CIRRUS LOGIC®

CS47L15

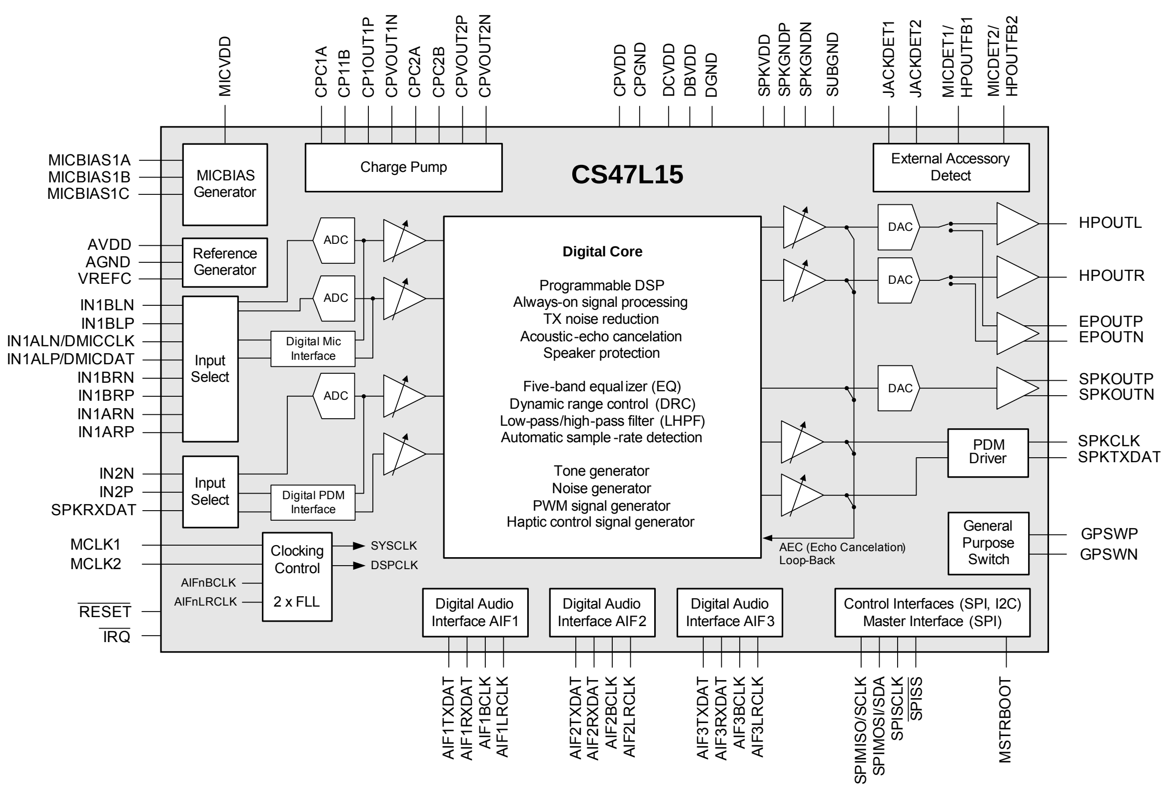
Smart Codec with Low-Power Audio DSP

Features

- 150 MIPS, 150 MMAC audio-signal processor
  - — Low-power, always-on voice trigger capability
  - — Speaker protection algorithm support
  - — Event loggers with time-stamp and interrupt functions
- Programmable wideband audio processing
  - — Transmit-path noise reduction and echo cancelation
- Integrated multichannel 24-bit hi-fi audio hub codec
  - — 98-dB signal-to-noise ratio (SNR) mic input (48 kHz)
  - — 127-dB SNR headphone playback (48 kHz)
  - — Low-power analog input modes
- Up to four analog or four digital microphone (DMIC) inputs
  - — Speaker-monitoring input path (analog or digital)
- Stereo headphone/earpiece/line output driver: 30 mW into 32-Ω load at 0.1% total harmonic distortion + noise (THD+N)
- Earpiece, speaker, and digital (pulse-density modulation, PDM) output interfaces
  - — Two-way stereo PDM interface
- Three full digital-audio interfaces
  - — Standard sample rates from 8 to 192 kHz
  - — Multichannel support on AIF1 and AIF2
- Self-boot capability from external non-volatile memory
- Flexible clocking, derived from MCLK*n* or AIF*n*
- Low-power frequency-locked loops (FLLs) support reference clocks down to 32 kHz
- Advanced accessory detection functions
- Configurable functions on up to 15 general-purpose input/output (GPIO) pins
- Small WLCSP package, 0.4-mm ball array

Applications

- Smartphones, tablets, and wearable technology
  - — Karaoke algorithm support



Preliminary Product Information

This document contains information for a product under development. Cirrus Logic reserves the right to modify this product.

Description

The CS47L15 is a highly integrated, low-power audio hub for smartphones, tablets, and other portable audio devices including wearable technology. It combines an advanced DSP feature set with a flexible, high-performance audio hub codec. The CS47L15 combines a programmable DSP core with a variety of power-efficient fixed-function audio processors. An SPI master interface is provided, for autonomous boot-up and configuration using an external non-volatile memory—enabling the CS47L15 to be used independently of a host processor.

The DSP core supports advanced audio processing functions such as wideband noise reduction, acoustic-echo cancelation (AEC), speech enhancement, karaoke, and many more. Low-power analog and digital interfaces provide flexible support for always-on voice applications and speaker-protection algorithms implemented on the programmable DSP core. The DSP core is integrated within a fully flexible, all-digital mixing and routing engine with sample-rate converters, for wide use-case flexibility. Support for third-party DSP programming provides far-reaching opportunities for product differentiation.

Three digital audio interfaces are provided, each supporting a wide range of standard audio sample rates and serial interface formats. Automatic sample-rate detection enables seamless wideband/narrowband voice-call handover. The DACs and output paths provide full support for high definition audio throughout the entire signal chain.

The stereo headphone driver provides ground-referenced output, with noise levels as low as 0.45 μV_{RMS} for hi-fi quality line or headphone output. The CS47L15 also features a mono bridge-tied load (BTL) earpiece output, mono 2.5-W Class D speaker driver, two channels of stereo PDM output, and an IEC-60958-3–compatible S/PDIF transmitter. A signal generator for controlling haptics devices is included; vibe actuators can connect directly to the Class D speaker output, or via an external driver on the PDM output interface.

The CS47L15 supports up to five analog inputs, and up to four PDM digital inputs. As many as four analog microphone connections can be supported; a separate analog input channel is provided for use in speaker-protection applications. Microphone activity detection with interrupt is available. A smart accessory interface supports most standard 3.5-mm accessories. Impedance sensing and measurement is provided for external accessory and push-button detection (Android™ headset specification compliant).

The CS47L15 supports SPI™ and I^2C interface modes for control-register access. The CS47L15 can also be configured as SPI master, enabling autonomous boot-up and configuration without dependency on a host processor. Two integrated FLLs support a wide range of system-clock frequencies. The device is powered from 1.8- and 1.2-V supplies. Separate MICVDD input can be supported, for microphone operation above 1.8 V. An additional supply is required for the Class D speaker drivers (typically direct connection to 4.2-V battery). The power, clocking, and output driver architectures are designed to maximize battery life in voice, music, and standby modes. Low-power (25 µW) Sleep Mode is supported, with configurable wake-up events.

Table of Contents

1 Pin Descriptions 4
1.1 WLCSP Pinout 4
1.2 Pin Descriptions 5
2 Typical Connection Diagram 8
3 Characteristics and Specifications 9
Table 3-1. Parameter Definitions 9
Table 3-2. Absolute Maximum Ratings 9
Table 3-3. Recommended Operating Conditions 10
Table 3-4. Analog Input Signal Level—IN1Axx, IN1Bxx, IN2x 10
Table 3-5. Analog Input Pin Characteristics 10
Table 3-6. Analog Input Gain—Programmable Gain Amplifiers (PGAs) 10
Table 3-7. Digital Input Signal Level—DMICDAT, SPKRXDAT 11
Table 3-8. Output Characteristics 11
Table 3-9. Input/Output Path Characteristics 11
Table 3-10. Digital Input/Output 15
Table 3-11. Miscellaneous Characteristics 16
Table 3-12. Device Reset Thresholds 17
Table 3-13. System Clock and Frequency-Locked Loop (FLL) 17
Table 3-14. Digital Microphone (DMIC) Interface Timing 18
Table 3-15. Digital Speaker (PDM) Interface Timing 18
Table 3-16. Digital Audio Interface—Master Mode 19
Table 3-17. Digital Audio Interface—Slave Mode 20
Table 3-18. Digital Audio Interface Timing—TDM Mode 20
Table 3-19. Control Interface Timing—Two-Wire (I2C) Mode 21
Table 3-20. Control Interface Timing—Four-Wire (SPI) Mode 22
Table 3-21. Master Interface Timing—SPI Master 23
Table 3-22. JTAG Interface Timing 24
Table 3-23. Typical Power Consumption 25
Table 3-24. Typical Signal Latency 25
4 Functional Description 26
4.1 Overview 26
4.2 Input Signal Path 29
4.3 Digital Core 42
4.4 DSP Firmware Control 77
4.5 DSP Peripheral Control 90
4.6 Digital Audio Interface 104
4.7 Digital Audio Interface Control 111
4.8 Output Signal Path 120
4.9 External Accessory Detection 133
4.10 Low Power Sleep Configuration 149
4.11 General-Purpose I/O 151
4.12 Interrupts 159
4.13 Clocking and Sample Rates 173
4.14 Control Interface and Master-Boot Interface 195
4.15 Control-Write Sequencer 202
4.16 Charge Pumps, Regulators, and Voltage Reference 210
4.17 JTAG Interface 213
4.18 Thermal, Short-Circuit, and Timer-Controlled Protection . 214
4.19 Power-Up, Resets, and Device ID 215
5 Applications 218
5.1 Recommended External Components 218
5.2 Resets Summary 228
5.3 Output-Signal Drive-Strength Control 228
5.4 Digital Audio Interface Clocking Configurations 229
5.5 PCB Layout Considerations 233
6 Register Map 234
7 Thermal Characteristics 274
8 Package Dimensions 275
9 Ordering Information 276
10 References 276
11 Revision History 276

1 Pin Descriptions

1.1 WLCSP Pinout

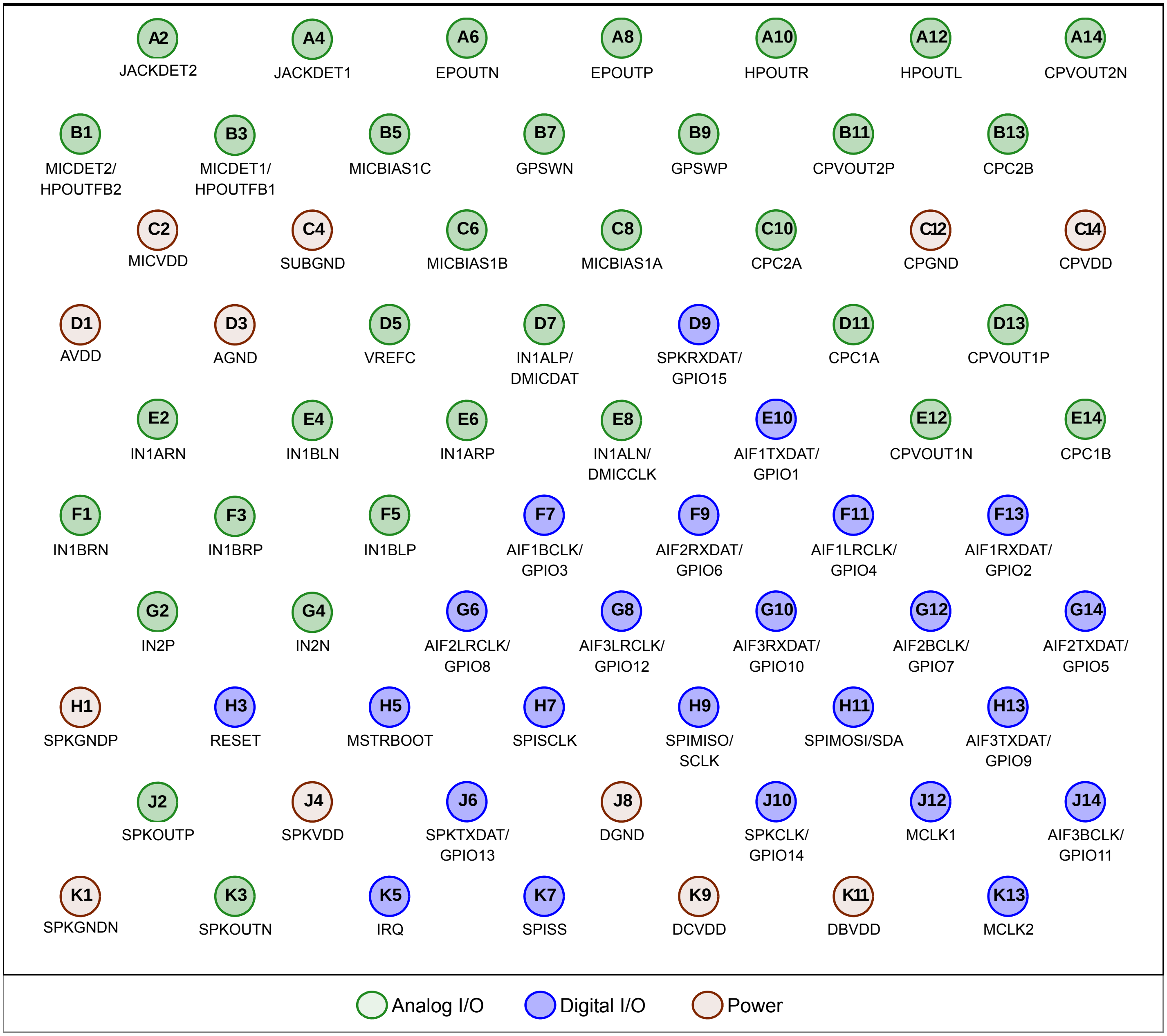


Figure 1-1. Top-Down (Through-Package) View—70-Ball WLCSP Package

1.2 Pin Descriptions

Table 1-1 describes each pin on the CS47L15. All digital output pins are CMOS outputs, unless otherwise stated.

Table 1-1. Pin Descriptions

PU = Pull-up, PD = Pull-down, K = Bus keeper, H = Hysteresis on CMOS input, Z = Hi-Z (High impedance), C = CMOS, OD = Open drain.

Pin Name	Pin #	Power Supply	I/O	Pin Description	Digital Pad Attributes	State at Reset [1]
Analog I/O						
CPC1A	D11	—	O	Charge pump fly-back capacitor 1 pin	—	—
CPC1B	E14	—	O	Charge pump fly-back capacitor 1 pin	—	—
CPC2A	C10	—	O	Charge pump fly-back capacitor 2 pin	—	—
CPC2B	B13	—	O	Charge pump fly-back capacitor 2 pin	—	—
CPVOUT1N	E12	—	O	Charge pump negative output 1 decoupling pin	—	Output
CPVOUT1P	D13	—	O	Charge pump positive output 1 decoupling pin	—	Output
CPVOUT2N	A14	—	O	Charge pump negative output 2 decoupling pin	—	Output
CPVOUT2P	B11	—	O	Charge pump positive output 2 decoupling pin	—	Output
EPOUTN	A6	—	O	Earpiece negative output	—	Output
EPOUTP	A8	—	O	Earpiece positive output	—	Output
GPSWN	B7	—	I/O	General-purpose bidirectional switch contact	—	—
GPSWP	B9	—	I/O	General-purpose bidirectional switch contact	—	—
HPOUTL	A12	—	O	Left headphone output	—	Output
HPOUTR	A10	—	O	Right headphone output	—	Output
IN1ALN/ DMICCLK	E8	MICVDD or MICBIAS*nx* [2]	I/O	Left-channel negative differential mic/line input /DMIC clock output	PD/H	IN1ALN input
IN1ALP/ DMICDAT	D7	MICVDD or MICBIAS*nx* [2]	I	Left-channel single-ended mic/line input/left-channel positive differential mic/line input/DMIC data input	PD/H	IN1ALP input
IN1ARN	E2	MICVDD	I	Right-channel negative differential mic/line input	—	Input
IN1ARP	E6	MICVDD	I	Right-channel single-ended mic/line input/ right-channel positive differential mic/line input	—	Input
IN1BLN	E4	MICVDD	I	Left-channel negative differential mic/line input. Also suitable for connection to external accessory interfaces.	—	Input
IN1BLP	F5	MICVDD	I	Left-channel single-ended mic/line input/left-channel positive differential mic/line input. Also suitable for connection to external accessory interfaces.	—	Input
IN1BRN	F1	MICVDD	I	Right-channel negative differential mic/line input. Also suitable for connection to external accessory interfaces.	—	Input
IN1BRP	F3	MICVDD	I	Right-channel single-ended mic/line input/ right-channel positive differential mic/line input. Also suitable for connection to external accessory interfaces.	—	Input
IN2N	G4	MICVDD	I	Negative differential analog input	—	Input
IN2P	G2	MICVDD	I	Positive differential analog input	—	Input
JACKDET1	A4	AVDD	I	Jack detect input 1	—	Input
JACKDET2	A2	AVDD	I	Jack detect input 2	—	Input
MICBIAS1A	C8	—	O	Microphone bias 1A	—	Output
MICBIAS1B	C6	—	O	Microphone bias 1B	—	Output
MICBIAS1C	B5	—	O	Microphone bias 1C	—	Output
MICDET1/ HPOUTFB1	B3	—	I	Microphone and accessory sense input 1/HPOUTL and HPOUTR ground feedback pin 1	—	Input
MICDET2/ HPOUTFB2	B1	—	I	Microphone and accessory sense input 2/HPOUTL and HPOUTR ground feedback pin 2	—	Input

Table 1-1. Pin Descriptions *(Cont.)*

PU = Pull-up, PD = Pull-down, K = Bus keeper, H = Hysteresis on CMOS input, Z = Hi-Z (High impedance), C = CMOS, OD = Open drain.

Pin Name	Pin #	Power Supply	I/O	Pin Description	Digital Pad Attributes	State at Reset [1]
SPKOUTN	K3	—	O	Speaker negative output	—	Output
SPKOUTP	J2	—	O	Speaker positive output	—	Output
VREFC	D5	—	O	Band-gap reference external capacitor connection	—	Output
				Digital I/O		
AIF1BCLK/ GPIO3	F7	DBVDD	I/O	Audio interface 1 bit clock/GPIO	PU/PD/K/H/ Z/C/OD	GPIO3 input with bus-keeper
AIF1LRCLK/ GPIO4	F11	DBVDD	I/O	Audio interface 1 left/right clock/GPIO	PU/PD/K/H/ Z/C/OD	GPIO4 input with bus-keeper
AIF1RXDAT/ GPIO2	F13	DBVDD	I/O	Audio interface 1 RX digital audio data/GPIO	PU/PD/K/H/ C/OD	GPIO2 input with bus-keeper
AIF1TXDAT/ GPIO1	E10	DBVDD	I/O	Audio interface 1 TX digital audio data/GPIO	PU/PD/K/H/ Z/C/OD	GPIO1 input with bus-keeper
AIF2BCLK/ GPIO7	G12	DBVDD	I/O	Audio interface 2 bit clock/GPIO	PU/PD/K/H/ Z/C/OD	GPIO7 input with bus-keeper
AIF2LRCLK/ GPIO8	G6	DBVDD	I/O	Audio interface 2 left/right clock/GPIO	PU/PD/K/H/ Z/C/OD	GPIO8 input with bus-keeper
AIF2RXDAT/ GPIO6	F9	DBVDD	I/O	Audio interface 2 RX digital audio data/GPIO	PU/PD/K/H/ C/OD	GPIO6 input with bus-keeper
AIF2TXDAT/ GPIO5	G14	DBVDD	I/O	Audio interface 2 TX digital audio data/GPIO. If the JTAG interface is configured, this pin provides the TDI input connection.	PU/PD/K/H/ Z/C/OD	GPIO5 input with bus-keeper
AIF3BCLK/ GPIO11	J14	DBVDD	I/O	Audio interface 3 bit clock/GPIO. If the JTAG interface is configured, this pin provides the TCK input connection.	PU/PD/K/H/ Z/C/OD	GPIO11 input with bus-keeper
AIF3LRCLK/ GPIO12	G8	DBVDD	I/O	Audio interface 3 left/right clock/GPIO. If the JTAG interface is configured, this pin provides the TDO output connection.	PU/PD/K/H/ Z/C/OD	GPIO12 input with bus-keeper
AIF3RXDAT/ GPIO10	G10	DBVDD	I/O	Audio interface 3 RX digital audio data/GPIO. If the JTAG interface is configured, this pin provides the TMS input connection.	PU/PD/K/H/ C/OD	GPIO10 input with bus-keeper
AIF3TXDAT/ GPIO9	H13	DBVDD	I/O	Audio interface 3 TX digital audio data/GPIO. If the JTAG interface is configured, this pin provides the TRST input connection.	PU/PD/K/H/ Z/C/OD	GPIO9 input with bus-keeper
$\overline{\text{IRQ}}$	K5	DBVDD	O	Interrupt request output (default is active low). The pin configuration is selectable CMOS or open drain.	C/OD	Output
MCLK1	J12	DBVDD	I	Master clock 1	H	Input
MCLK2	K13	DBVDD	I	Master clock 2	H	Input
MSTRBOOT	H5	DBVDD	I	Master boot mode select	PD/H	Input
$\overline{\text{RESET}}$	H3	DBVDD	I	Digital reset input (active low)	PU/PD/K/H	Input with pull-up
SPIMISO/ SCLK	H9	DBVDD	I/O	Control interface (SPI) Master In Slave Out data/I^{2}C clock input. SPIMISO is high impedance if $\overline{\text{SPISS}}$ is not asserted.	PD/H/C	Input
SPIMOSI/SDA	H11	DBVDD	I/O	Control interface (SPI) Master Out Slave In data/I^{2}C data input and output.	H/C/OD	Input
SPISCLK	H7	DBVDD	I/O	Control interface (SPI) clock	H/C	Input
$\overline{\text{SPISS}}$	K7	DBVDD	I/O	Control interface (SPI) slave select (SS)	H/C	Input
SPKCLK/ GPIO14	J10	DBVDD	I/O	Digital speaker (PDM) clock output/GPIO/I^{2}C clock input. GPIO output is selectable CMOS or open drain; SPKCLK output is CMOS.	PU/PD/K/H/ C/OD	GPIO14 input with bus-keeper
SPKRXDAT/ GPIO15	D9	DBVDD	I/O	Digital speaker (PDM) data input/GPIO/I^{2}C data input and output. GPIO output is selectable CMOS or open drain.	PU/PD/K/H/ C/OD	GPIO15 input with bus-keeper

Table 1-1. Pin Descriptions *(Cont.)*

PU = Pull-up, PD = Pull-down, K = Bus keeper, H = Hysteresis on CMOS input, Z = Hi-Z (High impedance), C = CMOS, OD = Open drain.

Pin Name	Pin #	Power Supply	I/O	Pin Description	Digital Pad Attributes	State at Reset [1]
SPKTXDAT/ GPIO13	J6	DBVDD	I/O	Digital speaker (PDM) data output/GPIO. GPIO output is selectable CMOS or open drain; SPKTXDAT output is CMOS.	PU/PD/K/H/ C/OD	GPIO13 input with bus-keeper
				Supply		
AGND	D3	—	—	Analog ground (return path for AVDD and MICVDD)	—	—
AVDD	D1	—	—	Analog supply	—	—
CPGND	C12	—	—	Charge pump ground (return path for CPVDD)	—	—
CPVDD	C14	—	—	Supply for charge pump	—	—
DBVDD	K11	—	—	Digital buffer (I/O) supply	—	—
DCVDD	K9	—	—	Digital core supply	—	—
DGND	J8	—	—	Digital ground (return path for DCVDD and DBVDD)	—	—
MICVDD	C2	—	—	Microphone bias supply (input to MICBIAS regulator)	—	—
SPKGNDN	K1	—	—	Speaker driver ground (return path for SPKVDD) [3]	—	—
SPKGNDP	H1	—	—	Speaker driver ground (return path for SPKVDD) [3]	—	—
SPKVDD	J4	—	—	Speaker driver supply	—	—
SUBGND	C4	—	—	Substrate ground	—	—

1. Note that the default conditions described are not valid if modified by the boot sequence or by a wake-up control sequence.
2. The analog input functions on these pins are referenced to the MICVDD power domain. The digital input/output functions are referenced to the MICVDD or MICBIAS1 power domain, as selected by the IN1_DMIC_SUP field.
3. Separate P/N ground connections are provided for the Class D speaker output, which provides flexible support for current monitoring and output-protection circuits. If this option is not used, these ground connections should be tied together on the PCB.

2 Typical Connection Diagram

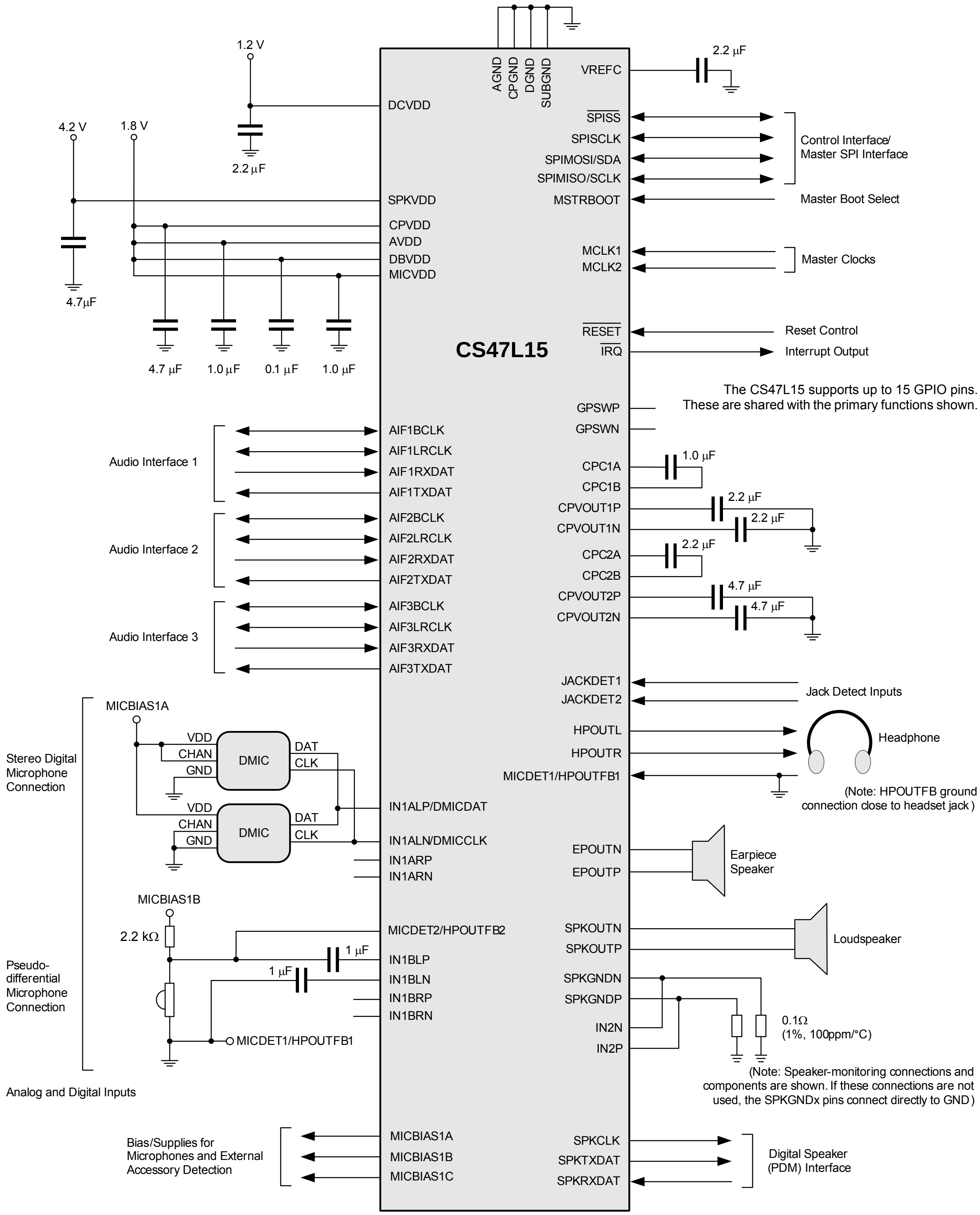


Figure 2-1. Typical Connection Diagram

3 Characteristics and Specifications

Table 3-1 defines parameters as they are characterized in this section.

Table 3-1. Parameter Definitions

Parameter	Definition
Channel separation	Left-to-right and right-to-left channel separation is the difference in level between the active channel (driven to maximum full scale output) and the measured signal level in the idle channel at the test signal frequency. The active channel is configured and supplied with an appropriate input signal to drive a full scale output, with signal measured at the output of the associated idle channel.
Common-mode rejection ratio (CMRR)	The ratio of a specified input signal (applied to both sides of a differential input), relative to the output signal that results from it.
Dynamic range (DR)	A measure of the difference between the maximum full scale output signal and the sum of all harmonic distortion products plus noise, with a low-level input signal applied. Typically, an input signal level 60 dB below full scale is used.
Power-supply rejection ratio (PSRR)	The ratio of a specified power supply variation relative to the output signal that results from it. PSRR is measured under quiescent signal path conditions.
Signal-to-noise ratio (SNR)	A measure of the difference in level between the maximum full scale output signal and the output with no input signal applied.
Total harmonic distortion (THD)	The ratio of the RMS sum of the harmonic distortion products in the specified bandwidth [1] relative to the RMS amplitude of the fundamental (i.e., test frequency) output.
Total harmonic distortion plus noise (THD+N)	The ratio of the RMS sum of the harmonic distortion products plus noise in the specified bandwidth [1] relative to the RMS amplitude of the fundamental (i.e., test frequency) output.

1. All performance measurements are specified with a 20-kHz low-pass brick-wall filter and, where noted, an A-weighted filter. The low-pass filter removes out-of-band noise.

Table 3-2. Absolute Maximum Ratings

Absolute maximum ratings are stress ratings only. Permanent damage to the device may be caused by continuously operating at or beyond these limits. Device functional operating limits and guaranteed performance specifications are given under electrical characteristics at the test conditions specified.

Parameter	Symbol	Minimum	Maximum
Supply voltages	DCVDD CPVDD DBVDD, AVDD, MICVDD SPKVDD	–0.3 V –0.3 V –0.3 V –0.3 V	1.6 V 2.5 V 5.0 V 6.0 V
Voltage range digital inputs	—	SUBGND – 0.3 V	DBVDD + 0.3 V
Voltage range analog inputs	IN1Axx, IN2xx IN1Bxx HPOUTFB*n* [1] MICDET*n* [1] JACKDET1 JACKDET2 [2], GPSWP, GPSWN	SUBGND – 0.3 V SUBGND – 0.9 V SUBGND – 0.3 V SUBGND – 0.3 V CPVOUT2N – 0.3 V [3] SUBGND – 0.3 V	MICVDD + 0.3 V MICVDD + 0.3 V SUBGND + 0.3 V MICVDD + 0.3 V AVDD + 0.3 V MICVDD + 0.3 V
Ground	AGND, DGND, CPGND, SPKGNDN, SPKGNDP	SUBGND – 0.3 V	SUBGND + 0.3 V
Operating temperature range	T_A	–40ºC	+85ºC
Operating junction temperature	T_J	–40ºC	+125ºC
Storage temperature after soldering	—	–65ºC	+150ºC

ESD-sensitive device. The CS47L15 is manufactured on a CMOS process. It is therefore generically susceptible to damage from excessive static voltages. Proper ESD precautions must be taken during handling and storage of this device. This device is qualified to current JEDEC ESD standards.

1. The HPOUTFB*n* and MICDET*n* functions share common pins. The absolute maximum rating varies according to the applicable function of each pin.
2. If AVDD > MICVDD the maximum JACKDET2 voltage is AVDD + 0.3 V.
3. CPVOUT2N is an internal supply, generated by the CS47L15 charge pump (CP). Its voltage can vary between CPGND and –CPVDD.

Table 3-3. Recommended Operating Conditions

Parameter		Symbol	Minimum	Typical	Maximum	Units
Digital supply range [1,2]	Core and FLL I/O	DCVDD [3] DBVDD	1.14 1.71	1.2 —	1.26 3.6	V V
Charge pump supply range	CPVDD	CPVDD	1.71	1.8	1.89	V
Speaker supply range		SPKVDD	2.4	—	5.5	V
Analog supply range		AVDD	1.71	1.8	1.89	V
Mic bias supply		MICVDD	1.71	1.8	3.6	V
Ground [4]		DGND, AGND, CPGND, SPKGNDN, SPKGNDP, SUBGND	—	0	—	V
Power supply rise time [5,6]		DCVDD All other supplies	100 100	— —	2000 —	µs µs
Operating temperature range		T_A	–40	—	85	ºC

1. When powering-up the CS47L15, the DBVDD and AVDD supplies must be enabled before DCVDD. The DCVDD domain must not be powered if DBVDD or AVDD is not present. There are no power-down sequencing requirements; the supplies may be disabled in any order.
2. When powering-up the CS47L15, RESET must be deasserted (high) before DCVDD is applied. RESET must be held high until at least 10 ms after DCVDD is applied.
3. Sleep mode is supported for when DCVDD is below the limits noted, provided that AVDD and DBVDD are present.
4. The impedance between DGND, AGND, and SUBGND must not exceed 0.1 Ω.
   The impedance between SPKGNDN, SPKGNDP, and SUBGND must not exceed 0.2 Ω.
5. If the DCVDD rise time exceeds 2 ms, RESET must be asserted (low) during the rise and held asserted until after DCVDD is within the recommended operating limits. This requirement takes precedence over Note 2 above.
6. The specified minimum power supply rise times assume a minimum decoupling capacitance of 100 nF per pin. However, Cirrus Logic strongly advises that the recommended decoupling capacitors are present on the PCB and that appropriate layout guidelines are observed. The specified minimum power supply rise times also assume a maximum PCB inductance of 10 nH between decoupling capacitor and pin.

Table 3-4. Analog Input Signal Level—IN1Axx, IN1Bxx, IN2x

Test conditions (unless specified otherwise): AVDD = 1.8V, sinusoid input signal; with the exception of the conditions noted, the following electrical characteristics are valid across the full range of recommended operating conditions.

Parameter		Minimum	Typical	Maximum	Units
Maximum input signal level (IN1Axx, IN1Bxx) [1, 2]	Single-ended configuration, 0 dB PGA gain	— —	0.5 –6	— —	V_{RMS} dBV
	Differential configuration [3], 0 dB PGA gain	— —	1 0	— —	V_{RMS} dBV
Maximum input signal level (IN2x) [4]	Differential configuration	— —	0.1 –20	— —	V_{RMS} dBV

Note: The maximum and full-scale input signal levels change in proportion with AVDD.

1. The maximum input signal level (before clipping occurs) is also the full-scale input signal level (0 dBFS) at the IN1 ADC outputs.
2. If Low-Power Mode is enabled, the maximum input signal level is reduced by 6 dB. The maximum input signal level corresponds to –6 dBFS at the IN1 ADC output in this case.
3. A 1.0V_{RMS} differential signal equates to 0.5V_{RMS}/–6dBV per input.
4. The maximum input signal level (before clipping occurs) corresponds to –6 dBFS at the IN2 ADC output.

Table 3-5. Analog Input Pin Characteristics

Test conditions (unless specified otherwise): T_A = +25ºC; with the exception of the condition noted, the following electrical characteristics are valid across the full range of recommended operating conditions.

Parameter		Minimum	Typical	Maximum	Units
Input resistance (IN1x)	Single-ended PGA input, All PGA gain settings Differential PGA input, All PGA gain settings	9 18	10 21	— —	kΩ kΩ
Input resistance (IN2x)		—	17	—	kΩ
Input capacitance		—	—	5	pF

Table 3-6. Analog Input Gain—Programmable Gain Amplifiers (PGAs)

The following electrical characteristics are valid across the full range of recommended operating conditions.

Parameter[1]		Minimum	Typical	Maximum	Units
Minimum programmable gain		—	0	—	dB
Maximum programmable gain		—	31	—	dB
Programmable gain step size	Guaranteed monotonic	—	1	—	dB

1. Note that PGA control is provided for the IN1x analog input channels only.

Table 3-7. Digital Input Signal Level—DMICDAT, SPKRXDAT

The following electrical characteristics are valid across the full range of recommended operating conditions.

Parameter		Minimum	Typical	Max	Units
Full-scale input level [1]	0 dBFS digital core input, 0 dB gain	—	–6	—	dBFS

1. The digital input signal level is measured in dBFS, where 0 dBFS is a signal level equal to the full-scale range (FSR) of the PDM input. The FSR is defined as the amplitude of a 1-kHz sine wave whose positive and negative peaks are represented by the maximum and minimum digital codes respectively—this is the largest 1-kHz sine wave that can fit in the digital output range without clipping.

Table 3-8. Output Characteristics

The following electrical characteristics are valid across the full range of recommended operating conditions.

Parameter			Minimum	Typical	Max	Units
Line/headphone/earpiece output driver (HPOUTL, HPOUTR)	Load resistance	Normal operation, Single-Ended Mode Normal operation, Differential (BTL) Mode Device survival with load applied indefinitely	6 15 0	— — —	— — —	Ω Ω Ω
	Load capacitance	Single-Ended Mode Differential (BTL) Mode	— —	— —	500 200	pF pF
Earpiece output driver (EPOUTP+EPOUTN)	Load resistance	Normal operation Device survival with load applied indefinitely	15 0	— —	— —	Ω Ω
	Load capacitance		—	—	200	pF
Speaker output driver (SPKOUTP+SPKOUTN)	Load resistance	Normal operation Device survival with load applied indefinitely	4 0	— —	— —	Ω Ω
	Load capacitance		—	—	200	pF
Digital speaker output (SPKTXDAT)	Full-scale output level [1]	0 dBFS digital core output, 0 dB gain	—	–6	—	dBFS

1. The digital output signal level is measured in dBFS, where 0 dBFS is a signal level equal to the full-scale range (FSR) of the PDM output. The FSR is defined as the amplitude of a 1-kHz sine wave whose positive and negative peaks are represented by the maximum and minimum digital codes respectively—this is the largest 1-kHz sine wave that can fit in the digital output range without clipping.

Table 3-9. Input/Output Path Characteristics

Test conditions (unless specified otherwise): DBVDD = CPVDD = AVDD = 1.8 V, DCVDD = 1.2 V; MICVDD = 2.5 V; SPKVDD = 4.2 V; T_A = +25ºC; 1 kHz sinusoid signal; Fs = 48 kHz; PGA gain = 0 dB, 24-bit audio data.

Parameter			Min	Typ	Max	Units
Line/headphone/earpiece output driver (HPOUTL, HPOUTR)	DC offset at Load	Single-ended mode Differential (BTL) mode	— —	50 75	— —	µV µV
Earpiece output driver (EPOUTP+EPOUTN)	DC offset at Load		—	75	—	µV
Speaker output driver (SPKOUTP+SPKOUTN)	DC offset at Load		—	300	—	µV
	SPKVDD leakage current		—	1	—	µA
Analog input paths (IN1xL, IN1xR) to ADC (Differential Input Mode)	SNR (A-weighted), defined in Table 3-1	48 kHz sample rate 16 kHz sample rate (wideband voice)	90 —	98 104	— —	dB dB
	THD, defined in Table 3-1	–1 dBV input	—	–87	—	dB
	THD+N, defined in Table 3-1	–1 dBV input	—	–88	–80	dB
	Channel separation (L/R), defined in Table 3-1	100 Hz to 10 kHz	—	109	—	dB
	Input-referred noise floor	A-weighted, PGA gain = +20 dB	—	2.7	—	μV_{RMS}
	CMRR, defined in Table 3-1	PGA gain = +30 dB PGA gain = 0 dB	— —	79 70	— —	dB dB
	PSRR (DBVDD, CPVDD, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	93 77	— —	dB dB
	PSRR (MICVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	98 90	— —	dB dB
	PSRR (DCVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	98 83	— —	dB dB
	PSRR (SPKVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	100 95	— —	dB dB

Table 3-9. Input/Output Path Characteristics *(Cont.)*

Test conditions (unless specified otherwise): DBVDD = CPVDD = AVDD = 1.8 V, DCVDD = 1.2 V; MICVDD = 2.5 V; SPKVDD = 4.2 V; T_A = +25ºC; 1 kHz sinusoid signal; Fs = 48 kHz; PGA gain = 0 dB, 24-bit audio data.

Parameter			Min	Typ	Max	Units
Analog input paths (IN1xL, IN1xR) to ADC (Single-Ended Input Mode)	SNR (A-weighted), defined in Table 3-1	48-kHz sample rate 16-kHz sample rate (wideband voice)	85 —	97 102	— —	dB dB
	THD, defined in Table 3-1	–7dBV input	—	–86	—	dB
	THD+N, defined in Table 3-1	–7dBV input	—	–85	–78	dB
	Channel separation (L/R), defined in Table 3-1	100 Hz to 10 kHz	—	107	—	dB
	Input-referred noise floor	A-weighted, PGA gain = +20 dB	—	4	—	μV_{RMS}
	PSRR (DBVDD, CPVDD, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	77 52	— —	dB dB
	PSRR (MICVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	100 90	— —	dB dB
	PSRR (DCVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	96 74	— —	dB dB
	PSRR (SPKVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	100 80	— —	dB dB
Analog input path (IN2) to ADC (Differential Input Mode)	SNR (A-weighted), defined in Table 3-1	48 kHz sample rate	—	70	—	dB
	THD, defined in Table 3-1	–21 dBV input	—	–65	—	dB
	THD+N, defined in Table 3-1	–21 dBV input	—	–63	—	dB
	Input-referred noise floor	A-weighted	—	28	—	μV_{RMS}
	CMRR, defined in Table 3-1		— —	60 70	— —	dB dB
	PSRR (DBVDD, CPVDD, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	50 50	— —	dB dB
	PSRR (MICVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	71 83	— —	dB dB
	PSRR (DCVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	50 50	— —	dB dB
	PSRR (SPKVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	50 50	— —	dB dB
DAC to line output (HPOUTL, HPOUTR; Load = 10 kΩ, 50 pF)	Full-scale output signal level	0 dBFS input	— —	1 0	— —	V_{RMS} dBV
	SNR, defined in Table 3-1	A-weighted, output signal = 1 V_{RMS}	—	127	—	dB
	Dynamic range, defined in Table 3-1	A-weighted, –60 dBFS input	105	114	—	dB
	THD, defined in Table 3-1	0 dBFS input	—	–94	—	dB
	THD+N, defined in Table 3-1	0 dBFS input	—	–92	–85	dB
	Channel separation (L/R), defined in Table 3-1	100 Hz to 10 kHz	—	105	—	dB
	Output noise floor	A-weighted	—	0.45	—	μV_{RMS}
	PSRR (DBVDD, CPVDD, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	124 80	— —	dB dB
	PSRR (MICVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	110 105	— —	dB dB
	PSRR (DCVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	126 90	— —	dB dB
	PSRR (SPKVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	110 100	— —	dB dB

Table 3-9. Input/Output Path Characteristics *(Cont.)*

Test conditions (unless specified otherwise): DBVDD = CPVDD = AVDD = 1.8 V, DCVDD = 1.2 V; MICVDD = 2.5 V; SPKVDD = 4.2 V; T_A = +25ºC; 1 kHz sinusoid signal; Fs = 48 kHz; PGA gain = 0 dB, 24-bit audio data.

Parameter			Min	Typ	Max	Units
DAC to headphone output (HPOUTL, HPOUTR; R_L = 32 Ω)	Maximum output power	0.1% THD+N	—	30	—	mW
	SNR, defined in Table 3-1	A-weighted, output signal = 1 V_{RMS}	—	127	—	dB
	Dynamic range, defined in Table 3-1	A-weighted, –60 dBFS input	105	115	—	dB
	THD, defined in Table 3-1	P_O = 25 mW	—	–94	—	dB
	THD+N, defined in Table 3-1	P_O = 25 mW	—	–92	—	dB
	THD, defined in Table 3-1	P_O = 20 mW	—	–92	—	dB
	THD+N, defined in Table 3-1	P_O = 20 mW	—	–90	–85	dB
	THD, defined in Table 3-1	P_O = 2 mW	—	–92	—	dB
	THD+N, defined in Table 3-1	P_O = 2 mW	—	–90	—	dB
	Channel separation (L/R), defined in Table 3-1	100 Hz to 10 kHz	—	102	—	dB
	Output noise floor	A-weighted	—	0.45	—	μV_{RMS}
	PSRR (DBVDD, CPVDD, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	124 80	— —	dB dB
	PSRR (MICVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	124 110	— —	dB dB
	PSRR (DCVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	126 90	— —	dB dB
	PSRR (SPKVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	110 100	— —	dB dB
DAC to headphone output (HPOUTL, HPOUTR; R_L = 16 Ω)	Maximum output power	0.1% THD+N	—	40	—	mW
	SNR, defined in Table 3-1	A-weighted, output signal = 1 V_{RMS}	—	127	—	dB
	Dynamic range, defined in Table 3-1	A-weighted, –60 dBFS input	105	114	—	dB
	THD, defined in Table 3-1	P_O = 25 mW	—	–90	—	dB
	THD+N, defined in Table 3-1	P_O = 25 mW	—	–88	—	dB
	THD, defined in Table 3-1	P_O = 20 mW	—	–90	—	dB
	THD+N, defined in Table 3-1	P_O = 20 mW	—	–88	–80	dB
	THD, defined in Table 3-1	P_O = 2 mW	—	–88	—	dB
	THD+N, defined in Table 3-1	P_O = 2 mW	—	–86	—	dB
	Channel separation (L/R), defined in Table 3-1	100 Hz to 10 kHz	—	100	—	dB
	Output noise floor	A-weighted	—	0.45	—	μV_{RMS}
	PSRR (DBVDD, CPVDD, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	124 80	— —	dB dB
	PSRR (MICVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	124 110	— —	dB dB
	PSRR (DCVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	126 90	— —	dB dB
	PSRR (SPKVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	110 100	— —	dB dB
DAC to earpiece output (EPOUTP+EPOUTN, R_L = 32 Ω BTL)	Maximum output power	0.1% THD+N	—	96	—	mW
	SNR, defined in Table 3-1	A-weighted, output signal = 1.41 V_{RMS}	—	128	—	dB
	Dynamic range, defined in Table 3-1	A-weighted, –60 dBFS input	105	118	—	dB
	THD, defined in Table 3-1	P_O = 75 mW	—	–92	—	dB
	THD+N, defined in Table 3-1	P_O = 75 mW	—	–88	—	dB
	THD, defined in Table 3-1	P_O = 5 mW	—	–88	—	dB
	THD+N, defined in Table 3-1	P_O = 5 mW	—	–86	—	dB
	Output noise floor	A-weighted	—	0.60	—	μV_{RMS}
	PSRR (DBVDD, CPVDD, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	85 85	— —	dB dB
	PSRR (MICVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	124 110	— —	dB dB
	PSRR (DCVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	126 86	— —	dB dB
	PSRR (SPKVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	110 105	— —	dB dB

Table 3-9. Input/Output Path Characteristics *(Cont.)*

Test conditions (unless specified otherwise): DBVDD = CPVDD = AVDD = 1.8 V, DCVDD = 1.2 V; MICVDD = 2.5 V; SPKVDD = 4.2 V; T_A = +25ºC; 1 kHz sinusoid signal; Fs = 48 kHz; PGA gain = 0 dB, 24-bit audio data.

Parameter			Min	Typ	Max	Units
DAC to earpiece output (EPOUTP+EPOUTN, R_L = 16 Ω BTL)	Maximum output power	0.1% THD+N	—	108	—	mW
	SNR, defined in Table 3-1	A-weighted, output signal = 1.41 V_{RMS}	—	128	—	dB
	Dynamic range, defined in Table 3-1	A-weighted, –60 dBFS input	105	118	—	dB
	THD, defined in Table 3-1	P_O = 75 mW	—	–89	—	dB
	THD+N, defined in Table 3-1	P_O = 75 mW	—	–87	—	dB
	THD, defined in Table 3-1	P_O = 5 mW	—	–90	—	dB
	THD+N, defined in Table 3-1	P_O = 5 mW	—	–88	—	dB
	Output noise floor	A-weighted	—	0.60	—	μV_{RMS}
	PSRR (DBVDD, CPVDD, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	85 85	— —	dB dB
	PSRR (MICVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	124 110	— —	dB dB
	PSRR (DCVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	126 86	— —	dB dB
	PSRR (SPKVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	108 110	— —	dB dB
DAC to speaker output (SPKOUTP+SPKOUTN, Load = 8 Ω, 22 μH, BTL)	Maximum output power	SPKVDD = 5.0 V, 1% THD+N SPKVDD = 4.2 V, 1% THD+N SPKVDD = 3.6 V, 1% THD+N	— — —	1.4 1.0 0.7	— — —	W W W
	SNR, defined in Table 3-1	A-weighted, output signal = 2.83 V_{RMS}	—	127	—	dB
	Dynamic range, defined in Table 3-1	A-weighted, –60 dBFS input	90	100	—	dB
	THD, defined in Table 3-1	P_O = 1.0 W	—	–40	—	dB
	THD+N, defined in Table 3-1	P_O = 1.0 W	—	–40	—	dB
	THD, defined in Table 3-1	P_O = 0.5 W	—	–61	—	dB
	THD+N, defined in Table 3-1	P_O = 0.5 W	—	–60	–50	dB
	Output noise floor	A-weighted	—	1.3	—	μV_{RMS}
	PSRR (DBVDD, CPVDD, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	110 90	— —	dB dB
	PSRR (MICVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	124 110	— —	dB dB
	PSRR (DCVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	125 105	— —	dB dB
	PSRR (SPKVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	125 105	— —	dB dB
DAC to speaker output (SPKOUTP+SPKOUTN, Load = 4 Ω, 15 μH, BTL)	Maximum output power	SPKVDD = 5.0 V, 1% THD+N SPKVDD = 4.2 V, 1% THD+N SPKVDD = 3.6 V, 1% THD+N	— — —	2.5 1.8 1.3	— — —	W W W
	SNR, defined in Table 3-1	A-weighted, output signal = 2.83 V_{RMS}	—	127	—	dB
	Dynamic range, defined in Table 3-1	A-weighted, –60 dBFS input	—	100	—	dB
	THD, defined in Table 3-1	P_O = 1.0 W	—	–40	—	dB
	THD+N, defined in Table 3-1	P_O = 1.0 W	—	–40	—	dB
	THD, defined in Table 3-1	P_O = 0.5 W	—	–61	—	dB
	THD+N, defined in Table 3-1	P_O = 0.5 W	—	–60	—	dB
	Output noise floor	A-weighted	—	1.3	—	μV_{RMS}
	PSRR (DBVDD, CPVDD, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	110 90	— —	dB dB
	PSRR (MICVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	124 110	— —	dB dB
	PSRR (DCVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	125 105	— —	dB dB
	PSRR (SPKVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	125 105	— —	dB dB

Table 3-10. Digital Input/Output

The following electrical characteristics are valid across the full range of recommended operating conditions.

Parameter			Minimum	Typical	Maximum	Units
Digital I/O (except DMICDAT and DMICCLK) [1,3]	Input HIGH level	V_{DBVDD} = 1.71–1.98 V V_{DBVDD} = 2.5 V ±10% V_{DBVDD} = 3.3 V ±10%	0.75 × DBVDD 0.8 × DBVDD 0.7 × DBVDD	— — —	— — —	V V V
	Input LOW level	V_{DBVDD} = 1.71–1.98 V V_{DBVDD} = 2.5 V ±10% V_{DBVDD} = 3.3 V ±10%	— — —	— — —	0.3 × DBVDD 0.25 × DBVDD 0.2 × DBVDD	V V V
	Output HIGH level (I_{OH} = 1 mA)	V_{DBVDD} = 1.71–1.98 V V_{DBVDD} = 2.5 V ±10% V_{DBVDD} = 3.3 V ±10%	0.75 × DBVDD 0.65 × DBVDD 0.7 × DBVDD	— — —	— — —	V V V
	Output LOW level (I_{OL} = 1mA)	V_{DBVDD} = 1.71–1.98 V V_{DBVDD} = 2.5 V ±10% V_{DBVDD} = 3.3 V ±10%	— — —	— — —	0.25 × DBVDD 0.3 × DBVDD 0.15 × DBVDD	V V V
	Input capacitance		—	—	5	pF
	Input leakage		–1	—	1	µA
	Pull-up/pull-down resistance (where applicable)	RESET pin All other pins	35 25	— —	55 50	kΩ kΩ
DMIC I/O (DMICDAT and DMICCLK) [2,3]	DMICDAT input HIGH Level		0.65 × V_{SUP}	—	—	V
	DMICDAT input LOW Level		—	—	0.35 × V_{SUP}	V
	DMICCLK output HIGH Level	I_{OH} = 1 mA	0.8 × V_{SUP}	—	—	V
	DMICCLK output LOW Level	I_{OL} = –1 mA	—		0.2 × V_{SUP}	V
	Input capacitance		—	25		pF
	Input leakage		–1	—	1	µA
GPIO*n*	Clock output frequency	GPIO pin as OPCLK or FLL output	—	—	50	MHz

1. Digital I/O is referenced to DBVDD.
2. DMICDAT and DMICCLK are referenced to a selectable supply, V_{SUP}, according to the IN1_DMIC_SUP field.
3. Note that digital input pins should not be left unconnected or floating.

Table 3-11. Miscellaneous Characteristics

Test conditions (unless specified otherwise): DBVDD = CPVDD = AVDD = 1.8 V, DCVDD = 1.2 V; MICVDD = 2.5 V; SPKVDD = 4.2 V; T_A = +25ºC; 1 kHz sinusoid signal; Fs = 48 kHz; PGA gain = 0 dB, 24-bit audio data.

Parameter			Min	Typ	Max	Units
Microphone bias (MICBIAS1A, MICBIAS1B, MICBIAS1C) [1]	Minimum Bias Voltage [2] Maximum Bias Voltage Bias Voltage output step size Bias Voltage accuracy		— — — –5%	1.5 2.8 0.1 —	— — — +5%	V V V V
	Bias Current [3]	Regulator Mode (MICB1_BYPASS = 0), $V_{MICVDD} - V_{MICBIAS}$ >200 mV Bypass Mode (MICBn_BYPASS = 1)	— —	— —	2.4 5.0	mA mA
	Output Noise Density	Regulator Mode (MICB1_BYPASS = 0), MICB1_LVL = 0x4, Load current = 1 mA, Measured at 1 kHz	—	50	—	nV/√Hz
	Integrated noise voltage	Regulator Mode (MICB1_BYPASS = 0), MICB1_LVL = 0x4, Load current = 1 mA, 100 Hz to 7 kHz, A-weighted	—	5	—	μV_{RMS}
	PSRR (DBVDD, CPVDD, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	100 80	— —	dB dB
	PSRR (MICVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	82 44	— —	dB dB
	PSRR (DCVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	100 80	— —	dB dB
	PSRR (SPKVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	100 80	— —	dB dB
	Load capacitance [3]	Regulator Mode (MICB1_BYPASS = 0), MICB1_EXT_CAP = 0 Regulator Mode (MICB1_BYPASS = 0), MICB1_EXT_CAP = 1	— 0.1	— 1.0	50 10	pF µF
	Output discharge resistance	MICB*nx*_ENA = 0, MICB*nx*_DISCH = 1	—	2	—	kΩ
General-purpose switch [4]	Switch resistance	Switch closed, I = 1 mA Switch open	— —	25 100	40 —	Ω MΩ
External Accessory Detect	Headphone detection load impedance range: Detection via HPOUTL (HPD_SENSE_SEL = 100) or HPOUTR (HPD_SENSE_SEL = 101)	HPD_IMPEDANCE_RANGE = 01 HPD_IMPEDANCE_RANGE = 10 HPD_IMPEDANCE_RANGE = 11	0 90 1	— — —	90 1000 10	Ω Ω kΩ
	Headphone detection load impedance range: Detection via MICDET*n* or JACKDET*n* pins		400	—	6000	Ω
	Headphone detection accuracy: (HPD_DACVAL, HPD_SENSE_SEL = 100 or 101)	HPD_IMPEDANCE_RANGE = 01 HPD_IMPEDANCE_RANGE = 10 HPD_IMPEDANCE_RANGE = 11	–10 –5 –10	— — —	+10 +5 +10	% % %
	Headphone detection accuracy (HPD_LVL, HPD_SENSE_SEL = 0XX or 11X)		–20	—	+20	%
	Microphone impedance detection range: (MICD1_ADC_MODE = 0, 2.2 kΩ ±2% MICBIAS resistor. [5]	for MICD1_LVL[0] = 1 for MICD1_LVL[1] = 1 for MICD1_LVL[2] = 1 for MICD1_LVL[3] = 1 for MICD1_LVL[8] = 1	0 110 210 360 1	— — — — —	70 180 290 680 30	Ω Ω Ω Ω kΩ
	Jack-detection input threshold voltage (JACKDET*n*)	Detection on JACKDET1, Jack insertion Detection on JACKDET1, Jack removal Detection on JACKDET2, Jack insertion Detection on JACKDET2, Jack removal	— — — —	0.9 1.65 0.27 0.9	— — — —	V V V V
	Pull-up resistance (JACKDET*n*)		—	1	—	MΩ
Frequency-Locked Loop (FLL1)	Output frequency	FLL output as SYSCLK source FLL output as DSPCLK source	90 135	— —	98.3 150	MHz MHz
	Lock Time	F_{REF} = 32 kHz, F_{OUT} (DSPCLK source) = 147.456 MHz F_{REF} = 12 MHz, F_{OUT} (DSPCLK source) = 147.456 MHz	— —	10 1	— —	ms ms
$\overline{\text{RESET}}$ pin input	$\overline{\text{RESET}}$ input pulse width [6]		1	—	—	µs

1. No capacitor on MICBIAS1x. In Regulator Mode, it is required that $V_{MICVDD} - V_{MICBIAS}$ > 200 mV.
2. Regulator Mode (MICB1_BYPASS = 0), Load current ≤ 1.0 mA.
3. Bias current and load capacitance specifications are for the sum of all enabled MICBIAS1*x* outputs.
4. The GPSWN pin voltage must not exceed GPSWP + 0.3 V. See Table 3-2 for voltage limits applicable to the GPSWP and GPSWN pins.
5. These characteristics assume no other component is connected to MICDET*n*.
6. To trigger a hardware reset, the $\overline{\text{RESET}}$ input must be asserted for longer than this duration.

Table 3-12. Device Reset Thresholds

The following electrical characteristics are valid across the full range of recommended operating conditions.

Parameter		Symbol	Minimum	Typical	Maximum	Units
AVDD reset threshold	V_{AVDD} rising V_{AVDD} falling	V_{AVDD}	— 1.06	— —	1.66 1.44	V V
DCVDD reset threshold	V_{DCVDD} rising V_{DCVDD} falling	V_{DCVDD}	— 0.40	— —	1.04 0.72	V V
DBVDD reset threshold	V_{DBVDD} rising V_{DBVDD} falling	V_{DBVDD}	— 1.06	— —	1.66 1.44	V V

Note: The reset thresholds are derived from simulations only, across all operational and process corners. Device performance is not assured outside the voltage ranges defined in Table 3-3.

Table 3-13. System Clock and Frequency-Locked Loop (FLL)

The following timing information is valid across the full range of recommended operating conditions.

Parameter			Minimum	Typical	Maximum	Units
Master clock timing (MCLK1, MCLK2) [1]	MCLK cycle time	MCLK as input to FLL, FLL1_REFCLK_DIV = 00	74	—	—	ns
		MCLK as input to FLL, FLL1_REFCLK_DIV = 01	37	—	—	ns
		MCLK as input to FLL, FLL1_REFCLK_DIV = 10	18	—	—	ns
		MCLK as input to FLL, FLL1_REFCLK_DIV = 11	12.5	—	—	ns
		MCLK as direct SYSCLK source	40	—	—	ns
	MCLK duty cycle	MCLK as input to FLL	80:20	—	20:80	%
		MCLK as direct SYSCLK source	60:40	—	40:60	%
Frequency-locked loop (FLL1)	FLL input frequency	FLL1_REFCLK_DIV = 00	0.032	—	13.5	MHz
		FLL1_REFCLK_DIV = 01	0.064	—	27	MHz
		FLL1_REFCLK_DIV = 11	0.128	—	54	MHz
		FLL1_REFCLK_DIV = 11	0.256	—	80	MHz
	FLL synchronizer input frequency	FLL1_SYNCCLK_DIV = 00	0.032	—	13.5	MHz
		FLL1_SYNCCLK_DIV = 01	0.064	—	27	MHz
		FLL1_SYNCCLK_DIV = 10	0.128	—	54	MHz
		FLL1_SYNCCLK_DIV = 11	0.256	—	80	MHz
Internal clocking	SYSCLK frequency	SYSCLK_FREQ = 000, SYSCLK_FRAC = 0	–1%	6.144	+1%	MHz
		SYSCLK_FREQ = 000, SYSCLK_FRAC = 1	–1%	5.6448	+1%	MHz
		SYSCLK_FREQ = 001, SYSCLK_FRAC = 0	–1%	12.288	+1%	MHz
		SYSCLK_FREQ = 001, SYSCLK_FRAC = 1	–1%	11.2896	+1%	MHz
		SYSCLK_FREQ = 010, SYSCLK_FRAC = 0	–1%	24.576	+1%	MHz
		SYSCLK_FREQ = 010, SYSCLK_FRAC = 1	–1%	22.5792	+1%	MHz
		SYSCLK_FREQ = 011, SYSCLK_FRAC = 0	–1%	49.152	+1%	MHz
		SYSCLK_FREQ = 011, SYSCLK_FRAC = 1	–1%	45.1584	+1%	MHz
		SYSCLK_FREQ = 100, SYSCLK_FRAC = 0	–1%	98.304	+1%	MHz
		SYSCLK_FREQ = 100, SYSCLK_FRAC = 1	–1%	90.3168	+1%	MHz
	DSPCLK frequency		5	—	150	MHz

1. If MCLK1 or MCLK2 is selected as a source for SYSCLK (either directly or via the FLL), the frequency must be within 1% of the SYSCLK_FREQ setting.

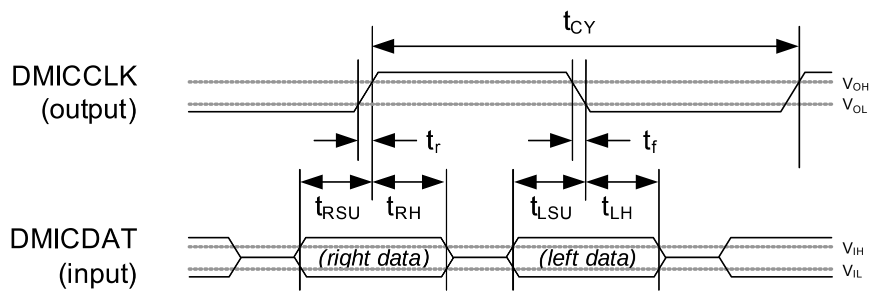
Table 3-14. Digital Microphone (DMIC) Interface Timing

The following timing information is valid across the full range of recommended operating conditions.

Parameter [1,2]	Symbol	Minimum	Typical	Maximum	Units
DMICCLK cycle time	t_{CY}	160	163	1432	ns
DMICCLK duty cycle	—	45	—	55	%
DMICCLK rise/fall time (25-pF load, 1.8-V supply)	t_r, t_f	5	—	30	ns
DMICDAT (Left) setup time to falling DMICCLK edge	t_{LSU}	15	—	—	ns
DMICDAT (Left) hold time from falling DMICCLK edge	t_{LH}	0	—	—	ns
DMICDAT (Right) setup time to rising DMICCLK edge	t_{RSU}	15	—	—	ns
DMICDAT (Right) hold time from rising DMICCLK edge	t_{RH}	0	—	—	ns

Note: The voltage reference for the IN1 interface is selectable, using the IN1_DMIC_SUP field—the interface is referenced to MICVDD or MICBIAS1.

1. DMIC interface timing



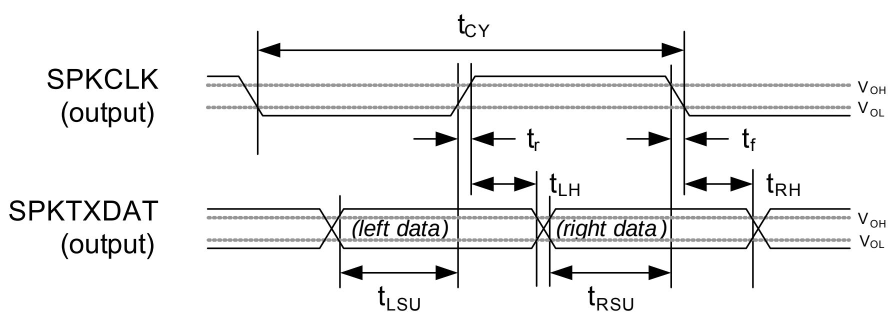
2. If the SPKRXDAT pin is configured for digital input, the SPKRXDAT timing requirements (with respect to SPKCLK) are the same as the DMICDAT timing requirements (with respect to DMICCLK).

Table 3-15. Digital Speaker (PDM) Interface Timing

The following timing information is valid across the full range of recommended operating conditions.

Parameter		Symbol	Minimum	Typical	Maximum	Units
Mode A [1]	SPKCLK cycle time	t_{CY}	160	163	358	ns
	SPKCLK duty cycle	—	45	—	55	%
	SPKCLK rise/fall time (25-pF load)	t_r, t_f	2	—	8	ns
	SPKTXDAT set-up time to SPKCLK rising edge (left channel)	t_{LSU}	30	—	—	ns
	SPKTXDAT hold time from SPKCLK rising edge (left channel)	t_{LH}	30	—	—	ns
	SPKTXDAT set-up time to SPKCLK falling edge (right channel)	t_{RSU}	30	—	—	ns
	SPKTXDAT hold time from SPKCLK falling edge (right channel)	t_{RH}	30	—	—	ns
Mode B [2]	SPKCLK cycle time	t_{CY}	160	163	358	ns
	SPKCLK duty cycle	—	45	—	55	%
	SPKCLK rise/fall time (25-pF load)	t_r, t_f	2	—	8	ns
	SPKTXDAT enable from SPKCLK rising edge (right channel)	t_{REN}	—	—	15	ns
	SPKTXDAT disable to SPKCLK falling edge (right channel)	t_{RDIS}	—	—	5	ns
	SPKTXDAT enable from SPKCLK falling edge (left channel)	t_{LEN}	—	—	15	ns
	SPKTXDAT disable to SPKCLK rising edge (left channel)	t_{LDIS}	—	—	5	ns

1. Digital speaker (PDM) interface timing—Mode A



2. Digital speaker (PDM) interface timing—Mode B

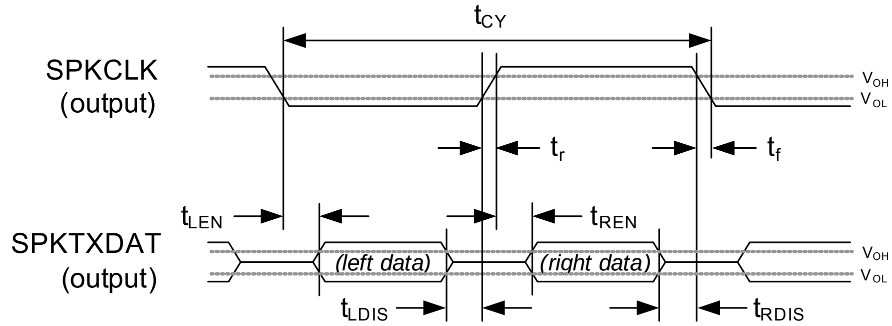


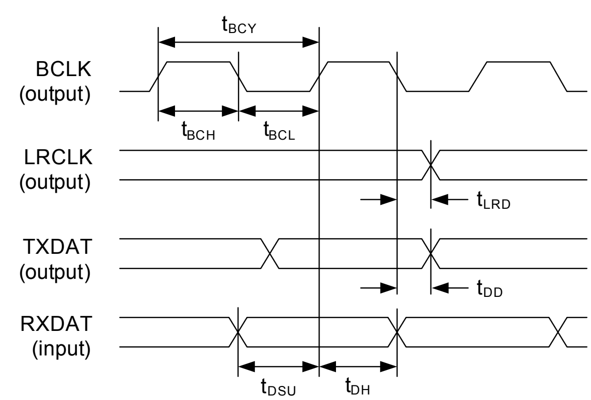
Table 3-16. Digital Audio Interface—Master Mode

Test conditions (unless specified otherwise): C_{LOAD} = 25 pF (output pins); BCLK slew (10% to 90%) = 3.7–5.6 ns; with the exception of the conditions noted, the following electrical characteristics are valid across the full range of recommended operating conditions.

Parameter [1]		Symbol	Minimum	Typical	Maximum	Units
Master Mode	AIF*n*BCLK cycle time	t_{BCY}	40	—	—	ns
	AIF*n*BCLK pulse width high	t_{BCH}	18	—	—	ns
	AIF*n*BCLK pulse width low	t_{BCL}	18	—	—	ns
	AIF*n*LRCLK propagation delay from BCLK falling edge [2]	t_{LRD}	0	—	8.3	ns
	AIF*n*TXDAT propagation delay from BCLK falling edge	t_{DD}	0	—	5	ns
	AIF*n*RXDAT setup time to BCLK rising edge	t_{DSU}	11	—	—	ns
	AIF*n*RXDAT hold time from BCLK rising edge	t_{DH}	0	—	—	ns
Master Mode, Slave LRCLK	AIF*n*LRCLK setup time to BCLK rising edge	t_{LRSU}	14	—	—	ns
	AIF*n*LRCLK hold time from BCLK rising edge	t_{LRH}	0	—	—	ns

Note: The descriptions above assume noninverted polarity of AIF*n*BCLK.

1. Digital audio interface timing—Master Mode. Note that BCLK and LRCLK outputs can be inverted if required; the figure shows the default, noninverted polarity.



2. The timing of the AIF*n*LRCLK signal is selectable. If the LRCLK advance option is enabled, the LRCLK transition is timed relative to the preceding BCLK edge. Under the required condition that BCLK is inverted in this case, the LRCLK transition is still timed relative to the falling BCLK edge.

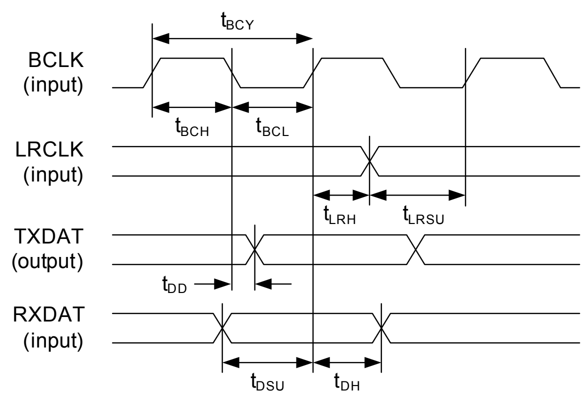
Table 3-17. Digital Audio Interface—Slave Mode

The following timing information is valid across the full range of recommended operating conditions, unless otherwise noted.

Parameter [1,2]		Symbol	Min	Typ	Max	Units
AIF*n*BCLK cycle time		t_{BCY}	40	—	—	ns
AIF*n*BCLK pulse width high	BCLK as direct SYSCLK source	t_{BCH}	16	—	—	ns
	All other conditions	t_{BCH}	14	—	—	ns
AIF*n*BCLK pulse width low	BCLK as direct SYSCLK source	t_{BCL}	16	—	—	ns
	All other conditions	t_{BCL}	14	—	—	ns
C_{LOAD} = 15 pF (output pins), BCLK slew (10%–90%) = 3 ns	AIF*n*LRCLK set-up time to BCLK rising edge	t_{LRSU}	7	—	—	ns
	AIF*n*LRCLK hold time from BCLK rising edge	t_{LRH}	0	—	—	ns
	AIF*n*TXDAT propagation delay from BCLK falling edge	t_{DD}	0	—	12.2	ns
	AIF*n*RXDAT set-up time to BCLK rising edge	t_{DSU}	2	—		ns
	AIF*n*RXDAT hold time from BCLK rising edge	t_{DH}	0	—		ns
	Master LRCLK, AIF*n*LRCLK propagation delay from BCLK falling edge	t_{LRD}	—	—	14.8	ns
C_{LOAD} = 25 pF (output pins), BCLK slew (10%–90%) = 6 ns	AIF*n*LRCLK set-up time to BCLK rising edge	t_{LRSU}	7	—		ns
	AIF*n*LRCLK hold time from BCLK rising edge	t_{LRH}	0	—		ns
	AIF*n*TXDAT propagation delay from BCLK falling edge	t_{DD}	0	—	14.2	ns
	AIF*n*RXDAT set-up time to BCLK rising edge	t_{DSU}	2	—		ns
	AIF*n*RXDAT hold time from BCLK rising edge	t_{DH}	0	—		ns
	Master LRCLK, AIF*n*LRCLK propagation delay from BCLK falling edge	t_{LRD}	—	—	15.9	ns

Note: The descriptions above assume noninverted polarity of AIF*n*BCLK.

1. Digital audio interface timing—Slave Mode. Note that BCLK and LRCLK inputs can be inverted if required; the figure shows the default, noninverted polarity.



2. If AIF*n*BCLK or AIF*n*LRCLK is selected as a source for SYSCLK (either directly or via the FLL), the frequency must be within 1% of the SYSCLK_FREQ setting.

Table 3-18. Digital Audio Interface Timing—TDM Mode

The following timing information is valid across the full range of recommended operating conditions, unless otherwise noted.

Parameter [1]		Min	Typ	Max	Units
Master Mode—C_{LOAD} (AIF*n*TXDAT) = 15 to 25 pF. BCLK slew (10%–90%) = 3.7ns to 5.6 ns.	AIF*n*TXDAT enable time from BCLK falling edge	0	—	—	ns
	AIF*n*TXDAT disable time from BCLK falling edge	—	—	6	ns
Slave Mode—C_{LOAD} (AIF*n*TXDAT) = 15 pF). BCLK slew (10%–90%) = 3 ns	AIF*n*TXDAT enable time from BCLK falling edge	2	—	—	ns
	AIF*n*TXDAT disable time from BCLK falling edge	—	—	12.2	ns
Slave Mode—C_{LOAD} (AIF*n*TXDAT) = 25 pF). BCLK slew (10%–90%) = 6 ns	AIF*n*TXDAT enable time from BCLK falling edge	2	—	—	ns
	AIF*n*TXDAT disable time from BCLK falling edge	—	—	14.2	ns

Note: If TDM operation is used on the AIF*n*TXDAT pins, it is important that two devices do not attempt to drive the AIF*n*TXDAT pin simultaneously. To support this requirement, the AIF*n*TXDAT pins can be configured to be tristated when not outputting data.

1. Digital audio interface timing—TDM Mode. The timing of the AIF*n*TXDAT tristating at the start and end of the data transmission is shown.

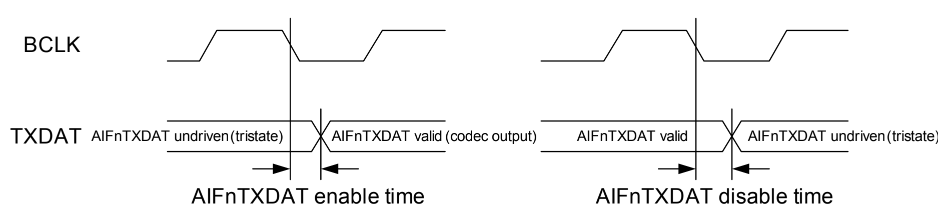


Table 3-19. Control Interface Timing—Two-Wire (I²C) Mode

The following timing information is valid across the full range of recommended operating conditions.

Parameter [1]		Symbol	Min	Typ	Max	Units
SCLK Frequency		—	—	—	3400	kHz
SCLK Low Pulse-Width		t_1	160	—	—	ns
SCLK High Pulse-Width		t_2	100	—	—	ns
Hold Time (Start Condition)		t_3	160	—	—	ns
Setup Time (Start Condition)		t_4	160	—	—	ns
SDA, SCLK Rise Time (10%–90%)	SCLK frequency > 1.7MHz SCLK frequency > 1MHz SCLK frequency ≤ 1MHz	t_6 t_6 t_6	— — —	— — —	80 160 2000	ns ns ns
SDA, SCLK Fall Time (90%–10%)	SCLK frequency > 1.7MHz SCLK frequency > 1MHz SCLK frequency ≤ 1MHz	t_7 t_7 t_7	— — —	— — —	60 160 200	ns ns ns
Setup Time (Stop Condition)		t_8	160	—	—	ns
SDA Setup Time (data input)		t_5	40	—	—	ns
SDA Hold Time (data input)		t_9	0	—	—	ns
SDA Valid Time (data/ACK output)	SCLK slew (90%–10%) = 20ns, C_{LOAD} (SDA) = 15 pF SCLK slew (90%–10%) = 60ns, C_{LOAD} (SDA) = 100 pF SCLK slew (90%–10%) = 160ns, C_{LOAD} (SDA) = 400 pF SCLK slew (90%–10%) = 200ns, C_{LOAD} (SDA) = 550 pF	t_{10} t_{10} t_{10} t_{10}	— — — —	— — — —	40 130 190 220	ns ns ns ns
Pulse width of spikes that are suppressed		t_{ps}	0	—	25	ns

1. Control interface timing—I²C Mode

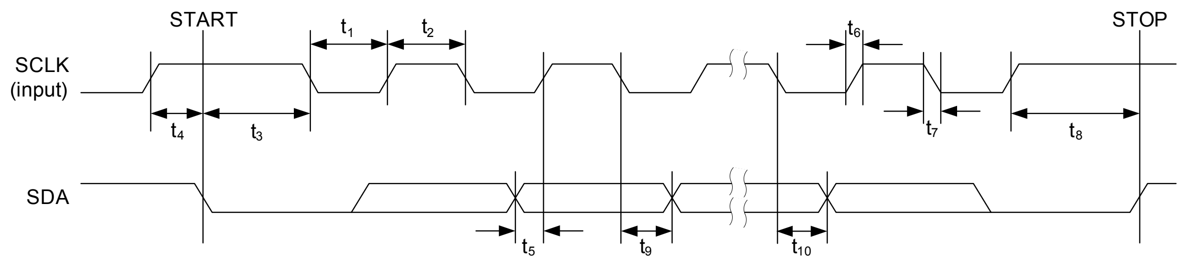
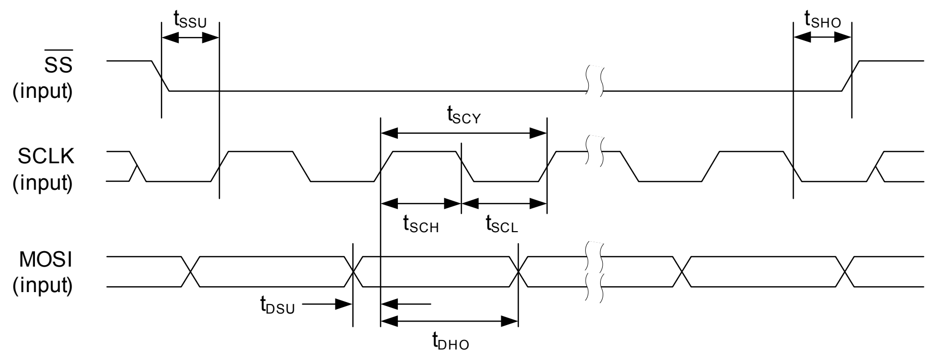


Table 3-20. Control Interface Timing—Four-Wire (SPI) Mode

The following timing information is valid across the full range of recommended operating conditions.

Parameter [1, 2]		Symbol	Min	Typ	Max	Units
$\overline{SS}$ falling edge to SCLK rising edge		t_{SSU}	2.6	—	—	ns
SCLK falling edge to $\overline{SS}$ rising edge		t_{SHO}	0	—	—	ns
SCLK pulse cycle time	SYSCLK disabled (SYSCLK_ENA = 0)	t_{SCY}	38.4	—	—	ns
	SYSCLK_ENA = 1, SYSCLK_FREQ = 000	t_{SCY}	76.8	—	—	ns
	SYSCLK_ENA = 1, SYSCLK_FREQ > 000	t_{SCY}	38.4	—	—	ns
SCLK pulse width low		t_{SCL}	15.3	—	—	ns
SCLK pulse width high		t_{SCH}	15.3	—	—	ns
MOSI to SCLK set-up time		t_{DSU}	1.5	—	—	ns
MOSI to SCLK hold time		t_{DHO}	1.7	—	—	ns
SCLK falling edge to MISO transition	SCLK slew (90%–10%) = 5 ns, C_{LOAD} (MISO) = 25 pF	t_{DL}	0	—	12.6	ns

1. Control interface timing—SPI Mode (write cycle)



2. Control interface timing—SPI Mode (read cycle)

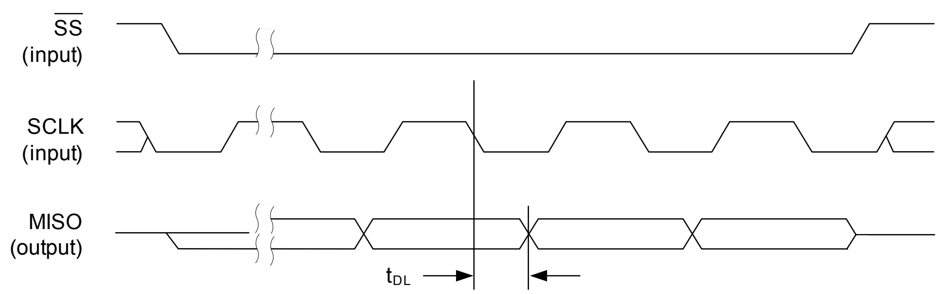


Table 3-21. Master Interface Timing—SPI Master

The following timing information is valid across the full range of recommended operating conditions.

Parameter [1]		Symbol	Min	Typ	Max	Units
$\overline{SS}$ falling edge to SCLK rising edge		t_{SSU}	13.88	—	—	ns
SCLK falling edge to $\overline{SS}$ rising edge		t_{SHO}	0	—	—	ns
SCLK pulse cycle time		t_{SCY}	27.77	—	—	ns
SCLK pulse width low		t_{SCL}	13.88	—	—	ns
SCLK pulse width high		t_{SCH}	13.88	—	—	ns
SCLK falling edge to MOSI transition	SCLK slew (90%–10%) = 5 ns, C_{LOAD} (MOSI) = 25 pF	t_{DL}	0	—	8.88	ns
MISO to SCLK set-up time		t_{DSU}	5	—	—	ns
MISO to SCLK hold time		t_{DHO}	5	—	—	ns

1. Master interface timing—SPI read cycle

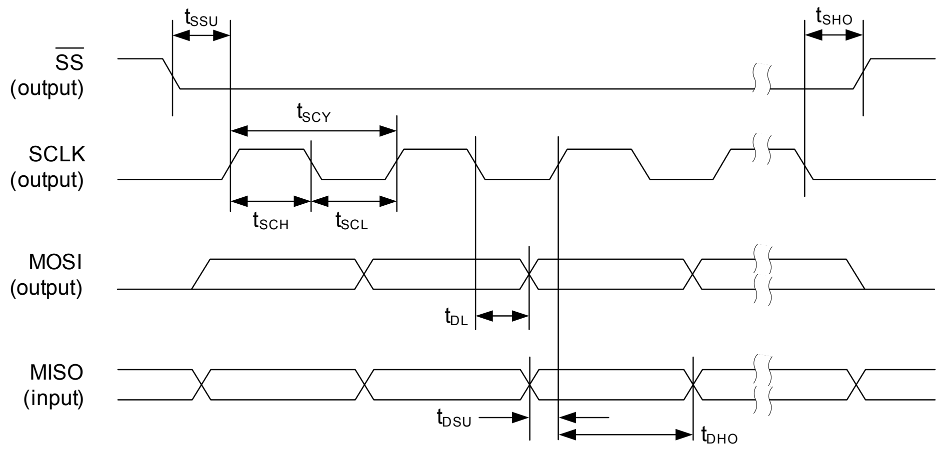


Table 3-22. JTAG Interface Timing

Test conditions (unless specified otherwise): C_{LOAD} = 25 pF (output pins); TCK slew (20%–80%) = 5 ns; with the exception of the conditions noted, the following electrical characteristics are valid across the full range of recommended operating conditions.

Parameter [1]	Symbol	Minimum	Typical	Maximum	Units
TCK cycle time	T_{CCY}	50	—	—	ns
TCK pulse width high	T_{CCH}	20	—	—	ns
TCK pulse width low	T_{CCL}	20	—	—	ns
TMS setup time to TCK rising edge	T_{MSU}	1	—	—	ns
TMS hold time from TCK rising edge	T_{MH}	2	—	—	ns
TDI setup time to TCK rising edge	T_{DSU}	1	—	—	ns
TDI hold time from TCK rising edge	T_{DH}	2	—	—	ns
TDO propagation delay from TCK falling edge	T_{DD}	0	—	17	ns
TRST setup time to TCK rising edge	T_{RSU}	3	—	—	ns
TRST hold time from TCK rising edge	T_{RH}	3	—	—	ns
TRST pulse width low	—	20	—	—	ns

1.JTAG Interface timing

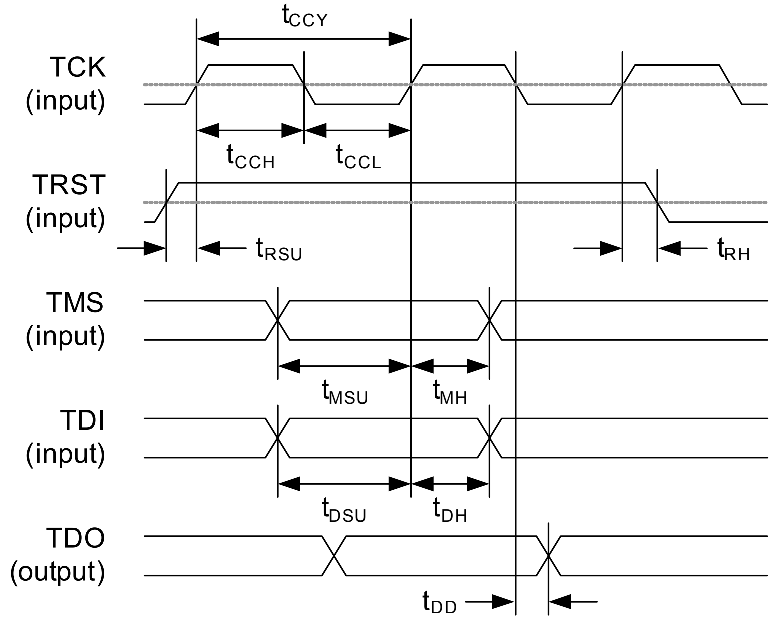


Table 3-23. Typical Power Consumption

Test conditions (unless specified otherwise): DBVDD = CPVDD = AVDD = 1.8 V, DCVDD = 1.2 V; MICVDD = 2.5 V; SPKVDD = 4.2 V; T_A = +25ºC; Fs = 48 kHz; 24-bit audio data, I^2S Slave Mode; SYSCLK = 24.576 MHz (direct MCLK1 input).

Operating Configuration		Typical $I_{1.2V}$ (mA)	Typical $I_{1.8V}$ (mA)	Typical $I_{2.5V}$ (mA)	Typical $I_{4.2V}$ (mA)	P_{TOT} (mW)
Headphone playback—AIF1 to DAC to HPOUT (stereo), 32-Ω load.	Quiescent	0.78	0.92	0.001	0.00	2.59
	1-kHz sine wave, P_O = 0.1 mW	0.87	3.6	0.001	0.00	7.6
Earpiece playback—AIF1 to DAC to EPOUT, 32-Ω load (BTL).	Quiescent	0.59	0.94	0.001	0.00	2.40
	1-kHz sine wave, P_O = 30 mW	0.62	61.68	0.001	0.00	112
Speaker playback—AIF1 to DAC to SPKOUT, 8-Ω, 22-µH load.	Quiescent	0.61	1.18	0.001	0.13	3.40
	1-kHz sine wave, P_O = 700 mW	0.66	1.18	0.001	187	790
Stereo line record—Analog line to ADC to AIF1	1-kHz sine wave, –1 dBFS output	1.11	2.22	0.001	0.00	5.33
Sleep Mode	Accessory detect enabled (JD1_ENA = 1)	0.000	0.014	0.000	0.000	0.025

Table 3-24. Typical Signal Latency

Test conditions (unless specified otherwise): DBVDD = CPVDD = AVDD = 1.8 V, DCVDD = 1.2 V; MICVDD = 2.5 V; SPKVDD = 4.2 V; T_A = +25ºC; Fs = 48 kHz; 24-bit audio data, I^2S Slave Mode; SYSCLK = 24.576 MHz (direct MCLK1 input).

Operating Configuration			Latency (µs)
AIF to DAC path	Digital input (AIFn) to analog output (HPOUT).	48 kHz input, 48 kHz output, Synchronous	332
		44.1 kHz input, 44.1 kHz output, Synchronous	358
		16 kHz input, 16 kHz output, Synchronous	550
		8 kHz input, 8 kHz output, Synchronous	1076
		8 kHz input, 48 kHz output, Isochronous [1]	1717
		16 kHz input, 48 kHz output, Isochronous [1]	1041
ADC to AIF path	Analog input (INn) to digital output (AIFn).[2]	48 kHz input, 48 kHz output, Synchronous	219
		44.1 kHz input, 44.1 kHz output, Synchronous	234
		16 kHz input, 16 kHz output, Synchronous	654
		8 kHz input, 8 kHz output, Synchronous	1323
		8 kHz input, 48 kHz output, Isochronous [1]	1802
		16 kHz input, 48 kHz output, Isochronous [1]	994

1. Signal is routed via the ISRC function in the isochronous cases only.
2. Digital core high-pass filter is included in the signal path

4 Functional Description

The CS47L15 is a highly integrated, low-power audio hub codec for mobile telephony, media players and wearable technology devices. It provides flexible, high-performance audio interfacing for handheld devices in a small and cost-effective package. It also provides exceptional levels of performance and signal-processing capability, suitable for a wide variety of mobile and handheld applications.

4.1 Overview

The CS47L15 block diagram is shown in Fig. 4-1.

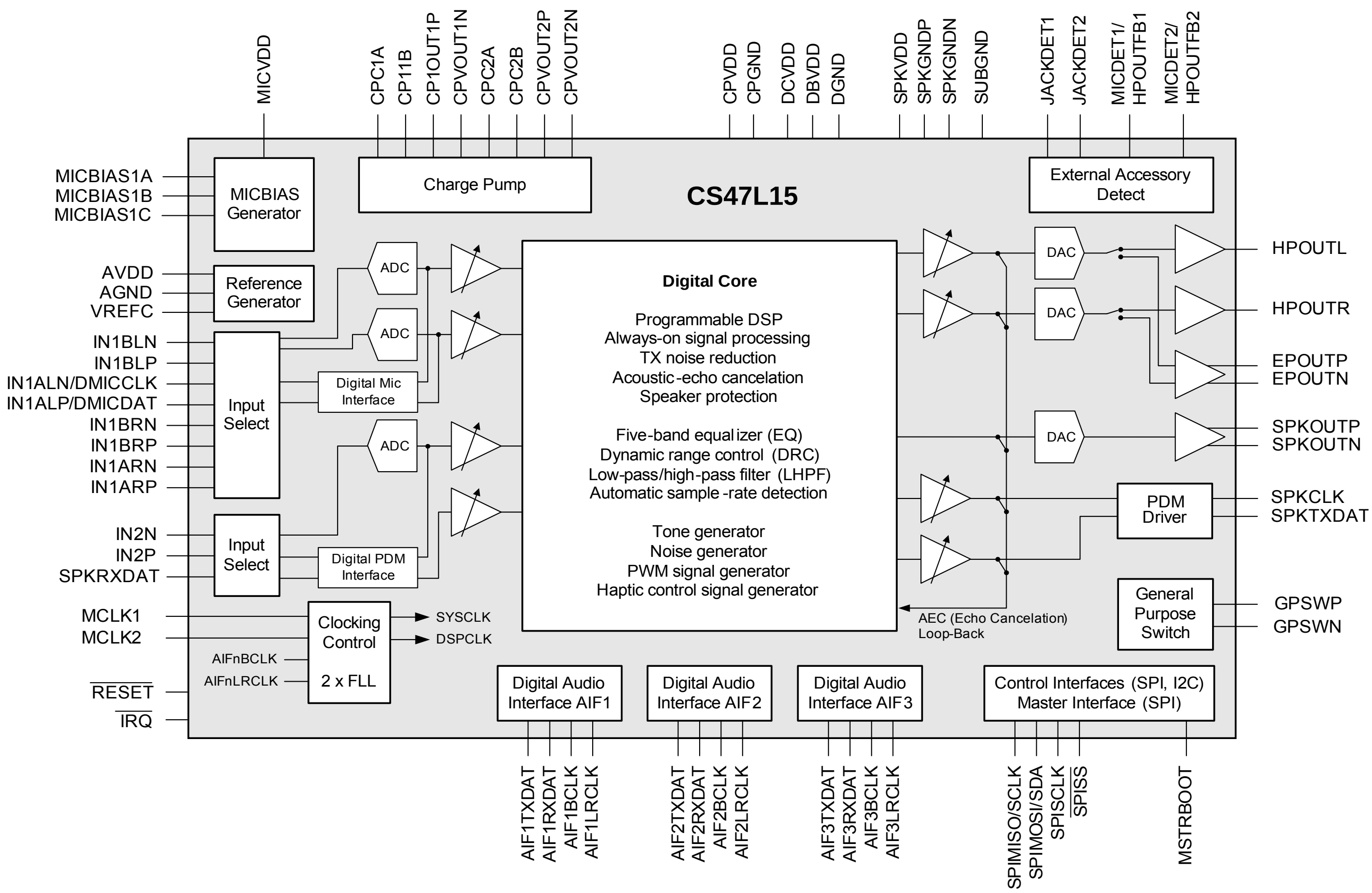


Figure 4-1. CS47L15 Block Diagram

The CS47L15 digital core provides a flexible capability for signal-processing algorithms, including transmit (TX) path noise reduction, acoustic-echo cancelation (AEC), and other programmable filters. Low-power analog and digital interfaces provide additional support for always-on voice applications and speaker-protection algorithms implemented on the DSP core. The DSP is supported by integrated general-purpose timers and event-logger functions. The DSP is ideally suited to the Cirrus Logic® SoundClear® suite of audio processing algorithms, such as the SoundClear Control always-on voice control software.

The CS47L15 digital core supports audio enhancements, such as dynamic range control (DRC) and multiband compression (MBC). Highly flexible digital mixing, including stereo full-duplex isochronous sample-rate conversion, provides use-case flexibility across a broad range of system architectures. A signal generator for controlling haptics vibe actuators is included.

The CS47L15 provides multiple digital audio interfaces to provide independent isochronous connections to different processors (e.g., application processor, baseband processor, and wireless transceiver). The DACs and output paths support high definition audio throughout the entire signal chain, enabling studio-quality playback without loss of detail or bandwidth.

A flexible clocking arrangement supports a wide variety of external clock references, including clocking derived from the digital audio interface. Two frequency-locked loop (FLL) circuits provide additional flexibility for system clocking, including low-power always-on operation. Seamless switching between clock sources is supported, and free-running modes are also available.

Unused circuitry can be disabled under software control to save power; low leakage currents enable extended standby/off time in portable battery-powered applications. The CS47L15 always-on circuitry can be used in conjunction with the Apps Processor to wake up the device following a headphone jack-detection event.

An SPI master interface is incorporated, enabling autonomous boot-up and configuration using an external non-volatile memory (e.g., EEPROM or flash memory). Versatile GPIO functionality is provided, including support for external accessory/push-button detection inputs. The CS47L15 also provides comprehensive interrupt functions, with status reporting.

4.1.1 Hi-Fi Audio Codec

The CS47L15 is a high-performance, low-power audio codec that uses a simple analog architecture. Three ADCs are incorporated, with multiplexers to support up to five analog inputs. Three DACs are incorporated, with two being switchable between the headphone and BTL-earpiece analog output paths.

Five analog inputs are provided (multiplexed into three input channels), supporting single-ended or differential input modes. As many as four analog microphone connections can be supported; a separate analog input channel is provided for use in speaker-protection applications. In differential input mode, SNR performance of 104 dB is supported (16 kHz sample rate, i.e., wideband voice mode). The ADC input paths can be bypassed, supporting up to four channels of digital (e.g., DMIC) input.

The analog outputs comprise a stereo headphone amplifier with ground-referenced output (30-mW per channel, 127 dB SNR), a mono (BTL) earpiece driver, and a mono Class D speaker driver capable of delivering 2.5 W into a 4-Ω load.

The CS47L15 output drivers are designed to support a range of different system architectures. Each output path supports independent signal mixing, equalization, filtering, and gain controls. This allows each signal path to be individually tailored for the load characteristics. All outputs have integrated pop and click suppression features.

The headphone and earpiece output drivers are ground-referenced, powered from an integrated charge pump, enabling high quality, power efficient headphone playback without any requirement for DC blocking capacitors. Ground loop feedback is incorporated, providing rejection of noise on the ground connections. Full support for high definition audio is provided throughout the entire signal chain from the digital audio interfaces through to the analog output.

The Class D speaker driver delivers excellent power efficiency. Speaker protection software is supported within the DSP core, enabling maximum audio output without risk of damage to the external speaker. High PSRR, low leakage and optimized supply voltage ranges enable powering from switching regulators or directly from the battery. Battery current consumption is minimized across a wide variety of voice communication and multimedia playback use cases.

The CS47L15 is cost optimized for a wide range of mobile applications, and incorporates a mono Class D power amplifier. For applications requiring more than one channel of power amplification (or when using the integrated Class D path to drive a haptics actuator), the PDM output channels can be used to drive external PDM-input speaker drivers. The PDM outputs can ease layout and electromagnetic compatibility by avoiding the need to run the Class D speaker output over a long distance and across interconnects.

4.1.2 Digital Audio Core

The CS47L15 uses a core architecture based on all-digital signal routing, making digital audio effects available on all signal paths, regardless of whether the source data input is analog or digital. The digital mixing desk allows different audio effects to be applied simultaneously on many independent paths, while supporting a variety of sample rates. A soft mute/unmute control ensures smooth transitions between use cases without interruption to other audio streams.

The CS47L15 digital core provides an extensive capability for programmable signal-processing algorithms. The SoundClear suite of software algorithms enable advanced audio features, such as transmit (TX) path noise reduction, AEC, wind-noise reduction, speech enhancement, karaoke, and other programmable filters. The DSP core is supported by peripheral timer and event logging functions, which provide additional capability for signal-processing applications. Audio enhancements such as DRC and MBC are also supported.

The CS47L15 is ideal for mobile telephony, providing enhanced voice communication quality for both near-end and far-end users in a wide variety of applications. The SoundClear Control voice command recognition software is supported, for low-power always-on features. Speaker Protection software is available, using analog or digital input paths to support current monitoring in the speaker output—this allows the Class D output to be optimized for the operational limits of the speaker, and enables maximum audio output while ensuring the loudspeakers are fully protected from damage.

The digital audio core incorporates a highly flexible digital mixing capability, including mixing between audio interfaces. The CS47L15 performs multichannel full-duplex isochronous sample-rate conversion, providing use-case flexibility across a broad range of system architectures. Automatic sample-rate detection is provided, enabling seamless wideband/narrowband voice call handover.

DRC functions are available for optimizing audio signal levels. In playback modes, the DRC can be used to maximize loudness, while limiting the signal level to avoid distortion, clipping, or battery droop, for high-power output drivers such as speaker amplifiers. In record modes, the DRC assists in applications where the signal level is unpredictable.

The five-band parametric EQ functions can be used to compensate for the frequency characteristics of the output transducers. EQ functions can be cascaded to provide additional frequency control. Programmable high-pass and low-pass filters are also available for general filtering applications, such as removal of wind and other low-frequency noise.

4.1.3 Digital Interfaces

Three serial digital audio interfaces (AIFs) each support PCM, TDM, and I^{2}S data formats for compatibility with most industry-standard chipsets. AIF1 supports six input/output channels; AIF2 supports four input/output channels; AIF3 supports two input/output channels. Bidirectional operation at sample rates up to 192 kHz is supported.

Four digital PDM input channels are available (two stereo interfaces). The IN1 digital input path is suitable for use with digital microphones, powered from the integrated MICBIAS power-supply regulator. Two PDM output channels are also available (one stereo interface); these are typically used for external power amplifiers. The IN2 digital input (SPKRXDAT) is synchronized to the PDM output interface, creating a bidirectional audio interface suitable for speaker-protection algorithms, using digital feedback from the external amplifier.

An IEC-60958-3–compatible S/PDIF transmitter is incorporated, enabling stereo S/PDIF output on a GPIO pin. Standard S/PDIF sample rates of 32–192 kHz are supported.

Control register access and high bandwidth data transfer are supported by a slave SPI/I^{2}C control interface. The slave interface operates up to 26 MHz in SPI Mode, or up to 3.4 MHz in I^{2}C Mode. The CS47L15 also supports an SPI master interface that can be used to download firmware and register-configuration data from an external non-volatile memory (e.g., EEPROM or flash memory).

4.1.4 Other Features

The CS47L15 supports autonomous boot-up and configuration from an external non-volatile memory. This enables the device to self-boot to an application-specific configuration and to be used independently of a host processor. The interface to the external memory is supported via the CS47L15 control interface, operating in SPI Master Mode.

The CS47L15 incorporates two 1-kHz tone generators that can be used for beep functions through any of the audio signal paths. The phase relationship between the two generators is configurable, providing flexibility in creating differential signals, or for test scenarios.

A white-noise generator is provided that can be routed within the digital core. The noise generator can provide comfort noise in cases where silence (digital mute) is not desirable.

Two pulse-width modulation (PWM) signal generators are incorporated. The duty cycle of each PWM signal can be modulated by an audio source or can be set to a fixed value using a control register setting. The PWM signal generators can be output directly on a GPIO pin.

The CS47L15 supports up to 15 GPIO pins, offering a range of input/output functions for interfacing, for detection of external hardware, and for providing logic outputs to other devices. The GPIOs are multiplexed with other functions. Comprehensive interrupt functionality is also provided for monitoring internal and external event conditions.

A signal generator for controlling haptics devices is included, compatible with both eccentric rotating mass (ERM) and linear resonant actuator (LRA) haptics devices. The haptics signal generator is highly configurable and can execute programmable drive event profiles, including reverse drive control. An external vibe actuator can be driven directly by the Class D speaker output.

A smart accessory interface is included, supporting most standard 3.5-mm accessories. Jack detection, accessory sensing, and impedance measurement is provided, for external accessory and push-button detection. Accessory detection can be used as a wake-up trigger from low-power standby. Microphone activity detection with interrupt is also available.

System clocking can be derived from the MCLK1 or MCLK2 input pins. Alternatively, the audio interfaces (configured in Slave Mode), can be used to provide a clock reference. The CS47L15 also provides two integrated FLL circuits for clock frequency conversion and stability. The flexible clocking architecture supports low-power always-on operation, with reference frequencies down to 32 kHz. Seamless switching between clock sources is supported; free-running FLL modes are also available.

The CS47L15 can be powered from 1.8- and 1.2-V external supplies. Separate MICVDD input can be supported (up to 3.6 V), for microphone operation above 1.8 V. A separate supply (4.2 V) is typically required for the Class D speaker driver.

4.2 Input Signal Path

The CS47L15 provides flexible input channels, supporting up to five analog inputs or up to four digital inputs. Selectable combinations of analog (mic or line) and digital inputs are multiplexed into two stereo input signal paths.

The IN1 signal paths support high performance analog and digital input modes. The analog paths support single-ended and differential input, programmable gain control, and are digitized using a high performance sigma-delta ADCs. The IN1 analog input paths can be configured for low-power operation, ideal for always-on applications. The digital paths connect directly to external digital microphones; the two-wire digital interface incorporates a dedicated clock source and supports stereo microphone operation.

The IN2 signal paths can be configured for analog or digital input modes. Mono analog (differential) input is supported; the analog configuration is optimized for low power operation and is ideally suited as an input path for speaker-protection applications. Stereo digital input can also be supported on the SPKRXDAT pin; the respective data input is synchronized with the digital speaker (PDM) output interface—these signal paths provide a bidirectional interface to an external speaker driver.

The microphone bias (MICBIAS) generator provides a low-noise reference for biasing electret condenser microphones (ECMs) or for use as a low-noise supply for MEMS microphones and digital microphones. Switchable outputs from the MICBIAS generator allows three separate reference/supply outputs to be independently controlled.

Digital volume control is available on all inputs (analog and digital), with programmable ramp control for smooth, glitch-free operation. A configurable signal-detect function is available on each input signal path.

The IN1 and IN2 signal paths and control fields are shown in Fig. 4-2.

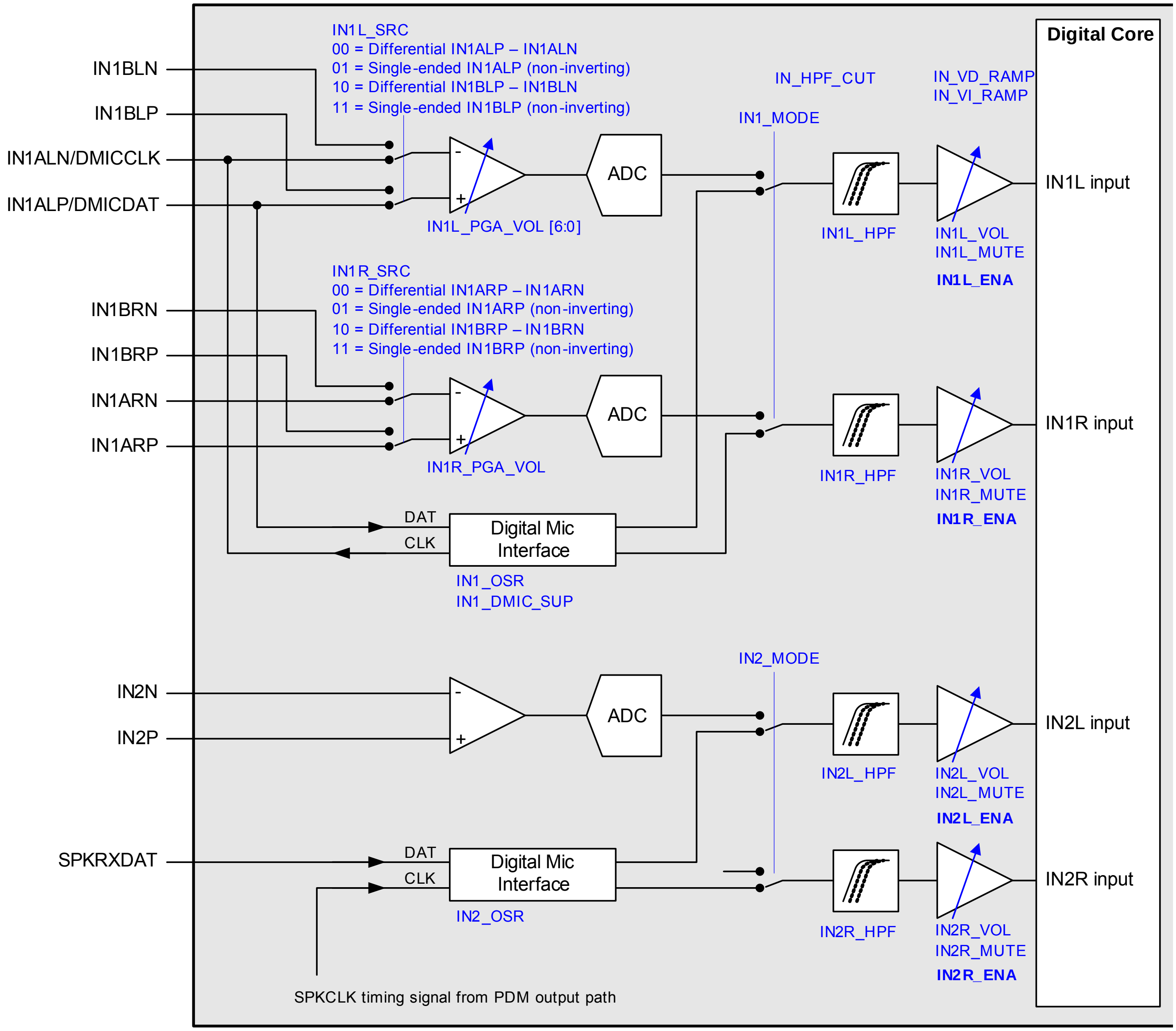


Figure 4-2. Input Signal Paths

4.2.1 Analog Microphone Input

Up to four analog microphones can be connected to the CS47L15, in single-ended or differential configuration. The input configuration and pin selection for the IN1 signal paths is controlled using IN1*x*_SRC, as described in Section 4.2.7.

Note: The IN2 analog input path is optimized for supporting speaker-protection applications. It is not suitable for connection to microphones.

The CS47L15 includes external accessory-detection circuits that can report the presence of a microphone and the status of a hook switch or other push buttons. When using this function, it is recommended to use the IN1BLP or IN1BRP analog microphone input paths to ensure best immunity to electrical transients arising from the push buttons.

For single-ended input, the microphone signal is connected to the noninverting input of the PGAs (IN1*x*P). The inverting inputs of the PGAs are connected to an internal reference in this configuration.

For differential input, the noninverted microphone signal is connected to the noninverting input of the PGAs (IN1*x*P), while the inverted (or noisy ground) signal is connected to the inverting input pins (IN1*x*N).

Note: Pseudodifferential connection is also possible—this is similar to the configuration shown in Fig. 4-4, but the GND connection is directly to the microphone (and IN1*x*N capacitor), instead of via a resistor. This is the recommended configuration if the external accessory detection functions on the CS47L15 are used. The IN1*x*_SRC field settings are the same for pseudodifferential connection as for differential.

The gain of the IN1 signal path PGAs is controlled via register settings, as defined in Section 4.2.7. Note that the input impedance of the analog input paths is fixed across all PGA gain settings.

The ECM analog input configurations are shown in Fig. 4-3 and Fig. 4-4. The integrated MICBIAS generator provides a low noise reference for biasing the ECMs.

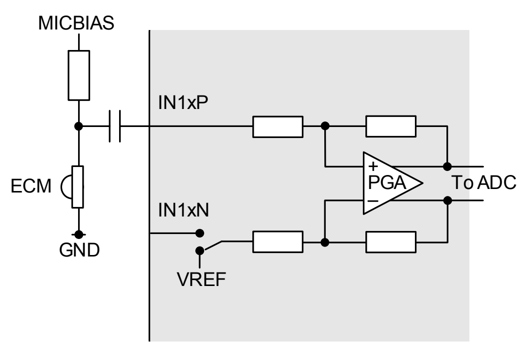


Figure 4-3. Single-Ended ECM Input

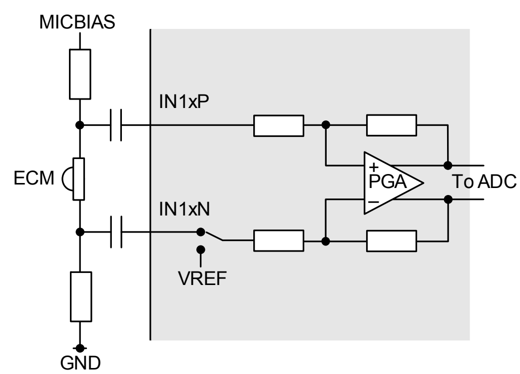


Figure 4-4. Differential ECM Input

Analog MEMS microphones can be connected to the CS47L15 in a similar manner to the ECM configurations. Typical configurations are shown in Fig. 4-5 and Fig. 4-6. In this configuration, the integrated MICBIAS generator provides a low-noise power supply for the microphones.

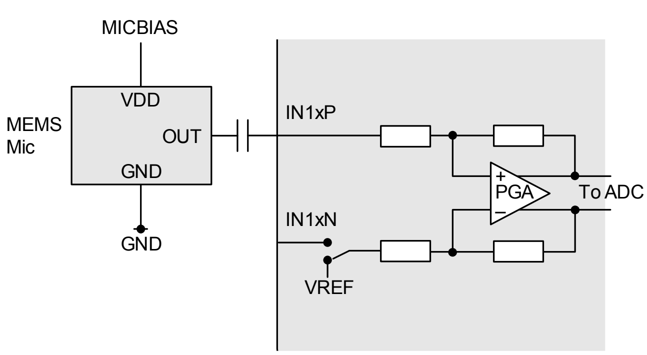


Figure 4-5. Single-Ended MEMS Input

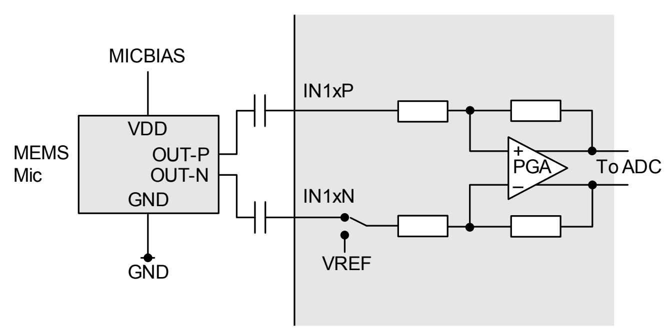


Figure 4-6. Differential MEMS Input

Note: It is also possible to use the MICVDD pin (instead of MICBIAS) as a reference or power supply for external microphones; the MICBIAS outputs are preferred because they offer better noise performance and independent enable/disable control.

4.2.2 Analog Line Input

Line input signals can be connected to the CS47L15 in a similar manner to the mic inputs.

Single-ended and differential configurations are supported on the IN1 pins, using the IN1*x*_SRC bits as described in Section 4.2.7. The IN1 analog line input configurations are shown in Fig. 4-7 and Fig. 4-8. Note that the microphone bias (MICBIAS) is not used for line input connections.

The gain of the IN1 signal path PGAs is controlled via register settings, as defined in Section 4.2.7. Note that the input impedance of the analog input paths is fixed across all PGA gain settings.

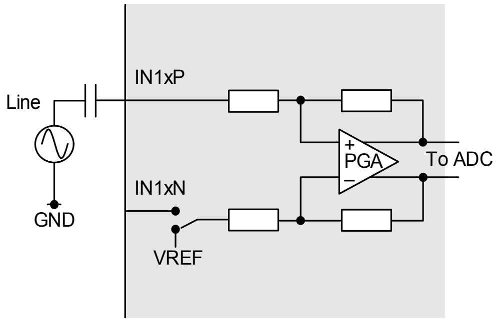


Figure 4-7. Single-Ended Line Input

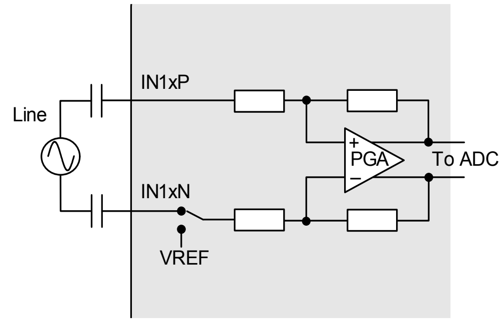


Figure 4-8. Differential Line Input

The IN2 analog input path supports differential connection only, as shown in Fig. 4-2. The IN2 analog line input configuration is shown in Fig. 4-9. The gain of the IN2 signal path PGA is fixed at 14 dB.

Note that IN2 analog input supports ground-referenced input signals only. Input capacitors must not be used on the IN2*x* pins.

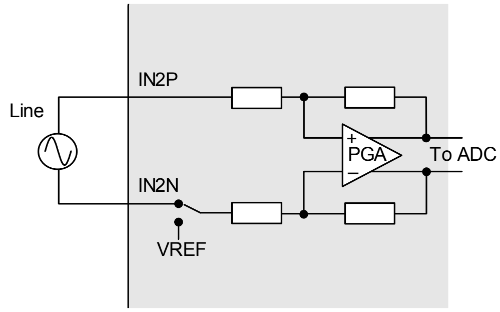


Figure 4-9. Differential Line Input

4.2.3 Analog Input—Speaker Current Monitoring

The IN2 analog input path is optimized for supporting speaker-protection applications. In these applications, the IN2 pins are used to provide feedback from current-monitoring connections on the Class D speaker outputs. Speaker-protection software, running on the integrated DSP core, enables the operational limits to be continually optimized for the particular loudspeaker and the prevailing conditions.

Typical connections for speaker-protection applications, including the analog feedback path to the IN2 pins, are shown in Fig. 4-10.

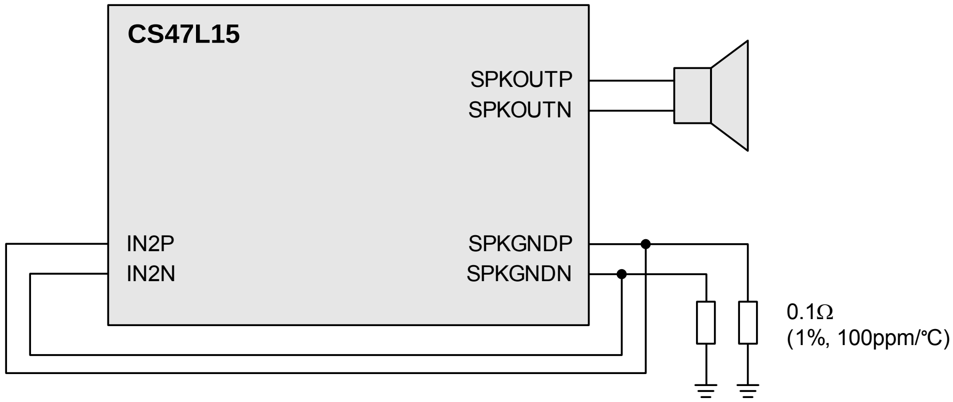


Figure 4-10. Speaker Current Monitoring Connection

See Section 4.8 for the details of the Class D speaker output.

4.2.4 Digital Input

The CS47L15 input signal paths support up to four channels of digital input—the IN1 and IN2 paths each support two digital input channels. Digital operation on input paths IN1 and IN2 is selected using IN*n*_MODE, as described in Section 4.2.7.

The IN1 (DMICDAT) and IN2 (SPKRXDAT) digital paths are described in Section 4.2.4.1 and Section 4.2.4.2 respectively.

4.2.4.1 IN1 Digital Input (DMICDAT)

The IN1 digital input path is designed to support digital microphone (DMIC) operation. In DMIC mode, two channels of audio data are multiplexed on the DMICDAT pin. If a DMIC input path is enabled, the CS47L15 outputs a clock signal on the DMICCLK pin—this is the timing reference for the DMICDAT input. The DMICCLK frequency is controlled by the IN1_OSR field, as described in Table 4-1 and Table 4-4.

Note that, if the 384- or 768-kHz DMICCLK frequency is selected for the DMIC input path, the maximum valid input path sample rate (all input paths) is restricted as described in Table 4-1.

The system clock, SYSCLK, must be present and enabled when using the DMICDAT input channels; see Section 4.13 for details regarding SYSCLK and the associated registers.

The DMICCLK frequencies in Table 4-1 assume that the SYSCLK frequency is a multiple of 6.144 MHz (SYSCLK_FRAC = 0). If the SYSCLK frequency is a multiple of 5.6448 MHz (SYSCLK_FRAC = 1), the DMICCLK frequencies are scaled accordingly.

Table 4-1. DMICCLK Frequency

Condition	DMICCLK Frequency	Valid Sample Rates	Signal Passband
IN1_OSR = 010	384 kHz	Up to 48 kHz	Up to 4 kHz
IN1_OSR = 011	768 kHz	Up to 96 kHz	Up to 8 kHz
IN1_OSR = 100	1.536 MHz	Up to 192 kHz	Up to 20 kHz
IN1_OSR = 101	3.072 MHz	Up to 192 kHz	Up to 20 kHz
IN1_OSR = 110	6.144 MHz	Up to 192 kHz	Up to 96 kHz

The voltage reference for the IN1 DMIC interface is selectable, using IN1_DMIC_SUP—the interface is referenced to MICVDD or MICBIAS1. The voltage reference selection should be set equal to the power supply of the respective microphones.

A pair of digital microphones is connected as shown in Fig. 4-11. The microphones must be configured to ensure that the left mic transmits a data bit when DMICCLK is high and the right mic transmits a data bit when DMICCLK is low. The CS47L15 samples the DMIC data at the end of each DMICCLK phase. Each microphone must tristate its data output when the other microphone is transmitting.

Note that the CS47L15 provides an integrated pull-down resistor on the DMICDAT pin. This provides a flexible capability for interfacing with other devices.

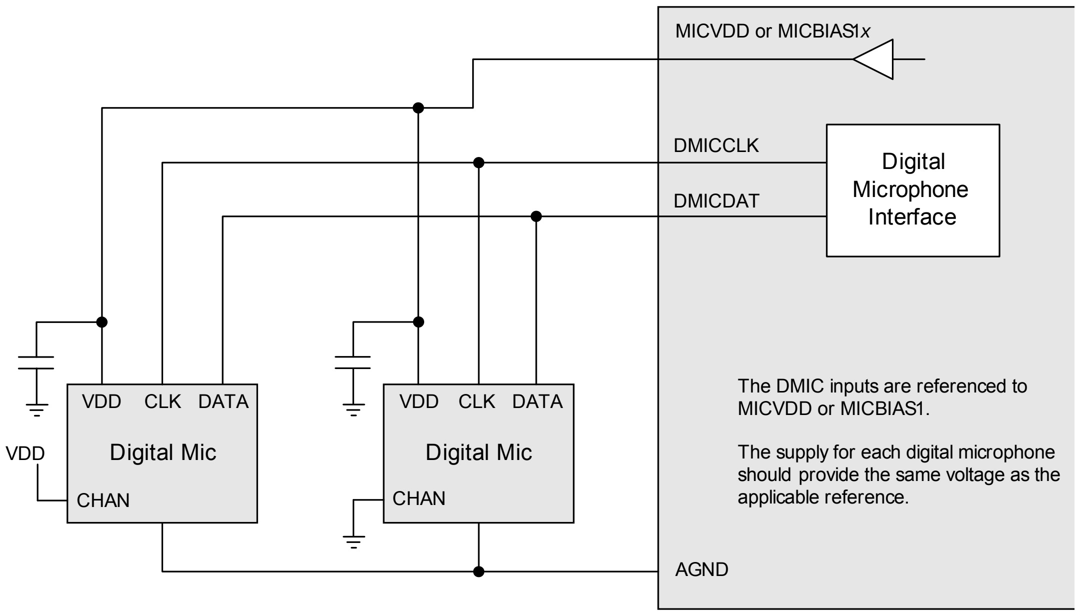


Figure 4-11. DMIC Input

Two DMIC channels are interleaved on DMICDAT. The DMIC interface timing is shown in Fig. 4-12. Each microphone must tristate its data output when the other microphone is transmitting. See Table 3-14 for a detailed timing specification of the DMIC interface.

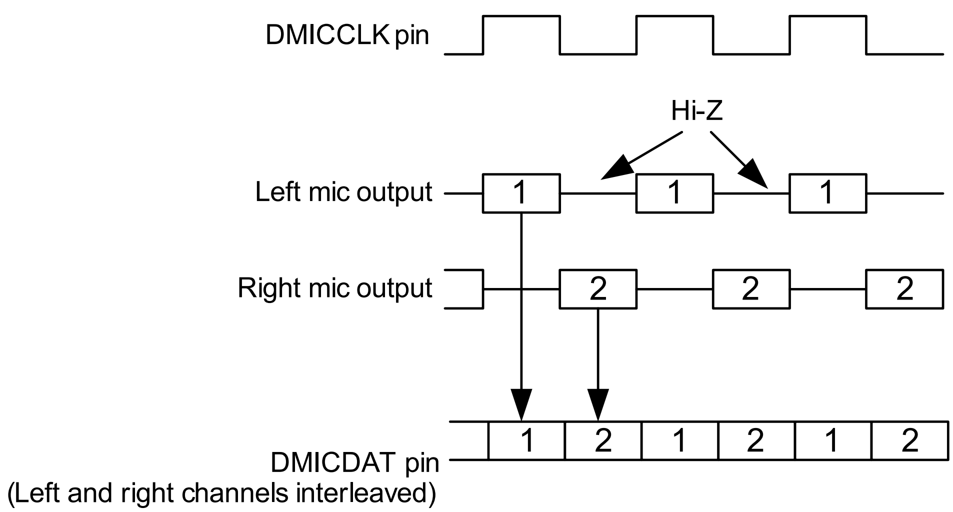


Figure 4-12. DMIC Interface Timing

4.2.4.2 IN2 Digital Input (SPKRXDAT)

The IN2 digital input path forms part of a bidirectional interface for external speaker drivers. If the IN2 path is configured for digital input, two channels of audio data are multiplexed on the SPKRXDAT pin. A timing reference signal is provided on the SPKCLK pin, which is common to the input (SPKRXDAT) and output (SPKTXDAT) paths of the digital speaker (PDM) interface.

The SPKCLK frequency is controlled using the OUT5_OSR field, as described in Table 4-55. The input signal timing is controlled by the IN2_OSR field—this field must be configured for the same frequency as the OUT5_OSR field.

The system clock, SYSCLK, must be present and enabled when using the SPKRXDAT input channels; see Section 4.13 for details regarding SYSCLK and the associated registers.

The SPKCLK frequencies in Table 4-2 assume that the SYSCLK frequency is a multiple of 6.144 MHz (SYSCLK_FRAC = 0). If the SYSCLK frequency is a multiple of 5.6448 MHz (SYSCLK_FRAC = 1), the SPKCLK frequencies are scaled accordingly.

Table 4-2. SPKCLK Frequency

Condition	SPKCLK Frequency	Valid Sample Rates	Signal Passband
IN2_OSR = 101	3.072 MHz	Up to 192 kHz	Up to 20 kHz
IN2_OSR = 110	6.144 MHz	Up to 192 kHz	Up to 96 kHz

Note: The SPKCLK frequency is controlled by the OUT5_OSR field (see Table 4-55). The descriptions shown here assume that the IN2_OSR and OUT5_OSR fields are configured for the same frequency.

The voltage reference for the IN2 digital input is DBVDD—this is the same voltage reference as the output pins of the digital speaker (PDM) interface.

Typical connections for an external speaker driver, incorporating the IN2 digital input (SPKRXDAT) path, are shown in Fig. 4-13. The left channel data is received when SPKCLK is high and the right channel data is received when SPKCLK is low. The CS47L15 samples the data at the end of each SPKCLK phase.

Note that the CS47L15 provides integrated pull-up and pull-down resistors on the SPKRXDAT pin. This provides a flexible capability for interfacing with other devices.

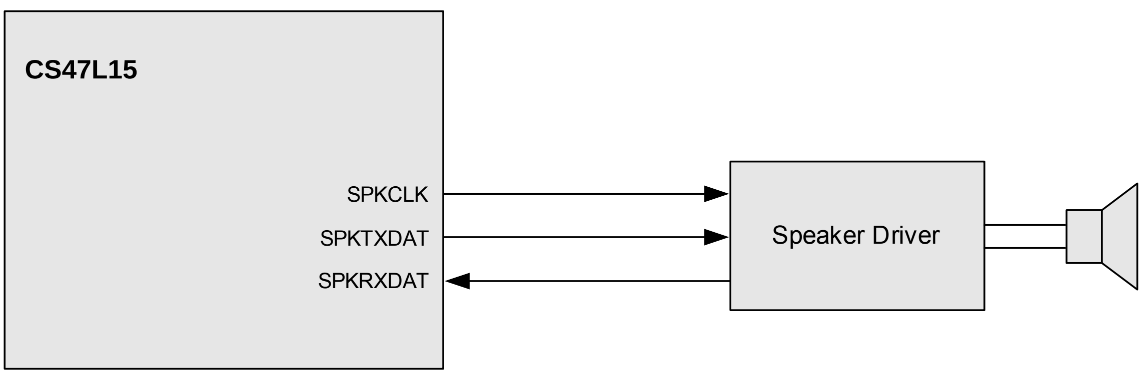


Figure 4-13. Digital Speaker (PDM) Connection with Feedback

The IN2 digital interface timing is similar to the DMIC timing shown in Fig. 4-12, with two audio channels interleaved on SPKRXDAT. See Table 3-14 for a detailed timing specification of the SPKRXDAT digital input.

4.2.5 Input Signal Path Enable

The input signal paths are enabled using the bits described in Table 4-3. The respective bits must be enabled for analog or digital input on the respective input paths.

The input signal paths are muted by default. It is recommended that deselecting the mute should be the final step of the path enable control sequence. Similarly, the mute should be selected as the first step of the path-disable control sequence. The input signal path mute functions are controlled using the bits described in Table 4-6.

The system clock, SYSCLK, must be configured and enabled before any audio path is enabled. See Section 4.13 for details of the system clocks.

The CS47L15 performs automatic checks to confirm that the SYSCLK frequency is high enough to support the input signal paths and associated ADCs. If the frequency is too low, an attempt to enable an input signal path fails. Note that active signal paths are not affected under such circumstances.

The status bits in Register R769 indicate the status of each of the input signal paths. If an underclocked error condition occurs, these bits indicate which input signal paths have been enabled.

Table 4-3. Input Signal Path Enable

Register Address	Bit	Label	Default	Description
R768 (0x0300) Input_Enables	3	IN2L_ENA	0	Input Path 2 (left) enable 0 = Disabled 1 = Enabled
	2	IN2R_ENA	0	Input Path 2 (right) enable 0 = Disabled 1 = Enabled
	1	IN1L_ENA	0	Input Path 1 (left) enable 0 = Disabled 1 = Enabled
	0	IN1R_ENA	0	Input Path 1 (right) enable 0 = Disabled 1 = Enabled
R769 (0x0301) Input_Enables_Status	3	IN2L_ENA_STS	0	Input Path 2 (left) enable status 0 = Disabled 1 = Enabled
	2	IN2R_ENA_STS	0	Input Path 2 (right) enable status 0 = Disabled 1 = Enabled
	1	IN1L_ENA_STS	0	Input Path 1 (left) enable status 0 = Disabled 1 = Enabled
	0	IN1R_ENA_STS	0	Input Path 1 (right) enable status 0 = Disabled 1 = Enabled

4.2.6 Input Signal Path Sample-Rate Control

The input signal paths may be selected as input to the digital mixers or signal-processing functions within the CS47L15 digital core. The sample rate for the input signal paths is configured using IN_RATE; see Table 4-24.

Note that sample-rate conversion is required when routing the input signal paths to any signal chain that is configured for a different sample rate.

4.2.7 Input Signal Path Configuration

The CS47L15 supports up to five analog inputs or up to four digital inputs. Selectable combinations of analog (mic or line) and digital inputs are multiplexed into two stereo input signal paths, as illustrated in Fig. 4-2.

- Input path IN1 can be configured for single-ended, differential, or digital operation. The analog input configuration and pin selection is controlled using the IN1x_SRC bits; digital input mode is selected by setting IN1_MODE.

  If digital input is selected, the IN1_DMICCLK_SRC field must be 00. Under default conditions, this field is locked and cannot be written. To change the value of this field, the user key must be set before writing to IN1_DMICCLK_SRC. It is recommended to clear the user key after writing to IN1_DMICCLK_SRC. See Table 4-105 for details of the user key control register.
- Input path IN2 can be configured for differential or digital operation. The analog mode supports mono, differential connection only; stereo digital input is selected by setting IN2_MODE.

  If analog input is selected (IN2_MODE=0), the IN2L_LP_MODE bit must be set. If digital input is selected (IN2_MODE=1), the IN2L_LP_MODE must be cleared.

A configurable high-pass filter (HPF) is provided on the left and right channels of each input path. The applicable cut-off frequency is selected using IN_HPF_CUT. The filter can be enabled on each path independently using the IN*nx*_HPF bits.

The IN1 analog input paths (single-ended or differential) each incorporate a PGA to provide gain in the range 0 dB to +31 dB in 1-dB steps. Note that these PGAs do not provide pop suppression functions; it is recommended that the gain should not be adjusted while the respective signal path is enabled. The analog input PGA gain is controlled using IN1L_PGA_VOL and IN1R_PGA_VOL.

The IN1 analog input paths can be configured for low-power operation, ideal for always-on applications. If the IN1 signal path is configured for analog input, low-power operation can be selected as described in Section 4.2.7.1.

The IN2 analog input path supports mono input only. The IN2 analog input PGA gain is fixed at 14 dB.

If the IN1 input signal path is configured for digital (DMIC) input, the voltage reference for the DMICDAT/DMICCLK pins is selectable using IN1_DMIC_SUP; the interface is referenced to MICVDD or MICBIAS1. The voltage reference selection controls the digital logic thresholds for the DMICDAT/DMICCLK pins (see Table 3-10)—it should be set equal to the applicable power supply of the respective microphones.

If the IN1 input signal path is configured for digital input, the DMICCLK frequency can be configured using the IN1_OSR field.

If the IN2 input signal path is configured for digital input, the interface clocking frequency is configured using the IN2_OSR field. The IN2_OSR field must select the same frequency as the OUT5_OSR bit (see Table 4-55).

The input signal paths are configured using the fields described in Table 4-4.

Table 4-4. Input Signal Path Configuration

Register Address	Bit	Label	Default	Description
R780 (0x030C) HPF_Control	2:0	IN_HPF_CUT[2:0]	010	Input Path HPF Select. Controls the cut-off frequency of the input path HPF circuits. 000 = 2.5 Hz; 010 = 10 Hz; 100 = 40 Hz 001 = 5 Hz; 011 = 20 Hz; All other codes are reserved
R784 (0x0310) IN1L_Control	15	IN1L_HPF	0	Input Path 1 (Left) HPF Enable 0 = Disabled 1 = Enabled
	12:11	IN1_DMIC_SUP[1:0]	00	Input Path 1 DMIC Reference Select (sets the DMICDAT and DMICCLK logic levels) 00 = MICVDD; All other codes are reserved 01 = MICBIAS1
	10	IN1_MODE	0	Input Path 1 Mode 0 = Analog input 1 = Digital input
	7:1	IN1L_PGA_VOL[6:0]	0x40	Input Path 1 (Left) PGA Volume (applicable to analog inputs only) 0x00 to 0x3F = Reserved; 0x42 = 2 dB; 0x60 to 0x7F = Reserved 0x40 = 0 dB; … (1-dB steps) 0x41 = 1 dB; 0x5F = 31 dB
R785 (0x0311) ADC_Digital_Volume_1L	14:13	IN1L_SRC[1:0]	00	Input Path 1 (Left) Source 00 = Differential (IN1ALP–IN1ALN); 10 = Differential (IN1BP–IN1BN) 01 = Single-ended (IN1ALP); 11 = Single-ended (IN1BP)
R786 (0x0312) DMIC1L_Control	10:8	IN1_OSR[2:0]	101	Input Path 1 Oversample Rate Control If analog input is selected, this field must be set to 101 (default). If digital input is selected, this field controls the DMICCLK frequency. 010 = 384 kHz; 100 = 1.536 MHz; 110 = 6.144 MHz 011 = 768 kHz; 101 = 3.072 MHz; All other codes are reserved
R788 (0x0314) IN1R_Control	15	IN1R_HPF	0	Input Path 1 (Right) HPF Enable 0 = Disabled 1 = Enabled
	12:11	IN1_DMICCLK_SRC[1:0]	01	Input Path 1 DMIC Clock Source 00 = DMICCLK1 All other codes are reserved. If digital input is selected, this field must be 00. Under default conditions, this field is locked and cannot be written. To change the value of this field, the user key must be set before writing to IN1_DMICCLK_SRC.
	7:1	IN1R_PGA_VOL[6:0]	0x40	Input Path 1 (Right) PGA Volume (applicable to analog inputs only) 0x00 to 0x3F = Reserved; 0x42 = 2 dB; 0x60 to 0x7F = Reserved 0x40 = 0 dB; … (1-dB steps) 0x41 = 1 dB; 0x5F = 31 dB
R789 (0x0315) ADC_Digital_Volume_1R	14:13	IN1R_SRC[1:0]	00	Input Path 1 (Right) Source 00 = Differential (IN1ARP–IN1ARN); 10 = Differential (IN1BRP–IN1BRN) 01 = Single-ended (IN1ARP); 11 = Single-ended (IN1BRP)

Table 4-4. Input Signal Path Configuration *(Cont.)*

Register Address	Bit	Label	Default	Description
R792 (0x0318) IN2L_Control	15	IN2L_HPF	0	Input Path 2 (Left) HPF Enable 0 = Disabled 1 = Enabled
	10	IN2_MODE	0	Input Path 2 Mode 0 = Analog input 1 = Digital input
R793 (0x0319) ADC_Digital_ Volume_2L	11	IN2L_LP_ MODE	1	Input Path 2 (Left) control If IN2_MODE = 0 (analog input), the IN2L_LP_MODE bit must be set. If IN2_MODE = 1 (digital input), the IN2L_LP_MODE bit must be cleared.
R794 (0x031A) DMIC2L_Control	10:8	IN2_ OSR[2:0]	101	Input Path 2 Oversample Rate Control If analog input is selected, this field must be set to 101 (default). If digital input is selected, this field must be set to the same frequency as OUT5_OSR. 101 = 3.072 MHz All other codes are reserved 110 = 6.144 MHz
R796 (0x031C) IN2R_Control	15	IN2R_HPF	0	Input Path 2 (Right) HPF Enable 0 = Disabled 1 = Enabled

4.2.7.1 IN1 Low-Power Mode Configuration

The IN1 input path supports low-power operation for analog input configurations. Note that, although the IN1L and IN1R signal paths can be enabled/disabled independently, the selection of Low-Power Mode is common to both channels.

The required register settings for selecting/deselecting Low-Power Mode are described in Table 4-5.

Table 4-5. IN1 Low-Power Mode Control Sequences

IN1 Low-Power Configuration	IN1 Normal (High-Performance) Configuration
• Write 100 to address 0x312, bits [10:8] • Write 001 to address 0x3A8, bits [13:11] • Write 11 to address 0x3C4, bits [1:0]	• Write 101 to address 0x312, bits [10:8] • Write 100 to address 0x3A8, bits [13:11] • Write 00 to address 0x3C4, bits [1:0]

4.2.8 Input Signal Path Digital Volume Control

A digital volume control is provided on each input signal path, providing –64 dB to +31.5 dB gain control in 0.5-dB steps. An independent mute control is also provided for each input signal path.

Whenever the gain or mute setting is changed, the signal path gain is ramped up or down to the new settings at a programmable rate. For increasing gain (or unmute), the rate is controlled by IN_VI_RAMP. For decreasing gain (or mute), the rate is controlled by IN_VD_RAMP.

Note: The IN_VI_RAMP and IN_VD_RAMP fields should not be changed while a volume ramp is in progress.

The IN_VU bits control the loading of the input signal path digital volume and mute controls. When IN_VU is cleared, the digital volume and mute settings are loaded into the respective control register, but do not change the signal path gain. The digital volume and mute settings on all of the input signal paths are updated when a 1 is written to IN_VU. This makes it possible to update the gain of multiple signal paths simultaneously.

Note that, although the digital-volume controls provide 0.5-dB steps, the internal circuits provide signal gain adjustment in 0.125-dB steps. This allows a very high degree of gain control and smooth volume ramping under all operating conditions.

Note: The 0 dBFS level of the IN1/IN2 digital input paths is not equal to the 0 dBFS level of the CS47L15 digital core. The maximum digital input signal level is –6 dBFS (see Table 3-7). Under 0 dB gain conditions, a –6 dBFS input signal corresponds to a 0 dBFS input to the CS47L15 digital core functions.

The digital volume control registers are described in Table 4-6 and Table 4-7.

Table 4-6. Input Signal Path Digital Volume Control

Register Address	Bit	Label	Default	Description
R777 (0x0309) Input_Volume_Ramp	6:4	IN_VD_RAMP[2:0]	010	Input Volume Decreasing Ramp Rate (seconds/6 dB) This field should not be changed while a volume ramp is in progress. 000 = 0 ms 011 = 2 ms 110 = 15 ms 001 = 0.5 ms 100 = 4 ms 111 = 30 ms 010 = 1 ms 101 = 8 ms
	2:0	IN_VI_RAMP[2:0]	010	Input Volume Increasing Ramp Rate (seconds/6 dB) This field should not be changed while a volume ramp is in progress. 000 = 0 ms 011 = 2 ms 110 = 15 ms 001 = 0.5 ms 100 = 4 ms 111 = 30 ms 010 = 1 ms 101 = 8 ms
R785 (0x0311) ADC_Digital_Volume_1L	9	IN_VU	See Footnote 1	Input Signal Paths Volume and Mute Update. Writing 1 to this bit causes the Input Signal Paths Volume and Mute settings to be updated simultaneously
	8	IN1L_MUTE	1	Input Path 1 (Left) Digital Mute 0 = Unmute 1 = Mute
	7:0	IN1L_VOL[7:0]	0x80	Input Path 1 (Left) Digital Volume (see Table 4-7 for volume register definition). –64 dB to +31.5 dB in 0.5-dB steps 0x00 = –64dB 0x80 = 0 dB 0xC0 to 0xFF = Reserved 0x01 = –63.5dB … (0.5-dB steps) … (0.5-dB steps) 0xBF = +31.5 dB
R789 (0x0315) ADC_Digital_Volume_1R	9	IN_VU	See Footnote 1	Input Signal Paths Volume and Mute Update. Writing 1 to this bit causes the Input Signal Paths Volume and Mute settings to be updated simultaneously
	8	IN1R_MUTE	1	Input Path 1 (Right) Digital Mute 0 = Unmute 1 = Mute
	7:0	IN1R_VOL[7:0]	0x80	Input Path 1 (Right) Digital Volume (see Table 4-7 for volume register definition). –64 dB to +31.5 dB in 0.5-dB steps 0x00 = –64dB 0x80 = 0 dB 0xC0 to 0xFF = Reserved 0x01 = –63.5dB … (0.5-dB steps) … (0.5-dB steps) 0xBF = +31.5 dB
R793 (0x0319) ADC_Digital_Volume_2L	9	IN_VU	See Footnote 1	Input Signal Paths Volume and Mute Update. Writing 1 to this bit causes the Input Signal Paths Volume and Mute settings to be updated simultaneously
	8	IN2L_MUTE	1	Input Path 2 (Left) Digital Mute 0 = Unmute 1 = Mute
	7:0	IN2L_VOL[7:0]	0x80	Input Path 2 (Left) Digital Volume (see Table 4-7 for volume register definition). –64 dB to +31.5 dB in 0.5-dB steps 0x00 = –64dB 0x80 = 0 dB 0xC0 to 0xFF = Reserved 0x01 = –63.5dB … (0.5-dB steps) … (0.5-dB steps) 0xBF = +31.5 dB
R797 (0x031D) ADC_Digital_Volume_2R	9	IN_VU	See Footnote 1	Input Signal Paths Volume and Mute Update. Writing 1 to this bit causes the Input Signal Paths Volume and Mute settings to be updated simultaneously
	8	IN2R_MUTE	1	Input Path 2 (Right) Digital Mute 0 = Unmute 1 = Mute
	7:0	IN2R_VOL[7:0]	0x80	Input Path 2 (Right) Digital Volume (see Table 4-7 for volume register definition). –64 dB to +31.5 dB in 0.5-dB steps 0x00 = –64dB 0x80 = 0 dB 0xC0 to 0xFF = Reserved 0x01 = –63.5dB … (0.5-dB steps) … (0.5-dB steps) 0xBF = +31.5 dB

1. Default is not applicable to these write-only bits

Table 4-7 lists the input signal path digital volume settings.

Table 4-7. Input Signal Path Digital Volume Range

Input Volume Register	Volume (dB)	Input Volume Register	Volume (dB)	Input Volume Register	Volume (dB)	Input Volume Register	Volume (dB)
0x00	–64.0	0x31	–39.5	0x62	–15.0	0x93	9.5
0x01	–63.5	0x32	–39.0	0x63	–14.5	0x94	10.0
0x02	–63.0	0x33	–38.5	0x64	–14.0	0x95	10.5
0x03	–62.5	0x34	–38.0	0x65	–13.5	0x96	11.0
0x04	–62.0	0x35	–37.5	0x66	–13.0	0x97	11.5
0x05	–61.5	0x36	–37.0	0x67	–12.5	0x98	12.0
0x06	–61.0	0x37	–36.5	0x68	–12.0	0x99	12.5
0x07	–60.5	0x38	–36.0	0x69	–11.5	0x9A	13.0
0x08	–60.0	0x39	–35.5	0x6A	–11.0	0x9B	13.5
0x09	–59.5	0x3A	–35.0	0x6B	–10.5	0x9C	14.0
0x0A	–59.0	0x3B	–34.5	0x6C	–10.0	0x9D	14.5
0x0B	–58.5	0x3C	–34.0	0x6D	–9.5	0x9E	15.0
0x0C	–58.0	0x3D	–33.5	0x6E	–9.0	0x9F	15.5
0x0D	–57.5	0x3E	–33.0	0x6F	–8.5	0xA0	16.0
0x0E	–57.0	0x3F	–32.5	0x70	–8.0	0xA1	16.5
0x0F	–56.5	0x40	–32.0	0x71	–7.5	0xA2	17.0
0x10	–56.0	0x41	–31.5	0x72	–7.0	0xA3	17.5
0x11	–55.5	0x42	–31.0	0x73	–6.5	0xA4	18.0
0x12	–55.0	0x43	–30.5	0x74	–6.0	0xA5	18.5
0x13	–54.5	0x44	–30.0	0x75	–5.5	0xA6	19.0
0x14	–54.0	0x45	–29.5	0x76	–5.0	0xA7	19.5
0x15	–53.5	0x46	–29.0	0x77	–4.5	0xA8	20.0
0x16	–53.0	0x47	–28.5	0x78	–4.0	0xA9	20.5
0x17	–52.5	0x48	–28.0	0x79	–3.5	0xAA	21.0
0x18	–52.0	0x49	–27.5	0x7A	–3.0	0xAB	21.5
0x19	–51.5	0x4A	–27.0	0x7B	–2.5	0xAC	22.0
0x1A	–51.0	0x4B	–26.5	0x7C	–2.0	0xAD	22.5
0x1B	–50.5	0x4C	–26.0	0x7D	–1.5	0xAE	23.0
0x1C	–50.0	0x4D	–25.5	0x7E	–1.0	0xAF	23.5
0x1D	–49.5	0x4E	–25.0	0x7F	–0.5	0xB0	24.0
0x1E	–49.0	0x4F	–24.5	0x80	0.0	0xB1	24.5
0x1F	–48.5	0x50	–24.0	0x81	0.5	0xB2	25.0
0x20	–48.0	0x51	–23.5	0x82	1.0	0xB3	25.5
0x21	–47.5	0x52	–23.0	0x83	1.5	0xB4	26.0
0x22	–47.0	0x53	–22.5	0x84	2.0	0xB5	26.5
0x23	–46.5	0x54	–22.0	0x85	2.5	0xB6	27.0
0x24	–46.0	0x55	–21.5	0x86	3.0	0xB7	27.5
0x25	–45.5	0x56	–21.0	0x87	3.5	0xB8	28.0
0x26	–45.0	0x57	–20.5	0x88	4.0	0xB9	28.5
0x27	–44.5	0x58	–20.0	0x89	4.5	0xBA	29.0
0x28	–44.0	0x59	–19.5	0x8A	5.0	0xBB	29.5
0x29	–43.5	0x5A	–19.0	0x8B	5.5	0xBC	30.0
0x2A	–43.0	0x5B	–18.5	0x8C	6.0	0xBD	30.5
0x2B	–42.5	0x5C	–18.0	0x8D	6.5	0xBE	31.0
0x2C	–42.0	0x5D	–17.5	0x8E	7.0	0xBF	31.5
0x2D	–41.5	0x5E	–17.0	0x8F	7.5	0xC0–0xFF	Reserved
0x2E	–41.0	0x5F	–16.5	0x90	8.0		
0x2F	–40.5	0x60	–16.0	0x91	8.5		
0x30	–40.0	0x61	–15.5	0x92	9.0		

4.2.9 Input Signal Path Signal-Detect Control

The CS47L15 provides a digital signal-detect function for the input signal path. This enables system actions to be triggered by signal detection and allows the device to remain in a low-power state until a valid audio signal is detected. A mute function is integrated with the signal-detect circuit, ensuring the respective digital audio path remains at zero until the detection threshold level is reached. Signal detection is also indicated via the interrupt controller.

The signal-detect function is supported on input paths IN1 and IN2 in analog and digital configurations (digital input is selected by setting the respective IN*n*_MODE bit). Note that the valid operating conditions for this function vary, depending on the applicable signal-path configuration.

- The signal-detect function is supported on analog input paths for sample rates up to 16 kHz.
- The signal-detect function is supported on digital input paths for sample rates up to 48 kHz.

For each input path, the signal-detect function is enabled by setting the respective IN*nx*_SIG_DET_ENA bit. The detection threshold level is set using IN_SIG_DET_THR—this applies to all input paths.

If the signal-detect function is enabled, the respective input channel is muted if the signal level is below the configured threshold. If the input signal exceeds the threshold level, the respective channel is immediately unmuted.

If the input signal falls below the threshold level, the mute is applied. To prevent erroneous behavior, a time delay is applied before muting the input signal—the channel is only muted if the signal level remains below the threshold level for longer than the hold time. The hold time is set using IN_SIG_DET_HOLD.

Note that the signal-level detection is performed in the digital domain, after the ADC, PGA, digital mute and digital volume controls—the respective input channel must be enabled and unmuted when using the signal-detect function.

The signal-detect function is an input to the interrupt control circuit and can be used to trigger an interrupt event; see Section 4.12. Note that the respective interrupt event represents the logic OR of the signal detection on all input channels and does not provide indication of which input channel caused the interrupt. To avoid multiple interrupts, the signal-detect interrupt can be reasserted only after all input channels have fallen below the trigger threshold level.

The input path signal-detection control registers are described in Table 4-8.

Table 4-8. Input Signal Path Signal-Detect Control

Register Address	Bit	Label	Default	Description
R786 (0x0312) DMIC1L_Control	15	IN1L_SIG_DET_ENA	0	Input Path 1 (Left) Signal-Detect Enable 0 = Disabled 1 = Enabled
R790 (0x0316) DMIC1R_Control	15	IN1R_SIG_DET_ENA	0	Input Path 1 (Right) Signal-Detect Enable 0 = Disabled 1 = Enabled
R794 (0x031A) DMIC2L_Control	15	IN2L_SIG_DET_ENA	0	Input Path 2 (Left) Signal-Detect Enable 0 = Disabled 1 = Enabled
R798 (0x031E) DMIC2R_Control	15	IN2R_SIG_DET_ENA	0	Input Path 2 (Right) Signal-Detect Enable 0 = Disabled 1 = Enabled
R832 (0x0340) Signal_Detect_Globals	8:4	IN_SIG_DET_THR[4:0]	0x00	Input Signal Path Signal-Detect Threshold 0x00 = –30.1 dB; 0x05 = –54.2 dB; 0x0A = –72.2 dB 0x01 = –36.1 dB; 0x06 = –56.7 dB; 0x0B = –74.7 dB 0x02 = –42.1 dB; 0x07 = –60.2 dB; 0x0C = –78.3 dB 0x03 = –48.2 dB; 0x08 = –66.2 dB; 0x0D = –80.8 dB 0x04 = –50.7 dB; 0x09 = –68.7 dB; All other codes are reserved
	3:0	IN_SIG_DET_HOLD[3:0]	0001	Input Signal Path Signal-Detect Hold Time (delay before signal detect indication is deasserted) 0000 = Reserved; ... (4-ms steps); 1100 = 96–100 ms 0001 = 4–8 ms; 1001 = 36–40 ms; 1101 = 192–196 ms 0010 = 8–12 ms; 1010 = 40–44 ms; 1110 = 384–388 ms 0011 = 12–16 ms; 1011 = 48–52 ms; 1111 = 768–772 ms

4.2.10 Digital Input (DMICDAT/SPKRXDAT) Pin Configuration

DMIC operation on the IN1 input path is selected using IN1_MODE, as described in Table 4-4. If DMIC is selected, the DMICCLK and DMICDAT pins are configured as digital output and input, respectively.

The CS47L15 provides an integrated pull-down resistor on the DMICDAT pin; this provides a flexible capability for interfacing with other devices. The DMICDAT pull-down resistor can be configured using the DMICDAT1_PD bit, as described in Table 4-9. Note that, if the IN1 DMIC input path is disabled, the pull-down is disabled on the DMICDAT pin.

Table 4-9. DMIC Interface Pull-Down Control

Register Address	Bit	Label	Default	Description
R840 (0x0348) Dig_Mic_Pad_Ctrl	0	DMICDAT1_PD	0	DMICDAT1 Pull-Down Control 0 = Disabled 1 = Enabled

The SPKRXDAT function is implemented on the SPKRXDAT/GPIO15 pin, which must be configured for digital audio input function when required. See Section 4.11 to configure the pin for SPKRXDAT operation.

Integrated pull-up and pull-down resistors can be enabled on the SPKRXDAT pin. This is provided as part of the GPIO functionality, and provides a flexible capability for interfacing with other devices. The pull-up and pull-down resistors can be configured independently using the fields described in Table 4-72.

If the pull-up and pull-down resistors are both enabled, the CS47L15 provides a bus keeper function on the SPKRXDAT pin. The bus-keeper function holds the logic level unchanged whenever the pin is undriven (e.g., if the signal is tristated).

4.3 Digital Core

The CS47L15 digital core provides extensive mixing and processing capabilities for multiple signal paths. The configuration is highly flexible, and virtually every conceivable input/output connection can be supported between the available processing blocks.

The digital core provides parametric equalization (EQ) functions, DRC, low-/high-pass filters (LHPF), and programmable DSP capability. The DSP can support functions such as wind-noise, side-tone, or other programmable filters, also dynamic range control and compression, or virtual surround sound and other audio enhancements.

The CS47L15 supports multiple signal paths through the digital core. Stereo full-duplex sample-rate conversion is provided to allow digital audio to be routed between input (ADC/DMIC) paths, output (DAC) paths, and digital audio interfaces (AIF1–AIF3) operating at different sample rates.

The DSP functions are highly programmable, using application-specific control sequences. Note that the DSP configuration data is lost whenever the DCVDD power domain is removed; the DSP configuration data must be downloaded to the CS47L15 each time the device is powered up.

The procedure for configuring the CS47L15 DSP functions is tailored to each customer's application; please contact your Cirrus Logic representative for more details.

The digital core incorporates a S/PDIF transmitter that can provide a stereo S/PDIF output on a GPIO pin. Standard sample rates of 32–192 kHz can be supported. The CS47L15 incorporates a tone generator that can be used for beep functions through any of the audio signal paths. A white-noise generator is incorporated, to provide comfort noise in cases where silence (digital mute) is not desirable.

A haptic signal generator is provided, for use with external haptic devices (e.g., mechanical vibration actuators). Two pulse-width modulation (PWM) signal generators are also provided; the PWM waveforms can be modulated by an audio source within the digital core, and can be output on a GPIO pin.

An overview of the digital-core mixing and signal-processing functions is provided in Fig. 4-14.

The control registers associated with the digital-core signal paths are shown in Fig. 4-15 through Fig. 4-29. The full list of digital mixer control registers (R1600–R2936) is provided in Section 6. Generic register field definitions are provided in Table 4-10.

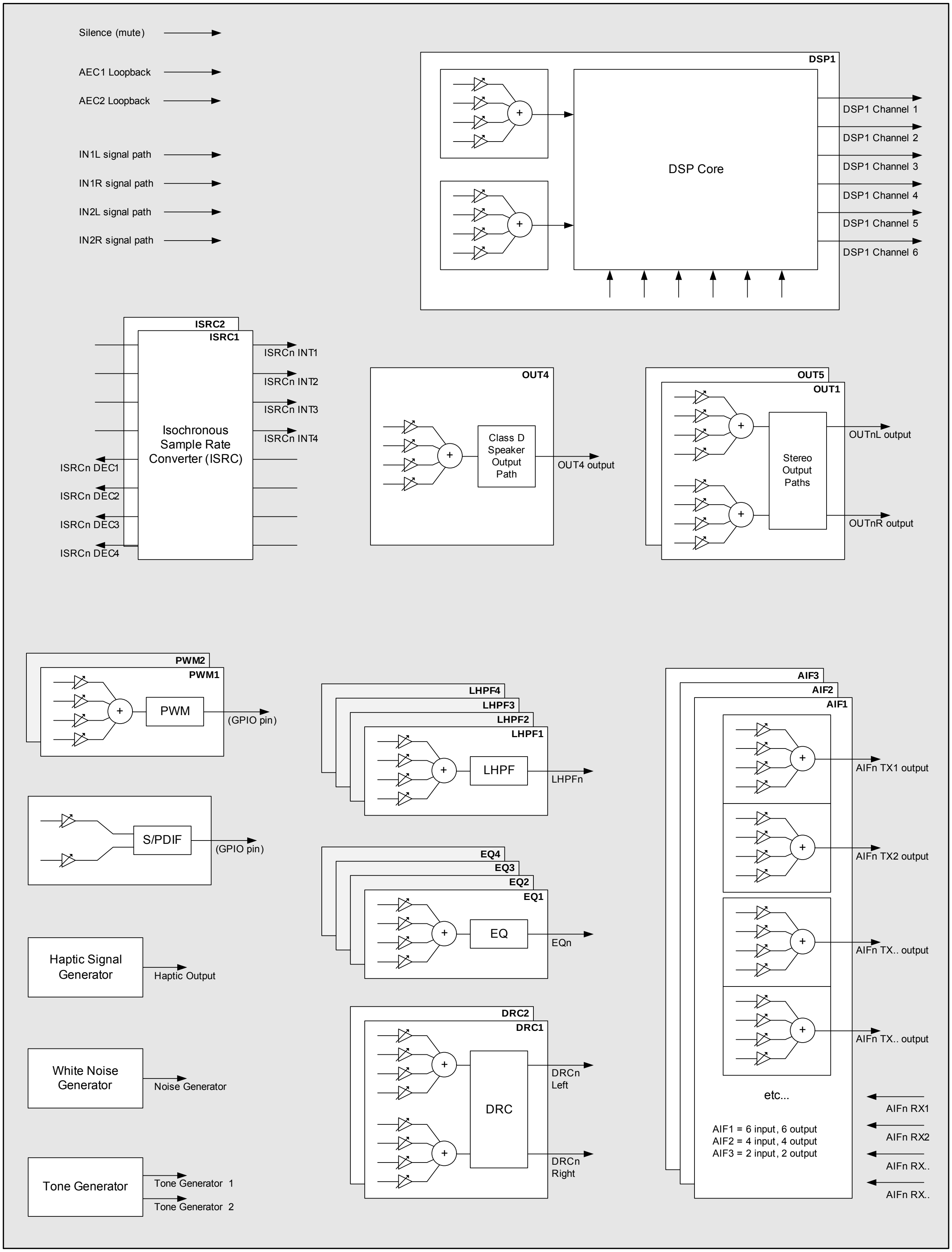


Figure 4-14. Digital Core

4.3.1 Digital-Core Mixers

The CS47L15 provides an extensive digital mixing capability. The digital-core mixing and signal-processing blocks are shown in Fig. 4-14. A four-input digital mixer is associated with many of these functions, as shown. The digital mixer circuit is identical in each instance, providing up to four selectable input sources, with independent volume control on each input.

The control registers associated with the digital-core signal paths are shown in Fig. 4-15–Fig. 4-29. The full list of digital mixer control registers (R1600–R2936) is provided in Section 6.

Further description of the associated control registers is provided throughout Section 4.3. Generic register field definitions are provided in Table 4-10.

The digital mixer input sources are selected using the associated x_SRC*n* fields; the volume control is implemented via the associated x_VOL*n* fields.

The ISRC and DSP auxiliary input functions support selectable input sources, but do not incorporate any digital mixing. The respective input source (x_SRC*n*) fields are identical to those of the digital mixers.

The x_SRC*n* fields select the input sources for the respective mixer or signal-processing block. Note that the selected input sources must be configured for the same sample rate as the blocks to which they are connected. Sample-rate conversion functions are available to support flexible interconnectivity; see Section 4.3.14.

A status bit is associated with each configurable input source. If an underclocked error condition occurs, these bits indicate which signal paths have been enabled.

The generic register field definition for the digital mixers is provided in Table 4-10.

Table 4-10. Digital-Core Mixer Control Registers

Register Address	Bit	Label	Default	Description
R1600 (0x0640) to R2936 (0x0B78)	15	x_STS*n* Valid for every digital core function input (digital mixers, DSP aux inputs, and ISRC inputs).	0	[Digital Core function] input *n* status 0 = Disabled 1 = Enabled
	7:1	x_VOL*n* Valid for every digital mixer input.	0x40	[Digital Core mixer] input *n* volume. (–32 dB to +16 dB in 1-dB steps) 0x00 to 0x20 = –32 dB ... (1-dB steps) 0x50 = +16 dB 0x21 = –31 dB 0x40 = 0 dB 0x51 to 0x7F = +16 dB 0x22 = –30 dB ... (1-dB steps)
	7:0	x_SRC*n* Valid for every digital core function input (digital mixers, DSP aux inputs, and ISRC inputs).	0x00	[Digital Core function] input *n* source select 0x00 = Silence (mute) 0x2A = AIF2 RX3 0x6B = DSP1 Channel 4 0x04 = Tone generator 1 0x2B = AIF2 RX4 0x6C = DSP1 Channel 5 0x05 = Tone generator 2 0x30 = AIF3 RX1 0x6D = DSP1 Channel 6 0x06 = Haptic generator 0x31 = AIF3 RX2 0xA0 = ISRC1 INT1 0x08 = AEC Loop-Back 1 0x50 = EQ1 0xA1 = ISRC1 INT2 0x09 = AEC Loop-Back 2 0x51 = EQ2 0xA2 = ISRC1 INT3 0x0D = Noise generator 0x52 = EQ3 0xA3 = ISRC1 INT4 0x10 = IN1L signal path 0x53 = EQ4 0xA4 = ISRC1 DEC1 0x11 = IN1R signal path 0x58 = DRC1 Left 0xA5 = ISRC1 DEC2 0x12 = IN2L signal path 0x59 = DRC1 Right 0xA6 = ISRC1 DEC3 0x13 = IN2R signal path 0x5A = DRC2 Left 0xA7 = ISRC1 DEC4 0x20 = AIF1 RX1 0x5B = DRC2 Right 0xA8 = ISRC2 INT1 0x21 = AIF1 RX2 0x60 = LHPF1 0xA9 = ISRC2 INT2 0x22 = AIF1 RX3 0x61 = LHPF2 0xAA = ISRC2 INT3 0x23 = AIF1 RX4 0x62 = LHPF3 0xAB = ISRC2 INT4 0x24 = AIF1 RX5 0x63 = LHPF4 0xAC = ISRC2 DEC1 0x25 = AIF1 RX6 0x68 = DSP1 Channel 1 0xAD = ISRC2 DEC2 0x28 = AIF2 RX1 0x69 = DSP1 Channel 2 0xAE = ISRC2 DEC3 0x29 = AIF2 RX2 0x6A = DSP1 Channel 3 0xAF = ISRC2 DEC4

4.3.2 Digital-Core Inputs

The digital core comprises multiple input paths, as shown in Fig. 4-15. Any of these inputs may be selected as a source to the digital mixers or signal-processing functions within the CS47L15 digital core.

Note that the outputs from other blocks within the digital core may also be selected as input to the digital mixers or signal-processing functions within the CS47L15 digital core. Those input sources, which are not shown in Fig. 4-15, are described separately throughout Section 4.3.

The hexadecimal numbers in Fig. 4-15 indicate the corresponding x_SRC*n* setting for selection of that signal as an input to another digital-core function.

The sample rate for the input signal paths is configured by using the applicable IN_RATE or AIF*n*_RATE field; see Table 4-24. Note that sample-rate conversion is required when routing the input signal paths to any signal chain that is configured for a different sample rate.

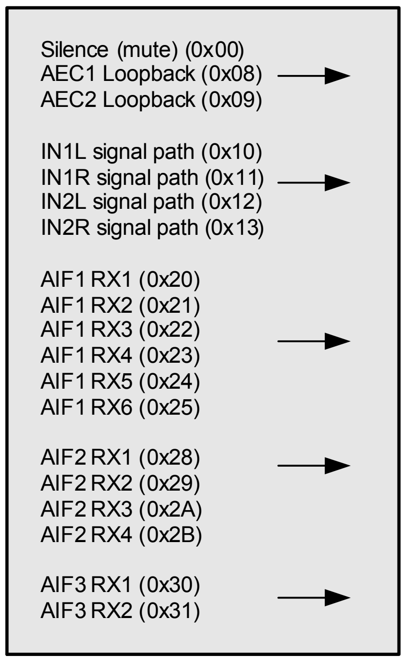


Figure 4-15. Digital-Core Inputs

4.3.3 Digital-Core Output Mixers

The digital core comprises multiple output paths. The output paths associated with AIF1–AIF3 are shown in Fig. 4-16. The output paths associated with OUT1, OUT4, and OUT5 are shown in Fig. 4-17.

A four-input mixer is associated with each output. The four input sources are selectable in each case, and independent volume control is provided for each path.

The AIF1–AIF3 output mixer control fields (see Fig. 4-16) are located at register addresses R1792–R1935 (0x0700–0x078F). The OUT1, OUT4, and OUT5 output mixer control fields (see Fig. 4-17) are located at addresses R1664–R1743 (0x0680–0x06CF).

The full list of digital mixer control registers (R1600–R2936) is provided in Section 6. Generic register field definitions are provided in Table 4-10.

The x_SRC*n* fields select the input sources for the respective mixers. Note that the selected input sources must be configured for the same sample rate as the mixer to which they are connected. Sample-rate conversion functions are available to support flexible interconnectivity; see Section 4.3.14.

The sample rate for the output signal paths is configured using the applicable OUT_RATE or AIF*n*_RATE fields; see Table 4-24. Note that sample-rate conversion is required when routing the output signal paths to any signal chain that is configured for a different sample rate.

The OUT_RATE or AIF*n*_RATE fields must not be changed if any of the respective x_SRC*n* fields is nonzero. The associated x_SRC*n* fields must be cleared before writing new values to OUT_RATE or AIF*n*_RATE. A minimum delay of 125 μs must be allowed between clearing the x_SRC*n* fields and writing to the associated OUT_RATE or AIF*n*_RATE fields. See Table 4-24 for details.

The CS47L15 performs automatic checks to confirm that the SYSCLK frequency is high enough to support the output mixer paths. If the frequency is too low, an attempt to enable an output mixer path fails. Note that active signal paths are not affected under such circumstances.

The status bits in registers R1600–R2936 indicate the status of each of the digital mixers. If an underclocked error condition occurs, these bits indicate which mixers have been enabled.

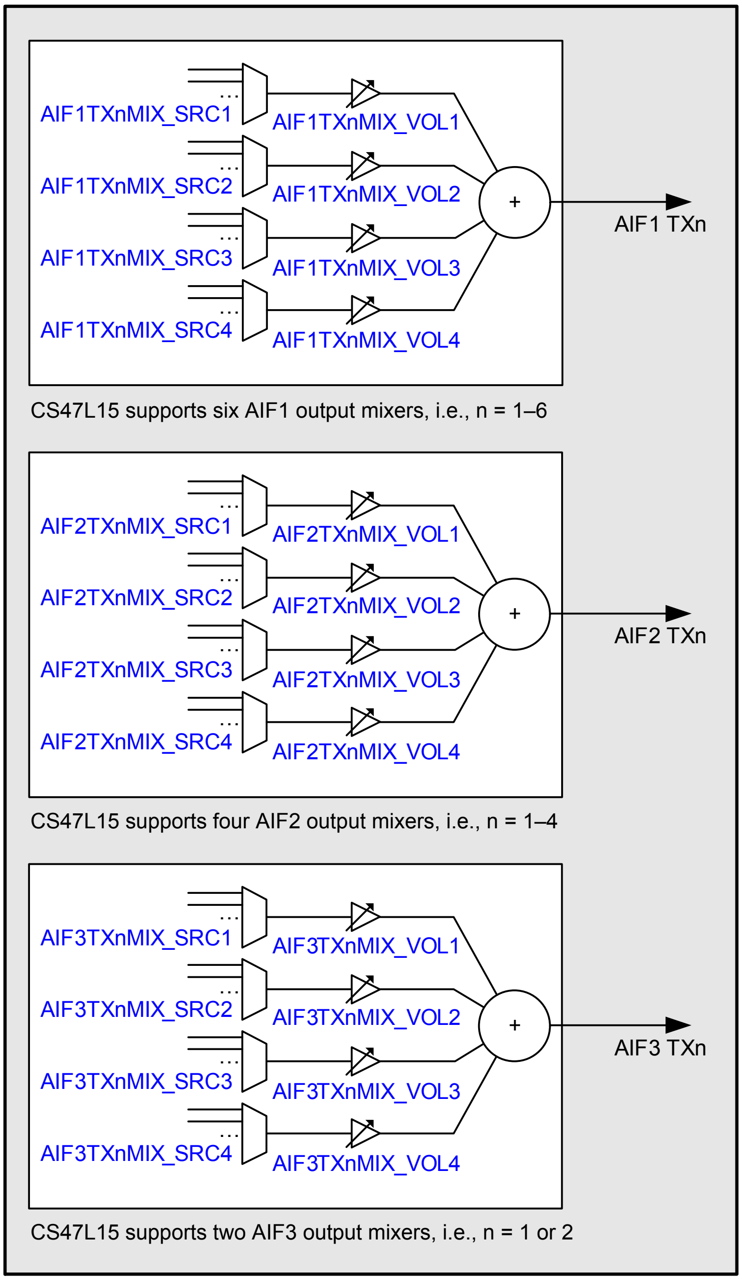


Figure 4-16. Digital-Core AIF Outputs

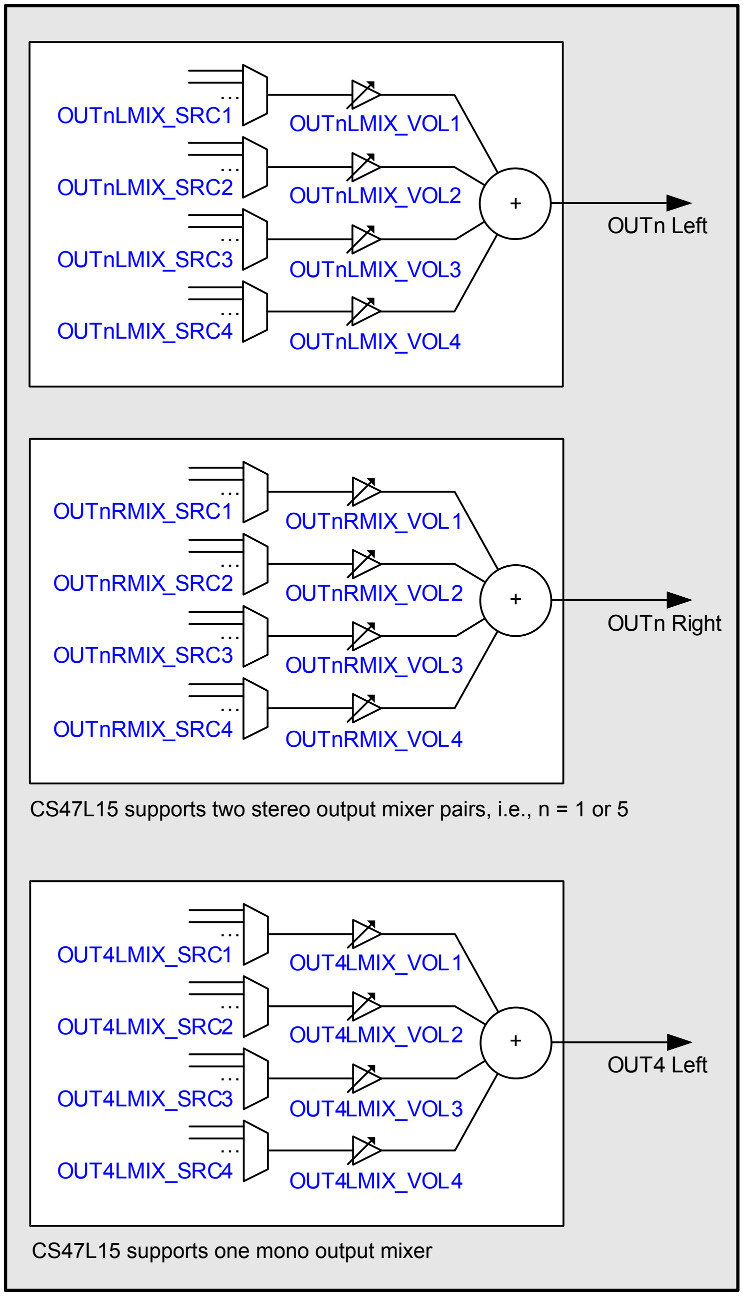


Figure 4-17. Digital-Core OUT*n* Outputs

4.3.4 Five-Band Parametric Equalizer (EQ)

The digital core provides four EQ processing blocks as shown in Fig. 4-18. A four-input mixer is associated with each EQ. The four input sources are selectable in each case, and independent volume control is provided for each path. Each EQ block supports one output.

The EQ provides selective control of five frequency bands as follows:

- The low-frequency band (Band 1) filter can be configured as a peak filter or as a shelving filter. If configured as a shelving filter, it provides adjustable gain below the Band 1 cut-off frequency. As a peak filter, it provides adjustable gain within a defined frequency band that is centered on the Band 1 frequency.
- The midfrequency bands (Band 2–Band 4) filters are peak filters that provide adjustable gain around the respective center frequency.

- The high-frequency band (Band 5) filter is a shelving filter that provides adjustable gain above the Band 5 cut-off frequency.

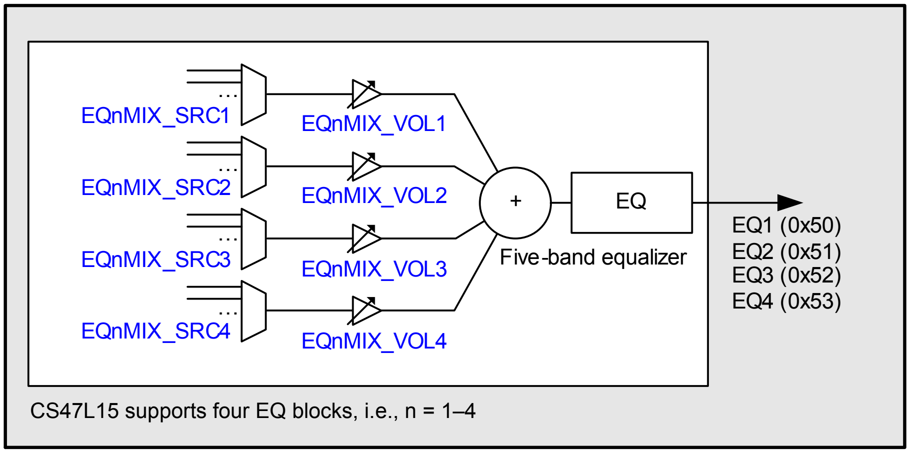


Figure 4-18. Digital-Core EQ Blocks

The EQ1–EQ4 mixer control fields (see Fig. 4-18) are located at register addresses R2176–R2207 (0x0880–0x089F).

The full list of digital-mixer control registers (R1600–R2936) is provided in Section 6. Generic register field definitions are provided in Table 4-10.

The x_SRC*n* fields select the input sources for the respective EQ processing blocks. Note that the selected input sources must be configured for the same sample rate as the EQ to which they are connected. Sample-rate conversion functions are available to support flexible interconnectivity; see Section 4.3.14.

The hexadecimal numbers in Fig. 4-18 indicate the corresponding x_SRC*n* setting for selection of that signal as an input to another digital-core function.

The sample rate for the EQ function is configured using FX_RATE; see Table 4-24. Note that the EQ, DRC, and LHPF functions must be configured for the same sample rate. Sample-rate conversion is required when routing the EQ signal paths to any signal chain that is configured for a different sample rate.

The FX_RATE field must not be changed if any of the associated x_SRC*n* fields is nonzero. The associated x_SRC*n* fields must be cleared before writing a new value to FX_RATE. A minimum delay of 125 µs must be allowed between clearing the x_SRC*n* fields and writing to FX_RATE. See Table 4-24 for details.

The cut-off or center frequencies for the five-band EQ are set by using the coefficients held in the registers identified in Table 4-11. These coefficients are derived using tools provided in Cirrus Logic's WISCE™ evaluation-board control software; please contact your Cirrus Logic representative for details.

Table 4-11. EQ Coefficient Registers

EQ	Register Addresses
EQ1	R3602 (0x0E10) to R3620 (0x0E24)
EQ2	R3624 (0x0E28) to R3642 (0x0E3A)
EQ3	R3646 (0x0E3E) to R3664 (0x0E53)
EQ4	R3668 (0x0E54) to R3686 (0x0E66)

The control registers associated with the EQ functions are described in Table 4-12.

Table 4-12. EQ Enable and Gain Control

Register Address	Bit	Label	Default	Description
R3585 (0x0E01) FX_Ctrl2	15:4	FX_STS[11:0]	0x00	LHPF, DRC, EQ Enable Status. Indicates the status of each of the respective signal-processing functions. Each bit is coded as follows: 0 = Disabled 1 = Enabled [11] = EQ4 [7] = DRC2 (Right) [3] = LHPF4 [10] = EQ3 [6] = DRC2 (Left) [2] = LHPF3 [9] = EQ2 [5] = DRC1 (Right) [1] = LHPF2 [8] = EQ1 [4] = DRC1 (Left) [0] = LHPF1
R3600 (0x0E10) EQ1_1	15:11	EQ1_B1_GAIN[4:0]	0x0C	EQ1 Band 1 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	10:6	EQ1_B2_GAIN[4:0]	0x0C	EQ1 Band 2 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	5:1	EQ1_B3_GAIN[4:0]	0x0C	EQ1 Band 3 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	0	EQ1_ENA	0	EQ1 Enable 0 = Disabled 1 = Enabled
R3601 (0x0E11) EQ1_2	15:11	EQ1_B4_GAIN[4:0]	0x0C	EQ1 Band 4 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	10:6	EQ1_B5_GAIN[4:0]	0x0C	EQ1 Band 5 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	0	EQ1_B1_MODE	0	EQ1 Band 1 Mode 0 = Shelving filter 1 = Peak filter
R3602 (0x0E12) to R3620 (0x0E24)	15:0	EQ1_B1_* EQ1_B2_* EQ1_B3_* EQ1_B4_* EQ1_B5_*	—	EQ1 Frequency Coefficients. Refer to WISCE evaluation board control software for the derivation of these field values.
R3622 (0x0E26) EQ2_1	15:11	EQ2_B1_GAIN[4:0]	0x0C	EQ2 Band 1 Gain [1] –12 dB to +12 dB in 1-dB steps
	10:6	EQ2_B2_GAIN[4:0]	0x0C	EQ2 Band 2 Gain [1] –12 dB to +12 dB in 1-dB steps
	5:1	EQ2_B3_GAIN[4:0]	0x0C	EQ2 Band 3 Gain [1] –12 dB to +12 dB in 1-dB steps
	0	EQ2_ENA	0	EQ2 Enable 0 = Disabled 1 = Enabled
R3623 (0x0E27) EQ2_2	15:11	EQ2_B4_GAIN[4:0]	0x0C	EQ2 Band 4 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	10:6	EQ2_B5_GAIN[4:0]	0x0C	EQ2 Band 5 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	0	EQ2_B1_MODE	0	EQ2 Band 1 Mode 0 = Shelving filter 1 = Peak filter
R3624 (0x0E28) to R3642 (0x0E3A)	15:0	EQ2_B1_* EQ2_B2_* EQ2_B3_* EQ2_B4_* EQ2_B5_*	—	EQ2 Frequency Coefficients. Refer to WISCE evaluation board control software for the derivation of these field values.
R3644 (0x0E3C) EQ3_1	15:11	EQ3_B1_GAIN[4:0]	0x0C	EQ3 Band 1 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	10:6	EQ3_B2_GAIN[4:0]	0x0C	EQ3 Band 2 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	5:1	EQ3_B3_GAIN[4:0]	0x0C	EQ3 Band 3 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	0	EQ3_ENA	0	EQ3 Enable 0 = Disabled 1 = Enabled
R3645 (0x0E3D) EQ3_2	15:11	EQ3_B4_GAIN[4:0]	0x0C	EQ3 Band 4 Gain[1] (–12 dB to +12 dB in 1-dB steps)
	10:6	EQ3_B5_GAIN[4:0]	0x0C	EQ3 Band 5 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	0	EQ3_B1_MODE	0	EQ3 Band 1 Mode 0 = Shelving filter 1 = Peak filter

Table 4-12. EQ Enable and Gain Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R3646 (0x0E3E) to R3664 (0x0E50)	15:0	EQ3_B1_* EQ3_B2_* EQ3_B3_* EQ3_B4_* EQ3_B5_*	—	EQ3 Frequency Coefficients. Refer to WISCE evaluation board control software for the derivation of these field values.
R3666 (0x0E52) EQ4_1	15:11	EQ4_B1_GAIN[4:0]	0x0C	EQ4 Band 1 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	10:6	EQ4_B2_GAIN[4:0]	0x0C	EQ4 Band 2 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	5:1	EQ4_B3_GAIN[4:0]	0x0C	EQ4 Band 3 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	0	EQ4_ENA	0	EQ4 Enable 0 = Disabled 1 = Enabled
R3667 (0x0E53) EQ4_2	15:11	EQ4_B4_GAIN[4:0]	0x0C	EQ4 Band 4 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	10:6	EQ4_B5_GAIN[4:0]	0x0C	EQ4 Band 5 Gain [1] (–12 dB to +12 dB in 1-dB steps
	0	EQ4_B1_MODE	0	EQ4 Band 1 Mode 0 = Shelving filter 1 = Peak filter
R3668 (0x0E54) to R3686 (0x0E66)	15:0	EQ4_B1_* EQ4_B2_* EQ4_B3_* EQ4_B4_* EQ4_B5_*	—	EQ4 Frequency Coefficients Refer to WISCE evaluation board control software for the derivation of these field values.

1.See Table 4-13 for gain range.

Table 4-13 lists the EQ gain control settings.

Table 4-13. EQ Gain-Control Range

EQ Gain Setting	Gain (dB)	EQ Gain Setting	Gain (dB)
00000	–12	01101	+1
00001	–11	01110	+2
00010	–10	01111	+3
00011	–9	10000	+4
00100	–8	10001	+5
00101	–7	10010	+6
00110	–6	10011	+7
00111	–5	10100	+8
01000	–4	10101	+9
01001	–3	10110	+10
01010	–2	10111	+11
01011	–1	11000	+12
01100	0	11001–11111	Reserved

The CS47L15 automatically checks to confirm whether the SYSCLK frequency is high enough to support the commanded EQ and digital mixing functions. If an attempt is made to enable an EQ signal path, and there are insufficient SYSCLK cycles to support it, the attempt does not succeed. Note that any signal paths that are already active are not affected under such circumstances.

The FX_STS field in register R3585 indicates the status of each of the EQ, DRC, and LHPF signal paths. If an underclocked error condition occurs, this field indicates which EQ, DRC, or LHPF signal paths have been enabled.

The status bits in registers R1600–R2936 indicate the status of each of the digital mixers. If an underclocked error condition occurs, these bits indicate which mixers have been enabled.

4.3.5 Dynamic Range Control (DRC)

The digital core provides two stereo DRC processing blocks, as shown in Fig. 4-19. A four-input mixer is associated with each DRC input channel. The input sources are selectable in each case, and independent volume control is provided for each path. The stereo DRC blocks support two outputs each.

The function of the DRC is to adjust the signal gain in conditions where the input amplitude is unknown or varies over a wide range, for example, when recording from microphones built into a handheld system or to restrict the dynamic range of an output signal path.

To improve intelligibility in the presence of loud impulsive noises, the DRC can apply compression and automatic level control to the signal path. It incorporates anticlip and quick-release features for handling transients.

The DRC also incorporates a noise-gate function that provides additional attenuation of very low-level input signals. This means that the signal path is quiet when no signal is present, giving an improvement in background noise level under these conditions.

A signal-detect function is provided within the DRC; this can be used to detect the presence of an audio signal and to trigger other events. It can also be used as an interrupt event or to trigger the control-write sequencer. Note that DRC triggering of the control-write sequencer is supported for DRC1 only.

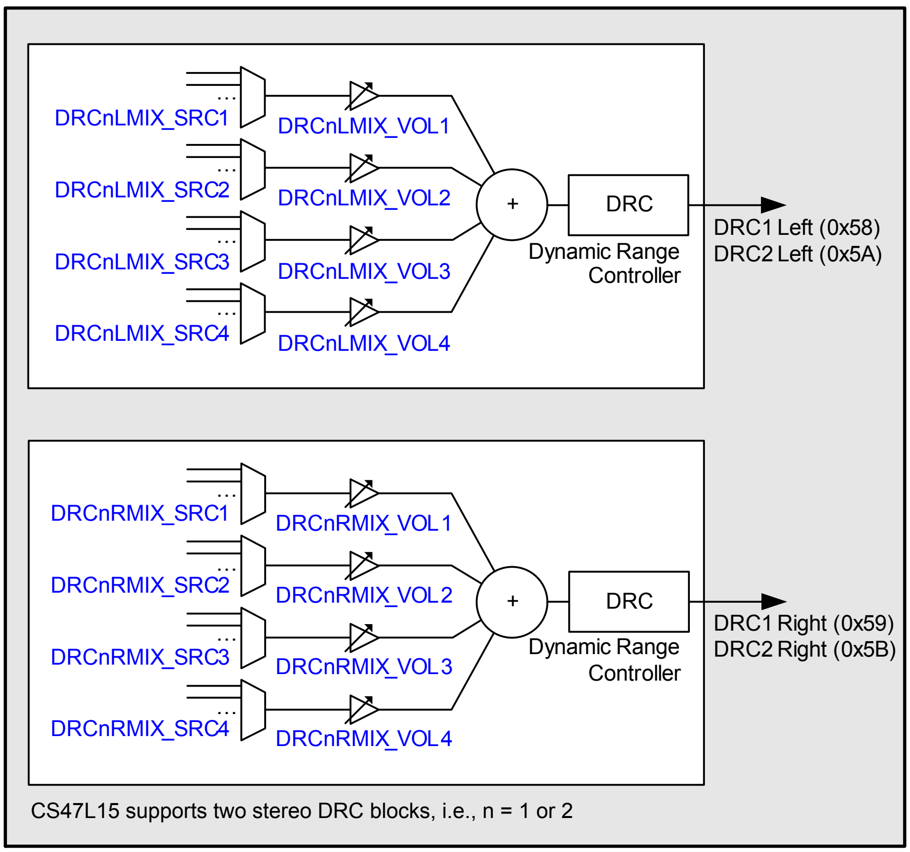


Figure 4-19. Dynamic Range Control (DRC) Block

The DRC1 and DRC2 mixer control fields (see Fig. 4-19) are located at register addresses R2240–R2271 (0x08C0–0x08DF).

The full list of digital mixer control registers (R1600–R2936) is provided in Section 6. Generic register field definitions are provided in Table 4-10.

The x_SRC*n* fields select the input sources for the respective DRC processing blocks. Note that the selected input sources must be configured for the same sample rate as the DRC to which they are connected. Sample-rate conversion functions are available to support flexible interconnectivity; see Section 4.3.14.

The hexadecimal numbers in Fig. 4-19 indicate the corresponding x_SRC*n* setting for selection of that signal as an input to another digital-core function.

The sample rate for the DRC function is configured using FX_RATE; see Table 4-24. Note that the EQ, DRC, and LHPF functions must all be configured for the same sample rate. Sample-rate conversion is required when routing the DRC signal paths to any signal chain that is configured for a different sample rate.

The FX_RATE field must not be changed if any of the associated x_SRC*n* fields is nonzero. The associated x_SRC*n* fields must be cleared before writing a new value to FX_RATE. A minimum delay of 125 µs must be allowed between clearing the x_SRC*n* fields and writing to FX_RATE. See Table 4-24 for details.

The DRC functions are enabled using the control registers described in Table 4-14.

Table 4-14. DRC Enable

Register Address	Bit	Label	Default	Description
R3712 (0x0E80) DRC1_ctrl1	1	DRC1L_ENA	0	DRC1 (left) enable 0 = Disabled 1 = Enabled
	0	DRC1R_ENA	0	DRC1 (right) enable 0 = Disabled 1 = Enabled
R3720 (0x0E88) DRC2_ctrl1	1	DRC2L_ENA	0	DRC2 (left) enable 0 = Disabled 1 = Enabled
	0	DRC2R_ENA	0	DRC2 (right) enable 0 = Disabled 1 = Enabled

The following description of the DRC is applicable to each of the DRCs. The associated control fields are described in Table 4-16 and Table 4-17 for DRC1 and DRC2 respectively.

4.3.5.1 DRC Compression, Expansion, and Limiting

The DRC supports two different compression regions, separated by a knee at a specific input amplitude. In the region above the knee, the compression slope DRC*n*_HI_COMP applies; in the region below the knee, the compression slope DRC*n*_LO_COMP applies. Note that *n* identifies the applicable DRC 1 or 2.

The DRC also supports a noise-gate region, where low-level input signals are heavily attenuated. This function can be enabled or disabled according to the application requirements. The DRC response in this region is defined by the expansion slope DRC*n*_NG_EXP.

For additional attenuation of signals in the noise-gate region, an additional knee can be defined (shown as Knee 2 in Fig. 4-20). When this knee is enabled, this introduces an infinitely steep drop-off in the DRC response pattern between the DRC*n*_LO_COMP and DRC*n*_NG_EXP regions.

The overall DRC compression characteristic in steady state (i.e., where the input amplitude is near constant) is shown in Fig. 4-20.

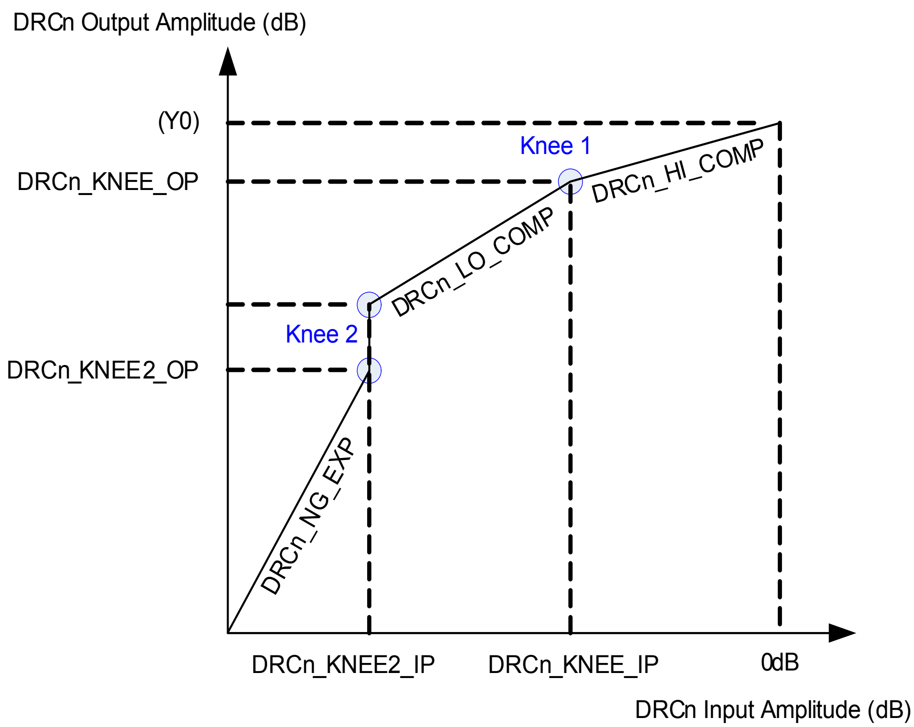


Figure 4-20. DRC Response Characteristic

The slope of the DRC response is determined by DRC*n*_HI_COMP and DRC*n*_LO_COMP. A slope of 1 indicates constant gain in this region. A slope less than 1 represents compression (i.e., a change in input amplitude produces only a smaller change in output amplitude). A slope of 0 indicates that the target output amplitude is the same across a range of input amplitudes; this is infinite compression.

When the noise gate is enabled, the DRC response in this region is determined by DRC*n*_NG_EXP. A slope of 1 indicates constant gain in this region. A slope greater than 1 represents expansion (i.e., a change in input amplitude produces a larger change in output amplitude).

When the DRC*n*_KNEE2_OP knee is enabled (Knee 2 in Fig. 4-20), this introduces the vertical line in the response pattern shown, resulting in infinitely steep attenuation at this point in the response.

The DRC parameters are listed in Table 4-15.

Table 4-15. DRC Response Parameters

Parameters	Parameter	Description
1	DRC*n*_KNEE_IP	Input level at Knee 1 (dB)
2	DRC*n*_KNEE_OP	Output level at Knee 2 (dB)
3	DRC*n*_HI_COMP	Compression ratio above Knee 1
4	DRC*n*_LO_COMP	Compression ratio below Knee 1
5	DRC*n*_KNEE2_IP	Input level at Knee 2 (dB)
6	DRC*n*_NG_EXP	Expansion ratio below Knee 2
7	DRC*n*_KNEE2_OP	Output level at Knee 2 (dB)

The noise gate is enabled by setting DRC*n*_NG_ENA. When the noise gate is not enabled, Parameters 5–7 (see Table 4-15) are ignored, and the DRC*n*_LO_COMP slope applies to all input signal levels below Knee 1.

The DRC*n*_KNEE2_OP knee is enabled by setting DRC*n*_KNEE2_OP_ENA. If this bit is not set, Parameter 7 is ignored and the Knee 2 position always coincides with the low end of the DRC*n*_LO_COMP region.

The Knee 1 point in Fig. 4-20 is determined by DRC*n*_KNEE_IP and DRC*n*_KNEE_OP.

Parameter Y0, the output level for a 0 dB input, is not specified directly but can be calculated from the other parameters using Eq. 4-1.

$$Y0 = DRCn\_KNEE\_OP - (DRCn\_KNEE\_IP \times DRCn\_HI\_COMP)$$

Equation 4-1. DRC Compression Calculation

4.3.5.2 Gain Limits

The minimum and maximum gain applied by the DRC is set by DRC*n*_MINGAIN, DRC*n*_MAXGAIN, and DRC*n*_NG_MINGAIN. These limits can be used to alter the DRC response from that shown in Fig. 4-20. If the range between maximum and minimum gain is reduced, the extent of the dynamic range control is reduced.

The minimum gain in the compression regions of the DRC response is set by DRC*n*_MINGAIN. The minimum gain in the noise-gate region is set by DRC*n*_NG_MINGAIN. The minimum gain limit prevents excessive attenuation of the signal path.

The maximum gain limit set by DRC*n*_MAXGAIN prevents quiet signals (or silence) from being excessively amplified.

4.3.5.3 Dynamic Characteristics

The dynamic behavior determines how quickly the DRC responds to changing signal levels. Note that the DRC responds to the average (RMS) signal amplitude over a period of time.

The DRC*n*_ATK determines how quickly the DRC gain decreases when the signal amplitude is high. The DRC*n*_DCY determines how quickly the DRC gain increases when the signal amplitude is low.

These fields are described in Table 4-16 and Table 4-17. The register defaults are suitable for general-purpose microphone use.

4.3.5.4 Anticlip Control

The DRC includes an anticlip feature to avoid signal clipping when the input amplitude rises very quickly. This feature uses a feed-forward technique for early detection of a rising signal level. Signal clipping is avoided by dynamically increasing the gain attack rate when required. The anticlip feature is enabled using the DRC*n*_ANTICLIP bit.

Note that the feed-forward processing increases the latency in the input signal path.

Note that the anticlip feature operates entirely in the digital domain. It cannot be used to prevent signal clipping in the analog domain nor in the source signal. Analog clipping can only be prevented by reducing the analog signal gain or by adjusting the source signal.

4.3.5.5 Quick Release Control

The DRC includes a quick-release feature to handle short transient peaks that are not related to the intended source signal. For example, in handheld microphone recording, transient signal peaks sometimes occur due to user handling, key presses or accidental tapping against the microphone. The quick-release feature ensures that these transients do not cause the intended signal to be masked by the longer time constant of DRC*n*_DCY.

The quick-release feature is enabled by setting the DRC*n*_QR bit. When this bit is enabled, the DRC measures the crest factor (peak to RMS ratio) of the input signal. A high crest factor is indicative of a transient peak that may not be related to the intended source signal. If the crest factor exceeds the level set by DRC*n*_QR_THR, the normal decay rate (DRC*n*_DCY) is ignored and a faster decay rate (DRC*n*_QR_DCY) is used instead.

4.3.5.6 Signal Activity Detect

The DRC incorporates a configurable signal-detect function, allowing the signal level at the DRC input to be monitored and to be used to trigger other events. This can be used to detect the presence of a microphone signal on an ADC or DMIC channel, or can be used to detect an audio signal received over the digital audio interface.

The DRC signal-detect function is enabled by setting DRC*n*_SIG_DET. Note that the respective DRC*n* must also be enabled. The detection threshold is either a peak level (crest factor) or an RMS level, depending on DRC*n*_SIG_DET_MODE. When peak level is selected, the threshold is determined by DRC*n*_SIG_DET_PK, which defines the applicable crest factor (peak-to-RMS ratio) threshold. If RMS level is selected, the threshold is set using DRC*n*_SIG_DET_RMS.

The DRC signal-detect function is an input to the interrupt control circuit and can be used to trigger an interrupt event; see Section 4.12.

The control-write sequencer can be triggered by the DRC1 signal-detect function. This is enabled by setting DRC1_WSEQ_SIG_DET_ENA. See Section 4.15.

Note that signal detection is supported on DRC1 and DRC2, but the triggering of the control-write sequencer is available on DRC1 only.

4.3.5.7 DRC Register Controls

The DRC1 control registers are described in Table 4-16.

Table 4-16. DRC1 Control Registers

Register Address	Bit	Label	Default	Description
R3585 (0x0E01) FX_Ctrl2	15:4	FX_STS[11:0]	0x00	LHPF, DRC, EQ enable status. Indicates the status of each of the respective signal-processing functions. Each bit is coded as follows: 0 = Disabled 1 = Enabled [11] = EQ4 [7] = DRC2 (Right) [3] = LHPF4 [10] = EQ3 [6] = DRC2 (Left) [2] = LHPF3 [9] = EQ2 [5] = DRC1 (Right) [1] = LHPF2 [8] = EQ1 [4] = DRC1 (Left) [0] = LHPF1
R3712 (0x0E80) DRC1_ctrl1	15:11	DRC1_SIG_DET_RMS[4:0]	0x00	DRC1 Signal-Detect RMS Threshold. RMS signal level for signal-detect to be indicated when DRC1_SIG_DET_MODE = 1. 0x00 = –30 dB …. (1.5-dB steps) 0x1F = –76.5 dB 0x01 = –31.5 dB 0x1E = –75 dB
	10:9	DRC1_SIG_DET_PK[1:0]	00	DRC1 Signal-Detect Peak Threshold. This is the Peak/RMS ratio, or Crest Factor, level for signal-detect to be indicated when DRC1_SIG_DET_MODE = 0. 00 = 12 dB 10 = 24 dB 01 = 18 dB 11 = 30 dB
	8	DRC1_NG_ENA	0	DRC1 Noise-Gate Enable 0 = Disabled 1 = Enabled
	7	DRC1_SIG_DET_MODE	0	DRC1 Signal-Detect Mode 0 = Peak threshold mode 1 = RMS threshold mode
	6	DRC1_SIG_DET	0	DRC1 Signal-Detect Enable 0 = Disabled 1 = Enabled
	5	DRC1_KNEE2_OP_ENA	0	DRC1 KNEE2_OP Enable 0 = Disabled 1 = Enabled
	4	DRC1_QR	1	DRC1 Quick-release Enable 0 = Disabled 1 = Enabled
	3	DRC1_ANTICLIP	1	DRC1 Anticlip Enable 0 = Disabled 1 = Enabled
	2	DRC1_WSEQ_SIG_DET_ENA	0	DRC1 Signal-Detect Write Sequencer Select 0 = Disabled 1 = Enabled

Table 4-16. DRC1 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R3713 (0x0E81) DRC1_ctrl2	12:9	DRC1_ATK[3:0]	0100	DRC1 Gain attack rate (seconds/6 dB) 0000 = Reserved 0101 = 2.9 ms 1010 = 92.8 ms 0001 = 181 µs 0110 = 5.8 ms 1011 = 185.6 ms 0010 = 363 µs 0111 = 11.6 ms 1100 to 1111 = Reserved 0011 = 726 µs 1000 = 23.2 ms 0100 = 1.45 ms 1001 = 46.4 ms
	8:5	DRC1_DCY[3:0]	1001	DRC1 Gain decay rate (seconds/6 dB) 0000 = 1.45 ms 0101 = 46.5 ms 1010 = 1.49 s 0001 = 2.9 ms 0110 = 93 ms 1011 = 2.97 s 0010 = 5.8 ms 0111 = 186 ms 1100 to 1111 = Reserved 0011 = 11.6 ms 1000 = 372 ms 0100 = 23.25 ms 1001 = 743 ms
	4:2	DRC1_ MINGAIN[2:0]	100	DRC1 Minimum gain to attenuate audio signals 000 = 0 dB 011 = –24 dB 11X = Reserved 001 = –12 dB 100 = –36 dB 010 = –18 dB 101 = Reserved
	1:0	DRC1_ MAXGAIN[1:0]	11	DRC1 Maximum gain to boost audio signals (dB) 00 = 12 dB 10 = 24 dB 01 = 18 dB 11 = 36 dB
R3714 (0x0E82) DRC1_ctrl3	15:12	DRC1_NG_ MINGAIN[3:0]	0000	DRC1 Minimum gain to attenuate audio signals when the Noise Gate is active. 0000 = –36 dB 0101 = –6 dB 1010 = 24 dB 0001 = –30 dB 0110 = 0 dB 1011 = 30 dB 0010 = –24 dB 0111 = 6 dB 1100 = 36 dB 0011 = –18 dB 1000 = 12 dB 1101 to 1111 = Reserved 0100 = –12 dB 1001 = 18 dB
	11:10	DRC1_NG_ EXP[1:0]	00	DRC1 Noise-Gate slope 00 = 1 (no expansion) 10 = 4 01 = 2 11 = 8
	9:8	DRC1_QR_ THR[1:0]	00	DRC1 Quick-release threshold (crest factor in dB) 00 = 12 dB 10 = 24 dB 01 = 18 dB 11 = 30 dB
	7:6	DRC1_QR_ DCY[1:0]	00	DRC1 Quick-release decay rate (seconds/6 dB) 00 = 0.725 ms 10 = 5.8 ms 01 = 1.45 ms 11 = Reserved
	5:3	DRC1_HI_ COMP[2:0]	011	DRC1 Compressor slope (upper region) 000 = 1 (no compression) 011 = 1/8 110 = Reserved 001 = 1/2 100 = 1/16 111 = Reserved 010 = 1/4 101 = 0
	2:0	DRC1_LO_ COMP[2:0]	000	DRC1 Compressor slope (lower region) 000 = 1 (no compression) 011 = 1/8 11X = Reserved 001 = 1/2 100 = 0 010 = 1/4 101 = Reserved
R3715 (0x0E83) DRC1_ctrl4	10:5	DRC1_KNEE_ IP[5:0]	0x00	DRC1 Input signal level at the compressor knee. 0x00 = 0 dB 0x02 = –1.5 dB 0x3C = –45 dB 0x01 = –0.75 dB … (–0.75-dB steps) 0x3D–-0x3F = Reserved
	4:0	DRC1_KNEE_ OP[4:0]	0x00	DRC1 Output signal at the compressor knee. 0x00 = 0 dB 0x02 = –1.5 dB 0x1E = –22.5 dB 0x01 = –0.75 dB … (–0.75 dB steps) 0x1F = Reserved
R3716 (0x0E84) DRC1_ctrl5	9:5	DRC1_KNEE2_ IP[4:0]	0x00	DRC1 Input signal level at the noise-gate threshold Knee 2. 0x00 = –36 dB 0x02 = –39 dB 0x1E = –81 dB 0x01 = –37.5 dB … (-1.5-dB steps) 0x1F = –82.5 dB Applicable if DRC1_NG_ENA = 1.
	4:0	DRC1_KNEE2_ OP[4:0]	0x00	DRC1 Output signal at the noise-gate threshold Knee 2. 0x00 = –30 dB 0x02 = –33 dB 0x1E = –75 dB 0x01 = –31.5 dB … (–1.5dB steps) 0x1F = –76.5 dB Applicable only if DRC1_KNEE2_OP_ENA = 1.

The DRC2 control registers are described in Table 4-17.

Table 4-17. DRC2 Control Registers

Register Address	Bit	Label	Default	Description
R3585 (0x0E01) FX_Ctrl2	15:4	FX_STS[11:0]	0x00	LHPF, DRC, EQ Enable Status. Indicates the status of each of the respective signal-processing functions. Each bit is coded as follows: 0 = Disabled 1 = Enabled [11] = EQ4 [7] = DRC2 (Right) [3] = LHPF4 [10] = EQ3 [6] = DRC2 (Left) [2] = LHPF3 [9] = EQ2 [5] = DRC1 (Right) [1] = LHPF2 [8] = EQ1 [4] = DRC1 (Left) [0] = LHPF1
R3720 (0x0E88) DRC2_ctrl1	15:11	DRC2_SIG_DET_RMS[4:0]	0x00	DRC2 Signal-Detect RMS Threshold. This is the RMS signal level for signal-detect to be indicated when DRC2_SIG_DET_MODE = 1. 0x00 = –30 dB …. (1.5-dB steps) 0x1F = –76.5 dB 0x01 = –31.5 dB 0x1E = –75 dB
	10:9	DRC2_SIG_DET_PK[1:0]	00	DRC2 Signal-Detect Peak Threshold. Peak/RMS ratio, or Crest Factor, level for signal-detect to be indicated when DRC2_SIG_DET_MODE = 0. 00 = 12 dB 10 = 24 dB 01 = 18 dB 11 = 30 dB
	8	DRC2_NG_ENA	0	DRC2 Noise-Gate Enable 0 = Disabled 1 = Enabled
	7	DRC2_SIG_DET_MODE	0	DRC2 Signal-Detect Mode 0 = Peak threshold mode 1 = RMS threshold mode
	6	DRC2_SIG_DET	0	DRC2 Signal-Detect Enable 0 = Disabled 1 = Enabled
	5	DRC2_KNEE2_OP_ENA	0	DRC2 KNEE2_OP Enable 0 = Disabled 1 = Enabled
	4	DRC2_QR	1	DRC2 Quick-release Enable 0 = Disabled 1 = Enabled
	3	DRC2_ANTICLIP	1	DRC2 Anticlip Enable 0 = Disabled 1 = Enabled
R3721 (0x0E89) DRC2_ctrl2	12:9	DRC2_ATK[3:0]	0100	DRC2 Gain attack rate (seconds/6 dB) 0000 = Reserved 0101 = 2.9 ms 1010 = 92.8 ms 0001 = 181 µs 0110 = 5.8 ms 1011 = 185.6 ms 0010 = 363 µs 0111 = 11.6 ms 1100 to 1111 = Reserved 0011 = 726 µs 1000 = 23.2 ms 0100 = 1.45 ms 1001 = 46.4 ms
	8:5	DRC2_DCY[3:0]	1001	DRC2 Gain decay rate (seconds/6 dB) 0000 = 1.45 ms 0101 = 46.5 ms 1010 = 1.49 s 0001 = 2.9 ms 0110 = 93 ms 1011 = 2.97 s 0010 = 5.8 ms 0111 = 186 ms 1100 to 1111 = Reserved 0011 = 11.6 ms 1000 = 372 ms 0100 = 23.25 ms 1001 = 743 ms
	4:2	DRC2_MINGAIN[2:0]	100	DRC2 Minimum gain to attenuate audio signals 000 = 0 dB 011 = –24 dB 11X = Reserved 001 = –12 dB (default) 100 = –36 dB 010 = –18 dB 101 = Reserved
	1:0	DRC2_MAXGAIN[1:0]	11	DRC2 Maximum gain to boost audio signals (dB) 00 = 12 dB 10 = 24 dB 01 = 18 dB 11 = 36 dB

Table 4-17. DRC2 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R3722 (0x0E8A) DRC2_ctrl3	15:12	DRC2_NG_ MINGAIN[3:0]	0000	DRC2 Minimum gain to attenuate audio signals when the Noise Gate is active. 0000 = –36 dB 0101 = –6 dB 1010 = 24 dB 0001 = –30 dB 0110 = 0 dB 1011 = 30 dB 0010 = –24 dB 0111 = 6 dB 1100 = 36 dB 0011 = –18 dB 1000 = 12 dB 1101 to 1111 = Reserved 0100 = –12 dB 1001 = 18 dB
	11:10	DRC2_NG_ EXP[1:0]	00	DRC2 Noise-Gate slope 00 = 1 (no expansion) 01 = 2 10 = 4 11 = 8
	9:8	DRC2_QR_ THR[1:0]	00	DRC2 Quick-release threshold (crest factor in dB) 00 = 12 dB 01 = 18 dB 10 = 24 dB 11 = 30 dB
	7:6	DRC2_QR_ DCY[1:0]	00	DRC2 Quick-release decay rate (seconds/6 dB) 00 = 0.725 ms 01 = 1.45 ms 10 = 5.8 ms 11 = Reserved
	5:3	DRC2_HI_ COMP[2:0]	011	DRC2 Compressor slope (upper region) 000 = 1 (no compression) 011 = 1/8 110–111 = Reserved 001 = 1/2 100 = 1/16 010 = 1/4 101 = 0
	2:0	DRC2_LO_ COMP[2:0]	000	DRC2 Compressor slope (lower region) 000 = 1 (no compression) 010 = 1/4 100 = 0 001 = 1/2 011 = 1/8 101–11X = Reserved
R3723 (0x0E8B) DRC2_ctrl4	10:5	DRC2_KNEE_ IP[5:0]	0x00	DRC2 Input signal level at the compressor knee. 0x00 = 0 dB 0x02 = –1.5 dB 0x3C = –45 dB 0x01 = –0.75 dB … (–0.75-dB steps) 0x3D–-0x3F = Reserved
	4:0	DRC2_KNEE_ OP[4:0]	0x00	DRC2 Output signal at the compressor knee. 0x00 = 0 dB 0x02 = –1.5 dB 0x1E = –22.5 dB 0x01 = –0.75 dB … (–0.75 dB steps) 0x1F = Reserved
R3724 (0x0E8C) DRC2_ctrl5	9:5	DRC2_ KNEE2_IP[4:0]	0x00	DRC2 Input signal level at the noise-gate threshold Knee 2. 0x00 = –36 dB 0x02 = –39 dB 0x1E = –81 dB 0x01 = –37.5 dB … (-1.5-dB steps) 0x1F = –82.5 dB Applicable only if DRC2_NG_ENA = 1.
	4:0	DRC2_ KNEE2_ OP[4:0]	0x00	DRC2 Output signal at the noise-gate threshold Knee 2. 0x00 = –30 dB 0x02 = –33 dB 0x1E = –75 dB 0x01 = –31.5 dB … (–1.5dB steps) 0x1F = –76.5 dB Applicable only if DRC2_KNEE2_OP_ENA = 1.

The CS47L15 performs automatic checks to confirm that the SYSCLK frequency is high enough to support the commanded DRC and digital mixing functions. If the frequency is too low, an attempt to enable a DRC signal path fails. Note that active signal paths are not affected under such circumstances.

The FX_STS field in register R3585 indicates the status of each of the EQ, DRC, and LHPF signal paths. If an underclocked error condition occurs, this field indicates which EQ, DRC, or LHPF signal paths have been enabled.

The status bits in registers R1600–R2936 indicate the status of each of the digital mixers. If an underclocked error condition occurs, these bits indicate which mixers have been enabled.

4.3.6 Low-/High-Pass Digital Filter (LHPF)

The digital core provides four LHPF processing blocks as shown in Fig. 4-21. A four-input mixer is associated with each filter. The four input sources are selectable in each case, and independent volume control is provided for each path. Each LHPF block supports one output.

The LHPF /HPF can be used to remove unwanted out-of-band noise from a signal path. Each filter can be configured either as a low-pass filter (LPF) or a high-pass filter (HPF).

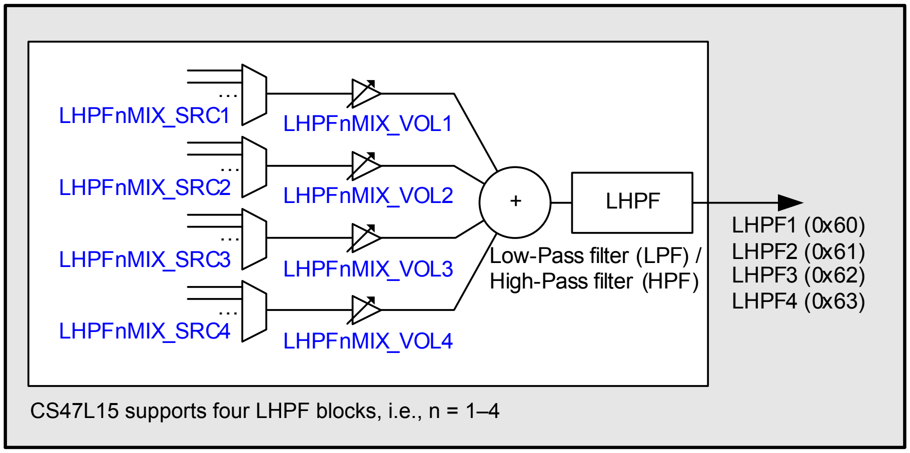


Figure 4-21. Digital-Core LPF/HPF Blocks

The LHPF1–LHPF4 mixer control fields, shown in Fig. 4-21, are located at register addresses R2304–R2335 (0x0900–0x091F).

The full list of digital mixer control registers (R1600–R2936) is provided in Section 6. Generic register field definitions are provided in Table 4-10.

The x_SRC*n* fields select the input sources for the respective LHPF processing blocks. Note that the selected input sources must be configured for the same sample rate as the LHPF to which they are connected. Sample-rate conversion functions are available to support flexible interconnectivity; see Section 4.3.14.

The hexadecimal numbers in Fig. 4-21 indicate the corresponding x_SRC*n* setting for selection of that signal as an input to another digital-core function.

The sample rate for the LHPF function is configured using FX_RATE; see Table 4-24. Note that the EQ, DRC, and LHPF functions must all be configured for the same sample rate. Sample-rate conversion is required when routing the LHPF signal paths to any signal chain that is configured for a different sample rate.

The FX_RATE field must not be changed if any of the associated x_SRC*n* fields is nonzero. The associated x_SRC*n* fields must be cleared before writing a new value to FX_RATE. A minimum delay of 125 µs must be allowed between clearing the x_SRC*n* fields and writing to FX_RATE. See Table 4-24 for details.

The control registers associated with the LHPF functions are described in Table 4-18.

The cut-off frequencies for the LHPF blocks are set using the coefficients held in registers R3777, R3781, R3785, and R3789 for LHPF1, LHPF2, LHPF3 and LHPF4 respectively. These coefficients are derived using tools provided in Cirrus Logic's WISCE evaluation board control software; please contact your Cirrus Logic representative for details.

Table 4-18. Low-Pass Filter/High-Pass Filter

Register Address	Bit	Label	Default	Description
R3585 (0x0E01) FX_Ctrl2	15:4	FX_STS[11:0]	0x00	LHPF, DRC, EQ Enable Status. Indicates the status of the respective signal-processing functions. Each bit is coded as follows: 0 = Disabled 1 = Enabled [11] = EQ4 [7] = DRC2 (Right) [3] = LHPF4 [10] = EQ3 [6] = DRC2 (Left) [2] = LHPF3 [9] = EQ2 [5] = DRC1 (Right) [1] = LHPF2 [8] = EQ1 [4] = DRC1 (Left) [0] = LHPF1
R3776 (0x0EC0) HPLPF1_1	1	LHPF1_MODE	0	Low-/High-Pass Filter 1 Mode 0 = Low Pass 1 = High Pass
	0	LHPF1_ENA	0	Low-/High-Pass Filter 1 Enable 0 = Disabled 1 = Enabled
R3777 (0x0EC1) HPLPF1_2	15:0	LHPF1_COEFF[15:0]	0x0000	Low-/High-Pass Filter 1 Frequency Coefficient Refer to WISCE evaluation board control software for the derivation of this field value.
R3780 (0x0EC4) HPLPF2_1	1	LHPF2_MODE	0	Low-/High-Pass Filter 2 Mode 0 = Low Pass 1 = High Pass
	0	LHPF2_ENA	0	Low-/High-Pass Filter 2 Enable 0 = Disabled 1 = Enabled
R3781 (0x0EC5) HPLPF2_2	15:0	LHPF2_COEFF[15:0]	0x0000	Low-/High-Pass Filter 2 Frequency Coefficient Refer to WISCE evaluation board control software for the derivation of this field value.
R3784 (0x0EC8) HPLPF3_1	1	LHPF3_MODE	0	Low-/High-Pass Filter 3 Mode 0 = Low Pass 1 = High Pass
	0	LHPF3_ENA	0	Low-/High-Pass Filter 3 Enable 0 = Disabled 1 = Enabled
R3785 (0x0EC9) HPLPF3_2	15:0	LHPF3_COEFF[15:0]	0x0000	Low-/High-Pass Filter 3 Frequency Coefficient Refer to WISCE evaluation board control software for the derivation of this field value.
R3788 (0x0ECC) HPLPF4_1	1	LHPF4_MODE	0	Low-/High-Pass Filter 4 Mode 0 = Low Pass 1 = High Pass
	0	LHPF4_ENA	0	Low-/High-Pass Filter 4 Enable 0 = Disabled 1 = Enabled
R3789 (0x0ECD) HPLPF4_2	15:0	LHPF4_COEFF[15:0]	0x0000	Low-/High-Pass Filter 4 Frequency Coefficient Refer to WISCE evaluation board control software for the derivation of this field value.

The CS47L15 performs automatic checks to confirm whether the SYSCLK frequency is high enough to support the commanded LHPF and digital mixing functions. If the frequency is too low, an attempt to enable an LHPF signal path fails. Note that active signal paths are not affected under such circumstances.

The FX_STS field in register R3585 indicates the status of each of the EQ, DRC, and LHPF signal paths. If an underclocked error condition occurs, this field indicates which EQ, DRC, or LHPF signal paths have been enabled.

The status bits in registers R1600–R2936 indicate the status of each of the digital mixers. If an underclocked error condition occurs, these bits indicate which mixers have been enabled.

4.3.7 Digital-Core DSP

The digital core provides one programmable DSP processing block as shown in Fig. 4-22. The DSP block supports eight inputs (Left, Right, Aux1, Aux2, … Aux6). A four-input mixer is associated with the left and right inputs, providing further expansion of the number of input paths. Each of the input sources is selectable, and independent volume control is provided for left and right input mixer channels. The DSP block supports six outputs.

The functionality of the DSP processing block is not fixed, and a wide range of audio enhancements algorithms may be performed. The procedure for configuring the CS47L15 DSP functions is tailored to each customer's application; please contact your Cirrus Logic representative for details.

For details of the DSP firmware requirements relating to clocking, register access, and code execution, refer to Section 4.4.3.

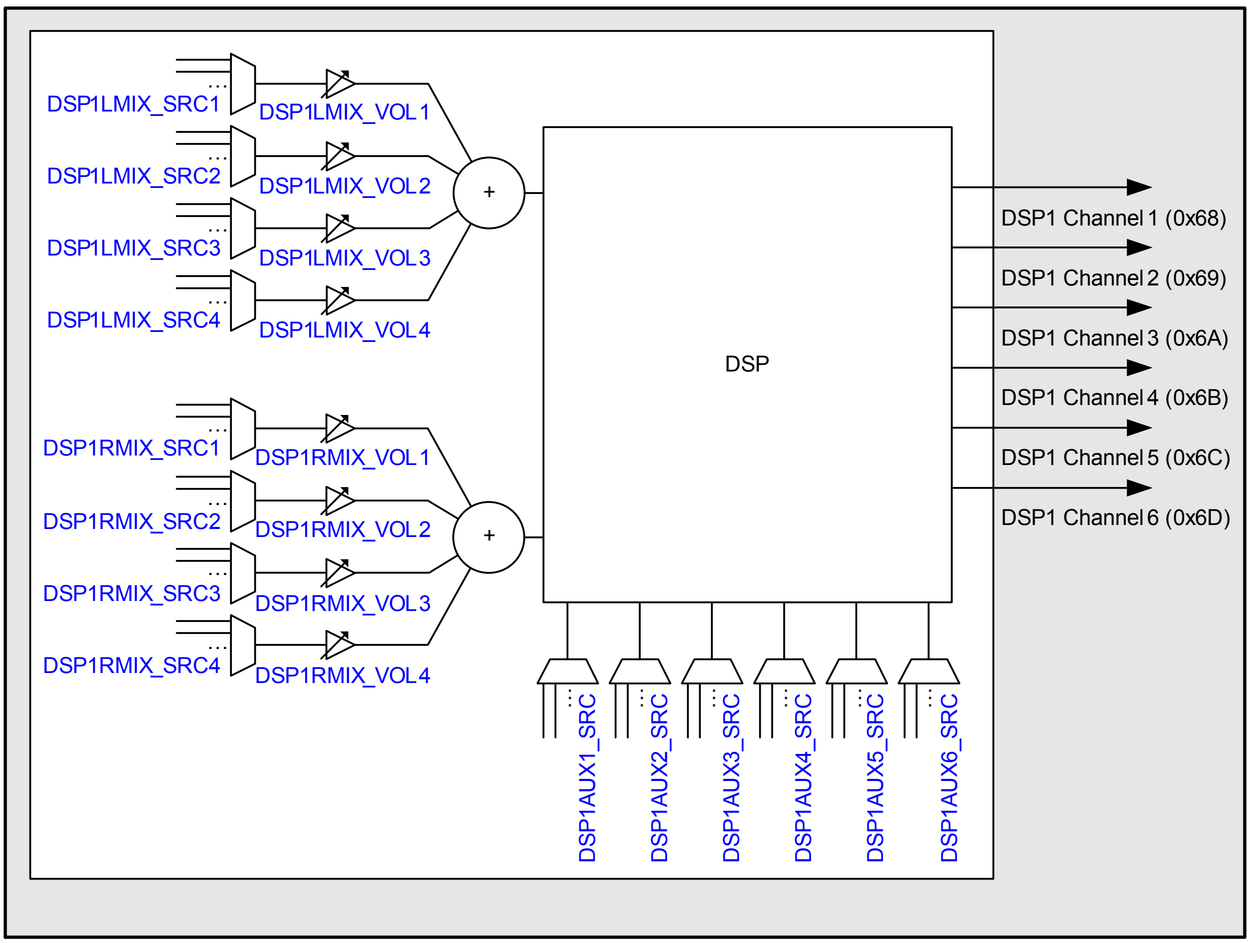


Figure 4-22. Digital-Core DSP Block

The DSP mixer input control fields (see Fig. 4-22) are located at register addresses R2368–R2424 (0x0940–0x0978).

The full list of digital mixer control registers (R1600–R2936) is provided in Section 6. Generic register field definitions are provided in Table 4-10.

The x_SRC*n* fields select the input sources for the DSP processing block. Note that the selected input sources must be configured for the same sample rate as the DSP. Sample-rate conversion functions are available to support flexible interconnectivity; see Section 4.3.14.

The hexadecimal numbers in Fig. 4-22 indicate the corresponding x_SRC*n* setting for selection of that signal as an input to another digital-core function.

The sample rate for the DSP functions is configured using the DSP1_RATE field; see Table 4-24. Sample-rate conversion is required when routing the DSP signal paths to any signal chain that is configured for a different sample rate.

The DSP1_RATE field must not be changed if any of the respective x_SRC*n* fields is nonzero. The associated x_SRC*n* fields must be cleared before writing new values to DSP1_RATE. A minimum delay of 125 µs must be allowed between clearing the x_SRC*n* fields and writing to the DSP1_RATE field. See Table 4-24 for details.

The CS47L15 performs automatic checks to confirm that the SYSCLK frequency is high enough to support the required DSP mixing functions. If the frequency is too low, an attempt to enable a DSP mixer path fails. Note that active signal paths are not affected under such circumstances.

The status bits in registers R1600–R2936 indicate the status of each of the digital mixers. If an underclocked error condition occurs, these bits indicate which mixers have been enabled.

4.3.8 S/PDIF Output Generator

The CS47L15 incorporates an IEC-60958-3–compatible S/PDIF output generator, as shown in Fig. 4-23; this provides a stereo S/PDIF output on a GPIO pin. The S/PDIF transmitter allows full control over the S/PDIF validity bits and channel status information.

The input sources to the S/PDIF transmitter are selectable for each channel, and independent volume control is provided for each path. The *TX1 and *TX2 fields control Channels A and B (respectively) of the S/PDIF output.

The S/PDIF signal can be output directly on a GPIO pin. See Section 4.11 to configure a GPIO pin for this function.

Note that the S/PDIF signal cannot be selected as input to the digital mixers or signal-processing functions within the CS47L15 digital core.

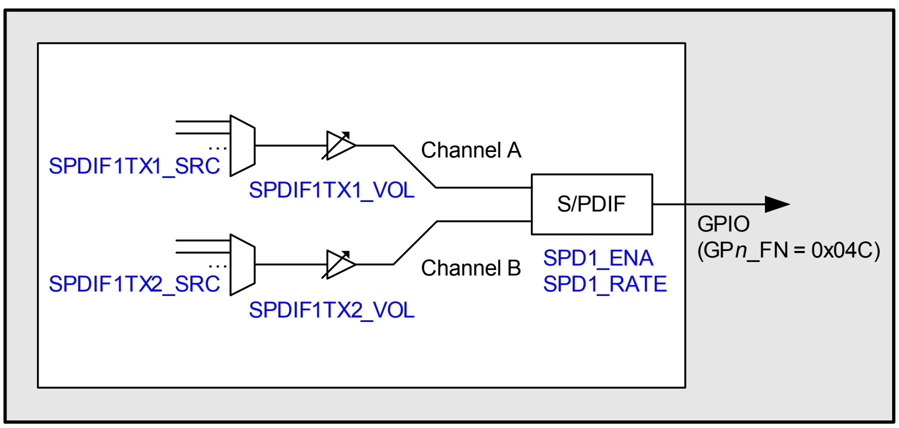


Figure 4-23. Digital-Core S/PDIF Output Generator

The S/PDIF input control fields (see Fig. 4-23) are located at register addresses R2048–R2057 (0x0800–0x0809).

The full list of digital mixer control registers (R1600–R2936) is provided in Section 6. Generic register field definitions are provided in Table 4-10.

The x_SRC*n* fields select the input sources for the two S/PDIF channels. Note that the selected input sources must be synchronized to the SYSCLK clocking domain, and configured for the same sample rate as the S/PDIF generator. Sample-rate conversion functions are available to support flexible interconnectivity; see Section 4.3.14.

The sample rate of the S/PDIF generator is configured using SPD1_RATE; see Table 4-24. The S/PDIF transmitter supports sample rates in the range 32–192 kHz. Note that sample-rate conversion is required when linking the S/PDIF generator to any signal chain that is configured for a different sample rate.

The SPD1_RATE field must not be changed if any of the associated x_SRC*n* fields is nonzero. The associated x_SRC*n* fields must be cleared before writing a new value to SPD1_RATE. A minimum delay of 125 µs must be allowed between clearing the x_SRC*n* fields and writing to SPD1_RATE. See Table 4-24 for details.

The S/PDIF generator is enabled by setting SPD1_ENA, as described in Table 4-19.

The S/PDIF output contains audio data derived from the selected sources. Audio samples up to 24-bit width can be accommodated. The validity bits and the channel status bits in the S/PDIF data are configured using the corresponding fields in registers R1474 (0x5C2) to R1477 (0x5C5).

Refer to the S/PDIF specification (IEC 60958-3 Digital Audio Interface - Consumer) for full details of the S/PDIF protocol and configuration parameters.

Table 4-19. S/PDIF Output Generator Control

Register Address	Bit	Label	Default	Description
R1474 (0x05C2) SPD1_TX_Control	13	SPD1_VAL2	0	S/PDIF Validity (Subframe B)
	12	SPD1_VAL1	0	S/PDIF Validity (Subframe A)
	0	SPD1_ENA	0	S/PDIF Generator Enable 0 = Disabled 1 = Enabled
R1475 (0x05C3) SPD1_TX_ Channel_Status_1	15:8	SPD1_CATCODE[7:0]	0x00	S/PDIF Category code
	7:6	SPD1_CHSTMODE[1:0]	00	S/PDIF Channel Status mode
	5:3	SPD1_PREEMPH[2:0]	000	S/PDIF Preemphasis mode
	2	SPD1_NOCOPY	0	S/PDIF Copyright status
	1	SPD1_NOAUDIO	0	S/PDIF Audio/nonaudio indication
	0	SPD1_PRO	0	S/PDIF Consumer Mode/Professional Mode
R1476 (0x05C4) SPD1_TX_ Channel_Status_2	15:12	SPD1_FREQ[3:0]	0000	S/PDIF Indicated sample frequency
	11:8	SPD1_CHNUM2[3:0]	1011	S/PDIF Channel number (Subframe B)
	7:4	SPD1_CHNUM1[3:0]	0000	S/PDIF Channel number (Subframe A)
	3:0	SPD1_SRCNUM[3:0]	0001	S/PDIF Source number
R1477 (0x05C5) SPD1_TX_ Channel_Status_3	11:8	SPD1_ORGSAMP[3:0]	0000	S/PDIF Original sample frequency
	7:5	SPD1_TXWL[2:0]	000	S/PDIF Audio sample word length
	4	SPD1_MAXWL	0	S/PDIF Maximum audio sample word length
	3:2	SPD1_SC31_30[1:0]	00	S/PDIF Channel Status [31:30]
	1:0	SPD1_CLKACU[1:0]	00	Transmitted Clock accuracy

The CS47L15 automatically checks to confirm whether the SYSCLK frequency is high enough to support the digital mixer paths. If an attempt is made to enable the S/PDIF generator, and there are insufficient SYSCLK cycles to support it, the attempt does not succeed. Note that any active signal paths are unaffected under such circumstances.

The status bits in registers R1600–R2936 indicate the status of each of the digital mixers. If an underclocked error condition occurs, these bits indicate which mixers have been enabled.

4.3.9 Tone Generator

The CS47L15 incorporates a tone generator that can be used for beep functions through any of the audio signal paths. The tone generator provides two 1-kHz outputs, with configurable phase relationship, offering flexibility to create differential signals or test scenarios.

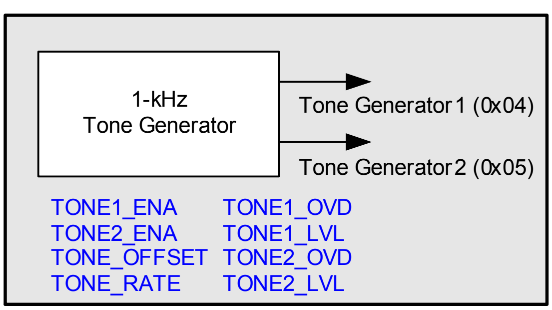


Figure 4-24. Digital-Core Tone Generator

The tone generator outputs can be selected as input to any of the digital mixers or signal-processing functions within the CS47L15 digital core. The hexadecimal numbers in Fig. 4-24 indicate the corresponding x_SRC*n* setting for selection of that signal as an input to another digital-core function.

The sample rate for the tone generator is configured using TONE_RATE. See Table 4-24. Note that sample-rate conversion is required when routing the tone generator outputs to any signal chain that is configured for a different sample rate.

The tone generator outputs are enabled by setting the TONE1_ENA and TONE2_ENA bits as described in Table 4-20. The phase relationship is configured using TONE_OFFSET.

The tone generator outputs can also provide a configurable DC signal level, for use as a test signal. The DC output is selected using the TONE*n*_OVD bits, and the DC signal amplitude is configured using the TONE*n*_LVL fields, as described in Table 4-20.

Table 4-20. Tone Generator Control

Register Address	Bit	Label	Default	Description
R32 (0x0020) Tone_Generator_1	9:8	TONE_OFFSET[1:0]	00	Tone Generator Phase Offset. Sets the phase of Tone Generator 2 relative to Tone Generator 1 00 = 0 degrees (in phase) 01 = 90 degrees ahead 10 = 180 degrees ahead 11 = 270 degrees ahead
	5	TONE2_OVD	0	Tone Generator 2 Override 0 = Disabled (1-kHz tone output) 1 = Enabled (DC signal output) The DC signal level, when selected, is configured using TONE2_LVL[23:0]
	4	TONE1_OVD	0	Tone Generator 1 Override 0 = Disabled (1-kHz tone output) 1 = Enabled (DC signal output) The DC signal level, when selected, is configured using TONE1_LVL[23:0]
	1	TONE2_ENA	0	Tone Generator 2 Enable 0 = Disabled 1 = Enabled
	0	TONE1_ENA	0	Tone Generator 1 Enable 0 = Disabled 1 = Enabled
R33 (0x0021) Tone_Generator_2	15:0	TONE1_LVL[23:8]	0x1000	Tone Generator 1 DC output level TONE1_LVL[23:8] is coded as 2’s complement. Bits [23:20] contain the integer portion; bits [19:0] contain the fractional portion. The digital core 0 dBFS level corresponds to 0x10_0000 (+1) or 0xF0_0000 (–1).
R34 (0x0022) Tone_Generator_3	7:0	TONE1_LVL[7:0]	0x00	Tone Generator 1 DC output level TONE1_LVL[23:8] is coded as 2’s complement. Bits [23:20] contain the integer portion; bits [19:0] contain the fractional portion. The digital core 0 dBFS level corresponds to 0x10_0000 (+1) or 0xF0_0000 (–1).
R35 (0x0023) Tone_Generator_4	15:0	TONE2_LVL[23:8]	0x1000	Tone Generator 2 DC output level TONE2_LVL[23:8] is coded as 2’s complement. Bits [23:20] contain the integer portion; bits [19:0] contain the fractional portion. The digital core 0 dBFS level corresponds to 0x10_0000 (+1) or 0xF0_0000 (–1).
R36 (0x0024) Tone_Generator_5	7:0	TONE2_LVL[7:0]	0x00	Tone Generator 2 DC output level TONE2_LVL[23:8] is coded as 2’s complement. Bits [23:20] contain the integer portion; bits [19:0] contain the fractional portion. The digital core 0 dBFS level corresponds to 0x10_0000 (+1) or 0xF0_0000 (–1).

4.3.10 Noise Generator

The CS47L15 incorporates a white-noise generator that can be routed within the digital core. The main purpose of the noise generator is to provide comfort noise in cases where silence (digital mute) is not desirable.

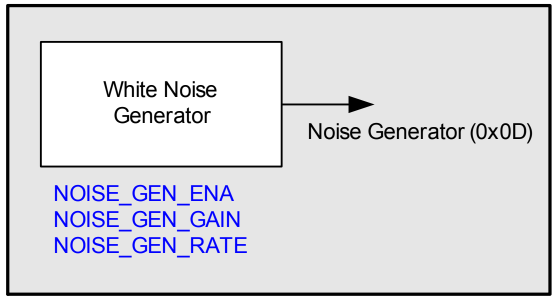


Figure 4-25. Digital-Core Noise Generator

The noise generator can be selected as input to any of the digital mixers or signal-processing functions within the CS47L15 digital core. The hexadecimal number (0x0D) in Fig. 4-25 indicates the corresponding x_SRC*n* setting for selection of the noise generator as an input to another digital-core function.

The sample rate for the noise generator is configured using the NOISE_GEN_RATE field. See Table 4-24. Note that sample-rate conversion is required when routing the noise generator output to any signal chain that is configured for a different sample rate.

The noise generator is enabled by setting NOISE_GEN_ENA, described in Table 4-21. The signal level is configured using NOISE_GEN_GAIN.

Table 4-21. Noise Generator Control

Register Address	Bit	Label	Default	Description
R160 (0x00A0) Comfort_Noise_ Generator	5	NOISE_GEN_ ENA	0	Noise Generator Enable 0 = Disabled 1 = Enabled
	4:0	NOISE_GEN_ GAIN[4:0]	0x00	Noise generator signal level 0x00 = –114 dBFS ...(6-dB steps) All other codes are reserved 0x01 = –108 dBFS 0x11 = –6 dBFS 0x02 = –102 dBFS 0x12 = 0 dBFS

4.3.11 Haptic Signal Generator

The CS47L15 incorporates a signal generator for use with haptic devices (e.g., mechanical vibration actuators). The haptic signal generator is compatible with both eccentric rotating mass (ERM) and linear resonant actuator (LRA) haptic devices.

The haptic signal generator is highly configurable, and includes the capability to execute a programmable event profile comprising three distinct operating phases.

The resonant frequency of the haptic signal output (for LRA devices) is selectable, providing support for many different actuator components.

The haptic signal generator is a digital signal generator, which is incorporated within the digital core of the CS47L15. The haptic signal may be routed, via one of the digital-core output mixers, to a Class D speaker output for connection to the external haptic device, as shown in Fig. 4-26. Note that the digital PDM output paths may also be used for haptic signal output.

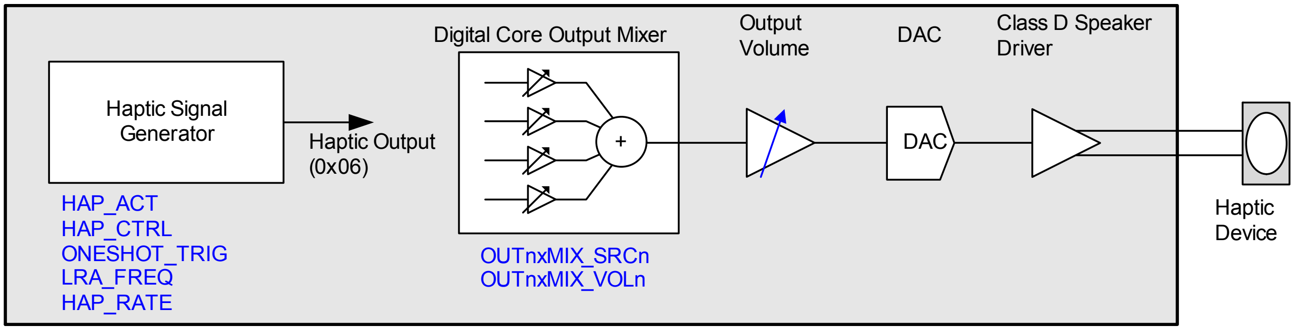


Figure 4-26. Digital-Core Haptic Signal Generator

The hexadecimal number (0x06) in Fig. 4-26 indicates the corresponding x_SRC*n* setting for selection of the haptic signal generator as an input to another digital-core function.

The haptic signal generator is selected as input to one of the digital-core output mixers by setting the x_SRC*n* field of the applicable output mixer to 0x06.

The sample rate for the haptic signal generator is configured using the HAP_RATE field. See Table 4-22. Note that sample-rate conversion is required when routing the haptic signal generator output to any signal chain that is configured for a different sample rate.

The haptic signal generator is configured for an ERM or LRA actuator using the HAP_ACT bit. The required resonant frequency is configured using the LRA_FREQ field. Note that the resonant frequency is only applicable to LRA actuators.

The signal generator can be enabled in continuous mode or configured for one-shot mode using the HAP_CTRL field, as described in Table 4-22. In one-shot mode, the output is triggered by writing to the ONESHOT_TRIG bit.

In one-shot mode, the signal generator profile comprises the distinct phases (1, 2, 3). The duration and intensity of each output phase is programmable.

In continuous mode, the signal intensity is controlled using the PHASE2_INTENSITY field only.

In the case of an ERM actuator (HAP_ACT = 0), the haptic output is a DC signal level, which may be positive or negative, as selected by the x_INTENSITY fields.

For an LRA actuator (HAP_ACT = 1), the haptic output is an AC signal; selecting a negative signal level corresponds to a 180° phase inversion. In some applications, phase inversion may be desirable during the final phase, to halt the physical motion of the haptic device.

Table 4-22. Haptic Signal Generator Control

Register Address	Bit	Label	Default	Description
R144 (0x0090) Haptics_Control_1	4	ONESHOT_TRIG	0	Haptic One-Shot Trigger. Writing 1 starts the one-shot profile (i.e., Phase 1, Phase 2, Phase 3)
	3:2	HAP_CTRL[1:0]	00	Haptic Signal Generator Control 00 = Disabled 10 = One-Shot 01 = Continuous 11 = Reserved
	1	HAP_ACT	0	Haptic Actuator Select 0 = Eccentric rotating mass (ERM) 1 = Linear resonant actuator (LRA)
R145 (0x0091) Haptics_Control_2	14:0	LRA_FREQ[14:0]	0x7FFF	Haptic Resonant Frequency. Selects the haptic signal frequency (LRA actuator only, HAP_ACT = 1) Haptic Frequency (Hz) = System Clock/(2 x (LRA_FREQ+1)), where System Clock = 6.144 MHz or 5.6448 MHz, derived by division from SYSCLK. Valid for haptic frequency in the range 100–250 Hz For 6.144-MHz System Clock: For 5.6448-MHz System Clock: 0x77FF = 100 Hz 0x6E3F = 100 Hz 0x4491 = 175 Hz 0x3EFF = 175 Hz 0x2FFF = 250 Hz 0x2C18 = 250 Hz

Table 4-22. Haptic Signal Generator Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R146 (0x0092) Haptics_phase_1_intensity	7:0	PHASE1_INTENSITY[7:0]	0x00	Haptic Output Level (Phase 1). Selects the signal intensity of Phase 1 in one-shot mode. Coded as 2's complement. Range is ± Full Scale (FS). For ERM actuator, this selects the DC signal level for the haptic output. For LRA actuator, this selects the AC peak amplitude; negative values correspond to a 180° phase shift.
R147 (0x0093) Haptics_Control_phase_1_duration	8:0	PHASE1_DURATION[8:0]	0x000	Haptic Output Duration (Phase 1). Selects the duration of Phase 1 in one-shot mode. 0x000 = 0 ms 0x002 = 1.25 ms 0x1FF = 319.375 ms 0x001 = 0.625 ms … (0.625-ms steps)
R148 (0x0094) Haptics_phase_2_intensity	7:0	PHASE2_INTENSITY[7:0]	0x00	Haptic Output Level (Phase 2) Selects the signal intensity in Continuous mode or Phase 2 of one-shot mode. Coded as 2's complement. Range is ± Full Scale (FS). For ERM actuator, this selects the DC signal level for the haptic output. For LRA actuator, this selects the AC peak amplitude; negative values correspond to a 180° phase shift.
R149 (0x0095) Haptics_phase_2_duration	10:0	PHASE2_DURATION[10:0]	0x000	Haptic Output Duration (Phase 2). Selects the duration of Phase 2 in one-shot mode. 0x000 = 0 ms 0x002 = 1.25 ms 0x7FF = 1279.375 ms 0x001 = 0.625 ms … (0.625-ms steps)
R150 (0x0096) Haptics_phase_3_intensity	7:0	PHASE3_INTENSITY[7:0]	0x00	Haptic Output Level (Phase 3). Selects the signal intensity of Phase 3 in one-shot mode. Coded as 2's complement. Range is ± Full Scale (FS). For ERM actuator, this selects the DC signal level for the haptic output. For LRA actuator, this selects the AC peak amplitude; negative values correspond to a 180° phase shift.
R151 (0x0097) Haptics_phase_3_duration	8:0	PHASE3_DURATION[8:0]	0x000	Haptic Output Duration (Phase 3). Selects the duration of Phase 3 in one-shot mode. 0x000 = 0 ms 0x002 = 1.25 ms 0x1FF = 319.375 ms 0x001 = 0.625 ms … (0.625-ms steps)
R152 (0x0098) Haptics_Status	0	ONESHOT_STS	0	Haptic One-Shot status 0 = One-Shot event not in progress 1 = One-Shot event in progress

4.3.12 PWM Generator

The CS47L15 incorporates two PWM signal generators as shown in Fig. 4-27. The duty cycle of each PWM signal can be modulated by an audio source, or can be set to a fixed value using a control register setting.

A four-input mixer is associated with each PWM generator. The four input sources are selectable in each case, and independent volume control is provided for each path.

PWM signal generators can be output directly on a GPIO pin. See Section 4.11 to configure a GPIO pin for this function.

Note that the PWM signal generators cannot be selected as input to the digital mixers or signal-processing functions within the CS47L15 digital core.

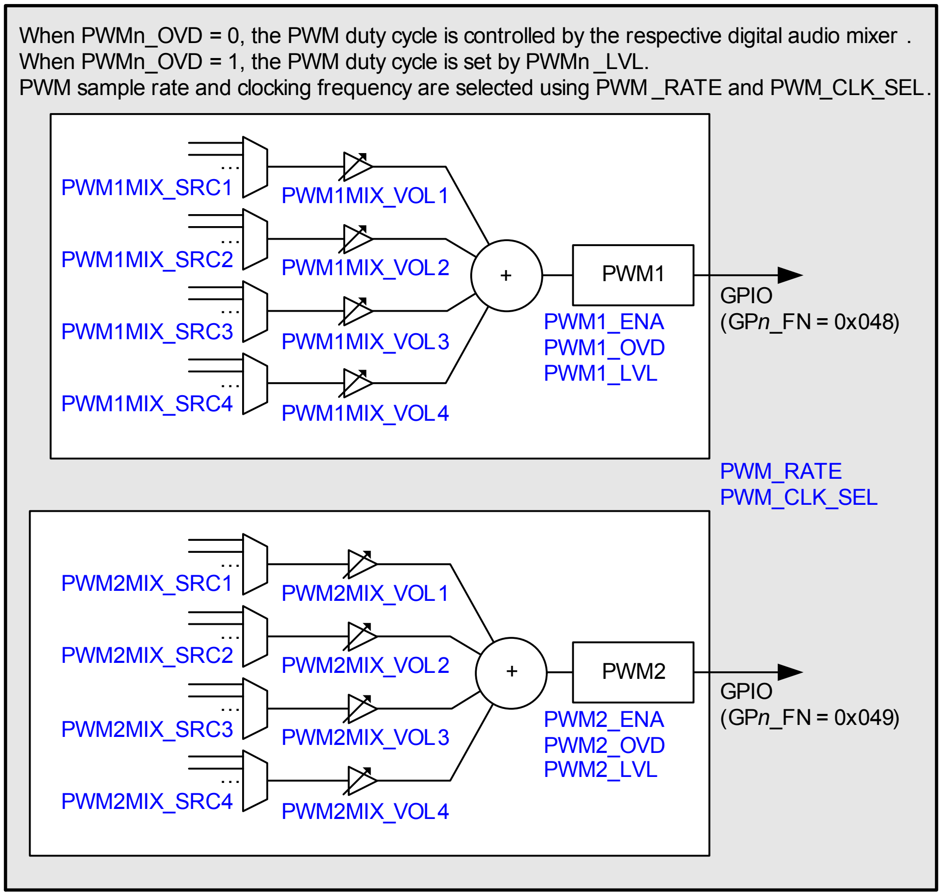


Figure 4-27. Digital-Core PWM Generator

The PWM1 and PWM2 mixer control fields (see Fig. 4-27) are located at register addresses R1600–R1615 (0x0640–0x064F).

The full list of digital mixer control registers (R1600–R2936) is provided in Section 6. Generic register field definitions are provided in Table 4-10.

The x_SRC*n* fields select the input sources for the respective mixers. Note that the selected input sources must be configured for the same sample rate as the mixer to which they are connected. Sample-rate conversion functions are available to support flexible interconnectivity; see Section 4.3.14.

The PWM sample rate (cycle time) is configured using PWM_RATE. See Table 4-24. Note that sample-rate conversion is required when linking the PWM generators to any signal chain that is configured for a different sample rate.

The PWM_RATE field must not be changed if any of the associated x_SRC*n* fields is nonzero. The associated x_SRC*n* fields must be cleared before writing a new value to PWM_RATE. A minimum delay of 125 µs must be allowed between clearing the x_SRC*n* fields and writing to PWM_RATE. See Table 4-24 for details.

The PWM generators are enabled by setting PWM1_ENA and PWM2_ENA, respectively, as described in Table 4-23.

Under default conditions (PWM*n*_OVD = 0), the duty cycle of the PWM generators is controlled by an audio signal path; a 4-input mixer is associated with each PWM generator, as shown in Fig. 4-27.

When the PWM*n*_OVD bit is set, the duty cycle of the respective PWM generator is set to a fixed ratio; in this case, the duty cycle ratio is configurable using the PWM*n*_LVL fields.

The PWM generator clock frequency is selected using PWM_CLK_SEL. For best performance, the highest available setting should be used. Note that the PWM generator clock must not be set to a higher frequency than SYSCLK.

Table 4-23. PWM Generator Control

Register Address	Bit	Label	Default	Description
R48 (0x0030) PWM_Drive_1	10:8	PWM_CLK_SEL[2:0]	000	PWM Clock Select 000 = 6.144 MHz (5.6448 MHz) 001 = 12.288 MHz (11.2896 MHz) 010 = 24.576 MHz (22.5792 MHz) All other codes are reserved. The frequencies in brackets apply for 44.1 kHz–related sample rates only. PWM_CLK_SEL controls the resolution of the PWM generator; higher settings correspond to higher resolution. The PWM Clock must be less than or equal to SYSCLK.
	5	PWM2_OVD	0	PWM2 Generator Override 0 = Disabled (PWM duty cycle is controlled by audio source) 1 = Enabled (PWM duty cycle is controlled by PWM2_LVL).
	4	PWM1_OVD	0	PWM1 Generator Override 0 = Disabled (PWM1 duty cycle is controlled by audio source) 1 = Enabled (PWM1 duty cycle is controlled by PWM1_LVL).
	1	PWM2_ENA	0	PWM2 Generator Enable 0 = Disabled 1 = Enabled
	0	PWM1_ENA	0	PWM1 Generator Enable 0 = Disabled 1 = Enabled
R49 (0x0031) PWM_Drive_2	9:0	PWM1_LVL[9:0]	0x100	PWM1 Override Level. Sets the PWM1 duty cycle when PWM1_OVD = 1. Coded as 2’s complement. 0x000 = 50% duty cycle 0x200 = 0% duty cycle
R50 (0x0032) PWM_Drive_3	9:0	PWM2_LVL[9:0]	0x100	PWM2 Override Level. Sets the PWM2 duty cycle when PWM2_OVD = 1. Coded as 2’s complement. 0x000 = 50% duty cycle 0x200 = 0% duty cycle

The CS47L15 automatically checks to confirm that the SYSCLK frequency is high enough to support the digital mixer paths. If an attempt is made to enable a PWM signal mixer path, without sufficient SYSCLK cycles to support it, the attempt fails. Note that any signal paths that are already active are not affected under such circumstances.

The status bits in registers R1600–R2936 indicate the status of each of the digital mixers. If an underclocked error condition occurs, these bits indicate which mixers have been enabled.

4.3.13 Sample-Rate Control

The CS47L15 supports multiple signal paths through the digital core. Stereo full-duplex sample-rate conversion is provided to allow digital audio to be routed between interfaces operating at different sample rates.

The master clock reference for the audio signal paths is SYSCLK, as described in Section 4.13. Every digital signal path must be synchronized to SYSCLK.

Up to three different sample rates may be in use at any time on the CS47L15; all of these sample rates must be synchronized to SYSCLK.

Sample-rate conversion is required when routing any audio path between digital functions that are configured for different sample rates.

There are two isochronous sample-rate converters: ISRC1 and ISRC2. Each ISRC supports two-way, four-channel conversion paths between sample rates on the SYSCLK domain. The ISRCs are described in Section 4.3.14.

The sample rate of different blocks within the CS47L15 digital core are controlled as shown in Fig. 4-28. The x_RATE fields select the applicable sample rate for each respective group of digital functions.

The x_RATE fields must not be changed if any of the x_SRC*n* fields associated with the respective functions is nonzero. The associated x_SRC*n* fields must be cleared before writing new values to the x_RATE fields. A minimum delay of 125 µs must be allowed between clearing the x_SRC*n* fields and writing to the associated x_RATE fields. See Table 4-24 for details.

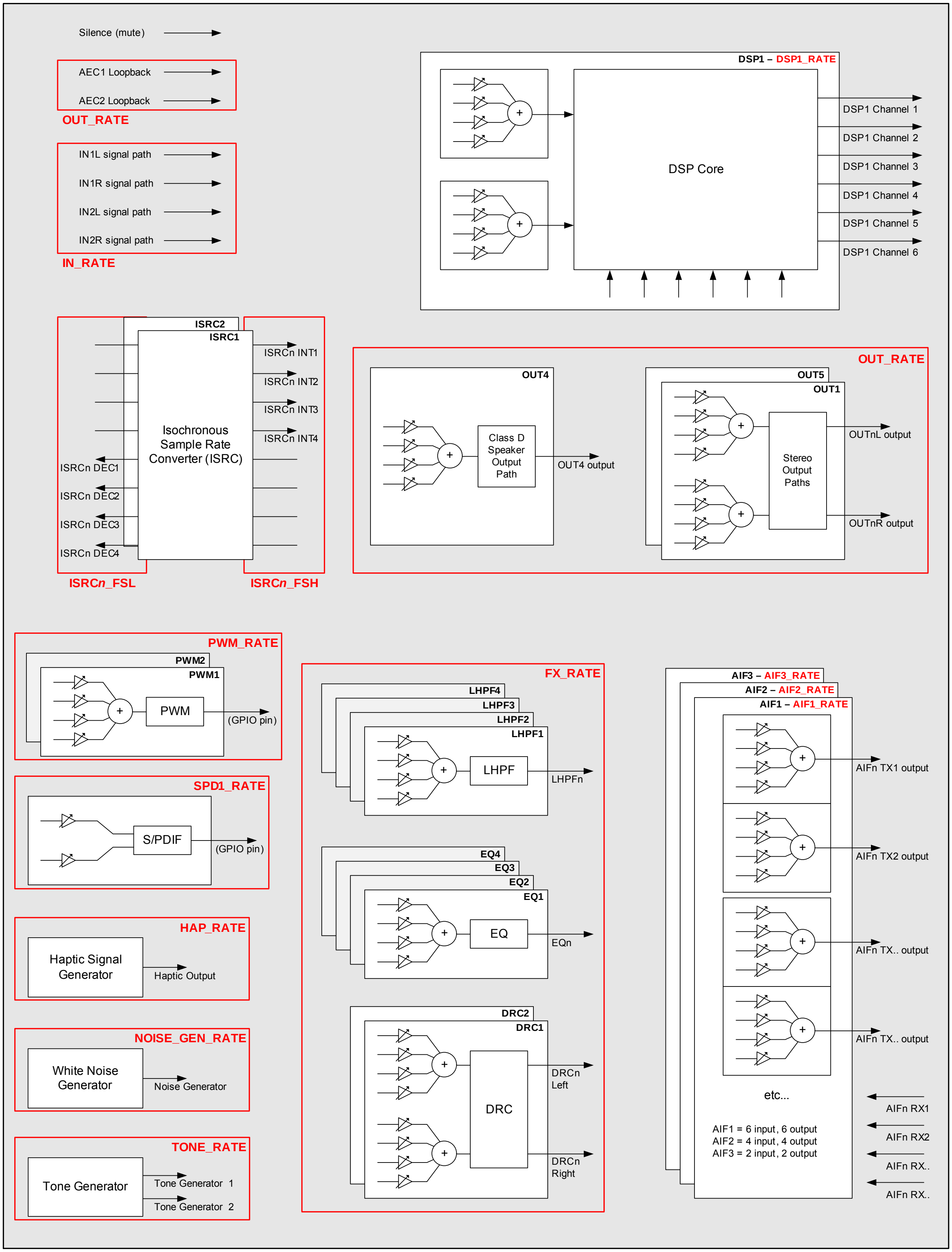


Figure 4-28. Digital-Core Sample-Rate Control

The input signal paths may be selected as input to the digital mixers or signal-processing functions. The sample rate for the input signal paths is configured using the IN_RATE field.

The output signal paths are derived from the respective output mixers. The sample rate for the output signal paths is configured using OUT_RATE. The sample rate of the AEC loop-back path is also set by OUT_RATE.

The AIF*n* RX inputs may be selected as input to the digital mixers or signal-processing functions. The AIF*n* TX outputs are derived from the respective output mixers. The sample rates for digital audio interfaces (AIF1–AIF3) are configured using the AIF*n*_RATE fields (where *n* identifies the applicable AIF 1, 2, or 3) respectively.

The EQ, DRC, and LHPF functions can be enabled in any signal path within the digital core. The sample rate for these functions is configured using FX_RATE. Note that the EQ, DRC, and LHPF functions must all be configured for the same sample rate.

The DSP functions can be enabled in any signal path within the digital core. The applicable sample rate is configured using the DSP1_RATE field.

The S/PDIF transmitter can be enabled on a GPIO pin. Stereo inputs to this function can be configured from any of the digital-core inputs, mixers, or signal-processing functions. The sample rate of the S/PDIF transmitter is configured using SPD1_RATE.

The tone generators and noise generator can be selected as input to any of the digital mixers or signal-processing functions. The sample rates for these sources are configured using the TONE_RATE and NOISE_GEN_RATE fields, respectively.

The haptic signal generator can be used to control an external vibe actuator, which can be driven directly by the Class D speaker output. The sample rate for the haptic signal generator is configured using HAP_RATE.

The PWM signal generators can be modulated by an audio source, derived from the associated signal mixers. The sample rate (cycle time) for the PWM signal generators is configured using PWM_RATE.

The sample-rate control registers are described in Table 4-24. Refer to the field descriptions for details of the valid selections in each case. The control registers associated with the ISRCs are described in Table 4-25.

Note that 32-bit register addressing is used from R12888 (0x3000) upwards; 16-bit format is used otherwise. The registers noted in Table 4-24 contain a mixture of 16-bit and 32-bit register addresses.

Table 4-24. Digital-Core Sample-Rate Control

Register Address	Bit	Label	Default	Description
R32 (0x0020) Tone_Generator_1	14:11	TONE_RATE[3:0]	0000	Tone Generator Sample Rate 0000 = SAMPLE_RATE_1 0001 = SAMPLE_RATE_2 0010 = SAMPLE_RATE_3 All other codes are reserved. The selected sample rate is valid in the range 8–192 kHz.
R48 (0x0030) PWM_Drive_1	14:11	PWM_RATE[3:0]	0000	PWM Frequency (sample rate) 0000 = SAMPLE_RATE_1 0001 = SAMPLE_RATE_2 0010 = SAMPLE_RATE_3 All other codes are reserved. The selected sample rate is valid in the range 8–192 kHz. All PWM*n*MIX_SRC*m* fields must be cleared before changing PWM_RATE.
R144 (0x0090) Haptics_Control_1	14:11	HAP_RATE[3:0]	0000	Haptic Signal Generator Sample Rate 0000 = SAMPLE_RATE_1 0001 = SAMPLE_RATE_2 0010 = SAMPLE_RATE_3 All other codes are reserved. The selected sample rate is valid in the range 8–192 kHz.

Table 4-24. Digital-Core Sample-Rate Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R160 (0x00A0) Comfort_Noise_Generator	14:11	NOISE_GEN_RATE[3:0]	0000	Noise Generator Sample Rate 0000 = SAMPLE_RATE_1 0001 = SAMPLE_RATE_2 0010 = SAMPLE_RATE_3 All other codes are reserved. The selected sample rate is valid in the range 8–192 kHz.
R776 (0x0308) Input_Rate	14:11	IN_RATE[3:0]	0000	Input Signal Paths Sample Rate 0000 = SAMPLE_RATE_1 0001 = SAMPLE_RATE_2 0010 = SAMPLE_RATE_3 All other codes are reserved. The selected sample rate is valid in the range 8–192 kHz. If 384 kHz/768 kHz DMIC rate is selected (IN1_OSR = 01X), the input paths sample rate is valid up to 48 kHz/96 kHz respectively.
R1032 (0x0408) Output_Rate_1	14:11	OUT_RATE[3:0]	0000	Output Signal Paths Sample Rate 0000 = SAMPLE_RATE_1 0001 = SAMPLE_RATE_2 0010 = SAMPLE_RATE_3 All other codes are reserved. The selected sample rate is valid in the range 8–192 kHz. All OUT*nx*MIX_SRC*m* fields must be cleared before changing OUT_RATE.
R1283 (0x0503) AIF1_Rate_Ctrl	14:11	AIF1_RATE[3:0]	0000	AIF*n* Audio Interface Sample Rate 0000 = SAMPLE_RATE_1 0001 = SAMPLE_RATE_2 0010 = SAMPLE_RATE_3 All other codes are reserved. The selected sample rate is valid in the range 8–192 kHz. All AIF*n*TXMIX_SRC*m* fields must be cleared before changing AIF*n*_RATE.
R1347 (0x0543) AIF2_Rate_Ctrl	14:11	AIF2_RATE[3:0]	0000	
R1411 (0x0583) AIF3_Rate_Ctrl	14:11	AIF3_RATE[3:0]	0000	
R1474 (0x05C2) SPD1_TX_Control	7:4	SPD1_RATE[3:0]	0000	S/PDIF Transmitter Sample Rate 0000 = SAMPLE_RATE_1 0001 = SAMPLE_RATE_2 0010 = SAMPLE_RATE_3 All other codes are reserved. The selected sample rate is valid in the range 32–192 kHz. All SPDIF1TX*n*_SRC fields must be cleared before changing SPD1_RATE.
R3584 (0x0E00) FX_Ctrl1	14:11	FX_RATE[3:0]	0000	FX Sample Rate (EQ, LHPF, DRC) 0000 = SAMPLE_RATE_1 0001 = SAMPLE_RATE_2 0010 = SAMPLE_RATE_3 All other codes are reserved. The selected sample rate is valid in the range 8–192 kHz. All EQ*n*MIX_SRC*m*, DRC*nx*MIX_SRC*m*, and LHPF*n*MIX_SRC*m* fields must be cleared before changing FX_RATE.
R1048064 (0x0F_FE00) DSP1_Config_1	14:11	DSP1_RATE[3:0]	0000	DSP1 Sample Rate 0000 = SAMPLE_RATE_1 0001 = SAMPLE_RATE_2 0010 = SAMPLE_RATE_3 All other codes are reserved. The selected sample rate is valid in the range 8–192 kHz. All DSP1xMIX_SRC*m* fields must be cleared before changing DSP1_RATE.

4.3.14 Isochronous Sample-Rate Converter (ISRC)

The CS47L15 supports multiple signal paths through the digital core. The ISRCs provide sample-rate conversion between synchronized sample rates on the SYSCLK clock domain.

There are two ISRCs on the CS47L15. Each ISRC provides four signal paths between two different sample rates, as shown in Fig. 4-29. The sample rates associated with each ISRC can each be set equal to SAMPLE_RATE_1, SAMPLE_RATE_2, or SAMPLE_RATE_3. See Section 4.13 for details of the sample-rate control registers.

Each ISRC supports sample rates in the range 8–192 kHz. The higher of the sample rates associated with each ISRC must be an integer multiple of the lower sample rate; all possible integer ratios are supported (i.e., up to 24).

Each ISRC converts between a sample rate selected by ISRC*n*_FSL and a sample rate selected by ISRC*n*_FSH, (where *n* identifies the applicable ISRC 1 or 2). Note that, in each case, the higher of the two sample rates must be selected by ISRC*n*_FSH.

The ISRC*n*_FSL and ISRC*n*_FSH fields must not be changed if any of the respective x_SRC*n* fields is nonzero. The associated x_SRC*n* fields must be cleared before writing new values to ISRC*n*_FSL or ISRC*n*_FSH. A minimum delay of 125 µs must be allowed between clearing the x_SRC*n* fields and writing to the associated ISRC*n*_FSL or ISRC*n*_FSH fields. See Table 4-25 for details.

The ISRC signal paths are enabled using the ISRC*n*_INT*m*_ENA and ISRC*n*_DEC*m*_ENA bits, as follows:

- The ISRC*n* interpolation paths (increasing sample rate) are enabled by setting the ISRC*n*_INT*m*_ENA bits, (where *m* identifies the applicable channel).
- The ISRC*n* decimation paths (decreasing sample rate) are enabled by setting the ISRC*n*_DEC*m*_ENA bits.

The CS47L15 performs automatic checks to confirm that the SYSCLK frequency is high enough to support the commanded ISRC and digital mixing functions. If the frequency is too low, an attempt to enable an ISRC signal path fails. Note that active signal paths are not affected under such circumstances.

The status bits in registers R1600–R2936 indicate the status of each of the digital mixers. If an underclocked error condition occurs, these bits indicate which mixers have been enabled.

The ISRC signal paths and control registers are shown in Fig. 4-29.

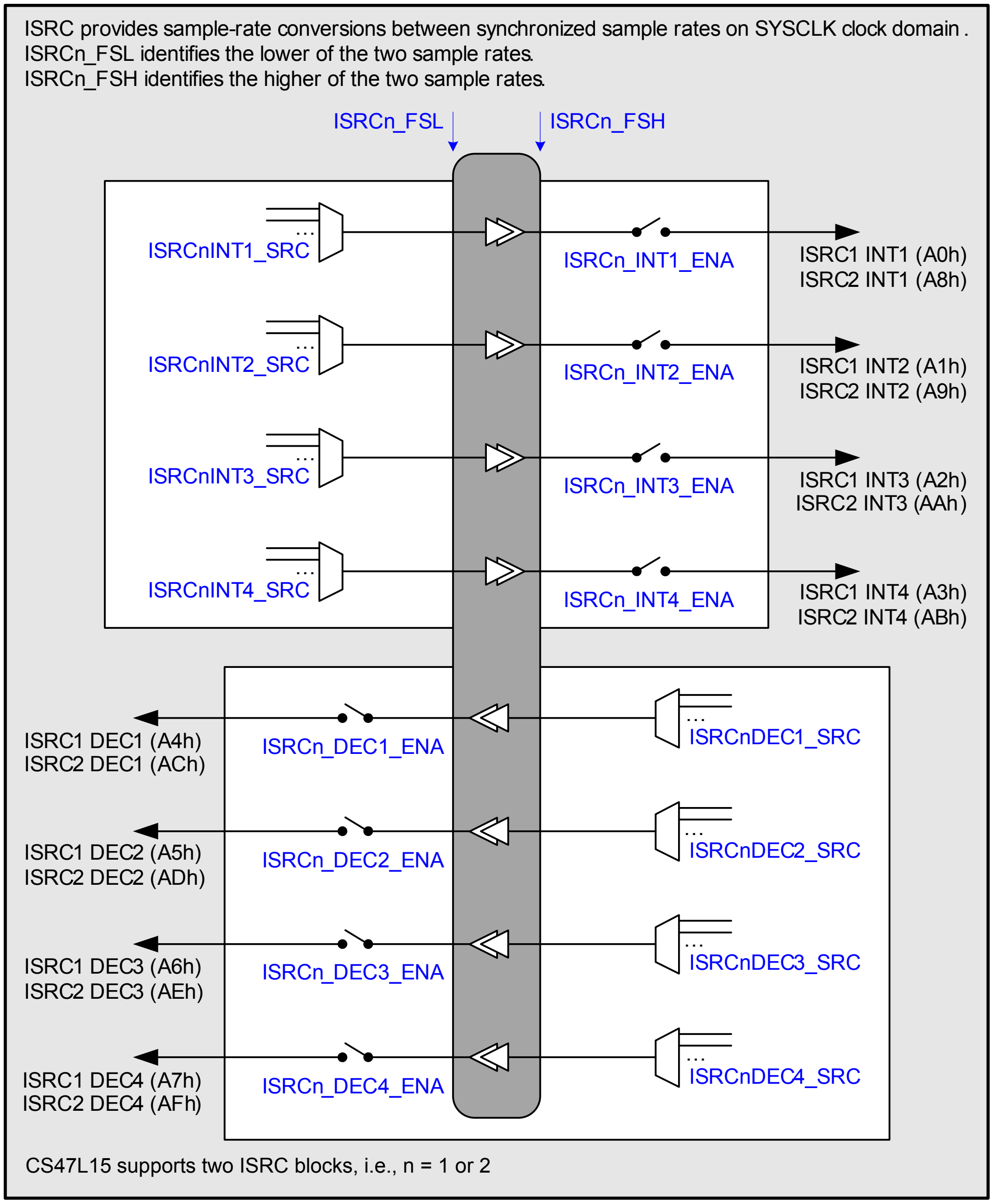


Figure 4-29. Isochronous Sample-Rate Converters (ISRCs)

The ISRC input control fields (see Fig. 4-29) are located at register addresses R2816–R2936 (0x0B00–0x0B78).

The full list of digital mixer control registers (R1600–R2936) is provided in Section 6. Generic register field definitions are provided in Table 4-10.

The x_SRC fields select the input sources for the respective ISRC processing blocks. Note that the selected input sources must be configured for the same sample rate as the ISRC to which they are connected.

The hexadecimal numbers in Fig. 4-29 indicate the corresponding x_SRC setting for selection of that signal as an input to another digital-core function.

The register bits associated with the ISRCs are described in Table 4-25.

Table 4-25. Digital-Core ISRC Control

Register Address	Bit	Label	Default	Description
R3824 (0x0EF0) ISRC1_CTRL_1	14:11	ISRC1_FSH[3:0]	0000	ISRC1 High Sample Rate (Sets the higher of the ISRC1 sample rates) 0000 = SAMPLE_RATE_1 0001 = SAMPLE_RATE_2 0010 = SAMPLE_RATE_3 All other codes are reserved. The selected sample rate is valid in the range 8 kHz to 192 kHz. All ISRC1_DEC*n*_SRC fields must be cleared before changing ISRC1_FSH.
R3825 (0x0EF1) ISRC1_CTRL_2	14:11	ISRC1_FSL[3:0]	0000	ISRC1 Low Sample Rate (Sets the lower of the ISRC1 sample rates) 0000 = SAMPLE_RATE_1 0001 = SAMPLE_RATE_2 0010 = SAMPLE_RATE_3 All other codes are reserved. The selected sample rate is valid in the range 8 kHz to 192 kHz. All ISRC1_INT*n*_SRC fields must be cleared before changing ISRC1_FSL.
R3826 (0x0EF2) ISRC1_CTRL_3	15	ISRC1_INT1_ENA	0	ISRC1 INT1 Enable (Interpolation Channel 1 path from ISRC1_FSL rate to ISRC1_FSH rate) 0 = Disabled 1 = Enabled
	14	ISRC1_INT2_ENA	0	ISRC1 INT2 Enable (Interpolation Channel 2 path from ISRC1_FSL rate to ISRC1_FSH rate) 0 = Disabled 1 = Enabled
	13	ISRC1_INT3_ENA	0	ISRC1 INT3 Enable (Interpolation Channel 3 path from ISRC1_FSL rate to ISRC1_FSH rate) 0 = Disabled 1 = Enabled
	12	ISRC1_INT4_ENA	0	ISRC1 INT4 Enable (Interpolation Channel 4 path from ISRC1_FSL rate to ISRC1_FSH rate) 0 = Disabled 1 = Enabled
	9	ISRC1_DEC1_ ENA	0	ISRC1 DEC1 Enable (Decimation Channel 1 path from ISRC1_FSH rate to ISRC1_FSL rate) 0 = Disabled 1 = Enabled
	8	ISRC1_DEC2_ ENA	0	ISRC1 DEC2 Enable (Decimation Channel 2 path from ISRC1_FSH rate to ISRC1_FSL rate) 0 = Disabled 1 = Enabled
	7	ISRC1_DEC3_ ENA	0	ISRC1 DEC3 Enable (Decimation Channel 3 path from ISRC1_FSH rate to ISRC1_FSL rate) 0 = Disabled 1 = Enabled
	6	ISRC1_DEC4_ ENA	0	ISRC1 DEC4 Enable (Decimation Channel 4 path from ISRC1_FSH rate to ISRC1_FSL rate) 0 = Disabled 1 = Enabled
R3827 (0x0EF3) ISRC2_CTRL_1	14:11	ISRC2_FSH[3:0]	0000	ISRC2 High Sample Rate (Sets the higher of the ISRC2 sample rates) 0000 = SAMPLE_RATE_1 0001 = SAMPLE_RATE_2 0010 = SAMPLE_RATE_3 All other codes are reserved. The selected sample rate is valid in the range 8 kHz to 192 kHz. All ISRC2_DEC*n*_SRC fields must be cleared before changing ISRC2_FSH.

Table 4-25. Digital-Core ISRC Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R3828 (0x0EF4) ISRC2_CTRL_2	14:11	ISRC2_FSL[3:0]	0000	ISRC2 Low Sample Rate (Sets the lower of the ISRC2 sample rates) 0000 = SAMPLE_RATE_1 0001 = SAMPLE_RATE_2 0010 = SAMPLE_RATE_3 All other codes are reserved. The selected sample rate is valid in the range 8 kHz to 192 kHz. All ISRC2_INT*n*_SRC fields must be cleared before changing ISRC2_FSL.
R3829 (0x0EF5) ISRC2_CTRL_3	15	ISRC2_INT1_ENA	0	ISRC2 INT1 Enable (Interpolation Channel 1 path from ISRC2_FSL rate to ISRC2_FSH rate) 0 = Disabled 1 = Enabled
	14	ISRC2_INT2_ENA	0	ISRC2 INT2 Enable (Interpolation Channel 2 path from ISRC2_FSL rate to ISRC2_FSH rate) 0 = Disabled 1 = Enabled
	13	ISRC2_INT3_ENA	0	ISRC2 INT3 Enable (Interpolation Channel 3 path from ISRC2_FSL rate to ISRC2_FSH rate) 0 = Disabled 1 = Enabled
	12	ISRC2_INT4_ENA	0	ISRC2 INT4 Enable (Interpolation Channel 4 path from ISRC2_FSL rate to ISRC2_FSH rate) 0 = Disabled 1 = Enabled
	9	ISRC2_DEC1_ ENA	0	ISRC2 DEC1 Enable (Decimation Channel 1 path from ISRC2_FSH rate to ISRC2_FSL rate) 0 = Disabled 1 = Enabled
	8	ISRC2_DEC2_ ENA	0	ISRC2 DEC2 Enable (Decimation Channel 2 path from ISRC2_FSH rate to ISRC2_FSL rate) 0 = Disabled 1 = Enabled
	7	ISRC2_DEC3_ ENA	0	ISRC2 DEC3 Enable (Decimation Channel 3 path from ISRC2_FSH rate to ISRC2_FSL rate) 0 = Disabled 1 = Enabled
	6	ISRC2_DEC4_ ENA	0	ISRC2 DEC4 Enable (Decimation Channel 4 path from ISRC2_FSH rate to ISRC2_FSL rate) 0 = Disabled 1 = Enabled

4.4 DSP Firmware Control

The CS47L15 digital core incorporates one programmable digital signal processing (DSP) block, capable of running a wide range of audio-enhancement functions. Different firmware configurations can be loaded onto the DSP, enabling the CS47L15 to be customized for specific application requirements. Full read/write access to the device register map is supported from the DSP core.

Examples of the DSP functions include multiband compressor (MBC), and the SoundClear™ suite of audio processing algorithms. The DSP can be clocked at up to 150MHz, corresponding to 150 MIPS.

DSP firmware can be configured using software packages provided by Cirrus Logic. A software programming guide can also be provided to assist users in developing their own software algorithms—please contact your Cirrus Logic representative for further information.

To use the programmable DSP, the required firmware configuration must first be loaded onto the device by writing the appropriate files to the CS47L15 register map. The firmware configuration comprises program, data, and coefficient content. In some cases, the coefficient content must be derived using tools provided in the WISCE evaluation board control software.

Details of the DSP firmware memory registers are provided in Section 4.4.1. Note that the WISCE evaluation board control software provides support for easy loading of program, data, and coefficient content onto the CS47L15. Please contact your Cirrus Logic representative for more details of the WISCE evaluation board control software.

After loading the DSP firmware, the DSP functions must be enabled using the associated control fields.

The audio signal paths to and from the DSP processing block are configured as described in Section 4.3. Note that the DSP firmware must be loaded and enabled before audio signal paths can be enabled.

4.4.1 DSP Firmware Memory and Register Mapping

The DSP firmware memory is programmed by writing to the registers referenced in Table 4-26. Note that clocking is not required for access to the firmware registers by the host processor.

The CS47L15 program, data, and coefficient register memory space is described in Table 4-26. The full register map listing is provided in Section 6.

The program firmware parameters are formatted as 40-bit words. For this reason, 3 x 32-bit register addresses are required for every 2 x 40-bit words.

Table 4-26. DSP Program, Data, and Coefficient Registers

DSP Number	Description	Register Address	Number of Registers	DSP Memory Size
DSP1	Program memory	0x08_0000–0x08_8FFE	18432	12k x 40-bit words
	X-Data memory	0x0A_0000–0x0A_9FFE	20480	20k x 24-bit words
	Y-Data memory	0x0C_0000–0x0C_1FFE	4096	4k x 24-bit words
	Coefficient memory	0x0E_0000–0x0E_1FFE	4096	4k x 24-bit words

The X-memory on the DSP supports read/write access to all register fields throughout the device, including the codec control registers, and the other firmware-memory regions of DSP core itself. Access to the register address space is supported using a number of register windows within the X-memory on the DSP.

Note that the register window space is additional to the X-data memory size described in Table 4-26.

Addresses 0xC000 to 0xDFFF in X-memory map directly to addresses 0x0000 to 0x1FFF in the device register space. This fixed register window contains primarily the codec control registers; it also includes the virtual DSP control registers (described in Section 4.4.7). Each X-memory address within this window maps onto one 16-bit register in the codec memory space.

Four movable register windows are also provided, starting at X-memory addresses 0xF000, 0xF400, 0xF800, and 0xFC00 respectively. Each window represents 1024 addresses in the X-memory space. The start address, within the corresponding device register space, for each window is configured using DSP1_EXT_[A/B/C/D]_PAGE (where A defines the first window, B defines the second window, etc.).

Two mapping modes are supported and are selected using the DSP1_EXT_[A/B/C/D]_PSIZE16 bits for the respective window. In 16-Bit Mode, each address within the window maps onto one 16-bit register in the device memory space; the window equates to 1024 x 16-bit registers. In 32-Bit Mode, each address within the window maps onto two 16-bit registers in the device memory space; the window equates to 1024 x 32-bit registers.

Note that the X-memory is only 24-bits wide; as a result, the upper 8 bits of the odd-numbered register addresses are not mapped, and cannot be accessed, in 32-Bit Mode.

The DSP1_EXT_[A/B/C/D]_PAGE fields are defined with an LSB = 512. Accordingly, the base address of each window must be aligned with 512-word boundaries. Note that the base addresses are entirely independent of each other; for example, overlapping windows are permissible if required, and there is no requirement for the A/B/C/D windows to be at incremental locations.

The register map window functions are shown in Fig. 4-30. Further information on the definition and usage of the DSP firmware memories is provided in the software programming guide; contact your Cirrus Logic representative if required.

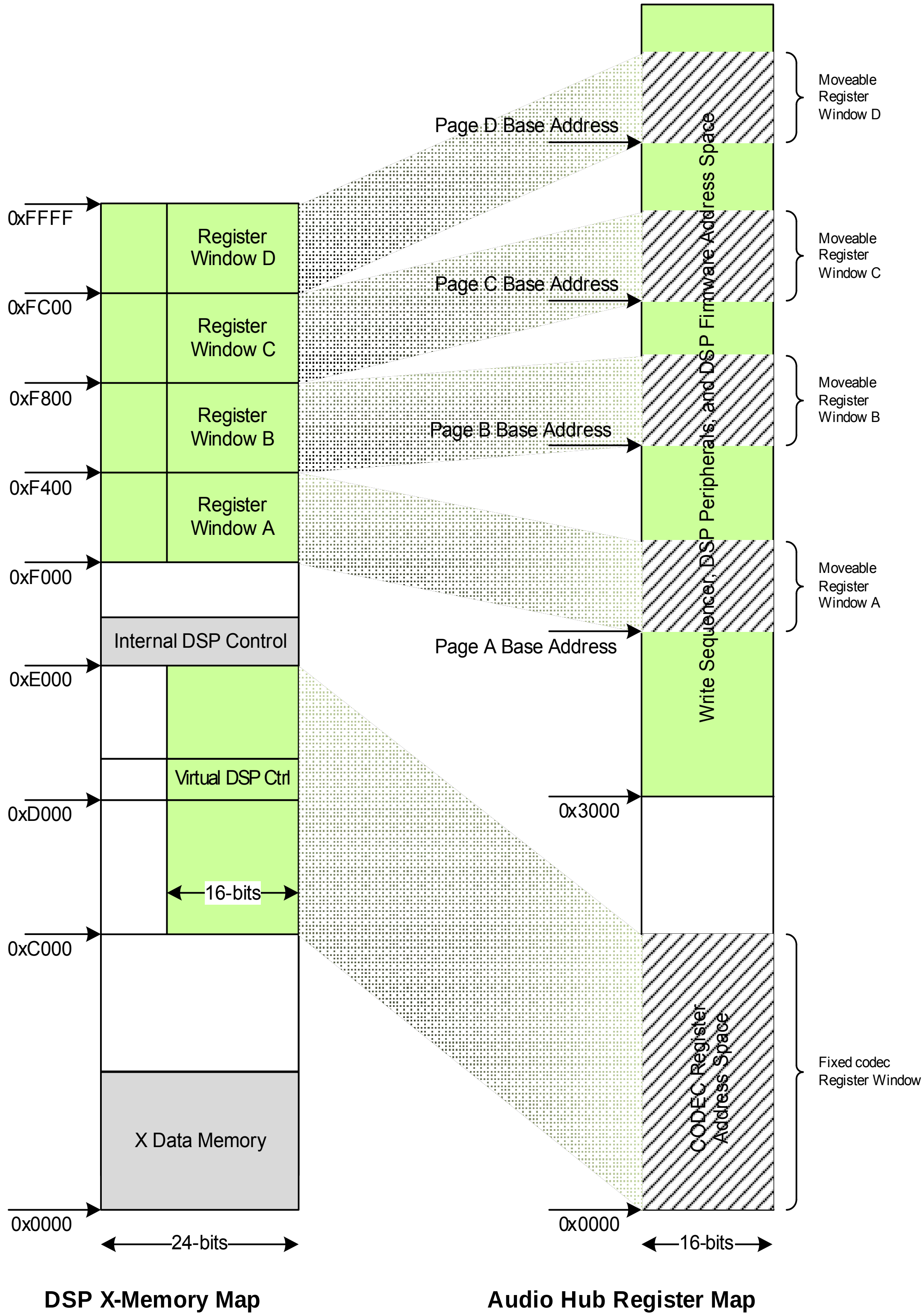


Figure 4-30. X-Data Memory Map

Note that the full CS47L15 register space is shown here as 16-bit width. (SPI/I2C register access uses 32-bit data width at 0x3000 and above.) However, the window base address fields (DSP1_EXT_[A/B/C/D]_PAGE) are referenced to 16-bit width, and 16-bit register mapping is shown. Hence, the device register map is shown here entirely as 16-bit width for ease of explanation.

The control registers associated with the register map window functions are described in Table 4-27.

Table 4-27. X-Data Memory and Clocking Control

Register Address	Bit	Label	Default	Description
R1048148 (0xF_FE54) DSP1_Ext_window_A	31	DSP1_EXT_A_PSIZE16	0	Register Window A page width select 0 = 32-bit 1 = 16-bit Note that, in 32-Bit Mode, only the lower 24 bits can be accessed.
	15:0	DSP1_EXT_A_PAGE[15:0]	0x0000	Sets the Base Address of Register Window A in X-memory. Coded as LSB = 512 (0x200)
R1048150 (0xF_FE56) DSP1_Ext_window_B	31	DSP1_EXT_B_PSIZE16	0	Register Window B page width select 0 = 32-bit 1 = 16-bit Note that, in 32-Bit Mode, only the lower 24 bits can be accessed.
	15:0	DSP1_EXT_B_PAGE[15:0]	0x0000	Sets the Base Address of Register Window B in X-memory. Coded as LSB = 512 (0x200)
R1048152 (0xF_FE58) DSP1_Ext_window_C	31	DSP1_EXT_C_PSIZE16	0	Register Window C page width select 0 = 32-bit 1 = 16-bit Note that, in 32-Bit Mode, only the lower 24 bits can be accessed.
	15:0	DSP1_EXT_C_PAGE[15:0]	0x0000	Sets the Base Address of Register Window C in X-memory. Coded as LSB = 512 (0x200)
R1048154 (0xF_FE5A) DSP1_Ext_window_D	31	DSP1_EXT_D_PSIZE16	0	Register Window D page width select 0 = 32-bit 1 = 16-bit Note that, in 32-Bit Mode, only the lower 24 bits can be accessed.
	15:0	DSP1_EXT_D_PAGE[15:0]	0x0000	Sets the Base Address of Register Window D in X-memory. Coded as LSB = 512 (0x200)

4.4.2 DSP Memory Locking

The DSP core has the capability for read/write access to all register fields throughout the device, including the codec control registers, DSP peripheral control registers, and the virtual DSP control registers. Access to these registers is supported via the DSP X-memory (using the register windows), as described in Section 4.4.1.

The CS47L15 provides a register-locking feature that blocks DSP register-write attempts to invalid register regions, preventing the firmware from making unintentional changes to register and memory contents. An interrupt event and associated debug information are generated if any write-access attempt is blocked; this can be used to assist software development and debug.

The register map and DSP firmware memories are partitioned into four regions; each region can be locked independently. This allows full flexibility to lock different register/memory regions according to the applicable DSP firmware configuration.

The DSP has direct access to its own X-, Y-, Z-, and P- memories; this is always enabled and cannot be locked. Access to the codec registers, DSP peripheral registers, and the virtual DSP registers is effected using the X-memory register windows (fixed codec window, and four configurable windows)—write access to these locations is governed by the register-locking configuration settings.

The virtual DSP registers occupy addresses within the codec register space; these registers represent one of the lockable regions within the register map—two independent locks are provided for the codec and virtual DSP registers.

Note: A DSP register window can be mapped onto the X-, Y-, Z-, or P- memory region of the DSP. In this event, write access via that window is governed by the register locks, potentially blocking the DSP from accessing its own memory. This is not the intended use of the register lock, however.

The lockable register/memory regions are defined in Table 4-28.

Table 4-28. DSP Memory Locking Regions

Region	Description	Register Address	Notes
Region 0	Virtual DSP registers	0x00_1000–0x00_2FFF	Excludes memory lock and watchdog reset registers
Region 1	Codec registers	0x00_0000–0x03_FFFE	Excludes virtual DSP registers
Region 2	DSP peripheral control registers	0x04_0000–0x07_FFFE	—
Region 3	DSP1 memory	0x08_0000–0x09_FFFE	—

The register locks are controlled using the DSP1_CTRL_REGION*m*_LOCK fields (where *m* identifies the register/memory region). The associated lock determines whether the DSP core is granted write access to region *m*. To change the lock status, two writes must be made to the respective register field:

- Writing 0x5555, followed by 0xAAAA, sets the respective lock
- Writing 0xCCCC, followed by 0x3333, clears the respective lock

The status of each lock can be read from the DSP1_CTRL_REGION*m*_LOCK_STS bits.

Write access to the DSP1_CTRL_REGION*m*_LOCK fields is always possible. This means that the DSP core always has write access for configuring the memory-access locks.

The DSP memory locking function is an input to the interrupt control circuit and can be used to trigger an interrupt event if an invalid register write is attempted—see Section 4.4.5. Additional status and control fields are provided for debug purposes, as described in Section 4.4.6.

The control registers associated with the DSP memory locking functions are described in Table 4-29.

Table 4-29. DSP Memory Locking Control

Register Address	Bit	Label	Default	Description
R1048164 (0xF_FE64) DSP1_Region_lock_sts_0	3	DSP1_CTRL_REGION3_LOCK_STS	0	DSP1 memory region *m* lock status 0 = Unlocked 1 = Locked (write access is blocked)
	2	DSP1_CTRL_REGION2_LOCK_STS	0	
	1	DSP1_CTRL_REGION1_LOCK_STS	0	
	0	DSP1_CTRL_REGION0_LOCK_STS	0	
R1048166 (0xF_FE66) DSP1_Region_lock_1___ DSP1_Region_lock_0	31:16	DSP1_CTRL_REGION1_LOCK[15:0]	See Footnote 1	DSP1 memory region *m* lock. Write 0x5555, then 0xAAAA, to set the lock. Write 0xCCCC, then 0x3333, to clear the lock.
	15:0	DSP1_CTRL_REGION0_LOCK[15:0]	See Footnote 1	
R1048168 (0xF_FE68) DSP1_Region_lock_3___ DSP1_Region_lock_2	31:16	DSP1_CTRL_REGION3_LOCK[15:0]	See Footnote 1	
	15:0	DSP1_CTRL_REGION2_LOCK[15:0]	See Footnote 1	

1. Default is not applicable to these write-only fields

4.4.3 DSP Firmware Control

The configuration and control of the DSP firmware is described in the following subsections.

4.4.3.1 DSP Memory

The DSP memory (program, X-data, Y-data, and coefficient) is enabled by setting DSP1_MEM_ENA. This memory must be enabled (DSP1_MEM_ENA = 1) for read/write access, code execution, and DMA functions. The DSP memory is disabled, and the contents lost, whenever the DSP1_MEM_ENA bit is cleared.

The DSP1_MEM_ENA bit is not affected by software reset; it remains in its previous state under software reset conditions. Accordingly, the DSP memory contents are maintained through software reset, provided DCVDD is held above its reset threshold.

The DSP firmware memory is always cleared under power-on reset, hardware reset, and Sleep Mode conditions. See Section 5.2 for a summary of the CS47L15 reset behavior.

4.4.3.2 DSP Clocking

Clocking is required for the DSP processing block, when executing software or when supporting DMA functions. (Note that clocking is not required for access to the firmware registers by the host processor.)

Clocking within the DSP is enabled and disabled automatically, as required by the DSP core and DMA channel status.

In normal operating conditions, the clock source for the DSP is derived from DSPCLK. See Section 4.13 for details of how to configure DSPCLK. See Section 4.4.3.4 for supported clocking configurations when DSPCLK is not enabled.

The clock frequency for the DSP is selected using DSP1_CLK_FREQ_SEL. The DSP clock frequency must be less than or equal to the DSPCLK frequency.

The DSP1_CLK_FREQ_STS field indicates the clock frequency for the DSP core. This can be used to confirm the clock frequency, in cases where code execution has a minimum clock frequency requirement. The DSP1_CLK_FREQ_STS field is only valid when the core is running code; typical usage of this field would be for the DSP core itself to read the clock status and to take action as applicable, in particular, if the available clock does not meet the application requirements.

Note that, depending on the DSPCLK frequency and the available clock dividers, the DSP1 clock frequency may differ from the selected clock. In most cases, the DSP1 clock frequency equals or exceeds the requested frequency. A lower frequency is implemented if limited by either the DSPCLK frequency or the maximum DSP1 clocking frequency.

The DSPCLK configuration provides input to the interrupt control circuit and can be used to trigger an interrupt event when the DSP1 clock frequency is less than the requested frequency; see Section 4.12.

4.4.3.3 DSP Code Execution

After the DSP firmware has been loaded, and the clocks configured, the DSP block is enabled by setting DSP1_CORE_ENA. When the DSP is configured and enabled, the firmware execution can be started by writing 1 to DSP1_START.

Alternative methods to trigger the firmware execution can also be configured using the DSP1_START_IN_SEL field.

Using the DSP1_START_IN_SEL field, the DSP firmware execution can be linked to the respective DMA function, the IRQ2 status, or to the FIFO status in one of the event loggers:

- DMA function: firmware execution commences when all enabled DSP input (WDMA) channel buffers have been filled, and all enabled DSP output (RDMA) channel buffers have been emptied
- IRQ2: firmware execution commences when one or more of the unmasked IRQ2 events has occurred
- Event logger status: firmware execution commences when the FIFO not-empty status is asserted within the respective event logger

To enable firmware execution on the DSP block, the DSP1_CORE_ENA bit must be set. Note that the usage of the DSP1_START bit may vary depending on the particular firmware that is being executed: in some applications (e.g., when an alternative trigger is selected using DSP1_START_IN_SEL), writing to the DSP1_START bit is not required.

4.4.3.4 DSP Operation without DSPCLK

In normal operating conditions, the clock source for the DSP block is derived from DSPCLK. The CS47L15 also supports DSP operation when DSPCLK is not enabled; this provides capability for always-on DSP applications.

The alternative clock source, for DSP clocking without DSPCLK, is the always-on FLL (FLL_AO). The FLL_AO output frequency range is approximately 45–50 MHz and is suitable for low-speed DSP clocking requirements.

The default FLL_AO settings are configured to provide a 49.152-MHz output, suitable for use as the always-on DSP clock source. Note that the FLL_AO control registers must always hold valid settings—either enabled and locked to an input reference clock, or else configured in FLL Hold Mode. See Section 4.13.9 for details of FLL_AO.

The always-on DSP clocking options are configured using the DSP1_FLL_AO_CLKENA and EVENTLOG*n*_FLL_AO_CLKENA bits:

- Setting DSP1_FLL_AO_CLKENA causes the DSP to be clocked directly from FLL_AO if DSP_CLK_ENA = 0. This allows the DSP core to execute firmware code while DSPCLK is absent.

- Setting EVENTLOG*n*_FLL_AO_CLKENA enables the DSP core to be clocked directly from FLL_AO if DSP_CLK_ENA = 0 and the FIFO not-empty status is asserted for the respective event logger. This allows the DSP core to execute firmware code while DSPCLK is absent, triggered by an event detected on one of the event loggers. Note that the DSP core is only clocked in this case if the start trigger for the DSP is derived from the status of the respective event logger (i.e., DSP1_START_IN_SEL selects the event logger as the start signal). See Section 4.5.1 for details of the event loggers; the EVENTLOG*n*_FLL_AO_CLKENA bits are defined in Table 4-34.

Note that these control bits do not automatically start DSP firmware execution—the DSP block must also be enabled using DSP1_CORE_ENA, and the start signal must be configured, as applicable.

The intended use case of the EVENTLOG*n*_FLL_AO_CLKENA bit is where the DSP core is configured to use an event logger status bit as its start condition. Note that, to support continued operation of the DSP core after the event log status is cleared (i.e., the FIFO buffer has been emptied), clocking of the DSP core must be enabled using DSP1_FLL_AO_CLKENA, or else by enabling DSPCLK as per the normal system clocking operation. One or other of these actions could be effected via the DSP firmware code.

The clock frequency for the DSP in these always-on clocking modes is selected using the DSP1_CLK_FREQ_SEL field (same as normal DSP clocking). Note that, depending on the FLL_AO output frequency and the available clock dividers, the DSP clock frequency may differ from the selected frequency. In most cases, the DSP clock frequency equals or exceeds the requested frequency. A lower frequency is implemented if limited by the FLL_AO frequency.

The DSP_CLK_SRC field is ignored in the always-on clocking modes. The DSP core reverts to the normal (DSPCLK) clocking configuration if DSP_CLK_ENA = 1.

4.4.3.5 DSP Watchdog Timer

A watchdog timer is provided for the DSP, which can be used to detect software lock-ups, and other conditions that require corrective action in order to resume the intended DSP behavior.

The DSP1 watchdog is enabled using DSP1_WDT_ENA. The timeout period is configured using DSP1_WDT_MAX_COUNT.

In normal operation, the watchdog should be reset regularly—this action is used to confirm that the DSP code is running correctly. The watchdog is reset by writing 0x5555, followed by 0xAAAA, to the DSP1_WDT_RESET field.

The watchdog status bit, DSP1_WDT_TIMEOUT_STS, is set if the timeout period elapses before the watchdog is reset; this event typically signals that a lock-up or other error condition has occurred.

The DSP watchdog is an input to the interrupt control circuit and can be used to trigger an interrupt event if the timeout period elapses—see Section 4.4.5.

Note that write access to the DSP1_WDT_RESET field is not affected by the register locking mechanism (see Section 4.4.2). This means that the DSP core always has write access to reset the watchdog.

4.4.3.6 DSP Control Registers

The DSP memory, clocking, code-execution, and watchdog control registers are described in Table 4-30.

The audio signal paths connecting to/from the DSP processing block are configured as described in Section 4.3. Note that the DSP firmware must be loaded and enabled before audio signal paths can be enabled.

Table 4-30. DSP Memory and Clocking Control

Register Address	Bit	Label	Default	Description
R1048064 (0xF_FE00) DSP1_Config_1	24	DSP1_FLL_AO_CLKENA	0	DSP1 always-on clock control Selects the DSP1 clocking if DSPCLK is disabled 0 = No clock 1 = DSP1 is clocked directly from FLL_AO
	4	DSP1_MEM_ENA	0	DSP1 memory control 0 = Disabled 1 = Enabled The DSP1 memory contents are lost when DSP1_MEM_ENA =0. Note that this bit is not affected by software reset; it remains in its previous condition.
	1	DSP1_CORE_ENA	0	DSP1 enable. Controls the DSP1 firmware execution 0 = Disabled 1 = Enabled
	0	DSP1_START	—	DSP1 start Write 1 to start DSP1 firmware execution
R1048066 (0xF_FE02) DSP1_Config_2	15:0	DSP1_CLK_FREQ_SEL[15:0]	0x0000	DSP1 clock frequency select Coded as LSB = 1/64 MHz, Valid from 5.6 to 148 MHz. The DSP1 clock must be less than or equal to the DSPCLK frequency. The DSP1 clock is generated by division of DSPCLK, and may differ from the selected frequency. The DSP1 clock frequency can be read from DSP1_CLK_FREQ_STS.
R1048070 (0xF_FE06) DSP1_Status_2	0	DSP1_CLK_AVAIL	0	DSP1 clock availability (read only) 0 = No Clock 1 = Clock Available This bit exists for legacy software support only; it is not recommended for future designs—it may be unreliable on the latest device architectures.
R1048072 (0xF_FE08) DSP1_Status_3	15:0	DSP1_CLK_FREQ_STS[15:0]	0x0000	DSP1 clock frequency (read only). Valid only when the respective DSP core is enabled. Coded as LSB = 1/64 MHz.
R1048074 (0xF_FE0A) DSP1_Watchdog_1	4:1	DSP1_WDT_MAX_COUNT[3:0]	0x0	DSP1 watchdog timeout value. 0x0 = 2 ms; 0x5 = 64 ms; 0xA = 2 s 0x1 = 4 ms; 0x6 = 128 ms; 0xB = 4 s 0x2 = 8 ms; 0x7 = 256 ms; 0xC = 8 s 0x3 = 16 ms; 0x8 = 512 ms; 0xD–0xF = reserved 0x4 = 32 ms; 0x9 = 1 s
	0	DSP1_WDT_ENA	0	DSP1 watchdog enable 0 = Disabled 1 = Enabled
R1048120 (0xF_FE38) DSP1_External_Start	4:0	DSP1_START_IN_SEL[4:0]	0x00	DSP1 firmware execution control. Selects the trigger for DSP1 firmware execution. 0x00 = DMA; 0x10 = Event Logger 1 0x0B = IRQ2; 0x11 = Event Logger 2 All other codes are reserved. Note that the DSP1_START bit also starts the DSP1 firmware execution, regardless of this field setting.
R1048158 (0xF_FE5E) DSP1_Watchdog_2	15:0	DSP1_WDT_RESET[15:0]	0x0000	DSP1 watchdog reset. Write 0x5555, followed by 0xAAAA, to reset the watchdog.
R1048186 (0xF_FE7A) DSP1_Region_lock_ctrl_0	13	DSP1_WDT_TIMEOUT_STS	0	DSP1 watchdog timeout status This bit, when set, indicates that the watchdog timeout has occurred. This bit is latched when set; it is cleared when the watchdog is disabled or reset.

4.4.4 DSP Direct Memory Access (DMA) Control

The DSP provides a multichannel DMA function; this is configured using the registers described in Table 4-31.

There are eight WDMA (DSP input) and six RDMA (DSP output) channels; these are enabled using the DSP1_WDMA_CHANNEL_ENABLE and DSP1_RDMA_CHANNEL_ENABLE fields. The status of each WDMA channel is indicated in DSP1_WDMA_ACTIVE_CHANNELS.

The DMA can access the X-data memory or Y-data memory associated with the DSP block. The applicable memory is selected using bit [15] of the respective x_START_ADDRESS field for each DMA channel.

The start address of each DMA channel is configured as described in Table 4-31. Note that the required address is defined relative to the base address of the selected (X-data or Y-data) memory.

The buffer length of the DMA channels is configured using the DSP1_DMA_BUFFER_LENGTH field. The selected buffer length applies to all enabled DMA channels.

Note that the start-address fields and buffer-length fields are defined in 24-bit DSP data word units. This means that the LSB of these fields represents one 24-bit DSP memory word. This differs from the CS47L15 register map layout described in Table 4-26.

The parameters of a DMA channel (i.e., start address or offset address) must not be changed while the respective DMA is enabled. All of the DMA channels must be disabled before changing the DMA buffer length.

Each DMA channel uses a twin buffer mechanism to support uninterrupted data flow through the DSP. The buffers are called *ping* and *pong*, and are of configurable size, as noted above. Data is transferred to/from each of the buffers in turn.

When the ping input data buffer is full, the DSP1_PING_FULL bit is set, and a DSP start signal is generated. The start signal from the DMA is typically used to start firmware execution, as noted in Table 4-30. Meanwhile, further DSP input data fills up the pong buffer.

When the pong input buffer is full, the DSP1_PONG_FULL bit is set, and another DSP start signal is generated. The DSP firmware must take care to read the input data from the applicable buffer, in accordance with the DSP1_PING_FULL and DSP1_PONG_FULL status bits.

Twin buffers are also used on the DSP output (RDMA) channels. The output ping buffers are emptied at the same time as the input ping buffers are filled; the output pong buffers are emptied at the same time that the input pong buffers are filled.

The DSP core supports 24-bit signal processing. Under default conditions, the DSP audio data is in 2’s complement Q3.20 format (i.e., 0xF00000 corresponds to the –1.0 level, and 0x100000 corresponds to the +1.0 level; a sine wave with peak values of ±1.0 corresponds to the 0 dBFS level). If DSP1_DMA_WORD_SEL is set, audio data is transferred to and from the DSP in Q0.23 format. The applicable format should be set according to the requirements of the specific DSP firmware.

Note that the DSP core is optimized for Q3.20 audio data processing; Q0.23 data can be supported, but the firmware implementation may incur a reduction in power efficiency due to the higher MIPS required for arithmetic operations in non-native data word format.

The DMA function is an input to the interrupt control circuit—see Section 4.4.5. The respective interrupt event is triggered if all enabled input (WDMA) channel buffers have been filled and all enabled output (RDMA) channel buffers have been emptied.

Further details of the DMA are provided in the software programming guide; contact your Cirrus Logic representative if required.

Table 4-31. DMA Control

Register Address	Bit	Label	Default	Description
R1048068 (0xF_FE04) DSP1_Status_1	31	DSP1_PING_FULL	0	DSP1 WDMA Ping Buffer Status 0 = Not Full 1 = Full
	30	DSP1_PONG_FULL	0	DSP1 WDMA Pong Buffer Status 0 = Not Full 1 = Full
	23:16	DSP1_WDMA_ACTIVE_ CHANNELS[7:0]	0x00	DSP1 WDMA Channel Status There are eight WDMA channels; each bit of this field indicates the status of the respective WDMA channel. Each bit is coded as follows: 0 = Inactive 1 = Active
R1048080 (0xF_FE10) DSP1_WDMA_Buffer_1	31:16	DSP1_START_ADDRESS_ WDMA_BUFFER_1[15:0]	0x0000	DSP1 WDMA Channel 1 Start Address Bit [15] = Memory select 0 = X-data memory 1 = Y-data memory Bits [14:0] = Address select The address is defined relative to the base address of the applicable data memory. The LSB represents one 24-bit DSP memory word. Note that the start address is also controlled by the respective DSP1_WDMA_CHANNEL_OFFSET bit.
	15:0	DSP1_START_ADDRESS_ WDMA_BUFFER_0[15:0]	0x0000	DSP1 WDMA Channel 0 Start Address Field description is as above.
R1048082 (0xF_FE12) DSP1_WDMA_Buffer_2	31:16	DSP1_START_ADDRESS_ WDMA_BUFFER_3[15:0]	0x0000	DSP1 WDMA Channel 3 Start Address Field description is as above.
	15:0	DSP1_START_ADDRESS_ WDMA_BUFFER_2[15:0]	0x0000	DSP1 WDMA Channel 2 Start Address Field description is as above.
R1048084 (0xF_FE14) DSP1_WDMA_Buffer_3	31:16	DSP1_START_ADDRESS_ WDMA_BUFFER_5[15:0]	0x0000	DSP1 WDMA Channel 5 Start Address Field description is as above.
	15:0	DSP1_START_ADDRESS_ WDMA_BUFFER_4[15:0]	0x0000	DSP1 WDMA Channel 4 Start Address Field description is as above.
R1048086 (0xF_FE16) DSP1_WDMA_Buffer_4	31:16	DSP1_START_ADDRESS_ WDMA_BUFFER_7[15:0]	0x0000	DSP1 WDMA Channel 7 Start Address Field description is as above.
	15:0	DSP1_START_ADDRESS_ WDMA_BUFFER_6[15:0]	0x0000	DSP1 WDMA Channel 6 Start Address Field description is as above.
R1048096 (0xF_FE20) DSP1_RDMA_Buffer_1	31:16	DSP1_START_ADDRESS_ RDMA_BUFFER_1[15:0]	0x0000	DSP1 RDMA Channel 1 Start Address Bit [15] = Memory select 0 = X-data memory 1 = Y-data memory Bits [14:0] = Address select The address is defined relative to the base address of the applicable data memory. The LSB represents one 24-bit DSP memory word. Note that the start address is also controlled by the respective DSP1_RDMA_CHANNEL_OFFSET bit.
	15:0	DSP1_START_ADDRESS_ RDMA_BUFFER_0[15:0]	0x0000	DSP1 RDMA Channel 0 Start Address Field description is as above.
R1048098 (0xF_FE22) DSP1_RDMA_Buffer_2	31:16	DSP1_START_ADDRESS_ RDMA_BUFFER_3[15:0]	0x0000	DSP1 RDMA Channel 3 Start Address Field description is as above.
	15:0	DSP1_START_ADDRESS_ RDMA_BUFFER_2[15:0]	0x0000	DSP1 RDMA Channel 2 Start Address Field description is as above.
R1048100 (0xF_FE24) DSP1_RDMA_Buffer_3	31:16	DSP1_START_ADDRESS_ RDMA_BUFFER_5[15:0]	0x0000	DSP1 RDMA Channel 5 Start Address Field description is as above.
	15:0	DSP1_START_ADDRESS_ RDMA_BUFFER_4[15:0]	0x0000	DSP1 RDMA Channel 4 Start Address Field description is as above.

Table 4-31. DMA Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R1048112 (0xF_FE30) DSP1_DMA_Config_1	23:16	DSP1_WDMA_CHANNEL_ ENABLE[7:0]	0x00	DSP1 WDMA Channel Enable There are eight WDMA channels; each bit of this field enables the respective WDMA channel. Each bit is coded as follows: 0 = Disabled 1 = Enabled
	13:0	DSP1_DMA_BUFFER_ LENGTH[13:0]	0x0000	DSP1 DMA Buffer Length Selects the amount of data transferred in each DMA channel. The LSB represents one 24-bit DSP memory word.
R1048114 (0xF_FE32) DSP1_DMA_Config_2	7:0	DSP1_WDMA_CHANNEL_ OFFSET[7:0]	0x00	DSP1 WDMA Channel Offset There are eight WDMA channels; each bit of this field offsets the start Address of the respective WDMA channel. Each bit is coded as follows: 0 = No offset 1 = Offset by 0x8000
R1048116 (0xF_FE34) DSP1_DMA_Config_3	21:16	DSP1_RDMA_CHANNEL_ OFFSET[5:0]	0x00	DSP1 RDMA Channel Offset There are six RDMA channels; each bit of this field offsets the start Address of the respective RDMA channel. Each bit is coded as follows: 0 = No offset 1 = Offset by 0x8000
	5:0	DSP1_RDMA_CHANNEL_ ENABLE[5:0]	0x00	DSP1 RDMA Channel Enable There are six RDMA channels; each bit of this field enables the respective RDMA channel. Each bit is coded as follows: 0 = Disabled 1 = Enabled
R1048118 (0xF_FE36) DSP1_DMA_Config_4	0	DSP1_DMA_WORD_SEL	0	DSP1 Data Word Format 0 = Q3.20 format (4 integer bits, 20 fractional bits) 1 = Q0.23 format (1 integer bit, 23 fractional bits) The data word format should be set according to the requirements of the applicable DSP firmware.

4.4.5 DSP Interrupts

The DSP core provides inputs to the interrupt circuit and can be used to trigger an interrupt event when the associated conditions occur. The following interrupts are provided for DSP core:

- DMA interrupt—Asserted when all enabled DSP input (WDMA) channel buffers have been filled, and all enabled DSP output (RDMA) channel buffers have been emptied
- DSP Start 1, DSP Start 2 interrupts—Asserted when the respective start signal is triggered
- DSP Busy interrupt—Asserted when the DSP is busy (i.e., when firmware execution or DMA processes are started)
- DSP Bus Error interrupt—Asserted when a locked register address, invalid memory address, or watchdog timeout error is detected

The CS47L15 also provides 16 control bits that allow the DSP core to generate programmable interrupt events. When a 1 is written to these bits (see Table 4-32), the respective DSP interrupt (DSP_IRQn_EINTx) is triggered. The associated interrupt bits are latched once set; they can be polled at any time or used to control the IRQ signal.

See Section 4.12 for further details.

Table 4-32. DSP Interrupts

Register Address	Bit	Label	Default	Description
R5632 (0x1600) ADSP2_IRQ0	1	DSP_IRQ2	0	DSP IRQ2. Write 1 to trigger the DSP_IRQ2_EINT*n* interrupt.
	0	DSP_IRQ1	0	DSP IRQ1. Write 1 to trigger the DSP_IRQ1_EINT*n* interrupt.
R5633 (0x1601) ADSP2_IRQ1	1	DSP_IRQ4	0	DSP IRQ4. Write 1 to trigger the DSP_IRQ4_EINT*n* interrupt.
	0	DSP_IRQ3	0	DSP IRQ3. Write 1 to trigger the DSP_IRQ3_EINT*n* interrupt.

Table 4-32. DSP Interrupts *(Cont.)*

Register Address	Bit	Label	Default	Description
R5634 (0x1602)	1	DSP_IRQ6	0	DSP IRQ6. Write 1 to trigger the DSP_IRQ6_EINT*n* interrupt.
ADSP2_IRQ2	0	DSP_IRQ5	0	DSP IRQ5. Write 1 to trigger the DSP_IRQ5_EINT*n* interrupt.
R5635 (0x1603)	1	DSP_IRQ8	0	DSP IRQ8. Write 1 to trigger the DSP_IRQ8_EINT*n* interrupt.
ADSP2_IRQ3	0	DSP_IRQ7	0	DSP IRQ7. Write 1 to trigger the DSP_IRQ7_EINT*n* interrupt.
R5636 (0x1604)	1	DSP_IRQ10	0	DSP IRQ10. Write 1 to trigger the DSP_IRQ10_EINT*n* interrupt.
ADSP2_IRQ4	0	DSP_IRQ9	0	DSP IRQ9. Write 1 to trigger the DSP_IRQ9_EINT*n* interrupt.
R5637 (0x1605)	1	DSP_IRQ12	0	DSP IRQ12. Write 1 to trigger the DSP_IRQ12_EINT*n* interrupt.
ADSP2_IRQ5	0	DSP_IRQ11	0	DSP IRQ11. Write 1 to trigger the DSP_IRQ11_EINT*n* interrupt.
R5638 (0x1606)	1	DSP_IRQ14	0	DSP IRQ14. Write 1 to trigger the DSP_IRQ14_EINT*n* interrupt.
ADSP2_IRQ6	0	DSP_IRQ13	0	DSP IRQ13. Write 1 to trigger the DSP_IRQ13_EINT*n* interrupt.
R5639 (0x1607)	1	DSP_IRQ16	0	DSP IRQ16. Write 1 to trigger the DSP_IRQ16_EINT*n* interrupt.
ADSP2_IRQ7	0	DSP_IRQ15	0	DSP IRQ15. Write 1 to trigger the DSP_IRQ15_EINT*n* interrupt.

4.4.6 DSP Debug Support

General-purpose registers are provided for the DSP. These have no assigned function and can be used to assist in algorithm development.

The JTAG interface provides test and debug access to the CS47L15, as described in Section 4.17. The JTAG interface clock can be enabled for the DSP core using DSP1_DBG_CLK_ENA. Note that, when the JTAG interface is used to access the DSP core, the DSP1_CORE_ENA bit must also be set.

The DSP1_LOCK_ERR_STS bit indicates that the DSP attempted to write to a locked register address. The DSP1_ ADDR_ERR_STS bit indicates that the DSP attempted to access an invalid memory address (i.e., an address whose contents are undefined). Once set, these bits remain set until a 1 is written to DSP1_ERR_CLEAR.

The DSP1_PMEM_ERR_ADDR and DSP1_XMEM_ERR_ADDR fields contain the program memory and X-data memory addresses associated with a locked register address error condition. If DSP1_LOCK_ERR_STS is set, these fields correspond to the first-detected locked register address error. Note that no subsequent error event can be reported in these fields until the DSP1_LOCK_ERR_STS is cleared.

Note: The DSP1_PMEM_ERR_ADDR value is the prefetched address of a code instruction that has not yet been executed; it does not point directly to the instruction that caused the error.

The DSP1_BUS_ERR_ADDR field indicates the register/memory address that resulted in a register-access error. The field relates either to a locked register address error or to an invalid memory address error, as follows:

- If DSP1_LOCK_ERR_STS is set, the DSP1_BUS_ERR_ADDR value corresponds to the first-detected locked register address error. Note that no subsequent error event can be reported in this field until DSP1_LOCK_ERR_ STS is cleared.
- If DSP1_ADDR_ERR_STS is set, and DSP1_LOCK_ERR_STS is clear, the DSP1_BUS_ERR_ADDR field corresponds to the most recent invalid memory address error.
- If the DSP1_LOCK_ERR_STS and DSP1_ADDR_ERR_STS are both clear, the DSP1_BUS_ERR_ADDR field is undefined.

Note: The DSP1_BUS_ERR_ADDR value is coded using a byte-referenced address, so the actual register address is equal to DSP1_BUS_ERR_ADDR / 2. If the register-access error is the result of an attempt to access the virtual DSP registers, a register address of 0 is reported.

If the DSP1_ERR_PAUSE bit is set, the DSP code execution stops immediately on detection of a locked register address error. This enables debug information to be retrieved from the DSP core during code development. In this event, code execution can be restarted by clearing the DSP1_ERR_PAUSE bit. Alternatively, the DSP core can restarted by clearing and setting DSP1_CORE_ENA (described in Section 4.4.3.3).

Table 4-33. DSP Debug Support

Register Address	Bit	Label	Default	Description
R1048064 (0xF_FE00) DSP1_Config_1	3	DSP1_DBG_CLK_ENA	0	DSP1 Debug Clock Enable 0 = Disabled 1 = Enabled
R1048128 (0xF_FE40) DSP1_Scratch_1	31:16	DSP1_SCRATCH_1[15:0]	0x0000	DSP1 Scratch Register 1
	15:0	DSP1_SCRATCH_0[15:0]	0x0000	DSP1 Scratch Register 0
R1048130 (0xF_FE42) DSP1_Scratch_2	31:16	DSP1_SCRATCH_3[15:0]	0x0000	DSP1 Scratch Register 3
	15:0	DSP1_SCRATCH_2[15:0]	0x0000	DSP1 Scratch Register 2
R1048146 (0xF_FE52) DSP1_Bus_Error_Addr	23:0	DSP1_BUS_ERR_ADDR[23:0]	0x00_0000	Contains the register address of a memory region lock or memory address error event. Note the associated register address is equal to DSP1_BUS_ERR_ADDR / 2.
R1048186 (0xF_FE7A) DSP1_Region_lock_ctrl_0	15	DSP1_LOCK_ERR_STS	0	DSP1 memory region lock error status. This bit, when set, indicates that DSP1 attempted to write to a locked register address. This bit is latched when set; it is cleared when a 1 is written to DSP1_ERR_CLEAR.
	14	DSP1_ADDR_ERR_STS	0	DSP1 memory address error status. This bit, when set, indicates that DSP1 attempted to access an undefined locked register address. This bit is latched when set; it is cleared when a 1 is written to DSP1_ERR_CLEAR.
	1	DSP1_ERR_PAUSE	0	DSP1 bus address error control. Configures the DSP1 response to a memory region lock error event. 0 = No action 1 = Pause DSP1 code execution
	0	DSP1_ERR_CLEAR	0	Write 1 to clear the memory region lock error and memory address error status bits.
R1048188 (0xF_FE7C) DSP1_PMEM_Err_Addr___XMEM_ERR_Addr	30:16	DSP1_PMEM_ERR_ADDR[14:0]	0x0000	Contains the program memory address of a memory region lock error event. Note this is the prefetched address of a subsequent instruction; it does not point directly to the address that caused the error.
	15:0	DSP1_XMEM_ERR_ADDR[15:0]	0x0000	Contains the X-data memory address of a memory region lock error event.

4.4.7 Virtual DSP Registers

The DSP control registers are described throughout Section 4.4. Each control register has a unique location within the CS47L15 register map.

An additional set of DSP control registers is also defined, which can be used in firmware to access the DSP control fields: the virtual DSP (or DSP 0) registers are defined at address R4096 (0x1000) in the device register map. The full register map listing is provided in Section 6.

Note that read/write access to the virtual DSP registers is only possible via firmware running on the integrated DSP core. When DSP firmware accesses the virtual registers, the registers are automatically mapped onto the DSP1 control registers. The virtual DSP registers are designed to allow software to be transferable across different DSPs (e.g., on multicore devices) without modification to the software code.

The virtual DSP registers are defined at register addresses R4096–R4192 (0x1000–0x1060) in the device register map. Note that these registers cannot be accessed directly at the addresses shown; they can be only accessed through DSP firmware code, using the register window function shown in Fig. 4-30. The virtual DSP registers are located at address 0xD000 in the X-data memory map.

4.5 DSP Peripheral Control

The CS47L15 incorporates a suite of DSP peripheral functions that can be integrated together to provide an enhanced capability for DSP applications. Configurable event log functions provide multichannel monitoring of internal and external signals. The general-purpose timers provide time-stamp data for the event logs; they also support the watchdog and other miscellaneous time-based functions. Maskable GPIO provides an efficient mechanism for the DSP core to access the required input and output signals.

The peripherals are designed to support a comprehensive DSP capability, operating with a high degree of autonomy from the host processor.

4.5.1 Event Loggers

The CS47L15 provides two event log functions, supporting multichannel, edge-sensitive monitoring and recording of internal or external signals.

4.5.1.1 Overview

The event loggers allow status information to be captured from a large number of sources, to be prioritized and acted upon as required. For the purposes of the event loggers, an event is recorded when a logic transition (edge) is detected on a selected signal source.

The logged events are held in a FIFO buffer, which is managed by the application software. A 32-bit time stamp, derived from one of the general-purpose timers, is associated and recorded with each FIFO index, to provide a comprehensive record of the detected events.

Each event logger must be associated with one of the general-purpose timers. The selected timer is the source of time stamp data for any logged events. If DSPCLK is disabled, the timer also provides the clock source for the event logger. (If DSPCLK is enabled, DSPCLK is used as the clock source instead.)

A maximum of one event per cycle of the clock source can be logged. If more than one event occurs within the cycle time, the highest priority (lowest channel number) event is logged at the rising edge of the clock. In this case, any lower priority events is queued, and is logged as soon as no higher priority events are pending. It is possible for recurring events on a high-priority channel to be logged, while low-priority ones remain queued. Note that recurring instances of events that are queued would not be logged.

The event logger can use a slow clock (e.g., 32 kHz), but higher clock frequencies may also be commonly used, depending on the application and use case. The clock frequency determines the maximum possible event logging rate.

4.5.1.2 Event Logger Control

The event logger is enabled by setting EVENTLOG*n*_ENA (where *n* identifies the respective event logger, 1 or 2).

The event logger can be reset by writing 1 to EVENTLOG*n*_RST. Executing this function clears all the event logger status flags and clears the contents of the FIFO buffer.

The associated timer (and time-stamp source) is selected using EVENTLOG*n*_TIME_SEL. Note that the event logger must be disabled (EVENTLOG*n*_ENA = 0) when selecting the timer source.

4.5.1.3 Input Channel Configuration

The event logger allows up to 16 input channels to be configured for detection and logging. The EVENTLOG*n*_CH*x*_SEL field selects the applicable input source for each channel (where *x* identifies the channel number, 1 to 16). The polarity selection and debounce options are configured using the EVENTLOG*n*_CH*x*_POL and EVENTLOG*n*_CH*x*_DB bits respectively.

The input channels can be enabled or disabled freely, using EVENTLOG*n*_CH*x*_ENA, without having to disable the event logger entirely. An input channel must be disabled whenever the associated x_SEL, x_POL, or x_DB fields are written. It is possible to reconfigure input channels while the event logger is enabled, provided the channels being reconfigured are disabled when doing so.

The available input sources include GPIO inputs, external accessory status (jack, mic, sensors), and signals generated by the integrated DSP core. A list of the valid input sources for the event loggers is provided in Table 4-35. Note that, to log both rising and falling events from any source, two separate input channels must be configured—one for each polarity.

If an input channel is configured for rising edge detection (EVENTLOG*n*_CH*x*_POL = 0), and the corresponding input signal is asserted (Logic 1) at the time when the event logger is enabled, an event is logged in respect of this initial state. Similarly, if an input channel is configured for falling edge detection, and is deasserted (Logic 0) when the event logger is enabled, a corresponding event is logged. If rising and falling edges are both configured for detection, an event is always logged in respect of the initial condition.

4.5.1.4 FIFO Buffer

Each event (signal transition) that meets the criteria of an enabled channel is written to the 16-stage FIFO buffer. The buffer is filled cyclically, but does not overwrite unread data when full. An error condition occurs if the buffer fills up completely.

Note that the FIFO behavior is not enforced or fully implemented in the device hardware, but assumes that a compatible software implementation is in place. New events are written to the buffer in a cyclic manner, but the data can be read out in any order, if desired. The designed FIFO behavior requires the software to update the read pointer (RPTR) in the intended manner for smooth operation.

The entire contents of the 16-stage FIFO buffer can be accessed directly in the register map. Each FIFO index (y = 0 to 15) comprises the EVENTLOG*n*_FIFO*y*_ID (identifying the source signal of the associated log event), the EVENTLOG*n*_FIFO*y*_POL (the polarity of the respective event transition), and the EVENTLOG*n*_FIFO*y*_TIME field (containing the 32-bit time stamp from the associated timer).

The FIFO buffer is managed using EVENTLOG*n*_FIFO_WPTR and EVENTLOG*n*_FIFO_RPTR. The write pointer (WPTR) field identifies the index location (0 to 15) in which the next event is logged. The read pointer (RPTR) field identifies the index location of the first set of unread data, if any exists. Both of these fields are initialized to 0 when the event logger is reset.

- If RPTR ≠ WPTR, the buffer contains new data. The number of new events is equal to the difference between the two pointer values (WPTR – RPTR, allowing for wraparound beyond Index 15). For example, if WPTR = 12 and RPTR = 8, this means that there are four unread data sets in the buffer, at index locations 8, 9, 10, and 11.

  After reading the new data from the buffer, the RPTR value should be incremented by the corresponding amount (e.g., increment by 4, in the example described above). Note that the RPTR value can either be incremented once for each read, or can be incremented in larger steps after a batch read.
- If RPTR = WPTR, the buffer is either empty (0 events) or full (16 events). In this case, the status bits described in Section 4.5.1.5 confirm the current status of the buffer.

4.5.1.5 Status Bits

The EVENTLOG*n*_NOT_EMPTY bit indicates whether the FIFO buffer is empty. When this bit is set, it indicates one or more new sets of data in the FIFO.

The EVENTLOG*n*_WMARK_STS bit indicates when the number of FIFO index locations available for new events reaches a configurable threshold, known as the watermark level. The watermark level is held in the EVENTLOG*n*_FIFO_WMARK field.

The EVENTLOG*n*_FULL bit indicates when the FIFO buffer is full. When this bit is set, it indicates that there are 16 sets of new event data in the FIFO. Note that this does not mean that a buffer overflow condition has occurred, but further events are not logged or indicated until the buffer has been cleared.

Note: Following a buffer full condition, the FIFO operation resumes as soon as the RPTR field has been updated to a new value. Writing the same value to RPTR does not restart the FIFO operation, even if the entire buffer contents have been read. After all of the required data has been read from the buffer, the RPTR value should be set equal to the WPTR value; an intermediate (different) value must also be written to the RPTR field in order to clear the buffer full status and restart the FIFO operation.

4.5.1.6 Interrupts, GPIO, Write Sequencer, and DSP Firmware Control

The control-write sequencer is automatically triggered whenever the NOT_EMPTY status of the event log buffer is asserted. A different control sequence may be configured for each event logger; see Section 4.15 for further details.

The event log status flags are inputs to the interrupt control circuit and can be used to trigger an interrupt event when the respective FIFO condition (full, not empty, or watermark level) occurs; see Section 4.12.

The event log status can be output directly on a GPIO pin as an external indication of the event logger; see Section 4.11 to configure a GPIO pin for this function.

The event log NOT_EMPTY status can also be selected as a start trigger for DSP firmware execution; see Section 4.4.

4.5.1.7 Event Logger Control Registers

The event logger control registers are described in Table 4-34.

Table 4-34. Event Logger (EVENTLOG*n*) Control

Register Address	Bit	Label	Default	Description
Event Log 1 Base Address = R294912 (0x4_8000) Event Log 2 Base Address = R295424 (0x4_8200)				
base address EVENTLOG*n*_CONTROL	8	EVENTLOG*n*_FLL_AO_CLKENA	0	Event Log DSP Clock Control Configures clocking of the DSP core if DSPCLK is disabled, according to the Event Log FIFO status. 0 = FIFO status has no effect on DSP clocking 1 = DSP core clocked directly from FLL_AO if Event Log *n* FIFO is not empty
	1	EVENTLOG*n*_RST	0	Event Log Reset Write 1 to reset the status outputs and clear the FIFO buffer.
	0	EVENTLOG*n*_ENA	0	Event Log Enable 0 = Disabled 1 = Enabled
Base address +0x04 EVENTLOG*n*_TIMER_SEL	1:0	EVENTLOG*n*_TIMER_SEL[1:0]	00	Event Log Timer Source Select 00 = Timer 1 01 = Timer 2 Note that the event log must be disabled when updating this field
Base address +0x0C EVENTLOG*n*_FIFO_CONTROL1	3:0	EVENTLOG*n*_FIFO_WMARK[3:0]	0x1	Event Log FIFO Watermark. The watermark status output is asserted when the number of FIFO locations available for new events is less than or equal to the FIFO watermark. Valid from 0 to 15.
Base address +0x0E EVENTLOG*n*_FIFO_POINTER1	18	EVENTLOG*n*_FULL	0	Event Log FIFO Full Status. This bit, when set, indicates that the FIFO buffer is full. It is cleared when a new value is written to the FIFO read pointer, or when the event log is Reset.
	17	EVENTLOG*n*_WMARK_STS	0	Event Log FIFO Watermark Status. This bit, when set, indicates that the FIFO space available for new events to be logged is less than or equal to the watermark threshold.
	16	EVENTLOG*n*_NOT_EMPTY	0	Event Log FIFO Not Empty Status. This bit, when set, indicates one or more new sets of logged event data in the FIFO.
	11:8	EVENTLOG*n*_FIFO_WPTR[3:0]	0x0	Event Log FIFO Write Pointer. Indicates the FIFO index location in which the next event is logged. This is a read-only field.
	3:0	EVENTLOG*n*_FIFO_RPTR[3:0]	0x0	Event Log FIFO Read Pointer. Indicates the FIFO index location of the first set of unread data, if any exists. For the intended FIFO behavior, this field must be incremented after the respective data has been read.

Table 4-34. Event Logger (EVENTLOGn) Control *(Cont.)*

Register Address	Bit	Label	Default	Description
Base address +0x20 EVENTLOG*n*_CH_ENABLE	15	EVENTLOG*n*_CH16_ENA	0	Event Log Channel 16 Enable 0 = Disabled, 1 = Enabled
	14	EVENTLOG*n*_CH15_ENA	0	Event Log Channel 15 Enable 0 = Disabled, 1 = Enabled
	13	EVENTLOG*n*_CH14_ENA	0	Event Log Channel 14 Enable 0 = Disabled, 1 = Enabled
	12	EVENTLOG*n*_CH13_ENA	0	Event Log Channel 13 Enable 0 = Disabled, 1 = Enabled
	11	EVENTLOG*n*_CH12_ENA	0	Event Log Channel 12 Enable 0 = Disabled, 1 = Enabled
	10	EVENTLOG*n*_CH11_ENA	0	Event Log Channel 11 Enable 0 = Disabled, 1 = Enabled
	9	EVENTLOG*n*_CH10_ENA	0	Event Log Channel 10 Enable 0 = Disabled, 1 = Enabled
	8	EVENTLOG*n*_CH9_ENA	0	Event Log Channel 9 Enable 0 = Disabled, 1 = Enabled
	7	EVENTLOG*n*_CH8_ENA	0	Event Log Channel 8 Enable 0 = Disabled, 1 = Enabled
	6	EVENTLOG*n*_CH7_ENA	0	Event Log Channel 7 Enable 0 = Disabled, 1 = Enabled
	5	EVENTLOG*n*_CH6_ENA	0	Event Log Channel 6 Enable 0 = Disabled, 1 = Enabled
	4	EVENTLOG*n*_CH5_ENA	0	Event Log Channel 5 Enable 0 = Disabled, 1 = Enabled
	3	EVENTLOG*n*_CH4_ENA	0	Event Log Channel 4 Enable 0 = Disabled, 1 = Enabled
	2	EVENTLOG*n*_CH3_ENA	0	Event Log Channel 3 Enable 0 = Disabled, 1 = Enabled
	1	EVENTLOG*n*_CH2_ENA	0	Event Log Channel 2 Enable 0 = Disabled, 1 = Enabled
	0	EVENTLOG*n*_CH1_ENA	0	Event Log Channel 1 Enable 0 = Disabled, 1 = Enabled
Base address +0x40 EVENTLOG*n*_CH1_DEFINE	15	EVENTLOG*n*_CH1_DB	0	Event Log Channel 1 debounce 0 = Disabled, 1 = Enabled Note that channel must be disabled when updating this field
	14	EVENTLOG*n*_CH1_POL	0	Event Log Channel 1 polarity 0 = Rising edge triggered, 1 = Falling edge triggered Note that channel must be disabled when updating this field
	8:0	EVENTLOG*n*_CH1_SEL[8:0]	0x000	Event Log Channel 1 source [1] Note that channel must be disabled when updating this field
Base address +0x42 EVENTLOG*n*_CH2_DEFINE	15	EVENTLOG*n*_CH2_DB	0	Event Log Channel 2 debounce 0 = Disabled, 1 = Enabled Note that channel must be disabled when updating this field
	14	EVENTLOG*n*_CH2_POL	0	Event Log Channel 2 polarity 0 = Rising edge triggered, 1 = Falling edge triggered Note that channel must be disabled when updating this field
	8:0	EVENTLOG*n*_CH2_SEL[8:0]	0x000	Event Log Channel 2 source [1] Field description is as above.
Base address +0x44 EVENTLOG*n*_CH3_DEFINE	15	EVENTLOG*n*_CH3_DB	0	Event Log Channel 3 debounce 0 = Disabled, 1 = Enabled Note that channel must be disabled when updating this field
	14	EVENTLOG*n*_CH3_POL	0	Event Log Channel 3 polarity 0 = Rising edge triggered, 1 = Falling edge triggered Note that channel must be disabled when updating this field
	8:0	EVENTLOG*n*_CH3_SEL[8:0]	0x000	Event Log Channel 3 source [1] Field description is as above.

Table 4-34. Event Logger (EVENTLOG*n*) Control *(Cont.)*

Register Address	Bit	Label	Default	Description
Base address +0x46 EVENTLOG*n*_CH4_DEFINE	15	EVENTLOG*n*_CH4_DB	0	Event Log Channel 4 debounce 0 = Disabled, 1 = Enabled Note that channel must be disabled when updating this field
	14	EVENTLOG*n*_CH4_POL	0	Event Log Channel 4 polarity 0 = Rising edge triggered, 1 = Falling edge triggered Note that channel must be disabled when updating this field
	8:0	EVENTLOG*n*_CH4_SEL[8:0]	0x000	Event Log Channel 4 source [1] Field description is as above.
Base address +0x48 EVENTLOG*n*_CH5_DEFINE	15	EVENTLOG*n*_CH5_DB	0	Event Log Channel 5 debounce 0 = Disabled, 1 = Enabled Note that channel must be disabled when updating this field
	14	EVENTLOG*n*_CH5_POL	0	Event Log Channel 5 polarity 0 = Rising edge triggered, 1 = Falling edge triggered Note that channel must be disabled when updating this field
	8:0	EVENTLOG*n*_CH5_SEL[8:0]	0x000	Event Log Channel 5 source [1] Field description is as above.
Base address +0x4A EVENTLOG*n*_CH6_DEFINE	15	EVENTLOG*n*_CH6_DB	0	Event Log Channel 6 debounce 0 = Disabled, 1 = Enabled Note that channel must be disabled when updating this field
	14	EVENTLOG*n*_CH6_POL	0	Event Log Channel 6 polarity 0 = Rising edge triggered, 1 = Falling edge triggered Note that channel must be disabled when updating this field
	8:0	EVENTLOG*n*_CH6_SEL[8:0]	0x000	Event Log Channel 6 source [1] Field description is as above.
Base address +0x4C EVENTLOG*n*_CH7_DEFINE	15	EVENTLOG*n*_CH7_DB	0	Event Log Channel 7 debounce 0 = Disabled, 1 = Enabled Note that channel must be disabled when updating this field
	14	EVENTLOG*n*_CH7_POL	0	Event Log Channel 7 polarity 0 = Rising edge triggered, 1 = Falling edge triggered Note that channel must be disabled when updating this field
	8:0	EVENTLOG*n*_CH7_SEL[8:0]	0x000	Event Log Channel 7 source [1] Field description is as above.
Base address +0x4E EVENTLOG*n*_CH8_DEFINE	15	EVENTLOG*n*_CH8_DB	0	Event Log Channel 8 debounce 0 = Disabled, 1 = Enabled Note that channel must be disabled when updating this field
	14	EVENTLOG*n*_CH8_POL	0	Event Log Channel 8 polarity 0 = Rising edge triggered, 1 = Falling edge triggered Note that channel must be disabled when updating this field
	8:0	EVENTLOG*n*_CH8_SEL[8:0]	0x000	Event Log Channel 8 source [1] Field description is as above.
Base address +0x50 EVENTLOG*n*_CH9_DEFINE	15	EVENTLOG*n*_CH9_DB	0	Event Log Channel 9 debounce 0 = Disabled, 1 = Enabled Note that channel must be disabled when updating this field
	14	EVENTLOG*n*_CH9_POL	0	Event Log Channel 9 polarity 0 = Rising edge triggered, 1 = Falling edge triggered Note that channel must be disabled when updating this field
	8:0	EVENTLOG*n*_CH9_SEL[8:0]	0x000	Event Log Channel 9 source [1] Field description is as above.
Base address +0x52 EVENTLOG*n*_CH10_ DEFINE	15	EVENTLOG*n*_CH10_DB	0	Event Log Channel 10 debounce 0 = Disabled, 1 = Enabled Note that channel must be disabled when updating this field
	14	EVENTLOG*n*_CH10_POL	0	Event Log Channel 10 polarity 0 = Rising edge triggered, 1 = Falling edge triggered Note that channel must be disabled when updating this field
	8:0	EVENTLOG*n*_CH10_SEL[8:0]	0x000	Event Log Channel 10 source [1] Field description is as above.

Table 4-34. Event Logger (EVENTLOG*n*) Control *(Cont.)*

Register Address	Bit	Label	Default	Description
Base address +0x54 EVENTLOG*n*_CH11_DEFINE	15	EVENTLOG*n*_CH11_DB	0	Event Log Channel 11 debounce 0 = Disabled, 1 = Enabled Note that channel must be disabled when updating this field
	14	EVENTLOG*n*_CH11_POL	0	Event Log Channel 11 polarity 0 = Rising edge triggered, 1 = Falling edge triggered Note that channel must be disabled when updating this field
	8:0	EVENTLOG*n*_CH11_SEL[8:0]	0x000	Event Log Channel 11 source [1] Field description is as above.
Base address +0x56 EVENTLOG*n*_CH12_DEFINE	15	EVENTLOG*n*_CH12_DB	0	Event Log Channel 12 debounce 0 = Disabled, 1 = Enabled Note that channel must be disabled when updating this field
	14	EVENTLOG*n*_CH12_POL	0	Event Log Channel 12 polarity 0 = Rising edge triggered, 1 = Falling edge triggered Note that channel must be disabled when updating this field
	8:0	EVENTLOG*n*_CH12_SEL[8:0]	0x000	Event Log Channel 12 source [1] Field description is as above.
Base address +0x58 EVENTLOG*n*_CH13_DEFINE	15	EVENTLOG*n*_CH13_DB	0	Event Log Channel 13 debounce 0 = Disabled, 1 = Enabled Note that channel must be disabled when updating this field
	14	EVENTLOG*n*_CH13_POL	0	Event Log Channel 13 polarity 0 = Rising edge triggered, 1 = Falling edge triggered Note that channel must be disabled when updating this field
	8:0	EVENTLOG*n*_CH13_SEL[8:0]	0x000	Event Log Channel 13 source [1] Field description is as above.
Base address +0x5A EVENTLOG*n*_CH14_DEFINE	15	EVENTLOG*n*_CH14_DB	0	Event Log Channel 14 debounce 0 = Disabled, 1 = Enabled Note that channel must be disabled when updating this field
	14	EVENTLOG*n*_CH14_POL	0	Event Log Channel 14 polarity 0 = Rising edge triggered, 1 = Falling edge triggered Note that channel must be disabled when updating this field
	8:0	EVENTLOG*n*_CH14_SEL[8:0]	0x000	Event Log Channel 14 source [1] Field description is as above.
Base address +0x5C EVENTLOG*n*_CH15_DEFINE	15	EVENTLOG*n*_CH15_DB	0	Event Log Channel 15 debounce 0 = Disabled, 1 = Enabled Note that channel must be disabled when updating this field
	14	EVENTLOG*n*_CH15_POL	0	Event Log Channel 15 polarity 0 = Rising edge triggered, 1 = Falling edge triggered Note that channel must be disabled when updating this field
	8:0	EVENTLOG*n*_CH15_SEL[8:0]	0x000	Event Log Channel 15 source [1] Field description is as above.
Base address +0x5E EVENTLOG*n*_CH16_DEFINE	15	EVENTLOG*n*_CH16_DB	0	Event Log Channel 16 debounce 0 = Disabled, 1 = Enabled Note that channel must be disabled when updating this field
	14	EVENTLOG*n*_CH16_POL	0	Event Log Channel 16 polarity 0 = Rising edge triggered, 1 = Falling edge triggered Note that channel must be disabled when updating this field
	8:0	EVENTLOG*n*_CH16_SEL[8:0]	0x000	Event Log Channel 16 source [1] Field description is as above.
Base address +0x80 EVENTLOG*n*_FIFO0_READ	12	EVENTLOG*n*_FIFO0_POL	0	Event Log FIFO Index 0 polarity 0 = Rising edge, 1 = Falling edge
	8:0	EVENTLOG*n*_FIFO0_ID[8:0]	0x000	Event Log FIFO Index 0 source [1]
Base address +0x82 EVENTLOG*n*_FIFO0_TIME	31:0	EVENTLOG*n*_FIFO0_TIME[31:0]	0x0000_0000	Event Log FIFO Index 0 Time
Base address +0x84 EVENTLOG*n*_FIFO1_READ	12	EVENTLOG*n*_FIFO1_POL	0	Event Log FIFO Index 1 polarity 0 = Rising edge, 1 = Falling edge
	8:0	EVENTLOG*n*_FIFO1_ID[8:0]	0x000	Event Log FIFO Index 1 source [1]

Table 4-34. Event Logger (EVENTLOG*n*) Control *(Cont.)*

Register Address	Bit	Label	Default	Description
Base address +0x86 EVENTLOG*n*_FIFO1_TIME	31:0	EVENTLOG*n*_FIFO1_TIME[31:0]	0x0000_0000	Event Log FIFO Index 1 Time
Base address +0x88 EVENTLOG*n*_FIFO2_READ	12	EVENTLOG*n*_FIFO2_POL	0	Event Log FIFO Index 2 polarity 0 = Rising edge, 1 = Falling edge
	8:0	EVENTLOG*n*_FIFO2_ID[8:0]	0x000	Event Log FIFO Index 2 source [1]
Base address +0x8A EVENTLOG*n*_FIFO2_TIME	31:0	EVENTLOG*n*_FIFO2_TIME[31:0]	0x0000_0000	Event Log FIFO Index 2 Time
Base address +0x8C EVENTLOG*n*_FIFO3_READ	12	EVENTLOG*n*_FIFO3_POL	0	Event Log FIFO Index 3 polarity 0 = Rising edge, 1 = Falling edge
	8:0	EVENTLOG*n*_FIFO3_ID[8:0]	0x000	Event Log FIFO Index 3 source [1]
Base address +0x8E EVENTLOG*n*_FIFO3_TIME	31:0	EVENTLOG*n*_FIFO3_TIME[31:0]	0x0000_0000	Event Log FIFO Index 3 Time
Base address +0x90 EVENTLOG*n*_FIFO4_READ	12	EVENTLOG*n*_FIFO4_POL	0	Event Log FIFO Index 4 polarity 0 = Rising edge, 1 = Falling edge
	8:0	EVENTLOG*n*_FIFO4_ID[8:0]	0x000	Event Log FIFO Index 4 source [1]
Base address +0x92 EVENTLOG*n*_FIFO4_TIME	31:0	EVENTLOG*n*_FIFO4_TIME[31:0]	0x0000_0000	Event Log FIFO Index 4 Time
Base address +0x94 EVENTLOG*n*_FIFO5_READ	12	EVENTLOG*n*_FIFO5_POL	0	Event Log FIFO Index 5 polarity 0 = Rising edge, 1 = Falling edge
	8:0	EVENTLOG*n*_FIFO5_ID[8:0]	0x000	Event Log FIFO Index 5 source [1]
Base address +0x96 EVENTLOG*n*_FIFO5_TIME	31:0	EVENTLOG*n*_FIFO5_TIME[31:0]	0x0000_0000	Event Log FIFO Index 5 Time
Base address +0x98 EVENTLOG*n*_FIFO6_READ	12	EVENTLOG*n*_FIFO6_POL	0	Event Log FIFO Index 6 polarity 0 = Rising edge, 1 = Falling edge
	8:0	EVENTLOG*n*_FIFO6_ID[8:0]	0x000	Event Log FIFO Index 6 source [1]
Base address +0x9A EVENTLOG*n*_FIFO6_TIME	31:0	EVENTLOG*n*_FIFO6_TIME[31:0]	0x0000_0000	Event Log FIFO Index 6 Time
Base address +0x9C EVENTLOG*n*_FIFO7_READ	12	EVENTLOG*n*_FIFO7_POL	0	Event Log FIFO Index 7 polarity 0 = Rising edge, 1 = Falling edge
	8:0	EVENTLOG*n*_FIFO7_ID[8:0]	0x000	Event Log FIFO Index 7 source [1]
Base address +0x9E EVENTLOG*n*_FIFO7_TIME	31:0	EVENTLOG*n*_FIFO7_TIME[31:0]	0x0000_0000	Event Log FIFO Index 7 Time
Base address +0xA0 EVENTLOG*n*_FIFO8_READ	12	EVENTLOG*n*_FIFO8_POL	0	Event Log FIFO Index 8 polarity 0 = Rising edge, 1 = Falling edge
	8:0	EVENTLOG*n*_FIFO8_ID[8:0]	0x000	Event Log FIFO Index 8 source [1]
Base address +0xA2 EVENTLOG*n*_FIFO8_TIME	31:0	EVENTLOG*n*_FIFO8_TIME[31:0]	0x0000_0000	Event Log FIFO Index 8 Time
Base address +0xA4 EVENTLOG*n*_FIFO9_READ	12	EVENTLOG*n*_FIFO9_POL	0	Event Log FIFO Index 9 polarity 0 = Rising edge, 1 = Falling edge
	8:0	EVENTLOG*n*_FIFO9_ID[8:0]	0x000	Event Log FIFO Index 9 source [1]
Base address +0xA6 EVENTLOG*n*_FIFO9_TIME	31:0	EVENTLOG*n*_FIFO9_TIME[31:0]	0x0000_0000	Event Log FIFO Index 9 Time
Base address +0xA8 EVENTLOG*n*_FIFO10_READ	12	EVENTLOG*n*_FIFO10_POL	0	Event Log FIFO Index 10 polarity 0 = Rising edge, 1 = Falling edge
	8:0	EVENTLOG*n*_FIFO10_ID[8:0]	0x000	Event Log FIFO Index 10 source [1]
Base address +0xAA EVENTLOG*n*_FIFO10_TIME	31:0	EVENTLOG*n*_FIFO10_TIME[31:0]	0x0000_0000	Event Log FIFO Index 10 Time
Base address +0xAC EVENTLOG*n*_FIFO11_READ	12	EVENTLOG*n*_FIFO11_POL	0	Event Log FIFO Index 11 polarity 0 = Rising edge, 1 = Falling edge
	8:0	EVENTLOG*n*_FIFO11_ID[8:0]	0x000	Event Log FIFO Index 11 source [1]
Base address +0xAE EVENTLOG*n*_FIFO11_TIME	31:0	EVENTLOG*n*_FIFO11_TIME[31:0]	0x0000_0000	Event Log FIFO Index 11 Time
Base address +0xB0 EVENTLOG*n*_FIFO12_READ	12	EVENTLOG*n*_FIFO12_POL	0	Event Log FIFO Index 12 polarity 0 = Rising edge, 1 = Falling edge
	8:0	EVENTLOG*n*_FIFO12_ID[8:0]	0x000	Event Log FIFO Index 12 source [1]

Table 4-34. Event Logger (EVENTLOG*n*) Control *(Cont.)*

Register Address	Bit	Label	Default	Description
Base address +0xB2 EVENTLOG*n*_FIFO12_TIME	31:0	EVENTLOG*n*_FIFO12_ TIME[31:0]	0x0000 _0000	Event Log FIFO Index 12 Time
Base address +0xB4 EVENTLOG*n*_FIFO13_ READ	12	EVENTLOG*n*_FIFO13_POL	0	Event Log FIFO Index 13 polarity 0 = Rising edge, 1 = Falling edge
	8:0	EVENTLOG*n*_FIFO13_ID[8:0]	0x000	Event Log FIFO Index 13 source [1]
Base address +0xB6 EVENTLOG*n*_FIFO13_TIME	31:0	EVENTLOG*n*_FIFO13_ TIME[31:0]	0x0000 _0000	Event Log FIFO Index 13 Time
Base address +0xB8 EVENTLOG*n*_FIFO14_ READ	12	EVENTLOG*n*_FIFO14_POL	0	Event Log FIFO Index 14 polarity 0 = Rising edge, 1 = Falling edge
	8:0	EVENTLOG*n*_FIFO14_ID[8:0]	0x000	Event Log FIFO Index 14 source [1]
Base address +0xBA EVENTLOG*n*_FIFO14_TIME	31:0	EVENTLOG*n*_FIFO14_ TIME[31:0]	0x0000 _0000	Event Log FIFO Index 14 Time
Base address +0xBC EVENTLOG*n*_FIFO15_ READ	12	EVENTLOG*n*_FIFO15_POL	0	Event Log FIFO Index 15 polarity 0 = Rising edge, 1 = Falling edge
	8:0	EVENTLOG*n*_FIFO15_ID[8:0]	0x000	Event Log FIFO Index 15 source [1]
Base address +0xBE EVENTLOG*n*_FIFO15_TIME	31:0	EVENTLOG*n*_FIFO15_ TIME[31:0]	0x0000 _0000	Event Log FIFO Index 15 Time

1. See Table 4-35 for valid channel source selections

4.5.1.8 Event Logger Input Sources

A list of the valid input sources for the event loggers is provided in Table 4-35.

The EDGE type noted is coded as *S* (single edge) or *D* (dual edge). Note that a single-edge input source only provides valid input to the event logger in the default (rising edge triggered) polarity.

Caution is advised when enabling IRQ1 or IRQ2 as an input source for the event loggers; a recursive loop, where the IRQ*n* signal is also an output from the same event logger, must be avoided.

Table 4-35. Event Logger Input Sources

ID	Description	Edge
3	irq1	D
4	irq2	D
9	sysclk_fail	S
24	fll1_lock	D
27	fll_ao_lock	D
32	frame_start_g1r1	S
33	frame_start_g1r2	S
34	frame_start_g1r3	S
80	hpdet	S
88	micdet1	S
89	micdet2	S
96	jd1_rise	S
97	jd1_fall	S
98	jd2_rise	S
99	jd2_fall	S
100	micd_clamp_rise	S
101	micd_clamp_fall	S
128	drc1_sig_det	D
129	drc2_sig_det	D
160	dsp_irq1	S
161	dsp_irq2	S
162	dsp_irq3	S
163	dsp_irq4	S
164	dsp_irq5	S
165	dsp_irq6	S
166	dsp_irq7	S
167	dsp_irq8	S
168	dsp_irq9	S
169	dsp_irq10	S
170	dsp_irq11	S
171	dsp_irq12	S
172	dsp_irq13	S
173	dsp_irq14	S
174	dsp_irq15	S
175	dsp_irq16	S
176	hp1l_sc	S
177	hp1r_sc	S
178	hp2l_sc	S
179	hp2r_sc	S
182	spkoutl_short	D
224	spk_shutdown	D
225	spk_overheat	S
226	spk_overheat_warn	S
256	gpio1	D
257	gpio2	D
258	gpio3	D
259	gpio4	D
260	gpio5	D
261	gpio6	D
262	gpio7	D
263	gpio8	D
264	gpio9	D
265	gpio10	D
266	gpio11	D
267	gpio12	D
268	gpio13	D
269	gpio14	D
270	gpio15	D
320	Timer1	S
321	Timer2	S
336	event1_not_empty	S
337	event2_not_empty	S
352	event1_full	S
353	event2_full	S
368	event1_wmark	S
369	event2_wmark	S
384	dsp1_dma	S
416	dsp1_start1	S
432	dsp1_start2	S
448	dsp1_start	S
464	dsp1_busy	D

4.5.2 General-Purpose Timers

The CS47L15 incorporates two general-purpose timers, which support a wide variety of uses. The general-purpose timers provide time-stamp data for the event logs; they also support the watchdog and other miscellaneous time-based functions, providing additional capability for signal-processing applications.

4.5.2.1 Overview

The timers allow time-stamp information to be associated with external signal detection, and other system events, enabling real-time data to be more easily integrated into user applications. The timers allow many advanced functions to be implemented with a high degree of autonomy from a host processor.

The timers can use either internal system clocks, or external clock signals, as a reference. The selected reference is scaled down, using configurable dividers, to the required clock count frequency.

4.5.2.2 Timer Control

The reference clock for each timer is selected using TIMER*n*_REFCLK_SRC, (where *n* identifies the applicable timer, 1 or 2).

If SYSCLK or DSPCLK is selected, a lower clock frequency, derived from the applicable system clock, can be selected using the TIMER*n*_REFCLK_FREQ_SEL field (for SYSCLK source) or the TIMER*n*_DSPCLK_FREQ_SEL field (for DSPCLK source). The applicable division ratio is determined automatically, assuming the respective clock source has been correctly configured as described in Section 4.13.

Note that, depending on the DSPCLK frequency and the available clock dividers, the timer reference clock may differ from the selected clock if DSPCLK is the selected source. In most cases, the reference clock frequency equals or exceeds the requested frequency. A lower frequency is implemented if limited by either the DSPCLK frequency or the maximum TIMER*n* clocking frequency.

If any source other than DSPCLK is selected, the clock can be further divided using TIMER*n*_REFCLK_DIV. Division ratios in the range 1 to 128 can be selected.

Note that, if DSPCLK is enabled, the CS47L15 synchronizes the selected reference clock to DSPCLK. As a result of this, if a non-DSPCLK is selected as source, the following additional constraints must be observed: the reference clock frequency (after TIMER*n*_REFCLK_FREQ_SEL and after TIMER*n*_REFCLK_DIV) must be less than DSPCLK / 3, and must be less than 12 MHz; it must also be close to 50% duty cycle. The TIMER*n*_REFCLK_DIV field can be used to ensure that these criteria are met.

One final division, controlled by TIMER*n*_PRESCALE, determines the timer count frequency. This field is valid for all clock reference sources; division ratios in the range 1 to 128 can be selected. The output from this division corresponds to the frequency at which the TIMER*n*_COUNT fields are incremented (or decremented).

The maximum count value of the timer is determined by the TIMER*n*_MAX_COUNT field. This is the final count value (when counting up), or the initial count value (when counting down). The current value of the timer counter can be read from the TIMER*n*_CUR_COUNT field.

The timer is started by writing 1 to TIMER*n*_START. Note that, if the timer is already running, it restarts from its initial value. The timer is stopped by writing 1 to TIMER*n*_STOP. The count direction (up or down) is selected using the TIMER*n*_DIR bit.

The TIMER*n*_CONTINUOUS bit selects whether the timer automatically restarts after the end-of-count condition has been reached. The TIMER*n*_RUNNING_STS indicates whether the timer is running, or if it has stopped.

Note that the timers should be stopped before making any changes to the respective configuration registers. The timer configuration should only be changed if TIMER*n*_RUNNING_STS = 0.

4.5.2.3 Interrupts, GPIO, and Class D Speaker Driver Control

The timer status is an input to the interrupt control circuit and can be used to trigger an interrupt event after the final count value is reached; see Section 4.12. Note that the interrupt does not occur immediately when the final count value is reached; the interrupt is triggered at the point when the next update to the timer count value would be due.

The timer status can be output directly on a GPIO pin as an external indication of the timer activity. See Section 4.11 to configure a GPIO pin for this function.

The timers can be used as a watchdog function to trigger a shutdown of the Class D speaker drivers. See Section 4.18 to configure this function.

4.5.2.4 Timer Block Diagram and Control Registers

The timer block is shown in Fig. 4-31.

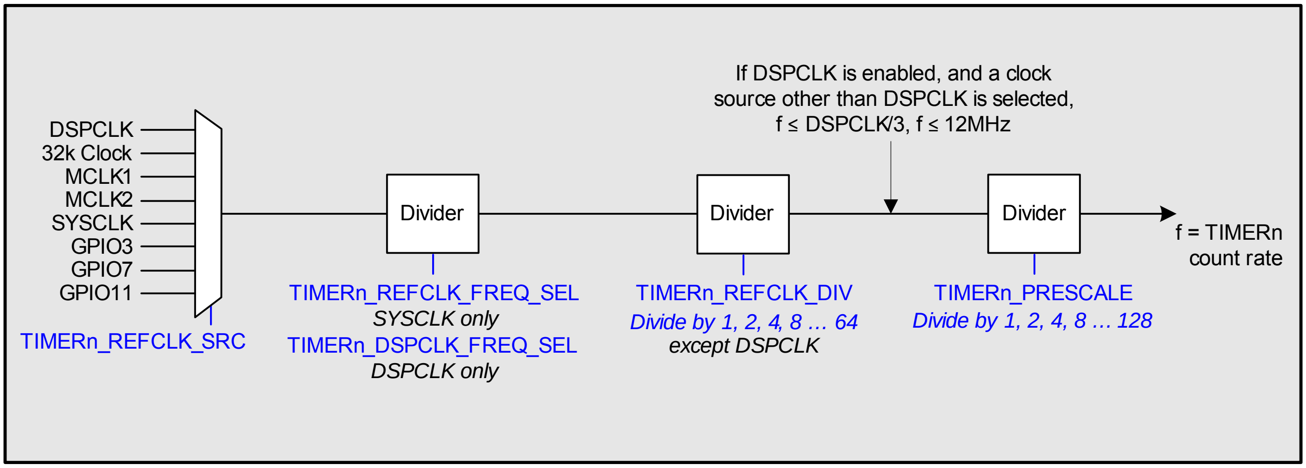


Figure 4-31. General-Purpose Timer

The timer control registers are described in Table 4-36.

Table 4-36. General-Purpose Timer (TIMERn) Control

Register Address	Bit	Label	Default	Description
Timer 1 Base Address = R311296 (0x4_C000) Timer 2 Base Address = R311424 (0x4_C080)				
Base address Timer*n*_Control	21	TIMER*n*_CONTINUOUS	0	Timer Continuous Mode select 0 = Single mode 1 = Continuous mode Timer must be stopped (TIMER*n*_RUNNING_STS = 0) when updating this field
	20	TIMER*n*_DIR	0	Timer Count Direction 0 = Down 1 = Up Timer must be stopped (TIMER*n*_RUNNING_STS = 0) when updating this field
	18:16	TIMER*n*_PRESCALE[2:0]	000	Timer Count Rate Prescale 000 = Divide by 1 011 = Divide by 8 110 = Divide by 64 001 = Divide by 2 100 = Divide by 16 111 = Divide by 128 010 = Divide by 4 101 = Divide by 32 Timer must be stopped (TIMER*n*_RUNNING_STS = 0) when updating this field
	14:12	TIMER*n*_REFCLK_DIV[2:0]	000	Timer Reference Clock Divide (Not valid for DSPCLK source). 000 = Divide by 1 011 = Divide by 8 110 = Divide by 64 001 = Divide by 2 100 = Divide by 16 111 = Divide by 128 010 = Divide by 4 101 = Divide by 32 If DSPCLK is enabled, and DSPCLK is not selected as source, the output frequency from this divider must be set less than or equal to DSPCLK / 3, and less than or equal to 12 MHz. If DSPCLK is disabled, the output of this divider is used as clock reference for any associated event logger. In this case, the divider output corresponds to the frequency of event logging opportunities on the respective modules. Timer must be stopped (TIMER*n*_RUNNING_STS = 0) when updating this field
	10:8	TIMER*n*_REFCLK_FREQ_SEL[2:0]	000	Timer Reference Frequency Select (SYSCLK source) 000 = 6.144 MHz (5.6448 MHz) 001 = 12.288 MHz (11.2896 MHz) 010 = 24.576 MHz (22.5792 MHz) 011 = 49.152 MHz (45.1584 MHz) All other codes are reserved. The selected frequency must be less than or equal to the frequency of the source. Timer must be stopped (TIMER*n*_RUNNING_STS = 0) when updating this field.
	3:0	TIMER*n*_REFCLK_SRC[3:0]	0000	Timer Reference Source Select. Timer must be stopped (TIMER*n*_RUNNING_STS=0) when updating this field. Codes not listed are reserved. 0000 = DSPCLK 0101 = MCLK2 1110 = GPIO7 0001 = 32-kHz clock 1000 = SYSCLK 1111 = GPIO11 0100 = MCLK1 1101 = GPIO3
Base address +0x02 Timer*n*_Count_Preset	31:0	TIMER*n*_MAX_COUNT[31:0]	0x0000_0000	Timer Maximum Count. Final count value (when counting up). Starting count value (when counting down). Timer must be stopped (TIMER*n*_RUNNING_STS = 0) when updating this field.
Base address +0x06 Timer*n*_Start_and_Stop	4	TIMER*n*_STOP	0	Timer Stop Control Write 1 to stop.
	0	TIMER*n*_START	0	Timer Start Control Write 1 to start. If the timer is already running, it restarts from its initial value.
Base address +0x08 Timer*n*_Status	0	TIMER*n*_RUNNING_STS	0	Timer Running Status 0 = Timer stopped 1 = Timer running
Base address +0x0A Timer*n*_Count_Readback	31:0	TIMER*n*_CUR_COUNT[31:0]	0x0000	Timer Current Count value

Table 4-36. General-Purpose Timer (TIMER*n*) Control *(Cont.)*

Register Address	Bit	Label	Default	Description
Base address +0x0C Timer*n*_DSP_ Clock_Config	15:0	TIMER*n*_ DSPCLK_ FREQ_ SEL[15:0]	0x0000	Timer Reference Frequency Select (DSPCLK source) Coded as LSB = 1/64 MHz, Valid from 5.6 MHz to 148 MHz. The timer reference frequency must be less than or equal to the DSPCLK frequency. The timer reference is generated by division of DSPCLK, and may differ from the selected frequency. The timer reference frequency can be read from TIMER*n*_DSPCLK_FREQ_ STS. Timer must be stopped (TIMER*n*_RUNNING_STS=0) when updating this field.
Base address +0x0E Timer*n*_DSP_ Clock_Status	15:0	TIMER*n*_ DSPCLK_ FREQ_ STS[15:0]	0x0000	Timer Reference Frequency (Read only) Only valid when DSPCLK is the selected clock source. Coded as LSB = 1/64 MHz.

4.5.3 DSP GPIO

The DSP GPIO function provides an advanced I/O capability, supporting enhanced flexibility for signal-processing applications.

4.5.3.1 Overview

The CS47L15 supports up to 15 GPIO pins; these are implemented as alternate functions to a pin-specific capability.

The GPIOs can be used to provide status outputs and control signals to external hardware; the supported functions include interrupt output, FLL clock output, accessory detection status, and S/PDIF or PWM-coded audio channels; see Section 4.11.

The GPIOs can support miscellaneous logic input and output, interfacing directly with the integrated DSPs, or with the Host Application software. A basic level of I/O functionality is described in Section 4.11, under the configuration where GP*n*_ FN = 0x001. The GP*n*_FN field selects the functionality for the respective pin, GPIO*n*.

The DSP GPIO pins are accessed using maskable sets of I/O control registers; this allows the selected combinations of GPIOs to be controlled with ease, regardless of how the allocation of GPIO pins has been implemented in hardware. In a typical use case, a different GPIO mask is defined for each DSP function; this provides a highly efficient mechanism for the DSP to access the required input and output signals.

4.5.3.2 DSP GPIO Control

The DSP GPIO function is selected by setting GP*n*_FN = 0x002 for the respective GPIO pin (where *n* identifies the applicable GPIO*n* pin).

Each DSP GPIO is controlled using bits that determine the direction (input/output) and the logic state (0/1) of the pin. These bits are replicated in four control sets; each which can determine the logic level of any DSP GPIO.

Mask bits are provided within each control set, to determine which of the control sets has control of each DSP GPIO. To avoid logic contention, a DSP GPIO output must be controlled (unmasked) in a maximum of one control set at any time.

Note that write access to the direction control bits (DSPGP*n*_SETx_DIR) and level control bits (DSPGP*n*_SETx_LVL) is only valid when the channel (DSPGPn) is unmasked in the respective control set. Writes to these fields are implemented for the unmasked DSP GPIOs, and are ignored in respect of the masked DSP GPIOs. Note that the level control bits (DSPGP*n*_SETx_LVL) provide output level control only—they cannot be used to read the status of DSP GPIO inputs.

The logic level of the unmasked DSP GPIO outputs in any control set can be configured using a single register write. Writing to the output level control registers determines the logic level of the unmasked DSP GPIOs in that set only; all other outputs are unaffected.

DSP GPIO status bits are provided, indicating the logic level of every input or output pin that is configured as a DSP GPIO. The DSPGP*n*_STS bits also provide logic-level indication for any pin that is configured as a GPIO input, with GP*n*_ FN = 0x001.Note that there is only one set of DSP GPIO status bits.

The status bits indicate the logic level of the DSP GPIO outputs. The respective pins are driven as outputs if configured as a DSP GPIO output, and unmasked in one of the control sets. Note that a DSP GPIO continues to be driven as an output, even if the mask bit is subsequently asserted in that set. The pin only ceases to be driven if it is configured as a DSP GPIO input and is unmasked in one of the control sets, or if the pin is configured as an input under a different GP*n*_FN field selection.

4.5.3.3 Common Functions to Standard GPIOs

The DSP GPIO functions are implemented alongside the standard GPIO capability, providing an alternative method of maskable I/O control for all of the GPIO pins. The DSP GPIO control bits in the register map are implemented in a manner that supports efficient read/write access for multiple GPIOs at once.

The DSP GPIO logic is shown in Fig. 4-32, which also shows the control fields relating to the standard GPIO.

The DSP GPIO function is selected by setting GP*n*_FN = 0x002 for the respective GPIO pin. Integrated pull-up and pull-down resistors are provided on each of the GPIO pins, which are also valid for DSP GPIO function. A bus keeper function is supported on the GPIO pins; this is enabled using the respective pull-up and pull-down control bits. The bus keeper function holds the logic level unchanged whenever the pin is undriven (e.g., if the signal is tristated). See Table 4-72 for details of the GPIO pull-up and pull-down control bits.

4.5.3.4 DSP GPIO Block Diagram and Control Registers

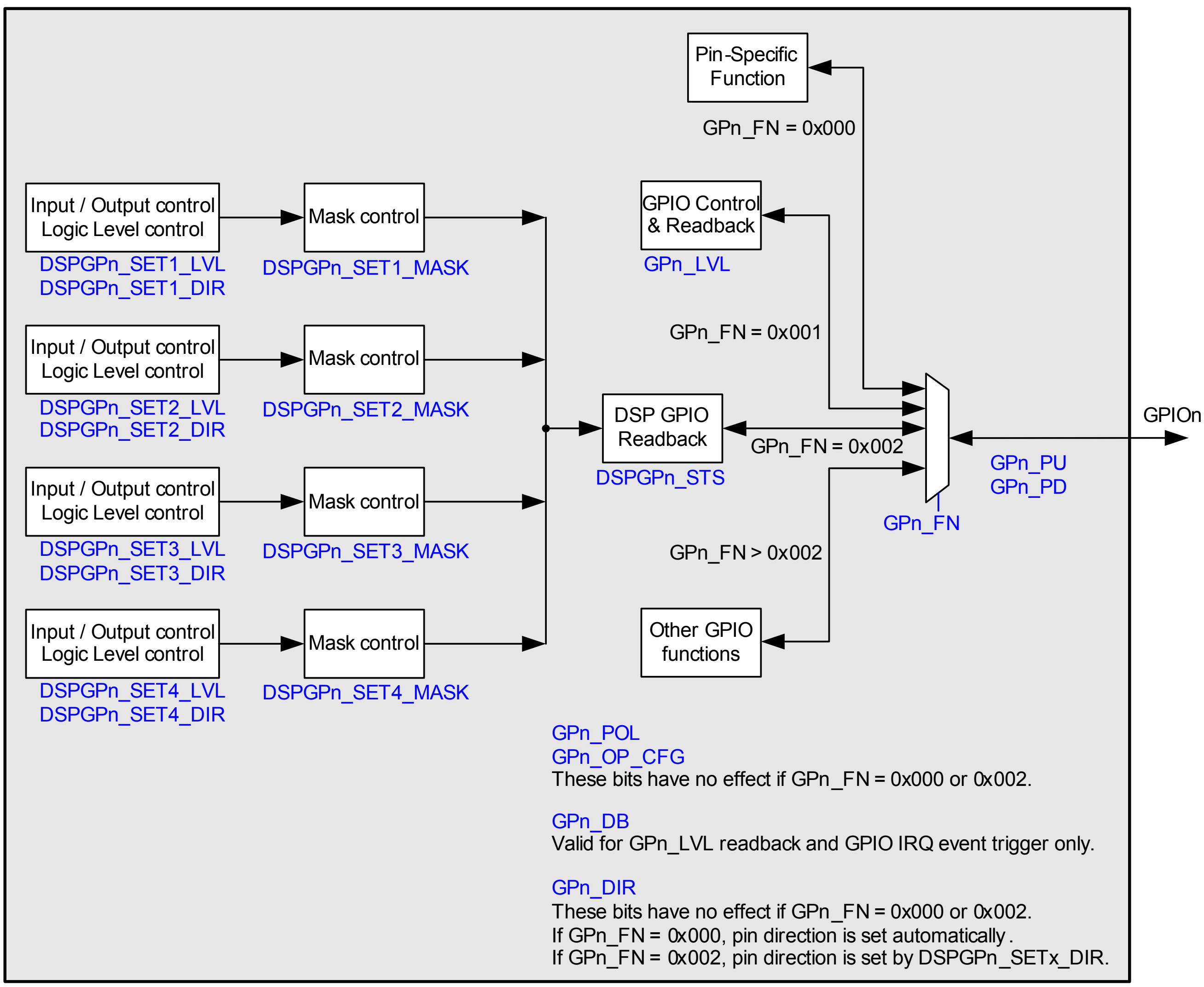


Figure 4-32. DSP GPIO Control

The control registers associated with the DSP GPIO are described in Table 4-37.

Table 4-37. DSP GPIO Control

Register Address	Bit	Label	Default	Description
R315392 (0x4_D000) DSPGP_Status_1	14	DSPGP15_STS	0	DSPGP15 Status Valid for DSPGP input and output
	13	DSPGP14_STS	0	DSPGP14 Status
	12	DSPGP13_STS	0	DSPGP13 Status
	11	DSPGP12_STS	0	DSPGP12 Status
	10	DSPGP11_STS	0	DSPGP11 Status
	9	DSPGP10_STS	0	DSPGP10 Status
	8	DSPGP9_STS	0	DSPGP9 Status
	7	DSPGP8_STS	0	DSPGP8 Status
	6	DSPGP7_STS	0	DSPGP7 Status
	5	DSPGP6_STS	0	DSPGP6 Status
	4	DSPGP5_STS	0	DSPGP5 Status
	3	DSPGP4_STS	0	DSPGP4 Status
	2	DSPGP3_STS	0	DSPGP3 Status
	1	DSPGP2_STS	0	DSPGP2 Status
	0	DSPGP1_STS	0	DSPGP1 Status
R315424 (0x4_D020) DSPGP_SET1_Mask_1 R315456 (0x4_D040) DSPGP_SET2_Mask_1 R315488 (0x4_D060) DSPGP_SET3_Mask_1 R315520 (0x4_D080) DSPGP_SET4_Mask_1	14	DSPGP15_SET*n*_MASK	1	DSP SET*n* GPIO15 Mask Control 0 = Unmasked 1 = Masked A GPIO pin should be unmasked in a maximum of one SET at any time.
	13	DSPGP14_SET*n*_MASK	1	DSP SET*n* GPIO14 Mask Control
	12	DSPGP13_SET*n*_MASK	1	DSP SET*n* GPIO13 Mask Control
	11	DSPGP12_SET*n*_MASK	1	DSP SET*n* GPIO12 Mask Control
	10	DSPGP11_SET*n*_MASK	1	DSP SET*n* GPIO11 Mask Control
	9	DSPGP10_SET*n*_MASK	1	DSP SET*n* GPIO10 Mask Control
	8	DSPGP9_SET*n*_MASK	1	DSP SET*n* GPIO9 Mask Control
	7	DSPGP8_SET*n*_MASK	1	DSP SET*n* GPIO8 Mask Control
	6	DSPGP7_SET*n*_MASK	1	DSP SET*n* GPIO7 Mask Control
	5	DSPGP6_SET*n*_MASK	1	DSP SET*n* GPIO6 Mask Control
	4	DSPGP5_SET*n*_MASK	1	DSP SET*n* GPIO5 Mask Control
	3	DSPGP4_SET*n*_MASK	1	DSP SET*n* GPIO4 Mask Control
	2	DSPGP3_SET*n*_MASK	1	DSP SET*n* GPIO3 Mask Control
	1	DSPGP2_SET*n*_MASK	1	DSP SET*n* GPIO2 Mask Control
	0	DSPGP1_SET*n*_MASK	1	DSP SET*n* GPIO1 Mask Control
R315432 (0x4_D028) DSPGP_SET1_Direction_1 R315464 (0x4_D048) DSPGP_SET2_Direction_1 R315496 (0x4_D068) DSPGP_SET3_Direction_1 R315528 (0x4_D088) DSPGP_SET4_Direction_1	14	DSPGP15_SET*n*_DIR	1	DSP SET*n* GPIO15 Direction Control 0 = Output 1 = Input
	13	DSPGP14_SET*n*_DIR	1	DSP SET*n* GPIO14 Direction Control
	12	DSPGP13_SET*n*_DIR	1	DSP SET*n* GPIO13 Direction Control
	11	DSPGP12_SET*n*_DIR	1	DSP SET*n* GPIO12 Direction Control
	10	DSPGP11_SET*n*_DIR	1	DSP SET*n* GPIO11 Direction Control
	9	DSPGP10_SET*n*_DIR	1	DSP SET*n* GPIO10 Direction Control
	8	DSPGP9_SET*n*_DIR	1	DSP SET*n* GPIO9 Direction Control
	7	DSPGP8_SET*n*_DIR	1	DSP SET*n* GPIO8 Direction Control
	6	DSPGP7_SET*n*_DIR	1	DSP SET*n* GPIO7 Direction Control
	5	DSPGP6_SET*n*_DIR	1	DSP SET*n* GPIO6 Direction Control
	4	DSPGP5_SET*n*_DIR	1	DSP SET*n* GPIO5 Direction Control
	3	DSPGP4_SET*n*_DIR	1	DSP SET*n* GPIO4 Direction Control
	2	DSPGP3_SET*n*_DIR	1	DSP SET*n* GPIO3 Direction Control
	1	DSPGP2_SET*n*_DIR	1	DSP SET*n* GPIO2 Direction Control
	0	DSPGP1_SET*n*_DIR	1	DSP SET*n* GPIO1 Direction Control

Table 4-37. DSP GPIO Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R315440 (0x4_D030) DSPGP_SET1_Level_1 R315472 (0x4_D050) DSPGP_SET2_Level_1 R315504 (0x4_D070) DSPGP_SET3_Level_1 R315536 (0x4_D090) DSPGP_SET4_Level_1	14	DSPGP15_SET*n*_LVL	0	DSP SET*n* GPIO15 Output Level 0 = Logic 0 1 = Logic 1
	13	DSPGP14_SET*n*_LVL	0	DSP SET*n* GPIO14 Output Level
	12	DSPGP13_SET*n*_LVL	0	DSP SET*n* GPIO13 Output Level
	11	DSPGP12_SET*n*_LVL	0	DSP SET*n* GPIO12 Output Level
	10	DSPGP11_SET*n*_LVL	0	DSP SET*n* GPIO11 Output Level
	9	DSPGP10_SET*n*_LVL	0	DSP SET*n* GPIO10 Output Level
	8	DSPGP9_SET*n*_LVL	0	DSP SET*n* GPIO9 Output Level
	7	DSPGP8_SET*n*_LVL	0	DSP SET*n* GPIO8 Output Level
	6	DSPGP7_SET*n*_LVL	0	DSP SET*n* GPIO7 Output Level
	5	DSPGP6_SET*n*_LVL	0	DSP SET*n* GPIO6 Output Level
	4	DSPGP5_SET*n*_LVL	0	DSP SET*n* GPIO5 Output Level
	3	DSPGP4_SET*n*_LVL	0	DSP SET*n* GPIO4 Output Level
	2	DSPGP3_SET*n*_LVL	0	DSP SET*n* GPIO3 Output Level
	1	DSPGP2_SET*n*_LVL	0	DSP SET*n* GPIO2 Output Level
	0	DSPGP1_SET*n*_LVL	0	DSP SET*n* GPIO1 Output Level

4.6 Digital Audio Interface

The CS47L15 provides three audio interfaces, AIF1, AIF2, and AIF3. Each of these is independently configurable on the respective transmit (TX) and receive (RX) paths. AIF1 supports up to six channels of input and output signal paths; AIF2 supports up to four channels of input and output signal paths; AIF3 supports up to two channels of input and output signal paths.

The data sources for the audio interface transmit (TX) paths can be selected from any of the CS47L15 input signal paths, or from the digital-core processing functions. The audio interface receive (RX) paths can be selected as inputs to any of the digital-core processing functions or digital-core outputs. See Section 4.3 for details of the digital-core routing options.

The digital audio interfaces provide flexible connectivity for multiple processors and other audio devices. Typical connections include applications processor, baseband processor, and wireless transceiver. A typical configuration is shown in Fig. 4-33.

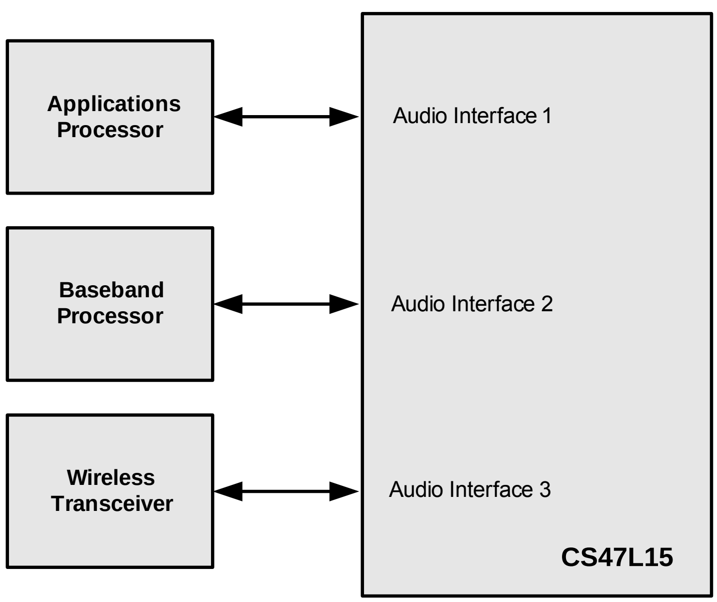


Figure 4-33. Typical AIF Connections

In the general case, the digital audio interface uses four pins:

- TXDAT: data output
- RXDAT: data input
- BCLK: bit clock, for synchronization
- LRCLK: left/right data-alignment clock

In Master Mode, the clock signals BCLK and LRCLK are outputs from the CS47L15. In Slave Mode, these signals are inputs, as shown in Section 4.6.1.

The following interface formats are supported on AIF1–AIF3:

- DSP Mode A.
- DSP Mode B
- I^{2}S
- Left-justified

The left-justified and DSP-B formats are valid in Master Mode only (i.e., BCLK and LRCLK are outputs from the CS47L15). These modes cannot be supported in Slave Mode.

The audio interface formats are described in Section 4.6.2. The bit order is MSB-first in each case; data words are encoded in 2's complement format. Mono PCM operation can be supported using the DSP modes. Refer to Table 3-16 through Table 3-18 for signal timing information.

4.6.1 Master and Slave Mode Operation

The CS47L15 digital audio interfaces can operate as a master or slave, as shown in Fig. 4-34 and Fig. 4-35. The associated control bits are described in Section 4.7.

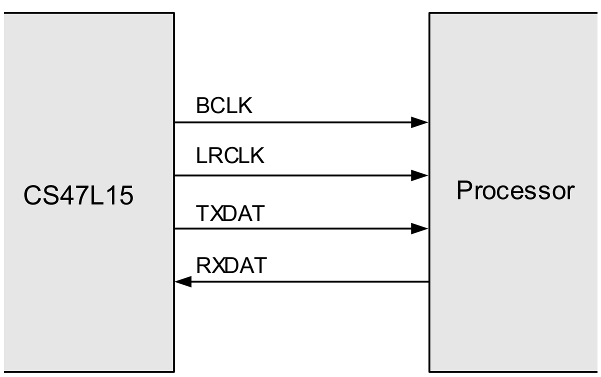


Figure 4-34. Master Mode

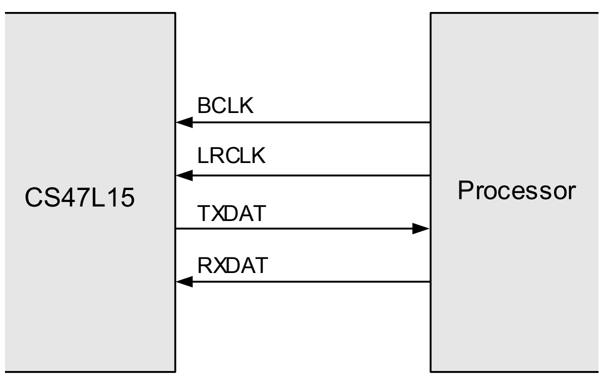


Figure 4-35. Slave Mode

4.6.2 Audio Data Formats

The CS47L15 digital audio interfaces can be configured to operate in I^{2}S, left-justified, DSP-A, or DSP-B interface modes. Note that left-justified and DSP-B modes are valid in Master Mode only (i.e., BCLK and LRCLK are outputs from the CS47L15).

The digital audio interfaces also provide flexibility to support multiple slots of audio data within each LRCLK frame. This flexibility allows multiple audio channels to be supported within a single LRCLK frame.

The data formats described in this section are generic descriptions, assuming only one stereo pair of audio samples per LRCLK frame. In these cases, the AIF is configured to transmit (or receive) in the first available position in each frame (i.e., the Slot 0 position).

The options for multichannel operation are described in Section 4.6.3.

The audio data modes supported by the CS47L15 are described as follows. Note that the BCLK and LRCLK signals are configurable—the polarity of these signals can be inverted if required, and the timing of the LRCLK transition can also be adjusted. The following descriptions all assume the default configuration (noninverted polarity, normal timing) of these signals.

- In DSP modes, the left channel MSB is available on either the first (Mode B) or second (Mode A) rising edge of BCLK following a rising edge of LRCLK. Right-channel data immediately follows left channel data. Depending on word length, BCLK frequency, and sample rate, there may be unused BCLK cycles between the LSB of the right channel data and the next sample.

  In Master Mode, the LRCLK output resembles the frame pulse shown in Fig. 4-36 and Fig. 4-37. In Slave Mode, it is possible to use any length of frame pulse less than 1/Fs, providing the falling edge of the frame pulse occurs at least one BCLK period before the rising edge of the next frame pulse.

  PCM operation is supported in DSP interface mode. CS47L15 data that is output on the left channel is read as mono data by the receiving equipment. Mono PCM data received by the CS47L15 is treated as left-channel data. This may be routed to the left/right playback paths using the control fields described in Section 4.3.

  DSP Mode A data format is shown in Fig. 4-36.

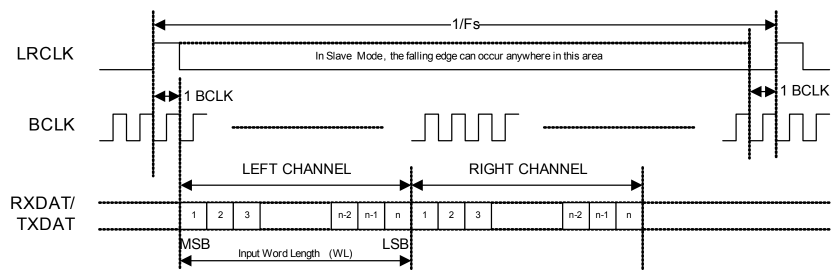


Figure 4-36. DSP Mode A Data Format

DSP Mode B data format is shown in Fig. 4-37.

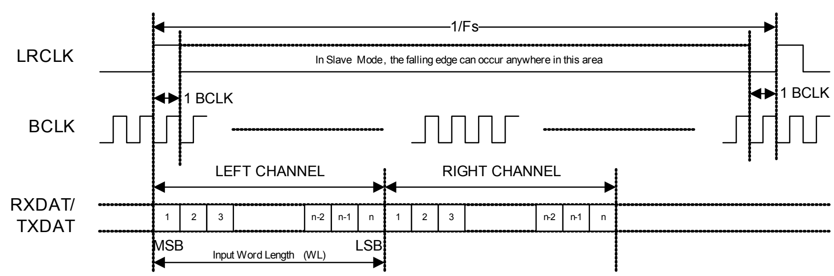


Figure 4-37. DSP Mode B Data Format

- In I2S Mode, the MSB is available on the second rising edge of BCLK following a LRCLK transition. The other bits up to the LSB are then transmitted in order. Depending on word length, BCLK frequency, and sample rate, there may be unused BCLK cycles between the LSB of one sample and the MSB of the next.

  I2S Mode data format is shown in Fig. 4-38.

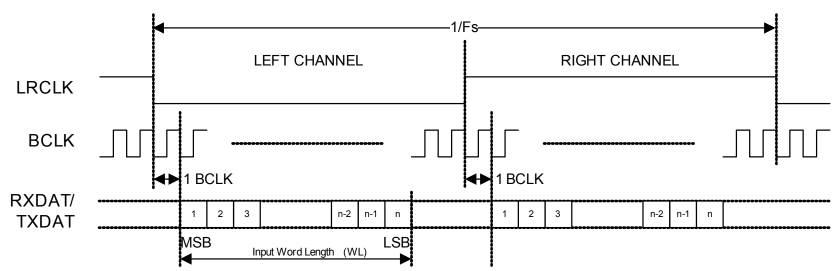


Figure 4-38. I2S Data Format (Assuming n-Bit Word Length)

- In Left-Justified Mode, the MSB is available on the first rising edge of BCLK following a LRCLK transition. The other bits up to the LSB are then transmitted in order. Depending on word length, BCLK frequency, and sample rate, there may be unused BCLK cycles before each LRCLK transition.

  Left-Justified Mode data format is shown in Fig. 4-39.

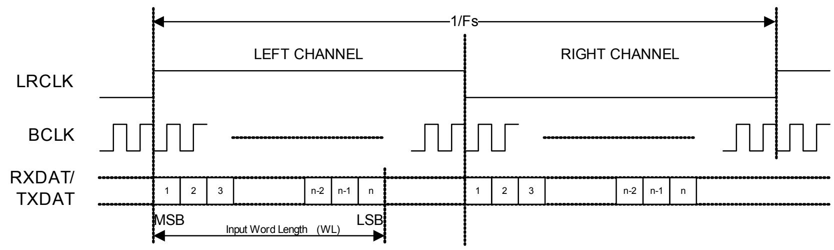


Figure 4-39. Left-Justified Data Format (Assuming n-Bit Word Length)

4.6.3 AIF Time-Slot Configuration

Digital audio interfaces AIF1 and AIF2 support multichannel operation, with up to six channels of input and output on AIF1, and up to four channels on AIF2. A high degree of flexibility is provided to define the position of the audio samples within each LRCLK frame; the audio channel samples may be arranged in any order within the frame.

AIF3 also provides flexible configuration options, but this interface supports only one stereo input and one stereo output path.

Note that, on each interface, all input and output channels must operate at the same sample rate (Fs).

Each of the audio channels can be enabled or disabled independently on the transmit (TX) and receive (RX) signal paths. For each enabled channel, the audio samples are assigned to one time slot within the LRCLK frame.

In DSP modes, the time slots are ordered consecutively from the start of the LRCLK frame. In I^2S and left-justified modes, the even-numbered time slots are arranged in the first half of the LRCLK frame, and the odd-numbered time slots are arranged in the second half of the frame.

The time slots are assigned independently for the transmit (TX) and receive (RX) signal paths. There is no requirement to assign every available time slot to an audio sample; slots may be left unused, if desired. Care is required, however, to ensure that no time slot is allocated to more than one audio channel.

The number of BCLK cycles within a slot is configurable; this is the slot-length. The number of valid data bits within a slot is also configurable; this is the word length. The number of BCLK cycles per LRCLK frame must be configured; it must be ensured that there are enough BCLK cycles within each LRCLK frame to transmit or receive all of the enabled audio channels.

Examples of the AIF time-slot configurations are shown in Fig. 4-40 through Fig. 4-43. One example is shown for each of the four possible data formats.

Fig. 4-40 shows an example of DSP Mode A format. Four enabled audio channels are shown, allocated to time slots 0 through 3.

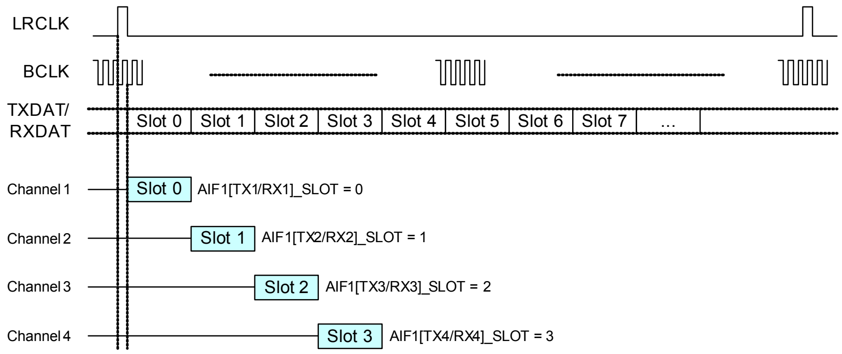


Figure 4-40. DSP Mode A Example

Fig. 4-41 shows an example of DSP Mode B format. Six enabled audio channels are shown, with time slots 4 and 5 unused.

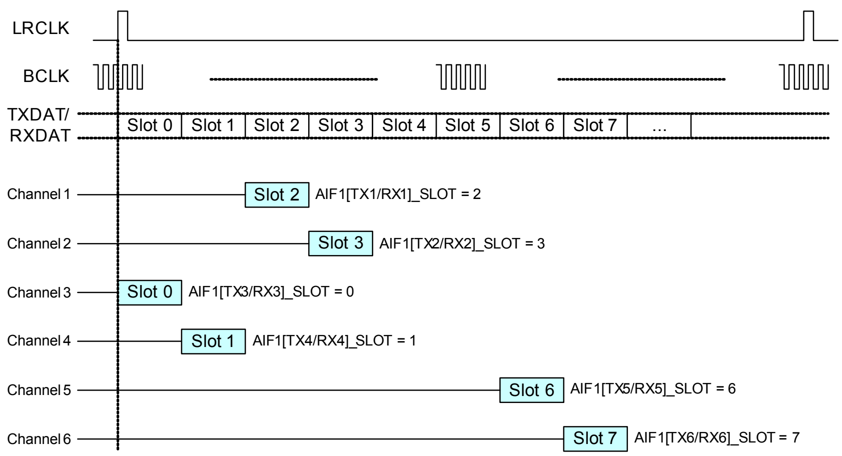


Figure 4-41. DSP Mode B Example

Fig. 4-42 shows an example of I2S format. Four enabled channels are shown, allocated to time slots 0 through 3.

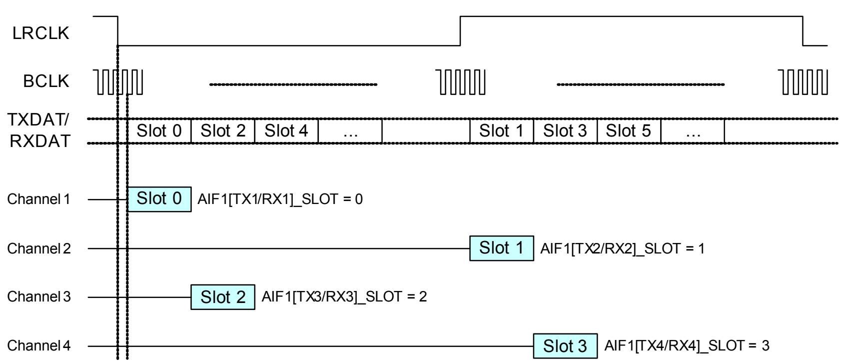


Figure 4-42. I2S Example

Fig. 4-43 shows an example of left-justified format. Six enabled channels are shown.

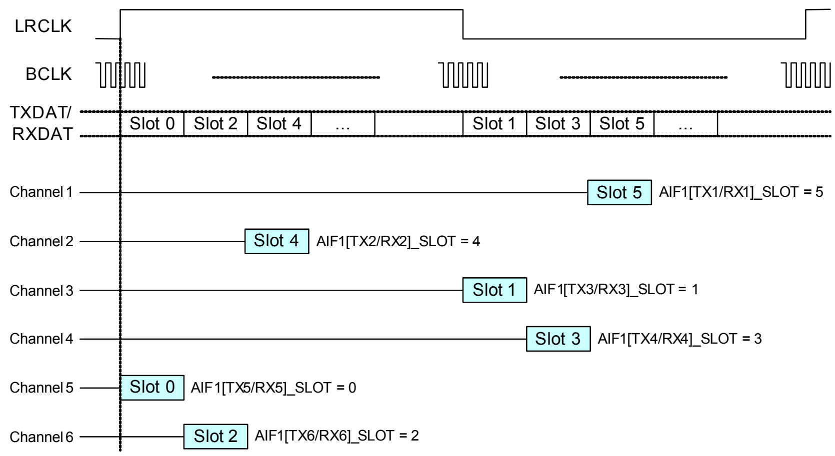


Figure 4-43. Left-Justified Example

4.6.4 TDM Operation Between Three or More Devices

The AIF operation described in Section 4.6.3 illustrates how multiple audio channels can be interleaved on a single TXDAT or RXDAT pin. The interface uses TDM to allocate time periods to each of the audio channels in turn.

This form of TDM is implemented between two devices, using the electrical connections shown Fig. 4-34 or Fig. 4-35.

It is also possible to implement TDM between three or more devices. This allows one codec to receive audio data from two other devices simultaneously on a single audio interface, as shown in Fig. 4-44, Fig. 4-45, and Fig. 4-46.

The CS47L15 provides full support for TDM operation. The TXDAT pin can be tristated when not transmitting data, in order to allow other devices to transmit on the same wire. The behavior of the TXDAT pin is configurable, to allow maximum flexibility to interface with other devices in this way.

Typical configurations of TDM operation between three devices are shown in Fig. 4-44, Fig. 4-45, and Fig. 4-46.

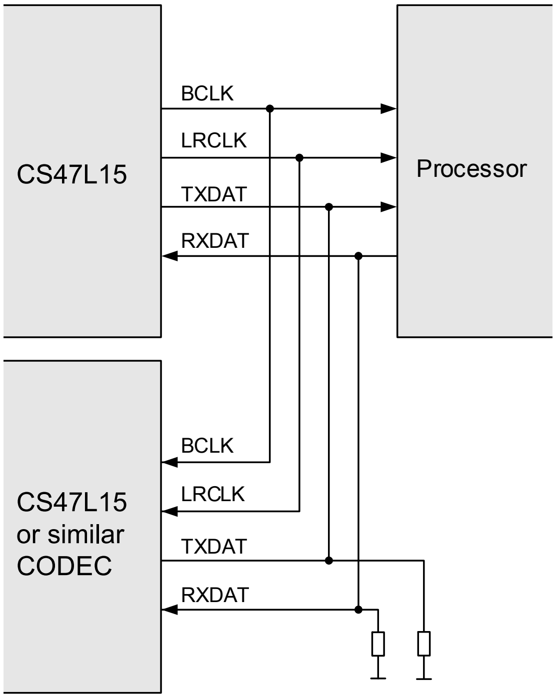


Figure 4-44. TDM with CS47L15 as Master

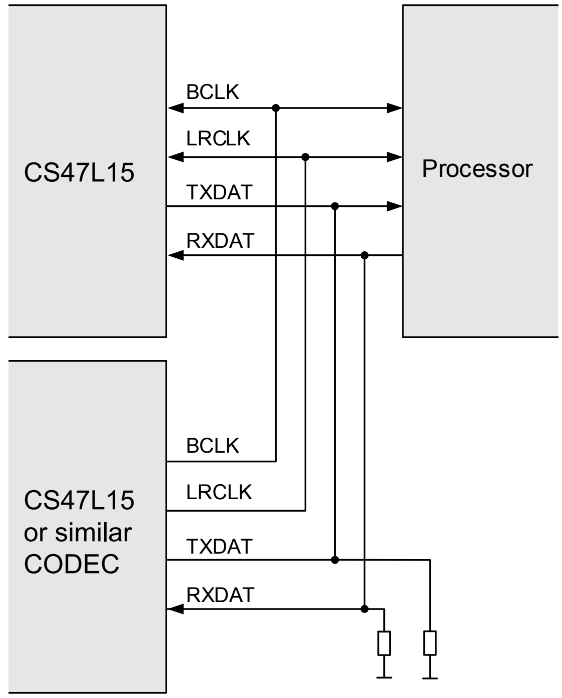


Figure 4-45. TDM with Other Codec as Master

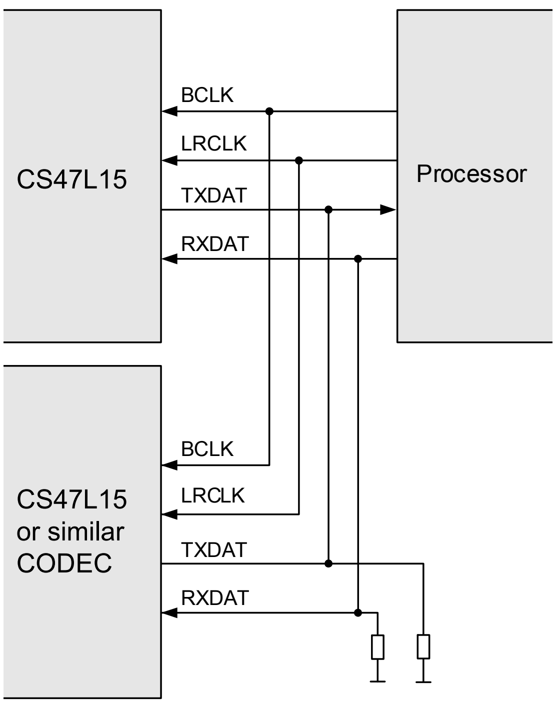


Figure 4-46. TDM with Processor as Master

Note: The CS47L15 is a 24-bit device. If the user operates the CS47L15 in 32-Bit Mode, the 8 LSBs are ignored on the receiving side and not driven on the transmitting side. It is therefore recommended to add a pull-down resistor if necessary to the RXDAT line and the TXDAT line in TDM mode.

4.7 Digital Audio Interface Control

This section describes the configuration of the CS47L15 digital audio interface paths.

AIF1 supports up to six input signal paths and up to six output signal paths; AIF2 supports up to four input signal paths and up to four output signal paths; AIF3 supports up to two channels of input and output signal paths. The digital audio interfaces can be configured as master or slave interfaces; mixed master/slave configurations are also possible.

Each input and output signal path can be independently enabled or disabled. The AIF output (TX) and AIF input (RX) paths use shared BCLK and LRCLK control signals.

The digital audio interface supports flexible data formats, selectable word length, configurable time-slot allocations, and TDM tristate control.

The audio interfaces can be reconfigured while enabled, including changes to the LRCLK frame length and the channel time-slot configurations. Care is required to ensure that any on-the-fly reconfiguration does not cause corruption to the active signal paths. Wherever possible, it is recommended to disable all channels before changing the AIF configuration.

4.7.1 AIF Sample-Rate Control

The AIF RX inputs may be selected as input to the digital mixers or signal-processing functions within the CS47L15 digital core. The AIF TX outputs are derived from the respective output mixers.

The sample rate for each digital audio interface AIF*n* is configured using the respective AIF*n*_RATE field—see Table 4-24.

Note that sample-rate conversion is required when routing the AIF paths to any signal chain that is configured for a different sample rate.

4.7.2 AIF Pin Configuration

The external connections associated with each digital audio interface (AIF) are implemented on multi-function GPIO pins, which must be configured for the respective AIF functions when required. The AIF connections are alternative functions available on specific GPIO pins. See Section 4.11 to configure the GPIO pins for AIF operation.

Integrated pull-up and pull-down resistors can be enabled on the AIF*n*LRCLK, AIF*n*BCLK and AIF*n*RXDAT pins. This is provided as part of the GPIO functionality, and provides a flexible capability for interfacing with other devices. Each of the pull-up and pull-down resistors can be configured independently using the fields described in Table 4-72.

If the pull-up and pull-down resistors are both enabled, the CS47L15 provides a bus keeper function on the respective pin. The bus-keeper function holds the logic level unchanged whenever the pin is undriven (e.g., if the signal is tristated).

4.7.3 AIF Master/Slave Control

The digital audio interfaces can operate in master or slave modes and also in mixed master/slave configurations. In Master Mode, the BCLK and LRCLK signals are generated by the CS47L15 when any of the respective digital audio interface channels is enabled. In Slave Mode, these outputs are disabled by default to allow another device to drive these pins.

Master Mode is selected on the AIF*n*BCLK pin by setting AIF*n*_BCLK_MSTR. In Master Mode, the AIF*n*BCLK signal is generated by the CS47L15 when one or more AIF*n* channels is enabled.

When the AIF*n*_BCLK_FRC bit is set in BCLK Master Mode, the AIF*n*BCLK signal is output at all times, including when none of the AIF*n* channels is enabled.

The AIF*n*BCLK signal can be inverted in master or slave modes using the AIF*n*_BCLK_INV bit.

Master Mode is selected on the AIF*n*LRCLK pin by setting AIF*n*_LRCLK_MSTR. In Master Mode, the AIF*n*LRCLK signal is generated by the CS47L15 when one or more AIF*n* channels is enabled.

When AIF*n*_LRCLK_FRC is set in LRCLK Master Mode, the AIF*n*LRCLK signal is output at all times, including when none of the AIF*n* channels is enabled. Note that AIF*n*LRCLK is derived from AIF*n*BCLK, and an internal or external AIF*n*BCLK signal must be present to generate AIF*n*LRCLK.

The AIF*n*LRCLK signal can be inverted in master or slave modes using the AIF*n*_LRCLK_INV bit.

The timing of the AIF*n*LRCLK signal is selectable using AIF*n*_LRCLK_ADV. If this bit is set, the LRCLK signal transition is advanced to the previous BCLK phase (as compared with the default behavior). Further details of this option, and conditions for valid use cases, are described in Section 4.7.3.1.

The AIF1 master/slave control registers are described in Table 4-38.

Table 4-38. AIF1 Master/Slave Control

Register Address	Bit	Label	Default	Description
R1280 (0x0500) AIF1_BCLK_Ctrl	7	AIF1_BCLK_INV	0	AIF1 Audio Interface BCLK Invert 0 = AIF1BCLK not inverted 1 = AIF1BCLK inverted
	6	AIF1_BCLK_FRC	0	AIF1 Audio Interface BCLK Output Control 0 = Normal 1 = AIF1BCLK always enabled in Master Mode
	5	AIF1_BCLK_MSTR	0	AIF1 Audio Interface BCLK Master Select 0 = AIF1BCLK Slave Mode 1 = AIF1BCLK Master Mode
R1282 (0x0502) AIF1_Rx_Pin_Ctrl	4	AIF1_LRCLK_ADV	0	AIF1 Audio Interface LRCLK Advance 0 = Normal 1 = AIF1LRCLK transition is advanced to the previous BCLK phase
	2	AIF1_LRCLK_INV	0	AIF1 Audio Interface LRCLK Invert 0 = AIF1LRCLK not inverted 1 = AIF1LRCLK inverted
	1	AIF1_LRCLK_FRC	0	AIF1 Audio Interface LRCLK Output Control 0 = Normal 1 = AIF1LRCLK always enabled in Master Mode
	0	AIF1_LRCLK_MSTR	0	AIF1 Audio Interface LRCLK Master Select 0 = AIF1LRCLK Slave Mode 1 = AIF1LRCLK Master Mode

The AIF2 master/slave control registers are described in Table 4-39.

Table 4-39. AIF2 Master/Slave Control

Register Address	Bit	Label	Default	Description
R1344 (0x0540) AIF2_BCLK_Ctrl	7	AIF2_BCLK_INV	0	AIF2 Audio Interface BCLK Invert 0 = AIF2BCLK not inverted 1 = AIF2BCLK inverted
	6	AIF2_BCLK_FRC	0	AIF2 Audio Interface BCLK Output Control 0 = Normal 1 = AIF2BCLK always enabled in Master Mode
	5	AIF2_BCLK_MSTR	0	AIF2 Audio Interface BCLK Master Select 0 = AIF2BCLK Slave Mode 1 = AIF2BCLK Master Mode
R1346 (0x0542) AIF2_Rx_Pin_Ctrl	4	AIF2_LRCLK_ADV	0	AIF2 Audio Interface LRCLK Advance 0 = Normal 1 = AIF2LRCLK transition is advanced to the previous BCLK phase
	2	AIF2_LRCLK_INV	0	AIF2 Audio Interface LRCLK Invert 0 = AIF2LRCLK not inverted 1 = AIF2LRCLK inverted
	1	AIF2_LRCLK_FRC	0	AIF2 Audio Interface LRCLK Output Control 0 = Normal 1 = AIF2LRCLK always enabled in Master Mode
	0	AIF2_LRCLK_MSTR	0	AIF2 Audio Interface LRCLK Master Select 0 = AIF2LRCLK Slave Mode 1 = AIF2LRCLK Master Mode

The AIF3 master/slave control registers are described in Table 4-40.

Table 4-40. AIF3 Master/Slave Control

Register Address	Bit	Label	Default	Description
R1408 (0x0580) AIF3_BCLK_Ctrl	7	AIF3_BCLK_INV	0	AIF3 Audio Interface BCLK Invert 0 = AIF3BCLK not inverted 1 = AIF3BCLK inverted
	6	AIF3_BCLK_FRC	0	AIF3 Audio Interface BCLK Output Control 0 = Normal 1 = AIF3BCLK always enabled in Master Mode
	5	AIF3_BCLK_MSTR	0	AIF3 Audio Interface BCLK Master Select 0 = AIF3BCLK Slave Mode 1 = AIF3BCLK Master Mode
R1410 (0x0582) AIF3_Rx_Pin_Ctrl	4	AIF3_LRCLK_ADV	0	AIF3 Audio Interface LRCLK Advance 0 = Normal 1 = AIF3LRCLK transition is advanced to the previous BCLK phase
	2	AIF3_LRCLK_INV	0	AIF3 Audio Interface LRCLK Invert 0 = AIF3LRCLK not inverted 1 = AIF3LRCLK inverted
	1	AIF3_LRCLK_FRC	0	AIF3 Audio Interface LRCLK Output Control 0 = Normal 1 = AIF3LRCLK always enabled in Master Mode
	0	AIF3_LRCLK_MSTR	0	AIF3 Audio Interface LRCLK Master Select 0 = AIF3LRCLK Slave Mode 1 = AIF3LRCLK Master Mode

4.7.3.1 LRCLK Advance

The timing of the AIF*n*LRCLK signal can be adjusted using AIF*n*_LRCLK_ADV. If this bit is set, the LRCLK signal transition is advanced to the previous BCLK phase (as compared with the default behavior).

The LRCLK-advance option (AIF*n*_LRCLK_ADV = 1) is valid for DSP-A mode only, operating in Master Mode.

Note: BCLK inversion must be enabled (AIF*n*_BCLK_INV = 1) if the LRCLK-advance option is enabled.

The adjusted interface timing (AIF*n*_LRCLK_ADV = 1), is shown in Fig. 4-47. The left-channel MSB is available on the second rising edge of BCLK, 1.5 BCLK cycles after the LRCLK rising edge—assuming the BCLK output is inverted.

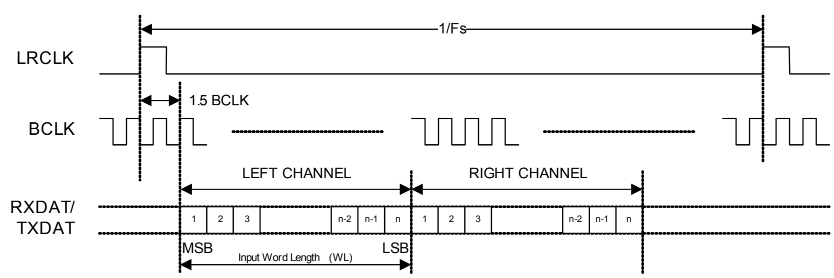


Figure 4-47. LRCLK advance—DSP-A Master Mode

4.7.4 AIF Signal Path Enable

The AIF1 interface supports up to six input (RX) channels and up to six output (TX) channels. Each channel is enabled or disabled using the bits defined in Table 4-41.

The AIF2 interface supports up to four input (RX) channels and up to four output (TX) channels. Each channel is enabled or disabled using the bits defined in Table 4-42.

The AIF3 interface supports up to two input (RX) channels and up to two output (TX) channels. Each channel is enabled or disabled using the bits defined in Table 4-43.

The system clock, SYSCLK, must be configured and enabled before any audio path is enabled. See Section 4.13 for details of the system clocks.

The audio interfaces can be reconfigured if enabled, including changes to the LRCLK frame length and the channel time-slot configurations. Care is required to ensure that this on-the-fly reconfiguration does not cause corruption to the active signal paths. Wherever possible, it is recommended to disable all channels before changing the AIF configuration.

The CS47L15 performs automatic checks to confirm that the SYSCLK frequency is high enough to support the commanded signal paths and processing functions. If the frequency is too low, an attempt to enable an AIF signal path fails. Note that active signal paths are not affected under such circumstances.

The AIF1 signal-path-enable bits are described in Table 4-41.

Table 4-41. AIF1 Signal Path Enable

Register Address	Bit	Label	Default	Description
R1305 (0x0519) AIF1_Tx_Enables	5	AIF1TX6_ENA	0	AIF1 Audio Interface TX Channel 6 Enable 0 = Disabled 1 = Enabled
	4	AIF1TX5_ENA	0	AIF1 Audio Interface TX Channel 5 Enable 0 = Disabled 1 = Enabled
	3	AIF1TX4_ENA	0	AIF1 Audio Interface TX Channel 4 Enable 0 = Disabled 1 = Enabled
	2	AIF1TX3_ENA	0	AIF1 Audio Interface TX Channel 3 Enable 0 = Disabled 1 = Enabled
	1	AIF1TX2_ENA	0	AIF1 Audio Interface TX Channel 2 Enable 0 = Disabled 1 = Enabled
	0	AIF1TX1_ENA	0	AIF1 Audio Interface TX Channel 1 Enable 0 = Disabled 1 = Enabled
R1306 (0x051A) AIF1_Rx_Enables	5	AIF1RX6_ENA	0	AIF1 Audio Interface RX Channel 6 Enable 0 = Disabled 1 = Enabled
	4	AIF1RX5_ENA	0	AIF1 Audio Interface RX Channel 5 Enable 0 = Disabled 1 = Enabled
	3	AIF1RX4_ENA	0	AIF1 Audio Interface RX Channel 4 Enable 0 = Disabled 1 = Enabled
	2	AIF1RX3_ENA	0	AIF1 Audio Interface RX Channel 3 Enable 0 = Disabled 1 = Enabled
	1	AIF1RX2_ENA	0	AIF1 Audio Interface RX Channel 2 Enable 0 = Disabled 1 = Enabled
	0	AIF1RX1_ENA	0	AIF1 Audio Interface RX Channel 1 Enable 0 = Disabled 1 = Enabled

The AIF2 signal-path-enable bits are described in Table 4-42.

Table 4-42. AIF2 Signal Path Enable

Register Address	Bit	Label	Default	Description
R1369 (0x0559) AIF2_Tx_Enables	3	AIF2TX4_ENA	0	AIF2 Audio Interface TX Channel 4 Enable 0 = Disabled 1 = Enabled
	2	AIF2RX3_ENA	0	AIF2 Audio Interface RX Channel 3 Enable 0 = Disabled 1 = Enabled
	1	AIF2RX2_ENA	0	AIF2 Audio Interface RX Channel 2 Enable 0 = Disabled 1 = Enabled
	0	AIF2TX1_ENA	0	AIF2 Audio Interface TX Channel 1 Enable 0 = Disabled 1 = Enabled
R1370 (0x055A) AIF2_Rx_Enables	3	AIF2RX4_ENA	0	AIF2 Audio Interface RX Channel 4 Enable 0 = Disabled 1 = Enabled
	2	AIF2RX3_ENA	0	AIF2 Audio Interface RX Channel 3 Enable 0 = Disabled 1 = Enabled
	1	AIF2RX2_ENA	0	AIF2 Audio Interface RX Channel 2 Enable 0 = Disabled 1 = Enabled
	0	AIF2RX1_ENA	0	AIF2 Audio Interface RX Channel 1 Enable 0 = Disabled 1 = Enabled

The AIF3 signal-path-enable bits are described in Table 4-43.

Table 4-43. AIF3 Signal Path Enable

Register Address	Bit	Label	Default	Description
R1433 (0x0599) AIF3_Tx_Enables	1	AIF3TX2_ENA	0	AIF3 Audio Interface TX Channel 2 Enable 0 = Disabled 1 = Enabled
	0	AIF3TX1_ENA	0	AIF3 Audio Interface TX Channel 1 Enable 0 = Disabled 1 = Enabled
R1434 (0x059A) AIF3_Rx_Enables	1	AIF3RX2_ENA	0	AIF3 Audio Interface RX Channel 2 Enable 0 = Disabled 1 = Enabled
	0	AIF3RX1_ENA	0	AIF3 Audio Interface RX Channel 1 Enable 0 = Disabled 1 = Enabled

4.7.5 AIF BCLK and LRCLK Control

The AIF*n*BCLK frequency is selected using the AIF*n*_BCLK_FREQ field. For each setting of this field, the actual frequency depends on whether AIF*n* is configured for a 48-kHz-related sample rate (SAMPLE_RATE_*n* = 01XXX or 10XXX) or a 44.1kHz-related sample rate (SAMPLE_RATE_*n* = 10XXX), as described in Table 4-44 through Table 4-46.

The selected AIF*n*BCLK rate must be less than or equal to SYSCLK/2. See Section 4.13 for details of SYSCLK clock domain, and the associated control registers.

The AIF*n*LRCLK frequency is controlled relative to AIF*n*BCLK by the AIF*n*_BCPF divider.

Note that the BCLK rate must be configured in master or slave modes, using the AIF*n*_BCLK_FREQ fields. The LRCLK rates only require to be configured in Master Mode.

The AIF1 BCLK/LRCLK control fields are described in Table 4-44.

Table 4-44. AIF1 BCLK and LRCLK Control

Register Address	Bit	Label	Default	Description
R1280 (0x0500) AIF1_ BCLK_Ctrl	4:0	AIF1_BCLK_ FREQ[4:0]	0x0C	AIF1BCLK Rate. The AIF1BCLK rate must be less than or equal to SYSCLK/2. 0x00–0x01 = Reserved 0x02 = 64 kHz (58.8 kHz) 0x03 = 96 kHz (88.2 kHz) 0x04 = 128 kHz (117.6 kHz) 0x05 = 192 kHz (176.4 kHz) 0x06 = 256 kHz (235.2 kHz) 0x07 = 384 kHz (352.8 kHz) 0x08 = 512 kHz (470.4 kHz) 0x09 = 768 kHz (705.6 kHz) 0x0A = 1.024 MHz (940.8 kHz) 0x0B = 1.536 MHz (1.4112 MHz) 0x0C = 2.048 MHz (1.8816 MHz) 0x0D = 3.072 MHz (2.8824 MHz) 0x0E = 4.096 MHz (3.7632 MHz) 0x0F = 6.144 MHz (5.6448 MHz) 0x10 = 8.192 MHz (7.5264 MHz) 0x11 = 12.288 MHz (11.2896 MHz) 0x12 = 24.576 MHz (22.5792 MHz) The frequencies in brackets apply for 44.1 kHz–related sample rates only (SAMPLE_RATE_ *n* = 01XXX).
R1286 (0x0506) AIF1_Rx_ BCLK_Rate	12:0	AIF1_ BCPF[12:0]	0x0040	AIF1LRCLK Rate. Selects the number of BCLK cycles per AIF1LRCLK frame. AIF1LRCLK clock = AIF1BCLK/AIF1_BCPF. Integer (LSB = 1), Valid from 8 to 8191.

The AIF2 BCLK/LRCLK control fields are described in Table 4-45.

Table 4-45. AIF2 BCLK and LRCLK Control

Register Address	Bit	Label	Default	Description
R1344 (0x0540) AIF2_ BCLK_Ctrl	4:0	AIF2_BCLK_ FREQ[4:0]	0x0C	AIF2BCLK Rate. The AIF2BCLK rate must be less than or equal to SYSCLK/2. 0x00–0x01 = Reserved 0x02 = 64 kHz (58.8 kHz) 0x03 = 96 kHz (88.2 kHz) 0x04 = 128 kHz (117.6 kHz) 0x05 = 192 kHz (176.4 kHz) 0x06 = 256 kHz (235.2 kHz) 0x07 = 384 kHz (352.8 kHz) 0x08 = 512 kHz (470.4 kHz) 0x09 = 768 kHz (705.6 kHz) 0x0A = 1.024 MHz (940.8 kHz) 0x0B = 1.536 MHz (1.4112 MHz) 0x0C = 2.048 MHz (1.8816 MHz) 0x0D = 3.072 MHz (2.8824 MHz) 0x0E = 4.096 MHz (3.7632 MHz) 0x0F = 6.144 MHz (5.6448 MHz) 0x10 = 8.192 MHz (7.5264 MHz) 0x11 = 12.288 MHz (11.2896 MHz) 0x12 = 24.576 MHz (22.5792 MHz) The frequencies in brackets apply for 44.1 kHz–related sample rates only (SAMPLE_RATE_ *n* = 01XXX).
R1350 (0x0546) AIF2_Rx_ BCLK_Rate	12:0	AIF2_ BCPF[12:0]	0x0040	AIF2LRCLK Rate. Selects the number of BCLK cycles per AIF2LRCLK frame. AIF2LRCLK clock = AIF2BCLK/AIF2_BCPF. Integer (LSB = 1), Valid from 8 to 8191.

The AIF3 BCLK/LRCLK control fields are described in Table 4-46.

Table 4-46. AIF3 BCLK and LRCLK Control

Register Address	Bit	Label	Default	Description
R1408 (0x0580) AIF3_ BCLK_Ctrl	4:0	AIF3_ BCLK_ FREQ[4:0]	0x0C	AIF3BCLK Rate. The AIF3BCLK rate must be less than or equal to SYSCLK/2. 0x00–0x01 = Reserved 0x02 = 64 kHz (58.8 kHz) 0x03 = 96 kHz (88.2 kHz) 0x04 = 128 kHz (117.6 kHz) 0x05 = 192 kHz (176.4 kHz) 0x06 = 256 kHz (235.2 kHz) 0x07 = 384 kHz (352.8 kHz) 0x08 = 512 kHz (470.4 kHz) 0x09 = 768 kHz (705.6 kHz) 0x0A = 1.024 MHz (940.8 kHz) 0x0B = 1.536 MHz (1.4112 MHz) 0x0C = 2.048 MHz (1.8816 MHz) 0x0D = 3.072 MHz (2.8824 MHz) 0x0E = 4.096 MHz (3.7632 MHz) 0x0F = 6.144 MHz (5.6448 MHz) 0x10 = 8.192 MHz (7.5264 MHz) 0x11 = 12.288 MHz (11.2896 MHz) 0x12 = 24.576 MHz (22.5792 MHz) The frequencies in brackets apply for 44.1 kHz–related sample rates only (SAMPLE_RATE_ *n* = 01XXX).
R1414 (0x0586) AIF3_Rx_ BCLK_Rate	12:0	AIF3_ BCPF[12:0]	0x0040	AIF3LRCLK Rate. Selects the number of BCLK cycles per AIF3LRCLK frame. AIF3LRCLK clock = AIF3BCLK/AIF3_BCPF. Integer (LSB = 1), Valid from 8 to 8191.

4.7.6 AIF Digital Audio Data Control

The fields controlling the audio data format, word length, and slot configurations for AIF1, AIF2, and AIF3 are described in Table 4-47, Table 4-48, and Table 4-49 respectively.

Note that left-justified and DSP-B modes are valid in Master Mode only (i.e., BCLK and LRCLK are outputs from the CS47L15).

The AIF*n* slot length is the number of BCLK cycles in one time slot within the overall LRCLK frame. The word length is the number of valid data bits within each time slot. If the word length is less than the slot length, there are unused BCLK cycles at the end of each time slot. The AIF*n* word length and slot length is independently selectable for the input (RX) and output (TX) paths.

For each AIF input (RX) and AIF output (TX) channel, the position of the audio data sample within the LRCLK frame is configurable. The x_SLOT fields define the time-slot position of the audio sample for the associated audio channel. Valid selections are Slot 0 upwards. The time slots are numbered as shown in Fig. 4-40 through Fig. 4-43.

Note that, in DSP modes, the time slots are ordered consecutively from the start of the LRCLK frame. In I^2S and left-justified modes, the even-numbered time slots are arranged in the first half of the LRCLK frame, and the odd-numbered time slots are arranged in the second half of the frame.

The AIF1 data control fields are described in Table 4-47.

Table 4-47. AIF1 Digital Audio Data Control

Register Address	Bit	Label	Default	Description
R1284 (0x0504) AIF1_Format	2:0	AIF1_FMT[2:0]	000	AIF1 Audio Interface Format 000 = DSP Mode A 001 = DSP Mode B 010 = I^2S mode 011 = Left-Justified mode Other codes are reserved.
R1287 (0x0507) AIF1_Frame_Ctrl_1	13:8	AIF1TX_WL[5:0]	0x18	AIF1 TX Word Length (Number of valid data bits per slot) Integer (LSB = 1); Valid from 16 to 32
	7:0	AIF1TX_SLOT_LEN[7:0]	0x18	AIF1 TX Slot Length (Number of BCLK cycles per slot) Integer (LSB = 1); Valid from 16 to 128
R1288 (0x0508) AIF1_Frame_Ctrl_2	13:8	AIF1RX_WL[5:0]	0x18	AIF1 RX Word Length (Number of valid data bits per slot) Integer (LSB = 1); Valid from 16 to 32
	7:0	AIF1RX_SLOT_LEN[7:0]	0x18	AIF1 RX Slot Length (Number of BCLK cycles per slot) Integer (LSB = 1); Valid from 16 to 128
R1289 (0x0509) to R1294 (0x050E)	5:0	AIF1TX1_SLOT[5:0]	0x0	AIF1 TX Channel n Slot position Defines the TX time slot position of the Channel n audio sample Integer (LSB=1); Valid from 0 to 63
	5:0	AIF1TX2_SLOT[5:0]	0x1	
	5:0	AIF1TX3_SLOT[5:0]	0x2	
	5:0	AIF1TX4_SLOT[5:0]	0x3	
	5:0	AIF1TX5_SLOT[5:0]	0x4	
	5:0	AIF1TX6_SLOT[5:0]	0x5	
R1297 (0x0511) to R1302 (0x0516)	5:0	AIF1RX1_SLOT[5:0]	0x0	AIF1 RX Channel n Slot position Defines the RX time slot position of the Channel n audio sample Integer (LSB=1); Valid from 0 to 63
	5:0	AIF1RX2_SLOT[5:0]	0x1	
	5:0	AIF1RX3_SLOT[5:0]	0x2	
	5:0	AIF1RX4_SLOT[5:0]	0x3	
	5:0	AIF1RX5_SLOT[5:0]	0x4	
	5:0	AIF1RX6_SLOT[5:0]	0x5	

The AIF2 data control fields are described in Table 4-48.

Table 4-48. AIF2 Digital Audio Data Control

Register Address	Bit	Label	Default	Description
R1348 (0x0544) AIF2_Format	2:0	AIF2_FMT[2:0]	000	AIF2 Audio Interface Format 000 = DSP Mode A 001 = DSP Mode B 010 = I2S mode 011 = Left-Justified mode Other codes are reserved.
R1351 (0x0547) AIF2_Frame_Ctrl_1	13:8	AIF2TX_WL[5:0]	0x18	AIF2 TX Word Length (Number of valid data bits per slot) Integer (LSB = 1); Valid from 16 to 32
	7:0	AIF2TX_SLOT_LEN[7:0]	0x18	AIF2 TX Slot Length (Number of BCLK cycles per slot) Integer (LSB = 1); Valid from 16 to 128
R1352 (0x0548) AIF2_Frame_Ctrl_2	13:8	AIF2RX_WL[5:0]	0x18	AIF2 RX Word Length (Number of valid data bits per slot) Integer (LSB = 1); Valid from 16 to 32
	7:0	AIF2RX_SLOT_LEN[7:0]	0x18	AIF2 RX Slot Length (Number of BCLK cycles per slot) Integer (LSB = 1); Valid from 16 to 128
R1353 (0x0549) to R1356 (0x054C)	5:0	AIF2TX1_SLOT[5:0]	0x0	AIF2 TX Channel n Slot position Defines the TX time slot position of the Channel n audio sample Integer (LSB=1); Valid from 0 to 63
	5:0	AIF2TX2_SLOT[5:0]	0x1	
	5:0	AIF2TX3_SLOT[5:0]	0x2	
	5:0	AIF2TX4_SLOT[5:0]	0x3	
R1361 (0x0551) to R1364 (0x0554)	5:0	AIF2RX1_SLOT[5:0]	0x0	AIF2 RX Channel n Slot position Defines the RX time slot position of the Channel n audio sample Integer (LSB=1); Valid from 0 to 63
	5:0	AIF2RX2_SLOT[5:0]	0x1	
	5:0	AIF2RX3_SLOT[5:0]	0x2	
	5:0	AIF2RX4_SLOT[5:0]	0x3	

The AIF3 data control fields are described in Table 4-49.

Table 4-49. AIF3 Digital Audio Data Control

Register Address	Bit	Label	Default	Description
R1412 (0x0584) AIF3_Format	2:0	AIF3_FMT[2:0]	000	AIF3 Audio Interface Format 000 = DSP Mode A 001 = DSP Mode B 010 = I2S mode 011 = Left-Justified mode Other codes are reserved.
R1415 (0x0587) AIF3_Frame_Ctrl_1	13:8	AIF3TX_WL[5:0]	0x18	AIF3 TX Word Length (Number of valid data bits per slot) Integer (LSB = 1); Valid from 16 to 32
	7:0	AIF3TX_SLOT_LEN[7:0]	0x18	AIF3 TX Slot Length (Number of BCLK cycles per slot) Integer (LSB = 1); Valid from 16 to 128
R1416 (0x0588) AIF3_Frame_Ctrl_2	13:8	AIF3RX_WL[5:0]	0x18	AIF3 RX Word Length (Number of valid data bits per slot) Integer (LSB = 1); Valid from 16 to 32
	7:0	AIF3RX_SLOT_LEN[7:0]	0x18	AIF3 RX Slot Length (Number of BCLK cycles per slot) Integer (LSB = 1); Valid from 16 to 128

Table 4-49. AIF3 Digital Audio Data Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R1417 (0x0589) AIF3_Frame_Ctrl_3	5:0	AIF3TX1_SLOT[5:0]	0x0	AIF3 TX Channel 1 Slot position Defines the TX time slot position of the Channel 1 audio sample Integer (LSB=1); Valid from 0 to 63
R1418 (0x058A) AIF3_Frame_Ctrl_4	5:0	AIF3TX2_SLOT[5:0]	0x1	AIF3 TX Channel 2 Slot position Defines the TX time slot position of the Channel 2 audio sample Integer (LSB=1); Valid from 0 to 63
R1425 (0x0591) AIF3_Frame_Ctrl_11	5:0	AIF3RX1_SLOT[5:0]	0x0	AIF3 RX Channel 1 Slot position Defines the RX time slot position of the Channel 1 audio sample Integer (LSB=1); Valid from 0 to 63
R1426 (0x0592) AIF3_Frame_Ctrl_12	5:0	AIF3RX2_SLOT[5:0]	0x1	AIF3 RX Channel 2 Slot position Defines the RX time slot position of the Channel 2 audio sample Integer (LSB=1); Valid from 0 to 63

4.7.7 AIF TDM and Tristate Control

The AIF*n* output pins are tristated when the AIF*n*_TRI bit is set. Note that this function only affects output pins configured for the respective AIF*n* function—a GPIO pin that is configured for a different function is not affected by AIF*n*_TRI. See Section 4.11 to configure the GPIO pins.

Under default conditions, the AIF*n*TXDAT output is held at Logic 0 when the CS47L15 is not transmitting data (i.e., during time slots that are not enabled for output by the CS47L15). If the AIF*n*TX_DAT_TRI bit is set, the CS47L15 tristates the respective AIF*n*TXDAT pin when not transmitting data, allowing other devices to drive the AIF*n*TXDAT connection.

The AIF1 TDM and tristate control fields are described in Table 4-50.

Table 4-50. AIF1 TDM and Tristate Control

Register Address	Bit	Label	Default	Description
R1281 (0x0501) AIF1_Tx_Pin_Ctrl	5	AIF1TX_DAT_TRI	0	AIF1TXDAT Tristate Control 0 = Logic 0 during unused time slots 1 = Tristated during unused time slots
R1283 (0x0503) AIF1_Rate_Ctrl	6	AIF1_TRI	0	AIF1 Audio Interface Tristate Control 0 = Normal 1 = AIF1 Outputs are tristated Note that this bit only affects output pins configured for the respective AIF1 function.

The AIF2 TDM and tristate control fields are described in Table 4-51.

Table 4-51. AIF2 TDM and Tristate Control

Register Address	Bit	Label	Default	Description
R1345 (0x0541) AIF2_Tx_Pin_Ctrl	5	AIF2TX_DAT_TRI	0	AIF2TXDAT Tristate Control 0 = Logic 0 during unused time slots 1 = Tristated during unused time slots
R1347 (0x0543) AIF2_Rate_Ctrl	6	AIF2_TRI	0	AIF2 Audio Interface Tristate Control 0 = Normal 1 = AIF2 Outputs are tristated Note that this bit only affects output pins configured for the respective AIF2 function.

The AIF3 TDM and tristate control fields are described in Table 4-52.

Table 4-52. AIF3 TDM and Tristate Control

Register Address	Bit	Label	Default	Description
R1409 (0x0581) AIF3_Tx_Pin_Ctrl	5	AIF3TX_DAT_TRI	0	AIF3TXDAT Tristate Control 0 = Logic 0 during unused time slots 1 = Tristated during unused time slots
R1411 (0x0583) AIF3_Rate_Ctrl	6	AIF3_TRI	0	AIF3 Audio Interface Tristate Control 0 = Normal 1 = AIF3 Outputs are tristated Note that this bit only affects output pins configured for the respective AIF3 function.

4.8 Output Signal Path

The CS47L15 provides three audio output signal paths. These outputs comprise ground-referenced headphone/earpiece drivers, differential speaker driver, and a digital output interface suitable for external speaker drivers. The output signal paths are summarized in Table 4-53.

Table 4-53. Output Signal Path Summary

Signal Path	Descriptions	Output Pins
OUT1L, OUT1R	Ground-referenced headphone/earpiece output	HPOUTL, HPOUTR or EPOUTP, EPOUTN
OUT4L	Differential speaker output	SPKOUTN, SPKOUTP
OUT5L, OUT5R	Digital speaker (PDM) output	SPKTXDAT, SPKCLK

The analog output paths incorporate high performance 24-bit sigma-delta DACs.

The headphone/earpiece output path is configurable as a stereo headphone driver (HPOUTL and HPOUTR pins), or as a differential earpiece driver (EPOUTP and EPOUTN pins). The ground-referenced headphone output path incorporates a common mode feedback path for rejection of system-related noise. The headphone and earpiece outputs each support direct connection to external loads, with no requirement for AC coupling capacitors.

The speaker output path is configured to drive a differential (BTL) output. The Class D design offers high efficiency at large signal levels. With a suitable choice of external speaker, the Class D output can drive a loudspeaker directly, without any additional filter components.

The digital output path provides a stereo pulse-density modulation (PDM) output interface, for connection to external audio devices. The PDM interface supports two digital output channels. The CS47L15 also supports a two-channel digital input path that is synchronized to the PDM interface; the two-way interface can be used to support digital feedback from a PDM speaker driver, enabling advanced speaker protection algorithms to be implemented.

Digital volume control is available on all outputs (analog and digital), with programmable ramp control for smooth, glitch-free operation. A configurable noise-gate function is available on each of the output signal paths. Any two of the output signal paths may be selected as input to the AEC loop-back paths.

The CS47L15 incorporates thermal protection functions, and provides short-circuit detection on the Class D speaker and headphone/earpiece output paths. The general-purpose timers (see Section 4.5.2) can also be used as a watchdog function, to trigger a shutdown of the Class D speaker drivers; see Section 4.18.

The Class D speaker output is designed to support monitoring of external loudspeakers, giving real-time feedback for algorithms such as Cirrus Logic's speaker-protection software, running on the DSP core. This enables loudspeakers to be protected against damage from excessive signal levels and other electro-mechanical constraints. This feature requires additional external component connections, as described in Section 4.8.8.

The CS47L15 output signal paths are shown in Fig. 4-48.

The OUT2, OUT3, and OUT4R paths are not implemented on this device.

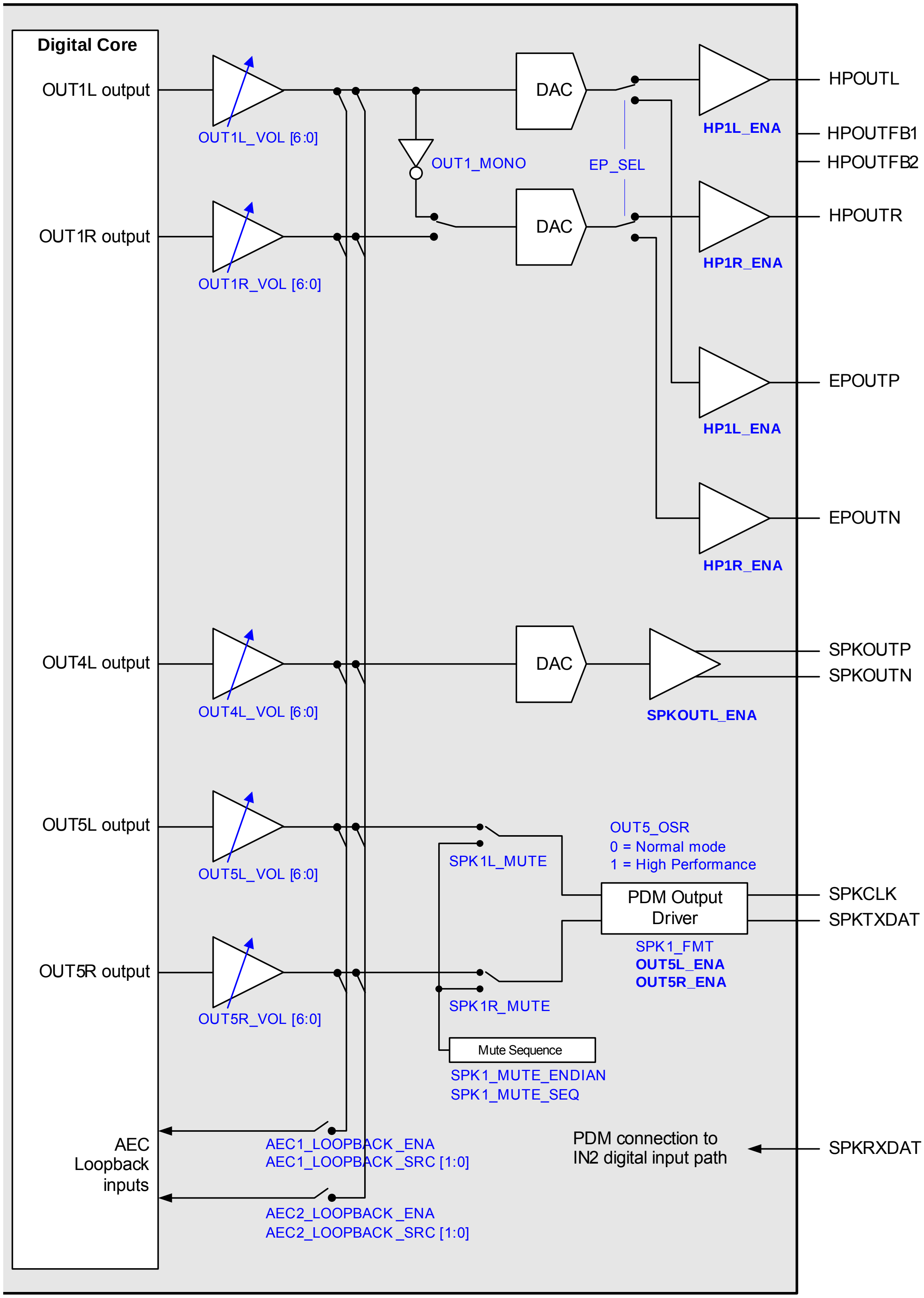


Figure 4-48. Output Signal Paths

4.8.1 Output Signal Path Enable

The output signal paths are enabled using the bits described in Table 4-54. The respective bits must be enabled for analog or digital output on the respective output paths.

The OUT1 path is associated with the headphone and the earpiece output drivers. The HP1L_ENA and HP1R_ENA bits control either the HPOUT or EPOUT drivers, depending on the EP_SEL register bit selection. See Table 4-56 for details of the EP_SEL register.

The output signal paths are muted by default. It is recommended that deselecting the mute should be the final step of the path enable control sequence. Similarly, the mute should be selected as the first step of the path disable control sequence. The output signal path mute functions are controlled using the bits described in Table 4-54.

The supply rails for the OUT1 outputs (HPOUT and EPOUT) are generated using an integrated dual-mode charge pump. The charge pump is enabled automatically by the CS47L15 when required by the output drivers; see Section 4.16.

The CS47L15 schedules a pop-suppressed control sequence to enable or disable the OUT1 and OUT4L signal paths. This is automatically managed by the control-write sequencer in response to setting the respective HP*nx*_ENA or SPKOUTL_ENA bits; see Section 4.15 for further details.

The output signal path enable/disable control sequences are inputs to the interrupt circuit and can be used to trigger an interrupt event when a sequence completes; see Section 4.12.

The system clock, SYSCLK, must be configured and enabled before any audio path is enabled. See Section 4.13 for details of the system clocks.

The CS47L15 performs automatic checks to confirm that the SYSCLK frequency is high enough to support the output signal paths and associated DACs. If the frequency is too low, an attempt to enable an output signal path fails. Note that active signal paths are not affected under such circumstances.

The status bits in Register R1025 and R1030 indicate the status of each of the output signal paths. If an underclocked error condition occurs, these bits indicate which signal paths have been enabled.

Table 4-54. Output Signal Path Enable

Register Address	Bit	Label	Default	Description
R1024 (0x0400) Output_Enables_1	9	OUT5L_ENA	0	Output Path 5 (left) enable 0 = Disabled 1 = Enabled
	8	OUT5R_ENA	0	Output Path 5 (right) enable 0 = Disabled 1 = Enabled
	7	SPKOUTL_ENA	0	Output Path 4 (left) enable 0 = Disabled 1 = Enabled
	1	HP1L_ENA	0	Output Path 1 (left) enable When EP_SEL = 0, this bit controls the HPOUTL output driver. When EP_SEL = 1, this bit controls the EPOUTP output driver. 0 = Disabled 1 = Enabled
	0	HP1R_ENA	0	Output Path 1 (right) enable When EP_SEL = 0, this bit controls the HPOUTR output driver. When EP_SEL = 1, this bit controls the EPOUTN output driver. 0 = Disabled 1 = Enabled

Table 4-54. Output Signal Path Enable *(Cont.)*

Register Address	Bit	Label	Default	Description
R1025 (0x0401) Output_Status_1	9	OUT5L_ENA_STS	0	Output Path 5 (left) enable status 0 = Disabled 1 = Enabled
	8	OUT5R_ENA_STS	0	Output Path 5 (right) enable status 0 = Disabled 1 = Enabled
	7	OUT4L_ENA_STS	0	Output Path 4 (left) enable status 0 = Disabled 1 = Enabled
R1030 (0x0406) Raw_Output_Status_1	1	OUT1L_ENA_STS	0	Output Path 1 (left) enable status 0 = Disabled 1 = Enabled
	0	OUT1R_ENA_STS	0	Output Path 1 (right) enable status 0 = Disabled 1 = Enabled

4.8.2 Output Signal Path Sample-Rate Control

The output signal paths are derived from the respective output mixers within the CS47L15 digital core. The sample rate for the output signal paths is configured using OUT_RATE—see Table 4-24.

Note that sample-rate conversion is required when routing the output signal paths to any signal chain that is configured for a different sample rate.

4.8.3 Output Signal Path Control

The OUT1 path is associated with the headphone and the earpiece output drivers. The EP_SEL bit controls which of these outputs can be used—it is not possible to enable the headphone and earpiece drivers simultaneously.

Under default register conditions, the OUT1 path is configured for stereo output. The path can be configured for mono differential (BTL) output using the OUT1_MONO bit; this is ideal for driving an earpiece or hearing aid coil.

When the OUT1_MONO bit is set, the respective right channel output is an inverted copy of the left channel output signal; this creates a differential output between the respective outputs. The left and right channel output drivers must both be enabled in Mono Mode; both channels should be enabled simultaneously using the fields described in Table 4-54.

The mono (BTL) signal paths are shown in Fig. 4-48. Note that, in Mono Mode, the effective gain of the signal path is increased by 6 dB.

For stereo output on HPOUTL and HPOUTR, the required settings are as follows:

- EP_SEL = 0
- OUT1_MONO = 0

For mono differential output on EPOUTP and EPOUTN, the required settings are as follows:

- EP_SEL = 1
- OUT1_MONO = 1

Note that the EP_SEL and OUT1_MONO bits should not be changed while the headphone or earpiece drivers are enabled. These bits should be configured before enabling the respective drivers, and should remain unchanged until after the drivers have been disabled. The HPOUT and EPOUT drivers are enabled using the HP1L_ENA and HP1R_ENA bits, as described in Table 4-54.

The SPKCLK frequency of the PDM output path (OUT5) is controlled by OUT5_OSR, as described in Table 4-55. When the OUT5_OSR bit is set, the audio performance is improved, but power consumption is also increased.

Note that the SPKCLK frequencies noted in Table 4-55 assume that the SYSCLK frequency is a multiple of 6.144 MHz (SYSCLK_FRAC=0). If the SYSCLK frequency is a multiple of 5.6448 MHz (SYSCLK_FRAC = 1), the SPKCLK frequency is scaled accordingly.

Table 4-55. SPKCLK Frequency

OUT5_OSR	Description	SPKCLK Frequency
0	Normal mode	3.072 MHz
1	High Performance mode	6.144 MHz

The output signal path control registers are defined in Table 4-56.

Table 4-56. Output Signal Path Control

Register Address	Bit	Label	Default	Description
R1024 (0x0400) Output_Enables_1	15	EP_SEL	0	Output Path 1 Output Driver select 0 = HPOUTL and HPOUTR 1 = EPOUTP and EPOUTN
R1040 (0x0410) Output_Path_ Config_1L	12	OUT1_MONO	0	Output Path 1 Mono Mode (Configures HPOUT and EPOUT as a mono differential output.) 0 = Disabled 1 = Enabled The gain of the signal path is increased by 6 dB in differential (mono) mode.
R1072 (0x0430) Output_Path_ Config_5L	13	OUT5_OSR	0	Output Path 5 Oversample Rate 0 = Normal mode 1 = High Performance mode

4.8.4 Output Signal Path Digital Volume Control

A digital volume control is provided on each of the output signal paths, providing –64 to +31.5 dB gain control in 0.5-dB steps. An independent mute control is also provided for each output signal path.

Whenever the gain or mute setting is changed, the signal path gain is ramped up or down to the new settings at a programmable rate. For increasing gain (or unmute), the rate is controlled by OUT_VI_RAMP. For decreasing gain (or mute), the rate is controlled by OUT_VD_RAMP.

Note: The OUT_VI_RAMP and OUT_VD_RAMP fields should not be changed while a volume ramp is in progress.

The OUT_VU bits control the loading of the output signal path digital volume and mute controls. When OUT_VU is cleared, the digital volume and mute settings are loaded into the respective control register, but do not change the signal path gain. The digital volume and mute settings on all of the output signal paths are updated when a 1 is written to OUT_VU. This makes it possible to update the gain of multiple signal paths simultaneously.

Note that, although the digital-volume controls provide 0.5-dB steps, the internal circuits provide signal gain adjustment in 0.125-dB steps. This allows a very high degree of gain control—smooth volume ramping under all operating conditions.

Note: The 0 dBFS level of the OUT5 digital output path is not equal to the 0 dBFS level of the CS47L15 digital core. The maximum digital output level is –6 dBFS (see Table 3-8). Under 0 dB gain conditions, a 0 dBFS output from the digital core corresponds to a –6 dBFS level in the PDM output.

The digital volume control registers are described in Table 4-57 and Table 4-58.

Table 4-57. Output Signal Path Digital Volume Control

Register Address	Bit	Label	Default	Description
R1033 (0x0409) Output_Volume_ Ramp	6:4	OUT_VD_ RAMP[2:0]	010	Output Volume Decreasing Ramp Rate (seconds/6 dB) This field should not be changed while a volume ramp is in progress. 000 = 0 ms 011 = 2 ms 110 = 15 ms 001 = 0.5 ms 100 = 4 ms 111 = 30 ms 010 = 1 ms 101 = 8 ms
	2:0	OUT_VI_ RAMP[2:0]	010	Output Volume Increasing Ramp Rate (seconds/6 dB) This field should not be changed while a volume ramp is in progress. 000 = 0 ms 011 = 2 ms 110 = 15 ms 001 = 0.5 ms 100 = 4 ms 111 = 30 ms 010 = 1 ms 101 = 8 ms
R1041 (0x0411) DAC_Digital_ Volume_1L	9	OUT_VU	See Footnote 1	Output Signal Paths Volume Update. Writing 1 to this bit causes the Output Signal Paths Volume and Mute settings to be updated simultaneously
	8	OUT1L_MUTE	1	Output Path 1 (Left) Digital Mute 0 = Unmute 1 = Mute
	7:0	OUT1L_VOL[7:0]	0x80	Output Path 1 (Left) Digital Volume (see Table 4-58 for volume register definition). –64 dB to +31.5 dB in 0.5-dB steps 0x00 = –64dB 0x80 = 0 dB 0xC0 to 0xFF = Reserved 0x01 = –63.5dB … (0.5-dB steps) … (0.5-dB steps) 0xBF = +31.5 dB
R1045 (0x0415) DAC_Digital_ Volume_1R	9	OUT_VU	See Footnote 1	Output Signal Paths Volume Update. Writing 1 to this bit causes the Output Signal Paths Volume and Mute settings to be updated simultaneously
	8	OUT1R_MUTE	1	Output Path 1 (Right) Digital Mute 0 = Unmute 1 = Mute
	7:0	OUT1R_VOL[7:0]	0x80	Output Path 1 (Right) Digital Volume (see Table 4-58 for volume register definition). –64 dB to +31.5 dB in 0.5-dB steps 0x00 = –64dB 0x80 = 0 dB 0xC0 to 0xFF = Reserved 0x01 = –63.5dB … (0.5-dB steps) … (0.5-dB steps) 0xBF = +31.5 dB
R1065 (0x0429) DAC_Digital_ Volume_4L	9	OUT_VU	See Footnote 1	Output Signal Paths Volume Update. Writing 1 to this bit causes the Output Signal Paths Volume and Mute settings to be updated simultaneously
	8	OUT4L_MUTE	1	Output Path 4 (Left) Digital Mute 0 = Unmute 1 = Mute
	7:0	OUT4L_VOL[7:0]	0x80	Output Path 4 (Left) Digital Volume (see Table 4-58 for volume register definition). –64 dB to +31.5 dB in 0.5-dB steps 0x00 = –64dB 0x80 = 0 dB 0xC0 to 0xFF = Reserved 0x01 = –63.5dB … (0.5-dB steps) … (0.5-dB steps) 0xBF = +31.5 dB
R1073 (0x0431) DAC_Digital_ Volume_5L	9	OUT_VU	See Footnote 1	Output Signal Paths Volume Update. Writing 1 to this bit causes the Output Signal Paths Volume and Mute settings to be updated simultaneously
	8	OUT5L_MUTE	1	Output Path 5 (Left) Digital Mute 0 = Unmute 1 = Mute
	7:0	OUT5L_VOL[7:0]	0x80	Output Path 5 (Left) Digital Volume (see Table 4-58 for volume register definition). –64 dB to +31.5 dB in 0.5-dB steps 0x00 = –64dB 0x80 = 0 dB 0xC0 to 0xFF = Reserved 0x01 = –63.5dB … (0.5-dB steps) … (0.5-dB steps) 0xBF = +31.5 dB

Table 4-57. Output Signal Path Digital Volume Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R1077 (0x0435) DAC_Digital_ Volume_5R	9	OUT_VU	See Footnote 1	Output Signal Paths Volume Update. Writing 1 to this bit causes the Output Signal Paths Volume and Mute settings to be updated simultaneously
	8	OUT5R_MUTE	1	Output Path 5 (Right) Digital Mute 0 = Unmute 1 = Mute
	7:0	OUT5R_VOL[7:0]	0x80	Output Path 5 (Right) Digital Volume (see Table 4-58 for volume register definition). –64 dB to +31.5 dB in 0.5-dB steps 0x00 = –64dB 0x80 = 0 dB 0xC0 to 0xFF = Reserved 0x01 = –63.5dB … (0.5-dB steps) … (0.5-dB steps) 0xBF = +31.5 dB

1. Default is not applicable to these write-only bits

Table 4-58 lists the output signal path digital volume settings.

Table 4-58. Output Signal Path Digital Volume Range

Output Volume Register	Volume (dB)	Output Volume Register	Volume (dB)	Output Volume Register	Volume (dB)	Output Volume Register	Volume (dB)
0x00	–64.0	0x31	–39.5	0x62	–15.0	0x93	9.5
0x01	–63.5	0x32	–39.0	0x63	–14.5	0x94	10.0
0x02	–63.0	0x33	–38.5	0x64	–14.0	0x95	10.5
0x03	–62.5	0x34	–38.0	0x65	–13.5	0x96	11.0
0x04	–62.0	0x35	–37.5	0x66	–13.0	0x97	11.5
0x05	–61.5	0x36	–37.0	0x67	–12.5	0x98	12.0
0x06	–61.0	0x37	–36.5	0x68	–12.0	0x99	12.5
0x07	–60.5	0x38	–36.0	0x69	–11.5	0x9A	13.0
0x08	–60.0	0x39	–35.5	0x6A	–11.0	0x9B	13.5
0x09	–59.5	0x3A	–35.0	0x6B	–10.5	0x9C	14.0
0x0A	–59.0	0x3B	–34.5	0x6C	–10.0	0x9D	14.5
0x0B	–58.5	0x3C	–34.0	0x6D	–9.5	0x9E	15.0
0x0C	–58.0	0x3D	–33.5	0x6E	–9.0	0x9F	15.5
0x0D	–57.5	0x3E	–33.0	0x6F	–8.5	0xA0	16.0
0x0E	–57.0	0x3F	–32.5	0x70	–8.0	0xA1	16.5
0x0F	–56.5	0x40	–32.0	0x71	–7.5	0xA2	17.0
0x10	–56.0	0x41	–31.5	0x72	–7.0	0xA3	17.5
0x11	–55.5	0x42	–31.0	0x73	–6.5	0xA4	18.0
0x12	–55.0	0x43	–30.5	0x74	–6.0	0xA5	18.5
0x13	–54.5	0x44	–30.0	0x75	–5.5	0xA6	19.0
0x14	–54.0	0x45	–29.5	0x76	–5.0	0xA7	19.5
0x15	–53.5	0x46	–29.0	0x77	–4.5	0xA8	20.0
0x16	–53.0	0x47	–28.5	0x78	–4.0	0xA9	20.5
0x17	–52.5	0x48	–28.0	0x79	–3.5	0xAA	21.0
0x18	–52.0	0x49	–27.5	0x7A	–3.0	0xAB	21.5
0x19	–51.5	0x4A	–27.0	0x7B	–2.5	0xAC	22.0
0x1A	–51.0	0x4B	–26.5	0x7C	–2.0	0xAD	22.5
0x1B	–50.5	0x4C	–26.0	0x7D	–1.5	0xAE	23.0
0x1C	–50.0	0x4D	–25.5	0x7E	–1.0	0xAF	23.5
0x1D	–49.5	0x4E	–25.0	0x7F	–0.5	0xB0	24.0
0x1E	–49.0	0x4F	–24.5	0x80	0.0	0xB1	24.5
0x1F	–48.5	0x50	–24.0	0x81	0.5	0xB2	25.0
0x20	–48.0	0x51	–23.5	0x82	1.0	0xB3	25.5
0x21	–47.5	0x52	–23.0	0x83	1.5	0xB4	26.0
0x22	–47.0	0x53	–22.5	0x84	2.0	0xB5	26.5
0x23	–46.5	0x54	–22.0	0x85	2.5	0xB6	27.0

Table 4-58. Output Signal Path Digital Volume Range *(Cont.)*

Output Volume Register	Volume (dB)	Output Volume Register	Volume (dB)	Output Volume Register	Volume (dB)	Output Volume Register	Volume (dB)
0x24	–46.0	0x55	–21.5	0x86	3.0	0xB7	27.5
0x25	–45.5	0x56	–21.0	0x87	3.5	0xB8	28.0
0x26	–45.0	0x57	–20.5	0x88	4.0	0xB9	28.5
0x27	–44.5	0x58	–20.0	0x89	4.5	0xBA	29.0
0x28	–44.0	0x59	–19.5	0x8A	5.0	0xBB	29.5
0x29	–43.5	0x5A	–19.0	0x8B	5.5	0xBC	30.0
0x2A	–43.0	0x5B	–18.5	0x8C	6.0	0xBD	30.5
0x2B	–42.5	0x5C	–18.0	0x8D	6.5	0xBE	31.0
0x2C	–42.0	0x5D	–17.5	0x8E	7.0	0xBF	31.5
0x2D	–41.5	0x5E	–17.0	0x8F	7.5	0xC0–0xFF	Reserved
0x2E	–41.0	0x5F	–16.5	0x90	8.0		
0x2F	–40.5	0x60	–16.0	0x91	8.5		
0x30	–40.0	0x61	–15.5	0x92	9.0		

4.8.5 Output Signal Path Noise-Gate Control

The CS47L15 provides a digital noise-gate function for each of the output signal paths. The noise gate ensures best noise performance when the signal path is idle. When the noise gate is enabled, and the applicable signal level is below the noise-gate threshold, the noise gate is activated, causing the signal path to be muted.

The noise-gate function is enabled by setting NGATE_ENA, as described in Table 4-59.

For each output path, the noise gate may be associated with one or more of the signal path threshold detection functions using the x_NGATE_SRC fields. When more than one signal threshold is selected, the output-path noise gate is only activated (i.e., muted) when all of the respective signal thresholds are satisfied.

For example, if the OUT1L noise gate is associated with the OUT1L and OUT1R signal paths, the OUT1L signal path is only muted if both the OUT1L and OUT1R signal levels are below the respective thresholds.

The noise-gate threshold (the signal level below which the noise gate is activated) is set using NGATE_THR. Note that, for each output path, the noise-gate threshold represents the signal level at the respective output pins; the threshold is therefore independent of the digital volume and PGA gain settings.

Note that, although there is only one noise-gate threshold level (NGATE_THR), each of the output-path noise gates may be activated independently, according to the respective signal content and the associated threshold configurations.

To prevent erroneous triggering, a time delay is applied before the gate is activated; the noise gate is only activated (i.e., muted) when the output levels are below the applicable signal level thresholds for longer than the noise-gate hold time. The hold time is set using the NGATE_HOLD field.

When the noise gate is activated, the CS47L15 gradually attenuates the respective signal path at the rate set by OUT_VD_RAMP (see Table 4-57). When the noise gate is deactivated, the output volume increases at the rate set by OUT_VI_RAMP.

Table 4-59. Output Signal Path Noise-Gate Control

Register Address	Bit	Label	Default	Description
R1043 (0x0413) Noise_Gate_Select_1L	11:0	OUT1L_NGATE_SRC[11:0]	0x001	Output Signal Path Noise-Gate Source. Enables one of more signal paths as inputs to the respective noise gate. If more than one signal path is enabled as an input, the noise gate is only activated (i.e., muted) when all of the respective signal thresholds are satisfied. Each bit is coded as 0 = Disabled, 1 = Enabled [11] = Reserved [7] = Reserved [3] = Reserved [10] = Reserved [6] = OUT4L [2] = Reserved [9] = OUT5R [5] = Reserved [1] = OUT1R [8] = OUT5L [4] = Reserved [0] = OUT1L
R1047 (0x0417) Noise_Gate_Select_1R	11:0	OUT1R_NGATE_SRC[11:0]	0x002	
R1067 (0x042B) Noise_Gate_Select_4L	11:0	OUT4L_NGATE_SRC[11:0]	0x040	
R1075 (0x0433) Noise_Gate_Select_5L	11:0	OUT5L_NGATE_SRC[11:0]	0x100	
R1079 (0x0437) Noise_Gate_Select_5R	11:0	OUT5R_NGATE_SRC[11:0]	0x200	
R1112 (0x0458) Noise_Gate_Control	5:4	NGATE_HOLD[1:0]	00	Output Signal Path Noise-Gate Hold Time (delay before noise gate is activated) 00 = 30 ms 10 = 250 ms 01 = 120 ms 11 = 500 ms
	3:1	NGATE_THR[2:0]	000	Output Signal Path Noise-Gate Threshold 000 = –78 dB 011 = –96 dB 110 = –114 dB 001 = –84 dB 100 = –102 dB 111 = –120 dB 010 = –90 dB 101 = –108 dB
	0	NGATE_ENA	0	Output Signal Path Noise-Gate Enable 0 = Disabled 1 = Enabled

4.8.6 Output Signal Path AEC Loop-Back

The CS47L15 incorporates two loop-back signal paths, which are ideally suited as a reference for AEC processing. Any two of the output signal paths may be selected as the AEC loop-back sources.

When configured with suitable DSP firmware, the CS47L15 can provide an integrated AEC capability. The AEC loop-back feature also enables convenient hook-up to an external device for implementing the required signal-processing algorithms.

The AEC loop-back source is connected after the respective digital volume controls, as shown in Fig. 4-48. The AEC loop-back signals can be selected as input to any of the digital mixers within the CS47L15 digital core. The sample rate for the AEC loop-back paths is configured using OUT_RATE—see Table 4-24.

The AEC loop-back function is enabled using the AEC*n*_LOOPBACK_ENA bits (where *n* identifies the applicable path, AEC1 or AEC2). The source signals for the Transmit Path AEC function are selected using the AEC*n*_LOOPBACK_SRC bits.

The CS47L15 performs automatic checks to confirm that the SYSCLK frequency is high enough to support the AEC loop-back function. If the frequency is too low, an attempt to enable this function fails. Note that active signal paths are not affected under such circumstances.

The AEC*n*_ENA_STS bits indicate the status of the AEC loop-back functions. If an underclocked error condition occurs, these bits indicate whether the AEC loop-back function has been enabled.

Table 4-60. Output Signal Path AEC Loop-Back Control

Register Address	Bit	Label	Default	Description
R1104 (0x0450) DAC_AEC_ Control_1	5:2	AEC1_LOOPBACK_ SRC[3:0]	0000	Input source for Tx AEC1 function 0000 = OUT1L 0110 = OUT4L 1001 = OUT5R 0001 = OUT1R 1000 = OUT5L All other codes are reserved
	1	AEC1_ENA_STS	0	Transmit (Tx) Path AEC1 Control Status 0 = Disabled 1 = Enabled
	0	AEC1_LOOPBACK_ ENA	0	Transmit (Tx) Path AEC1 Control 0 = Disabled 1 = Enabled
R1105 (0x0451) DAC_AEC_ Control_2	5:2	AEC2_LOOPBACK_ SRC[3:0]	0000	Input source for Tx AEC2 function 0000 = OUT1L 0110 = OUT4L 1001 = OUT5R 0001 = OUT1R 1000 = OUT5L All other codes are reserved
	1	AEC2_ENA_STS	0	Transmit (Tx) Path AEC2 Control Status 0 = Disabled 1 = Enabled
	0	AEC2_LOOPBACK_ ENA	0	Transmit (Tx) Path AEC2 Control 0 = Disabled 1 = Enabled

4.8.7 Headphone and Earpiece Outputs

The headphone/earpiece driver outputs, HPOUTL, HPOUTR, EPOUTP, and EPOUTN, are suitable for direct connection to external headphones and earpieces. The outputs are ground referenced, eliminating any requirement for AC coupling capacitors.

The headphone output (HPOUTL, HPOUTR) incorporates a common-mode, or ground-loop, feedback path that provides rejection of system-related ground noise. The feedback pin must be connected to ground for normal operation of the headphone output.

The ground feedback path for HPOUTL and HPOUTR is selected using HP1_GND_SEL—see Table 4-61. Note that the selected pin should be connected to GND as close as possible to the respective headphone jack ground pin, as shown in Fig. 4-49.

Table 4-61. Headphone Output (HPOUT) Ground Feedback Control

Register Address	Bit	Label	Default	Description
R1042 (0x0412) Output_Path_ Config_1	2:0	HP1_GND_ SEL[2:0]	000	HPOUT ground feedback pin select 000 = HPOUTFB1 001 = HPOUTFB2 All other codes are reserved

The earpiece output (EPOUTP, EPOUTN) does not support common-mode feedback. The HP1_GND_SEL bit has no effect if the earpiece output is selected (EP_SEL = 1).

The headphone and earpiece connections are shown in Fig. 4-49.

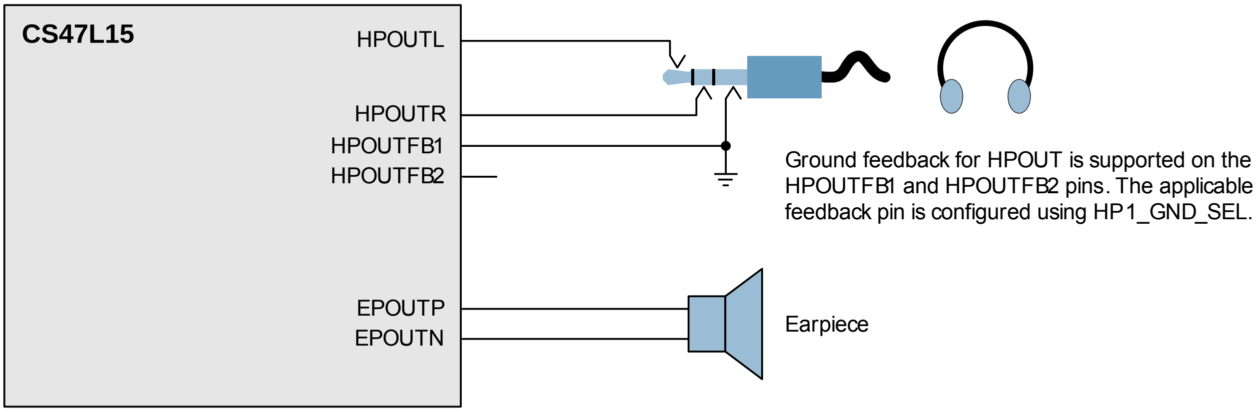


Figure 4-49. Headphone and Earpiece Connection

4.8.8 Speaker Outputs (Analog)

The speaker driver outputs SPKOUTP and SPKOUTN provide differential (BTL) outputs suitable for direct connection to an external loudspeaker. The integrated Class D speaker driver provides high efficiency at large signal levels.

The speaker driver signal path incorporates a boost function that shifts the signal levels between the AVDD and SPKVDD voltage domains. The boost is preconfigured (+12 dB) for the recommended AVDD and SPKVDD operating voltages (see Table 3-3).

Ultralow leakage and high PSRR allow the speaker supply SPKVDD to be connected directly to a lithium battery.

Note that SYSCLK must be present and enabled when using the Class D speaker output; see Section 4.13 for details of SYSCLK and the associated control fields.

The OUT4L output signal path is associated with the analog outputs SPKOUTP and SPKOUTN.

The Class D speaker output is a pulse-width modulated signal, and requires external filtering in order to recreate the audio signal. With a suitable choice of external speakers, the speakers themselves can provide the necessary filtering. See Section 5 for further information on Class D speaker connections.

The external speaker connection is shown in Fig. 4-50, assuming a suitable speaker is chosen to provide the PWM filtering.

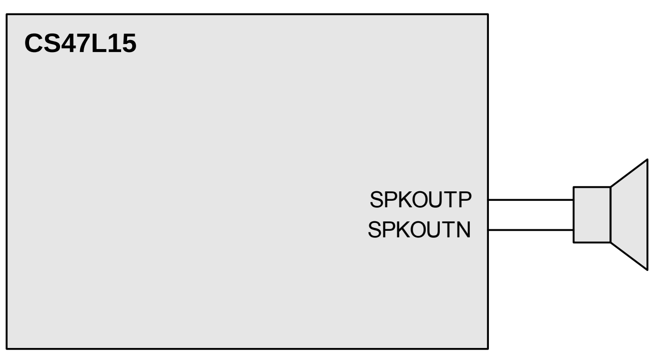


Figure 4-50. Speaker Connection

The speaker output path is designed to support monitoring of external loudspeakers, giving real-time feedback for algorithms such as Cirrus Logic's speaker-protection software. Specific external connections are necessary when using this feature, as detailed below.

The speaker-protection software, implemented on the integrated DSP core, enables loudspeakers to be protected from excessive signal levels and other electro-mechanical constraints. The monitoring circuit enables the operational limits to be continually optimized for the particular loudspeaker and the prevailing conditions. Factors such as cone excursion, resonance, and thermal behavior of the loudspeaker are modeled in the speaker-protection software. As a result, the maximum audio output can be achieved, while ensuring the loudspeakers are also fully protected from damage.

Separate P/N ground connections are provided for the speaker driver; these pins relate to the positive/negative output transistors respectively, to allow comprehensive current monitoring in the output path, as an input to the speaker protection algorithm.

The external speaker connections, incorporating the output current monitoring requirements, are shown in Fig. 4-51. Note that, if output current monitoring is not required, these connections should be tied directly to ground on the PCB.

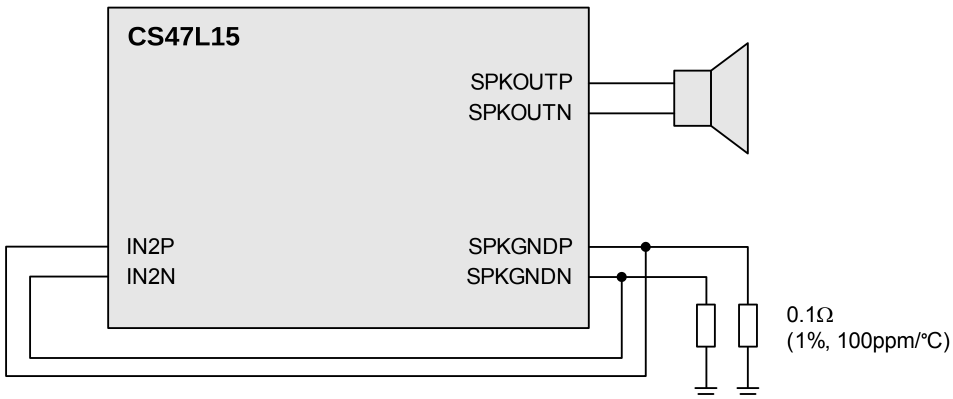


Figure 4-51. Speaker Output Current Monitoring Connections (Speaker Protection)

Please contact your Cirrus Logic representative for further information on the Speaker Protection software.

4.8.9 Speaker Outputs (Digital PDM)

The CS47L15 supports a two-channel pulse-density modulation (PDM) digital speaker interface; the PDM outputs are associated with the OUT5L and OUT5R output signal paths.

The external connections associated with the PDM outputs are implemented on multi-function GPIO pins, which must be configured for the respective PDM functions when required. The PDM output connections are alternative functions available on specific GPIO pins. See Section 4.11 to configure the GPIO pins for the PDM output.

The PDM digital speaker interface is a stereo interface; the OUT5L and OUT5R output signal paths are interleaved on the SPKTXDAT output, and clocked using SPKCLK.

Note that the PDM interface supports two different operating modes; these are selected using SPK1_FMT. See Table 3-15 for detailed timing information in both modes.

- If SPK1_FMT = 0 (Mode A), the left PDM channel is valid at the rising edge of SPKCLK; the right PDM channel is valid at the falling edge of SPKCLK.
- If SPK1_FMT = 1 (Mode B), the left PDM channel is valid during the low phase of SPKCLK; the right PDM channel is valid during the high phase of SPKCLK.

The PDM interface timing is shown in Fig. 4-52.

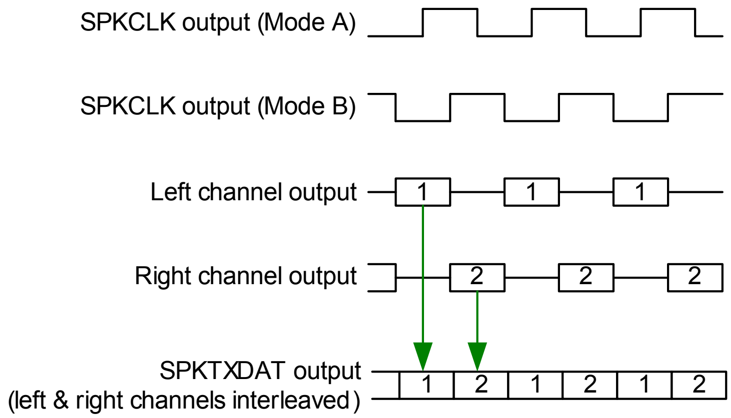


Figure 4-52. Digital Speaker (PDM) Interface Timing

Clocking for the PDM interface is derived from SYSCLK. Note that SYSCLK_ENA must also be set. See Section 4.13 for further details of the system clocks and control registers.

If the OUT5L or OUT5R output signal path is enabled, the PDM interface clock signal is output on the SPKCLK pin.

The output signal paths support normal and high performance operating modes, as described in Section 4.8.3. The SPKCLK frequency is set according to the operating mode of the relevant output path, as described in Table 4-55. The OUT5_OSR bit is defined in Table 4-56.

The PDM output channels can be independently muted. When muted, the default output on each channel is a DSD-compliant silent stream (0110_1001b). The mute output code can be programmed to other values if required, using the SPK1_MUTE_SEQ field. The mute output code can be transmitted MSB-first or LSB-first; this is selectable using the SPK1_MUTE_ENDIAN bit.

Note that the PDM Mute function is not a soft-mute; the audio output is interrupted immediately when the PDM mute is asserted. It is recommended to use the output signal path mute function before applying the PDM mute. See Table 4-57 for details of the OUT5L_MUTE and OUT5R_MUTE bits.

The PDM output interface registers are described in Table 4-62.

Table 4-62. Digital Speaker (PDM) Output Control

Register Address	Bit	Label	Default	Description
R1168 (0x0490) PDM_SPK1_ CTRL_1	13	SPK1R_MUTE	0	PDM Speaker Output 1 (Right) Mute 0 = Audio output (OUT5R) 1 = Mute Sequence output
	12	SPK1L_MUTE	0	PDM Speaker Output 1 (Left) Mute 0 = Audio output (OUT5L) 1 = Mute Sequence output
	8	SPK1_MUTE_ ENDIAN	0	PDM Speaker Output 1 Mute Sequence Control 0 = Mute sequence is LSB first 1 = Mute sequence output is MSB first
	7:0	SPK1_MUTE_ SEQ[7:0]	0x69	PDM Speaker Output 1 Mute Sequence Defines the 8-bit code that is output on muted SPKTXDAT channels.
R1169 (0x0491) PDM_SPK1_ CTRL_2	0	SPK1_FMT	0	PDM Speaker Output 1 timing format 0 = Mode A (PDM data is valid at the rising/falling edges of SPKCLK) 1 = Mode B (PDM data is valid during the high/low phase of SPKCLK)

The digital speaker (PDM) outputs SPKTXDAT and SPKCLK are intended for direct connection to a compatible external speaker driver. A typical configuration is shown in Fig. 4-53.

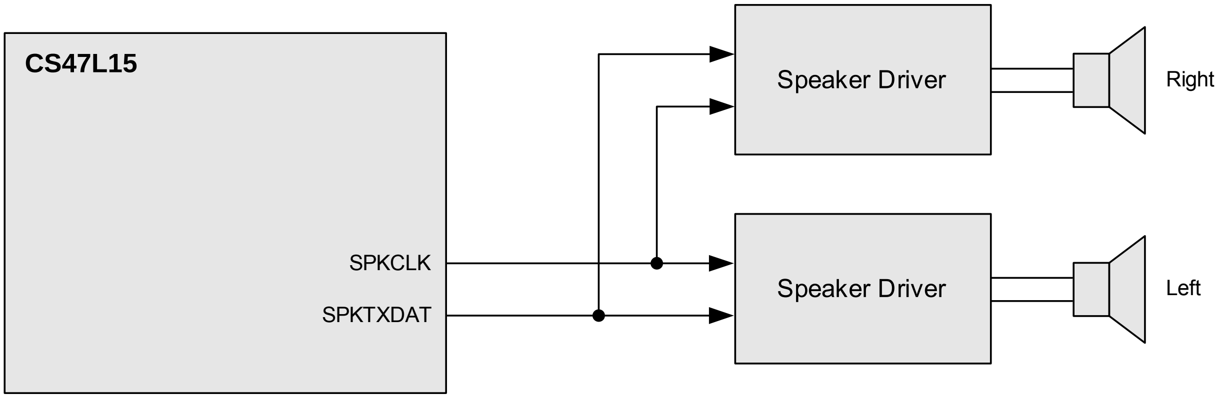


Figure 4-53. Digital Speaker (PDM) Connection

The CS47L15 supports a two-channel digital input path that is synchronized to the PDM interface; this allows a bidirectional audio interface to be supported, using SPKCLK as a shared clock. The PDM interface can be used in this way to support digital feedback from an external speaker driver, enabling advanced speaker protection algorithms to be implemented. See Section 4.2 to configure the SPKRXDAT digital input path.

Typical connections for an external speaker driver, incorporating the digital feedback path, are shown in Fig. 4-54.

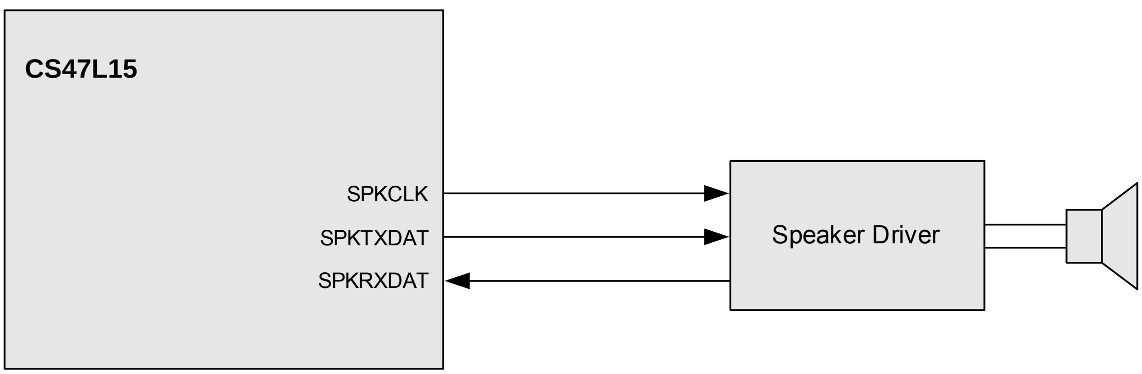


Figure 4-54. Digital Speaker (PDM) Connection with Feedback

4.9 External Accessory Detection

The CS47L15 provides external accessory detection functions that can sense the presence and impedance of external components. This can be used to detect the insertion or removal of an external headphone or headset, and to provide an indication of key/button push events.

Jack insertion is detected using the JACKDET1 and JACKDET2 pins, which must be connected to a switch contact within the jack sockets. An interrupt event is generated whenever a jack insertion or jack removal event is detected.

Suppression of pops and clicks caused by jack insertion or removal is provided using the MICDET clamp function. This function can also be used to trigger interrupt events, and to trigger the control-write sequencer. The integrated general-purpose switch can be synchronized with the MICDET clamp, to provide additional pop-suppression capability.

Microphones, push buttons, and other accessories can be detected via the MICDET1 or MICDET2 pins. The presence of a microphone, and the status of a hook switch can be detected. This feature can also be used to detect push-button operation. (Note that accessory detection is also possible via the HPOUT*x* and JACKDET*n* pins, subject to some additional constraints.)

Headphone impedance can be detected via the HPOUTL and HPOUTR pins; this can be used to set different gain levels or other configuration settings according to the type of load connected. For example, different settings may be applicable to headphone or line output loads. (Note that impedance measurement is also possible via the MICDET*n* and JACKDET*n* pins, subject to some additional constraints.)

The internal 32-kHz clock must be present and enabled when using the microphone detect or headphone detect functions; the 32-kHz clock is also required for the jack detect function, assuming input debounce is enabled. See Section 4.13 for details of the internal 32-kHz clock and associated control fields.

4.9.1 Jack Detect

The CS47L15 provides support for jack insertion switch detection. The jack insertion status can be read using the relevant register status bits. A jack insertion or removal can also be used to trigger an interrupt event.

The jack-detect interrupt (IRQ) functionality is maintained in Sleep Mode (see Section 4.10). This enables a jack insertion event to be used to trigger a wake-up of the CS47L15.

Jack insertion and removal is detected using the JACKDET1 and JACKDET2 pins. The recommended external connections are shown in Fig. 4-55. Note that the logic thresholds associated with the two JACKDET differ from each other, as described in Table 3-11—this provides support for different jack switch configurations.

The jack detect feature is enabled using the JD*n*_ENA bits (where *n* = 1 or 2 for JACKDET1 or JACKDET2 respectively); the jack insertion status can be read using JD*n*_STS*x*. Note that the JD*n*_STS1 and JD*n*_STS2 bits provide the same information in respect of the applicable JACKDET*n* input.

The jack detect input debounce is selected using the JD*n*_DB bits, as described in Table 4-63. Note that, under normal operating conditions, the debounce circuit uses the 32-kHz clock, which must be enabled whenever input debounce functions are required. Input debounce is not provided in Sleep Mode; the JD*n*_DB bits have no effect in Sleep Mode.

Note that the jack detect signals, JD1 and JD2, can be used as inputs to the MICDET clamp function—this provides additional functionality relating to jack insertion and removal events.

An interrupt request (IRQ) event is generated whenever a jack insertion or jack removal is detected (see Section 4.12). Separate mask bits are provided, to allow IRQ events on the rising and/or falling edges of the JD1 or JD2 signals.

The control registers associated with the jack detect function are described in Table 4-63.

Table 4-63. Jack Detect Control

Register Address	Bit	Label	Default	Description
R723 (0x02D3) Jack_detect_ analog	1	JD2_ENA	0	JACKDET2 enable 0 = Disabled 1 = Enabled
	0	JD1_ENA	0	JACKDET1 enable 0 = Disabled 1 = Enabled
R6278 (0x1886) IRQ1_Raw_ Status_7	2	JD2_STS1	0	JACKDET2 input status 0 = Jack not detected 1 = Jack is detected (Assumes the JACKDET2 pin is pulled low on jack insertion.)
	0	JD1_STS1	0	JACKDET1 input status 0 = Jack not detected 1 = Jack is detected (Assumes the JACKDET1 pin is pulled low on jack insertion.)

Table 4-63. Jack Detect Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R6534 (0x1986) IRQ2_Raw_ Status_7	2	JD2_STS2	0	JACKDET2 input status 0 = Jack not detected 1 = Jack is detected (Assumes the JACKDET2 pin is pulled low on jack insertion.)
	0	JD1_STS2	0	JACKDET1 input status 0 = Jack not detected 1 = Jack is detected (Assumes the JACKDET1 pin is pulled low on jack insertion.)
R6662 (0x1A06) Interrupt_ Debounce_7	2	JD2_DB	0	JACKDET2 input debounce 0 = Disabled 1 = Enabled
	0	JD1_DB	0	JACKDET1 input debounce 0 = Disabled 1 = Enabled

A recommended connection circuit, including headphone output on HPOUT and microphone connections, is shown in Fig. 4-55. See Section 5.1 for details of recommended external components.

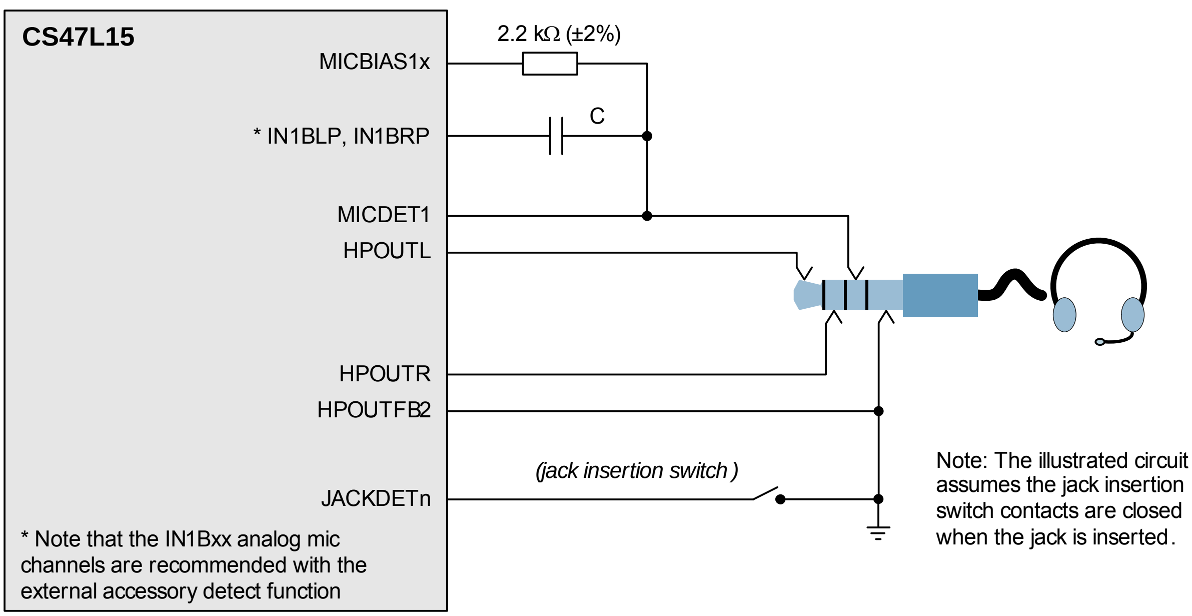


Figure 4-55. Jack Detect and External Accessory Connections

The internal comparator circuit used to detect the JACKDET*n* status is shown in Fig. 4-56. The threshold voltages for the jack detect circuit are noted in Table 3-11. Note that separate thresholds are defined for jack insertion and removal.

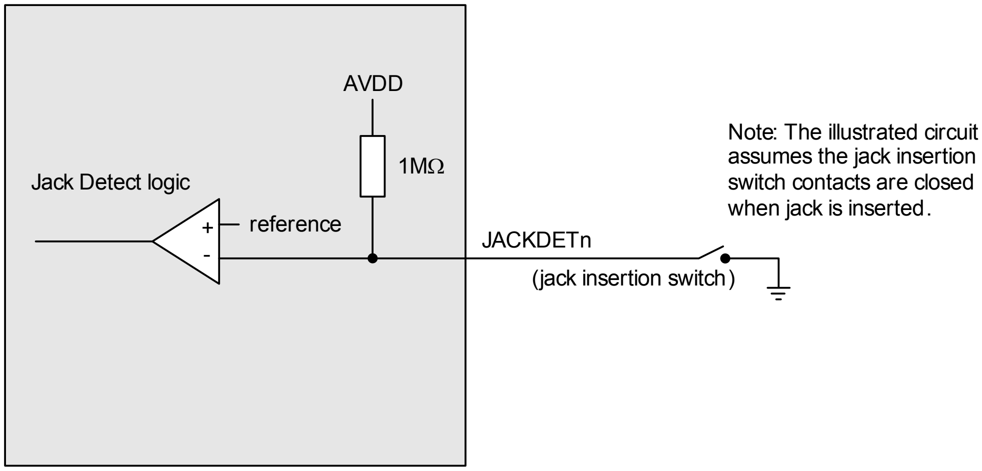


Figure 4-56. Jack Detect Comparator

4.9.2 Jack Pop Suppression (MICDET Clamp and GP Switch)

Under typical configuration of a 3.5-mm headphone/accessory jack connection, there is a risk of pops and clicks arising from jack insertion or removal. This can occur if the headphone load makes momentary contact with the MICBIAS output when the jack is not fully inserted.

The CS47L15 provides a MICDET clamp function to suppress pops and clicks caused by jack insertion or removal. It can be controlled directly, or can be activated by a configurable logic function derived from the JACKDET*n* inputs. The clamp status can be read using the relevant register status bit. The clamp status can also be used to trigger an interrupt (IRQ) event or to trigger the control-write sequencer.

A general-purpose analog switch is incorporated, which can be configured to augment the MICDET clamp functions and to support the pop-suppression circuits, as described in Section 4.9.2.3.

4.9.2.1 MICDET Clamp Control

The MICDET clamp function can be configured using the MICD_CLAMP_MODE field. Selectable logic conditions (derived from the JD1 and JD2 signals—see Table 4-63) provide support for different jack-detect circuit configurations. Setting the MICD_CLAMP_OVD bit enables the MICDET clamp, regardless of other conditions.

Note: The MICD_CLAMP_OVD bit is set by default. Accordingly, the MICDET clamp is always enabled following power-on reset, hardware reset, or software reset.

The MICDET clamp functionality (including the external IRQ) is maintained in Sleep Mode (see Section 4.10). This enables a jack insertion event to be used to trigger a wake-up of the CS47L15. The recommended control sequence for the jack detect and MICDET clamp control is described in Section 4.9.2.5.

If the MICDET clamp is enabled, the MICDET1/HPOUTFB1 and MICDET2/HPOUTFB2 pins are shorted together. The grounding of the MICDET pin is achieved via the applicable HPOUTFB pin—it is assumed that the HPOUTFB connection is grounded externally, as shown in Fig. 4-57.

The selectable logic conditions supported by the MICD_CLAMP_MODE field provides flexibility in selecting the appropriate conditions for controlling the MICDET clamp. The status of the clamp can be read using the MICD_CLAMP_STS*x* bits. Note that the MICD_CLAMP_STS1 and MICD_CLAMP_STS2 bits provide the same information. The status of the clamp in the overridden (MICD_CLAMP_OVD = 1) state is not indicated.

The MICDET clamp debounce is selected by setting MICD_CLAMP_DB, as described in Table 4-64. Note that, under normal operating conditions, the debounce circuit uses the 32-kHz clock, which must be enabled whenever input debounce functions are required. Input debounce is not provided in Sleep Mode; the MICD_CLAMP_DB bit has no effect in Sleep Mode.

The MICDET clamp function is shown in Fig. 4-57. Note that the jack plug is shown partially removed, with the MICDET1 pin in contact with the headphone load.

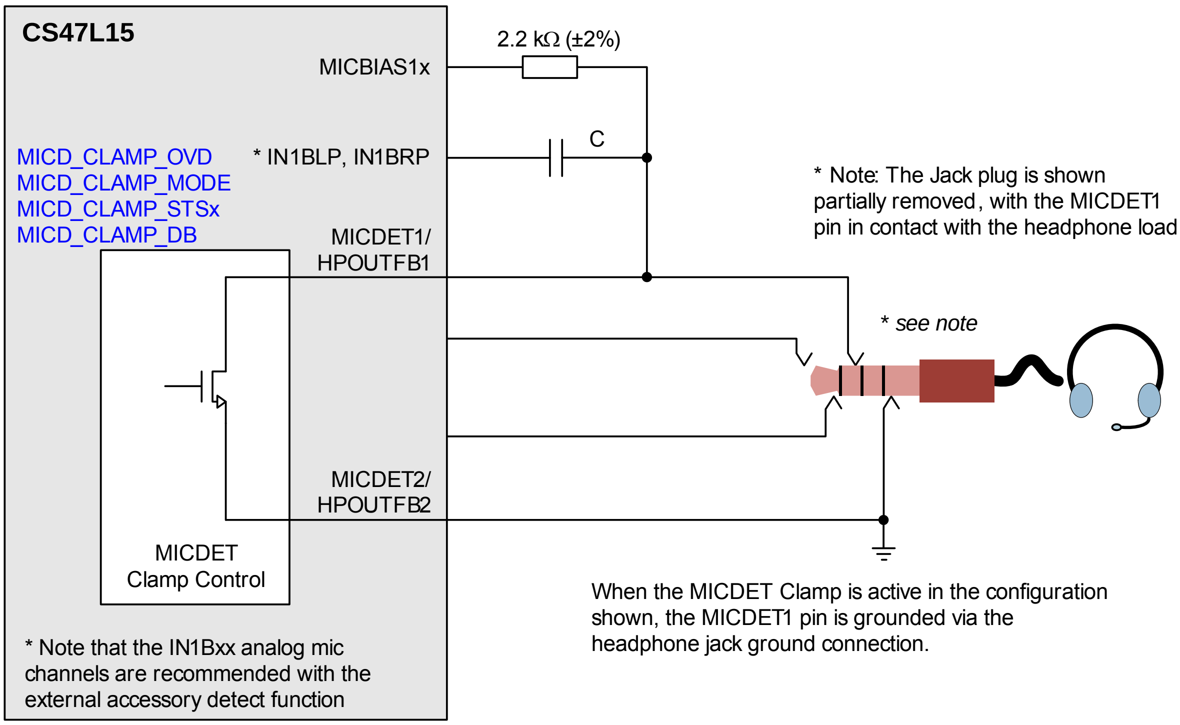


Figure 4-57. MICDET Clamp Circuit

4.9.2.2 Interrupts and Write-Sequencer Control

An interrupt request (IRQ) event can be generated in response to the MICDET clamp status. A MICDET clamp interrupt is generated whenever the logic condition of the JD*n* signals cause a change in the clamp status. Separate maskable interrupts are provided for the rising and falling edges of the MICDET clamp status—see Section 4.12.

The control-write sequencer can be triggered by the MICDET clamp status. This is enabled using the WSEQ_ENA_MICD_CLAMP_FALL and WSEQ_ENA_MICD_CLAMP_RISE bits. Note that the control-sequencer events are only valid if the clamp status changed in response to the JD*n* signals. See Section 4.15 for details of the control-write sequencer.

4.9.2.3 Pop Suppression using General-Purpose Switch

In applications where a large decoupling capacitance is present on the MICBIAS output, the MICDET clamp function may be unable to discharge the capacitor sufficiently to eliminate pops and clicks associated with jack insertion and removal. In this case, it may be desirable to use the general-purpose switch on the CS47L15 to provide isolation from the MICBIAS output; an example circuit is shown in Fig. 4-58.

The general-purpose switch is configured using SW1_MODE. This field allows the switch to be disabled, enabled, or synchronized to the MICDET clamp status, as described in Table 4-64.

For jack pop suppression, it is recommended to set SW1_MODE = 11. In this case, the switch contacts are open whenever the MICDET clamp status bits are set (clamp enabled), and the switch contacts are closed whenever the MICDET clamp status bits are clear (clamp disabled).

A typical pop-suppression circuit, incorporating the general-purpose switch and MICDET clamp function, is shown in Fig. 4-58. Normal accessory functions are supported when the switch contacts (GPSWP and GPSWN) are closed, and the MICDET clamp is disabled. Ground clamping of MICDET, and isolation of MICBIAS are achieved when the switch contacts are open, and the MICDET clamp is enabled.

Note that the MICDET clamp function must also be configured appropriately if using this method of pop suppression control.

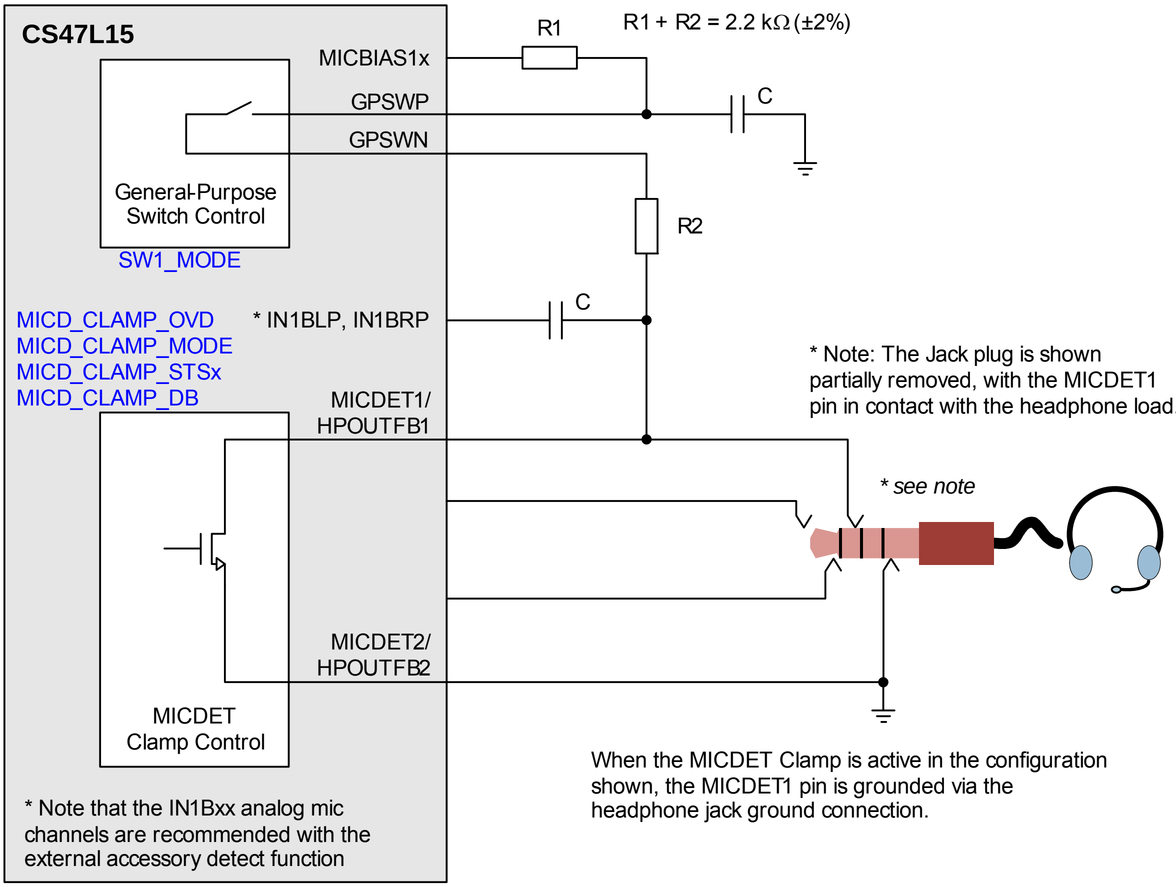


Figure 4-58. General-Purpose Switch Circuit

4.9.2.4 MICDET Clamp Control Registers

The control registers associated with the MICDET clamp and general-purpose switch functions are described in Table 4-64.

Table 4-64. MICDET Clamp and General-Purpose Switch Control

Register Address	Bit	Label	Default	Description
R65 (0x0041) Sequence_control	7	WSEQ_ENA_ MICD_CLAMP_ FALL	0	MICDET Clamp (Falling) Write Sequencer Select 0 = Disabled 1 = Enabled
	6	WSEQ_ENA_ MICD_CLAMP_ RISE	0	MICDET Clamp (Rising) Write Sequencer Select 0 = Disabled 1 = Enabled

Table 4-64. MICDET Clamp and General-Purpose Switch Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R710 (0x02C6) Micd_Clamp_ control	4	MICD_CLAMP_ OVD	1	MICDET Clamp Override 0 = Disabled (clamp is controlled by MICD_CLAMP_MODE) 1 = Enabled (clamp is enabled)
	3:0	MICD_CLAMP_ MODE[3:0]	0000	MICDET Clamp Mode 0x0 = Disabled 0x1 = Enabled (MICDET1/MICDET2 shorted together) 0x2–0x3 = Reserved 0x4 = Enabled if JD1=0 0x5 = Enabled if JD1=1 0x6 = Enabled if JD2=0 0x7 = Enabled if JD2=1 0x8 = Enabled if JD1=0 or JD2=0 0x9 = Enabled if JD1=0 or JD2=1 0xA = Enabled if JD1=1 or JD2=0 0xB = Enabled if JD1=1 or JD2=1 0xC = Enabled if JD1=0 and JD2=0 0xD = Enabled if JD1=0 and JD2=1 0xE = Enabled if JD1=1 and JD2=0 0xF = Enabled if JD1=1 and JD2=1
R712 (0x02C8) GP_Switch_1	1:0	SW1_ MODE[1:0]	00	General-purpose Switch control 00 = Disabled (switch open) 01 = Enabled (switch closed) 10 = Enabled if MICDET clamp status is set 11 = Enabled if MICDET clamp status is clear
R6278 (0x1886) IRQ1_Raw_ Status_7	4	MICD_CLAMP_ STS1	0	MICDET Clamp status 0 = Clamp disabled 1 = Clamp enabled
R6534 (0x1986) IRQ2_Raw_ Status_7	4	MICD_CLAMP_ STS2	0	MICDET Clamp status 0 = Clamp disabled 1 = Clamp enabled
R6662 (0x1A06) Interrupt_ Debounce_7	4	MICD_CLAMP_ DB	0	MICDET Clamp debounce 0 = Disabled 1 = Enabled

4.9.2.5 Control Sequence for Jack Detect and MICDET Clamp

A summary of the jack detect and MICDET clamp functionality, and the recommended usage in typical applications, is described as follows.

- On device power-up, and following reset, the MICDET clamp is enabled due to the default setting of MICD_CLAMP_ OVD; this ensures no spurious output can occur during jack insertion. It is recommended to keep the MICDET clamp enabled (MICD_CLAMP_OVD = 1) until after a jack insertion has been detected.

  The MICDET_CLAMP_MODE field should be set according to the required JD1/JD2 logic condition (configured to enable the clamp when jack is removed).
- Jack insertion is indicated using the JD1/JD2 signals or MICDET clamp interrupt (assuming that the MICDET_ CLAMP_MODE field has been correctly set for the applicable JD1/JD2 signal configuration); the associated status bits can be read directly, or associated signals can be unmasked as inputs to the interrupt controller.

  After jack insertion has been detected, the applicable headset functions (headphone, microphone, accessory detect) may then be enabled.

  If the headset function requires MICBIAS to be enabled on the respective jack, the MICDET clamp should be disabled (MICD_CLAMP_OVD = 0) immediately before enabling the MICBIAS (or immediately before enabling MICD_ENA). Note that, if MICBIAS is not required on the respective jack, the clamp should not be disabled (e.g., for headphone-only operation).
- Jack removal is also indicated using the JD1/JD2 signals or MICDET clamp interrupts. The associated status bits can be read directly, or can be unmasked as inputs to the interrupt controller. The MICDET clamp ensures fast and automatic silencing of the jack outputs.

  Under typical use cases, the respective MICBIAS generator and headset audio paths should all be disabled following jack removal.

  After jack removal has been detected, the MICDET clamp override bit (MICD_CLAMP_OVD) should be set, to make the system ready for a jack insertion.

The recommended control sequence for jack detect and MICDET clamp is summarized in Table 4-65.

Table 4-65. Control Sequence for Jack Detect and MICDET Clamp

Event	Device Actions	Recommended User Actions
Initial condition	Clamp enabled by default	Configure MICDET_CLAMP_MODE
Jack insertion	Jack insertion signaled via IRQ	For headphone-only operation: Enable output signal paths For other use cases: Disable clamp, MICD_CLAMP_OVD = 0 Enable MICBIAS and MICDET Enable I/O signal paths
Jack removal	Jack removal signaled via IRQ, Clamp enabled automatically	Disable MICBIAS and MICDET Disable I/O signal paths Enable clamp MICD_CLAMP_OVD = 1

4.9.3 Microphone Detect

The CS47L15 microphone detection circuit measures the impedance of an external load connected to one of the MICDET pins. This feature can be used to detect the presence of a microphone, and the status of the associated hook switch. It can also be used to detect push-button status or the connection of other external accessories.

4.9.3.1 Microphone Detect Control

The microphone detection circuit measures the external impedance connected to the MICDET*n* pins. In the discrete measurement mode, the function reports whether the measured impedance lies within one of eight predefined levels. In the ADC measurement mode, a more specific result is provided in the form of a 7-bit ADC output.

Note that microphone/accessory detection is also possible via the HPOUT*x* and JACKDET*n* pins, subject to some additional constraints. If the measurement (sense) pin is connected to MICVDD or MICBIAS1*x* (typically via a 2.2-kΩ bias resistor), MICDET*n* must always be used.

The microphone detection circuit typically uses one of the MICBIAS outputs as a reference. The CS47L15 automatically enables the appropriate MICBIAS output when required in order to perform the detection function; this allows the detection function to be supported in low-power standby operating conditions.

The internal 32-kHz clock must be present and enabled when using the microphone detection function; see Section 4.13 for details.

To configure the microphone detection circuit, the applicable pin connections for the intended measurement must be written to the MICD1_SENSE_SEL and MICD1_GND_SEL fields. The detection circuit measures the external impedance between the pins selected by these two fields; the valid selections for each are defined in Table 4-66.

Note: There is no requirement for the SENSE and GND pin selections to be uniquely assigned between the microphone detect and headphone detect functions—the same pin may be used as a SENSE or GND connection for more than one of the detection functions. If multiple microphone/headphone detections are enabled, the respective measurements are automatically scheduled in isolation to each other. See Section 4.9.4 for details of the headphone detect function.

The microphone detection circuit uses MICVDD, or any one of the MICBIAS1*x* sources, as a reference. The applicable source is configured using the MICD1_BIAS_SRC field. If HPOUT*x* or JACKDET*n* is selected as the measurement pin (MICD1_SENSE_SEL = 1XX), MICD1_BIAS_SRC should be set to 1111.

The microphone detection function is enabled by setting MICD1_ENA.

When microphone detection is enabled, the CS47L15 performs a number of measurements in order to determine the external impedance between the selected pins. The measurement process is repeated at a cyclic rate controlled by MICD1_RATE. The MICD1_RATE field selects the delay between completion of one measurement and the start of the next. When the microphone detection result has settled, the CS47L15 indicates valid data by setting MICD1_VALID.

The discrete measurement mode and ADC measurement mode provide different capabilities for microphone detection. The control requirements and the measurement indication mechanisms differ according to the selected mode, as follows:

- In the discrete measurement mode (MICD1_ADC_MODE = 0), the measured impedance is only deemed valid after more than one successive measurement has produced the same result. The MICD1_DBTIME field provides control of the debounce period; this can be either two measurements or four measurements.

  When the microphone detection result has settled (i.e., after the applicable debounce period), the CS47L15 indicates valid data by setting the MICD1_VALID bit. The measured impedance is indicated using the MICD1_LVL and MICD1_STS bits, as described in Table 4-66.

  The MICD1_VALID bit, when set, remains asserted for as long as the microphone detection function is enabled (i.e., while MICD1_ENA = 1). If the detected impedance changes, the MICD1_LVL and MICD1_STS fields change, but the MICD1_VALID bit remains set, indicating valid data at all times.

  The detection circuit supports up to eight impedance levels (including the no-accessory-detected level), enabling detection of a typical microphone and up to six push buttons. Each measurement level can be enabled or disabled independently; this provides flexibility according to the required thresholds, and offers a faster measurement time in some applications. The MICD1_LVL_SEL field is described in Section 4.9.3.3. The default configuration supports a maximum of four push buttons, in accordance with the Android™ wired headset specification.

  Note that, for typical headset detection, the choice of external resistance values must take into account the impedance of the microphone—the detected impedance corresponds to the combined parallel resistance of the microphone and any asserted push button. Examples of suitable external components are described in Section 5.1.8.
- In the ADC measurement mode (MICD1_ADC_MODE = 1), the detection function generates two output results, contained within the MICD1_ADCVAL and MICD1_ADCVAL_DIFF fields. These fields contain the most recent measurement value (MICD1_ADCVAL) and the measurement difference value (MICD1_ADCVAL_DIFF). The difference value indicates the difference between the latest measurement and the previous measurement; this can be used to determine whether the measurement is stable and reliable.

  In ADC measurement mode, the detection function must be disabled before the measurement can be read. When the CS47L15 indicates valid data (MICD1_VALID = 1), the detection must be disabled by setting MICD1_ENA = 0.

  Note that MICD1_ADCVAL and MICD1_ADCVAL_DIFF do not follow a linear coding. The appropriate test condition for accepting the measurement value (or for rescheduling the measurement) varies depending on the application requirements, and depending on the expected impedance value.

The microphone detection functions are inputs to the interrupt control circuit and can be used to trigger an interrupt event every time an accessory insertion, removal, or impedance change is detected; see Section 4.12.

The fields associated with microphone detection (or other accessories) are described in Table 4-66. The external circuit configuration is shown in Fig. 4-59.

Table 4-66. Microphone Detect Control

Register Address	Bit	Label	Default	Description
R674 (0x02A2) Mic_Detect_1_Control_0	15	MICD1_ADC_MODE	0	Mic Detect 1 Measurement Mode 0 = Discrete Mode 1 = ADC Mode
	7:4	MICD1_SENSE_SEL[3:0]	0001	Mic Detect 1 Sense Select 0000 = MICDET1 0001 = MICDET2 0100 = HPOUTL 0101 = HPOUTR 0110 = JACKDET1 0111 = JACKDET2 All other codes are reserved
	2:0	MICD1_GND_SEL[2:0]	000	Mic Detect 1 Ground Select 000 = MICDET1/HPOUTFB1 001 = MICDET2/HPOUTFB2 All other codes are reserved

Table 4-66. Microphone Detect Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R675 (0x02A3) Mic_Detect_1_ Control_1	15:12	MICD1_BIAS_ STARTTIME[3:0]	0001	Mic Detect 1 Bias Start-up Delay (Selects the delay time between enabling the MICBIASnx reference and performing the MICDET function.) 0000 = 0 ms (continuous) 0101 = 4 ms 1010 = 128 ms 0001 = 0.25 ms 0110 = 8 ms 1011 = 256 ms 0010 = 0.5 ms 0111 = 16 ms 1100 = 512 ms 0011 = 1 ms 1000 = 32 ms 1101 = 24 ms 0100 = 2 ms 1001 = 64 ms 1110 to 1111 = 512 ms
	11:8	MICD1_ RATE[3:0]	0001	Mic Detect 1 Rate (Selects the delay between successive MICDET measurements.) 0000 = 0 ms (continuous) 0101 = 4 ms 1010 = 128 ms 0001 = 0.25 ms 0110 = 8 ms 1011 = 256 ms 0010 = 0.5 ms 0111 = 16 ms 1100 = 512 ms 0011 = 1 ms 1000 = 32 ms 1101 = 24 ms 0100 = 2 ms 1001 = 64 ms 1110 to 1111 = 512 ms
	7:4	MICD1_BIAS_ SRC[3:0]	0000	Mic Detect 1 Reference Select 0000 = MICBIAS1A 0010 = MICBIAS1C All other codes are reserved 0001 = MICBIAS1B 1111 = MICVDD
	1	MICD1_DBTIME	1	Mic Detect 1 Debounce 0 = 2 measurements 1 = 4 measurements Only valid if MICD1_ADC_MODE = 0.
	0	MICD1_ENA	0	Mic Detect 1 Enable 0 = Disabled 1 = Enabled
R676 (0x02A4) Mic_Detect_1_ Control_2	7:0	MICD1_LVL_ SEL[7:0]	1001_ 1111	Mic Detect 1 Level Select (enables mic/accessory detection in specific impedance ranges) [7] = Enable 1–30 kΩ detection [3] = Not used [6] = Not used [2] = Enable 360–680 Ω detection [5] = Not used [1] = Enable 210–290 Ω detection [4] = Not used [0] = Enable 110–180 Ω detection Only valid if MICD1_ADC_MODE = 0.
R677 (0x02A5) Mic_Detect_1_ Control_3	10:2	MICD1_LVL[8:0]	0_ 0000_ 0000	Mic Detect 1 Level (indicates the measured impedance) [8] = 1–30 kΩ [3] = 360–680 Ω [7] = Not used [2] = 210–290 Ω [6] = Not used [1] = 110–180 Ω [5] = Not used [0] = 0–70 Ω [4] = Not used Accessory detection is assured within the specified impedance limits. Note that other impedance conditions, including loads >30 kΩ, may also be indicated using these bits. Only valid if MICD1_ADC_MODE = 0.
	1	MICD1_VALID	0	Mic Detect 1 Data Valid 0 = Not Valid 1 = Valid
	0	MICD1_STS	0	Mic Detect 1 Status 0 = Mic/accessory not detected 1 = Mic/accessory detected Mic/accessory detection is assured for load impedance up to 30 kΩ. Only valid when MICD1_ADC_MODE = 0.
R683 (0x02AB) Mic_Detect_1_ Control_4	15:8	MICD1_ ADCVAL_ DIFF[7:0]	0x00	Mic Detect 1 ADC Level (Difference) Only valid if MICD1_ADC_MODE = 1.
	6:0	MICD1_ ADCVAL[6:0]	0x00	Mic Detect 1 ADC Level Only valid if MICD1_ADC_MODE = 1.

The external connections for the microphone detect circuit are shown in Fig. 4-59. In typical applications, it can be used to detect a microphone or button press.

Note that, when using the microphone detect circuit, it is recommended to use the IN1BLP or IN1BRP analog microphone input paths to ensure best immunity to electrical transients arising from the external accessory.

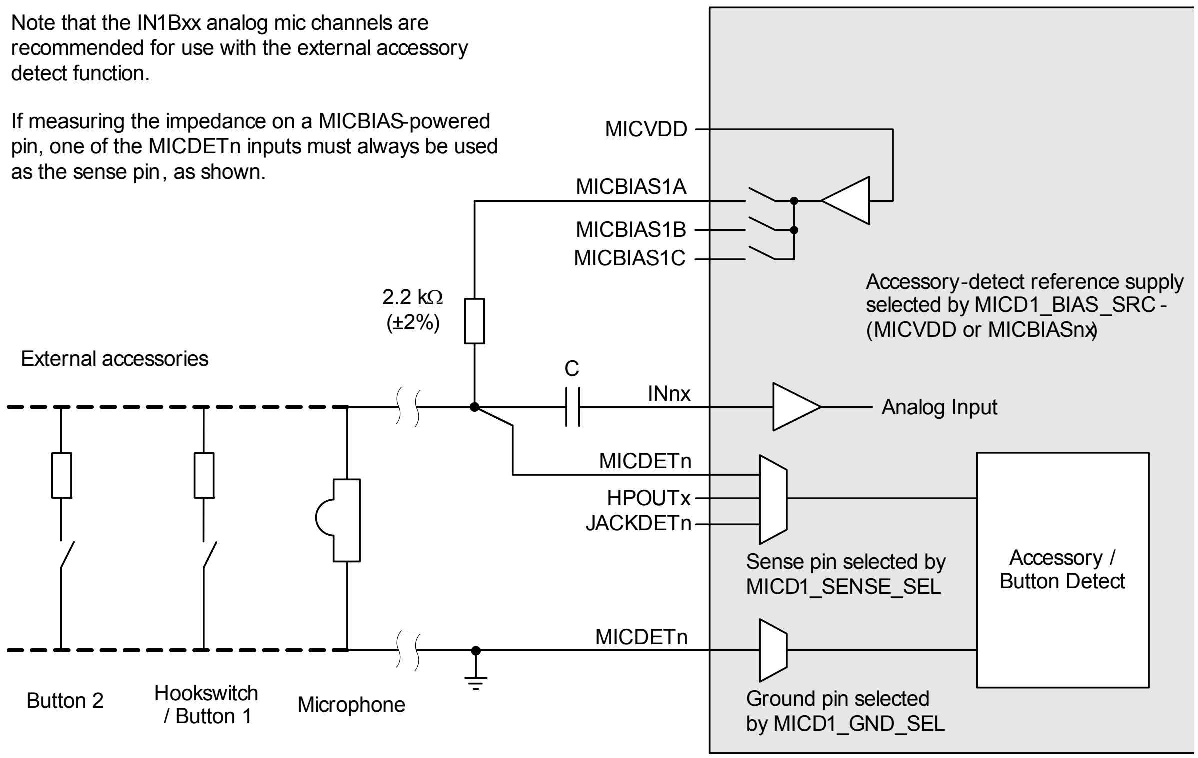


Figure 4-59. Microphone- and Accessory-Detect Interface

4.9.3.2 MICBIAS Reference Control

The voltage reference for the microphone detection is configured using the MICD1_BIAS_SRC field, as described in Table 4-66. The microphone detection function automatically enables the applicable reference when required for impedance measurement.

If the selected reference (MICBIAS1*x*) is not already enabled, the microphone detect circuit automatically enables the respective MICBIAS output for short periods of time only, every time the impedance measurement is scheduled. To allow time for the associated circuitry to stabilize, a time delay is applied before the measurement is performed; this is configured using MICD1_BIAS_STARTTIME, as described in Table 4-66. If the measurement rate setting (MICD1_RATE) is greater than 0x0, the delay (MICD1_BIAS_STARTTIME) should be set to 0.25 ms or more.

Note: The microphone detection automatically enables the applicable MICBIAS1*x* output switch, every time the impedance measurement is scheduled. The MICBIAS generator is not controlled automatically—the MICBIAS1 generator must be enabled using the MICB1_ENA bit, as described in Table 4-103.

The timing of the microphone detect function is shown in Fig. 4-60. Two different cases are shown, according to whether MICBIAS1*x* is enabled periodically by the impedance measurement function, or is enabled at all times.

If the selected reference (MICBIAS1*x*) is not enabled continuously, the respective MICBIAS1*x* discharge bits should be cleared. The MICBIAS control registers are described in Section 4.16.

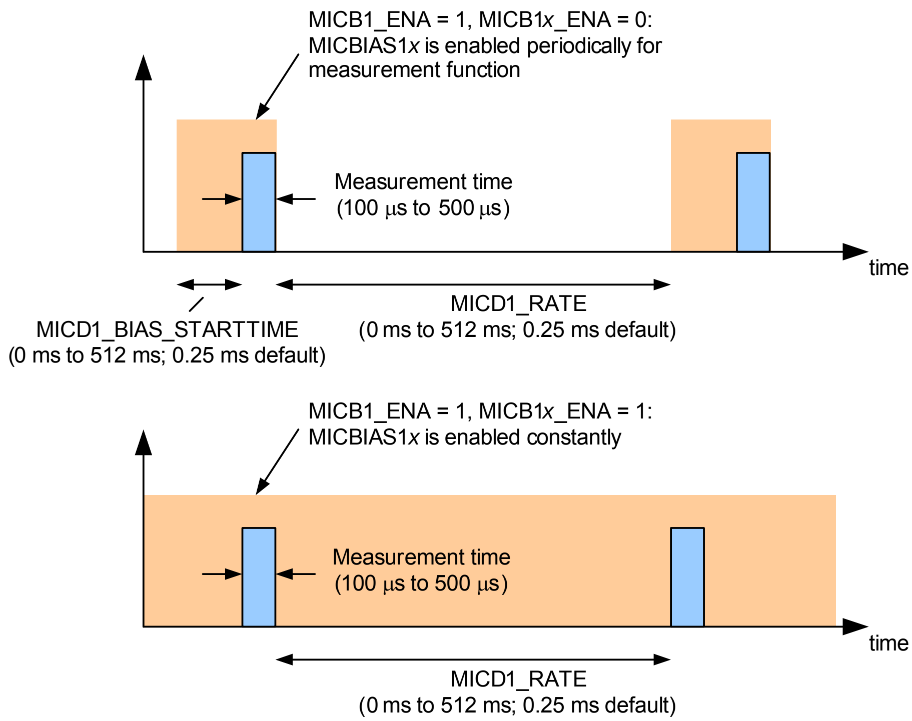


Figure 4-60. Microphone- and Accessory-Detect Timing

4.9.3.3 Measurement Range Control

If the discrete measurement mode is selected (MICD1_ADC_MODE = 000), the MICD1_LVL_SEL[7:0] bits allow each of the impedance measurement levels to be enabled or disabled independently. This allows the function to be tailored to the particular application requirements.

If one or more bits MICD1_LVL_SEL is cleared, the corresponding impedance level is disabled. Any measured impedance which lies in a disabled level is reported as the next lowest, enabled level.

For example, the MICD1_LVL_SEL[2] bit enables the detection of a 360–680 Ω impedance. If MICD1_LVL_SEL[2] = 0, an external impedance in this range is indicated in the next lowest detection range (210–290 Ω); this would be reported in the MICD1_LVL field as MICD1_LVL[2] = 1.

With default register configuration, and all measurement levels enabled, the CS47L15 can detect the presence of a typical microphone and up to four push buttons. It is possible to configure the detection circuit for up to eight push buttons, by adjusting the impedance detection thresholds. However, adjustment of the detection thresholds is outside the scope of this datasheet—please contact your local Cirrus Logic representative for further information, if required.

The measurement time varies between 100–500 µs, depending on the impedance of the external load, and depending on how many impedance measurement levels are enabled. A high impedance is measured faster than a low impedance.

4.9.3.4 External Components

The external connections for the microphone detect circuit are shown in Fig. 4-59. Examples of suitable external components are described in Section 5.1.8.

The accuracy of the microphone detect function is assured whenever the connected load is within the applicable limits specified in Table 3-11. It is required that a 2.2-kΩ (2%) resistor must also be connected between the measurement (SENSE) pin and the selected MICBIAS reference—different resistor values lead to inaccuracy in the impedance measurement.

Note that, for typical headset detection, the choice of external resistance values must take into account the impedance of the microphone—the detected impedance corresponds to the combined parallel resistance of the microphone and any asserted push button.

4.9.4 Headphone Detect

The CS47L15 headphone detection circuit measures the impedance of an external headphone load. This feature can be used to set different gain levels or to apply other configuration settings according to the type of load connected. Separate monitor pins are provided for headphone detection on the left and right channels of HPOUT.

4.9.4.1 Headphone Detection Control

The headphone detection circuit measures the external impedance connected to the HPOUTL or HPOUTR pin. In typical usage, this provides measurement of the load impedance on the headphone outputs.

Note that impedance measurement is also possible via the MICDET*n* and JACKDET*n* pins, subject to some additional constraints. If the measurement (sense) pin is connected to one of the headphone outputs, HPOUTL, HPOUTR*,* or JACKDET1 must always be used. The valid measurement range and the measurement accuracy are reduced, if using the MICDET*n* or JACKDET*n* pins.

To configure the headphone detection circuit, the applicable pin connections for the intended measurement must be written to the HPD_SENSE_SEL and HPD_GND_SEL fields. The headphone detection circuit measures the external impedance between the pins selected by these two fields; the valid selections for each are defined in Table 4-69.

- When measuring the load impedance on the HPOUT output paths, the HPD_GND_SEL selection should be the same MICDETn/HPOUTFBn pin as the ground feedback pin for the headphone output. See Section 4.8.7 to configure the ground feedback pin for HPOUT.

The HPD_FRC_SEL field must also be configured, to select where the measurement current is applied. As a general rule, this should be the same as the HPD_SENSE_SEL pin. Other configurations can be used if required—for example, to improve measurement accuracy in cases where the SENSE input path includes significant unwanted resistance.

Note: There is no requirement for the SENSE and GND pin selections to be uniquely assigned between the microphone detect and headphone detect functions—the same pin may be used as a SENSE or GND connection for more than one of the detection functions. If multiple microphone/headphone detections are enabled, the respective measurements are automatically scheduled in isolation to each other. See Section 4.9.3 for details of the microphone detect function.

Headphone detection on the selected channel is commanded by writing 1 to HPD_POLL.

The impedance measurement range is configured using HPD_IMPEDANCE_RANGE. This field should be set in accordance with the expected load impedance. Note that a number of separate measurements are typically required to determine the load impedance; the recommended control requirements are described in Section 4.9.4.2.

Note: Setting HPD_IMPEDANCE_RANGE is not required for detection on the MICDET*n* or JACKDET*n* pins (HPD_SENSE_SEL = 0XX or 11X). The impedance measurement range, and measurement accuracy, in these cases are different to the HPOUTL and HPOUTR measurements.

For correct operation, the respective output drivers must be disabled when headphone detection is commanded on HPOUTL or HPOUTR. The required settings are shown in Table 4-67.

Table 4-67. Output Configuration for Headphone Detect

Description	Requirement
HPOUTL Impedance measurement	HP1L_ENA = 0
HPOUTR Impedance measurement	HP1R_ENA = 0

Note: The applicable headphone outputs configuration must be maintained until after the headphone detection has completed. See Table 4-54 for details of the HP1L_ENA and HP1R_ENA bits.

If headphone detection is performed using a measurement pin that is not connected to one of the headphone outputs, the HPD_OVD_ENA bit should be cleared.

If headphone detection is performed using a measurement pin that is also connected to one of the MICBIAS outputs, the respective MICBIAS output must be disabled and floating (MICB*nx*_ENA = 0, MICB*nx*_DISCH = 0).

When headphone detection is commanded, the CS47L15 uses an adjustable current source to determine the connected impedance. A sweep of measurement currents is applied. The rate of this sweep can be adjusted using HPD_CLK_DIV and HPD_RATE.

4.9.4.2 Measurement Output

The headphone detection process typically comprises a number of separate measurements (for different impedance ranges). Completion of each measurement is indicated by HPD_DONE. When this bit is set, the measurement result can be read from the HPD_DACVAL field, and decoded as described in Eq. 4-2.

$$\text{Impedance }(\Omega) = \frac{C_0 + (C_1 \times \mathit{Offset})}{\left[\frac{(\text{HPD\_DACVAL} + 0.5)}{C_2}\right] - \left[\frac{1}{C_3(1 + (C_4 \times \mathit{Gradient}))}\right]} - C_5$$

Equation 4-2. Headphone Impedance Calculation

The associated parameters for decoding the measurement result are defined Table 4-68. The applicable values are dependent on the HPD_IMPEDANCE_RANGE setting in each case. The *Offset* and *Gradient* values are derived from register fields that are factory-calibrated for each device.

Table 4-68. Headphone Measurement Decode Parameters

Parameter	HPD_IMPEDANCE_ RANGE = 01	HPD_IMPEDANCE_ RANGE = 10	HPD_IMPEDANCE_ RANGE = 11
C_0	1.0	9.633	100.684
C_1	–0.0043	–0.0795	–0.9494
C_2	7975	7300	7300
C_3	69.6	62.9	63.2
C_4	0.0055	0.0045	0.0045
C_5 HPD_SENSE_SEL = 0100 or 0101 All other cases	33.35 0.85	33.35 0.85	33.35 0.85
Offset	HP_OFFSET_01	HP_OFFSET_10	HP_OFFSET_11
Gradient	HP_GRADIENT_0X	HP_GRADIENT_1X	HP_GRADIENT_1X

Note that, to achieve the specified measurement accuracy, the above equation must be calculated to an accuracy of at least 5 decimal places throughout.

The impedance measurement result is valid if 169 ≤ HPD_DACVAL ≤ 1019. (In case of any contradiction with the HPD_ IMPEDANCE_RANGE description, the HPD_DACVAL validity takes precedence.)

If the external impedance is entirely unknown (i.e., it could lie in any of the HPD_IMPEDANCE_RANGE regions), it is recommended to test initially with HPD_IMPEDANCE_RANGE = 01. If the resultant HPD_DACVAL is < 169, the impedance is higher than the selected measurement range, so the test should be scheduled again, after incrementing HPD_IMPEDANCE_RANGE.

Each measurement is triggered by writing 1 to HPD_POLL. Completion of each measurement is indicated by HPD_DONE. Note that, after HPD_DONE has been asserted, it remains asserted until the next measurement has been commanded.

Note: A simpler, but less accurate, procedure for headphone impedance measurement is also supported, using the HPD_LVL field. When the HPD_DONE bit is set, indicating completion of a measurement, the impedance can be read directly from the HPD_LVL field, provided that the value lies within the range of the applicable HPD_ IMPEDANCE_RANGE setting.

Note that, for detection using the MICDET*n* or JACKDET*n* pins, the HPD_LVL field is the only supported measurement output option. The HPD_IMPEDANCE_RANGE field is not valid for detection on the MICDET*n* or JACKDET*n* pins. See Table 4-69 for further description of the HPD_LVL field.

The headphone detection function is an input to the interrupt control circuit and can be used to trigger an interrupt event on completion of the headphone detection; see Section 4.12.

The fields associated with headphone detection are described in Table 4-69. The external circuit configuration is shown Fig. 4-61.

Note that 32-bit register addressing is used from R12888 (0x3000) upwards; 16-bit format is used otherwise. The registers noted in Table 4-69 contain a mixture of 16- and 32-bit register addresses.

Table 4-69. Headphone Detect Control

Register Address	Bit	Label	Default	Description
R665 (0x0299) Headphone_ Detect_0	15	HPD_OVD_ ENA	0	Headphone Detect Output Override Enable This bit, when set, causes the HPD_OUT_SEL headphone output channel to be automatically configured for headphone detection each time headphone detection is scheduled. Note that the respective output driver must also be disabled (HP1x_ ENA = 0) for the duration of a headphone output impedance measurement. 0 = Disabled 1 = Enabled
	14:12	HPD_OUT_ SEL[2:0]	000	Headphone Detect Output Channel Select 000 = HPOUTL All other codes are reserved 001 = HPOUTR
	11:8	HPD_FRC_ SEL[3:0]	000	Headphone Detect Measurement Current Pin Select 0000 = MICDET1 0110 = JACKDET1 0001 = MICDET2 0111 = JACKDET2 0100 = HPOUTL All other codes are reserved 0101 = HPOUTR
	7:4	HPD_SENSE_ SEL[3:0]	0000	Headphone Detect Sense Pin Select 0000 = MICDET1 0110 = JACKDET1 0001 = MICDET2 0111 = JACKDET2 0100 = HPOUTL All other codes are reserved 0101 = HPOUTR
	2:0	HPD_GND_ SEL[2:0]	000	Headphone Detect Ground Pin Select 000 = MICDET1/HPOUTFB1 All other codes are reserved 001 = MICDET2/HPOUTFB2
R667 (0x029B) Headphone_ Detect_1	10:9	HPD_ IMPEDANCE_ RANGE[1:0]	00	Headphone Detect Range 00 = Reserved 01 = 0 Ω to 90 Ω 10 = 90 Ω to 1000 Ω 11 = 1 kΩ to 10 kΩ Only valid when HPD_SENSE_SEL = 0100 or 0101.
	4:3	HPD_CLK_ DIV[1:0]	00	Headphone Detect Clock Rate (Selects the clocking rate of the headphone detect adjustable current source. Decreasing the clock rate gives a slower measurement time.) 00 = 32 kHz 01 = 16 kHz 10 = 8 kHz 11 = 4 kHz
	2:1	HPD_ RATE[1:0]	00	Headphone Detect Sweep Rate (Selects the step size between successive measurements. Increasing the step size gives a faster measurement time.) 00 = 1 01 = 2 10 = 4 11 = Reserved
	0	HPD_POLL	0	Headphone Detect Enable Write 1 to start HP Detect function

Table 4-69. Headphone Detect Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R668 (0x029C) Headphone_ Detect_2	15	HPD_DONE	0	Headphone Detect Status 0 = HP Detect not complete 1 = HP Detect done
	14:0	HPD_ LVL[14:0]	0x0000	Headphone Detect Level For HPOUTL or HPOUTR measurement (HPD_SENSE_SEL = 0100 or 0101), HPD_ LVL is valid from 4 Ω to10 kΩ, within the range selected by HPD_IMPEDANCE_ RANGE. 74 = 4 Ω or less 75 = 4.5 Ω 76 = 5 Ω 77 = 5.5 Ω … 20,066 = 10 kΩ If HPD_LVL reports a value outside the valid range, the range should be adjusted and the measurement repeated. A 0-Ω result may be reported if the measurement is less than the minimum value for the selected range. For all other measurements, HPD_LVL is valid from 400 Ω to 6 kΩ only. 800 = 400 Ω or less 801 = 400.5 Ω 802 = 401 Ω 803 = 401.5 Ω … 12,000 = 6 kΩ
R669 (0x029D) Headphone_ Detect_3	9:0	HPD_ DACVAL[9:0]	0x000	Headphone Detect Level (Coded as integer, LSB = 1). See separate description for full decode information.
R131076 (0x20004) OTP_HPDET_Cal_ 1	31:24	HP_OFFSET_ 11[7:0]	See Footnote 1	Headphone Detect Calibration field. Signed number, LSB = 0.25. Range is –31.75 to +31.75. Default value is factory-set per device.
	23:16	HP_OFFSET_ 10[7:0]	See Footnote 1	Headphone Detect Calibration field. Signed number, LSB = 0.25. Range is –31.75 to +31.75. Default value is factory-set per device.
	15:8	HP_OFFSET_ 01[7:0]	See Footnote 1	Headphone Detect Calibration field. Signed number, LSB = 0.25. Range is –31.75 to +31.75. Default value is factory-set per device.
R131078 (0x20006) OTP_HPDET_Cal_ 2	15:8	HP_ GRADIENT_ 1X[7:0]	See Footnote 1	Headphone Detect Calibration field. Signed number, LSB = 0.25. Range is –31.75 to +31.75. Default value is factory-set per device.
	7:0	HP_ GRADIENT_ 0X[7:0]	See Footnote 1	Headphone Detect Calibration field. Signed number, LSB = 0.25. Range is –31.75 to +31.75. Default value is factory-set per device.

1. Default value is factory-set per device.

The external connections for the headphone detect circuit are shown in Fig. 4-61.

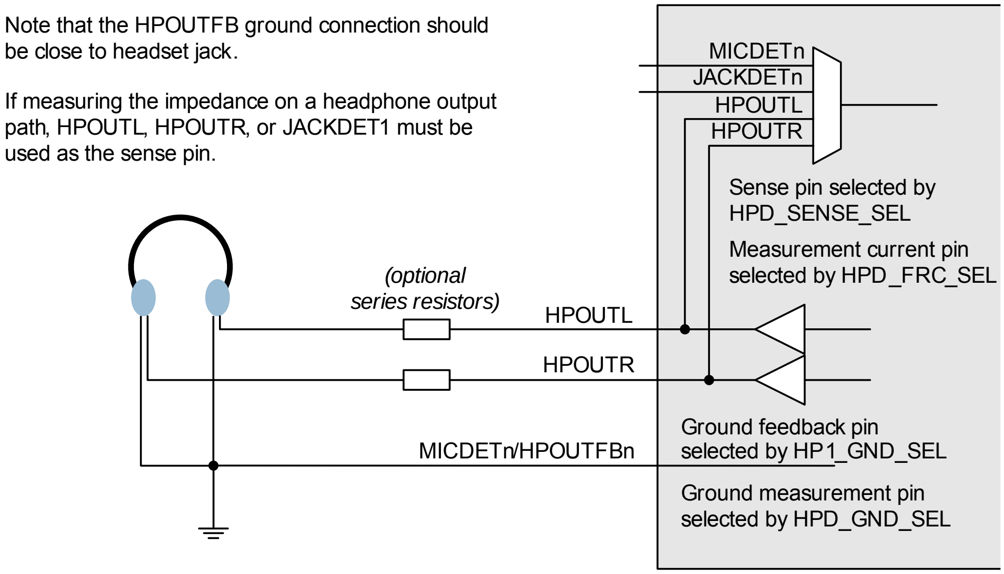


Figure 4-61. Headphone Detect Interface

Under default conditions, the measurement time varies between 17–244 ms, depending on the impedance of the external load. A high impedance is measured faster than a low impedance.

4.10 Low Power Sleep Configuration

The CS47L15 supports a low-power Sleep Mode, in which most functions are disabled and power consumption is minimized. The CS47L15 enters Sleep Mode when the DCVDD supply is removed. Note that the AVDD and DBVDD supplies must be present throughout the Sleep Mode duration.

In Sleep Mode, the CS47L15 can generate an interrupt event in response to a change in voltage on the JACKDET1 or JACKDET2 pins. This enables a jack insertion event (or other digital logic transition) to be used to trigger a wake-up of the CS47L15.

The system clocks (SYSCLK, DSPCLK) should be disabled before selecting Sleep Mode. The external clock input (MCLK*n*) may also be stopped, if desired.

The functionality and control fields associated with Sleep Mode are supported via an internal always-on supply domain.

The always-on control registers are listed in Table 4-70. These fields are maintained (i.e., not reset) in Sleep Mode.

Note that the control interface is not supported in Sleep Mode; read/write access to the always-on registers is not possible. Access to the register map using any of the control interfaces should be ceased before selecting Sleep Mode.

Table 4-70. Sleep Mode Always-On Control Registers

Register Address	Label	Reference
R710 (0x02C6)	MICD_CLAMP_OVD	See Section 4.9
	MICD_CLAMP_MODE[3:0]	
R723 (0x02D3)	JD2_ENA	
	JD1_ENA	

Table 4-70. Sleep Mode Always-On Control Registers *(Cont.)*

Register Address	Label	Reference
R6150 (0x1806)	MICD_CLAMP_FALL_EINT1	See Section 4.12
	MICD_CLAMP_RISE_EINT1	
	JD2_FALL_EINT1	
	JD2_RISE_EINT1	
	JD1_FALL_EINT1	
	JD1_RISE_EINT1	
R6214 (0x1846)	IM_MICD_CLAMP_FALL_EINT1	
	IM_MICD_CLAMP_RISE_EINT1	
	IM_JD2_FALL_EINT1	
	IM_JD2_RISE_EINT1	
	IM_JD1_FALL_EINT1	
	IM_JD1_RISE_EINT1	
R6784 (0x1A80)	IM_IRQ1	
	IRQ_POL	
	IRQ_OP_CFG	
R6864 (0x1AD0)	RESET_PU	See Section 4.19
	RESET_PD	

The always-on digital I/O pins are listed in Table 4-71. All other digital input pins have no effect in Sleep Mode; all other digital output pins are undriven (floating).

The $\overline{\text{IRQ}}$ output is normally deasserted in Sleep Mode. In Sleep Mode, the $\overline{\text{IRQ}}$ output can be asserted only in response to the JACKDET1 or JACKDET2 inputs. If the $\overline{\text{IRQ}}$ output is asserted in Sleep Mode, it can be deasserted only after a wake-up transition.

Output drivers and bus keepers are disabled in Sleep Mode, for all pins not on the always-on domain; this means that the logic level on these pins is undefined. If a defined logic state is required during Sleep Mode (e.g., as input to another device), an external pull resistor may be required. If an external pull resistor is connected to a pin that also supports a bus keeper function, the pull resistance should be chosen carefully, taking into account the resistance of the bus keeper. See Section 4.11.1 for specific notes concerning the GPIO pins.

Table 4-71. Sleep Mode Always-On Digital Input/Output Pins

Pin Name	Description	Reference
$\overline{\text{IRQ}}$	Interrupt Request output	See Section 4.12
JACKDET1	Jack Detect input 1	See Section 4.9
JACKDET2	Jack Detect input 2	See Section 4.9
$\overline{\text{RESET}}$	Digital Reset input (active low)	See Section 4.19

The always-on functionality includes the JD1 and JD2 control signals, which provide support for the low-power Sleep Mode. The MICDET clamp status signal is also supported; this is controlled by a selectable logic function, derived from JD1 and/or JD2.

The JD1, JD2 and MICDET clamp status signals are derived from the JACKDET1 and JACKDET2 inputs, and can be used to trigger the interrupt controller.

- The JD1 and JD2 signals are derived from the jack detect function (see Section 4.9). These inputs can be used to trigger a response to a jack insertion or jack removal detection.

  When these signals are enabled, the JD1 and JD2 signals indicate the status of the JACKDET1 and JACKDET2 input pins respectively. See Table 4-63 for details of the associated control fields.
- The MICDET clamp status is controlled by the JD1 and/or JD2 signals (see Section 4.9). The configurable logic provides flexibility in selecting the appropriate conditions for activating the MICDET clamp. The clamp status can be used to trigger a response to a jack insertion or jack removal detection.

  The MICDET clamp function is configured using MICD_CLAMP_MODE, as described in Table 4-64.

The interrupt functionality associated with these signals is part of the always-on functionality, enabling the CS47L15 to provide indication of jack insertion or jack removal to the host processor in Sleep Mode; see Section 4.12.

Note that the JACKDET1 and JACKDET2 inputs do not result in a wake-up transition directly; a wake-up transition only occurs by reapplication of DCVDD. In a typical application, the JACKDET*n* inputs provide a signal to the applications processor, via the $\overline{\text{IRQ}}$ output; if a wake-up transition is required, this is triggered by the applications processor enabling the DCVDD supply.

4.11 General-Purpose I/O

The CS47L15 provides a number of GPIO functions to enable interfacing and detection of external hardware and to provide logic outputs to other devices. The GPIO input functions can be used to generate an interrupt (IRQ) event. The GPIO and interrupt circuits support the following functions:

- Pin-specific alternative functions for external interfaces (AIF, PDM)
- Logic input/button detect (GPIO input)
- Logic 1 and Logic 0 output (GPIO output)
- Interrupt (IRQ) status output
- Clock output
- Frequency-locked loop (FLL) status output
- FLL clock output
- IEC-60958-3–compatible S/PDIF output
- Pulse-width modulation (PWM) signal output
- Overtemperature, speaker short-circuit protection, and speaker shutdown status output
- General-purpose timer status output
- Event logger FIFO buffer status output

Logic input and output (GPIO) can be supported in two different ways on the CS47L15. The standard mechanism described in this section provides a comprehensive suite of options including input debounce, and selectable output drive configuration. The DSP GPIO circuit is tailored towards more advanced requirements typically demanded by DSP software features. The DSP GPIO functions are described in Section 4.5.3.

The CS47L15 also incorporates a general-purpose switch feature, which can be used as a controllable analog switch, as described in Section 4.11.16.

If the master-boot function is selected, the GPIO13 and GPIO14 pins support an I^2C control interface that provides read/write access to the CS47L15 control registers. If the I^2C control interface is enabled, the respective GPIO configuration registers have no effect and the GPIO pins cannot be assigned any other function. See Section 4.14 for details of the master-boot function.

If the JTAG interface is enabled, the GPIO5 and GPIO9–11 pins are configured as a JTAG interface that provides test and debug access to the CS47L15 DSP core. The respective GPIO configuration registers have no effect in this case, and the GPIO pins cannot be assigned any other function. See Section 4.17 for details of the JTAG interface.

4.11.1 GPIO Control

For each GPIO, the selected function is determined by the GP*n*_FN field, where *n* identifies the GPIO pin (1–15). The pin direction, set by GP*n*_DIR, must be set according to function selected by GP*n*_FN.

If a pin is configured as a GPIO input (GP*n*_DIR = 1, GP*n*_FN = 0x001), the logic level at the pin can be read from the respective GP*n*_LVL bit. Note that GP*n*_LVL is not affected by the GP*n*_POL bit.

A debounce circuit can be enabled on any GPIO input, to avoid false event triggers. This is enabled on each pin by setting the respective GP*n*_DB bit. The debounce circuit uses the 32-kHz clock, which must be enabled whenever input debounce functions are required. The debounce time is configurable using the GP_DBTIME field. See Section 4.13 for further details of the CS47L15 clocking configuration.

Each of the GPIO pins is an input to the interrupt control circuit and can be used to trigger an interrupt event. An interrupt event is triggered on the rising and falling edges of the GPIO input. The associated interrupt bit is latched once set; it can be polled at any time or used to control the IRQ signal. See Section 4.12 for details of the interrupt event handling.

Integrated pull-up and pull-down resistors are provided on each of the GPIO pins; these can be configured independently using the GP*n*_PU and GP*n*_PD fields. When the pull-up and pull-down control bits are both enabled, the CS47L15 provides a bus keeper function on the respective pin. The bus keeper function holds the logic level unchanged whenever the pin is undriven (e.g., if the signal is tristated).

Note: The bus keeper is enabled by default on all GPIO pins and, if not actively driven, may result in either a Logic 0 or Logic 1 at the respective input on start-up. If an external pull resistor is connected (e.g., to control the logic level in Sleep Mode), the chosen resistance should take account of the bus keeper resistance (see Table 3-10). A strong pull resistor (e.g., 10 kΩ) is required, if a specific start-up condition is to be forced by the external pull component.

If a pin is configured as a GPIO output (GP*n*_DIR = 0, GP*n*_FN = 0x001), its level can be set to Logic 0 or Logic 1 using the GP*n*_LVL field. Note that the GP*n*_LVL bits are write-only when the respective GPIO pin is configured as an output.

If a pin is configured as an output (GP*n*_DIR = 0), the polarity can be inverted using the GP*n*_POL bit. When GP*n*_POL = 1, the selected output function is inverted. In the case of logic level output (GP*n*_FN = 0x001), the external output is the opposite logic level to GP*n*_LVL when GP*n*_POL = 1. Note that, if GP*n*_FN = 0x000 or 0x002, the GP*n*_POL bit has no effect on the respective GPIO pin.

A GPIO output can be either CMOS driven or open drain. This is selected on each pin using the respective GP*n*_OP_CFG bit. Note that if GP*n*_FN = 0x000 or 0x002, the GP*n*_OP_CFG bit has no effect on the respective GPIO pin—the respective pin output is CMOS in this case.

The register fields that control the GPIO pins are described in Table 4-72.

Table 4-72. GPIO Control

Register Address	Bit	Label	Default	Description
R5888 (0x1700) GPIO1_CTRL_1 to R5916 (0x171C) GPIO15_CTRL_1	15	GP*n*_LVL	See Footnote 2	GPIO*n* level. Write to this bit to set a GPIO output. Read from this bit to read GPIO input level. For output functions only, if GP*n*_POL is set, the GP*n*_LVL bit is the opposite logic level to the external pin. Note that, if GP*n*_DIR = 0, the GP*n*_LVL bit is write-only.
	14	GP*n*_OP_CFG	0	GPIO*n* Output Configuration 0 = CMOS 1 = Open drain Note that, if GP*n*_FN = 0x000 or 0x002, this bit has no effect on the GPIO*n* output. If GPn_FN = 0x000, the pin configuration is set according to the applicable pin-specific function (see Table 4-74). If GPn_FN = 0x002, the pin configuration is CMOS.
	13	GP*n*_DB	1	GPIO*n* Input Debounce 0 = Disabled 1 = Enabled
	12	GP*n*_POL	0	GPIO*n* Output Polarity Select 0 = Noninverted (Active High) 1 = Inverted (Active Low) Note that, if GP*n*_FN = 0x000 or 0x002, this bit has no effect on the GPIO*n* output.
	8:0	GP*n*_FN[8:0]	0x001	GPIO*n* Pin Function (see Table 4-73 for details)

Table 4-72. GPIO Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R5889 (0x1701) GPIO1_CTRL_2 to R5917 (0x171D) GPIO15_CTRL_2	15	GP*n*_DIR	1	GPIO*n* Pin Direction 0 = Output 1 = Input The GPn_DIR bit has no effect if GPn_FN = 0x000 or 0x002. If GPn_FN = 0x000, the pin direction is set according to the applicable pin-specific function (see Table 4-74). If GPn_FN = 0x002, the pin direction is set according to the DSP GPIO configuration.
	14	GP*n*_PU	1	GPIO*n* Pull-Up Enable 0 = Disabled 1 = Enabled **Note:** If GP*n*_PD and GP*n*_PU are both set, a bus keeper function is enabled on the respective GPIO*n* pin.
	13	GP*n*_PD	1	GPIO*n* Pull-Down Enable 0 = Disabled 1 = Enabled **Note:** If GP*n*_PD and GP*n*_PU are both set, a bus keeper function is enabled on the respective GPIO*n* pin.
R6848 (0x1AC0) GPIO_Debounce_ Config	3:0	GP_DBTIME[3:0]	0000	GPIO Input debounce time 0x0 = 100 µs 0x1 = 1.5 ms 0x2 = 3 ms 0x3 = 6 ms 0x4 = 12 ms 0x5 = 24 ms 0x6 = 48 ms 0x7 = 96 ms 0x8 = 192 ms 0x9 = 384 ms 0xA = 768 ms 0xB to 0xF = Reserved

1. *n* is a number (1–15) that identifies the individual GPIO.
2. The default value of GP*n*_LVL depends upon whether the pin is actively driven by another device. If the pin is actively driven, the bus keeper maintains this logic level. If the pin is not actively driven, the bus keeper may establish either a Logic 1 or Logic 0 as the initial input level.

4.11.2 GPIO Function Select

The available GPIO functions are described in Table 4-73. The function of each GPIO is set using GP*n*_FN, where *n* identifies the GPIO pin (1–15). Note that the respective GP*n*_DIR must also be set according to whether the function is an input or output.

Table 4-73. GPIO Function Select

GP*n*_FN	Description	Comments
0x000	Pin-specific alternate function	Alternate functions supporting digital microphone, digital audio interface, master control interface, and PDM output functions.
0x001	Button-detect input/logic-level output	GP*n*_DIR = 0: GPIO pin logic level is set by GP*n*_LVL. GP*n*_DIR = 1: Button detect or logic level input.
0x002	DSP GPIO	Low latency input/output for DSP functions.
0x003	IRQ1 output	Interrupt (IRQ1) output 0 = IRQ1 not asserted 1 = IRQ1 asserted
0x004	IRQ2 output	Interrupt (IRQ2) output 0 = IRQ2 not asserted 1 = IRQ2 asserted
0x010	FLL1 clock	Clock output from FLL1
0x013	FLL_AO clock	Clock output from FLL_AO

Table 4-73. GPIO Function Select *(Cont.)*

GP*n*_FN	Description	Comments
0x018	FLL1 lock	Indicates FLL1 lock status 0 = Not locked 1 = Locked
0x01B	FLL_AO lock	Indicates FLL_AO lock status 0 = Not locked 1 = Locked
0x040	OPCLK clock output	Configurable clock output derived from SYSCLK
0x048	PWM1 output	Configurable PWM output PWM1
0x049	PWM2 output	Configurable PWM output PWM2
0x04C	S/PDIF output	IEC-60958-3–compatible S/PDIF output
0x0B6	SPKOUTL short circuit status	SPKOUT short circuit status 0 = Normal 1 = Short Circuit detected
0x0E0	Speaker shutdown status	Speaker shutdown status 0 = Normal 1 = Speaker shutdown completed (due to overheat temperature, short-circuit protection, or general-purpose timer condition)
0x0E1	Speaker overheat shutdown	Indicates shutdown temperature status 0 = Temperature is below shutdown level 1 = Temperature is above shutdown level
0x0E2	Speaker overheat warning	Indicates warning temperature status 0 = Temperature is below warning level 1 = Temperature is above warning level
0x140	Timer 1 status	Timer 1 status A pulse is output after the timer reaches its final count value.
0x141	Timer 2 status	Timer 2 status A pulse is output after the timer reaches its final count value.
0x150	Event Log 1 FIFO not-empty status	Event Log 1 FIFO Not-Empty status 0 = FIFO Empty 1 = FIFO Not Empty
0x151	Event Log 2 FIFO not-empty status	Event Log 2 FIFO Not-Empty status 0 = FIFO Empty 1 = FIFO Not Empty

4.11.3 Pin-Specific Alternative Function—GP*n*_FN = 0x000

The CS47L15 GPIO capability is multiplexed with the pin-specific functions listed in Table 4-74. The alternate functions are selected by setting the respective GP*n*_FN fields to 0x000, as described in Section 4.11.1. Note that each function is unique to the associated pin and can be supported only on that pin.

If the alternate function is selected on a GPIO pin, the pin direction (input or output) and the output driver configuration (CMOS or open drain) are set automatically as described in Table 4-74. The respective GP*n*_DIR and GP*n*_OP_CFG bits have no effect in this case.

Table 4-74. GPIO Alternate Functions

Name	Condition	Description	Direction	Output Driver Configuration
AIF1BCLK/GPIO3	GP3_FN = 0x000	Audio Interface 1 bit clock	Digital I/O	CMOS
AIF1LRCLK/GPIO4	GP4_FN = 0x000	Audio Interface 1 left/right clock	Digital I/O	CMOS
AIF1RXDAT/GPIO2	GP2_FN = 0x000	Audio Interface 1 RX digital audio data	Digital input	—
AIF1TXDAT/GPIO1	GP1_FN = 0x000	Audio Interface 1 TX digital audio data	Digital output	CMOS
AIF2BCLK/GPIO7	GP7_FN = 0x000	Audio Interface 2 bit clock	Digital I/O	CMOS
AIF2LRCLK/GPIO8	GP8_FN = 0x000	Audio Interface 2 left/right clock	Digital I/O	CMOS
AIF2RXDAT/GPIO6	GP6_FN = 0x000	Audio Interface 2 RX digital audio data	Digital input	—
AIF2TXDAT/GPIO5	GP5_FN = 0x000	Audio Interface 2 TX digital audio data	Digital output	CMOS
AIF3BCLK/GPIO11	GP11_FN = 0x000	Audio Interface 3 bit clock	Digital I/O	CMOS

Table 4-74. GPIO Alternate Functions *(Cont.)*

Name	Condition	Description	Direction	Output Driver Configuration
AIF3LRCLK/GPIO12	GP12_FN = 0x000	Audio Interface 3 left/right clock	Digital I/O	CMOS
AIF3RXDAT/GPIO10	GP10_FN = 0x000	Audio Interface 3 RX digital audio data	Digital input	—
AIF3TXDAT/GPIO9	GP9_FN = 0x000	Audio Interface 3 TX digital audio data	Digital output	CMOS
SPKCLK/GPIO14	GP14_FN = 0x000	Digital speaker (PDM) clock	Digital output	CMOS
SPKTXDAT/GPIO13	GP13_FN = 0x000	Digital speaker (PDM) TX data	Digital output	CMOS
SPKRXDAT/GPIO15	GP15_FN = 0x000	Digital speaker (PDM) RX data	Digital input	—

If the master-boot function is selected, the GPIO13 and GPIO14 pins support an I²C control interface that provides read/write access to the CS47L15 control registers. If the I²C control interface is enabled, the respective GPIO configuration registers have no effect and the GPIO pins cannot be assigned any other function. See Section 4.14 for details of the master-boot function.

If the JTAG interface is enabled, the GPIO5 and GPIO9–11 pins are configured as a JTAG interface. In this case, the respective GPIO configuration registers have no effect, and the GPIO pins cannot be assigned any other function. See Section 4.17 for details of the JTAG interface.

4.11.4 Button Detect (GPIO Input)—GP*n*_FN = 0x001

Button-detect functionality can be selected on a GPIO pin by setting the respective GPIO fields as described in Section 4.11.1. The same functionality can be used to support a jack-detect input function.

It is recommended to enable the GPIO input debounce feature when using GPIOs as button input or jack-detect input.

The GP*n*_LVL fields may be read to determine the logic levels on a GPIO input, after the selectable debounce controls. Note that GP*n*_LVL is not affected by the GP*n*_POL bit.

The debounced GPIO signals are also inputs to the interrupt-control circuit. An interrupt event is triggered on the rising and falling edges of the GPIO input. The associated interrupt bits are latched once set; they can be polled at any time or used to control the IRQ signal. See Section 4.12 for details of the interrupt event handling.

4.11.5 Logic 1 and Logic 0 Output (GPIO Output)—GP*n*_FN = 0x001

The CS47L15 can be programmed to drive a logic high or logic low level on a GPIO pin by selecting the GPIO Output function as described in Section 4.11.1.

The output logic level is selected using the respective GP*n*_LVL bit. Note that, if a GPIO pin is configured as an output, the respective GP*n*_LVL bits are write-only.

The polarity of the GPIO output can be inverted using the GP*n*_POL bits. If GP*n*_POL = 1, the external output is the opposite logic level to GP*n*_LVL.

4.11.6 DSP GPIO (Low-Latency DSP Input/Output)—GP*n*_FN = 0x002

The DSP GPIO function provides an advanced I/O capability for signal-processing applications. The DSP GPIO pins are accessed using maskable sets of I/O control registers; this allows the selected combinations of GPIOs to be controlled with ease, regardless of how the allocation of GPIO pins has been implemented in hardware.

The DSP GPIO function is selected by setting the respective GPIO fields as described in Section 4.11.1.

A full description of the DSP GPIO function is provided in Section 4.5.3.

Note that, if GP*n*_FN is set to 0x002, the respective pin direction (input or output) is set according to the DSP GPIO configuration for that pin—the GP*n*_DIR control bit has no effect in this case.

4.11.7 Interrupt (IRQ) Status Output—GP*n*_FN = 0x003, 0x004

The CS47L15 has an interrupt controller, which can be used to indicate when any selected interrupt events occur. Individual interrupts may be masked in order to configure the interrupt as required. See Section 4.12 for full definition of all supported interrupt events.

The interrupt controller supports two separate interrupt request (IRQ) outputs. The IRQ1 or IRQ2 status may be output directly on a GPIO pin by setting the respective GPIO fields as described in Section 4.11.1.

Note that the IRQ1 status is output on the $\overline{\text{IRQ}}$ pin at all times.

4.11.8 Frequency-Locked Loop (FLL) Clock Output—GP*n*_FN = 0x010, 0x013

Clock outputs derived from the FLLs may be output on a GPIO pin. The GPIO output from each FLL (FLL1 or FLL_AO) is controlled by the respective FLL*n*_GPCLK_DIV and FLL*n*_GPCLK_ENA fields, as described in Table 4-75.

It is recommended to disable the clock output (FLL*n*_GPCLK_ENA = 0) before making any change to the respective FLL*n*_GPCLK_DIV field.

Note that FLL*n*_GPCLK_DIV and FLL*n*_GPCLK_ENA affect the GPIO outputs only; they do not affect the FLL frequency. The maximum output frequency supported for GPIO output is noted in Table 3-10.

The FLL clock outputs may be output directly on a GPIO pin by setting the respective GPIO fields as described in Section 4.11.1.

See Section 4.13 for details of the CS47L15 system clocking and how to configure the FLLs.

Table 4-75. FLL Clock Output Control

Register Address	Bit	Label	Default	Description
R394 (0x018A) FLL1_GPIO_Clock	7:1	FLL1_GPCLK_ DIV[6:0]	0x02	FLL1 GPIO Clock Divider 0x00 = Reserved 0x01 = Reserved 0x02 = Divide by 2 0x03 = Divide by 3 0x04 = Divide by 4 … 0x7F = Divide by 127 (F_{GPIO} = F_{VCO}/FLL1_GPCLK_DIV)
	0	FLL1_GPCLK_ ENA	0	FLL1 GPIO Clock Enable 0 = Disabled 1 = Enabled
R490 (0x01EA) FLL_AO_GPIO_ Clock	7:1	FLL_AO_GPCLK_ DIV[6:0]	0x01	FLL_AO GPIO Clock Divider 0x00 = Divide by 1 0x01 = Divide by 1 0x02 = Divide by 2 0x03 = Divide by 3 0x04 = Divide by 4 … 0x7F = Divide by 127 (F_{GPIO} = F_{VCO}/FLL_AO_GPCLK_DIV)
	0	FLL_AO_GPCLK_ ENA	0	FLL_AO GPIO Clock Enable 0 = Disabled 1 = Enabled

4.11.9 Frequency-Locked Loop (FLL) Status Output—GP*n*_FN = 0x018, 0x01B

The CS47L15 provides FLL status flags, which may be used to control other events. The FLL lock signals indicate whether FLL lock has been achieved. See Section 4.13.8 and Section 4.13.9 for details of the FLLs.

The FLL lock signals may be output directly on a GPIO pin by setting the respective GPIO fields as described in Section 4.11.1.

The FLL lock signals are inputs to the interrupt controller circuit. An interrupt event is triggered on the rising edge of these signals. The associated interrupt bits are latched once set; they can be polled at any time or used to control the IRQ signal. See Section 4.12 for details of the interrupt event handling.

4.11.10 OPCLK Clock Output—GP*n*_FN = 0x040

A clock output (OPCLK) derived from SYSCLK can be output on a GPIO pin. The OPCLK frequency is controlled by OPCLK_DIV and OPCLK_SEL. The OPCLK output is enabled by setting OPCLK_ENA, as described in Table 4-76.

It is recommended to disable the clock output (OPCLK_ENA = 0) before making any change to OPCLK_DIV or OPCLK_SEL.

The OPCLK clock can be output directly on a GPIO pin by setting the respective GPIO fields as described in Section 4.11.1.

Note that the OPCLK source frequency cannot be higher than the SYSCLK frequency. The maximum output frequency supported for GPIO output is noted in Table 3-10.

See Section 4.13 for details of the SYSCLK system clock.

Table 4-76. OPCLK Control

Register Address	Bit	Label	Default	Description
R329 (0x0149) Output_system_clock	15	OPCLK_ENA	0	OPCLK Enable 0 = Disabled 1 = Enabled
	7:3	OPCLK_DIV[4:0]	0x00	OPCLK Divider 0x02 = Divide by 2 0x04 = Divide by 4 0x06 = Divide by 6 … (even numbers only) 0x1E = Divide by 30 Note that only even numbered divisions (2, 4, 6, etc.) are valid selections. All other codes are reserved when the OPCLK signal is enabled.
	2:0	OPCLK_SEL[2:0]	000	OPCLK Source Frequency 000 = 6.144 MHz (5.6448 MHz) 001 = 12.288 MHz (11.2896 MHz) 010 = 24.576 MHz (22.5792 MHz) 011 = 49.152 MHz (45.1584 MHz) All other codes are reserved The frequencies in brackets apply for 44.1 kHz–related SYSCLK rates only (i.e., SAMPLE_RATE_*n* = 01XXX). The OPCLK Source Frequency must be less than or equal to the SYSCLK frequency.

4.11.11 Pulse-Width Modulation (PWM) Signal Output—GP*n*_FN = 0x048, 0x049

The CS47L15 incorporates two PWM signal generators, which can be enabled as GPIO outputs. The duty cycle of each PWM signal can be modulated by an audio source, or can be set to a fixed value using a control register setting.

The PWM outputs may be output directly on a GPIO pin by setting the respective GPIO fields as described in Section 4.11.1.

See Section 4.3.12 for details of how to configure the PWM signal generators.

4.11.12 S/PDIF Audio Output—GP*n*_FN = 0x04C

The CS47L15 incorporates an IEC-60958-3–compatible S/PDIF transmitter, which can be selected as a GPIO output. The S/PDIF transmitter supports stereo audio channels and allows full control over the S/PDIF validity bits and channel status information.

The S/PDIF signal may be output directly on a GPIO pin by setting the respective GPIO fields as described in Section 4.11.1.

See Section 4.3.8 for details of how to configure the S/PDIF output generator.

4.11.13 Overtemperature, Short-Circuit Protection, and Speaker Shutdown Status Output—GP*n*_FN = 0x0B6, 0x0E0, 0x0E1, 0x0E2.

The CS47L15 incorporates a temperature sensor, which detects when the device temperature is within normal limits or if the device is approaching a hazardous temperature condition.

The temperature status may be output directly on a GPIO pin by setting the respective GPIO fields as described in Section 4.11.1. A GPIO pin can be used to indicate either an Overheat Warning Temperature event or an Overheat Shutdown Temperature event.

The CS47L15 provides short-circuit protection on the Class D speaker outputs, and on each of the headphone output paths.

The status of the Class D speaker short-circuit detection circuits may be output directly on a GPIO pin by setting the respective GPIO fields as described in Section 4.11.1.

If the Overheat Shutdown Temperature is exceeded, or if a short circuit is detected on the Class D speaker outputs, the Class D speaker outputs are automatically disabled in order to protect the device. The general-purpose timers can be used as a watchdog function to trigger a shutdown of the Class D speaker drivers. Further details of the Speaker Shutdown functions are described in Section 4.18. When the speaker driver shutdown is complete, the Speaker Shutdown signal is asserted. The speaker driver shutdown status can also be output directly on a GPIO pin.

The Overtemperature, short-circuit protection, and Speaker Shutdown status flags are inputs to the interrupt control circuit. An interrupt event may be triggered on the applicable edges of these signals. The associated interrupt bit is latched once set; it can be polled at any time or used to control the IRQ signal. See Section 4.12 for details of the interrupt event handling.

4.11.14 General-Purpose Timer Status Output—GP*n*_FN = 0x140, 0x141

The general-purpose timers can count up or down, and support continuous or single count modes. Status outputs indicating the progress of these timers are provided. See Section 4.5.2 for details of the general-purpose timers.

A logic signal from the general-purpose timers may be output directly on a GPIO pin by setting the respective GPIO fields as described in Section 4.11.1. This logic signal is pulsed high whenever the respective timer reaches its final count value.

The general-purpose timers also provide inputs to the interrupt control circuit. An interrupt event is triggered whenever the respective timer reaches its final count value. The associated interrupt bits are latched once set; they can be polled at any time or used to control the IRQ signal. See Section 4.12 for details of the interrupt event handling.

4.11.15 Event Logger FIFO Buffer Status Output—GP*n*_FN = 0x150, 0x151

The event loggers are each provided with a 16-stage FIFO buffer, in which any detected events (signal transitions) are recorded. Status outputs for each FIFO buffer are provided. See Section 4.5.1 for details of the event loggers.

A logic signal from the event loggers may be output directly on a GPIO pin by setting the respective GPIO fields as described in Section 4.11.1. This logic signal is set high whenever the FIFO not-empty condition is true.

The event loggers also provide inputs to the interrupt control circuit. An interrupt event is triggered whenever the respective FIFO condition occurs. The associated interrupt bits are latched once set; they can be polled at any time or used to control the $\overline{\text{IRQ}}$ signal. See Section 4.12 for details of the interrupt event handling.

4.11.16 General-Purpose Switch

The CS47L15 provides a general-purpose switch, which can be used as a controllable analog switch for external functions. The switch is implemented between the GPSWP and GPSWN pins. Note that this feature is entirely independent of the GPIO*n* pins.

The general-purpose switch is configured using SW1_MODE. This field allows the switch to be disabled, enabled, or synchronized to the MICDET clamp status, as described in Table 4-77.

The switch is a bidirectional analog switch, offering flexibility in the potential circuit applications. Refer to Table 3-2 and Table 3-10 for further details.

The switch can be used in conjunction with the MICDET clamp function to suppress pops and clicks associated with jack insertion and removal. An example circuit is shown in Fig. 4-58 within the External Accessory Detection section. Note that the MICDET clamp function must also be configured appropriately when using this method of pop suppression.

Table 4-77. General-Purpose Switch Control

Register Address	Bit	Label	Default	Description
R712 (0x02C8) GP_Switch_1	1:0	SW1_MODE[1:0]	00	General-purpose Switch control 00 = Disabled (open) 01 = Enabled (closed) 10 = Enabled when MICDET clamp is active 11 = Enabled when MICDET clamp is not active

4.12 Interrupts

The interrupt controller has multiple inputs. These include the jack detect and GPIO input pins, DSP_IRQn flags, headphone/accessory detection, FLL lock detection, and status flags from DSP peripheral functions. See Table 4-78 and Table 4-79 for a full definition of the interrupt controller inputs. Any combination of these inputs can be used to trigger an interrupt request event.

The interrupt controller supports two sets of interrupt registers. This allows two separate interrupt request (IRQ) outputs to be generated, and for each IRQ to report a different set of input or status conditions.

For each interrupt request (IRQ1 and IRQ2) output, there is an interrupt register field associated with each of the interrupt inputs. These fields are asserted whenever a logic edge is detected on the respective input. Some inputs are triggered on rising edges only; some are triggered on both edges. Separate rising and falling interrupt bits are provided for the JD1 and JD2 signals. The interrupt register fields for IRQ1 are described in Table 4-78. The interrupt register fields for IRQ2 are described in Table 4-79. The interrupt flags can be polled at any time or in response to the interrupt request output being signaled via the $\overline{\text{IRQ}}$ pin or a GPIO pin.

All interrupts are edge triggered, as noted above. Many are triggered on both the rising and falling edges and, therefore, the interrupt bits cannot indicate which edge has been detected. The raw status fields described in Table 4-78 and Table 4-79 indicate the current value of the corresponding inputs to the interrupt controller. Note that the raw status bits associated with IRQ1 and IRQ2 provide the same information. The status of any GPIO (or DSP GPIO) inputs can also be read using the GPIO (or DSP GPIO) control fields, as described in Table 4-72 and Table 4-37.

Individual mask bits can enable or disable different functions from the interrupt controller. The mask bits are described in Table 4-78 (for IRQ1) and Table 4-79 (for IRQ2). Note that a masked interrupt input does not assert the corresponding interrupt register field and does not cause the associated interrupt request output to be asserted.

The interrupt request outputs represent the logical OR of the associated interrupt registers. IRQ1 is derived from the x_ EINT1 registers; IRQ2 is derived from the x_EINT2 registers. The interrupt register fields are latching fields and, once they are set, they are not reset until a 1 is written to the respective bits. The interrupt request outputs are not reset until each of the associated interrupts has been reset.

A debounce circuit can be enabled on any GPIO input, to avoid false event triggers. This is enabled on each pin using the fields described in Table 4-72. The GPIO debounce circuit uses the 32-kHz clock, which must be enabled whenever the GPIO debounce function is required.

The IRQ outputs can be globally masked using the IM_IRQ1 and IM_IRQ2 bits. When not masked, the IRQ status can be read from IRQ1_STS and IRQ2_STS for the respective IRQ outputs.

The IRQ1 output is provided externally on the $\overline{\text{IRQ}}$ pin. Under default conditions, this output is active low. The polarity can be inverted using IRQ_POL. The $\overline{\text{IRQ}}$ output can be either CMOS driven or open drain; this is selected using the IRQ_ OP_CFG bit.

The IRQ2 status can be used to trigger DSP firmware execution; see Section 4.4. This allows the DSP firmware execution to be linked to external events (e.g., jack detection, or GPIO input), or to any of the status conditions flagged by the interrupt registers.

The IRQ1 and IRQ2 signals may be output on a GPIO pin; see Section 4.11.

The CS47L15 interrupt controller circuit is shown in Fig. 4-62. (Note that not all interrupt inputs are shown.) The control fields associated with IRQ1 and IRQ2 are described in Table 4-78 and Table 4-79 respectively. The global interrupt mask bits, status bits, and output configuration fields are described Table 4-80.

Note that, under default register conditions, the boot done status is the only unmasked interrupt source; a falling edge on the $\overline{\text{IRQ}}$ pin indicates completion of the boot sequence.

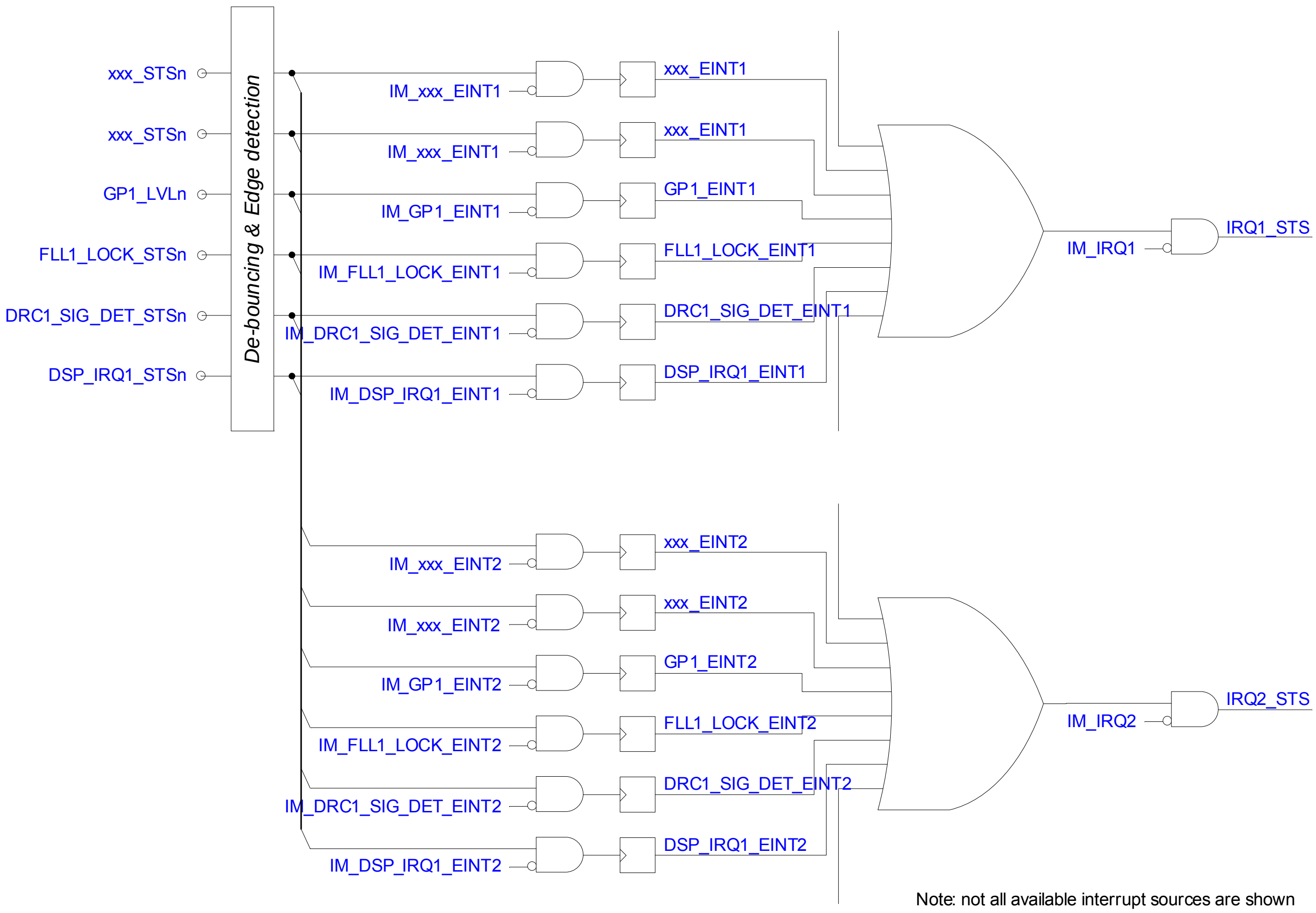


Figure 4-62. Interrupt Controller

The IRQ1 interrupt, mask, and status control registers are described in Table 4-78.

Table 4-78. Interrupt 1 Control Registers

Register Address	Bit	Label	Default	Description
R6144 (0x1800) IRQ1_Status_1	12	CTRLIF_ERR_EINT1	0	Control Interface Error Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	9	SYSCLK_FAIL_EINT1	0	SYSCLK Fail Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	7	BOOT_DONE_EINT1	0	Boot Done Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.

Table 4-78. Interrupt 1 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6145 (0x1801) IRQ1_Status_2	15	FLL_AO_REF_LOST_EINT1	0	FLL_AO Reference Lost Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	14	DSPCLK_ERR_EINT1	0	DSPCLK Error Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	12	SYSCLK_ERR_EINT1	0	SYSCLK Error Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	11	FLL_AO_LOCK_EINT1	0	FLL_AO Lock Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	8	FLL1_LOCK_EINT1	0	FLL1 Lock Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6149 (0x1805) IRQ1_Status_6	9	MICDET2_EINT1	0	Mic/Accessory Detect 2 Interrupt (Detection event triggered) Note: Cleared when a 1 is written.
	8	MICDET1_EINT1	0	Mic/Accessory Detect 1 Interrupt (Detection event triggered) Note: Cleared when a 1 is written.
	0	HPDET_EINT1	0	Headphone Detect Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6150 (0x1806) IRQ1_Status_7	5	MICD_CLAMP_FALL_EINT1	0	MICDET Clamp Interrupt (Falling edge triggered) Note: Cleared when a 1 is written.
	4	MICD_CLAMP_RISE_EINT1	0	MICDET Clamp Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	3	JD2_FALL_EINT1	0	JD2 Interrupt (Falling edge triggered) Note: Cleared when a 1 is written.
	2	JD2_RISE_EINT1	0	JD2 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	JD1_FALL_EINT1	0	JD1 Interrupt (Falling edge triggered) Note: Cleared when a 1 is written.
	0	JD1_RISE_EINT1	0	JD1 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6152 (0x1808) IRQ1_Status_9	2	INPUTS_SIG_DET_EINT1	0	Input Path Signal-Detect Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	1	DRC2_SIG_DET_EINT1	0	DRC2 Signal-Detect Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	0	DRC1_SIG_DET_EINT1	0	DRC1 Signal-Detect Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.

Table 4-78. Interrupt 1 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6154 (0x180A) IRQ1_Status_11	15	DSP_IRQ16_EINT1	0	DSP IRQ16 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	14	DSP_IRQ15_EINT1	0	DSP IRQ15 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	13	DSP_IRQ14_EINT1	0	DSP IRQ14 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	12	DSP_IRQ13_EINT1	0	DSP IRQ13 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	11	DSP_IRQ12_EINT1	0	DSP IRQ12 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	10	DSP_IRQ11_EINT1	0	DSP IRQ11 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	9	DSP_IRQ10_EINT1	0	DSP IRQ10 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	8	DSP_IRQ9_EINT1	0	DSP IRQ9 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	7	DSP_IRQ8_EINT1	0	DSP IRQ8 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	6	DSP_IRQ7_EINT1	0	DSP IRQ7 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	5	DSP_IRQ6_EINT1	0	DSP IRQ6 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	4	DSP_IRQ5_EINT1	0	DSP IRQ5 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	3	DSP_IRQ4_EINT1	0	DSP IRQ4 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	2	DSP_IRQ3_EINT1	0	DSP IRQ3 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	DSP_IRQ2_EINT1	0	DSP IRQ2 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	DSP_IRQ1_EINT1	0	DSP IRQ1 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6155 (0x180B) IRQ1_Status_12	6	SPKOUTL_SC_EINT1	0	SPKOUT Short Circuit Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	3	HP2R_SC_EINT1	0	EPOUTN Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	2	HP2L_SC_EINT1	0	EPOUTP Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	HP1R_SC_EINT1	0	HPOUTR Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	HP1L_SC_EINT1	0	HPOUTL Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6156 (0x180C) IRQ1_Status_13	6	SPKOUTL_ENABLE_DONE_EINT1	0	SPKOUT Enable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	HP1R_ENABLE_DONE_EINT1	0	HPOUTR/EPOUTN Enable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	HP1L_ENABLE_DONE_EINT1	0	HPOUTL/EPOUTP Enable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6157 (0x180D) IRQ1_Status_14	6	SPKOUTL_DISABLE_DONE_EINT1	0	SPKOUTL Disable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	HP1R_DISABLE_DONE_EINT1	0	HPOUTR/EPOUTN Disable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	HP1L_DISABLE_DONE_EINT1	0	HPOUTL/EPOUTP Disable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.

Table 4-78. Interrupt 1 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6158 (0x180E) IRQ1_Status_15	2	SPK_OVERHEAT_WARN_EINT1	0	Speaker Overheat Warning Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	SPK_OVERHEAT_EINT1	0	Speaker Overheat Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	SPK_SHUTDOWN_EINT1	0	Speaker Shutdown Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
R6160 (0x1810) IRQ1_Status_17	14	GP15_EINT1	0	GPIO15 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	13	GP14_EINT1	0	GPIO14 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	12	GP13_EINT1	0	GPIO13 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	11	GP12_EINT1	0	GPIO12 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	10	GP11_EINT1	0	GPIO11 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	9	GP10_EINT1	0	GPIO10 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	8	GP9_EINT1	0	GPIO9 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	7	GP8_EINT1	0	GPIO8 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	6	GP7_EINT1	0	GPIO7 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	5	GP6_EINT1	0	GPIO6 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	4	GP5_EINT1	0	GPIO5 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	3	GP4_EINT1	0	GPIO4 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	2	GP3_EINT1	0	GPIO3 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	1	GP2_EINT1	0	GPIO2 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	0	GP1_EINT1	0	GPIO1 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
R6164 (0x1814) IRQ1_Status_21	1	TIMER2_EINT1	0	Timer 2 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	TIMER1_EINT1	0	Timer 1 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6165 (0x1815) IRQ1_Status_22	1	EVENT2_NOT_EMPTY_EINT1	0	Event Log 2 FIFO Not Empty Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	EVENT1_NOT_EMPTY_EINT1	0	Event Log 1 FIFO Not Empty Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6166 (0x1816) IRQ1_Status_23	1	EVENT2_FULL_EINT1	0	Event Log 2 FIFO Full Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	EVENT1_FULL_EINT1	0	Event Log 1 FIFO Full Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6167 (0x1817) IRQ1_Status_24	1	EVENT2_WMARK_EINT1	0	Event Log 2 FIFO Watermark Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	EVENT1_WMARK_EINT1	0	Event Log 1 FIFO Watermark Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6168 (0x1818) IRQ1_Status_25	0	DSP1_DMA_EINT1	00	DSP1 DMA Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6170 (0x181A) IRQ1_Status_27	0	DSP1_START1_EINT1	0	DSP1 Start 1 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.

Table 4-78. Interrupt 1 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6171 (0x181B) IRQ1_Status_28	0	DSP1_START2_EINT1	0	DSP1 Start 2 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6173 (0x181D) IRQ1_Status_30	0	DSP1_BUSY_EINT1	0	DSP1 Busy Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6176 (0x1820) IRQ1_Status_33	0	DSP1_BUS_ERR_EINT1	0	DSP1 Bus Error Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6208 (0x1840) to R6240 (0x1860)		IM_*	See Footnote [1]	For each x_EINT1 interrupt bit in R6144 to R6176, a corresponding mask bit (IM_*) is provided in R6208 to R6240. The mask bits are coded as follows: 0 = Do not mask interrupt 1 = Mask interrupt
R6272 (0x1880) IRQ1_Raw_Status_1	12	CTRLIF_ERR_STS1	0	Control Interface Error Status 0 = Normal 1 = Control Interface Error
	7	BOOT_DONE_STS1	0	Boot Status 0 = Busy (boot sequence in progress) 1 = Idle (boot sequence completed) Control register writes should not be attempted until Boot Sequence has completed.
R6273 (0x1881) IRQ1_Raw_Status_2	15	FLL_AO_REF_LOST_STS1	0	FLL_AO Reference Lost Status 0 = Normal 1 = Reference Lost
	14	DSPCLK_ERR_STS1	0	DSPCLK Error Interrupt Status 0 = Normal 1 = Insufficient DSPCLK cycles for the requested DSP1 clock frequency
	12	SYSCLK_ERR_STS1	0	SYSCLK Error Interrupt Status 0 = Normal 1 = Insufficient SYSCLK cycles for the requested signal path functionality
	11	FLL_AO_LOCK_STS1	0	FLL_AO Lock Status 0 = Not locked 1 = Locked
	8	FLL1_LOCK_STS1	0	FLL1 Lock Status 0 = Not locked 1 = Locked
R6278 (0x1886) IRQ1_Raw_Status_7	4	MICD_CLAMP_STS1	0	MICDET Clamp status 0 = Clamp not active 1 = Clamp active
	2	JD2_STS1	0	JACKDET2 input status 0 = Jack not detected 1 = Jack is detected (Assumes the JACKDET2 pin is pulled low on jack insertion.)
	0	JD1_STS1	0	JACKDET1 input status 0 = Jack not detected 1 = Jack is detected (Assumes the JACKDET1 pin is pulled low on jack insertion.)
R6280 (0x1888) IRQ1_Raw_Status_9	2	INPUTS_SIG_DET_STS1	0	Input Path Signal-Detect Status 0 = Normal 1 = Signal detected
	1	DRC2_SIG_DET_STS1	0	DRC2 Signal-Detect Status 0 = Normal 1 = Signal detected
	0	DRC1_SIG_DET_STS1	0	DRC1 Signal-Detect Status 0 = Normal 1 = Signal detected

Table 4-78. Interrupt 1 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6283 (0x188B) IRQ1_Raw_ Status_12	6	SPKOUTL_SC_STS1	0	SPKOUT Short Circuit Status 0 = Normal 1 = Short Circuit detected
	3	HP2R_SC_STS1	0	EPOUTN Short Circuit Status 0 = Normal 1 = Short Circuit detected
	2	HP2L_SC_STS1	0	EPOUTP Short Circuit Status 0 = Normal 1 = Short Circuit detected
	1	HP1R_SC_STS1	0	HPOUTR Short Circuit Status 0 = Normal 1 = Short Circuit detected
	0	HP1L_SC_STS1	0	HPOUTL Short Circuit Status 0 = Normal 1 = Short Circuit detected
R6284 (0x188C) IRQ1_Raw_ Status_13	6	SPKOUTL_ENABLE_DONE_STS1	0	SPKOUT Enable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	1	HP1R_ENABLE_DONE_STS1	0	HPOUTR/EPOUTN Enable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	0	HP1L_ENABLE_DONE_STS1	0	HPOUTL/EPOUTP Enable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
R6285 (0x188D) IRQ1_Raw_ Status_14	6	SPKOUTL_DISABLE_DONE_STS1	0	SPKOUT Disable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	1	HP1R_DISABLE_DONE_STS1	0	HPOUTR/EPOUTN Disable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	0	HP1L_DISABLE_DONE_STS1	0	HPOUTL/EPOUTP Disable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
R6286 (0x188E) IRQ1_Raw_ Status_15	2	SPK_OVERHEAT_WARN_STS1	0	Speaker Overheat Warning Status 0 = Normal 1 = Warning temperature exceeded
	1	SPK_OVERHEAT_STS1	0	Speaker Overheat Status 0 = Normal 1 = Shutdown temperature exceeded
	0	SPK_SHUTDOWN_STS1	0	Speaker Shutdown Status 0 = Normal 1 = Speaker Shutdown completed (due to Overheat Temperature or Short Circuit condition)

Table 4-78. Interrupt 1 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6288 (0x1890) IRQ1_Raw_ Status_17	14	GP15_STS1	0	GPIO*n* Input status. Reads back the logic level of GPIO*n*. Only valid for pins configured as GPIO input (does not include DSPGPIO inputs).
	13	GP14_STS1	0	
	12	GP13_STS1	0	
	11	GP12_STS1	0	
	10	GP11_STS1	0	
	9	GP10_STS1	0	
	8	GP9_STS1	0	
	7	GP8_STS1	0	
	6	GP7_STS1	0	
	5	GP6_STS1	0	
	4	GP5_STS1	0	
	3	GP4_STS1	0	
	2	GP3_STS1	0	
	1	GP2_STS1	0	
	0	GP1_STS1	0	
R6293 (0x1895) IRQ1_Raw_ Status_22	1	EVENT2_NOT_EMPTY_STS1	0	Event Log *n* FIFO Not Empty status 0 = FIFO Empty 1 = FIFO Not Empty
	0	EVENT1_NOT_EMPTY_STS1	0	
R6294 (0x1896) IRQ1_Raw_ Status_23	1	EVENT2_FULL_STS1	0	Event Log *n* FIFO Full status 0 = FIFO Not Full 1 = FIFO Full
	0	EVENT1_FULL_STS1	0	
R6295 (0x1897) IRQ1_Raw_ Status_24	1	EVENT2_WMARK_STS1	0	Event Log *n* FIFO Watermark status 0 = FIFO Watermark not reached 1 = FIFO Watermark reached
	0	EVENT1_WMARK_STS1	0	
R6296 (0x1898) IRQ1_Raw_ Status_25	0	DSP1_DMA_STS1	00	DSP1 DMA status 0 = Normal 1 = All enabled WDMA buffers filled, and all enabled RDMA buffers emptied
R6301 (0x189D) IRQ1_Raw_ Status_30	0	DSP1_BUSY_STS1	0	DSP1 Busy status 0 = DSP Idle 1 = DSP Busy

1. The BOOT_DONE_EINT1 interrupt is 0 (unmasked) by default; all other interrupts are 1 (masked) by default.

The IRQ2 interrupt, mask, and status control registers are described in Table 4-79.

Table 4-79. Interrupt 2 Control Registers

Register Address	Bit	Label	Default	Description
R6400 (0x1900) IRQ2_Status_1	12	CTRLIF_ERR_EINT2	0	Control Interface Error Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	9	SYSCLK_FAIL_EINT2	0	SYSCLK Fail Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	7	BOOT_DONE_EINT2	0	Boot Done Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6401 (0x1901) IRQ2_Status_2	15	FLL_AO_REF_LOST_EINT2	0	FLL_AO Reference Lost Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	14	DSPCLK_ERR_EINT2	0	DSPCLK Error Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	12	SYSCLK_ERR_EINT2	0	SYSCLK Error Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	11	FLL_AO_LOCK_EINT2	0	FLL_AO Lock Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	8	FLL1_LOCK_EINT2	0	FLL1 Lock Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.

Table 4-79. Interrupt 2 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6405 (0x1905) IRQ2_Status_6	9	MICDET2_EINT2	0	Mic/Accessory Detect 2 Interrupt (Detection event triggered) Note: Cleared when a 1 is written.
	8	MICDET1_EINT2	0	Mic/Accessory Detect 1 Interrupt (Detection event triggered) Note: Cleared when a 1 is written.
	0	HPDET_EINT2	0	Headphone Detect Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6406 (0x1906) IRQ2_Status_7	5	MICD_CLAMP_FALL_EINT2	0	MICDET Clamp Interrupt (Falling edge triggered) Note: Cleared when a 1 is written.
	4	MICD_CLAMP_RISE_EINT2	0	MICDET Clamp Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	3	JD2_FALL_EINT2	0	JD2 Interrupt (Falling edge triggered) Note: Cleared when a 1 is written.
	2	JD2_RISE_EINT2	0	JD2 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	JD1_FALL_EINT2	0	JD1 Interrupt (Falling edge triggered) Note: Cleared when a 1 is written.
	0	JD1_RISE_EINT2	0	JD1 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6408 (0x1908) IRQ2_Status_9	2	INPUTS_SIG_DET_EINT2	0	Input Path Signal-Detect Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	1	DRC2_SIG_DET_EINT2	0	DRC2 Signal-Detect Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	0	DRC1_SIG_DET_EINT2	0	DRC1 Signal-Detect Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.

Table 4-79. Interrupt 2 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6410 (0x190A) IRQ2_Status_11	15	DSP_IRQ16_EINT2	0	DSP IRQ16 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	14	DSP_IRQ15_EINT2	0	DSP IRQ15 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	13	DSP_IRQ14_EINT2	0	DSP IRQ14 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	12	DSP_IRQ13_EINT2	0	DSP IRQ13 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	11	DSP_IRQ12_EINT2	0	DSP IRQ12 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	10	DSP_IRQ11_EINT2	0	DSP IRQ11 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	9	DSP_IRQ10_EINT2	0	DSP IRQ10 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	8	DSP_IRQ9_EINT2	0	DSP IRQ9 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	7	DSP_IRQ8_EINT2	0	DSP IRQ8 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	6	DSP_IRQ7_EINT2	0	DSP IRQ7 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	5	DSP_IRQ6_EINT2	0	DSP IRQ6 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	4	DSP_IRQ5_EINT2	0	DSP IRQ5 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	3	DSP_IRQ4_EINT2	0	DSP IRQ4 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	2	DSP_IRQ3_EINT2	0	DSP IRQ3 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	DSP_IRQ2_EINT2	0	DSP IRQ2 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	DSP_IRQ1_EINT2	0	DSP IRQ1 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6411 (0x190B) IRQ2_Status_12	6	SPKOUTL_SC_EINT2	0	SPKOUT Short Circuit Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	3	HP2R_SC_EINT2	0	EPOUTN Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	2	HP2L_SC_EINT2	0	EPOUTP Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	HP1R_SC_EINT2	0	HPOUTR Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	HP1L_SC_EINT2	0	HPOUTL Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6412 (0x190C) IRQ2_Status_13	6	SPKOUTL_ENABLE_DONE_EINT2	0	SPKOUT Enable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	HP1R_ENABLE_DONE_EINT2	0	HPOUTR/EPOUTN Enable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	HP1L_ENABLE_DONE_EINT2	0	HPOUTL/EPOUTP Enable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6413 (0x190D) IRQ2_Status_14	6	SPKOUTL_DISABLE_DONE_EINT2	0	SPKOUT Disable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	HP1R_DISABLE_DONE_EINT2	0	HPOUTR/EPOUTN Disable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	HP1L_DISABLE_DONE_EINT2	0	HPOUTL/EPOUTP Disable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.

Table 4-79. Interrupt 2 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6414 (0x190E) IRQ2_Status_15	2	SPK_OVERHEAT_WARN_EINT2	0	Speaker Overheat Warning Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	SPK_OVERHEAT_EINT2	0	Speaker Overheat Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	SPK_SHUTDOWN_EINT2	0	Speaker Shutdown Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
R6416 (0x1910) IRQ2_Status_17	14	GP15_EINT2	0	GPIO15 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	13	GP14_EINT2	0	GPIO14 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	12	GP13_EINT2	0	GPIO13 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	11	GP12_EINT2	0	GPIO12 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	10	GP11_EINT2	0	GPIO11 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	9	GP10_EINT2	0	GPIO10 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	8	GP9_EINT2	0	GPIO9 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	7	GP8_EINT2	0	GPIO8 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	6	GP7_EINT2	0	GPIO7 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	5	GP6_EINT2	0	GPIO6 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	4	GP5_EINT2	0	GPIO5 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	3	GP4_EINT2	0	GPIO4 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	2	GP3_EINT2	0	GPIO3 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	1	GP2_EINT2	0	GPIO2 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	0	GP1_EINT2	0	GPIO1 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
R6420 (0x1914) IRQ2_Status_21	1	TIMER2_EINT2	0	Timer 2 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	TIMER1_EINT2	0	Timer 1 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6421 (0x1915) IRQ2_Status_22	1	EVENT2_NOT_EMPTY_EINT2	0	Event Log 2 FIFO Not Empty Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	EVENT1_NOT_EMPTY_EINT2	0	Event Log 1 FIFO Not Empty Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6422 (0x1916) IRQ2_Status_23	1	EVENT2_FULL_EINT2	0	Event Log 2 FIFO Full Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	EVENT1_FULL_EINT2	0	Event Log 1 FIFO Full Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6423 (0x1917) IRQ2_Status_24	1	EVENT2_WMARK_EINT2	0	Event Log 2 FIFO Watermark Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	EVENT1_WMARK_EINT2	0	Event Log 1 FIFO Watermark Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6424 (0x1918) IRQ2_Status_25	0	DSP1_DMA_EINT2	00	DSP1 DMA Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6426 (0x191A) IRQ2_Status_27	0	DSP1_START1_EINT2	0	DSP1 Start 1 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.

Table 4-79. Interrupt 2 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6427 (0x191B) IRQ2_Status_28	0	DSP1_START2_EINT2	0	DSP1 Start 2 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6429 (0x191D) IRQ2_Status_30	0	DSP1_BUSY_EINT2	0	DSP1 Busy Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6432 (0x1920) IRQ2_Status_33	0	DSP1_BUS_ERR_EINT2	0	DSP1 Bus Error Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6464 (0x1940) to R6496 (0x1960)		IM_*	1	For each x_EINT2 interrupt bit in R6400 to R6432, a corresponding mask bit (IM_*) is provided in R6464 to R6496. The mask bits are coded as follows: 0 = Do not mask interrupt 1 = Mask interrupt
R6528 (0x1980) IRQ2_Raw_ Status_1	12	CTRLIF_ERR_STS2	0	Control Interface Error Status 0 = Normal 1 = Control Interface Error
	7	BOOT_DONE_STS2	0	Boot Status 0 = Busy (boot sequence in progress) 1 = Idle (boot sequence completed) Control register writes should not be attempted until Boot Sequence has completed.
R6529 (0x1981) IRQ2_Raw_ Status_2	15	FLL_AO_REF_LOST_STS2	0	FLL_AO Reference Lost Status 0 = Normal 1 = Reference Lost
	14	DSPCLK_ERR_STS2	0	DSPCLK Error Interrupt Status 0 = Normal 1 = Insufficient DSPCLK cycles for the requested DSP1 clock frequency
	12	SYSCLK_ERR_STS2	0	SYSCLK Error Interrupt Status 0 = Normal 1 = Insufficient SYSCLK cycles for the requested signal path functionality
	11	FLL_AO_LOCK_STS2	0	FLL_AO Lock Status 0 = Not locked 1 = Locked
	8	FLL1_LOCK_STS2	0	FLL1 Lock Status 0 = Not locked 1 = Locked
R6534 (0x1986) IRQ2_Raw_ Status_7	4	MICD_CLAMP_STS2	0	MICDET Clamp status 0 = Clamp not active 1 = Clamp active
	2	JD2_STS2	0	JACKDET2 input status 0 = Jack not detected 1 = Jack is detected (Assumes the JACKDET2 pin is pulled low on jack insertion.)
	0	JD1_STS2	0	JACKDET1 input status 0 = Jack not detected 1 = Jack is detected (Assumes the JACKDET1 pin is pulled low on jack insertion.)
R6536 (0x1988) IRQ2_Raw_ Status_9	2	INPUTS_SIG_DET_STS2	0	Input Path Signal-Detect Status 0 = Normal 1 = Signal detected
	1	DRC2_SIG_DET_STS2	0	DRC2 Signal-Detect Status 0 = Normal 1 = Signal detected
	0	DRC1_SIG_DET_STS2	0	DRC1 Signal-Detect Status 0 = Normal 1 = Signal detected

Table 4-79. Interrupt 2 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6539 (0x198B) IRQ2_Raw_ Status_12	6	SPKOUTL_SC_STS2	0	SPKOUT Short Circuit Status 0 = Normal 1 = Short Circuit detected
	3	HP2R_SC_STS2	0	EPOUTN Short Circuit Status 0 = Normal 1 = Short Circuit detected
	2	HP2L_SC_STS2	0	EPOUTP Short Circuit Status 0 = Normal 1 = Short Circuit detected
	1	HP1R_SC_STS2	0	HPOUTR Short Circuit Status 0 = Normal 1 = Short Circuit detected
	0	HP1L_SC_STS2	0	HPOUTL Short Circuit Status 0 = Normal 1 = Short Circuit detected
R6540 (0x198C) IRQ2_Raw_ Status_13	6	SPKOUTL_ENABLE_DONE_STS2	0	SPKOUT Enable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	1	HP1R_ENABLE_DONE_STS2	0	HPOUTR/EPOUTN Enable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	0	HP1L_ENABLE_DONE_STS2	0	HPOUTL/EPOUTP Enable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
R6541 (0x198D) IRQ2_Raw_ Status_14	6	SPKOUTL_DISABLE_DONE_STS2	0	SPKOUT Disable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	1	HP1R_DISABLE_DONE_STS2	0	HPOUTR/EPOUTN Disable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	0	HP1L_DISABLE_DONE_STS2	0	HPOUTL/EPOUTP Disable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
R6542 (0x198E) IRQ2_Raw_ Status_15	2	SPK_OVERHEAT_WARN_STS2	0	Speaker Overheat Warning Status 0 = Normal 1 = Warning temperature exceeded
	1	SPK_OVERHEAT_STS2	0	Speaker Overheat Status 0 = Normal 1 = Shutdown temperature exceeded
	0	SPK_SHUTDOWN_STS2	0	Speaker Shutdown Status 0 = Normal 1 = Speaker Shutdown completed (due to Overheat Temperature or Short Circuit condition)

Table 4-79. Interrupt 2 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6544 (0x1990) IRQ2_Raw_ Status_17	14	GP15_STS2	0	GPIO*n* Input status Reads back the logic level of GPIO*n*. Only valid for pins configured as GPIO input (does not include DSPGPIO inputs).
	13	GP14_STS2	0	
	12	GP13_STS2	0	
	11	GP12_STS2	0	
	10	GP11_STS2	0	
	9	GP10_STS2	0	
	8	GP9_STS2	0	
	7	GP8_STS2	0	
	6	GP7_STS2	0	
	5	GP6_STS2	0	
	4	GP5_STS2	0	
	3	GP4_STS2	0	
	2	GP3_STS2	0	
	1	GP2_STS2	0	
	0	GP1_STS2	0	
R6549 (0x1995) IRQ2_Raw_ Status_22	1	EVENT2_NOT_EMPTY_STS2	0	Event Log *n* FIFO Not Empty status 0 = FIFO Empty 1 = FIFO Not Empty
	0	EVENT1_NOT_EMPTY_STS2	0	
R6550 (0x1996) IRQ2_Raw_ Status_23	1	EVENT2_FULL_STS2	0	Event Log *n* FIFO Full status 0 = FIFO Not Full 1 = FIFO Full
	0	EVENT1_FULL_STS2	0	
R6551 (0x1997) IRQ2_Raw_ Status_24	1	EVENT2_WMARK_STS2	0	Event Log *n* FIFO Watermark status 0 = FIFO Watermark not reached 1 = FIFO Watermark reached
	0	EVENT1_WMARK_STS2	0	
R6552 (0x1998) IRQ2_Raw_ Status_25	0	DSP1_DMA_STS2	00	DSP1 DMA status 0 = Normal 1 = All enabled WDMA buffers filled, and all enabled RDMA buffers emptied
R6557 (0x199D) IRQ2_Raw_ Status_30	0	DSP1_BUSY_STS2	0	DSP1 Busy status 0 = DSP Idle 1 = DSP Busy

The IRQ output and polarity control registers are described in Table 4-80.

Table 4-80. Interrupt Control Registers

Register Address	Bit	Label	Default	Description
R6784 (0x1A80) IRQ1_CTRL	11	IM_IRQ1	0	IRQ1 Output Interrupt mask. 0 = Do not mask interrupt. 1 = Mask interrupt.
	10	IRQ_POL	1	IRQ Output Polarity Select 0 = Noninverted (Active High) 1 = Inverted (Active Low)
	9	IRQ_OP_CFG	0	IRQ Output Configuration 0 = CMOS 1 = Open drain
R6786 (0x1A82) IRQ2_CTRL	11	IM_IRQ2	0	IRQ2 Output Interrupt mask. 0 = Do not mask interrupt. 1 = Mask interrupt.
R6816 (0x1AA0) Interrupt_Raw_ Status_1	1	IRQ2_STS	0	IRQ2 Status. IRQ2_STS is the logical OR of all unmasked x_EINT2 interrupts. 0 = Not asserted 1 = Asserted
	0	IRQ1_STS	0	IRQ1 Status. IRQ1_STS is the logical OR of all unmasked x_EINT1 interrupts. 0 = Not asserted 1 = Asserted

4.13 Clocking and Sample Rates

The CS47L15 requires a clock reference for its internal functions and also for the input (ADC) paths, output (DAC) paths, and digital audio interfaces. Under typical clocking configurations, all commonly used audio sample rates can be derived directly from the external reference; for additional flexibility, the CS47L15 incorporates two FLL circuits to perform frequency conversion and filtering.

External clock signals may be connected via MCLK1 and MCLK2. In AIF Slave Modes, the BCLK signals may be used as a reference for the system clocks. To avoid audible glitches, all clock configurations must be set up before enabling playback.

4.13.1 System Clocking Overview

The CS47L15 supports two primary clock domains—SYSCLK and DSPCLK.

The SYSCLK clock domain is the reference clock for all the audio signal paths on the CS47L15. Up to three different sample rates may be independently selected for specific audio interfaces and other input/output signal paths.

The DSPCLK clock domain is the reference clock for the programmable DSP core on the CS47L15. A wide range of DSPCLK frequencies can be supported, and a programmable clock divider is also provided for the DSP core, allowing the DSP clocking (and power consumption) to be optimized according to the applicable processing requirements. See Section 4.3 for further details.

Note that there is no requirement for DSPCLK to be synchronized to SYSCLK. The DSPCLK controls the software execution in the DSP core; audio outputs from the DSP are synchronized to SYSCLK, regardless of the applicable DSPCLK rate.

Excluding the DSP, each subsystem within the CS47L15 digital core is clocked at a dynamically controlled rate, limited by the SYSCLK frequency. For maximum signal mixing and processing capacity, it is recommended that the highest possible SYSCLK frequency is configured.

The DSP core is clocked at the DSPCLK rate (or supported divisions of the DSPCLK frequency). The DSPCLK configuration must ensure that sufficient clock cycles are available for the applicable processing requirements. The requirements vary, according to the particular software that is in use.

4.13.2 Sample-Rate Control

The CS47L15 audio signal paths are synchronized to the SYSCLK system clock.

Different sample rates may be selected for each of the audio interfaces (AIF1, AIF2, AIF3), and for the input (ADC) and output (DAC) paths, but each enabled interface must still be synchronized to SYSCLK.

The CS47L15 can support a maximum of three different sample rates at any time. The supported sample rates range from 8kHz to 192kHz.

The applicable sample rates are selected using SAMPLE_RATE_1, SAMPLE_RATE_2 and SAMPLE_RATE_3. These must each be numerically related to each other and to the SYSCLK frequency (further details of these requirements are provided in Table 4-81 and the accompanying text).

Each of the audio interfaces, input paths, and output paths is associated with one of the sample rates selected by the SAMPLE_RATE_*n* fields.

Note that, when any of the SAMPLE_RATE_*n* fields is written to, the activation of the new setting is automatically synchronized by the CS47L15 to ensure continuity of all active signal paths. The SAMPLE_RATE_*n*_STS bits provide indication of the sample rate selections that have been implemented.

The following restrictions must be observed regarding the sample-rate control configuration:

- All external clock references (MCLK input or Slave Mode AIF input) must be within 1% of the applicable register field settings.
- The input (ADC/DMIC) sample rate is valid from 8–192 kHz. If 384- or 768-kHz DMIC clock rate is selected on any of the input paths, the supported sample rate is valid only up to 48 or 96 kHz respectively.

- The S/PDIF sample rate is valid from 32–192 kHz.
- The isochronous sample-rate converters (ISRCs) support sample rates 8–192 kHz. For each ISRC, the higher sample rate must be an integer multiple of the lower rate.

4.13.3 Automatic Sample-Rate Detection

The CS47L15 supports automatic sample-rate detection on the digital audio interfaces (AIF1–AIF3). Note that this is only possible when the respective interface is operating in Slave Mode (i.e., when LRCLK and BCLK are inputs to the CS47L15).

Automatic sample-rate detection is enabled by setting RATE_EST_ENA. The LRCLK input pin selected for sample-rate detection is set using LRCLK_SRC.

As many as four audio sample rates can be configured for automatic detection; these sample rates are selected using the SAMPLE_RATE_DETECT_*n* fields. Note that the function only detects sample rates that match one of the SAMPLE_RATE_DETECT_*n* fields.

If one of the selected audio sample rates is detected on the selected LRCLK input, the control-write sequencer is triggered. A unique sequence of actions may be programmed for each of the detected sample rates. Note that the applicable control sequences must be programmed by the user for each detection outcome; see Section 4.15.

The TRIG_ON_STARTUP bit controls whether the sample-rate detection circuit responds to the initial detection of the applicable interface (i.e., when the AIF*n* interface starts up).

- If TRIG_ON_STARTUP = 0, the detection circuit only responds (i.e., trigger the control-write sequencer) to a change in the detected sample rate—the initial sample-rate detection is ignored. (Note that the initial sample-rate detection is the first detection of a sample rate that matches one of the SAMPLE_RATE_DETECT_*n* fields.)
- If TRIG_ON_STARTUP = 1, the detection circuit triggers the control-write sequencer whenever a selected sample rate is detected, including when the AIF interface starts up, or when the sample-rate detection is first enabled.

As described above, setting TRIG_ON_STARTUP = 0 is designed to inhibit any response to the initial detection of a sample rate that matches one of the SAMPLE_RATE_DETECT_*n* fields. Note that, if the LRCLK_SRC setting is changed, or if the detection function is disabled and reenabled, a subsequent detection of a matching sample rate may trigger the control-write sequencer, regardless of the TRIG_ON_STARTUP setting.

There are some restrictions to be observed regarding the automatic sample-rate detection configuration, as noted in the following:

- The same sample rate must not be selected on more than one of the SAMPLE_RATE_DETECT_*n* fields.
- Sample rates 192 kHz and 176.4 kHz must not be selected concurrently.
- Sample rates 96 kHz and 88.2 kHz must not be selected concurrently.

The control registers associated with the automatic sample-rate detection function are described in Table 4-82.

4.13.4 System Clock Configuration

The system clocks (SYSCLK and DSPCLK) may be provided directly from external inputs (MCLK, or Slave Mode BCLK inputs). Alternatively, these clocks can be derived using the integrated FLLs, with MCLK, BCLK or LRCLK as a reference. Each clock is configured independently, as described in the following sections.

The SYSCLK clock must be configured and enabled before any audio path is enabled. The DSPCLK clock must be configured and enabled, if running firmware applications on any of the DSP cores.

4.13.4.1 SYSCLK Configuration

The required SYSCLK frequency is dependent on the SAMPLE_RATE_*n* fields. Table 4-81 illustrates the valid SYSCLK frequencies for every supported sample rate.

The SYSCLK frequency must be valid for all of the SAMPLE_RATE_*n* fields. It follows that all of the SAMPLE_RATE_*n* fields must select numerically-related values, that is, all from the same group of sample rates as represented in Table 4-81.

Table 4-81. SYSCLK Frequency Selection

<table>
<tr><th>SYSCLK Frequency (MHz)</th><th>SYSCLK_FREQ</th><th>SYSCLK_FRAC</th><th>Sample Rate (kHz)</th><th>SAMPLE_RATE_n</th></tr>
<tr><td rowspan="8">6.144
12.288
24.576
49.152
98.304</td><td rowspan="8">000
001
010
011
100</td><td rowspan="8">0</td><td>12</td><td>0x01</td></tr>
<tr><td>24</td><td>0x02</td></tr>
<tr><td>48</td><td>0x03</td></tr>
<tr><td>96</td><td>0x04</td></tr>
<tr><td>192</td><td>0x05</td></tr>
<tr><td>8</td><td>0x11</td></tr>
<tr><td>16</td><td>0x12</td></tr>
<tr><td>32</td><td>0x13</td></tr>
<tr><td rowspan="5">5.6448
11.2896
22.5792
45.1584
90.3168</td><td rowspan="5">000
001
010
011
100</td><td rowspan="5">1</td><td>11.025</td><td>0x09</td></tr>
<tr><td>22.05</td><td>0x0A</td></tr>
<tr><td>44.1</td><td>0x0B</td></tr>
<tr><td>88.2</td><td>0x0C</td></tr>
<tr><td>176.4</td><td>0x0D</td></tr>
</table>

Note: The SAMPLE_RATE_*n* fields must each be set to a value from the same group of sample rates, and from the same group as the SYSCLK frequency.

SYSCLK_SRC is used to select the SYSCLK source, as described in Table 4-82. The source may be MCLK*n*, AIF*n*BCLK, or FLL*n*. If an FLL circuit is selected as the source, the relevant FLL must be enabled and configured, as described in Section 4.13.8 and Section 4.13.9.

Note: FLL_AO is designed to support low-power always-on use cases only; for hi-fi audio use cases, it is recommended to use FLL1.

If FLL_AO is selected as SYSCLK source, two different clock frequencies are available—the loop frequency (45–50 MHz) or a higher frequency (loop frequency multiplied by 2). If either of these clocks is the SYSCLK source, FLL_AO must be enabled and configured. The FLL_AO_FREQ field must also be configured for the applicable (loop) frequency.

SYSCLK_FREQ and SYSCLK_FRAC must be set according to the frequency of the selected SYSCLK source.

The SYSCLK-referenced circuits within the digital core are clocked at a dynamically controlled rate that is limited by the SYSCLK frequency. For maximum signal mixing and processing capacity, the highest possible SYSCLK frequency should be used.

The SAMPLE_RATE_*n* fields are set according to the sample rates that are required by one or more of the CS47L15 audio interfaces. The CS47L15 supports sample rates ranging from 8–192 kHz.

The SYSCLK signal is enabled by setting SYSCLK_ENA. The applicable clock source (MCLK*n*, AIF*n*BCLK, or FLL*n*) must be enabled before setting SYSCLK_ENA. This bit should be cleared before stopping or removing the applicable clock source.

The CS47L15 supports seamless switching between clock sources. To change the SYSCLK configuration while SYSCLK is enabled, the SYSCLK_FRAC, SYSCLK_FREQ, and SYSCLK_SRC fields must be updated together in one register write operation. Note that, if changing the frequency only (not the source), SYSCLK_ENA should be cleared before the clock frequency is updated. The current SYSCLK frequency and source can be read from the SYSCLK_FREQ_STS and SYSCLK_SRC_STS fields respectively.

The CS47L15 performs automatic checks to confirm that the SYSCLK frequency is high enough to support the commanded signal paths and processing functions. If the frequency is too low, an attempt to enable a signal path or processing function fails. Note that active signal paths are not affected under such circumstances.

The SYSCLK frequency check provides input to the interrupt-control circuit and can be used to trigger an interrupt event if the frequency is not high enough to support the commanded functionality; see Section 4.12.

4.13.4.2 DSPCLK Configuration

The required DSPCLK frequency depends on the requirements of firmware loaded on the DSP core. The DSP is clocked at the DSPCLK rate or at supported divisions of the DSPCLK frequency. The DSPCLK configuration must ensure that sufficient clock cycles are available for the applicable processing requirements. The requirements vary, according to the particular software that is in use.

A configurable clock divider is also provided for the DSP core, allowing the DSP clocking (and power consumption) to be optimized according to the applicable processing requirements; see Section 4.4 for details.

DSP_CLK_FREQ must be configured for the applicable DSPCLK frequency. This field is coded in LSB units of 1/64 MHz. Note that, if the field coding cannot represent the DSPCLK frequency exactly, the DSPCLK frequency must be rounded down in the DSP_CLK_FREQ field.

The suggested method for calculating DSP_CLK_FREQ is to multiply the DSPCLK frequency by 64, round down to the nearest integer, and use the resulting integer as DSP_CLK_FREQ (LSB = 1).

DSP_CLK_SRC is used to select the DSPCLK source, as described in Table 4-82. The source may be MCLK*n*, AIF*n*BCLK, or FLL*n*. If an FLL circuit is selected as the source, the relevant FLL must be enabled and configured, as described in Section 4.13.8 and Section 4.13.9.

Note: If FLL1 is selected as DSPCLK source, the DSPCLK frequency is F_{VCO} × 1.5.

If FLL_AO is selected as DSPCLK source, two different clock frequencies are available—the loop frequency (45–50 MHz) or a higher frequency (loop frequency multiplied by 3). If either of these clocks is the DSPCLK source, the FLL_AO must be enabled and configured. The FLL_AO_FREQ must also be configured for the applicable (loop) frequency.

The DSPCLK signal is enabled by setting DSP_CLK_ENA. The applicable clock source (MCLK*n*, AIF*n*BCLK, or FLL) must be enabled before setting DSP_CLK_ENA. This bit should be cleared when reconfiguring the clock sources.

The CS47L15 supports seamless switching between clock sources. To change the DSPCLK configuration while DSPCLK is enabled, the DSP_CLK_FREQ field must be updated before DSP_CLK_SRC. The new configuration becomes effective when the DSP_CLK_SRC field is written. Note that, if changing the frequency only (not the source), the DSP_CLK_ENA bit should be cleared before the clock frequency is updated. The current DSPCLK frequency and source can be read from the DSP_CLK_FREQ_STS and DSP_CLK_SRC_STS fields respectively.

In a typical application, DSPCLK and SYSCLK are derived from a single FLL source. Note that there is no requirement for DSPCLK to be synchronized to SYSCLK. The DSPCLK controls the software execution in the DSP core; audio outputs from the DSP are synchronized to SYSCLK, regardless of the applicable DSPCLK rate.

Under specific conditions, the CS47L15 can provide clocking to the DSP core when DSPCLK is disabled. This capability is supported using the always-on FLL (FLL_AO), either in Free-Running Mode or locked to a valid clock reference. See Section 4.4.3 for further details.

4.13.5 Miscellaneous Clock Controls

The CS47L15 incorporates a 32-kHz clock circuit, which is required for input signal debounce, and microphone/accessory detect circuits. The 32-kHz clock must be configured and enabled whenever any of these features are in use.

The 32-kHz clock can be generated automatically from SYSCLK, or may be input directly as MCLK1 or MCLK2. The 32-kHz clock source is selected using CLK_32K_SRC. The 32-kHz clock is enabled by setting CLK_32K_ENA.

A clock output (OPCLK) derived from SYSCLK can be output on a GPIO pin. See Section 4.11 for details on configuring a GPIO pin for this function.

The CS47L15 provides integrated pull-down resistors on the MCLK1 and MCLK2 pins. This provides a flexible capability for interfacing with other devices.

The clocking scheme for the CS47L15 is shown in Fig. 4-63.

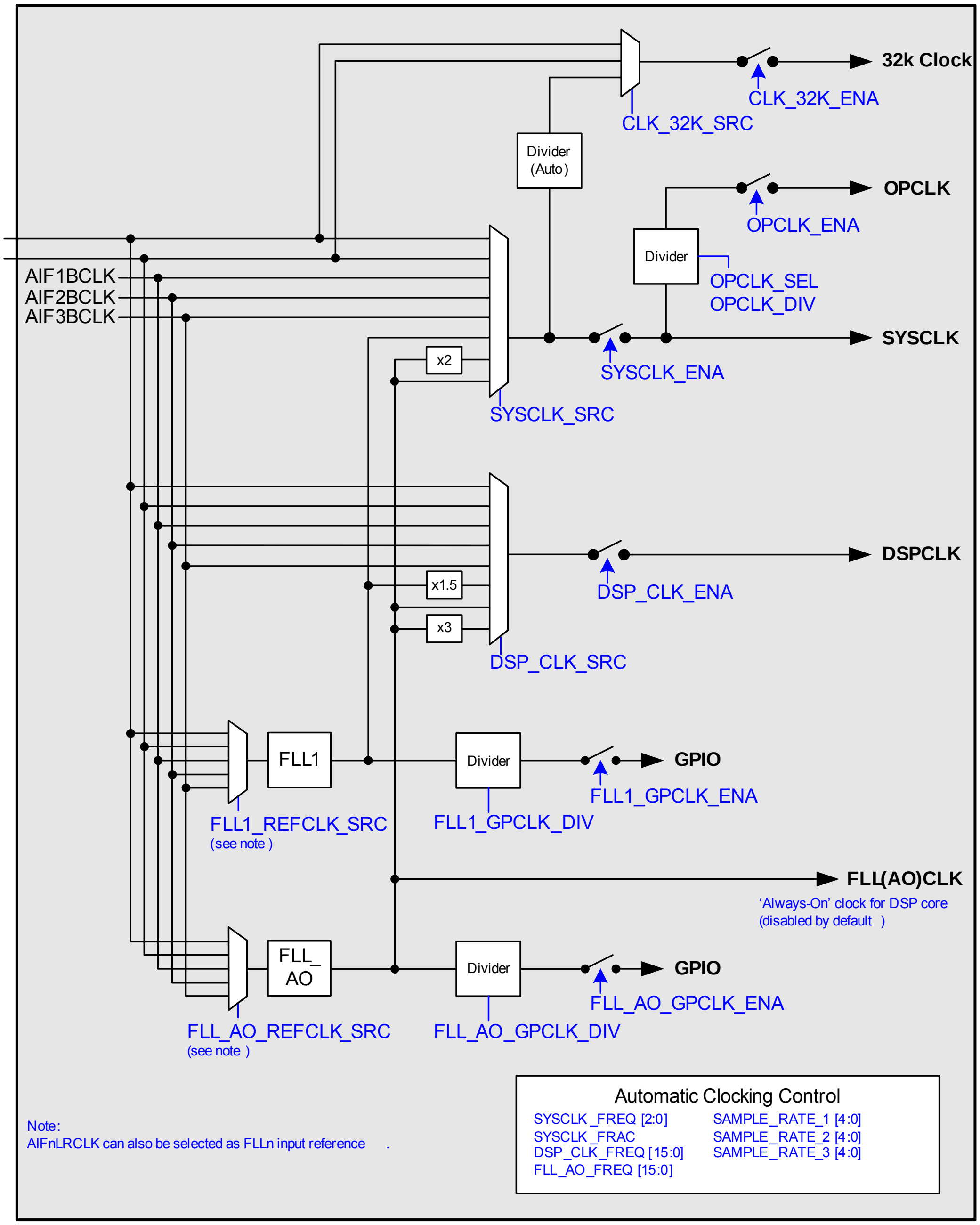


Figure 4-63. System Clocking

The CS47L15 clocking control registers are described in Table 4-82.

Table 4-82. Clocking Control

Register Address	Bit	Label	Default	Description
R256 (0x0100) Clock_32k_1	6	CLK_32K_ENA	0	32kHz Clock Enable 0 = Disabled 1 = Enabled
	1:0	CLK_32K_ SRC[1:0]	10	32kHz Clock Source 00 = MCLK1 (direct) 01 = MCLK2 (direct) 10 = SYSCLK (automatically divided) 11 = Reserved
R257 (0x0101) System_Clock_1	15	SYSCLK_FRAC	0	SYSCLK Frequency 0 = SYSCLK is a multiple of 6.144MHz 1 = SYSCLK is a multiple of 5.6448MHz
	10:8	SYSCLK_ FREQ[2:0]	100	SYSCLK Frequency 000 = 6.144 MHz (5.6448 MHz) 001 = 12.288 MHz (11.2896 MHz) 010 = 24.576 MHz (22.5792 MHz) 011 = 49.152 MHz (45.1584 MHz) 100 = 98.304 MHz (90.3168 MHz) All other codes are reserved The frequencies in brackets apply for 44.1 kHz–related sample rates only (i.e., SAMPLE_RATE_*n* = 01XXX).
	6	SYSCLK_ENA	0	SYSCLK Control 0 = Disabled 1 = Enabled SYSCLK should only be enabled after the applicable clock source has been configured and enabled. Set this bit to 0 when reconfiguring the clock sources. All digital core (audio mixer) x_SRC fields must be cleared before clearing SYSCLK_ENA = 0.
	3:0	SYSCLK_ SRC[3:0]	0100	SYSCLK Source 0000 = MCLK1 0001 = MCLK2 0100 = FLL1 0111 = FLL_AO (x2) 1000 = AIF1BCLK 1001 = AIF2BCLK 1010 = AIF3BCLK 1111 = FLL_AO All other codes are reserved
R258 (0x0102) Sample_rate_1	4:0	SAMPLE_RATE_ 1[4:0]	0x11	Sample Rate 1 Select 0x00 = None 0x01 = 12 kHz 0x02 = 24 kHz 0x03 = 48 kHz 0x04 = 96 kHz 0x05 = 192 kHz 0x09 = 11.025 kHz 0x0A = 22.05 kHz 0x0B = 44.1 kHz 0x0C = 88.2 kHz 0x0D = 176.4 kHz 0x11 = 8 kHz 0x12 = 16 kHz 0x13 = 32 kHz All other codes are reserved
R259 (0x0103) Sample_rate_2	4:0	SAMPLE_RATE_ 2[4:0]	0x11	Sample Rate 2 Select Field coding is same as SAMPLE_RATE_1.

Table 4-82. Clocking Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R260 (0x0104) Sample_rate_3	4:0	SAMPLE_RATE_3[4:0]	0x11	Sample Rate 3 Select Field coding is same as SAMPLE_RATE_1.
R266 (0x010A) Sample_rate_1_status	4:0	SAMPLE_RATE_1_STS[4:0]	0x00	Sample Rate 1 Status (Read only) Field coding is same as SAMPLE_RATE_1.
R267 (0x010B) Sample_rate_2_status	4:0	SAMPLE_RATE_2_STS[4:0]	0x00	Sample Rate 2 Status (Read only) Field coding is same as SAMPLE_RATE_1.
R268 (0x010C) Sample_rate_3_status	4:0	SAMPLE_RATE_3_STS[4:0]	0x00	Sample Rate 3 Status (Read only) Field coding is same as SAMPLE_RATE_1.
R288 (0x0120) DSP_Clock_1	6	DSP_CLK_ENA	0	DSPCLK Control 0 = Disabled 1 = Enabled DSPCLK should only be enabled after the applicable clock source has been configured and enabled. Set this bit to 0 when reconfiguring the clock sources.
	3:0	DSP_CLK_SRC[3:0]	0100	DSPCLK Source 0000 = MCLK1 0001 = MCLK2 0100 = FLL1 (x1.5) 0111 = FLL_AO (x3) 1000 = AIF1BCLK 1001 = AIF2BCLK 1010 = AIF3BCLK 1111 = FLL_AO All other codes are reserved
R290 (0x0122) DSP_Clock_2	15:0	DSP_CLK_FREQ[15:0]	0x0000	DSPCLK Frequency Coded as LSB = 1/64 MHz, Valid from 5.6 MHz to 148 MHz. Note that, if this field is written while DSPCLK is enabled, the new frequency does not become effective until DSP_CLK_SRC is updated. To reconfigure DSPCLK while DSPCLK is enabled, the DSP_CLK_FREQ field must be updated before DSP_CLK_SRC.
R292 (0x0124) DSP_Clock_3	15:0	FLL_AO_FREQ[15:0]	0x0000	FLL_AO Frequency Coded as LSB = 1/64 MHz, Valid from 45 MHz to 50 MHz.
R294 (0x0126) DSP_Clock_4	15:0	DSP_CLK_FREQ_STS[15:0]	0x0000	DSPCLK Frequency (Read only) Coded as LSB = 1/64 MHz.
R295 (0x0127) DSP_Clock_5	3:0	DSP_CLK_SRC_STS[3:0]	0000	DSPCLK Source (Read only) 0000 = MCLK1 0001 = MCLK2 0100 = FLL1 (F_{VCO} x 1.5) 0111 = FLL_AO (F_{NCO} x 3) 1000 = AIF1BCLK 1001 = AIF2BCLK 1010 = AIF3BCLK 1111 = FLL_AO (F_{NCO}) All other codes are reserved

Table 4-82. Clocking Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R329 (0x0149) Output_system_clock	15	OPCLK_ENA	0	OPCLK Enable 0 = Disabled 1 = Enabled
	7:3	OPCLK_DIV[4:0]	0x00	OPCLK Divider 0x02 = Divide by 2 0x04 = Divide by 4 0x06 = Divide by 6 … (even numbers only) 0x1E = Divide by 30 Note that only even numbered divisions (2, 4, 6, etc.) are valid selections. All other codes are reserved when the OPCLK signal is enabled.
	2:0	OPCLK_SEL[2:0]	000	OPCLK Source Frequency 000 = 6.144 MHz (5.6448 MHz) 001 = 12.288 MHz (11.2896 MHz) 010 = 24.576 MHz (22.5792 MHz) 011 = 49.152 MHz (45.1584 MHz) All other codes are reserved The frequencies in brackets apply for 44.1 kHz–related SYSCLK rates only (i.e., SAMPLE_RATE_*n* = 01XXX). The OPCLK Source Frequency must be less than or equal to the SYSCLK frequency.
R334 (0x014E) Clock_Gen_Pad_Ctrl	8	MCLK2_PD	0	MCLK2 Pull-Down Control 0 = Disabled 1 = Enabled
	7	MCLK1_PD	0	MCLK1 Pull-Down Control 0 = Disabled 1 = Enabled
R338 (0x0152) Rate_Estimator_1	4	TRIG_ON_STARTUP	0	Automatic Sample-Rate Detection Start-Up select 0 = Do not trigger Write Sequencer on initial detection 1 = Always trigger the Write Sequencer on sample-rate detection
	3:1	LRCLK_SRC[2:0]	000	Automatic Sample-Rate Detection source 000 = AIF1LRCLK 010 = AIF2LRCLK 100 = AIF3LRCLK All other codes are reserved
	0	RATE_EST_ENA	0	Automatic Sample-Rate Detection control 0 = Disabled 1 = Enabled
R339 (0x0153) Rate_Estimator_2	4:0	SAMPLE_RATE_DETECT_A[4:0]	0x00	Automatic Detection Sample Rate A (Up to four different sample rates can be configured for automatic detection.) Field coding is same as SAMPLE_RATE_*n*.
R340 (0x0154) Rate_Estimator_3	4:0	SAMPLE_RATE_DETECT_B[4:0]	0x00	Automatic Detection Sample Rate B (Up to four different sample rates can be configured for automatic detection.) Field coding is same as SAMPLE_RATE_*n*.
R341 (0x0155) Rate_Estimator_4	4:0	SAMPLE_RATE_DETECT_C[4:0]	0x00	Automatic Detection Sample Rate C (Up to four different sample rates can be configured for automatic detection.) Field coding is same as SAMPLE_RATE_*n*.

Table 4-82. Clocking Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R342 (0x0156) Rate_Estimator_5	4:0	SAMPLE_RATE_ DETECT_D[4:0]	0x00	Automatic Detection Sample Rate D (Up to four different sample rates can be configured for automatic detection.) Field coding is same as SAMPLE_RATE_*n*.
R352 (0x0160) Clocking_debug_5	6:4	SYSCLK_FREQ_ STS[2:0]	000	SYSCLK Frequency (Read only) 000 = 6.144 MHz (5.6448 MHz) 001 = 12.288 MHz (11.2896 MHz) 010 = 24.576 MHz (22.5792 MHz) 011 = 49.152 MHz (45.1584 MHz) 100 = 98.304 MHz (90.3168 MHz) All other codes are reserved The frequencies in brackets apply for 44.1 kHz–related sample rates only (i.e., SAMPLE_RATE_*n* = 01XXX).
	3:0	SYSCLK_SRC_ STS[3:0]	0000	SYSCLK Source (Read only) 0000 = MCLK1 0001 = MCLK2 0100 = FLL1 0111 = FLL_AO (x2) 1000 = AIF1BCLK 1001 = AIF2BCLK 1010 = AIF3BCLK 1111 = FLL_AO All other codes are reserved

In AIF Slave Modes, it is important to ensure that SYSCLK is synchronized with the associated external LRCLK. This can be achieved by selecting an MCLK input that is derived from the same reference as the LRCLK, or can be achieved by selecting the external BCLK or LRCLK signal as a reference input to one of the FLLs, as a source for SYSCLK.

If the AIF clock domain is not synchronized with the LRCLK, clicks arising from dropped or repeated audio samples occur, due to the inherent tolerances of multiple, asynchronous, system clocks. See Section 5.4 for further details on valid clocking configurations.

4.13.6 BCLK and LRCLK Control

The digital audio interfaces (AIF1–AIF3) use BCLK and LRCLK signals for synchronization. In Master Mode, these are output signals, generated by the CS47L15. In Slave Mode, these are input signals to the CS47L15. It is also possible to support mixed master/slave operation.

The BCLK and LRCLK signals are controlled as shown in Fig. 4-64. See Section 4.7 for details of the associated control fields.

Note that the BCLK and LRCLK signals are synchronized to SYSCLK. See Section 4.3.13 for further details.

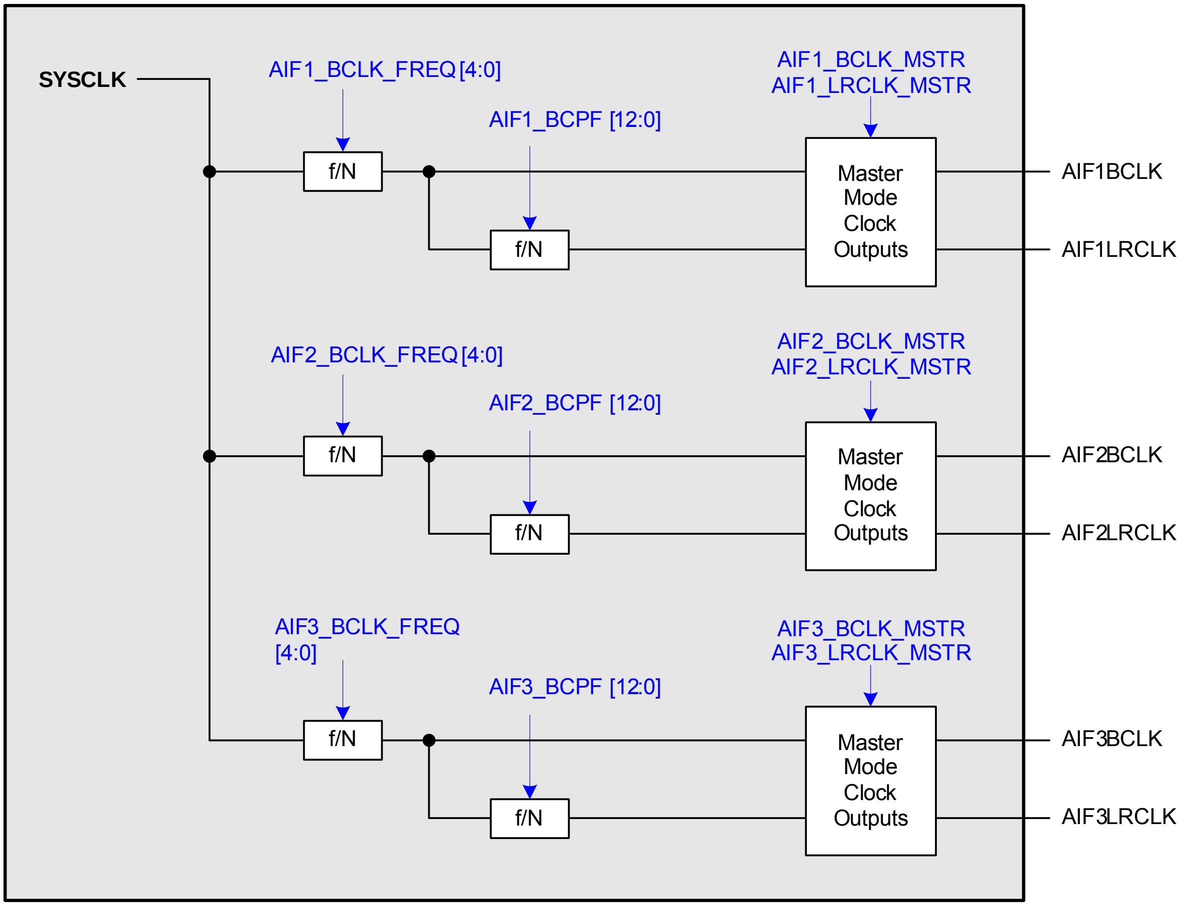


Figure 4-64. BCLK and LRCLK Control

4.13.7 Control Interface Clocking

Register map access is possible with or without a system clock—there is no requirement for SYSCLK, or any other system clock, to be enabled when accessing the register map.

See Section 4.14 for details of control register access.

4.13.8 Frequency-Locked Loop (FLL1)

Two integrated FLLs are provided to support the clocking requirements of the CS47L15. These can be configured according to the available reference clocks and the application requirements. The reference clock may use a high frequency (e.g., 12.288 MHz) or low frequency (e.g., 32.768 kHz). The FLL is tolerant of jitter and may be used to generate a stable output clock from a less stable input reference.

There are two FLL implementations on the CS47L15:

- FLL1 incorporates two subsystems—the main loop and the synchronizer loop—providing an advanced capability to use more than one reference clock to achieve best performance. FLL1 is described in the following subsections.
- FLL_AO is low-power FLL that supports additional always-on capability to provide system clocking when other references are unavailable or disabled. FLL_AO is described in Section 4.13.9. Note that FLL_AO is designed to support low-power always-on use cases only; for hi-fi audio use cases, it is recommended to use FLL1.

4.13.8.1 Overview

The FLL characteristics are summarized in Table 3-11. In normal operation, the FLL output is frequency locked to an input clock reference. The FLL can be used to generate a free-running clock in the absence of any external reference, as described in Section 4.13.8.7. Configurable spread-spectrum modulation can be applied to the FLL outputs, to control electro-magnetic interference (EMI) effects.

The FLL comprises two subsystems—the main loop and the synchronizer loop; these can be used together to maintain best frequency accuracy and noise (jitter) performance across multiple use cases. The two-loop design enables the FLL to synchronize effectively to an input clock that may be intermittent or noisy, while also achieving the performance benefits of a stable clock reference that may be asynchronous to the audio data.

The main loop takes a constant and stable clock reference as its input. For best performance, a high-frequency (e.g., 12.288 MHz) reference is recommended. The main FLL loop is free running without any clock reference if the input signal is removed; it can also be configured to initiate an output in the absence of any reference signal.

The synchronizer loop takes a separate clock reference as its input. The synchronizer input may be intermittent (e.g., during voice calls only). The FLL uses the synchronizer input, when available, as the frequency reference. To achieve the designed performance advantage, the synchronizer input must be synchronous with the audio data.

Note that, if only a single clock input reference is used, this must be configured as the main FLL input reference. The synchronizer should be disabled in this case.

The synchronizer loop should only be used when the main loop clock reference is present. If the input reference to the main FLL is intermittent, or may be interrupted unexpectedly, the synchronizer should be disabled.

4.13.8.2 FLL Enable

The FLL is enabled by setting FLL1_ENA. The FLL synchronizer is enabled by setting FLL1_SYNC_ENA. The FLL should be fully configured before setting the FLL1_ENA bit—this should be set as the final step of the FLL-enable sequence.

The FLL1_SYNC_ENA bit should not be changed if FLL1_ENA is set—the FLL1_ENA bit should be cleared before setting or clearing FLL1_SYNC_ENA.

The FLL supports configurable free-running operation, using the FLL1_FREERUN bit described in Section 4.13.8.7. Note that, once the FLL output has been established, the FLL is always free running if the input reference clock is stopped, regardless of the FLL1_FREERUN bit.

To disable the FLL while the input reference clock has stopped, FLL1_FREERUN must be set before clearing the FLL1_ ENA bit.

When changing FLL settings, it is recommended to disable the FLL by clearing the FLL1_ENA bit before updating the other register fields. When changing the input reference frequency F_{REF}, the FLL should be reset by clearing the FLL1_ENA bit before updating the affected register fields.

Note that some of the FLL configuration registers can be updated while the FLL is enabled, as described in Section 4.13.8.4. As a general rule, however, it is recommended to configure the FLL (and FLL Synchronizer, if applicable), before setting the corresponding x_ENA bits.

The FLL configuration is shown in Fig. 4-65.

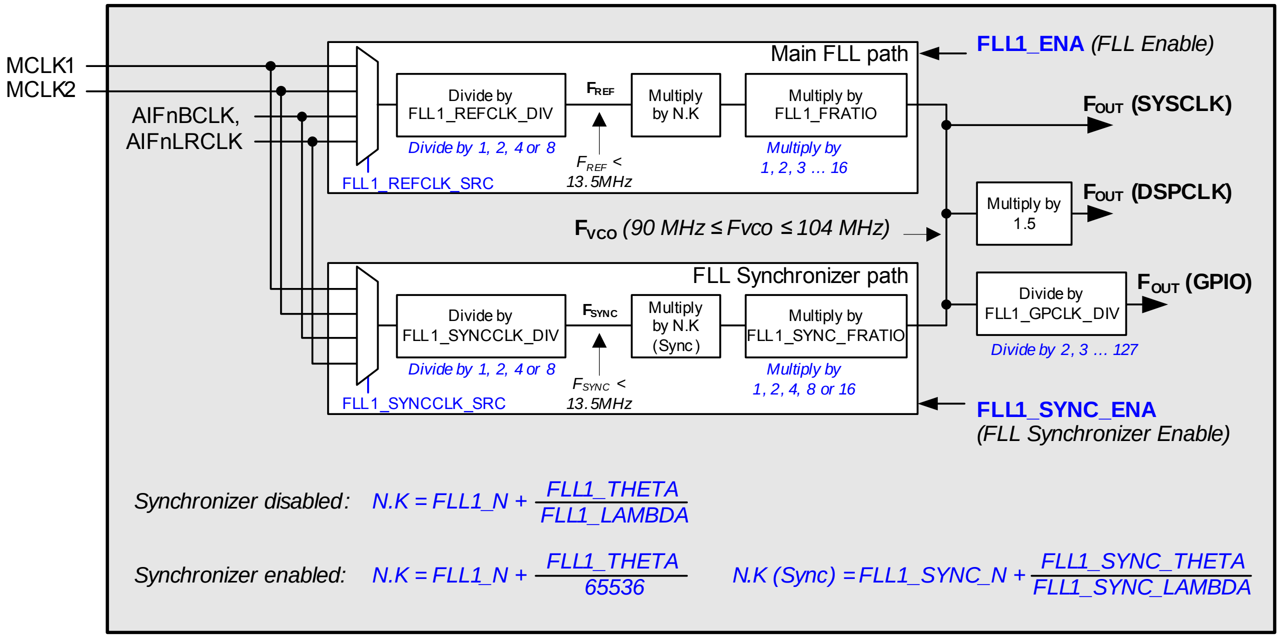


Figure 4-65. FLL Configuration

The procedure for configuring the FLL is described in the following subsections. Note that the configuration of the main FLL path and the FLL synchronizer path are very similar. One or both paths must be configured, depending on the application requirements:

- If a single clock input reference is used, only the main FLL path should be used.
- If the input reference to the main FLL is intermittent, or may be interrupted unexpectedly, only the main FLL path should be used.
- If two clock input references are used, the constant or low-noise clock is configured on the main FLL path and the high-accuracy clock is configured on the FLL synchronizer path. Note that the synchronizer input must be synchronous with the audio data.

4.13.8.3 Input Frequency Control

The main input reference is selected using FLL1_REFCLK_SRC. The synchronizer input reference is selected using FLL1_SYNCCLK_SRC. The available options in each case are MCLK1, MCLK2, AIF*n*BCLK, or AIF*n*LRCLK.

The FLL1_REFCLK_DIV field controls a programmable divider on the main input reference. The FLL1_SYNCCLK_DIV field controls a programmable divider on the synchronizer input reference. Each input can be divided by 1, 2, 4 or 8. The divider should be set to bring each reference down to 13.5 MHz or below. For best performance, it is recommended that the highest possible frequency—within the 13.5 MHz limit—should be selected.

4.13.8.4 Output Frequency Control—Main Loop

The FLL output frequency, relative to the main input reference F_{REF}, is a function of the following:

- The FLL oscillator frequency, F_{VCO}
- The frequency ratio set by FLL1_FRATIO
- The real number represented by N.K. (N = integer; K = fractional portion)

The F_{VCO} frequency must be in the range 90–104 MHz.

If the FLL is selected as SYSCLK source, the respective F_{VCO} frequency must be exactly 98.304 MHz (for 48 kHz–related sample rates) or 90.3168 MHz (for 44.1 kHz–related sample rates).

If the FLL is selected as DSPCLK source, the DSPCLK frequency is $F_{VCO} \times 1.5$. Note that the DSPCLK can be divided to lower frequencies for clocking the DSP core.

The FLL clock can be configured as a GPIO output; a programmable divider supports division ratios in the range 2 through 127, enabling a wide range of GPIO clock output frequencies.

Note: The chosen F_{VCO} frequency can be used to support multiple outputs simultaneously (e.g., SYSCLK, DSPCLK, and GPIO), as shown in Fig. 4-65.

The FLL oscillator frequency, F_{VCO} is set according to the following equation:

$$F_{VCO} = (F_{REF} \text{ x N.K x FLL1\_FRATIO})$$

The value of N.K can thus be determined as follows:

$$\text{N.K} = F_{VCO} / (\text{FLL1\_FRATIO x } F_{REF})$$

It is recommended to calculate N.K using an initial assumption of FLL1_FRATIO = 1. If N > 1023, FLL1_FRATIO should be incremented until N < 1024.

Note that, in the above equations, the following interpretations are assumed:

- F_{REF} is the input frequency, after division by FLL1_REFCLK_DIV, where applicable
- FLL1_FRATIO is the F_{VCO} clock ratio (1, 2, 3, … 16)

The value of N is held in FLL1_N.

The value of K is determined by the FLL1_THETA and FLL1_LAMBDA fields. In integer mode (K = 0), FLL1_THETA must be set to 0. (FLL1_LAMBDA is ignored in integer mode.) In fractional mode (K > 0), the FLL1_THETA and FLL1_LAMBDA fields can be derived as described in Section 4.13.8.6.

The FLL1_N, FLL1_THETA, and FLL1_LAMBDA fields are all coded as integers (LSB = 1).

The FLL1_CTRL_UPD bit controls the updating of the FLL1_N and FLL1_THETA fields:

- If the FLL1_N or FLL1_THETA fields are updated while the FLL is enabled (FLL1_ENA = 1), the new values are only effective when a 1 is written to FLL1_CTRL_UPD. This makes it possible to update the two fields simultaneously, without disabling the FLL.

  Note that, if the FLL is disabled (FLL1_ENA = 0), the FLL1_N and FLL1_THETA fields can be updated without writing to FLL1_CTRL_UPD.

The FLL1_GAIN and FLL1_PHASE_ENA fields should be set as shown in Table 4-83, depending on F_{REF}, FLL1_THETA, and whether the FLL synchronizer is enabled.

Table 4-83. Selection of FLL1_GAIN and FLL1_PHASE_ENA

Condition		FLL1_GAIN	FLL1_PHASE_ENA
Synchronizer disabled (FLL1_SYNC_ENA = 0) and FLL Integer Mode (FLL1_THETA = 0)	F_{REF} < 768 kHz	0x2	1
	$F_{REF} \geq$ 768 kHz	0x3	
Synchronizer enabled (FLL1_SYNC_ENA = 1) or FLL Fractional Mode (FLL1_THETA > 0)	F_{REF} < 100 kHz	0x0	0
	100 kHz $\leq F_{REF}$ < 375 kHz	0x2	
	375 kHz $\leq F_{REF}$ < 1.5 MHz	0x3	
	1.5 MHz $\leq F_{REF}$ < 6.0 MHz	0x4	
	$F_{REF} \geq$ 6.0 MHz	0x5	

Note: F_{REF} is the input frequency, after division by FLL1_REFCLK_DIV, where applicable.

4.13.8.5 Output Frequency Control—Synchronizer Loop

A similar procedure applies for the derivation of the FLL synchronizer parameters—assuming that this function is used.

The FLL1_SYNC_FRATIO field selects the frequency division ratio of the FLL synchronizer input. The FLL1_GAIN and FLL1_SYNC_DFSAT fields are used to optimize the FLL, according to the input frequency. These fields should be set as described in Table 4-84.

Note: The FLL1_SYNC_FRATIO coding differs from that of FLL1_FRATIO.

Table 4-84. Selection of FLL1_SYNC_FRATIO, FLL1_SYNC_GAIN, FLL1_SYNC_DFSAT

Condition	FLL1_SYNC_FRATIO	FLL1_SYNC_GAIN	FLL1_SYNC_DFSAT
1 MHz $\le F_{SYNC}$ < 13.5 MHz	0x0 (divide by 1)	0x4 (16x gain)	0 (wide bandwidth)
256 kHz $\le F_{SYNC}$ < 1 MHz	0x1 (divide by 2)	0x2 (4x gain)	0 (wide bandwidth)
128 kHz $\le F_{SYNC}$ < 256 kHz	0x2 (divide by 4)	0x0 (1x gain)	0 (wide bandwidth)
64 kHz $\le F_{SYNC}$ < 128 kHz	0x3 (divide by 8)	0x0 (1x gain)	1 (narrow bandwidth)
F_{SYNC} < 64 kHz	0x4 (divide by 16)	0x0 (1x gain)	1 (narrow bandwidth)

Note: F_{SYNC} is the synchronizer input frequency, after division by FLL1_SYNCCLK_DIV, where applicable.

The FLL oscillator frequency, F_{VCO}, is the same frequency calculated as described in Section 4.13.8.4.

The value of N.K (Sync) can then be determined as follows:

N.K (Sync) = F_{VCO} / (FLL1_SYNC_FRATIO x F_{SYNC})

Note that, in the above equation, the following interpretations are assumed:

- F_{SYNC} is the synchronizer input frequency, after division by FLL1_SYNCCLK_DIV, where applicable
- FLL1_SYNC_FRATIO is the F_{VCO} clock ratio (1, 2, 4, 8, or 16)

The value of N (Sync) is held in FLL1_SYNC_N.

The value of K (Sync) is determined by the FLL1_SYNC_THETA and FLL1_SYNC_LAMBDA fields. See Section 4.13.8.6 to derive the recommended settings for these fields.

The FLL1_SYNC_N, FLL1_SYNC_THETA, and FLL1_SYNC_LAMBDA fields are all coded as integers (LSB = 1).

4.13.8.6 Calculation of Theta and Lambda

In Fractional Mode, with the synchronizer disabled (K > 0, and FLL1_SYNC_ENA = 0), FLL1_THETA and FLL1_LAMBDA are calculated with the following steps:

1. Calculate GCD(FLL) using the Greatest Common Denominator function:
   GCD(FLL) = GCD(FLL1_FRATIO x F_{REF}, F_{VCO}),
   where GCD(x, y) is the greatest common denominator of x and y.
   F_{REF} is the input frequency, after division by FLL1_REFCLK_DIV, where applicable.
2. Calculate FLL1_THETA and FLL1_LAMBDA using the following equations:
   FLL1_THETA = (F_{VCO} – (FLL_N x FLL1_FRATIO x F_{REF})) / GCD(FLL)
   FLL1_LAMBDA = (FLL1_FRATIO x F_{REF}) / GCD(FLL)

Note that the values of FLL1_THETA and FLL1_LAMBDA must be coprime (i.e., not divisible by any common integer). The calculation above ensures that the values are coprime. The value of K must be less than 1 (i.e., FLL1_THETA must be less than FLL1_LAMBDA).

If the synchronizer is enabled, the FLL1_SYNC_THETA and FLL1_SYNC_LAMBDA fields are calculated in the same manner described above, using the corresponding synchronizer parameters.

In Fractional Mode, with the synchronizer enabled (K > 0, and FLL1_SYNC_ENA = 1), FLL1_THETA is calculated as FLL1_THETA = K x 65536. The FLL1_LAMBDA field is ignored in this case, and the coprime requirement for FLL1_ LAMBDA and FLL1_THETA is not applicable.

4.13.8.7 Free-Running FLL Mode

The FLL can generate a clock signal even if no external reference is available. This may be because the normal input reference has been interrupted, or may be during a standby or start-up period when no initial reference clock is available.

Free-Running FLL Mode is enabled by setting FLL1_FREERUN. Note that FLL1_ENA must also be enabled in Free-Running FLL Mode.

In Free-Running FLL Mode, the normal feedback mechanism of the FLL is halted and the FLL oscillates independently of the external input references.

If the FLL was previously operating normally (with an input reference clock), the FLL output frequency remains unchanged when Free-Running FLL Mode is enabled. The FLL output is independent of the input reference while operating with FLL1_FREERUN = 1.

The main FLL loop always runs freely if the input reference clock is stopped (regardless of the FLL1_FREERUN setting). If FLL1_FREERUN = 0, the FLL relocks to the input reference whenever it is available.

In Free-Running FLL Mode, (with FLL1_FREERUN = 1), the FLL integrator value (part of the feedback mechanism) can be commanded directly using FLL1_FRC_INTEG_VAL. The integrator value in this field is applied to the FLL when a 1 is written to FLL1_FRC_INTEG_UPD.

If the FLL is started up in Free-Running FLL Mode, (i.e., it was not previously running), the default value of FLL1_FRC_INTEG_VAL is applied.

The FLL integrator value (part of the feedback mechanism) can be read from the FLL1_INTEG field; the value of this field may be stored for later use. Note that the value of FLL1_INTEG is only valid if FLL1_FREERUN = 1 and the FLL1_INTEG_VALID = 1.

The FLL integrator setting does not ensure a specific output frequency for the FLL across all devices and operating conditions; some level of variation applies.

The free-running FLL clock may be selected as the SYSCLK or DSPCLK source, as shown in Fig. 4-63.

4.13.8.8 Spread-Spectrum FLL Control

The CS47L15 can apply modulation to the FLL output, using spread-spectrum techniques. This can be used to control the EMI characteristics of the circuits that are clocked via the FLL.

The FLL can be configured for triangle modulation, zero mean frequency modulation (ZMFM), or dither. The amplitude and frequency parameters of the spread spectrum functions is also programmable, using the fields described in Section 4.13.8.9.

4.13.8.9 FLL Control Registers

The FLL control registers are described in Table 4-85.

Example settings for a variety of reference frequencies and output frequencies are shown in Section 4.13.8.12.

Table 4-85. FLL1 Register Map

Register Address	Bit	Label	Default	Description
R369 (0x0171) FLL1_Control_1	1	FLL1_FREERUN	1	FLL1 Free-Running Mode Enable 0 = Disabled 1 = Enabled The FLL feedback mechanism is halted in Free-Running FLL Mode, and the latest integrator setting is maintained
	0	FLL1_ENA	0	FLL1 Enable 0 = Disabled 1 = Enabled This should be set as the final step of the FLL1 enable sequence, i.e., after the other FLL fields have been configured.
R370 (0x0172) FLL1_Control_2	15	FLL1_CTRL_UPD	0	FLL1 Control Update Write 1 to apply the FLL1_N and FLL1_THETA field settings. (Only valid if FLL1_ENA = 1)
	9:0	FLL1_N[9:0]	0x008	FLL1 Integer multiply for F_{REF} (LSB = 1) If updated while the FLL is enabled, the new value is only effective when a 1 is written to FLL1_CTRL_UPD.

Table 4-85. FLL1 Register Map *(Cont.)*

Register Address	Bit	Label	Default	Description
R371 (0x0173) FLL1_Control_3	15:0	FLL1_ THETA[15:0]	0x0018	FLL1 Fractional multiply for F_{REF}. Sets the numerator (multiply) part of the FLL1_ THETA / FLL1_LAMBDA ratio. Coded as LSB = 1. If updated while the FLL is enabled, the new value is only effective when a 1 is written to FLL1_CTRL_UPD.
R372 (0x0174) FLL1_Control_4	15:0	FLL1_ LAMBDA[15:0]	0x007D	FLL1 Fractional multiply for F_{REF} This field sets the denominator (dividing) part of the FLL1_THETA / FLL1_LAMBDA ratio. Coded as LSB = 1.
R373 (0x0175) FLL1_Control_5	11:8	FLL1_ FRATIO[3:0]	0x0	FLL1 F_{VCO} clock divider 0x0 = 1 0x2 = 3 … 0x1 = 2 0x3 = 4 0xF = 16
R374 (0x0176) FLL1_Control_6	7:6	FLL1_REFCLK_ DIV[1:0]	00	FLL1 Clock Reference Divider 00 = 1 10 = 4 01 = 2 11 = 8 MCLK (or other input reference) must be divided down to ≤13.5 MHz.
	3:0	FLL1_REFCLK_ SRC[3:0]	0000	FLL1 Clock source 0000 = MCLK1 1001 = AIF2BCLK 1101 = AIF2LRCLK 0001 = MCLK2 1010 = AIF3BCLK 1110 = AIF3LRCLK 1000 = AIF1BCLK 1100 = AIF1LRCLK All other codes are reserved
R375 (0x0177) FLL1_Loop_ Filter_Test_1	15	FLL1_FRC_ INTEG_UPD	0	Write 1 to apply the FLL1_FRC_INTEG_VAL setting. (Only valid if FLL1_FREERUN = 1)
	11:0	FLL1_FRC_ INTEG_ VAL[11:0]	0x281	FLL1 Forced Integrator Value
R376 (0x0178) FLL1_NCO_Test_ 0	15	FLL1_INTEG_ VALID	0	FLL1 Integrator Valid. Indicates whether FLL1_INTEG is valid 0 = Not valid 1 = Valid
	11:0	FLL1_ INTEG[11:0]	0x000	FLL1 Integrator Value (Read-only). Indicates the current FLL1 integrator setting. Only valid if FLL1_INTEG_VALID = 1.
R377 (0x0179) FLL1_Control_7	5:2	FLL1_GAIN[3:0]	0000	FLL1 Gain 0000 = 1 0011 = 8 0110 = 64 0001 = 2 0100 = 16 0111 = 128 0010 = 4 0101 = 32 1000–1111 = 256
R378 (0x017A) FLL1_Control_8	11	FLL1_PHASE_ ENA	1	FLL1 Phase Integrator Control 0 = Disabled 1 = Enabled
R385 (0x0181) FLL1_ Synchroniser_1	0	FLL1_SYNC_ ENA	0	FLL1 Synchronizer Enable 0 = Disabled 1 = Enabled This should be set as the final step of the FLL1 synchronizer enable sequence, i.e., after the other synchronizer fields have been configured.
R386 (0x0182) FLL1_ Synchroniser_2	9:0	FLL1_SYNC_ N[9:0]	0x000	FLL1 Integer multiply for F_{SYNC} (LSB = 1)
R387 (0x0183) FLL1_ Synchroniser_3	15:0	FLL1_SYNC_ THETA[15:0]	0x0000	FLL1 Fractional multiply for F_{SYNC} This field sets the numerator (multiply) part of the FLL1_SYNC_THETA / FLL1_ SYNC_LAMBDA ratio. Coded as LSB = 1.
R388 (0x0184) FLL1_ Synchroniser_4	15:0	FLL1_SYNC_ LAMBDA[15:0]	0x0000	FLL1 Fractional multiply for F_{SYNC} This field sets the denominator (dividing) part of the FLL1_SYNC_THETA / FLL1_ SYNC_LAMBDA ratio. Coded as LSB = 1.
R389 (0x0185) FLL1_ Synchroniser_5	10:8	FLL1_SYNC_ FRATIO[2:0]	000	FLL1 Synchronizer F_{VCO} clock divider 000 = 1 010 = 4 1XX = 16 001 = 2 011 = 8

Table 4-85. FLL1 Register Map *(Cont.)*

Register Address	Bit	Label	Default	Description
R390 (0x0186) FLL1_ Synchroniser_6	7:6	FLL1_ SYNCCLK_ DIV[1:0]	00	FLL1 Synchronizer Clock Reference Divider 00 = 1 10 = 4 01 = 2 11 = 8 MCLK (or other input reference) must be divided down to ≤13.5 MHz.
	3:0	FLL1_ SYNCCLK_SRC	0000	FLL1 Synchronizer Clock source 0000 = MCLK1 1001 = AIF2BCLK 1101 = AIF2LRCLK 0001 = MCLK2 1010 = AIF3BCLK 1110 = AIF3LRCLK 1000 = AIF1BCLK 1100 = AIF1LRCLK All other codes are reserved
R391 (0x0187) FLL1_ Synchroniser_7	5:2	FLL1_SYNC_ GAIN[3:0]	0000	FLL1 Synchronizer Gain 0000 = 1 0011 = 8 0110 = 64 0001 = 2 0100 = 16 0111 = 128 0010 = 4 0101 = 32 1000–1111 = 256
	0	FLL1_SYNC_ DFSAT	1	FLL1 Synchronizer Bandwidth 0 = Wide bandwidth 1 = Narrow bandwidth
R393 (0x0189) FLL1_Spread_ Spectrum	5:4	FLL1_SS_ AMPL[1:0]	00	FLL1 Spread Spectrum Amplitude. Controls the extent of the spread-spectrum modulation. 00 = 0.7% (triangle), 0.7% (ZMFM, dither) 10 = 2.3% (triangle), 2.6% (ZMFM, dither) 01 = 1.1% (triangle), 1.3% (ZMFM, dither) 11 = 4.6% (triangle), 5.2% (ZMFM, dither)
	3:2	FLL1_SS_ FREQ[1:0]	00	FLL1 Spread Spectrum Frequency. Controls the spread spectrum modulation frequency in Triangle Mode. 00 = 439 kHz 10 = 1.17 MHz 01 = 878 kHz 11 = 1.76 MHz
	1:0	FLL1_SS_ SEL[1:0]	00	FLL1 Spread Spectrum Select. 00 = Disabled 10 = Triangle 01 = Zero Mean Frequency (ZMFM) 11 = Dither

4.13.8.10 FLL Interrupts and GPIO Output

The CS47L15 provides an FLL lock signal, which indicates whether FLL lock has been achieved (i.e., the FLL is locked to the input reference signal).

The FLL lock signal is an input to the interrupt control circuit and can be used to trigger an interrupt event; see Section 4.12.

The FLL lock signal can be output directly on a GPIO pin as an external indication of the FLL status. See Section 4.11 to configure a GPIO pin for these functions.

Clock output signals derived from the FLL can be output on a GPIO pin. See Section 4.11 to configure a GPIO pin for this function.

The FLL clocking configuration is shown in Fig. 4-65.

4.13.8.11 Example FLL Calculation

The following example illustrates how to derive the FLL1 register fields to generate an oscillator frequency (F_{VCO}) of 98.304 MHz from a 12.000-MHz reference clock (F_{REF}). This is suitable for generating SYSCLK at 98.304 MHz and/or DSPCLK at 147.456 MHz.

1. Set FLL1_REFCLK_DIV to generate $F_{REF} \le 13.5$ MHz:
   FLL1_REFCLK_DIV = 00 (divide by 1)
2. Calculate N.K as given by N.K = F_{VCO} / (FLL1_FRATIO × F_{REF}). Assume FLL1_FRATIO = 0x0 (divide by 1).
   N.K = 98304000 / (1 × 12000000) = 8.192
3. Confirm that the calculated value of N is less than 1024.
4. Determine FLL1_N from the integer portion of N.K:
   FLL1_N = 8 (0x008)

5. Determine GCD(FLL), as given by GCD(FLL) = GCD(FLL1_FRATIO × F_{REF}, F_{VCO}):

   GCD(FLL) = GCD(1 × 12000000, 98304000) = 96000

6. Determine FLL1_THETA, as given by FLL1_THETA = (F_{VCO} – (FLL1_N × FLL1_FRATIO × F_{REF})) / GCD(FLL):

   FLL1_THETA = ((98304000) – (8 × 1 × 12000000)) / 96000
   FLL1_THETA = 24 (0x0018)

7. Determine FLL1_LAMBDA, as given by FLL1_LAMBDA = (FLL1_FRATIO x F_{REF}) / GCD(FLL):

   FLL1_LAMBDA = (1 × 12000000) / 96000
   FLL1_LAMBDA = 125 (0x007D)

8. Determine FLL1_GAIN and FLL1_PHASE_ENA as specified in Section 4.13.8.4:

   FLL1_GAIN = 0x5
   FLL1_PHASE_ENA = 1

4.13.8.12Example FLL Settings

Table 4-86 shows FLL settings for generating an oscillator frequency (F_{VCO}) of 98.304 MHz from a variety of low- and high-frequency reference inputs. This is suitable for generating SYSCLK at 98.304 MHz and/or DSPCLK at 147.456 MHz.

Table 4-86. Example FLL Settings

F_{SOURCE}	F_{VCO} (MHz)[1]	F_{REF} Divider[2]	FRATIO[2]	N.K [3]	FLL1_N	FLL1_ THETA	FLL1_ LAMBDA	FLL1_ GAIN [4]	FLL1_ PHASE_ ENA [4]
32.000 kHz	98.304	1	4	768	0x300	0x0000	0x0001	0x2	1
32.768 kHz	98.304	1	3	1000	0x3E8	0x0000	0x0001	0x2	1
48 kHz	98.304	1	3	682.6667	0x2AA	0x0002	0x0003	0x0	0
128 kHz	98.304	1	1	768	0x300	0x0000	0x0001	0x2	1
512 kHz	98.304	1	1	192	0x0C0	0x0000	0x0001	0x2	1
1.536 MHz	98.304	1	1	64	0x040	0x0000	0x0001	0x3	1
3.072 MHz	98.304	1	1	32	0x020	0x0000	0x0001	0x3	1
11.2896 MHz	98.304	1	1	8.7075	0x008	0x0068	0x0093	0x5	0
12.000 MHz	98.304	1	1	8.192	0x008	0x0018	0x007D	0x5	0
12.288 MHz	98.304	1	1	8	0x008	0x0000	0x0001	0x3	1
13.000 MHz	98.304	1	1	7.5618	0x007	0x0391	0x0659	0x5	0
19.200 MHz	98.304	2	1	10.24	0x00A	0x0006	0x0019	0x5	0
24 MHz	98.304	2	1	8.192	0x008	0x0018	0x007D	0x5	0
26 MHz	98.304	2	1	7.5618	0x007	0x0391	0x0659	0x5	0
27 MHz	98.304	2	1	7.2818	0x007	0x013D	0x0465	0x5	0

1. F_{VCO} = (F_{SOURCE}/F_{REF} Divider) × N.K × FRATIO
2. See Table 4-85 for the coding of the FLL1_REFCLK_DIV and FLL1_FRATIO fields.
3. N.K values are represented in the FLL1_N, FLL1_THETA, and FLL1_LAMBDA fields.
4. Assumes the FLL synchronizer is disabled. If the FLL synchronizer is enabled, see Table 4-83 for required settings.

The FLL synchronizer, when used, is configured similarly to the FLL main loop, using the corresponding register fields. Note that the recommended FRATIO and GAIN settings on the FLL synchronizer circuit differ from those of the main loop—the FLL1_SYNC_FRATIO, FLL1_GAIN and FLL1_SYNC_DFSAT fields are set as described in Table 4-84.

Note that the FLL1_SYNC_FRATIO coding differs from that of FLL1_FRATIO.

Table 4-87 shows FLL synchronizer settings for generating an oscillator frequency (F_{VCO}) of 98.304 MHz from a variety of low- and high-frequency reference inputs.

Table 4-87. Example FLL Synchronizer Settings

F_{SOURCE}	F_{VCO} (MHz) [1]	F_{REF} Divider[2]	FRATIO[2]	N.K [3]	FLL1_ SYNC_N	FLL1_ SYNC_ THETA	FLL1_ SYNC_ LAMBDA	FLL1_ SYNC_ GAIN	FLL1_ SYNC_ DFSAT
32.000 kHz	294.912	1	16	192	0x0C0	0x0000	0x0001	0x0	1
32.768 kHz	294.912	1	16	187.5	0x0BB	0x0001	0x0002	0x0	1
48 kHz	294.912	1	16	128	0x080	0x0000	0x0001	0x0	1
128 kHz	294.912	1	4	192	0x0C0	0x0000	0x0001	0x0	0
512 kHz	294.912	1	2	96	0x060	0x0000	0x0001	0x2	0
1.536 MHz	294.912	1	1	64	0x040	0x0000	0x0001	0x4	0
3.072 MHz	294.912	1	1	32	0x020	0x0000	0x0001	0x4	0
11.2896 MHz	294.912	1	1	8.7075	0x008	0x0068	0x0093	0x4	0
12.000 MHz	294.912	1	1	8.192	0x008	0x0018	0x007D	0x4	0
12.288 MHz	294.912	1	1	8	0x008	0x0000	0x0001	0x4	0
13.000 MHz	294.912	1	1	7.5618	0x007	0x0391	0x0659	0x4	0
19.200 MHz	294.912	2	1	10.24	0x00A	0x0006	0x0019	0x4	0
24 MHz	294.912	2	1	8.192	0x008	0x0018	0x007D	0x4	0
26 MHz	294.912	2	1	7.5618	0x007	0x0391	0x0659	0x4	0
27 MHz	294.912	2	1	7.2818	0x007	0x013D	0x0465	0x4	0

1. $F_{VCO} = (F_{SOURCE}/F_{REF}$ Divider) × N.K × FRATIO
2. See Table 4-85 for the coding of the FLL1_REFCLK_DIV and FLL1_SYNC_FRATIO fields.
3. N.K values are represented in the FLL1_SYNC_N, FLL1_SYNC_THETA, and FLL1_SYNC_LAMBDA fields.

4.13.9 Frequency-Locked Loop (FLL_AO)

Two integrated FLLs are provided to support the clocking requirements of the CS47L15. These can be configured according to the available reference clocks and the application requirements. The reference clock may use a high frequency (e.g., 12.288 MHz) or low frequency (e.g., 32.768 kHz). The FLL is tolerant of jitter and may be used to generate a stable output clock from a less stable input reference.

There are two FLL implementations on the CS47L15:

- FLL1 provides an advanced capability to use more than one reference clock to achieve best performance. See Section 4.13.8.
- FLL_AO is low-power FLL that supports additional always-on capability to provide system clocking when other references are unavailable or disabled. FLL_AO is described in the following subsections. Note that FLL_AO is designed to support low-power always-on use cases only; for hi-fi audio use cases, it is recommended to use FLL1.

4.13.9.1 Overview

The FLL_AO characteristics are summarized in Table 3-11. In normal operation, the FLL output is frequency-locked to an input clock reference. The FLL can also be used to generate a free-running clock in the absence of any external reference, as described in Section 4.13.9.5.

FLL_AO is a low-power FLL that can be configured as the source for SYSCLK or DSPCLK system clocks. It also supports always-on functions—it can be used to provide clocking for the DSP core if DSPCLK is not enabled (e.g., for always-on DSP applications). See Section 4.4.3.4 to configure FLL_AO for always-on DSP operation.

The default FLL_AO settings are configured to provide a 49.152-MHz output, without any input reference required. The FLL_AO can be used in its default settings or can be reconfigured for different input/output frequencies. The FLL_AO control registers must always hold valid settings—either enabled and locked to an input reference clock or configured in FLL Hold Mode.

FLL_AO takes a constant and stable clock reference as its input. Under typical application conditions, a low-frequency (e.g., 32.768 kHz) reference is used. FLL_AO is free running without any clock reference if the input signal is removed; it can also initiate an output in the absence of any reference signal.

4.13.9.2 FLL Enable

FLL_AO is enabled by setting FLL_AO_ENA. In normal operation, the FLL_AO output is frequency locked to the selected input reference.

FLL_AO supports free-running operation in FLL Hold Mode, using the FLL_AO_HOLD bit described in Section 4.13.9.5. If the FLL is enabled and FLL Hold Mode is selected, the configured output frequency is maintained without any input reference required. Note that, once the FLL output has been established, FLL_AO always runs freely if the input reference clock is stopped, regardless of the FLL_AO_HOLD bit.

To disable FLL_AO, FLL_AO_HOLD must be set before clearing FLL_AO_ENA. FLL_AO_HOLD must always be set if the FLL is disabled; this holds the oscillator loop-configuration settings, in readiness for always-on system requirements.

FLL_AO_HOLD should remain set when enabling FLL_AO. If normal (input-reference locked) FLL operation is required, FLL_AO_HOLD should be cleared after FLL_AO_ENA has been set.

When changing FLL_AO settings, FLL_AO_HOLD must be set before writing to the configuration registers. FLL_AO_HOLD must not be cleared until after the new register values have been written. Note that, if the FLL is disabled, the FLL_AO_HOLD bit must remain set until after FLL_AO_ENA has been set.

Under default conditions, FLL_AO is preconfigured to generate 49.152-MHz output, without any input reference required. Setting FLL_AO_ENA without changing any other control bits enables this reference clock output, which may be selected as SYSCLK or DSPCLK source, as shown in Fig. 4-63.

The FLL_AO configuration is shown in Fig. 4-66.

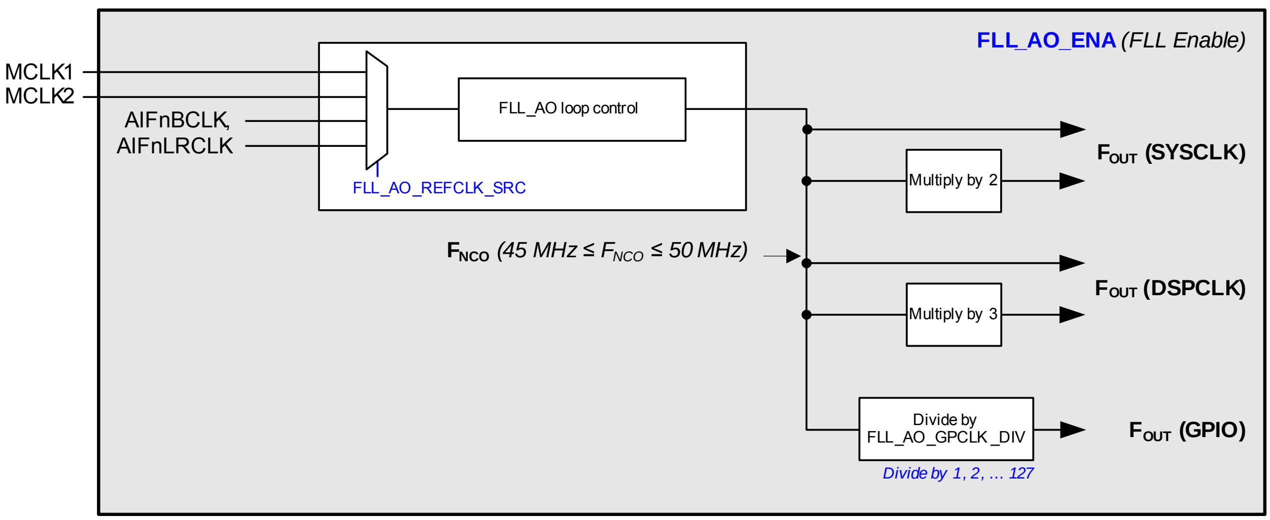


Figure 4-66. FLL_AO Configuration

The procedure for configuring FLL_AO is described in the following subsections. The associated register control fields are described in Table 4-88.

4.13.9.3 Input Frequency Control

The main input reference is selected using FLL_AO_REFCLK_SRC. The available options in each case are MCLK1, MCLK2, AIF*n*BCLK, or AIF*n*LRCLK.

The FLL_AO reference clock provides input to the interrupt control circuit and can be used to trigger an interrupt event when the input reference is stopped; see Section 4.12.

4.13.9.4 Output Frequency Control

If FLL_AO is selected as SYSCLK source, the associated multiplexer can select the FLL_AO oscillator frequency (equal to F_{NCO}) or a multiplied frequency (equal to F_{NCO} x 2). For hi-fi audio use, F_{NCO} must be exactly 49.152 MHz for 48 kHz–related sample rates or 45.1584 MHz for 44.1 kHz–related sample rates.

If FLL_AO is selected as DSPCLK source, the associated multiplexer can select the basic frequency (equal to F_{NCO}) or a multiplied frequency (equal to F_{NCO} x 3). Note that the DSPCLK can be divided to lower frequencies for clocking the DSP core.

If FLL_AO is selected as a GPIO output, a programmable divider supports division ratios in the range 1 through 127, enabling a wide range of GPIO clock output frequencies.

Note: The chosen F_{NCO} frequency can be used to support multiple outputs simultaneously (e.g., SYSCLK and DSPCLK); each FLL clock output path is controlled by a separate divider function, as shown in Fig. 4-66.

4.13.9.5 FLL Hold Mode

FLL Hold Mode enables the FLL to generate a clock signal even if no external reference clock is available, such as when the normal input reference has been interrupted during a standby or start-up period. FLL Hold Mode is selected by setting FLL_AO_HOLD.

- If the FLL is enabled and FLL Hold Mode is selected, the normal feedback mechanism of the FLL is halted and the FLL oscillates independently of the external input references—the FLL output frequency remains unchanged if FLL Hold Mode is enabled.
- If the FLL is enabled and the input reference clock is stopped, the loop always runs freely, regardless of the FLL_AO_HOLD setting. If FLL_AO_HOLD = 0, the FLL relocks to the input reference whenever it is available.
- If the FLL is disabled and FLL Hold Mode is selected, the latest oscillator loop configuration is held for later use. Note that this is the default condition of FLL_AO: preconfigured to generate 49.152-MHz output with no input reference required.

Note: For specified CS47L15 functionality, FLL_AO_HOLD must be set before disabling the FLL and must always be set if the FLL is disabled.

4.13.9.6 FLL Control Registers

The FLL_AO control registers are described in Table 4-88.

Example settings for a variety of reference frequencies and output frequencies are shown in Section 4.13.9.8.

Table 4-88. FLL_AO Register Map

Register Address	Bit	Label	Default	Description
R465 (0x01D1) FLL_AO_Control_1	2	FLL_AO_HOLD	1	FLL_AO Hold Mode Enable 0 = Disabled 1 = Enabled The FLL feedback mechanism is halted in FLL Hold Mode, and the latest integrator setting is maintained. This bit must always be set if FLL_AO is disabled.
	0	FLL_AO_ENA	0	FLL_AO Enable 0 = Disabled 1 = Enabled
R470 (0x01D6) FLL_AO_Control_6	3:0	FLL_AO_REFCLK_SRC[3:0]	0100	FLL_AO Clock source 0000 = MCLK1 0001 = MCLK2 1000 = AIF1BCLK 1001 = AIF2BCLK 1010 = AIF3BCLK 1100 = AIF1LRCLK 1101 = AIF2LRCLK 1110 = AIF3LRCLK All other codes are reserved

4.13.9.7 FLL Interrupts and GPIO Output

For each FLL, the CS47L15 provides an FLL lock signal, which indicates whether FLL lock has been achieved (i.e., the FLL is locked to the input reference signal).

The FLL lock signals are inputs to the interrupt control circuit and can be used to trigger an interrupt event; see Section 4.12.

The FLL lock signal can be output directly on a GPIO pin as an external indication of the FLL status. See Section 4.11 to configure a GPIO pin for these functions.

Clock output signals derived from the FLL can be output on a GPIO pin. See Section 4.11 to configure a GPIO pin for this function.

The FLL_AO configuration is shown in Fig. 4-66.

4.13.9.8 Example FLL Settings

Table 4-89 shows FLL settings for generating an oscillator frequency (F_{NCO}) of 45.1854 MHz or 49.152 MHz from a variety of low-frequency reference inputs.

Table 4-89. Example FLL_AO Settings

Input Reference	Configuration Sequence—45.1584 MHz output	Configuration Sequence—49.152 MHz output
32.000 kHz	• Write 0x02C1 to address 0x01D2 • Write 0x0003 to address 0x01D3 • Write 0x0005 to address 0x01D4 • Write 0x0002 to address 0x01D5 • Write 0x8001 to address 0x01D6 • Write 0x0004 to address 0x01D8 • Write 0x0077 to address 0x01DA • Write 0x06D8 to address 0x01DC • Write 0x0005 to address 0x01DD • Write 0x82C1 to address 0x01D2	• Write 0x0300 to address 0x01D2 • Write 0x0000 to address 0x01D3 • Write 0x0001 to address 0x01D4 • Write 0x0002 to address 0x01D5 • Write 0x8001 to address 0x01D6 • Write 0x0004 to address 0x01D8 • Write 0x0077 to address 0x01DA • Write 0x06D8 to address 0x01DC • Write 0x0085 to address 0x01DD • Write 0x8300 to address 0x01D2
32.768 kHz	• Write 0x02B1 to address 0x01D2 • Write 0x0001 to address 0x01D3 • Write 0x0010 to address 0x01D4 • Write 0x0002 to address 0x01D5 • Write 0x8001 to address 0x01D6 • Write 0x0004 to address 0x01D8 • Write 0x0077 to address 0x01DA • Write 0x06D8 to address 0x01DC • Write 0x0005 to address 0x01DD • Write 0x82B1 to address 0x01D2	• Write 0x02EE to address 0x01D2 • Write 0x0000 to address 0x01D3 • Write 0x0001 to address 0x01D4 • Write 0x0002 to address 0x01D5 • Write 0x8001 to address 0x01D6 • Write 0x0004 to address 0x01D8 • Write 0x0077 to address 0x01DA • Write 0x06D8 to address 0x01DC • Write 0x0085 to address 0x01DD • Write 0x82EE to address 0x01D2
44.100 kHz	• Write 0x0200 to address 0x01D2 • Write 0x0000 to address 0x01D3 • Write 0x0001 to address 0x01D4 • Write 0x0002 to address 0x01D5 • Write 0x8001 to address 0x01D6 • Write 0x0004 to address 0x01D8 • Write 0x0077 to address 0x01DA • Write 0x06D8 to address 0x01DC • Write 0x0085 to address 0x01DD • Write 0x8200 to address 0x01D2	• Write 0x022D to address 0x01D2 • Write 0x0029 to address 0x01D3 • Write 0x0093 to address 0x01D4 • Write 0x0002 to address 0x01D5 • Write 0x8001 to address 0x01D6 • Write 0x0004 to address 0x01D8 • Write 0x0077 to address 0x01DA • Write 0x06D8 to address 0x01DC • Write 0x0005 to address 0x01DD • Write 0x822D to address 0x01D2
48.000 kHz	• Write 0x01D6 to address 0x01D2 • Write 0x0002 to address 0x01D3 • Write 0x0005 to address 0x01D4 • Write 0x0002 to address 0x01D5 • Write 0x8001 to address 0x01D6 • Write 0x0004 to address 0x01D8 • Write 0x0077 to address 0x01DA • Write 0x06D8 to address 0x01DC • Write 0x0005 to address 0x01DD • Write 0x81D6 to address 0x01D2	• Write 0x0200 to address 0x01D2 • Write 0x0000 to address 0x01D3 • Write 0x0001 to address 0x01D4 • Write 0x0002 to address 0x01D5 • Write 0x8001 to address 0x01D6 • Write 0x0004 to address 0x01D8 • Write 0x0077 to address 0x01DA • Write 0x06D8 to address 0x01DC • Write 0x0085 to address 0x01DD • Write 0x8200 to address 0x01D2

Notes: For correct FLL_AO configuration, the register values must be written in the sequence shown. The sequence must be executed in full, regardless of the previous contents of the respective registers.

The example FLL_AO settings assume MCLK2 is input source. The register 0x01D6 value should be amended, if a different input source is used. See Table 4-88 for the applicable register field definitions.

To enable the FLL_AO output, the FLL_AO_HOLD and FLL_AO_ENA control bits must also be written. See Section 4.13.9.2 for further details.

4.14 Control Interface and Master-Boot Interface

The CS47L15 supports a control interface for read/write access to its control registers. The control interface is a slave interface and can be configured in 4-wire SPI or 2-wire I^{2}C modes.

The CS47L15 also supports a master interface that can be used to download firmware and register-configuration data from an external non-volatile memory (e.g., EEPROM or flash memory). This enables the device to self-boot to an application-specific configuration and to be used independently of a host processor. The master interface operates in 4-wire SPI mode.

The control interface and master-boot interface selection is configured at power-up and following hardware reset, according to the logic level applied to the MSTRBOOT, SPISCLK, and SPISS pins. This is described in Table 4-90.

If the master-boot function is selected, the SPI interface pins are assigned to the master-boot interface (for connection to an external memory). In this case, the GPIO13 and GPIO14 pins support an I^{2}C control interface—this is intended for development purposes and can be used to provide register access for a debug tool if the master-boot function is selected. The I^{2}C control interface is enabled by default (if the master-boot function is selected); it can be disabled by clearing I2C_DEBUG.

Note that, if the digital speaker (PDM) interface is required following the master-boot start-up configuration, the I^{2}C control interface on GPIO13/14 must be disabled. The external memory can be programmed to disable the I^{2}C control interface.

Table 4-90. Control Interface and Master-Boot Interface Selection

MSTRBOOT	SPISCLK	SPISS	Control Interface Configuration	Master Boot Interface Configuration
Logic 0	Logic 0	Logic 1	Slave I^{2}C: • SDA—Data input/output • SCLK—Interface clock input	—
	—	—	Slave SPI: • SPIMISO—Data output • SPIMOSI—Data input • SPISCLK—Interface clock input • SPISS—Slave select input	—
Logic 1	—	—	Slave I^{2}C: • GPIO13—Data input/output (SDA) • GPIO14—Interface clock input (SCLK) **Note:** Slave I^{2}C interface can be disabled (e.g., to support SPKCLK/SPKTXDAT functions)	Master SPI: • SPIMISO—Data input • SPIMOSI—Data output • SPISCLK—Interface clock output • SPISS—Slave select output

The control interface and master-boot interface configurations are illustrated in Fig. 4-67, Fig. 4-68, and Fig. 4-69.

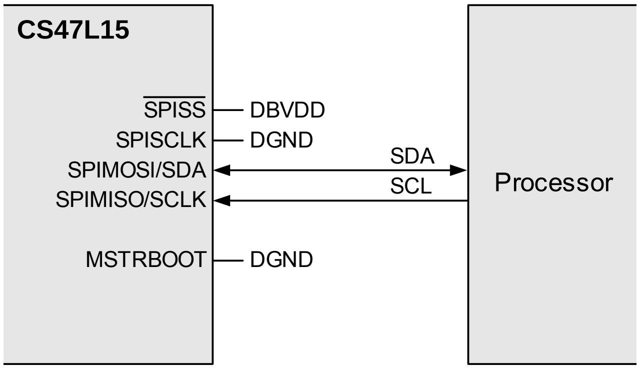


Figure 4-67. I2C Slave Control Interface

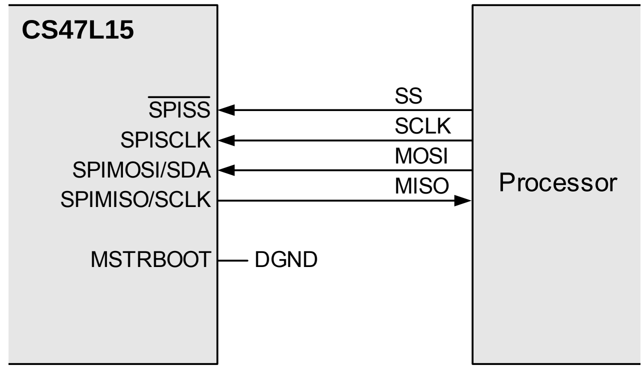


Figure 4-68. SPI Slave Control Interface

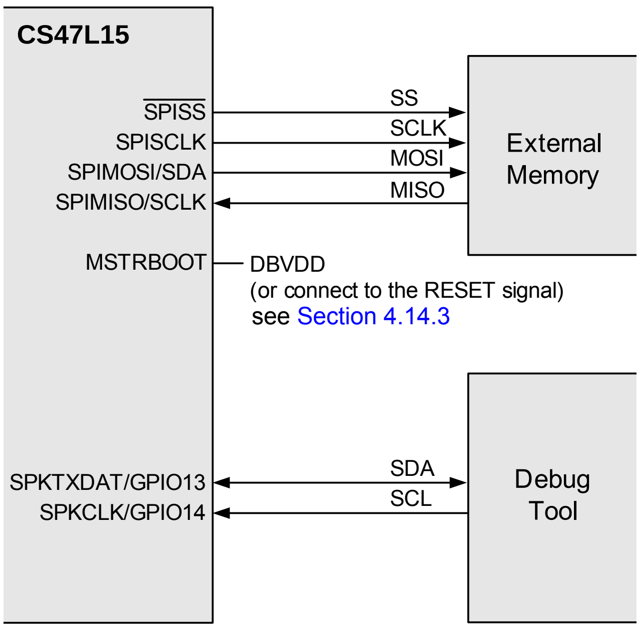


Figure 4-69. I2C Slave and SPI Master Interfaces

The control interface function can be supported with or without system clocking—there is no requirement for SYSCLK, or any other system clock, to be enabled when accessing the register map.

The CS47L15 executes a boot sequence following power-on reset, hardware reset, software reset, or wake-up from Sleep Mode. Note that control register writes should not be attempted until the boot sequence has completed. See Section 4.19.1 for further details.

The CS47L15 provides an integrated pull-down resistor on the SPIMISO/SCLK pin. This provides a flexible capability for interfacing with other devices. A pull-down resistor is also provided on the MSTRBOOT pin. The pull-downs are controlled using the MISO_SCLK_PD and MSTRBOOT_PD bits, as described in Table 4-91.

Note: When writing to the MISO_SCLK_PD bit, take care not to change other nonzero bits that are configured at the same register address.

Table 4-91. Control Interface Pull-Down

Register Address	Bit	Label	Default	Description
R8 (0x0008) Ctrl_IF_CFG_1	7	MISO_SCLK_PD	0	SPIMISO/SCLK Pull-Down Control 0 = Disabled 1 = Enabled
R18 (0x0012) Ctrl_IF_Pin_Cfg_1	10	I2C_DEBUG	1	I2C Debug Interface Control 0 = Disabled 1 = Enabled The I2C debug interface is supported on the GPIO13/GPIO14 pins if master-boot is selected.
R334 (0x014E) Clock_Gen_Pad_Ctrl	9	MSTRBOOT_PD	1	MSTRBOOT Pull-Down Control 0 = Disabled 1 = Enabled

A detailed description of the SPI and I2C control interface modes is provided in Section 4.14.3 and Section 4.14.3. The master-boot interface function is described in Section 4.14.3.

4.14.1 Four-Wire (SPI) Control Interface

The SPI control interface mode uses the SS, SCLK, MOSI, and MISO pin functions, as described in Table 4-90.

In write operations (R/W = 0), the MOSI pin input is driven by the controlling device.

In read operations (R/W = 1), the MOSI pin is ignored following receipt of the valid register address.

If SS is asserted (Logic 0), the MISO output is actively driven when outputting data and is high impedance at other times. If SS is not asserted, the MISO output is high impedance.

The high-impedance state of the MISO output allows the pin to be shared with other slaves. An internal pull-down resistor can be enabled on the SPIMISO pin, as described in Table 4-91.

Data transfers in SPI mode must use the applicable SPI message format, according to the register address space that is being accessed:

- When accessing register addresses below R12288 (0x3000), the applicable SPI protocol comprises a 31-bit register address and 16-bit data words.
- When accessing register addresses from R12888 (0x3000) upwards, the applicable SPI protocol comprises a 31-bit register address and 32-bit data words.
- Note that, in all cases, the complete SPI message protocol also includes a read/write bit and a 16-bit padding phase (see Fig. 4-70 and Fig. 4-71 below).

Continuous read and write modes enable multiple register operations to be scheduled faster than is possible with single register operations. In these modes, the CS47L15 automatically increments the register address at the end of each data word, for as long as SS is held low and SCLK is toggled. Successive data words can be input/output every 16 (or 32) clock cycles (depending on the applicable register address space).

The SPI protocol is shown in Fig. 4-70 and Fig. 4-71. Note that 16-bit data words are shown, but the equivalent protocol also applies to 32-bit data words.

Fig. 4-70 shows a single register write to a specified address.

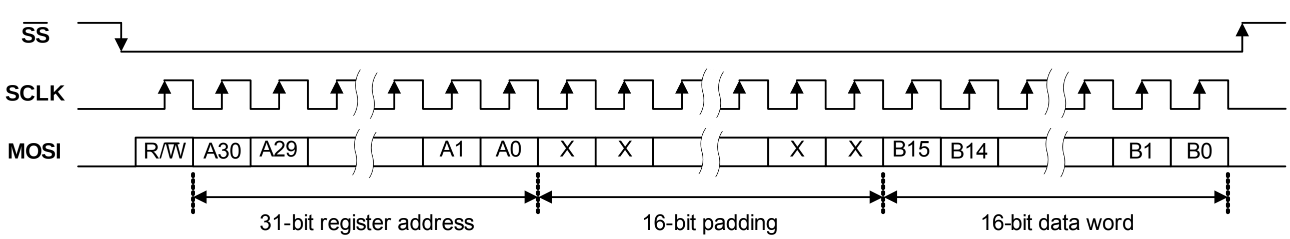


Figure 4-70. Control Interface SPI Register Write (16-Bit Data Words)

Fig. 4-71 shows a single register read from a specified address.

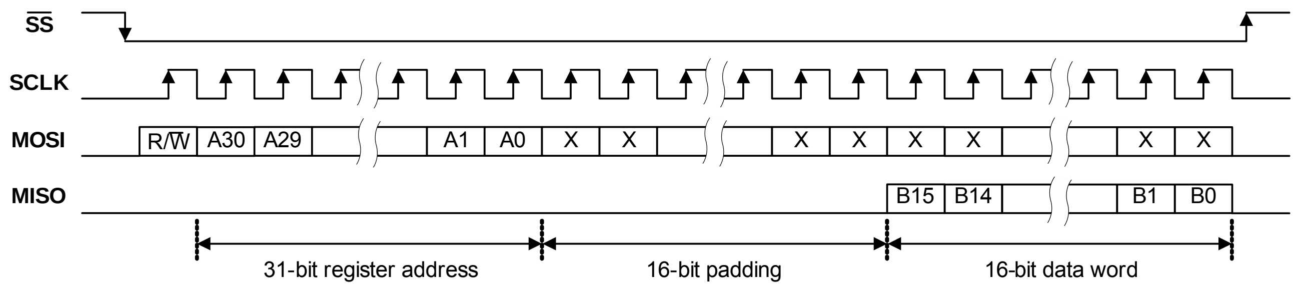


Figure 4-71. Control Interface SPI Register Read (16-Bit Data Words)

See Table 3-20 for a detailed timing specification of the SPI control interface.

4.14.2 Two-Wire (I2C) Control Interface

The I2C control interface mode uses the SCLK and SDA pin functions, as described in Table 4-90.

In I2C Mode, the CS47L15 is a slave device on the control interface; SCLK is a clock input, while SDA is a bidirectional data pin. To allow arbitration of multiple slaves (and/or multiple masters) on the same interface, the CS47L15 transmits Logic 1 by tristating the SDA pin, rather than pulling it high. An external pull-up resistor is required to pull the SDA line high so that the Logic 1 can be recognized by the master.

In order to allow many devices to share a single two-wire control bus, every device on the bus has a unique 8-bit device ID (this is not the same as the address of each register in the CS47L15).

The CS47L15 device ID is 0011_0100 (0x34). Note that the LSB of the device ID is the read/write bit; this bit is set to Logic 1 for read and Logic 0 for write.

The CS47L15 operates as a slave device only. The controller indicates the start of data transfer with a high-to-low transition on SDA while SCLK remains high. This indicates that a device ID and subsequent address/data bytes follow. The CS47L15 responds to the start condition and shifts in the next 8 bits on SDA (8-bit device ID, including read/write bit, MSB first). If the device ID received matches the device ID of the CS47L15, the CS47L15 responds by pulling SDA low on the next clock pulse (ACK). If the device ID is not recognized or the R/W bit is set incorrectly, the CS47L15 returns to the idle condition and waits for a new start condition.

If the device ID matches the device ID of the CS47L15, the data transfer continues. The controller indicates the end of data transfer with a low-to-high transition on SDA while SCLK remains high. After receiving a complete address and data sequence the CS47L15 returns to the idle state and waits for another start condition. If a start or stop condition is detected out of sequence at any point during data transfer (i.e., SDA changes while SCLK is high), the device returns to the idle condition.

Data transfers in I2C mode must use the applicable I2C message format, according to the register address space that is being accessed:

- When accessing register addresses below R12288 (0x3000), the applicable I2C protocol comprises a 32-bit register address and 16-bit data words.
- When accessing register addresses from R12888 (0x3000) upwards, the applicable I2C protocol comprises a 32-bit register address and 32-bit data words.
- Note that, in all cases, the complete I2C message protocol also includes a device ID, a read/write bit, and other signaling bits (see Fig. 4-72 and Fig. 4-73).

The CS47L15 supports the following read and write operations:

- Single write
- Single read
- Multiple write
- Multiple read

Continuous (multiple) read and write modes allow register operations to be scheduled faster than is possible with single register operations. In these modes, the CS47L15 automatically increments the register address after each data word. Successive data words can be input/output every 2 (or 4) data bytes, depending on the applicable register address space.

The I2C protocol for a single, 16-bit register write operation is shown in Fig. 4-72.

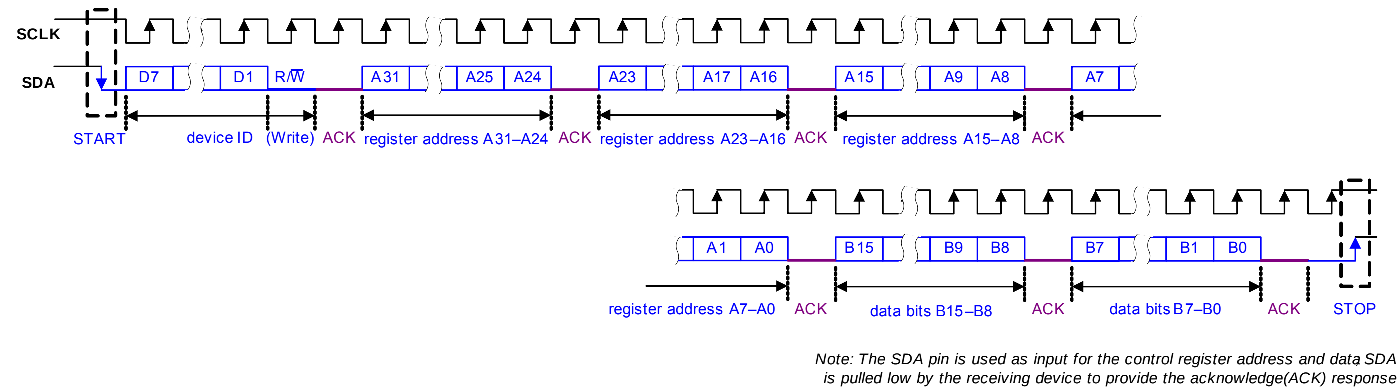


Figure 4-72. Control Interface I2C Register Write (16-Bit Data Words)

The I2C protocol for a single, 16-bit register read operation is shown in Fig. 4-73.

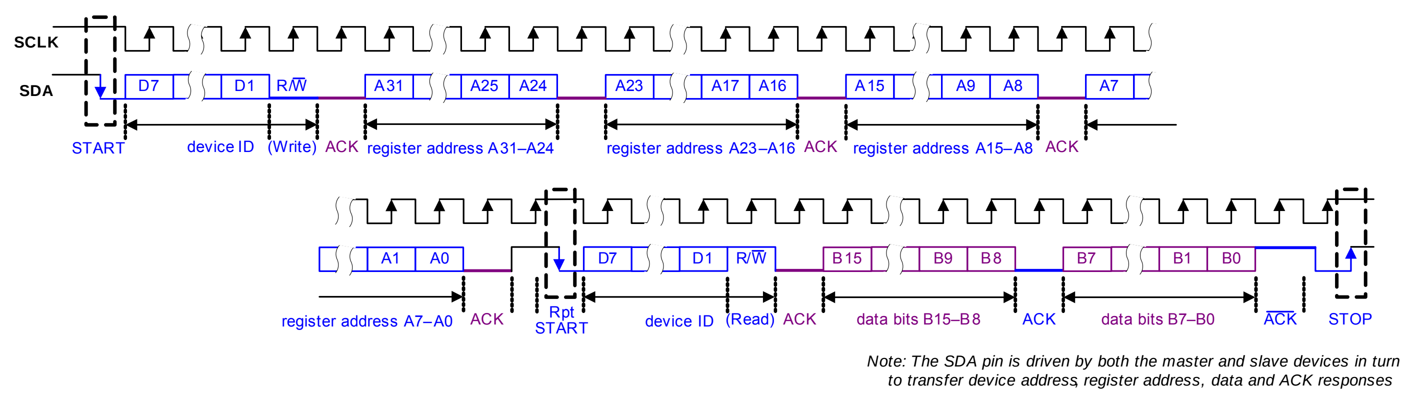


Figure 4-73. Control Interface I2C Register Read (16-Bit Data Words)

See Table 3-19 for a detailed timing specification of the I2C control interface.

The control interface also supports other register operations; the interface protocol for these operations is shown in Fig. 4-74 through Fig. 4-77. The terminology used in the following figures is detailed in Table 4-92.

Note that 16-bit data words are shown in these illustrations. The equivalent protocol is also applicable to 32-bit words, with 4 data bytes transmitted (or received) instead of 2.

Table 4-92. Control Interface (I2C) Terminology

Terminology	Description
S	Start condition
Sr	Repeated start
A	Acknowledge (SDA low)
$\overline{A}$	Not acknowledge (SDA high)

Table 4-92. Control Interface (I²C) Terminology *(Cont.)*

Terminology	Description
P	Stop condition
R/$\overline{W}$	Read/not write 0 = Write; 1 = Read
[White field]	Data flow from bus master to CS47L15
[Gray field]	Data flow from CS47L15 to bus master

Fig. 4-74 shows a single register write to a specified address.

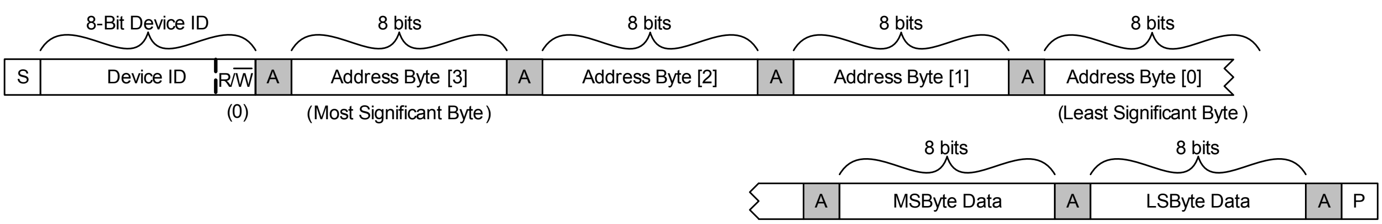


Figure 4-74. Single-Register Write to Specified Address

Fig. 4-75 shows a single register read from a specified address.

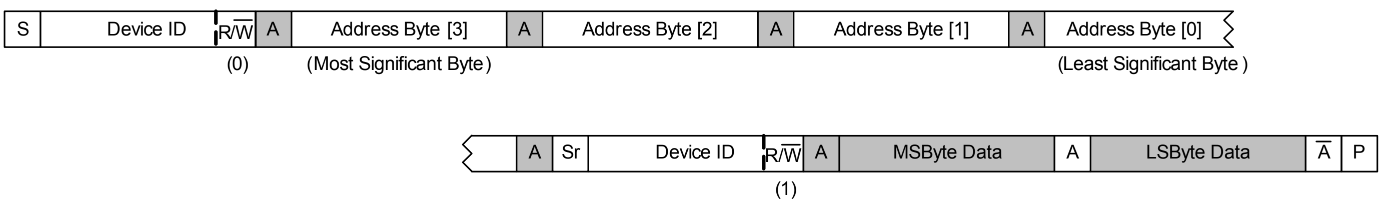


Figure 4-75. Single-Register Read from Specified Address

Fig. 4-76 shows a multiple register write to a specified address.

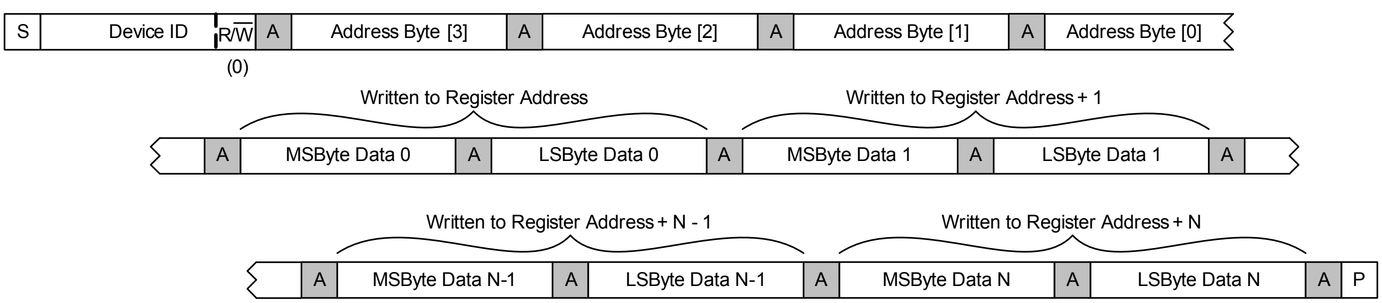


Figure 4-76. Multiple-Register Write to Specified Address

Fig. 4-77 shows a multiple register read from a specified address.

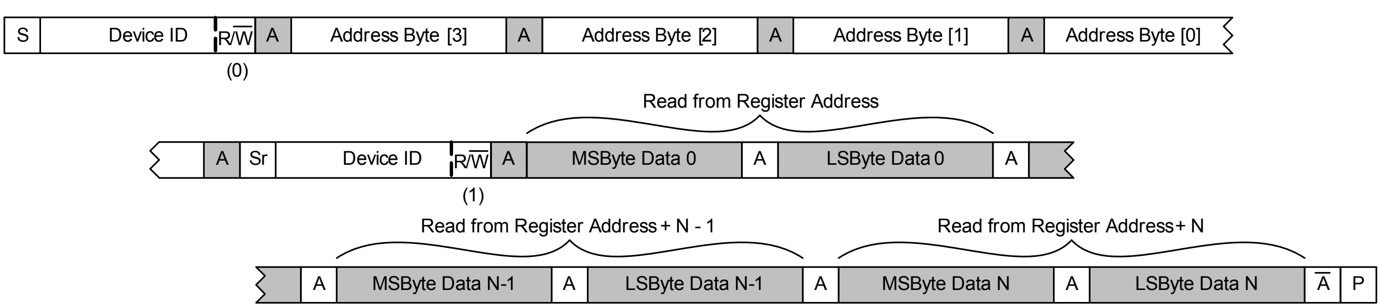


Figure 4-77. Multiple-Register Read from Specified Address

4.14.3 SPI Master-Boot Interface

The SPI master-boot interface mode uses the SS, SCLK, MOSI, and MISO pin functions, as described in Table 4-90. The interface connects directly to an external non-volatile memory (e.g., EEPROM or flash memory), enabling the CS47L15 to self-boot to an application-specific configuration and to be used independently of a host processor.

The SPI master-boot interface is selected using the MSTRBOOT pin—if a Logic 1 is detected on the MSTRBOOT pin during device start-up, the CS47L15 downloads the firmware and register-configuration data over the SPI master interface. This self-boot function is scheduled as part of power-on reset or hardware reset (assuming a Logic 1 is detected on the MSTRBOOT pin).

Note that, if a Logic 1 is applied to the MSTRBOOT pin, the output pins of the SPI interface are actively driven—including during reset. To allow programming of the external memory, the output pins of the SPI interface must be tristated by applying a Logic 0 to the MSTRBOOT input. The CS47L15 should be held in reset during memory programming by asserting the RESET input (Logic 0) as described in Section 4.19.2. It is recommended to connect the RESET and MSTRBOOT pins as shown in Fig. 4-78.

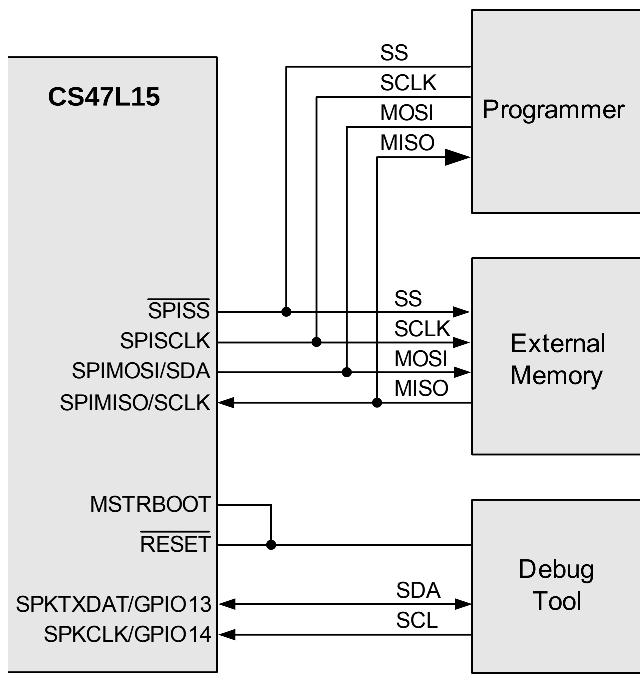


Figure 4-78. SPI Master-Boot Connections

The external memory data contents are compiled using a dedicated application-development tool. The compiled data includes a boot header that contains identifier fields, interface timing parameters, CRC data, and other fields that describe the associated data packets. The firmware and register-configuration data is contained within data packets; these may be formatted in a number of different ways to optimize the overall file size and electrical/timing requirements. Please contact your local Cirrus Logic representative for details of the external memory development tool.

The CS47L15 reads the external memory using SPI Mode 0 bus protocol. Two types of SPI read instruction are supported—the standard read instruction is used by default; the fast read instruction is used if the external memory contents are configured to enable this option.

Continuous read modes are used to enable multiple register operations to be scheduled faster than is possible with single register operations. In these modes, the CS47L15 (and the external memory) automatically increment the register address at the end of each data word, for as long as SS is held low and SCLK is toggled.

The standard read instruction is shown in Fig. 4-79.

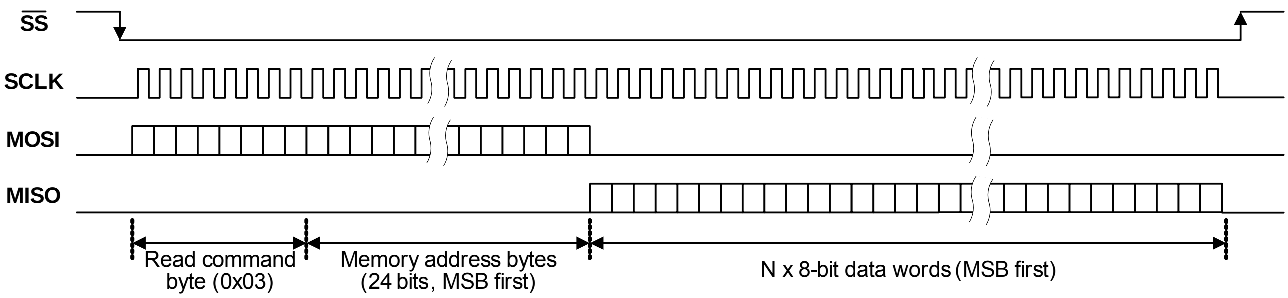


Figure 4-79. SPI Master Standard Read Instruction

The fast read instruction is shown in Fig. 4-80.

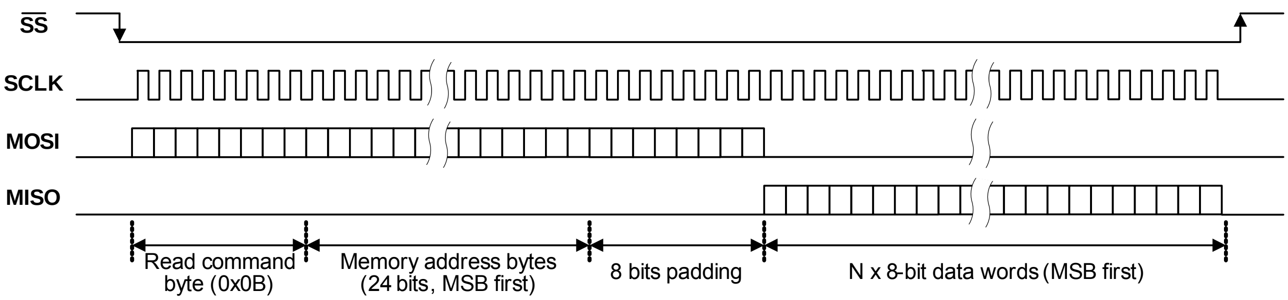


Figure 4-80. SPI Master Fast Read Instruction

See Table 3-21 for a detailed timing specification of the SPI master interface.

Refer to Section 5.1.9 for recommended external memory components.

4.15 Control-Write Sequencer

The control-write sequencer is a programmable unit that forms part of the CS47L15 control interface logic. It provides the ability to perform a sequence of register-write operations with the minimum of demands on the host processor—the sequence may be initiated by a single operation from the host processor and then left to execute independently.

Default sequences for pop-suppressed start-up and shutdown of each headphone/earpiece output driver are provided (these are scheduled automatically when the respective output paths are enabled or disabled). Other control sequences can be programmed, and may be associated with sample-rate detection, DRC, MICDET clamp, or event-logger status; these sequences are automatically scheduled whenever a corresponding event is detected.

When a sequence is initiated, the sequencer performs a series of predefined register writes. The start index of a control sequence within the sequencer's memory may be commanded directly by the host processor. The applicable start index for each of the sequences associated with sample-rate detection, DRC, or MICDET clamp, or event logger status is held in a user-programmed control register.

The control-write sequencer may be triggered by a number of different events. Multiple sequences are queued if necessary, and each is scheduled in turn.

The control-write sequencer can be supported with or without system clocking—there is no requirement for SYSCLK or for any other system clock to be enabled when using the control-write sequencer. The timing accuracy of the sequencer operation is improved when SYSCLK is present, but the general functionality is supported with or without SYSCLK.

4.15.1 Initiating a Sequence

The fields associated with running the control-write sequencer are described in Table 4-93.

The CS47L15 provides 16 general-purpose trigger bits for the write sequencer to allow easy triggering of the associated control sequences. Writing 1 to the trigger bit initiates a control sequence, starting at the respective index position within the control-write sequencer memory.

The WSEQ_TRG1_INDEX field defines the sequencer start index corresponding to the WSEQ_TRG1 trigger control bit. Equivalent start index fields are provided for each of the trigger control bits, as described in Table 4-93. Note that a sequencer start index of 0x1FF causes the respective sequence to be aborted.

The general-purpose control sequences are undefined following power-on reset, a hardware reset, or a Sleep Mode transition. The general-purpose control sequences must be reconfigured by the host processor following any of these events. Note that all control sequences are maintained in the sequencer memory through software reset.

The write sequencer can also be commanded using control bits in register R22 (0x16). In this case, the write sequencer is enabled using the WSEQ_ENA bit and the index location of the first command in the sequence is held in the WSEQ_START_INDEX field. Writing 1 to the WSEQ_START bit commands the sequencer to execute a control sequence, starting at the specified index position. Note that, if the sequencer is already running, the WSEQ_START command is queued and executed when the sequencer becomes available.

Note: The mechanism for queuing multiple sequence requests has limitations when the WSEQ_START bit is used to trigger the write sequencer. If a sequence is initiated using the WSEQ_START bit, no other control sequences should be triggered until the sequence completes. The WSEQ_BUSY bit (described in Table 4-99) provides an indication of the sequencer status and can be used to confirm the sequence has completed.

Multiple control sequences triggered by any other method are queued if necessary, and scheduled in turn.

The write sequencer can be interrupted by writing 1 to the WSEQ_ABORT bit. Note that this command only aborts a sequence that is currently running; if other sequence commands are pending and not yet started, these sequences are not aborted by writing to the WSEQ_ABORT bit.

The write sequencer stores up to 252 register-write commands. These are defined in registers R12288 (0x3000) through R12790 (0x31F6). See Table 4-100 for a description of these registers.

Table 4-93. Write Sequencer Control—Initiating a Sequence

Register Address	Bit	Label	Default	Description
R22 (0x0016)	11	WSEQ_ABORT	0	Writing 1 to this bit aborts the current sequence.
Write_Sequencer_Ctrl_0	10	WSEQ_START	0	Writing 1 to this bit starts the write sequencer at the index location selected by WSEQ_START_INDEX. At the end of the sequence, this bit is reset by the write sequencer.
	9	WSEQ_ENA	0	Write Sequencer Enable 0 = Disabled 1 = Enabled Only applies to sequences triggered using the WSEQ_START bit.
	8:0	WSEQ_START_INDEX[8:0]	0x000	Sequence Start Index. Contains the index location in the sequencer memory of the first command in the selected sequence. Only applies to sequences triggered using the WSEQ_START bit. Valid from 0 to 251 (0x0FB).

Table 4-93. Write Sequencer Control—Initiating a Sequence *(Cont.)*

Register Address	Bit	Label	Default	Description
R66 (0x0042) Spare_Triggers	15	WSEQ_TRG16	0	Write Sequence Trigger 16 Write 1 to trigger
	14	WSEQ_TRG15	0	Write Sequence Trigger 15 Write 1 to trigger
	13	WSEQ_TRG14	0	Write Sequence Trigger 14 Write 1 to trigger
	12	WSEQ_TRG13	0	Write Sequence Trigger 13 Write 1 to trigger
	11	WSEQ_TRG12	0	Write Sequence Trigger 12 Write 1 to trigger
	10	WSEQ_TRG11	0	Write Sequence Trigger 11 Write 1 to trigger
	9	WSEQ_TRG10	0	Write Sequence Trigger 10 Write 1 to trigger
	8	WSEQ_TRG9	0	Write Sequence Trigger 9 Write 1 to trigger
	7	WSEQ_TRG8	0	Write Sequence Trigger 8 Write 1 to trigger
	6	WSEQ_TRG7	0	Write Sequence Trigger 7 Write 1 to trigger
	5	WSEQ_TRG6	0	Write Sequence Trigger 6 Write 1 to trigger
	4	WSEQ_TRG5	0	Write Sequence Trigger 5 Write 1 to trigger
	3	WSEQ_TRG4	0	Write Sequence Trigger 4 Write 1 to trigger
	2	WSEQ_TRG3	0	Write Sequence Trigger 3 Write 1 to trigger
	1	WSEQ_TRG2	0	Write Sequence Trigger 2 Write 1 to trigger
	0	WSEQ_TRG1	0	Write Sequence Trigger 1 Write 1 to trigger
R75 (0x004B) Spare_Sequence_ Select_1	8:0	WSEQ_TRG1_ INDEX[8:0]	0x1FF	Write Sequence trigger 1 start index. Contains the index location in the sequencer memory of the first command in the sequence associated with the WSEQ_TRG1 trigger. Valid from 0 to 251 (0x0FB).
R76 (0x004C) Spare_Sequence_ Select_2	8:0	WSEQ_TRG2_ INDEX[8:0]	0x1FF	Write Sequence trigger 1 start index. Contains the index location in the sequencer memory of the first command in the sequence associated with the WSEQ_TRG2 trigger. Valid from 0 to 251 (0x0FB).
R77 (0x004D) Spare_Sequence_ Select_3	8:0	WSEQ_TRG3_ INDEX[8:0]	0x1FF	Write Sequence trigger 1 start index. Contains the index location in the sequencer memory of the first command in the sequence associated with the WSEQ_TRG3 trigger. Valid from 0 to 251 (0x0FB).
R78 (0x004E) Spare_Sequence_ Select_4	8:0	WSEQ_TRG4_ INDEX[8:0]	0x1FF	Write Sequence trigger 1 start index. Contains the index location in the sequencer memory of the first command in the sequence associated with the WSEQ_TRG4 trigger. Valid from 0 to 251 (0x0FB).
R79 (0x004F) Spare_Sequence_ Select_5	8:0	WSEQ_TRG5_ INDEX[8:0]	0x1FF	Write Sequence trigger 1 start index. Contains the index location in the sequencer memory of the first command in the sequence associated with the WSEQ_TRG5 trigger. Valid from 0 to 251 (0x0FB).
R80 (0x0050) Spare_Sequence_ Select_6	8:0	WSEQ_TRG6_ INDEX[8:0]	0x1FF	Write Sequence trigger 1 start index. Contains the index location in the sequencer memory of the first command in the sequence associated with the WSEQ_TRG6 trigger. Valid from 0 to 251 (0x0FB).
R89 (0x0059) Spare_Sequence_ Select_7	8:0	WSEQ_TRG7_ INDEX[8:0]	0x1FF	Write Sequence trigger 1 start index. Contains the index location in the sequencer memory of the first command in the sequence associated with the WSEQ_TRG7 trigger. Valid from 0 to 251 (0x0FB).
R90 (0x005A) Spare_Sequence_ Select_8	8:0	WSEQ_TRG8_ INDEX[8:0]	0x1FF	Write Sequence trigger 1 start index. Contains the index location in the sequencer memory of the first command in the sequence associated with the WSEQ_TRG8 trigger. Valid from 0 to 251 (0x0FB).

Table 4-93. Write Sequencer Control—Initiating a Sequence *(Cont.)*

Register Address	Bit	Label	Default	Description
R91 (0x005B) Spare_Sequence_ Select_9	8:0	WSEQ_TRG9_ INDEX[8:0]	0x1FF	Write Sequence trigger 1 start index. Contains the index location in the sequencer memory of the first command in the sequence associated with the WSEQ_TRG9 trigger. Valid from 0 to 251 (0x0FB).
R92 (0x005C) Spare_Sequence_ Select_10	8:0	WSEQ_ TRG10_ INDEX[8:0]	0x1FF	Write Sequence trigger 1 start index. Contains the index location in the sequencer memory of the first command in the sequence associated with the WSEQ_TRG10 trigger. Valid from 0 to 251 (0x0FB).
R93 (0x005D) Spare_Sequence_ Select_11	8:0	WSEQ_ TRG11_ INDEX[8:0]	0x1FF	Write Sequence trigger 1 start index. Contains the index location in the sequencer memory of the first command in the sequence associated with the WSEQ_TRG11 trigger. Valid from 0 to 251 (0x0FB).
R94 (0x005E) Spare_Sequence_ Select_12	8:0	WSEQ_ TRG12_ INDEX[8:0]	0x1FF	Write Sequence trigger 1 start index. Contains the index location in the sequencer memory of the first command in the sequence associated with the WSEQ_TRG12 trigger. Valid from 0 to 251 (0x0FB).
R104 (0x0068) Spare_Sequence_ Select_13	8:0	WSEQ_ TRG13_ INDEX[8:0]	0x1FF	Write Sequence trigger 1 start index. Contains the index location in the sequencer memory of the first command in the sequence associated with the WSEQ_TRG13 trigger. Valid from 0 to 251 (0x0FB).
R105 (0x0069) Spare_Sequence_ Select_14	8:0	WSEQ_ TRG14_ INDEX[8:0]	0x1FF	Write Sequence trigger 1 start index. Contains the index location in the sequencer memory of the first command in the sequence associated with the WSEQ_TRG14 trigger. Valid from 0 to 251 (0x0FB).
R106 (0x006A) Spare_Sequence_ Select_15	8:0	WSEQ_ TRG15_ INDEX[8:0]	0x1FF	Write Sequence trigger 1 start index. Contains the index location in the sequencer memory of the first command in the sequence associated with the WSEQ_TRG15 trigger. Valid from 0 to 251 (0x0FB).
R107 (0x006B) Spare_Sequence_ Select_16	8:0	WSEQ_ TRG16_ INDEX[8:0]	0x1FF	Write Sequence trigger 1 start index. Contains the index location in the sequencer memory of the first command in the sequence associated with the WSEQ_TRG16 trigger. Valid from 0 to 251 (0x0FB).

4.15.2 Automatic Sample-Rate Detection Sequences

The CS47L15 supports automatic sample-rate detection on the digital audio interfaces (AIF1–AIF3) when operating in AIF Slave Mode. Automatic sample-rate detection is enabled by setting RATE_EST_ENA—see Table 4-82.

As many as four audio sample rates can be configured for automatic detection; these sample rates are selected using the SAMPLE_RATE_DETECT_*n* fields. If a selected audio sample rate is detected, the control-write sequencer is triggered. The applicable start index location within the sequencer memory is separately configurable for each detected sample rate.

The WSEQ_SAMPLE_RATE_DETECT_A_INDEX field defines the sequencer start index corresponding to the SAMPLE_ RATE_DETECT_A sample rate. Equivalent start index fields are defined for the other sample rates, as described in Table 4-94.

Note that a sequencer start index of 0x1FF causes the respective sequence to be aborted.

The automatic sample-rate detection control sequences are undefined following power-on reset, a hardware reset, or a Sleep Mode transition. The automatic sample-rate detection control sequences must be reconfigured by the host processor following any of these events. Note that all control sequences are maintained in the sequencer memory through software reset.

See Section 4.13 for further details of the automatic sample-rate detection function.

Table 4-94. Write Sequence Control—Automatic Sample-Rate Detection

Register Address	Bit	Label	Default	Description
R97 (0x0061) Sample_Rate_ Sequence_Select_1	8:0	WSEQ_SAMPLE_ RATE_DETECT_ A_INDEX[8:0]	0x1FF	Sample Rate A Write Sequence start index. Contains the index location in the sequencer memory of the first command in the sequence associated with Sample Rate A detection. Valid from 0 to 251 (0x0FB).
R98 (0x0062) Sample_Rate_ Sequence_Select_2	8:0	WSEQ_SAMPLE_ RATE_DETECT_ B_INDEX[8:0]	0x1FF	Sample Rate B Write Sequence start index. Contains the index location in the sequencer memory of the first command in the sequence associated with Sample Rate B detection. Valid from 0 to 251 (0x0FB).
R99 (0x0063) Sample_Rate_ Sequence_Select_3	8:0	WSEQ_SAMPLE_ RATE_DETECT_ C_INDEX[8:0]	0x1FF	Sample Rate C Write Sequence start index. Contains the index location in the sequencer memory of the first command in the sequence associated with Sample Rate C detection. Valid from 0 to 251 (0x0FB).
R100 (0x0064) Sample_Rate_ Sequence_Select_4	8:0	WSEQ_SAMPLE_ RATE_DETECT_ D_INDEX[8:0]	0x1FF	Sample Rate D Write Sequence start index. Contains the index location in the sequencer memory of the first command in the sequence associated with Sample Rate D detection. Valid from 0 to 251 (0x0FB).

4.15.3 DRC Signal-Detect Sequences

The DRC function within the CS47L15 digital core provides a configurable signal-detect function. This allows the signal level at the DRC input to be monitored and used to trigger other events.

The DRC signal-detect functions are enabled and configured using the fields described in Table 4-16 and Table 4-17 for DRC1 and DRC2 respectively.

A control-write sequence can be associated with a rising edge and/or a falling edge of the DRC1 signal-detect output. This is enabled by setting DRC1_WSEQ_SIG_DET_ENA, as described in Table 4-16.

Note that signal detection is supported on DRC1 and DRC2, but the triggering of the control-write sequencer is available on DRC1 only.

When the DRC signal-detect sequence is enabled, the control-write sequencer is triggered whenever the DRC1 signal-detect output transitions (high or low). The applicable start index location within the sequencer memory is separately configurable for each logic condition.

The WSEQ_DRC1_SIG_DET_RISE_SEQ_INDEX field defines the sequencer start index corresponding to a DRC1 signal-detect rising edge event, as described in Table 4-95. The WSEQ_DRC1_SIG_DET_FALL_SEQ_INDEX field defines the sequencer start index corresponding to a DRC1 signal-detect falling edge event.

Note that a sequencer start index of 0x1FF causes the respective sequence to be aborted.

The DRC signal-detect sequences cannot be independently enabled for rising and falling edges. Instead, a start index of 0x1FF can be used to disable the sequence for either edge, if required.

The DRC signal-detect control sequences are undefined following power-on reset, a hardware reset, or a Sleep Mode transition. The DRC signal-detect control sequences must be reconfigured by the host processor following any of these events. Note that all control sequences are maintained in the sequencer memory through software reset.

See Section 4.3.5 for further details of the DRC function.

Table 4-95. Write Sequencer Control—DRC Signal-Detect

Register Address	Bit	Label	Default	Description
R110 (0x006E) Trigger_ Sequence_ Select_32	8:0	WSEQ_DRC1_ SIG_DET_RISE_ INDEX[8:0]	0x1FF	DRC1 Signal-Detect (Rising) Write Sequence start index. Contains the index location in the sequencer memory of the first command in the sequence associated with DRC1 Signal-Detect (Rising) detection. Valid from 0 to 251 (0x0FB).
R111 (0x006F) Trigger_ Sequence_ Select_33	8:0	WSEQ_DRC1_ SIG_DET_FALL_ INDEX[8:0]	0x1FF	DRC1 Signal-Detect (Falling) Write Sequence start index. Contains the index location in the sequencer memory of the first command in the sequence associated with DRC1 Signal-Detect (Falling) detection. Valid from 0 to 251 (0x0FB).

4.15.4 MICDET Clamp Sequences

The CS47L15 supports external accessory detection functions, including the MICDET clamp circuit. The MICDET clamp status can be used to trigger the control-write sequencer. The MICDET clamp is controlled by the JD1 and/or JD2 signals, as described in Table 4-64.

A control-write sequence can be associated with a rising edge and/or a falling edge of the MICDET clamp status. This is configured using the fields described in Table 4-64.

If one of the selected logic conditions is detected, the control-write sequencer is triggered. The applicable start index location within the sequencer memory is separately configurable for the rising and falling edge conditions.

The WSEQ_MICD_CLAMP_RISE_INDEX field defines the sequencer start index corresponding to a MICDET clamp rising edge (clamp active) event, as described in Table 4-96. The WSEQ_MICD_CLAMP_FALL_INDEX field defines the sequencer start index corresponding to a MICDET clamp falling edge event.

Note that a sequencer start index of 0x1FF causes the respective sequence to be aborted.

The MICDET clamp control sequences are undefined following power-on reset, a hardware reset, or a Sleep Mode transition. The MICDET clamp control sequences must be reconfigured by the host processor following any of these events. Note that all control sequences are maintained in the sequencer memory through software reset.

See Section 4.9 for further details of the MICDET clamp status signals.

Table 4-96. Write Sequencer Control—MICDET Clamp

Register Address	Bit	Label	Default	Description
R102 (0x0066) Always_On_Triggers_ Sequence_Select_1	8:0	WSEQ_MICD_ CLAMP_RISE_ INDEX[8:0]	0x1FF	MICDET Clamp (Rising) Write Sequence start index. Contains the index location in the sequencer memory of the first command in the sequence associated with MICDET clamp (Rising) detection. Valid from 0 to 251 (0x0FB).
R103 (0x0067) Always_On_Triggers_ Sequence_Select_2	8:0	WSEQ_MICD_ CLAMP_FALL_ INDEX[8:0]	0x1FF	MICDET Clamp (Falling) Write Sequence start index. Contains the index location in the sequencer memory of the first command in the sequence associated with MICDET clamp (Falling) detection. Valid from 0 to 251 (0x0FB).

4.15.5 Event Logger Sequences

The CS47L15 provides two event log functions, for monitoring and recording internal or external signals. The logged events are held in a FIFO buffer, from which the application software can read details of the detected logic transitions.

The control-write sequencer is automatically triggered whenever the NOT_EMPTY status of the event log buffer is asserted. A different control sequence may be configured for each of the event loggers.

The WSEQ_EVENTLOG*n*_INDEX field defines the sequencer start index corresponding to respective event logger (where *n* is 1 or 2), as described in Table 4-97.

Note that a sequencer start index of 0x1FF causes the respective sequence to be aborted.

The event logger control sequences are undefined following power-on reset, a hardware reset, or a Sleep Mode transition. The event logger control sequences must be reconfigured by the host processor following any of these events. Note that all control sequences are maintained in the sequencer memory through software reset.

See Section 4.5.1 for further details of the event loggers.

Table 4-97. Write Sequencer Control—Event Loggers

Register Address	Bit	Label	Default	Description
R120 (0x0078) Eventlog_ Sequence_ Select_1	8:0	WSEQ_ EVENTLOG1_ INDEX[8:0]	0x1FF	Event Log 1 Write Sequence start index. Contains the index location in the sequencer memory of the first command in the sequence associated with Event Log 1 FIFO Not-Empty detection. Valid from 0 to 251 (0x0FB).
R121 (0x0079) Eventlog_ Sequence_ Select_2	8:0	WSEQ_ EVENTLOG2_ INDEX[8:0]	0x1FF	Event Log 2 Write Sequence start index. Contains the index location in the sequencer memory of the first command in the sequence associated with Event Log 2 FIFO Not-Empty detection. Valid from 0 to 251 (0x0FB).

4.15.6 Boot Sequence

The CS47L15 executes a boot sequence following power-on reset, hardware reset, software reset, or wake-up from Sleep Mode. The boot sequence configures the CS47L15 with factory-set trim (calibration) data. See Section 4.19.5 for further details.

The start index location of the boot sequence is 224 (0x0E0). See Table 4-102 for details of the write sequencer memory allocation.

The boot sequence can be commanded at any time by writing 1 to the WSEQ_BOOT_START bit.

Table 4-98. Write Sequencer Control—Boot Sequence

Register Address	Bit	Label	Default	Description
R24 (0x0018) Write_Sequencer_ Ctrl_2	1	WSEQ_BOOT_ START	0	Writing 1 to this bit starts the write sequencer at the index location configured for the Boot Sequence. The Boot Sequence start index is 224 (0x0E0).

4.15.7 Sequencer Status Indication

The status of the write sequencer can be read using WSEQ_BUSY and WSEQ_CURRENT_INDEX, as described in Table 4-99. When the WSEQ_BUSY bit is asserted, this indicates that the write sequencer is busy.

The index address of the most recent write sequencer command can be read from the WSEQ_CURRENT_INDEX field. This can be used to provide a precise indication of the write sequencer progress.

Table 4-99. Write Sequencer Control—Status Indication

Register Address	Bit	Label	Default	Description
R23 (0x0017) Write_Sequencer_ Ctrl_1	9	WSEQ_BUSY (read only)	0	Sequencer Busy flag (Read Only). 0 = Sequencer idle 1 = Sequencer busy
	8:0	WSEQ_CURRENT_ INDEX[8:0] (read only)	0x000	Sequence Current Index. This indicates the memory location of the most recently accessed command in the write sequencer memory. Coding is the same as WSEQ_START_INDEX.

4.15.8 Programming a Sequence

A control-write sequence comprises a series of write operations to data bits within the control register map. Standard write operations are defined by 5 fields, contained within a single 32-bit register. An extended instruction set is also defined; the associated actions makes use of alternate definitions of the 32-bit registers.

The sequencer instruction fields are replicated 252 times, defining each of the sequencer’s 252 possible index addresses. Many sequences can be stored in the sequencer memory at the same time, with each assigned a unique range of index addresses. The WSEQ_DELAY*n* field is used to identify the end-of-sequence position, as described below.

The general definition of the sequencer instruction fields is described as follows, where *n* denotes the sequencer index address (valid from 0 to 251):

- WSEQ_DATA_WIDTH*n* is a 3-bit field that identifies the width of the data block to be written. Note that the maximum value of this field selects a width of 8 bits; writes to fields that are larger than 8 bits wide must be performed using two separate operations of the write sequencer.
- WSEQ_ADDR*n* is a 12-bit field containing the register address in which the data should be written. The applicable register address is referenced to the base address currently configured for the sequencer—it is calculated as: (base address * 512) + WSEQ_ADDR*n*. Note that the base address is configured using the sequencer's extended instruction set.
- WSEQ_DELAY*n* is a 4-bit field that controls the waiting time between the current step and the next step in the sequence (i.e., the delay occurs after the write in which it was called). The total delay time per step (including execution) is defined below, giving a useful range of execution/delay times from 3.3 µs up to 1 s per step.

  If WSEQ_DELAYn = 0x0 or 0xF, the step execution time is 3.3 µs

  For all other values, the step execution time is 61.44 µs x $((2^{\text{WSEQ\_DELAY}}) - 1)$

  Setting this field to 0xF identifies the step as the last in the sequence
- WSEQ_DATA_START*n* is a 4-bit field that identifies the LSB position within the selected control register to which the data should be written. For example, setting WSEQ_DATA_START*n* = 0100 selects bit [4] as the LSB position of the data to be written.
- WSEQ_DATA*n* is an 8-bit field that contains the data to be written to the selected control register. The WSEQ_DATA_WIDTH*n* field determines how many of these bits are written to the selected control register; the most significant bits (above the number indicated by WSEQ_DATA_WIDTH*n*) are ignored.

The extended instruction set for the write sequencer is accessed by setting WSEQ_MODE*n* (bit [28]) in the respective sequencer definition register. The extended instruction set comprises the following functions:

- If bits [31:24] = 0x11, the register base address is set equal to the value contained in bits [23:0].
- If bits [31:16] = 0x12FF, the sequencer performs an unconditional jump to the index location defined in bits [15:0]. The index location is valid in the range 0 to 251 (0x0FB).
- All other settings within the extended instruction set are reserved.

The control field definitions for Step 0 are described in Table 4-100. The equivalent definitions also apply to Step 1 through Step 251, in the subsequent register address locations.

Table 4-100. Write Sequencer Control—Programming a Sequence

Register Address	Bit	Label	Default	Description
R12288 (0x3000) WSEQ_Sequence_1	31:29	WSEQ_DATA_WIDTH0[2:0]	000	Width of the data block written in this sequence step. 000 = 1 bit; 001 = 2 bits; 010 = 3 bits; 011 = 4 bits; 100 = 5 bits; 101 = 6 bits; 110 = 7 bits; 111 = 8 bits
	28	WSEQ_MODE0	0	Extended Sequencer Instruction select 0 = Basic instruction set 1 = Extended instruction set
	27:16	WSEQ_ADDR0[11:0]	0x000	Control Register Address to be written to in this sequence step. The register address is calculated as: (Base Address * 512) + WSEQ_ADDRn. Base Address is 0x00_0000 by default, and is configured using the sequencer's extended instruction set.
	15:12	WSEQ_DELAY0[3:0]	0000	Time delay after executing this step. 0x0 = 3.3 µs 0x1 to 0xE = 61.44 µs x $((2^{\text{WSEQ\_DELAY}})–1)$ 0xF = End of sequence marker
	11:8	WSEQ_DATA_START0[3:0]	0000	Bit position of the LSB of the data block written in this sequence step. 0000 = Bit 0 … 1111 = Bit 15
	7:0	WSEQ_DATA0[7:0]	0x00	Data to be written in this sequence step. When the data width is less than 8 bits, one or more of the MSBs of WSEQ_DATA*n* are ignored. It is recommended that unused bits be cleared.

4.15.9 Sequencer Memory Definition

The write sequencer memory defines up to 252 write operations; these are indexed as 0 to 251 in the sequencer memory map.

The write sequencer memory reverts to its default contents following power-on reset, a hardware reset, or a Sleep Mode transition. In these cases, the sequence memory contains the boot sequence and the OUT1–OUT4 signal path enable/disable sequences; the remainder of the sequence memory is undefined.

User-defined sequences can be programmed after power-up. The user-defined control sequences must be reconfigured by the host processor following power-on reset, a hardware reset, or a Sleep Mode transition. Note that all control sequences are maintained in the sequencer memory through software reset. See Section 5.2 for a summary of the CS47L15 memory reset conditions.

The default control sequences can be overwritten in the sequencer memory, if required. Note that the headphone and earpiece output path enable bits (HP*nx*_ENA, SPKOUT*x*_ENA) always trigger the write sequencer (at the predetermined start index addresses).

Writing 1 to the WSEQ_LOAD_MEM bit clears the sequencer memory to the power-on reset state.

Table 4-101. Write Sequencer Control—Load Memory Control

Register Address	Bit	Label	Default	Description
R24 (0x0018) Write_Sequencer_Ctrl_2	0	WSEQ_LOAD_MEM	0	Writing 1 to this bit resets the sequencer memory to the power-on reset state.

The sequencer memory is summarized in Table 4-102. User-defined sequences should be assigned space within the allocated portion (user space) of the write sequencer memory.

The start index for the user-defined sequences is configured using the fields described in Table 4-93 through Table 4-97.

Table 4-102. Write Sequencer Memory Allocation

Description	Sequence Index Range
Default Sequences	0 to 155
User Space	156 to 223
Boot Sequence	224 to 251

4.16 Charge Pumps, Regulators, and Voltage Reference

The CS47L15 incorporates a charge-pump circuit to support the ground-referenced headphone/earpiece driver. It also provides a MICBIAS generator (with three switchable outputs), which provide low noise reference voltages suitable for biasing ECM-type microphones or powering digital microphones.

Refer to Section 5.1 for recommended external components.

4.16.1 Charge Pump (CP) Control

The charge pump (CP) circuit is used to generate the positive and negative supply rails for the analog output drivers. The charge pump is enabled automatically by the CS47L15 when required. The charge pump circuit is shown in Fig. 4-81.

Note that decoupling capacitors and flyback capacitors are required for these circuits. Refer to Section 5.1 for recommended external components.

4.16.2 Microphone Bias (MICBIAS) Control

A single MICBIAS generator is incorporated, which provides a low-noise reference suitable for biasing ECM-type microphones or powering digital microphones. The MICBIAS generator is powered from MICVDD, as shown in Fig. 4-81. Refer to Section 5.1.3 for recommended external components.

Switchable outputs from the MICBIAS generator allows three separate reference/supply outputs to be independently controlled. The MICBIAS regulator is enabled using the MICB1_ENA bit. The MICBIAS output switches are enabled using MICB1A_ENA, MICB1B_ENA, and MICB1C_ENA.

Note that, to enable any of the MICBIAS1*x* outputs, the regulator and the respective output switch must both be enabled.

When a MICBIAS output is disabled, it can be configured to be floating or to be actively discharged. This is configured using the MICB1*x*_DISCH bits (for each of the switched outputs), and the MICB1_DISCH bit (for the MICBIAS regulator). Each discharge path is only effective when the respective output, or regulator, is disabled.

The MICBIAS generator can operate in Regulator Mode or in Bypass Mode. The applicable mode is selected using the MICB1_BYPASS bit.

In Regulator Mode (MICB1_BYPASS = 0), the output voltage is selected using the MICB1_LVL field. In this mode, MICVDD must be at least 200mV greater than the required MICBIAS output voltage. The MICBIAS outputs are powered from the MICVDD pin and use the internal band-gap circuit as a reference.

In Regulator Mode, the MICBIAS regulator is designed to operate without external decoupling capacitors. The regulator can be configured to support a capacitive load if required, using the MICB1_EXT_CAP bit. (This may be appropriate for a DMIC supply.) It is important that the external capacitance is compatible with the MICB1_EXT_CAP setting. The compatible load conditions are detailed in Table 3-11.

In Bypass Mode (MICB1_BYPASS = 1), the outputs (MICBIAS1*x*), when enabled, are connected directly to MICVDD. This enables a low power operating state. Note that the MICB1_EXT_CAP setting is not applicable in Bypass Mode—there are no restrictions on the external MICBIAS capacitance in Bypass Mode.

The MICBIAS generator incorporates a pop-free control circuit to ensure smooth transitions when the MICBIAS outputs are enabled or disabled in Bypass Mode; this feature is enabled using the MICB1_RATE bit.

The MICBIAS generator is shown in Fig. 4-81. The MICBIAS control fields are described in Table 4-103.

The maximum output current for the MICBIAS regulator is noted in Table 3-11. This limit must be observed across all three MICBIAS1*x* outputs, especially if more than one microphone is connected to the regulator simultaneously. Note that the maximum output current differs between Regulator Mode and Bypass Mode.

4.16.3 Voltage-Reference Circuit

The CS47L15 incorporates a voltage-reference circuit, powered by AVDD. This circuit ensures the accuracy of the MICBIAS voltage settings.

4.16.4 Block Diagram and Control Registers

The charge-pump and regulator circuits are shown in Fig. 4-81. Note that decoupling capacitors and flyback capacitors are required for these circuits. Refer to Section 5.1 for recommended external components.

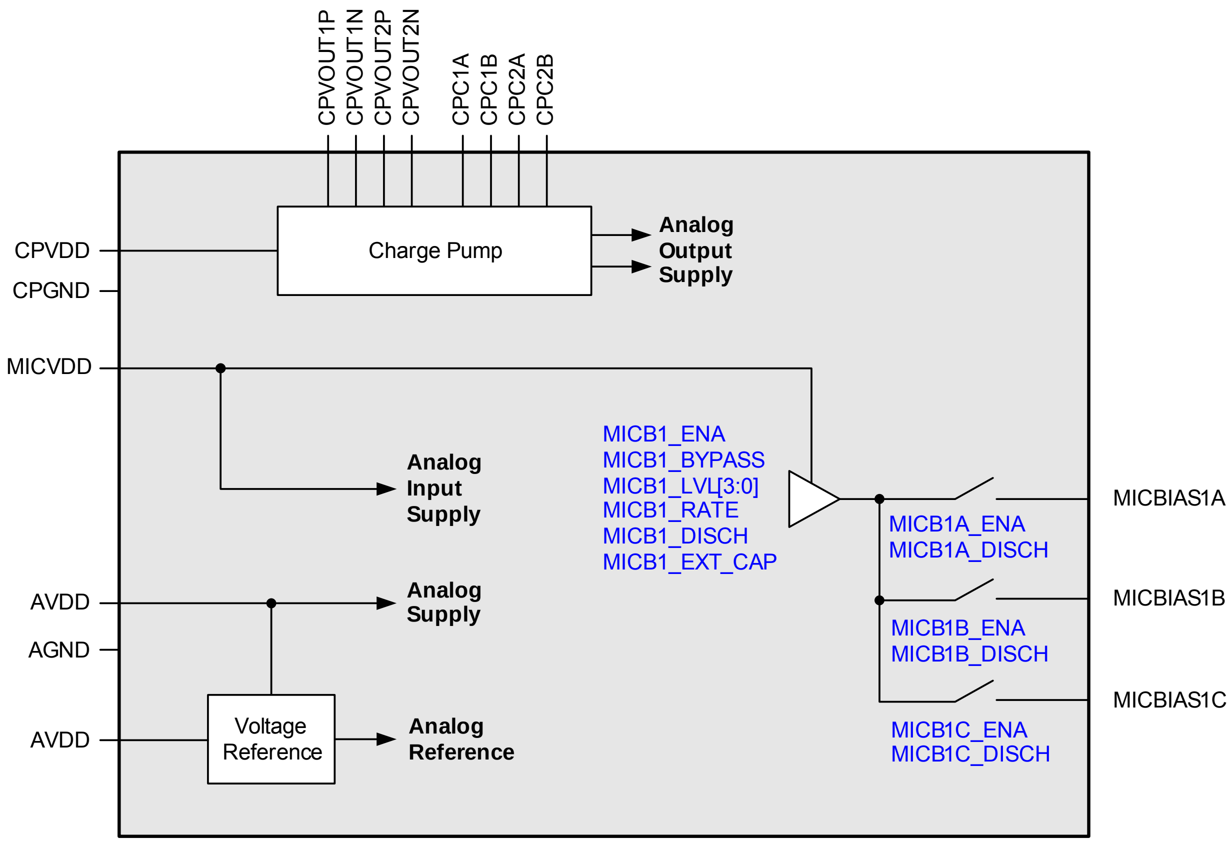


Figure 4-81. Charge Pumps and Regulators

The charge-pump and regulator control registers are described in Table 4-103.

Table 4-103. Charge-Pump and MICBIAS Control Registers

Register Address	Bit	Label	Default	Description
R536 (0x0218) Mic_Bias_Ctrl_1	15	MICB1_EXT_CAP	0	Microphone Bias 1 External Capacitor (when MICB1_BYPASS = 0). Configures the MICBIAS1 regulator according to the specified capacitance connected to the MICBIAS1x outputs. 0 = No external capacitor 1 = External capacitor connected
	8:5	MICB1_LVL[3:0]	0x7	Microphone Bias 1 Voltage Control (when MICB1_BYPASS = 0) 0x0 = 1.5 V … (0.1-V steps) 0xD to 0xF = 2.8 V 0x1 = 1.6 V 0xC = 2.7 V
	3	MICB1_RATE	0	Microphone Bias 1 Rate (Bypass Mode) 0 = Fast start-up/shutdown 1 = Pop-free start-up/shutdown
	2	MICB1_DISCH	1	Microphone Bias 1 Discharge 0 = MICBIAS1 floating when disabled 1 = MICBIAS1 discharged when disabled
	1	MICB1_BYPASS	1	Microphone Bias 1 Mode 0 = Regulator Mode 1 = Bypass Mode
	0	MICB1_ENA	0	Microphone Bias 1 Enable 0 = Disabled 1 = Enabled

Table 4-103. Charge-Pump and MICBIAS Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R540 (0x021C) Mic_Bias_Ctrl_5	9	MICB1C_DISCH	1	Microphone Bias 1C Discharge 0 = MICBIAS1B floating when disabled 1 = MICBIAS1B discharged when disabled
	8	MICB1C_ENA	0	Microphone Bias 1C Enable 0 = Disabled 1 = Enabled
	5	MICB1B_DISCH	1	Microphone Bias 1B Discharge 0 = MICBIAS1B floating when disabled 1 = MICBIAS1B discharged when disabled
	4	MICB1B_ENA	0	Microphone Bias 1B Enable 0 = Disabled 1 = Enabled
	1	MICB1A_DISCH	1	Microphone Bias 1A Discharge 0 = MICBIAS1A floating when disabled 1 = MICBIAS1A discharged when disabled
	0	MICB1A_ENA	0	Microphone Bias 1A Enable 0 = Disabled 1 = Enabled

4.17 JTAG Interface

The JTAG interface provides test and debug access to the CS47L15 DSP core. The interface comprises five connections that are multiplexed with AIF2/AIF3 pins, as noted in Table 4-104.

Table 4-104. JTAG Interface Connections

Pin No	Pin Name	JTAG Function	JTAG Description
J14	AIF3BCLK/GPIO11	TCK	Clock input
G14	AIF2TXDAT/GPIO5	TDI	Data input
G8	AIF3LRCLK/GPIO12	TDO	Data output
G10	AIF3RXDAT/GPIO10	TMS	Mode select input
H13	AIF3TXDAT/GPIO9	TRST	Test access port reset input (active low)

The JTAG interface is selected by setting the DSP_JTAG_MODE bit. If the JTAG interface is selected, the AIF and GPIO functions on the respective pins are disabled.

Note that, under default register conditions, DSP_JTAG_MODE is locked to prevent accidental selection—the user key must be set before writing to DSP_JTAG_MODE. The user key is set by writing 0x5555, followed by 0xAAAA, to the USER_KEY_CTRL field.

It is recommended to clear the user key after writing to DSP_JTAG_MODE. (Note that clearing the user key does not change the value of DSP_JTAG_MODE.) The user key is cleared by writing 0xCCCC, followed by 0x3333, to USER_KEY_CTRL.

For normal operation (test and debug access disabled), the JTAG interface should be disabled or held in reset. If DSP_JTAG_MODE = 0, the JTAG interface is disabled. If DSP_JTAG_MODE = 1, the JTAG interface is held in reset if the TRST pin is Logic 0. An internal pull-down resistor can be used to hold the TRST pin at Logic 0 (i.e., JTAG interface in reset) when not actively driven.

Integrated pull-up and pull-down resistors can be enabled on each of the JTAG pins. This is provided as part of the GPIO functionality, and provides a flexible capability for interfacing with other devices. The pull-up and pull-down resistors can be configured independently using the fields described in Table 4-72.

If the JTAG interface is enabled (TRST deasserted and TCK active) at the time of any reset, a software reset must be scheduled, with the TCK input stopped or TRST asserted (Logic 0), before using the JTAG interface.

It is recommended to always schedule a software reset before starting the JTAG clock or deasserting the JTAG reset. In this event, the JTAG interface should be held in its reset state until the software reset has completed, and the BOOT_DONE_STSx bits have been set. See Section 4.19.3 for further details of the CS47L15 software reset.

The JTAG interface control registers are described in Table 4-105.

Table 4-105. JTAG Interface Control

Register Address	Bit	Label	Default	Description
R140 (0x008C) User_Key_Ctrl	15:0	USER_KEY_ CTRL	0x0000	User Key Control Write 0x5555, then 0xAAAA, to set the key. (Registers unlocked.) Write 0xCCCC, then 0x3333, to clear the key. (Registers locked.)
R334 (0x014E) Clock_Gen_Pad_Ctrl	11	DSP_JTAG_ MODE	0	DSP JTAG Mode Enable 0 = Disabled 1 = Enabled Under default conditions, this bit is locked and cannot be written. To change the value of this bit, the user key must be set before writing to DSP_JTAG_MODE.

4.18 Thermal, Short-Circuit, and Timer-Controlled Protection

The CS47L15 incorporates thermal protection, short-circuit detection, and timer-controlled speaker disable functions; these are described in the following subsections.

4.18.1 Thermal Shutdown

The temperature sensor detects when the device temperature is within normal limits or if the device is approaching a hazardous temperature condition.

The temperature sensor is an input to the interrupt control circuit and can be used to trigger an interrupt event; see Section 4.12. A two-stage indication is provided, via the SPK_OVERHEAT_WARN_EINT*n* and SPK_OVERHEAT_EINT*n* interrupts.

If the upper temperature threshold (SPK_OVERHEAT_EINT*n*) is exceeded, the Class D speaker outputs are automatically disabled in order to protect the device. When the speaker driver shutdown is complete, a further interrupt, SPK_SHUTDOWN_EINT*n*, is asserted.

4.18.2 Short Circuit Protection

The short-circuit detection function for the Class D speaker output is triggered when the respective output driver is enabled (see Table 4-54). If a short circuit is detected at this time, the enable does not succeed, and the output driver is not enabled.

The Class D speaker short-circuit detection provides inputs to the interrupt control circuit and can be used to trigger an interrupt event; see Section 4.12. If the Class D speaker short-circuit condition is detected, the respective driver is automatically disabled in order to protect the device. When the speaker driver shutdown is complete, a further interrupt, SPK_SHUTDOWN_EINT*n*, is asserted.

To enable the Class D speaker outputs following a short-circuit detection, the host processor must disable and reenable the output drivers. Note that the short-circuit status bits are always cleared when the drivers are disabled.

The short-circuit detection function for the headphone and earpiece output paths operates continuously if the respective output driver is enabled. If a short circuit is detected on the headphone or earpiece output, current limiting is applied to protect the respective output driver. Note that the driver continues to operate, but the output is current-limited.

The headphone and earpiece short-circuit detection function provides input to the interrupt control circuit and can be used to trigger an interrupt event when a short-circuit condition is detected; see Section 4.12.

4.18.3 Timer-Controlled Speaker Shutdown

The general-purpose timers (see Section 4.5.2) can also be used to trigger a shutdown of the Class D speaker driver. This is configured using the SPK_SHUTDOWN_TIMER_SEL field, as described in Table 4-106.

If one of the general-purpose timers is selected for the speaker shutdown function, and the respective timer reaches its final count value, the Class D speaker driver is automatically disabled. When the driver shutdown is complete, an interrupt event (SPK_SHUTDOWN_EINT*n*) is signaled.

To enable the Class D speaker output following a timeout condition, the host processor must disable and reenable the output driver using the control bits described in Table 4-54.

Table 4-106. Speaker Shutdown—Timer Control

Register Address	Bit	Label	Default	Description
R620 (0x026D) SPK_Watchdog_1	3:0	SPK_SHUTDOWN_ TIMER_SEL[3:0]	0x0	Speaker Shutdown Timer select. 0x0 = Disabled 0x1 = Timer 1 0x2 = Timer 2 All other codes are reserved

4.18.4 GPIO Output

The thermal status, Class D speaker short-circuit protection, and Class D speaker shutdown flags can be output directly on a GPIO pin as an external indication of the associated events. See Section 4.11 to configure a GPIO pin for this function.

4.19 Power-Up, Resets, and Device ID

The CS47L15 incorporates a power-on reset function to control the device start-up procedure. Hardware- and software-controlled reset functions are also supported. The resets and the sleep/wake-up state transitions provide similar functionality, and are described in the following subsections.

The CS47L15 device ID can be read from the Software_Reset (R0) control register, as described in Section 4.19.8.

4.19.1 Power-On Reset (POR)

The CS47L15 remains in the reset state until AVDD, DBVDD, and DCVDD are above their respective reset thresholds. Note that specified device performance is not assured outside the voltage ranges defined in Table 3-3.

The POR sequence is scheduled on initial power-up, when AVDD, DBVDD, and DCVDD are above their respective reset thresholds. After the initial power-up, the POR is also scheduled following an interrupt to the DBVDD or AVDD supplies.

4.19.2 Hardware Reset

The CS47L15 provides a hardware reset function, which is executed whenever the $\overline{\text{RESET}}$ input is asserted (Logic 0). The $\overline{\text{RESET}}$ input is active low and is referenced to the DBVDD power domain. A hardware reset causes all of the CS47L15 control registers to be reset to their default states.

An internal pull-up resistor is enabled by default on the $\overline{\text{RESET}}$ pin; this can be configured using the RESET_PU bit. A pull-down resistor is also available, as described in Table 4-107. When the pull-up and pull-down resistors are both enabled, the CS47L15 provides a bus keeper function on the $\overline{\text{RESET}}$ pin. The bus keeper function holds the input logic level unchanged whenever the external circuit removes the drive (e.g., if the signal is tristated).

Table 4-107. Reset Pull-Up/Pull-Down Configuration

Register Address	Bit	Label	Default	Description
R6864 (0x1AD0) AOD_Pad_Ctrl	1	RESET_PU	1	RESET Pull-up enable 0 = Disabled 1 = Enabled **Note:** If RESET_PD and RESET_PU are both set, a bus keeper function is enabled on the RESET pin.
	0	RESET_PD	0	RESET Pull-down enable 0 = Disabled 1 = Enabled **Note:** If RESET_PD and RESET_PU are both set, a bus keeper function is enabled on the RESET pin.

4.19.3 Software Reset

A software reset is executed by writing any value to register R0. A software reset causes most of the CS47L15 control registers to be reset to their default states. Note that the control-write sequencer memory is retained during software reset.

4.19.4 Wake-Up

The CS47L15 is in Sleep Mode when AVDD and DBVDD are present, and DCVDD is below its reset threshold. (Note that specific control requirements are also applicable for entering Sleep Mode, as described in Section 4.10.)

In Sleep Mode, most of the digital core (and control registers) are held in reset; selected functions and control registers are maintained via an always-on internal supply domain. See Section 4.10 for details of the always-on functions.

A wake-up transition (from Sleep Mode) is similar to a software reset, but selected functions and control registers are maintained via an always-on internal supply domain—the always-on registers are not reset during wake-up. See Section 4.10 for details of the always-on functions.

4.19.5 Boot Sequence

Following power-on reset, hardware reset, software reset, or wake-up from Sleep Mode, a boot sequence is executed. The BOOT_DONE_STS*x* bits (see Table 4-109) are asserted on completion of the boot sequence. Control-register writes should not be attempted until BOOT_DONE_STS*x* has been asserted. Note that the BOOT_DONE_STS1 and BOOT_DONE_STS2 bits provide the same information.

The BOOT_DONE_STS*x* status is an input to the interrupt control circuit and can be used to trigger an interrupt event on completion of the boot sequence; see Section 4.12. Under default register conditions, a falling edge on the IRQ pin indicates completion of the boot sequence.

For details of the boot sequence, see Section 4.15.

An additional sequence of initialization settings must be written after the boot sequence has completed—this is specified in Table 4-108. The host system should ensure the CS47L15 is ready (i.e., BOOT_DONE_STS*x* is set) before scheduling these register operations.

Note: If the master-boot function is selected (see Section 4.14), the initialization sequence must be incorporated within the device configuration file on the external EEPROM.

Table 4-108. CS47L15 Initialization Sequence

Control Register Writes
• Write 0x5555 to address 0x008C • Write 0xAAAA to address 0x008C • Write 0x0080 to address 0x0314 • Write 0x6023 to address 0x04A8 • Write 0x6023 to address 0x04A9 • Write 0x0008 to address 0x04D4 • Write 0x0F00 to address 0x04CF • Write 0xCCCC to address 0x008C • Write 0x3333 to address 0x008C

If the master-boot function is selected, the $\overline{\text{IRQ}}$ pin is asserted (Logic 0) after the normal boot sequence has completed. The subsequent behavior depends upon what event caused the boot sequence to occur:

- If the boot sequence was scheduled due to a power-on reset or hardware reset, the CS47L15 downloads data from the external EEPROM, and is configured according to the applicable user program data. Clearing the interrupt, and the subsequent behavior of the $\overline{\text{IRQ}}$ output, is dependent on the user program data.
- If the boot sequence was scheduled due to a software reset or wake-up from Sleep Mode, the CS47L15 does not download data from the external EEPROM. Caution is advised if scheduling a software reset or wake-up transition in master-boot applications.

The BOOT_DONE_STS*x* bits are defined in Table 4-109.

Table 4-109. Device Boot-Up Status

Register Address	Bit	Label	Default	Description
R6272 (0x1880) IRQ1_Raw_ Status_1	7	BOOT_DONE_ STS1	0	Boot Status 0 = Busy (boot sequence in progress) 1 = Idle (boot sequence completed) Control register writes should not be attempted until Boot Sequence has completed.
R6528 (0x1980) IRQ2_Raw_ Status_1	7	BOOT_DONE_ STS2	0	Boot Status 0 = Busy (boot sequence in progress) 1 = Idle (boot sequence completed) Control register writes should not be attempted until Boot Sequence has completed.

4.19.6 Digital I/O Status in Reset

Table 1-1 describes the default status of the CS47L15 digital I/O pins on completion of power-on reset and before any register writes. The same default conditions are also applicable on completion of a hardware reset or software reset.

The default conditions are also applicable following a wake-up transition, except for the $\overline{\text{IRQ}}$ and $\overline{\text{RESET}}$ pins—these are always-on pins whose configuration is unchanged in Sleep Mode and during a wake-up transition.

Note that the default conditions described in Table 1-1 are not valid if modified by the boot sequence or by a wake-up control sequence. See Section 4.15 for details of these functions.

4.19.7 Write Sequencer and DSP Firmware Memory Control in Reset and Wake-Up

The control-write sequencer memory reverts to its default state following power-on reset, a hardware reset, or a Sleep Mode transition. The control sequences (including any user-defined sequences) are maintained in the sequencer memory through software reset.

The DSP firmware memory contents are cleared following power-on reset, a hardware reset, or a Sleep Mode transition. The firmware memory contents are not affected by software reset, provided DCVDD is held above its reset threshold.

See Section 5.2 for a summary of the CS47L15 memory reset conditions.

4.19.8 Device ID

The device ID can be read from Register R0. The hardware revision can be read from Register R1.

The software revision can be read from Register R2. The software revision code is incremented if software driver compatibility or software feature support is changed.

Table 4-110. Device Reset and ID

Register Address	Bit	Label	Default	Description
R0 (0x0000) Software_Reset	15:0	SW_RST_DEV_ ID[15:0]	0x6370	Writing to this register resets all registers to their default state. Reading from this register indicates Device ID 0x6370.
R1 (0x0001) Hardware_ Revision	7:0	HW_ REVISION[7:0]	—	Hardware Device revision. This field is incremented for every new revision of the device.
R2 (0x0002) Software_Revision	7:0	SW_ REVISION[7:0]	—	Software Device revision. This field is incremented if software driver compatibility or software feature support is changed.

5 Applications

5.1 Recommended External Components

This section provides information on the recommended external components for use with the CS47L15.

5.1.1 Analog Input Paths

The CS47L15 supports up to five analog audio input connections. Four analog inputs are multiplexed on the IN1 signal path; a mono analog input is also supported on the IN2 signal path.

The IN1*x*P and IN1*x*N pins are biased to the internal DC reference, VREF. (Note that this reference voltage is present on the VREFC pin.) A DC-blocking capacitor is required when connecting to these input pins. The choice of capacitor is determined by the filter that is formed between that capacitor and the impedance of the input pin. The circuit is shown in Fig. 5-1.

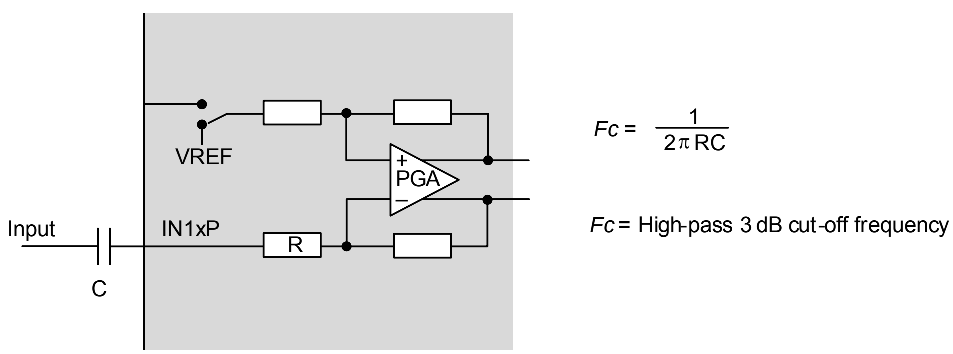


Figure 5-1. Audio Input Path DC-Blocking Capacitor (IN1x pins only)

In accordance with the CS47L15 input pin resistance (see Table 3-5), a 1-µF capacitance gives good results in most cases, with a 3-dB cut-off frequency around 13 Hz.

Ceramic capacitors are suitable, but take care to ensure the desired capacitance is maintained at the AVDD operating voltage. Also, ceramic capacitors may show microphonic effects, where vibrations and mechanical conditions give rise to electrical signals. This is particularly problematic for microphone input paths where a large signal gain is required.

A single capacitor is required for a single-ended line or microphone input connection. For a differential input connection, a DC-blocking capacitor is required on both input pins. The external connections for single-ended and differential microphones, incorporating the CS47L15 microphone bias circuit, are shown in Fig. 5-2.

The IN2P and IN2N pins support ground-referenced input signals only. Input capacitors must not be used on the IN2*x* pins.

5.1.2 Digital Input Paths

The CS47L15 supports up to four channels of digital input. Two channels of audio data can be multiplexed on the DMICDAT pin and a further two channels can be multiplexed on the SPKRXDAT pin.

The external connections for digital microphones, incorporating the CS47L15 microphone bias circuit, are shown in Fig. 5-4. The data on the DMICDAT input pin is clocked using the DMICCLK signal. Ceramic decoupling capacitors for the digital microphones may be required—refer to the specific recommendations for the application microphones.

If two microphones are connected to DMICDAT, the microphones must be configured to ensure that the left mic transmits a data bit when DMICCLK is high, and the right mic transmits a data bit when DMICCLK is low. The CS47L15 samples the DMIC data at the end of each DMICCLK phase. Each microphone must tristate its data output when the other microphone is transmitting. An integrated pull-down resistor can be enabled on the DMICDAT pin if required.

The voltage reference for the DMICDAT/DMICCLK interface is selectable. It is important that the selected reference for the CS47L15 interface is compatible with the applicable configuration of the external microphone.

Digital audio input is also supported on the SPKRXDAT pin; this digital input path forms the receive (RX) side of the digital speaker (PDM) output interface. Two channels of audio data are multiplexed on the SPKRXDAT pin; the data on the SPKRXDAT input pin is clocked using the SPKCLK signal. The voltage reference for the SPKCLK, SPKRXDAT, and SPKTXDAT pins is DBVDD.

If two digital microphones are connected to the SPKRXDAT pin, each microphone must tristate its data output when the other microphone is transmitting. Ceramic decoupling capacitors for the digital microphones may be required.

5.1.3 Microphone Bias Circuit

The CS47L15 is designed to interface easily with analog or digital microphones.

Each microphone requires a bias current (electret condenser microphones) or voltage supply (silicon microphones); these can be provided by the MICBIAS regulator on the CS47L15. A single MICBIAS generator is available, with switchable outputs allowing three separate reference/supply outputs to be independently controlled.

Note that the MICVDD pin can also be used (instead of MICBIAS1*x*) as a reference or power supply for external microphones. The MICBIAS outputs are recommended, as these offer better noise performance and independent enable/disable control.

Analog microphones may be connected in single-ended or differential configurations, as shown in Fig. 5-2. The differential configuration provides better performance due to its rejection of common-mode noise; the single-ended method provides a reduction in external component count.

A bias resistor is required when using an ECM. The bias resistor should be chosen according to the minimum operating impedance of the microphone and MICBIAS voltage so that the maximum bias current of the CS47L15 is not exceeded.

A 2.2-kΩ bias resistor is recommended; this provides compatibility with a wide range of microphone components.

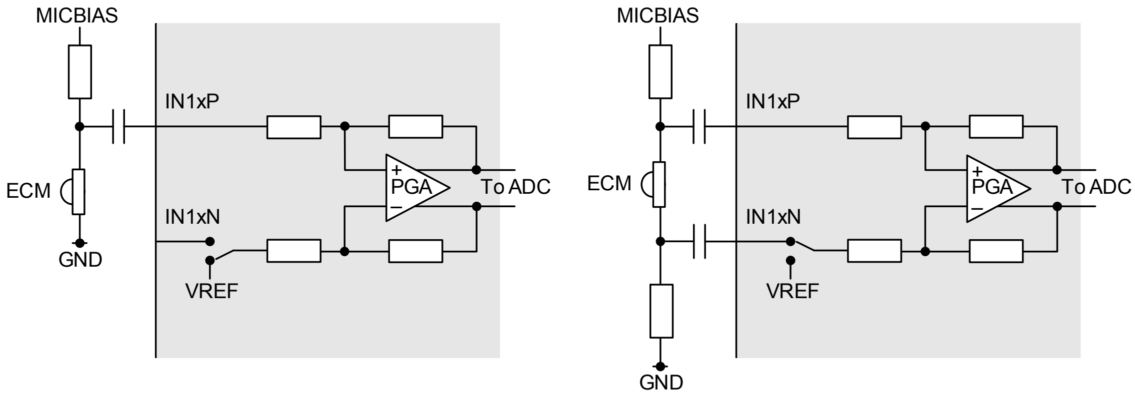


Figure 5-2. Single-Ended and Differential ECM Microphone Connections

Analog MEMS microphones can be connected to the CS47L15 as shown in Fig. 5-3. In this configuration, the MICBIAS generators provide a low-noise supply for the microphones; a bias resistor is not required.

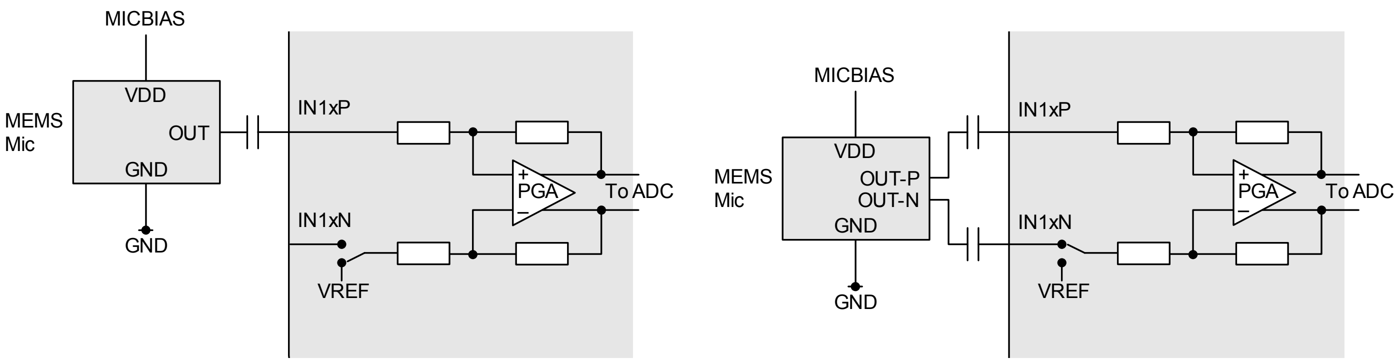


Figure 5-3. Single-Ended and Differential Analog MEMS Microphone Connections

DMIC connection to the CS47L15 is shown in Fig. 5-4. Note that ceramic decoupling capacitors at the DMIC power supply pins may be required—refer to the specific recommendations for the application microphones.

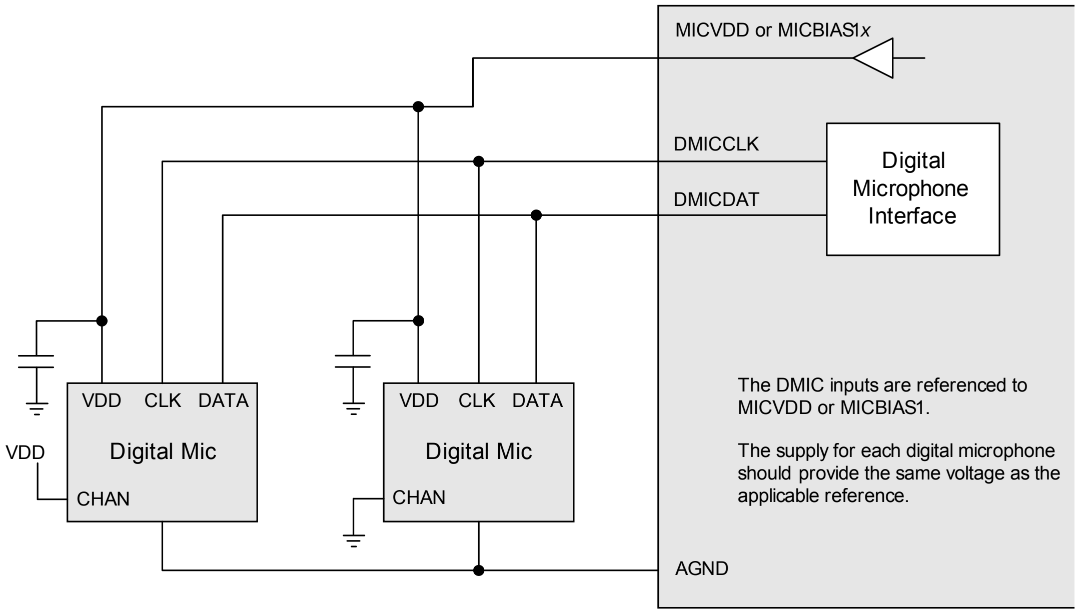


Figure 5-4. DMIC Connection

The MICBIAS generator can operate in Regulator Mode or in Bypass Mode. See Section 4.16 for details of the MICBIAS generator.

In Regulator Mode, the MICBIAS regulator is designed to operate without external decoupling capacitors. The regulator can be configured to support a capacitive load if required (e.g., for DMIC supply decoupling). The compatible load conditions are detailed in Table 3-11.

If the capacitive load on the MICBIAS1*x* outputs exceeds the specified conditions for Regulator Mode (e.g., due to a decoupling capacitor or long PCB trace), the MICBIAS generator must be configured in Bypass Mode.

The maximum output current for the MICBIAS regulator is noted in Table 3-11. This limit must be observed in respect of all enabled MICBIAS1*x* outputs, especially if more than one microphone is connected. Note that the maximum output current differs between Regulator Mode and Bypass Mode. The MICBIAS output voltage can be adjusted using register control in Regulator Mode.

5.1.4 Headphone/Earpiece Driver Output Path

The CS47L15 provides a stereo headphone output driver and a mono (differential) earpiece output driver. Note that the respective output signal path is common to both drivers; only one of these drivers may be enabled at any time. These outputs are all ground referenced, allowing direct connection to the external loads. There is no requirement for DC-blocking capacitors.

Under default register conditions, the headphone/earpiece output path is configured for stereo output on HPOUTL and HPOUTR; this is ideal for stereo headphone loads. In Mono Mode, with the earpiece output driver selected, the output path is configured for mono (differential) output on EPOUTP and EPOUTN; this is suitable for an earpiece or hearing coil load.

The headphone output (HPOUTL, HPOUTR) incorporates a common-mode, or ground-loop, feedback path that provides rejection of system-related ground noise. The feedback pin must be connected to ground for normal operation of the headphone output.

The ground feedback path for HPOUTL and HPOUTR is selected using HP1_GND_SEL. Note that the selected pin should be connected to GND as close as possible to the respective headphone jack ground pin, as shown in Fig. 5-5.

Note that the earpiece output (EPOUTP, EPOUTN) does not support common-mode feedback.

It is recommended to ensure that the electrical characteristics of the PCB traces for each output pair are closely matched. This is particularly important to matching the two traces of a differential (BTL) output.

Typical headphone and earpiece connections are shown in Fig. 5-5.

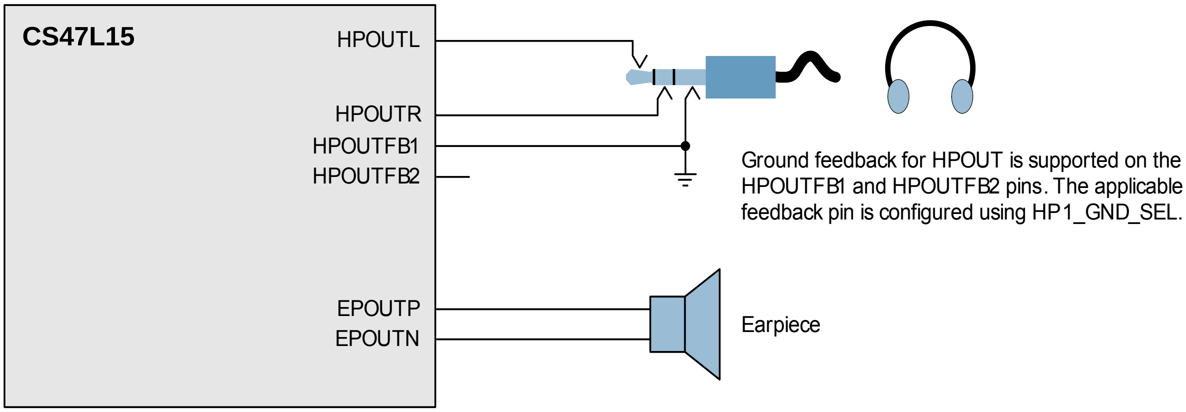


Figure 5-5. Headphone and Earpiece Connection

It is common for ESD diodes to be wired to pins that link to external connectors. This provides protection from potentially harmful ESD effects. In a typical application, ESD diodes are recommended if the headphone path is used for external headphone or line output.

The HPOUT outputs are ground-referenced, and the respective voltages may swing between +1.8V and –1.8V. The ESD diode configuration must be carefully chosen.

The recommended ESD diode configuration for these ground-referenced outputs is shown in Fig. 5-6. The back-to-back arrangement prevents clipping and distortion of the output signal.

Note that similar care is required when connecting the CS47L15 outputs to external circuits that provide input path ESD protection; the configuration on those input circuits must be correctly designed to accommodate ground-referenced signals.

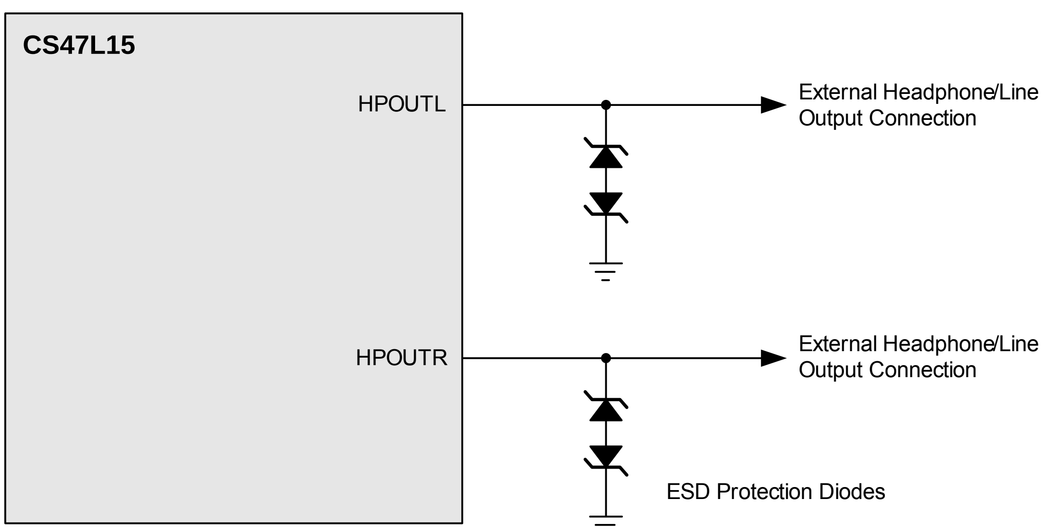


Figure 5-6. ESD Diode Configuration for External Output Connections

5.1.5 Speaker-Driver Output Path

The CS47L15 incorporates a Class D speaker driver, offering high amplifier efficiency at large signal levels. As the Class D output is a pulse-width modulated signal, the choice of speakers and tracking of signals is critical for ensuring good performance and reducing EMI.

The efficiency of the speaker driver is affected by the series resistance between the CS47L15 and the speaker (e.g., PCB track loss and inductor ESR) as shown in Fig. 5-7. This resistance should be as low as possible to maximize efficiency.

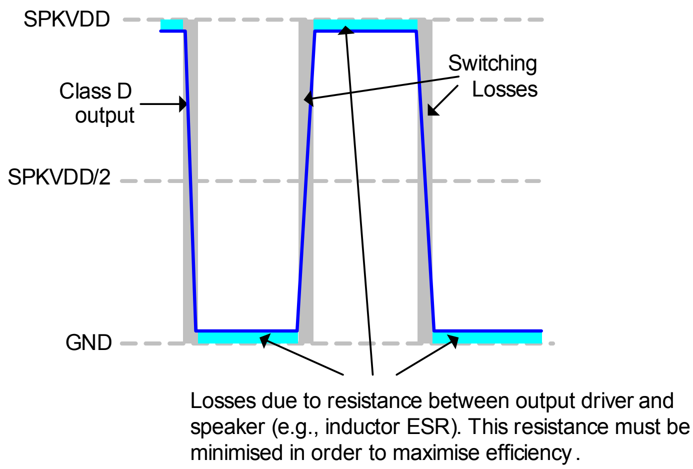
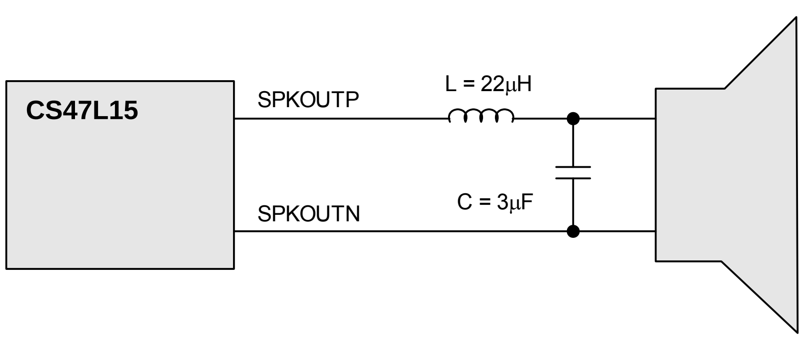


Figure 5-7. Speaker Connection Losses

The Class D output requires external filtering to recreate the audio signal. This may be implemented using a 2nd order LC or 1st order RC filter, or else may be achieved by using a loudspeaker whose internal inductance provides the required filter response. An LC or RC filter should be used if the loudspeaker characteristics are unknown or unsuitable, or if the length of the loudspeaker connection is likely to lead to EMI problems.

In applications where it is necessary to provide Class D filter components, a second-order LC filter is the recommended solution as it provides more attenuation at higher frequencies and minimizes power dissipated in the filter when compared to a first order RC filter (lower ESR). This maximizes both rejection of unwanted switching frequencies and overall speaker efficiency. A suitable implementation is shown in Fig. 5-8.



$$F_C = \frac{1}{2\pi\sqrt{LC}}$$

F_C = Low-pass 3 dB cut-off frequency

Figure 5-8. Class D Output Filter Components

A simple equivalent circuit of a loudspeaker consists of a series-connected resistor and inductor, as shown in Fig. 5-9. This circuit provides a low-pass filter for the speaker output. If the loudspeaker characteristics are suitable, the loudspeaker itself can be used in place of the filter components described earlier. This is known as filterless operation.

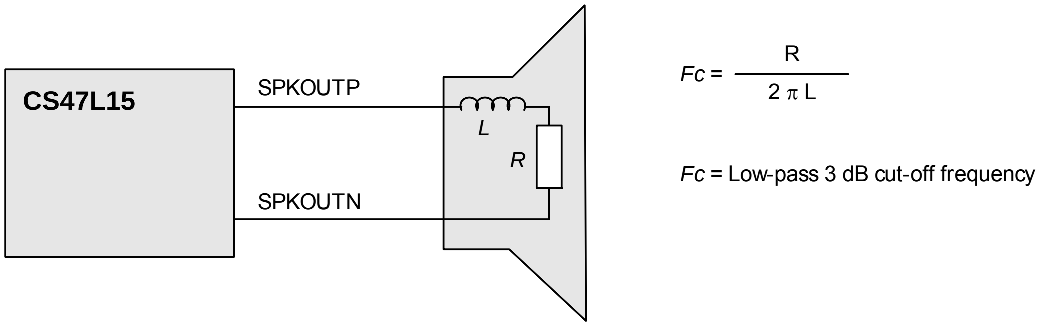


Figure 5-9. Speaker Equivalent Circuit for Filterless Operation

For filterless Class D operation, it is important to ensure that a speaker with suitable inductance is chosen. For example, if we know the speaker impedance is 8 Ω and the desired cut-off frequency is 20 kHz, the optimum speaker inductance may be calculated as shown in Eq. 5-1.

$$L = \frac{R}{2\pi Fc} = \frac{8\Omega}{2\pi \times 20kHz} = 64\mu H$$

Equation 5-1. Speaker Inductance Calculation

An 8-Ω loudspeaker typically has an inductance in the range 20–100 µH; however, it should be noted that a loudspeaker inductance is not constant across the relevant frequencies for Class D operation (up to and beyond the Class D switching frequency). Care should be taken to ensure that the cut-off frequency of the loudspeaker’s filtering is low enough to suppress the high-frequency energy of the Class D switching and, in so doing, to prevent speaker damage. The Class D outputs of the CS47L15 operate at much higher frequencies than is recommended for most speakers, and it must be ensured that the cut-off frequency is low enough to protect the speaker.

The Class D speaker outputs are designed to support monitoring of external loudspeakers, giving real-time feedback for algorithms such as Cirrus Logic’s speaker protection software. This enables maximum audio output to be achieved, while ensuring the loudspeakers are also fully protected from damage.

The external speaker connections, incorporating the output current monitoring requirements, are shown in Fig. 5-10. Note that, if output current monitoring is not required on one or more speaker channels, the respective ground connections should be tied directly to ground on the PCB.

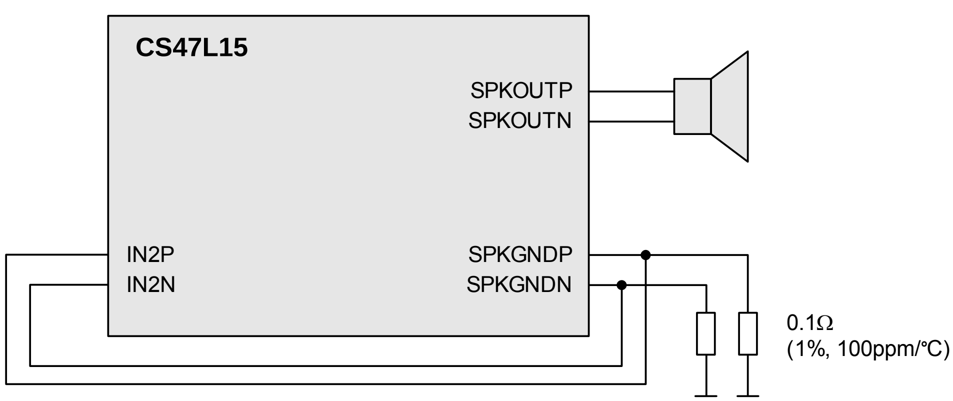


Figure 5-10. Speaker Output Current Monitoring Connections (Speaker Protection)

5.1.6 Power Supply/Reference Decoupling

Electrical coupling exists particularly in digital logic systems where switching in one subsystem causes fluctuations on the power supply. This effect occurs because the inductance of the power supply acts in opposition to the changes in current flow that are caused by the logic switching. The resultant variations (spikes) in the power-supply voltage can cause malfunctions and unintentional behavior in other components. A decoupling (bypass) capacitor can be used as an energy storage component that provides power to the decoupled circuit for the duration of these power-supply variations, protecting it from malfunctions that could otherwise arise.

Coupling also occurs in a lower frequency form when ripple is present on the power supply rail caused by changes in the load current or by limitations of the power-supply regulation method. In audio components such as the CS47L15, these variations can alter the performance of the signal path, leading to degradation in signal quality. A decoupling capacitor can be used to filter these effects by presenting the ripple voltage with a low-impedance path that does not affect the circuit to be decoupled.

These coupling effects are addressed by placing a capacitor between the supply rail and the corresponding ground reference. In the case of systems comprising multiple power supply rails, decoupling should be provided on each rail.

PCB layout is also a contributory factor for coupling effects. If multiple power supply rails are connected to a single supply source, it is recommended to provide separate PCB tracks connecting each rail to the supply. See Section 5.5 for PCB-layout recommendations.

The recommended power-supply decoupling capacitors for CS47L15 are detailed in Table 5-1.

Table 5-1. Power Supply Decoupling Capacitors

Power Supply	Decoupling Capacitor
AVDD	1.0 µF ceramic
CPVDD	4.7 µF ceramic
DBVDD	0.1 µF ceramic [1]
DCVDD	2.2 µF ceramic
MICVDD	1.0 µF ceramic
SPKVDD	4.7 µF ceramic
VREFC	2.2 µF ceramic

1. Total capacitance of 4.7 µF is required for the DBVDD domain. This can be provided by dedicated DBVDD decoupling or by other capacitors on the same power rail.

All decoupling capacitors should be placed as close as possible to the CS47L15 device. The connection between AGND, the AVDD decoupling capacitor, and the main system ground should be made at a single point as close as possible to the AGND balls of the CS47L15.

Due to the wide tolerance of many types of ceramic capacitors, care must be taken to ensure that the selected components provide the required capacitance across the required temperature and voltage ranges in the intended application. Ceramic capacitors with X5R dielectric are recommended.

5.1.7 Charge-Pump Components

The CS47L15 incorporates a charge-pump circuit that generates the CPVOUT*nx* supply rails for the headphone/earpiece drivers. Decoupling capacitors are required on each of the charge-pump outputs. Two fly-back capacitors are also required.

The recommended charge-pump capacitors for CS47L15 are detailed in Table 5-2.

Table 5-2. Charge-Pump External Capacitors

Description	Capacitor
CPVOUT1P decoupling	2.2 µF ceramic
CPVOUT1N decoupling	2.2 µF ceramic
CP fly-back 1 (connect between CPC1A and CPC1B)	1.0 µF ceramic
CPVOUT2P decoupling	4.7 µF ceramic
CPVOUT2N decoupling	4.7 µF ceramic
CP fly-back 2 (connect between CPC2A and CPC2B)	2.2 µF ceramic

Ceramic capacitors are recommended for these charge-pump requirements. Care must be taken to ensure that the selected components provide the required capacitance across the required temperature and voltage ranges in the intended application. Ceramic capacitors with X5R dielectric are recommended.

The positioning of the charge-pump capacitors is important. These capacitors (particularly the fly-back capacitors) must be placed as close as possible to the CS47L15.

5.1.8 External Accessory Detection Components

The external accessory detection circuit measures jack insertion using the JACKDET1 and JACKDET2 pins. The insertion switch status is detected using an internal pull-up resistor circuit on the respective pin. Note that the logic thresholds associated with the two JACKDET differ from each other, as described in Table 3-11—this provides support for different jack switch configurations.

Microphone detection and key-button press detection is supported using the MICDETn pins. The applicable pin should be connected to one of the MICBIAS1*x* outputs, via a 2.2-kΩ bias resistor, as described in Section 5.1.3. Note that, when using the external accessory detection function, the MICBIAS1*x* resistor must be 2.2 kΩ ±2%.

A recommended circuit configuration, including headphone output on HPOUT and microphone connections, is shown in Fig. 5-11. See Section 5.1.1 for details of the DC-blocking microphone input capacitor selection.

The recommended external components and connections for microphone/push-button detection are shown in Fig. 5-11.

Note that, when using the microphone detect circuit, it is recommended to use the IN1BLP or IN1BRP analog microphone input paths to ensure best immunity to electrical transients arising from the external accessory.

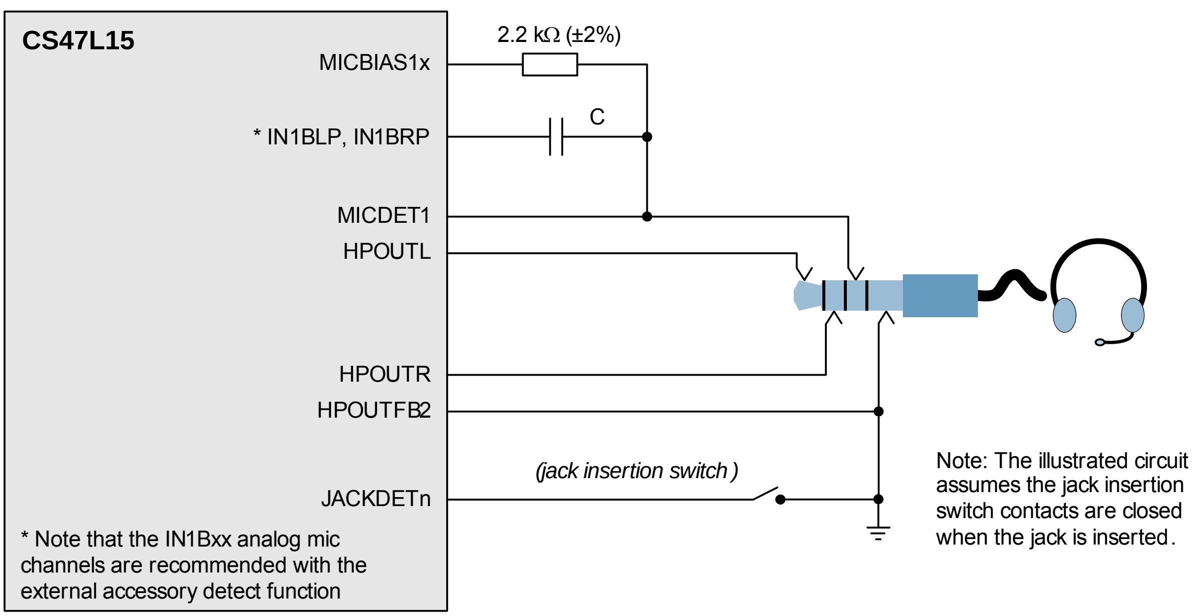


Figure 5-11. External Accessory Detection

The accessory detection circuit measures the impedance of an external load connected to one of the MICDET pins.

The microphone-detection circuit uses MICVDD, MICBIAS1A, MICBIAS1B, or MICBIAS1C as a reference. The applicable source is configured using MICD1_BIAS_SRC.

The CS47L15 can detect the presence of a typical microphone and up to six push buttons, using the components shown in Fig. 5-12. When the microphone detection circuit is enabled, each of the push buttons shown causes a different bit in the MICD1_LVL field to be set.

The choice of external resistor values must take into account the impedance of the microphone—the detected impedance corresponds to the combined parallel resistance of the microphone and any asserted push button. The components shown in Fig. 5-12 are examples only, assuming default impedance measurement ranges and a microphone impedance of 1 kΩ or higher.

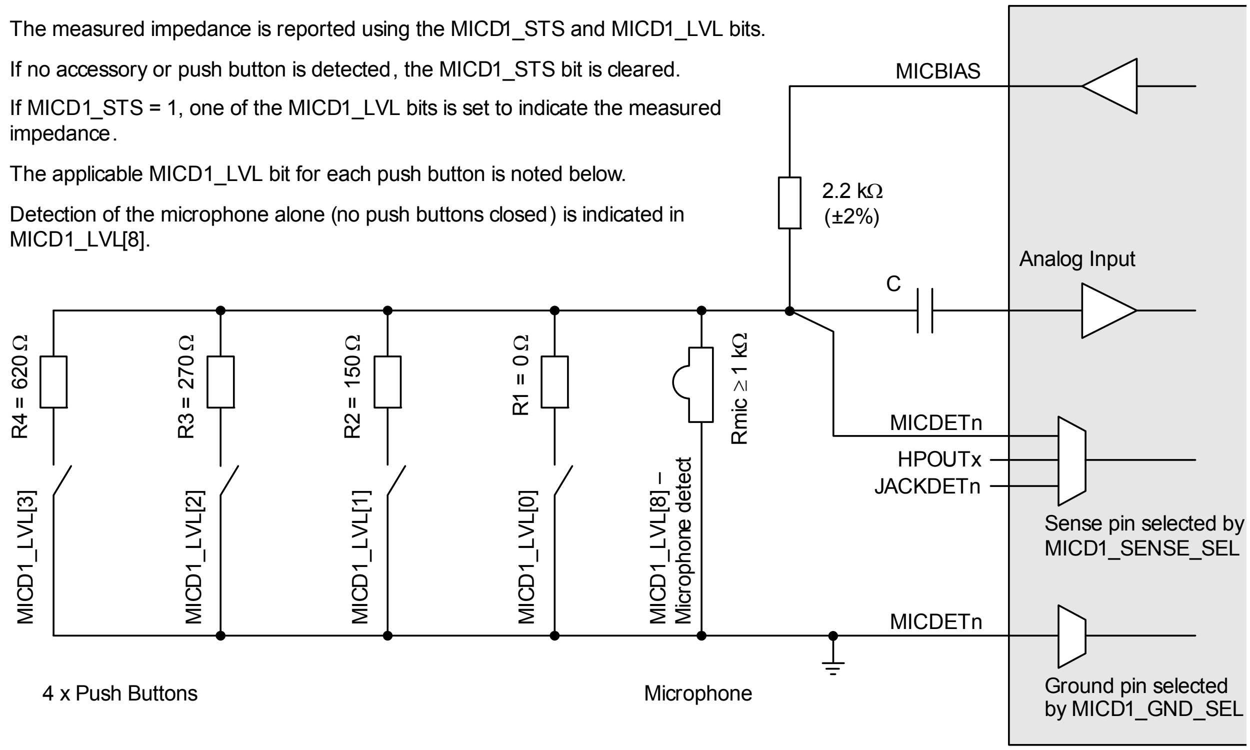


Figure 5-12. External Accessory Detect Components

5.1.9 External Memory Components

The CS47L15 supports a master interface that can be used to download firmware and register-configuration data from an external non-volatile memory (e.g., EEPROM or flash memory). This enables the device to self-boot to an application-specific configuration and to be used independently of a host processor.

Compatible external-memory devices should be selected to meet the following criteria:

- Four-wire SPI interface (slave select, clock, data in, data out)
- SPI Mode 0 bus protocol support
- Memory size 500 kBit (minimum), 2–8 MBit (recommended)
- SPI speed 6 MHz (minimum), 20–40 MHz (recommended)
- Operating voltage compatible with DBVDD

The CS47L15 reads the external memory using the read instruction sequences illustrated in Fig. 4-79 and Fig. 4-80. As a minimum requirement, the external memory must support the standard read instruction shown in Fig. 4-79.

The following memory devices are recommended for use with the CS47L15 master-boot function. These devices have been chosen for compatibility with the CS47L15 master-boot function, and also for compatibility with the CS47L15 development tools. Please contact your local Cirrus Logic representative for details of the external memory development tool.

- Microchip Technology SST25WF080B (8 MBit, 40 MHz)
- Winbound Electronics W25Q80BWSVIG (8 MBit, 80 MHz)
- Atmel AT25DL161 (16 MBit, 100 MHz)

5.2 Resets Summary

Table 5-3 summarizes of the CS47L15 registers and other programmable memory under different reset conditions. The associated events and conditions are listed as follows:

- A power-on reset occurs when AVDD or DBVDD is below its respective reset threshold. Note that DCVDD is also required for initial start-up; subsequent interruption to DCVDD should only be permitted as part of a control sequence for entering Sleep Mode.
- A hardware reset occurs when the $\overline{\text{RESET}}$ input is asserted (Logic 0).
- A software reset occurs when register R0 is written to.
- Sleep Mode is selected when DCVDD is removed. Note that the AVDD and DBVDD supplies must be present throughout the Sleep Mode duration.

Table 5-3. Memory Reset Summary

Reset Type	Always-On Registers [1]	Other Registers	Control-Write Sequencer Memory	DSP Firmware Memory
Power-on reset	Reset	Reset	Reset	Reset
Hardware reset	Reset	Reset	Reset	Reset
Software reset	Reset	Reset	Retained	Retained [2]
Sleep Mode	Retained	Reset	Reset	Reset

1. See Section 4.10 for details of Sleep Mode and the always-on registers.
2. To retain the DSP firmware memory contents during software reset, it must be ensured that DCVDD is held above its reset threshold.

5.3 Output-Signal Drive-Strength Control

The CS47L15 supports configurable drive-strength control for the digital output pins. This can be used to assist system-level integration and design considerations.

The drive-strength control bits are described in Table 5-4. Note that, in the case of bidirectional pins (e.g., GPIO*n*), the drive-strength control bits are only applicable if the pin is configured as an output.

Table 5-4. Output Drive-Strength and Slew-Rate Control

Register Address	Bit	Label	Default	Description
R5889 (0x1701) GPIO1_CTRL2	12:11	GP1_DRV_STR[1:0]	01	AIF1TXDAT/GPIO1 output drive strength 00 = 4 mA 01 = 8 mA 10 = 12 mA 11 = 16 mA
R5891 (0x1703) GPIO2_CTRL2	12:11	GP2_DRV_STR[1:0]	01	AIF1RXDAT/GPIO2 output drive strength Field description is as above.
R5893 (0x1705) GPIO3_CTRL2	12:11	GP3_DRV_STR[1:0]	01	AIF1BCLK/GPIO3 output drive strength Field description is as above.
R5895 (0x1707) GPIO4_CTRL2	12:11	GP4_DRV_STR[1:0]	01	AIF1LRCLK/GPIO4 output drive strength Field description is as above.
R5897 (0x1709) GPIO5_CTRL2	12:11	GP5_DRV_STR[1:0]	01	AIF2TXDAT/GPIO5 output drive strength Field description is as above.
R5899 (0x170B) GPIO6_CTRL2	12:11	GP6_DRV_STR[1:0]	01	AIF2RXDAT/GPIO6 output drive strength Field description is as above.
R5901 (0x170D) GPIO7_CTRL2	12:11	GP7_DRV_STR[1:0]	01	AIF2BCLK/GPIO7 output drive strength Field description is as above.
R5903 (0x170F) GPIO8_CTRL2	12:11	GP8_DRV_STR[1:0]	01	AIF2LRCLK/GPIO8 output drive strength Field description is as above.
R5905 (0x1711) GPIO9_CTRL2	12:11	GP9_DRV_STR[1:0]	01	AIF3TXDAT/GPIO9 output drive strength Field description is as above.
R5907 (0x1713) GPIO10_CTRL2	12:11	GP10_DRV_STR[1:0]	01	AIF3RXDAT/GPIO10 output drive strength Field description is as above.
R5909 (0x1715) GPIO11_CTRL2	12:11	GP11_DRV_STR[1:0]	01	AIF3BCLK/GPIO11 output drive strength Field description is as above.

Table 5-4. Output Drive-Strength and Slew-Rate Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R5911 (0x1717) GPIO12_CTRL2	12:11	GP12_DRV_ STR[1:0]	01	AIF3LRCLK/GPIO12 output drive strength Field description is as above.
R5913 (0x1719) GPIO13_CTRL2	12:11	GP13_DRV_ STR[1:0]	01	SPKTXDAT/GPIO13 output drive strength Field description is as above.
R5915 (0x171B) GPIO14_CTRL2	12:11	GP14_DRV_ STR[1:0]	01	SPKCLK/GPIO14 output drive strength Field description is as above.
R5917 (0x171D) GPIO15_CTRL2	12:11	GP15_DRV_ STR[1:0]	01	SPKRXTDAT/GPIO15 output drive strength Field description is as above.

5.4 Digital Audio Interface Clocking Configurations

The digital audio interfaces (AIF1–AIF3) can be configured in master or slave modes. In all applications, it is important that the system clocking configuration is correctly designed. Incorrect clock configurations lead to audible clicks arising from dropped or repeated audio samples; this is caused by the inherent tolerances of multiple asynchronous system clocks.

To ensure reliable clocking of the audio interface functions, the external interface clocks (e.g., BCLK, LRCLK) must be derived from the same clock source as SYSCLK.

In AIF Master Mode, the external BCLK and LRCLK signals are generated by the CS47L15 and synchronization of these signals with SYSCLK is ensured. In this case, clocking of the AIF is typically derived from the MCLK1 or MCLK2 inputs, either directly or via the FLL circuit. Alternatively, another AIF*n* interface (configured in Slave Mode) can be used to provide the reference clock to which the AIF master can be synchronized.

In AIF Slave Mode, the external BCLK and LRCLK signals are generated by another device, as inputs to the CS47L15. In this case, the system clock (SYSCLK) must be generated from a source that is synchronized to the external BCLK and LRCLK inputs.

In a typical Slave Mode application, the BCLK input is selected as the clock reference, using the FLL to perform frequency shifting. The MCLK1 or MCLK2 inputs can also be used, but only if the selected clock is synchronized externally to the BCLK and LRCLK inputs.

The valid AIF clocking configurations are listed in Table 5-5 for AIF Master and AIF Slave Modes.

Table 5-5. AIF Clocking Configurations

AIF Mode	Clocking Configuration
AIF Master Mode	SYSCLK_SRC selects MCLK1 or MCLK2 as SYSCLK source.
	SYSCLK_SRC selects FLL1 as SYSCLK source; FLL1_REFCLK_SRC selects MCLK1 or MCLK2 as FLL1 source.
	SYSCLK_SRC selects FLL1 as SYSCLK source; FLL1_REFCLK_SRC selects a different interface (BCLK, LRCLK) as FLL1 source.
AIF Slave Mode	SYSCLK_SRC selects FLL1 as SYSCLK source; FLL1_REFCLK_SRC selects BCLK as FLL1 source.
	SYSCLK_SRC selects MCLK1 or MCLK2 as SYSCLK source, provided MCLK is externally synchronized to the BCLK input.
	SYSCLK_SRC selects FLL1 as SYSCLK source; FLL1_REFCLK_SRC selects MCLK1 or MCLK2 as FLL1 source, provided MCLK is externally synchronized to the BCLK input.
	SYSCLK_SRC selects FLL1 as SYSCLK source; FLL1_REFCLK_SRC selects a different interface (BCLK, LRCLK) as FLL1 source, provided the other interface is externally synchronized to the BCLK input.

In each case, the SYSCLK frequency must be a valid ratio to the LRCLK frequency; the supported clocking rates are defined by the SYSCLK_FREQ and SAMPLE_RATE_*n* fields.

The valid AIF clocking configurations are shown in Fig. 5-13 to Fig. 5-19. Note that, where MCLK1 is shown as the clock source, it is equally possible to select MCLK2 as the clock source.

Fig. 5-13 shows AIF Master Mode operation, using MCLK as the clock reference.

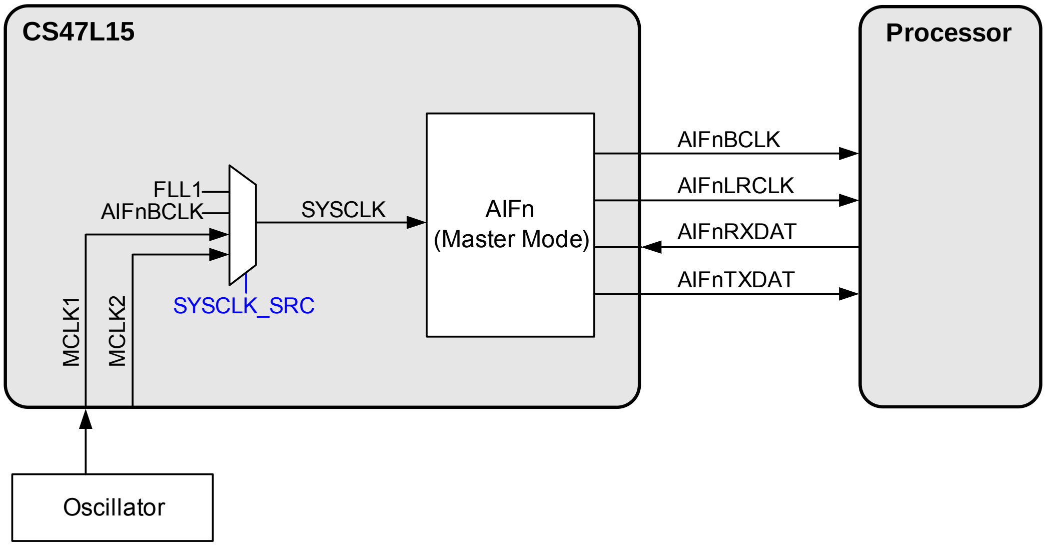


Figure 5-13. AIF Master Mode, Using MCLK as Reference

Fig. 5-14 shows AIF Master Mode operation, using MCLK as the clock reference. In this example, the FLL is used to generate the system clock, with MCLK as the reference.

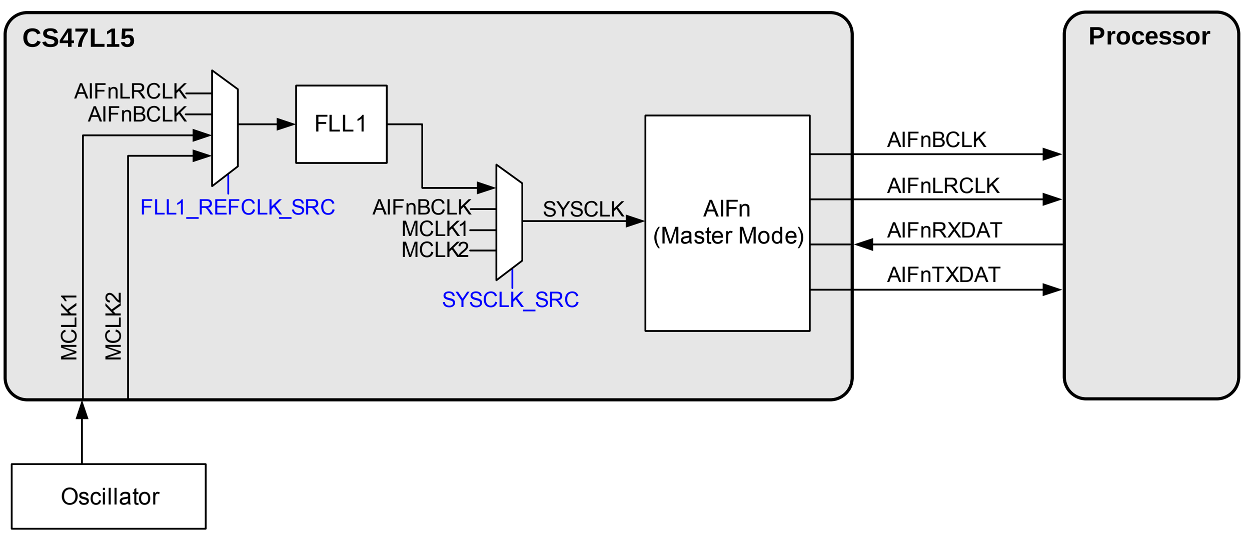


Figure 5-14. AIF Master Mode, Using MCLK and FLL as Reference

Fig. 5-15 shows AIF Master Mode operation, using a separate interface as the clock reference. In this example, the FLL is used to generate the system clock, with LRCLK or BCLK input (from a separate AIFn slave interface) as the reference.

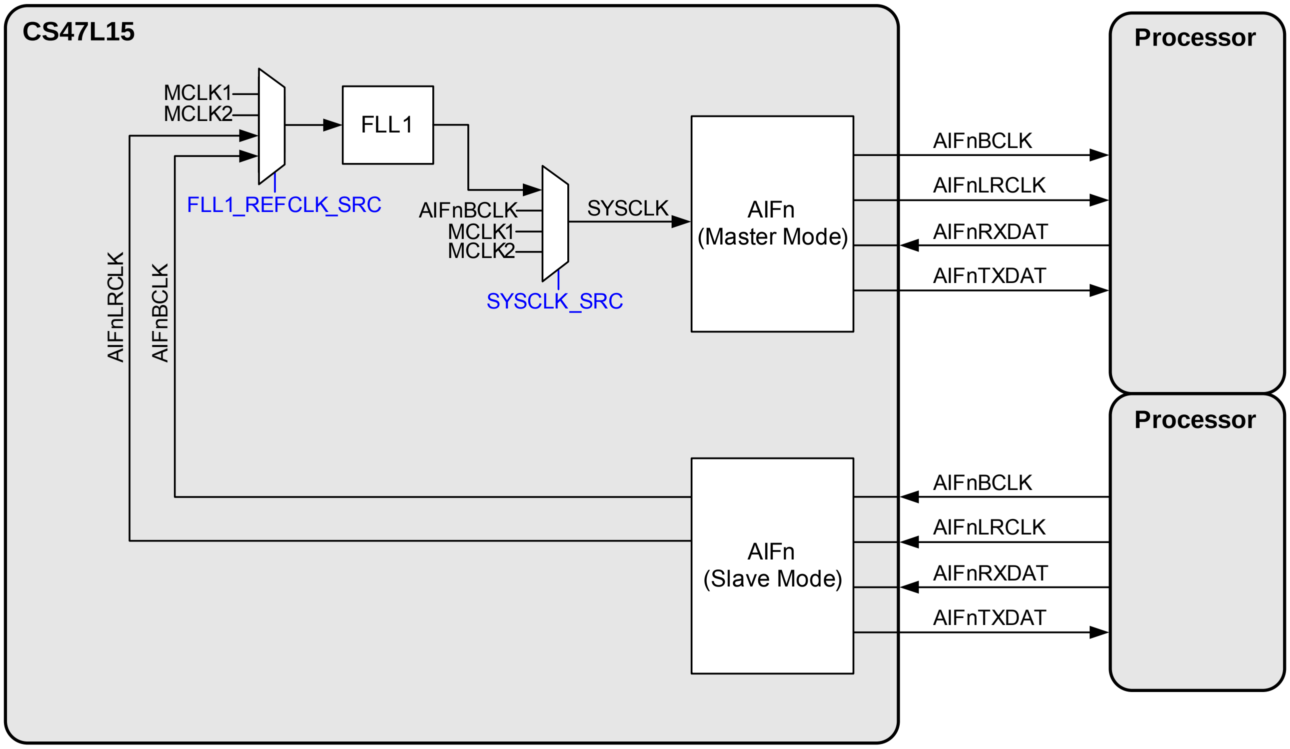


Figure 5-15. AIF Master Mode, Using Another Interface as Reference

Fig. 5-16 shows AIF Slave Mode operation, using BCLK as the clock reference. In this example, the FLL is used to generate the system clock, with BCLK as the reference.

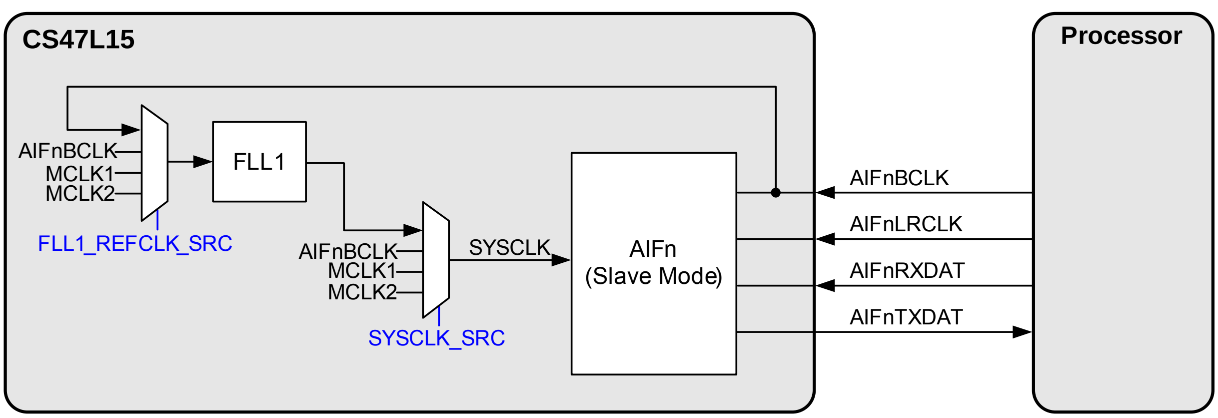


Figure 5-16. AIF Slave Mode, Using BCLK and FLL as Reference

Fig. 5-17 shows AIF Slave Mode operation, using MCLK as the clock reference. For correct operation, the MCLK input must be fully synchronized to the audio interface.

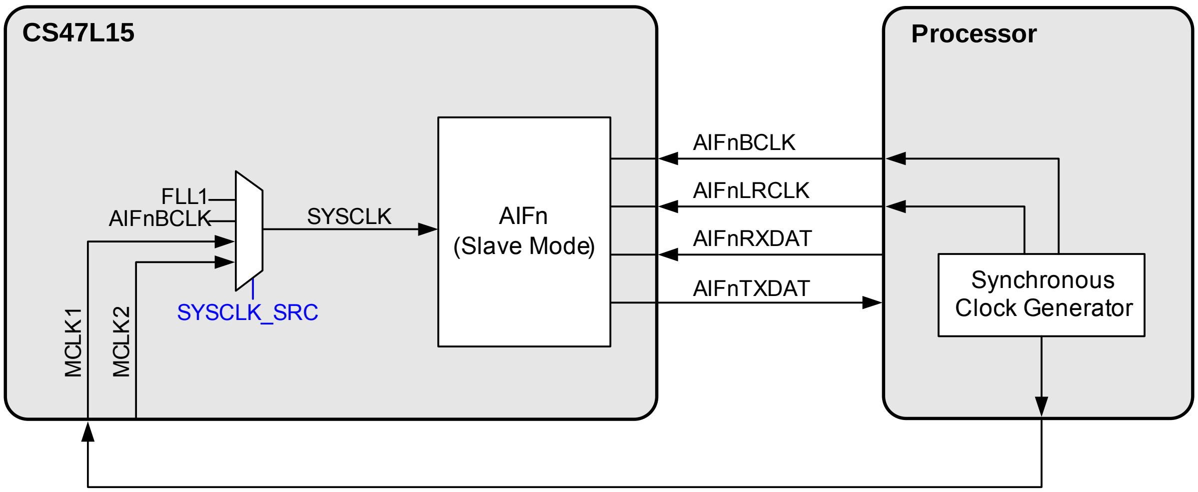


Figure 5-17. AIF Slave Mode, Using MCLK as Reference

Fig. 5-18 shows AIF Slave Mode operation, using MCLK as the clock reference. For correct operation, the MCLK input must be fully synchronized to the audio interface. In this example, the FLL is used to generate the system clock, with MCLK as the reference.

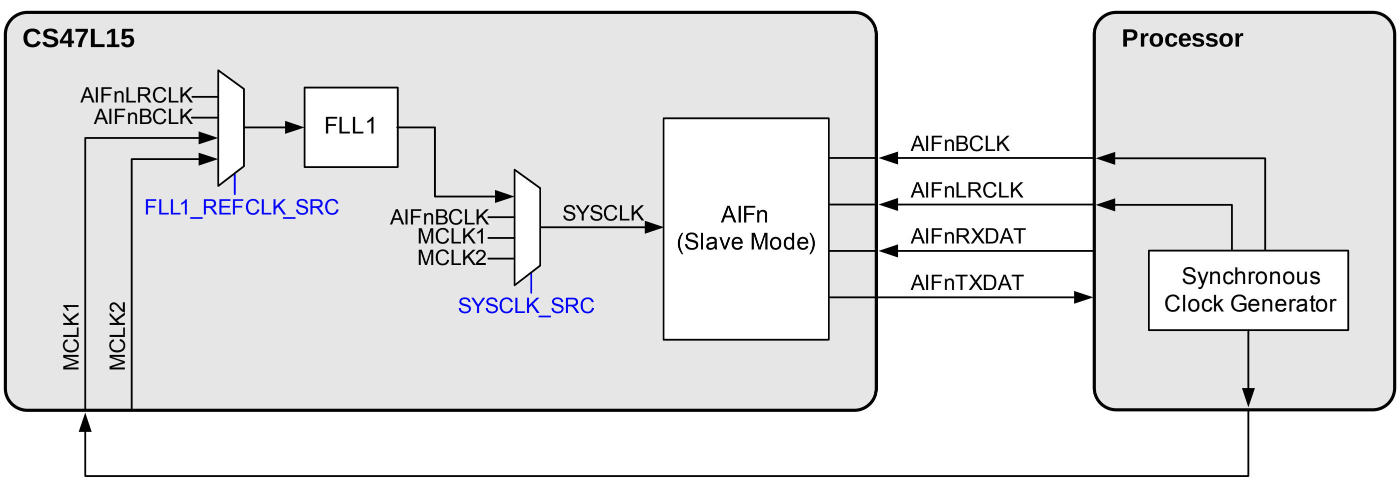


Figure 5-18. AIF Slave Mode, Using MCLK and FLL as Reference

Fig. 5-19 shows AIF Slave Mode operation, using a separate interface as the clock reference. In this example, the FLL is used to generate the system clock, with LRCLK or BCLK input (from a separate AIFn slave interface) as the reference. For correct operation, the reference input must be fully synchronized to the other audio interfaces.

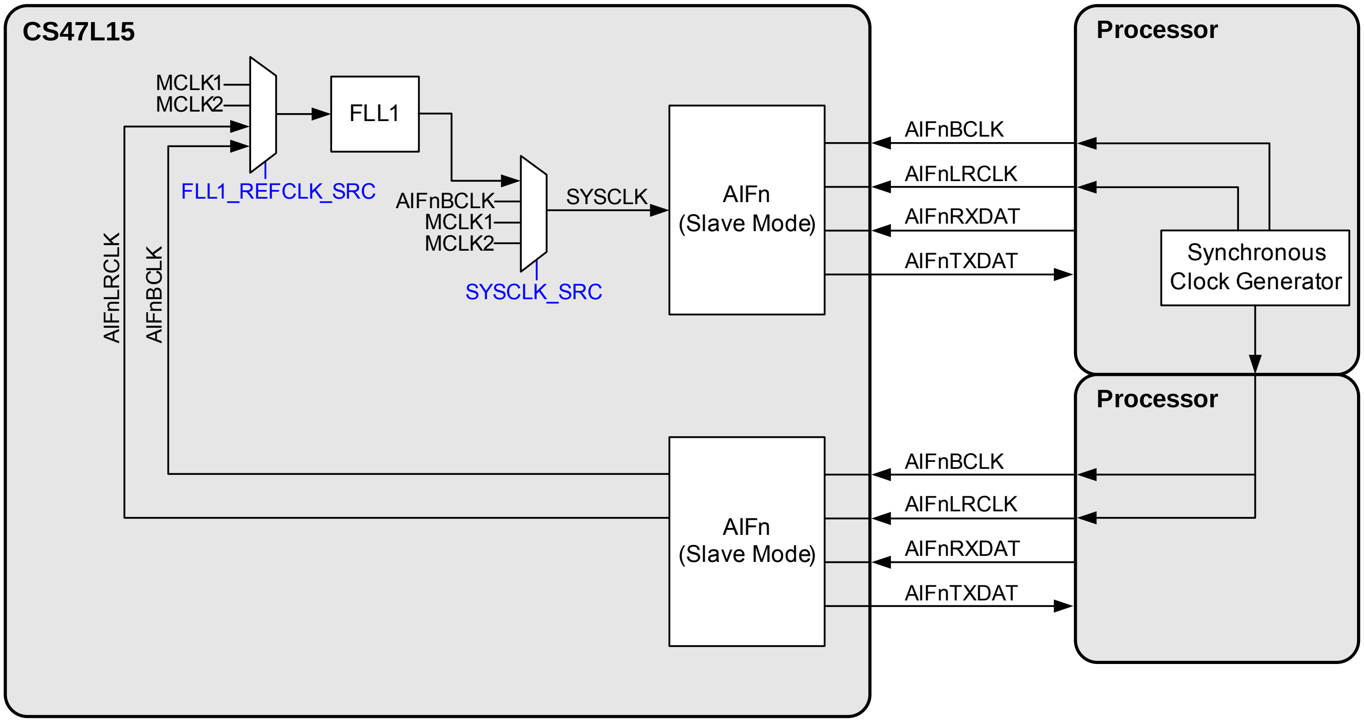


Figure 5-19. AIF Slave Mode, Using Another Interface as Reference

5.5 PCB Layout Considerations

Poor PCB layout degrades the performance and is a contributory factor in EMI, ground bounce, and resistive voltage losses. All external components should be placed as close to the CS47L15 device as possible, with current loop areas kept as small as possible.

PCB layout should be carefully considered, to ensure optimum performance of the CS47L15. Poor PCB layout degrades the performance and is a contributory factor in EMI, ground bounce, and resistive voltage losses. All external components should be placed close to the CS47L15, with current loop areas kept as small as possible. The following specific considerations should be noted:

- Placement of the charge pump capacitors is a high priority requirement—these capacitors (particularly the fly-back capacitors) must be placed as close as possible to the CS47L15.
- Decoupling capacitors should be placed as close as possible to the CS47L15. The connection between AGND, the AVDD decoupling capacitor, and the main system ground should be made at a single point as close as possible to the AGND ball of the CS47L15.
- The VREFC capacitor should be placed as close as possible to the CS47L15. The ground connection to the VREFC capacitor should be as close as possible to the AGND ball of the CS47L15.
- If multiple power supply rails are connected to a single supply source, it is recommended to provide separate PCB tracks connecting each rail to the supply. This configuration is also known as *star connection*.
- If power supply rails are routed between different layers of the PCB, it is recommended to use several track vias, in order to minimize resistive voltage losses.
- Differential input signal tracks should be routed as a pair, ensuring similar length/width dimensions on each track. Input signal paths should be kept away from high frequency digital signals.
- Differential output signal tracks should be routed as a pair, ensuring similar length/width dimensions on each track. The tracks should provide a low resistance path from the device output pin to the load (< 1% of the minimum load).

- The headphone output ground-feedback pins should be connected to GND as close as possible to the respective headphone jack ground pin. The ground-feedback PCB track should follow the same route as the respective output signal paths.

6 Register Map

The CS47L15 control registers are listed in the following tables. Note that only the register addresses described here should be accessed; writing to other addresses may result in undefined behavior. Register bits that are not documented should not be changed from the default values.

The CS47L15 register map is defined in two regions:

- The codec register space (below 0x3000) is defined in 16-bit word format
- The DSP register space (from 0x3000 upwards) is defined in 32-bit word format

It is important to ensure that all control interface register operations use the applicable data word format, in accordance with the applicable register addresses.

The 16-bit codec register space is described in Table 6-1.

Table 6-1. Register Map Definition—16-bit region

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R0 (0h)	Software_Reset	SW_RST_DEV_ID [15:0]																6370h
R1 (1h)	Hardware_Revision	0	0	0	0	0	0	0	0	HW_REVISION [7:0]								0000h
R2 (2h)	Software_Revision	0	0	0	0	0	0	0	0	SW_REVISION [7:0]								0000h
R8 (8h)	Ctrl_IF_CFG_1	0	0	1	1	0	1	1	1	MISO_SCLK_PD	0	1	1	1	0	1	1	373Bh
R18 (12h)	Ctrl_IF_Pin_Cfg_1	1	0	1	0	0	I2C_DEBUG	0	0	0	0	0	0	0	0	0	1	A401h
R22 (16h)	Write_Sequencer_Ctrl_0	0	0	0	0	WSEQ_ABORT	WSEQ_START	WSEQ_ENA	WSEQ_START_INDEX [8:0]									0000h
R23 (17h)	Write_Sequencer_Ctrl_1	0	0	0	0	0	0	WSEQ_BUSY	WSEQ_CURRENT_INDEX [8:0]									0000h
R24 (18h)	Write_Sequencer_Ctrl_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	WSEQ_BOOT_START	WSEQ_LOAD_MEM	0000h
R32 (20h)	Tone_Generator_1	0	TONE_RATE [3:0]				0	TONE_OFFSET [1:0]		0	0	TONE2_OVD	TONE1_OVD	0	0	TONE2_ENA	TONE1_ENA	0000h
R33 (21h)	Tone_Generator_2	TONE1_LVL [23:8]																1000h
R34 (22h)	Tone_Generator_3	0	0	0	0	0	0	0	0	TONE1_LVL [7:0]								0000h
R35 (23h)	Tone_Generator_4	TONE2_LVL [23:8]																1000h
R36 (24h)	Tone_Generator_5	0	0	0	0	0	0	0	0	TONE2_LVL [7:0]								0000h
R48 (30h)	PWM_Drive_1	0	PWM_RATE [3:0]				PWM_CLK_SEL [2:0]			0	0	PWM2_OVD	PWM1_OVD	0	0	PWM2_ENA	PWM1_ENA	0000h
R49 (31h)	PWM_Drive_2	0	0	0	0	0	0	PWM1_LVL [9:0]										0100h
R50 (32h)	PWM_Drive_3	0	0	0	0	0	0	PWM2_LVL [9:0]										0100h
R65 (41h)	Sequence_control	0	0	0	0	0	0	0	0	WSEQ_ENA_MICD_CLAMP_FALL	WSEQ_ENA_MICD_CLAMP_RISE	0	0	0	0	0	0	0000h
R66 (42h)	Spare_Triggers	WSEQ_TRG16	WSEQ_TRG15	WSEQ_TRG14	WSEQ_TRG13	WSEQ_TRG12	WSEQ_TRG11	WSEQ_TRG10	WSEQ_TRG9	WSEQ_TRG8	WSEQ_TRG7	WSEQ_TRG6	WSEQ_TRG5	WSEQ_TRG4	WSEQ_TRG3	WSEQ_TRG2	WSEQ_TRG1	0000h
R75 (4Bh)	Spare_Sequence_Select_1	0	0	0	0	0	0	0	WSEQ_TRG1_INDEX [8:0]									01FFh
R76 (4Ch)	Spare_Sequence_Select_2	0	0	0	0	0	0	0	WSEQ_TRG2_INDEX [8:0]									01FFh
R77 (4Dh)	Spare_Sequence_Select_3	0	0	0	0	0	0	0	WSEQ_TRG3_INDEX [8:0]									01FFh
R78 (4Eh)	Spare_Sequence_Select_4	0	0	0	0	0	0	0	WSEQ_TRG4_INDEX [8:0]									01FFh
R79 (4Fh)	Spare_Sequence_Select_5	0	0	0	0	0	0	0	WSEQ_TRG5_INDEX [8:0]									01FFh
R80 (50h)	Spare_Sequence_Select_6	0	0	0	0	0	0	0	WSEQ_TRG6_INDEX [8:0]									01FFh

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R89 (59h)	Spare_Sequence_Select_7	0	0	0	0	0	0	0	WSEQ_TRG7_INDEX [8:0]									01FFh
R90 (5Ah)	Spare_Sequence_Select_8	0	0	0	0	0	0	0	WSEQ_TRG8_INDEX [8:0]									01FFh
R91 (5Bh)	Spare_Sequence_Select_9	0	0	0	0	0	0	0	WSEQ_TRG9_INDEX [8:0]									01FFh
R92 (5Ch)	Spare_Sequence_Select_10	0	0	0	0	0	0	0	WSEQ_TRG10_INDEX [8:0]									01FFh
R93 (5Dh)	Spare_Sequence_Select_11	0	0	0	0	0	0	0	WSEQ_TRG11_INDEX [8:0]									01FFh
R94 (5Eh)	Spare_Sequence_Select_12	0	0	0	0	0	0	0	WSEQ_TRG12_INDEX [8:0]									01FFh
R97 (61h)	Sample_Rate_Sequence_Select_1	0	0	0	0	0	0	0	WSEQ_SAMPLE_RATE_DETECT_A_INDEX [8:0]									01FFh
R98 (62h)	Sample_Rate_Sequence_Select_2	0	0	0	0	0	0	0	WSEQ_SAMPLE_RATE_DETECT_B_INDEX [8:0]									01FFh
R99 (63h)	Sample_Rate_Sequence_Select_3	0	0	0	0	0	0	0	WSEQ_SAMPLE_RATE_DETECT_C_INDEX [8:0]									01FFh
R100 (64h)	Sample_Rate_Sequence_Select_4	0	0	0	0	0	0	0	WSEQ_SAMPLE_RATE_DETECT_D_INDEX [8:0]									01FFh
R102 (66h)	Always_On_Triggers_Sequence_Select_1	0	0	0	0	0	0	0	WSEQ_MICD_CLAMP_RISE_INDEX [8:0]									01FFh
R103 (67h)	Always_On_Triggers_Sequence_Select_2	0	0	0	0	0	0	0	WSEQ_MICD_CLAMP_FALL_INDEX [8:0]									01FFh
R104 (68h)	Spare_Sequence_Select_13	0	0	0	0	0	0	0	WSEQ_TRG13_INDEX [8:0]									01FFh
R105 (69h)	Spare_Sequence_Select_14	0	0	0	0	0	0	0	WSEQ_TRG14_INDEX [8:0]									01FFh
R106 (6Ah)	Spare_Sequence_Select_15	0	0	0	0	0	0	0	WSEQ_TRG15_INDEX [8:0]									01FFh
R107 (6Bh)	Spare_Sequence_Select_16	0	0	0	0	0	0	0	WSEQ_TRG16_INDEX [8:0]									01FFh
R110 (6Eh)	Trigger_Sequence_Select_32	0	0	0	0	0	0	0	WSEQ_DRC1_SIG_DET_RISE_INDEX [8:0]									01FFh
R111 (6Fh)	Trigger_Sequence_Select_33	0	0	0	0	0	0	0	WSEQ_DRC1_SIG_DET_FALL_INDEX [8:0]									01FFh
R120 (78h)	Eventlog_Sequence_Select_1	0	0	0	0	0	0	0	WSEQ_EVENTLOG1_INDEX [8:0]									01FFh
R121 (79h)	Eventlog_Sequence_Select_2	0	0	0	0	0	0	0	WSEQ_EVENTLOG2_INDEX [8:0]									01FFh
R140 (8Ch)	User_Key_Ctrl	USER_KEY_CTRL [15:0]																0000h
R144 (90h)	Haptics_Control_1	0	HAP_RATE [3:0]				0	0	0	0	0	0	ONESHOT_TRIG	HAP_CTRL [1:0]		HAP_ACT	0	0000h
R145 (91h)	Haptics_Control_2	0	LRA_FREQ [14:0]															7FFFh
R146 (92h)	Haptics_phase_1_intensity	0	0	0	0	0	0	0	0	PHASE1_INTENSITY [7:0]								0000h
R147 (93h)	Haptics_phase_1_duration	0	0	0	0	0	0	0	PHASE1_DURATION [8:0]									0000h
R148 (94h)	Haptics_phase_2_intensity	0	0	0	0	0	0	0	0	PHASE2_INTENSITY [7:0]								0000h
R149 (95h)	Haptics_phase_2_duration	0	0	0	0	0	PHASE2_DURATION [10:0]											0000h
R150 (96h)	Haptics_phase_3_intensity	0	0	0	0	0	0	0	0	PHASE3_INTENSITY [7:0]								0000h
R151 (97h)	Haptics_phase_3_duration	0	0	0	0	0	0	0	PHASE3_DURATION [8:0]									0000h
R152 (98h)	Haptics_Status	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	ONESHOT_STS	0000h
R160 (A0h)	Comfort_Noise_Generator	0	NOISE_GEN_RATE [3:0]				0	0	0	0	0	NOISE_GEN_ENA	NOISE_GEN_GAIN [4:0]					0000h
R256 (100h)	Clock_32k_1	0	0	0	0	0	0	0	0	0	CLK_32K_ENA	0	0	0	0	CLK_32K_SRC [1:0]		0002h
R257 (101h)	System_Clock_1	SYSCLK_FRAC	0	0	0	0	SYSCLK_FREQ [2:0]			0	SYSCLK_ENA	0	0	SYSCLK_SRC [3:0]				0404h
R258 (102h)	Sample_rate_1	0	0	0	0	0	0	0	0	0	0	0	SAMPLE_RATE_1 [4:0]					0011h
R259 (103h)	Sample_rate_2	0	0	0	0	0	0	0	0	0	0	0	SAMPLE_RATE_2 [4:0]					0011h
R260 (104h)	Sample_rate_3	0	0	0	0	0	0	0	0	0	0	0	SAMPLE_RATE_3 [4:0]					0011h
R266 (10Ah)	Sample_rate_1_status	0	0	0	0	0	0	0	0	0	0	0	SAMPLE_RATE_1_STS [4:0]					0000h
R267 (10Bh)	Sample_rate_2_status	0	0	0	0	0	0	0	0	0	0	0	SAMPLE_RATE_2_STS [4:0]					0000h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R268 (10Ch)	Sample_rate_3_status	0	0	0	0	0	0	0	0	0	0	0	SAMPLE_RATE_3_STS [4:0]					0000h
R288 (120h)	DSP_Clock_1	0	0	0	0	0	0	1	1	0	DSP_CLK_ENA	0	0	DSP_CLK_SRC [3:0]				0304h
R290 (122h)	DSP_Clock_2	DSP_CLK_FREQ [15:0]																0000h
R292 (124h)	DSP_Clock_3	FLL_AO_FREQ [15:0]																0000h
R294 (126h)	DSP_Clock_4	DSP_CLK_FREQ_STS [15:0]																0000h
R295 (127h)	DSP_Clock_5	0	0	0	0	0	0	0	0	0	0	0	0	DSP_CLK_SRC_STS [3:0]				0000h
R329 (149h)	Output_system_clock	OPCLK_ENA	0	0	0	0	0	0	0	OPCLK_DIV [4:0]					OPCLK_SEL [2:0]			0000h
R334 (14Eh)	Clock_Gen_Pad_Ctrl	0	0	0	0	DSP_JTAG_MODE	1	MSTRBOOT_PD	MCLK2_PD	MCLK1_PD	1	1	0	0	0	0	0	0660h
R338 (152h)	Rate_Estimator_1	0	0	0	0	0	0	0	0	0	0	0	TRIG_ON_STARTUP	LRCLK_SRC [2:0]			RATE_EST_ENA	0000h
R339 (153h)	Rate_Estimator_2	0	0	0	0	0	0	0	0	0	0	0	SAMPLE_RATE_DETECT_A [4:0]					0000h
R340 (154h)	Rate_Estimator_3	0	0	0	0	0	0	0	0	0	0	0	SAMPLE_RATE_DETECT_B [4:0]					0000h
R341 (155h)	Rate_Estimator_4	0	0	0	0	0	0	0	0	0	0	0	SAMPLE_RATE_DETECT_C [4:0]					0000h
R342 (156h)	Rate_Estimator_5	0	0	0	0	0	0	0	0	0	0	0	SAMPLE_RATE_DETECT_D [4:0]					0000h
R352 (160h)	Clocking_debug_5	0	0	0	0	0	0	0	0	0	SYSCLK_FREQ_STS [2:0]			SYSCLK_SRC_STS [3:0]				0000h
R369 (171h)	FLL1_Control_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLL1_FREERUN	FLL1_ENA	0002h
R370 (172h)	FLL1_Control_2	FLL1_CTRL_UPD	0	0	0	0	0	FLL1_N [9:0]										0008h
R371 (173h)	FLL1_Control_3	FLL1_THETA [15:0]																0018h
R372 (174h)	FLL1_Control_4	FLL1_LAMBDA [15:0]																007Dh
R373 (175h)	FLL1_Control_5	0	0	0	0	FLL1_FRATIO [3:0]				0	0	0	0	0	0	0	0	0000h
R374 (176h)	FLL1_Control_6	0	0	0	0	0	0	0	0	FLL1_REFCLK_DIV [1:0]		0	0	FLL1_REFCLK_SRC [3:0]				0000h
R375 (177h)	FLL1_Loop_Filter_Test_1	FLL1_FRC_INTEG_UPD	0	0	0	FLL1_FRC_INTEG_VAL [11:0]												0281h
R376 (178h)	FLL1_NCO_Test_0	FLL1_INTEG_VALID	0	0	0	FLL1_INTEG [11:0]												0000h
R377 (179h)	FLL1_Control_7	0	0	0	0	0	0	0	0	0	0	FLL1_GAIN [3:0]				0	0	0000h
R378 (17Ah)	FLL1_Control_8	0	0	1	0	FLL1_PHASE_ENA	0	0	1	0	0	0	0	0	1	1	0	2906h
R385 (181h)	FLL1_Synchroniser_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLL1_SYNC_ENA	0000h
R386 (182h)	FLL1_Synchroniser_2	0	0	0	0	0	0	FLL1_SYNC_N [9:0]										0000h
R387 (183h)	FLL1_Synchroniser_3	FLL1_SYNC_THETA [15:0]																0000h
R388 (184h)	FLL1_Synchroniser_4	FLL1_SYNC_LAMBDA [15:0]																0000h
R389 (185h)	FLL1_Synchroniser_5	0	0	0	0	0	FLL1_SYNC_FRATIO [2:0]			0	0	0	0	0	0	0	0	0000h
R390 (186h)	FLL1_Synchroniser_6	0	0	0	0	0	0	0	0	FLL1_SYNCCLK_DIV [1:0]		0	0	FLL1_SYNCCLK_SRC [3:0]				0000h
R391 (187h)	FLL1_Synchroniser_7	0	0	0	0	0	0	0	0	0	0	FLL1_SYNC_GAIN [3:0]				0	FLL1_SYNC_DFSAT	0001h
R393 (189h)	FLL1_Spread_Spectrum	0	0	0	0	0	0	0	0	0	0	FLL1_SS_AMPL [1:0]		FLL1_SS_FREQ [1:0]		FLL1_SS_SEL [1:0]		0000h
R394 (18Ah)	FLL1_GPIO_Clock	0	0	0	0	0	0	0	0	FLL1_GPCLK_DIV [6:0]							FLL1_GPCLK_ENA	0004h
R465 (1D1h)	FLL_AO_Control_1	0	0	0	0	0	0	0	0	0	0	0	0	0	FLL_AO_HOLD	0	FLL_AO_ENA	0004h
R470 (1D6h)	FLL_AO_Control_6	1	0	0	0	0	0	0	0	0	0	0	0	FLL_AO_REFCLK_SRC [3:0]				8004h
R490 (1EAh)	FLL_AO_GPIO_Clock	0	0	0	0	0	0	0	0	FLL_AO_GPCLK_DIV [6:0]							FLL_AO_GPCLK_ENA	0002h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R536 (218h)	Mic_Bias_Ctrl_1	MICB1_EXT_CAP	0	0	0	0	0	0	MICB1_LVL [3:0]				0	MICB1_RATE	MICB1_DISCH	MICB1_BYPASS	MICB1_ENA	00E6h
R540 (21Ch)	Mic_Bias_Ctrl_5	0	0	0	0	0	0	MICB1C_DISCH	MICB1C_ENA	0	0	MICB1B_DISCH	MICB1B_ENA	0	0	MICB1A_DISCH	MICB1A_ENA	0222h
R620 (26Ch)	SPK_Watchdog_1	0	0	0	0	0	0	0	0	0	0	0	0	SPK_SHUTDOWN_TIMER_SEL [3:0]				0000h
R665 (299h)	Headphone_Detect_0	HPD_OVD_ENA	HPD_OUT_SEL [2:0]			HPD_FRC_SEL [3:0]				HPD_SENSE_SEL [3:0]				0	HPD_GND_SEL [2:0]			0000h
R667 (29Bh)	Headphone_Detect_1	0	0	0	0	0	HPD_IMPEDANCE_RANGE [1:0]		0	0	0	0	HPD_CLK_DIV [1:0]		HPD_RATE [1:0]		HPD_POLL (M)	0000h
R668 (29Ch)	Headphone_Detect_2	HPD_DONE	HPD_LVL [14:0]															0000h
R669 (29Dh)	Headphone_Detect_3	0	0	0	0	0	0	HPD_DACVAL [9:0]										0000h
R674 (2A2h)	Mic_Detect_1_Control_0	MICD1_ADC_MODE	0	0	0	0	0	0	0	MICD1_SENSE_SEL [3:0]				0	MICD1_GND_SEL [2:0]			0010h
R675 (2A3h)	Mic_Detect_1_Control_1	MICD1_BIAS_STARTTIME [3:0]				MICD1_RATE [3:0]				MICD1_BIAS_SRC [3:0]				0	0	MICD1_DBTIME	MICD1_ENA	1102h
R676 (2A4h)	Mic_Detect_1_Control_2	0	0	0	0	0	0	0	0	MICD1_LVL_SEL [7:0]								009Fh
R677 (2A5h)	Mic_Detect_1_Control_3	0	0	0	0	0	MICD1_LVL [8:0]									MICD1_VALID	MICD1_STS	0000h
R683 (2ABh)	Mic_Detect_1_Control_4	MICD1_ADCVAL_DIFF [7:0]								0	MICD1_ADCVAL [6:0]							0000h
R710 (2C6h)	Micd_Clamp_control	0	0	0	0	0	0	0	0	0	0	0	MICD_CLAMP_OVD	MICD_CLAMP_MODE [3:0]				0010h
R712 (2C8h)	GP_Switch_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SW1_MODE [1:0]		0000h
R723 (2D3h)	Jack_detect_analogue	0	0	0	0	0	0	0	0	0	0	0	0	0	0	JD2_ENA	JD1_ENA	0000h
R768 (300h)	Input_Enables	0	0	0	0	0	0	0	0	0	0	0	0	IN2L_ENA	IN2R_ENA	IN1L_ENA	IN1R_ENA	0000h
R769 (301h)	Input_Enables_Status	0	0	0	0	0	0	0	0	0	0	0	0	IN2L_ENA_STS	IN2R_ENA_STS	IN1L_ENA_STS	IN1R_ENA_STS	0000h
R776 (308h)	Input_Rate	0	IN_RATE [3:0]				0	0	0	0	0	0	0	0	0	0	0	0000h
R777 (309h)	Input_Volume_Ramp	0	0	0	0	0	0	0	0	0	IN_VD_RAMP [2:0]			0	IN_VI_RAMP [2:0]			0022h
R780 (30Ch)	HPF_Control	0	0	0	0	0	0	0	0	0	0	0	0	0	IN_HPF_CUT [2:0]			0002h
R784 (310h)	IN1L_Control	IN1L_HPF	0	0	IN1_DMIC_SUP [1:0]		IN1_MODE	0	0	IN1L_PGA_VOL [6:0]							0	0080h
R785 (311h)	ADC_Digital_Volume_1L	0	IN1L_SRC [1:0]		0	0	0	IN_VU	IN1L_MUTE	IN1L_VOL [7:0]								0180h
R786 (312h)	DMIC1L_Control	IN1L_SIG_DET_ENA	0	0	0	0	IN1_OSR [2:0]			0	0	0	0	0	0	0	0	0500h
R788 (314h)	IN1R_Control	IN1R_HPF	0	0	IN1_DMICCLK_SRC [1:0] (K)		0	0	0	IN1R_PGA_VOL [6:0]							0	0080h
R789 (315h)	ADC_Digital_Volume_1R	0	IN1R_SRC [1:0]		0	0	0	IN_VU	IN1R_MUTE	IN1R_VOL [7:0]								0180h
R790 (316h)	DMIC1R_Control	IN1R_SIG_DET_ENA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h
R792 (318h)	IN2L_Control	IN2L_HPF	0	0	0	0	IN2_MODE	0	0	0	0	0	0	0	0	0	0	0000h
R793 (319h)	ADC_Digital_Volume_2L	0	0	0	0	IN2L_LP_MODE	0	IN_VU	IN2L_MUTE	IN2L_VOL [7:0]								0980h
R794 (31Ah)	DMIC2L_Control	IN2L_SIG_DET_ENA	0	0	0	0	IN2_OSR [2:0]			0	0	0	0	0	0	0	0	0500h
R796 (31Ch)	IN2R_Control	IN2R_HPF	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0800h
R797 (31Dh)	ADC_Digital_Volume_2R	0	0	0	0	1	0	IN_VU	IN2R_MUTE	IN2R_VOL [7:0]								0980h
R798 (31Eh)	DMIC2R_Control	IN2R_SIG_DET_ENA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h
R832 (340h)	Signal_Detect_Globals	0	0	0	0	0	0	0	IN_SIG_DET_THR [4:0]					IN_SIG_DET_HOLD [3:0]				0001h
R840 (348h)	Dig_Mic_Pad_Ctrl	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DMICDAT1_PD	0000h
R1024 (400h)	Output_Enables_1	EP_SEL	0	0	0	0	0	OUT5L_ENA	OUT5R_ENA	SPKOUTL_ENA	0	0	0	0	0	HP1L_ENA	HP1R_ENA	0000h
R1025 (401h)	Output_Status_1	0	0	0	0	0	0	OUT5L_ENA_STS	OUT5R_ENA_STS	OUT4L_ENA_STS	0	0	0	0	0	0	0	0000h
R1030 (406h)	Raw_Output_Status_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	OUT1L_ENA_STS	OUT1R_ENA_STS	0000h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R1032 (408h)	Output_Rate_1	0	OUT_RATE [3:0]				0	0	0	0	0	0	0	0	0	0	0	0000h
R1033 (409h)	Output_Volume_Ramp	0	0	0	0	0	0	0	0	0	OUT_VD_RAMP [2:0]			0	OUT_VI_RAMP [2:0]			0022h
R1040 (410h)	Output_Path_Config_1L	0	0	0	OUT1_MONO	0	0	0	0	1	0	0	0	0	0	0	0	0080h
R1041 (411h)	DAC_Digital_Volume_1L	0	0	0	0	0	0	OUT_VU	OUT1L_MUTE	OUT1L_VOL [7:0]								0180h
R1042 (412h)	Output_Path_Config_1	0	0	0	0	0	0	0	0	0	0	0	0	0	HP1_GND_SEL [2:0]			0000h
R1043 (413h)	Noise_Gate_Select_1L	0	0	0	0	OUT1L_NGATE_SRC [11:0]												0001h
R1045 (415h)	DAC_Digital_Volume_1R	0	0	0	0	0	0	OUT_VU	OUT1R_MUTE	OUT1R_VOL [7:0]								0180h
R1047 (417h)	Noise_Gate_Select_1R	0	0	0	0	OUT1R_NGATE_SRC [11:0]												0002h
R1065 (429h)	DAC_Digital_Volume_4L	0	0	0	0	0	0	OUT_VU	OUT4L_MUTE	OUT4L_VOL [7:0]								0180h
R1067 (42Bh)	Noise_Gate_Select_4L	0	0	0	0	OUT4L_NGATE_SRC [11:0]												0040h
R1072 (430h)	Output_Path_Config_5L	0	0	OUT5_OSR	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h
R1073 (431h)	DAC_Digital_Volume_5L	0	0	0	0	0	0	OUT_VU	OUT5L_MUTE	OUT5L_VOL [7:0]								0180h
R1075 (433h)	Noise_Gate_Select_5L	0	0	0	0	OUT5L_NGATE_SRC [11:0]												0100h
R1077 (435h)	DAC_Digital_Volume_5R	0	0	0	0	0	0	OUT_VU	OUT5R_MUTE	OUT5R_VOL [7:0]								0180h
R1079 (437h)	Noise_Gate_Select_5R	0	0	0	0	OUT5R_NGATE_SRC [11:0]												0200h
R1104 (450h)	DAC_AEC_Control_1	0	0	0	0	0	0	0	0	0	0	AEC1_LOOPBACK_SRC [3:0]				AEC1_ENA_STS	AEC1_LOOPBACK_ENA	0000h
R1105 (451h)	DAC_AEC_Control_2	0	0	0	0	0	0	0	0	0	0	AEC2_LOOPBACK_SRC [3:0]				AEC2_ENA_STS	AEC2_LOOPBACK_ENA	0000h
R1112 (458h)	Noise_Gate_Control	0	0	0	0	0	0	0	0	0	0	NGATE_HOLD [1:0]		NGATE_THR [2:0]			NGATE_ENA	0000h
R1168 (490h)	PDM_SPK1_CTRL_1	0	0	SPK1R_MUTE	SPK1L_MUTE	0	0	0	SPK1_MUTE_ENDIAN	SPK1_MUTE_SEQ [7:0]								0069h
R1169 (491h)	PDM_SPK1_CTRL_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SPK1_FMT	0000h
R1280 (500h)	AIF1_BCLK_Ctrl	0	0	0	0	0	0	0	0	AIF1_BCLK_INV	AIF1_BCLK_FRC	AIF1_BCLK_MSTR	AIF1_BCLK_FREQ [4:0]					000Ch
R1281 (501h)	AIF1_Tx_Pin_Ctrl	0	0	0	0	0	0	0	0	0	0	AIF1TX_DAT_TRI	0	0	0	0	0	0000h
R1282 (502h)	AIF1_Rx_Pin_Ctrl	0	0	0	0	0	0	0	0	0	0	0	AIF1_LRCLK_ADV	0	AIF1_LRCLK_INV	AIF1_LRCLK_FRC	AIF1_LRCLK_MSTR	0000h
R1283 (503h)	AIF1_Rate_Ctrl	0	AIF1_RATE [3:0]				0	0	0	0	AIF1_TRI	0	0	0	0	0	0	0000h
R1284 (504h)	AIF1_Format	0	0	0	0	0	0	0	0	0	0	0	0	0	AIF1_FMT [2:0]			0000h
R1286 (506h)	AIF1_Rx_BCLK_Rate	0	0	0	AIF1_BCPF [12:0]													0040h
R1287 (507h)	AIF1_Frame_Ctrl_1	0	0	AIF1TX_WL [5:0]						AIF1TX_SLOT_LEN [7:0]								1818h
R1288 (508h)	AIF1_Frame_Ctrl_2	0	0	AIF1RX_WL [5:0]						AIF1RX_SLOT_LEN [7:0]								1818h
R1289 (509h)	AIF1_Frame_Ctrl_3	0	0	0	0	0	0	0	0	0	0	AIF1TX1_SLOT [5:0]						0000h
R1290 (50Ah)	AIF1_Frame_Ctrl_4	0	0	0	0	0	0	0	0	0	0	AIF1TX2_SLOT [5:0]						0001h
R1291 (50Bh)	AIF1_Frame_Ctrl_5	0	0	0	0	0	0	0	0	0	0	AIF1TX3_SLOT [5:0]						0002h
R1292 (50Ch)	AIF1_Frame_Ctrl_6	0	0	0	0	0	0	0	0	0	0	AIF1TX4_SLOT [5:0]						0003h
R1293 (50Dh)	AIF1_Frame_Ctrl_7	0	0	0	0	0	0	0	0	0	0	AIF1TX5_SLOT [5:0]						0004h
R1294 (50Eh)	AIF1_Frame_Ctrl_8	0	0	0	0	0	0	0	0	0	0	AIF1TX6_SLOT [5:0]						0005h
R1297 (511h)	AIF1_Frame_Ctrl_11	0	0	0	0	0	0	0	0	0	0	AIF1RX1_SLOT [5:0]						0000h
R1298 (512h)	AIF1_Frame_Ctrl_12	0	0	0	0	0	0	0	0	0	0	AIF1RX2_SLOT [5:0]						0001h
R1299 (513h)	AIF1_Frame_Ctrl_13	0	0	0	0	0	0	0	0	0	0	AIF1RX3_SLOT [5:0]						0002h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R1300 (514h)	AIF1_Frame_Ctrl_14	0	0	0	0	0	0	0	0	0	0	AIF1RX4_SLOT [5:0]						0003h
R1301 (515h)	AIF1_Frame_Ctrl_15	0	0	0	0	0	0	0	0	0	0	AIF1RX5_SLOT [5:0]						0004h
R1302 (516h)	AIF1_Frame_Ctrl_16	0	0	0	0	0	0	0	0	0	0	AIF1RX6_SLOT [5:0]						0005h
R1305 (519h)	AIF1_Tx_Enables	0	0	0	0	0	0	0	0	0	0	AIF1TX6_ENA	AIF1TX5_ENA	AIF1TX4_ENA	AIF1TX3_ENA	AIF1TX2_ENA	AIF1TX1_ENA	0000h
R1306 (51Ah)	AIF1_Rx_Enables	0	0	0	0	0	0	0	0	0	0	AIF1RX6_ENA	AIF1RX5_ENA	AIF1RX4_ENA	AIF1RX3_ENA	AIF1RX2_ENA	AIF1RX1_ENA	0000h
R1344 (540h)	AIF2_BCLK_Ctrl	0	0	0	0	0	0	0	0	AIF2_BCLK_INV	AIF2_BCLK_FRC	AIF2_BCLK_MSTR	AIF2_BCLK_FREQ [4:0]					000Ch
R1345 (541h)	AIF2_Tx_Pin_Ctrl	0	0	0	0	0	0	0	0	0	0	AIF2TX_DAT_TRI	0	0	0	0	0	0000h
R1346 (542h)	AIF2_Rx_Pin_Ctrl	0	0	0	0	0	0	0	0	0	0	0	AIF2_LRCLK_ADV	0	AIF2_LRCLK_INV	AIF2_LRCLK_FRC	AIF2_LRCLK_MSTR	0000h
R1347 (543h)	AIF2_Rate_Ctrl	0	AIF2_RATE [3:0]				0	0	0	0	AIF2_TRI	0	0	0	0	0	0	0000h
R1348 (544h)	AIF2_Format	0	0	0	0	0	0	0	0	0	0	0	0	0	AIF2_FMT [2:0]			0000h
R1350 (546h)	AIF2_Rx_BCLK_Rate	0	0	0	AIF2_BCPF [12:0]													0040h
R1351 (547h)	AIF2_Frame_Ctrl_1	0	0	AIF2TX_WL [5:0]						AIF2TX_SLOT_LEN [7:0]								1818h
R1352 (548h)	AIF2_Frame_Ctrl_2	0	0	AIF2RX_WL [5:0]						AIF2RX_SLOT_LEN [7:0]								1818h
R1353 (549h)	AIF2_Frame_Ctrl_3	0	0	0	0	0	0	0	0	0	0	AIF2TX1_SLOT [5:0]						0000h
R1354 (54Ah)	AIF2_Frame_Ctrl_4	0	0	0	0	0	0	0	0	0	0	AIF2TX2_SLOT [5:0]						0001h
R1355 (54Bh)	AIF2_Frame_Ctrl_5	0	0	0	0	0	0	0	0	0	0	AIF2TX3_SLOT [5:0]						0002h
R1356 (54Ch)	AIF2_Frame_Ctrl_6	0	0	0	0	0	0	0	0	0	0	AIF2TX4_SLOT [5:0]						0003h
R1361 (551h)	AIF2_Frame_Ctrl_11	0	0	0	0	0	0	0	0	0	0	AIF2RX1_SLOT [5:0]						0000h
R1362 (552h)	AIF2_Frame_Ctrl_12	0	0	0	0	0	0	0	0	0	0	AIF2RX2_SLOT [5:0]						0001h
R1363 (553h)	AIF2_Frame_Ctrl_13	0	0	0	0	0	0	0	0	0	0	AIF2RX3_SLOT [5:0]						0002h
R1364 (554h)	AIF2_Frame_Ctrl_14	0	0	0	0	0	0	0	0	0	0	AIF2RX4_SLOT [5:0]						0003h
R1369 (559h)	AIF2_Tx_Enables	0	0	0	0	0	0	0	0	0	0	0	0	AIF2TX4_ENA	AIF2TX3_ENA	AIF2TX2_ENA	AIF2TX1_ENA	0000h
R1370 (55Ah)	AIF2_Rx_Enables	0	0	0	0	0	0	0	0	0	0	0	0	AIF2RX4_ENA	AIF2RX3_ENA	AIF2RX2_ENA	AIF2RX1_ENA	0000h
R1408 (580h)	AIF3_BCLK_Ctrl	0	0	0	0	0	0	0	0	AIF3_BCLK_INV	AIF3_BCLK_FRC	AIF3_BCLK_MSTR	AIF3_BCLK_FREQ [4:0]					000Ch
R1409 (581h)	AIF3_Tx_Pin_Ctrl	0	0	0	0	0	0	0	0	0	0	AIF3TX_DAT_TRI	0	0	0	0	0	0000h
R1410 (582h)	AIF3_Rx_Pin_Ctrl	0	0	0	0	0	0	0	0	0	0	0	AIF3_LRCLK_ADV	0	AIF3_LRCLK_INV	AIF3_LRCLK_FRC	AIF3_LRCLK_MSTR	0000h
R1411 (583h)	AIF3_Rate_Ctrl	0	AIF3_RATE [3:0]				0	0	0	0	AIF3_TRI	0	0	0	0	0	0	0000h
R1412 (584h)	AIF3_Format	0	0	0	0	0	0	0	0	0	0	0	0	0	AIF3_FMT [2:0]			0000h
R1414 (586h)	AIF3_Rx_BCLK_Rate	0	0	0	AIF3_BCPF [12:0]													0040h
R1415 (587h)	AIF3_Frame_Ctrl_1	0	0	AIF3TX_WL [5:0]						AIF3TX_SLOT_LEN [7:0]								1818h
R1416 (588h)	AIF3_Frame_Ctrl_2	0	0	AIF3RX_WL [5:0]						AIF3RX_SLOT_LEN [7:0]								1818h
R1417 (589h)	AIF3_Frame_Ctrl_3	0	0	0	0	0	0	0	0	0	0	AIF3TX1_SLOT [5:0]						0000h
R1418 (58Ah)	AIF3_Frame_Ctrl_4	0	0	0	0	0	0	0	0	0	0	AIF3TX2_SLOT [5:0]						0001h
R1425 (591h)	AIF3_Frame_Ctrl_11	0	0	0	0	0	0	0	0	0	0	AIF3RX1_SLOT [5:0]						0000h
R1426 (592h)	AIF3_Frame_Ctrl_12	0	0	0	0	0	0	0	0	0	0	AIF3RX2_SLOT [5:0]						0001h
R1433 (599h)	AIF3_Tx_Enables	0	0	0	0	0	0	0	0	0	0	0	0	0	0	AIF3TX2_ENA	AIF3TX1_ENA	0000h
R1434 (59Ah)	AIF3_Rx_Enables	0	0	0	0	0	0	0	0	0	0	0	0	0	0	AIF3RX2_ENA	AIF3RX1_ENA	0000h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R1474 (5C2h)	SPD1_TX_Control	0	0	SPD1_VAL2	SPD1_VAL1	0	0	0	0	SPD1_RATE [3:0]				0	0	0	SPD1_ENA	0000h
R1475 (5C3h)	SPD1_TX_Channel_Status_1	SPD1_CATCODE [7:0]								SPD1_CHSTMODE [1:0]		SPD1_PREEMPH [2:0]			SPD1_NOCOPY	SPD1_NOAUDIO	SPD1_PRO	0000h
R1476 (5C4h)	SPD1_TX_Channel_Status_2	SPD1_FREQ [3:0]				SPD1_CHNUM2 [3:0]				SPD1_CHNUM1 [3:0]				SPD1_SRCNUM [3:0]				0001h
R1477 (5C5h)	SPD1_TX_Channel_Status_3	0	0	0	0	SPD1_ORGSAMP [3:0]				SPD1_TXWL [2:0]			SPD1_MAXWL	SPD1_CS31_30 [1:0]		SPD1_CLKACU [1:0]		0000h
R1600 (640h)	PWM1MIX_Input_1_Source	PWM1MIX_STS1	0	0	0	0	0	0	0	PWM1MIX_SRC1 [7:0]								0000h
R1601 (641h)	PWM1MIX_Input_1_Volume	0	0	0	0	0	0	0	0	PWM1MIX_VOL1 [6:0]							0	0080h
R1602 (642h)	PWM1MIX_Input_2_Source	PWM1MIX_STS2	0	0	0	0	0	0	0	PWM1MIX_SRC2 [7:0]								0000h
R1603 (643h)	PWM1MIX_Input_2_Volume	0	0	0	0	0	0	0	0	PWM1MIX_VOL2 [6:0]							0	0080h
R1604 (644h)	PWM1MIX_Input_3_Source	PWM1MIX_STS3	0	0	0	0	0	0	0	PWM1MIX_SRC3 [7:0]								0000h
R1605 (645h)	PWM1MIX_Input_3_Volume	0	0	0	0	0	0	0	0	PWM1MIX_VOL3 [6:0]							0	0080h
R1606 (646h)	PWM1MIX_Input_4_Source	PWM1MIX_STS4	0	0	0	0	0	0	0	PWM1MIX_SRC4 [7:0]								0000h
R1607 (647h)	PWM1MIX_Input_4_Volume	0	0	0	0	0	0	0	0	PWM1MIX_VOL4 [6:0]							0	0080h
R1608 (648h)	PWM2MIX_Input_1_Source	PWM2MIX_STS1	0	0	0	0	0	0	0	PWM2MIX_SRC1 [7:0]								0000h
R1609 (649h)	PWM2MIX_Input_1_Volume	0	0	0	0	0	0	0	0	PWM2MIX_VOL1 [6:0]							0	0080h
R1610 (64Ah)	PWM2MIX_Input_2_Source	PWM2MIX_STS2	0	0	0	0	0	0	0	PWM2MIX_SRC2 [7:0]								0000h
R1611 (64Bh)	PWM2MIX_Input_2_Volume	0	0	0	0	0	0	0	0	PWM2MIX_VOL2 [6:0]							0	0080h
R1612 (64Ch)	PWM2MIX_Input_3_Source	PWM2MIX_STS3	0	0	0	0	0	0	0	PWM2MIX_SRC3 [7:0]								0000h
R1613 (64Dh)	PWM2MIX_Input_3_Volume	0	0	0	0	0	0	0	0	PWM2MIX_VOL3 [6:0]							0	0080h
R1614 (64Eh)	PWM2MIX_Input_4_Source	PWM2MIX_STS4	0	0	0	0	0	0	0	PWM2MIX_SRC4 [7:0]								0000h
R1615 (64Fh)	PWM2MIX_Input_4_Volume	0	0	0	0	0	0	0	0	PWM2MIX_VOL4 [6:0]							0	0080h
R1664 (680h)	OUT1LMIX_Input_1_Source	OUT1LMIX_STS1	0	0	0	0	0	0	0	OUT1LMIX_SRC1 [7:0]								0000h
R1665 (681h)	OUT1LMIX_Input_1_Volume	0	0	0	0	0	0	0	0	OUT1LMIX_VOL1 [6:0]							0	0080h
R1666 (682h)	OUT1LMIX_Input_2_Source	OUT1LMIX_STS2	0	0	0	0	0	0	0	OUT1LMIX_SRC2 [7:0]								0000h
R1667 (683h)	OUT1LMIX_Input_2_Volume	0	0	0	0	0	0	0	0	OUT1LMIX_VOL2 [6:0]							0	0080h
R1668 (684h)	OUT1LMIX_Input_3_Source	OUT1LMIX_STS3	0	0	0	0	0	0	0	OUT1LMIX_SRC3 [7:0]								0000h
R1669 (685h)	OUT1LMIX_Input_3_Volume	0	0	0	0	0	0	0	0	OUT1LMIX_VOL3 [6:0]							0	0080h
R1670 (686h)	OUT1LMIX_Input_4_Source	OUT1LMIX_STS4	0	0	0	0	0	0	0	OUT1LMIX_SRC4 [7:0]								0000h
R1671 (687h)	OUT1LMIX_Input_4_Volume	0	0	0	0	0	0	0	0	OUT1LMIX_VOL4 [6:0]							0	0080h
R1672 (688h)	OUT1RMIX_Input_1_Source	OUT1RMIX_STS1	0	0	0	0	0	0	0	OUT1RMIX_SRC1 [7:0]								0000h
R1673 (689h)	OUT1RMIX_Input_1_Volume	0	0	0	0	0	0	0	0	OUT1RMIX_VOL1 [6:0]							0	0080h
R1674 (68Ah)	OUT1RMIX_Input_2_Source	OUT1RMIX_STS2	0	0	0	0	0	0	0	OUT1RMIX_SRC2 [7:0]								0000h
R1675 (68Bh)	OUT1RMIX_Input_2_Volume	0	0	0	0	0	0	0	0	OUT1RMIX_VOL2 [6:0]							0	0080h
R1676 (68Ch)	OUT1RMIX_Input_3_Source	OUT1RMIX_STS3	0	0	0	0	0	0	0	OUT1RMIX_SRC3 [7:0]								0000h
R1677 (68Dh)	OUT1RMIX_Input_3_Volume	0	0	0	0	0	0	0	0	OUT1RMIX_VOL3 [6:0]							0	0080h
R1678 (68Eh)	OUT1RMIX_Input_4_Source	OUT1RMIX_STS4	0	0	0	0	0	0	0	OUT1RMIX_SRC4 [7:0]								0000h
R1679 (68Fh)	OUT1RMIX_Input_4_Volume	0	0	0	0	0	0	0	0	OUT1RMIX_VOL4 [6:0]							0	0080h
R1712 (6B0h)	OUT4LMIX_Input_1_Source	OUT4LMIX_STS1	0	0	0	0	0	0	0	OUT4LMIX_SRC1 [7:0]								0000h
R1713 (6B1h)	OUT4LMIX_Input_1_Volume	0	0	0	0	0	0	0	0	OUT4LMIX_VOL1 [6:0]							0	0080h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R1714 (6B2h)	OUT4LMIX_Input_2_Source	OUT4LMIX_STS2	0	0	0	0	0	0	0	OUT4LMIX_SRC2 [7:0]								0000h
R1715 (6B3h)	OUT4LMIX_Input_2_Volume	0	0	0	0	0	0	0	0	OUT4LMIX_VOL2 [6:0]							0	0080h
R1716 (6B4h)	OUT4LMIX_Input_3_Source	OUT4LMIX_STS3	0	0	0	0	0	0	0	OUT4LMIX_SRC3 [7:0]								0000h
R1717 (6B5h)	OUT4LMIX_Input_3_Volume	0	0	0	0	0	0	0	0	OUT4LMIX_VOL3 [6:0]							0	0080h
R1718 (6B6h)	OUT4LMIX_Input_4_Source	OUT4LMIX_STS4	0	0	0	0	0	0	0	OUT4LMIX_SRC4 [7:0]								0000h
R1719 (6B7h)	OUT4LMIX_Input_4_Volume	0	0	0	0	0	0	0	0	OUT4LMIX_VOL4 [6:0]							0	0080h
R1728 (6C0h)	OUT5LMIX_Input_1_Source	OUT5LMIX_STS1	0	0	0	0	0	0	0	OUT5LMIX_SRC1 [7:0]								0000h
R1729 (6C1h)	OUT5LMIX_Input_1_Volume	0	0	0	0	0	0	0	0	OUT5LMIX_VOL1 [6:0]							0	0080h
R1730 (6C2h)	OUT5LMIX_Input_2_Source	OUT5LMIX_STS2	0	0	0	0	0	0	0	OUT5LMIX_SRC2 [7:0]								0000h
R1731 (6C3h)	OUT5LMIX_Input_2_Volume	0	0	0	0	0	0	0	0	OUT5LMIX_VOL2 [6:0]							0	0080h
R1732 (6C4h)	OUT5LMIX_Input_3_Source	OUT5LMIX_STS3	0	0	0	0	0	0	0	OUT5LMIX_SRC3 [7:0]								0000h
R1733 (6C5h)	OUT5LMIX_Input_3_Volume	0	0	0	0	0	0	0	0	OUT5LMIX_VOL3 [6:0]							0	0080h
R1734 (6C6h)	OUT5LMIX_Input_4_Source	OUT5LMIX_STS4	0	0	0	0	0	0	0	OUT5LMIX_SRC4 [7:0]								0000h
R1735 (6C7h)	OUT5LMIX_Input_4_Volume	0	0	0	0	0	0	0	0	OUT5LMIX_VOL4 [6:0]							0	0080h
R1736 (6C8h)	OUT5RMIX_Input_1_Source	OUT5RMIX_STS1	0	0	0	0	0	0	0	OUT5RMIX_SRC1 [7:0]								0000h
R1737 (6C9h)	OUT5RMIX_Input_1_Volume	0	0	0	0	0	0	0	0	OUT5RMIX_VOL1 [6:0]							0	0080h
R1738 (6CAh)	OUT5RMIX_Input_2_Source	OUT5RMIX_STS2	0	0	0	0	0	0	0	OUT5RMIX_SRC2 [7:0]								0000h
R1739 (6CBh)	OUT5RMIX_Input_2_Volume	0	0	0	0	0	0	0	0	OUT5RMIX_VOL2 [6:0]							0	0080h
R1740 (6CCh)	OUT5RMIX_Input_3_Source	OUT5RMIX_STS3	0	0	0	0	0	0	0	OUT5RMIX_SRC3 [7:0]								0000h
R1741 (6CDh)	OUT5RMIX_Input_3_Volume	0	0	0	0	0	0	0	0	OUT5RMIX_VOL3 [6:0]							0	0080h
R1742 (6CEh)	OUT5RMIX_Input_4_Source	OUT5RMIX_STS4	0	0	0	0	0	0	0	OUT5RMIX_SRC4 [7:0]								0000h
R1743 (6CFh)	OUT5RMIX_Input_4_Volume	0	0	0	0	0	0	0	0	OUT5RMIX_VOL4 [6:0]							0	0080h
R1792 (700h)	AIF1TX1MIX_Input_1_Source	AIF1TX1MIX_STS1	0	0	0	0	0	0	0	AIF1TX1MIX_SRC1 [7:0]								0000h
R1793 (701h)	AIF1TX1MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF1TX1MIX_VOL1 [6:0]							0	0080h
R1794 (702h)	AIF1TX1MIX_Input_2_Source	AIF1TX1MIX_STS2	0	0	0	0	0	0	0	AIF1TX1MIX_SRC2 [7:0]								0000h
R1795 (703h)	AIF1TX1MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF1TX1MIX_VOL2 [6:0]							0	0080h
R1796 (704h)	AIF1TX1MIX_Input_3_Source	AIF1TX1MIX_STS3	0	0	0	0	0	0	0	AIF1TX1MIX_SRC3 [7:0]								0000h
R1797 (705h)	AIF1TX1MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF1TX1MIX_VOL3 [6:0]							0	0080h
R1798 (706h)	AIF1TX1MIX_Input_4_Source	AIF1TX1MIX_STS4	0	0	0	0	0	0	0	AIF1TX1MIX_SRC4 [7:0]								0000h
R1799 (707h)	AIF1TX1MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF1TX1MIX_VOL4 [6:0]							0	0080h
R1800 (708h)	AIF1TX2MIX_Input_1_Source	AIF1TX2MIX_STS1	0	0	0	0	0	0	0	AIF1TX2MIX_SRC1 [7:0]								0000h
R1801 (709h)	AIF1TX2MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF1TX2MIX_VOL1 [6:0]							0	0080h
R1802 (70Ah)	AIF1TX2MIX_Input_2_Source	AIF1TX2MIX_STS2	0	0	0	0	0	0	0	AIF1TX2MIX_SRC2 [7:0]								0000h
R1803 (70Bh)	AIF1TX2MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF1TX2MIX_VOL2 [6:0]							0	0080h
R1804 (70Ch)	AIF1TX2MIX_Input_3_Source	AIF1TX2MIX_STS3	0	0	0	0	0	0	0	AIF1TX2MIX_SRC3 [7:0]								0000h
R1805 (70Dh)	AIF1TX2MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF1TX2MIX_VOL3 [6:0]							0	0080h
R1806 (70Eh)	AIF1TX2MIX_Input_4_Source	AIF1TX2MIX_STS4	0	0	0	0	0	0	0	AIF1TX2MIX_SRC4 [7:0]								0000h
R1807 (70Fh)	AIF1TX2MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF1TX2MIX_VOL4 [6:0]							0	0080h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R1808 (710h)	AIF1TX3MIX_Input_1_Source	AIF1TX3MIX_STS1	0	0	0	0	0	0	0	AIF1TX3MIX_SRC1 [7:0]								0000h
R1809 (711h)	AIF1TX3MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF1TX3MIX_VOL1 [6:0]							0	0080h
R1810 (712h)	AIF1TX3MIX_Input_2_Source	AIF1TX3MIX_STS2	0	0	0	0	0	0	0	AIF1TX3MIX_SRC2 [7:0]								0000h
R1811 (713h)	AIF1TX3MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF1TX3MIX_VOL2 [6:0]							0	0080h
R1812 (714h)	AIF1TX3MIX_Input_3_Source	AIF1TX3MIX_STS3	0	0	0	0	0	0	0	AIF1TX3MIX_SRC3 [7:0]								0000h
R1813 (715h)	AIF1TX3MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF1TX3MIX_VOL3 [6:0]							0	0080h
R1814 (716h)	AIF1TX3MIX_Input_4_Source	AIF1TX3MIX_STS4	0	0	0	0	0	0	0	AIF1TX3MIX_SRC4 [7:0]								0000h
R1815 (717h)	AIF1TX3MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF1TX3MIX_VOL4 [6:0]							0	0080h
R1816 (718h)	AIF1TX4MIX_Input_1_Source	AIF1TX4MIX_STS1	0	0	0	0	0	0	0	AIF1TX4MIX_SRC1 [7:0]								0000h
R1817 (719h)	AIF1TX4MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF1TX4MIX_VOL1 [6:0]							0	0080h
R1818 (71Ah)	AIF1TX4MIX_Input_2_Source	AIF1TX4MIX_STS2	0	0	0	0	0	0	0	AIF1TX4MIX_SRC2 [7:0]								0000h
R1819 (71Bh)	AIF1TX4MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF1TX4MIX_VOL2 [6:0]							0	0080h
R1820 (71Ch)	AIF1TX4MIX_Input_3_Source	AIF1TX4MIX_STS3	0	0	0	0	0	0	0	AIF1TX4MIX_SRC3 [7:0]								0000h
R1821 (71Dh)	AIF1TX4MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF1TX4MIX_VOL3 [6:0]							0	0080h
R1822 (71Eh)	AIF1TX4MIX_Input_4_Source	AIF1TX4MIX_STS4	0	0	0	0	0	0	0	AIF1TX4MIX_SRC4 [7:0]								0000h
R1823 (71Fh)	AIF1TX4MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF1TX4MIX_VOL4 [6:0]							0	0080h
R1824 (720h)	AIF1TX5MIX_Input_1_Source	AIF1TX5MIX_STS1	0	0	0	0	0	0	0	AIF1TX5MIX_SRC1 [7:0]								0000h
R1825 (721h)	AIF1TX5MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF1TX5MIX_VOL1 [6:0]							0	0080h
R1826 (722h)	AIF1TX5MIX_Input_2_Source	AIF1TX5MIX_STS2	0	0	0	0	0	0	0	AIF1TX5MIX_SRC2 [7:0]								0000h
R1827 (723h)	AIF1TX5MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF1TX5MIX_VOL2 [6:0]							0	0080h
R1828 (724h)	AIF1TX5MIX_Input_3_Source	AIF1TX5MIX_STS3	0	0	0	0	0	0	0	AIF1TX5MIX_SRC3 [7:0]								0000h
R1829 (725h)	AIF1TX5MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF1TX5MIX_VOL3 [6:0]							0	0080h
R1830 (726h)	AIF1TX5MIX_Input_4_Source	AIF1TX5MIX_STS4	0	0	0	0	0	0	0	AIF1TX5MIX_SRC4 [7:0]								0000h
R1831 (727h)	AIF1TX5MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF1TX5MIX_VOL4 [6:0]							0	0080h
R1832 (728h)	AIF1TX6MIX_Input_1_Source	AIF1TX6MIX_STS1	0	0	0	0	0	0	0	AIF1TX6MIX_SRC1 [7:0]								0000h
R1833 (729h)	AIF1TX6MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF1TX6MIX_VOL1 [6:0]							0	0080h
R1834 (72Ah)	AIF1TX6MIX_Input_2_Source	AIF1TX6MIX_STS2	0	0	0	0	0	0	0	AIF1TX6MIX_SRC2 [7:0]								0000h
R1835 (72Bh)	AIF1TX6MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF1TX6MIX_VOL2 [6:0]							0	0080h
R1836 (72Ch)	AIF1TX6MIX_Input_3_Source	AIF1TX6MIX_STS3	0	0	0	0	0	0	0	AIF1TX6MIX_SRC3 [7:0]								0000h
R1837 (72Dh)	AIF1TX6MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF1TX6MIX_VOL3 [6:0]							0	0080h
R1838 (72Eh)	AIF1TX6MIX_Input_4_Source	AIF1TX6MIX_STS4	0	0	0	0	0	0	0	AIF1TX6MIX_SRC4 [7:0]								0000h
R1839 (72Fh)	AIF1TX6MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF1TX6MIX_VOL4 [6:0]							0	0080h
R1856 (740h)	AIF2TX1MIX_Input_1_Source	AIF2TX1MIX_STS1	0	0	0	0	0	0	0	AIF2TX1MIX_SRC1 [7:0]								0000h
R1857 (741h)	AIF2TX1MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF2TX1MIX_VOL1 [6:0]							0	0080h
R1858 (742h)	AIF2TX1MIX_Input_2_Source	AIF2TX1MIX_STS2	0	0	0	0	0	0	0	AIF2TX1MIX_SRC2 [7:0]								0000h
R1859 (743h)	AIF2TX1MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF2TX1MIX_VOL2 [6:0]							0	0080h
R1860 (744h)	AIF2TX1MIX_Input_3_Source	AIF2TX1MIX_STS3	0	0	0	0	0	0	0	AIF2TX1MIX_SRC3 [7:0]								0000h
R1861 (745h)	AIF2TX1MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF2TX1MIX_VOL3 [6:0]							0	0080h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R1862 (746h)	AIF2TX1MIX_Input_4_Source	AIF2TX1MIX_STS4	0	0	0	0	0	0	0	AIF2TX1MIX_SRC4 [7:0]								0000h
R1863 (747h)	AIF2TX1MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF2TX1MIX_VOL4 [6:0]							0	0080h
R1864 (748h)	AIF2TX2MIX_Input_1_Source	AIF2TX2MIX_STS1	0	0	0	0	0	0	0	AIF2TX2MIX_SRC1 [7:0]								0000h
R1865 (749h)	AIF2TX2MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF2TX2MIX_VOL1 [6:0]							0	0080h
R1866 (74Ah)	AIF2TX2MIX_Input_2_Source	AIF2TX2MIX_STS2	0	0	0	0	0	0	0	AIF2TX2MIX_SRC2 [7:0]								0000h
R1867 (74Bh)	AIF2TX2MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF2TX2MIX_VOL2 [6:0]							0	0080h
R1868 (74Ch)	AIF2TX2MIX_Input_3_Source	AIF2TX2MIX_STS3	0	0	0	0	0	0	0	AIF2TX2MIX_SRC3 [7:0]								0000h
R1869 (74Dh)	AIF2TX2MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF2TX2MIX_VOL3 [6:0]							0	0080h
R1870 (74Eh)	AIF2TX2MIX_Input_4_Source	AIF2TX2MIX_STS4	0	0	0	0	0	0	0	AIF2TX2MIX_SRC4 [7:0]								0000h
R1871 (74Fh)	AIF2TX2MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF2TX2MIX_VOL4 [6:0]							0	0080h
R1872 (750h)	AIF2TX3MIX_Input_1_Source	AIF2TX3MIX_STS1	0	0	0	0	0	0	0	AIF2TX3MIX_SRC1 [7:0]								0000h
R1873 (751h)	AIF2TX3MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF2TX3MIX_VOL1 [6:0]							0	0080h
R1874 (752h)	AIF2TX3MIX_Input_2_Source	AIF2TX3MIX_STS2	0	0	0	0	0	0	0	AIF2TX3MIX_SRC2 [7:0]								0000h
R1875 (753h)	AIF2TX3MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF2TX3MIX_VOL2 [6:0]							0	0080h
R1876 (754h)	AIF2TX3MIX_Input_3_Source	AIF2TX3MIX_STS3	0	0	0	0	0	0	0	AIF2TX3MIX_SRC3 [7:0]								0000h
R1877 (755h)	AIF2TX3MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF2TX3MIX_VOL3 [6:0]							0	0080h
R1878 (756h)	AIF2TX3MIX_Input_4_Source	AIF2TX3MIX_STS4	0	0	0	0	0	0	0	AIF2TX3MIX_SRC4 [7:0]								0000h
R1879 (757h)	AIF2TX3MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF2TX3MIX_VOL4 [6:0]							0	0080h
R1880 (758h)	AIF2TX4MIX_Input_1_Source	AIF2TX4MIX_STS1	0	0	0	0	0	0	0	AIF2TX4MIX_SRC1 [7:0]								0000h
R1881 (759h)	AIF2TX4MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF2TX4MIX_VOL1 [6:0]							0	0080h
R1882 (75Ah)	AIF2TX4MIX_Input_2_Source	AIF2TX4MIX_STS2	0	0	0	0	0	0	0	AIF2TX4MIX_SRC2 [7:0]								0000h
R1883 (75Bh)	AIF2TX4MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF2TX4MIX_VOL2 [6:0]							0	0080h
R1884 (75Ch)	AIF2TX4MIX_Input_3_Source	AIF2TX4MIX_STS3	0	0	0	0	0	0	0	AIF2TX4MIX_SRC3 [7:0]								0000h
R1885 (75Dh)	AIF2TX4MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF2TX4MIX_VOL3 [6:0]							0	0080h
R1886 (75Eh)	AIF2TX4MIX_Input_4_Source	AIF2TX4MIX_STS4	0	0	0	0	0	0	0	AIF2TX4MIX_SRC4 [7:0]								0000h
R1887 (75Fh)	AIF2TX4MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF2TX4MIX_VOL4 [6:0]							0	0080h
R1920 (780h)	AIF3TX1MIX_Input_1_Source	AIF3TX1MIX_STS1	0	0	0	0	0	0	0	AIF3TX1MIX_SRC1 [7:0]								0000h
R1921 (781h)	AIF3TX1MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF3TX1MIX_VOL1 [6:0]							0	0080h
R1922 (782h)	AIF3TX1MIX_Input_2_Source	AIF3TX1MIX_STS2	0	0	0	0	0	0	0	AIF3TX1MIX_SRC2 [7:0]								0000h
R1923 (783h)	AIF3TX1MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF3TX1MIX_VOL2 [6:0]							0	0080h
R1924 (784h)	AIF3TX1MIX_Input_3_Source	AIF3TX1MIX_STS3	0	0	0	0	0	0	0	AIF3TX1MIX_SRC3 [7:0]								0000h
R1925 (785h)	AIF3TX1MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF3TX1MIX_VOL3 [6:0]							0	0080h
R1926 (786h)	AIF3TX1MIX_Input_4_Source	AIF3TX1MIX_STS4	0	0	0	0	0	0	0	AIF3TX1MIX_SRC4 [7:0]								0000h
R1927 (787h)	AIF3TX1MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF3TX1MIX_VOL4 [6:0]							0	0080h
R1928 (788h)	AIF3TX2MIX_Input_1_Source	AIF3TX2MIX_STS1	0	0	0	0	0	0	0	AIF3TX2MIX_SRC1 [7:0]								0000h
R1929 (789h)	AIF3TX2MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF3TX2MIX_VOL1 [6:0]							0	0080h
R1930 (78Ah)	AIF3TX2MIX_Input_2_Source	AIF3TX2MIX_STS2	0	0	0	0	0	0	0	AIF3TX2MIX_SRC2 [7:0]								0000h
R1931 (78Bh)	AIF3TX2MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF3TX2MIX_VOL2 [6:0]							0	0080h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R1932 (78Ch)	AIF3TX2MIX_Input_3_Source	AIF3TX2MIX_STS3	0	0	0	0	0	0	0	AIF3TX2MIX_SRC3 [7:0]								0000h
R1933 (78Dh)	AIF3TX2MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF3TX2MIX_VOL3 [6:0]							0	0080h
R1934 (78Eh)	AIF3TX2MIX_Input_4_Source	AIF3TX2MIX_STS4	0	0	0	0	0	0	0	AIF3TX2MIX_SRC4 [7:0]								0000h
R1935 (78Fh)	AIF3TX2MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF3TX2MIX_VOL4 [6:0]							0	0080h
R2048 (800h)	SPDIF1TX1MIX_Input_1_Source	SPDIF1TX1_STS	0	0	0	0	0	0	0	SPDIF1TX1_SRC [7:0]								0000h
R2049 (801h)	SPDIF1TX1MIX_Input_1_Volume	0	0	0	0	0	0	0	0	SPDIF1TX1_VOL [6:0]							0	0080h
R2056 (808h)	SPDIF1TX2MIX_Input_1_Source	SPDIF1TX2_STS	0	0	0	0	0	0	0	SPDIF1TX2_SRC [7:0]								0000h
R2057 (809h)	SPDIF1TX2MIX_Input_1_Volume	0	0	0	0	0	0	0	0	SPDIF1TX2_VOL [6:0]							0	0080h
R2176 (880h)	EQ1MIX_Input_1_Source	EQ1MIX_STS1	0	0	0	0	0	0	0	EQ1MIX_SRC1 [7:0]								0000h
R2177 (881h)	EQ1MIX_Input_1_Volume	0	0	0	0	0	0	0	0	EQ1MIX_VOL1 [6:0]							0	0080h
R2178 (882h)	EQ1MIX_Input_2_Source	EQ1MIX_STS2	0	0	0	0	0	0	0	EQ1MIX_SRC2 [7:0]								0000h
R2179 (883h)	EQ1MIX_Input_2_Volume	0	0	0	0	0	0	0	0	EQ1MIX_VOL2 [6:0]							0	0080h
R2180 (884h)	EQ1MIX_Input_3_Source	EQ1MIX_STS3	0	0	0	0	0	0	0	EQ1MIX_SRC3 [7:0]								0000h
R2181 (885h)	EQ1MIX_Input_3_Volume	0	0	0	0	0	0	0	0	EQ1MIX_VOL3 [6:0]							0	0080h
R2182 (886h)	EQ1MIX_Input_4_Source	EQ1MIX_STS4	0	0	0	0	0	0	0	EQ1MIX_SRC4 [7:0]								0000h
R2183 (887h)	EQ1MIX_Input_4_Volume	0	0	0	0	0	0	0	0	EQ1MIX_VOL4 [6:0]							0	0080h
R2184 (888h)	EQ2MIX_Input_1_Source	EQ2MIX_STS1	0	0	0	0	0	0	0	EQ2MIX_SRC1 [7:0]								0000h
R2185 (889h)	EQ2MIX_Input_1_Volume	0	0	0	0	0	0	0	0	EQ2MIX_VOL1 [6:0]							0	0080h
R2186 (88Ah)	EQ2MIX_Input_2_Source	EQ2MIX_STS2	0	0	0	0	0	0	0	EQ2MIX_SRC2 [7:0]								0000h
R2187 (88Bh)	EQ2MIX_Input_2_Volume	0	0	0	0	0	0	0	0	EQ2MIX_VOL2 [6:0]							0	0080h
R2188 (88Ch)	EQ2MIX_Input_3_Source	EQ2MIX_STS3	0	0	0	0	0	0	0	EQ2MIX_SRC3 [7:0]								0000h
R2189 (88Dh)	EQ2MIX_Input_3_Volume	0	0	0	0	0	0	0	0	EQ2MIX_VOL3 [6:0]							0	0080h
R2190 (88Eh)	EQ2MIX_Input_4_Source	EQ2MIX_STS4	0	0	0	0	0	0	0	EQ2MIX_SRC4 [7:0]								0000h
R2191 (88Fh)	EQ2MIX_Input_4_Volume	0	0	0	0	0	0	0	0	EQ2MIX_VOL4 [6:0]							0	0080h
R2192 (890h)	EQ3MIX_Input_1_Source	EQ3MIX_STS1	0	0	0	0	0	0	0	EQ3MIX_SRC1 [7:0]								0000h
R2193 (891h)	EQ3MIX_Input_1_Volume	0	0	0	0	0	0	0	0	EQ3MIX_VOL1 [6:0]							0	0080h
R2194 (892h)	EQ3MIX_Input_2_Source	EQ3MIX_STS2	0	0	0	0	0	0	0	EQ3MIX_SRC2 [7:0]								0000h
R2195 (893h)	EQ3MIX_Input_2_Volume	0	0	0	0	0	0	0	0	EQ3MIX_VOL2 [6:0]							0	0080h
R2196 (894h)	EQ3MIX_Input_3_Source	EQ3MIX_STS3	0	0	0	0	0	0	0	EQ3MIX_SRC3 [7:0]								0000h
R2197 (895h)	EQ3MIX_Input_3_Volume	0	0	0	0	0	0	0	0	EQ3MIX_VOL3 [6:0]							0	0080h
R2198 (896h)	EQ3MIX_Input_4_Source	EQ3MIX_STS4	0	0	0	0	0	0	0	EQ3MIX_SRC4 [7:0]								0000h
R2199 (897h)	EQ3MIX_Input_4_Volume	0	0	0	0	0	0	0	0	EQ3MIX_VOL4 [6:0]							0	0080h
R2200 (898h)	EQ4MIX_Input_1_Source	EQ4MIX_STS1	0	0	0	0	0	0	0	EQ4MIX_SRC1 [7:0]								0000h
R2201 (899h)	EQ4MIX_Input_1_Volume	0	0	0	0	0	0	0	0	EQ4MIX_VOL1 [6:0]							0	0080h
R2202 (89Ah)	EQ4MIX_Input_2_Source	EQ4MIX_STS2	0	0	0	0	0	0	0	EQ4MIX_SRC2 [7:0]								0000h
R2203 (89Bh)	EQ4MIX_Input_2_Volume	0	0	0	0	0	0	0	0	EQ4MIX_VOL2 [6:0]							0	0080h
R2204 (89Ch)	EQ4MIX_Input_3_Source	EQ4MIX_STS3	0	0	0	0	0	0	0	EQ4MIX_SRC3 [7:0]								0000h
R2205 (89Dh)	EQ4MIX_Input_3_Volume	0	0	0	0	0	0	0	0	EQ4MIX_VOL3 [6:0]							0	0080h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R2206 (89Eh)	EQ4MIX_Input_4_Source	EQ4MIX_STS4	0	0	0	0	0	0	0	EQ4MIX_SRC4 [7:0]								0000h
R2207 (89Fh)	EQ4MIX_Input_4_Volume	0	0	0	0	0	0	0	0	EQ4MIX_VOL4 [6:0]							0	0080h
R2240 (8C0h)	DRC1LMIX_Input_1_Source	DRC1LMIX_STS1	0	0	0	0	0	0	0	DRC1LMIX_SRC1 [7:0]								0000h
R2241 (8C1h)	DRC1LMIX_Input_1_Volume	0	0	0	0	0	0	0	0	DRC1LMIX_VOL1 [6:0]							0	0080h
R2242 (8C2h)	DRC1LMIX_Input_2_Source	DRC1LMIX_STS2	0	0	0	0	0	0	0	DRC1LMIX_SRC2 [7:0]								0000h
R2243 (8C3h)	DRC1LMIX_Input_2_Volume	0	0	0	0	0	0	0	0	DRC1LMIX_VOL2 [6:0]							0	0080h
R2244 (8C4h)	DRC1LMIX_Input_3_Source	DRC1LMIX_STS3	0	0	0	0	0	0	0	DRC1LMIX_SRC3 [7:0]								0000h
R2245 (8C5h)	DRC1LMIX_Input_3_Volume	0	0	0	0	0	0	0	0	DRC1LMIX_VOL3 [6:0]							0	0080h
R2246 (8C6h)	DRC1LMIX_Input_4_Source	DRC1LMIX_STS4	0	0	0	0	0	0	0	DRC1LMIX_SRC4 [7:0]								0000h
R2247 (8C7h)	DRC1LMIX_Input_4_Volume	0	0	0	0	0	0	0	0	DRC1LMIX_VOL4 [6:0]							0	0080h
R2248 (8C8h)	DRC1RMIX_Input_1_Source	DRC1RMIX_STS1	0	0	0	0	0	0	0	DRC1RMIX_SRC1 [7:0]								0000h
R2249 (8C9h)	DRC1RMIX_Input_1_Volume	0	0	0	0	0	0	0	0	DRC1RMIX_VOL1 [6:0]							0	0080h
R2250 (8CAh)	DRC1RMIX_Input_2_Source	DRC1RMIX_STS2	0	0	0	0	0	0	0	DRC1RMIX_SRC2 [7:0]								0000h
R2251 (8CBh)	DRC1RMIX_Input_2_Volume	0	0	0	0	0	0	0	0	DRC1RMIX_VOL2 [6:0]							0	0080h
R2252 (8CCh)	DRC1RMIX_Input_3_Source	DRC1RMIX_STS3	0	0	0	0	0	0	0	DRC1RMIX_SRC3 [7:0]								0000h
R2253 (8CDh)	DRC1RMIX_Input_3_Volume	0	0	0	0	0	0	0	0	DRC1RMIX_VOL3 [6:0]							0	0080h
R2254 (8CEh)	DRC1RMIX_Input_4_Source	DRC1RMIX_STS4	0	0	0	0	0	0	0	DRC1RMIX_SRC4 [7:0]								0000h
R2255 (8CFh)	DRC1RMIX_Input_4_Volume	0	0	0	0	0	0	0	0	DRC1RMIX_VOL4 [6:0]							0	0080h
R2256 (8D0h)	DRC2LMIX_Input_1_Source	DRC2LMIX_STS1	0	0	0	0	0	0	0	DRC2LMIX_SRC1 [7:0]								0000h
R2257 (8D1h)	DRC2LMIX_Input_1_Volume	0	0	0	0	0	0	0	0	DRC2LMIX_VOL1 [6:0]							0	0080h
R2258 (8D2h)	DRC2LMIX_Input_2_Source	DRC2LMIX_STS2	0	0	0	0	0	0	0	DRC2LMIX_SRC2 [7:0]								0000h
R2259 (8D3h)	DRC2LMIX_Input_2_Volume	0	0	0	0	0	0	0	0	DRC2LMIX_VOL2 [6:0]							0	0080h
R2260 (8D4h)	DRC2LMIX_Input_3_Source	DRC2LMIX_STS3	0	0	0	0	0	0	0	DRC2LMIX_SRC3 [7:0]								0000h
R2261 (8D5h)	DRC2LMIX_Input_3_Volume	0	0	0	0	0	0	0	0	DRC2LMIX_VOL3 [6:0]							0	0080h
R2262 (8D6h)	DRC2LMIX_Input_4_Source	DRC2LMIX_STS4	0	0	0	0	0	0	0	DRC2LMIX_SRC4 [7:0]								0000h
R2263 (8D7h)	DRC2LMIX_Input_4_Volume	0	0	0	0	0	0	0	0	DRC2LMIX_VOL4 [6:0]							0	0080h
R2264 (8D8h)	DRC2RMIX_Input_1_Source	DRC2RMIX_STS1	0	0	0	0	0	0	0	DRC2RMIX_SRC1 [7:0]								0000h
R2265 (8D9h)	DRC2RMIX_Input_1_Volume	0	0	0	0	0	0	0	0	DRC2RMIX_VOL1 [6:0]							0	0080h
R2266 (8DAh)	DRC2RMIX_Input_2_Source	DRC2RMIX_STS2	0	0	0	0	0	0	0	DRC2RMIX_SRC2 [7:0]								0000h
R2267 (8DBh)	DRC2RMIX_Input_2_Volume	0	0	0	0	0	0	0	0	DRC2RMIX_VOL2 [6:0]							0	0080h
R2268 (8DCh)	DRC2RMIX_Input_3_Source	DRC2RMIX_STS3	0	0	0	0	0	0	0	DRC2RMIX_SRC3 [7:0]								0000h
R2269 (8DDh)	DRC2RMIX_Input_3_Volume	0	0	0	0	0	0	0	0	DRC2RMIX_VOL3 [6:0]							0	0080h
R2270 (8DEh)	DRC2RMIX_Input_4_Source	DRC2RMIX_STS4	0	0	0	0	0	0	0	DRC2RMIX_SRC4 [7:0]								0000h
R2271 (8DFh)	DRC2RMIX_Input_4_Volume	0	0	0	0	0	0	0	0	DRC2RMIX_VOL4 [6:0]							0	0080h
R2304 (900h)	HPLP1MIX_Input_1_Source	LHPF1MIX_STS1	0	0	0	0	0	0	0	LHPF1MIX_SRC1 [7:0]								0000h
R2305 (901h)	HPLP1MIX_Input_1_Volume	0	0	0	0	0	0	0	0	LHPF1MIX_VOL1 [6:0]							0	0080h
R2306 (902h)	HPLP1MIX_Input_2_Source	LHPF1MIX_STS2	0	0	0	0	0	0	0	LHPF1MIX_SRC2 [7:0]								0000h
R2307 (903h)	HPLP1MIX_Input_2_Volume	0	0	0	0	0	0	0	0	LHPF1MIX_VOL2 [6:0]							0	0080h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R2308 (904h)	HPLP1MIX_Input_3_Source	LHPF1MIX_STS3	0	0	0	0	0	0	0	LHPF1MIX_SRC3 [7:0]								0000h
R2309 (905h)	HPLP1MIX_Input_3_Volume	0	0	0	0	0	0	0	0	LHPF1MIX_VOL3 [6:0]							0	0080h
R2310 (906h)	HPLP1MIX_Input_4_Source	LHPF1MIX_STS4	0	0	0	0	0	0	0	LHPF1MIX_SRC4 [7:0]								0000h
R2311 (907h)	HPLP1MIX_Input_4_Volume	0	0	0	0	0	0	0	0	LHPF1MIX_VOL4 [6:0]							0	0080h
R2312 (908h)	HPLP2MIX_Input_1_Source	LHPF2MIX_STS1	0	0	0	0	0	0	0	LHPF2MIX_SRC1 [7:0]								0000h
R2313 (909h)	HPLP2MIX_Input_1_Volume	0	0	0	0	0	0	0	0	LHPF2MIX_VOL1 [6:0]							0	0080h
R2314 (90Ah)	HPLP2MIX_Input_2_Source	LHPF2MIX_STS2	0	0	0	0	0	0	0	LHPF2MIX_SRC2 [7:0]								0000h
R2315 (90Bh)	HPLP2MIX_Input_2_Volume	0	0	0	0	0	0	0	0	LHPF2MIX_VOL2 [6:0]							0	0080h
R2316 (90Ch)	HPLP2MIX_Input_3_Source	LHPF2MIX_STS3	0	0	0	0	0	0	0	LHPF2MIX_SRC3 [7:0]								0000h
R2317 (90Dh)	HPLP2MIX_Input_3_Volume	0	0	0	0	0	0	0	0	LHPF2MIX_VOL3 [6:0]							0	0080h
R2318 (90Eh)	HPLP2MIX_Input_4_Source	LHPF2MIX_STS4	0	0	0	0	0	0	0	LHPF2MIX_SRC4 [7:0]								0000h
R2319 (90Fh)	HPLP2MIX_Input_4_Volume	0	0	0	0	0	0	0	0	LHPF2MIX_VOL4 [6:0]							0	0080h
R2320 (910h)	HPLP3MIX_Input_1_Source	LHPF3MIX_STS1	0	0	0	0	0	0	0	LHPF3MIX_SRC1 [7:0]								0000h
R2321 (911h)	HPLP3MIX_Input_1_Volume	0	0	0	0	0	0	0	0	LHPF3MIX_VOL1 [6:0]							0	0080h
R2322 (912h)	HPLP3MIX_Input_2_Source	LHPF3MIX_STS2	0	0	0	0	0	0	0	LHPF3MIX_SRC2 [7:0]								0000h
R2323 (913h)	HPLP3MIX_Input_2_Volume	0	0	0	0	0	0	0	0	LHPF3MIX_VOL2 [6:0]							0	0080h
R2324 (914h)	HPLP3MIX_Input_3_Source	LHPF3MIX_STS3	0	0	0	0	0	0	0	LHPF3MIX_SRC3 [7:0]								0000h
R2325 (915h)	HPLP3MIX_Input_3_Volume	0	0	0	0	0	0	0	0	LHPF3MIX_VOL3 [6:0]							0	0080h
R2326 (916h)	HPLP3MIX_Input_4_Source	LHPF3MIX_STS4	0	0	0	0	0	0	0	LHPF3MIX_SRC4 [7:0]								0000h
R2327 (917h)	HPLP3MIX_Input_4_Volume	0	0	0	0	0	0	0	0	LHPF3MIX_VOL4 [6:0]							0	0080h
R2328 (918h)	HPLP4MIX_Input_1_Source	LHPF4MIX_STS1	0	0	0	0	0	0	0	LHPF4MIX_SRC1 [7:0]								0000h
R2329 (919h)	HPLP4MIX_Input_1_Volume	0	0	0	0	0	0	0	0	LHPF4MIX_VOL1 [6:0]							0	0080h
R2330 (91Ah)	HPLP4MIX_Input_2_Source	LHPF4MIX_STS2	0	0	0	0	0	0	0	LHPF4MIX_SRC2 [7:0]								0000h
R2331 (91Bh)	HPLP4MIX_Input_2_Volume	0	0	0	0	0	0	0	0	LHPF4MIX_VOL2 [6:0]							0	0080h
R2332 (91Ch)	HPLP4MIX_Input_3_Source	LHPF4MIX_STS3	0	0	0	0	0	0	0	LHPF4MIX_SRC3 [7:0]								0000h
R2333 (91Dh)	HPLP4MIX_Input_3_Volume	0	0	0	0	0	0	0	0	LHPF4MIX_VOL3 [6:0]							0	0080h
R2334 (91Eh)	HPLP4MIX_Input_4_Source	LHPF4MIX_STS4	0	0	0	0	0	0	0	LHPF4MIX_SRC4 [7:0]								0000h
R2335 (91Fh)	HPLP4MIX_Input_4_Volume	0	0	0	0	0	0	0	0	LHPF4MIX_VOL4 [6:0]							0	0080h
R2368 (940h)	DSP1LMIX_Input_1_Source	DSP1LMIX_STS1	0	0	0	0	0	0	0	DSP1LMIX_SRC1 [7:0]								0000h
R2369 (941h)	DSP1LMIX_Input_1_Volume	0	0	0	0	0	0	0	0	DSP1LMIX_VOL1 [6:0]							0	0080h
R2370 (942h)	DSP1LMIX_Input_2_Source	DSP1LMIX_STS2	0	0	0	0	0	0	0	DSP1LMIX_SRC2 [7:0]								0000h
R2371 (943h)	DSP1LMIX_Input_2_Volume	0	0	0	0	0	0	0	0	DSP1LMIX_VOL2 [6:0]							0	0080h
R2372 (944h)	DSP1LMIX_Input_3_Source	DSP1LMIX_STS3	0	0	0	0	0	0	0	DSP1LMIX_SRC3 [7:0]								0000h
R2373 (945h)	DSP1LMIX_Input_3_Volume	0	0	0	0	0	0	0	0	DSP1LMIX_VOL3 [6:0]							0	0080h
R2374 (946h)	DSP1LMIX_Input_4_Source	DSP1LMIX_STS4	0	0	0	0	0	0	0	DSP1LMIX_SRC4 [7:0]								0000h
R2375 (947h)	DSP1LMIX_Input_4_Volume	0	0	0	0	0	0	0	0	DSP1LMIX_VOL4 [6:0]							0	0080h
R2376 (948h)	DSP1RMIX_Input_1_Source	DSP1RMIX_STS1	0	0	0	0	0	0	0	DSP1RMIX_SRC1 [7:0]								0000h
R2377 (949h)	DSP1RMIX_Input_1_Volume	0	0	0	0	0	0	0	0	DSP1RMIX_VOL1 [6:0]							0	0080h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R2378 (94Ah)	DSP1RMIX_Input_2_Source	DSP1RMIX_STS2	0	0	0	0	0	0	0	DSP1RMIX_SRC2 [7:0]								0000h
R2379 (94Bh)	DSP1RMIX_Input_2_Volume	0	0	0	0	0	0	0	0	DSP1RMIX_VOL2 [6:0]							0	0080h
R2380 (94Ch)	DSP1RMIX_Input_3_Source	DSP1RMIX_STS3	0	0	0	0	0	0	0	DSP1RMIX_SRC3 [7:0]								0000h
R2381 (94Dh)	DSP1RMIX_Input_3_Volume	0	0	0	0	0	0	0	0	DSP1RMIX_VOL3 [6:0]							0	0080h
R2382 (94Eh)	DSP1RMIX_Input_4_Source	DSP1RMIX_STS4	0	0	0	0	0	0	0	DSP1RMIX_SRC4 [7:0]								0000h
R2383 (94Fh)	DSP1RMIX_Input_4_Volume	0	0	0	0	0	0	0	0	DSP1RMIX_VOL4 [6:0]							0	0080h
R2384 (950h)	DSP1AUX1MIX_Input_1_Source	DSP1AUX1_STS	0	0	0	0	0	0	0	DSP1AUX1_SRC [7:0]								0000h
R2392 (958h)	DSP1AUX2MIX_Input_1_Source	DSP1AUX2_STS	0	0	0	0	0	0	0	DSP1AUX2_SRC [7:0]								0000h
R2400 (960h)	DSP1AUX3MIX_Input_1_Source	DSP1AUX3_STS	0	0	0	0	0	0	0	DSP1AUX3_SRC [7:0]								0000h
R2408 (968h)	DSP1AUX4MIX_Input_1_Source	DSP1AUX4_STS	0	0	0	0	0	0	0	DSP1AUX4_SRC [7:0]								0000h
R2416 (970h)	DSP1AUX5MIX_Input_1_Source	DSP1AUX5_STS	0	0	0	0	0	0	0	DSP1AUX5_SRC [7:0]								0000h
R2424 (978h)	DSP1AUX6MIX_Input_1_Source	DSP1AUX6_STS	0	0	0	0	0	0	0	DSP1AUX6_SRC [7:0]								0000h
R2816 (B00h)	ISRC1DEC1MIX_Input_1_Source	ISRC1DEC1_STS	0	0	0	0	0	0	0	ISRC1DEC1_SRC [7:0]								0000h
R2824 (B08h)	ISRC1DEC2MIX_Input_1_Source	ISRC1DEC2_STS	0	0	0	0	0	0	0	ISRC1DEC2_SRC [7:0]								0000h
R2832 (B10h)	ISRC1DEC3MIX_Input_1_Source	ISRC1DEC3_STS	0	0	0	0	0	0	0	ISRC1DEC3_SRC [7:0]								0000h
R2840 (B18h)	ISRC1DEC4MIX_Input_1_Source	ISRC1DEC4_STS	0	0	0	0	0	0	0	ISRC1DEC4_SRC [7:0]								0000h
R2848 (B20h)	ISRC1INT1MIX_Input_1_Source	ISRC1INT1_STS	0	0	0	0	0	0	0	ISRC1INT1_SRC [7:0]								0000h
R2856 (B28h)	ISRC1INT2MIX_Input_1_Source	ISRC1INT2_STS	0	0	0	0	0	0	0	ISRC1INT2_SRC [7:0]								0000h
R2864 (B30h)	ISRC1INT3MIX_Input_1_Source	ISRC1INT3_STS	0	0	0	0	0	0	0	ISRC1INT3_SRC [7:0]								0000h
R2872 (B38h)	ISRC1INT4MIX_Input_1_Source	ISRC1INT4_STS	0	0	0	0	0	0	0	ISRC1INT4_SRC [7:0]								0000h
R2880 (B40h)	ISRC2DEC1MIX_Input_1_Source	ISRC2DEC1_STS	0	0	0	0	0	0	0	ISRC2DEC1_SRC [7:0]								0000h
R2888 (B48h)	ISRC2DEC2MIX_Input_1_Source	ISRC2DEC2_STS	0	0	0	0	0	0	0	ISRC2DEC2_SRC [7:0]								0000h
R2896 (B50h)	ISRC2DEC3MIX_Input_1_Source	ISRC2DEC3_STS	0	0	0	0	0	0	0	ISRC2DEC3_SRC [7:0]								0000h
R2904 (B58h)	ISRC2DEC4MIX_Input_1_Source	ISRC2DEC4_STS	0	0	0	0	0	0	0	ISRC2DEC4_SRC [7:0]								0000h
R2912 (B60h)	ISRC2INT1MIX_Input_1_Source	ISRC2INT1_STS	0	0	0	0	0	0	0	ISRC2INT1_SRC [7:0]								0000h
R2920 (B68h)	ISRC2INT2MIX_Input_1_Source	ISRC2INT2_STS	0	0	0	0	0	0	0	ISRC2INT2_SRC [7:0]								0000h
R2928 (B70h)	ISRC2INT3MIX_Input_1_Source	ISRC2INT3_STS	0	0	0	0	0	0	0	ISRC2INT3_SRC [7:0]								0000h
R2936 (B78h)	ISRC2INT4MIX_Input_1_Source	ISRC2INT4_STS	0	0	0	0	0	0	0	ISRC2INT4_SRC [7:0]								0000h
R3584 (E00h)	FX_Ctrl1	0	FX_RATE [3:0]				0	0	0	0	0	0	0	0	0	0	0	0000h
R3585 (E01h)	FX_Ctrl2	FX_STS [11:0]												0	0	1	0	0002h
R3600 (E10h)	EQ1_1	EQ1_B1_GAIN [4:0]					EQ1_B2_GAIN [4:0]					EQ1_B3_GAIN [4:0]					EQ1_ENA	6318h
R3601 (E11h)	EQ1_2	EQ1_B4_GAIN [4:0]					EQ1_B5_GAIN [4:0]					0	0	0	0	0	EQ1_B1_MODE	6300h
R3602 (E12h)	EQ1_3	EQ1_B1_A [15:0]																0FC8h
R3603 (E13h)	EQ1_4	EQ1_B1_B [15:0]																03FEh
R3604 (E14h)	EQ1_5	EQ1_B1_PG [15:0]																00E0h
R3605 (E15h)	EQ1_6	EQ1_B2_A [15:0]																1EC4h
R3606 (E16h)	EQ1_7	EQ1_B2_B [15:0]																F136h
R3607 (E17h)	EQ1_8	EQ1_B2_C [15:0]																0409h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R3608 (E18h)	EQ1_9	EQ1_B2_PG [15:0]																04CCh
R3609 (E19h)	EQ1_10	EQ1_B3_A [15:0]																1C9Bh
R3610 (E1Ah)	EQ1_11	EQ1_B3_B [15:0]																F337h
R3611 (E1Bh)	EQ1_12	EQ1_B3_C [15:0]																040Bh
R3612 (E1Ch)	EQ1_13	EQ1_B3_PG [15:0]																0CBBh
R3613 (E1Dh)	EQ1_14	EQ1_B4_A [15:0]																16F8h
R3614 (E1Eh)	EQ1_15	EQ1_B4_B [15:0]																F7D9h
R3615 (E1Fh)	EQ1_16	EQ1_B4_C [15:0]																040Ah
R3616 (E20h)	EQ1_17	EQ1_B4_PG [15:0]																1F14h
R3617 (E21h)	EQ1_18	EQ1_B5_A [15:0]																058Ch
R3618 (E22h)	EQ1_19	EQ1_B5_B [15:0]																0563h
R3619 (E23h)	EQ1_20	EQ1_B5_PG [15:0]																4000h
R3620 (E24h)	EQ1_21	EQ1_B1_C [15:0]																0B75h
R3622 (E26h)	EQ2_1	EQ2_B1_GAIN [4:0]					EQ2_B2_GAIN [4:0]					EQ2_B3_GAIN [4:0]					EQ2_ENA	6318h
R3623 (E27h)	EQ2_2	EQ2_B4_GAIN [4:0]					EQ2_B5_GAIN [4:0]					0	0	0	0	0	EQ2_B1_MODE	6300h
R3624 (E28h)	EQ2_3	EQ2_B1_A [15:0]																0FC8h
R3625 (E29h)	EQ2_4	EQ2_B1_B [15:0]																03FEh
R3626 (E2Ah)	EQ2_5	EQ2_B1_PG [15:0]																00E0h
R3627 (E2Bh)	EQ2_6	EQ2_B2_A [15:0]																1EC4h
R3628 (E2Ch)	EQ2_7	EQ2_B2_B [15:0]																F136h
R3629 (E2Dh)	EQ2_8	EQ2_B2_C [15:0]																0409h
R3630 (E2Eh)	EQ2_9	EQ2_B2_PG [15:0]																04CCh
R3631 (E2Fh)	EQ2_10	EQ2_B3_A [15:0]																1C9Bh
R3632 (E30h)	EQ2_11	EQ2_B3_B [15:0]																F337h
R3633 (E31h)	EQ2_12	EQ2_B3_C [15:0]																040Bh
R3634 (E32h)	EQ2_13	EQ2_B3_PG [15:0]																0CBBh
R3635 (E33h)	EQ2_14	EQ2_B4_A [15:0]																16F8h
R3636 (E34h)	EQ2_15	EQ2_B4_B [15:0]																F7D9h
R3637 (E35h)	EQ2_16	EQ2_B4_C [15:0]																040Ah
R3638 (E36h)	EQ2_17	EQ2_B4_PG [15:0]																1F14h
R3639 (E37h)	EQ2_18	EQ2_B5_A [15:0]																058Ch
R3640 (E38h)	EQ2_19	EQ2_B5_B [15:0]																0563h
R3641 (E39h)	EQ2_20	EQ2_B5_PG [15:0]																4000h
R3642 (E3Ah)	EQ2_21	EQ2_B1_C [15:0]																0B75h
R3644 (E3Ch)	EQ3_1	EQ3_B1_GAIN [4:0]					EQ3_B2_GAIN [4:0]					EQ3_B3_GAIN [4:0]					EQ3_ENA	6318h
R3645 (E3Dh)	EQ3_2	EQ3_B4_GAIN [4:0]					EQ3_B5_GAIN [4:0]					0	0	0	0	0	EQ3_B1_MODE	6300h
R3646 (E3Eh)	EQ3_3	EQ3_B1_A [15:0]																0FC8h
R3647 (E3Fh)	EQ3_4	EQ3_B1_B [15:0]																03FEh

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R3648 (E40h)	EQ3_5	EQ3_B1_PG [15:0]																00E0h
R3649 (E41h)	EQ3_6	EQ3_B2_A [15:0]																1EC4h
R3650 (E42h)	EQ3_7	EQ3_B2_B [15:0]																F136h
R3651 (E43h)	EQ3_8	EQ3_B2_C [15:0]																0409h
R3652 (E44h)	EQ3_9	EQ3_B2_PG [15:0]																04CCh
R3653 (E45h)	EQ3_10	EQ3_B3_A [15:0]																1C9Bh
R3654 (E46h)	EQ3_11	EQ3_B3_B [15:0]																F337h
R3655 (E47h)	EQ3_12	EQ3_B3_C [15:0]																040Bh
R3656 (E48h)	EQ3_13	EQ3_B3_PG [15:0]																0CBBh
R3657 (E49h)	EQ3_14	EQ3_B4_A [15:0]																16F8h
R3658 (E4Ah)	EQ3_15	EQ3_B4_B [15:0]																F7D9h
R3659 (E4Bh)	EQ3_16	EQ3_B4_C [15:0]																040Ah
R3660 (E4Ch)	EQ3_17	EQ3_B4_PG [15:0]																1F14h
R3661 (E4Dh)	EQ3_18	EQ3_B5_A [15:0]																058Ch
R3662 (E4Eh)	EQ3_19	EQ3_B5_B [15:0]																0563h
R3663 (E4Fh)	EQ3_20	EQ3_B5_PG [15:0]																4000h
R3664 (E50h)	EQ3_21	EQ3_B1_C [15:0]																0B75h
R3666 (E52h)	EQ4_1	EQ4_B1_GAIN [4:0]					EQ4_B2_GAIN [4:0]					EQ4_B3_GAIN [4:0]					EQ4_ENA	6318h
R3667 (E53h)	EQ4_2	EQ4_B4_GAIN [4:0]					EQ4_B5_GAIN [4:0]					0	0	0	0	0	EQ4_B1_MODE	6300h
R3668 (E54h)	EQ4_3	EQ4_B1_A [15:0]																0FC8h
R3669 (E55h)	EQ4_4	EQ4_B1_B [15:0]																03FEh
R3670 (E56h)	EQ4_5	EQ4_B1_PG [15:0]																00E0h
R3671 (E57h)	EQ4_6	EQ4_B2_A [15:0]																1EC4h
R3672 (E58h)	EQ4_7	EQ4_B2_B [15:0]																F136h
R3673 (E59h)	EQ4_8	EQ4_B2_C [15:0]																0409h
R3674 (E5Ah)	EQ4_9	EQ4_B2_PG [15:0]																04CCh
R3675 (E5Bh)	EQ4_10	EQ4_B3_A [15:0]																1C9Bh
R3676 (E5Ch)	EQ4_11	EQ4_B3_B [15:0]																F337h
R3677 (E5Dh)	EQ4_12	EQ4_B3_C [15:0]																040Bh
R3678 (E5Eh)	EQ4_13	EQ4_B3_PG [15:0]																0CBBh
R3679 (E5Fh)	EQ4_14	EQ4_B4_A [15:0]																16F8h
R3680 (E60h)	EQ4_15	EQ4_B4_B [15:0]																F7D9h
R3681 (E61h)	EQ4_16	EQ4_B4_C [15:0]																040Ah
R3682 (E62h)	EQ4_17	EQ4_B4_PG [15:0]																1F14h
R3683 (E63h)	EQ4_18	EQ4_B5_A [15:0]																058Ch
R3684 (E64h)	EQ4_19	EQ4_B5_B [15:0]																0563h
R3685 (E65h)	EQ4_20	EQ4_B5_PG [15:0]																4000h
R3686 (E66h)	EQ4_21	EQ4_B1_C [15:0]																0B75h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R3712 (E80h)	DRC1_ctrl1	DRC1_SIG_DET_RMS [4:0]					DRC1_SIG_DET_PK [1:0]		DRC1_NG_ENA	DRC1_SIG_DET_MODE	DRC1_SIG_DET	DRC1_KNEE2_OP_ENA	DRC1_QR	DRC1_ANTICLIP	DRC1_WSEQ_SIG_DET_ENA	DRC1L_ENA	DRC1R_ENA	0018h
R3713 (E81h)	DRC1_ctrl2	0	0	0	DRC1_ATK [3:0]				DRC1_DCY [3:0]				DRC1_MINGAIN [2:0]			DRC1_MAXGAIN [1:0]		0933h
R3714 (E82h)	DRC1_ctrl3	DRC1_NG_MINGAIN [3:0]				DRC1_NG_EXP [1:0]		DRC1_QR_THR [1:0]		DRC1_QR_DCY [1:0]		DRC1_HI_COMP [2:0]			DRC1_LO_COMP [2:0]			0018h
R3715 (E83h)	DRC1_ctrl4	0	0	0	0	0	DRC1_KNEE_IP [5:0]						DRC1_KNEE_OP [4:0]					0000h
R3716 (E84h)	DRC1_ctrl5	0	0	0	0	0	0	DRC1_KNEE2_IP [4:0]					DRC1_KNEE2_OP [4:0]					0000h
R3720 (E88h)	DRC2_ctrl1	DRC2_SIG_DET_RMS [4:0]					DRC2_SIG_DET_PK [1:0]		DRC2_NG_ENA	DRC2_SIG_DET_MODE	DRC2_SIG_DET	DRC2_KNEE2_OP_ENA	DRC2_QR	DRC2_ANTICLIP	0	DRC2L_ENA	DRC2R_ENA	0018h
R3721 (E89h)	DRC2_ctrl2	0	0	0	DRC2_ATK [3:0]				DRC2_DCY [3:0]				DRC2_MINGAIN [2:0]			DRC2_MAXGAIN [1:0]		0933h
R3722 (E8Ah)	DRC2_ctrl3	DRC2_NG_MINGAIN [3:0]				DRC2_NG_EXP [1:0]		DRC2_QR_THR [1:0]		DRC2_QR_DCY [1:0]		DRC2_HI_COMP [2:0]			DRC2_LO_COMP [2:0]			0018h
R3723 (E8Bh)	DRC2_ctrl4	0	0	0	0	0	DRC2_KNEE_IP [5:0]						DRC2_KNEE_OP [4:0]					0000h
R3724 (E8Ch)	DRC2_ctrl5	0	0	0	0	0	0	DRC2_KNEE2_IP [4:0]					DRC2_KNEE2_OP [4:0]					0000h
R3776 (EC0h)	HPLPF1_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	LHPF1_MODE	LHPF1_ENA	0000h
R3777 (EC1h)	HPLPF1_2	LHPF1_COEFF [15:0]																0000h
R3780 (EC4h)	HPLPF2_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	LHPF2_MODE	LHPF2_ENA	0000h
R3781 (EC5h)	HPLPF2_2	LHPF2_COEFF [15:0]																0000h
R3784 (EC8h)	HPLPF3_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	LHPF3_MODE	LHPF3_ENA	0000h
R3785 (EC9h)	HPLPF3_2	LHPF3_COEFF [15:0]																0000h
R3788 (ECCh)	HPLPF4_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	LHPF4_MODE	LHPF4_ENA	0000h
R3789 (ECDh)	HPLPF4_2	LHPF4_COEFF [15:0]																0000h
R3824 (EF0h)	ISRC1_CTRL_1	0	ISRC1_FSH [3:0]				0	0	0	0	0	0	0	0	0	0	0	0000h
R3825 (EF1h)	ISRC1_CTRL_2	0	ISRC1_FSL [3:0]				0	0	0	0	0	0	0	0	0	0	1	0001h
R3826 (EF2h)	ISRC1_CTRL_3	ISRC1_INT1_ENA	ISRC1_INT2_ENA	ISRC1_INT3_ENA	ISRC1_INT4_ENA	0	0	ISRC1_DEC1_ENA	ISRC1_DEC2_ENA	ISRC1_DEC3_ENA	ISRC1_DEC4_ENA	0	0	0	0	0	0	0000h
R3827 (EF3h)	ISRC2_CTRL_1	0	ISRC2_FSH [3:0]				0	0	0	0	0	0	0	0	0	0	0	0000h
R3828 (EF4h)	ISRC2_CTRL_2	0	ISRC2_FSL [3:0]				0	0	0	0	0	0	0	0	0	0	1	0001h
R3829 (EF5h)	ISRC2_CTRL_3	ISRC2_INT1_ENA	ISRC2_INT2_ENA	ISRC2_INT3_ENA	ISRC2_INT4_ENA	0	0	ISRC2_DEC1_ENA	ISRC2_DEC2_ENA	ISRC2_DEC3_ENA	ISRC2_DEC4_ENA	0	0	0	0	0	0	0000h
R5632 (1600h)	ADSP2_IRQ0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP_IRQ2	DSP_IRQ1	0000h
R5633 (1601h)	ADSP2_IRQ1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP_IRQ4	DSP_IRQ3	0000h
R5634 (1602h)	ADSP2_IRQ2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP_IRQ6	DSP_IRQ5	0000h
R5635 (1603h)	ADSP2_IRQ3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP_IRQ8	DSP_IRQ7	0000h
R5636 (1604h)	ADSP2_IRQ4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP_IRQ10	DSP_IRQ9	0000h
R5637 (1605h)	ADSP2_IRQ5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP_IRQ12	DSP_IRQ11	0000h
R5638 (1606h)	ADSP2_IRQ6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP_IRQ14	DSP_IRQ13	0000h
R5639 (1607h)	ADSP2_IRQ7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP_IRQ16	DSP_IRQ15	0000h
R5888 (1700h)	GPIO1_CTRL_1	GP1_LVL	GP1_OP_CFG	GP1_DB	GP1_POL	GP1_IP_CFG	0	GP1_FN [9:0]										2801h
R5889 (1701h)	GPIO1_CTRL_2	GP1_DIR	GP1_PU	GP1_PD	GP1_DRV_STR [1:0]		0	0	0	0	0	0	0	0	0	0	0	E800h
R5890 (1702h)	GPIO2_CTRL_1	GP2_LVL	GP2_OP_CFG	GP2_DB	GP2_POL	GP2_IP_CFG	0	GP2_FN [9:0]										2801h
R5891 (1703h)	GPIO2_CTRL_2	GP2_DIR	GP2_PU	GP2_PD	GP2_DRV_STR [1:0]		0	0	0	0	0	0	0	0	0	0	0	E800h
R5892 (1704h)	GPIO3_CTRL_1	GP3_LVL	GP3_OP_CFG	GP3_DB	GP3_POL	GP3_IP_CFG	0	GP3_FN [9:0]										2801h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R5893 (1705h)	GPIO3_CTRL_2	GP3_DIR	GP3_PU	GP3_PD	GP3_DRV_STR [1:0]		0	0	0	0	0	0	0	0	0	0	0	E800h
R5894 (1706h)	GPIO4_CTRL_1	GP4_LVL	GP4_OP_CFG	GP4_DB	GP4_POL	GP4_IP_CFG	0	GP4_FN [9:0]										2801h
R5895 (1707h)	GPIO4_CTRL_2	GP4_DIR	GP4_PU	GP4_PD	GP4_DRV_STR [1:0]		0	0	0	0	0	0	0	0	0	0	0	E800h
R5896 (1708h)	GPIO5_CTRL_1	GP5_LVL	GP5_OP_CFG	GP5_DB	GP5_POL	GP5_IP_CFG	0	GP5_FN [9:0]										2801h
R5897 (1709h)	GPIO5_CTRL_2	GP5_DIR	GP5_PU	GP5_PD	GP5_DRV_STR [1:0]		0	0	0	0	0	0	0	0	0	0	0	E800h
R5898 (170Ah)	GPIO6_CTRL_1	GP6_LVL	GP6_OP_CFG	GP6_DB	GP6_POL	GP6_IP_CFG	0	GP6_FN [9:0]										2801h
R5899 (170Bh)	GPIO6_CTRL_2	GP6_DIR	GP6_PU	GP6_PD	GP6_DRV_STR [1:0]		0	0	0	0	0	0	0	0	0	0	0	E800h
R5900 (170Ch)	GPIO7_CTRL_1	GP7_LVL	GP7_OP_CFG	GP7_DB	GP7_POL	GP7_IP_CFG	0	GP7_FN [9:0]										2801h
R5901 (170Dh)	GPIO7_CTRL_2	GP7_DIR	GP7_PU	GP7_PD	GP7_DRV_STR [1:0]		0	0	0	0	0	0	0	0	0	0	0	E800h
R5902 (170Eh)	GPIO8_CTRL_1	GP8_LVL	GP8_OP_CFG	GP8_DB	GP8_POL	GP8_IP_CFG	0	GP8_FN [9:0]										2801h
R5903 (170Fh)	GPIO8_CTRL_2	GP8_DIR	GP8_PU	GP8_PD	GP8_DRV_STR [1:0]		0	0	0	0	0	0	0	0	0	0	0	E800h
R5904 (1710h)	GPIO9_CTRL_1	GP9_LVL	GP9_OP_CFG	GP9_DB	GP9_POL	GP9_IP_CFG	0	GP9_FN [9:0]										2801h
R5905 (1711h)	GPIO9_CTRL_2	GP9_DIR	GP9_PU	GP9_PD	GP9_DRV_STR [1:0]		0	0	0	0	0	0	0	0	0	0	0	E800h
R5906 (1712h)	GPIO10_CTRL_1	GP10_LVL	GP10_OP_CFG	GP10_DB	GP10_POL	GP10_IP_CFG	0	GP10_FN [9:0]										2801h
R5907 (1713h)	GPIO10_CTRL_2	GP10_DIR	GP10_PU	GP10_PD	GP10_DRV_STR [1:0]		0	0	0	0	0	0	0	0	0	0	0	E800h
R5908 (1714h)	GPIO11_CTRL_1	GP11_LVL	GP11_OP_CFG	GP11_DB	GP11_POL	GP11_IP_CFG	0	GP11_FN [9:0]										2801h
R5909 (1715h)	GPIO11_CTRL_2	GP11_DIR	GP11_PU	GP11_PD	GP11_DRV_STR [1:0]		0	0	0	0	0	0	0	0	0	0	0	E800h
R5910 (1716h)	GPIO12_CTRL_1	GP12_LVL	GP12_OP_CFG	GP12_DB	GP12_POL	GP12_IP_CFG	0	GP12_FN [9:0]										2801h
R5911 (1717h)	GPIO12_CTRL_2	GP12_DIR	GP12_PU	GP12_PD	GP12_DRV_STR [1:0]		0	0	0	0	0	0	0	0	0	0	0	E800h
R5912 (1718h)	GPIO13_CTRL_1	GP13_LVL	GP13_OP_CFG	GP13_DB	GP13_POL	GP13_IP_CFG	0	GP13_FN [9:0]										2801h
R5913 (1719h)	GPIO13_CTRL_2	GP13_DIR	GP13_PU	GP13_PD	GP13_DRV_STR [1:0]		0	0	0	0	0	0	0	0	0	0	0	E800h
R5914 (171Ah)	GPIO14_CTRL_1	GP14_LVL	GP14_OP_CFG	GP14_DB	GP14_POL	GP14_IP_CFG	0	GP14_FN [9:0]										2801h
R5915 (171Bh)	GPIO14_CTRL_2	GP14_DIR	GP14_PU	GP14_PD	GP14_DRV_STR [1:0]		0	0	0	0	0	0	0	0	0	0	0	E800h
R5916 (171Ch)	GPIO15_CTRL_1	GP15_LVL	GP15_OP_CFG	GP15_DB	GP15_POL	GP15_IP_CFG	0	GP15_FN [9:0]										2801h
R5917 (171Dh)	GPIO15_CTRL_2	GP15_DIR	GP15_PU	GP15_PD	GP15_DRV_STR [1:0]		0	0	0	0	0	0	0	0	0	0	0	E800h
R6144 (1800h)	IRQ1_Status_1	0	0	0	CTRLIF_ERR_EINT1	0	0	SYSCLK_FAIL_EINT1	0	BOOT_DONE_EINT1	0	0	0	0	0	0	0	0000h
R6145 (1801h)	IRQ1_Status_2	FLL_AO_REF_LOST_EINT1	DSPCLK_ERR_EINT1	0	SYSCLK_ERR_EINT1	FLL_AO_LOCK_EINT1	0	0	FLL1_LOCK_EINT1	0	0	0	0	0	0	0	0	0000h
R6149 (1805h)	IRQ1_Status_6	0	0	0	0	0	0	MICDET2_EINT1	MICDET1_EINT1	0	0	0	0	0	0	0	HPDET_EINT1	0000h
R6150 (1806h)	IRQ1_Status_7	0	0	0	0	0	0	0	0	0	0	MICD_CLAMP_FALL_EINT1	MICD_CLAMP_RISE_EINT1	JD2_FALL_EINT1	JD2_RISE_EINT1	JD1_FALL_EINT1	JD1_RISE_EINT1	0000h
R6152 (1808h)	IRQ1_Status_9	0	0	0	0	0	0	0	0	0	0	0	0	0	INPUTS_SIG_DET_EINT1	DRC2_SIG_DET_EINT1	DRC1_SIG_DET_EINT1	0000h
R6154 (180Ah)	IRQ1_Status_11	DSP_IRQ16_EINT1	DSP_IRQ15_EINT1	DSP_IRQ14_EINT1	DSP_IRQ13_EINT1	DSP_IRQ12_EINT1	DSP_IRQ11_EINT1	DSP_IRQ10_EINT1	DSP_IRQ9_EINT1	DSP_IRQ8_EINT1	DSP_IRQ7_EINT1	DSP_IRQ6_EINT1	DSP_IRQ5_EINT1	DSP_IRQ4_EINT1	DSP_IRQ3_EINT1	DSP_IRQ2_EINT1	DSP_IRQ1_EINT1	0000h
R6155 (180Bh)	IRQ1_Status_12	0	0	0	0	0	0	0	0	0	SPKOUTL_SC_EINT1	0	0	HP2R_SC_EINT1	HP2L_SC_EINT1	HP1R_SC_EINT1	HP1L_SC_EINT1	0000h
R6156 (180Ch)	IRQ1_Status_13	0	0	0	0	0	0	0	0	0	SPKOUTL_ENABLE_DONE_EINT1	0	0	0	0	HP1R_ENABLE_DONE_EINT1	HP1L_ENABLE_DONE_EINT1	0000h
R6157 (180Dh)	IRQ1_Status_14	0	0	0	0	0	0	0	0	0	SPKOUTL_DISABLE_DONE_EINT1	0	0	0	0	HP1R_DISABLE_DONE_EINT1	HP1L_DISABLE_DONE_EINT1	0000h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R6158 (180Eh)	IRQ1_Status_15	0	0	0	0	0	0	0	0	0	0	0	0	0	SPK_OVERHEAT_WARN_EINT1	SPK_OVERHEAT_EINT1	SPK_SHUTDOWN_EINT1	0000h
R6159 (180Fh)	IRQ1_Status_16	0	0	0	0	0	0	0	0	MIF4_OVERCLOCKED_EINT1	0	0	0	0	0	0	0	0000h
R6160 (1810h)	IRQ1_Status_17	0	GP15_EINT1	GP14_EINT1	GP13_EINT1	GP12_EINT1	GP11_EINT1	GP10_EINT1	GP9_EINT1	GP8_EINT1	GP7_EINT1	GP6_EINT1	GP5_EINT1	GP4_EINT1	GP3_EINT1	GP2_EINT1	GP1_EINT1	0000h
R6164 (1814h)	IRQ1_Status_21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TIMER2_EINT1	TIMER1_EINT1	0000h
R6165 (1815h)	IRQ1_Status_22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT2_NOT_EMPTY_EINT1	EVENT1_NOT_EMPTY_EINT1	0000h
R6166 (1816h)	IRQ1_Status_23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT2_FULL_EINT1	EVENT1_FULL_EINT1	0000h
R6167 (1817h)	IRQ1_Status_24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT2_WMARK_EINT1	EVENT1_WMARK_EINT1	0000h
R6168 (1818h)	IRQ1_Status_25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_DMA_EINT1	0000h
R6170 (181Ah)	IRQ1_Status_27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_START1_EINT1	0000h
R6171 (181Bh)	IRQ1_Status_28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_START2_EINT1	0000h
R6173 (181Dh)	IRQ1_Status_30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_BUSY_EINT1	0000h
R6174 (181Eh)	IRQ1_Status_31	0	0	0	0	0	0	0	0	0	0	0	0	MIF4_DONE_EINT1	0	0	0	0000h
R6175 (181Fh)	IRQ1_Status_32	0	0	0	0	0	0	0	0	0	0	0	0	MIF4_BLOCK_EINT1	0	0	0	0000h
R6176 (1820h)	IRQ1_Status_33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_BUS_ERR_EINT1	0000h
R6208 (1840h)	IRQ1_Mask_1	0	0	0	IM_CTRLIF_ERR_EINT1	0	0	IM_SYSCLK_FAIL_EINT1	0	IM_BOOT_DONE_EINT1	0	0	0	0	0	0	0	1200h
R6209 (1841h)	IRQ1_Mask_2	IM_FLL_AO_REF_LOST_EINT1	IM_DSPCLK_ERR_EINT1	0	IM_SYSCLK_ERR_EINT1	IM_FLL_AO_LOCK_EINT1	0	0	IM_FLL1_LOCK_EINT1	0	0	0	0	0	0	0	0	D900h
R6213 (1845h)	IRQ1_Mask_6	0	0	0	0	0	0	IM_MICDET2_EINT1	IM_MICDET1_EINT1	0	0	0	0	0	0	0	IM_HPDET_EINT1	0301h
R6214 (1846h)	IRQ1_Mask_7	0	0	0	0	0	0	0	0	0	0	IM_MICD_CLAMP_FALL_EINT1	IM_MICD_CLAMP_RISE_EINT1	IM_JD2_FALL_EINT1	IM_JD2_RISE_EINT1	IM_JD1_FALL_EINT1	IM_JD1_RISE_EINT1	003Fh
R6216 (1848h)	IRQ1_Mask_9	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_INPUTS_SIG_DET_EINT1	IM_DRC2_SIG_DET_EINT1	IM_DRC1_SIG_DET_EINT1	0007h
R6218 (184Ah)	IRQ1_Mask_11	IM_DSP_IRQ16_EINT1	IM_DSP_IRQ15_EINT1	IM_DSP_IRQ14_EINT1	IM_DSP_IRQ13_EINT1	IM_DSP_IRQ12_EINT1	IM_DSP_IRQ11_EINT1	IM_DSP_IRQ10_EINT1	IM_DSP_IRQ9_EINT1	IM_DSP_IRQ8_EINT1	IM_DSP_IRQ7_EINT1	IM_DSP_IRQ6_EINT1	IM_DSP_IRQ5_EINT1	IM_DSP_IRQ4_EINT1	IM_DSP_IRQ3_EINT1	IM_DSP_IRQ2_EINT1	IM_DSP_IRQ1_EINT1	FFFFh
R6219 (184Bh)	IRQ1_Mask_12	0	0	0	0	0	0	0	0	0	IM_SPKOUTL_SC_EINT1	0	0	IM_HP2R_SC_EINT1	IM_HP2L_SC_EINT1	IM_HP1R_SC_EINT1	IM_HP1L_SC_EINT1	004Fh
R6220 (184Ch)	IRQ1_Mask_13	0	0	0	0	0	0	0	0	0	IM_SPKOUTL_ENABLE_DONE_EINT1	0	0	0	0	IM_HP1R_ENABLE_DONE_EINT1	IM_HP1L_ENABLE_DONE_EINT1	0043h
R6221 (184Dh)	IRQ1_Mask_14	0	0	0	0	0	0	0	0	0	IM_SPKOUTL_DISABLE_DONE_EINT1	0	0	0	0	IM_HP1R_DISABLE_DONE_EINT1	IM_HP1L_DISABLE_DONE_EINT1	0043h
R6222 (184Eh)	IRQ1_Mask_15	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_SPK_OVERHEAT_WARN_EINT1	IM_SPK_OVERHEAT_EINT1	IM_SPK_SHUTDOWN_EINT1	0007h
R6223 (184Fh)	IRQ1_Mask_16	0	0	0	0	0	0	0	0	IM_MIF4_OVERCLOCKED_EINT1	0	0	0	0	0	0	0	0080h
R6224 (1850h)	IRQ1_Mask_17	0	IM_GP15_EINT1	IM_GP14_EINT1	IM_GP13_EINT1	IM_GP12_EINT1	IM_GP11_EINT1	IM_GP10_EINT1	IM_GP9_EINT1	IM_GP8_EINT1	IM_GP7_EINT1	IM_GP6_EINT1	IM_GP5_EINT1	IM_GP4_EINT1	IM_GP3_EINT1	IM_GP2_EINT1	IM_GP1_EINT1	7FFFh
R6228 (1854h)	IRQ1_Mask_21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_TIMER2_EINT1	IM_TIMER1_EINT1	0003h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R6229 (1855h)	IRQ1_Mask_22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_EVENT2_NOT_EMPTY_EINT1	IM_EVENT1_NOT_EMPTY_EINT1	0003h
R6230 (1856h)	IRQ1_Mask_23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_EVENT2_FULL_EINT1	IM_EVENT1_FULL_EINT1	0003h
R6231 (1857h)	IRQ1_Mask_24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_EVENT2_WMARK_EINT1	IM_EVENT1_WMARK_EINT1	0003h
R6232 (1858h)	IRQ1_Mask_25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP1_DMA_EINT1	0001h
R6234 (185Ah)	IRQ1_Mask_27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP1_START1_EINT1	0001h
R6235 (185Bh)	IRQ1_Mask_28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP1_START2_EINT1	0001h
R6237 (185Dh)	IRQ1_Mask_30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP1_BUSY_EINT1	0001h
R6238 (185Eh)	IRQ1_Mask_31	0	0	0	0	0	0	0	0	0	0	0	0	IM_MIF4_DONE_EINT1	0	0	0	0008h
R6239 (185Fh)	IRQ1_Mask_32	0	0	0	0	0	0	0	0	0	0	0	0	IM_MIF4_BLOCK_EINT1	0	0	0	0008h
R6240 (1860h)	IRQ1_Mask_33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP1_BUS_ERR_EINT1	0008h
R6272 (1880h)	IRQ1_Raw_Status_1	0	0	0	CTRLIF_ERR_STS1	0	0	0	0	BOOT_DONE_STS1	0	0	0	0	0	0	0	0000h
R6273 (1881h)	IRQ1_Raw_Status_2	FLL_AO_REF_LOST_STS1	DSPCLK_ERR_STS1	0	SYSCLK_ERR_STS1	FLL_AO_LOCK_STS1	0	0	FLL1_LOCK_STS1	0	0	0	0	0	0	0	0	0000h
R6278 (1886h)	IRQ1_Raw_Status_7	0	0	0	0	0	0	0	0	0	0	0	MICD_CLAMP_STS1	0	JD2_STS1	0	JD1_STS1	0000h
R6280 (1888h)	IRQ1_Raw_Status_9	0	0	0	0	0	0	0	0	0	0	0	0	0	INPUTS_SIG_DET_STS1	DRC2_SIG_DET_STS1	DRC1_SIG_DET_STS1	0000h
R6283 (188Bh)	IRQ1_Raw_Status_12	0	0	0	0	0	0	0	0	0	SPKOUTL_SC_STS1	0	0	HP2R_SC_STS1	HP2L_SC_STS1	HP1R_SC_STS1	HP1L_SC_STS1	0000h
R6284 (188Ch)	IRQ1_Raw_Status_13	0	0	0	0	0	0	0	0	0	SPKOUTL_ENABLE_DONE_STS1	0	0	0	0	HP1R_ENABLE_DONE_STS1	HP1L_ENABLE_DONE_STS1	0000h
R6285 (188Dh)	IRQ1_Raw_Status_14	0	0	0	0	0	0	0	0	0	SPKOUTL_DISABLE_DONE_STS1	0	0	0	0	HP1R_DISABLE_DONE_STS1	HP1L_DISABLE_DONE_STS1	0000h
R6286 (188Eh)	IRQ1_Raw_Status_15	0	0	0	0	0	0	0	0	0	0	0	0	0	SPK_OVERHEAT_WARN_STS1	SPK_OVERHEAT_STS1	SPK_SHUTDOWN_STS1	0000h
R6287 (188Fh)	IRQ1_Raw_Status_16	0	0	0	0	0	0	0	0	MIF4_OVERCLOCKED_STS1	0	0	0	0	0	0	0	0000h
R6288 (1890h)	IRQ1_Raw_Status_17	0	GPIO15_STS1	GPIO14_STS1	GPIO13_STS1	GPIO12_STS1	GPIO11_STS1	GPIO10_STS1	GPIO9_STS1	GPIO8_STS1	GPIO7_STS1	GPIO6_STS1	GPIO5_STS1	GPIO4_STS1	GPIO3_STS1	GPIO2_STS1	GPIO1_STS1	0000h
R6293 (1895h)	IRQ1_Raw_Status_22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT2_NOT_EMPTY_STS1	EVENT1_NOT_EMPTY_STS1	0000h
R6294 (1896h)	IRQ1_Raw_Status_23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT2_FULL_STS1	EVENT1_FULL_STS1	0000h
R6295 (1897h)	IRQ1_Raw_Status_24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT2_WMARK_STS1	EVENT1_WMARK_STS1	0000h
R6296 (1898h)	IRQ1_Raw_Status_25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_DMA_STS1	0000h
R6301 (189Dh)	IRQ1_Raw_Status_30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_BUSY_STS1	0000h
R6302 (189Eh)	IRQ1_Raw_Status_31	0	0	0	0	0	0	0	0	0	0	0	0	MIF4_DONE_STS1	0	0	0	0000h
R6303 (189Fh)	IRQ1_Raw_Status_32	0	0	0	0	0	0	0	0	0	0	0	0	MIF4_BLOCK_STS1	0	0	0	0000h
R6400 (1900h)	IRQ2_Status_1	0	0	0	CTRLIF_ERR_EINT2	0	0	SYSCLK_FAIL_EINT2	0	BOOT_DONE_EINT2	0	0	0	0	0	0	0	0000h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R6401 (1901h)	IRQ2_Status_2	FLL_AO_REF_LOST_EINT2	DSPCLK_ERR_EINT2	0	SYSCLK_ERR_EINT2	FLL_AO_LOCK_EINT2	0	0	FLL1_LOCK_EINT2	0	0	0	0	0	0	0	0	0000h
R6405 (1905h)	IRQ2_Status_6	0	0	0	0	0	0	MICDET2_EINT2	MICDET1_EINT2	0	0	0	0	0	0	0	HPDET_EINT2	0000h
R6406 (1906h)	IRQ2_Status_7	0	0	0	0	0	0	0	0	0	0	MICD_CLAMP_FALL_EINT2	MICD_CLAMP_RISE_EINT2	JD2_FALL_EINT2	JD2_RISE_EINT2	JD1_FALL_EINT2	JD1_RISE_EINT2	0000h
R6408 (1908h)	IRQ2_Status_9	0	0	0	0	0	0	0	0	0	0	0	0	0	INPUTS_SIG_DET_EINT2	DRC2_SIG_DET_EINT2	DRC1_SIG_DET_EINT2	0000h
R6410 (190Ah)	IRQ2_Status_11	DSP_IRQ16_EINT2	DSP_IRQ15_EINT2	DSP_IRQ14_EINT2	DSP_IRQ13_EINT2	DSP_IRQ12_EINT2	DSP_IRQ11_EINT2	DSP_IRQ10_EINT2	DSP_IRQ9_EINT2	DSP_IRQ8_EINT2	DSP_IRQ7_EINT2	DSP_IRQ6_EINT2	DSP_IRQ5_EINT2	DSP_IRQ4_EINT2	DSP_IRQ3_EINT2	DSP_IRQ2_EINT2	DSP_IRQ1_EINT2	0000h
R6411 (190Bh)	IRQ2_Status_12	0	0	0	0	0	0	0	0	0	SPKOUTL_SC_EINT2	0	0	HP2R_SC_EINT2	HP2L_SC_EINT2	HP1R_SC_EINT2	HP1L_SC_EINT2	0000h
R6412 (190Ch)	IRQ2_Status_13	0	0	0	0	0	0	0	0	0	SPKOUTL_ENABLE_DONE_EINT2	0	0	0	0	HP1R_ENABLE_DONE_EINT2	HP1L_ENABLE_DONE_EINT2	0000h
R6413 (190Dh)	IRQ2_Status_14	0	0	0	0	0	0	0	0	0	SPKOUTL_DISABLE_DONE_EINT2	0	0	0	0	HP1R_DISABLE_DONE_EINT2	HP1L_DISABLE_DONE_EINT2	0000h
R6414 (190Eh)	IRQ2_Status_15	0	0	0	0	0	0	0	0	0	0	0	0	0	SPK_OVERHEAT_WARN_EINT2	SPK_OVERHEAT_EINT2	SPK_SHUTDOWN_EINT2	0000h
R6415 (190Fh)	IRQ2_Status_16	0	0	0	0	0	0	0	0	MIF4_OVERCLOCKED_EINT2	0	0	0	0	0	0	0	0000h
R6416 (1910h)	IRQ2_Status_17	0	GP15_EINT2	GP14_EINT2	GP13_EINT2	GP12_EINT2	GP11_EINT2	GP10_EINT2	GP9_EINT2	GP8_EINT2	GP7_EINT2	GP6_EINT2	GP5_EINT2	GP4_EINT2	GP3_EINT2	GP2_EINT2	GP1_EINT2	0000h
R6420 (1914h)	IRQ2_Status_21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TIMER2_EINT2	TIMER1_EINT2	0000h
R6421 (1915h)	IRQ2_Status_22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT2_NOT_EMPTY_EINT2	EVENT1_NOT_EMPTY_EINT2	0000h
R6422 (1916h)	IRQ2_Status_23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT2_FULL_EINT2	EVENT1_FULL_EINT2	0000h
R6423 (1917h)	IRQ2_Status_24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT2_WMARK_EINT2	EVENT1_WMARK_EINT2	0000h
R6424 (1918h)	IRQ2_Status_25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_DMA_EINT2	0000h
R6426 (191Ah)	IRQ2_Status_27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_START1_EINT2	0000h
R6427 (191Bh)	IRQ2_Status_28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_START2_EINT2	0000h
R6429 (191Dh)	IRQ2_Status_30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_BUSY_EINT2	0000h
R6430 (191Eh)	IRQ2_Status_31	0	0	0	0	0	0	0	0	0	0	0	0	MIF4_DONE_EINT2	0	0	0	0000h
R6431 (191Fh)	IRQ2_Status_32	0	0	0	0	0	0	0	0	0	0	0	0	MIF4_BLOCK_EINT2	0	0	0	0000h
R6432 (1920h)	IRQ2_Status_33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_BUS_ERR_EINT2	0000h
R6464 (1940h)	IRQ2_Mask_1	0	0	0	IM_CTRLIF_ERR_EINT2	0	0	IM_SYSCLK_FAIL_EINT2	0	IM_BOOT_DONE_EINT2	0	0	0	0	0	0	0	1280h
R6465 (1941h)	IRQ2_Mask_2	IM_FLL_AO_REF_LOST_EINT2	IM_DSPCLK_ERR_EINT2	0	IM_SYSCLK_ERR_EINT2	IM_FLL_AO_LOCK_EINT2	0	0	IM_FLL1_LOCK_EINT2	0	0	0	0	0	0	0	0	D900h
R6469 (1945h)	IRQ2_Mask_6	0	0	0	0	0	0	IM_MICDET2_EINT2	IM_MICDET1_EINT2	0	0	0	0	0	0	0	IM_HPDET_EINT2	0301h
R6470 (1946h)	IRQ2_Mask_7	0	0	0	0	0	0	0	0	0	0	IM_MICD_CLAMP_FALL_EINT2	IM_MICD_CLAMP_RISE_EINT2	IM_JD2_FALL_EINT2	IM_JD2_RISE_EINT2	IM_JD1_FALL_EINT2	IM_JD1_RISE_EINT2	003Fh
R6472 (1948h)	IRQ2_Mask_9	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_INPUTS_SIG_DET_EINT2	IM_DRC2_SIG_DET_EINT2	IM_DRC1_SIG_DET_EINT2	0007h
R6474 (194Ah)	IRQ2_Mask_11	IM_DSP_IRQ16_EINT2	IM_DSP_IRQ15_EINT2	IM_DSP_IRQ14_EINT2	IM_DSP_IRQ13_EINT2	IM_DSP_IRQ12_EINT2	IM_DSP_IRQ11_EINT2	IM_DSP_IRQ10_EINT2	IM_DSP_IRQ9_EINT2	IM_DSP_IRQ8_EINT2	IM_DSP_IRQ7_EINT2	IM_DSP_IRQ6_EINT2	IM_DSP_IRQ5_EINT2	IM_DSP_IRQ4_EINT2	IM_DSP_IRQ3_EINT2	IM_DSP_IRQ2_EINT2	IM_DSP_IRQ1_EINT2	FFFFh

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R6475 (194Bh)	IRQ2_Mask_12	0	0	0	0	0	0	0	0	0	IM_SPKOUTL_SC_EINT2	0	0	IM_HP2R_SC_EINT2	IM_HP2L_SC_EINT2	IM_HP1R_SC_EINT2	IM_HP1L_SC_EINT2	004Fh
R6476 (194Ch)	IRQ2_Mask_13	0	0	0	0	0	0	0	0	0	IM_SPKOUTL_ENABLE_DONE_EINT2	0	0	0	0	IM_HP1R_ENABLE_DONE_EINT2	IM_HP1L_ENABLE_DONE_EINT2	0043h
R6477 (194Dh)	IRQ2_Mask_14	0	0	0	0	0	0	0	0	0	IM_SPKOUTL_DISABLE_DONE_EINT2	0	0	0	0	IM_HP1R_DISABLE_DONE_EINT2	IM_HP1L_DISABLE_DONE_EINT2	0043h
R6478 (194Eh)	IRQ2_Mask_15	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_SPK_OVERHEAT_WARN_EINT2	IM_SPK_OVERHEAT_EINT2	IM_SPK_SHUTDOWN_EINT2	0007h
R6479 (194Fh)	IRQ2_Mask_16	0	0	0	0	0	0	0	0	IM_MIF4_OVERCLOCKED_EINT2	0	0	0	0	0	0	0	0080h
R6480 (1950h)	IRQ2_Mask_17	0	IM_GP15_EINT2	IM_GP14_EINT2	IM_GP13_EINT2	IM_GP12_EINT2	IM_GP11_EINT2	IM_GP10_EINT2	IM_GP9_EINT2	IM_GP8_EINT2	IM_GP7_EINT2	IM_GP6_EINT2	IM_GP5_EINT2	IM_GP4_EINT2	IM_GP3_EINT2	IM_GP2_EINT2	IM_GP1_EINT2	7FFFh
R6484 (1954h)	IRQ2_Mask_21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_TIMER2_EINT2	IM_TIMER1_EINT2	0003h
R6485 (1955h)	IRQ2_Mask_22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_EVENT2_NOT_EMPTY_EINT2	IM_EVENT1_NOT_EMPTY_EINT2	0003h
R6486 (1956h)	IRQ2_Mask_23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_EVENT2_FULL_EINT2	IM_EVENT1_FULL_EINT2	0003h
R6487 (1957h)	IRQ2_Mask_24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_EVENT2_WMARK_EINT2	IM_EVENT1_WMARK_EINT2	0003h
R6488 (1958h)	IRQ2_Mask_25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP1_DMA_EINT2	0001h
R6490 (195Ah)	IRQ2_Mask_27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP1_START1_EINT2	0001h
R6491 (195Bh)	IRQ2_Mask_28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP1_START2_EINT2	0001h
R6493 (195Dh)	IRQ2_Mask_30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP1_BUSY_EINT2	0001h
R6494 (195Eh)	IRQ2_Mask_31	0	0	0	0	0	0	0	0	0	0	0	0	IM_MIF4_DONE_EINT2	0	0	0	0008h
R6495 (195Fh)	IRQ2_Mask_32	0	0	0	0	0	0	0	0	0	0	0	0	IM_MIF4_BLOCK_EINT2	0	0	0	0008h
R6496 (1960h)	IRQ1_Mask_33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP1_BUS_ERR_EINT2	0000h
R6528 (1980h)	IRQ2_Raw_Status_1	0	0	0	CTRLIF_ERR_STS2	0	0	0	0	BOOT_DONE_STS2	0	0	0	0	0	0	0	0000h
R6529 (1981h)	IRQ2_Raw_Status_2	FLL_AO_REF_LOST_STS2	DSPCLK_ERR_STS2	0	SYSCLK_ERR_STS2	FLL_AO_LOCK_STS2	0	0	FLL1_LOCK_STS2	0	0	0	0	0	0	0	0	0000h
R6534 (1986h)	IRQ2_Raw_Status_7	0	0	0	0	0	0	0	0	0	0	0	MICD_CLAMP_STS2	0	JD2_STS2	0	JD1_STS2	0000h
R6536 (1988h)	IRQ2_Raw_Status_9	0	0	0	0	0	0	0	0	0	0	0	0	0	INPUTS_SIG_DET_STS2	DRC2_SIG_DET_STS2	DRC1_SIG_DET_STS2	0000h
R6539 (198Bh)	IRQ2_Raw_Status_12	0	0	0	0	0	0	0	0	0	SPKOUTL_SC_STS2	0	0	HP2R_SC_STS2	HP2L_SC_STS2	HP1R_SC_STS2	HP1L_SC_STS2	0000h
R6540 (198Ch)	IRQ2_Raw_Status_13	0	0	0	0	0	0	0	0	0	SPKOUTL_ENABLE_DONE_STS2	0	0	0	0	HP1R_ENABLE_DONE_STS2	HP1L_ENABLE_DONE_STS2	0000h
R6541 (198Dh)	IRQ2_Raw_Status_14	0	0	0	0	0	0	0	0	0	SPKOUTL_DISABLE_DONE_STS2	0	0	0	0	HP1R_DISABLE_DONE_STS2	HP1L_DISABLE_DONE_STS2	0000h
R6542 (198Eh)	IRQ2_Raw_Status_15	0	0	0	0	0	0	0	0	0	0	0	0	0	SPK_OVERHEAT_WARN_STS2	SPK_OVERHEAT_STS2	SPK_SHUTDOWN_STS2	0000h
R6543 (198Fh)	IRQ2_Raw_Status_16	0	0	0	0	0	0	0	0	MIF4_OVERCLOCKED_STS2	0	0	0	0	0	0	0	0000h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R6544 (1990h)	IRQ2_Raw_Status_17	0	GPIO15_STS2	GPIO14_STS2	GPIO13_STS2	GPIO12_STS2	GPIO11_STS2	GPIO10_STS2	GPIO9_STS2	GPIO8_STS2	GPIO7_STS2	GPIO6_STS2	GPIO5_STS2	GPIO4_STS2	GPIO3_STS2	GPIO2_STS2	GPIO1_STS2	0000h
R6549 (1995h)	IRQ2_Raw_Status_22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT2_NOT_EMPTY_STS2	EVENT1_NOT_EMPTY_STS2	0000h
R6550 (1996h)	IRQ2_Raw_Status_23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT2_FULL_STS2	EVENT1_FULL_STS2	0000h
R6551 (1997h)	IRQ2_Raw_Status_24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT2_WMARK_STS2	EVENT1_WMARK_STS2	0000h
R6552 (1998h)	IRQ2_Raw_Status_25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_DMA_STS2	0000h
R6557 (199Dh)	IRQ2_Raw_Status_30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_BUSY_STS2	0000h
R6558 (199Eh)	IRQ2_Raw_Status_31	0	0	0	0	0	0	0	0	0	0	0	0	MIF4_DONE_STS2	0	0	0	0000h
R6559 (199Fh)	IRQ2_Raw_Status_32	0	0	0	0	0	0	0	0	0	0	0	0	MIF4_BLOCK_STS2	0	0	0	0000h
R6662 (1A06h)	Interrupt_Debounce_7	0	0	0	0	0	0	0	0	0	0	0	MICD_CLAMP_DB	0	JD2_DB	0	JD1_DB	0000h
R6784 (1A80h)	IRQ1_CTRL	0	1	0	0	IM_IRQ1	IRQ_POL	IRQ_OP_CFG	0	0	0	0	0	0	0	0	0	4400h
R6786 (1A82h)	IRQ2_CTRL	0	0	0	0	IM_IRQ2	0	0	0	0	0	0	0	0	0	0	0	0000h
R6816 (1AA0h)	Interrupt_Raw_Status_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IRQ2_STS	IRQ1_STS	0000h
R6848 (1AC0h)	GPIO_Debounce_Config	0	0	0	0	0	0	0	0	0	0	0	0	GP_DBTIME [3:0]				0000h
R6864 (1AD0h)	AOD_Pad_Ctrl	0	1	0	0	0	0	0	0	0	0	0	0	0	0	RESET_PU	RESET_PD	4002h

The 32-bit DSP register space is described in Table 6-2.

Table 6-2. Register Map Definition—32-bit region

Register	Name	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0	Default
R12288 (3000h)	WSEQ_Sequence_1	WSEQ_DATA_WIDTH0 [2:0]			WSEQ_ADDR0 [12:0]													0000F000h
		WSEQ_DELAY0 [3:0]				WSEQ_DATA_START0 [3:0]				WSEQ_DATA0 [7:0]								
R12290 (3002h)	WSEQ_Sequence_2	WSEQ_DATA_WIDTH1 [2:0]			WSEQ_ADDR1 [12:0]													0000F000h
		WSEQ_DELAY1 [3:0]				WSEQ_DATA_START1 [3:0]				WSEQ_DATA1 [7:0]								
R12292 (3004h)	WSEQ_Sequence_3	WSEQ_DATA_WIDTH2 [2:0]			WSEQ_ADDR2 [12:0]													0000F000h
		WSEQ_DELAY2 [3:0]				WSEQ_DATA_START2 [3:0]				WSEQ_DATA2 [7:0]								
R12294 (3006h)	WSEQ_Sequence_4	WSEQ_DATA_WIDTH3 [2:0]			WSEQ_ADDR3 [12:0]													0000F000h
		WSEQ_DELAY3 [3:0]				WSEQ_DATA_START3 [3:0]				WSEQ_DATA3 [7:0]								
R12296 (3008h)	WSEQ_Sequence_5	WSEQ_DATA_WIDTH4 [2:0]			WSEQ_ADDR4 [12:0]													82253719h
		WSEQ_DELAY4 [3:0]				WSEQ_DATA_START4 [3:0]				WSEQ_DATA4 [7:0]								
R12298 (300Ah)	WSEQ_Sequence_6	WSEQ_DATA_WIDTH5 [2:0]			WSEQ_ADDR5 [12:0]													C2300001h
		WSEQ_DELAY5 [3:0]				WSEQ_DATA_START5 [3:0]				WSEQ_DATA5 [7:0]								
R12300 (300Ch)	WSEQ_Sequence_7	WSEQ_DATA_WIDTH6 [2:0]			WSEQ_ADDR6 [12:0]													02251301h
		WSEQ_DELAY6 [3:0]				WSEQ_DATA_START6 [3:0]				WSEQ_DATA6 [7:0]								
R12302 (300Eh)	WSEQ_Sequence_8	WSEQ_DATA_WIDTH7 [2:0]			WSEQ_ADDR7 [12:0]													8225191Fh
		WSEQ_DELAY7 [3:0]				WSEQ_DATA_START7 [3:0]				WSEQ_DATA7 [7:0]								
R12304 (3010h)	WSEQ_Sequence_9	WSEQ_DATA_WIDTH8 [2:0]			WSEQ_ADDR8 [12:0]													82310B00h
		WSEQ_DELAY8 [3:0]				WSEQ_DATA_START8 [3:0]				WSEQ_DATA8 [7:0]								
R12306 (3012h)	WSEQ_Sequence_10	WSEQ_DATA_WIDTH9 [2:0]			WSEQ_ADDR9 [12:0]													E231023Bh
		WSEQ_DELAY9 [3:0]				WSEQ_DATA_START9 [3:0]				WSEQ_DATA9 [7:0]								
R12308 (3014h)	WSEQ_Sequence_11	WSEQ_DATA_WIDTH10 [2:0]			WSEQ_ADDR10 [12:0]													02313B01h
		WSEQ_DELAY10 [3:0]				WSEQ_DATA_START10 [3:0]				WSEQ_DATA10 [7:0]								
R12310 (3016h)	WSEQ_Sequence_12	WSEQ_DATA_WIDTH11 [2:0]			WSEQ_ADDR11 [12:0]													62300000h
		WSEQ_DELAY11 [3:0]				WSEQ_DATA_START11 [3:0]				WSEQ_DATA11 [7:0]								
R12312 (3018h)	WSEQ_Sequence_13	WSEQ_DATA_WIDTH12 [2:0]			WSEQ_ADDR12 [12:0]													E2314288h
		WSEQ_DELAY12 [3:0]				WSEQ_DATA_START12 [3:0]				WSEQ_DATA12 [7:0]								
R12314 (301Ah)	WSEQ_Sequence_14	WSEQ_DATA_WIDTH13 [2:0]			WSEQ_ADDR13 [12:0]													02310B00h
		WSEQ_DELAY13 [3:0]				WSEQ_DATA_START13 [3:0]				WSEQ_DATA13 [7:0]								
R12316 (301Ch)	WSEQ_Sequence_15	WSEQ_DATA_WIDTH14 [2:0]			WSEQ_ADDR14 [12:0]													02310B00h
		WSEQ_DELAY14 [3:0]				WSEQ_DATA_START14 [3:0]				WSEQ_DATA14 [7:0]								
R12318 (301Eh)	WSEQ_Sequence_16	WSEQ_DATA_WIDTH15 [2:0]			WSEQ_ADDR15 [12:0]													02250E01h
		WSEQ_DELAY15 [3:0]				WSEQ_DATA_START15 [3:0]				WSEQ_DATA15 [7:0]								
R12320 (3020h)	WSEQ_Sequence_17	WSEQ_DATA_WIDTH16 [2:0]			WSEQ_ADDR16 [12:0]													42310C02h
		WSEQ_DELAY16 [3:0]				WSEQ_DATA_START16 [3:0]				WSEQ_DATA16 [7:0]								

Table 6-2. Register Map Definition—32-bit region *(Cont.)*

Register	Name	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0	Default
R12322 (3022h)	WSEQ_Sequence_18	WSEQ_DATA_WIDTH17 [2:0]			WSEQ_ADDR17 [12:0]													E2310227h
		WSEQ_DELAY17 [3:0]				WSEQ_DATA_START17 [3:0]				WSEQ_DATA17 [7:0]								
R12324 (3024h)	WSEQ_Sequence_19	WSEQ_DATA_WIDTH18 [2:0]			WSEQ_ADDR18 [12:0]													02313B01h
		WSEQ_DELAY18 [3:0]				WSEQ_DATA_START18 [3:0]				WSEQ_DATA18 [7:0]								
R12326 (3026h)	WSEQ_Sequence_20	WSEQ_DATA_WIDTH19 [2:0]			WSEQ_ADDR19 [12:0]													E2314266h
		WSEQ_DELAY19 [3:0]				WSEQ_DATA_START19 [3:0]				WSEQ_DATA19 [7:0]								
R12328 (3028h)	WSEQ_Sequence_21	WSEQ_DATA_WIDTH20 [2:0]			WSEQ_ADDR20 [12:0]													E2315294h
		WSEQ_DELAY20 [3:0]				WSEQ_DATA_START20 [3:0]				WSEQ_DATA20 [7:0]								
R12330 (302Ah)	WSEQ_Sequence_22	WSEQ_DATA_WIDTH21 [2:0]			WSEQ_ADDR21 [12:0]													02310B00h
		WSEQ_DELAY21 [3:0]				WSEQ_DATA_START21 [3:0]				WSEQ_DATA21 [7:0]								
R12332 (302Ch)	WSEQ_Sequence_23	WSEQ_DATA_WIDTH22 [2:0]			WSEQ_ADDR22 [12:0]													02310B00h
		WSEQ_DELAY22 [3:0]				WSEQ_DATA_START22 [3:0]				WSEQ_DATA22 [7:0]								
R12334 (302Eh)	WSEQ_Sequence_24	WSEQ_DATA_WIDTH23 [2:0]			WSEQ_ADDR23 [12:0]													E2251734h
		WSEQ_DELAY23 [3:0]				WSEQ_DATA_START23 [3:0]				WSEQ_DATA23 [7:0]								
R12336 (3030h)	WSEQ_Sequence_25	WSEQ_DATA_WIDTH24 [2:0]			WSEQ_ADDR24 [12:0]													0225F501h
		WSEQ_DELAY24 [3:0]				WSEQ_DATA_START24 [3:0]				WSEQ_DATA24 [7:0]								
R12338 (3032h)	WSEQ_Sequence_26	WSEQ_DATA_WIDTH25 [2:0]			WSEQ_ADDR25 [12:0]													0000F000h
		WSEQ_DELAY25 [3:0]				WSEQ_DATA_START25 [3:0]				WSEQ_DATA25 [7:0]								
R12340 (3034h)	WSEQ_Sequence_27	WSEQ_DATA_WIDTH26 [2:0]			WSEQ_ADDR26 [12:0]													0000F000h
		WSEQ_DELAY26 [3:0]				WSEQ_DATA_START26 [3:0]				WSEQ_DATA26 [7:0]								
R12342 (3036h)	WSEQ_Sequence_28	WSEQ_DATA_WIDTH27 [2:0]			WSEQ_ADDR27 [12:0]													0000F000h
		WSEQ_DELAY27 [3:0]				WSEQ_DATA_START27 [3:0]				WSEQ_DATA27 [7:0]								
R12344 (3038h)	WSEQ_Sequence_29	WSEQ_DATA_WIDTH28 [2:0]			WSEQ_ADDR28 [12:0]													0000F000h
		WSEQ_DELAY28 [3:0]				WSEQ_DATA_START28 [3:0]				WSEQ_DATA28 [7:0]								
R12346 (303Ah)	WSEQ_Sequence_30	WSEQ_DATA_WIDTH29 [2:0]			WSEQ_ADDR29 [12:0]													0000F000h
		WSEQ_DELAY29 [3:0]				WSEQ_DATA_START29 [3:0]				WSEQ_DATA29 [7:0]								
R12348 (303Ch)	WSEQ_Sequence_31	WSEQ_DATA_WIDTH30 [2:0]			WSEQ_ADDR30 [12:0]													0000F000h
		WSEQ_DELAY30 [3:0]				WSEQ_DATA_START30 [3:0]				WSEQ_DATA30 [7:0]								
R12350 (303Eh)	WSEQ_Sequence_32	WSEQ_DATA_WIDTH31 [2:0]			WSEQ_ADDR31 [12:0]													02253A01h
		WSEQ_DELAY31 [3:0]				WSEQ_DATA_START31 [3:0]				WSEQ_DATA31 [7:0]								
R12352 (3040h)	WSEQ_Sequence_33	WSEQ_DATA_WIDTH32 [2:0]			WSEQ_ADDR32 [12:0]													C2251300h
		WSEQ_DELAY32 [3:0]				WSEQ_DATA_START32 [3:0]				WSEQ_DATA32 [7:0]								
R12354 (3042h)	WSEQ_Sequence_34	WSEQ_DATA_WIDTH33 [2:0]			WSEQ_ADDR33 [12:0]													02250B00h
		WSEQ_DELAY33 [3:0]				WSEQ_DATA_START33 [3:0]				WSEQ_DATA33 [7:0]								
R12356 (3044h)	WSEQ_Sequence_35	WSEQ_DATA_WIDTH34 [2:0]			WSEQ_ADDR34 [12:0]													0225FF01h
		WSEQ_DELAY34 [3:0]				WSEQ_DATA_START34 [3:0]				WSEQ_DATA34 [7:0]								
R12358 (3046h)	WSEQ_Sequence_36	WSEQ_DATA_WIDTH35 [2:0]			WSEQ_ADDR35 [12:0]													0000F000h
		WSEQ_DELAY35 [3:0]				WSEQ_DATA_START35 [3:0]				WSEQ_DATA35 [7:0]								
R12360 (3048h)	WSEQ_Sequence_37	WSEQ_DATA_WIDTH36 [2:0]			WSEQ_ADDR36 [12:0]													0000F000h
		WSEQ_DELAY36 [3:0]				WSEQ_DATA_START36 [3:0]				WSEQ_DATA36 [7:0]								
R12362 (304Ah)	WSEQ_Sequence_38	WSEQ_DATA_WIDTH37 [2:0]			WSEQ_ADDR37 [12:0]													0000F000h
		WSEQ_DELAY37 [3:0]				WSEQ_DATA_START37 [3:0]				WSEQ_DATA37 [7:0]								
R12364 (304Ch)	WSEQ_Sequence_39	WSEQ_DATA_WIDTH38 [2:0]			WSEQ_ADDR38 [12:0]													0000F000h
		WSEQ_DELAY38 [3:0]				WSEQ_DATA_START38 [3:0]				WSEQ_DATA38 [7:0]								
R12366 (304Eh)	WSEQ_Sequence_40	WSEQ_DATA_WIDTH39 [2:0]			WSEQ_ADDR39 [12:0]													0000F000h
		WSEQ_DELAY39 [3:0]				WSEQ_DATA_START39 [3:0]				WSEQ_DATA39 [7:0]								
R12368 (3050h)	WSEQ_Sequence_41	WSEQ_DATA_WIDTH40 [2:0]			WSEQ_ADDR40 [12:0]													0000F000h
		WSEQ_DELAY40 [3:0]				WSEQ_DATA_START40 [3:0]				WSEQ_DATA40 [7:0]								
R12370 (3052h)	WSEQ_Sequence_42	WSEQ_DATA_WIDTH41 [2:0]			WSEQ_ADDR41 [12:0]													0000F000h
		WSEQ_DELAY41 [3:0]				WSEQ_DATA_START41 [3:0]				WSEQ_DATA41 [7:0]								
R12372 (3054h)	WSEQ_Sequence_43	WSEQ_DATA_WIDTH42 [2:0]			WSEQ_ADDR42 [12:0]													0000F000h
		WSEQ_DELAY42 [3:0]				WSEQ_DATA_START42 [3:0]				WSEQ_DATA42 [7:0]								
R12374 (3056h)	WSEQ_Sequence_44	WSEQ_DATA_WIDTH43 [2:0]			WSEQ_ADDR43 [12:0]													0000F000h
		WSEQ_DELAY43 [3:0]				WSEQ_DATA_START43 [3:0]				WSEQ_DATA43 [7:0]								
R12376 (3058h)	WSEQ_Sequence_45	WSEQ_DATA_WIDTH44 [2:0]			WSEQ_ADDR44 [12:0]													82263719h
		WSEQ_DELAY44 [3:0]				WSEQ_DATA_START44 [3:0]				WSEQ_DATA44 [7:0]								
R12378 (305Ah)	WSEQ_Sequence_46	WSEQ_DATA_WIDTH45 [2:0]			WSEQ_ADDR45 [12:0]													C2300001h
		WSEQ_DELAY45 [3:0]				WSEQ_DATA_START45 [3:0]				WSEQ_DATA45 [7:0]								
R12380 (305Ch)	WSEQ_Sequence_47	WSEQ_DATA_WIDTH46 [2:0]			WSEQ_ADDR46 [12:0]													02261301h
		WSEQ_DELAY46 [3:0]				WSEQ_DATA_START46 [3:0]				WSEQ_DATA46 [7:0]								
R12382 (305Eh)	WSEQ_Sequence_48	WSEQ_DATA_WIDTH47 [2:0]			WSEQ_ADDR47 [12:0]													8226191Fh
		WSEQ_DELAY47 [3:0]				WSEQ_DATA_START47 [3:0]				WSEQ_DATA47 [7:0]								
R12384 (3060h)	WSEQ_Sequence_49	WSEQ_DATA_WIDTH48 [2:0]			WSEQ_ADDR48 [12:0]													82310B02h
		WSEQ_DELAY48 [3:0]				WSEQ_DATA_START48 [3:0]				WSEQ_DATA48 [7:0]								
R12386 (3062h)	WSEQ_Sequence_50	WSEQ_DATA_WIDTH49 [2:0]			WSEQ_ADDR49 [12:0]													E231023Bh
		WSEQ_DELAY49 [3:0]				WSEQ_DATA_START49 [3:0]				WSEQ_DATA49 [7:0]								
R12388 (3064h)	WSEQ_Sequence_51	WSEQ_DATA_WIDTH50 [2:0]			WSEQ_ADDR50 [12:0]													02313B01h
		WSEQ_DELAY50 [3:0]				WSEQ_DATA_START50 [3:0]				WSEQ_DATA50 [7:0]								
R12390 (3066h)	WSEQ_Sequence_52	WSEQ_DATA_WIDTH51 [2:0]			WSEQ_ADDR51 [12:0]													62300000h
		WSEQ_DELAY51 [3:0]				WSEQ_DATA_START51 [3:0]				WSEQ_DATA51 [7:0]								
R12392 (3068h)	WSEQ_Sequence_53	WSEQ_DATA_WIDTH52 [2:0]			WSEQ_ADDR52 [12:0]													E2314288h
		WSEQ_DELAY52 [3:0]				WSEQ_DATA_START52 [3:0]				WSEQ_DATA52 [7:0]								

Table 6-2. Register Map Definition—32-bit region *(Cont.)*

Register	Name	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0	Default
R12394 (306Ah)	WSEQ_Sequence_54	WSEQ_DATA_WIDTH53 [2:0]			WSEQ_ADDR53 [12:0]													02310B00h
		WSEQ_DELAY53 [3:0]				WSEQ_DATA_START53 [3:0]				WSEQ_DATA53 [7:0]								
R12396 (306Ch)	WSEQ_Sequence_55	WSEQ_DATA_WIDTH54 [2:0]			WSEQ_ADDR54 [12:0]													02310B00h
		WSEQ_DELAY54 [3:0]				WSEQ_DATA_START54 [3:0]				WSEQ_DATA54 [7:0]								
R12398 (306Eh)	WSEQ_Sequence_56	WSEQ_DATA_WIDTH55 [2:0]			WSEQ_ADDR55 [12:0]													02260E01h
		WSEQ_DELAY55 [3:0]				WSEQ_DATA_START55 [3:0]				WSEQ_DATA55 [7:0]								
R12400 (3070h)	WSEQ_Sequence_57	WSEQ_DATA_WIDTH56 [2:0]			WSEQ_ADDR56 [12:0]													42310C03h
		WSEQ_DELAY56 [3:0]				WSEQ_DATA_START56 [3:0]				WSEQ_DATA56 [7:0]								
R12402 (3072h)	WSEQ_Sequence_58	WSEQ_DATA_WIDTH57 [2:0]			WSEQ_ADDR57 [12:0]													E2310227h
		WSEQ_DELAY57 [3:0]				WSEQ_DATA_START57 [3:0]				WSEQ_DATA57 [7:0]								
R12404 (3074h)	WSEQ_Sequence_59	WSEQ_DATA_WIDTH58 [2:0]			WSEQ_ADDR58 [12:0]													02313B01h
		WSEQ_DELAY58 [3:0]				WSEQ_DATA_START58 [3:0]				WSEQ_DATA58 [7:0]								
R12406 (3076h)	WSEQ_Sequence_60	WSEQ_DATA_WIDTH59 [2:0]			WSEQ_ADDR59 [12:0]													E2314266h
		WSEQ_DELAY59 [3:0]				WSEQ_DATA_START59 [3:0]				WSEQ_DATA59 [7:0]								
R12408 (3078h)	WSEQ_Sequence_61	WSEQ_DATA_WIDTH60 [2:0]			WSEQ_ADDR60 [12:0]													E2315294h
		WSEQ_DELAY60 [3:0]				WSEQ_DATA_START60 [3:0]				WSEQ_DATA60 [7:0]								
R12410 (307Ah)	WSEQ_Sequence_62	WSEQ_DATA_WIDTH61 [2:0]			WSEQ_ADDR61 [12:0]													02310B00h
		WSEQ_DELAY61 [3:0]				WSEQ_DATA_START61 [3:0]				WSEQ_DATA61 [7:0]								
R12412 (307Ch)	WSEQ_Sequence_63	WSEQ_DATA_WIDTH62 [2:0]			WSEQ_ADDR62 [12:0]													02310B00h
		WSEQ_DELAY62 [3:0]				WSEQ_DATA_START62 [3:0]				WSEQ_DATA62 [7:0]								
R12414 (307Eh)	WSEQ_Sequence_64	WSEQ_DATA_WIDTH63 [2:0]			WSEQ_ADDR63 [12:0]													E2261734h
		WSEQ_DELAY63 [3:0]				WSEQ_DATA_START63 [3:0]				WSEQ_DATA63 [7:0]								
R12416 (3080h)	WSEQ_Sequence_65	WSEQ_DATA_WIDTH64 [2:0]			WSEQ_ADDR64 [12:0]													0226F501h
		WSEQ_DELAY64 [3:0]				WSEQ_DATA_START64 [3:0]				WSEQ_DATA64 [7:0]								
R12418 (3082h)	WSEQ_Sequence_66	WSEQ_DATA_WIDTH65 [2:0]			WSEQ_ADDR65 [12:0]													0000F000h
		WSEQ_DELAY65 [3:0]				WSEQ_DATA_START65 [3:0]				WSEQ_DATA65 [7:0]								
R12420 (3084h)	WSEQ_Sequence_67	WSEQ_DATA_WIDTH66 [2:0]			WSEQ_ADDR66 [12:0]													0000F000h
		WSEQ_DELAY66 [3:0]				WSEQ_DATA_START66 [3:0]				WSEQ_DATA66 [7:0]								
R12422 (3086h)	WSEQ_Sequence_68	WSEQ_DATA_WIDTH67 [2:0]			WSEQ_ADDR67 [12:0]													0000F000h
		WSEQ_DELAY67 [3:0]				WSEQ_DATA_START67 [3:0]				WSEQ_DATA67 [7:0]								
R12424 (3088h)	WSEQ_Sequence_69	WSEQ_DATA_WIDTH68 [2:0]			WSEQ_ADDR68 [12:0]													0000F000h
		WSEQ_DELAY68 [3:0]				WSEQ_DATA_START68 [3:0]				WSEQ_DATA68 [7:0]								
R12426 (308Ah)	WSEQ_Sequence_70	WSEQ_DATA_WIDTH69 [2:0]			WSEQ_ADDR69 [12:0]													0000F000h
		WSEQ_DELAY69 [3:0]				WSEQ_DATA_START69 [3:0]				WSEQ_DATA69 [7:0]								
R12428 (308Ch)	WSEQ_Sequence_71	WSEQ_DATA_WIDTH70 [2:0]			WSEQ_ADDR70 [12:0]													0000F000h
		WSEQ_DELAY70 [3:0]				WSEQ_DATA_START70 [3:0]				WSEQ_DATA70 [7:0]								
R12430 (308Eh)	WSEQ_Sequence_72	WSEQ_DATA_WIDTH71 [2:0]			WSEQ_ADDR71 [12:0]													02263A01h
		WSEQ_DELAY71 [3:0]				WSEQ_DATA_START71 [3:0]				WSEQ_DATA71 [7:0]								
R12432 (3090h)	WSEQ_Sequence_73	WSEQ_DATA_WIDTH72 [2:0]			WSEQ_ADDR72 [12:0]													C2261300h
		WSEQ_DELAY72 [3:0]				WSEQ_DATA_START72 [3:0]				WSEQ_DATA72 [7:0]								
R12434 (3092h)	WSEQ_Sequence_74	WSEQ_DATA_WIDTH73 [2:0]			WSEQ_ADDR73 [12:0]													02260B00h
		WSEQ_DELAY73 [3:0]				WSEQ_DATA_START73 [3:0]				WSEQ_DATA73 [7:0]								
R12436 (3094h)	WSEQ_Sequence_75	WSEQ_DATA_WIDTH74 [2:0]			WSEQ_ADDR74 [12:0]													0226FF01h
		WSEQ_DELAY74 [3:0]				WSEQ_DATA_START74 [3:0]				WSEQ_DATA74 [7:0]								
R12438 (3096h)	WSEQ_Sequence_76	WSEQ_DATA_WIDTH75 [2:0]			WSEQ_ADDR75 [12:0]													0000F000h
		WSEQ_DELAY75 [3:0]				WSEQ_DATA_START75 [3:0]				WSEQ_DATA75 [7:0]								
R12440 (3098h)	WSEQ_Sequence_77	WSEQ_DATA_WIDTH76 [2:0]			WSEQ_ADDR76 [12:0]													0000F000h
		WSEQ_DELAY76 [3:0]				WSEQ_DATA_START76 [3:0]				WSEQ_DATA76 [7:0]								
R12442 (309Ah)	WSEQ_Sequence_78	WSEQ_DATA_WIDTH77 [2:0]			WSEQ_ADDR77 [12:0]													0000F000h
		WSEQ_DELAY77 [3:0]				WSEQ_DATA_START77 [3:0]				WSEQ_DATA77 [7:0]								
R12444 (309Ch)	WSEQ_Sequence_79	WSEQ_DATA_WIDTH78 [2:0]			WSEQ_ADDR78 [12:0]													0000F000h
		WSEQ_DELAY78 [3:0]				WSEQ_DATA_START78 [3:0]				WSEQ_DATA78 [7:0]								
R12446 (309Eh)	WSEQ_Sequence_80	WSEQ_DATA_WIDTH79 [2:0]			WSEQ_ADDR79 [12:0]													0000F000h
		WSEQ_DELAY79 [3:0]				WSEQ_DATA_START79 [3:0]				WSEQ_DATA79 [7:0]								
R12448 (30A0h)	WSEQ_Sequence_81	WSEQ_DATA_WIDTH80 [2:0]			WSEQ_ADDR80 [12:0]													0000F000h
		WSEQ_DELAY80 [3:0]				WSEQ_DATA_START80 [3:0]				WSEQ_DATA80 [7:0]								
R12450 (30A2h)	WSEQ_Sequence_82	WSEQ_DATA_WIDTH81 [2:0]			WSEQ_ADDR81 [12:0]													0000F000h
		WSEQ_DELAY81 [3:0]				WSEQ_DATA_START81 [3:0]				WSEQ_DATA81 [7:0]								
R12452 (30A4h)	WSEQ_Sequence_83	WSEQ_DATA_WIDTH82 [2:0]			WSEQ_ADDR82 [12:0]													0000F000h
		WSEQ_DELAY82 [3:0]				WSEQ_DATA_START82 [3:0]				WSEQ_DATA82 [7:0]								
R12454 (30A6h)	WSEQ_Sequence_84	WSEQ_DATA_WIDTH83 [2:0]			WSEQ_ADDR83 [12:0]													0000F000h
		WSEQ_DELAY83 [3:0]				WSEQ_DATA_START83 [3:0]				WSEQ_DATA83 [7:0]								
R12456 (30A8h)	WSEQ_Sequence_85	WSEQ_DATA_WIDTH84 [2:0]			WSEQ_ADDR84 [12:0]													026D0101h
		WSEQ_DELAY84 [3:0]				WSEQ_DATA_START84 [3:0]				WSEQ_DATA84 [7:0]								
R12458 (30AAh)	WSEQ_Sequence_86	WSEQ_DATA_WIDTH85 [2:0]			WSEQ_ADDR85 [12:0]													44B00004h
		WSEQ_DELAY85 [3:0]				WSEQ_DATA_START85 [3:0]				WSEQ_DATA85 [7:0]								
R12460 (30ACh)	WSEQ_Sequence_87	WSEQ_DATA_WIDTH86 [2:0]			WSEQ_ADDR86 [12:0]													04020701h
		WSEQ_DELAY86 [3:0]				WSEQ_DATA_START86 [3:0]				WSEQ_DATA86 [7:0]								
R12462 (30AEh)	WSEQ_Sequence_88	WSEQ_DATA_WIDTH87 [2:0]			WSEQ_ADDR87 [12:0]													04AE5801h
		WSEQ_DELAY87 [3:0]				WSEQ_DATA_START87 [3:0]				WSEQ_DATA87 [7:0]								
R12464 (30B0h)	WSEQ_Sequence_89	WSEQ_DATA_WIDTH88 [2:0]			WSEQ_ADDR88 [12:0]													A4AE201Fh
		WSEQ_DELAY88 [3:0]				WSEQ_DATA_START88 [3:0]				WSEQ_DATA88 [7:0]								

Table 6-2. Register Map Definition—32-bit region *(Cont.)*

Register	Name	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0	Default
R12466 (30B2h)	WSEQ_Sequence_90	WSEQ_DATA_WIDTH89 [2:0]			WSEQ_ADDR89 [12:0]													A4AE201Fh
		WSEQ_DELAY89 [3:0]				WSEQ_DATA_START89 [3:0]				WSEQ_DATA89 [7:0]								
R12468 (30B4h)	WSEQ_Sequence_91	WSEQ_DATA_WIDTH90 [2:0]			WSEQ_ADDR90 [12:0]													A4AE301Dh
		WSEQ_DELAY90 [3:0]				WSEQ_DATA_START90 [3:0]				WSEQ_DATA90 [7:0]								
R12470 (30B6h)	WSEQ_Sequence_92	WSEQ_DATA_WIDTH91 [2:0]			WSEQ_ADDR91 [12:0]													A4AE203Ch
		WSEQ_DELAY91 [3:0]				WSEQ_DATA_START91 [3:0]				WSEQ_DATA91 [7:0]								
R12472 (30B8h)	WSEQ_Sequence_93	WSEQ_DATA_WIDTH92 [2:0]			WSEQ_ADDR92 [12:0]													A4AE303Ch
		WSEQ_DELAY92 [3:0]				WSEQ_DATA_START92 [3:0]				WSEQ_DATA92 [7:0]								
R12474 (30BAh)	WSEQ_Sequence_94	WSEQ_DATA_WIDTH93 [2:0]			WSEQ_ADDR93 [12:0]													026D4F01h
		WSEQ_DELAY93 [3:0]				WSEQ_DATA_START93 [3:0]				WSEQ_DATA93 [7:0]								
R12476 (30BCh)	WSEQ_Sequence_95	WSEQ_DATA_WIDTH94 [2:0]			WSEQ_ADDR94 [12:0]													026D0100h
		WSEQ_DELAY94 [3:0]				WSEQ_DATA_START94 [3:0]				WSEQ_DATA94 [7:0]								
R12478 (30BEh)	WSEQ_Sequence_96	WSEQ_DATA_WIDTH95 [2:0]			WSEQ_ADDR95 [12:0]													04B00200h
		WSEQ_DELAY95 [3:0]				WSEQ_DATA_START95 [3:0]				WSEQ_DATA95 [7:0]								
R12480 (30C0h)	WSEQ_Sequence_97	WSEQ_DATA_WIDTH96 [2:0]			WSEQ_ADDR96 [12:0]													04C7F101h
		WSEQ_DELAY96 [3:0]				WSEQ_DATA_START96 [3:0]				WSEQ_DATA96 [7:0]								
R12482 (30C2h)	WSEQ_Sequence_98	WSEQ_DATA_WIDTH97 [2:0]			WSEQ_ADDR97 [12:0]													0000F000h
		WSEQ_DELAY97 [3:0]				WSEQ_DATA_START97 [3:0]				WSEQ_DATA97 [7:0]								
R12484 (30C4h)	WSEQ_Sequence_99	WSEQ_DATA_WIDTH98 [2:0]			WSEQ_ADDR98 [12:0]													0000F000h
		WSEQ_DELAY98 [3:0]				WSEQ_DATA_START98 [3:0]				WSEQ_DATA98 [7:0]								
R12486 (30C6h)	WSEQ_Sequence_100	WSEQ_DATA_WIDTH99 [2:0]			WSEQ_ADDR99 [12:0]													0000F000h
		WSEQ_DELAY99 [3:0]				WSEQ_DATA_START99 [3:0]				WSEQ_DATA99 [7:0]								
R12488 (30C8h)	WSEQ_Sequence_101	WSEQ_DATA_WIDTH100 [2:0]			WSEQ_ADDR100 [12:0]													0000F000h
		WSEQ_DELAY100 [3:0]				WSEQ_DATA_START100 [3:0]				WSEQ_DATA100 [7:0]								
R12490 (30CAh)	WSEQ_Sequence_102	WSEQ_DATA_WIDTH101 [2:0]			WSEQ_ADDR101 [12:0]													0000F000h
		WSEQ_DELAY101 [3:0]				WSEQ_DATA_START101 [3:0]				WSEQ_DATA101 [7:0]								
R12492 (30CCh)	WSEQ_Sequence_103	WSEQ_DATA_WIDTH102 [2:0]			WSEQ_ADDR102 [12:0]													0000F000h
		WSEQ_DELAY102 [3:0]				WSEQ_DATA_START102 [3:0]				WSEQ_DATA102 [7:0]								
R12494 (30CEh)	WSEQ_Sequence_104	WSEQ_DATA_WIDTH103 [2:0]			WSEQ_ADDR103 [12:0]													0000F000h
		WSEQ_DELAY103 [3:0]				WSEQ_DATA_START103 [3:0]				WSEQ_DATA103 [7:0]								
R12496 (30D0h)	WSEQ_Sequence_105	WSEQ_DATA_WIDTH104 [2:0]			WSEQ_ADDR104 [12:0]													0000F000h
		WSEQ_DELAY104 [3:0]				WSEQ_DATA_START104 [3:0]				WSEQ_DATA104 [7:0]								
R12498 (30D2h)	WSEQ_Sequence_106	WSEQ_DATA_WIDTH105 [2:0]			WSEQ_ADDR105 [12:0]													0000F000h
		WSEQ_DELAY105 [3:0]				WSEQ_DATA_START105 [3:0]				WSEQ_DATA105 [7:0]								
R12500 (30D4h)	WSEQ_Sequence_107	WSEQ_DATA_WIDTH106 [2:0]			WSEQ_ADDR106 [12:0]													026D0101h
		WSEQ_DELAY106 [3:0]				WSEQ_DATA_START106 [3:0]				WSEQ_DATA106 [7:0]								
R12502 (30D6h)	WSEQ_Sequence_108	WSEQ_DATA_WIDTH107 [2:0]			WSEQ_ADDR107 [12:0]													A4AE101Dh
		WSEQ_DELAY107 [3:0]				WSEQ_DATA_START107 [3:0]				WSEQ_DATA107 [7:0]								
R12504 (30D8h)	WSEQ_Sequence_109	WSEQ_DATA_WIDTH108 [2:0]			WSEQ_ADDR108 [12:0]													A4AE0003h
		WSEQ_DELAY108 [3:0]				WSEQ_DATA_START108 [3:0]				WSEQ_DATA108 [7:0]								
R12506 (30DAh)	WSEQ_Sequence_110	WSEQ_DATA_WIDTH109 [2:0]			WSEQ_ADDR109 [12:0]													04AE1800h
		WSEQ_DELAY109 [3:0]				WSEQ_DATA_START109 [3:0]				WSEQ_DATA109 [7:0]								
R12508 (30DCh)	WSEQ_Sequence_111	WSEQ_DATA_WIDTH110 [2:0]			WSEQ_ADDR110 [12:0]													04024700h
		WSEQ_DELAY110 [3:0]				WSEQ_DATA_START110 [3:0]				WSEQ_DATA110 [7:0]								
R12510 (30DEh)	WSEQ_Sequence_112	WSEQ_DATA_WIDTH111 [2:0]			WSEQ_ADDR111 [12:0]													A4AE0003h
		WSEQ_DELAY111 [3:0]				WSEQ_DATA_START111 [3:0]				WSEQ_DATA111 [7:0]								
R12512 (30E0h)	WSEQ_Sequence_113	WSEQ_DATA_WIDTH112 [2:0]			WSEQ_ADDR112 [12:0]													026D0F00h
		WSEQ_DELAY112 [3:0]				WSEQ_DATA_START112 [3:0]				WSEQ_DATA112 [7:0]								
R12514 (30E2h)	WSEQ_Sequence_114	WSEQ_DATA_WIDTH113 [2:0]			WSEQ_ADDR113 [12:0]													04C7F301h
		WSEQ_DELAY113 [3:0]				WSEQ_DATA_START113 [3:0]				WSEQ_DATA113 [7:0]								
R12516 (30E4h)	WSEQ_Sequence_115	WSEQ_DATA_WIDTH114 [2:0]			WSEQ_ADDR114 [12:0]													0000F000h
		WSEQ_DELAY114 [3:0]				WSEQ_DATA_START114 [3:0]				WSEQ_DATA114 [7:0]								
R12518 (30E6h)	WSEQ_Sequence_116	WSEQ_DATA_WIDTH115 [2:0]			WSEQ_ADDR115 [12:0]													0000F000h
		WSEQ_DELAY115 [3:0]				WSEQ_DATA_START115 [3:0]				WSEQ_DATA115 [7:0]								
R12520 (30E8h)	WSEQ_Sequence_117	WSEQ_DATA_WIDTH116 [2:0]			WSEQ_ADDR116 [12:0]													0000F000h
		WSEQ_DELAY116 [3:0]				WSEQ_DATA_START116 [3:0]				WSEQ_DATA116 [7:0]								
R12522 (30EAh)	WSEQ_Sequence_118	WSEQ_DATA_WIDTH117 [2:0]			WSEQ_ADDR117 [12:0]													0000F000h
		WSEQ_DELAY117 [3:0]				WSEQ_DATA_START117 [3:0]				WSEQ_DATA117 [7:0]								
R12524 (30ECh)	WSEQ_Sequence_119	WSEQ_DATA_WIDTH118 [2:0]			WSEQ_ADDR118 [12:0]													0000F000h
		WSEQ_DELAY118 [3:0]				WSEQ_DATA_START118 [3:0]				WSEQ_DATA118 [7:0]								
R12526 (30EEh)	WSEQ_Sequence_120	WSEQ_DATA_WIDTH119 [2:0]			WSEQ_ADDR119 [12:0]													0000F000h
		WSEQ_DELAY119 [3:0]				WSEQ_DATA_START119 [3:0]				WSEQ_DATA119 [7:0]								
R12528 (30F0h)	WSEQ_Sequence_121	WSEQ_DATA_WIDTH120 [2:0]			WSEQ_ADDR120 [12:0]													0000F000h
		WSEQ_DELAY120 [3:0]				WSEQ_DATA_START120 [3:0]				WSEQ_DATA120 [7:0]								
R12530 (30F2h)	WSEQ_Sequence_122	WSEQ_DATA_WIDTH121 [2:0]			WSEQ_ADDR121 [12:0]													0000F000h
		WSEQ_DELAY121 [3:0]				WSEQ_DATA_START121 [3:0]				WSEQ_DATA121 [7:0]								
R12532 (30F4h)	WSEQ_Sequence_123	WSEQ_DATA_WIDTH122 [2:0]			WSEQ_ADDR122 [12:0]													0000F000h
		WSEQ_DELAY122 [3:0]				WSEQ_DATA_START122 [3:0]				WSEQ_DATA122 [7:0]								
R12534 (30F6h)	WSEQ_Sequence_124	WSEQ_DATA_WIDTH123 [2:0]			WSEQ_ADDR123 [12:0]													0000F000h
		WSEQ_DELAY123 [3:0]				WSEQ_DATA_START123 [3:0]				WSEQ_DATA123 [7:0]								
R12536 (30F8h)	WSEQ_Sequence_125	WSEQ_DATA_WIDTH124 [2:0]			WSEQ_ADDR124 [12:0]													0000F000h
		WSEQ_DELAY124 [3:0]				WSEQ_DATA_START124 [3:0]				WSEQ_DATA124 [7:0]								

Table 6-2. Register Map Definition—32-bit region *(Cont.)*

Register	Name	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0	Default
R12538 (30FAh)	WSEQ_Sequence_126	WSEQ_DATA_WIDTH125 [2:0]			WSEQ_ADDR125 [12:0]													0000F000h
		WSEQ_DELAY125 [3:0]				WSEQ_DATA_START125 [3:0]				WSEQ_DATA125 [7:0]								
R12540 (30FCh)	WSEQ_Sequence_127	WSEQ_DATA_WIDTH126 [2:0]			WSEQ_ADDR126 [12:0]													0000F000h
		WSEQ_DELAY126 [3:0]				WSEQ_DATA_START126 [3:0]				WSEQ_DATA126 [7:0]								
R12542 (30FEh)	WSEQ_Sequence_128	WSEQ_DATA_WIDTH127 [2:0]			WSEQ_ADDR127 [12:0]													0000F000h
		WSEQ_DELAY127 [3:0]				WSEQ_DATA_START127 [3:0]				WSEQ_DATA127 [7:0]								
R12544 (3100h)	WSEQ_Sequence_129	WSEQ_DATA_WIDTH128 [2:0]			WSEQ_ADDR128 [12:0]													110007FFh
		WSEQ_DELAY128 [3:0]				WSEQ_DATA_START128 [3:0]				WSEQ_DATA128 [7:0]								
R12546 (3102h)	WSEQ_Sequence_130	WSEQ_DATA_WIDTH129 [2:0]			WSEQ_ADDR129 [12:0]													00000100h
		WSEQ_DELAY129 [3:0]				WSEQ_DATA_START129 [3:0]				WSEQ_DATA129 [7:0]								
R12548 (3104h)	WSEQ_Sequence_131	WSEQ_DATA_WIDTH130 [2:0]			WSEQ_ADDR130 [12:0]													A0340000h
		WSEQ_DELAY130 [3:0]				WSEQ_DATA_START130 [3:0]				WSEQ_DATA130 [7:0]								
R12550 (3106h)	WSEQ_Sequence_132	WSEQ_DATA_WIDTH131 [2:0]			WSEQ_ADDR131 [12:0]													E0310000h
		WSEQ_DELAY131 [3:0]				WSEQ_DATA_START131 [3:0]				WSEQ_DATA131 [7:0]								
R12552 (3108h)	WSEQ_Sequence_133	WSEQ_DATA_WIDTH132 [2:0]			WSEQ_ADDR132 [12:0]													A0300800h
		WSEQ_DELAY132 [3:0]				WSEQ_DATA_START132 [3:0]				WSEQ_DATA132 [7:0]								
R12554 (310Ah)	WSEQ_Sequence_134	WSEQ_DATA_WIDTH133 [2:0]			WSEQ_ADDR133 [12:0]													E0300000h
		WSEQ_DELAY133 [3:0]				WSEQ_DATA_START133 [3:0]				WSEQ_DATA133 [7:0]								
R12556 (310Ch)	WSEQ_Sequence_135	WSEQ_DATA_WIDTH134 [2:0]			WSEQ_ADDR134 [12:0]													11000206h
		WSEQ_DELAY134 [3:0]				WSEQ_DATA_START134 [3:0]				WSEQ_DATA134 [7:0]								
R12558 (310Eh)	WSEQ_Sequence_136	WSEQ_DATA_WIDTH135 [2:0]			WSEQ_ADDR135 [12:0]													C0080040h
		WSEQ_DELAY135 [3:0]				WSEQ_DATA_START135 [3:0]				WSEQ_DATA135 [7:0]								
R12560 (3110h)	WSEQ_Sequence_137	WSEQ_DATA_WIDTH136 [2:0]			WSEQ_ADDR136 [12:0]													00080800h
		WSEQ_DELAY136 [3:0]				WSEQ_DATA_START136 [3:0]				WSEQ_DATA136 [7:0]								
R12562 (3112h)	WSEQ_Sequence_138	WSEQ_DATA_WIDTH137 [2:0]			WSEQ_ADDR137 [12:0]													A000001Dh
		WSEQ_DELAY137 [3:0]				WSEQ_DATA_START137 [3:0]				WSEQ_DATA137 [7:0]								
R12564 (3114h)	WSEQ_Sequence_139	WSEQ_DATA_WIDTH138 [2:0]			WSEQ_ADDR138 [12:0]													60090008h
		WSEQ_DELAY138 [3:0]				WSEQ_DATA_START138 [3:0]				WSEQ_DATA138 [7:0]								
R12566 (3116h)	WSEQ_Sequence_140	WSEQ_DATA_WIDTH139 [2:0]			WSEQ_ADDR139 [12:0]													60090808h
		WSEQ_DELAY139 [3:0]				WSEQ_DATA_START139 [3:0]				WSEQ_DATA139 [7:0]								
R12568 (3118h)	WSEQ_Sequence_141	WSEQ_DATA_WIDTH140 [2:0]			WSEQ_ADDR140 [12:0]													11000000h
		WSEQ_DELAY140 [3:0]				WSEQ_DATA_START140 [3:0]				WSEQ_DATA140 [7:0]								
R12570 (311Ah)	WSEQ_Sequence_142	WSEQ_DATA_WIDTH141 [2:0]			WSEQ_ADDR141 [12:0]													01200600h
		WSEQ_DELAY141 [3:0]				WSEQ_DATA_START141 [3:0]				WSEQ_DATA141 [7:0]								
R12572 (311Ch)	WSEQ_Sequence_143	WSEQ_DATA_WIDTH142 [2:0]			WSEQ_ADDR142 [12:0]													01010600h
		WSEQ_DELAY142 [3:0]				WSEQ_DATA_START142 [3:0]				WSEQ_DATA142 [7:0]								
R12574 (311Eh)	WSEQ_Sequence_144	WSEQ_DATA_WIDTH143 [2:0]			WSEQ_ADDR143 [12:0]													41D10005h
		WSEQ_DELAY143 [3:0]				WSEQ_DATA_START143 [3:0]				WSEQ_DATA143 [7:0]								
R12576 (3120h)	WSEQ_Sequence_145	WSEQ_DATA_WIDTH144 [2:0]			WSEQ_ADDR144 [12:0]													E1220080h
		WSEQ_DELAY144 [3:0]				WSEQ_DATA_START144 [3:0]				WSEQ_DATA144 [7:0]								
R12578 (3122h)	WSEQ_Sequence_146	WSEQ_DATA_WIDTH145 [2:0]			WSEQ_ADDR145 [12:0]													E1220825h
		WSEQ_DELAY145 [3:0]				WSEQ_DATA_START145 [3:0]				WSEQ_DATA145 [7:0]								
R12580 (3124h)	WSEQ_Sequence_147	WSEQ_DATA_WIDTH146 [2:0]			WSEQ_ADDR146 [12:0]													E1240080h
		WSEQ_DELAY146 [3:0]				WSEQ_DATA_START146 [3:0]				WSEQ_DATA146 [7:0]								
R12582 (3126h)	WSEQ_Sequence_148	WSEQ_DATA_WIDTH147 [2:0]			WSEQ_ADDR147 [12:0]													E124080Ch
		WSEQ_DELAY147 [3:0]				WSEQ_DATA_START147 [3:0]				WSEQ_DATA147 [7:0]								
R12584 (3128h)	WSEQ_Sequence_149	WSEQ_DATA_WIDTH148 [2:0]			WSEQ_ADDR148 [12:0]													61200007h
		WSEQ_DELAY148 [3:0]				WSEQ_DATA_START148 [3:0]				WSEQ_DATA148 [7:0]								
R12586 (312Ah)	WSEQ_Sequence_150	WSEQ_DATA_WIDTH149 [2:0]			WSEQ_ADDR149 [12:0]													01200601h
		WSEQ_DELAY149 [3:0]				WSEQ_DATA_START149 [3:0]				WSEQ_DATA149 [7:0]								
R12588 (312Ch)	WSEQ_Sequence_151	WSEQ_DATA_WIDTH150 [2:0]			WSEQ_ADDR150 [12:0]													110007FFh
		WSEQ_DELAY150 [3:0]				WSEQ_DATA_START150 [3:0]				WSEQ_DATA150 [7:0]								
R12590 (312Eh)	WSEQ_Sequence_152	WSEQ_DATA_WIDTH151 [2:0]			WSEQ_ADDR151 [12:0]													E0020080h
		WSEQ_DELAY151 [3:0]				WSEQ_DATA_START151 [3:0]				WSEQ_DATA151 [7:0]								
R12592 (3130h)	WSEQ_Sequence_153	WSEQ_DATA_WIDTH152 [2:0]			WSEQ_ADDR152 [12:0]													E0020825h
		WSEQ_DELAY152 [3:0]				WSEQ_DATA_START152 [3:0]				WSEQ_DATA152 [7:0]								
R12594 (3132h)	WSEQ_Sequence_154	WSEQ_DATA_WIDTH153 [2:0]			WSEQ_ADDR153 [12:0]													00000401h
		WSEQ_DELAY153 [3:0]				WSEQ_DATA_START153 [3:0]				WSEQ_DATA153 [7:0]								
R12596 (3134h)	WSEQ_Sequence_155	WSEQ_DATA_WIDTH154 [2:0]			WSEQ_ADDR154 [12:0]													00030001h
		WSEQ_DELAY154 [3:0]				WSEQ_DATA_START154 [3:0]				WSEQ_DATA154 [7:0]								
R12598 (3136h)	WSEQ_Sequence_156	WSEQ_DATA_WIDTH155 [2:0]			WSEQ_ADDR155 [12:0]													0000F101h
		WSEQ_DELAY155 [3:0]				WSEQ_DATA_START155 [3:0]				WSEQ_DATA155 [7:0]								
R12600 (3138h)	WSEQ_Sequence_157	WSEQ_DATA_WIDTH156 [2:0]			WSEQ_ADDR156 [12:0]													0000F000h
		WSEQ_DELAY156 [3:0]				WSEQ_DATA_START156 [3:0]				WSEQ_DATA156 [7:0]								
R12602 (313Ah)	WSEQ_Sequence_158	WSEQ_DATA_WIDTH157 [2:0]			WSEQ_ADDR157 [12:0]													0000F000h
		WSEQ_DELAY157 [3:0]				WSEQ_DATA_START157 [3:0]				WSEQ_DATA157 [7:0]								
R12604 (313Ch)	WSEQ_Sequence_159	WSEQ_DATA_WIDTH158 [2:0]			WSEQ_ADDR158 [12:0]													0000F000h
		WSEQ_DELAY158 [3:0]				WSEQ_DATA_START158 [3:0]				WSEQ_DATA158 [7:0]								
R12606 (313Eh)	WSEQ_Sequence_160	WSEQ_DATA_WIDTH159 [2:0]			WSEQ_ADDR159 [12:0]													0000F000h
		WSEQ_DELAY159 [3:0]				WSEQ_DATA_START159 [3:0]				WSEQ_DATA159 [7:0]								
R12608 (3140h)	WSEQ_Sequence_161	WSEQ_DATA_WIDTH160 [2:0]			WSEQ_ADDR160 [12:0]													0000F000h
		WSEQ_DELAY160 [3:0]				WSEQ_DATA_START160 [3:0]				WSEQ_DATA160 [7:0]								

Table 6-2. Register Map Definition—32-bit region *(Cont.)*

Register	Name	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0	Default
R12610 (3142h)	WSEQ_Sequence_162	WSEQ_DATA_WIDTH161 [2:0]			WSEQ_ADDR161 [12:0]													0000F000h
		WSEQ_DELAY161 [3:0]				WSEQ_DATA_START161 [3:0]				WSEQ_DATA161 [7:0]								
R12612 (3144h)	WSEQ_Sequence_163	WSEQ_DATA_WIDTH162 [2:0]			WSEQ_ADDR162 [12:0]													0000F000h
		WSEQ_DELAY162 [3:0]				WSEQ_DATA_START162 [3:0]				WSEQ_DATA162 [7:0]								
R12614 (3146h)	WSEQ_Sequence_164	WSEQ_DATA_WIDTH163 [2:0]			WSEQ_ADDR163 [12:0]													0000F000h
		WSEQ_DELAY163 [3:0]				WSEQ_DATA_START163 [3:0]				WSEQ_DATA163 [7:0]								
R12616 (3148h)	WSEQ_Sequence_165	WSEQ_DATA_WIDTH164 [2:0]			WSEQ_ADDR164 [12:0]													0000F000h
		WSEQ_DELAY164 [3:0]				WSEQ_DATA_START164 [3:0]				WSEQ_DATA164 [7:0]								
R12618 (314Ah)	WSEQ_Sequence_166	WSEQ_DATA_WIDTH165 [2:0]			WSEQ_ADDR165 [12:0]													0000F000h
		WSEQ_DELAY165 [3:0]				WSEQ_DATA_START165 [3:0]				WSEQ_DATA165 [7:0]								
R12620 (314Ch)	WSEQ_Sequence_167	WSEQ_DATA_WIDTH166 [2:0]			WSEQ_ADDR166 [12:0]													0000F000h
		WSEQ_DELAY166 [3:0]				WSEQ_DATA_START166 [3:0]				WSEQ_DATA166 [7:0]								
R12622 (314Eh)	WSEQ_Sequence_168	WSEQ_DATA_WIDTH167 [2:0]			WSEQ_ADDR167 [12:0]													0000F000h
		WSEQ_DELAY167 [3:0]				WSEQ_DATA_START167 [3:0]				WSEQ_DATA167 [7:0]								
R12624 (3150h)	WSEQ_Sequence_169	WSEQ_DATA_WIDTH168 [2:0]			WSEQ_ADDR168 [12:0]													0000F000h
		WSEQ_DELAY168 [3:0]				WSEQ_DATA_START168 [3:0]				WSEQ_DATA168 [7:0]								
R12626 (3152h)	WSEQ_Sequence_170	WSEQ_DATA_WIDTH169 [2:0]			WSEQ_ADDR169 [12:0]													0000F000h
		WSEQ_DELAY169 [3:0]				WSEQ_DATA_START169 [3:0]				WSEQ_DATA169 [7:0]								
R12628 (3154h)	WSEQ_Sequence_171	WSEQ_DATA_WIDTH170 [2:0]			WSEQ_ADDR170 [12:0]													0000F000h
		WSEQ_DELAY170 [3:0]				WSEQ_DATA_START170 [3:0]				WSEQ_DATA170 [7:0]								
R12630 (3156h)	WSEQ_Sequence_172	WSEQ_DATA_WIDTH171 [2:0]			WSEQ_ADDR171 [12:0]													0000F000h
		WSEQ_DELAY171 [3:0]				WSEQ_DATA_START171 [3:0]				WSEQ_DATA171 [7:0]								
R12632 (3158h)	WSEQ_Sequence_173	WSEQ_DATA_WIDTH172 [2:0]			WSEQ_ADDR172 [12:0]													0000F000h
		WSEQ_DELAY172 [3:0]				WSEQ_DATA_START172 [3:0]				WSEQ_DATA172 [7:0]								
R12634 (315Ah)	WSEQ_Sequence_174	WSEQ_DATA_WIDTH173 [2:0]			WSEQ_ADDR173 [12:0]													0000F000h
		WSEQ_DELAY173 [3:0]				WSEQ_DATA_START173 [3:0]				WSEQ_DATA173 [7:0]								
R12636 (315Ch)	WSEQ_Sequence_175	WSEQ_DATA_WIDTH174 [2:0]			WSEQ_ADDR174 [12:0]													0000F000h
		WSEQ_DELAY174 [3:0]				WSEQ_DATA_START174 [3:0]				WSEQ_DATA174 [7:0]								
R12638 (315Eh)	WSEQ_Sequence_176	WSEQ_DATA_WIDTH175 [2:0]			WSEQ_ADDR175 [12:0]													0000F000h
		WSEQ_DELAY175 [3:0]				WSEQ_DATA_START175 [3:0]				WSEQ_DATA175 [7:0]								
R12640 (3160h)	WSEQ_Sequence_177	WSEQ_DATA_WIDTH176 [2:0]			WSEQ_ADDR176 [12:0]													0000F000h
		WSEQ_DELAY176 [3:0]				WSEQ_DATA_START176 [3:0]				WSEQ_DATA176 [7:0]								
R12642 (3162h)	WSEQ_Sequence_178	WSEQ_DATA_WIDTH177 [2:0]			WSEQ_ADDR177 [12:0]													0000F000h
		WSEQ_DELAY177 [3:0]				WSEQ_DATA_START177 [3:0]				WSEQ_DATA177 [7:0]								
R12644 (3164h)	WSEQ_Sequence_179	WSEQ_DATA_WIDTH178 [2:0]			WSEQ_ADDR178 [12:0]													0000F000h
		WSEQ_DELAY178 [3:0]				WSEQ_DATA_START178 [3:0]				WSEQ_DATA178 [7:0]								
R12646 (3166h)	WSEQ_Sequence_180	WSEQ_DATA_WIDTH179 [2:0]			WSEQ_ADDR179 [12:0]													0000F000h
		WSEQ_DELAY179 [3:0]				WSEQ_DATA_START179 [3:0]				WSEQ_DATA179 [7:0]								
R12648 (3168h)	WSEQ_Sequence_181	WSEQ_DATA_WIDTH180 [2:0]			WSEQ_ADDR180 [12:0]													0000F000h
		WSEQ_DELAY180 [3:0]				WSEQ_DATA_START180 [3:0]				WSEQ_DATA180 [7:0]								
R12650 (316Ah)	WSEQ_Sequence_182	WSEQ_DATA_WIDTH181 [2:0]			WSEQ_ADDR181 [12:0]													0000F000h
		WSEQ_DELAY181 [3:0]				WSEQ_DATA_START181 [3:0]				WSEQ_DATA181 [7:0]								
R12652 (316Ch)	WSEQ_Sequence_183	WSEQ_DATA_WIDTH182 [2:0]			WSEQ_ADDR182 [12:0]													0000F000h
		WSEQ_DELAY182 [3:0]				WSEQ_DATA_START182 [3:0]				WSEQ_DATA182 [7:0]								
R12654 (316Eh)	WSEQ_Sequence_184	WSEQ_DATA_WIDTH183 [2:0]			WSEQ_ADDR183 [12:0]													0000F000h
		WSEQ_DELAY183 [3:0]				WSEQ_DATA_START183 [3:0]				WSEQ_DATA183 [7:0]								
R12656 (3170h)	WSEQ_Sequence_185	WSEQ_DATA_WIDTH184 [2:0]			WSEQ_ADDR184 [12:0]													0000F000h
		WSEQ_DELAY184 [3:0]				WSEQ_DATA_START184 [3:0]				WSEQ_DATA184 [7:0]								
R12658 (3172h)	WSEQ_Sequence_186	WSEQ_DATA_WIDTH185 [2:0]			WSEQ_ADDR185 [12:0]													0000F000h
		WSEQ_DELAY185 [3:0]				WSEQ_DATA_START185 [3:0]				WSEQ_DATA185 [7:0]								
R12660 (3174h)	WSEQ_Sequence_187	WSEQ_DATA_WIDTH186 [2:0]			WSEQ_ADDR186 [12:0]													0000F000h
		WSEQ_DELAY186 [3:0]				WSEQ_DATA_START186 [3:0]				WSEQ_DATA186 [7:0]								
R12662 (3176h)	WSEQ_Sequence_188	WSEQ_DATA_WIDTH187 [2:0]			WSEQ_ADDR187 [12:0]													0000F000h
		WSEQ_DELAY187 [3:0]				WSEQ_DATA_START187 [3:0]				WSEQ_DATA187 [7:0]								
R12664 (3178h)	WSEQ_Sequence_189	WSEQ_DATA_WIDTH188 [2:0]			WSEQ_ADDR188 [12:0]													0000F000h
		WSEQ_DELAY188 [3:0]				WSEQ_DATA_START188 [3:0]				WSEQ_DATA188 [7:0]								
R12666 (317Ah)	WSEQ_Sequence_190	WSEQ_DATA_WIDTH189 [2:0]			WSEQ_ADDR189 [12:0]													0000F000h
		WSEQ_DELAY189 [3:0]				WSEQ_DATA_START189 [3:0]				WSEQ_DATA189 [7:0]								
R12668 (317Ch)	WSEQ_Sequence_191	WSEQ_DATA_WIDTH190 [2:0]			WSEQ_ADDR190 [12:0]													0000F000h
		WSEQ_DELAY190 [3:0]				WSEQ_DATA_START190 [3:0]				WSEQ_DATA190 [7:0]								
R12670 (317Eh)	WSEQ_Sequence_192	WSEQ_DATA_WIDTH191 [2:0]			WSEQ_ADDR191 [12:0]													0000F000h
		WSEQ_DELAY191 [3:0]				WSEQ_DATA_START191 [3:0]				WSEQ_DATA191 [7:0]								
R12672 (3180h)	WSEQ_Sequence_193	WSEQ_DATA_WIDTH192 [2:0]			WSEQ_ADDR192 [12:0]													0000F000h
		WSEQ_DELAY192 [3:0]				WSEQ_DATA_START192 [3:0]				WSEQ_DATA192 [7:0]								
R12674 (3182h)	WSEQ_Sequence_194	WSEQ_DATA_WIDTH193 [2:0]			WSEQ_ADDR193 [12:0]													0000F000h
		WSEQ_DELAY193 [3:0]				WSEQ_DATA_START193 [3:0]				WSEQ_DATA193 [7:0]								
R12676 (3184h)	WSEQ_Sequence_195	WSEQ_DATA_WIDTH194 [2:0]			WSEQ_ADDR194 [12:0]													0000F000h
		WSEQ_DELAY194 [3:0]				WSEQ_DATA_START194 [3:0]				WSEQ_DATA194 [7:0]								
R12678 (3186h)	WSEQ_Sequence_196	WSEQ_DATA_WIDTH195 [2:0]			WSEQ_ADDR195 [12:0]													0000F000h
		WSEQ_DELAY195 [3:0]				WSEQ_DATA_START195 [3:0]				WSEQ_DATA195 [7:0]								
R12680 (3188h)	WSEQ_Sequence_197	WSEQ_DATA_WIDTH196 [2:0]			WSEQ_ADDR196 [12:0]													0000F000h
		WSEQ_DELAY196 [3:0]				WSEQ_DATA_START196 [3:0]				WSEQ_DATA196 [7:0]								

Table 6-2. Register Map Definition—32-bit region *(Cont.)*

Register	Name	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0	Default
R12682 (318Ah)	WSEQ_Sequence_198	WSEQ_DATA_WIDTH197 [2:0]			WSEQ_ADDR197 [12:0]													0000F000h
		WSEQ_DELAY197 [3:0]				WSEQ_DATA_START197 [3:0]				WSEQ_DATA197 [7:0]								
R12684 (318Ch)	WSEQ_Sequence_199	WSEQ_DATA_WIDTH198 [2:0]			WSEQ_ADDR198 [12:0]													0000F000h
		WSEQ_DELAY198 [3:0]				WSEQ_DATA_START198 [3:0]				WSEQ_DATA198 [7:0]								
R12686 (318Eh)	WSEQ_Sequence_200	WSEQ_DATA_WIDTH199 [2:0]			WSEQ_ADDR199 [12:0]													0000F000h
		WSEQ_DELAY199 [3:0]				WSEQ_DATA_START199 [3:0]				WSEQ_DATA199 [7:0]								
R12688 (3190h)	WSEQ_Sequence_201	WSEQ_DATA_WIDTH200 [2:0]			WSEQ_ADDR200 [12:0]													0000F000h
		WSEQ_DELAY200 [3:0]				WSEQ_DATA_START200 [3:0]				WSEQ_DATA200 [7:0]								
R12690 (3192h)	WSEQ_Sequence_202	WSEQ_DATA_WIDTH201 [2:0]			WSEQ_ADDR201 [12:0]													0000F000h
		WSEQ_DELAY201 [3:0]				WSEQ_DATA_START201 [3:0]				WSEQ_DATA201 [7:0]								
R12692 (3194h)	WSEQ_Sequence_203	WSEQ_DATA_WIDTH202 [2:0]			WSEQ_ADDR202 [12:0]													0000F000h
		WSEQ_DELAY202 [3:0]				WSEQ_DATA_START202 [3:0]				WSEQ_DATA202 [7:0]								
R12694 (3196h)	WSEQ_Sequence_204	WSEQ_DATA_WIDTH203 [2:0]			WSEQ_ADDR203 [12:0]													0000F000h
		WSEQ_DELAY203 [3:0]				WSEQ_DATA_START203 [3:0]				WSEQ_DATA203 [7:0]								
R12696 (3198h)	WSEQ_Sequence_205	WSEQ_DATA_WIDTH204 [2:0]			WSEQ_ADDR204 [12:0]													0000F000h
		WSEQ_DELAY204 [3:0]				WSEQ_DATA_START204 [3:0]				WSEQ_DATA204 [7:0]								
R12698 (319Ah)	WSEQ_Sequence_206	WSEQ_DATA_WIDTH205 [2:0]			WSEQ_ADDR205 [12:0]													0000F000h
		WSEQ_DELAY205 [3:0]				WSEQ_DATA_START205 [3:0]				WSEQ_DATA205 [7:0]								
R12700 (319Ch)	WSEQ_Sequence_207	WSEQ_DATA_WIDTH206 [2:0]			WSEQ_ADDR206 [12:0]													0000F000h
		WSEQ_DELAY206 [3:0]				WSEQ_DATA_START206 [3:0]				WSEQ_DATA206 [7:0]								
R12702 (319Eh)	WSEQ_Sequence_208	WSEQ_DATA_WIDTH207 [2:0]			WSEQ_ADDR207 [12:0]													0000F000h
		WSEQ_DELAY207 [3:0]				WSEQ_DATA_START207 [3:0]				WSEQ_DATA207 [7:0]								
R12704 (31A0h)	WSEQ_Sequence_209	WSEQ_DATA_WIDTH208 [2:0]			WSEQ_ADDR208 [12:0]													0000F000h
		WSEQ_DELAY208 [3:0]				WSEQ_DATA_START208 [3:0]				WSEQ_DATA208 [7:0]								
R12706 (31A2h)	WSEQ_Sequence_210	WSEQ_DATA_WIDTH209 [2:0]			WSEQ_ADDR209 [12:0]													0000F000h
		WSEQ_DELAY209 [3:0]				WSEQ_DATA_START209 [3:0]				WSEQ_DATA209 [7:0]								
R12708 (31A4h)	WSEQ_Sequence_211	WSEQ_DATA_WIDTH210 [2:0]			WSEQ_ADDR210 [12:0]													0000F000h
		WSEQ_DELAY210 [3:0]				WSEQ_DATA_START210 [3:0]				WSEQ_DATA210 [7:0]								
R12710 (31A6h)	WSEQ_Sequence_212	WSEQ_DATA_WIDTH211 [2:0]			WSEQ_ADDR211 [12:0]													0000F000h
		WSEQ_DELAY211 [3:0]				WSEQ_DATA_START211 [3:0]				WSEQ_DATA211 [7:0]								
R12712 (31A8h)	WSEQ_Sequence_213	WSEQ_DATA_WIDTH212 [2:0]			WSEQ_ADDR212 [12:0]													0000F000h
		WSEQ_DELAY212 [3:0]				WSEQ_DATA_START212 [3:0]				WSEQ_DATA212 [7:0]								
R12714 (31AAh)	WSEQ_Sequence_214	WSEQ_DATA_WIDTH213 [2:0]			WSEQ_ADDR213 [12:0]													0000F000h
		WSEQ_DELAY213 [3:0]				WSEQ_DATA_START213 [3:0]				WSEQ_DATA213 [7:0]								
R12716 (31ACh)	WSEQ_Sequence_215	WSEQ_DATA_WIDTH214 [2:0]			WSEQ_ADDR214 [12:0]													0000F000h
		WSEQ_DELAY214 [3:0]				WSEQ_DATA_START214 [3:0]				WSEQ_DATA214 [7:0]								
R12718 (31AEh)	WSEQ_Sequence_216	WSEQ_DATA_WIDTH215 [2:0]			WSEQ_ADDR215 [12:0]													0000F000h
		WSEQ_DELAY215 [3:0]				WSEQ_DATA_START215 [3:0]				WSEQ_DATA215 [7:0]								
R12720 (31B0h)	WSEQ_Sequence_217	WSEQ_DATA_WIDTH216 [2:0]			WSEQ_ADDR216 [12:0]													0000F000h
		WSEQ_DELAY216 [3:0]				WSEQ_DATA_START216 [3:0]				WSEQ_DATA216 [7:0]								
R12722 (31B2h)	WSEQ_Sequence_218	WSEQ_DATA_WIDTH217 [2:0]			WSEQ_ADDR217 [12:0]													0000F000h
		WSEQ_DELAY217 [3:0]				WSEQ_DATA_START217 [3:0]				WSEQ_DATA217 [7:0]								
R12724 (31B4h)	WSEQ_Sequence_219	WSEQ_DATA_WIDTH218 [2:0]			WSEQ_ADDR218 [12:0]													0000F000h
		WSEQ_DELAY218 [3:0]				WSEQ_DATA_START218 [3:0]				WSEQ_DATA218 [7:0]								
R12726 (31B6h)	WSEQ_Sequence_220	WSEQ_DATA_WIDTH219 [2:0]			WSEQ_ADDR219 [12:0]													0000F000h
		WSEQ_DELAY219 [3:0]				WSEQ_DATA_START219 [3:0]				WSEQ_DATA219 [7:0]								
R12728 (31B8h)	WSEQ_Sequence_221	WSEQ_DATA_WIDTH220 [2:0]			WSEQ_ADDR220 [12:0]													0000F000h
		WSEQ_DELAY220 [3:0]				WSEQ_DATA_START220 [3:0]				WSEQ_DATA220 [7:0]								
R12730 (31BAh)	WSEQ_Sequence_222	WSEQ_DATA_WIDTH221 [2:0]			WSEQ_ADDR221 [12:0]													0000F000h
		WSEQ_DELAY221 [3:0]				WSEQ_DATA_START221 [3:0]				WSEQ_DATA221 [7:0]								
R12732 (31BCh)	WSEQ_Sequence_223	WSEQ_DATA_WIDTH222 [2:0]			WSEQ_ADDR222 [12:0]													0000F000h
		WSEQ_DELAY222 [3:0]				WSEQ_DATA_START222 [3:0]				WSEQ_DATA222 [7:0]								
R12734 (31BEh)	WSEQ_Sequence_224	WSEQ_DATA_WIDTH223 [2:0]			WSEQ_ADDR223 [12:0]													0000F000h
		WSEQ_DELAY223 [3:0]				WSEQ_DATA_START223 [3:0]				WSEQ_DATA223 [7:0]								
R12736 (31C0h)	WSEQ_Sequence_225	WSEQ_DATA_WIDTH224 [2:0]			WSEQ_ADDR224 [12:0]													FFFFFFFFh
		WSEQ_DELAY224 [3:0]				WSEQ_DATA_START224 [3:0]				WSEQ_DATA224 [7:0]								
R12738 (31C2h)	WSEQ_Sequence_226	WSEQ_DATA_WIDTH225 [2:0]			WSEQ_ADDR225 [12:0]													FFFFFFFFh
		WSEQ_DELAY225 [3:0]				WSEQ_DATA_START225 [3:0]				WSEQ_DATA225 [7:0]								
R12740 (31C4h)	WSEQ_Sequence_227	WSEQ_DATA_WIDTH226 [2:0]			WSEQ_ADDR226 [12:0]													FFFFFFFFh
		WSEQ_DELAY226 [3:0]				WSEQ_DATA_START226 [3:0]				WSEQ_DATA226 [7:0]								
R12742 (31C6h)	WSEQ_Sequence_228	WSEQ_DATA_WIDTH227 [2:0]			WSEQ_ADDR227 [12:0]													FFFFFFFFh
		WSEQ_DELAY227 [3:0]				WSEQ_DATA_START227 [3:0]				WSEQ_DATA227 [7:0]								
R12744 (31C8h)	WSEQ_Sequence_229	WSEQ_DATA_WIDTH228 [2:0]			WSEQ_ADDR228 [12:0]													FFFFFFFFh
		WSEQ_DELAY228 [3:0]				WSEQ_DATA_START228 [3:0]				WSEQ_DATA228 [7:0]								
R12746 (31CAh)	WSEQ_Sequence_230	WSEQ_DATA_WIDTH229 [2:0]			WSEQ_ADDR229 [12:0]													FFFFFFFFh
		WSEQ_DELAY229 [3:0]				WSEQ_DATA_START229 [3:0]				WSEQ_DATA229 [7:0]								
R12748 (31CCh)	WSEQ_Sequence_231	WSEQ_DATA_WIDTH230 [2:0]			WSEQ_ADDR230 [12:0]													FFFFFFFFh
		WSEQ_DELAY230 [3:0]				WSEQ_DATA_START230 [3:0]				WSEQ_DATA230 [7:0]								
R12750 (31CEh)	WSEQ_Sequence_232	WSEQ_DATA_WIDTH231 [2:0]			WSEQ_ADDR231 [12:0]													FFFFFFFFh
		WSEQ_DELAY231 [3:0]				WSEQ_DATA_START231 [3:0]				WSEQ_DATA231 [7:0]								
R12752 (31D0h)	WSEQ_Sequence_233	WSEQ_DATA_WIDTH232 [2:0]			WSEQ_ADDR232 [12:0]													FFFFFFFFh
		WSEQ_DELAY232 [3:0]				WSEQ_DATA_START232 [3:0]				WSEQ_DATA232 [7:0]								

Table 6-2. Register Map Definition—32-bit region *(Cont.)*

Register	Name	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0	Default
R12754 (31D2h)	WSEQ_Sequence_234	WSEQ_DATA_WIDTH233 [2:0]			WSEQ_ADDR233 [12:0]													FFFFFFFFh
		WSEQ_DELAY233 [3:0]				WSEQ_DATA_START233 [3:0]				WSEQ_DATA233 [7:0]								
R12756 (31D4h)	WSEQ_Sequence_235	WSEQ_DATA_WIDTH234 [2:0]			WSEQ_ADDR234 [12:0]													FFFFFFFFh
		WSEQ_DELAY234 [3:0]				WSEQ_DATA_START234 [3:0]				WSEQ_DATA234 [7:0]								
R12758 (31D6h)	WSEQ_Sequence_236	WSEQ_DATA_WIDTH235 [2:0]			WSEQ_ADDR235 [12:0]													FFFFFFFFh
		WSEQ_DELAY235 [3:0]				WSEQ_DATA_START235 [3:0]				WSEQ_DATA235 [7:0]								
R12760 (31D8h)	WSEQ_Sequence_237	WSEQ_DATA_WIDTH236 [2:0]			WSEQ_ADDR236 [12:0]													FFFFFFFFh
		WSEQ_DELAY236 [3:0]				WSEQ_DATA_START236 [3:0]				WSEQ_DATA236 [7:0]								
R12762 (31DAh)	WSEQ_Sequence_238	WSEQ_DATA_WIDTH237 [2:0]			WSEQ_ADDR237 [12:0]													FFFFFFFFh
		WSEQ_DELAY237 [3:0]				WSEQ_DATA_START237 [3:0]				WSEQ_DATA237 [7:0]								
R12764 (31DCh)	WSEQ_Sequence_239	WSEQ_DATA_WIDTH238 [2:0]			WSEQ_ADDR238 [12:0]													FFFFFFFFh
		WSEQ_DELAY238 [3:0]				WSEQ_DATA_START238 [3:0]				WSEQ_DATA238 [7:0]								
R12766 (31DEh)	WSEQ_Sequence_240	WSEQ_DATA_WIDTH239 [2:0]			WSEQ_ADDR239 [12:0]													FFFFFFFFh
		WSEQ_DELAY239 [3:0]				WSEQ_DATA_START239 [3:0]				WSEQ_DATA239 [7:0]								
R12768 (31E0h)	WSEQ_Sequence_241	WSEQ_DATA_WIDTH240 [2:0]			WSEQ_ADDR240 [12:0]													FFFFFFFFh
		WSEQ_DELAY240 [3:0]				WSEQ_DATA_START240 [3:0]				WSEQ_DATA240 [7:0]								
R12770 (31E2h)	WSEQ_Sequence_242	WSEQ_DATA_WIDTH241 [2:0]			WSEQ_ADDR241 [12:0]													FFFFFFFFh
		WSEQ_DELAY241 [3:0]				WSEQ_DATA_START241 [3:0]				WSEQ_DATA241 [7:0]								
R12772 (31E4h)	WSEQ_Sequence_243	WSEQ_DATA_WIDTH242 [2:0]			WSEQ_ADDR242 [12:0]													FFFFFFFFh
		WSEQ_DELAY242 [3:0]				WSEQ_DATA_START242 [3:0]				WSEQ_DATA242 [7:0]								
R12774 (31E6h)	WSEQ_Sequence_244	WSEQ_DATA_WIDTH243 [2:0]			WSEQ_ADDR243 [12:0]													FFFFFFFFh
		WSEQ_DELAY243 [3:0]				WSEQ_DATA_START243 [3:0]				WSEQ_DATA243 [7:0]								
R12776 (31E8h)	WSEQ_Sequence_245	WSEQ_DATA_WIDTH244 [2:0]			WSEQ_ADDR244 [12:0]													FFFFFFFFh
		WSEQ_DELAY244 [3:0]				WSEQ_DATA_START244 [3:0]				WSEQ_DATA244 [7:0]								
R12778 (31EAh)	WSEQ_Sequence_246	WSEQ_DATA_WIDTH245 [2:0]			WSEQ_ADDR245 [12:0]													FFFFFFFFh
		WSEQ_DELAY245 [3:0]				WSEQ_DATA_START245 [3:0]				WSEQ_DATA245 [7:0]								
R12780 (31ECh)	WSEQ_Sequence_247	WSEQ_DATA_WIDTH246 [2:0]			WSEQ_ADDR246 [12:0]													FFFFFFFFh
		WSEQ_DELAY246 [3:0]				WSEQ_DATA_START246 [3:0]				WSEQ_DATA246 [7:0]								
R12782 (31EEh)	WSEQ_Sequence_248	WSEQ_DATA_WIDTH247 [2:0]			WSEQ_ADDR247 [12:0]													FFFFFFFFh
		WSEQ_DELAY247 [3:0]				WSEQ_DATA_START247 [3:0]				WSEQ_DATA247 [7:0]								
R12784 (31F0h)	WSEQ_Sequence_249	WSEQ_DATA_WIDTH248 [2:0]			WSEQ_ADDR248 [12:0]													FFFFFFFFh
		WSEQ_DELAY248 [3:0]				WSEQ_DATA_START248 [3:0]				WSEQ_DATA248 [7:0]								
R12786 (31F2h)	WSEQ_Sequence_250	WSEQ_DATA_WIDTH249 [2:0]			WSEQ_ADDR249 [12:0]													FFFFFFFFh
		WSEQ_DELAY249 [3:0]				WSEQ_DATA_START249 [3:0]				WSEQ_DATA249 [7:0]								
R12788 (31F4h)	WSEQ_Sequence_251	WSEQ_DATA_WIDTH250 [2:0]			WSEQ_ADDR250 [12:0]													FFFFFFFFh
		WSEQ_DELAY250 [3:0]				WSEQ_DATA_START250 [3:0]				WSEQ_DATA250 [7:0]								
R12790 (31F6h)	WSEQ_Sequence_252	WSEQ_DATA_WIDTH251 [2:0]			WSEQ_ADDR251 [12:0]													FFFFFFFFh
		WSEQ_DELAY251 [3:0]				WSEQ_DATA_START251 [3:0]				WSEQ_DATA251 [7:0]								
R131076 (20004h)	OTP_HPDET_Cal_1	HP_OFFSET_11 [7:0]								HP_OFFSET_10 [7:0]								00000000h
		HP_OFFSET_01 [7:0]								0	0	0	0	0	0	0	0	
R131078 (20006h)	OTP_HPDET_Cal_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		HP_GRADIENT_1X [7:0]								HP_GRADIENT_0X [7:0]								
R265216 (40C00h)	MIF4_SPI_CLK_ CONFIG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	MIF4_SCLK_FREQ_SEL [5:0]						
R265222 (40C06h)	MIF4_SPI_CLK_ STATUS_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		MIF4_SCLK_FREQ_STS [15:0]																
R265224 (40C08h)	MIF4_SPI_CONFIG_1	0	0	0	0	MIF4_SS_IDLE_COUNT [3:0]				0	0	0	0	MIF4_SS_DELAY_COUNT [3:0]				00000000h
		0	0	0	0	0	0	0	MIF4_3_ WIRE	0	MIF4_ DPHA	MIF4_ CPHA	MIF4_ CPOL	0	MIF4_SS_SEL [2:0]			
R265226 (40C0Ah)	MIF4_SPI_CONFIG_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MIF4_SS_ OVD	
R265228 (40C0Ch)	MIF4_SPI_CONFIG_4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MIF4_ WDT_ENA	
R265344 (40C80h)	MIF4_SPI_STATUS_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	MIF4_ ABORT_ STS	MIF4_ DONE_ STS	
R265346 (40C82h)	MIF4_SPI_STATUS_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MIF4_ STALL_ STS	
R265472 (40D00h)	MIF4_CONFIG_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MIF4_ START	
R265474 (40D02h)	MIF4_CONFIG_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MIF4_ ABORT	
R265478 (40D06h)	MIF4_CONFIG_4	0	0	0	0	0	0	0	0	0	0	0	MIF4_TX_LENGTH [20:16]					00000000h
		MIF4_TX_LENGTH [15:0]																
R265488 (40D10h)	MIF4_CONFIG_5	0	0	0	0	0	0	0	0	0	0	0	MIF4_RX_LENGTH [20:16]					00000000h
		MIF4_RX_LENGTH [15:0]																
R265490 (40D12h)	MIF4_CONFIG_6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	MIF4_READ_WRITE_ SEL [1:0]		0	0	0	MIF4_TX_BLOCK_LENGTH [6:0]							

Table 6-2. Register Map Definition—32-bit region *(Cont.)*

Register	Name	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0	Default
R265492 (40D14h)	MIF4_CONFIG_7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	MIF4_RX_BLOCK_LENGTH [6:0]							
R265494 (40D16h)	MIF4_CONFIG_8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	MIF4_WORD_SIZE [2:0]			
R265496 (40D18h)	MIF4_CONFIG_9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	MIF4_RX_DONE (U)	0	0	0	0	0	0	0	MIF4_TX_DONE	
R265600 (40D80h)	MIF4_STATUS_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MIF4_BUSY_STS	00000001h
		0	0	0	0	0	0	0	MIF4_RX_REQUEST	0	0	0	0	0	0	0	MIF4_TX_REQUEST	
R265602 (40D82h)	MIF4_STATUS_2	0	0	0	0	0	0	0	0	0	0	0	MIF4_TX_BYTE_COUNT [20:16]					00000000h
		MIF4_TX_BYTE_COUNT [15:0]																
R265604 (40D84h)	MIF4_STATUS_3	0	0	0	0	0	0	0	0	0	0	0	MIF4_RX_BYTE_COUNT [20:16]					00000000h
		MIF4_RX_BYTE_COUNT [15:0]																
R265728 (40E00h)	MIF4_TX_1	MIF4_TX_BYTE4 [7:0]								MIF4_TX_BYTE3 [7:0]								00000000h
		MIF4_TX_BYTE2 [7:0]								MIF4_TX_BYTE1 [7:0]								
R265730 (40E02h)	MIF4_TX_2	MIF4_TX_BYTE8 [7:0]								MIF4_TX_BYTE7 [7:0]								00000000h
		MIF4_TX_BYTE6 [7:0]								MIF4_TX_BYTE5 [7:0]								
R265732 (40E04h)	MIF4_TX_3	MIF4_TX_BYTE12 [7:0]								MIF4_TX_BYTE11 [7:0]								00000000h
		MIF4_TX_BYTE10 [7:0]								MIF4_TX_BYTE9 [7:0]								
R265734 (40E06h)	MIF4_TX_4	MIF4_TX_BYTE16 [7:0]								MIF4_TX_BYTE15 [7:0]								00000000h
		MIF4_TX_BYTE14 [7:0]								MIF4_TX_BYTE13 [7:0]								
R265736 (40E08h)	MIF4_TX_5	MIF4_TX_BYTE20 [7:0]								MIF4_TX_BYTE19 [7:0]								00000000h
		MIF4_TX_BYTE18 [7:0]								MIF4_TX_BYTE17 [7:0]								
R265738 (40E0Ah)	MIF4_TX_6	MIF4_TX_BYTE24 [7:0]								MIF4_TX_BYTE23 [7:0]								00000000h
		MIF4_TX_BYTE22 [7:0]								MIF4_TX_BYTE21 [7:0]								
R265740 (40E0Ch)	MIF4_TX_7	MIF4_TX_BYTE28 [7:0]								MIF4_TX_BYTE27 [7:0]								00000000h
		MIF4_TX_BYTE26 [7:0]								MIF4_TX_BYTE25 [7:0]								
R265742 (40E0Eh)	MIF4_TX_8	MIF4_TX_BYTE32 [7:0]								MIF4_TX_BYTE31 [7:0]								00000000h
		MIF4_TX_BYTE30 [7:0]								MIF4_TX_BYTE29 [7:0]								
R265744 (40E10h)	MIF4_TX_9	MIF4_TX_BYTE36 [7:0]								MIF4_TX_BYTE35 [7:0]								00000000h
		MIF4_TX_BYTE34 [7:0]								MIF4_TX_BYTE33 [7:0]								
R265746 (40E12h)	MIF4_TX_10	MIF4_TX_BYTE40 [7:0]								MIF4_TX_BYTE39 [7:0]								00000000h
		MIF4_TX_BYTE38 [7:0]								MIF4_TX_BYTE37 [7:0]								
R265748 (40E14h)	MIF4_TX_11	MIF4_TX_BYTE44 [7:0]								MIF4_TX_BYTE43 [7:0]								00000000h
		MIF4_TX_BYTE42 [7:0]								MIF4_TX_BYTE41 [7:0]								
R265750 (40E16h)	MIF4_TX_12	MIF4_TX_BYTE48 [7:0]								MIF4_TX_BYTE47 [7:0]								00000000h
		MIF4_TX_BYTE46 [7:0]								MIF4_TX_BYTE45 [7:0]								
R265752 (40E18h)	MIF4_TX_13	MIF4_TX_BYTE52 [7:0]								MIF4_TX_BYTE51 [7:0]								00000000h
		MIF4_TX_BYTE50 [7:0]								MIF4_TX_BYTE49 [7:0]								
R265754 (40E1Ah)	MIF4_TX_14	MIF4_TX_BYTE56 [7:0]								MIF4_TX_BYTE55 [7:0]								00000000h
		MIF4_TX_BYTE54 [7:0]								MIF4_TX_BYTE53 [7:0]								
R265756 (40E1Ch)	MIF4_TX_15	MIF4_TX_BYTE60 [7:0]								MIF4_TX_BYTE59 [7:0]								00000000h
		MIF4_TX_BYTE58 [7:0]								MIF4_TX_BYTE57 [7:0]								
R265758 (40E1Eh)	MIF4_TX_16	MIF4_TX_BYTE64 [7:0]								MIF4_TX_BYTE63 [7:0]								00000000h
		MIF4_TX_BYTE62 [7:0]								MIF4_TX_BYTE61 [7:0]								
R265984 (40F00h)	MIF4_RX_1	MIF4_RX_BYTE4 [7:0]								MIF4_RX_BYTE3 [7:0]								00000000h
		MIF4_RX_BYTE2 [7:0]								MIF4_RX_BYTE1 [7:0]								
R265986 (40F02h)	MIF4_RX_2	MIF4_RX_BYTE8 [7:0]								MIF4_RX_BYTE7 [7:0]								00000000h
		MIF4_RX_BYTE6 [7:0]								MIF4_RX_BYTE5 [7:0]								
R265988 (40F04h)	MIF4_RX_3	MIF4_RX_BYTE12 [7:0]								MIF4_RX_BYTE11 [7:0]								00000000h
		MIF4_RX_BYTE10 [7:0]								MIF4_RX_BYTE9 [7:0]								
R265990 (40F06h)	MIF4_RX_4	MIF4_RX_BYTE16 [7:0]								MIF4_RX_BYTE15 [7:0]								00000000h
		MIF4_RX_BYTE14 [7:0]								MIF4_RX_BYTE13 [7:0]								
R265992 (40F08h)	MIF4_RX_5	MIF4_RX_BYTE20 [7:0]								MIF4_RX_BYTE19 [7:0]								00000000h
		MIF4_RX_BYTE18 [7:0]								MIF4_RX_BYTE17 [7:0]								
R265994 (40F0Ah)	MIF4_RX_6	MIF4_RX_BYTE24 [7:0]								MIF4_RX_BYTE23 [7:0]								00000000h
		MIF4_RX_BYTE22 [7:0]								MIF4_RX_BYTE21 [7:0]								
R265996 (40F0Ch)	MIF4_RX_7	MIF4_RX_BYTE28 [7:0]								MIF4_RX_BYTE27 [7:0]								00000000h
		MIF4_RX_BYTE26 [7:0]								MIF4_RX_BYTE25 [7:0]								
R265998 (40F0Eh)	MIF4_RX_8	MIF4_RX_BYTE32 [7:0]								MIF4_RX_BYTE31 [7:0]								00000000h
		MIF4_RX_BYTE30 [7:0]								MIF4_RX_BYTE29 [7:0]								
R266000 (40F10h)	MIF4_RX_9	MIF4_RX_BYTE36 [7:0]								MIF4_RX_BYTE35 [7:0]								00000000h
		MIF4_RX_BYTE34 [7:0]								MIF4_RX_BYTE33 [7:0]								
R266002 (40F12h)	MIF4_RX_10	MIF4_RX_BYTE40 [7:0]								MIF4_RX_BYTE39 [7:0]								00000000h
		MIF4_RX_BYTE38 [7:0]								MIF4_RX_BYTE37 [7:0]								
R266004 (40F14h)	MIF4_RX_11	MIF4_RX_BYTE44 [7:0]								MIF4_RX_BYTE43 [7:0]								00000000h
		MIF4_RX_BYTE42 [7:0]								MIF4_RX_BYTE41 [7:0]								
R266006 (40F16h)	MIF4_RX_12	MIF4_RX_BYTE48 [7:0]								MIF4_RX_BYTE47 [7:0]								00000000h
		MIF4_RX_BYTE46 [7:0]								MIF4_RX_BYTE45 [7:0]								
R266008 (40F18h)	MIF4_RX_13	MIF4_RX_BYTE52 [7:0]								MIF4_RX_BYTE51 [7:0]								00000000h
		MIF4_RX_BYTE50 [7:0]								MIF4_RX_BYTE49 [7:0]								

Table 6-2. Register Map Definition—32-bit region *(Cont.)*

Register	Name	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0	Default
R266010 (40F1Ah)	MIF4_RX_14	MIF4_RX_BYTE56 [7:0]								MIF4_RX_BYTE55 [7:0]								00000000h
		MIF4_RX_BYTE54 [7:0]								MIF4_RX_BYTE53 [7:0]								
R266012 (40F1Ch)	MIF4_RX_15	MIF4_RX_BYTE60 [7:0]								MIF4_RX_BYTE59 [7:0]								00000000h
		MIF4_RX_BYTE58 [7:0]								MIF4_RX_BYTE57 [7:0]								
R266014 (40F1Eh)	MIF4_RX_16	MIF4_RX_BYTE64 [7:0]								MIF4_RX_BYTE63 [7:0]								00000000h
		MIF4_RX_BYTE62 [7:0]								MIF4_RX_BYTE61 [7:0]								
R294912 (48000h)	EVENTLOG1_CONTROL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	EVENTLOG1_FLL_AO_CLKENA	0	0	0	0	0	0	EVENTLOG1_RST	EVENTLOG1_ENA	
R294916 (48004h)	EVENTLOG1_TIMER_SEL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENTLOG1_TIMER_SEL [1:0]		
R294924 (4800Ch)	EVENTLOG1_FIFO_CONTROL1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000001h
		0	0	0	0	0	0	0	0	0	0	0	0	EVENTLOG1_FIFO_WMARK [3:0]				
R294926 (4800Eh)	EVENTLOG1_FIFO_POINTER1	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENTLOG1_FULL	EVENTLOG1_WMARK_STS	EVENTLOG1_NOT_EMPTY	00000000h
		0	0	0	0	EVENTLOG1_FIFO_WPTR [3:0]				0	0	0	0	EVENTLOG1_FIFO_RPTR [3:0]				
R294944 (48020h)	EVENTLOG1_CH_ENABLE1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH16_ENA	EVENTLOG1_CH15_ENA	EVENTLOG1_CH14_ENA	EVENTLOG1_CH13_ENA	EVENTLOG1_CH12_ENA	EVENTLOG1_CH11_ENA	EVENTLOG1_CH10_ENA	EVENTLOG1_CH9_ENA	EVENTLOG1_CH8_ENA	EVENTLOG1_CH7_ENA	EVENTLOG1_CH6_ENA	EVENTLOG1_CH5_ENA	EVENTLOG1_CH4_ENA	EVENTLOG1_CH3_ENA	EVENTLOG1_CH2_ENA	EVENTLOG1_CH1_ENA	
R294976 (48040h)	EVENTLOG1_CH1_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH1_DB	EVENTLOG1_CH1_POL	0	0	0	0	EVENTLOG1_CH1_SEL [9:0]										
R294978 (48042h)	EVENTLOG1_CH2_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH2_DB	EVENTLOG1_CH2_POL	0	0	0	0	EVENTLOG1_CH2_SEL [9:0]										
R294980 (48044h)	EVENTLOG1_CH3_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH3_DB	EVENTLOG1_CH3_POL	0	0	0	0	EVENTLOG1_CH3_SEL [9:0]										
R294982 (48046h)	EVENTLOG1_CH4_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH4_DB	EVENTLOG1_CH4_POL	0	0	0	0	EVENTLOG1_CH4_SEL [9:0]										
R294984 (48048h)	EVENTLOG1_CH5_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH5_DB	EVENTLOG1_CH5_POL	0	0	0	0	EVENTLOG1_CH5_SEL [9:0]										
R294986 (4804Ah)	EVENTLOG1_CH6_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH6_DB	EVENTLOG1_CH6_POL	0	0	0	0	EVENTLOG1_CH6_SEL [9:0]										
R294988 (4804Ch)	EVENTLOG1_CH7_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH7_DB	EVENTLOG1_CH7_POL	0	0	0	0	EVENTLOG1_CH7_SEL [9:0]										
R294990 (4804Eh)	EVENTLOG1_CH8_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH8_DB	EVENTLOG1_CH8_POL	0	0	0	0	EVENTLOG1_CH8_SEL [9:0]										
R294992 (48050h)	EVENTLOG1_CH9_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH9_DB	EVENTLOG1_CH9_POL	0	0	0	0	EVENTLOG1_CH9_SEL [9:0]										
R294994 (48052h)	EVENTLOG1_CH10_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH10_DB	EVENTLOG1_CH10_POL	0	0	0	0	EVENTLOG1_CH10_SEL [9:0]										
R294996 (48054h)	EVENTLOG1_CH11_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH11_DB	EVENTLOG1_CH11_POL	0	0	0	0	EVENTLOG1_CH11_SEL [9:0]										
R294998 (48056h)	EVENTLOG1_CH12_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH12_DB	EVENTLOG1_CH12_POL	0	0	0	0	EVENTLOG1_CH12_SEL [9:0]										
R295000 (48058h)	EVENTLOG1_CH13_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH13_DB	EVENTLOG1_CH13_POL	0	0	0	0	EVENTLOG1_CH13_SEL [9:0]										
R295002 (4805Ah)	EVENTLOG1_CH14_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH14_DB	EVENTLOG1_CH14_POL	0	0	0	0	EVENTLOG1_CH14_SEL [9:0]										
R295004 (4805Ch)	EVENTLOG1_CH15_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH15_DB	EVENTLOG1_CH15_POL	0	0	0	0	EVENTLOG1_CH15_SEL [9:0]										

Table 6-2. Register Map Definition—32-bit region *(Cont.)*

Register	Name	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0	Default
R295006 (4805Eh)	EVENTLOG1_CH16_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH16_DB	EVENTLOG1_CH16_POL	0	0	0	0	EVENTLOG1_CH16_SEL [9:0]										
R295040 (48080h)	EVENTLOG1_FIFO0_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO0_POL	0	0	EVENTLOG1_FIFO0_ID [9:0]										
R295042 (48082h)	EVENTLOG1_FIFO0_TIME	EVENTLOG1_FIFO0_TIME [31:16]																00000000h
		EVENTLOG1_FIFO0_TIME [15:0]																
R295044 (48084h)	EVENTLOG1_FIFO1_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO1_POL	0	0	EVENTLOG1_FIFO1_ID [9:0]										
R295046 (48086h)	EVENTLOG1_FIFO1_TIME	EVENTLOG1_FIFO1_TIME [31:16]																00000000h
		EVENTLOG1_FIFO1_TIME [15:0]																
R295048 (48088h)	EVENTLOG1_FIFO2_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO2_POL	0	0	EVENTLOG1_FIFO2_ID [9:0]										
R295050 (4808Ah)	EVENTLOG1_FIFO2_TIME	EVENTLOG1_FIFO2_TIME [31:16]																00000000h
		EVENTLOG1_FIFO2_TIME [15:0]																
R295052 (4808Ch)	EVENTLOG1_FIFO3_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO3_POL	0	0	EVENTLOG1_FIFO3_ID [9:0]										
R295054 (4808Eh)	EVENTLOG1_FIFO3_TIME	EVENTLOG1_FIFO3_TIME [31:16]																00000000h
		EVENTLOG1_FIFO3_TIME [15:0]																
R295056 (48090h)	EVENTLOG1_FIFO4_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO4_POL	0	0	EVENTLOG1_FIFO4_ID [9:0]										
R295058 (48092h)	EVENTLOG1_FIFO4_TIME	EVENTLOG1_FIFO4_TIME [31:16]																00000000h
		EVENTLOG1_FIFO4_TIME [15:0]																
R295060 (48094h)	EVENTLOG1_FIFO5_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO5_POL	0	0	EVENTLOG1_FIFO5_ID [9:0]										
R295062 (48096h)	EVENTLOG1_FIFO5_TIME	EVENTLOG1_FIFO5_TIME [31:16]																00000000h
		EVENTLOG1_FIFO5_TIME [15:0]																
R295064 (48098h)	EVENTLOG1_FIFO6_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO6_POL	0	0	EVENTLOG1_FIFO6_ID [9:0]										
R295066 (4809Ah)	EVENTLOG1_FIFO6_TIME	EVENTLOG1_FIFO6_TIME [31:16]																00000000h
		EVENTLOG1_FIFO6_TIME [15:0]																
R295068 (4809Ch)	EVENTLOG1_FIFO7_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO7_POL	0	0	EVENTLOG1_FIFO7_ID [9:0]										
R295070 (4809Eh)	EVENTLOG1_FIFO7_TIME	EVENTLOG1_FIFO7_TIME [31:16]																00000000h
		EVENTLOG1_FIFO7_TIME [15:0]																
R295072 (480A0h)	EVENTLOG1_FIFO8_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO8_POL	0	0	EVENTLOG1_FIFO8_ID [9:0]										
R295074 (480A2h)	EVENTLOG1_FIFO8_TIME	EVENTLOG1_FIFO8_TIME [31:16]																00000000h
		EVENTLOG1_FIFO8_TIME [15:0]																
R295076 (480A4h)	EVENTLOG1_FIFO9_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO9_POL	0	0	EVENTLOG1_FIFO9_ID [9:0]										
R295078 (480A6h)	EVENTLOG1_FIFO9_TIME	EVENTLOG1_FIFO9_TIME [31:16]																00000000h
		EVENTLOG1_FIFO9_TIME [15:0]																
R295080 (480A8h)	EVENTLOG1_FIFO10_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO10_POL	0	0	EVENTLOG1_FIFO10_ID [9:0]										
R295082 (480AAh)	EVENTLOG1_FIFO10_TIME	EVENTLOG1_FIFO10_TIME [31:16]																00000000h
		EVENTLOG1_FIFO10_TIME [15:0]																

Table 6-2. Register Map Definition—32-bit region *(Cont.)*

Register	Name	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0	Default
R295084 (480ACh)	EVENTLOG1_FIFO11_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO11_POL	0	0	EVENTLOG1_FIFO11_ID [9:0]										
R295086 (480AEh)	EVENTLOG1_FIFO11_TIME	EVENTLOG1_FIFO11_TIME [31:16]																00000000h
		EVENTLOG1_FIFO11_TIME [15:0]																
R295088 (480B0h)	EVENTLOG1_FIFO12_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO12_POL	0	0	EVENTLOG1_FIFO12_ID [9:0]										
R295090 (480B2h)	EVENTLOG1_FIFO12_TIME	EVENTLOG1_FIFO12_TIME [31:16]																00000000h
		EVENTLOG1_FIFO12_TIME [15:0]																
R295092 (480B4h)	EVENTLOG1_FIFO13_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO13_POL	0	0	EVENTLOG1_FIFO13_ID [9:0]										
R295094 (480B6h)	EVENTLOG1_FIFO13_TIME	EVENTLOG1_FIFO13_TIME [31:16]																00000000h
		EVENTLOG1_FIFO13_TIME [15:0]																
R295096 (480B8h)	EVENTLOG1_FIFO14_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO14_POL	0	0	EVENTLOG1_FIFO14_ID [9:0]										
R295098 (480BAh)	EVENTLOG1_FIFO14_TIME	EVENTLOG1_FIFO14_TIME [31:16]																00000000h
		EVENTLOG1_FIFO14_TIME [15:0]																
R295100 (480BCh)	EVENTLOG1_FIFO15_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO15_POL	0	0	EVENTLOG1_FIFO15_ID [9:0]										
R295102 (480BEh)	EVENTLOG1_FIFO15_TIME	EVENTLOG1_FIFO15_TIME [31:16]																00000000h
		EVENTLOG1_FIFO15_TIME [15:0]																
R295424 (48200h)	EVENTLOG2_CONTROL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	EVENTLOG2_FLL_AO_CLKENA	0	0	0	0	0	0	EVENTLOG2_RST	EVENTLOG2_ENA	
R295428 (48204h)	EVENTLOG2_TIMER_SEL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENTLOG2_TIMER_SEL [1:0]		
R295436 (4820Ch)	EVENTLOG2_FIFO_CONTROL1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000001h
		0	0	0	0	0	0	0	0	0	0	0	0	EVENTLOG2_FIFO_WMARK [3:0]				
R295438 (4820Eh)	EVENTLOG2_FIFO_POINTER1	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENTLOG2_FULL	EVENTLOG2_WMARK_STS	EVENTLOG2_NOT_EMPTY	00000000h
		0	0	0	0	EVENTLOG2_FIFO_WPTR [3:0]				0	0	0	0	EVENTLOG2_FIFO_RPTR [3:0]				
R295456 (48220h)	EVENTLOG2_CH_ENABLE1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG2_CH16_ENA	EVENTLOG2_CH15_ENA	EVENTLOG2_CH14_ENA	EVENTLOG2_CH13_ENA	EVENTLOG2_CH12_ENA	EVENTLOG2_CH11_ENA	EVENTLOG2_CH10_ENA	EVENTLOG2_CH9_ENA	EVENTLOG2_CH8_ENA	EVENTLOG2_CH7_ENA	EVENTLOG2_CH6_ENA	EVENTLOG2_CH5_ENA	EVENTLOG2_CH4_ENA	EVENTLOG2_CH3_ENA	EVENTLOG2_CH2_ENA	EVENTLOG2_CH1_ENA	
R295488 (48240h)	EVENTLOG2_CH1_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG2_CH1_DB	EVENTLOG2_CH1_POL	0	0	0	0	EVENTLOG2_CH1_SEL [9:0]										
R295490 (48242h)	EVENTLOG2_CH2_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG2_CH2_DB	EVENTLOG2_CH2_POL	0	0	0	0	EVENTLOG2_CH2_SEL [9:0]										
R295492 (48244h)	EVENTLOG2_CH3_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG2_CH3_DB	EVENTLOG2_CH3_POL	0	0	0	0	EVENTLOG2_CH3_SEL [9:0]										
R295494 (48246h)	EVENTLOG2_CH4_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG2_CH4_DB	EVENTLOG2_CH4_POL	0	0	0	0	EVENTLOG2_CH4_SEL [9:0]										
R295496 (48248h)	EVENTLOG2_CH5_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG2_CH5_DB	EVENTLOG2_CH5_POL	0	0	0	0	EVENTLOG2_CH5_SEL [9:0]										
R295498 (4824Ah)	EVENTLOG2_CH6_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG2_CH6_DB	EVENTLOG2_CH6_POL	0	0	0	0	EVENTLOG2_CH6_SEL [9:0]										
R295500 (4824Ch)	EVENTLOG2_CH7_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG2_CH7_DB	EVENTLOG2_CH7_POL	0	0	0	0	EVENTLOG2_CH7_SEL [9:0]										

Table 6-2. Register Map Definition—32-bit region *(Cont.)*

Register	Name	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0	Default
R295502 (4824Eh)	EVENTLOG2_CH8_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG2_CH8_DB	EVENTLOG2_CH8_POL	0	0	0	0	EVENTLOG2_CH8_SEL [9:0]										
R295504 (48250h)	EVENTLOG2_CH9_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG2_CH9_DB	EVENTLOG2_CH9_POL	0	0	0	0	EVENTLOG2_CH9_SEL [9:0]										
R295506 (48252h)	EVENTLOG2_CH10_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG2_CH10_DB	EVENTLOG2_CH10_POL	0	0	0	0	EVENTLOG2_CH10_SEL [9:0]										
R295508 (48254h)	EVENTLOG2_CH11_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG2_CH11_DB	EVENTLOG2_CH11_POL	0	0	0	0	EVENTLOG2_CH11_SEL [9:0]										
R295510 (48256h)	EVENTLOG2_CH12_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG2_CH12_DB	EVENTLOG2_CH12_POL	0	0	0	0	EVENTLOG2_CH12_SEL [9:0]										
R295512 (48258h)	EVENTLOG2_CH13_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG2_CH13_DB	EVENTLOG2_CH13_POL	0	0	0	0	EVENTLOG2_CH13_SEL [9:0]										
R295514 (4825Ah)	EVENTLOG2_CH14_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG2_CH14_DB	EVENTLOG2_CH14_POL	0	0	0	0	EVENTLOG2_CH14_SEL [9:0]										
R295516 (4825Ch)	EVENTLOG2_CH15_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG2_CH15_DB	EVENTLOG2_CH15_POL	0	0	0	0	EVENTLOG2_CH15_SEL [9:0]										
R295518 (4825Eh)	EVENTLOG2_CH16_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG2_CH16_DB	EVENTLOG2_CH16_POL	0	0	0	0	EVENTLOG2_CH16_SEL [9:0]										
R295552 (48280h)	EVENTLOG2_FIFO0_ READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG2_FIFO0_POL	0	0	EVENTLOG2_FIFO0_ID [9:0]										
R295554 (48282h)	EVENTLOG2_FIFO0_ TIME	EVENTLOG2_FIFO0_TIME [31:16]																00000000h
		EVENTLOG2_FIFO0_TIME [15:0]																
R295556 (48284h)	EVENTLOG2_FIFO1_ READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG2_FIFO1_POL	0	0	EVENTLOG2_FIFO1_ID [9:0]										
R295558 (48286h)	EVENTLOG2_FIFO1_ TIME	EVENTLOG2_FIFO1_TIME [31:16]																00000000h
		EVENTLOG2_FIFO1_TIME [15:0]																
R295560 (48288h)	EVENTLOG2_FIFO2_ READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG2_FIFO2_POL	0	0	EVENTLOG2_FIFO2_ID [9:0]										
R295562 (4828Ah)	EVENTLOG2_FIFO2_ TIME	EVENTLOG2_FIFO2_TIME [31:16]																00000000h
		EVENTLOG2_FIFO2_TIME [15:0]																
R295564 (4828Ch)	EVENTLOG2_FIFO3_ READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG2_FIFO3_POL	0	0	EVENTLOG2_FIFO3_ID [9:0]										
R295566 (4828Eh)	EVENTLOG2_FIFO3_ TIME	EVENTLOG2_FIFO3_TIME [31:16]																00000000h
		EVENTLOG2_FIFO3_TIME [15:0]																
R295568 (48290h)	EVENTLOG2_FIFO4_ READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG2_FIFO4_POL	0	0	EVENTLOG2_FIFO4_ID [9:0]										
R295570 (48292h)	EVENTLOG2_FIFO4_ TIME	EVENTLOG2_FIFO4_TIME [31:16]																00000000h
		EVENTLOG2_FIFO4_TIME [15:0]																
R295572 (48294h)	EVENTLOG2_FIFO5_ READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG2_FIFO5_POL	0	0	EVENTLOG2_FIFO5_ID [9:0]										
R295574 (48296h)	EVENTLOG2_FIFO5_ TIME	EVENTLOG2_FIFO5_TIME [31:16]																00000000h
		EVENTLOG2_FIFO5_TIME [15:0]																
R295576 (48298h)	EVENTLOG2_FIFO6_ READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG2_FIFO6_POL	0	0	EVENTLOG2_FIFO6_ID [9:0]										
R295578 (4829Ah)	EVENTLOG2_FIFO6_ TIME	EVENTLOG2_FIFO6_TIME [31:16]																00000000h
		EVENTLOG2_FIFO6_TIME [15:0]																

Table 6-2. Register Map Definition—32-bit region *(Cont.)*

Register	Name	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0	Default
R295580 (4829Ch)	EVENTLOG2_FIFO7_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG2_FIFO7_POL	0	0	EVENTLOG2_FIFO7_ID [9:0]										
R295582 (4829Eh)	EVENTLOG2_FIFO7_TIME	EVENTLOG2_FIFO7_TIME [31:16]																00000000h
		EVENTLOG2_FIFO7_TIME [15:0]																
R295584 (482A0h)	EVENTLOG2_FIFO8_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG2_FIFO8_POL	0	0	EVENTLOG2_FIFO8_ID [9:0]										
R295586 (482A2h)	EVENTLOG2_FIFO8_TIME	EVENTLOG2_FIFO8_TIME [31:16]																00000000h
		EVENTLOG2_FIFO8_TIME [15:0]																
R295588 (482A4h)	EVENTLOG2_FIFO9_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG2_FIFO9_POL	0	0	EVENTLOG2_FIFO9_ID [9:0]										
R295590 (482A6h)	EVENTLOG2_FIFO9_TIME	EVENTLOG2_FIFO9_TIME [31:16]																00000000h
		EVENTLOG2_FIFO9_TIME [15:0]																
R295592 (482A8h)	EVENTLOG2_FIFO10_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG2_FIFO10_POL	0	0	EVENTLOG2_FIFO10_ID [9:0]										
R295594 (482AAh)	EVENTLOG2_FIFO10_TIME	EVENTLOG2_FIFO10_TIME [31:16]																00000000h
		EVENTLOG2_FIFO10_TIME [15:0]																
R295596 (482ACh)	EVENTLOG2_FIFO11_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG2_FIFO11_POL	0	0	EVENTLOG2_FIFO11_ID [9:0]										
R295598 (482AEh)	EVENTLOG2_FIFO11_TIME	EVENTLOG2_FIFO11_TIME [31:16]																00000000h
		EVENTLOG2_FIFO11_TIME [15:0]																
R295600 (482B0h)	EVENTLOG2_FIFO12_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG2_FIFO12_POL	0	0	EVENTLOG2_FIFO12_ID [9:0]										
R295602 (482B2h)	EVENTLOG2_FIFO12_TIME	EVENTLOG2_FIFO12_TIME [31:16]																00000000h
		EVENTLOG2_FIFO12_TIME [15:0]																
R295604 (482B4h)	EVENTLOG2_FIFO13_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG2_FIFO13_POL	0	0	EVENTLOG2_FIFO13_ID [9:0]										
R295606 (482B6h)	EVENTLOG2_FIFO13_TIME	EVENTLOG2_FIFO13_TIME [31:16]																00000000h
		EVENTLOG2_FIFO13_TIME [15:0]																
R295608 (482B8h)	EVENTLOG2_FIFO14_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG2_FIFO14_POL	0	0	EVENTLOG2_FIFO14_ID [9:0]										
R295610 (482BAh)	EVENTLOG2_FIFO14_TIME	EVENTLOG2_FIFO14_TIME [31:16]																00000000h
		EVENTLOG2_FIFO14_TIME [15:0]																
R295612 (482BCh)	EVENTLOG2_FIFO15_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG2_FIFO15_POL	0	0	EVENTLOG2_FIFO15_ID [9:0]										
R295614 (482BEh)	EVENTLOG2_FIFO15_TIME	EVENTLOG2_FIFO15_TIME [31:16]																00000000h
		EVENTLOG2_FIFO15_TIME [15:0]																
R311296 (4C000h)	Timer1_Control	0	0	0	0	0	0	0	0	0	0	TIMER1_CONTINUOUS	TIMER1_DIR	0	TIMER1_PRESCALE [2:0]			00000000h
		0	TIMER1_REFCLK_DIV [2:0]			0	TIMER1_REFCLK_FREQ_SEL [2:0]			0	0	0	0	TIMER1_REFCLK_SRC [3:0]				
R311298 (4C002h)	Timer1_Count_Preset	TIMER1_MAX_COUNT [31:16]																00000000h
		TIMER1_MAX_COUNT [15:0]																
R311302 (4C006h)	Timer1_Start_and_Stop	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	TIMER1_STOP	0	0	0	TIMER1_START	
R311304 (4C008h)	Timer1_Status	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TIMER1_RUNNING_STS	
R311306 (4C00Ah)	Timer1_Count_Readback	TIMER1_CUR_COUNT [31:16]																00000000h
		TIMER1_CUR_COUNT [15:0]																
R311308 (4C00Ch)	Timer1_DSP_Clock_Config	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		TIMER1_DSPCLK_FREQ_SEL [15:0]																
R311310 (4C00Eh)	Timer1_DSP_Clock_Status	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		TIMER1_DSPCLK_FREQ_STS [15:0]																

Table 6-2. Register Map Definition—32-bit region *(Cont.)*

Register	Name	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0	Default
R311424 (4C080h)	Timer2_Control	0	0	0	0	0	0	0	0	0	0	TIMER2_CONTINUOUS	TIMER2_DIR	0	TIMER2_PRESCALE [2:0]			00000000h
		0	TIMER2_REFCLK_DIV [2:0]			0	TIMER2_REFCLK_FREQ_SEL [2:0]			0	0	0	0	TIMER2_REFCLK_SRC [3:0]				
R311426 (4C082h)	Timer2_Count_Preset	TIMER2_MAX_COUNT [31:16]																00000000h
		TIMER2_MAX_COUNT [15:0]																
R311430 (4C086h)	Timer2_Start_and_Stop	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	TIMER2_STOP	0	0	0	TIMER2_START	
R311432 (4C088h)	Timer2_Status	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TIMER2_RUNNING_STS	
R311434 (4C08Ah)	Timer2_Count_Readback	TIMER2_CUR_COUNT [31:16]																00000000h
		TIMER2_CUR_COUNT [15:0]																
R311436 (4C08Ch)	Timer2_DSP_Clock_Config	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		TIMER2_DSPCLK_FREQ_SEL [15:0]																
R311438 (4C08Eh)	Timer2_DSP_Clock_Status	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		TIMER2_DSPCLK_FREQ_STS [15:0]																
R315392 (4D000h)	DSPGP_Status_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	DSPGP15_STS	DSPGP14_STS	DSPGP13_STS	DSPGP12_STS	DSPGP11_STS	DSPGP10_STS	DSPGP9_STS	DSPGP8_STS	DSPGP7_STS	DSPGP6_STS	DSPGP5_STS	DSPGP4_STS	DSPGP3_STS	DSPGP2_STS	DSPGP1_STS	
R315424 (4D020h)	DSPGP_SET1_Mask_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00007FFFh
		0	DSPGP15_SET1_MASK	DSPGP14_SET1_MASK	DSPGP13_SET1_MASK	DSPGP12_SET1_MASK	DSPGP11_SET1_MASK	DSPGP10_SET1_MASK	DSPGP9_SET1_MASK	DSPGP8_SET1_MASK	DSPGP7_SET1_MASK	DSPGP6_SET1_MASK	DSPGP5_SET1_MASK	DSPGP4_SET1_MASK	DSPGP3_SET1_MASK	DSPGP2_SET1_MASK	DSPGP1_SET1_MASK	
R315432 (4D028h)	DSPGP_SET1_Direction_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00007FFFh
		0	DSPGP15_SET1_DIR	DSPGP14_SET1_DIR	DSPGP13_SET1_DIR	DSPGP12_SET1_DIR	DSPGP11_SET1_DIR	DSPGP10_SET1_DIR	DSPGP9_SET1_DIR	DSPGP8_SET1_DIR	DSPGP7_SET1_DIR	DSPGP6_SET1_DIR	DSPGP5_SET1_DIR	DSPGP4_SET1_DIR	DSPGP3_SET1_DIR	DSPGP2_SET1_DIR	DSPGP1_SET1_DIR	
R315440 (4D030h)	DSPGP_SET1_Level_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	DSPGP15_SET1_LVL	DSPGP14_SET1_LVL	DSPGP13_SET1_LVL	DSPGP12_SET1_LVL	DSPGP11_SET1_LVL	DSPGP10_SET1_LVL	DSPGP9_SET1_LVL	DSPGP8_SET1_LVL	DSPGP7_SET1_LVL	DSPGP6_SET1_LVL	DSPGP5_SET1_LVL	DSPGP4_SET1_LVL	DSPGP3_SET1_LVL	DSPGP2_SET1_LVL	DSPGP1_SET1_LVL	
R315456 (4D040h)	DSPGP_SET2_Mask_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00007FFFh
		0	DSPGP15_SET2_MASK	DSPGP14_SET2_MASK	DSPGP13_SET2_MASK	DSPGP12_SET2_MASK	DSPGP11_SET2_MASK	DSPGP10_SET2_MASK	DSPGP9_SET2_MASK	DSPGP8_SET2_MASK	DSPGP7_SET2_MASK	DSPGP6_SET2_MASK	DSPGP5_SET2_MASK	DSPGP4_SET2_MASK	DSPGP3_SET2_MASK	DSPGP2_SET2_MASK	DSPGP1_SET2_MASK	
R315464 (4D048h)	DSPGP_SET2_Direction_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00007FFFh
		0	DSPGP15_SET2_DIR	DSPGP14_SET2_DIR	DSPGP13_SET2_DIR	DSPGP12_SET2_DIR	DSPGP11_SET2_DIR	DSPGP10_SET2_DIR	DSPGP9_SET2_DIR	DSPGP8_SET2_DIR	DSPGP7_SET2_DIR	DSPGP6_SET2_DIR	DSPGP5_SET2_DIR	DSPGP4_SET2_DIR	DSPGP3_SET2_DIR	DSPGP2_SET2_DIR	DSPGP1_SET2_DIR	
R315472 (4D050h)	DSPGP_SET2_Level_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	DSPGP15_SET2_LVL	DSPGP14_SET2_LVL	DSPGP13_SET2_LVL	DSPGP12_SET2_LVL	DSPGP11_SET2_LVL	DSPGP10_SET2_LVL	DSPGP9_SET2_LVL	DSPGP8_SET2_LVL	DSPGP7_SET2_LVL	DSPGP6_SET2_LVL	DSPGP5_SET2_LVL	DSPGP4_SET2_LVL	DSPGP3_SET2_LVL	DSPGP2_SET2_LVL	DSPGP1_SET2_LVL	
R315488 (4D060h)	DSPGP_SET3_Mask_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00007FFFh
		0	DSPGP15_SET3_MASK	DSPGP14_SET3_MASK	DSPGP13_SET3_MASK	DSPGP12_SET3_MASK	DSPGP11_SET3_MASK	DSPGP10_SET3_MASK	DSPGP9_SET3_MASK	DSPGP8_SET3_MASK	DSPGP7_SET3_MASK	DSPGP6_SET3_MASK	DSPGP5_SET3_MASK	DSPGP4_SET3_MASK	DSPGP3_SET3_MASK	DSPGP2_SET3_MASK	DSPGP1_SET3_MASK	
R315496 (4D068h)	DSPGP_SET3_Direction_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00007FFFh
		0	DSPGP15_SET3_DIR	DSPGP14_SET3_DIR	DSPGP13_SET3_DIR	DSPGP12_SET3_DIR	DSPGP11_SET3_DIR	DSPGP10_SET3_DIR	DSPGP9_SET3_DIR	DSPGP8_SET3_DIR	DSPGP7_SET3_DIR	DSPGP6_SET3_DIR	DSPGP5_SET3_DIR	DSPGP4_SET3_DIR	DSPGP3_SET3_DIR	DSPGP2_SET3_DIR	DSPGP1_SET3_DIR	
R315504 (4D070h)	DSPGP_SET3_Level_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	DSPGP15_SET3_LVL	DSPGP14_SET3_LVL	DSPGP13_SET3_LVL	DSPGP12_SET3_LVL	DSPGP11_SET3_LVL	DSPGP10_SET3_LVL	DSPGP9_SET3_LVL	DSPGP8_SET3_LVL	DSPGP7_SET3_LVL	DSPGP6_SET3_LVL	DSPGP5_SET3_LVL	DSPGP4_SET3_LVL	DSPGP3_SET3_LVL	DSPGP2_SET3_LVL	DSPGP1_SET3_LVL	
R315520 (4D080h)	DSPGP_SET4_Mask_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00007FFFh
		0	DSPGP15_SET4_MASK	DSPGP14_SET4_MASK	DSPGP13_SET4_MASK	DSPGP12_SET4_MASK	DSPGP11_SET4_MASK	DSPGP10_SET4_MASK	DSPGP9_SET4_MASK	DSPGP8_SET4_MASK	DSPGP7_SET4_MASK	DSPGP6_SET4_MASK	DSPGP5_SET4_MASK	DSPGP4_SET4_MASK	DSPGP3_SET4_MASK	DSPGP2_SET4_MASK	DSPGP1_SET4_MASK	
R315528 (4D088h)	DSPGP_SET4_Direction_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00007FFFh
		0	DSPGP15_SET4_DIR	DSPGP14_SET4_DIR	DSPGP13_SET4_DIR	DSPGP12_SET4_DIR	DSPGP11_SET4_DIR	DSPGP10_SET4_DIR	DSPGP9_SET4_DIR	DSPGP8_SET4_DIR	DSPGP7_SET4_DIR	DSPGP6_SET4_DIR	DSPGP5_SET4_DIR	DSPGP4_SET4_DIR	DSPGP3_SET4_DIR	DSPGP2_SET4_DIR	DSPGP1_SET4_DIR	
R315536 (4D090h)	DSPGP_SET4_Level_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	DSPGP15_SET4_LVL	DSPGP14_SET4_LVL	DSPGP13_SET4_LVL	DSPGP12_SET4_LVL	DSPGP11_SET4_LVL	DSPGP10_SET4_LVL	DSPGP9_SET4_LVL	DSPGP8_SET4_LVL	DSPGP7_SET4_LVL	DSPGP6_SET4_LVL	DSPGP5_SET4_LVL	DSPGP4_SET4_LVL	DSPGP3_SET4_LVL	DSPGP2_SET4_LVL	DSPGP1_SET4_LVL	
R328704 (50400h)	RA_EVENTLOG_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000003h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	RA_EVENTLOG_STS [1:0]		
R328708 (50404h)	RA_EVENTLOG_Thread_Ctrl_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	RA_EVENTLOG_SHARE_STS [1:0]		
R328712 (50408h)	RA_EVENTLOG_Thread_Ctrl_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000002h
		0	0	0	0	0	0	0	0	0	0	RA_EVENTLOG_NUM [5:0]						
R328720 (50410h)	RA_EVENTLOG1_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_EVENTLOG1_INUSE_STS	RA_EVENTLOG1_SHARE	0	0	0	0	0	0	0	0	0	RA_EVENTLOG1_OWNER [4:0]					

Table 6-2. Register Map Definition—32-bit region *(Cont.)*

Register	Name	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0	Default
R328722 (50412h)	RA_EVENTLOG1_Thread_Ctrl_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	RA_EVENTLOG1_IN_USE_SET [4:0]					
R328724 (50414h)	RA_EVENTLOG1_Thread_Ctrl_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	RA_EVENTLOG1_IN_USE_CLR [4:0]					
R328728 (50418h)	RA_EVENTLOG1_Thread_Ctrl_Debug_1	RA_EVENTLOG1_IN_USE_DBG0 [31:16]																00000000h
		RA_EVENTLOG1_IN_USE_DBG0 [15:0]																
R328736 (50420h)	RA_EVENTLOG2_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_EVENTLOG2_IN_USE_STS	RA_EVENTLOG2_SHARE	0	0	0	0	0	0	0	0	0	RA_EVENTLOG2_OWNER [4:0]					
R328738 (50422h)	RA_EVENTLOG2_Thread_Ctrl_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	RA_EVENTLOG2_IN_USE_SET [4:0]					
R328744 (50428h)	RA_EVENTLOG2_Thread_Ctrl_Debug_1	RA_EVENTLOG2_IN_USE_DBG0 [31:16]																00000000h
		RA_EVENTLOG2_IN_USE_DBG0 [15:0]																
R329728 (50800h)	RA_TIMER_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000003h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	RA_TIMER_STS [1:0]		
R329732 (50804h)	RA_TIMER_Thread_Ctrl_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	RA_TIMER_SHARE_STS [1:0]		
R329736 (50808h)	RA_TIMER_Thread_Ctrl_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000002h
		0	0	0	0	0	0	0	0	0	0	RA_TIMER_NUM [5:0]						
R329744 (50810h)	RA_TIMER1_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_TIMER1_IN_USE_STS	RA_TIMER1_SHARE	0	0	0	0	0	0	0	0	0	RA_TIMER1_OWNER [4:0]					
R329746 (50812h)	RA_TIMER1_Thread_Ctrl_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	RA_TIMER1_IN_USE_SET [4:0]					
R329748 (50814h)	RA_TIMER1_Thread_Ctrl_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	RA_TIMER1_IN_USE_CLR [4:0]					
R329750 (50816h)	RA_TIMER1_Thread_Ctrl_4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000001h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RA_TIMER1_CAP_EVT	
R329752 (50818h)	RA_TIMER1_Thread_Ctrl_Debug_1	RA_TIMER1_IN_USE_DBG0 [31:16]																00000000h
		RA_TIMER1_IN_USE_DBG0 [15:0]																
R329760 (50820h)	RA_TIMER2_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_TIMER2_IN_USE_STS	RA_TIMER2_SHARE	0	0	0	0	0	0	0	0	0	RA_TIMER2_OWNER [4:0]					
R329762 (50822h)	RA_TIMER2_Thread_Ctrl_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	RA_TIMER2_IN_USE_SET [4:0]					
R329764 (50824h)	RA_TIMER2_Thread_Ctrl_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	RA_TIMER2_IN_USE_CLR [4:0]					
R329766 (50826h)	RA_TIMER2_Thread_Ctrl_4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000001h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RA_TIMER2_CAP_EVT	
R329768 (50828h)	RA_TIMER2_Thread_Ctrl_Debug_1	RA_TIMER2_IN_USE_DBG0 [31:16]																00000000h
		RA_TIMER2_IN_USE_DBG0 [15:0]																
R330752 (50C00h)	RA_DSPGP_SET_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	000000FFh
		0	0	0	0	0	0	0	0	RA_DSPGP_SET_STS [7:0]								
R330756 (50C04h)	RA_DSPGP_SET_Thread_Ctrl_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	RA_DSPGP_SET_SHARE_STS [7:0]								
R330760 (50C08h)	RA_DSPGP_SET_Thread_Ctrl_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000008h
		0	0	0	0	0	0	0	0	0	0	RA_DSPGP_SET_NUM [5:0]						
R330768 (50C10h)	RA_DSPGP_SET1_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_DSPGP_SET1_IN_USE_STS	RA_DSPGP_SET1_SHARE	0	0	0	0	0	0	0	0	0	RA_DSPGP_SET1_OWNER [4:0]					
R330770 (50C12h)	RA_DSPGP_SET1_Thread_Ctrl_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	RA_DSPGP_SET1_IN_USE_SET [4:0]					
R330772 (50C14h)	RA_DSPGP_SET1_Thread_Ctrl_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	RA_DSPGP_SET1_IN_USE_CLR [4:0]					
R330776 (50C18h)	RA_DSPGP_SET1_Thread_Ctrl_Debug_1	RA_DSPGP_SET1_IN_USE_DBG0 [31:16]																00000000h
		RA_DSPGP_SET1_IN_USE_DBG0 [15:0]																
R330784 (50C20h)	RA_DSPGP_SET2_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_DSPGP_SET2_IN_USE_STS	RA_DSPGP_SET2_SHARE	0	0	0	0	0	0	0	0	0	RA_DSPGP_SET2_OWNER [4:0]					
R330786 (50C22h)	RA_DSPGP_SET2_Thread_Ctrl_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	RA_DSPGP_SET2_IN_USE_SET [4:0]					
R330788 (50C24h)	RA_DSPGP_SET2_Thread_Ctrl_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	RA_DSPGP_SET2_IN_USE_CLR [4:0]					

Table 6-2. Register Map Definition—32-bit region *(Cont.)*

Register	Name	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0	Default
R330792 (50C28h)	RA_DSPGP_SET2_Thread_Ctrl_Debug_1	RA_DSPGP_SET2_IN_USE_DBG0 [31:16]																00000000h
		RA_DSPGP_SET2_IN_USE_DBG0 [15:0]																
R330800 (50C30h)	RA_DSPGP_SET3_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_DSPGP_SET3_IN_USE_STS	RA_DSPGP_SET3_SHARE	0	0	0	0	0	0	0	0	0	RA_DSPGP_SET3_OWNER [4:0]					
R330802 (50C32h)	RA_DSPGP_SET3_Thread_Ctrl_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	RA_DSPGP_SET3_IN_USE_SET [4:0]					
R330808 (50C38h)	RA_DSPGP_SET3_Thread_Ctrl_Debug_1	RA_DSPGP_SET3_IN_USE_DBG0 [31:16]																00000000h
		RA_DSPGP_SET3_IN_USE_DBG0 [15:0]																
R330816 (50C40h)	RA_DSPGP_SET4_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_DSPGP_SET4_IN_USE_STS	RA_DSPGP_SET4_SHARE	0	0	0	0	0	0	0	0	0	RA_DSPGP_SET4_OWNER [4:0]					
R330818 (50C42h)	RA_DSPGP_SET4_Thread_Ctrl_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	RA_DSPGP_SET4_IN_USE_SET [4:0]					
R330820 (50C44h)	RA_DSPGP_SET4_Thread_Ctrl_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	RA_DSPGP_SET4_IN_USE_CLR [4:0]					
R330824 (50C48h)	RA_DSPGP_SET4_Thread_Ctrl_Debug_1	RA_DSPGP_SET4_IN_USE_DBG0 [31:16]																00000000h
		RA_DSPGP_SET4_IN_USE_DBG0 [15:0]																
R333952 (51880h)	RA_MIF4_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000001h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RA_MIF4_STS	
R333956 (51884h)	RA_MIF4_Thread_Ctrl_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RA_MIF4_SHARE_STS	
R333960 (51888h)	RA_MIF4_Thread_Ctrl_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000001h
		0	0	0	0	0	0	0	0	0	0	RA_MIF4_NUM [5:0]						
R333968 (51890h)	RA_MIF41_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_MIF41_IN_USE_STS	RA_MIF41_SHARE	0	0	0	0	0	0	0	0	0	RA_MIF41_OWNER [4:0]					
R333970 (51892h)	RA_MIF41_Thread_Ctrl_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	RA_MIF41_IN_USE_SET [4:0]					
R333972 (51894h)	RA_MIF41_Thread_Ctrl_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	RA_MIF41_IN_USE_CLR [4:0]					
R333976 (51898h)	RA_MIF41_Thread_Ctrl_Debug_1	RA_MIF41_IN_USE_DBG0 [31:16]																00000000h
		RA_MIF41_IN_USE_DBG0 [15:0]																
R524288 (80000h)	DSP1_PMEM_0	0	0	0	0	0	0	0	0	DSP1_PM_START [39:32]								00000000h
		DSP1_PM_START [31:16]																
R524290 (80002h)	DSP1_PMEM_1	DSP1_PM_START [15:0]																00000000h
		0	0	0	0	0	0	0	0	DSP1_PM_1 [39:32]								
R524292 (80004h)	DSP1_PMEM_2	DSP1_PM_1 [31:16]																00000000h
		DSP1_PM_1 [15:0]																
R561146 (88FFAh)	DSP1_PMEM_18429	0	0	0	0	0	0	0	0	DSP1_PM_12286 [39:32]								00000000h
		DSP1_PM_12286 [31:16]																
R561148 (88FFCh)	DSP1_PMEM_18430	DSP1_PM_12286 [15:0]																00000000h
		0	0	0	0	0	0	0	0	DSP1_PM_END [39:32]								
R561150 (88FFEh)	DSP1_PMEM_18431	DSP1_PM_END [31:16]																00000000h
		DSP1_PM_END [15:0]																
R610304 (95000h)	DSP1_PMEM_ROM_0	0	0	0	0	0	0	0	0	DSP1_PM_ROM_START [39:32]								00000000h
		DSP1_PM_ROM_START [31:16]																
R610306 (95002h)	DSP1_PMEM_ROM_1	DSP1_PM_ROM_START [15:0]																00000000h
		0	0	0	0	0	0	0	0	DSP1_PM_ROM_1 [39:32]								
R610308 (95004h)	DSP1_PMEM_ROM_2	DSP1_PM_ROM_1 [31:16]																00000000h
		DSP1_PM_ROM_1 [15:0]																
R613370 (95BFAh)	DSP1_PMEM_ROM_1533	0	0	0	0	0	0	0	0	DSP1_PM_ROM_1022 [39:32]								00000000h
		DSP1_PM_ROM_1022 [31:16]																
R613372 (95BFCh)	DSP1_PMEM_ROM_1534	DSP1_PM_ROM_1022 [15:0]																00000000h
		0	0	0	0	0	0	0	0	DSP1_PM_ROM_END [39:32]								
R613374 (95BFEh)	DSP1_PMEM_ROM_1535	DSP1_PM_ROM_END [31:16]																00000000h
		DSP1_PM_ROM_END [15:0]																
R655360 (A0000h)	DSP1_XMEM_0	0	0	0	0	0	0	0	0	DSP1_XM_START [23:16]								00000000h
		DSP1_XM_START [15:0]																
R655362 (A0002h)	DSP1_XMEM_1	0	0	0	0	0	0	0	0	DSP1_XM_1 [23:16]								00000000h
		DSP1_XM_1 [15:0]																
R696316 (A9FFCh)	DSP1_XMEM_20478	0	0	0	0	0	0	0	0	DSP1_XM_20478 [23:16]								00000000h
		DSP1_XM_20478 [15:0]																
R696318 (A9FFEh)	DSP1_XMEM_20479	0	0	0	0	0	0	0	0	DSP1_XM_END [23:16]								00000000h
		DSP1_XM_END [15:0]																
R786432 (C0000h)	DSP1_YMEM_0	0	0	0	0	0	0	0	0	DSP1_YM_START [23:16]								00000000h
		DSP1_YM_START [15:0]																

Table 6-2. Register Map Definition—32-bit region *(Cont.)*

Register	Name	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0	Default
R786434 (C0002h)	DSP1_YMEM_1	0	0	0	0	0	0	0	0	DSP1_YM_1 [23:16]								00000000h
		DSP1_YM_1 [15:0]																
R794620 (C1FFCh)	DSP1_YMEM_4094	0	0	0	0	0	0	0	0	DSP1_YM_4094 [23:16]								00000000h
		DSP1_YM_4094 [15:0]																
R794622 (C1FFEh)	DSP1_YMEM_4095	0	0	0	0	0	0	0	0	DSP1_YM_END [23:16]								00000000h
		DSP1_YM_END [15:0]																
R917504 (E0000h)	DSP1_ZMEM_0	0	0	0	0	0	0	0	0	DSP1_ZM_START [23:16]								00000000h
		DSP1_ZM_START [15:0]																
R917506 (E0002h)	DSP1_ZMEM_1	0	0	0	0	0	0	0	0	DSP1_ZM_1 [23:16]								00000000h
		DSP1_ZM_1 [15:0]																
R925692 (E1FFCh)	DSP1_ZMEM_4094	0	0	0	0	0	0	0	0	DSP1_ZM_4094 [23:16]								00000000h
		DSP1_ZM_4094 [15:0]																
R925694 (E1FFEh)	DSP1_ZMEM_4095	0	0	0	0	0	0	0	0	DSP1_ZM_END [23:16]								00000000h
		DSP1_ZM_END [15:0]																
R1048064 (FFE00h)	DSP1_Config_1	0	0	0	0	0	0	0	DSP1_FLL_AO_CLKENA	0	0	0	0	0	0	0	0	00000000h
		0	DSP1_RATE [3:0]				0	0	0	0	0	0	DSP1_MEM_ENA	DSP1_DBG_CLK_ENA	0	DSP1_CORE_ENA	DSP1_START	
R1048066 (FFE02h)	DSP1_Config_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP1_CLK_FREQ_SEL [15:0]																
R1048068 (FFE04h)	DSP1_Status_1	DSP1_PING_FULL	DSP1_PONG_FULL	0	0	0	0	0	0	DSP1_WDMA_ACTIVE_CHANNELS [7:0]								00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R1048070 (FFE06h)	DSP1_Status_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_CLK_AVAIL	
R1048072 (FFE08h)	DSP1_Status_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP1_CLK_FREQ_STS [15:0]																
R1048074 (FFE0Ah)	DSP1_Watchdog_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	DSP1_WDT_MAX_COUNT [3:0]				DSP1_WDT_ENA	
R1048080 (FFE10h)	DSP1_WDMA_Buffer_1	DSP1_START_ADDRESS_WDMA_BUFFER_1 [15:0]																00000000h
		DSP1_START_ADDRESS_WDMA_BUFFER_0 [15:0]																
R1048082 (FFE12h)	DSP1_WDMA_Buffer_2	DSP1_START_ADDRESS_WDMA_BUFFER_3 [15:0]																00000000h
		DSP1_START_ADDRESS_WDMA_BUFFER_2 [15:0]																
R1048084 (FFE14h)	DSP1_WDMA_Buffer_3	DSP1_START_ADDRESS_WDMA_BUFFER_5 [15:0]																00000000h
		DSP1_START_ADDRESS_WDMA_BUFFER_4 [15:0]																
R1048086 (FFE16h)	DSP1_WDMA_Buffer_4	DSP1_START_ADDRESS_WDMA_BUFFER_7 [15:0]																00000000h
		DSP1_START_ADDRESS_WDMA_BUFFER_6 [15:0]																
R1048096 (FFE20h)	DSP1_RDMA_Buffer_1	DSP1_START_ADDRESS_RDMA_BUFFER_1 [15:0]																00000000h
		DSP1_START_ADDRESS_RDMA_BUFFER_0 [15:0]																
R1048098 (FFE22h)	DSP1_RDMA_Buffer_2	DSP1_START_ADDRESS_RDMA_BUFFER_3 [15:0]																00000000h
		DSP1_START_ADDRESS_RDMA_BUFFER_2 [15:0]																
R1048100 (FFE24h)	DSP1_RDMA_Buffer_3	DSP1_START_ADDRESS_RDMA_BUFFER_5 [15:0]																00000000h
		DSP1_START_ADDRESS_RDMA_BUFFER_4 [15:0]																
R1048112 (FFE30h)	DSP1_DMA_Config_1	0	0	0	0	0	0	0	0	DSP1_WDMA_CHANNEL_ENABLE [7:0]								00000000h
		0	0	DSP1_DMA_BUFFER_LENGTH [13:0]														
R1048114 (FFE32h)	DSP1_DMA_Config_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	DSP1_WDMA_CHANNEL_OFFSET [7:0]								
R1048116 (FFE34h)	DSP1_DMA_Config_3	0	0	0	0	0	0	0	0	0	0	DSP1_RDMA_CHANNEL_OFFSET [5:0]						00000000h
		0	0	0	0	0	0	0	0	0	0	DSP1_RDMA_CHANNEL_ENABLE [5:0]						
R1048118 (FFE36h)	DSP1_DMA_Config_4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_DMA_WORD_SEL	
R1048120 (FFE38h)	DSP1_External_Start	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	DSP1_START_IN_SEL [4:0]					
R1048128 (FFE40h)	DSP1_Scratch_1	DSP1_SCRATCH_1 [15:0]																00000000h
		DSP1_SCRATCH_0 [15:0]																
R1048130 (FFE42h)	DSP1_Scratch_2	DSP1_SCRATCH_3 [15:0]																00000000h
		DSP1_SCRATCH_2 [15:0]																
R1048146 (FFE52h)	DSP1_Bus_Error_Addr	0	0	0	0	0	0	0	0	DSP1_BUS_ERROR_ADDR [23:16]								00000000h
		DSP1_BUS_ERROR_ADDR [15:0]																
R1048148 (FFE54h)	DSP1_Ext_window_A	DSP1_EXT_A_PSIZE16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP1_EXT_A_PAGE [15:0]																
R1048150 (FFE56h)	DSP1_Ext_window_B	DSP1_EXT_B_PSIZE16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP1_EXT_B_PAGE [15:0]																

Table 6-2. Register Map Definition—32-bit region *(Cont.)*

Register	Name	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0	Default
R1048152 (FFE58h)	DSP1_Ext_window_C	DSP1_EXT_C_PSIZE16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP1_EXT_C_PAGE [15:0]																
R1048154 (FFE5Ah)	DSP1_Ext_window_D	DSP1_EXT_D_PSIZE16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP1_EXT_D_PAGE [15:0]																
R1048158 (FFE5Eh)	DSP1_Watchdog_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP1_WDT_RESET [15:0]																
R1048160 (FFE60h)	DSP1_Identity	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	DSP1_CORE_NUMBER [4:0]					
R1048164 (FFE64h)	DSP1_Region_lock_sts_0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	DSP1_CTRL_REGION3_LOCK_STS	DSP1_CTRL_REGION2_LOCK_STS	DSP1_CTRL_REGION1_LOCK_STS	DSP1_CTRL_REGION0_LOCK_STS	
R1048166 (FFE66h)	DSP1_Region_lock_1__DSP1_Region_lock_0	DSP1_CTRL_REGION1_LOCK [15:0]																00000000h
		DSP1_CTRL_REGION0_LOCK [15:0]																
R1048168 (FFE68h)	DSP1_Region_lock_3__DSP1_Region_lock_2	DSP1_CTRL_REGION3_LOCK [15:0]																00000000h
		DSP1_CTRL_REGION2_LOCK [15:0]																
R1048186 (FFE7Ah)	DSP1_Region_lock_ctrl_0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP1_LOCK_ERR_STS	DSP1_ADDR_ERR_STS	DSP1_WDT_TIMEOUT_STS	0	0	0	0	0	0	0	0	DSP1_SLAVE_DBG_ENA	0	0	DSP1_ERR_PAUSE	DSP1_ERR_CLEAR	
R1048188 (FFE7Ch)	DSP1_PMEM_ERR_ADDR__DSP1_XMEM_ERR_ADDR	0	DSP1_PMEM_ERR_ADDR [14:0]															00000000h
		DSP1_XMEM_ERR_ADDR [15:0]																

7 Thermal Characteristics

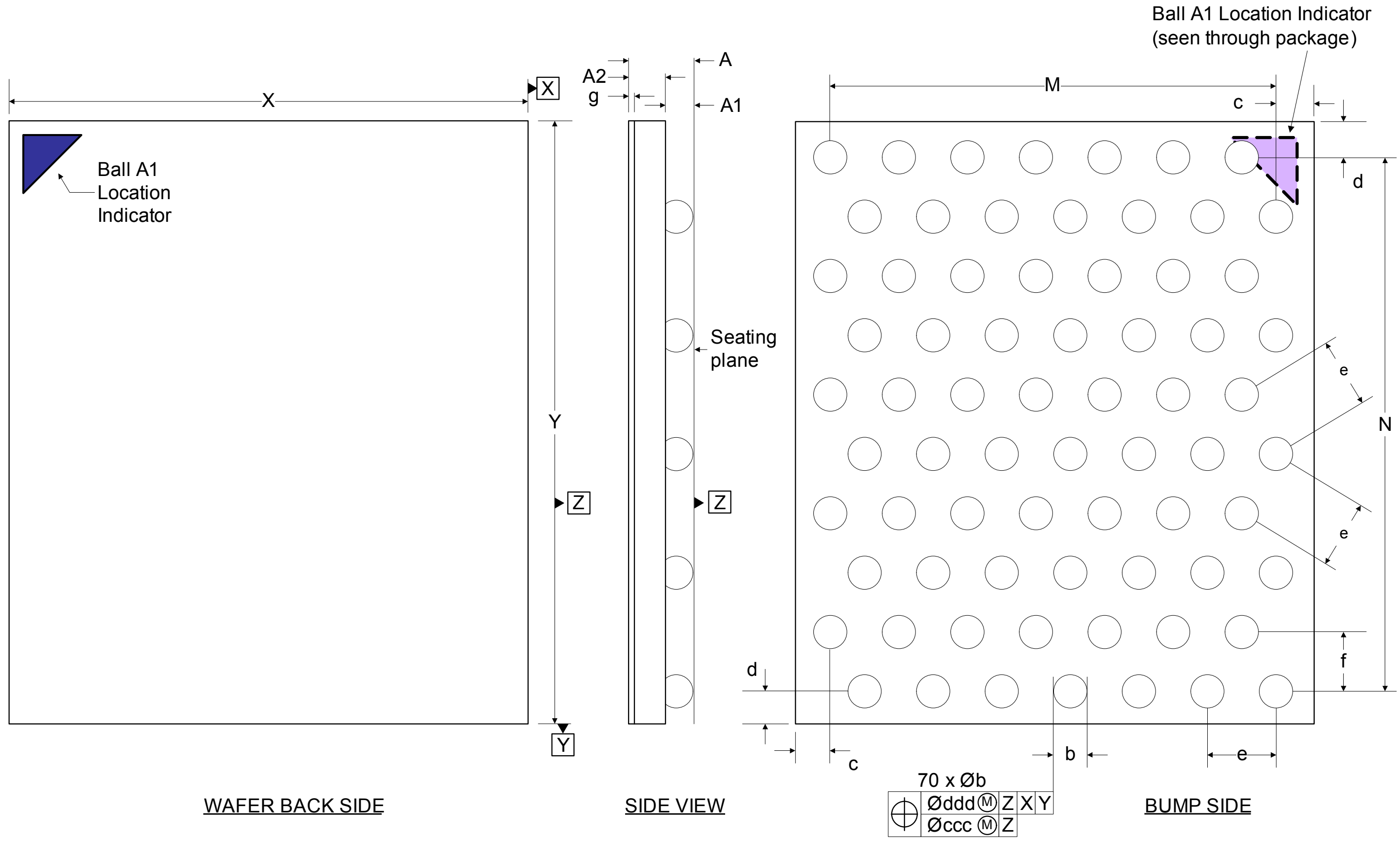
Table 7-1. Typical JEDEC Four-Layer, 2s2p Board Thermal Characteristics

Parameter	Symbol	WLCSP	Units
Junction-to-ambient thermal resistance	θ_{JA}	43.4	°C/W
Junction-to-board thermal resistance	θ_{JB}	16.2	°C/W
Junction-to-case thermal resistance	θ_{JC}	2.87	°C/W
Junction-to-board thermal-characterization parameter	Ψ_{JB}	16.1	°C/W
Junction-to-package-top thermal-characterization parameter	Ψ_{JT}	0.17	°C/W

Notes:

- Natural convection at the maximum recommended operating temperature T_A (see Table 3-3)
- Four-layer, 2s2p PCB as specified by JESD51-9 and JESD51-11; dimensions: 101.5 x 114.5 x 1.6 mm
- Thermal parameters as defined by JESD51-12

8 Package Dimensions



Notes:
- Dimensioning and tolerances per ASME Y 14.5M–2009.
- The Ball A1 position indicator is for illustration purposes only and may not be to scale.
- Dimension "b" applies to the solder sphere diameter and is measured at the midpoint between the package body and the seating plane Datum Z.

Table 8-1. WLCSP Package Dimensions

Dimension	Millimeters		
	Minimum	Nominal	Maximum
A	0.474	0.504	0.534
A1	0.172	0.202	0.232
A2	0.287	0.302	0.317
M	BSC	2.6	BSC
N	BSC	3.1176	BSC
b	0.247	0.262	0.277
c	0.2017	0.2057	0.2097
d	0.1940	0.1980	0.2020
e	BSC	0.40	BSC
f	BSC	0.3464	BSC
g	REF	0.022	REF
X	2.9864	3.0114	3.0364
Y	3.4886	3.5136	3.5386
ccc = 0.05 ddd = 0.15			

Note: Controlling dimension is millimeters.

9 Ordering Information

Table 9-1. Ordering Information

Product	Description	Package	Halogen Free	Pb Free	Grade	Temperature Range	Container	Order #
CS47L15	Smart Codec with Low-Power Audio DSP	70-ball WLCSP	Yes	Yes	Commercial	–40 to +85°C	Tape and Reel [1]	CS47L15–CWZR

1. Reel quantity = 6,000

10 References

- Google Inc, *Android Wired Headset Specification, Version 1.1.* https://source.android.com/accessories/headset-spec.html
- International Electrotechnical Commission, *IEC60958-3 Digital Audio Interface—Consumer.* http://www.ansi.org/

11 Revision History

Table 11-1. Revision History

Revision	Changes
PP1 NOV '16	• Power sequencing requirements updated (Table 3-3). • IN2 noise specification updated (Table 3-9). • MICD accessory detection description updated (Table 4-66). • MICBIAS control for accessory detection updated (Table 4-66, Section 4.9.3.2). • Register bit position of GPn_POL corrected (Table 4-72). • FLL control requirements updated (Section 4.13.8). • Initialization sequence (patch) requirements added (Section 4.19.5).

Contacting Cirrus Logic Support

For all product questions and inquiries, contact a Cirrus Logic Sales Representative.
To find the one nearest you, go to www.cirrus.com.