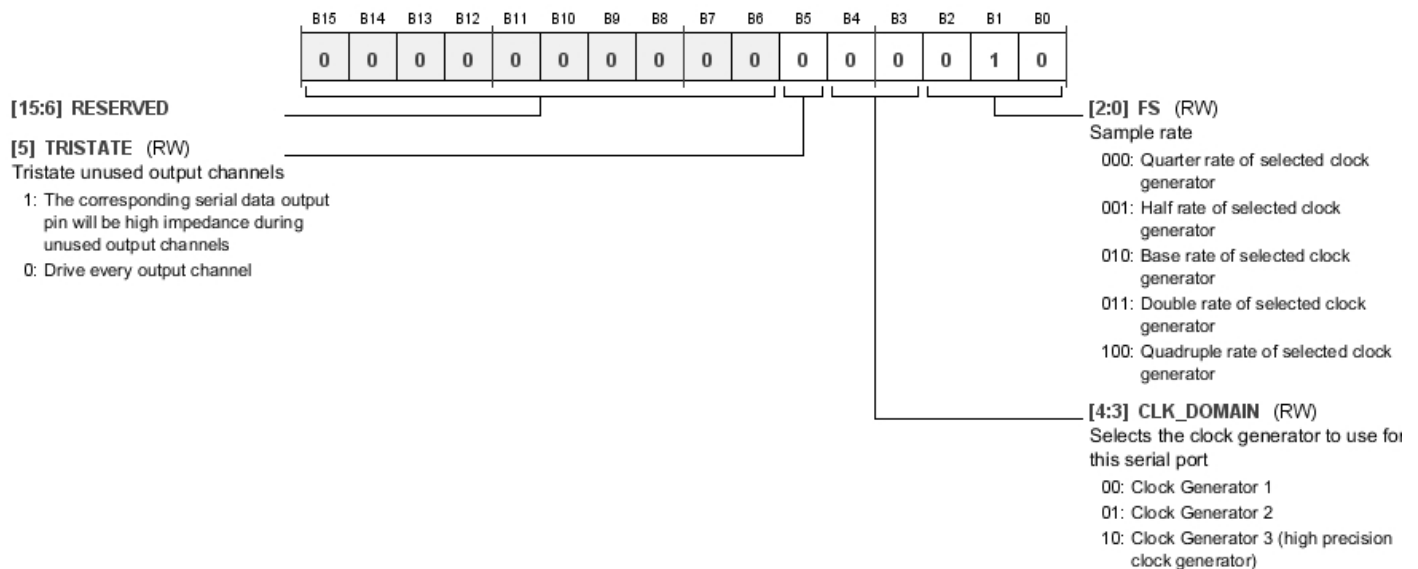


Table 82. Bit Descriptions for SERIAL_BYTE_x_1

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|----------|----------|--|-------|--------|
| [15:6] | RESERVED | | | 0x000 | RW |
| 5 | TRISTATE | 1 0 | Tristate unused output channels. This bit has no effect on serial input ports. The corresponding serial data output pin is high impedance during unused output channels Drive every output channel | 0x0 | RW |

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|------------|---------------------------------|--|-------|--------|
| [4:3] | CLK_DOMAIN | 00 01 10 | Selects the clock generator to use for the serial port. These bits select the clock generator to use for this serial port when it is configured as a clock master. This setting is valid only when Bits[15:13] (LRCLK_SRC) of the corresponding SERIAL_BYTE_x_0 register are set to 0b100 (master mode) and Bits[12:10] (BCLK_SRC) are set to 0b100 (master mode). 00 Clock Generator 1 01 Clock Generator 2 10 Clock Generator 3 (high precision clock generator) | 0x0 | RW |
| [2:0] | FS | 000 001 010 011 100 | Sample rate. These bits set the sample rate to use for the serial port when it is configured as a clock master. This setting is valid only when Bits[15:13] (LRCLK_SRC) of the corresponding SERIAL_BYTE_x_0 register are set to 0b100 (master mode) and Bits[12:10] BCLK_SRC are set to 0b100 (master mode). Bits[4:3] (CLK_DOMAIN) select which clock generator to use, and Bits[2:0] (FS) select which of the five clock generator outputs to use. 000 Quarter rate of selected clock generator 001 Half rate of selected clock generator 010 Base rate of selected clock generator 011 Double rate of selected clock generator 100 Quadruple rate of selected clock generator | 0x2 | RW |

FLEXIBLE TDM INTERFACE REGISTERS

FTDM Mapping for the Serial Inputs Register

Address: 0xF300 to 0xF33F (Increments of 0x1), Reset: 0x0000, Name: FTDM_INx

These 64 registers correspond to the 64 bytes of data that combine to form the 16 audio channels derived from the data streams being input to the SDATA_IN2 and SDATA_IN3 pins.

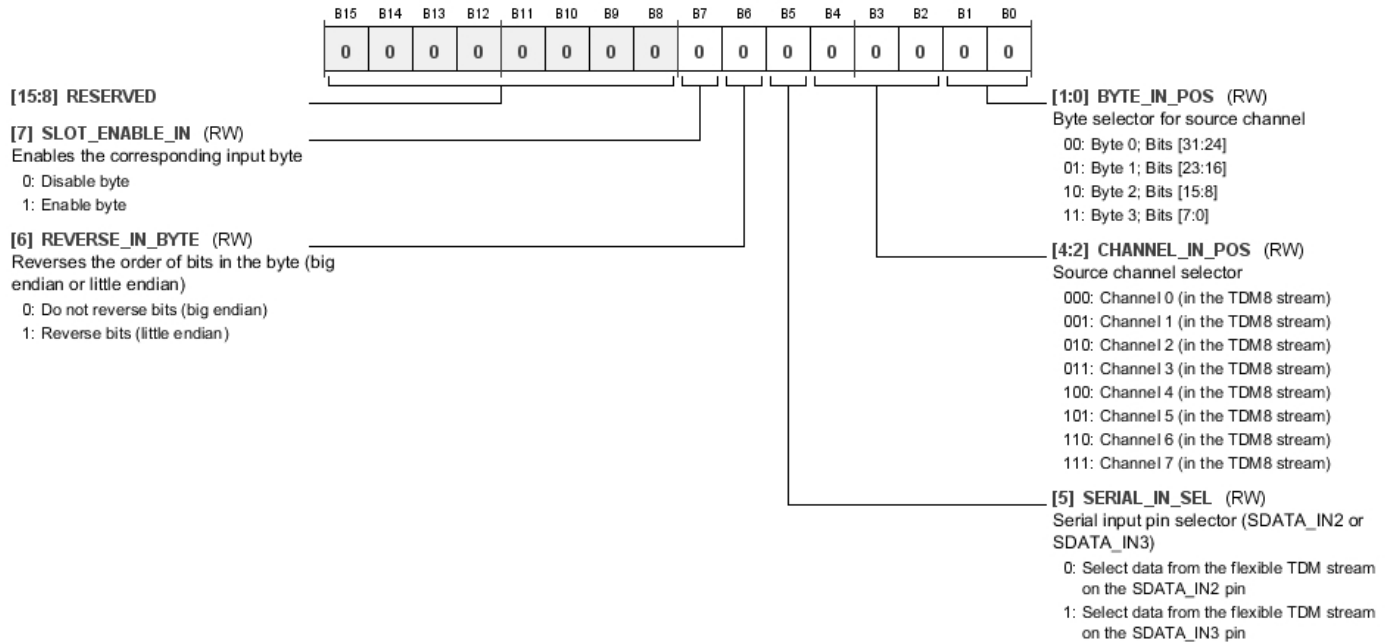


Table 83. Bit Descriptions for FTDM_INx

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-----------------|--|---|-------|--------|
| [15:8] | RESERVED | | | 0x0 | RW |
| 7 | SLOT_ENABLE_IN | 0 1 | Enables the corresponding input byte. This bit determines whether or not the slot is active. If active, valid data is input from the corresponding data slot on the selected channel of the selected input pin. If disabled, input data from the corresponding data slot on the selected channel of the selected input pin is ignored. 0 Disable byte 1 Enable byte | 0x0 | RW |
| 6 | REVERSE_IN_BYTE | 0 1 | Reverses the order of bits in the byte (big endian or little endian). This bit changes the endianness of the data bits within the byte by optionally reversing the order of the bits from MSB to LSB. 0 Do not reverse bits (big endian) 1 Reverse bits (little endian) | 0x0 | RW |
| 5 | SERIAL_IN_SEL | 0 1 | Serial input pin selector (SDATA_IN2 or SDATA_IN3). If this bit = 0b0, the slot is mapped to Audio Channel 32 to Audio Channel 39. If this bit = 0b1, the slot is mapped to Audio Channel 40 to Audio Channel 47. The exact channel assignment is determined by Bits[4:2] (CHANNEL_IN_POS). 0 Select data from the flexible TDM stream on the SDATA_IN2 pin 1 Select data from the flexible TDM stream on the SDATA_IN3 pin | 0x0 | RW |
| [4:2] | CHANNEL_IN_POS | 000 001 010 011 100 101 110 111 | Source channel selector. These bits map the slot to an audio input channel. If Bit 5 (SERIAL_IN_SEL) = 0b0, Position 0 maps to Channel 32, Position 1 maps to Channel 33, and so on. . If Bit 5 (SERIAL_IN_SEL) = 0b1, Position 0 maps to Channel 40, Position 1 maps to Channel 41, and so on. 000 Channel 0 (in the TDM8 stream) 001 Channel 1 (in the TDM8 stream) 010 Channel 2 (in the TDM8 stream) 011 Channel 3 (in the TDM8 stream) 100 Channel 4 (in the TDM8 stream) 101 Channel 5 (in the TDM8 stream) 110 Channel 6 (in the TDM8 stream) 111 Channel 7 (in the TDM8 stream) | 0x0 | RW |
| [1:0] | BYTE_IN_POS | 00 01 10 11 | Byte selector for source channel. These bits determine which byte the slot fills in the channel selected by Bit 5 (SERIAL_IN_SEL) and Bits[4:2] (CHANNEL_IN_POS). Each channel consists of four bytes that are selectable by the four options available in this bit field. 00 Byte 0; Bits[31:24] 01 Byte 1; Bits[23:16] 10 Byte 2; Bits[15:8] 11 Byte 3; Bits[7:0] | 0x0 | RW |

FTDM Mapping for the Serial Outputs Register

Address: 0xF380 to 0xF3BF (Increments of 0x1), Reset: 0x0000, Name: FTDM_OUTx

These 64 registers correspond to the 64 data slots for the flexible TDM output modes on the SDATA_OUT2 and SDATA_OUT3 pins. Slot 0 to Slot 31 are available for use on SDATA_OUT2, and Slot 32 to Slot 63 are available for use on SDATA_OUT3. Each slot can potentially hold one byte of data. Slots are mapped to corresponding audio channels in the serial ports by Bits[5:0] in these registers.

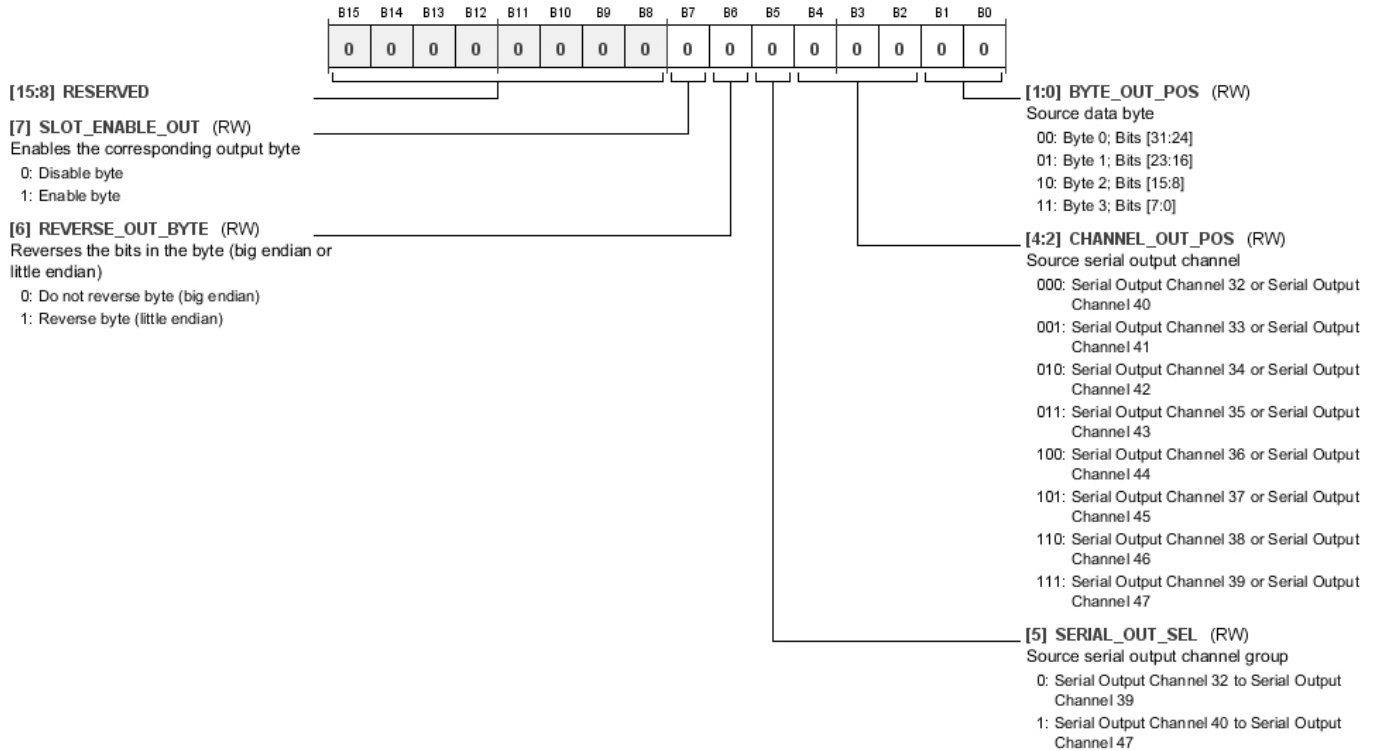


Table 84. Bit Descriptions for FTDM_OUTx

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|------------------|----------|--|-------|--------|
| [15:8] | RESERVED | | | 0x0 | RW |
| 7 | SLOT_ENABLE_OUT | 0 1 | Enables the corresponding output byte. This bit determines whether or not the slot is active. If Bit 7 (SLOT_ENABLE_OUT) = 0b0 and Bit 5 (TRISTATE) of the corresponding serial output port = 0b1, the corresponding output pin is high impedance during the period in which the corresponding flexible TDM slot is output. If Bit 7 (SLOT_ENABLE_OUT) = 0b0, and Bit 5 (TRISTATE) of the corresponding serial output port = 0b0, the corresponding output pin drives logic low during the period in which the corresponding flexible TDM slot is output. If Bit 7 (SLOT_ENABLE_OUT) = 0b1, the corresponding serial output pin outputs valid data during the period in which the corresponding flexible TDM slot is output. | 0x0 | RW |
| 6 | REVERSE_OUT_BYTE | 0 1 | Reverses the bits in the byte (big endian or little endian). This bit changes the endianness of the data bits within the corresponding flexible TDM slot by optionally reversing the order of the bits from MSB to LSB. | 0x0 | RW |
| 5 | SERIAL_OUT_SEL | 0 1 | Source serial output channel group. This bit, together with Bits[4:2] (CHANNEL_OUT_POS), selects which serial output channel is the source of data for the corresponding flexible TDM output slot. | 0x0 | RW |

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|-----------------|----------|--|-------|--------|
| [4:2] | CHANNEL_OUT_POS | | Source serial output channel. These bits, along with Bit 5 (SERIAL_OUT_SEL), select which serial output channel is the source of data for the corresponding flexible TDM output slot. If Bit 5 (SERIAL_OUT_SEL) = 0b0, Bits[4:2] (CHANNEL_OUT_POS) select serial output channels between Serial Output Channel 32 and Serial Output Channel 39. If Bit 5 (SERIAL_OUT_SEL) = 0b1, Bits[4:2] (CHANNEL_OUT_POS) selects serial output channels between Serial Output Channel 40 and Serial Output Channel 47. | 0x0 | RW |
| | | 000 | Serial Output Channel 32 or Serial Output Channel 40 | | |
| | | 001 | Serial Output Channel 33 or Serial Output Channel 41 | | |
| | | 010 | Serial Output Channel 34 or Serial Output Channel 42 | | |
| | | 011 | Serial Output Channel 35 or Serial Output Channel 43 | | |
| | | 100 | Serial Output Channel 36 or Serial Output Channel 44 | | |
| | | 101 | Serial Output Channel 37 or Serial Output Channel 45 | | |
| | | 110 | Serial Output Channel 38 or Serial Output Channel 46 | | |
| | | 111 | Serial Output Channel 39 or Serial Output Channel 47 | | |
| [1:0] | BYTE_OUT_POS | | Source data byte. These bits determine which data byte is used from the corresponding serial output channel (selected by setting Bit 5 (SERIAL_OUT_SEL) and Bits[4:2] (CHANNEL_OUT_POS)). Because there can be up to 32 bits in the data-word, four bytes are available. | 0x0 | RW |
| | | 00 | Byte 0; Bits[31:24] | | |
| | | 01 | Byte 1; Bits[23:16] | | |
| | | 10 | Byte 2; Bits[15:8] | | |
| | | 11 | Byte 3; Bits[7:0] | | |

DSP CORE CONTROL REGISTERS

Hibernate Setting Register

Address: 0xF400, Reset: 0x0000, Name: HIBERNATE

When hibernation mode is activated, the DSP core continues processing the current audio sample or block, and then enters a low power hibernation state. If Bit 0 (HIBERNATE) is set to 0b1 when the DSP core is processing audio, wait at least the duration of one sample before attempting to modify any other control registers. If Bit 0 (HIBERNATE) is set to 0b1 when the DSP core is processing audio, and block processing is used in the signal flow, wait at least the duration of one block plus the duration of one sample before attempting to modify any other control registers. During hibernation, interrupts to the core are disabled. This prevents audio from flowing into or out of the DSP core. Because DSP processing ceases when hibernation is active, there is a significant drop in the current consumption on the DVDD supply.

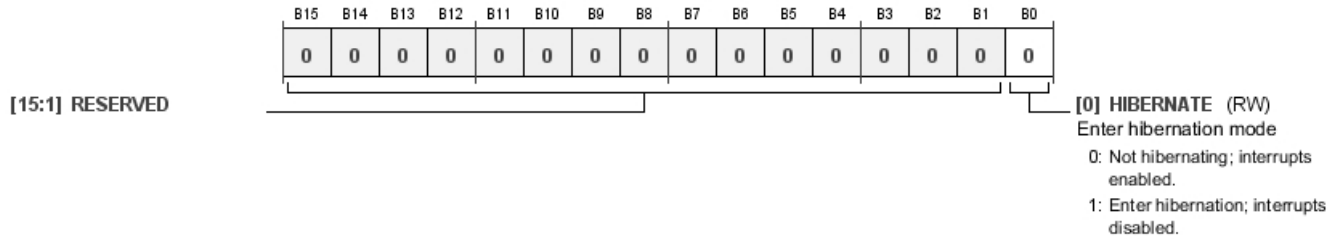


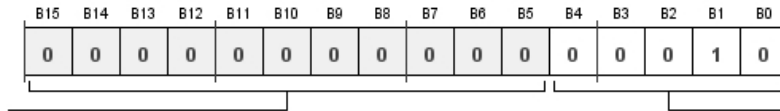
Table 85. Bit Descriptions for Hibernate

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-----------|----------|--|-------|--------|
| [15:1] | RESERVED | | | 0x0 | RW |
| 0 | HIBERNATE | 0 1 | Enter hibernation mode. This bit disables incoming interrupts and tells the DSP core to go to a low power sleep mode after the next audio sample or block has finished processing. It causes the DSP to enter hibernation mode by masking all interrupts. Not hibernating; interrupts enabled. Enter hibernation; interrupts disabled. | 0x0 | RW |

Start Pulse Selection Register

Address: 0xF401, Reset: 0x0002, Name: START_PULSE

This register selects the start pulse that marks the beginning of each audio frame in the DSP core. This effectively sets the sample rate of the audio going through the DSP. This start pulse can originate from either an internally generated pulse (from Clock Generator 1 or Clock Generator 2) or from an external clock that is received on one of the LRCLK pins of one of the serial ports. Any audio input or output from the DSP core that is asynchronous to this DSP start pulse rate must go through an ASRC. If asynchronous audio signals (that is, signals that are not synchronized to whatever start pulse is selected) are input to the DSP without first going through an ASRC, samples are skipped or doubled, leading to distortion and audible artifacts in the audio signal.



[15:5] RESERVED

[4:0] START_PULSE (RW)

Start pulse selection

- 00000: Base sample rate $\div 4$ (12 kHz for 48 kHz base sample rate) (1/4 output of Clock Generator 1)
- 00001: Base sample rate $\div 2$ (24 kHz for 48 kHz base sample rate) (1/2 output of Clock Generator 1)
- 00010: Base sample rate (48 kHz for 48 kHz base sample rate) ($\times 1$ output of Clock Generator 1)
- 00011: Base sample rate $\times 2$ (96 kHz for 48 kHz base sample rate) ($\times 2$ output of Clock Generator 1)
- 00100: Base sample rate $\times 4$ (192 kHz for 48 kHz base sample rate) ($\times 4$ output of Clock Generator 1)
- 00101: Base sample rate $\div 6$ (8 kHz for 48 kHz base sample rate) (1/4 output of Clock Generator 2)
- 00110: Base sample rate $\div 3$ (16 kHz for 48 kHz base sample rate) (1/2 output of Clock Generator 2)
- 00111: $2 \times$ base sample rate $\div 3$ (32 kHz for 48 kHz base sample rate) ($\times 1$ output of Clock Generator 2)
- 01000: Serial Input Port 0 sample rate (Register 0xF201 (SERIAL_BYTE_0_1), Bits[4:0])
- 01001: Serial Input Port 1 sample rate (Register 0xF205 (SERIAL_BYTE_1_1), Bits[4:0])
- 01010: Serial Input Port 2 sample rate (Register 0xF209 (SERIAL_BYTE_2_1), Bits[4:0])
- 01011: Serial Input Port 3 sample rate (Register 0xF20D (SERIAL_BYTE_3_1), Bits[4:0])
- 01100: Serial Output Port 0 sample rate (Register 0xF211 (SERIAL_BYTE_4_1), Bits[4:0])
- 01101: Serial Output Port 1 sample rate (Register 0xF215 (SERIAL_BYTE_5_1), Bits[4:0])
- 01110: Serial Output Port 2 sample rate (Register 0xF219 (SERIAL_BYTE_6_1), Bits[4:0])
- 01111: Serial Output Port 3 sample rate (Register 0xF21D (SERIAL_BYTE_7_1), Bits[4:0])
- 10000: S/PDIF receiver sample rate (derived from the S/PDIF input stream)

Table 86. Bit Descriptions for START_PULSE

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-------------|---|-------------|-------|--------|
| [15:5] | RESERVED | | | 0x0 | RW |
| [4:0] | START_PULSE | 00000 Base sample rate ÷ 4 (12 kHz for 48 kHz base sample rate) (1/4 output of Clock Generator 1) 00001 Base sample rate ÷ 2 (24 kHz for 48 kHz base sample rate) (1/2 output of Clock Generator 1) 00010 Base sample rate (48 kHz for 48 kHz base sample rate) (×1 output of Clock Generator 1) 00011 Base sample rate × 2 (96 kHz for 48 kHz base sample rate) (×2 output of Clock Generator 1) 00100 Base sample rate × 4 (192 kHz for 48 kHz base sample rate) (×4 output of Clock Generator 1) 00101 Base sample rate ÷ 6 (8 kHz for 48 kHz base sample rate) (1/4 output of Clock Generator 2) 00110 Base sample rate ÷ 3 (16 kHz for 48 kHz base sample rate) (1/2 output of Clock Generator 2) 00111 2× base sample rate ÷ 3 (32 kHz for 48 kHz base sample rate) (×1 output of Clock Generator 2) 01000 Serial Input Port 0 sample rate (Register 0xF201 (SERIAL_BYTE_0_1), Bits[4:0]) 01001 Serial Input Port 1 sample rate (Register 0xF205 (SERIAL_BYTE_1_1), Bits[4:0]) 01010 Serial Input Port 2 sample rate (Register 0xF209 (SERIAL_BYTE_2_1), Bits[4:0]) 01011 Serial Input Port 3 sample rate (Register 0xF20D (SERIAL_BYTE_3_1), Bits[4:0]) 01100 Serial Output Port 0 sample rate (Register 0xF211 (SERIAL_BYTE_4_1), Bits[4:0]) 01101 Serial Output Port 1 sample rate (Register 0xF215 (SERIAL_BYTE_5_1), Bits[4:0]) 01110 Serial Output Port 2 sample rate (Register 0xF219 (SERIAL_BYTE_6_1), Bits[4:0]) 01111 Serial Output Port 3 sample rate (Register 0xF21D (SERIAL_BYTE_7_1), Bits[4:0]) 10000 S/PDIF receiver sample rate (derived from the S/PDIF input stream) | 0x02 | RW | |

Instruction to Start the Core Register

Address: 0xF402, Reset: 0x0000, Name: START_CORE

Enables the DSP core and initiates the program counter, which then begins incrementing through the program memory and executing instruction codes. This register is edge triggered, meaning that a rising edge on Bit 0 (START_CORE), that is, a transition from 0b0 to 0b1, initiates the program counter. A falling edge on Bit 0 (START_CORE), that is, a transition from 0b1 to 0b0, has no effect. To stop the DSP core, use Register 0xF400 (HIBERNATE), Bit 0 (HIBERNATE).

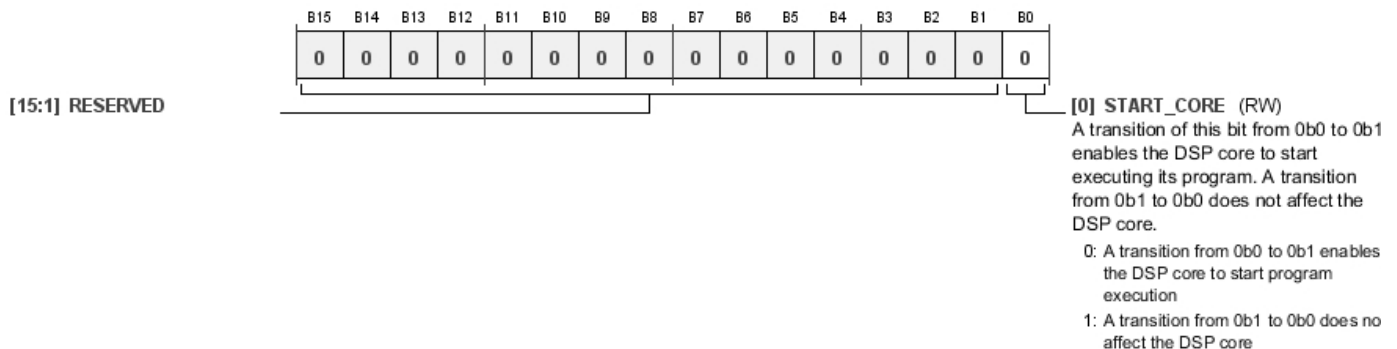


Table 87. Bit Descriptions for START_CORE

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|----------|----------|-------------|-------|--------|
| [15:1] | RESERVED | | | 0x0 | RW |

| Bits | Bit Name | Settings | Description | Reset | Access |
|------|------------|----------|--|-------|--------|
| 0 | START_CORE | | A transition of this bit from 0b0 to 0b1 enables the DSP core to start executing its program. A transition from 0b1 to 0b0 does not affect the DSP core. | 0x0 | RW |
| | | 0 | A transition from 0b0 to 0b1 enables the DSP core to start program execution | | |
| | | 1 | A transition from 0b1 to 0b0 does not affect the DSP core | | |

Instruction to Stop the Core Register

Address: 0xF403, Reset: 0x0000, Name: KILL_CORE

Bit 0 (KILL_CORE) halts the DSP core immediately, even when it is in an undefined state. Because halting the DSP core immediately can lead to memory corruption, and it must be used only in debugging situations. This register is edge triggered, meaning that a rising edge on Bit 0 (KILL_CORE), that is, a transition from 0b0 to 0b1, halts the core. A falling edge on Bit 0 (KILL_CORE), that is, a transition from 0b1 to 0b0, has no effect. To stop the DSP core after the next audio frame or block, use Register 0xF400 (HIBERNATE), Bit 0 (HIBERNATE).

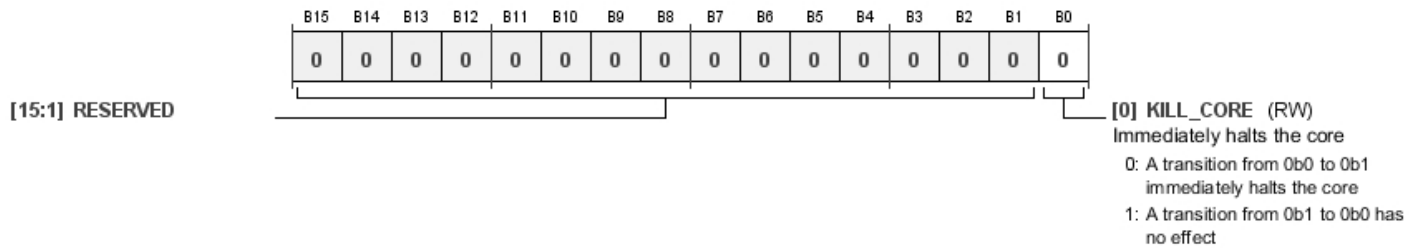


Table 88. Bit Descriptions for KILL_CORE

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-----------|----------|--|-------|--------|
| [15:1] | RESERVED | | | 0x0 | RW |
| 0 | KILL_CORE | | Immediately halts the core. When this bit transitions from 0b0 to 0b1, the core immediately halts. This can bring about undesired effects and, therefore, should be used only in debugging. To stop the core while it is running, use Register 0xF400 (HIBERNATE) to halt the core in a controlled manner. | 0x0 | RW |
| | | 0 | A transition from 0b0 to 0b1 immediately halts the core | | |
| | | 1 | A transition from 0b1 to 0b0 has no effect | | |

Start Address of the Program Register

Address: 0xF404, Reset: 0x0000, Name: START_ADDRESS

This register sets the program address where the program counter begins after the DSP core is enabled, using Register 0xF402, Bit 0 (START_CORE). The SigmaStudio compiler automatically sets the program start address; therefore, the user is not required to manually modify the value of this register.



Table 89. Bit Descriptions for START_ADDRESS

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|---------------|----------|------------------------|--------|--------|
| [15:0] | START_ADDRESS | | Program start address. | 0x0000 | RW |

Core Status Register

Address: 0xF405, Reset: 0x0000, Name: CORE_STATUS

This read only register allows the user to check the status of the DSP core. To manually modify the core status, use Register 0xF400 (HIBERNATE), Register 0xF402 (START_CORE), and Register 0xF403 (KILL_CORE).

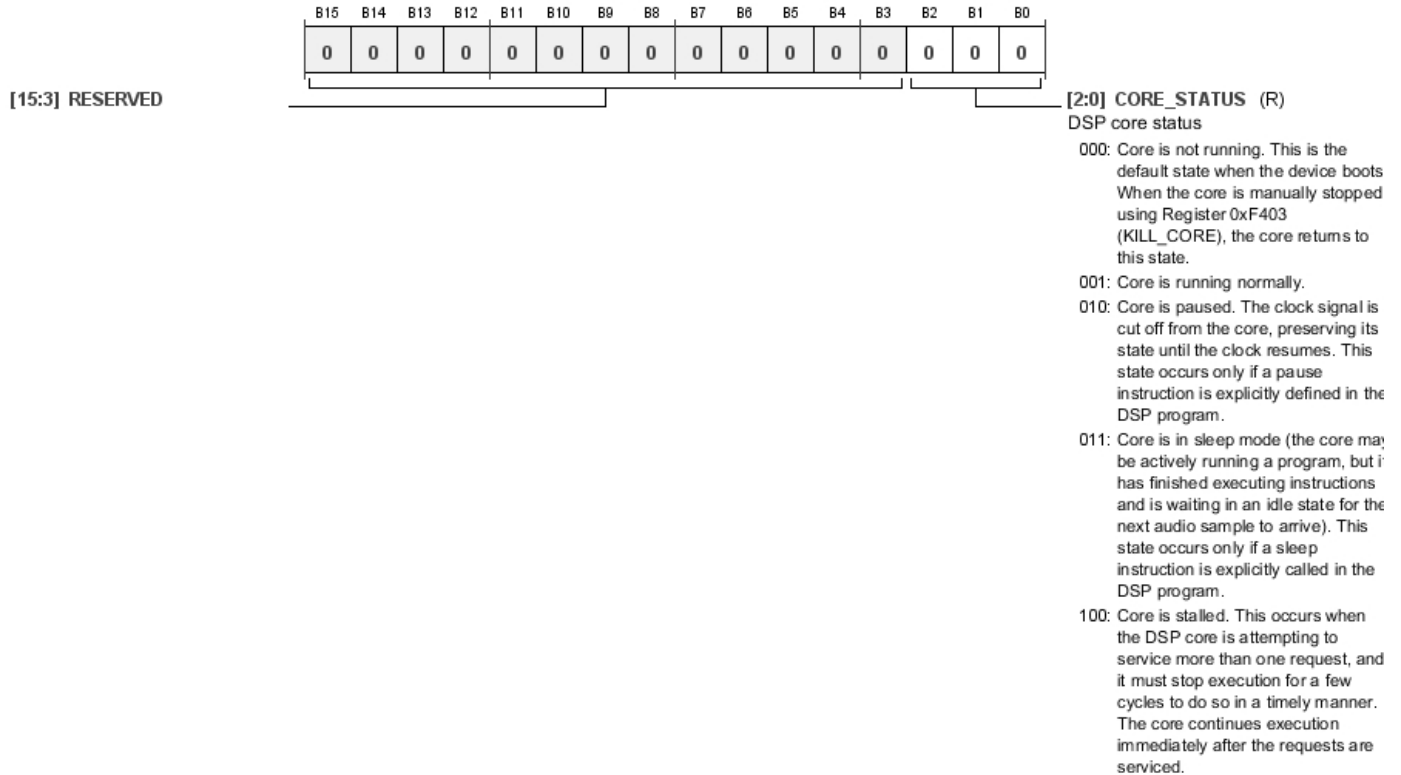


Table 90. Bit Descriptions for CORE_STATUS

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-------------|----------|--|-------|--------|
| [15:3] | RESERVED | | | 0x0 | RW |
| [2:0] | CORE_STATUS | | DSP core status. These bits display the status of the DSP core at the moment the value is read. | 0x0 | RW |
| | | 000 | Core is not running. This is the default state when the device boots. When the core is manually stopped using Register 0xF403 (KILL_CORE), the core returns to this state. | | |
| | | 001 | Core is running normally. | | |
| | | 010 | Core is paused. The clock signal is cut off from the core, preserving its state until the clock resumes. This state occurs only if a pause instruction is explicitly defined in the DSP program. | | |
| | | 011 | Core is in sleep mode (the core may be actively running a program, but it has finished executing instructions and is waiting in an idle state for the next audio sample to arrive). This state occurs only if a sleep instruction is explicitly called in the DSP program. | | |
| | | 100 | Core is stalled. This occurs when the DSP core is attempting to service more than one request, and it must stop execution for a few cycles to do so in a timely manner. The core continues execution immediately after the requests are serviced. | | |

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.

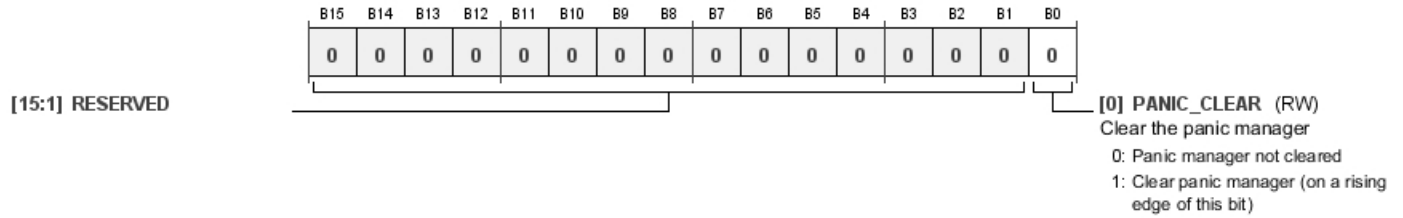


Table 91. Bit Descriptions for PANIC_CLEAR

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

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, should be subject to error reporting.

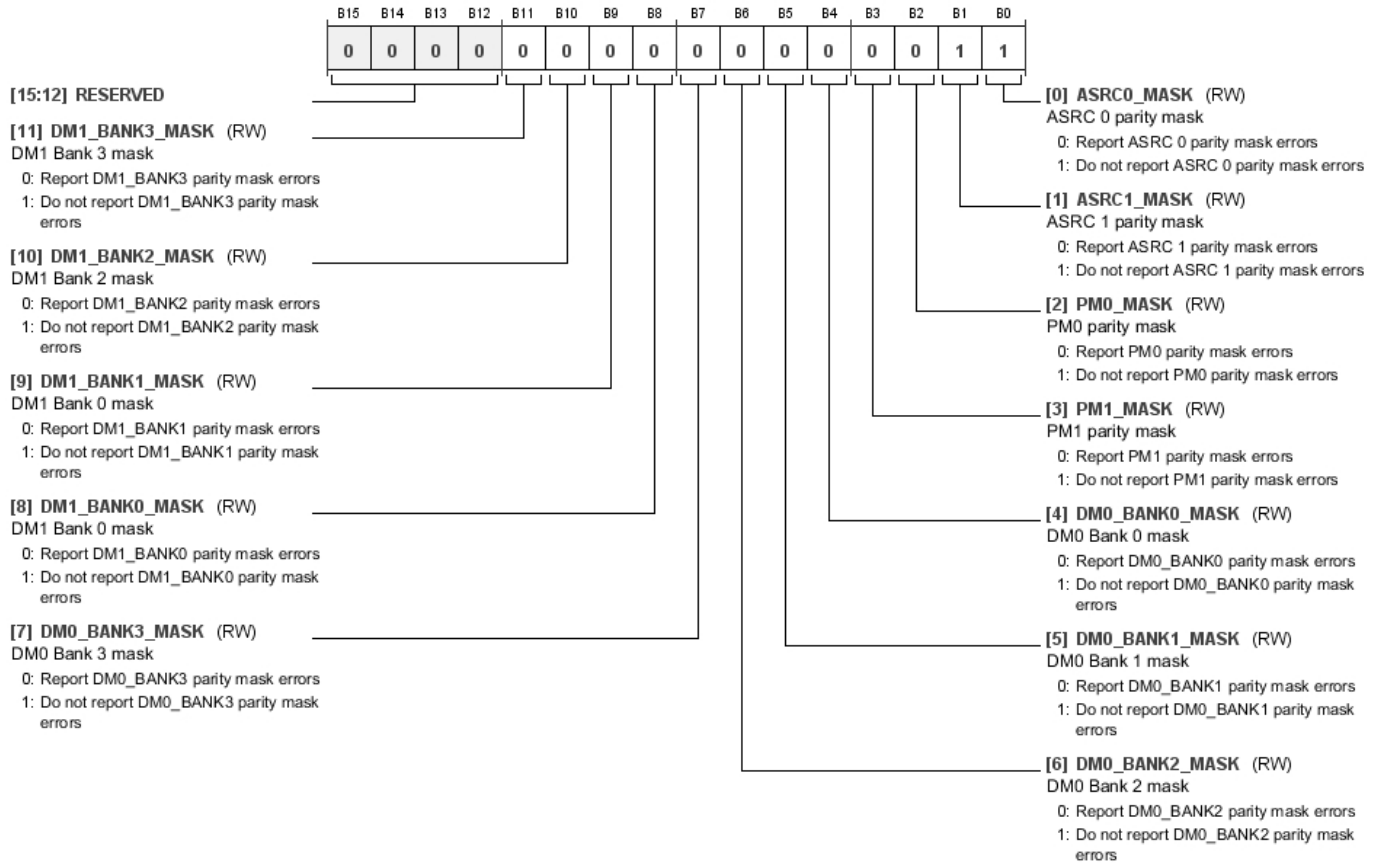


Table 92. Bit Descriptions for PANIC_PARITY_MASK

| Bits | Bit Name | Settings | Description | Reset | Access |
|---------|----------------|----------|---|-------|--------|
| [15:12] | RESERVED | | | 0x0 | RW |
| 11 | DM1_BANK3_MASK | 0 1 | DM1 Bank 3 mask. Report DM1_BANK3 parity mask errors Do not report DM1_BANK3 parity mask errors | 0x0 | RW |
| 10 | DM1_BANK2_MASK | 0 1 | DM1 Bank 2 mask. Report DM1_BANK2 parity mask errors Do not report DM1_BANK2 parity mask errors | 0x0 | RW |
| 9 | DM1_BANK1_MASK | 0 1 | DM1 Bank 1 mask. Report DM1_BANK1 parity mask errors Do not report DM1_BANK1 parity mask errors | 0x0 | RW |
| 8 | DM1_BANK0_MASK | 0 1 | DM1 Bank 0 mask. Report DM1_BANK0 parity mask errors Do not report DM1_BANK0 parity mask errors | 0x0 | RW |
| 7 | DM0_BANK3_MASK | 0 1 | DM0 Bank 3 mask. Report DM0_BANK3 parity mask errors Do not report DM0_BANK3 parity mask errors | 0x0 | RW |

| Bits | Bit Name | Settings | Description | Reset | Access |
|------|----------------|----------|---|-------|--------|
| 6 | DM0_BANK2_MASK | 0 1 | DM0 Bank 2 mask. Report DM0_BANK2 parity mask errors Do not report DM0_BANK2 parity mask errors | 0x0 | RW |
| 5 | DM0_BANK1_MASK | 0 1 | DM0 Bank 1 mask. Report DM0_BANK1 parity mask errors Do not report DM0_BANK1 parity mask errors | 0x0 | RW |
| 4 | DM0_BANK0_MASK | 0 1 | DM0 Bank 0 mask. Report DM0_BANK0 parity mask errors Do not report DM0_BANK0 parity mask errors | 0x0 | RW |
| 3 | PM1_MASK | 0 1 | PM1 parity mask. Report PM1 parity mask errors Do not report PM1 parity mask errors | 0x0 | RW |
| 2 | PM0_MASK | 0 1 | PM0 parity mask. Report PM0 parity mask errors Do not report PM0 parity mask errors | 0x0 | RW |
| 1 | ASRC1_MASK | 0 1 | ASRC 1 parity mask. Report ASRC 1 parity mask errors Do not report ASRC 1 parity mask errors | 0x1 | RW |
| 0 | ASRC0_MASK | 0 1 | ASRC 0 parity mask. Report ASRC 0 parity mask errors Do not report ASRC 0 parity mask errors | 0x1 | RW |

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.

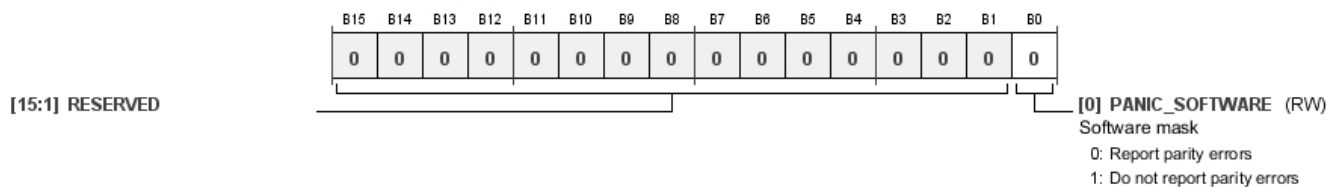


Table 93. Bit Descriptions for PANIC_SOFTWARE_MASK

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

Panic Mask 1 Register

Address: 0xF424, Reset: 0x0000, Name: PANIC_WD_MASK

The panic manager checks and reports watchdog errors. Register 0xF424 (PANIC_WD_MASK) allows the user to configure whether watchdog errors are reported to the panic manager or ignored.

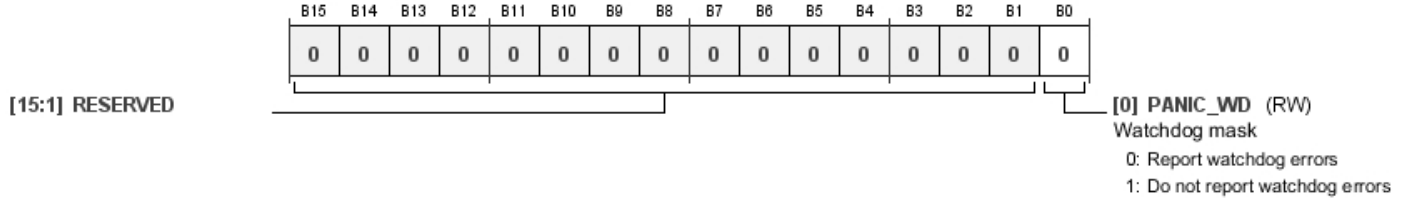


Table 94. Bit Descriptions for PANIC_WD_MASK

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|----------|----------|---|-------|--------|
| [15:1] | RESERVED | | | 0x0 | RW |
| 0 | PANIC_WD | 0 1 | Watchdog mask. 0 Report watchdog errors 1 Do not report watchdog errors | 0x0 | RW |

Panic Mask 2 Register

Address: 0xF425, Reset: 0x0000, Name: PANIC_STACK_MASK

The panic manager checks and reports stack errors. Register 0xF425 (PANIC_STACK_MASK) allows the user to configure whether stack errors are reported to the panic manager or ignored.

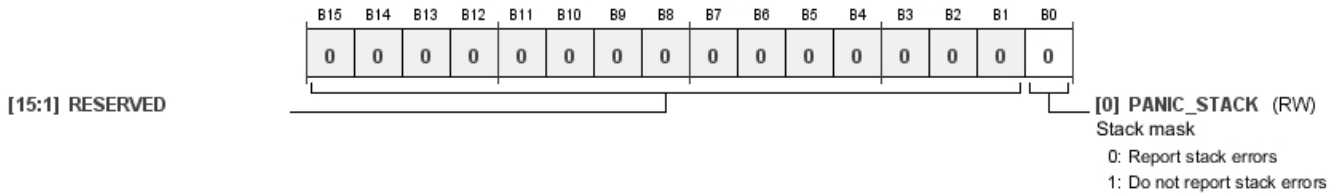


Table 95. Bit Descriptions for PANIC_STACK_MASK

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-------------|----------|--|-------|--------|
| [15:1] | RESERVED | | | 0x0 | RW |
| 0 | PANIC_STACK | 0 1 | Stack mask. 0 Report stack errors 1 Do not report stack errors | 0x0 | RW |

Panic Mask 3 Register

Address: 0xF426, Reset: 0x0000, Name: PANIC_LOOP_MASK

The panic manager checks and reports software errors related to looping code sections. Register 0xF426 (PANIC_LOOP_MASK) allows the user to configure whether loop errors are reported to the panic manager or ignored.

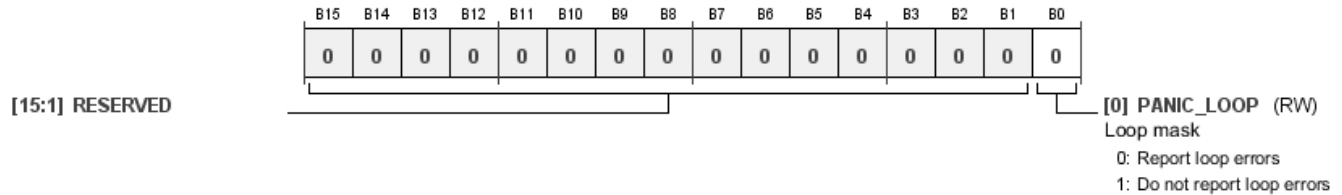


Table 96. Bit Descriptions for PANIC_LOOP_MASK

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|------------|----------|---|-------|--------|
| [15:1] | RESERVED | | | 0x0 | RW |
| 0 | PANIC_LOOP | 0 1 | Loop mask. 0 Report loop errors 1 Do not report loop errors | 0x0 | RW |

Panic Flag Register

Address: 0xF427, Reset: 0x0000, Name: PANIC_FLAG

This register acts as the master error flag for the panic manager. If any error is encountered in any functional block whose panic manager mask is disabled, this register logs that an error has occurred. Individual functional block masks are configured using Register 0xF422 (PANIC_PARITY_MASK), Register 0xF423 (PANIC_SOFTWARE_MASK), Register 0xF424 (PANIC_WD_MASK), Register 0xF425 (PANIC_STACK_MASK), and Register 0xF426 (PANIC_LOOP_MASK).

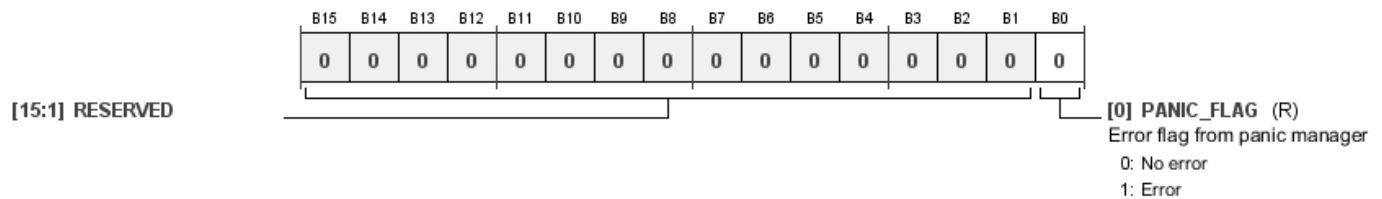


Table 97. Bit Descriptions for PANIC_FLAG

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|------------|----------|---|-------|--------|
| [15:1] | RESERVED | | | 0x0 | RW |
| 0 | PANIC_FLAG | 0 1 | Error flag from panic manager. This error flag bit is sticky. When an error is reported, this bit goes high, and it stays high until the user resets it using Register 0xF421 (PANIC_CLEAR). 0 No error 1 Error | 0x0 | R |

Panic Code Register

Address: 0xF428, Reset: 0x0000, Name: PANIC_CODE

When Register 0xF427 (PANIC_FLAG) indicates that an error has occurred, this register provides details revealing which subsystem is reporting an error. If several errors occur, this register reports only the first error that occurs. Subsequent errors are ignored until the register is cleared by toggling Register 0xF421 (PANIC_CLEAR).

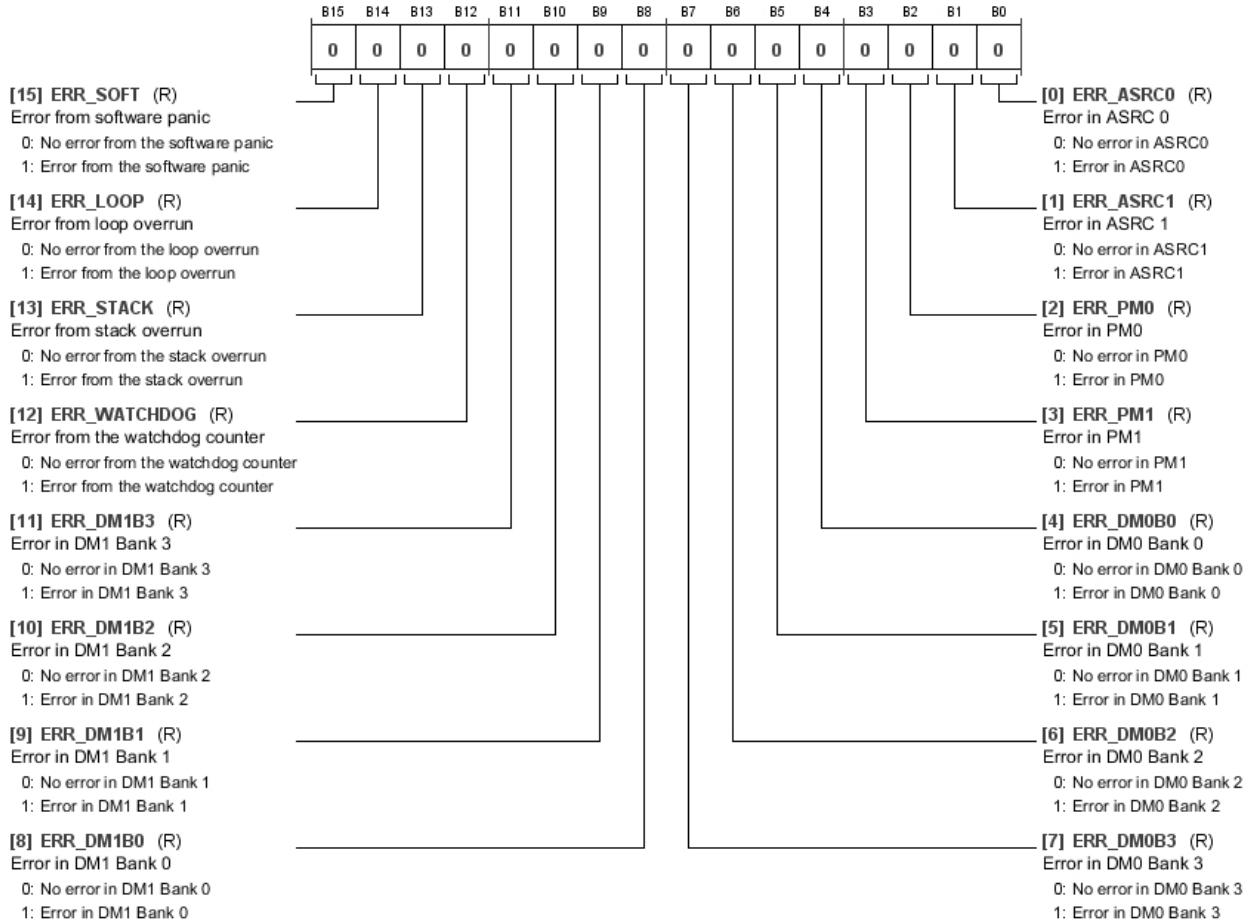


Table 98. Bit Descriptions for PANIC_CODE

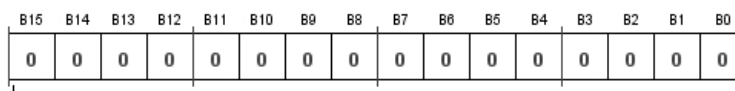
| Bits | Bit Name | Settings | Description | Reset | Access |
|------|--------------|----------|---|-------|--------|
| 15 | ERR_SOFT | 0 1 | Error from software panic. No error from the software panic Error from the software panic | 0x0 | R |
| 14 | ERR_LOOP | 0 1 | Error from loop overrun. No error from the loop overrun Error from the loop overrun | 0x0 | R |
| 13 | ERR_STACK | 0 1 | Error from stack overrun. No error from the stack overrun Error from the stack overrun | 0x0 | R |
| 12 | ERR_WATCHDOG | 0 1 | Error from the watchdog counter. No error from the watchdog counter Error from the watchdog counter | 0x0 | R |
| 11 | ERR_DM1B3 | 0 1 | Error in DM1 Bank 3. No error in DM1 Bank 3 Error in DM1 Bank 3 | 0x0 | R |

| Bits | Bit Name | Settings | Description | Reset | Access |
|------|-----------|----------|---|-------|--------|
| 10 | ERR_DM1B2 | 0 1 | Error in DM1 Bank 2. No error in DM1 Bank 2 Error in DM1 Bank 2 | 0x0 | R |
| 9 | ERR_DM1B1 | 0 1 | Error in DM1 Bank 1. No error in DM1 Bank 1 Error in DM1 Bank 1 | 0x0 | R |
| 8 | ERR_DM1B0 | 0 1 | Error in DM1 Bank 0. No error in DM1 Bank 0 Error in DM1 Bank 0 | 0x0 | R |
| 7 | ERR_DM0B3 | 0 1 | Error in DM0 Bank 3. No error in DM0 Bank 3 Error in DM0 Bank 3 | 0x0 | R |
| 6 | ERR_DM0B2 | 0 1 | Error in DM0 Bank 2. No error in DM0 Bank 2 Error in DM0 Bank 2 | 0x0 | R |
| 5 | ERR_DM0B1 | 0 1 | Error in DM0 Bank 1. No error in DM0 Bank 1 Error in DM0 Bank 1 | 0x0 | R |
| 4 | ERR_DM0B0 | 0 1 | Error in DM0 Bank 0. No error in DM0 Bank 0 Error in DM0 Bank 0 | 0x0 | R |
| 3 | ERR_PM1 | 0 1 | Error in PM1. No error in PM1 Error in PM1 | 0x0 | R |
| 2 | ERR_PM0 | 0 1 | Error in PM0. No error in PM0 Error in PM0 | 0x0 | R |
| 1 | ERR_ASRC1 | 0 1 | Error in ASRC 1. No error in ASRC 1 Error in ASRC 1 | 0x0 | R |
| 0 | ERR_ASRC0 | 0 1 | Error in ASRC 0. No error in ASRC 0 Error in ASRC 0 | 0x0 | R |

Execute Stage Error Program Count Register

Address: 0xF432, Reset: 0x0000, Name: EXECUTE_COUNT

When a software error occurs, this register logs the program instruction count at the time when the error occurred for software debugging purposes.



[15:0] EXECUTE_COUNT (RW)
Program count in the execute stage when the error occurred

Table 99. Bit Descriptions for EXECUTE_COUNT

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|---------------|----------|---|--------|--------|
| [15:0] | EXECUTE_COUNT | | Program count in the execute stage when the error occurred. | 0x0000 | RW |

Watchdog Maximum Count Register

Address: 0xF443, Reset: 0x0000, Name: WATCHDOG_MAXCOUNT

This register is designed to start counting at a specified number and decrement by 1 for each clock cycle of the system clock in the core. The counter is reset to the maximum value each time the program counter jumps to the beginning of the program to begin processing another audio frame (this is implemented in the DSP program code generated by SigmaStudio). If the counter reaches 0, a watchdog error flag is raised in the panic manager. The watchdog is typically set to begin counting from a number slightly larger than the maximum number of instructions expected to execute in the program, such that an error occurs if the program does not finish in time for the next incoming sample.

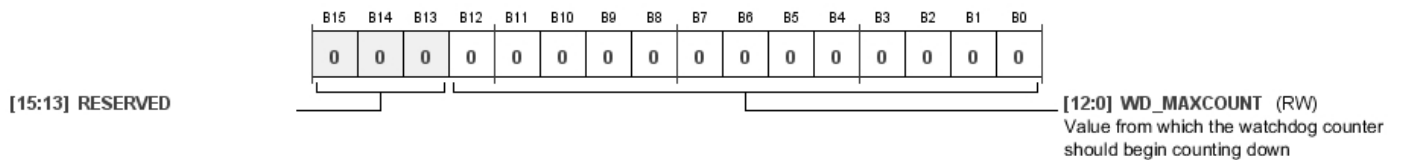


Table 100. Bit Descriptions for WATCHDOG_MAXCOUNT

| Bits | Bit Name | Settings | Description | Reset | Access |
|---------|-------------|----------|---|--------|--------|
| [15:13] | RESERVED | | | 0x0 | RW |
| [12:0] | WD_MAXCOUNT | | Value from which the watchdog counter should begin counting down. | 0x0000 | RW |

Watchdog Prescale Register

Address: 0xF444, Reset: 0x0000, Name: WATCHDOG_PRESCALE

The watchdog prescaler is a number that is multiplied by the setting in Register 0xF443 (WATCHDOG_MAXCOUNT) to achieve very large counts for the watchdog, if necessary. Using the largest prescale factor of 128 × 1024 and the largest watchdog maximum count of 64 × 1024, a very large watchdog counter, on the order of 8.5 billion clock cycles, can be achieved.

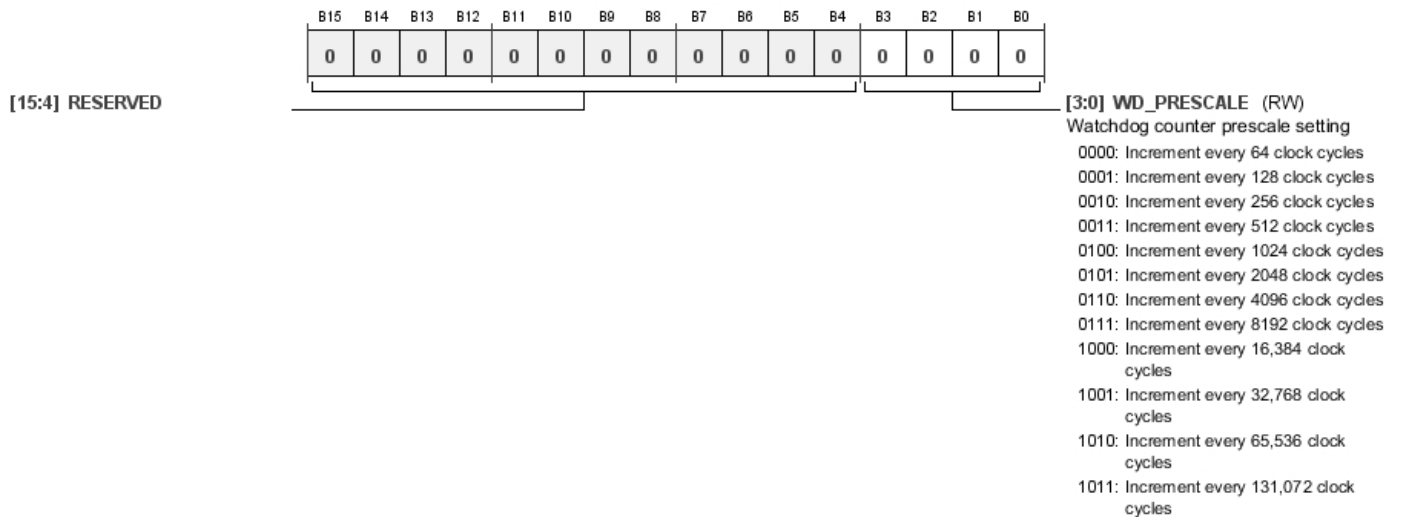


Table 101. Bit Descriptions for WATCHDOG_PRESCALE

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-------------|--|--|-------|--------|
| [15:4] | RESERVED | | | 0x0 | RW |
| [3:0] | WD_PRESCALE | 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 | Watchdog counter prescale setting. Increment every 64 clock cycles Increment every 128 clock cycles Increment every 256 clock cycles Increment every 512 clock cycles Increment every 1024 clock cycles Increment every 2048 clock cycles Increment every 4096 clock cycles Increment every 8192 clock cycles Increment every 16,384 clock cycles Increment every 32,768 clock cycles Increment every 65,536 clock cycles Increment every 131,072 clock cycles | 0x0 | RW |

DSP PROGRAM EXECUTION REGISTERS

Enable Block Interrupts Register

Address: 0xF450, Reset: 0x0000, Name: BLOCKINT_EN

This register enables block interrupts, which are necessary when frequency domain processing is required in the audio processing program. If block processing algorithms are used in SigmaStudio, SigmaStudio automatically sets this register accordingly. The user does not need to manually change the value of this register after SigmaStudio has configured it.

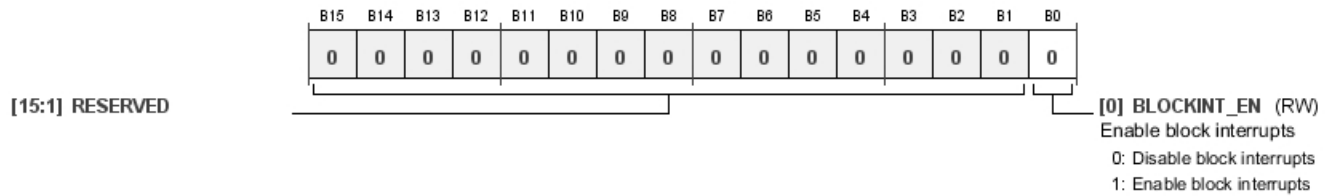


Table 102. Bit Descriptions for BLOCKINT_EN

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-------------|----------|---|-------|--------|
| [15:1] | RESERVED | | | 0x0 | RW |
| 0 | BLOCKINT_EN | 0 1 | Enable block interrupts. Disable block interrupts Enable block interrupts | 0x0 | RW |

Value for the Block Interrupt Counter Register

Address: 0xF451, Reset: 0x0000, Name: BLOCKINT_VALUE

This 16-bit register controls the duration in audio frames of a block. A counter increments each time a new frame start pulse is received by the DSP core. When the counter reaches the value determined by this register, a block interrupt is generated and the counter is reset. If block processing algorithms are used in SigmaStudio, SigmaStudio automatically sets this register accordingly. The user does not need to manually change the value of this register after SigmaStudio has configured it.



Table 103. Bit Descriptions for BLOCKINT_VALUE

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|----------------|----------|--|--------|--------|
| [15:0] | BLOCKINT_VALUE | | Value for the block interrupt counter. | 0x0000 | RW |

Program Counter, Bits[23:16] Register

Address: 0xF460, Reset: 0x0000, Name: PROG_CNTR0

This register, in combination with Register 0xF461 (PROG_CNTR1), stores the current value of the program counter.

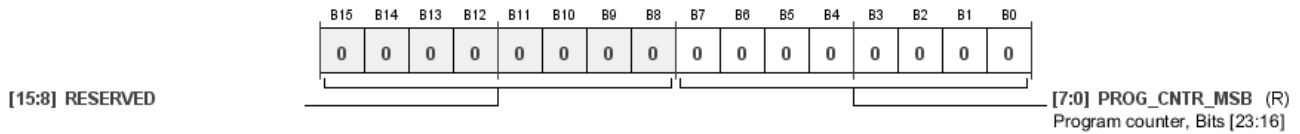


Table 104. Bit Descriptions for PROG_CNTR0

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|---------------|----------|-------------------------------|-------|--------|
| [15:8] | RESERVED | | | 0x0 | RW |
| [7:0] | PROG_CNTR_MSB | | Program counter, Bits[23:16]. | 0x00 | R |

Program Counter, Bits[15:0] Register

Address: 0xF461, Reset: 0x0000, Name: PROG_CNTR1

This register, in combination with Register 0xF460 (PROG_CNTR0), stores the current value of the program counter.

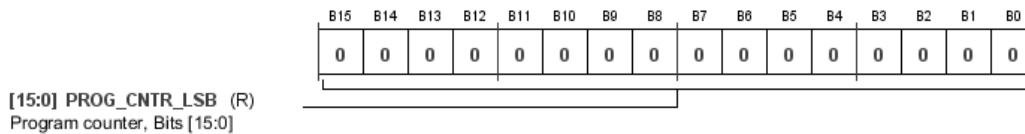


Table 105. Bit Descriptions for PROG_CNTR1

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|---------------|----------|------------------------------|--------|--------|
| [15:0] | PROG_CNTR_LSB | | Program counter, Bits[15:0]. | 0x0000 | R |

Program Counter Clear Register

Address: 0xF462, Reset: 0x0000, Name: PROG_CNTR_CLEAR

Enabling and disabling Bit 0 (PROG_CNTR_CLEAR) resets Register 0xF465 (PROG_CNTR_MAXLENGTH0) and Register 0xF466 (PROG_CNTR_MAXLENGTH1).

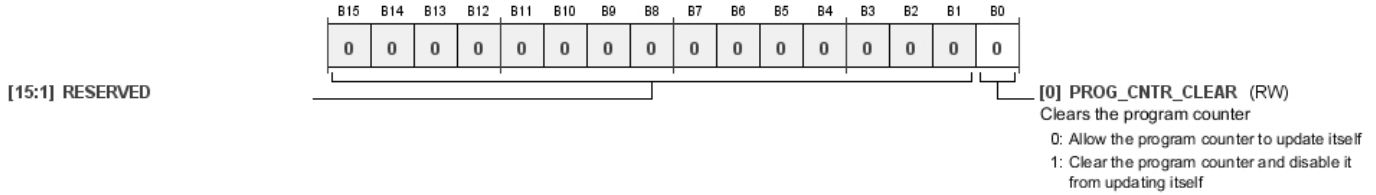


Table 106. Bit Descriptions for PROG_CNTR_CLEAR

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-----------------|----------|--|-------|--------|
| [15:1] | RESERVED | | | 0x0 | RW |
| 0 | PROG_CNTR_CLEAR | 0 1 | Clears the program counter. Allow the program counter to update itself Clear the program counter and disable it from updating itself | 0x0 | RW |

Program Counter Length, Bits[23:16] Register

Address: 0xF463, Reset: 0x0000, Name: PROG_CNTR_LENGTH0

This register, in combination with Register 0xF464 (PROG_CNTR_LENGTH1), keeps track of the peak value reached by the program counter during the last audio frame or block. It can be cleared using Register 0xF462 (PROG_CNTR_CLEAR).

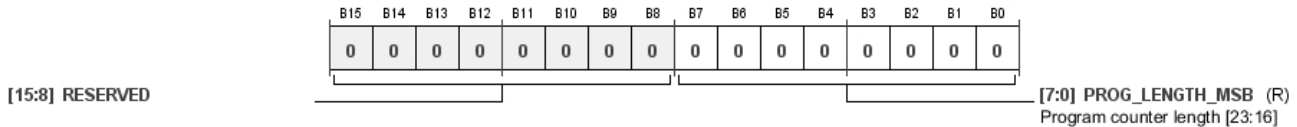


Table 107. Bit Descriptions for PROG_CNTR_LENGTH0

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-----------------|----------|--------------------------------------|-------|--------|
| [15:8] | RESERVED | | | 0x0 | RW |
| [7:0] | PROG_LENGTH_MSB | | Program counter length, Bits[23:16]. | 0x00 | R |

Program Counter Length, Bits[15:0] Register

Address: 0xF464, Reset: 0x0000, Name: PROG_CNTR_LENGTH1

This register, in combination with Register 0xF463 (PROG_CNTR_LENGTH0), keeps track of the peak value reached by the program counter during the last audio frame or block. It can be cleared using Register 0xF462 (PROG_CNTR_CLEAR).

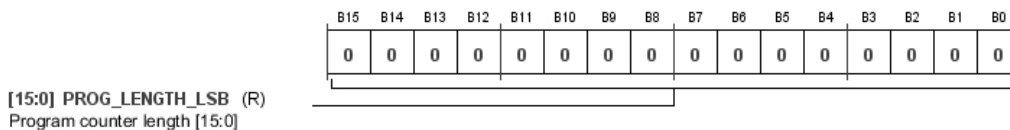


Table 108. Bit Descriptions for PROG_CNTR_LENGTH1

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-----------------|----------|-------------------------------------|--------|--------|
| [15:0] | PROG_LENGTH_LSB | | Program counter length, Bits[15:0]. | 0x0000 | R |

Program Counter Max Length, Bits[23:16] Register

Address: 0xF465, Reset: 0x0000, Name: PROG_CNTR_MAXLENGTH0

This register, in combination with Register 0xF466 (PROG_CNTR_MAXLENGTH1), keeps track of the highest peak value reached by the program counter since the DSP core started. It can be cleared using Register 0xF462 (PROG_CNTR_CLEAR).

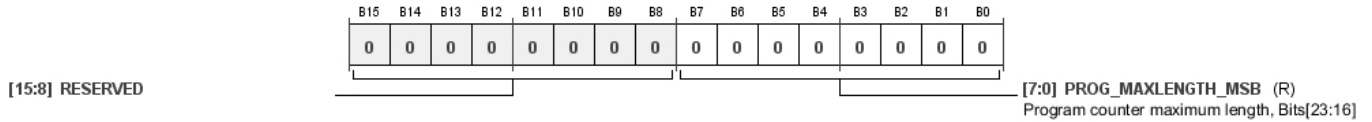


Table 109. Bit Descriptions for PROG_CNTR_MAXLENGTH0

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|--------------------|----------|--|-------|--------|
| [15:8] | RESERVED | | | 0x0 | RW |
| [7:0] | PROG_MAXLENGTH_MSB | | Program counter maximum length, Bits[23:16]. | 0x00 | R |

Program Counter Max Length, Bits[15:0] Register

Address: 0xF466, Reset: 0x0000, Name: PROG_CNTR_MAXLENGTH1

This register, in combination with Register 0xF465 (PROG_CNTR_MAXLENGTH0), keeps track of the highest peak value reached by the program counter since the DSP core started. It can be cleared using Register 0xF462 (PROG_CNTR_CLEAR).

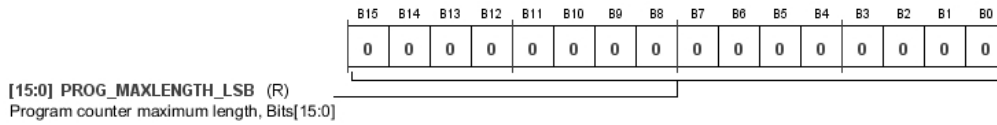


Table 110. Bit Descriptions for PROG_CNTR_MAXLENGTH1

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|--------------------|----------|---|--------|--------|
| [15:0] | PROG_MAXLENGTH_LSB | | Program counter maximum length, Bits[15:0]. | 0x0000 | R |

MULTIPURPOSE PIN CONFIGURATION REGISTERS

Multipurpose Pin Mode Register

Address: 0xF510 to 0xF51D (Increments of 0x1), Reset: 0x0000, Name: MPx_MODE

These 14 registers configure the multipurpose pins. Certain multipurpose pins can function as audio clock pins, control bus pins, or general-purpose input or output (GPIO) pins.

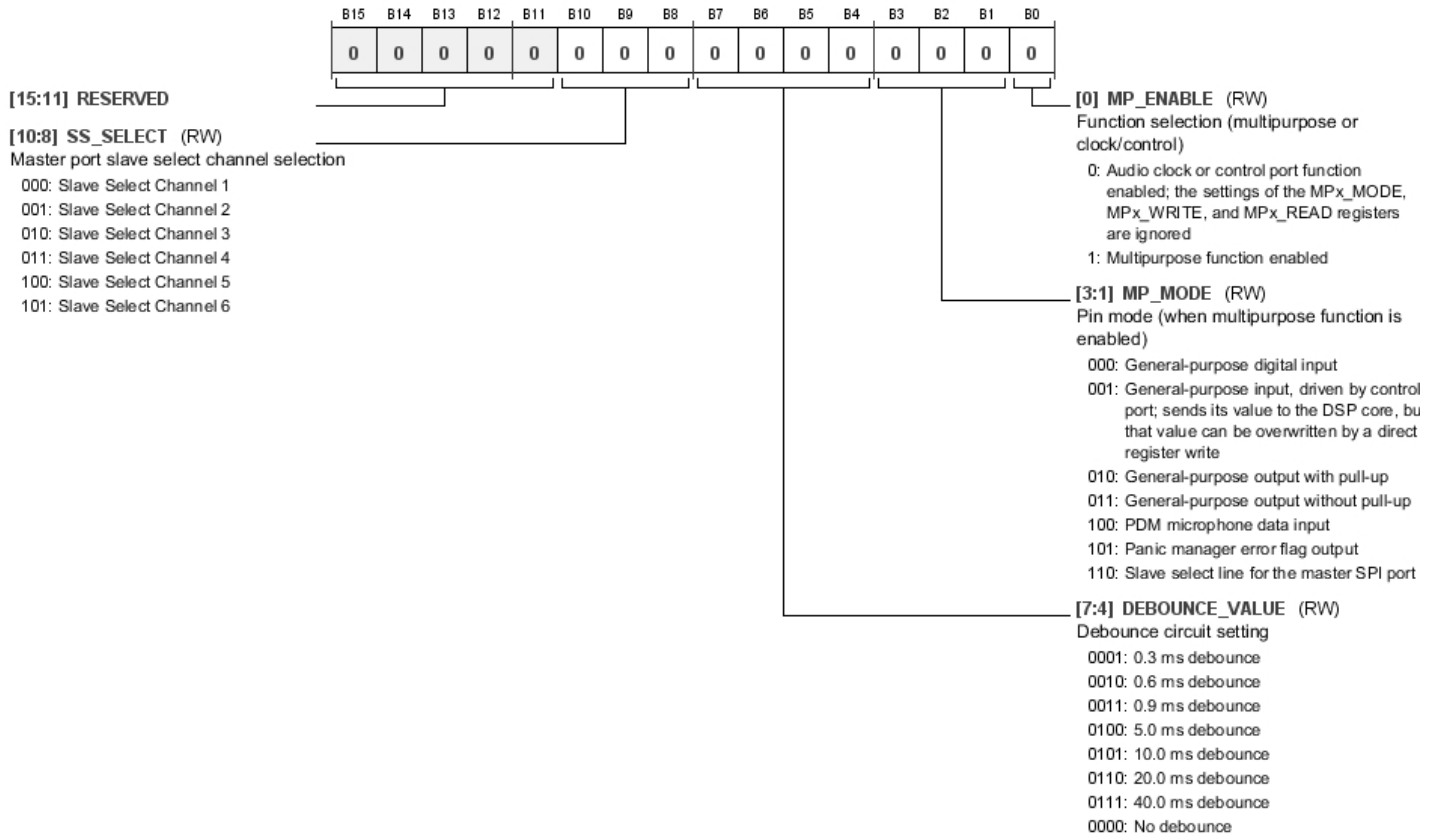


Table 111. Bit Descriptions for MPx_MODE

| Bits | Bit Name | Settings | Description | Reset | Access |
|---------|-----------|----------|--|-------|--------|
| [15:11] | RESERVED | | | 0x0 | RW |
| [10:8] | SS_SELECT | | Master port slave select channel selection. If the pin is configured as a slave select line (Bits[3:1] (MP_MODE) = 0b110), these bits configure which slave select channel the pin corresponds to. This allows multiple slave devices to be connected to the SPI master port, all using different slave select lines. The first slave select signal (Slave Select 0) is always routed to the SS_M/MP0 pin. The remaining six slave select lines can be routed to any multipurpose pin that has been configured as a slave select output. | 0x0 | RW |
| | | 000 | Slave Select Channel 1 | | |
| | | 001 | Slave Select Channel 2 | | |
| | | 010 | Slave Select Channel 3 | | |
| | | 011 | Slave Select Channel 4 | | |
| | | 100 | Slave Select Channel 5 | | |
| | | 101 | Slave Select Channel 6 | | |

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|----------------|---|--|-------|--------|
| [7:4] | DEBOUNCE_VALUE | 0001 0.3 ms debounce 0010 0.6 ms debounce 0011 0.9 ms debounce 0100 5.0 ms debounce 0101 10.0 ms debounce 0110 20.0 ms debounce 0111 40.0 ms debounce 0000 No debounce | Debounce circuit setting. These bits configure the duration of the debounce circuitry when the corresponding pin is configured as an input (Bits[3:1] (MP_MODE) = 0b000). | 0x0 | RW |
| [3:1] | MP_MODE | 000 General-purpose digital input 001 General-purpose input, driven by control port; sends its value to the DSP core, but that value can be overwritten by a direct register write 010 General-purpose output with pull-up 011 General-purpose output without pull-up 100 PDM microphone data input 101 Panic manager error flag output 110 Slave select line for the master SPI port | Pin mode (when multipurpose function is enabled). These bits select the function of the corresponding pin if it is enabled in multipurpose mode (Bit 0 (MP_ENABLE) = 0b1). | 0x0 | RW |
| 0 | MP_ENABLE | 0 Audio clock or control port function enabled; the settings of the MPx_MODE, MPx_WRITE, and MPx_READ registers are ignored 1 Multipurpose function enabled | Function selection (multipurpose or clock/control). This bit selects whether the corresponding pin is used as a multipurpose pin or as its primary function (which could be either an audio clock or control bus pin). | 0x0 | RW |

Multipurpose Pin Write Value Register

Address: 0xF520 to 0xF52D (Increments of 0x1), Reset: 0x0000, Name: MPx_WRITE

If a multipurpose pin is configured as an output driven by the control port (the corresponding Bits[3:1] (MP_MODE) = 0b001), the value that is output from the DSP core can be configured by directly writing to these registers.

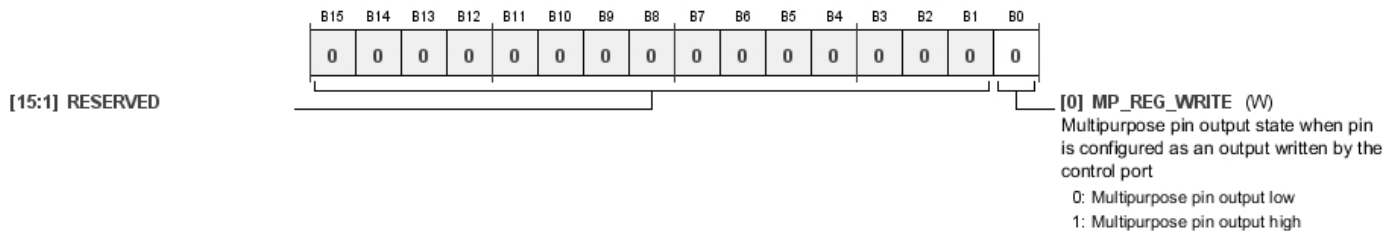


Table 112. Bit Descriptions for MPx_WRITE

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|--------------|---|---|-------|--------|
| [15:1] | RESERVED | | | 0x0 | W |
| 0 | MP_REG_WRITE | 0 Multipurpose pin output low 1 Multipurpose pin output high | Multipurpose pin output state when pin is configured as an output written by the control port. This register configures the value seen by the DSP core for the corresponding multipurpose pin input. The pin can have two states: logic low (off) or logic high (on). | 0x0 | W |

Multipurpose Pin Read Value Registers

Address: 0xF530 to 0xF53D (Increments of 0x1), Reset: 0x0000, Name: MP_x_READ

These registers log the current state of the multipurpose pins when they are configured as inputs. The pins can have two states: logic low (off) or logic high (on).

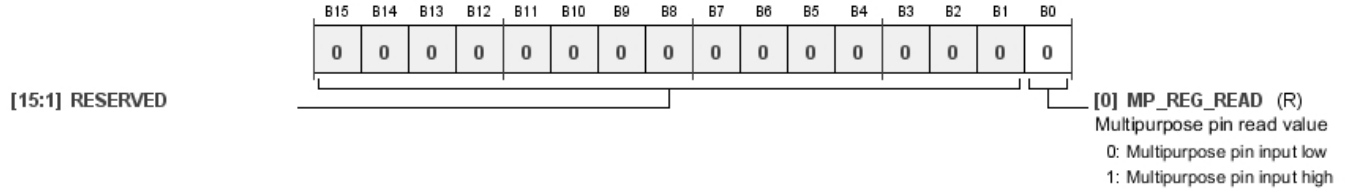


Table 113. Bit Descriptions for MP_x_READ

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-------------|----------|---|-------|--------|
| [15:1] | RESERVED | | | 0x0 | R |
| 0 | MP_REG_READ | 0 1 | Multipurpose pin read value. Multipurpose pin input low Multipurpose pin input high | 0x0 | R |

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.

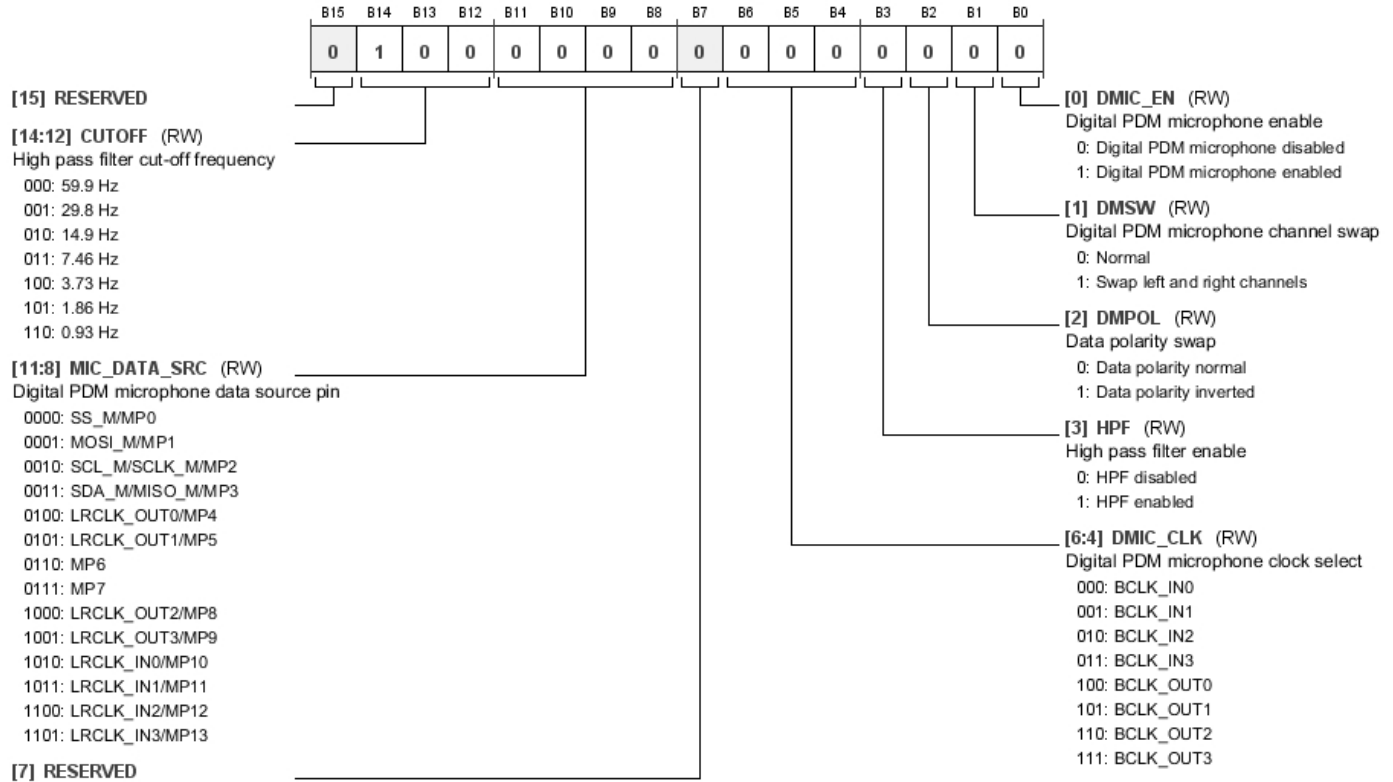


Table 114. Bit Descriptions for DMIC_CTRLx

| Bits | Bit Name | Settings | Description | Reset | Access |
|---------|----------|---|---|-------|--------|
| 15 | RESERVED | | | 0x0 | RW |
| [14:12] | CUTOFF | 000 001 010 011 100 101 110 | 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. 59.9 Hz 29.8 Hz 14.9 Hz 7.46 Hz 3.73 Hz 1.86 Hz 0.93 Hz | 0x4 | RW |

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|--------------|--|--|-------|--------|
| [11:8] | MIC_DATA_SRC | 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 0111 MP7 1000 LRCLK_OUT2/MP8 1001 LRCLK_OUT3/MP9 1010 LRCLK_IN0/MP10 1011 LRCLK_IN1/MP11 1100 LRCLK_IN2/MP12 1101 LRCLK_IN3/MP13 | 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 |
| 7 | RESERVED | | | 0x0 | RW |
| [6:4] | DMIC_CLK | 000 BCLK_IN0 001 BCLK_IN1 010 BCLK_IN2 011 BCLK_IN3 100 BCLK_OUT0 101 BCLK_OUT1 110 BCLK_OUT2 111 BCLK_OUT3 | Digital PDM microphone clock select. A valid bit clock signal must be assigned to the PDM microphones. Any of the four BCLK_INPUTx or four BCLK_OUTPUTx signals can be used. A trace must connect the selected pin to the clock input pin on the corresponding PDM microphone(s). If the corresponding BCLK_x pin is not configured in master mode, use an external clock source, with the BCLK_x pin and the PDM microphone acting as slaves. | 0x0 | RW |
| 3 | HPF | 0 HPF disabled 1 HPF enabled | High-pass filter enable. This bit enables or disables a high-pass filter to remove dc components from the microphone data signals. The cutoff of the filter is controlled by Bits[14:12] (CUTOFF). | 0x0 | RW |
| 2 | DMPOL | 0 Data polarity normal 1 Data polarity inverted | Data polarity swap. When this bit is set to 0b0, a logic high data input is treated as logic high, and a logic low data input is treated as logic low. When this bit is set to 0b1, the opposite is true: a logic high data input is treated as a logic low, and a logic low data input is treated as logic high. This effectively inverts the amplitude of the incoming audio data. | 0x0 | RW |
| 1 | DMSW | 0 Normal 1 Swap left and right channels | Digital PDM microphone channel swap. In DMIC_CTRL0, this bit swaps PDM Microphone Channel 0 and PDM Microphone Channel 1. In the DMIC_CTRL1 register, this bit swaps PDM Microphone Channel 2 and PDM Microphone Channel 3. | 0x0 | RW |
| 0 | DMIC_EN | 0 Digital PDM microphone disabled 1 Digital PDM microphone enabled | Digital PDM microphone enable. This bit enables or disables the data input from the PDM microphones. | 0x0 | RW |

ASRC STATUS AND CONTROL REGISTERS

ASRC Lock Status Register

Address: 0xF580, Reset: 0x0000, Name: ASRC_LOCK

This register contains eight bits that represent the lock status of each ASRC stereo pair. Lock status requires three conditions: the output target rate is set, the input rate is steady and has been detected, and the ratio between input and output rates has been calculated. If all of these conditions are true for a given stereo ASRC, the corresponding lock bit is low. If any of these conditions is not true, the corresponding lock bit is high.

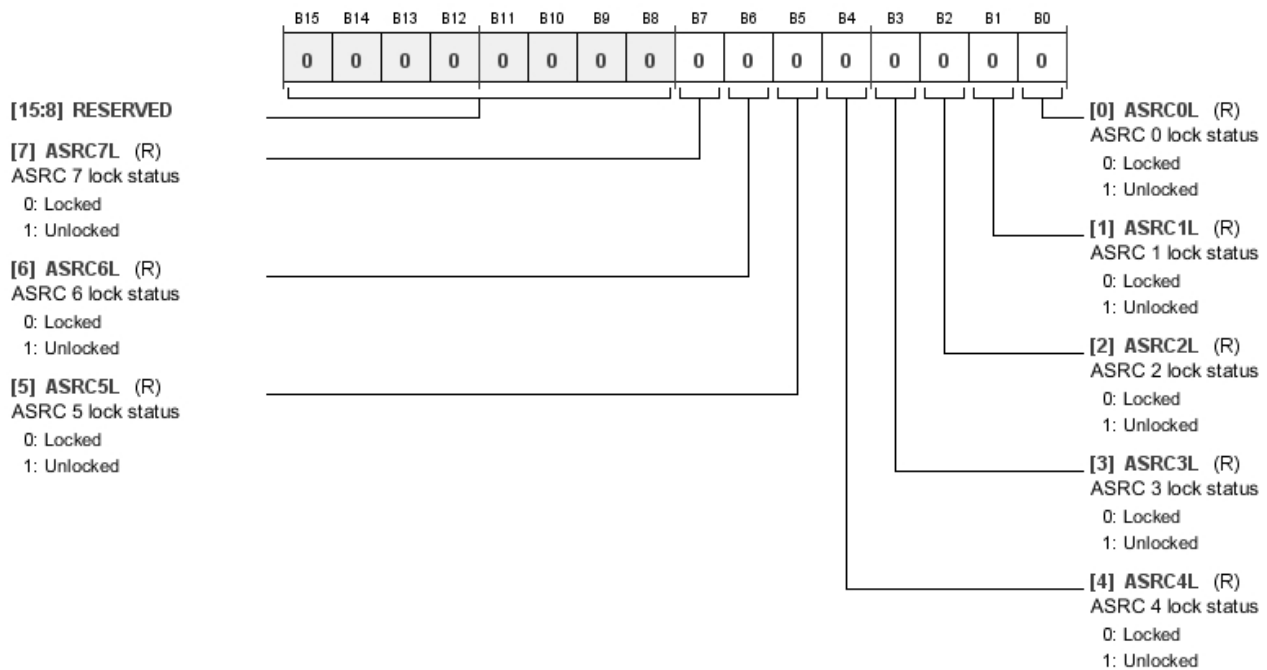


Table 115. Bit Descriptions for ASRC_LOCK

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|----------|----------|---|-------|--------|
| [15:8] | RESERVED | | | 0x0 | RW |
| 7 | ASRC7L | 0 1 | ASRC 7 lock status. Locked Unlocked | 0x0 | R |
| 6 | ASRC6L | 0 1 | ASRC 6 lock status. Locked Unlocked | 0x0 | R |
| 5 | ASRC5L | 0 1 | ASRC 5 lock status. Locked Unlocked | 0x0 | R |
| 4 | ASRC4L | 0 1 | ASRC 4 lock status. Locked Unlocked | 0x0 | R |
| 3 | ASRC3L | 0 1 | ASRC 3 lock status. Locked Unlocked | 0x0 | R |
| 2 | ASRC2L | 0 1 | ASRC 2 lock status. Locked Unlocked | 0x0 | R |

| Bits | Bit Name | Settings | Description | Reset | Access |
|------|----------|----------|---|-------|--------|
| 1 | ASRC1L | 0 1 | ASRC 1 lock status. Locked Unlocked | 0x0 | R |
| 0 | ASRC0L | 0 1 | ASRC 0 lock status. Locked Unlocked | 0x0 | R |

ASRC Mute Register

Address: 0xF581, Reset: 0x0000, Name: ASRC_MUTE

This register contains controls related to the muting of audio on ASRC channels. Bits[7:0] (ASRCxM) are individual mute controls for each stereo ASRC. Bit 8 (ASRC_RAMP0) and Bit 9 (ASRC_RAMP1) enable or disable an optional volume ramp-up and ramp-down to smoothly transition between muted and unmuted states. The mute and unmute ramps are linear. The duration of the ramp is determined by the sample rate of the DSP core, which is set by Register 0xF401 (START_PULSE). The ramp takes exactly 2048 input samples to complete. For example, if the sample rate of audio entering an ASRC channel is 48 kHz, the duration of the ramp is 2048/48,000 = 42.7 ms. If the sample rate of audio entering an ASRC channel is 6 kHz, the duration of the ramp is 2048/6000 = 341.3 ms. Bit 10 (LOCKMUTE) allows the ASRCs to automatically mute themselves in the event that lock status is lost or not attained.

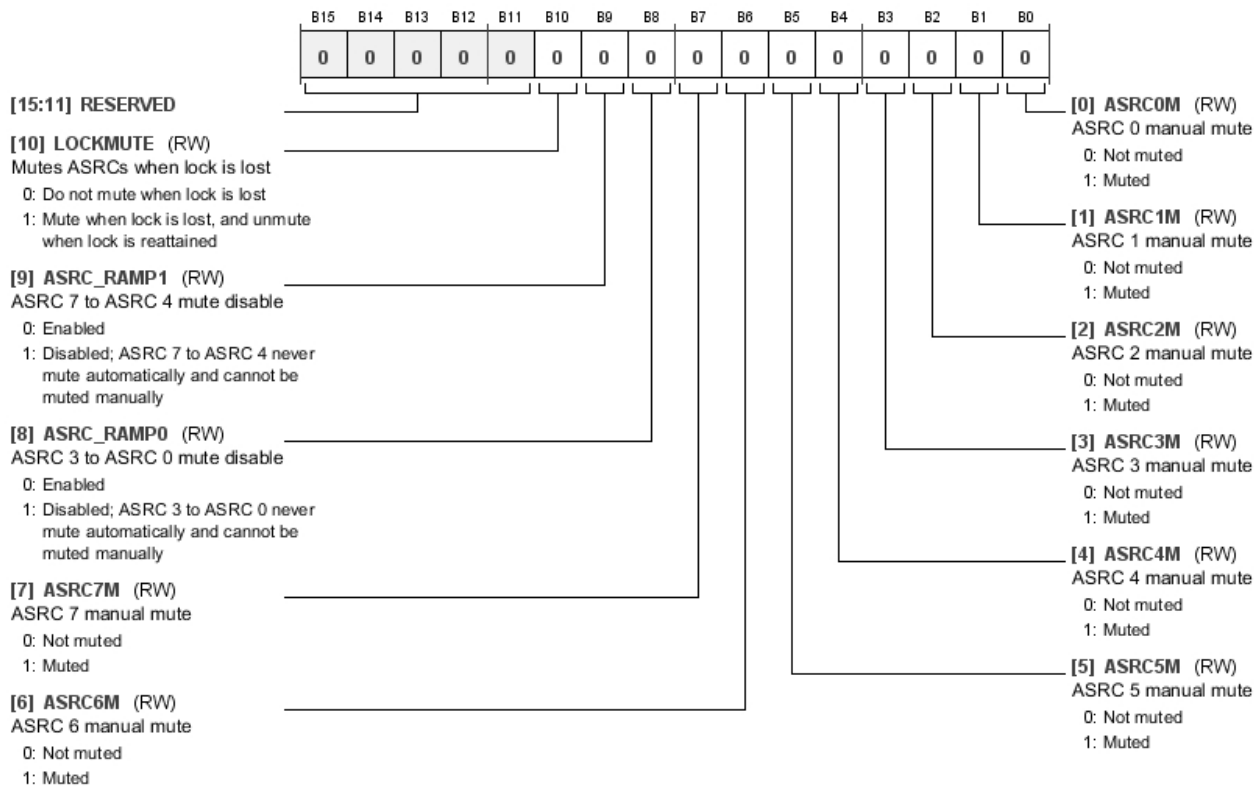


Table 116. Bit Descriptions for ASRC_MUTE

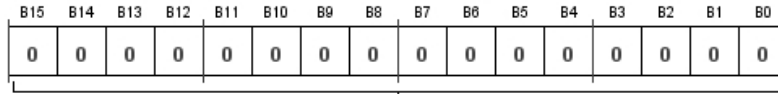
| Bits | Bit Name | Settings | Description | Reset | Access |
|---------|----------|----------|-------------|-------|--------|
| [15:11] | RESERVED | | | 0x0 | RW |

| Bits | Bit Name | Settings | Description | Reset | Access |
|------|------------|----------|--|-------|--------|
| 10 | LOCKMUTE | 0 1 | Mutes ASRCs when lock is lost. When this bit is enabled, individual stereo ASRCs automatically mute on the event that lock status is lost (for example, if the sample rate of the input suddenly changes and the ASRC needs to regain lock), provided that the corresponding ASRC_RAMPx bit is set to 0b0 (enabled). This automatic mute uses a volume ramp instead of an instantaneous mute to avoid click-and-pop noises on the output. When lock status is attained again (and the corresponding ASRC_RAMPx and ASRCxM bits are set to 0b0 (enabled) and 0b0 (unmuted), respectively), the ASRC automatically unmutes using a volume ramp. However, because there is a period of uncertainty when the ASRC is attaining lock, there still may be noise on the ASRC outputs when the input signal returns. Measures must be taken in the DSP program to delay the unmuting of the ASRC output signals if this noise is not desired. The individual ASRCxM mute bits override the automatic LOCKMUTE behavior. 0 Do not mute when lock is lost 1 Mute when lock is lost, and unmute when lock is regained | 0x0 | RW |
| 9 | ASRC_RAMP1 | 0 1 | ASRC 7 to ASRC 4 mute disable. ASRC 7 to ASRC 4 (Channel 15 to Channel 8) are defined as ASRC Block 1. This bit enables or disables mute ramping for all ASRCs in Block 1. If this bit is 0b1, Bit 7 (ASRC7M), Bit 6 (ASRC6M), Bit 5 (ASRC5M), and Bit 4 (ASRC4M) are ignored, and the outputs of ASRC 7 to ASRC 4 are active at all times. 0 Enabled 1 Disabled; ASRC 7 to ASRC 4 never mute automatically and cannot be muted manually | 0x0 | RW |
| 8 | ASRC_RAMPO | 0 1 | ASRC 3 to ASRC 0 mute disable. ASRC 3 to ASRC 0 (Channel 7 to Channel 0) are defined as ASRC Block 0. This bit enables or disables mute ramping for all ASRCs in Block 0. If this bit is 0b1, Bit 3 (ASRC3M), Bit 2 (ASRC2M), Bit 1 (ASRC1M), and Bit 0 (ASRC0M) are ignored, and the outputs of ASRC 3 to ASRC 0 are active at all times. 0 Enabled 1 Disabled; ASRC 3 to ASRC 0 never mute automatically and cannot be muted manually | 0x0 | RW |
| 7 | ASRC7M | 0 1 | ASRC 7 manual mute. 0 Not muted 1 Muted | 0x0 | RW |
| 6 | ASRC6M | 0 1 | ASRC 6 manual mute. 0 Not muted 1 Muted | 0x0 | RW |
| 5 | ASRC5M | 0 1 | ASRC 5 manual mute. 0 Not muted 1 Muted | 0x0 | RW |
| 4 | ASRC4M | 0 1 | ASRC 4 manual mute. 0 Not muted 1 Muted | 0x0 | RW |
| 3 | ASRC3M | 0 1 | ASRC 3 manual mute. 0 Not muted 1 Muted | 0x0 | RW |
| 2 | ASRC2M | 0 1 | ASRC 2 manual mute. 0 Not muted 1 Muted | 0x0 | RW |
| 1 | ASRC1M | 0 1 | ASRC 1 manual mute. 0 Not muted 1 Muted | 0x0 | RW |
| 0 | ASRC0M | 0 1 | ASRC 0 manual mute. 0 Not muted 1 Muted | 0x0 | RW |

ASRC Ratio Registers

Address: 0xF582 to 0xF589 (Increments of 0x1), Reset: 0x0000, Name: ASRCx_RATIO

These eight read only registers contain the sample rate conversion ratio of the ASRC, which is calculated as the ratio between the detected input rate and the selected target output rate. The format of the value stored in these registers is 4.12 format. For example, a ratio of 1 is shown as 0b0001000000000000 (0x1000). A ratio of 2 is shown as 0b0010000000000000 (0x2000). A ratio of 0.5 is shown as 0b0000100000000000 (0x0800).



[15:0] ASRC_RATIO (RW)
Output rate of the ASRC in 4.12 format

Table 117. Bit Descriptions for ASRCx_RATIO

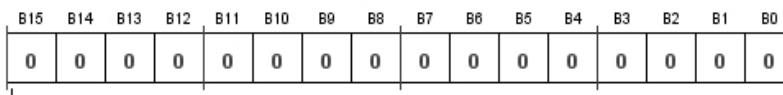
| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|------------|----------|--|--------|--------|
| [15:0] | ASRC_RATIO | | Output rate of the ASRC in 4.12 format. The value of this register represents the input to output rate of the corresponding ASRC. It is stored in 4.12 format. | 0x0000 | RW |

AUXILIARY ADC REGISTERS

Auxiliary ADC Read Value Register

Address: 0xF5A0 to 0xF5A5 (Increments of 0x1), Reset: 0x0000, Name: ADC_READx

These six register contains the output data of the auxiliary ADC for the corresponding channel. Each of the six channels of the ADC are updated once per audio frame. The format for the value in this register is 6.10 format, but the top six bits are always zero, meaning that the effective format is 0.10 format. If, for example, the input to the corresponding auxiliary ADC channel is equal to AVDD (the full-scale analog input voltage), this register reads its maximum value of 0b0000001111111111 (0x3FF). If the input to the auxiliary ADC channel is AVDD/2, this register reads 0b0000001000000000 (0x200). If the input to the auxiliary ADC channel is AVDD/4, this register reads 0b0000000100000000 (0x100).



[15:0] ADC_VALUE (RW)
ADC input value in 0.10 format, as a proportion of AVDD

Table 118. Bit Descriptions for ADC_READx

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-----------|----------|--|--------|--------|
| [15:0] | ADC_VALUE | | ADC input value in 0.10 format, as a proportion of AVDD. Instantaneous value of the sampled data on the ADC input. The top six bits are not used, and the least significant 10 bits contain the value of the ADC input. The minimum value of 0 maps to 0 V, and the maximum value of 1023 maps to 3.3 V ± 10% (equal to the AVDD supply). Values between 0 and 1023 are linearly mapped to dc voltages between 0 V and AVDD. | 0x0000 | RW |

S/PDIF INTERFACE REGISTERS

S/PDIF Receiver Lock Bit Detection Register

Address: 0xF600, Reset: 0x0000, Name: SPDIF_LOCK_DET

This register contains a flag that monitors the S/PDIF receiver and provides a way to check the validity of the input signal.

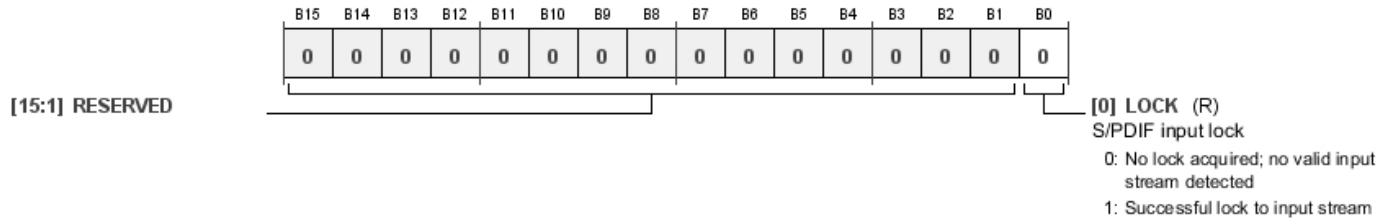


Table 119. Bit Descriptions for SPDIF_LOCK_DET

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|----------|----------|---|-------|--------|
| [15:1] | RESERVED | | | 0x0 | RW |
| 0 | LOCK | 0 1 | S/PDIF input lock. 0 No lock acquired; no valid input stream detected 1 Successful lock to input stream | 0x0 | R |

S/PDIF Receiver Control Register

Address: 0xF601, Reset: 0x0000, Name: SPDIF_RX_CTRL

This register provides controls that govern the behavior of the S/PDIF receiver.

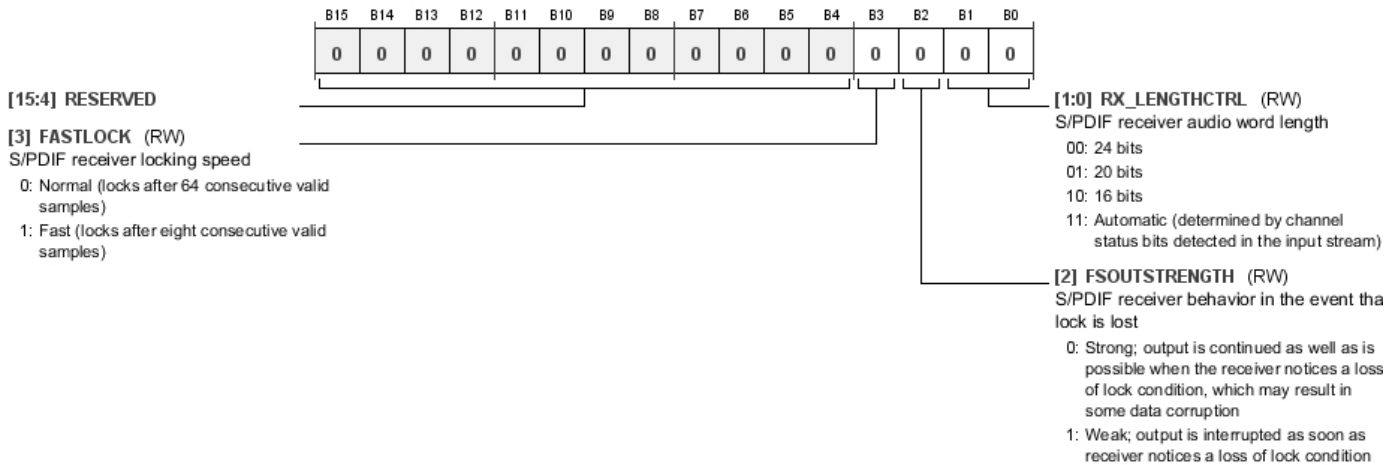


Table 120. Bit Descriptions for SPDIF_RX_CTRL

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|---------------|----------|--|-------|--------|
| [15:4] | RESERVED | | | 0x0 | RW |
| 3 | FASTLOCK | 0 1 | S/PDIF receiver locking speed. 0 Normal (locks after 64 consecutive valid samples) 1 Fast (locks after eight consecutive valid samples) | 0x0 | RW |
| 2 | FSOUTSTRENGTH | 0 1 | S/PDIF receiver behavior in the event that lock is lost. FSOUTSTRENGTH applies to the output of the recovered frame clock from the S/PDIF receiver. 0 Strong; output is continued as well as is possible when the receiver notices a loss of lock condition, which may result in some data corruption 1 Weak; output is interrupted as soon as receiver notices a loss of lock condition | 0x0 | RW |

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|---------------|----------------------|---|-------|--------|
| [1:0] | RX_LENGTHCTRL | 00 01 10 11 | S/PDIF receiver audio word length. 24 bits 20 bits 16 bits Automatic (determined by channel status bits detected in the input stream) | 0x0 | RW |

Decoded Signals From the S/PDIF Receiver Register

Address: 0xF602, Reset: 0x0000, Name: SPDIF_RX_DECODE

This register monitors the embedded nonaudio data bits in the incoming S/PDIF stream and decodes them, providing insight into the data format of the S/PDIF input stream.

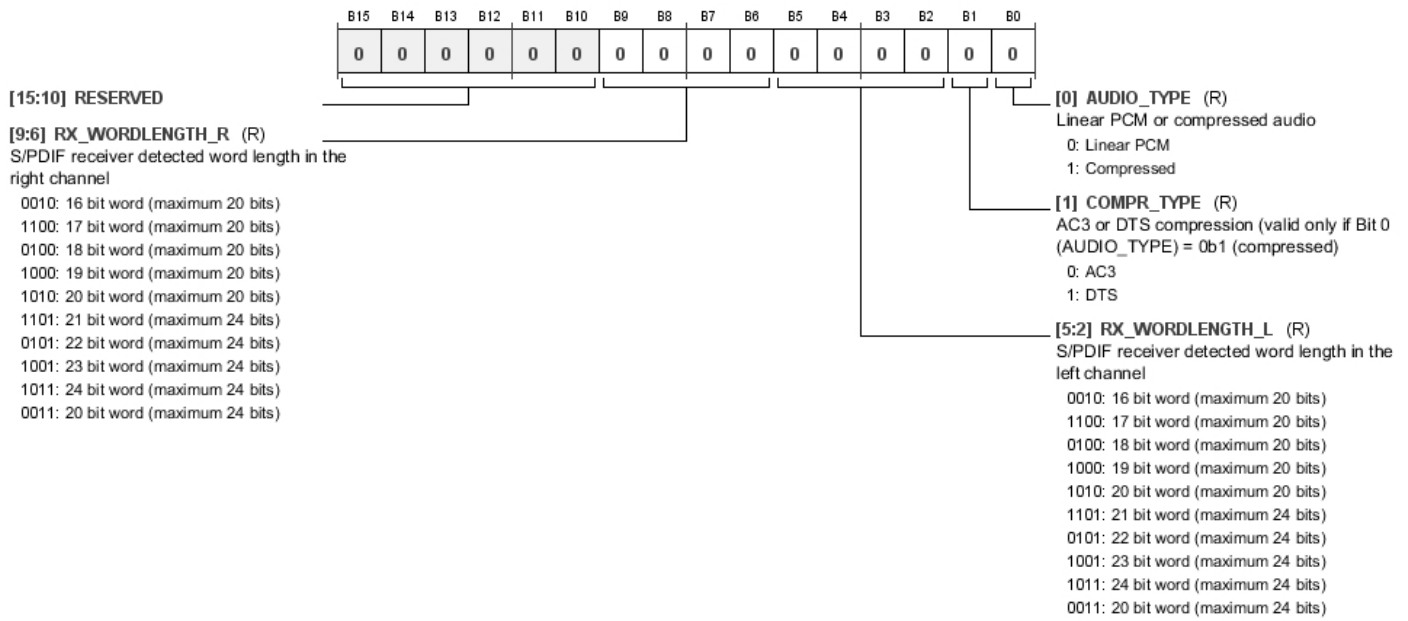


Table 121. Bit Descriptions for SPDIF_RX_DECODE

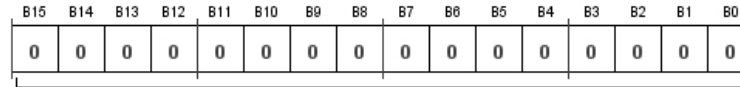
| Bits | Bit Name | Settings | Description | Reset | Access |
|---------|-----------------|--|--|-------|--------|
| [15:10] | RESERVED | | | 0x0 | RW |
| [9:6] | RX_WORDLENGTH_R | 0010 1100 0100 1000 1010 1101 0101 1001 1011 0011 | S/PDIF receiver detected word length in the right channel. 16 bit word (maximum 20 bits) 17 bit word (maximum 20 bits) 18 bit word (maximum 20 bits) 19 bit word (maximum 20 bits) 20 bit word (maximum 20 bits) 21 bit word (maximum 24 bits) 22 bit word (maximum 24 bits) 23 bit word (maximum 24 bits) 24 bit word (maximum 24 bits) 20 bit word (maximum 24 bits) | 0x0 | R |

| Bits | Bit Name | Settings | Description | Reset | Access |
|-------|-----------------|--|---|-------|--------|
| [5:2] | RX_WORDLENGTH_L | 0010 16 bit word (maximum 20 bits) 1100 17 bit word (maximum 20 bits) 0100 18 bit word (maximum 20 bits) 1000 19 bit word (maximum 20 bits) 1010 20 bit word (maximum 20 bits) 1101 21 bit word (maximum 24 bits) 0101 22 bit word (maximum 24 bits) 1001 23 bit word (maximum 24 bits) 1011 24 bit word (maximum 24 bits) 0011 20 bit word (maximum 24 bits) | S/PDIF receiver detected word length in the left channel. | 0x0 | R |
| 1 | COMPR_TYPE | 0 AC3 1 DTS | AC3 or DTS compression (valid only if Bit 0 (AUDIO_TYPE) = 0b1 (compressed)). | 0x0 | R |
| 0 | AUDIO_TYPE | 0 Linear PCM 1 Compressed | Linear PCM or compressed audio. | 0x0 | R |

Compression Mode From the S/PDIF Receiver Register

Address: 0xF603, Reset: 0x0000, Name: SPDIF_RX_COMPRMODE

If the incoming S/PDIF data has been encoded using a compression algorithm, this register displays the 16-bit code that represents the type of compression being used.



[15:0] COMPR_MODE (R)
 Compression mode detected by the S/PDIF receiver

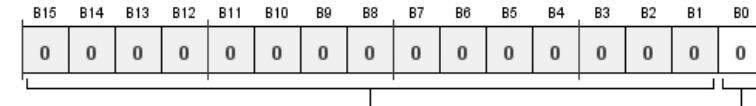
Table 122. Bit Descriptions for SPDIF_RX_COMPRMODE

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|------------|----------|---|--------|--------|
| [15:0] | COMPR_MODE | | Compression mode detected by the S/PDIF receiver. | 0x0000 | R |

Automatically Resume S/PDIF Receiver Audio Input Register

Address: 0xF604, Reset: 0x0000, Name: SPDIF_RESTART

When the S/PDIF receiver loses lock on the incoming S/PDIF signal, which can occur due to issues with signal integrity, the receiver automatically mutes itself. This register determines whether the S/PDIF receiver then automatically resumes outputting data if the S/PDIF receiver subsequently begins to receive valid data and a lock condition is reattained. By default, the S/PDIF receiver does not automatically resume audio when lock is lost (Register 0xF604 (SPDIF_RESTART), Bit 0 (RESTART_AUDIO) = 0b0); and, therefore, the user must manually reset the S/PDIF receiver by toggling Register 0xF604 (SPDIF_RESTART), Bit 0 (RESTART_AUDIO), from 0b0 to 0b1 and then back to 0b0 again. To ensure that the S/PDIF receiver always begins outputting data when a valid input signal is detected, set Register 0xF604 (SPDIF_RESTART), Bit 0 (RESTART_AUDIO), to 0b1 at all times.



[15:1] RESERVED

[0] RESTART_AUDIO (RW)
 Allows the S/PDIF receiver to automatically resume outputting audio when it successfully recovers from a loss of lock
 0: Do not automatically restart the audio when a relock occurs
 1: Restarts the audio automatically when a relock occurs, and resets Register 0xF605 (SPDIF_LOSS_OF_LOCK), Bit 0 (LOSS_OF_LOCK)

Table 123. Bit Descriptions for SPDIF_RESTART

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|---------------|----------|---|-------|--------|
| [15:1] | RESERVED | | | 0x0 | RW |
| 0 | RESTART_AUDIO | 0 1 | Allows the S/PDIF receiver to automatically resume outputting audio when it successfully recovers from a loss of lock. Do not automatically restart the audio when a relock occurs Restarts the audio automatically when a relock occurs, and resets Register 0xF605 (SPDIF_LOSS_OF_LOCK), Bit 0 (LOSS_OF_LOCK) | 0x0 | RW |

S/PDIF Receiver Loss of Lock Detection Register

Address: 0xF605, Reset: 0x0000, Name: SPDIF_LOSS_OF_LOCK

This bit monitors the S/PDIF lock status and checks to see if the lock is lost during operation of the S/PDIF receiver. This condition can arise when, for example, a valid S/PDIF input signal was present for an extended period of time, but signal integrity worsened for a brief period, causing the receiver to then lose its lock to the input signal. In this case, Bit 0 (LOSS_OF_LOCK) transitions from 0b0 to 0b1 and remains set at 0b1 indefinitely. This indicates that, at some point during the operation of the device, lock to the input stream was lost. Bit 0 (LOSS_OF_LOCK) stays high at 0b1 until Register 0xF604 (SPDIF_RESTART), Bit 0 (RESTART_AUDIO), is set to 0b1, which clears Bit 0 (LOSS_OF_LOCK) back to 0b0. At that point, Register 0xF604 (SPDIF_RESTART), Bit 0 (RESTART_AUDIO), can be reset to 0b0 if required.

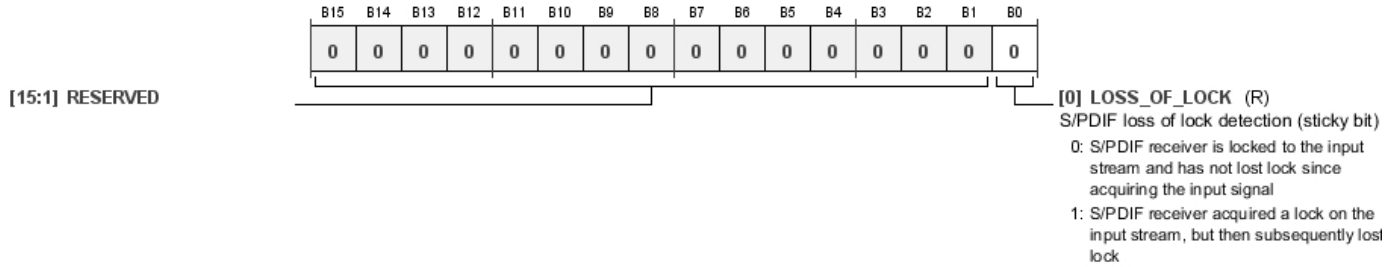


Table 124. Bit Descriptions for SPDIF_LOSS_OF_LOCK

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|--------------|---|---|-------|--------|
| [15:1] | RESERVED | | | 0x0 | RW |
| 0 | LOSS_OF_LOCK | 0 S/PDIF receiver is locked to the input stream and has not lost lock since acquiring the input signal 1 S/PDIF receiver acquired a lock on the input stream but then, subsequently, lost lock | S/PDIF loss of lock detection (sticky bit). | 0x0 | R |

S/PDIF Receiver Auxiliary Outputs Enable Register

Address: 0xF608, Reset: 0x0000, Name: SPDIF_AUX_EN

The S/PDIF receiver decodes embedded nonaudio data bits on the incoming data stream, including channel status, user data, validity bits, and parity bits. This information, together with the decoded audio data, can optionally be output on one of the SDATA_OUTx pins using Register 0xF608 (SPDIF_AUX_EN). The serial output port selected by Bits[3:0] (TDMOUT) outputs an 8-channel TDM stream containing this decoded information.

Channel 0 in the TDM8 stream contains the 24 audio bits from the left S/PDIF input channel, followed by eight zero bits.

Channel 1 in the TDM8 stream contains 20 zero bits, the parity bit, validity bit, user data bit, and the channel status bit from the left S/PDIF input channel, followed by eight zero bits.

Channel 2 in the TDM8 stream contains 22 zero bits, followed by the compression type bit (0b0 represents AC3 and 0b1 represents DTS) and the audio type bit (0b0 represents PCM and 0b1 represents compressed), followed by eight zero bits.

Channel 3 in the TDM8 stream contains 32 zero bits.

Channel 4 in the TDM8 stream contains the 24 audio bits from the right S/PDIF input channel, followed by eight zero bits.

Channel 5 in the TDM8 stream contains 20 zero bits followed by the parity bit, validity bit, user data bit, and channel status bit from the right S/PDIF input channel, followed by eight zero bits.

Channel 6 in the TDM8 stream contains 32 zero bits.

Channel 7 in the TDM8 stream contains 23 zero bits, the block start bit, and eight zero bits.

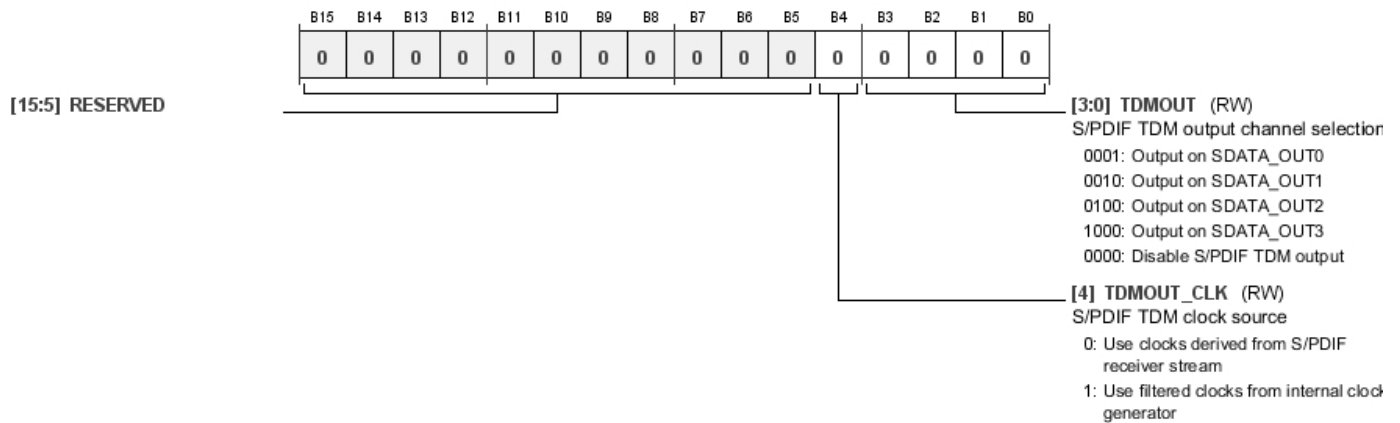


Table 125. Bit Descriptions for SPDIF_AUX_EN

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|------------|--------------------------------------|---|-------|--------|
| [15:5] | RESERVED | | | 0x0 | RW |
| 4 | TDMOUT_CLK | 0 1 | S/PDIF TDM clock source. When Bits[3:0] (TDMOUT) are configured to output S/PDIF receiver data on one of the SDATA_OUTx pins, the corresponding serial port must be set in master mode; and Bit 4 (TDMOUT_CLK) configures which clock signals are used on the corresponding BCLK_OUTx and LRCLK_OUTx pins. If Bit 4 (TDMOUT_CLK) = 0b0, the clock signals recovered from the S/PDIF input signal are used to clock the serial output. If Bit 4 (TDMOUT_CLK) = 0b1, the output of Clock Generator 3 is used to clock serial output; and Register 0xF026 (CLK_GEN3_SRC), Bits[3:0] (FREF_PIN), must be 0b1110, and Register 0xF026 (CLK_GEN3_SRC), Bit 4 (CLK_GEN3_SRC), must be 0b1. | 0x0 | RW |
| [3:0] | TDMOUT | 0001 0010 0100 1000 0000 | S/PDIF TDM output channel selection. | 0x0 | RW |

S/PDIF Receiver Auxiliary Bits Ready Flag Register

Address: 0xF60F, Reset: 0x0000, Name: SPDIF_RX_AUXBIT_READY

The decoded channel status, user data, validity, and parity bits are recovered from the input signal one frame at a time until a full block of 192 frames is received. When all of the 192 frames are received and decoded, Bit 0 (AUXBITS_READY), changes state from 0b0 to 0b1, indicating that the full block of data has been recovered and is available to be read from the corresponding registers.

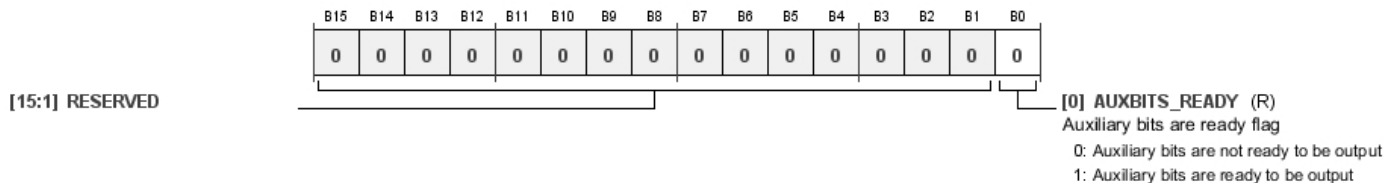


Table 126. Bit Descriptions for SPDIF_RX_AUXBIT_READY

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|---------------|----------|--|-------|--------|
| [15:1] | RESERVED | | | 0x0 | RW |
| 0 | AUXBITS_READY | 0 1 | Auxiliary bits are ready flag. 0 Auxiliary bits are not ready to be output 1 Auxiliary bits are ready to be output | 0x0 | R |

S/PDIF Receiver Channel Status Bits (Left) Register

Address: 0xF610 to 0xF61B (Increments of 0x1), Reset: 0x0000, Name: SPDIF_RX_CS_LEFT_x

These 12 registers store the 192 channel status bits decoded from the left channel of the S/PDIF input stream.

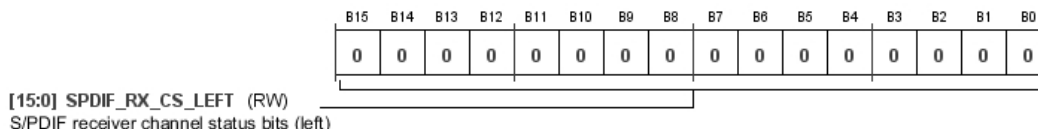


Table 127. Bit Descriptions for SPDIF_RX_CS_LEFT_x

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|------------------|----------|---|--------|--------|
| [15:0] | SPDIF_RX_CS_LEFT | | S/PDIF receiver channel status bits (left). | 0x0000 | R |

S/PDIF Receiver Channel Status Bits (Right) Register

Address: 0xF620 to 0xF62B (Increments of 0x1), Reset: 0x0000, Name: SPDIF_RX_CS_RIGHT_x

These 12 registers store the 192 channel status bits decoded from the right channel of the S/PDIF input stream.

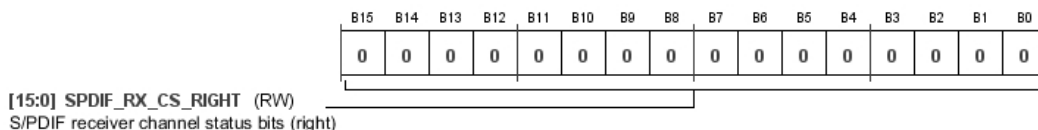


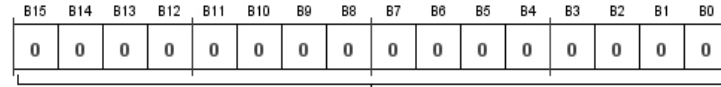
Table 128. Bit Descriptions for SPDIF_RX_CS_RIGHT_x

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-------------------|----------|--|--------|--------|
| [15:0] | SPDIF_RX_CS_RIGHT | | S/PDIF receiver channel status bits (right). | 0x0000 | R |

S/PDIF Receiver User Data Bits (Left) Register

Address: 0xF630 to 0xF63B (Increments of 0x1), Reset: 0x0000, Name: SPDIF_RX_UD_LEFT_x

These 12 registers store the 192 user data bits decoded from the left channel of the S/PDIF input stream.



[15:0] SPDIF_RX_UD_LEFT (RW)
S/PDIF receiver user data bits (left)

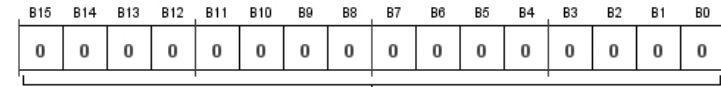
Table 129. Bit Descriptions for SPDIF_RX_UD_LEFT_x

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|------------------|----------|--|--------|--------|
| [15:0] | SPDIF_RX_UD_LEFT | | S/PDIF receiver user data bits (left). | 0x0000 | R |

S/PDIF Receiver User Data Bits (Right) Register

Address: 0xF640 to 0xF64B (Increments of 0x1), Reset: 0x0000, Name: SPDIF_RX_UD_RIGHT_x

These 12 registers store the 192 user data bits decoded from the right channel of the S/PDIF input stream.



[15:0] SPDIF_RX_UD_RIGHT (RW)
S/PDIF receiver user data bits (right)

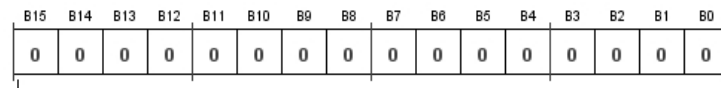
Table 130. Bit Descriptions for SPDIF_RX_UD_RIGHT_x

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-------------------|----------|---|--------|--------|
| [15:0] | SPDIF_RX_UD_RIGHT | | S/PDIF receiver user data bits (right). | 0x0000 | R |

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.



[15:0] SPDIF_RX_VB_LEFT (RW)
S/PDIF receiver validity bits (left)

Table 131. 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.

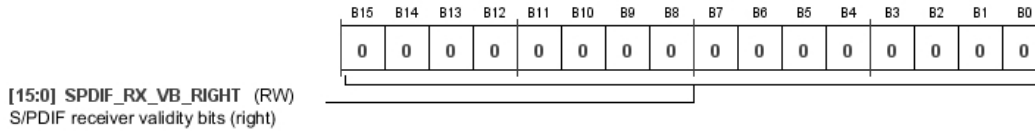


Table 132. Bit Descriptions for SPDIF_RX_VB_RIGHT_x

| 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.

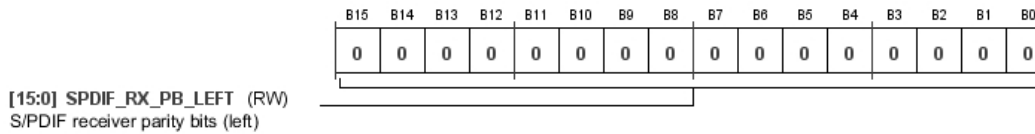


Table 133. Bit Descriptions for SPDIF_RX_PB_LEFT_x

| 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.

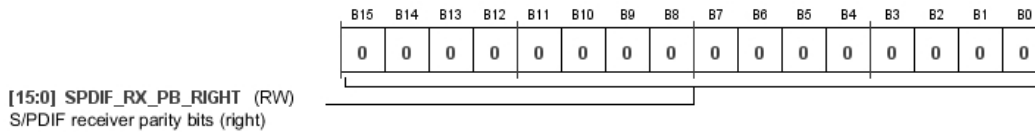


Table 134. Bit Descriptions for SPDIF_RX_PB_RIGHT_x

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

S/PDIF Transmitter Enable Register

Address: 0xF690, Reset: 0x0000, Name: SPDIF_TX_EN

This register enables or disables the S/PDIF transmitter. When the transmitter is disabled, it outputs a constant stream of zero data. When the S/PDIF transmitter is disabled, it still consumes power. To power down the S/PDIF transmitter for the purpose of power savings, set Register 0xF051 (POWER_ENABLE1), Bit 2 (TX_PWR) = 0b0.

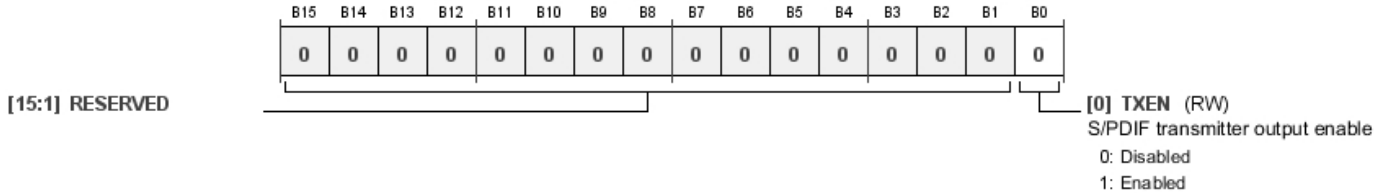


Table 135. Bit Descriptions for SPDIF_TX_EN

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|----------|----------|--|-------|--------|
| [15:1] | RESERVED | | | 0x0 | RW |
| 0 | TXEN | 0 1 | S/PDIF transmitter output enable. Disabled Enabled | 0x0 | RW |

S/PDIF Transmitter Control Register

Address: 0xF691, Reset: 0x0000, Name: SPDIF_TX_CTRL

This register controls the length of the audio data-words output by the S/PDIF transmitter. The maximum word length is 24 bits. If a shorter word length is selected using Bits[1:0] (TX_LENGTHCTRL), the extraneous bits are truncated, starting with the least significant bit. If Bits[1:0] (TX_LENGTHCTRL) = 0b11, the decoded channel status bits on the input stream of the S/PDIF receiver automatically set the word length on the S/PDIF transmitter.

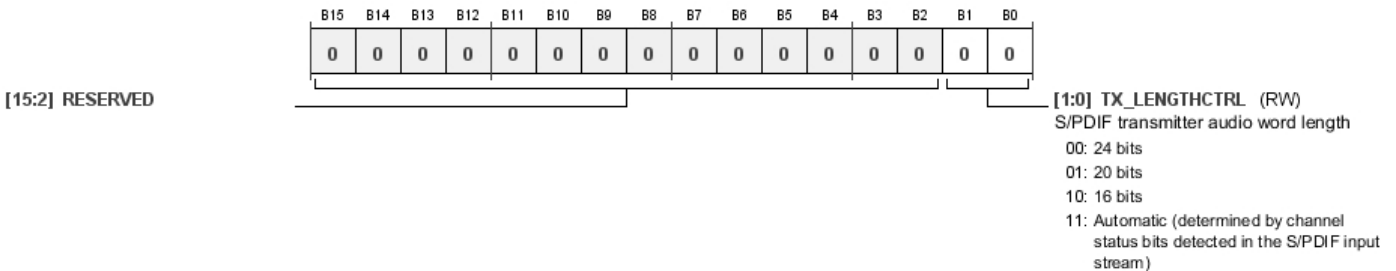


Table 136. Bit Descriptions for SPDIF_TX_CTRL

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|---------------|----------------------|---|-------|--------|
| [15:2] | RESERVED | | | 0x0 | RW |
| [1:0] | TX_LENGTHCTRL | 00 01 10 11 | S/PDIF transmitter audio word length. 24 bits 20 bits 16 bits Automatic (determined by channel status bits detected in the S/PDIF input stream) | 0x0 | RW |

S/PDIF Transmitter Auxiliary Bits Source Select Register

Address: 0xF69F, Reset: 0x0000, Name: SPDIF_TX_AUXBIT_SOURCE

This register configures whether the encoded nonaudio data bits in the output data stream of the S/PDIF transmitter are copied directly from the S/PDIF receiver or set manually using the corresponding control registers. If the data is configured manually, all channel status, parity, user data, and validity bits can be manually set using the following registers: SPDIF_TX_CS_LEFT_x, SPDIF_TX_CS_RIGHT_x, SPDIF_TX_UD_LEFT_x, SPDIF_TX_UD_RIGHT_x, SPDIF_TX_VB_LEFT_x, SPDIF_TX_VB_RIGHT_x, SPDIF_TX_PB_LEFT_x, and SPDIF_TX_PB_RIGHT_x.

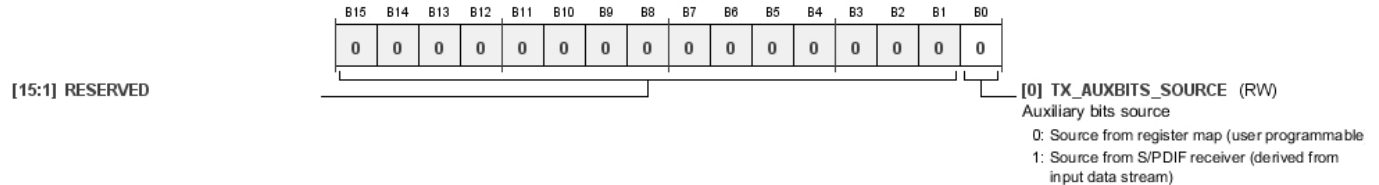


Table 137. Bit Descriptions for SPDIF_TX_AUXBIT_SOURCE

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-------------------|----------|--|-------|--------|
| [15:1] | RESERVED | | | 0x0 | RW |
| 0 | TX_AUXBITS_SOURCE | 0 1 | Auxiliary bits source. 0 Source from register map (user programmable) 1 Source from S/PDIF receiver (derived from input data stream) | 0x0 | RW |

S/PDIF Transmitter Channel Status Bits (Left) Register

Address: 0xF6A0 to 0xF6AB (Increments of 0x1), Reset: 0x0000, Name: SPDIF_TX_CS_LEFT_x

These 12 registers allow the 192 channel status bits encoded on the left channel of the output data stream of the S/PDIF transmitter to be manually configured. For these bits to be output properly on the S/PDIF transmitter, Register 0xF69F (SPDIF_TX_AUXBIT_SOURCE), Bit 0 (TX_AUXBITS_SOURCE), must be set to 0b0.

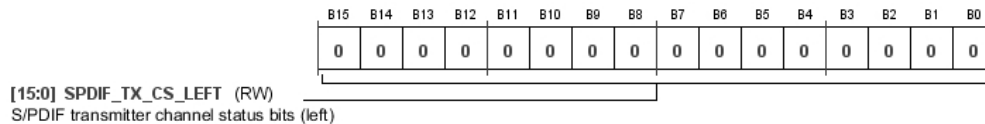


Table 138. Bit Descriptions for SPDIF_TX_CS_LEFT_x

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|------------------|----------|--|--------|--------|
| [15:0] | SPDIF_TX_CS_LEFT | | S/PDIF transmitter channel status bits (left). | 0x0000 | RW |

S/PDIF Transmitter Channel Status Bits (Right) Register

Address: 0xF6B0 to 0xF6BB (Increments of 0x1), Reset: 0x0000, Name: SPDIF_TX_CS_RIGHT_x

These 12 registers allow the 192 channel status bits encoded on the right channel of the output data stream of the S/PDIF transmitter to be manually configured. For these bits to be output properly on the S/PDIF transmitter, Register 0xF69F (SPDIF_TX_AUXBIT_SOURCE), Bit 0 (TX_AUXBITS_SOURCE), must be set to 0b0.

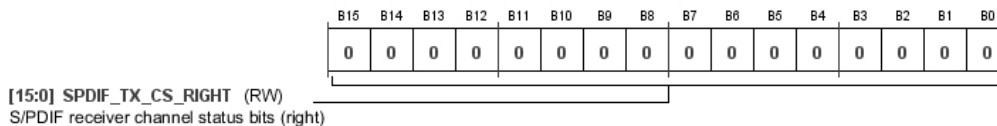


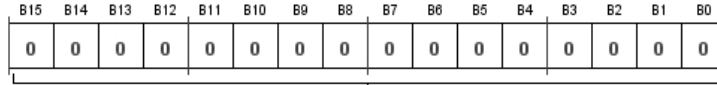
Table 139. Bit Descriptions for SPDIF_TX_CS_RIGHT_x

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-------------------|----------|--|--------|--------|
| [15:0] | SPDIF_TX_CS_RIGHT | | S/PDIF receiver channel status bits (right). | 0x0000 | RW |

S/PDIF Transmitter User Data Bits (Left) Register

Address: 0xF6C0 to 0xF6CB (Increments of 0x1), Reset: 0x0000, Name: SPDIF_TX_UD_LEFT_x

These 12 registers allow the 192 user data bits encoded on the left channel of the output data stream of the S/PDIF transmitter to be manually configured. For these bits to be output properly on the S/PDIF transmitter, Register 0xF69F (SPDIF_TX_AUXBIT_SOURCE), Bit 0 (TX_AUXBITS_SOURCE), must be set to 0b0.



[15:0] SPDIF_TX_UD_LEFT (RW)
S/PDIF transmitter user data bits (left)

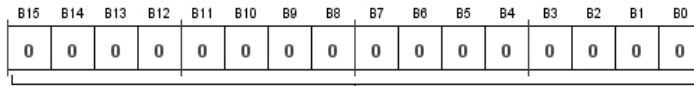
Table 140. Bit Descriptions for SPDIF_TX_UD_LEFT_x

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|------------------|----------|---|--------|--------|
| [15:0] | SPDIF_TX_UD_LEFT | | S/PDIF transmitter user data bits (left). | 0x0000 | RW |

S/PDIF Transmitter User Data Bits (Right) Register

Address: 0xF6D0 to 0xF6DB (Increments of 0x1), Reset: 0x0000, Name: SPDIF_TX_UD_RIGHT_x

These 12 registers allow the 192 user data bits encoded on the right channel of the output data stream of the S/PDIF transmitter to be manually configured. For these bits to be output properly on the S/PDIF transmitter, Register 0xF69F (SPDIF_TX_AUXBIT_SOURCE), Bit 0 (TX_AUXBITS_SOURCE), must be set to 0b0.



[15:0] SPDIF_TX_UD_RIGHT (RW)
S/PDIF transmitter user data bits (right)

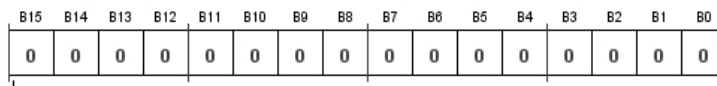
Table 141. Bit Descriptions for SPDIF_TX_UD_RIGHT_x

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-------------------|----------|--|--------|--------|
| [15:0] | SPDIF_TX_UD_RIGHT | | S/PDIF transmitter user data bits (right). | 0x0000 | RW |

S/PDIF Transmitter Validity Bits (Left) Register

Address: 0xF6E0 to 0xF6EB (Increments of 0x1), Reset: 0x0000, Name: SPDIF_TX_VB_LEFT_x

These 12 registers allow the 192 validity bits encoded on the left channel of the output data stream of the S/PDIF transmitter to be manually configured. For these bits to be output properly on the S/PDIF transmitter, Register 0xF69F (SPDIF_TX_AUXBIT_SOURCE), Bit 0 (TX_AUXBITS_SOURCE), must be set to 0b0.



[15:0] SPDIF_TX_VB_LEFT (RW)
S/PDIF transmitter validity bits (left)

Table 142. Bit Descriptions for SPDIF_TX_VB_LEFT_x

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|------------------|----------|--|--------|--------|
| [15:0] | SPDIF_TX_VB_LEFT | | S/PDIF transmitter validity bits (left). | 0x0000 | RW |

S/PDIF Transmitter Validity Bits (Right) Register

Address: 0xF6F0 to 0xF6FB (Increments of 0x1), Reset: 0x0000, Name: SPDIF_TX_VB_RIGHT_x

These 12 registers allow the 192 validity bits encoded on the right channel of the output data stream of the S/PDIF transmitter to be manually configured. For these bits to be output properly on the S/PDIF transmitter, Register 0xF69F (SPDIF_TX_AUXBIT_SOURCE), Bit 0 (TX_AUXBITS_SOURCE), must be set to 0b0.

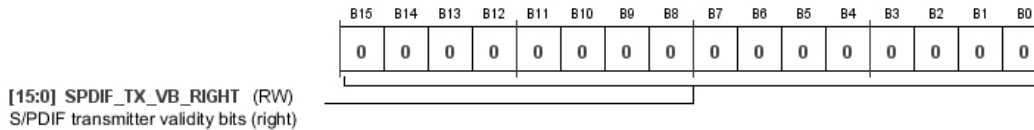


Table 143. Bit Descriptions for SPDIF_TX_VB_RIGHT_x

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-------------------|----------|---|--------|--------|
| [15:0] | SPDIF_TX_VB_RIGHT | | S/PDIF transmitter validity bits (right). | 0x0000 | RW |

S/PDIF Transmitter Parity Bits (Left) Register

Address: 0xF700 to Address 0xF70B (Increments of 0x1), Reset: 0x0000, Name: SPDIF_TX_PB_LEFT_x

These 12 registers allow the 192 parity bits encoded on the left channel of the output data stream of the S/PDIF transmitter to be manually configured. For these bits to be output properly on the S/PDIF transmitter, Register 0xF69F (SPDIF_TX_AUXBIT_SOURCE), Bit 0 (TX_AUXBITS_SOURCE), must be set to 0b0.

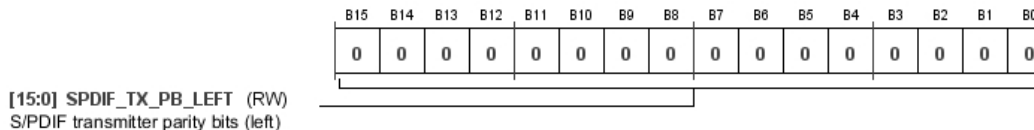


Table 144. Bit Descriptions for SPDIF_TX_PB_LEFT_x

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|------------------|----------|--|--------|--------|
| [15:0] | SPDIF_TX_PB_LEFT | | S/PDIF transmitter parity bits (left). | 0x0000 | RW |

S/PDIF Transmitter Parity Bits (Right) Register

Address: 0xF710 to Address 0xF71B (Increments of 0x1), Reset: 0x0000, Name: SPDIF_TX_PB_RIGHT_x

These 12 registers allow the 192 parity bits encoded on the right channel of the output data stream of the S/PDIF transmitter to be manually configured. For these bits to be output properly on the S/PDIF transmitter, Register 0xF69F (SPDIF_TX_AUXBIT_SOURCE), Bit 0 (TX_AUXBITS_SOURCE), must be set to 0b0.

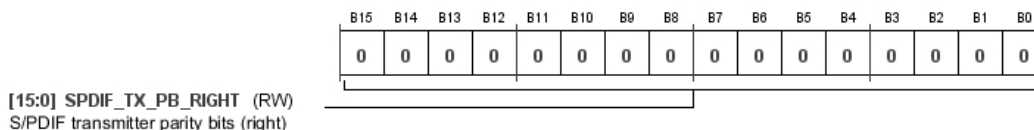


Table 145. Bit Descriptions for SPDIF_TX_PB_RIGHT_x

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-------------------|----------|---|--------|--------|
| [15:0] | SPDIF_TX_PB_RIGHT | | S/PDIF transmitter parity bits (right). | 0x0000 | RW |

HARDWARE INTERFACING REGISTERS

BCLK Input Pins Drive Strength and Slew Rate Register

Address: 0xF780 to 0xF783 (Increments of 0x1), Reset: 0x0018, Name: BCLK_INx_PIN

These registers configure the drive strength, slew rate, and pull resistors for the BCLK_INx pins. Register 0xF780 corresponds to BCLK_IN0, Register 0xF781 corresponds to BCLK_IN1, Register 0xF782 corresponds to BCLK_IN2, and Register 0xF783 corresponds to BCLK_IN3.

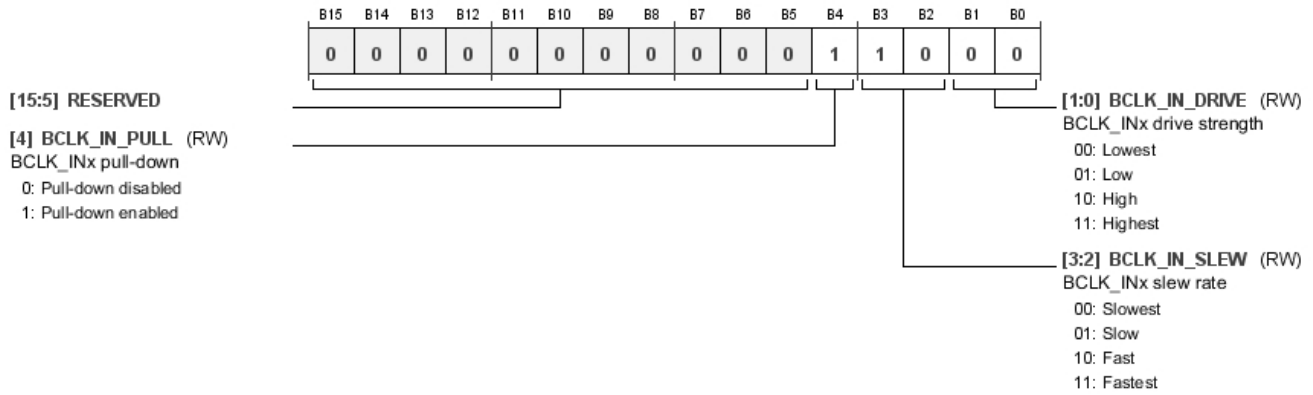


Table 146. Bit Descriptions for BCLK_INx_PIN

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

BCLK Output Pins Drive Strength and Slew Rate Register

Address: 0xF784 to 0xF787 (Increments of 0x1), Reset: 0x0018, Name: BCLK_OUTx_PIN

These registers configure the drive strength, slew rate, and pull resistors for the BCLK_OUTx pins. Register 0xF784 corresponds to BCLK_OUT0, Register 0xF785 corresponds to BCLK_OUT1, Register 0xF786 corresponds to BCLK_OUT2, and Register 0xF787 corresponds to BCLK_OUT3.

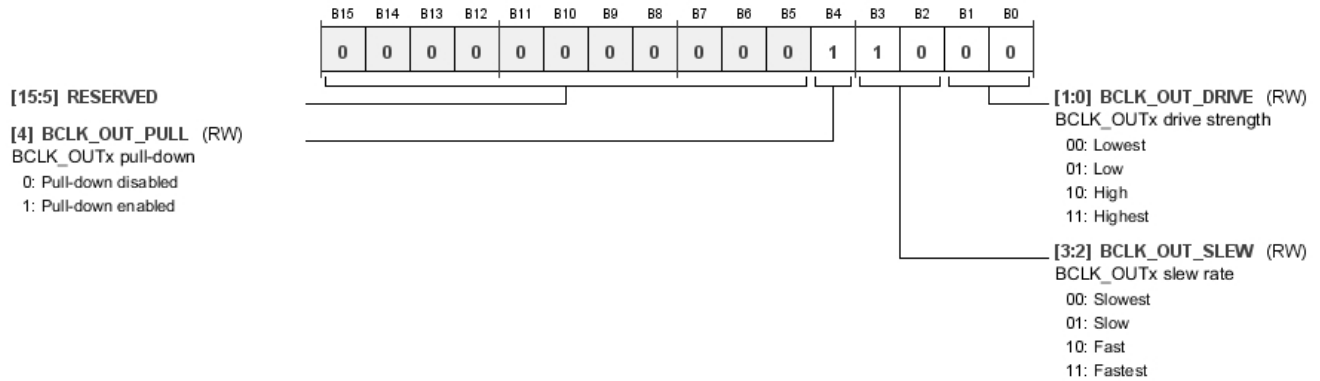


Table 147. Bit Descriptions for BCLK_OUTx_PIN

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

LRCLK Input Pins Drive Strength and Slew Rate Register

Address: 0xF788 to 0xF78B (Increments of 0x1), Reset: 0x0018, Name: LRCLK_INx_PIN

These registers configure the drive strength, slew rate, and pull resistors for the LRCLK_INx pins. Register 0xF788 corresponds to LRCLK_IN0/MP10, Register 0xF789 corresponds to LRCLK_IN1/MP11, Register 0xF78A corresponds to LRCLK_IN2/MP12, and Register 0xF78B corresponds to LRCLK_IN3/MP13.

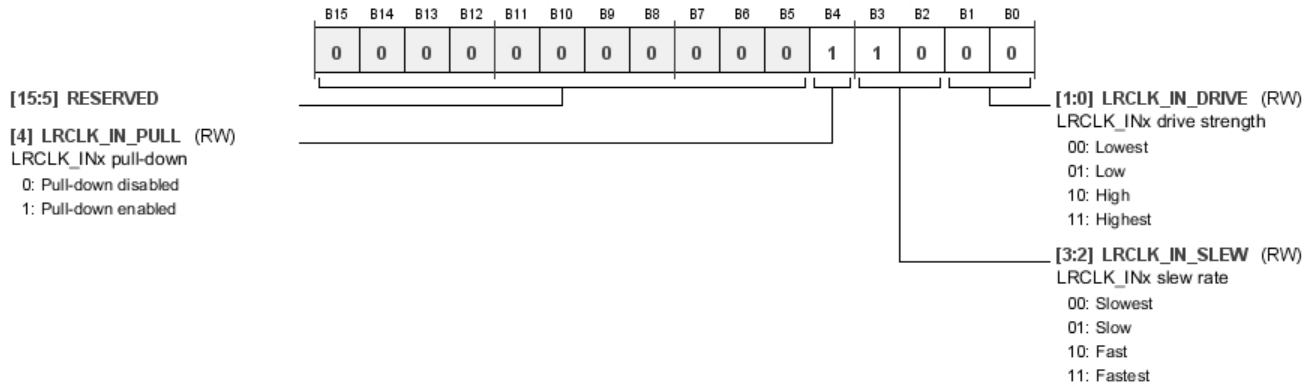


Table 148. Bit Descriptions for LRCLK_INx_PIN

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

LRCLK Output Pins Drive Strength and Slew Rate Register

Address: 0xF78C to 0xF78F (Increments of 0x1), Reset: 0x0018, Name: LRCLK_OUTx_PIN

These registers configure the drive strength, slew rate, and pull resistors for the LRCLK_OUTx pins. Register 0xF78C corresponds to LRCLK_OUT0/MP4, Register 0xF78D corresponds to LRCLK_OUT1/MP5, Register 0xF78E corresponds to LRCLK_OUT2/MP8, and Register 0xF78F corresponds to LRCLK_OUT3/MP9.

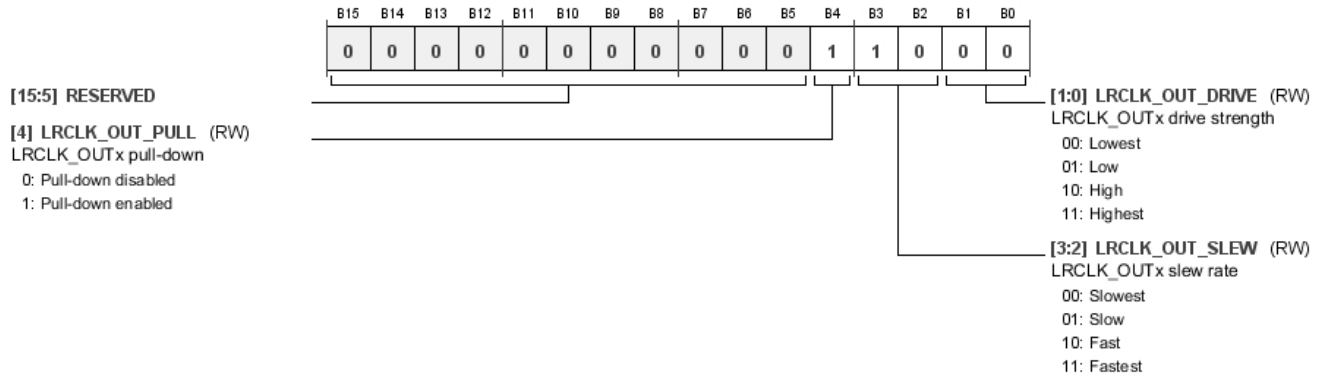


Table 149. Bit Descriptions for LRCLK_OUTx_PIN

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

SDATA Input Pins Drive Strength and Slew Rate Register

Address: 0xF790 to 0xF793 (Increments of 0x1), Reset: 0x0018, Name: SDATA_INx_PIN

These registers configure the drive strength, slew rate, and pull resistors for the SDATA_INx pins. Register 0xF790 corresponds to SDATA_IN0, Register 0xF791 corresponds to SDATA_IN1, Register 0xF792 corresponds to SDATA_IN2, and Register 0xF793 corresponds to SDATA_IN3.

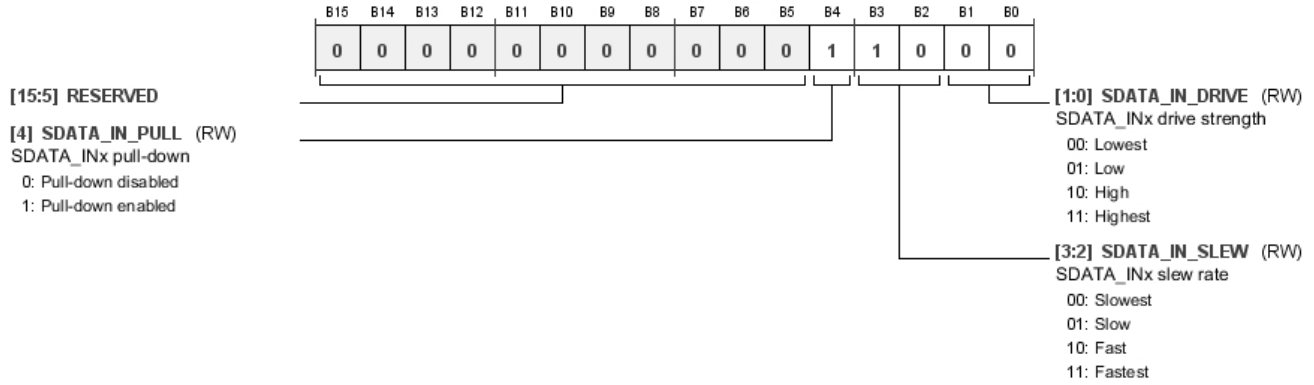


Table 150. Bit Descriptions for SDATA_INx_PIN

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

SDATA Output Pins Drive Strength and Slew Rate Register

Address: 0xF794 to 0xF797 (Increments of 0x1), Reset: 0x0008, Name: SDATA_OUTx_PIN

These registers configure the drive strength, slew rate, and pull resistors for the SDATA_OUTx pins. Register 0xF794 corresponds to SDATA_OUT0, Register 0xF795 corresponds to SDATA_OUT1, Register 0xF796 corresponds to SDATA_OUT2, and Register 0xF797 corresponds to SDATA_OUT3.

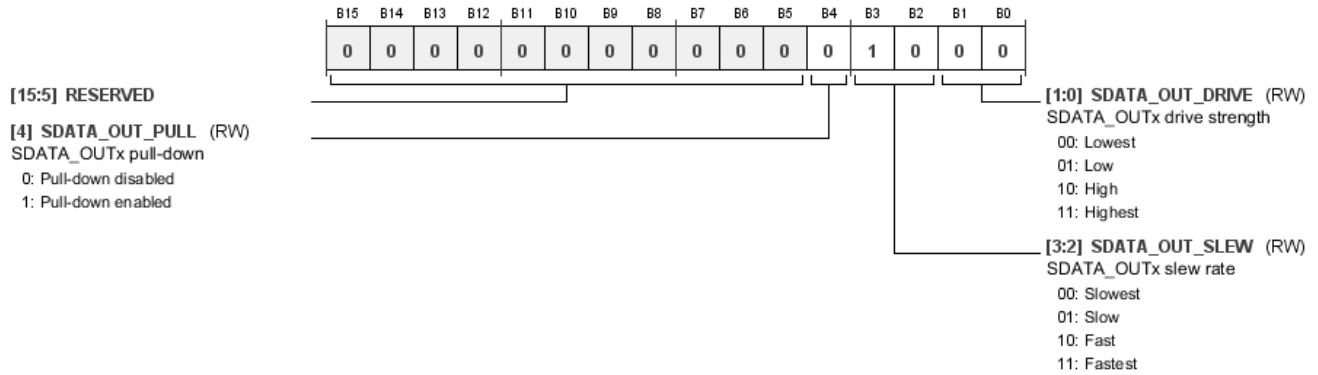


Table 151. Bit Descriptions for SDATA_OUTx_PIN

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

S/PDIF Transmitter Pin Drive Strength and Slew Rate Register

Address: 0xF798, Reset: 0x0008, Name: SPDIF_TX_PIN

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

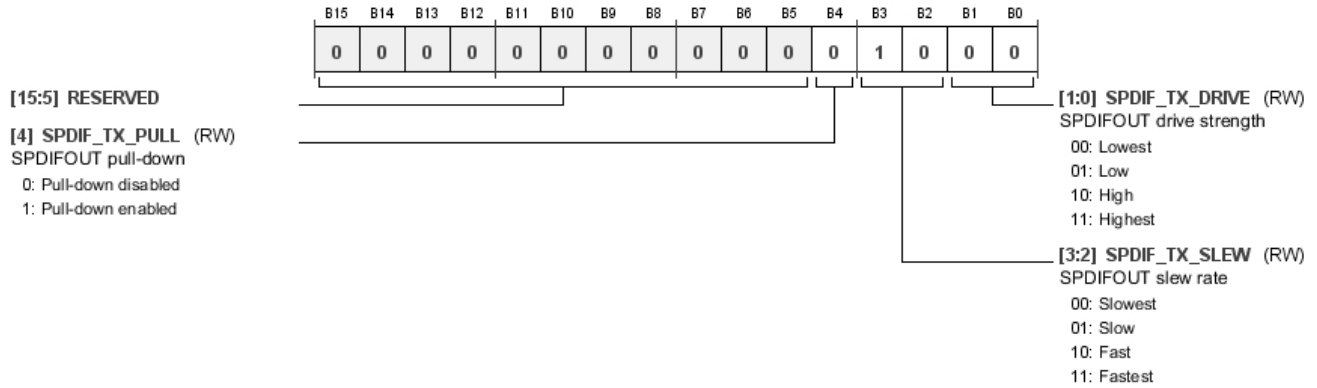


Table 152. Bit Descriptions for SPDIF_TX_PIN

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

SCLK/SCL Pin Drive Strength and Slew Rate Register

Address: 0xF799, Reset: 0x0008, Name: SCLK_SCL_PIN

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

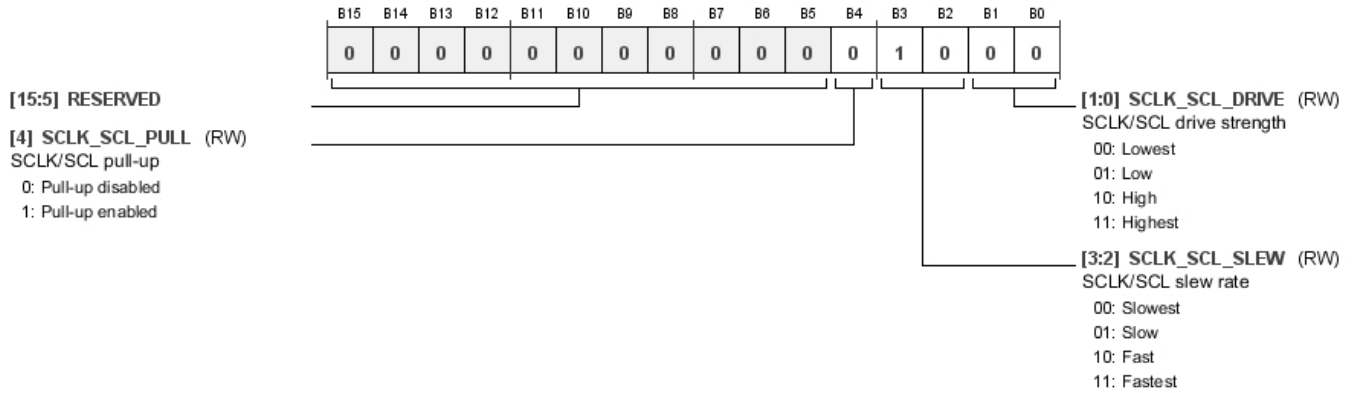


Table 153. Bit Descriptions for SCLK_SCL_PIN

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|----------------|----------------------|--|-------|--------|
| [15:5] | RESERVED | | | 0x0 | RW |
| 4 | SCLK_SCL_PULL | 0 1 | SCLK/SCL pull-up. 0 Pull-up disabled 1 Pull-up enabled | 0x0 | RW |
| [3:2] | SCLK_SCL_SLEW | 00 01 10 11 | SCLK/SCL slew rate. 00 Slowest 01 Slow 10 Fast 11 Fastest | 0x2 | RW |
| [1:0] | SCLK_SCL_DRIVE | 00 01 10 11 | SCLK/SCL drive strength. 00 Lowest 01 Low 10 High 11 Highest | 0x0 | RW |

MISO/SDA Pin Drive Strength and Slew Rate Register

Address: 0xF79A, Reset: 0x0008, Name: MISO_SDA_PIN

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

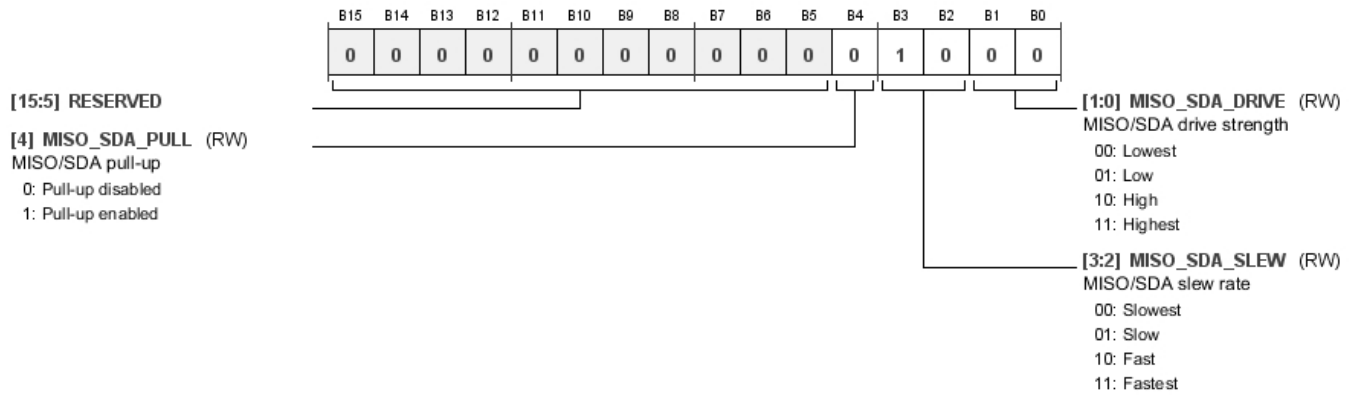


Table 154. Bit Descriptions for MISO_SDA_PIN

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|----------------|----------------------|--|-------|--------|
| [15:5] | RESERVED | | | 0x0 | RW |
| 4 | MISO_SDA_PULL | 0 1 | MISO/SDA pull-up. Pull-up disabled Pull-up enabled | 0x0 | RW |
| [3:2] | MISO_SDA_SLEW | 00 01 10 11 | MISO/SDA slew rate. Slowest Slow Fast Fastest | 0x2 | RW |
| [1:0] | MISO_SDA_DRIVE | 00 01 10 11 | MISO/SDA drive strength. Lowest Low High Highest | 0x0 | RW |

SS/ADDR0 Pin Drive Strength and Slew Rate Register

Address: 0xF79B, Reset: 0x0018, Name: SS_PIN

This register configures the drive strength, slew rate, and pull resistors for the SS/ADDR0 pin.

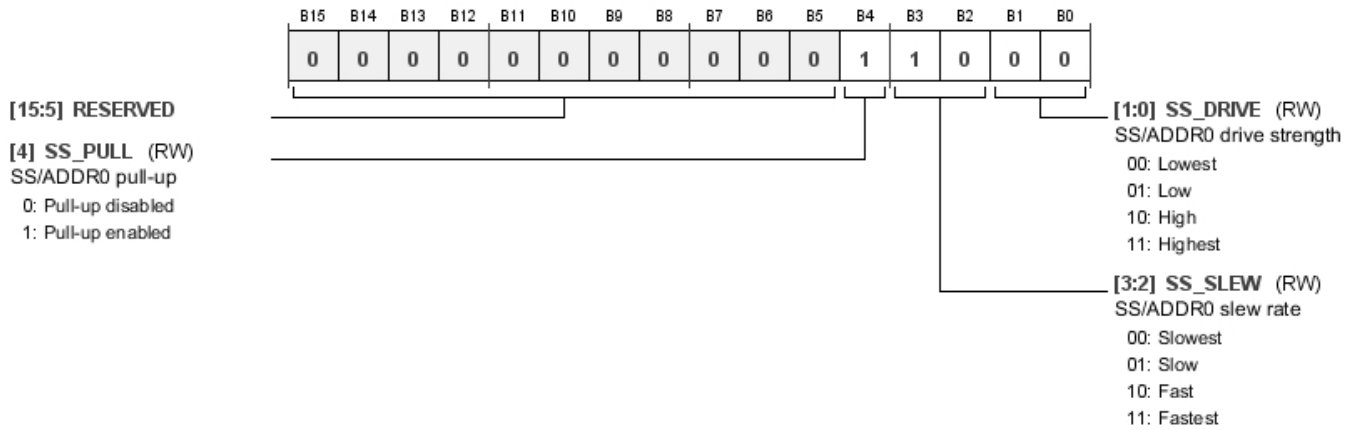


Table 155. Bit Descriptions for SS_PIN

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|----------|----------------------|--|-------|--------|
| [15:5] | RESERVED | | | 0x0 | RW |
| 4 | SS_PULL | 0 1 | SS/ADDR0 pull-up. Pull-up disabled Pull-up enabled | 0x1 | RW |
| [3:2] | SS_SLEW | 00 01 10 11 | SS/ADDR0 slew rate. Slowest Slow Fast Fastest | 0x2 | RW |
| [1:0] | SS_DRIVE | 00 01 10 11 | SS/ADDR0 drive strength. Lowest Low High Highest | 0x0 | RW |

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.

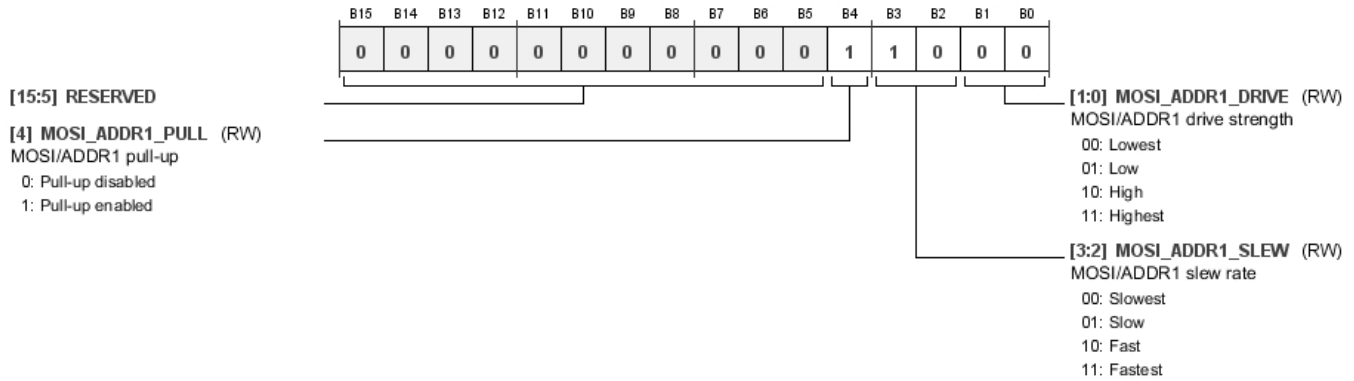


Table 156. Bit Descriptions for MOSI_ADDR1_PIN

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

SCL_M/SCLK_M/MP2 Pin Drive Strength and Slew Rate Register

Address: 0xF79D, Reset: 0x0008, Name: SCLK_SCL_M_PIN

This register configures the drive strength, slew rate, and pull resistors for the SCL_M/SCLK_M/MP2 pin.

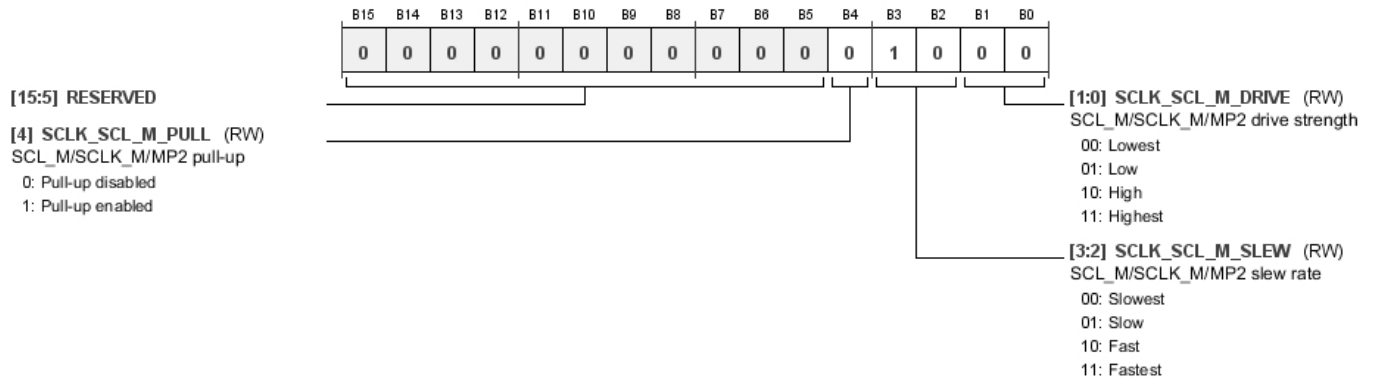


Table 157. Bit Descriptions for SCLK_SCL_M_PIN

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|------------------|----------------------|--|-------|--------|
| [15:5] | RESERVED | | | 0x0 | RW |
| 4 | SCLK_SCL_M_PULL | 0 1 | SCL_M/SCLK_M/MP2 pull-up. Pull-up disabled Pull-up enabled | 0x0 | RW |
| [3:2] | SCLK_SCL_M_SLEW | 00 01 10 11 | SCL_M/SCLK_M/MP2 slew rate. Slowest Slow Fast Fastest | 0x2 | RW |
| [1:0] | SCLK_SCL_M_DRIVE | 00 01 10 11 | SCL_M/SCLK_M/MP2 drive strength. Lowest Low High Highest | 0x0 | RW |

SDA_M/MISO_M/MP3 Pin Drive Strength and Slew Rate Register

Address: 0xF79E, Reset: 0x0008, Name: MISO_SDA_M_PIN

This register configures the drive strength, slew rate, and pull resistors for the SDA_M/MISO_M/MP3 pin.

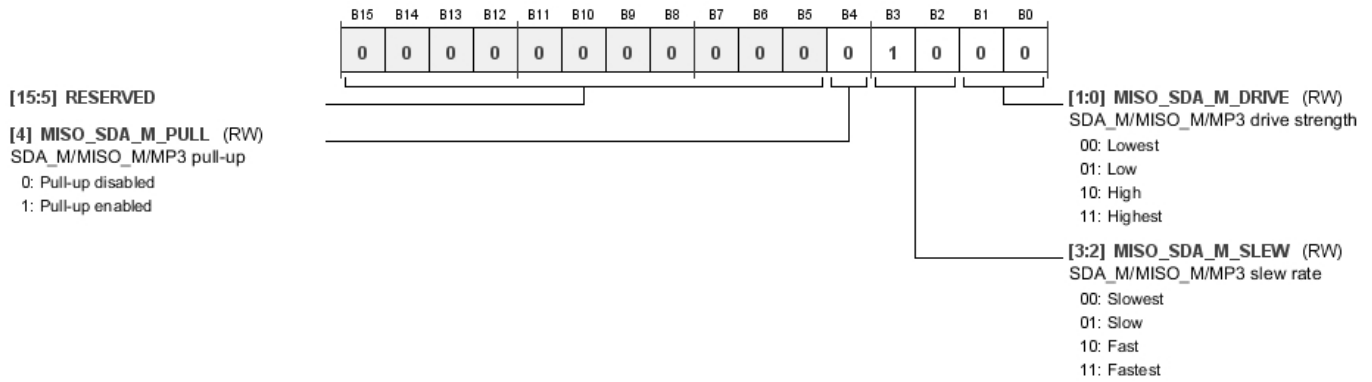


Table 158. Bit Descriptions for MISO_SDA_M_PIN

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|------------------|----------------------|--|-------|--------|
| [15:5] | RESERVED | | | 0x0 | RW |
| 4 | MISO_SDA_M_PULL | 0 1 | SDA_M/MISO_M/MP3 pull-up. 0 Pull-up disabled 1 Pull-up enabled | 0x0 | RW |
| [3:2] | MISO_SDA_M_SLEW | 00 01 10 11 | SDA_M/MISO_M/MP3 slew rate. 00 Slowest 01 Slow 10 Fast 11 Fastest | 0x2 | RW |
| [1:0] | MISO_SDA_M_DRIVE | 00 01 10 11 | SDA_M/MISO_M/MP3 drive strength. 00 Lowest 01 Low 10 High 11 Highest | 0x0 | RW |

SS_M/MP0 Pin Drive Strength and Slew Rate Register

Address: 0xF79F, Reset: 0x0018, Name: SS_M_PIN

This register configures the drive strength, slew rate, and pull resistors for the SS_M/MP0 pin.

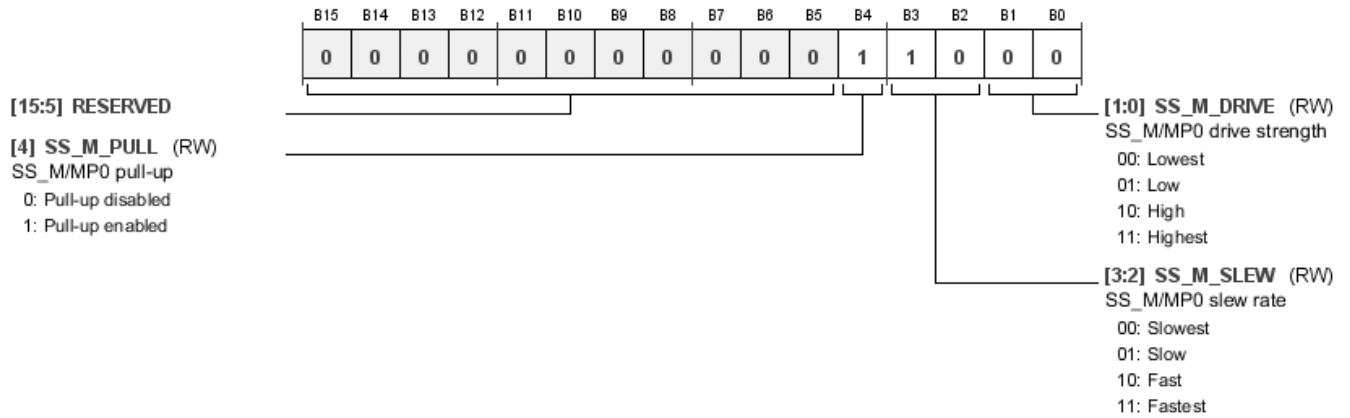


Table 159. Bit Descriptions for SS_M_PIN

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|------------|----------------------|--|-------|--------|
| [15:5] | RESERVED | | | 0x0 | RW |
| 4 | SS_M_PULL | 0 1 | SS_M/MP0 pull-up. Pull-up disabled Pull-up enabled | 0x1 | RW |
| [3:2] | SS_M_SLEW | 00 01 10 11 | SS_M/MP0 slew rate. Slowest Slow Fast Fastest | 0x2 | RW |
| [1:0] | SS_M_DRIVE | 00 01 10 11 | SS_M/MP0 drive strength. Lowest Low High Highest | 0x0 | RW |

MOSI_M/MP1 Pin Drive Strength and Slew Rate Register

Address: 0xF7A0, Reset: 0x0018, Name: MOSI_M_PIN

This register configures the drive strength, slew rate, and pull resistors for the MOSI_M/MP1 pin.

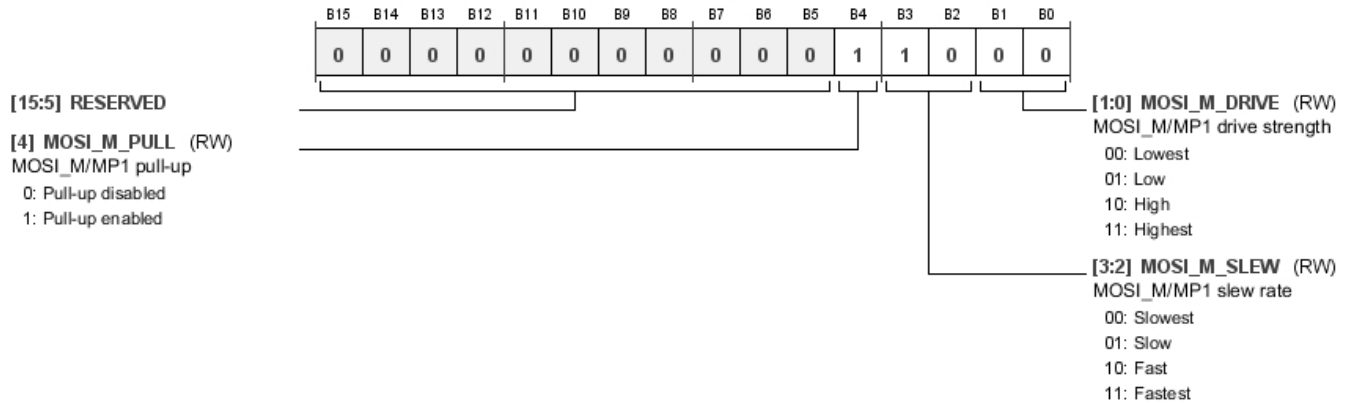


Table 160. Bit Descriptions for MOSI_M_PIN

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|--------------|----------------------|--|-------|--------|
| [15:5] | RESERVED | | | 0x0 | RW |
| 4 | MOSI_M_PULL | 0 1 | MOSI_M/MP1 pull-up. Pull-up disabled Pull-up enabled | 0x1 | RW |
| [3:2] | MOSI_M_SLEW | 00 01 10 11 | MOSI_M/MP1 slew rate. Slowest Slow Fast Fastest | 0x2 | RW |
| [1:0] | MOSI_M_DRIVE | 00 01 10 11 | MOSI_M/MP1 drive strength. Lowest Low High Highest | 0x0 | RW |

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.

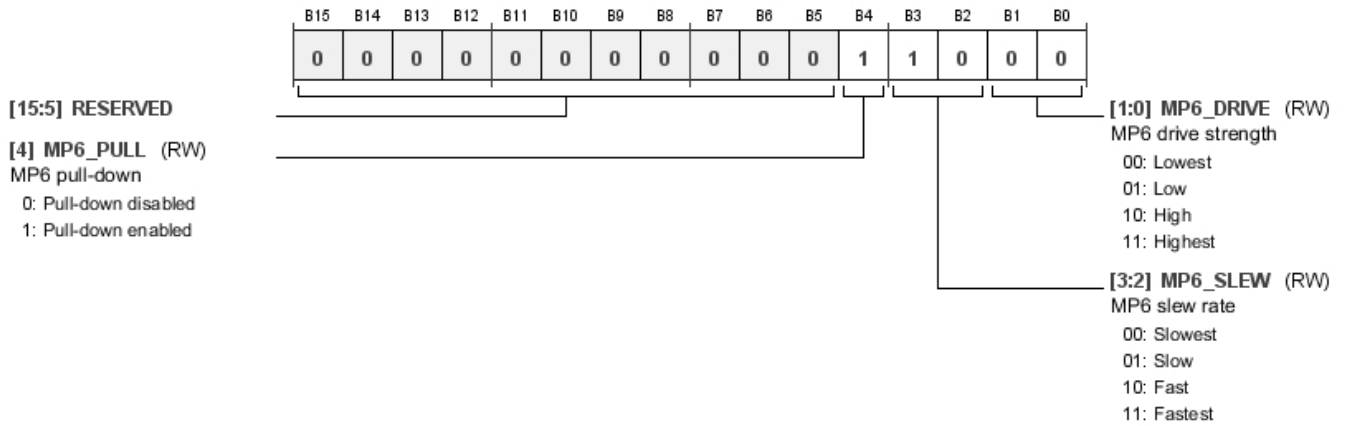


Table 161. Bit Descriptions for MP6_PIN

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

MP7 Pin Drive Strength and Slew Rate Register

Address: 0xF7A2, Reset: 0x0018, Name: MP7_PIN

This register configures the drive strength, slew rate, and pull resistors for the MP7 pin.

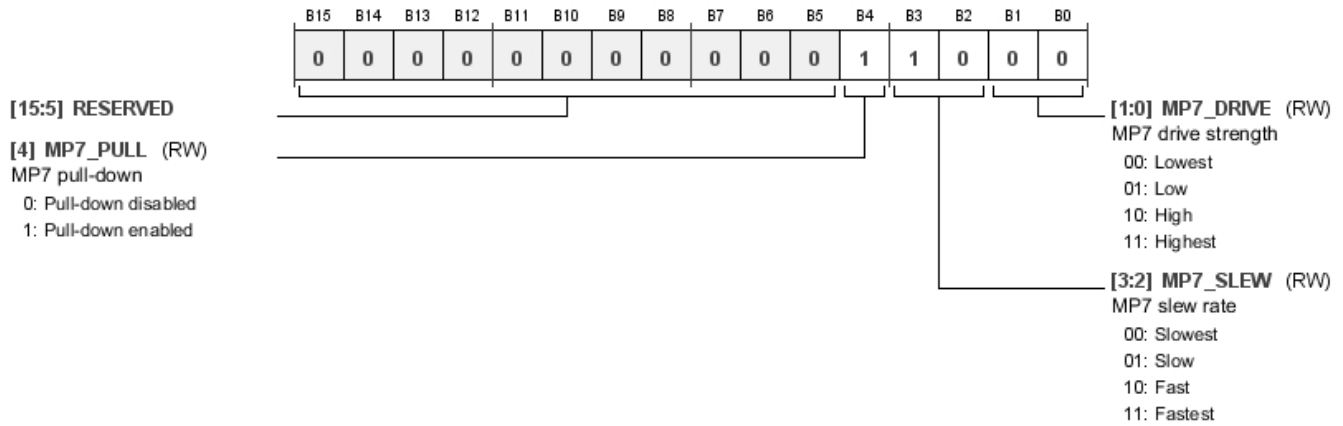


Table 162. Bit Descriptions for MP7_PIN

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|-----------|----------------------|---|-------|--------|
| [15:5] | RESERVED | | | 0x0 | RW |
| 4 | MP7_PULL | 0 1 | MP7 pull-down. 0 Pull-down disabled 1 Pull-down enabled | 0x1 | RW |
| [3:2] | MP7_SLEW | 00 01 10 11 | MP7 slew rate. 00 Slowest 01 Slow 10 Fast 11 Fastest | 0x2 | RW |
| [1:0] | MP7_DRIVE | 00 01 10 11 | MP7 drive strength. 00 Lowest 01 Low 10 High 11 Highest | 0x0 | RW |

CLKOUT Pin Drive Strength and Slew Rate Register

Address: 0xF7A3, Reset: 0x0008, Name: CLKOUT_PIN

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

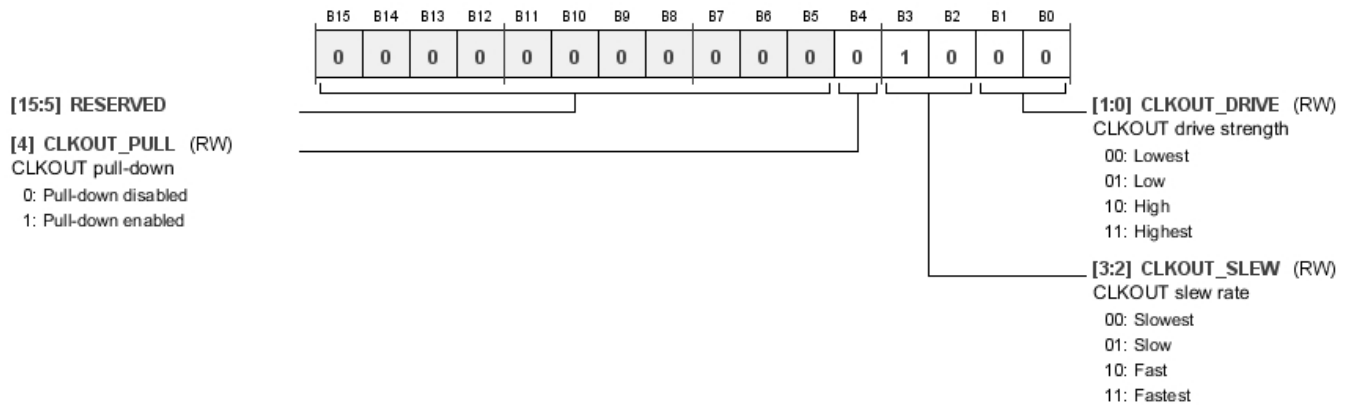


Table 163. Bit Descriptions for CLKOUT_PIN

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

SOFT RESET REGISTER

Address: 0xF890, Reset: 0x0001, Name: SOFT_RESET

SOFT_RESET provides the capability to reset all control registers in the device or put it into a state similar to a hardware reset, where the RESET pin is pulled low to ground. All control registers are reset to their default values, except for the PLL registers: Register 0xF000 (PLL_CTRL0), Register 0xF001 (PLL_CTRL1), Register 0xF002 (PLL_CLK_SRC), Register 0xF003 (PLL_ENABLE), Register 0xF004 (PLL_LOCK), Register 0xF005 (MCLK_OUT), and Register 0xF006 (PLL_WATCHDOG), as well as registers related to the panic manager. The I²C and SPI slave ports remain operational, and the user can write new values to the PLL registers while the soft reset is active. If SPI slave mode is enabled, the device remains in SPI slave mode during and after the soft reset state. To reset the device to I²C slave mode, the device must undergo a hardware reset by pulling the RESET pin low to ground. Bit 0 (SOFT_RESET) is active low, meaning that setting it to 0b1 enables normal operation and setting it to 0b0 enables the soft reset state.

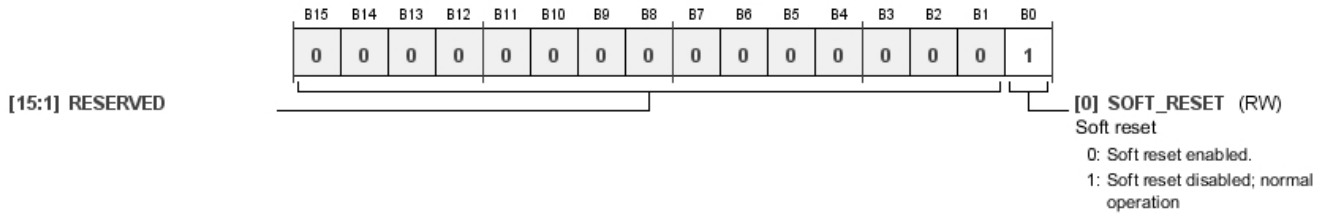


Table 164. Bit Descriptions for SOFT_RESET

| Bits | Bit Name | Settings | Description | Reset | Access |
|--------|------------|----------|--|-------|--------|
| [15:1] | RESERVED | | | 0x0 | RW |
| 0 | SOFT_RESET | 0 1 | Soft reset. Soft reset enabled Soft reset disabled; normal operation | 0x1 | RW |

APPLICATIONS INFORMATION

PCB DESIGN CONSIDERATIONS

A solid ground plane is a necessity for maintaining signal integrity and minimizing EMI radiation. If the PCB has two ground planes, they can be stitched together using vias that are spread evenly throughout the board.

Power Supply Bypass Capacitors

Bypass each power supply pin to its nearest appropriate ground pin with a single 100 nF capacitor and, optionally, with an additional 10 nF capacitor in parallel. Make the connections to each side of the capacitor as short as possible, and keep the trace on a single layer with no vias. For maximum effectiveness, place the capacitor either equidistant from the power and ground pins or, when equidistant placement is not possible, slightly nearer to the power pin (see Figure 80). Establish the thermal connections to the planes on the far side of the capacitor.

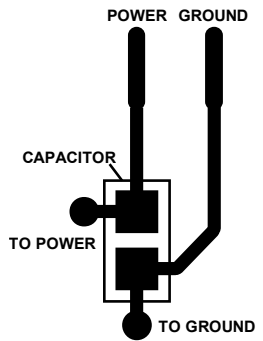


Figure 80. Recommended Power Supply Bypass Capacitor Layout

Typically, a single 100 nF capacitor for each power ground pin pair is sufficient. However, if there is excessive high frequency noise in the system, use an additional 10 nF capacitor in parallel (see Figure 81). In that case, place the 10 nF capacitor between the ADAU1452 and the 100 nF capacitor, and establish the thermal connections on the far side of the 100 nF capacitor.

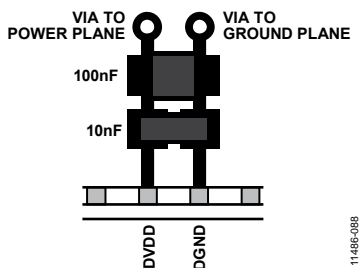


Figure 81. Layout for Multiple Power Supply Bypass Capacitors

To provide a current reservoir in case of sudden current spikes, use a 10 μF capacitor for each named supply (DVDD, AVDD, PVDD, and IOVDD) as shown in Figure 82.

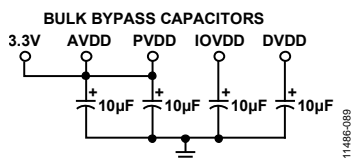


Figure 82. Bulk Capacitor Schematic

Parts Placement

Place all 100 nF bypass capacitors, which are recommended for every analog, digital, and PLL power ground pair, as near as possible to the ADAU1452. Bypass each of the AVDD, DVDD, PVDD, and IOVDD supply signals on the board with an additional single bulk capacitor (10 μF to 47 μF).

Keep all traces in the crystal resonator circuit (see Figure 14) as short as possible to minimize stray capacitance. Do not connect any long board traces to the crystal oscillator circuit components because such traces may affect crystal startup and operation.

Grounding

Use a single ground plane in the application layout. Place all components in an analog signal path away from digital signals.

Exposed Pad PCB Design

The ADAU1452 package includes an exposed pad for improved heat dissipation. When designing a board for such a package, give special consideration to the following:

- Place a copper layer, equal in size to the exposed pad, on all layers of the board, from top to bottom. Connect the copper layers to a dedicated copper board layer (see Figure 83).

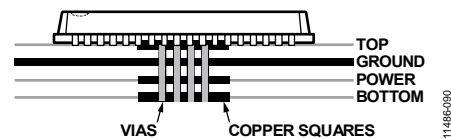


Figure 83. Exposed Pad Layout Example—Side View

- Place vias such that all layers of copper are connected, allowing for efficient heat and energy conductivity. For an example, see Figure 84, which shows 49 vias arranged in a 7 × 7 grid in the pad area.

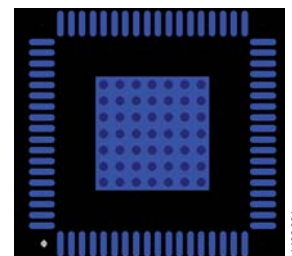


Figure 84. Exposed Pad Layout Example—Top View

PLL Loop Filter

To minimize jitter, connect the single resistor and two capacitors in the PLL loop filter to the PLLFILT and PVDD pins with short traces.

Power Supply Isolation with Ferrite Beads

Ferrite beads can be used for supply isolation. When using ferrite beads, always place the beads outside the local high frequency decoupling capacitors, as shown in Figure 85. If the ferrite beads are placed between the supply pin and the decoupling capacitor, high frequency noise is reflected back into the IC because there is no suitable return path to ground. As a result, EMI increases, creating noisy supplies.

EOS/ESD Protection

Although the ADAU1452 has robust internal protection circuitry against overvoltages and electrostatic discharge, an external transient voltage suppressor (TVS) is recommended for all systems to prevent damage to the IC. For examples, see [Application Note AN-311](#).

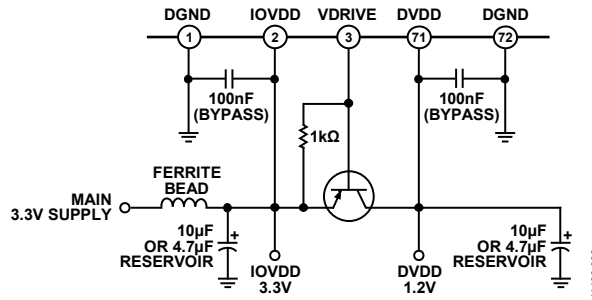


Figure 85. Ferrite Bead Power Supply Isolation Circuit Example

TYPICAL APPLICATIONS BLOCK DIAGRAM

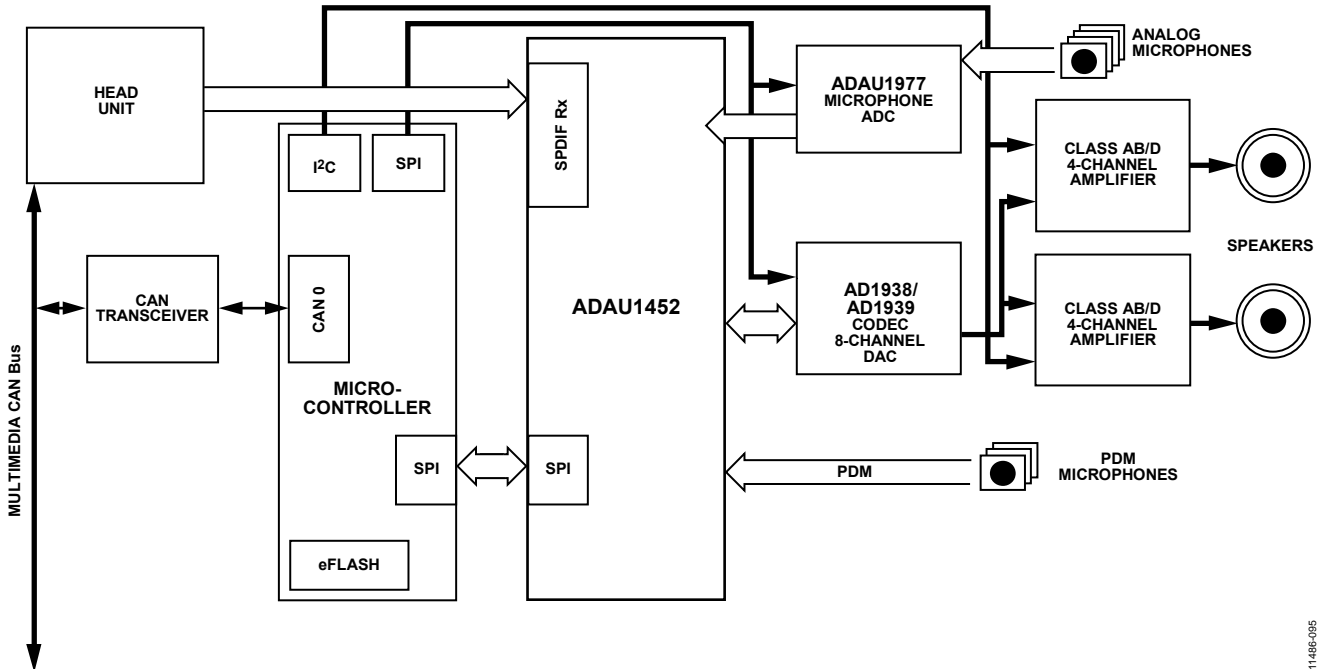


Figure 86. Automotive Infotainment Amplifier Block Diagram

EXAMPLE PCB LAYOUT

Several external components, such as capacitors, resistors, and a transistor, are required for proper operation of the device. An example of the connection and layout of these components is shown in Figure 87. Thick black lines represent traces, gray rectangles represent components, and white circles with a thick black ring represent thermal via connections to power or ground planes. If a 1.2 V supply is available in the system, the transistor circuit (including the associated 1 kΩ resistor) can be removed, and 1.2 V can be connected directly to the DVDD power net, with the VDRIVE pin left floating.

The analog (AVDD), PLL (PVDD), and interface (IOVDD) supply pins each have local 100 nF bypass capacitors to provide high frequency return currents with a short path to ground.

The digital (DVDD) supply pins each have up to three local bypass capacitors, as follows:

- The 10 nF bypass capacitor, placed closest to the pin, acts as a return path for very high frequency currents resulting from the nominal 294 MHz operating frequency of the DSP core.
- The 100 nF bypass capacitor acts as a return path for high frequency currents from the DSP and other digital circuitry.
- The 1 μF bypass capacitor is required to provide a local current supply for sudden spikes in current that occur at the beginning of each audio frame when the DSP core switches from idle mode to operating mode.

Of these three bypass capacitors, the most important is the 100 nF bypass capacitor, which is required for proper power supply bypassing. The 10 nF and 1 μF capacitors can optionally be used to improve the EMI/EMC performance of the system.

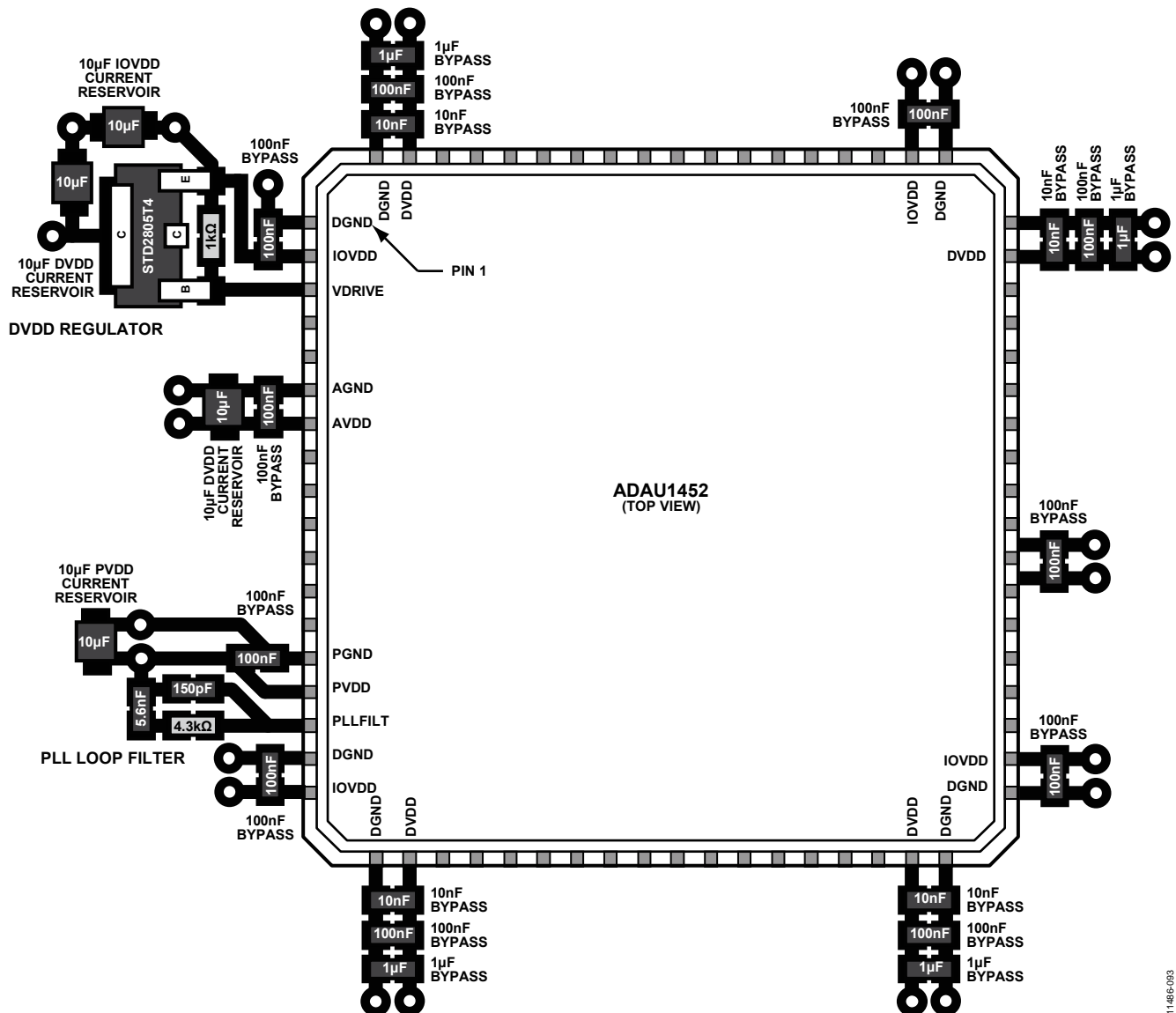


Figure 87. Supporting Component Placement and Layout

PCB MANUFACTURING GUIDELINES

The soldering profile in Figure 88 is recommended for the LFCSP package. Refer to [Application Note AN-772](#) for more information about PCB manufacturing guidelines.

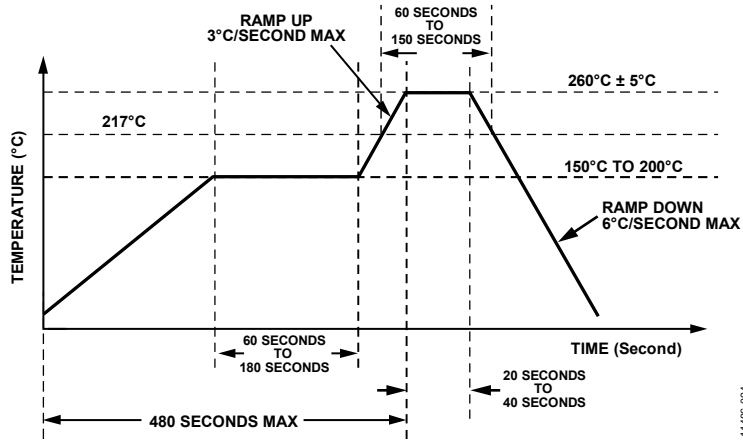


Figure 88. Soldering Profile

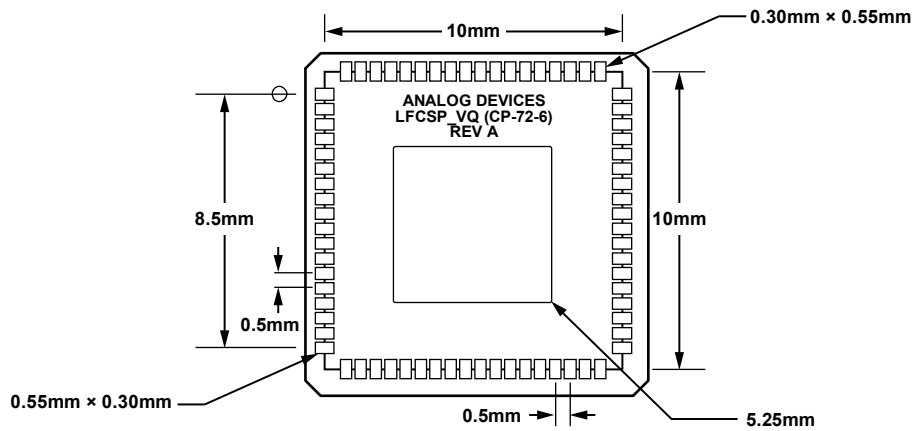
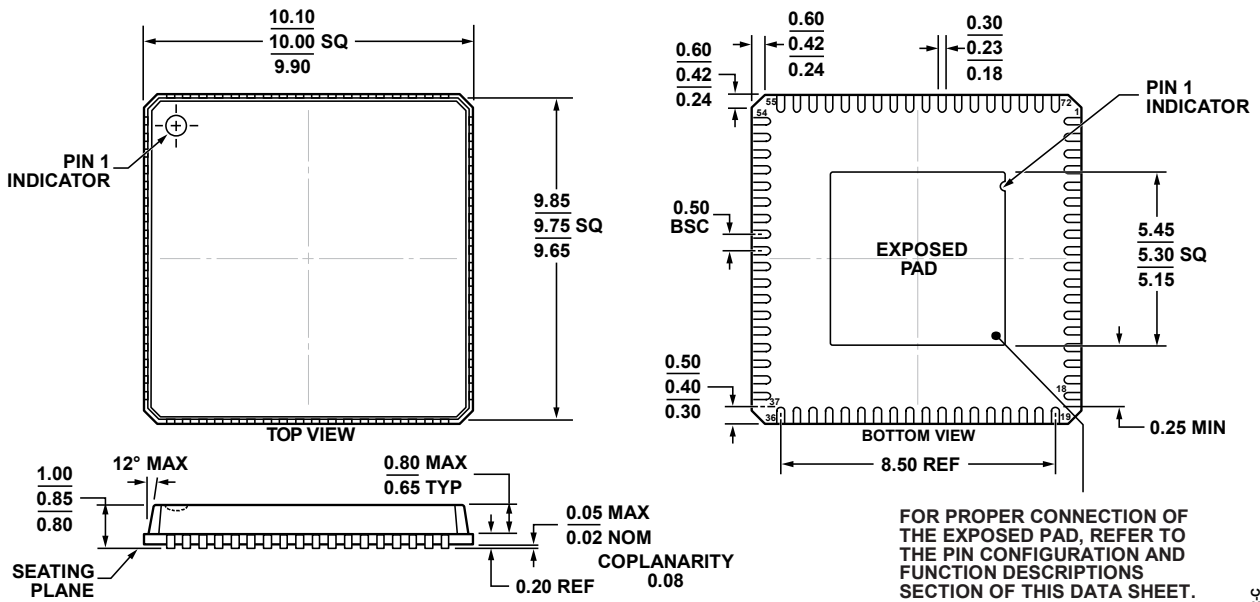


Figure 89. PCB Decal Dimensions

OUTLINE DIMENSIONS



COMPLIANT TO JEDEC STANDARDS MO-220-VNND-4
 Figure 90. 72-Lead Lead Frame Chip Scale Package [LFCSP_VQ]
 10 mm × 10 mm Body, Very Thin Quad
 (CP-72-6)
 Dimensions shown in millimeters

06-25-2012-C

ORDERING GUIDE

| Model ^{1,2} | Temperature Range | Package Description | Package Option |
|----------------------|-------------------|-------------------------------------|----------------|
| ADAU1452WBCPZ | -40°C to +105°C | 72-Lead LFCSP_VQ | CP-72-6 |
| ADAU1452WBCPZ-RL | -40°C to +105°C | 72-Lead LFCSP_VQ, 13" Tape and Reel | CP-72-6 |
| EVAL-ADAU1452MINIZ | | Evaluation Board | |

¹ Z = RoHS compliant part.
² W = Qualified for Automotive Applications.

AUTOMOTIVE PRODUCTS

The ADAU1452W models are available with controlled manufacturing to support the quality and reliability requirements of automotive applications. Note that these automotive models may have specifications that differ from the commercial models; therefore, designers should review the Specifications section of this data sheet carefully. Only the automotive grade products shown are available for use in automotive applications. Contact your local Analog Devices account representative for specific product ordering information and to obtain the specific Automotive Reliability reports for these models.

NOTES

I²C refers to a communications protocol originally developed by Philips Semiconductors (now NXP Semiconductors).