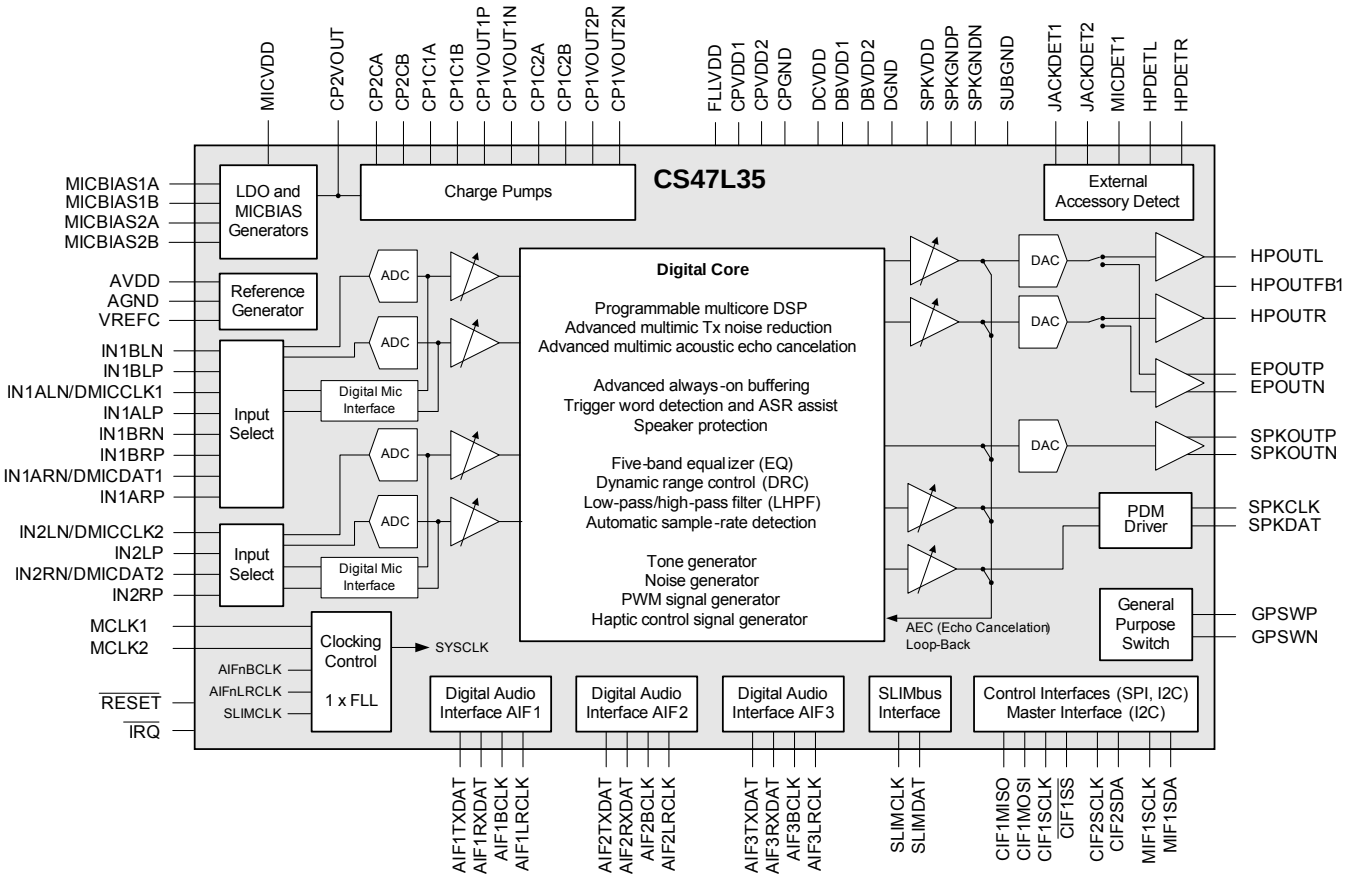
Smart Codec with Low-Power Audio DSP

Features

- 450 MIPS, 450 MMAC multicore audio-signal processor
- Programmable wideband, multimic audio processing
  - Transmit-path noise reduction and echo cancelation
  - Wind-noise, side-tone, and other programmable filters
- Integrated multichannel 24-bit hi-fi audio hub codec
  - 101-dB signal-to-noise ratio (SNR) mic input (48 kHz)
  - 122-dB SNR headphone playback (48 kHz)
- Up to six analog or four digital microphone (DMIC) inputs
- 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
- SLIMbus® audio and control interface
- Three full digital-audio interfaces
  - Standard sample rates from 8 to 192 kHz
  - Multichannel time-division multiplexing (TDM) support on AIF1
- Flexible clocking, derived from MCLK*n*, AIF*n*, or SLIMbus
- Low-power frequency-locked loop (FLL) supports reference clocks down to 32 kHz
- Configurable functions on up to 16 general-purpose input/output (GPIO) pins
- Sensor-hub connectivity, with event time-stamp functions
- Integrated regulators and charge pumps
- Small W-CSP package, 0.4-mm ball array

Applications

- Smartphones and multimedia handsets
- Tablets and mobile Internet devices



Description

The CS47L35 is a highly integrated, low-power audio hub for smartphones, tablets, and other portable audio devices. It combines an advanced DSP feature set with a flexible, high-performance audio hub codec. The CS47L35 combines three programmable DSP cores with a variety of power-efficient fixed-function audio processors. Flexible GPIO and an I²C master interface are also incorporated, enabling sensor-hub connectivity.

The DSP cores support multiple concurrent audio features, including multimic wideband noise reduction, high-performance acoustic-echo cancelation (AEC), speech enhancement, advanced media enhancement, and many more. The DSP cores are supported by 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.

A SLIMbus interface supports multichannel audio paths and host control register access. Three further 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 stereo headphone driver provides ground-referenced output, with noise levels as low as 0.8 μV_{RMS} for hi-fi quality line or headphone output. The CS47L35 also features a mono bridge-tied load (BTL) earpiece output, mono 2.7-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 CS47L35 supports up to six analog inputs, and up to four PDM digital inputs. 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 CS47L35 is configured using the SLIMbus, SPI™, or I²C interfaces. The integrated FLL provides support for a wide range of system-clock frequencies. The device is powered from 1.8- and 1.2-V supplies. (A separate 4.2-V battery supply is typically required for the Class D speaker drivers). The power, clocking, and output driver architectures are designed to maximize battery life in voice, music, and standby modes. Low-power (10 μA) Sleep Mode is supported, with configurable wake-up events.

Table of Contents

1 Pin Descriptions

1.1 WLCSP Pinout

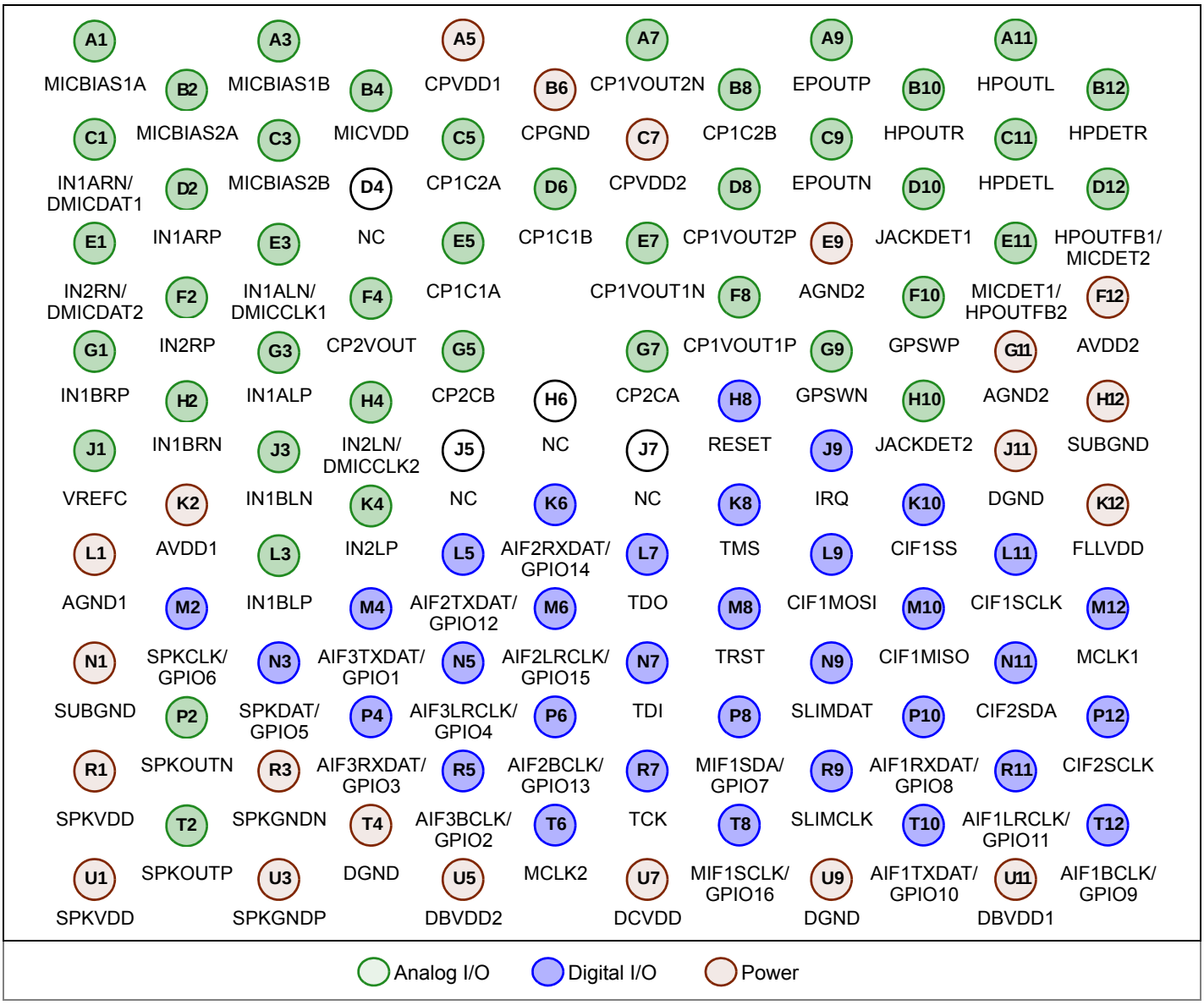


Figure 1-1. Top-Down (Through-Package) View—101-Ball WLCSP Package

1.2 Pin Descriptions

Table 1-1 describes each pin on the CS47L35. Note that pins that share a common name should be tied together on the printed circuit board (PCB). All digital output pins are CMOS outputs, unless otherwise stated.

Table 1-1. Pin Descriptions

Name	Ball #	Power Supply	I/O	Description
			Analog I/O	
CP1C1A	E5	—	O	Charge Pump 1 fly-back capacitor 1 pin
CP1C1B	D6	—	O	Charge Pump 1 fly-back capacitor 1 pin
CP1C2A	C5	—	O	Charge Pump 1 fly-back capacitor 2 pin
CP1C2B	B8	—	O	Charge Pump 1 fly-back capacitor 2 pin
CP1VOUT1N	E7	—	O	Charge Pump 1 negative output 1 decoupling pin
CP1VOUT1P	F8	—	O	Charge Pump 1 positive output 1 decoupling pin
CP1VOUT2N	A7	—	O	Charge Pump 1 negative output 2 decoupling pin
CP1VOUT2P	D8	—	O	Charge Pump 1 positive output 2 decoupling pin
CP2CA	G7	—	O	Charge Pump 2 fly-back capacitor pin
CP2CB	G5	—	O	Charge Pump 2 fly-back capacitor pin
CP2VOUT	F4	—	O	Charge Pump 2 output decoupling pin/supply for LDO2
EPOUTN	C9	—	O	Earpiece negative output
EPOUTP	A9	—	O	Earpiece positive output
GPSWN	G9	—	I/O	General-purpose bidirectional switch contact
GPSWP	F10	—	I/O	General-purpose bidirectional switch contact
HPDETL	C11	—	I	Headphone left (HPOUTL) sense input
HPDETR	B12	—	I	Headphone right (HPOUTR) sense input
HPOUTFB1/MICDET2	D12	—	I	HPOUTL and HPOUTR ground feedback pin 1/mic and accessory sense input 2
HPOUTL	A11	—	O	Left headphone output
HPOUTR	B10	—	O	Right headphone output
IN1ALN/DMICCLK1	E3	MICVDD or MICBIAS*nx*	I/O	Left-channel negative differential mic/line input /DMIC Clock Output 1
IN1ALP	G3	MICVDD	I	Left-channel single-ended mic/line input/left-channel positive differential mic/line input
IN1ARN/DMICDAT1	C1	MICVDD or MICBIAS*nx*	I	Right-channel negative differential mic/line input/DMIC Data Input 1
IN1ARP	D2	MICVDD	I	Right-channel single-ended mic/line input/right-channel positive differential mic/line input
IN1BLN	J3	MICVDD	I	Left-channel negative differential mic/line input. Also suitable for connection to external accessory interfaces.
IN1BLP	L3	MICVDD	I	Left-channel single-ended mic/line input/left-channel positive differential mic/line input. Also suitable for connection to external accessory interfaces.
IN1BRN	H2	MICVDD	I	Right-channel negative differential mic/line input. Also suitable for connection to external accessory interfaces.
IN1BRP	G1	MICVDD	I	Right-channel single-ended mic/line input/right-channel positive differential mic/line input. Also suitable for connection to external accessory interfaces.
IN2LN/DMICCLK2	H4	MICVDD or MICBIAS*nx*	I/O	Left-channel negative differential mic/line input/DMIC Clock Output 2

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

Name	Ball #	Power Supply	I/O	Description
IN2LP	K4	MICVDD	I	Left-channel single-ended mic/line input/left-channel positive differential mic/line input
IN2RN/DMICDAT2	E1	MICVDD or MICBIAS*nx*	I	Right-channel negative differential mic/line input/DMIC Data Input 2
IN2RP	F2	MICVDD	I	Right-channel single-ended mic/line input/right-channel positive differential mic/line input
JACKDET1	D10	AVDD	I	Jack detect input 1
JACKDET2	H10	AVDD	I	Jack detect input 2
MICBIAS1A	A1	—	O	Microphone bias 1A
MICBIAS1B	A3	—	O	Microphone bias 1B
MICBIAS2A	B2	—	O	Microphone bias 2A
MICBIAS2B	C3	—	O	Microphone bias 2B
MICDET1/HPOUTFB2	E11	—	I	Microphone and accessory sense input 1/HPOUTL and HPOUTR ground feedback pin 2
MICVDD	B4	—	O	LDO2 output decoupling pin (generated internally by CS47L35). (Can also be used as reference/supply for external microphones.)
SPKOUTN	P2	—	O	Speaker negative output
SPKOUTP	T2	—	O	Speaker positive output
VREFC	J1	—	O	Band-gap reference external capacitor connection
				Digital I/O
AIF1BCLK/GPIO9	T12	DBVDD1	I/O	Audio interface 1 bit clock/GPIO. GPIO output is selectable CMOS or open drain; BCLK output is CMOS.
AIF1LRCLK/GPIO11	R11	DBVDD1	I/O	Audio interface 1 left/right clock/GPIO. GPIO output is selectable CMOS or open drain; LRCLK output is CMOS.
AIF1RXDAT/GPIO8	P10	DBVDD1	I/O	Audio interface 1 RX digital audio data/GPIO. GPIO output is selectable CMOS or open drain.
AIF1TXDAT/GPIO10	T10	DBVDD1	I/O	Audio interface 1 TX digital audio data/GPIO. GPIO output is selectable CMOS or open drain; TXDAT output is CMOS.
AIF2BCLK/GPIO13	P6	DBVDD2	I/O	Audio interface 2 bit clock/GPIO. GPIO output is selectable CMOS or open drain; BCLK output is CMOS.
AIF2LRCLK/GPIO15	M6	DBVDD2	I/O	Audio interface 2 left/right clock/GPIO. GPIO output is selectable CMOS or open drain; LRCLK output is CMOS.
AIF2RXDAT/GPIO14	K6	DBVDD2	I/O	Audio interface 2 RX digital audio data/GPIO. GPIO output is selectable CMOS or open drain.
AIF2TXDAT/GPIO12	L5	DBVDD2	I/O	Audio interface 2 TX digital audio data/GPIO. GPIO output is selectable CMOS or open drain; TXDAT output is CMOS.
AIF3BCLK/GPIO2	R5	DBVDD2	I/O	Audio interface 3 bit clock/GPIO. GPIO output is selectable CMOS or open drain; BCLK output is CMOS.
AIF3LRCLK/GPIO4	N5	DBVDD2	I/O	Audio interface 3 left/right clock/GPIO. GPIO output is selectable CMOS or open drain; LRCLK output is CMOS.
AIF3RXDAT/GPIO3	P4	DBVDD2	I/O	Audio interface 3 RX digital audio data/GPIO. GPIO output is selectable CMOS or open drain.
AIF3TXDAT/GPIO1	M4	DBVDD2	I/O	Audio interface 3 TX digital audio data/GPIO. GPIO output is selectable CMOS or open drain; TXDAT output is CMOS.
CIF1MISO	M10	DBVDD1	O	Control interface 1 (SPI) Master In Slave Out data. The CIFMISO is high impedance if $\overline{\text{CIF1SS}}$ is not asserted.
CIF1MOSI	L9	DBVDD1	I	Control interface 1 (SPI) Master Out Slave In data
CIF1SCLK	L11	DBVDD1	I	Control interface 1 (SPI) clock input
$\overline{\text{CIF1SS}}$	K10	DBVDD1	I	Control interface 1 (SPI) slave select (SS)
CIF2SCLK	P12	DBVDD1	I	Control interface 2 (I^2C) clock input

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

Name	Ball #	Power Supply	I/O	Description
CIF2SDA	N11	DBVDD1	I/O	Control interface 2 (I^2C) data input and output. The SDA output is open drain.
$\overline{IRQ}$	J9	DBVDD1	O	Interrupt request output (default is active low). The pin configuration is selectable CMOS or open drain.
MCLK1	M12	DBVDD1	I	Master clock 1
MCLK2	T6	DBVDD2	I	Master clock 2
MIF1SCLK/GPIO16	T8	DBVDD1	I/O	Master (I^2C) Interface 1 clock output/GPIO. GPIO output is selectable CMOS or open drain; SCLK output is open drain.
MIF1SDA/GPIO7	P8	DBVDD1	I/O	Master (I^2C) Interface 1 data input and output/GPIO. GPIO output is selectable CMOS or open drain; SDA output is open drain.
$\overline{RESET}$	H8	DBVDD1	I	Digital reset input (active low)
SLIMCLK	R9	DBVDD1	I/O	SLIMbus clock I/O
SLIMDAT	N9	DBVDD1	I/O	SLIMbus data I/O
SPKCLK/GPIO6	M2	DBVDD2	I/O	Digital speaker (PDM) 1 clock output/GPIO. GPIO output is selectable CMOS or open drain; SPKCLK output is CMOS.
SPKDAT/GPIO5	N3	DBVDD2	I/O	Digital speaker (PDM) 1 data output/GPIO. GPIO output is selectable CMOS or open drain; SPKDAT output is CMOS.
TCK	R7	DBVDD2	I	JTAG clock input. Internal pull-down holds this pin at Logic 0 for normal operation.
TDI	N7	DBVDD2	I	JTAG data input. Internal pull-down holds this pin at Logic 0 for normal operation.
TDO	L7	DBVDD2	O	JTAG data output
TMS	K8	DBVDD2	I	JTAG mode select input. Internal pull-down holds this pin at Logic 0 for normal operation.
TRST	M8	DBVDD2	I	JTAG test access port reset (active low). Internal pull-down holds this pin at Logic 0 for normal operation.
				Supply
AGND1	L1	—	—	Analog ground (return path for AVDD1)
AGND2	E9, G11	—	—	Analog ground (return path for AVDD2)
AVDD1	K2	—	—	Analog supply
AVDD2	F12	—	—	Analog supply
CPGND	B6	—	—	Charge pump ground (return path for CPVDD1, CPVDD2)
CPVDD1	A5	—	—	Supply for Charge Pump 1 and Charge Pump 2
CPVDD2	C7	—	—	Secondary supply for Charge Pump 1
DBVDD1	U11	—	—	Digital buffer (I/O) supply (core functions, AIF1, CIF1, CIF2, SLIMbus, MIF1)
DBVDD2	U5	—	—	Digital buffer (I/O) supply (AIF2, AIF3, PDM, MCLK2, JTAG)

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

Name	Ball #	Power Supply	I/O	Description
DCVDD	U7	—	—	Digital core supply
DGND	J11, T4, U9	—	—	Digital ground (return path for DCVDD and DBVDD*n*)
FLLVDD	K12	—	—	Analog supply (FLL1)
SPKGNDN	R3	—	—	Speaker driver ground (return path for SPKVDD) [1]
SPKGNDP	U3	—	—	Speaker driver ground (return path for SPKVDD) [1]
SPKVDD	R1, U1	—	—	Speaker driver supply
SUBGND	H12, N1	—	—	Substrate ground
No Connect				
NC	D4, H6, J5, J7	—	—	—

1. 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

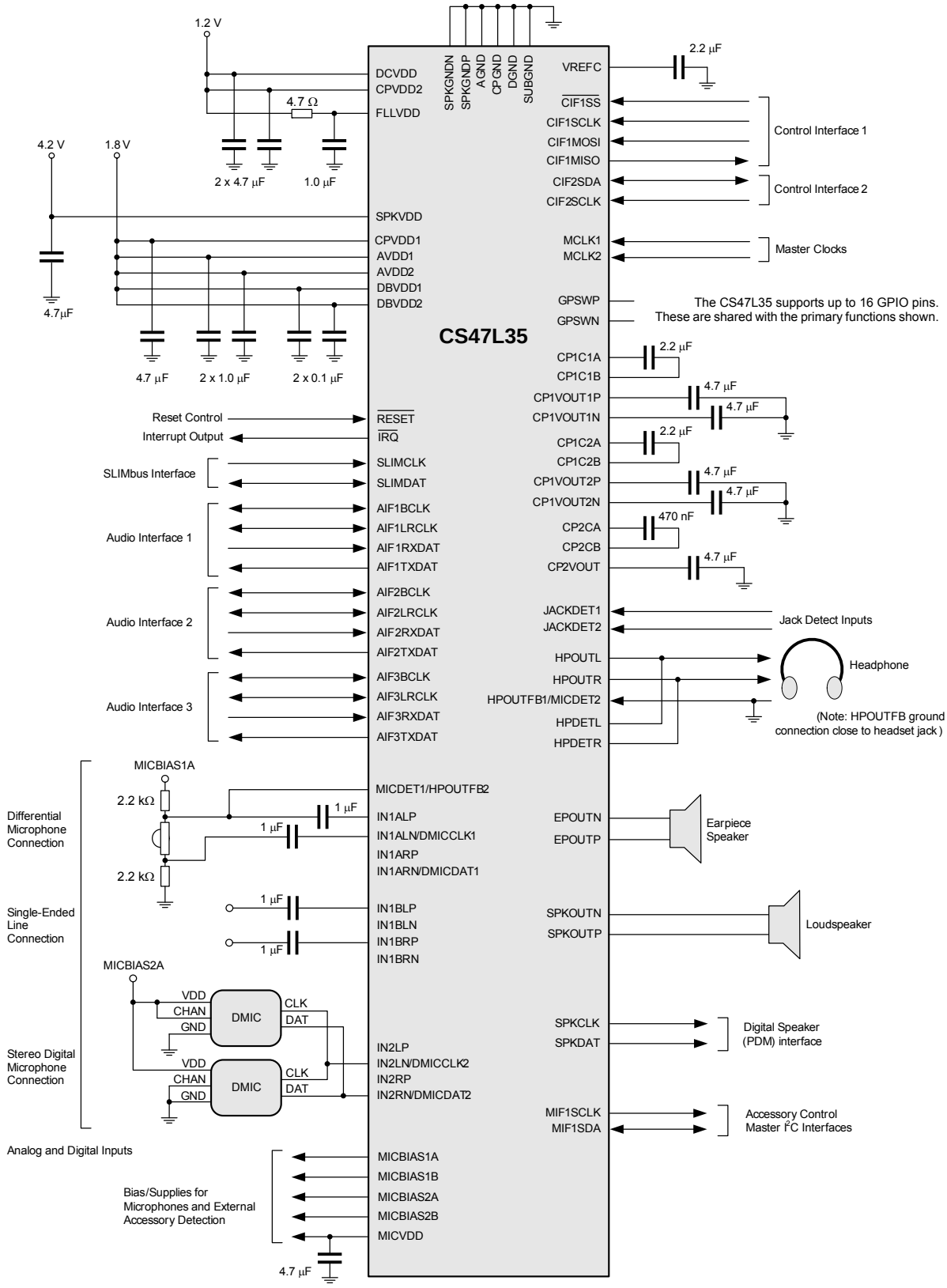


Figure 2-1. Typical Connection Diagram

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 [1], FLLVDD [1] CPVDD1, CPVDD2 DBVDD1, DBVDD2, AVDD [2], 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	DBVDD1 domain DBVDD2 domain	— —	SUBGND – 0.3 V SUBGND – 0.3 V	DBVDD1 + 0.3 V DBVDD2 + 0.3 V
Voltage range analog inputs		IN1Axx, IN2xx IN1Bxx HPOUTFB*n* [3] MICDET*n* [3] JACKDET1, HPDETL, HPDETR JACKDET2 [4], GPSWP, GPSWN	SUBGND – 0.3 V SUBGND – 0.9 V SUBGND – 0.3 V SUBGND – 0.3 V CP1VOUT2N – 0.3 V [5] 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 [6], DGND, CPGND, SPKGND	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 CS47L35 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 DCVDD and FLLVDD pins should be tied together. The associated power domain is referred to as DCVDD.
2. The AVDD1 and AVDD2 pins should be tied together. The associated power domain is referred to as AVDD.
3. The HPOUTFB*n* and MICDET*n* functions share common pins. The absolute maximum rating varies according to the applicable function of each pin.
4. If AVDD > MICVDD (e.g., if LDO2 is disabled), the maximum JACKDET2 voltage is AVDD + 0.3 V.
5. CP1VOUT2N is an internal supply, generated by the CS47L35 charge pump (CP1). Its voltage can vary between CPGND and –CPVDD1.
6. The AGND1 and AGND2 pins should be tied together. The associated ground domain is referred to as AGND.

Table 3-3. Recommended Operating Conditions

Parameter		Symbol	Minimum	Typical	Maximum	Units
Digital supply range [1]	Core and FLL	DCVDD [2], FLLVDD [3]	1.14	1.2	1.26	V
Digital supply range	I/O	DBVDD1, DBVDD2	1.71	—	3.6 [4]	V
Charge pump supply range	CPVDD1	CPVDD1	1.71	1.8	1.89	V
	CPVDD2	CPVDD2	1.14	1.2	1.26	V
Speaker supply range		SPKVDD	2.4		5.5	V
Analog supply range [5,6]		AVDD	1.71	1.8	1.89	V
Mic bias supply [7]		MICVDD	0.9	2.5	3.78	V
Ground [8]		DGND, AGND, CPGND, SPKGND, SUBGND	—	0	—	V
Power supply rise time [9,10]		DCVDD	10	—	2000	µs
		All other supplies	10	—	—	µs
Operating temperature range		T_A	–40	—	85	ºC

Note: There are no power sequencing requirements; the supplies may be enabled and disabled in any order.

1. The DCVDD and FLLVDD pins should be tied together. The associated power domain is referred to as DCVDD.
2. Sleep mode is supported for when DCVDD is below the limits noted, provided that AVDD and DBVDD1 are present
3. It is recommended to connect a 4.7-Ω resistor in series with the FLLVDD pin connection. Note that the minimum voltage limit applies at the supply end of the 4.7-Ω resistor in this case.
4. If the SLIMbus interface is enabled, the maximum DBVDD1 voltage is 1.98 V.
5. The AVDD1 and AVDD2 pins should be tied together. The associated power domain is referred to as AVDD
6. The AGND1 and AGND2 pins should be tied together. The associated ground domain is referred to as AGND
7. An internal charge pump and LDO (powered by CPVDD1) provide the mic bias supply; the MICVDD pin must not be connected to an external supply.
8. The impedance between DGND, AGND, and SUBGND must not exceed 0.1 Ω.
   The impedance between SPKGND and SUBGND must not exceed 0.2 Ω.
9. 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.
10. 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—IN1AL, IN1BL, IN1AR, IN1BR, IN2L, IN2R

Test conditions (unless specified otherwise): AVDD = 1.8V; 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
Full-scale input signal level (0 dBFS output)	Single-ended PGA input, 0 dB PGA gain	—	0.5	—	V_{RMS}
		—	–6	—	dBV
	Differential PGA input, 0 dB PGA gain	—	1	—	V_{RMS}
		—	0	—	dBV

Notes:
- The full-scale input signal level is also the maximum analog input level, before clipping occurs.
- The full-scale input signal level changes in proportion with AVDD. For differential input, it is calculated as AVDD/1.8.
- A 1.0V_{RMS} differential signal equates to 0.5V_{RMS}/–6dBV per input.
- A sinusoidal input signal is assumed.

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	Single-ended PGA input, All PGA gain settings	9	10.5	—	kΩ
	Differential PGA input, All PGA gain settings	18	21	—	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		Minimum	Typical	Maximum	Units
Minimum programmable gain		—	0	—	dB
Maximum programmable gain		—	31	—	dB
Programmable gain step size	Guaranteed monotonic	—	1	—	dB

Table 3-7. Digital Input Signal Level—DMICDAT1, DMICDAT2

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

Parameter		Minimum	Typical	Max	Units
Full-scale input signal level (0 dBFS output)	0 dB gain	—	–6	—	dBFS

Note: The DMIC 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	6	—	—	Ω
		Normal operation, Differential (BTL) Mode	15	—	—	Ω
		Device survival with load applied indefinitely	0	—	—	Ω
	Load capacitance	Single-Ended Mode	—	—	500	pF
		Differential (BTL) Mode	—	—	200	pF
Earpiece output driver (EPOUT)	Load resistance	Normal operation	15	—	—	Ω
		Device survival with load applied indefinitely	0	—	—	Ω
	Load capacitance		—	—	200	pF
Speaker output driver (SPKOUTP+SPKOUTN)	Load resistance	Normal operation	4	—	—	Ω
		Device survival with load applied indefinitely	0	—	—	Ω
	Load capacitance		—	—	200	pF
Digital speaker output (SPKDAT)	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): DBVDD1 = DBVDD2 = CPVDD1 = AVDD = 1.8 V, DCVDD = FLLVDD = CPVDD2 = 1.2 V; MICVDD = 2.5 V (powered from internal LDO); 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	—	50	—	µV
		Differential (BTL) mode	—	75	—	µV
Earpiece output driver (EPOUT+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 (IN*n*L, IN*n*R) to ADC (Differential Input Mode)	SNR (A-weighted), defined in Table 3-1	48 kHz sample rate	91	101	—	dB
		16 kHz sample rate (wideband voice)	—	105	—	dB
	THD, defined in Table 3-1	–1 dBV input	—	–87	—	dB
	THD+N, defined in Table 3-1	–1 dBV input	—	–86	–76	dB
	Channel separation (L/R), defined in Table 3-1	100 Hz to 10 kHz	—	100	—	dB
	Input-referred noise floor	A-weighted, PGA gain = +20 dB	—	3.2	—	μV_{RMS}
	CMRR, defined in Table 3-1	PGA gain = +30 dB	—	80	—	dB
		PGA gain = 0 dB	—	70	—	dB
	PSRR (DBVDD*n*, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	90	—	dB
		100 mV (peak-peak) 10 kHz	—	75	—	dB
	PSRR (DCVDD, FLLVDD, CPVDD2), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	95	—	dB
		100 mV (peak-peak) 10 kHz	—	90	—	dB
	PSRR (SPKVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	100	—	dB
		100 mV (peak-peak) 10 kHz	—	95	—	dB

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

Test conditions (unless specified otherwise): DBVDD1 = DBVDD2 = CPVDD1 = AVDD = 1.8 V, DCVDD = FLLVDD = CPVDD2 = 1.2 V; MICVDD = 2.5 V (powered from internal LDO); 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 (IN*n*LP, IN*n*RP) to ADC (Single-Ended Input Mode)	SNR (A-weighted), defined in Table 3-1	48-kHz sample rate	89	99	—	dB
		16-kHz sample rate (wideband voice)	—	102	—	dB
	THD, defined in Table 3-1	–7dB V input	—	–86	—	dB
	THD+N, defined in Table 3-1	–7dB V input	—	–85	–75	dB
	Channel separation (L/R), defined in Table 3-1	100 Hz to 10 kHz	—	100	—	dB
	Input-referred noise floor	A-weighted, PGA gain = +20 dB	—	4	—	μV_{RMS}
	PSRR (DBVDD*n*, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	77	—	dB
		100 mV (peak-peak) 10 kHz	—	50	—	dB
	PSRR (DCVDD, FLLVDD, CPVDD2), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	95	—	dB
		100 mV (peak-peak) 10 kHz	—	65	—	dB
	PSRR (SPKVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	100	—	dB
		100 mV (peak-peak) 10 kHz	—	80	—	dB
DAC to line output (HPOUTL, HPOUTR; Load = 10 kΩ, 50 pF)	Full-scale output signal level	0 dBFS input	—	1	—	V_{RMS}
			—	0	—	dBV
	SNR, defined in Table 3-1	A-weighted, output signal = 1 V_{RMS}	—	122	—	dB
	Dynamic range, defined in Table 3-1	A-weighted, –60 dBFS input	105	115	—	dB
	THD, defined in Table 3-1	0 dBFS input	—	–95	—	dB
	THD+N, defined in Table 3-1	0 dBFS input	—	–93	–83	dB
	Channel separation (L/R), defined in Table 3-1	100 Hz to 10 kHz	—	100	—	dB
	Output noise floor	A-weighted	—	0.8	—	μV_{RMS}
	PSRR (DBVDD*n*, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	100	—	dB
		100 mV (peak-peak) 10 kHz	—	73	—	dB
	PSRR (DCVDD, FLLVDD, CPVDD2), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	105	—	dB
		100 mV (peak-peak) 10 kHz	—	80	—	dB
	PSRR (SPKVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	110	—	dB
		100 mV (peak-peak) 10 kHz	—	100	—	dB
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}	—	122	—	dB
	Dynamic range, defined in Table 3-1	A-weighted, –60 dBFS input	105	115	—	dB
	THD, defined in Table 3-1	P_O = 20 mW	—	–93	—	dB
	THD+N, defined in Table 3-1	P_O = 20 mW	—	–91	—	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	–80	dB
	Channel separation (L/R), defined in Table 3-1	100 Hz to 10 kHz	—	100	—	dB
	Output noise floor	A-weighted	—	0.8	—	μV_{RMS}
	PSRR (DBVDD*n*, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	100	—	dB
		100 mV (peak-peak) 10 kHz	—	73	—	dB
	PSRR (DCVDD, FLLVDD, CPVDD2), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	105	—	dB
		100 mV (peak-peak) 10 kHz	—	80	—	dB
	PSRR (SPKVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	110	—	dB
		100 mV (peak-peak) 10 kHz	—	100	—	dB
DAC to headphone output (HPOUTL, HPOUTR; R_L = 16 Ω)	Maximum output power	0.1% THD+N	—	39	—	mW
	SNR, defined in Table 3-1	A-weighted, output signal = 1 V_{RMS}	—	122	—	dB
	Dynamic range, defined in Table 3-1	A-weighted, –60 dBFS input	105	115	—	dB
	THD, defined in Table 3-1	P_O = 20 mW	—	–89	—	dB
	THD+N, defined in Table 3-1	P_O = 20 mW	—	–88	—	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	–80	dB
	Channel separation (L/R), defined in Table 3-1	100 Hz to 10 kHz	—	100	—	dB
	Output noise floor	A-weighted	—	0.8	—	μV_{RMS}
	PSRR (DBVDD*n*, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	100	—	dB
		100 mV (peak-peak) 10 kHz	—	73	—	dB
	PSRR (DCVDD, FLLVDD, CPVDD2), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	105	—	dB
		100 mV (peak-peak) 10 kHz	—	80	—	dB
	PSRR (SPKVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	110	—	dB
		100 mV (peak-peak) 10 kHz	—	100	—	dB

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

Test conditions (unless specified otherwise): DBVDD1 = DBVDD2 = CPVDD1 = AVDD = 1.8 V, DCVDD = FLLVDD = CPVDD2 = 1.2 V; MICVDD = 2.5 V (powered from internal LDO); 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 = 32 Ω BTL)	Maximum output power	0.1% THD+N	—	99	—	mW
	SNR, defined in Table 3-1	A-weighted, output signal = 1.41 V_{RMS}	—	127	—	dB
	Dynamic range, defined in Table 3-1	A-weighted, –60 dBFS input	108	118	—	dB
	THD, defined in Table 3-1	P_O = 75 mW	—	–94	—	dB
	THD+N, defined in Table 3-1	P_O = 75 mW	—	–92	—	dB
	THD, defined in Table 3-1	P_O = 5 mW	—	–94	—	dB
	THD+N, defined in Table 3-1	P_O = 5 mW	—	–92	–82	dB
	Output noise floor	A-weighted	—	0.6	—	μV_{RMS}
	PSRR (DBVDD*n*, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	125 90	— —	dB dB
	PSRR (DCVDD, FLLVDD, CPVDD2), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	125 90	— —	dB dB
	PSRR (SPKVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	125 110	— —	dB dB
DAC to earpiece output (EPOUTP+EPOUTN, R_L = 16 Ω BTL)	Maximum output power	0.1% THD+N	—	110	—	mW
	SNR, defined in Table 3-1	A-weighted, output signal = 1.41 V_{RMS}	—	127	—	dB
	Dynamic range, defined in Table 3-1	A-weighted, –60 dBFS input	108	118	—	dB
	THD, defined in Table 3-1	P_O = 75 mW	—	–87	—	dB
	THD+N, defined in Table 3-1	P_O = 75 mW	—	–85	—	dB
	THD, defined in Table 3-1	P_O = 5 mW	—	–92	—	dB
	THD+N, defined in Table 3-1	P_O = 5 mW	—	–90	–80	dB
	Output noise floor	A-weighted	—	0.6	—	μV_{RMS}
	PSRR (DBVDD*n*, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	125 90	— —	dB dB
	PSRR (DCVDD, FLLVDD, CPVDD2), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	125 90	— —	dB dB
	PSRR (SPKVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	125 110	— —	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	92	102	—	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	—	–70	—	dB
	THD+N, defined in Table 3-1	P_O = 0.5 W	—	–70	–60	dB
	Output noise floor	A-weighted	—	1.3	—	μV_{RMS}
	PSRR (DBVDD*n*, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	125 90	— —	dB dB
	PSRR (DCVDD, FLLVDD, CPVDD2), 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	— —	120 90	— —	dB dB

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

Test conditions (unless specified otherwise): DBVDD1 = DBVDD2 = CPVDD1 = AVDD = 1.8 V, DCVDD = FLLVDD = CPVDD2 = 1.2 V; MICVDD = 2.5 V (powered from internal LDO); 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 speaker output (SPKOUTP+SPKOUTN, Load = 4 Ω, 15 µH, BTL)	Maximum output power	SPKVDD = 5.0 V, 1% THD+N	—	2.7	—	W
		SPKVDD = 4.2 V, 1% THD+N	—	1.9	—	W
		SPKVDD = 3.6 V, 1% THD+N	—	1.4	—	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	—	102	—	dB
	THD, defined in Table 3-1	P_O = 1.0 W	—	–71	—	dB
	THD+N, defined in Table 3-1	P_O = 1.0 W	—	–70	—	dB
	THD, defined in Table 3-1	P_O = 0.5 W	—	–71	—	dB
	THD+N, defined in Table 3-1	P_O = 0.5 W	—	–70	—	dB
	Output noise floor	A-weighted	—	1.3	—	μV_{RMS}
	PSRR (DBVDD*n*, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	125	—	dB
		100 mV (peak-peak) 10 kHz	—	90	—	dB
	PSRR (DCVDD, FLLVDD, CPVDD2), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	125	—	dB
		100 mV (peak-peak) 10 kHz	—	105	—	dB
	PSRR (SPKVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	120	—	dB
		100 mV (peak-peak) 10 kHz	—	90	—	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 DMICDAT1/2 and DMICCLK1/2) [1,3]	Input HIGH level	V_{DBVDDn} = 1.71–1.98 V	0.75 × DBVDD*n*	—	—	V
		V_{DBVDDn} = 2.5 V ±10%	0.8 × DBVDD*n*	—	—	V
		V_{DBVDDn} = 3.3 V ±10%	0.7 × DBVDD*n*	—	—	V
	Input LOW level	V_{DBVDDn} = 1.71–1.98 V	—	—	0.3 × DBVDD*n*	V
		V_{DBVDDn} = 2.5 V ±10%	—	—	0.25 × DBVDD*n*	V
		V_{DBVDDn} = 3.3 V ±10%	—	—	0.2 × DBVDD*n*	V
	Output HIGH level (I_{OH} = 1 mA)	V_{DBVDDn} = 1.71–1.98 V	0.75 × DBVDD*n*	—	—	V
		V_{DBVDDn} = 2.5 V ±10%	0.65 × DBVDD*n*	—	—	V
		V_{DBVDDn} = 3.3 V ±10%	0.7 × DBVDD*n*	—	—	V
	Output LOW level (I_{OL} = 1mA)	V_{DBVDDn} = 1.71–1.98 V	—	—	0.25 × DBVDD*n*	V
		V_{DBVDDn} = 2.5 V ±10%	—	—	0.3 × DBVDD*n*	V
		V_{DBVDDn} = 3.3 V ±10%	—	—	0.15 × DBVDD*n*	V
	Input capacitance		—	—	5	pF
	Input leakage		–1	—	1	µA
	Pull-up/pull-down resistance (where applicable)		35	—	55	kΩ
DMIC I/O (DMICDAT1/2 and DMICCLK1/2) [2,3]	DMICDAT*n* input HIGH Level		0.65 × V_{SUP}	—	—	V
	DMICDAT*n* input LOW Level		—	—	0.35 × V_{SUP}	V
	DMICCLK*n* output HIGH Level	I_{OH} = 1 mA	0.8 × V_{SUP}	—	—	V
	DMICCLK*n* 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 DBVDD1 or DBVDD2.
2. DMICDAT1/2 and DMICCLK1/2 are referenced to a selectable supply, V_{SUP}, according to the IN*n*_DMIC_SUP fields.
3. Note that digital input pins should not be left unconnected or floating.

Table 3-11. Miscellaneous Characteristics

Test conditions (unless specified otherwise): DBVDD1 = DBVDD2 = CPVDD1 = AVDD = 1.8 V, DCVDD = FLLVDD = CPVDD2 = 1.2 V; MICVDD = 2.5 V (powered from internal LDO); 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, MICBIAS2A, MICBIAS2B) [1]	Minimum Bias Voltage [2]		—	1.5	—	V
	Maximum Bias Voltage		—	2.8	—	V
	Bias Voltage output step size		—	0.1	—	V
	Bias Voltage accuracy		–5%	—	+5%	V
	Bias Current [3]	Regulator Mode (MICB*n*_BYPASS = 0), $V_{MICVDD} - V_{MICBIAS}$ >200 mV	—	—	2.4	mA
		Bypass Mode (MICB*n*_BYPASS = 1)	—	—	5.0	mA
	Output Noise Density	Regulator Mode (MICB*n*_BYPASS = 0), MICB*n*_LVL = 0x4, Load current = 1 mA, Measured at 1 kHz	—	45	—	nV/√Hz
	Integrated noise voltage	Regulator Mode (MICB*n*_BYPASS = 0), MICB*n*_LVL = 0x4, Load current = 1 mA, 100 Hz to 7 kHz, A-weighted	—	4	—	μV_{RMS}
	PSRR (DBVDD*n*, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	105	—	dB
		100 mV (peak-peak) 10 kHz	—	85	—	dB
	PSRR (DCVDD, FLLVDD, CPVDD2), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	110	—	dB
		100 mV (peak-peak) 10 kHz	—	90	—	dB
	PSRR (SPKVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	105	—	dB
		100 mV (peak-peak) 10 kHz	—	90	—	dB
	Load capacitance [3]	Regulator Mode (MICB*n*_BYPASS = 0), MICB*n*_EXT_CAP = 0	—	—	50	pF
		Regulator Mode (MICB*n*_BYPASS = 0), MICB*n*_EXT_CAP = 1	0.1	1.0	10	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	—	25	40	Ω
		Switch open	—	100	—	MΩ
External Accessory Detect	Load impedance detection range: Detection via HPDETL (ACCDET_MODE = 001) or HPDETR (ACCDET_MODE = 010)	HP_IMPEDANCE_RANGE = 00	4	—	30	Ω
		HP_IMPEDANCE_RANGE = 01	8	—	100	Ω
		HP_IMPEDANCE_RANGE = 10	100	—	1000	Ω
		HP_IMPEDANCE_RANGE = 11	1000	—	10000	Ω
	Load impedance detection range: Detection via MICDET1 or MICDET2 pin (ACCDET_MODE = 100)		400	—	6000	Ω
	Load impedance detection accuracy (HP_DACVAL, ACCDET_MODE = 001 or 010)	HP_IMPEDANCE_RANGE = 01 or 10	–5	—	+5	%
		HP_IMPEDANCE_RANGE = 00 or 11	–10	—	+10	%
	Load impedance detection accuracy (HP_LVL, ACCDET_MODE = 001, 010, or 100)		–20	—	+20	%
	Load impedance detection range—Detection via MICDET1 or MICDET2 pin (ACCDET_MODE = 000); 2.2 kΩ (±2%) MICBIAS resistor. [5]	for MICD_LVL[0] = 1	0	—	70	Ω
		for MICD_LVL[1] = 1	110	—	180	Ω
		for MICD_LVL[2] = 1	210	—	290	Ω
		for MICD_LVL[3] = 1	360	—	680	Ω
		for MICD_LVL[8] = 1	1000	—	30000	Ω
	Jack-detection input threshold voltage (JACKDET*n*)	Detection on JACKDET1, Jack insertion	—	0.9	—	V
		Detection on JACKDET1, Jack removal	—	1.65	—	V
		Detection on JACKDET2, Jack insertion	—	0.27	—	V
		Detection on JACKDET2, Jack removal	—	0.9	—	V
	Pull-up resistance (JACKDET*n*)		—	1	—	MΩ
MICVDD Charge Pump and Regulator (CP2 and LDO2)	Output voltage		0.9	2.7	3.3	V
	Programmable output voltage step size	LDO2_VSEL = 0x00–0x14 (0.9–1.4V)	—	25	—	mV
		LDO2_VSEL = 0x14 to 0x27 (1.4 V–3.3 V)	—	100	—	mV
	Maximum output current		—	8	—	mA
	Start-up time	4.7 µF on MICVDD	—	1.5	2.5	ms
Frequency-Locked Loop (FLL1)	Output frequency	FLL output as SYSCLK source	90	—	98.3	MHz
		FLL output as DSPCLK source	135	—	150	MHz
	Lock Time	F_{REF} = 32 kHz, F_{OUT} (DSPCLK source) = 147.456 MHz	—	10	—	ms
		F_{REF} = 12 MHz, F_{OUT} (DSPCLK source) = 147.456 MHz	—	1	—	ms
$\overline{\text{RESET}}$ pin input	$\overline{\text{RESET}}$ input pulse width [6]		1	—	—	µs

1. No capacitor on MICBIAS*n*. In Regulator Mode, it is required that $V_{MICVDD} - V_{MICBIAS}$ > 200 mV.
2. Regulator Mode (MICB*n*_BYPASS = 0), Load current ≤ 1.0 mA.
3. Bias current and load capacitance specifications are per MICBIAS generator (MICBIAS1 or MICBIAS2).
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}	—	—	1.66	V
	V_{AVDD} falling		1.06	—	1.44	V
DCVDD reset threshold	V_{DCVDD} rising	V_{DCVDD}	—	—	1.04	V
	V_{DCVDD} falling		0.49	—	0.66	V
DBVDD1 Reset threshold	V_{DBVDD1} rising	V_{DBVDD1}	—	—	1.66	V
	V_{DBVDD1} falling		1.06	—	1.44	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]	Symbol	Minimum	Typical	Maximum	Units
DMICCLK*n* cycle time	t_{CY}	160	163	1432	ns
DMICCLK*n* duty cycle	—	45	—	55	%
DMICCLK*n* rise/fall time (25-pF load, 1.8-V supply)	t_r, t_f	5	—	30	ns
DMICDAT*n* (Left) setup time to falling DMICCLK edge	t_{LSU}	15	—	—	ns
DMICDAT*n* (Left) hold time from falling DMICCLK edge	t_{LH}	0	—	—	ns
DMICDAT*n* (Right) setup time to rising DMICCLK edge	t_{RSU}	15	—	—	ns
DMICDAT*n* (Right) hold time from rising DMICCLK edge	t_{RH}	0	—	—	ns

Note: The voltage reference for the IN1 and IN2 DMIC interfaces is selectable, using the IN*n*_DMIC_SUP fields—each interface may be referenced to MICVDD, MICBIAS1B, MICBIAS2A, or MICBIAS2B levels.

1. DMIC interface timing

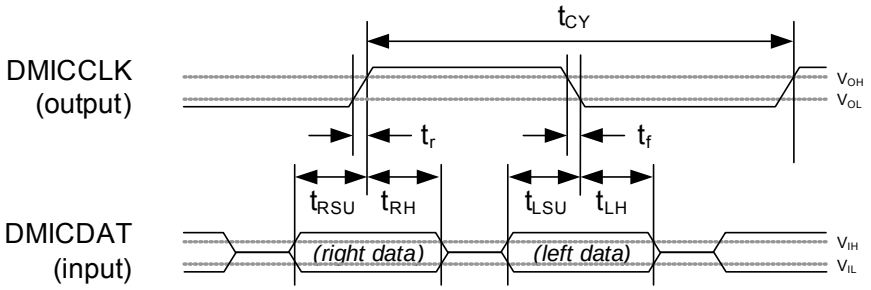
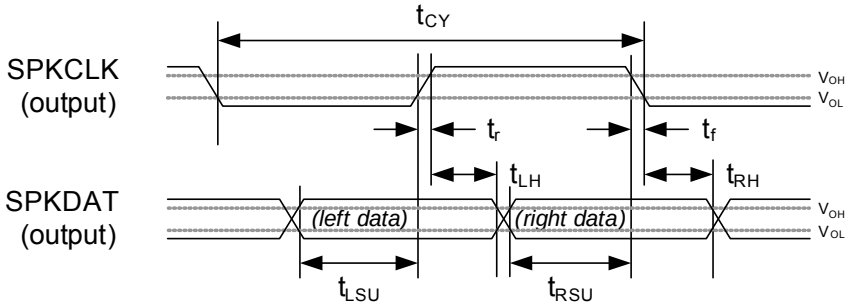


Table 3-15. Digital Speaker (PDM) Interface Timing

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
	SPKDAT set-up time to SPKCLK rising edge (left channel)	t_{LSU}	30	—	—	ns
	SPKDAT hold time from SPKCLK rising edge (left channel)	t_{LH}	30	—	—	ns
	SPKDAT set-up time to SPKCLK falling edge (right channel)	t_{RSU}	30	—	—	ns
	SPKDAT 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
	SPKDAT enable from SPKCLK rising edge (right channel)	t_{REN}	—	—	15	ns
	SPKDAT disable to SPKCLK falling edge (right channel)	t_{RDIS}	—	—	5	ns
	SPKDAT enable from SPKCLK falling edge (left channel)	t_{LEN}	—	—	15	ns
	SPKDAT 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

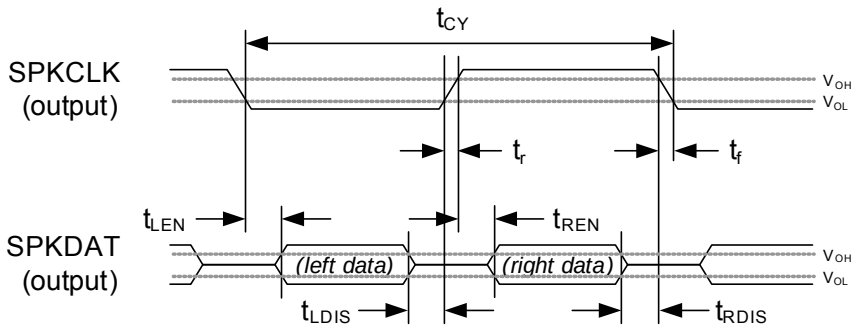


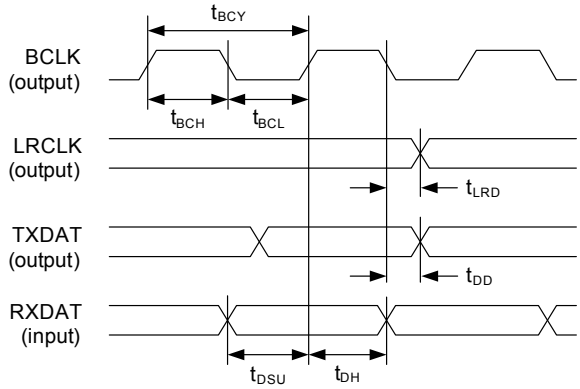
Table 3-16. Digital Audio Interface—Master Mode

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.

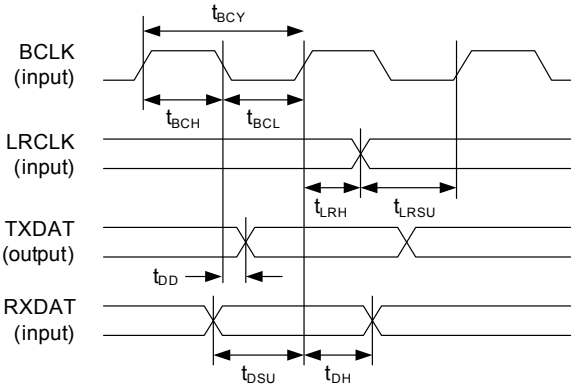
Table 3-17. Digital Audio Interface—Slave Mode

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.

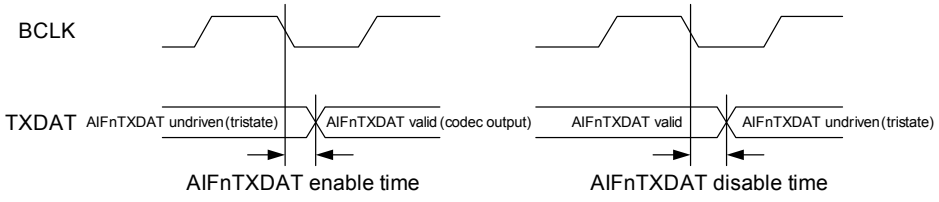


Table 3-19. Control Interface Timing—Two-Wire (I²C) Mode

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	t_6	—	—	80	ns
	SCLK frequency > 1MHz	t_6	—	—	160	ns
	SCLK frequency ≤ 1MHz	t_6	—	—	2000	ns
SDA, SCLK Fall Time (90%–10%)	SCLK frequency > 1.7MHz	t_7	—	—	60	ns
	SCLK frequency > 1MHz	t_7	—	—	160	ns
	SCLK frequency ≤ 1MHz	t_7	—	—	200	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	t_{10}	—	—	40	ns
	SCLK slew (90%–10%) = 60ns, C_{LOAD} (SDA) = 100 pF	t_{10}	—	—	130	ns
	SCLK slew (90%–10%) = 160ns, C_{LOAD} (SDA) = 400 pF	t_{10}	—	—	190	ns
	SCLK slew (90%–10%) = 200ns, C_{LOAD} (SDA) = 550 pF	t_{10}	—	—	220	ns
Pulse width of spikes that are suppressed		t_{ps}	0	—	25	ns

1. Control interface timing—I²C Mode

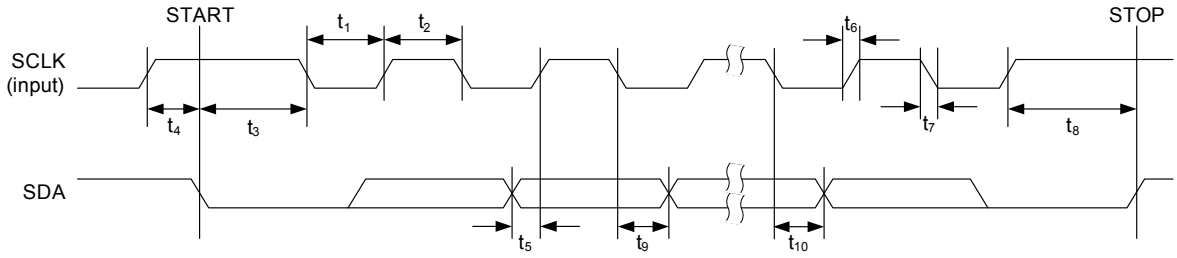
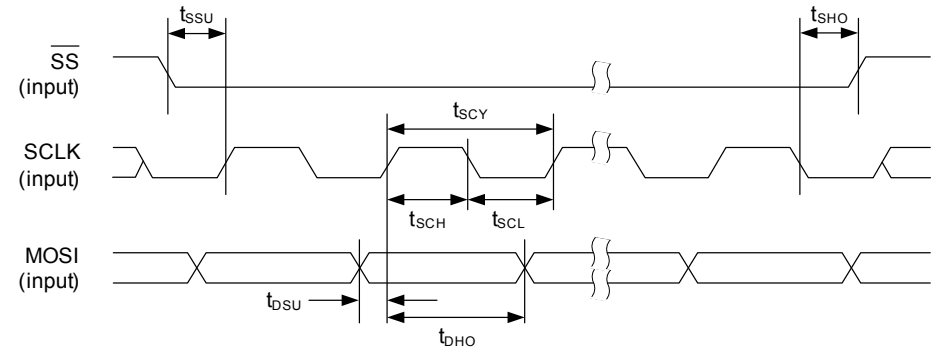


Table 3-20. Control Interface Timing—Four-Wire (SPI) Mode

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}	50.0	—	—	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)

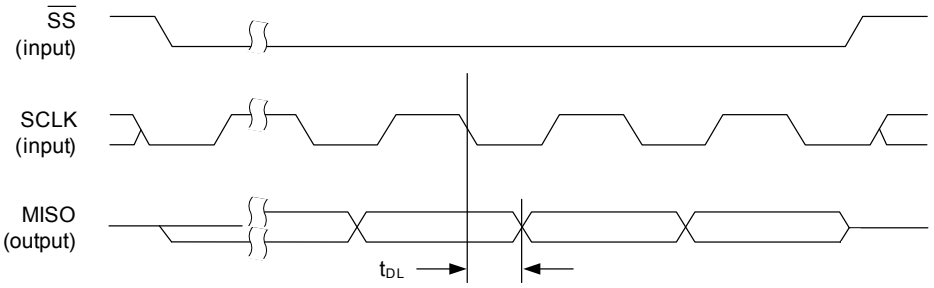


Table 3-21. SLIMbus Interface Timing

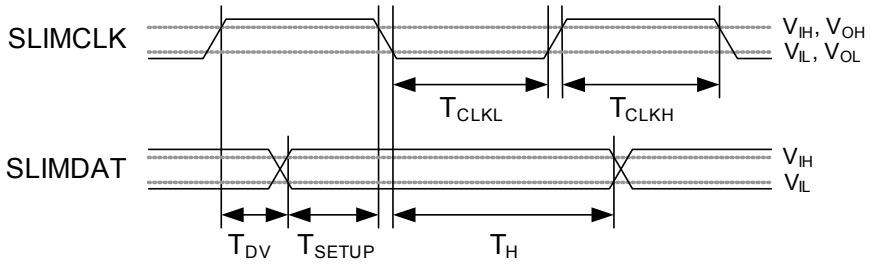
The following timing information is valid across the full range of recommended operating conditions.

Parameter [1]			Symbol	Minimum	Typ	Maximum	Units
SLIMCLK input	SLIMCLK cycle time		—	35	—	—	ns
	SLIMCLK pulse width high		T_{CLKH}	12	—	—	ns
	SLIMCLK pulse width low		T_{CLKL}	12	—	—	ns
SLIMCLK output	SLIMCLK cycle time		—	40	—	—	ns
	SLIMCLK pulse width high		T_{CLKH}	12	—	—	ns
	SLIMCLK pulse width low		T_{CLKL}	12	—	—	ns
	SLIMCLK slew rate (20%–80%)	C_{LOAD} = 15 pF, SLIMCLK_DRV_STR = 0	SR_{CLK}	0.09 x V_{DBVDD1}	—	0.22 x V_{DBVDD1}	V/ns
		C_{LOAD} = 70 pF, SLIMCLK_DRV_STR = 0	SR_{CLK}	0.02 x V_{DBVDD1}	—	0.05 x V_{DBVDD1}	V/ns
		C_{LOAD} = 70 pF, SLIMCLK_DRV_STR = 1	SR_{CLK}	0.04 x V_{DBVDD1}	—	0.11 x V_{DBVDD1}	V/ns
SLIMDAT input	SLIMDAT setup time to SLIMCLK falling edge		T_{SETUP}	3.5	—	—	ns
	SLIMDAT hold time from SLIMCLK falling edge		T_H	2	—	—	ns
SLIMDAT output	SLIMDAT time for data output valid (relative to SLIMCLK rising edge)	SLIMDAT_DRV_STR = 0, DBVDD1 = 1.71 V	T_{DV}	—	4.7	8.1	ns
		C_{LOAD} = 15 pF, SLIMDAT_DRV_STR = 1, DBVDD1 = 1.71 V	T_{DV}	—	4.3	7.3	ns
		C_{LOAD} = 30 pF, SLIMDAT_DRV_STR = 0, DBVDD1 = 1.71 V	T_{DV}	—	6.8	11.8	ns
		C_{LOAD} = 30 pF, SLIMDAT_DRV_STR = 1, DBVDD1 = 1.71 V	T_{DV}	—	5.8	10.0	ns
		C_{LOAD} = 50 pF, SLIMDAT_DRV_STR = 0, DBVDD1 = 1.71 V	T_{DV}	—	9.6	16.6	ns
		C_{LOAD} = 50 pF, SLIMDAT_DRV_STR = 1, DBVDD1 = 1.71 V	T_{DV}	—	7.9	13.7	ns
		C_{LOAD} = 70 pF, SLIMDAT_DRV_STR = 0, DBVDD1 = 1.71 V	T_{DV}	—	12.4	21.5	ns
		C_{LOAD} = 70 pF, SLIMDAT_DRV_STR = 1, DBVDD1 = 1.71 V	T_{DV}	—	10.0	17.4	ns
	SLIMDAT slew rate (20%–80%)	C_{LOAD} = 15 pF, SLIMDAT_DRV_STR = 0	SR_{DATA}	—	—	0.64 x V_{DBVDD1}	V/ns
		C_{LOAD} = 30 pF, SLIMDAT_DRV_STR = 0	SR_{DATA}	—	—	0.35 x V_{DBVDD1}	V/ns
		C_{LOAD} = 30pF, SLIMDAT_DRV_STR = 1	SR_{DATA}	—	—	0.46 x V_{DBVDD1}	V/ns
		C_{LOAD} = 70pF, SLIMDAT_DRV_STR = 0	SR_{DATA}	—	—	0.16 x V_{DBVDD1}	V/ns
		C_{LOAD} = 70pF, SLIMCLK_DRV_STR = 1	SR_{DATA}	—	—	0.21 x V_{DBVDD1}	V/ns
Other parameters	Driver disable time		T_{DD}	—	—	6	ns
	Bus holder output impedance	0.1 x V_{DBVDD1} < V < 0.9 x V_{DBVDD1}	R_{DATAS}	18	—	50	kΩ

Notes:

- The signal timing information describes the timing requirements of the SLIMbus interface as a whole, not just the CS47L35 device.
- T_{DV} is the propagation delay from the rising SLIMCLK edge (at CS47L35 input) to the SLIMDAT output being achieved at the input to all devices across the bus.
- T_{SETUP} is the set-up time for SLIMDAT input (at CS47L35), relative to the falling SLIMCLK edge (at CS47L35).
- T_H is the hold time for SLIMDAT input (at CS47L35) relative to the falling SLIMCLK edge (at CS47L35).
- For more details of the interface timing, refer to the *MIPI Alliance Specification for Serial Low-power Inter-Chip Media Bus (SLIMbus)*

1.SLIMbus interface timing.



V_{IL}, V_{IH} are the 35%/65% levels of the respective inputs.
V_{OL}, V_{OH} are the 20%/80% levels of the respective outputs.
The SLIMDAT output delay (T_{DV}) is with respect to the input pads of all receiving devices

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

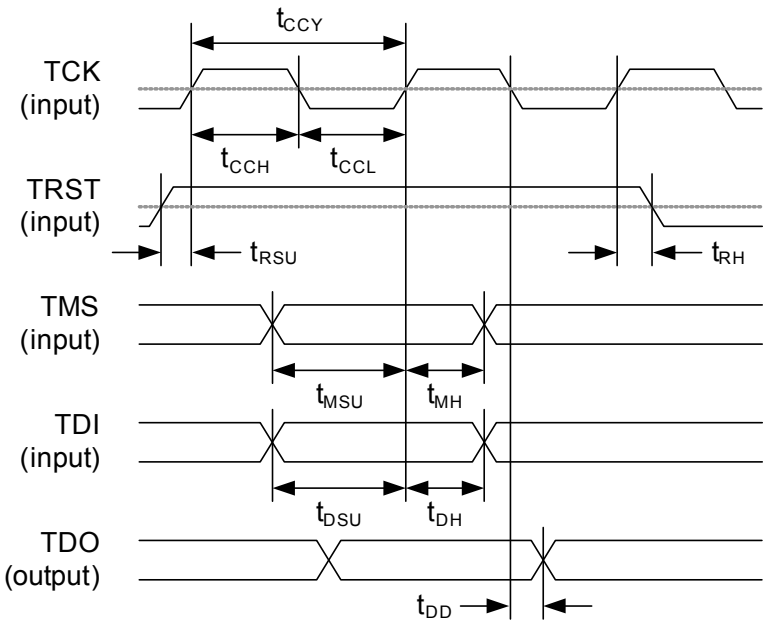


Table 3-23. Typical Power Consumption

Test conditions (unless specified otherwise): DBVDD1 = DBVDD2 = CPVDD1 = AVDD = 1.8 V, DCVDD = FLLVDD = CPVDD2 = 1.2 V; MICVDD = Off (CP2 and LDO2 disabled); SPKVDD = 4.2 V; T_A = +25°C; Fs = 48 kHz; 24-bit audio data, I²S Slave Mode; SYSCLK = 24.576 MHz (direct MCLK1 input).

Operating Configuration			Typical $I_{1.2V}$ (mA)	Typical $I_{1.8V}$ (mA)	Typical $I_{4.2V}$ (mA)	P_{TOT} (mW)
Headphone playback	AIF1 to DAC to HPOUT (stereo), 32-Ω load.	Quiescent	1.00	0.75	0.00	2.54
		1-kHz sine wave, P_O = 10 mW	1.06	36.5	0.00	67.0
Earpiece playback	AIF1 to DAC to EPOUT, 32-Ω load (BTL).	Quiescent	0.86	0.75	0.00	2.38
		1-kHz sine wave, P_O = 30 mW	0.86	61.8	0.00	112
Speaker playback	AIF1 to DAC to SPKOUT, 8-Ω, 22-µH load.	Quiescent	0.76	1.03	0.10	3.19
		1-kHz sine wave, P_O = 700 mW	0.79	1.10	180	759
Stereo line record	Analog line to ADC to AIF1, MICVDD = 1.8V (CP2 and LDO2 bypass enabled).	1-kHz sine wave, -1 dBFS output	1.23	2.52	0.00	6.01
Sleep Mode	Accessory detect enabled (JD1_ENA = 1)		0.000	0.013	0.000	0.023

Table 3-24. Typical Signal Latency

Test conditions (unless specified otherwise): DBVDD1 = DBVDD2 = CPVDD1 = AVDD = 1.8 V, DCVDD = FLLVDD = CPVDD2 = 1.2 V; MICVDD = Off (CP2 and LDO2 disabled); SPKVDD = 4.2 V; T_A = +25°C; Fs = 48 kHz; 24-bit audio data, I²S Slave Mode; SYSCLK = 24.576 MHz (direct MCLK1 input).

Operating Configuration			Latency (µs)
AIF to DAC path	Digital input (AIFn) to analog output (HPOUT). Signal is routed via the ISRC function in the isochronous cases only.	48 kHz input, 48 kHz output, Synchronous	344
		44.1 kHz input, 44.1 kHz output, Synchronous	371
		16 kHz input, 16 kHz output, Synchronous	665
		8 kHz input, 8 kHz output, Synchronous	1105
		8 kHz input, 48 kHz output, Isochronous	1660
		16 kHz input, 48 kHz output, Isochronous	1170
ADC to AIF path	Analog input (INn) to digital output (AIFn). Digital core high-pass filter is included in the signal path. Signal is routed via the ISRC function in the isochronous cases only.	48 kHz input, 48 kHz output, Synchronous	210
		44.1 kHz input, 44.1 kHz output, Synchronous	225
		16 kHz input, 16 kHz output, Synchronous	620
		8 kHz input, 8 kHz output, Synchronous	1210
		8 kHz input, 48 kHz output, Isochronous	1765
		16 kHz input, 48 kHz output, Isochronous	965

4 Functional Description

The CS47L35 is a highly integrated, low-power audio hub codec for mobile telephony and portable 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 devices.

4.1 Overview

The CS47L35 block diagram is shown in Fig. 4-1.

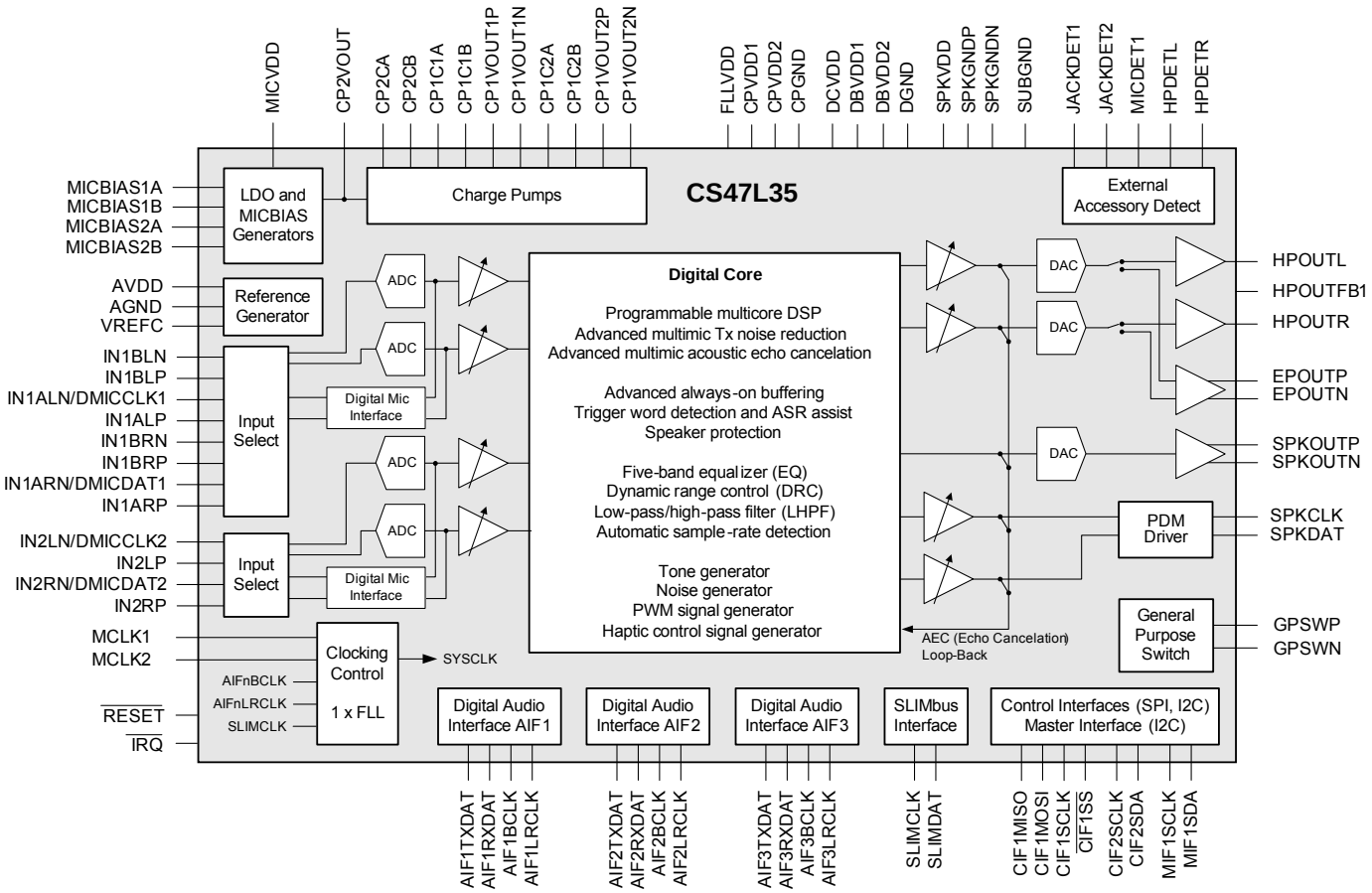


Figure 4-1. CS47L35 Block Diagram

The CS47L35 digital core provides an extensive capability for signal-processing algorithms, including transmit (TX) path noise reduction, acoustic-echo cancelation (AEC), and other programmable filters. The DSPs are ideally suited to the Cirrus Logic® SoundClear® suite of audio processing algorithms, such as the SoundClear Control always-on voice control software.

The digital core provides signal-processing capability for sensor-hub functions. The integration of external sensors with the programmable DSP enables increased contextual awareness in a variety of advanced user applications.

The CS47L35 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 CS47L35 provides multiple digital audio interfaces, including SLIMbus, to provide independent isochronous connections to different processors (e.g., application processor, baseband processor, and wireless transceiver).

A flexible clocking arrangement supports a wide variety of external clock references, including clocking derived from the digital audio interface. The frequency-locked loop (FLL) circuit provides additional flexibility.

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 CS47L35 always-on circuitry can be used in conjunction with the Apps Processor to wake up the device following a headphone jack-detection event.

Versatile GPIO functionality is provided, including support for external accessory/push-button detection inputs. Comprehensive interrupt functions, with status reporting, are also provided.

4.1.1 Hi-Fi Audio Codec

The CS47L35 is a high-performance, low-power audio codec that uses a simple analog architecture. Four ADCs are incorporated, with multiplexers to support up to six analog inputs. Three DACs are incorporated, with two being switchable between the headphone and BTL-earpiece analog output paths.

The analog outputs comprise a 30-mW (122 dB SNR) stereo headphone amplifier with ground-referenced output, a mono (BTL) earpiece driver, and a mono Class D speaker driver capable of delivering 2.7 W into a 4-Ω load. Six analog inputs are provided (multiplexed into four input channels), each supporting single-ended or differential input modes. In differential mode, the input path SNR is 105 dB (16 kHz sample rate, i.e., wideband voice mode). The ADC input paths can be bypassed, supporting up to four channels of DMIC input.

The audio codec is controlled directly via register access. The simple analog architecture, combined with the integrated tone generator, enables straightforward device configuration and testing, minimizing debug time and reducing software effort.

The CS47L35 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.

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 CS47L35 is cost optimized for a wide range of mobile phone 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 CS47L35 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. This helps support many new audio use cases. Soft mute and unmute control allows smooth transitions between use cases without interrupting existing audio streams elsewhere.

The CS47L35 digital core provides an extensive capability for programmable signal-processing algorithms. The SoundClear suite of software algorithms enable advanced multimic audio features, such as transmit (TX) path noise reduction, AEC, wind-noise reduction, and other programmable filters. Enhancements such as DRC and MBC are also provided.

The CS47L35 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 input paths to support current monitoring in the speaker output; this allows the Class D output to be continually optimized for the operational limits of the speaker, and enables maximum audio output while ensuring the loudspeakers are fully protected from damage.

The digital core also provides signal-processing capability for sensor-hub functions of the CS47L35. Sensors and accessories can be connected through the master I²C interface; the programmable DSP, together with peripheral timer and event logging functions, enables applications to use these inputs to support increased contextual awareness, including advanced motion sensing and navigation functionality.

Highly flexible digital mixing, including mixing between audio interfaces, is possible. The CS47L35 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 I²S data formats for compatibility with most industry-standard chipsets. AIF1 supports six input/output channels; AIF2 and AIF3 support two input/output channels each. Bidirectional operation at sample rates up to 192 kHz is supported.

Four digital PDM input channels are available (two stereo interfaces); these are typically used for digital microphones, powered from the integrated MICBIAS power-supply regulators. Two PDM output channels are also available (one stereo interface); these are typically used for external power amplifiers. Embedded mute codes provide a control mechanism for external PDM-input devices.

The CS47L35 features a SLIMbus interface, compliant with the MIPI® SLIMbus specification, providing six channels of audio input/output. Mixed audio sample rates are supported on the SLIMbus interface. The SLIMbus interface also supports read/write access to the CS47L35 control registers.

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 interface and a slave I²C control interface. The SPI interface operates up to 26 MHz; the I²C slave interface operates up to 3.4 MHz. Full access to the register map is also provided via the SLIMbus port.

The CS47L35 incorporates one master I²C interface, offering capability for additional sensor/accessory input. Typical sensors include accelerometers, gyroscopes and magnetometers for motion sensing and navigation applications. Other example accessories include barometers, or ambient light sensors, for environmental awareness.

4.1.4 Other Features

The CS47L35 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 CS47L35 supports up to 16 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.

The CS47L35 incorporates four general-purpose timers, providing support for the sensor-hub connectivity. Sensor event logging, and other real time application functions, allows many advanced functions to be implemented with a high degree of autonomy from a host processor.

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 SLIMbus interface, or the audio interfaces (configured in Slave Mode), can be used to provide a clock reference. The integrated FLL circuit provides support for a wide range of clocking configurations, including the use of a 32-kHz input clock reference.

The CS47L35 can be powered from 1.8- and 1.2-V external supplies. A separate supply (4.2 V) is typically required for the Class D speaker driver. Integrated charge-pump and LDO-regulator circuits are used to generate supply rails for internal functions and to support powering or biasing of external microphones.

4.2 Input Signal Path

The CS47L35 provides flexible input channels, supporting up to six 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 analog input paths support single-ended and differential modes, programmable gain control, and are digitized using a high performance sigma-delta ADC.

The digital input paths interface directly with external digital microphones; a separate microphone interface clock is provided for two separate stereo pairs of digital microphones. Digital delay can be applied to any of the digital input paths; this can be used for phase adjustment of any digital input, including directional control of multiple microphones.

Two microphone bias (MICBIAS) generators provide 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 generators allow four 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.

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

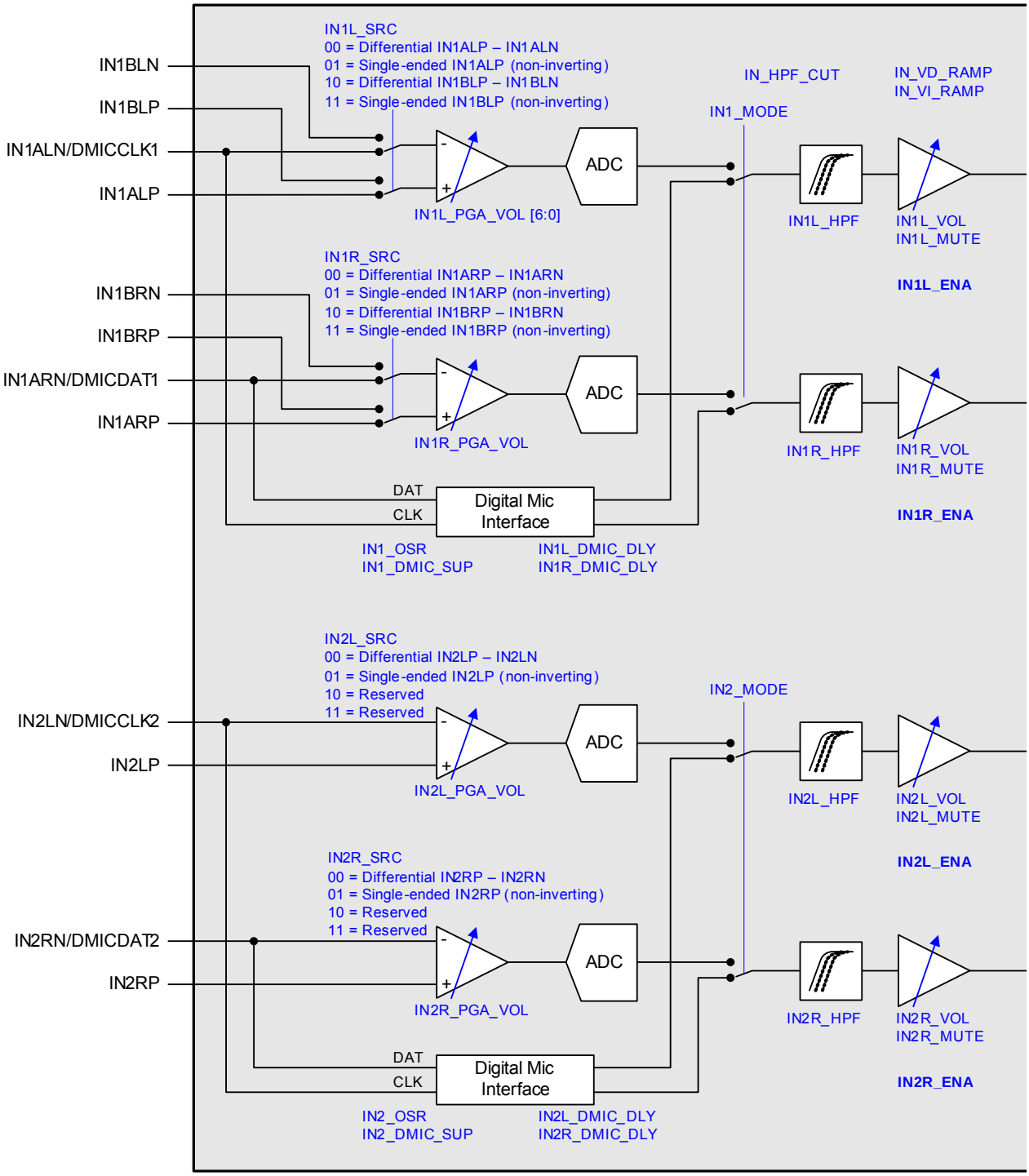


Figure 4-2. Input Signal Paths

4.2.1 Analog Microphone Input

Up to six analog microphones can be connected to the CS47L35, either in single-ended or differential mode. The applicable mode, and input pin selection, is controlled using IN*nx*_SRC, as described in Section 4.2.6.

The CS47L35 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 (IN*n*LP or IN*n*RP). 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 (IN*n*LP or IN*n*RP), while the inverted (or noisy ground) signal is connected to the inverting input pins (IN*n*LN or IN*n*RN).

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 IN*nx*N capacitor), instead of via a resistor. This is the recommended configuration if the external accessory detection functions on the CS47L35 are used. The IN*nx*_SRC field settings are the same for pseudodifferential connection as for differential.

The gain of the input PGAs is controlled via register settings, as defined in Section 4.2.6. 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 generators provide a low noise reference for biasing the ECMs.

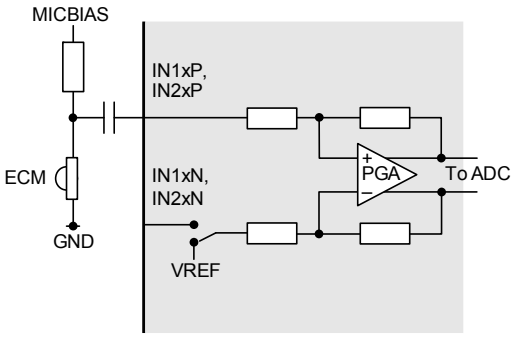


Figure 4-3. Single-Ended ECM Input

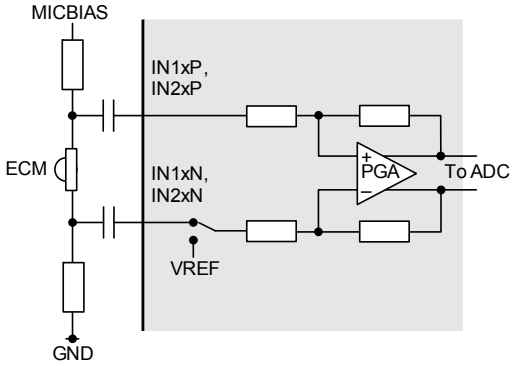


Figure 4-4. Differential ECM Input

Analog MEMS microphones can be connected to the CS47L35 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 generators provide a low-noise power supply for the microphones.

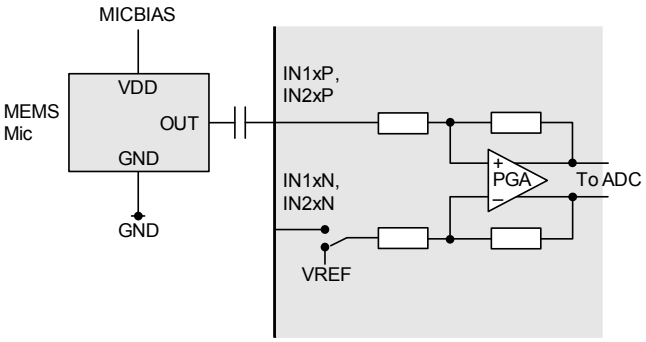


Figure 4-5. Single-Ended MEMS Input

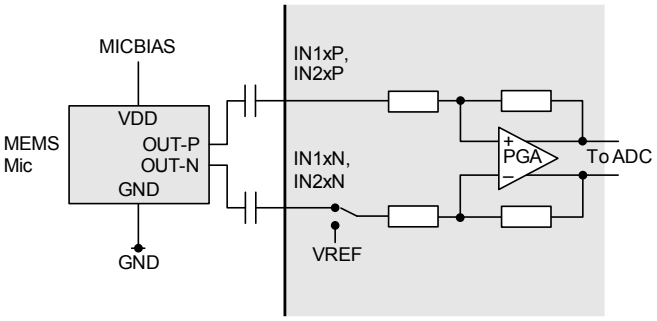


Figure 4-6. Differential MEMS Input

Note: The MICVDD pin can also be used (instead of MICBIAS) as a reference or power supply for external microphones. The MICBIAS outputs are recommended, because they offer better noise performance and independent enable/disable control.

4.2.2 Analog Line Input

Line inputs can be connected to the CS47L35 in a similar manner to the mic inputs. Single-ended and differential modes are supported on each analog input path. The mode is selected using IN*nx*_SRC, as described in Section 4.2.6.

The 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.

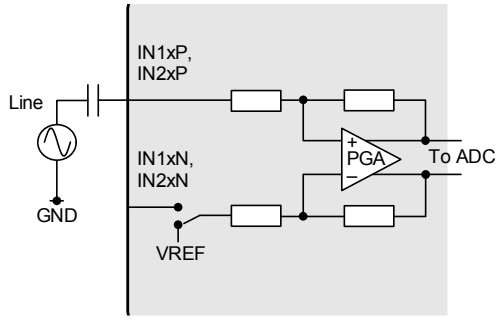


Figure 4-7. Single-Ended Line Input

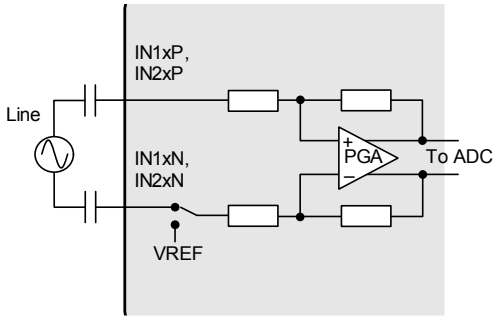


Figure 4-8. Differential Line Input

4.2.3 DMIC Input

As many as four digital microphones can be connected to the CS47L35. DMIC operation on input paths IN1 and IN2 is selected using IN*n*_MODE, as described in Section 4.2.6.

In DMIC mode, two channels of audio data are multiplexed on the associated DMICDAT*n* pin. Each stereo DMIC interface is clocked using the respective DMICCLK*n* pin.

If DMIC input is enabled, the CS47L35 outputs a clock signal on the applicable DMICCLK*n* pins. The DMICCLK*n* frequency is controlled by the respective IN*n*_OSR field, as described in Table 4-1 and Table 4-3.

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

The DMICCLK*n* 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*n* frequencies are scaled accordingly.

Table 4-1. DMICCLK Frequency

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

The voltage reference for the IN1 and IN2 DMIC interfaces is selectable, using IN*n*_DMIC_SUP; each interface may be referenced to MICVDD or to the MICBIAS1B, MICBIAS2A, or MICBIAS2B levels. The voltage reference for each digital input path should be set equal to the applicable power supply of the respective microphones.

A pair of digital microphones is connected as shown in Fig. 4-9. 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 CS47L35 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 CS47L35 provides integrated pull-down resistors on the DMICDAT*n* pins. This provides a flexible capability for interfacing with other devices.

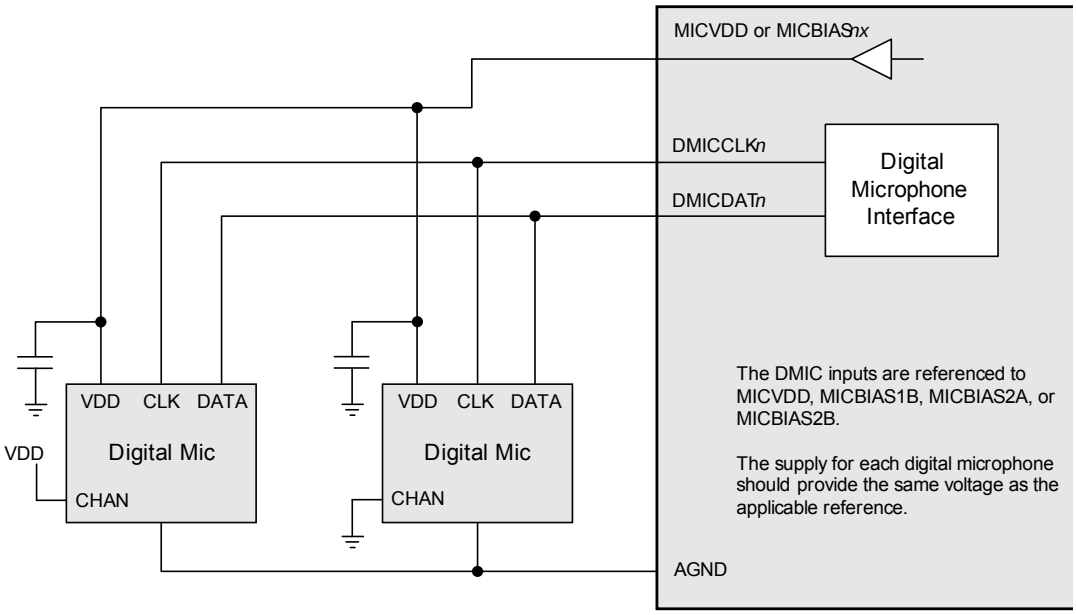


Figure 4-9. DMIC Input

Two DMIC channels are interleaved on DMICDAT*n*. The DMIC interface timing is shown in Fig. 4-10. Each microphone must tristate its data output when the other microphone is transmitting.

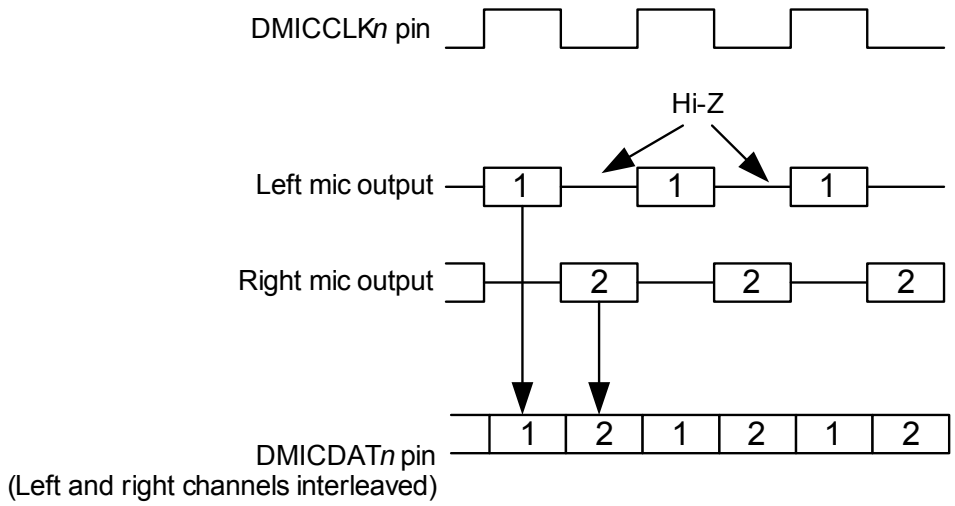


Figure 4-10. DMIC Interface Timing

When DMIC input is enabled, the CS47L35 outputs a clock signal on the applicable DMICCLK pins. The DMICCLK frequency is selectable, as described in Table 4-1.

Note that SYSCLK must be present and enabled when using the DMIC inputs; see Section 4.16 for details regarding SYSCLK and the associated registers.

4.2.4 Input Signal Path Enable

The input signal paths are enabled using the bits described in Table 4-2. 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 4-4.

The MICVDD power domain must be enabled when using the analog input signal paths. This power domain is provided using an internal charge pump (CP2) and LDO regulator (LDO2). See Section 4.19 for details of these circuits.

The system clock, SYSCLK, must be configured and enabled before any audio path is enabled. The input signal paths should be kept disabled (IN*nx*_ENA = 0) if SYSCLK is not enabled. The 32-kHz clock may also be required, depending on the path configuration. See Section 4.16 for details of the system clocks (including requirements for reconfiguring SYSCLK while audio paths are enabled).

The CS47L35 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 4-2. 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

4.2.5 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 CS47L35 digital core. The sample rate for the input signal paths is configured using IN_RATE; see Table 4-21.

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.

4.2.6 Input Signal Path Configuration

The CS47L35 supports up to six 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.

Input paths IN1 and IN2 can be configured as single-ended, differential, or DMIC configuration. The input signal path configuration is selected using IN*n*_MODE and IN*n*x_SRC.

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*n*x_HPF bits.

The analog input signal 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 IN*n*L_PGA_VOL and IN*n*R_PGA_VOL. Note that separate volume control is provided for the left and right channels of each stereo pair.

If the IN1 or IN2 input signal path is configured for DMIC input, the voltage reference for the associated input/output pins is selectable using the IN*n*_DMIC_SUP fields—each interface may be referenced to MICVDD, or to the MICBIAS1B, MICBIAS2A, or MICBIAS2B levels. The voltage reference for each digital input path should be set equal to the applicable power supply of the respective microphones.

When the input signal path is configured for DMIC input, the respective DMICCLK*n* frequency can be configured using the IN*n*_OSR bits.

A digital delay may be applied to any of the DMIC input channels. This feature can be used for phase adjustment of any digital input, including directional control of multiple microphones. The delay is controlled using IN*n*L_DMIC_DLY and IN*n*R_DMIC_DLY.

The MICVDD voltage is generated by an internal charge pump and LDO regulator. The MICBIAS*nx* outputs are derived from MICVDD; see Section 4.19.

The input signal paths are configured using the fields described in Table 4-3.

Table 4-3. 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 001 = 5 Hz 010 = 10 Hz 011 = 20 Hz 100 = 40 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 DMICDAT1 and DMICCLK1 logic levels) 00 = MICVDD 01 = MICBIAS1B 10 = MICBIAS2A 11 = MICBIAS2B
	10	IN1_MODE	00	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 0x40 = 0 dB 0x41 = 1 dB 0x42 = 2 dB … (1-dB steps) 0x5F = 31 dB 0x60 to 0x7F = Reserved
R785 (0x0311) ADC_Digital_Volume_1L	14:13	IN1L_SRC[1:0]	00	Input Path 1 (Left) Source 00 = Differential (IN1ALP–IN1ALN) 01 = Single-ended (IN1ALP) 10 = Differential (IN1BP–IN1BN) 11 = Single-ended (IN1BP)
R786 (0x0312) DMIC1L_Control	10:8	IN1_OSR[2:0]	101	Input Path 1 DMIC Oversample Rate (applicable to digital inputs only). 000 = Reserved 001 = Reserved 010 = 384 kHz 011 = 768 kHz 100 = 1.536 MHz 101 = 3.072 MHz 110 = 6.144 MHz 111 = Reserved If IN1_OSR=010 or 011, the maximum Input Path sample rate (all input paths) is 48 kHz or 96 kHz respectively.
	5:0	IN1L_DMIC_DLY[5:0]	0x00	Input Path 1 (Left) Digital Delay (applicable to digital input only) LSB = 1 sample, Range is 0 to 63. (Sample rate is controlled by IN1_OSR.)

Table 4-3. Input Signal Path Configuration *(Cont.)*

Register Address	Bit	Label	Default	Description
R788 (0x0314) IN1R_Control	15	IN1R_HPF	0	Input Path 1 (Right) HPF Enable 0 = Disabled 1 = Enabled
	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)
R790 (0x0316) DMIC1R_Control	5:0	IN1R_DMIC_DLY[5:0]	0x00	Input Path 1 (Right) Digital Delay (applicable to digital input only) LSB = 1 sample, Range is 0 to 63. (Sample rate is controlled by IN1_OSR.)
R792 (0x0318) IN2L_Control	15	IN2L_HPF	0	Input Path 2 (Left) HPF Enable 0 = Disabled 1 = Enabled
	12:11	IN2_DMIC_SUP[1:0]	00	Input Path 2 DMIC Reference Select (sets the DMICDAT2 and DMICCLK2 logic levels) 00 = MICVDD 10 = MICBIAS2A 01 = MICBIAS1B 11 = MICBIAS2B
	10	IN2_MODE	00	Input Path 2 Mode 0 = Analog input 1 = Digital input
	7:1	IN2L_PGA_VOL[6:0]	0x40	Input Path 2 (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
R793 (0x0319) ADC_Digital_Volume_2L	14:13	IN2L_SRC[1:0]	00	Input Path 2 (Left) Source 00 = Differential (IN2LP–IN2LN) 10 = Reserved 01 = Single-ended (IN2LP) 11 = Reserved
R794 (0x031A) DMIC2L_Control	10:8	IN2_OSR[2:0]	101	Input Path 2 DMIC Oversample Rate (applicable to digital inputs only). 000 = Reserved 011 = 768 kHz 110 = 6.144 MHz 001 = Reserved 100 = 1.536 MHz 111 = Reserved 010 = 384 kHz 101 = 3.072 MHz If IN2_OSR=010 or 011, the maximum Input Path sample rate (all input paths) is 48 kHz or 96 kHz respectively.
	5:0	IN2L_DMIC_DLY[5:0]	0x00	Input Path 2 (Left) Digital Delay (applicable to digital input only) LSB = 1 sample, Range is 0 to 63. (Sample rate is controlled by IN2_OSR.)
R796 (0x031C) IN2R_Control	15	IN2R_HPF	0	Input Path 2 (Right) HPF Enable 0 = Disabled 1 = Enabled
	7:1	IN2R_PGA_VOL[6:0]	0x40	Input Path 2 (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
R797 (0x0319) ADC_Digital_Volume_2R	14:13	IN2R_SRC[1:0]	00	Input Path 2 (Right) Source 00 = Differential (IN2RP–IN2RN) 10 = Reserved 01 = Single-ended (IN2ARP) 11 = Reserved
R798 (0x031E) DMIC2R_Control	5:0	IN2R_DMIC_DLY[5:0]	0x00	Input Path 2 (Right) Digital Delay (applicable to digital input only) LSB = 1 sample, Range is 0 to 63. (Sample rate is controlled by IN2_OSR.)

4.2.7 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 CS47L35 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 CS47L35 digital core functions.

The digital volume control registers are described in Table 4-4 and Table 4-5.

Table 4-4. 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 4-5 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 4-5 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 4-4. Input Signal Path Digital Volume Control *(Cont.)*

Register Address	Bit	Label	Default	Description
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 4-5 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 4-5 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 4-5 lists the input signal path digital volume settings.

Table 4-5. 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

Table 4-5. Input Signal Path Digital Volume Range *(Cont.)*

Input Volume Register	Volume (dB)	Input Volume Register	Volume (dB)	Input Volume Register	Volume (dB)	Input Volume Register	Volume (dB)
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		

4.2.8 DMIC Pin Configuration

DMIC operation on input paths IN1 and IN2 is selected using IN*n*_MODE, as described in Table 4-3. If DMIC is selected, the respective DMICCLK*n* and DMICDAT*n* pins are configured as digital outputs and inputs, respectively.

The CS47L35 provides integrated pull-down resistors on each of the DMICDAT*n* pins. This provides a flexible capability for interfacing with other devices.

The DMICDAT1 and DMICDAT2 pull-down resistors can be configured independently using the bits described in Table 4-6. Note that, if the DMICDAT*n* DMIC input paths are disabled, the pull-down is disabled on the respective pin.

Table 4-6. DMIC Interface Pull-Down Control

Register Address	Bit	Label	Default	Description
R840 (0x0348) Dig_Mic_Pad_Ctrl	1	DMICDAT2_PD	0	DMICDAT2 Pull-Down Control 0 = Disabled 1 = Enabled
	0	DMICDAT1_PD	0	DMICDAT1 Pull-Down Control 0 = Disabled 1 = Enabled

4.3 Digital Core

The CS47L35 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 CS47L35 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) paths, output (DAC) paths, Digital Audio Interfaces (AIF1–AIF3) and SLIMbus paths 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 CS47L35 each time the device is powered up.

The procedure for configuring the CS47L35 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 CS47L35 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. 4-11.

The control registers associated with the digital-core signal paths are shown in Fig. 4-12 through Fig. 4-27. The full list of digital mixer control registers (R1600–R2936) is provided in Section 6. Generic register field definitions are provided in Table 4-7.

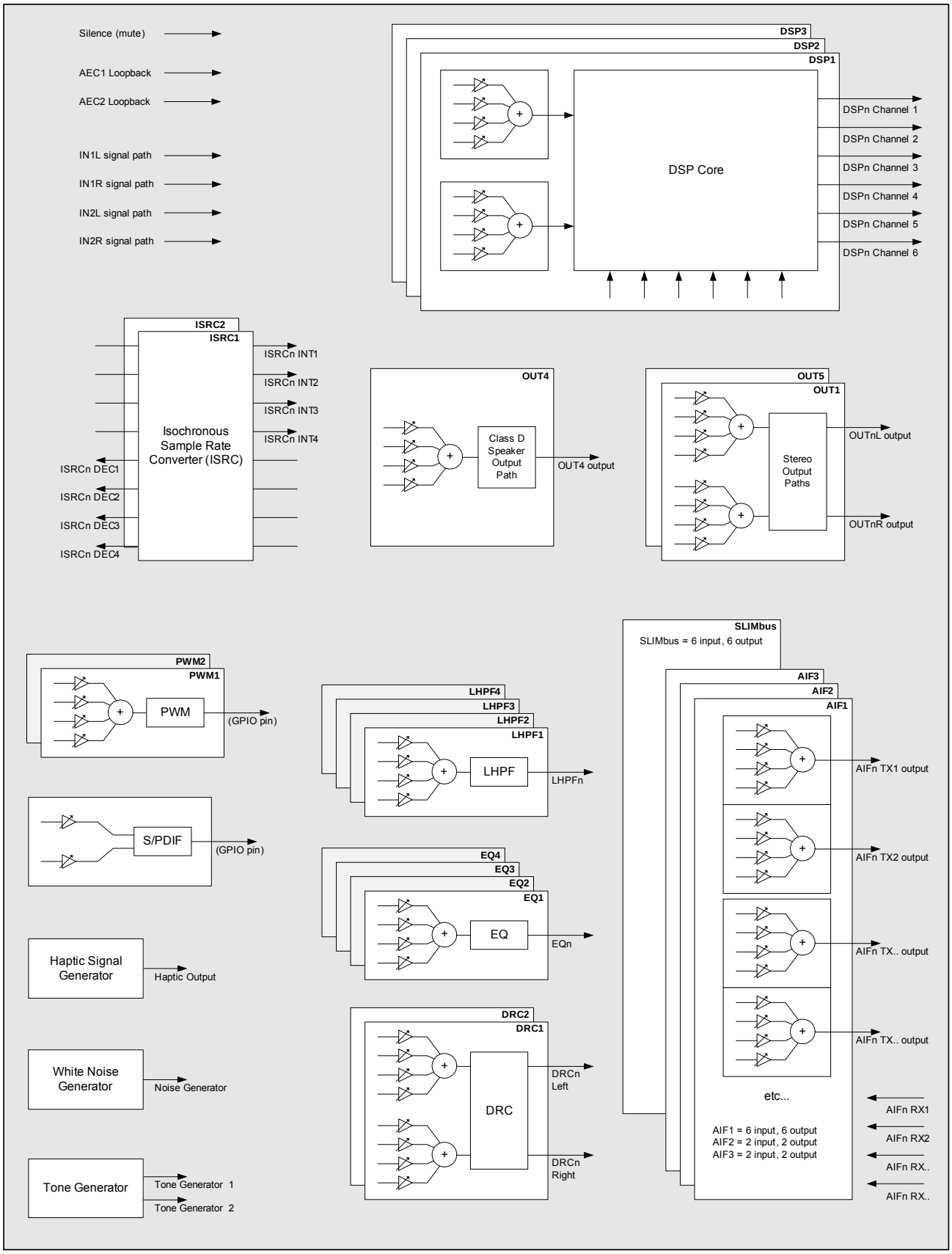


Figure 4-11. Digital Core

4.3.1 Digital-Core Mixers

The CS47L35 provides an extensive digital mixing capability. The digital-core mixing and signal-processing blocks are shown in Fig. 4-11. 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. 4-12–Fig. 4-27. 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 4-7.

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.

The x_SRC*n* fields for all digital core functions should be held at 0x00 if SYSCLK is not enabled—SYSCLK must be present and enabled before selecting other values for these fields. See Section 4.16.4 for further details (including requirements for reconfiguring SYSCLK while digital core mixers are enabled).

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 4-7.

Table 4-7. 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) 0x3C = SLIMbus RX5 0x75 = DSP2 Channel 6 0x04 = Tone generator 1 0x3D = SLIMbus RX6 0x78 = DSP3 Channel 1 0x05 = Tone generator 2 0x50 = EQ1 0x79 = DSP3 Channel 2 0x06 = Haptic generator 0x51 = EQ2 0x7A = DSP3 Channel 3 0x08 = AEC Loop-Back 1 0x52 = EQ3 0x7B = DSP3 Channel 4 0x09 = AEC Loop-Back 2 0x53 = EQ4 0x7C = DSP3 Channel 5 0x0D = Noise generator 0x58 = DRC1 Left 0x7D = DSP3 Channel 6 0x10 = IN1L signal path 0x59 = DRC1 Right 0xA0 = ISRC1 INT1 0x11 = IN1R signal path 0x5A = DRC2 Left 0xA1 = ISRC1 INT2 0x12 = IN2L signal path 0x5B = DRC2 Right 0xA2 = ISRC1 INT3 0x13 = IN2R signal path 0x60 = LHPF1 0xA3 = ISRC1 INT4 0x20 = AIF1 RX1 0x61 = LHPF2 0xA4 = ISRC1 DEC1 0x21 = AIF1 RX2 0x62 = LHPF3 0xA5 = ISRC1 DEC2 0x22 = AIF1 RX3 0x63 = LHPF4 0xA6 = ISRC1 DEC3 0x23 = AIF1 RX4 0x68 = DSP1 Channel 1 0xA7 = ISRC1 DEC4 0x24 = AIF1 RX5 0x69 = DSP1 Channel 2 0xA8 = ISRC2 INT1 0x25 = AIF1 RX6 0x6A = DSP1 Channel 3 0xA9 = ISRC2 INT2 0x28 = AIF2 RX1 0x6B = DSP1 Channel 4 0xAA = ISRC2 INT3 0x29 = AIF2 RX2 0x6C = DSP1 Channel 5 0xAB = ISRC2 INT4 0x30 = AIF3 RX1 0x6D = DSP1 Channel 6 0xAC = ISRC2 DEC1 0x31 = AIF3 RX2 0x70 = DSP2 Channel 1 0xAD = ISRC2 DEC2 0x38 = SLIMbus RX1 0x71 = DSP2 Channel 2 0xAE = ISRC2 DEC3 0x39 = SLIMbus RX2 0x72 = DSP2 Channel 3 0xAF = ISRC2 DEC4 0x3A = SLIMbus RX3 0x73 = DSP2 Channel 4 0x3B = SLIMbus RX4 0x74 = DSP2 Channel 5

4.3.2 Digital-Core Inputs

The digital core comprises multiple input paths, as shown in Fig. 4-12. Any of these inputs may be selected as a source to the digital mixers or signal-processing functions within the CS47L35 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 CS47L35 digital core. Those input sources, which are not shown in Fig. 4-12, are described separately throughout Section 4.3.

The hexadecimal numbers in Fig. 4-12 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, AIF*n*_RATE, or SLIMRX*n*_RATE field; see Table 4-21. 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.

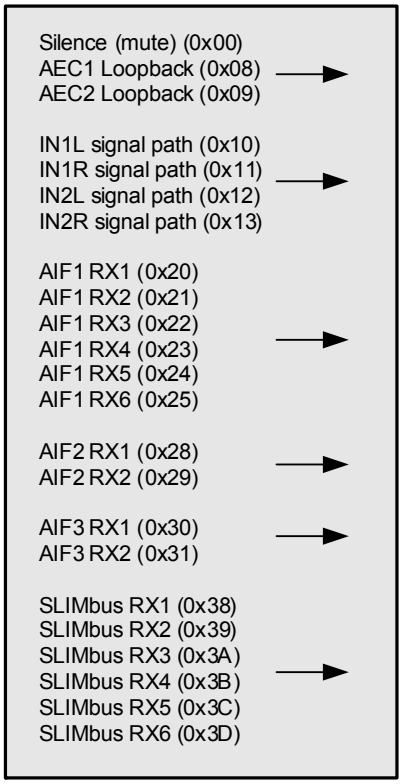


Figure 4-12. Digital-Core Inputs

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. 4-13. The output paths associated with OUT1, OUT4, and OUT5 are shown in Fig. 4-14. The output paths associated with the SLIMbus interface are shown in Fig. 4-15.

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. 4-13) are located at register addresses R1792–R1935 (0x0700–0x078F). The OUT1, OUT4, and OUT5 output mixer control fields (see Fig. 4-14) are located at addresses R1664–R1743 (0x0680–0x06CF). The SLIMbus output mixer control fields (see Fig. 4-15) are located at addresses R1984–R2031 (0x07C0–0x07EF).

The full list of digital mixer control registers (R1600–R2936) is provided in Section 6. Generic register field definitions are provided in Table 4-7.

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 x_SRC*n* fields for all digital core functions should be held at 0x00 if SYSCLK is not enabled—SYSCLK must be present and enabled before selecting other values for these fields. See Section 4.16.4 for further details (including requirements for reconfiguring SYSCLK while digital core mixers are enabled).

The sample rate for the output signal paths is configured using the applicable OUT_RATE, AIF*n*_RATE, or SLIMTX*n*_RATE fields; see Table 4-21. 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, AIF*n*_RATE, or SLIMTX*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, AIF*n*_RATE, or SLIMTX*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, AIF*n*_RATE, or SLIMTX*n*_RATE fields. See Table 4-21 for details.

The CS47L35 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.

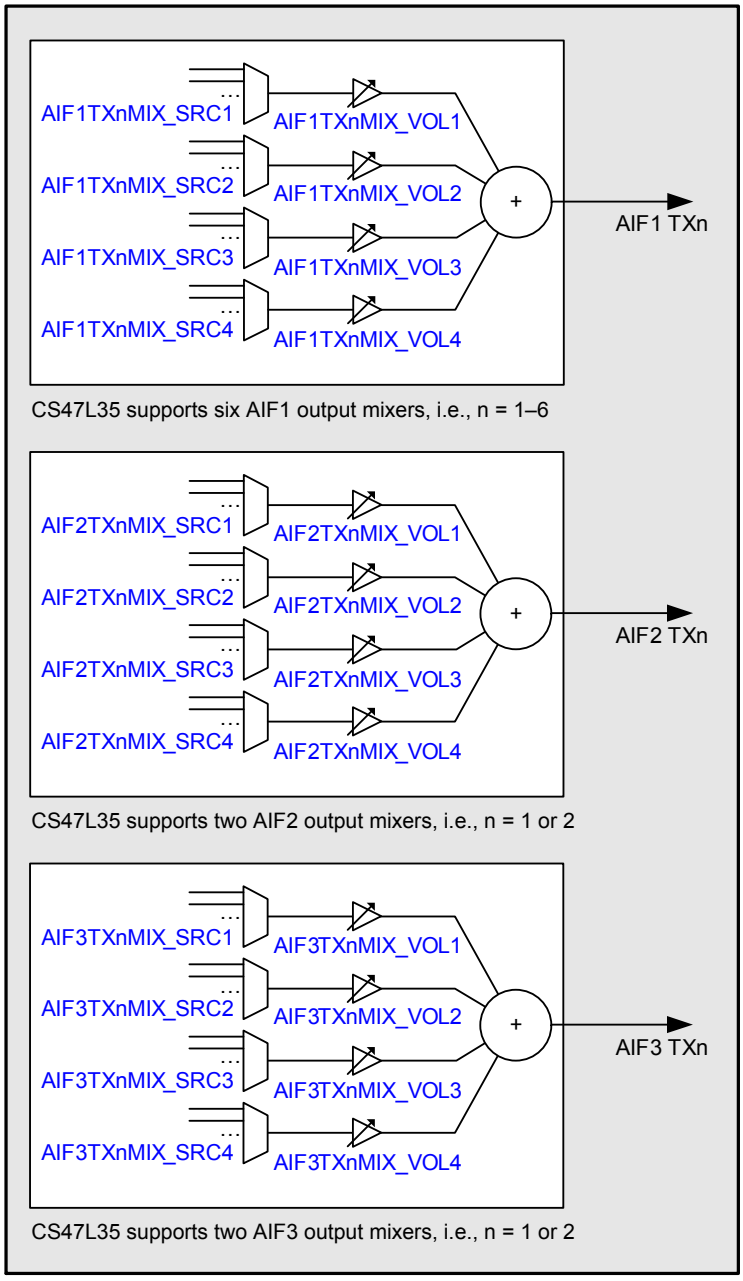


Figure 4-13. Digital-Core AIF Outputs

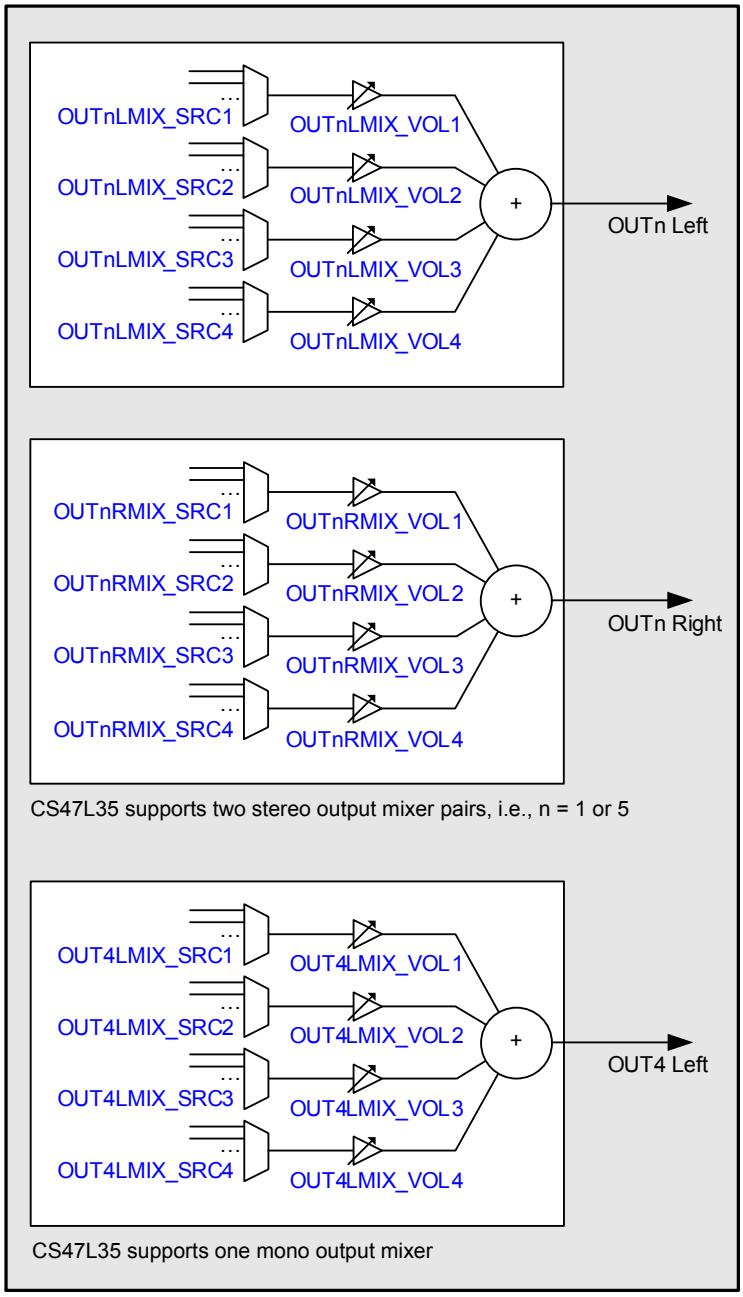


Figure 4-14. Digital-Core OUT*n* Outputs

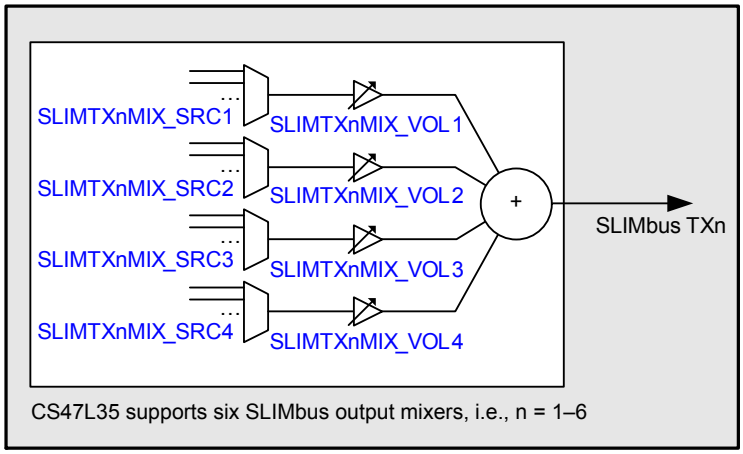


Figure 4-15. Digital-Core SLIMbus Outputs

4.3.4 Five-Band Parametric Equalizer (EQ)

The digital core provides four EQ processing blocks as shown in Fig. 4-16. 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.

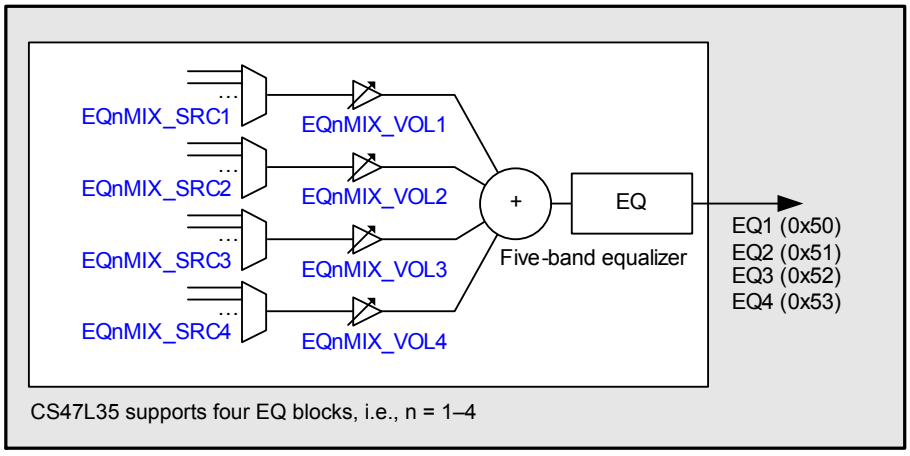


Figure 4-16. Digital-Core EQ Blocks

The EQ1–EQ4 mixer control fields (see Fig. 4-16) 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 4-7.

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. 4-16 indicate the corresponding x_SRC*n* setting for selection of that signal as an input to another digital-core function.

The EQ blocks should be kept disabled (EQ*n*_ENA = 0) if SYSCLK is not enabled. The x_SRC*n* fields for all digital core functions should be held at 0x00 if SYSCLK is not enabled. SYSCLK must be present and enabled before selecting other values for these fields. See Section 4.16.4 for further details (including requirements for reconfiguring SYSCLK while digital core functions are enabled).

The sample rate for the EQ function is configured using FX_RATE; see Table 4-21. 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 4-21 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 4-8. 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 4-8. 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 4-9.

Table 4-9. 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.

Table 4-9. EQ Enable and Gain Control *(Cont.)*

Register Address	Bit	Label	Default	Description
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
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 4-10 for gain range.

Table 4-10 lists the EQ gain control settings.

Table 4-10. 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 CS47L35 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. 4-17. 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.

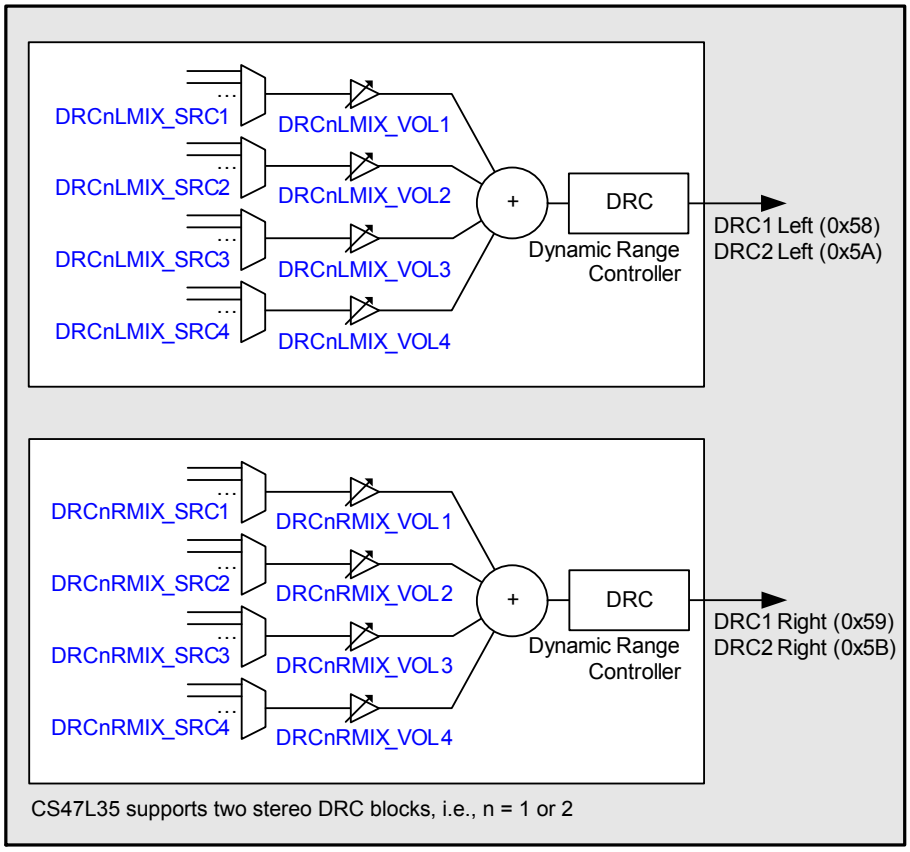


Figure 4-17. Dynamic Range Control (DRC) Block

The DRC1 and DRC2 mixer control fields (see Fig. 4-17) 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 4-7.

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. 4-17 indicate the corresponding x_SRC*n* setting for selection of that signal as an input to another digital-core function.

The DRC blocks should be kept disabled (DRC*nx*_ENA = 0) if SYSCLK is not enabled. The x_SRC*n* fields for all digital core functions should be held at 0x00 if SYSCLK is not enabled. SYSCLK must be present and enabled before selecting other values for these fields. See Section 4.16.4 for further details (including requirements for reconfiguring SYSCLK while digital core functions are enabled).

The sample rate for the DRC function is configured using FX_RATE; see Table 4-21. 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 4-21 for details.

The DRC functions are enabled using the control registers described in Table 4-11.

Table 4-11. 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 4-13 and Table 4-14 for DRC1 and DRC2.

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. 4-18). 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. 4-18.

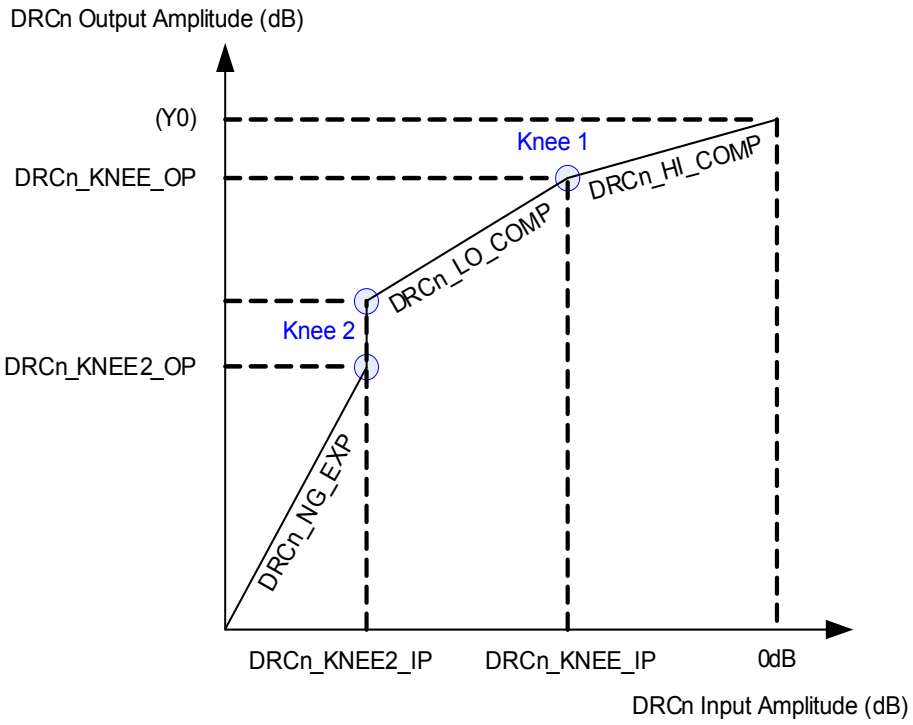


Figure 4-18. DRC Response Characteristic

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. 4-18), 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 4-12.

Table 4-12. 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 4-12) 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. 4-18 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. 4-18. 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 4-13. 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 4.15.

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 4.18.

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 4-13.

Table 4-13. 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 4-13. 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 4-14.

Table 4-14. 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 4-14. 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 0001 = –30 dB 0010 = –24 dB 0011 = –18 dB 0100 = –12 dB 0101 = –6 dB 0110 = 0 dB 0111 = 6 dB 1000 = 12 dB 1001 = 18 dB 1010 = 24 dB 1011 = 30 dB 1100 = 36 dB 1101 to 1111 = Reserved
	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) 001 = 1/2 010 = 1/4 011 = 1/8 100 = 1/16 101 = 0 110–111 = Reserved
	2:0	DRC2_LO_ COMP[2:0]	000	DRC2 Compressor slope (lower region) 000 = 1 (no compression) 001 = 1/2 010 = 1/4 011 = 1/8 100 = 0 101–11X = Reserved
R3723 (0x0E8B) DRC2_ctrl4	10:5	DRC2_KNEE_ IP[5:0]	0x00	DRC2 Input signal level at the compressor knee. 0x00 = 0 dB 0x01 = –0.75 dB 0x02 = –1.5 dB … (–0.75-dB steps) 0x3C = –45 dB 0x3D–-0x3F = Reserved
	4:0	DRC2_KNEE_ OP[4:0]	0x00	DRC2 Output signal at the compressor knee. 0x00 = 0 dB 0x01 = –0.75 dB 0x02 = –1.5 dB … (–0.75 dB steps) 0x1E = –22.5 dB 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 0x01 = –37.5 dB 0x02 = –39 dB … (-1.5-dB steps) 0x1E = –81 dB 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 0x01 = –31.5 dB 0x02 = –33 dB … (–1.5dB steps) 0x1E = –75 dB 0x1F = –76.5 dB Applicable only if DRC2_KNEE2_OP_ENA = 1.

The CS47L35 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. 4-19. 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).

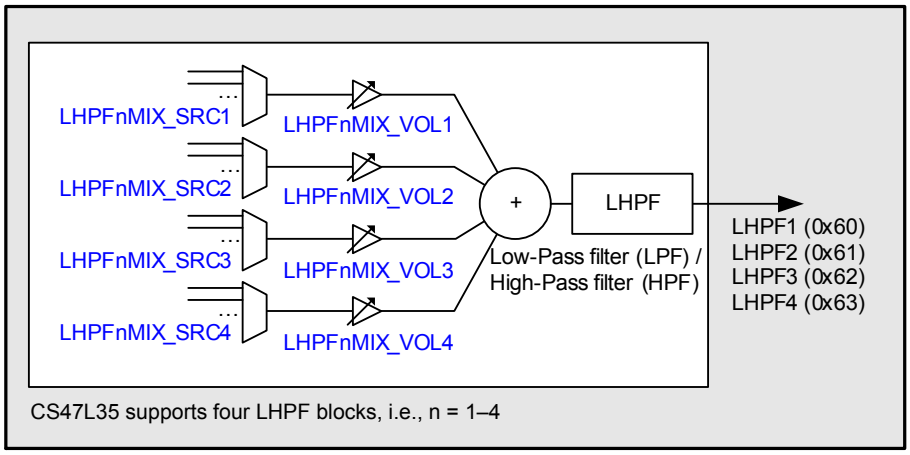


Figure 4-19. Digital-Core LPF/HPF Blocks

The LHPF1–LHPF4 mixer control fields, shown in Fig. 4-19, 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 4-7.

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

The LHPF blocks should be kept disabled (LHPF*n*_ENA = 0) if SYSCLK is not enabled. The x_SRC*n* fields for all digital core functions should be held at 0x00 if SYSCLK is not enabled. SYSCLK must be present and enabled before selecting other values for these fields. See Section 4.16.4 for further details (including requirements for reconfiguring SYSCLK while digital core functions are enabled).

The sample rate for the LHPF function is configured using FX_RATE; see Table 4-21. 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 4-21 for details.

The control registers associated with the LHPF functions are described in Table 4-15.

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 4-15. 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 CS47L35 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 three programmable DSP processing blocks as shown in Fig. 4-20. Each 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. Each DSP block supports six outputs.

The functionality of the DSP processing blocks is not fixed, and a wide range of audio enhancements algorithms may be performed. The procedure for configuring the CS47L35 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 4.4.2.

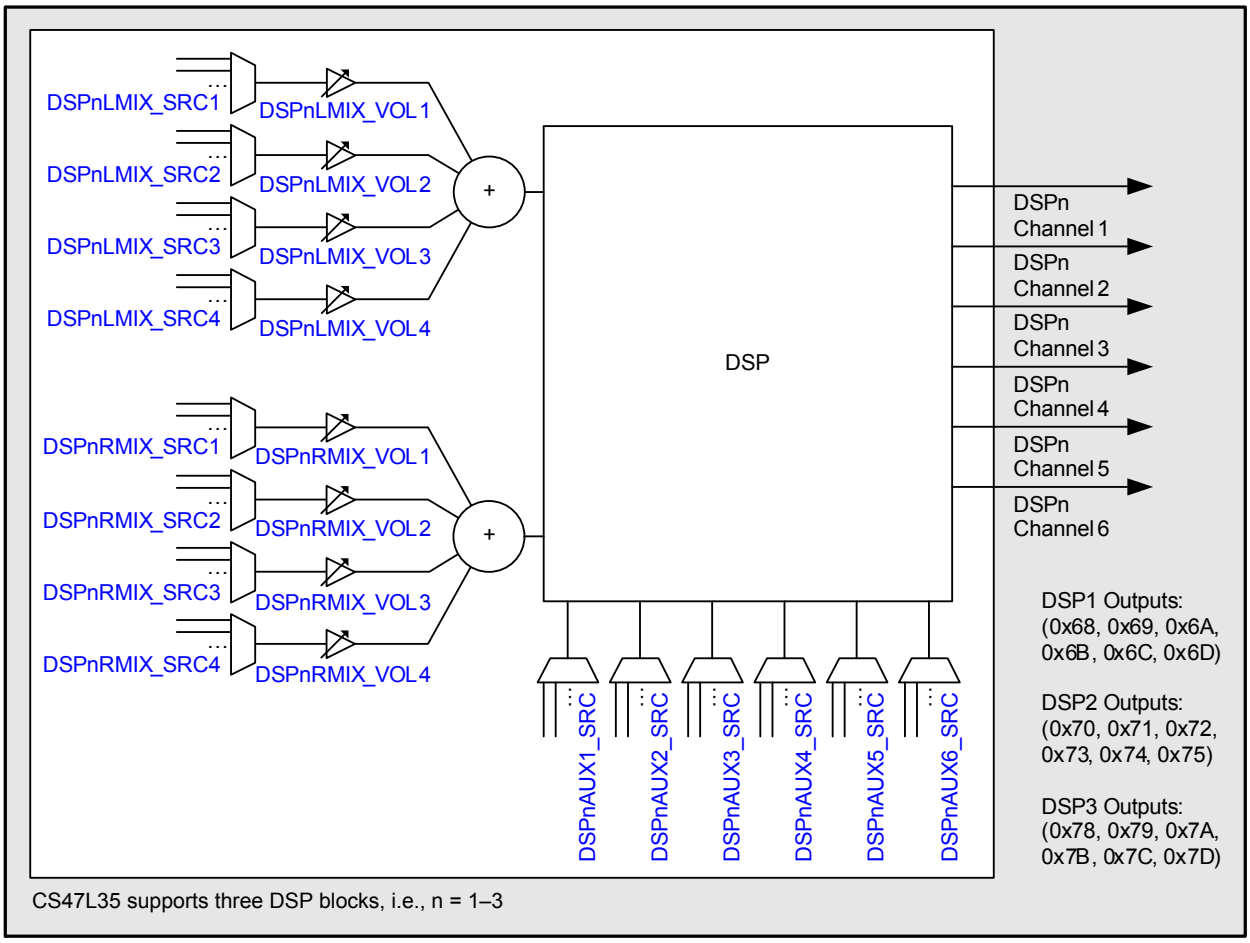


Figure 4-20. Digital-Core DSP Blocks

The DSP*n* mixer input control fields (see Fig. 4-20) are located at register addresses R2368–R2511 (0x0940–0x09CF).

The full list of digital mixer control registers (R1600–R2936) is provided in Section 6. Generic register field definitions are provided in Table 4-7.

The x_SRC*n* fields select the input sources for the respective DSP processing blocks. Note that the selected input sources must be configured for the same sample rate as the DSP 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. 4-20 indicate the corresponding x_SRC*n* setting for selection of that signal as an input to another digital-core function.

The x_SRC*n* fields for all digital core functions should be held at 0x00 if SYSCLK is not enabled—SYSCLK must be present and enabled before selecting other values for these fields. See Section 4.16.4 for further details (including requirements for reconfiguring SYSCLK while digital core mixers are enabled).

The sample rate for each of the DSP functions is configured using the respective DSP*n*_RATE field; see Table 4-21. Sample-rate conversion is required when routing the DSP*n* signal paths to any signal chain that is configured for a different sample rate.

The DSP*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 DSP*n*_RATE. A minimum delay of 125 µs must be allowed between clearing the x_SRC*n* fields and writing to the associated DSP*n*_RATE fields. See Table 4-21 for details.

The CS47L35 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 CS47L35 incorporates an IEC-60958-3–compatible S/PDIF output generator, as shown in Fig. 4-21; 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 4.14 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 CS47L35 digital core.

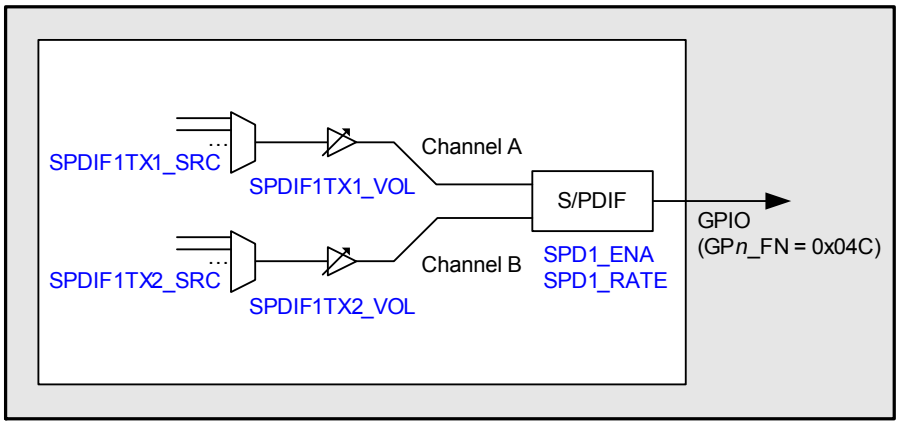


Figure 4-21. Digital-Core S/PDIF Output Generator

The S/PDIF input control fields (see Fig. 4-21) 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 4-7.

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 S/PDIF output generator should be kept disabled (SPD1_ENA = 0) if SYSCLK is not enabled. The x_SRC*n* fields for all digital core functions should be held at 0x00 if SYSCLK is not enabled. SYSCLK must be present and enabled before selecting other values for these fields. See Section 4.16.4 for further details (including requirements for reconfiguring SYSCLK while digital core functions are enabled).

The sample rate of the S/PDIF generator is configured using SPD1_RATE; see Table 4-21. 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 4-21 for details.

The S/PDIF generator is enabled by setting SPD1_ENA, as described in Table 4-16.

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 4-16. 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 CS47L35 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 CS47L35 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.

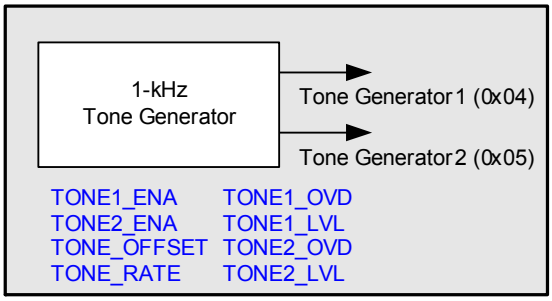


Figure 4-22. Digital-Core Tone Generator

The tone generators can be selected as input to any of the digital mixers or signal-processing functions within the CS47L35 digital core. 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 tone generators is configured using TONE_RATE. See Table 4-21. 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 generators are enabled by setting the TONE1_ENA and TONE2_ENA bits as described in Table 4-17. The phase relationship is configured using TONE_OFFSET.

The tone generators 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 4-17.

SYSCLK must be present and enabled before setting the TONE*n*_ENA bits. The tone generators should be kept disabled (TONE*n*_ENA = 0) if SYSCLK is not enabled. See Section 4.16.4 for further details (including requirements for reconfiguring SYSCLK while digital core functions are enabled).

Table 4-17. 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).

Table 4-17. Tone Generator Control *(Cont.)*

Register Address	Bit	Label	Default	Description
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 CS47L35 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.

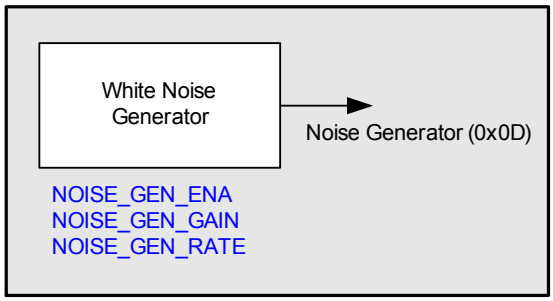


Figure 4-23. Digital-Core Noise Generator

The noise generator can be selected as input to any of the digital mixers or signal-processing functions within the CS47L35 digital core. The hexadecimal number (0x0D) in Fig. 4-23 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 4-21. 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 4-18. The signal level is configured using NOISE_GEN_GAIN.

SYSCLK must be present and enabled before setting NOISE_GEN_ENA. The noise generator should be kept disabled (NOISE_GEN_ENA = 0) if SYSCLK is not enabled. See Section 4.16.4 for further details (including requirements for reconfiguring SYSCLK while digital core functions are enabled).

Table 4-18. 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 CS47L35 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 CS47L35. 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. 4-24. Note that the digital PDM output paths may also be used for haptic signal output.

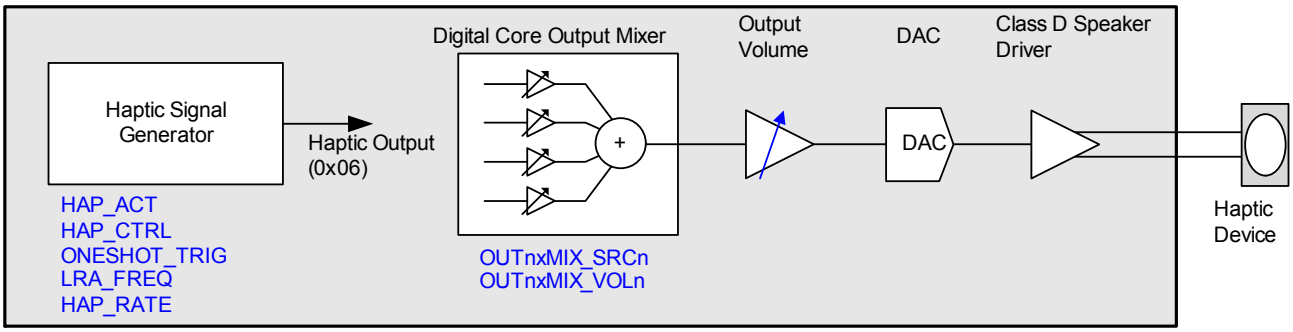


Figure 4-24. Digital-Core Haptic Signal Generator

The hexadecimal number (0x06) in Fig. 4-24 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.

SYSCLK must be present and enabled before setting HAP_CTRL > 00. The haptic signal generator should be kept disabled (HAP_CTRL = 00) if SYSCLK is not enabled. See Section 4.16.4 for further details (including requirements for reconfiguring SYSCLK while digital core functions are enabled).

The sample rate for the haptic signal generator is configured using the HAP_RATE field. See Table 4-19. 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 4-19. 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 4-19. 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
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 CS47L35 incorporates two PWM signal generators as shown in Fig. 4-25. 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 4.14 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 CS47L35 digital core.

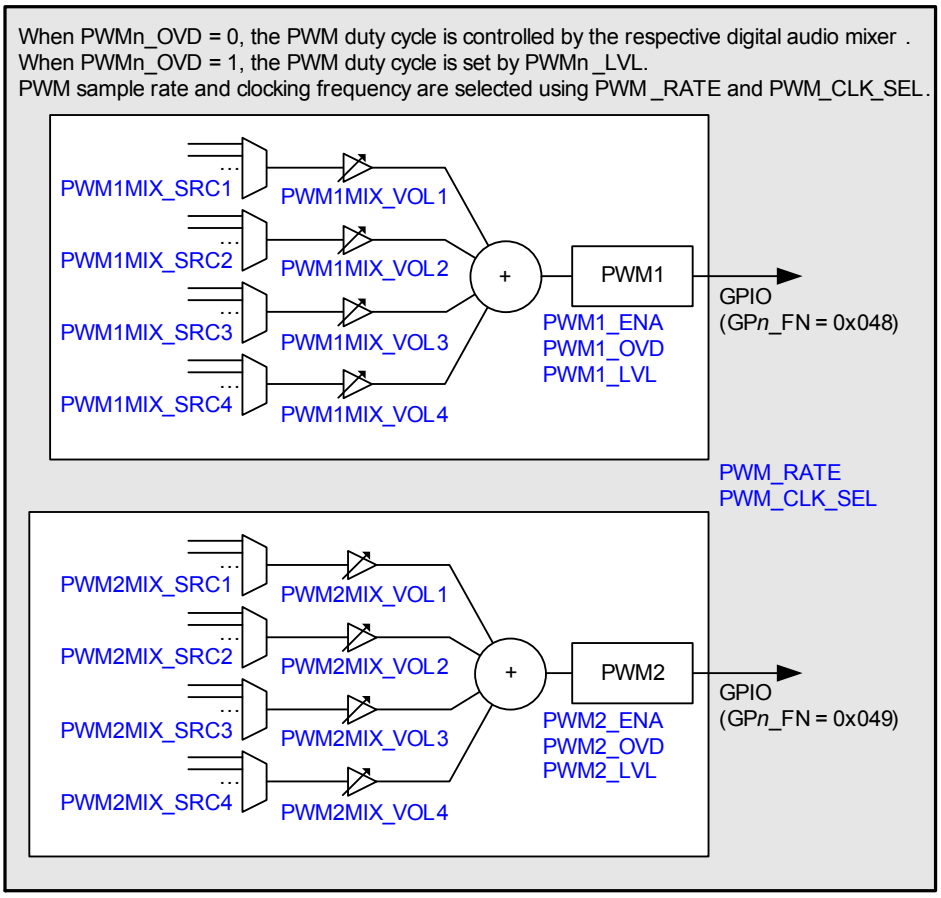


Figure 4-25. Digital-Core PWM Generator

The PWM1 and PWM2 mixer control fields (see Fig. 4-25) 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 4-7.

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 generators should be kept disabled (PWM*n*_ENA = 0) if SYSCLK is not enabled. The x_SRC*n* fields for all digital core functions should be held at 0x00 if SYSCLK is not enabled. SYSCLK must be present and enabled before selecting other values for these fields. See Section 4.16.4 for further details (including requirements for reconfiguring SYSCLK while digital core functions are enabled).

The PWM sample rate (cycle time) is configured using PWM_RATE. See Table 4-21. 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 4-21 for details.

The PWM generators are enabled by setting PWM1_ENA and PWM2_ENA, respectively, as described in Table 4-20.

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. 4-25.

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 4-20. 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 CS47L35 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 CS47L35 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 4.16. Every digital signal path must be synchronized to SYSCLK.

Up to three different sample rates may be in use at any time on the CS47L35; 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 CS47L35 digital core are controlled as shown in Fig. 4-26. 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 4-21 for details.

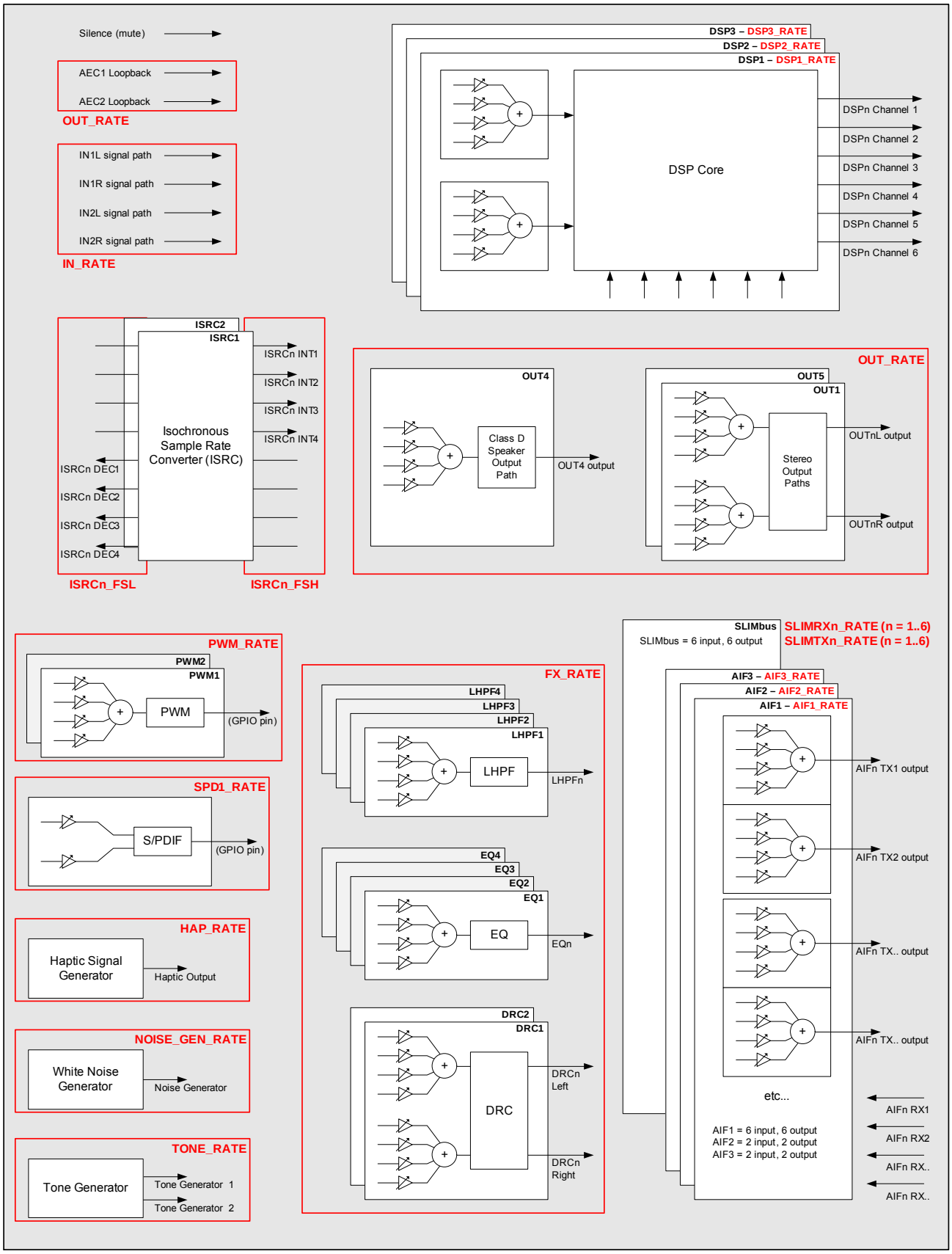


Figure 4-26. Digital-Core Sample-Rate Control

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 SLIMbus interface supports up to six input channels and six output channels. The sample rate of each channel can be configured independently, using SLIMTX*n*_RATE and SLIMRX*n*_RATE.

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*n* functions can be enabled in any signal path within the digital core. The applicable sample rates are configured using the DSP*n*_RATE fields (where *n* identifies the applicable DSP block, 1 to 3) respectively.

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 4-21. 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 4-22.

Note that 32-bit register addressing is used from R12888 (0x3000) upwards; 16-bit format is used otherwise. The registers noted in Table 4-21 contain a mixture of 16-bit and 32-bit register addresses.

Table 4-21. 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 PWMnMIX_SRCm 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 4-21. 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 8kHz to 192kHz. If 384 kHz/768 kHz DMIC rate is selected on any of the input paths (INn_ 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*n*xMIX_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.
R1509 (0x05E5) SLIMbus_Rates_1	14:11	SLIMRX2_ RATE[3:0]	0000	SLIMbus RX Channel *n* 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.
	6:3	SLIMRX1_ RATE[3:0]	0000	
R1510 (0x05E6) SLIMbus_Rates_2	14:11	SLIMRX4_ RATE[3:0]	0000	
	6:3	SLIMRX3_ RATE[3:0]	0000	
R1511 (0x05E7) SLIMbus_Rates_3	14:11	SLIMRX6_ RATE[3:0]	0000	
	6:3	SLIMRX5_ RATE[3:0]	0000	
R1513 (0x05E9) SLIMbus_Rates_5	14:11	SLIMTX2_ RATE[3:0]	0000	SLIMbus TX Channel *n* 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 SLIMTX*n*MIX_SRC*m* fields must be cleared before changing SLIMTX*n*_RATE.
	6:3	SLIMTX1_ RATE[3:0]	0000	
R1514 (0x05EA) SLIMbus_Rates_6	14:11	SLIMTX4_ RATE[3:0]	0000	
	6:3	SLIMTX3_ RATE[3:0]	0000	
R1515 (0x05EB) SLIMbus_Rates_7	14:11	SLIMTX6_ RATE[3:0]	0000	
	6:3	SLIMTX5_ RATE[3:0]	0000	

Table 4-21. Digital-Core Sample-Rate Control *(Cont.)*

Register Address	Bit	Label	Default	Description
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 DSP*n*xMIX_SRC*m* fields must be cleared before changing DSP*n*_RATE.
R1572352 (0x17_FE00) DSP2_Config_1	14:11	DSP2_RATE[3:0]	0000	
R2096640 (0x1F_FE00) DSP3_Config_1	14:11	DSP3_RATE[3:0]	0000	

4.3.14 Isochronous Sample-Rate Converter (ISRC)

The CS47L35 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 CS47L35. Each ISRC provides four signal paths between two different sample rates, as shown in Fig. 4-27. 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 4.16 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 4-22 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 CS47L35 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. 4-27.

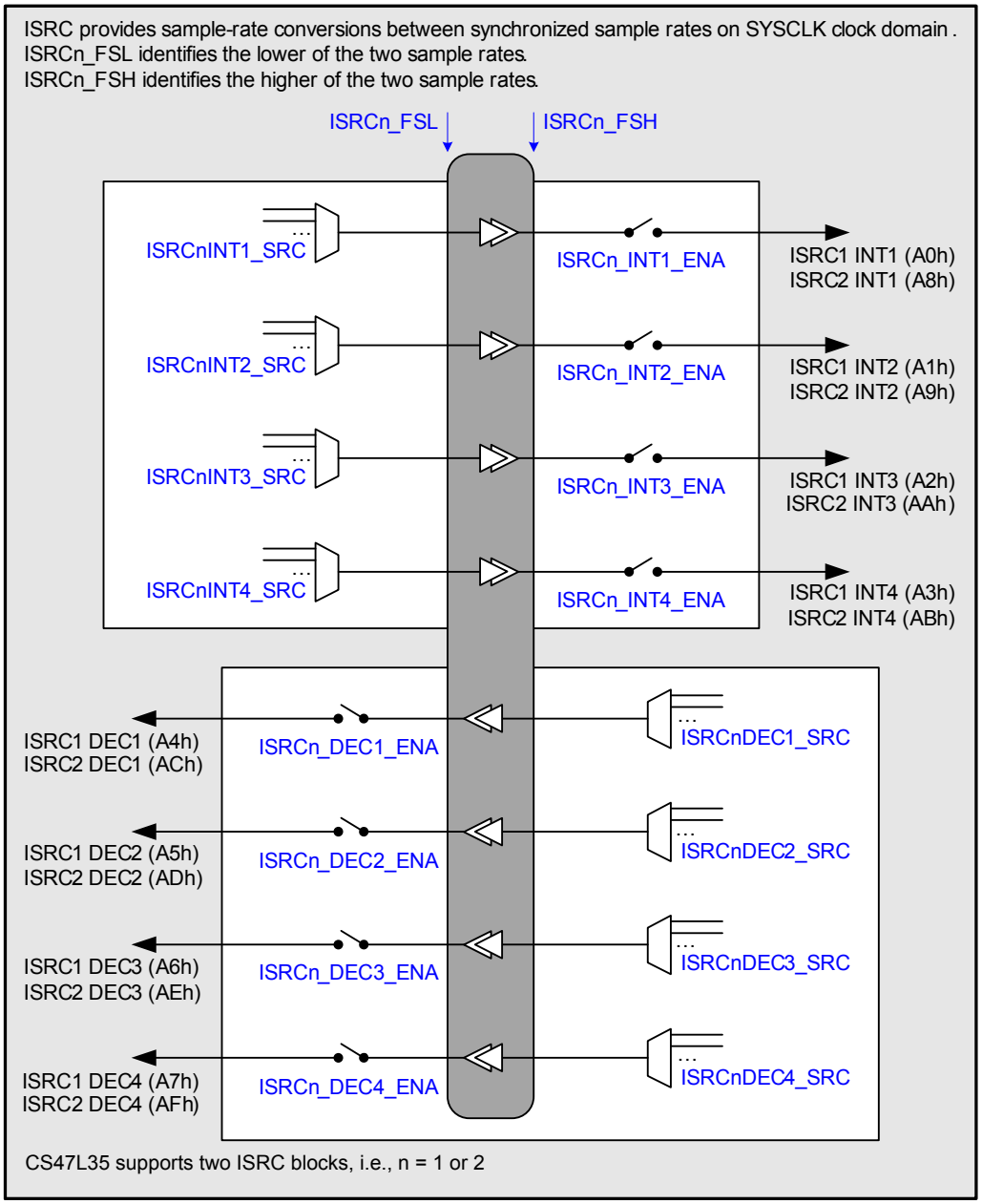


Figure 4-27. Isochronous Sample-Rate Converters (ISRCs)

The ISRC input control fields (see Fig. 4-27) 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 4-7.

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. 4-27 indicate the corresponding x_SRC setting for selection of that signal as an input to another digital-core function.

The ISRC paths should be kept disabled (ISRC*n*_INT*m*_ENA = 0, ISRC*n*_DEC*m*_ENA = 0) if SYSCLK is not enabled. The x_SRC*n* fields for all digital core functions should be held at 0x00 if SYSCLK is not enabled. SYSCLK must be present and enabled before selecting other values for these fields. See Section 4.16.4 for further details (including requirements for reconfiguring SYSCLK while digital core functions are enabled).

The register bits associated with the ISRCs are described in Table 4-22.

Table 4-22. 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_DECn_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_INTn_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_DECn_SRC fields must be cleared before changing ISRC2_FSH.

Table 4-22. 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_INTn_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 CS47L35 digital core incorporates three DSP processing blocks, capable of running a wide range of audio-enhancement functions. Different firmware configurations can be loaded onto each DSP, enabling the CS47L35 to be highly customized for specific application requirements. Full read/write access to the device register map is supported from each DSP core, including access to the firmware registers of the other DSPs. Synchronization of different DSPs is supported, and shared data memory space is provided for the DSP2 and DSP3 blocks; these features enable enhanced processing capabilities for the associated DSPs.

Examples of the DSP functions include multiband compressor (MBC), and the SoundClear™ suite of audio processing algorithms. Note that it is possible to implement more than one type of audio enhancement function on a single DSP; the precise combinations of functions vary from one firmware configuration to another.

The DSP blocks each employ the same internal architecture and provide an equivalent processing capability. Note that the DSPs differ in terms of the firmware memory sizes associated with each. The DSPs can be clocked at up to 150MHz, corresponding to 150 MIPS each.

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 DSP blocks, the required firmware configuration must first be loaded onto the device by writing the appropriate files to the CS47L35 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 CS47L35. 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 blocks 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 4-23. Note that clocking is not required for access to the firmware registers by the host processor.

The CS47L35 program, data, and coefficient register memory space is described in Table 4-23. The full register map listing is provided in Section 6. The shared DSP2/DSP3 memory space is implemented at two different register address locations; reading or writing at either address accesses the same memory data.

If multiple DSPs write to a shared-memory address at the same time, the address at which the collision occurred is reported in the DSP2_DUALMEM_COLLISION_ADDR and DSP3_DUALMEM_COLLISION_ADDR fields. Note that these fields are coded in 24-bit DSP data word units, and are defined relative to the base address of the shared-memory area. The DSP2_DUALMEM_COLLISION_ADDR and DSP3_DUALMEM_COLLISION_ADDR fields provide the same information.

The DSP memory controller provides an input to the interrupt control circuit. An interrupt event is triggered if a memory collision occurs. Note that the DSP software should be written to ensure this never happens; the interrupt is intended for development purposes only. See Section 4.15 for details of the interrupt-event handling.

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 4-23. DSP Program, Data, and Coefficient Registers

DSP Number	Description	Register Address	Number of Registers	DSP Memory Size
DSP1	Program memory	0x08_0000–0x08_5FFE	12288	8k x 40-bit words
	X-Data memory	0x0A_0000–0x0A_7FFE	16384	16k 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
DSP2	Program memory	0x10_0000–0x10_EFFE	30720	20k x 40-bit words
	X-Data memory	0x12_0000–0x12_BFFE	24576	24k x 24-bit words
	X-Data memory (Shared DSP2/DSP3)	0x13_6000–0x13_7FFE	4096	4k x 24-bit words
	Y-Data memory	0x14_0000–0x14_BFFE	24576	24k x 24-bit words
	Coefficient memory	0x16_0000–0x16_1FFE	4096	4k x 24-bit words
DSP3	Program memory	0x18_0000–0x18_EFFE	30720	20k x 40-bit words
	X-Data memory	0x1A_0000–0x1B_1FFE	36864	36k x 24-bit words
	X-Data memory (Shared DSP2/DSP3)	0x1B_6000–0x1B_7FFE	4096	4k x 24-bit words
	Y-Data memory	0x1C_0000–0x1C_BFFE	24576	24k x 24-bit words
	Coefficient memory	0x1E_0000–0x1E_1FFE	4096	4k x 24-bit words

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

The register window space is additional to the X-data memory sizes described in Table 4-23. Note that the X-memory addresses of these register windows are the same for all DSP cores, regardless of the different X-memory sizes.

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 4.4.6). 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 DSP*n*_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 DSP*n*_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 DSP*n*_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. 4-28. 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.

Note that SYSCLK must be present and enabled, if the DSP firmware requires read or write access to control registers below address 0x40000. See Section 4.16 for further details of the CS47L35 system clocks.

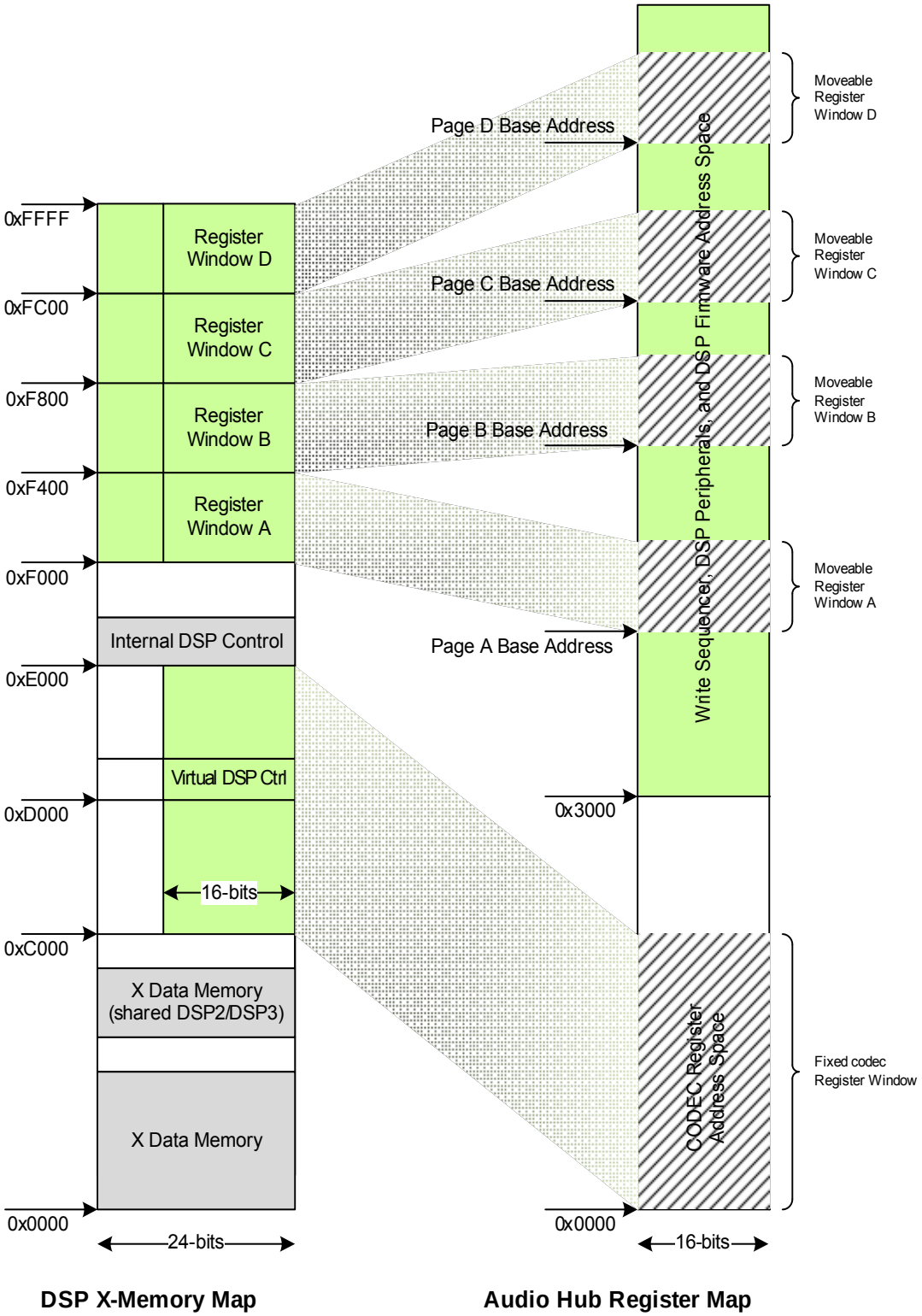


Figure 4-28. X-Data Memory Map

Note that the full CS47L35 register space is shown here as 16-bit width. (SPI/I²C/SLIMbus register access uses 32-bit data width at 0x3000 and above.) However, the window base address fields (DSP*n*_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 4-24.

Table 4-24. X-Data Memory and Clocking Control

Register Address	Bit	Label	Default	Description
DSP1 Base Address = R1048064 (0x0F_FE00) DSP2 Base Address = R1572352 (0x17_FE00) DSP3 Base Address = R2096640 (0x1F_FE00)				
Base address + 0x54 DSP*n*_Ext_window_A	31	DSP*n*_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	DSP*n*_EXT_A_PAGE[15:0]	0x0000	Sets the Base Address of Register Window A in X-memory. Coded as LSB = 512 (0x200)
Base address + 0x56 DSP*n*_Ext_window_B	31	DSP*n*_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	DSP*n*_EXT_B_PAGE[15:0]	0x0000	Sets the Base Address of Register Window B in X-memory. Coded as LSB = 512 (0x200)
Base address + 0x58 DSP*n*_Ext_window_C	31	DSP*n*_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	DSP*n*_EXT_C_PAGE[15:0]	0x0000	Sets the Base Address of Register Window C in X-memory. Coded as LSB = 512 (0x200)
Base address + 0x5A DSP*n*_Ext_window_D	31	DSP*n*_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	DSP*n*_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 Firmware Control

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

4.4.2.1 DSP Memory

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

The DSP*n*_MEM_ENA bits are not affected by software reset; these bits remain in their 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 for a summary of the CS47L35 reset behavior.

4.4.2.2 DSP Clocking

Clocking is required for each of the DSP processing blocks, 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 each DSP is enabled and disabled automatically, as required by the respective DSP core and DMA channel status.

The clock source for each DSP is derived from DSPCLK. See Section 4.16 for details of how to configure DSPCLK.

The clock frequency for each DSP is selected using the DSP*n*_CLK_SEL field (where *n* identifies the applicable DSP block, 1 to 3). The DSP clock frequency must be less than or equal to the DSPCLK frequency.

Note that the DSP*n*_CLK_SEL fields select a range of frequencies for each valid decode value. The clock frequency for each DSP is derived as DSPCLK divided by 1, 2, 4, 8, or 16. The required division ratios, within the selected DSP clock frequency ranges, are configured automatically for each DSP core.

The DSP*n*_CLK_SEL_STS fields indicate the clock frequency range for the respective DSP cores. These can be used to confirm the clock frequency, in cases where code execution has a minimum clock frequency requirement. The DSP*n*_CLK_SEL_STS field is only valid when the respective 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 the DSP*n*_CLK_SEL_STS fields indicate a range of frequencies for each decode value. The exact clock frequency for each DSP cannot be provided directly by the CS47L35, but can be derived using knowledge of the DSPCLK frequency, if available.

4.4.2.3 DSP Code Execution

After the DSP firmware has been loaded, and the clocks configured, the DSP blocks are enabled by setting the DSP*n*_CORE_ENA bits. When the DSP is configured and enabled, the firmware execution can be started by writing 1 to the respective DSP*n*_START bit.

Alternative methods to trigger the firmware execution can also be configured using the DSP*n*_START_IN_SEL fields. Note that this provides the capability to synchronously trigger multiple DSP blocks.

Using the DSP*n*_START_IN_SEL fields, the DSP firmware execution can be linked to the respective DMA function, the IRQ2 status, DSP*n* start signals from another DSP, 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
- DSP*n* start signals: firmware execution commences when the respective start signal is triggered in the selected DSP core
- 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 respective DSP block, the DSP*n*_CORE_ENA bit must be set. Note that the usage of the DSP*n*_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 DSP*n*_START_IN_SEL), writing to the DSP*n*_START bit is not required.

The DSPCLK system clock must be configured and enabled before any DSP processing core is enabled. The DSP blocks should be kept disabled (DSPn_CORE_ENA = 0) if DSPCLK is not enabled. See Section 4.16 for details of the system clocks (including requirements for reconfiguring DSPCLK while DSP cores are enabled).

4.4.2.4 DSP Control Registers

The DSP memory, clocking, and code-execution control registers are described in Table 4-25.

The audio signal paths connecting to/from the DSP processing blocks 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 4-25. DSP Memory and Clocking Control

Register Address	Bit	Label	Default	Description
DSP1 Base Address = R1048064 (0x0F_FE00) DSP2 Base Address = R1572352 (0x17_FE00) DSP3 Base Address = R2096640 (0x1F_FE00)				
Base address DSP*n*_Config_1	18:16	DSP*n*_CLK_SEL[2:0]	000	DSP*n* clock frequency select 000 = 5.5 MHz to 9.375 MHz 001 = 9.375 MHz to 18.75 MHz 010 = 18.75 MHz to 37.5 MHz 011 = 37.5 MHz to 75 MHz 100 = 75 MHz to 150 MHz All other codes are reserved. Note that, because DSPCLK could be any frequency (within the valid ranges), it is not possible to quote exact frequencies in this field definition. The DSP*n* Clock must be less than or equal to the DSPCLK frequency. The exact frequency is derived as DSPCLK divided by 1, 2, 4, 8, or 16.
	4	DSP*n*_MEM_ENA	0	DSP*n* memory control 0 = Disabled 1 = Enabled The DSPn memory contents are lost when DSPn_MEM_ENA =0. Note that this bit is not affected by software reset; it remains in its previous condition.
	1	DSP*n*_CORE_ENA	0	DSP*n* enable. Controls the DSP*n* firmware execution 0 = Disabled 1 = Enabled
	0	DSP*n*_START	—	DSP*n* start Write 1 to start DSP*n* firmware execution
Base address +0x06 DSP*n*_Status_2	31:16	DSP*n*_DUALMEM_COLLISION_ADDR[15:0]	0x0000	DSP*n* dual memory collision address. In the event of a DSP*n* memory access collision, this field reports the address at which the collision occurred. The address is defined relative to the base address of the shared data memory. The LSB represents one 24-bit DSP memory word. **Note:** Valid for DSP2 and DSP3 only.
	3:1	DSP*n*_CLK_SEL_STS[2:0]	000	DSP*n* clock frequency (read only). Valid only when the respective DSP Core is enabled. 000 = 5.5 MHz to 9.375 MHz 001 = 9.375 MHz to 18.75 MHz 010 = 18.75 MHz to 37.5 MHz 011 = 37.5 MHz to 75 MHz 100 = 75 MHz to 150 MHz All other codes are reserved Note that, because DSPCLK could be any frequency (within the valid ranges), it is not possible to quote exact frequencies in this field definition. The exact frequency is derived as DSPCLK divided by 1, 2, 4, 8, or 16.
	0	DSP*n*_CLK_AVAIL	0	DSP*n* 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.
Base address +0x38 DSP*n*_External_Start	4:0	DSP*n*_START_IN_SEL[4:0]	0x00	DSP*n* firmware execution control. Selects the trigger for DSP*n* firmware execution. 0x00 = DMA 0x01 = DSP1 Start 1 0x02 = DSP1 Start 2 0x03 = DSP2 Start 1 0x04 = DSP2 Start 2 0x05 = DSP3 Start 1 0x06 = DSP3 Start 2 0x0B = IRQ2 0x10 = Event Logger 1 0x11 = Event Logger 2 0x12 = Event Logger 3 0x13 = Event Logger 4 All other codes are reserved. Note that the DSP*n*_START bit also starts the DSP*n* firmware execution, regardless of this field setting.

4.4.3 DSP Direct Memory Access (DMA) Control

Each DSP provides a multichannel DMA function; this is configured using the registers described in Table 4-26.

There are eight WDMA (DSP input) and six RDMA (DSP output) channels for each DSP; these are enabled using the DSP*n*_WDMA_CHANNEL_ENABLE and DSP*n*_RDMA_CHANNEL_ENABLE fields. The status of each WDMA channel is indicated in DSP*n*_WDMA_ACTIVE_CHANNELS.

The DMA can access the X-data memory or Y-data memory associated with the respective DSP. 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 4-26. 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 DSP*n*_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 CS47L35 register map layout described in Table 4-23.

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 DSP*n*_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 4-25. Meanwhile, further DSP input data fills up the pong buffer.

When the pong input buffer is full, the DSP*n*_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 DSP*n*_PING_FULL and DSP*n*_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 cores support 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 DSP*n*_DMA_WORD_SEL is set, audio data is transferred to and from DSP*n* in Q0.23 format. The applicable format should be set according to the requirements of the specific DSP firmware.

Note that the DSP cores are 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 DSPCLK system clock must be configured and enabled before any DMA channel is enabled. The DMA channels should be kept disabled (DSP*n*_[WDMA/RDMA]_CHANNEL_ENABLE = 0x00) if DSPCLK is not enabled. See Section 4.16 for details of the system clocks (including requirements for reconfiguring DSPCLK while DMA channels are enabled).

The DMA function is an input to the interrupt control circuit—see Section 4.4.4. 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 4-26. DMA Control

Register Address	Bit	Label	Default	Description
DSP1 Base Address = R1048064 (0x0F_FE00) DSP2 Base Address = R1572352 (0x17_FE00) DSP3 Base Address = R2096640 (0x1F_FE00)				
Base address +0x04 DSP*n*_Status_1	31	DSP*n*_PING_FULL	0	DSP*n* WDMA Ping Buffer Status 0 = Not Full 1 = Full
	30	DSP*n*_PONG_FULL	0	DSP*n* WDMA Pong Buffer Status 0 = Not Full 1 = Full
	23:16	DSP*n*_WDMA_ACTIVE_CHANNELS[7:0]	0x00	DSP*n* 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
Base address +0x10 DSP*n*_WDMA_Buffer_1	31:16	DSP*n*_START_ADDRESS_WDMA_BUFFER_1[15:0]	0x0000	DSP*n* 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 DSP*n*_WDMA_CHANNEL_OFFSET bit.
	15:0	DSP*n*_START_ADDRESS_WDMA_BUFFER_0[15:0]	0x0000	DSP*n* WDMA Channel 0 Start Address Field description is as above.
Base address +0x12 DSP*n*_WDMA_Buffer_2	31:16	DSP*n*_START_ADDRESS_WDMA_BUFFER_3[15:0]	0x0000	DSP*n* WDMA Channel 3 Start Address Field description is as above.
	15:0	DSP*n*_START_ADDRESS_WDMA_BUFFER_2[15:0]	0x0000	DSP*n* WDMA Channel 2 Start Address Field description is as above.
Base address +0x14 DSP*n*_WDMA_Buffer_3	31:16	DSP*n*_START_ADDRESS_WDMA_BUFFER_5[15:0]	0x0000	DSP*n* WDMA Channel 5 Start Address Field description is as above.
	15:0	DSP*n*_START_ADDRESS_WDMA_BUFFER_4[15:0]	0x0000	DSP*n* WDMA Channel 4 Start Address Field description is as above.
Base address +0x16 DSP*n*_WDMA_Buffer_4	31:16	DSP*n*_START_ADDRESS_WDMA_BUFFER_7[15:0]	0x0000	DSP*n* WDMA Channel 7 Start Address Field description is as above.
	15:0	DSP*n*_START_ADDRESS_WDMA_BUFFER_6[15:0]	0x0000	DSP*n* WDMA Channel 6 Start Address Field description is as above.
Base address +0x20 DSP*n*_RDMA_Buffer_1	31:16	DSP*n*_START_ADDRESS_RDMA_BUFFER_1[15:0]	0x0000	DSP*n* 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 DSP*n*_RDMA_CHANNEL_OFFSET bit.
	15:0	DSP*n*_START_ADDRESS_RDMA_BUFFER_0[15:0]	0x0000	DSP*n* RDMA Channel 0 Start Address Field description is as above.
Base address +0x22 DSP*n*_RDMA_Buffer_2	31:16	DSP*n*_START_ADDRESS_RDMA_BUFFER_3[15:0]	0x0000	DSP*n* RDMA Channel 3 Start Address Field description is as above.
	15:0	DSP*n*_START_ADDRESS_RDMA_BUFFER_2[15:0]	0x0000	DSP*n* RDMA Channel 2 Start Address Field description is as above.

Table 4-26. DMA Control *(Cont.)*

Register Address	Bit	Label	Default	Description
Base address +0x24 DSP*n*_RDMA_Buffer_3	31:16	DSP*n*_START_ADDRESS_RDMA_BUFFER_5[15:0]	0x0000	DSP*n* RDMA Channel 5 Start Address Field description is as above.
	15:0	DSP*n*_START_ADDRESS_RDMA_BUFFER_4[15:0]	0x0000	DSP*n* RDMA Channel 4 Start Address Field description is as above.
Base address +0x30 DSP*n*_DMA_Config_1	23:16	DSP*n*_WDMA_CHANNEL_ENABLE[7:0]	0x00	DSP*n* 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	DSP*n*_DMA_BUFFER_LENGTH[13:0]	0x0000	DSP*n* DMA Buffer Length Selects the amount of data transferred in each DMA channel. The LSB represents one 24-bit DSP memory word.
Base address +0x32 DSP*n*_DMA_Config_2	7:0	DSP*n*_WDMA_CHANNEL_OFFSET[7:0]	0x00	DSP*n* 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
Base address +0x34 DSP*n*_DMA_Config_3	21:16	DSP*n*_RDMA_CHANNEL_OFFSET[5:0]	0x00	DSP*n* 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	DSP*n*_RDMA_CHANNEL_ENABLE[5:0]	0x00	DSP*n* 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
Base address +0x36 DSP*n*_DMA_Config_4	0	DSP*n*_DMA_WORD_SEL	0	DSP*n* 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.

4.4.4 DSP Interrupts

The DSP cores provide inputs to the interrupt circuit and can be used to trigger an interrupt event when the associated conditions occur. For each DSP, the following interrupts are provided:

- 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)

The CS47L35 also provides 16 control bits that allow the DSP cores to generate programmable interrupt events. When a 1 is written to these bits (see Table 4-27), 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 4.15 for further details.

Table 4-27. DSP Interrupts

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

4.4.5 DSP Debug Support

General-purpose registers are provided for each 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 CS47L35, as described in Section 4.20. The JTAG interface clock can be enabled independently for each DSP core using the DSP*n*_DBG_CLK_ENA bits. Note that, when the JTAG interface is used to access any DSP core, the respective DSP*n*_CORE_ENA bit must also be set.

Table 4-28. DSP Debug Support

Register Address	Bit	Label	Default	Description
DSP1 Base Address = R1048064 (0x0F_FE00) DSP2 Base Address = R1572352 (0x17_FE00) DSP3 Base Address = R2096640 (0x1F_FE00)				
Base address DSP*n*_Config_1	3	DSP*n*_DBG_CLK_ENA	0	DSP*n* Debug Clock Enable 0 = Disabled 1 = Enabled
Base address +0x40	31:16	DSP*n*_SCRATCH_1[15:0]	0x0000	DSP*n* Scratch Register 1
DSP*n*_Scratch_1	15:0	DSP*n*_SCRATCH_0[15:0]	0x0000	DSP*n* Scratch Register 0
Base address +0x42	31:16	DSP*n*_SCRATCH_3[15:0]	0x0000	DSP*n* Scratch Register 3
DSP*n*_Scratch_2	15:0	DSP*n*_SCRATCH_2[15:0]	0x0000	DSP*n* Scratch Register 2

4.4.6 Virtual DSP Registers

The DSP control registers, described throughout Section 4.4 are implemented for each DSP core. Each control register has a unique location within the CS47L35 register map.

An additional set of DSP control registers is also defined, which can be used in firmware to access any of the DSPs: 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 cores. When DSP firmware accesses the virtual registers, the registers are automatically mapped onto the control registers corresponding to whichever DSP core is making the read/write access. For example, if DSP1 accesses these registers, they are mapped onto the DSP1 control registers. If DSP2 accesses the virtual registers, they are mapped onto the DSP2 control registers.

The virtual DSP registers are designed to allow software to be transferable to any of the DSPs 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. 4-28. The virtual DSP registers are located at address 0xD000 in the X-data memory map.

4.5 DSP Peripheral Control

The CS47L35 incorporates a suite of DSP peripheral functions that can be integrated together to support the sensor-hub capability. A master I²C interface is provided for external sensor connectivity. Configurable event log functions provide multichannel monitoring of internal and external signals. The general-purpose timers provide time-stamp data for the event logs and support watchdog and other miscellaneous time-based functions. Maskable GPIO provides an efficient mechanism for multiple DSPs to access the respective input and output signals.

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

4.5.1 Master Interface (MIF)

The CS47L35 incorporates an I²C master interface, offering a flexible capability for additional sensor/accessory input.

4.5.1.1 Overview

The master interface (MIF1) can support single- and multiple-master I²C operation up to 1 MHz. The master interface supports 7- and 10-bit slave addressing modes. Master device arbitration algorithms are implemented, in accordance with the standard I²C protocol. A watchdog timer is provided for detecting interface error conditions.

The master interface is ideally suited for connection to external sensors such as accelerometers, gyroscopes, and magnetometers for motion-sensing and navigation applications. Other example accessories include barometers, and ambient light sensors, for environmental awareness. Flow-control bits for the TX and RX data buffers enable easy integration with external devices and with internal DSP functions.

4.5.1.2 Clocking Configuration

Clocking for the master interface is derived from DSPCLK, which must be enabled and present when using the MIF function. Standard I²C bus rates can be supported for typical DSPCLK frequencies using the register settings described in Table 4-29.

Table 4-29. Master Interface Clock Configuration

Register Address	Condition	Value
R262144 (0x40000)	—	0x0000_0006
R262208 (0x40040)	—	0x0000_0000
R262210 (0x40042)	10 kHz I²C Mode	0x01CC_01CC
	100 kHz I²C Mode	0x002E_002E
	400 kHz I²C Mode	0x000C_000C
	1 MHz I²C Mode	0x0005_0005

Note: It is assumed that the DSPCLK frequency is one of the nominal (typical) frequencies specified in Table 4-93.

The DSPCLK system clock must be configured and enabled before a Master Interface transaction is scheduled. The Master Interfaces should be kept idle if DSPCLK is not enabled. See Section 4.16 for details of the system clocks (including requirements for reconfiguring DSPCLK while DSP peripherals are enabled).

4.5.1.3 Transmit and Receive Data Buffers

The transmit (master write) and receive (master read) actions are each supported by 16-byte data buffers, allowing I²C transfers of up to 2,097,152 data bytes (2 MB). The number of data bytes transferred in each I²C operation is selected using MIF1_TX_LENGTH or MIF1_RX_LENGTH.

Data to be transmitted is managed using the TX data buffers; the application software must load data into the buffer registers (MIF1_TX_BYTEx) and then write 1 to the MIF1_TX_DONE bit to commit that data for transmission. The MIF1_TX_REQUEST bit indicates when the buffer registers are ready for loading new data. Internal buffering of the TX data enables uninterrupted I²C writes. If new data is not ready for transmission, SCLK halts until the buffer registers have been filled.

Data received on the interface is managed using the RX data buffers; the MIF1_RX_REQUEST bit indicates when the buffer registers contain new data. The application software must read the buffer registers (MIF1_RX_BYTEn), and then write 1 to the MIF1_RX_DONE bit to confirm the data has been read. Internal buffering of the RX data enables uninterrupted I²C reads. If the buffers are not ready to receive new data, SCLK halts until the buffer registers have been read.

The master interface divides each I²C transaction into one or more data blocks. The block length is configurable using the MIF1_TX_BLOCK_LENGTH and MIF1_RX_BLOCK_LENGTH fields. The block length is equal to the number of bytes transmitted/received for each TX_DONE/RX_DONE action. The maximum block length is 16 bytes, corresponding to the size of the TX and RX data buffers.

Note: The order in which the data bytes in the TX/RX buffers are transferred depends on the selected MIF1_WORD_SIZE setting. Correct setting of the word size ensures that each data word is transmitted/received most-significant byte first.

The Master Interface is configured for Read (RX) or Write (TX) operation using the MIF1_READ_WRITE_SEL bit. Each I2C transfer is started by writing 1 to the MIF1_START bit. In the case of a Master Write, data must be committed to the TX data buffers using the TX_DONE bit, to enable the transfer to proceed—note that the first block of transmit data can be committed to the TX buffers before or after writing to the START bit for the respective transfer.

4.5.1.4 Interrupts and Status Bits

The MIF1_BUSY_STS bit indicates when the master interface is executing an I²C transaction. This bit is set during each I²C transaction, and cleared on completion. An interrupt event is also triggered on completion of the I²C transfer, if the corresponding MIF1_DONE_EINTx is unmasked as an input to the IRQ circuit.

Additional status bits are provided to indicate watchdog timeout, loss of bus arbitration, or a NACK error signal received. Table 4-30 describes these bits.

Note that the MIF done indication is asserted each time an I²C transfer completes, including when an error condition has occurred. It is recommended that the master interface status bits be checked after each I²C transaction, so corrective action can be taken when necessary.

The master interface provides inputs to the interrupt control circuit. An interrupt event is triggered on completion of each TX/RX block, and on completion of the I²C transaction; see Section 4.15.

4.5.1.5 External Connections

The external connections associated with the I²C master interface (MIF) are implemented on multi-function GPIO pins, which must be configured for the respective MIF functions when required. The MIF1SCLK and MIF1SDA connections are pin-specific alternative functions available on specific GPIO pins only. See Section 4.14 to configure the GPIO pins for the MIF operation.

Fig. 4-29 shows a typical master I²C write transfer.

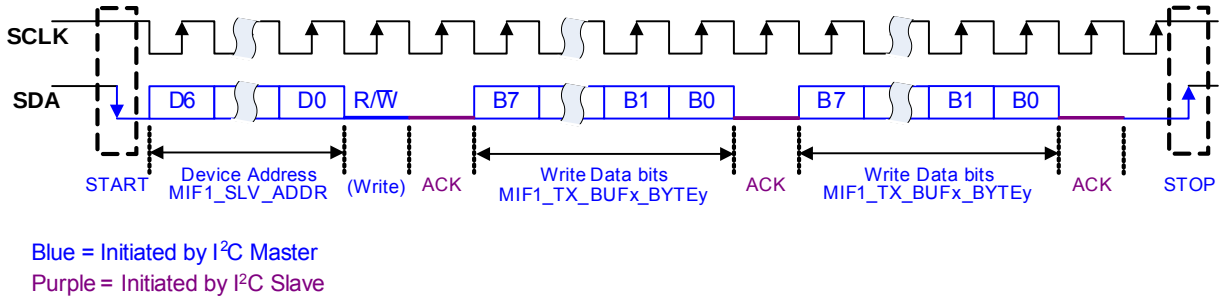


Figure 4-29. Master I²C Write

Fig. 4-30 shows a typical master I²C read transfer.

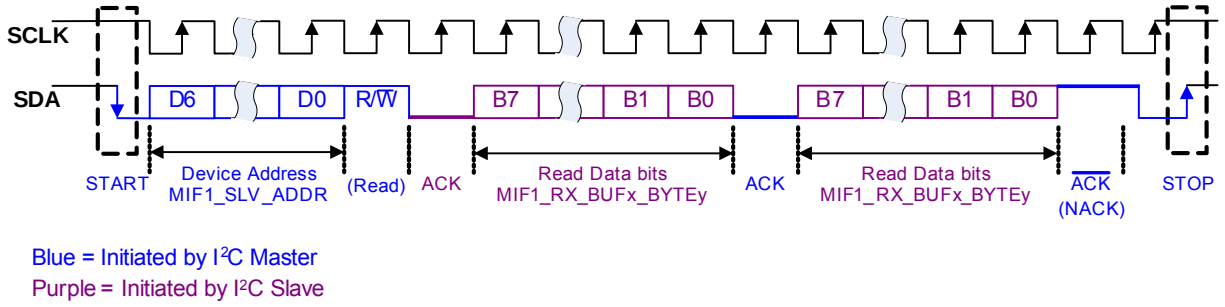


Figure 4-30. Master I²C Read

Fig. 4-31 shows a typical master I²C write/read transfer; the read transaction is preceded by a repeated start.

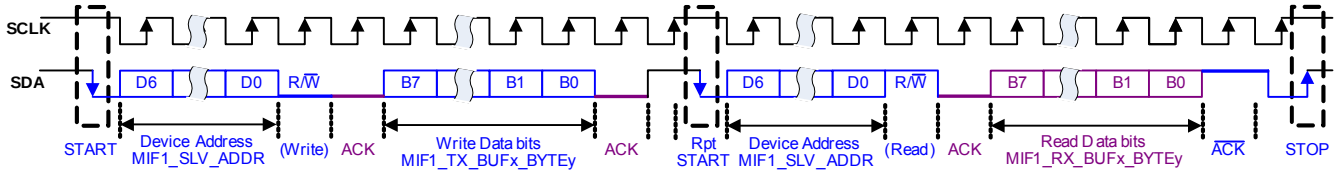


Figure 4-31. Master I²C Write and Read

4.5.1.6 Master Interface Control Registers

The MIF control registers are described in Table 4-30.

Table 4-30. Master Interface (MIF1) Control

Register Address	Bit	Label	Default	Description
R262146 (0x40002) MIF1_I2C_CONFIG_2	10:1	MIF1_SLV_ADDR[9:0]	0x000	Address of Slave on which transactions are executed. For 7-Bit Mode, lower 7 bits of field are used.
	0	MIF1_ADDR_MODE	0	Selects the addressing mode of I²C Master 0 = 7-Bit Mode 1 = 10-Bit Mode

Table 4-30. Master Interface (MIF1) Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R262148 (0x40004) MIF1_I2C_CONFIG_3	3	MIF1_NACK_ RESPONSE	0	Selects the action taken if NACK is received from Slave. 0 = Stop Condition sent. 1 = Stop Condition not sent; next transaction commences with a Repeated Start. Note that, if the Stop Condition is not sent, the master retains control of the bus until a subsequent action is scheduled. The next transaction commences with a Repeated Start in this case.
	2	MIF1_SCL_MON_ENA	1	Enables bus monitoring functions on SCLK 0 = Disabled 1 = Enabled This feature enables support for clock stretching by slave devices, and enables bus synchronization as part of multimaster operation.
	1	MIF1_RPT_START	0	Selects the action taken on completion of a bus transaction. 0 = Stop Condition sent. 1 = Stop Condition not sent; next transaction commences with a Repeated Start. Note that, if the Stop Condition is not sent, the master retains control of the bus until a subsequent action is scheduled. The next transaction commences with a Repeated Start in this case.
	0	MIF1_START_BYTE_ ENA	0	Selects whether a Start Byte is transmitted before an I²C transaction. 0 = Disabled 1 = Enabled The Start Byte is a dummy transaction that provides support for bus devices that use low-frequency polling to detect I²C activity. The Start Byte, when enabled, is transmitted before the Slave Address bytes. It is not acknowledged on the bus by any device.
R262152 (0x40008) MIF1_I2C_CONFIG_5	0	MIF1_WDT_ENA	0	Watchdog Timer (WDT) control 0 = Disabled 1 = Enabled When bus monitoring functions are enabled (MIF1_SCL_MON_ ENA = 1), the watchdog timer is used to detect the SCLK line being pulled low for a prolonged duration.
R262272 (0x40080) MIF1_I2C_STATUS_1	2	MIF1_WDT_TIMEOUT_ STS	0	Watchdog Timer (WDT) Error Status. This bit, when set, indicates that the WDT expired during the I²C transaction. This bit is latched when set; it is only cleared on next I²C transaction.
	1	MIF1_ARBIT_LOST_ STS	0	Arbitration Error Status. This bit, when set, indicates that arbitration was lost during the I2C transaction. This bit is latched when set; it is only cleared on next I²C transaction.
	0	MIF1_NACK_STS	0	NACK Error Status. This bit, when set, indicates that a NACK Error signal was received during the I²C transaction. This bit is latched when set; it is only cleared on next I²C transaction.
R262400 (0x40100) MIF1_CONFIG_1	0	MIF1_START	0	Starts the I²C transaction Write 1 to start.
R262404 (0x40104) MIF1_CONFIG_3	17:16	MIF1_WORD_SIZE[1:0]	00	Selects the data word format. I²C transactions are made up of 1-Byte data words; the sequence order of these words differs according to the applicable word format. Correct setting of the MIF1_WORD_SIZE field ensures that each data word is transmitted/received as MSB first. 00 = 8-bit (1, 2, 3, 4, 5, 6, 7, 8, etc) 01 = 16-bit (2, 1, 4, 3, 6, 5, 8, 7, etc) 10 = 32-bit (4, 3, 2, 1, 8, 7, 6, 5, etc) The bracketed numbers describe the order in which applicable MIF1_ [TX|RX]_BYTE*x* fields are transmitted/received over the I²C interface.
	0	MIF1_READ_WRITE_ SEL	0	Selects the I²C Command type 0 = Master Write 1 = Master Read

Table 4-30. Master Interface (MIF1) Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R262406 (0x40106) MIF1_CONFIG_4	20:0	MIF1_TX_ LENGTH[20:0]	0x00_ 0000	Selects the total number of data bytes in an I²C Write operation. 0x00_0000 = 1 byte 0x00_0001 = 2 bytes 0x00_0002 = 3 bytes … 0x1F_FFFF = 2,097,152 bytes
R262416 (0x40110) MIF1_CONFIG_5	20:0	MIF1_RX_ LENGTH[20:0]	0x00_ 0000	Selects the total number of data bytes in an I²C Read operation. 0x00_0000 = 1 byte 0x00_0001 = 2 bytes 0x00_0002 = 3 bytes … 0x1F_FFFF = 2,097,152 bytes
R262418 (0x40112) MIF1_CONFIG_6	7:0	MIF1_TX_BLOCK_ LENGTH[7:0]	0x10	Selects the interval at which the MIF1_BLOCK Interrupt is triggered during I²C Write operations. 0x00 = 1 byte 0x01 = 1 byte 0x02 = 2 bytes … 0x10 = 16 bytes All other codes are reserved
R262420 (0x40114) MIF1_CONFIG_7	7:0	MIF1_RX_BLOCK_ LENGTH[7:0]	0x10	Selects the interval at which the MIF1_BLOCK Interrupt is triggered during I²C Read operations. 0x00 = 1 byte 0x01 = 1 byte 0x02 = 2 bytes … 0x10 = 16 bytes All other codes are reserved
R262422 (0x40116) MIF1_CONFIG_8	4	MIF1_RX_DONE	0	RX Buffer access control bit. Write 1 to indicate that data in the RX Buffer has been read. In normal operation, a 1 is written after reading the RX buffer. This causes the MIF1_RX_REQUEST bit to be cleared. (Note that, if further data is available to read, the MIF1_RX_REQUEST bit remains set in this case.)
	0	MIF1_TX_DONE	0	TX Buffer access control bit. Write 1 to indicate the TX Buffer has been filled with data for transmission. In normal operation, a 1 is written after writing the TX buffer. This causes the MIF1_TX_REQUEST bit to be cleared.
R262528 (0x40180) MIF1_STATUS_1	8	MIF1_BUSY_STS	0	MIF Busy Status. This bit, when set, indicates that the master interface is executing an I²C transaction.
	8	MIF1_RX_REQUEST	0	RX Buffer flow control bit 0 = No data available to read 1 = Buffer data is available to read
	1	MIF1_TX_REQUEST	0	TX Buffer flow control bit 0 = TX buffer not available to write 1 = TX buffer is available to write
R262530 (0x40182) MIF1_STATUS_2	20:0	MIF1_BYTE_ COUNT[20:0]	0x00_ 0000	Number of data bytes transferred in current transaction. Note that this field is cleared on completion of the I²C transaction.
R262656 (0x40200) MIF1_TX_1	31:24	MIF1_TX_BYTE4[7:0]	0x00	TX Byte 4
	23:16	MIF1_TX_BYTE3[7:0]	0x00	TX Byte 3
	15:8	MIF1_TX_BYTE2[7:0]	0x00	TX Byte 2
	7:0	MIF1_TX_BYTE1[7:0]	0x00	TX Byte 1
R262658 (0x40202) MIF1_TX_2	31:24	MIF1_TX_BYTE8[7:0]	0x00	TX Byte 8
	23:16	MIF1_TX_BYTE7[7:0]	0x00	TX Byte 7
	15:8	MIF1_TX_BYTE6[7:0]	0x00	TX Byte 6
	7:0	MIF1_TX_BYTE5[7:0]	0x00	TX Byte 5

Table 4-30. Master Interface (MIF1) Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R262660 (0x40204) MIF1_TX_3	31:24	MIF1_TX_BYTE12[7:0]	0x00	TX Byte 12
	23:16	MIF1_TX_BYTE11[7:0]	0x00	TX Byte 11
	15:8	MIF1_TX_BYTE10[7:0]	0x00	TX Byte 10
	7:0	MIF1_TX_BYTE9[7:0]	0x00	TX Byte 9
R262662 (0x40206) MIF1_TX_4	31:24	MIF1_TX_BYTE16[7:0]	0x00	TX Byte 16
	23:16	MIF1_TX_BYTE15[7:0]	0x00	TX Byte 15
	15:8	MIF1_TX_BYTE14[7:0]	0x00	TX Byte 14
	7:0	MIF1_TX_BYTE13[7:0]	0x00	TX Byte 13
R262912 (0x40300) MIF1_RX_1	31:24	MIF1_RX_BYTE4[7:0]	0x00	RX Byte 4
	23:16	MIF1_RX_BYTE3[7:0]	0x00	RX Byte 3
	15:8	MIF1_RX_BYTE2[7:0]	0x00	RX Byte 2
	7:0	MIF1_RX_BYTE1[7:0]	0x00	RX Byte 1
R262914 (0x40302) MIF1_RX_2	31:24	MIF1_RX_BYTE8[7:0]	0x00	RX Byte 8
	23:16	MIF1_RX_BYTE7[7:0]	0x00	RX Byte 7
	15:8	MIF1_RX_BYTE6[7:0]	0x00	RX Byte 6
	7:0	MIF1_RX_BYTE5[7:0]	0x00	RX Byte 5
R262916 (0x40304) MIF1_RX_3	31:24	MIF1_RX_BYTE12[7:0]	0x00	RX Byte 12
	23:16	MIF1_RX_BYTE11[7:0]	0x00	RX Byte 11
	15:8	MIF1_RX_BYTE10[7:0]	0x00	RX Byte 10
	7:0	MIF1_RX_BYTE9[7:0]	0x00	RX Byte 9
R262918 (0x40306) MIF1_RX_4	31:24	MIF1_RX_BYTE16[7:0]	0x00	RX Byte 16
	23:16	MIF1_RX_BYTE15[7:0]	0x00	RX Byte 15
	15:8	MIF1_RX_BYTE14[7:0]	0x00	RX Byte 14
	7:0	MIF1_RX_BYTE13[7:0]	0x00	RX Byte 13

4.5.2 Event Loggers

The CS47L35 provides four event log functions, supporting multichannel, edge-sensitive monitoring and recording of internal or external signals.

4.5.2.1 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.

4.5.2.2 Event Logger Control

The event logger is enabled by setting EVENTLOG*n*_ENA (where *n* identifies the respective event logger, 1–4).

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.

4.5.2.3 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 cores. A list of the valid input sources for the event loggers is provided in Table 4-32. 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.

4.5.2.4 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 4.5.2.5 confirm the current status of the buffer.

4.5.2.5 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.

4.5.2.6 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 4.18 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 4.15.

The event log status can be output directly on a GPIO pin as an external indication of the event logger; see Section 4.14 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.

4.5.2.7 Event Logger Control Registers

The event logger control registers are described in Table 4-31.

Table 4-31. 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) Event Log 3 Base Address = R295936 (0x4_8400) Event Log 4 Base Address = R296448 (0x4_8600)				
base address EVENTLOG*n*_CONTROL	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 10 = Timer 3 11 = Timer 4 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.

Table 4-31. Event Logger (EVENTLOG*n*) Control *(Cont.)*

Register Address	Bit	Label	Default	Description
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.
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

Table 4-31. Event Logger (EVENTLOG*n*) Control *(Cont.)*

Register Address	Bit	Label	Default	Description
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.
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.

Table 4-31. Event Logger (EVENTLOG*n*) Control *(Cont.)*

Register Address	Bit	Label	Default	Description
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.
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*_CH15x_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.

Table 4-31. Event Logger (EVENTLOG*n*) Control *(Cont.)*

Register Address	Bit	Label	Default	Description
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]
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]

Table 4-31. Event Logger (EVENTLOG*n*) Control *(Cont.)*

Register Address	Bit	Label	Default	Description
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]
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 4-32 for valid channel source selections

4.5.2.8 Event Logger Input Sources

A list of the valid input sources for the event loggers is provided in Table 4-32.

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.

Take care 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 4-32. Event Logger Input Sources

ID	Description	Edge
3	irq1	D
4	irq2	D
9	sysclk_fail	S
24	fll1_lock	D
32	frame_start_g1r1	S
33	frame_start_g1r2	S
34	frame_start_g1r3	S
80	hpdet	S
88	micdet	S
96	jd1_rise	S

ID	Description	Edge
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

ID	Description	Edge
323	Timer4	S
336	event1_not_empty	S
337	event2_not_empty	S
338	event3_not_empty	S
339	event4_not_empty	S
352	event1_full	S
353	event2_full	S
354	event3_full	S
355	event4_full	S
368	event1_wmark	S

Table 4-32. Event Logger Input Sources *(Cont.)*

ID	Description	Edge
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

ID	Description	Edge
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
271	gpio16	D
320	Timer1	S
321	Timer2	S
322	Timer3	S

ID	Description	Edge
369	event2_wmark	S
370	event3_wmark	S
371	event4_wmark	S
384	dsp1_dma	S
385	dsp2_dma	S
386	dsp3_dma	S
416	dsp1_start1	S
417	dsp2_start1	S
418	dsp3_start1	S
432	dsp1_start2	S
433	dsp2_start2	S
434	dsp3_start2	S
448	dsp1_start	S
449	dsp2_start	S
450	dsp3_start	S
464	dsp1_busy	D
465	dsp2_busy	D
466	dsp3_busy	D
480	mif1_done	S
496	mif1_block	S

4.5.3 General-Purpose Timers

The CS47L35 incorporates four general-purpose timers, which support a wide variety of uses. In particular, these timers provide essential support for the sensor-hub capability.

4.5.3.1 Overview

The timers allow time stamp information to be associated with external sensor activity, 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.

4.5.3.2 Timer Control

The reference clock for each timer is selected using TIMER*n*_REFCLK_SRC, (where *n* identifies the applicable timer, 1–4).

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. The applicable division ratio is determined automatically, assuming the respective clock source has been correctly configured as described in Section 4.16.

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 CS47L35 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 when TIMER*n*_RUNNING_STS = 0.

The reference clock for each Timer should be configured and enabled before starting the Timer, and whenever the Timer is running. If the reference clock is interrupted while the Timer is running, the Timer operation pauses, and resumes again when the clock restarts. See Section 4.16 for details of the system clocks (including requirements for reconfiguring DSPCLK while DSP peripherals are enabled).

4.5.3.3 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 4.15. 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 4.14 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 4.21 to configure this function.

4.5.3.4 Timer Block Diagram and Control Registers

The timer block is shown in Fig. 4-32.

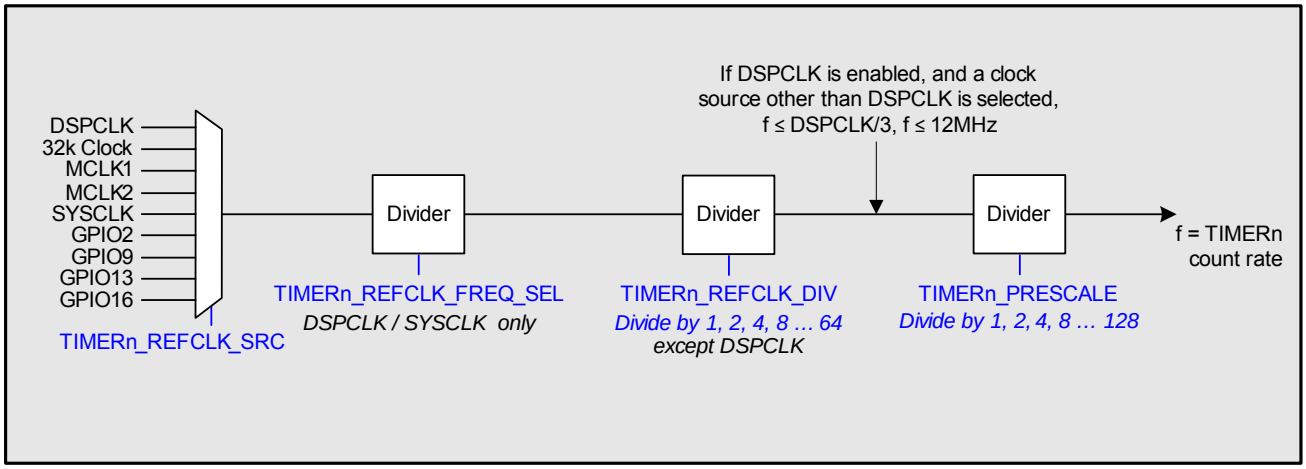


Figure 4-32. General-Purpose Timer

The timer control registers are described in Table 4-33.

Table 4-33. General-Purpose Timer (TIMER*n*) Control

Register Address	Bit	Label	Default	Description
Timer 1 Base Address = R311296 (0x4_C000) Timer 2 Base Address = R311424 (0x4_C080) Timer 3 Base Address = R311552 (0x4_C100) Timer 4 Base Address = R311680 (0x4_C180)				
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 Only valid when SYSCLK or DSPCLK is the source. The selected frequency must be less than or equal to the frequency of the respective source. All other codes are reserved Note that, because DSPCLK could be any frequency (within the valid ranges), it is not possible to quote exact frequencies in this field definition. The exact frequency is derived as DSPCLK divided by 1, 2, 4, 8, or 16. Timer must be stopped (TIMER*n*_RUNNING_STS=0) when updating this field. SYSCLK Source: DSPCLK Source: 000 = 6.144 MHz (5.6448 MHz) 000 = 5.5 MHz to 9.375 MHz 001 = 12.288 MHz (11.2896 MHz) 001 = 9.375 MHz to 18.75 MHz 010 = 24.576 MHz (22.5792 MHz) 010 = 18.75 MHz to 37.5 MHz 011 = 49.152 MHz (45.1584 MHz) 011 = 37.5 MHz to 75 MHz All other codes are reserved
	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 1101 = GPIO9 0001 = 32kHz clock 1000 = SYSCLK 1110 = GPIO13 0100 = MCLK1 1100 = GPIO2 1111 = GPIO16
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.

Table 4-33. General-Purpose Timer (TIMER*n*) Control *(Cont.)*

Register Address	Bit	Label	Default	Description
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

4.5.4 DSP GPIO

The DSP GPIO function provides an advanced I/O capability, supporting the requirements of the CS47L35 as a multipurpose sensor hub.

4.5.4.1 Overview

The CS47L35 supports up to 16 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 4.14.

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 4.14, 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, one GPIO mask is defined for each DSP, or for each functional process; this provides a highly efficient mechanism for each DSP to independently access the respective input and output signals.

4.5.4.2 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.

4.5.4.3 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. 4-33, 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 4-83 for details of the GPIO pull-up and pull-down control bits.

4.5.4.4 DSP GPIO Block Diagram and Control Registers

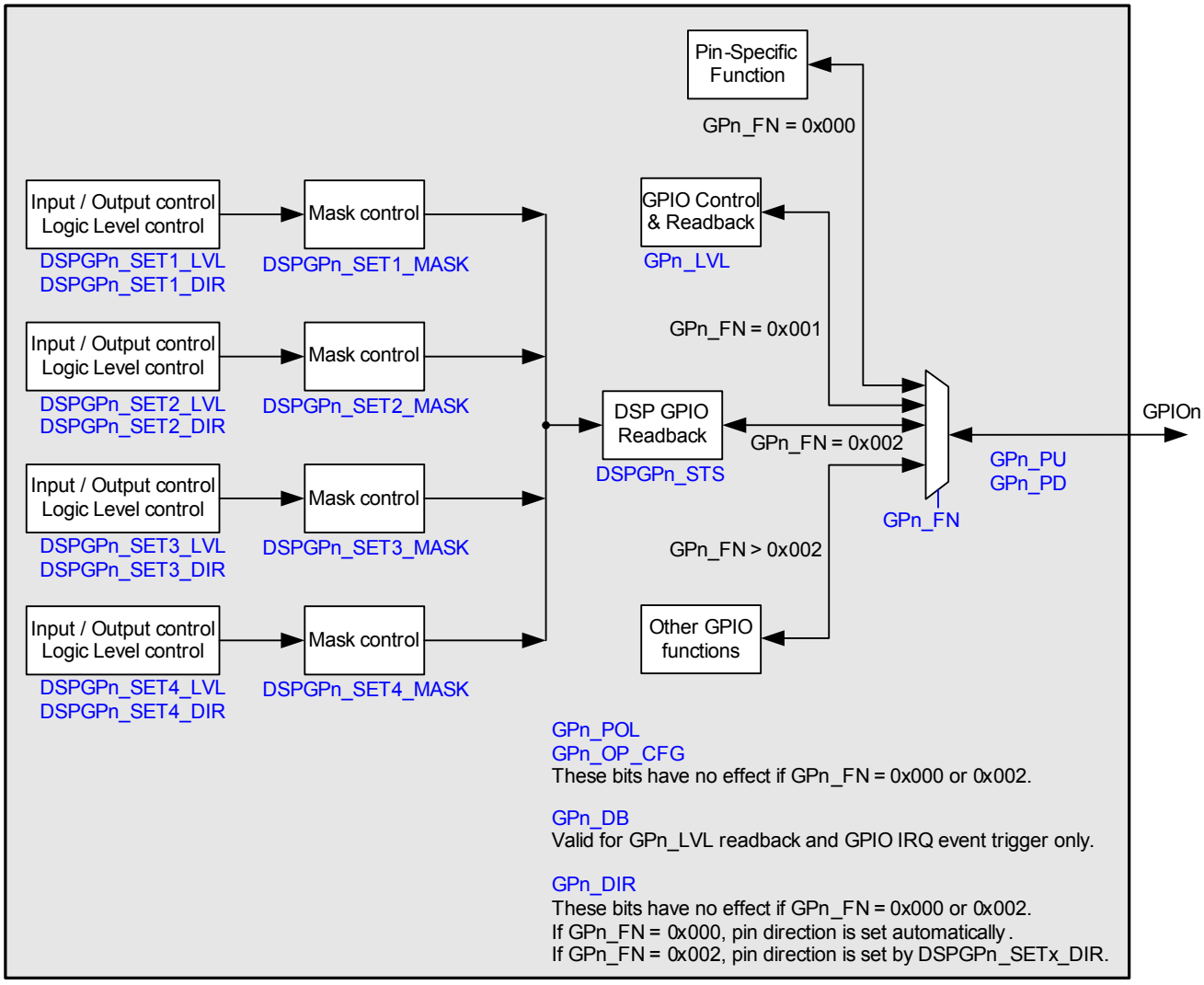


Figure 4-33. DSP GPIO Control

The control registers associated with the DSP GPIO are described in Table 4-34.

Table 4-34. DSP GPIO Control

Register Address	Bit	Label	Default	Description
R315392 (0x4_D000) DSPGP_Status_1	15	DSPGP16_STS	0	DSPGP16 Status Valid for DSPGP input and output
	14	DSPGP15_STS	0	DSPGP15 Status
	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	15	DSPGP16_SET*n*_MASK	1	DSP SET*n* GPIO16 Mask Control 0 = Unmasked 1 = Masked A GPIO pin should be unmasked in a maximum of one SET at any time.
	14	DSPGP15_SET*n*_MASK	1	DSP SET*n* GPIO15 Mask Control
	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	15	DSPGP16_SET*n*_DIR	1	DSP SET*n* GPIO16 Direction Control 0 = Output 1 = Input
	14	DSPGP15_SET*n*_DIR	1	DSP SET*n* GPIO15 Direction Control
	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 4-34. 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	15	DSPGP16_SET*n*_LVL	0	DSP SET*n* GPIO16 Output Level 0 = Logic 0 1 = Logic 1
	14	DSPGP15_SET*n*_LVL	0	DSP SET*n* GPIO15 Output Level
	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 CS47L35 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 and AIF3 support 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 CS47L35 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. Note that the SLIMbus interface also provides digital audio input/output paths, providing options for additional interfaces. A typical configuration is shown in Fig. 4-34.

The audio interface AIF1 is referenced to DBVDD1; interfaces AIF2 and AIF3 are referenced to DBVDD2. This enables the CS47L35 to connect easily between application subsystems on different voltage domains.

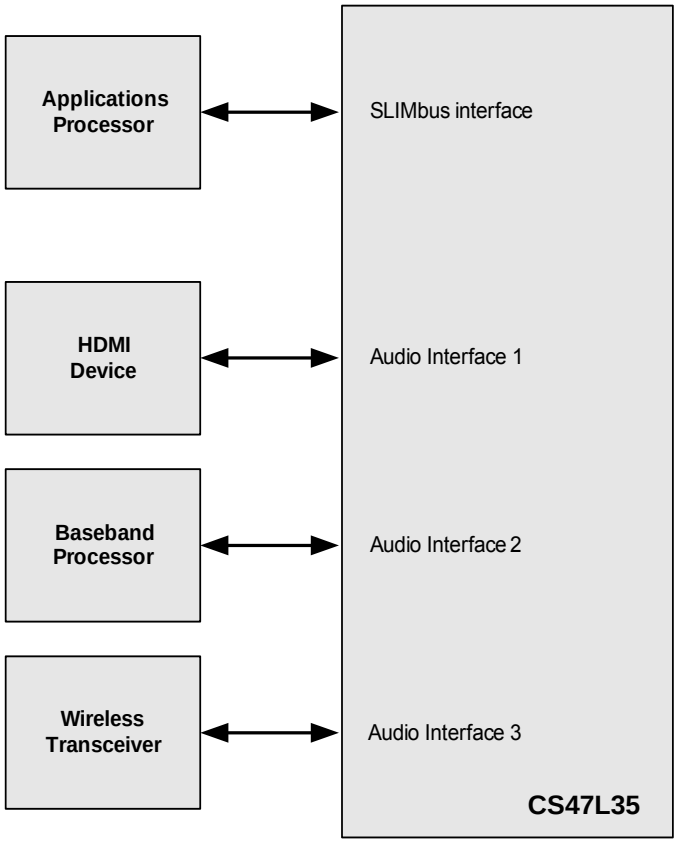


Figure 4-34. Typical AIF Connections

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 CS47L35. 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^2S
- Left-justified

The left-justified and DSP-B formats are valid in Master Mode only (i.e., BCLK and LRCLK are outputs from the CS47L35). 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 CS47L35 digital audio interfaces can operate as a master or slave, as shown in Fig. 4-35 and Fig. 4-36. The associated control bits are described in Section 4.7.

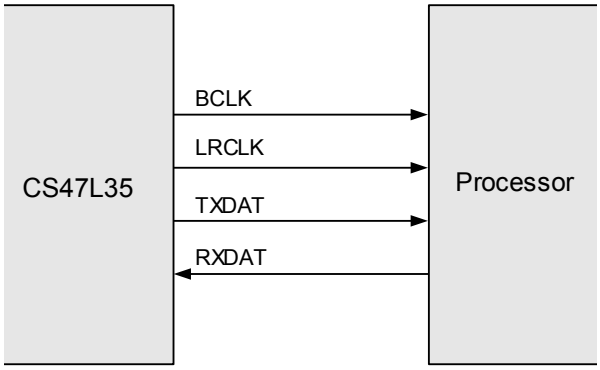


Figure 4-35. Master Mode

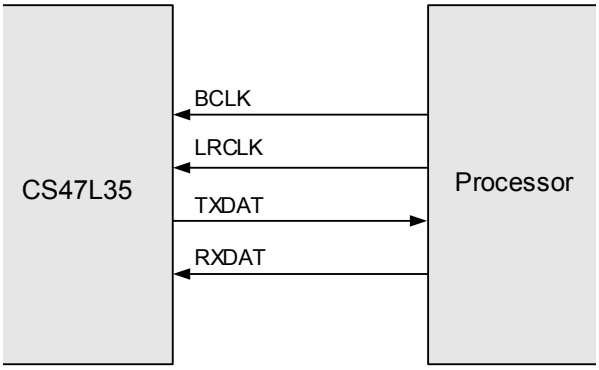


Figure 4-36. Slave Mode

4.6.2 Audio Data Formats

The CS47L35 digital audio interfaces can be configured to operate in I²S, 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 CS47L35).

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 CS47L35 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. 4-37 and Fig. 4-38. 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. CS47L35 data that is output on the left channel is read as mono data by the receiving equipment. Mono PCM data received by the CS47L35 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. 4-37.

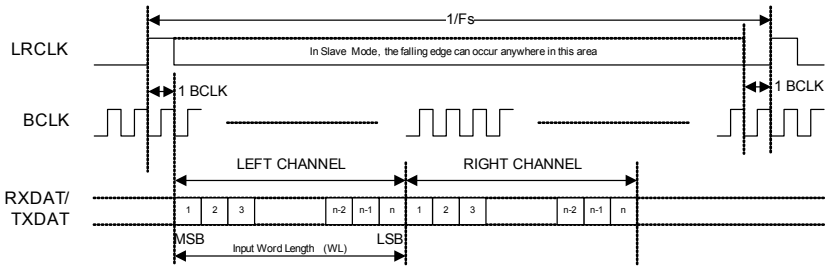


Figure 4-37. DSP Mode A Data Format

DSP Mode B data format is shown in Fig. 4-38.

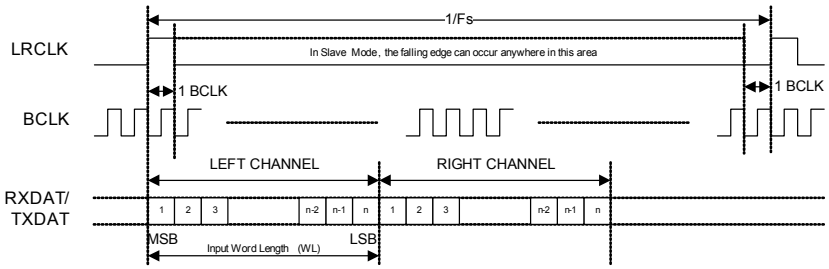


Figure 4-38. DSP Mode B Data Format

- 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. 4-39.

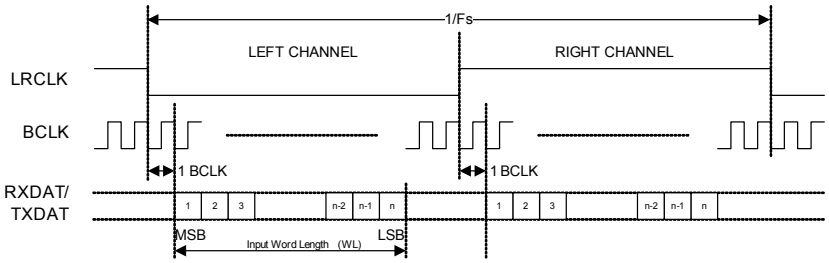


Figure 4-39. I²S Data Format (Assuming n-Bit Word Length)

- 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. 4-40.

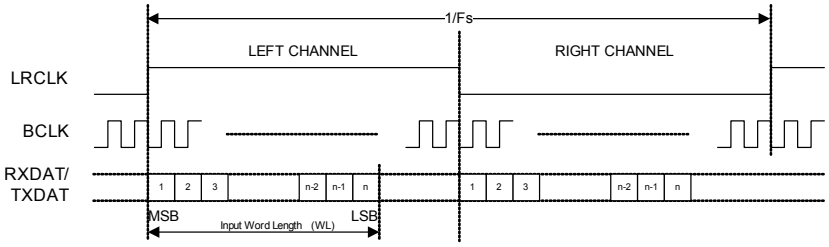


Figure 4-40. Left-Justified Data Format (Assuming n-Bit Word Length)

4.6.3 AIF Time-Slot Configuration

Digital audio interface AIF1 supports multichannel operation, with up to six channels of input and output in each case. 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.

AIF2 and AIF3 also provide flexible configuration options, but these interfaces support 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. 4-41 through Fig. 4-44. One example is shown for each of the four possible data formats.

Fig. 4-41 shows an example of DSP Mode A format. Four enabled audio channels are shown, allocated to time slots 0 through 3.

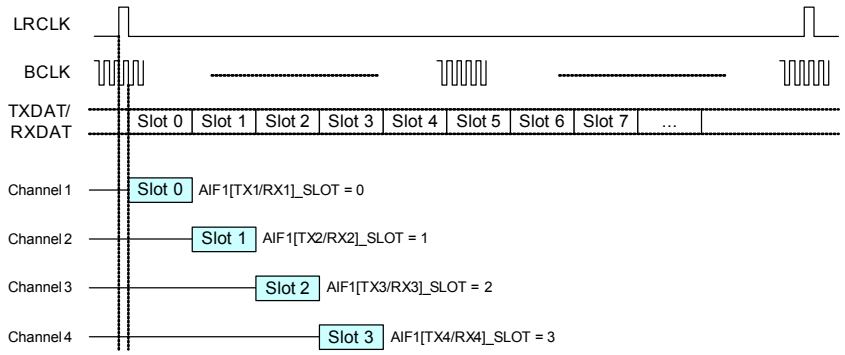


Figure 4-41. DSP Mode A Example

Fig. 4-42 shows an example of DSP Mode B format. Six enabled audio channels are shown, with time slots 4 and 5 unused.

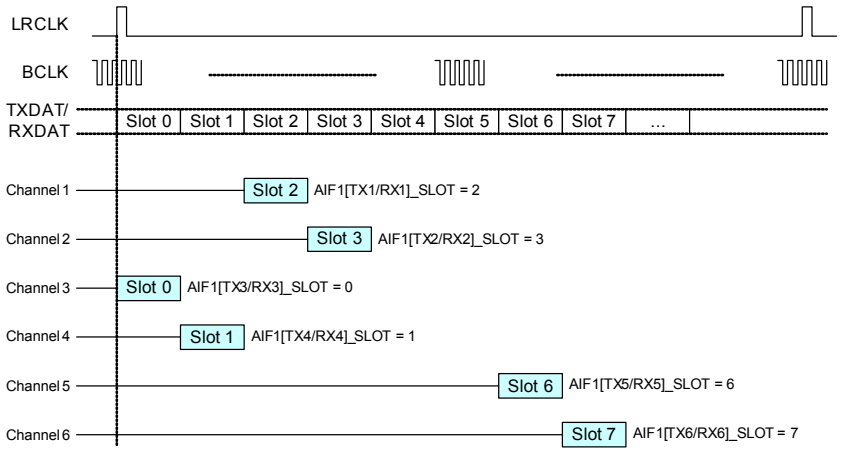


Figure 4-42. DSP Mode B Example

Fig. 4-43 shows an example of I²S format. Four enabled channels are shown, allocated to time slots 0 through 3.

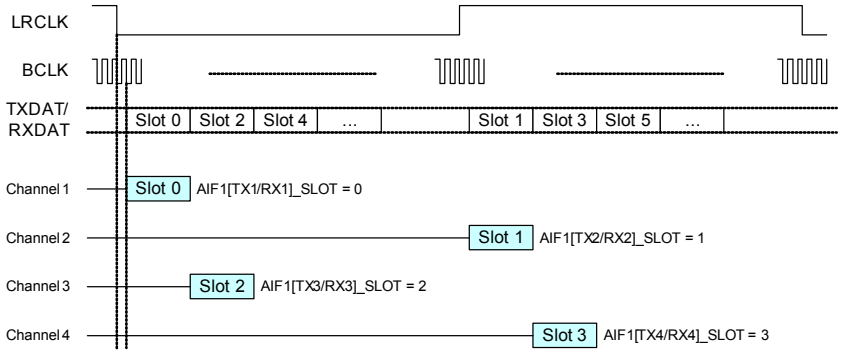


Figure 4-43. I²S Example

Fig. 4-44 shows an example of left-justified format. Six enabled channels are shown.

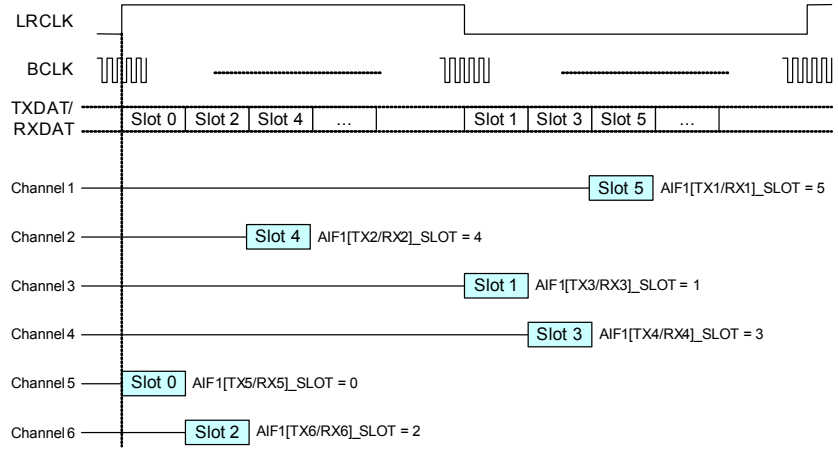


Figure 4-44. Left-Justified Example

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. 4-35 or Fig. 4-36.

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. 4-45, Fig. 4-46, and Fig. 4-47.

The CS47L35 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. 4-45, Fig. 4-46, and Fig. 4-47.

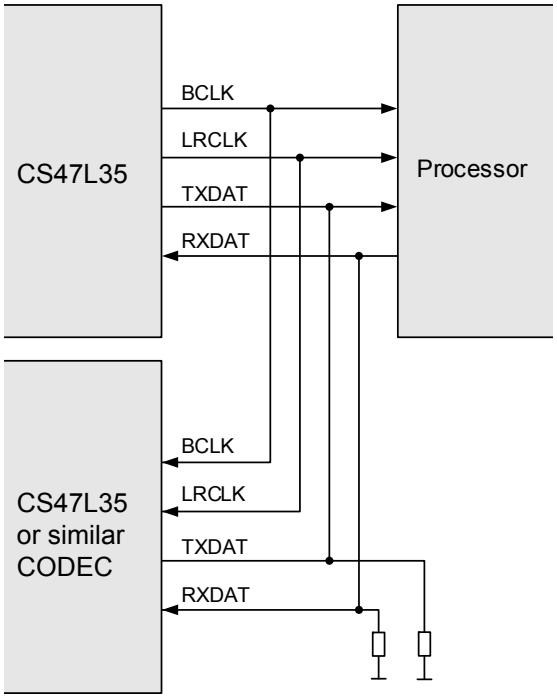


Figure 4-45. TDM with CS47L35 as Master

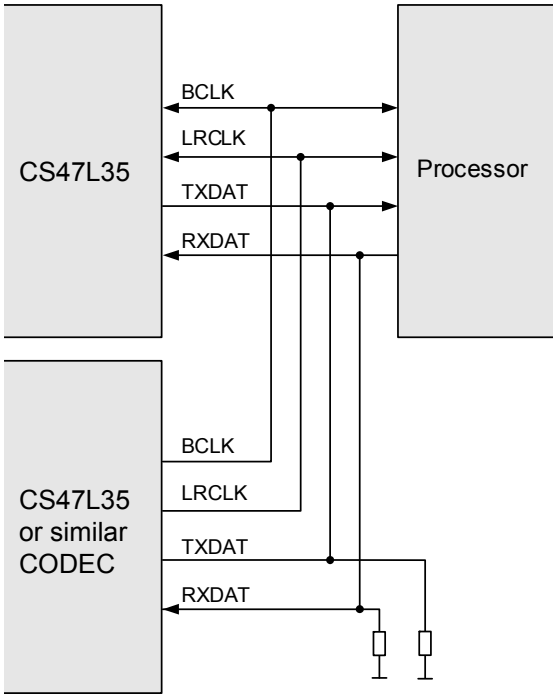


Figure 4-46. TDM with Other Codec as Master

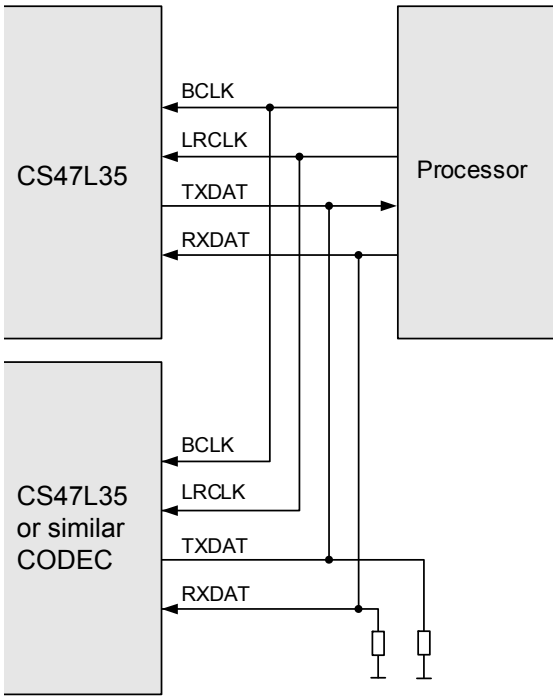


Figure 4-47. TDM with Processor as Master

Note: The CS47L35 is a 24-bit device. If the user operates the CS47L35 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 CS47L35 digital audio interface paths.

AIF1 supports up to six input signal paths and up to six output signal paths; AIF2 and AIF3 support 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 CS47L35 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 4-21.

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 pin-specific alternative functions available on specific GPIO pins. See Section 4.14 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 4-83.

If the pull-up and pull-down resistors are both enabled, the CS47L35 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 CS47L35 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 CS47L35 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_FRC bit should be held at 0 if SYSCLK is not enabled. SYSCLK must be present and enabled before setting the AIF*n*_BCLK_FRC bit. See Section 4.16.4 for further details (including requirements for reconfiguring SYSCLK while AIF clock signals are 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 CS47L35 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_FRC bit should be held at 0 if SYSCLK is not enabled. SYSCLK must be present and enabled before setting the AIF*n*_LRCLK_FRC bit. See Section 4.16.4 for further details (including requirements for reconfiguring SYSCLK while AIF clock signals are enabled).

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 4-35.

Table 4-35. 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

Table 4-35. AIF1 Master/Slave Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R1282 (0x0502) AIF1_Rx_Pin_Ctrl	3	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 4-36.

Table 4-36. 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	3	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 4-37.

Table 4-37. 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

Table 4-37. AIF3 Master/Slave Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R1410 (0x0582) AIF3_Rx_Pin_Ctrl	3	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. 4-48. 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.

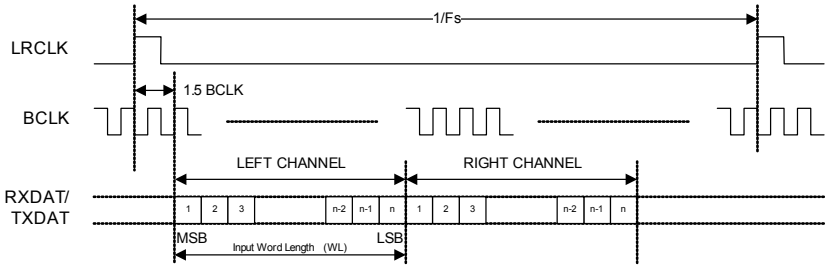


Figure 4-48. LRCLK advance—DSP-A Master Mode

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 4-38.

The AIF2 and AIF3 interfaces support 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 4-39 and Table 4-40.

The system clock, SYSCLK, must be configured and enabled before any audio path is enabled. The AIF signal paths should be kept disabled (AIF*n*_TX*m*_ENA = 0, AIF*n*_RX*m*_ENA = 0) if SYSCLK is not enabled. See Section 4.16 for details of the system clocks (including requirements for reconfiguring SYSCLK while audio paths are enabled).

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 CS47L35 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 4-38.

Table 4-38. 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 4-39.

Table 4-39. AIF2 Signal Path Enable

Register Address	Bit	Label	Default	Description
R1369 (0x0559) AIF2_Tx_Enables	1	AIF2TX2_ENA	0	AIF2 Audio Interface TX 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	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 4-40.

Table 4-40. 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 4-41 through Table 4-43.

The selected AIF*n*BCLK rate must be less than or equal to SYSCLK/2. See Section 4.16 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 4-41.

Table 4-41. 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 4-42.

Table 4-42. 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 4-43.

Table 4-43. 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 4-44, Table 4-45, and Table 4-46 respectively.

Note that left-justified and DSP-B modes are valid in Master Mode only (i.e., BCLK and LRCLK are outputs from the CS47L35).

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. 4-41 through Fig. 4-44.

Note that, in DSP modes, the time slots are ordered consecutively from the start of the LRCLK frame. In I²S 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 4-44.

Table 4-44. 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 = I²S 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 4-45.

Table 4-45. 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) AIF2_Frame_Ctrl_3	5:0	AIF2TX1_SLOT[5:0]	0x0	AIF2 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

Table 4-45. AIF2 Digital Audio Data Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R1354 (0x054A) AIF2_Frame_Ctrl_4	5:0	AIF2TX2_SLOT[5:0]	0x1	AIF2 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
R1361 (0x0551) AIF2_Frame_Ctrl_11	5:0	AIF2RX1_SLOT[5:0]	0x0	AIF2 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
R1362 (0x0552) AIF2_Frame_Ctrl_12	5:0	AIF2RX2_SLOT[5:0]	0x1	AIF2 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

The AIF3 data control fields are described in Table 4-46.

Table 4-46. 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
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 4.14 to configure the GPIO pins.

Under default conditions, the AIF*n*TXDAT output is held at Logic 0 when the CS47L35 is not transmitting data (i.e., during time slots that are not enabled for output by the CS47L35). If the AIF*n*TX_DAT_TRI bit is set, the CS47L35 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 4-47.

Table 4-47. 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 4-48.

Table 4-48. 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 4-49.

Table 4-49. 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 SLIMbus Interface

The SLIMbus protocol is highly configurable and adaptable, supporting multiple audio signal paths, and mixed sample rates simultaneously. It also supports control messaging and associated communications between devices.

4.8.1 SLIMbus Devices

The SLIMbus components comprise different device classes (manager, framer, interface, generic). Each component on the bus has an interface device, which provides bus management services for the respective component. One or more components on the bus provide manager and framer device functions; the manager has the capabilities to administer the bus, although the framer is responsible for driving the CLK line and for driving the DATA required to establish the frame structure on the bus. Note that only one manager and one framer device is active at any time. The framer function can be transferred between devices when required. Generic devices provide the basic SLIMbus functionality for the associated ports, and for the transport protocol by which audio signal paths are established on the bus.

4.8.2 SLIMbus Frame Structure

The SLIMbus bit stream is formatted within a defined structure of cells, slots, subframes, frames, and superframes:

- A single data bit is known as a cell.
- Four cells make a slot.

- A frame consists of 192 slots.
- Eight frames make a superframe.

The bit stream structure is configurable to some extent, but the superframe definition always comprises 1536 slots. The transmitted/received bit rate can be configured according to system requirements and can be changed dynamically without interruption to active audio paths.

The SLIMbus CLK frequency (also the bus bit rate) is defined by a root frequency (RF) and a clock gear (CG). In the top clock gear (Gear 10), the CLK frequency is equal to the root frequency. Each reduction in the clock gear halves the CLK frequency, and doubles the duration of the superframe.

The SLIMbus bandwidth typically comprises control space (for bus messages, synchronization, etc.) and data space (for audio paths). The precise allocation is configurable and can be entirely control space, if required.

The subframe definition comprises the number of slots per subframe (6, 8, 24 or 32 slots) and the number of these slots per subframe allocated as control space. The applicable combination of subframe length and control space width are defined by the Subframe Mode (SM) parameter.

The SLIMbus frame always comprises 192 slots, regardless of the subframe definition. A number of slots are allocated to control space, as noted above; the remaining slots are allocated to data space. Some of the control space is required for framing information and for the guide channel (see Section 4.8.3); the remainder of the control space are allocated to the message channel.

4.8.3 Control Space

Framing information is provided in slots 0 and 96 of every frame. Slot 0 contains a 4-bit synchronization code; slot 96 contains the 32-bit framing information, transmitted 4 bits at a time over the eight frames that make up the SLIMbus superframe. The clock gear, root frequency, subframe configuration, along with some other parameters, are encoded within the framing information.

The guide channel occupies two slots within Frame 0. This provides the necessary information for a SLIMbus component to acquire and verify the frame synchronization. The guide channel occupies the first two control space slots within the first frame of the bit stream, excluding the framing information slots. Note that the exact slot allocation depends upon the applicable subframe mode.

The message channel is allocated all of the control space not used by the framing information or the guide channel. The message channel enables SLIMbus devices to communicate with each other, using a priority-based mechanism defined in the MIPI specification.

Messages may be broadcast to all devices on the bus, or can be addressed to specific devices using their allocated logical address (LA) or enumeration address (EA). Note that, device-specific messages are directed to a particular device (i.e., manager, framer, interface, or generic) within a component on the bus.

4.8.4 Data Space

The data space can be organized into a maximum of 256 data channels. Each channel, identified by a unique channel number (CN), is a stream of one or more contiguous slots, organized in a consistent data structure that repeats at a fixed interval.

A data channel is defined by its segment length (SL, number of contiguous slots allocated), segment interval (spacing between the first slots of successive segments), and segment offset (the slot number of the first allocated slot within the superframe). The segment interval and segment offset are collectively defined by a segment distribution (SD), by which the SLIMbus manager may configure or reconfigure any data channel.

Each segment may comprise TAG, AUX, and DATA portions. Any of these portions may have a length of zero; the exact composition depends on the transport protocol (TP) for the associated channel. The DATA portion must be wide enough to accommodate one full word of the data channel contents. Data words cannot be spread across multiple segments.

The segment interval for each data channel represents the minimum spacing between consecutive data samples for that channel. (Note that the minimum spacing applies if every allocated segment is populated with new data; in many cases, additional bandwidth is allocated and not every allocated segment is used.)

The segment interval gives rise to segment windows for each data channel, aligned to the start of every superframe. The segment window boundaries define the times within which each new data sample must be buffered, ready for transmission—adherence to these fixed boundaries allows slot allocations to be moved within a segment window, without altering the signal latency. The segment interval may be either shorter or longer than the frame length, but there is always an integer number of segment windows per superframe.

The TP defines the flow control or handshaking method used by the ports associated with a data channel. The applicable flow control modes depend on the relationship between the audio sample rate (flow rate) and the SLIMbus CLK frequency. If the two rates are synchronized and integer related, no flow control is needed. In other cases, the flow may be regulated by the use of a presence bit, which can be set by the source device (pushed protocol), or by the sink device (pulled protocol).

The data-channel structure is defined in terms of the TP, SD, and SL parameters.

The data-channel content definition includes a presence rate (PR) parameter (describing the nominal sample rate for the audio channel) and a frequency locked (FL) bit (identifying whether the data source is synchronized to the SLIMbus CLK). The data length (DL) parameter defines the size of each data sample (number of slots). The auxiliary bits format (AF) and data type (DT) parameters provide support for non-PCM encoded data channels; the channel link (CL) parameter is an indicator that channel CN is related to the previous channel, CN-1.

For a given root frequency and clock gear, the SL and SD parameters define the amount of SLIMbus bandwidth that is allocated to a given data channel. The minimum bandwidth requirements of a data channel are represented by the presence rate (PR) and data length (DL) parameters. The allocated SLIMbus bandwidth must be equal to or greater than the bandwidth of the data to be transferred.

The segment interval defines the repetition rate of the SLIMbus slots allocated to consecutive data samples for a given data channel. The presence rate (PR) is the nominal sample rate of the audio path. The segment rate (determined by the segment interval value) must be equal to or greater than the presence rate for a given data channel. The following constraints must be observed when configuring a SLIMbus channel:

- If pushed or pulled transport protocol is selected, the segment rate must be greater than the presence rate to ensure that samples are not dropped as a result of clock drift.
- If isochronous transport protocol is selected, the segment rate must be equal to the presence rate. Isochronous transport protocol should be selected only if the data source is frequency locked to the SLIMbus CLK (i.e., the data source is synchronized to the SLIMbus framer device).

4.9 SLIMbus Control Sequences

This section describes the messages and general protocol associated with the SLIMbus system.

Note: The SLIMbus specification permits flexibility in core message support for different components. See Section 4.10 for details regarding which messages are supported on each of the SLIMbus devices present on the CS47L35.

4.9.1 Device Management and Configuration

This section describes the SLIMbus messages associated with configuring all devices on the SLIMbus interface.

When the SLIMbus interface starts up, it is required that only one component provides the manager and framer device functions. Other devices can request connection to the bus after they have gained synchronization.

The REPORT_PRESENT (DC, DCV) message may be issued by devices attempting to connect to the bus. The payload of this message contains the device class (DC) and device class version (DCV) parameters, describing the type of device that is attempting to connect. This message may be issued autonomously by the connecting device, or else in response to a REQUEST_SELF_ANNOUNCEMENT message from the manager device.

After positively acknowledging the REPORT_PRESENT message, the manager device then issues the ASSIGN_LOGICAL_ADDRESS (LA) message to allow the other device to connect to the bus. The payload of this message contains the logical address (LA) parameter only; this is the unique address by which the connected device sends and receives SLIMbus messages. The device is then said to be enumerated.

Once a device has been successfully connected to the bus, the logical address (LA) parameter can be changed at any time using the CHANGE_LOGICAL_ADDRESS (LA) message.

The RESET_DEVICE message commands an individual SLIMbus device to perform its reset procedure. As part of the reset, all associated ports are reset, and any associated data channels are canceled. Note that, if the RESET_DEVICE command is issued to an interface device, it causes a component reset (i.e., all devices within the associated component are reset). Under a component reset, every associated device releases its logical address, and the component becomes disconnected from the bus.

4.9.2 Information Management

A memory map of information elements is defined for each device. This is arranged in 3 x 1-kB blocks, comprising core value elements, device class-specific value elements, and user value elements respectively, as described in the MIPI specification. Note that the contents of the user information portion for each CS47L35 SLIMbus device are reserved.

Read/write access is implemented using the messages described as follows. Specific elements within the information map are identified using the element code (EC) parameter. In the case of read access, a unique transaction ID (TID) is assigned to each message relating to a particular read/write request.

- The REQUEST_INFORMATION (TID, EC) message is used to instruct a device to respond with the indicated information. The payload of this message contains the transaction ID (TID) and the element code (EC).
- The REQUEST_CLEAR_INFORMATION (TID, EC, CM) message is used to instruct a device to respond with the indicated information, and also to clear all, or parts, of the same information slice. The payload of this message contains the transaction ID (TID), element code (EC), and clear mask (CM). The clear mask field is used to select which elements are to be cleared as part of the instruction.
- The REPLY_INFORMATION (TID, IS) message is used to provide output of a requested parameter. The payload of this message contains the transaction ID (TID) and the information slice (IS). The information slice bytes contain the value of the requested parameter.
- The CLEAR_INFORMATION (EC, CM) message is used to clear all, or parts, of the indicated information slice. The payload of this message contains the element code (EC) and clear mask (CM). The clear mask field is used to select which elements are to be cleared as part of the instruction.
- The REPORT_INFORMATION (EC, IS) message is used to inform other devices about a change in a specified element in the information map. The payload of this message contains the element code (EC) and the information slice (IS). The information slice bytes contain the new value of the applicable parameter.

4.9.3 Value Management (Including Register Access)

A memory map of value elements is defined for each device. This is arranged in 3 x 1-kB blocks, comprising core value elements, device class-specific value elements, and user value elements respectively, as described in the MIPI specification. These elements are typically parameters used to configure device behavior.

The user value elements of the interface device are used on CS47L35 to support read/write access to the register map. Details of how to access specific registers are described in Section 4.10.

Note that, with the exception of the user value elements of the interface device, the contents of the user value portion for each CS47L35 SLIMbus device are reserved.

Read/write access is implemented using the messages described as follows. Specific elements within the value map are identified using the element code (EC) parameter. In the case of read access, a unique transaction ID (TID) is assigned to each message relating to a particular read/write request.

- The REQUEST_VALUE (TID, EC) message is used to instruct a device to respond with the indicated information. The payload of this message contains the transaction ID (TID) and the element code (EC).
- The REPLY_VALUE (TID, VS) message is used to provide output of a requested parameter. The payload of this message contains the transaction ID (TID) and the value slice (VS). The value slice bytes contain the value of the requested parameter.

- The CHANGE_VALUE (EC, VU) message is used to write data to a specified element in the value map. The payload of this message contains the element code (EC) and the value update (VU). The value update bytes contain the new value of the applicable parameter.

4.9.4 Frame and Clocking Management

This section describes the SLIMbus messages associated with changing the frame or clocking configuration. One or more configuration messages may be issued as part of a reconfiguration sequence; all of the updated parameters become active at once, when the reconfiguration boundary is reached.

- The BEGIN_RECONFIGURATION message is issued to define a reconfiguration boundary point: subsequent NEXT_* messages become active at the first valid superframe boundary following receipt of the RECONFIGURE_NOW message. (A valid boundary must be at least two slots after the end of the RECONFIGURE_NOW message.) Both of these messages have no payload content.
- The NEXT_ACTIVE_FRAMER (LAIF, NCo, NCi) message is used to select a new device as the active framer. The payload of this message includes the logical address, incoming framer (LAIF). Two other fields (NCo, NCi) define the number of clock cycles for which the CLK line shall be inactive during the handover.
- The NEXT_SUBFRAME_MODE (SM) and NEXT_CLOCK_GEAR (CG) messages are used to reconfigure the SLIMbus clocking or framing definition. The payload of each is the respective subframe mode (SM) or clock gear (CG) respectively.
- The NEXT_PAUSE_CLOCK (RT) message instructs the active framer to pause the bus. The payload of the message contains the restart time (RT), which indicates whether the interruption is to be of a specified time and/or phase duration.
- The NEXT_RESET_BUS message instructs all components on the bus to be reset. In this case, all devices on the bus are reset and are disconnected from the bus. Subsequent reconnection to the bus follows the same process as when the bus is first initialized.
- The NEXT_SHUTDOWN_BUS message instructs all devices that the bus is to be shut down.

4.9.5 Data Channel Configuration

This section describes the SLIMbus messages associated with configuring a SLIMbus data channel. Note that the manager device is responsible for allocating the available bandwidth as required for each data channel.

- The CONNECT_SOURCE (PN, CN) and CONNECT_SINK (PN, CN) messages are issued to the respective devices, defining the ports between which a data channel is to be established. Note that multiple destinations (sinks) can be configured for a channel, if required. The payload of each message contains the port number (PN) and the channel number (CN) parameters.
- The BEGIN_RECONFIGURATION message is issued to define a Reconfiguration Boundary point: subsequent NEXT_* messages become active at the first valid superframe boundary following receipt of the RECONFIGURE_NOW message. A valid boundary must be at least two slots after the end of the RECONFIGURE_NOW message.
- The NEXT_DEFINE_CHANNEL (CN, TP, SD, SL) message informs the associated devices of the structure of the data channel. The payload of this message contains the channel number (CN), TP, SD, and SL parameters for the data channel.
- The NEXT_DEFINE_CONTENT (CN, FL, PR, AF, DT, CL, DL), or CHANGE_CONTENT (CN, FL, PR, AF, DT, CL, DL) message provides more detailed information about the data channel contents. The payload of this message contains the channel number (CN), frequency locked (FL), presence rate (PR), auxiliary bits format (AF), data type (DT), channel link (CL), and data length (DL) parameters.
- The NEXT_ACTIVATE_CHANNEL (CN) message instructs the channel to be activated at the next reconfiguration boundary. The payload of this message contains the channel number (CN) only.
- The RECONFIGURE_NOW message completes the reconfiguration sequence, causing all of the NEXT_ messages since the BEGIN_RECONFIGURATION to become active at the next valid superframe boundary. (A valid boundary must be at least two slots after the end of the RECONFIGURE_NOW message.)

- Active channels can be reconfigured using the CHANGE_CONTENT, NEXT_DEFINE_CONTENT, or NEXT_DEFINE_CHANNEL messages. Note that these changes can be effected without interrupting the data channel; the NEXT_DEFINE_CHANNEL, for example, may be used to change a segment distribution, in order to reallocate the SLIMbus bandwidth.
- An active channel can be paused using the NEXT_DEACTIVATE_CHANNEL message and reinstated using the NEXT_ACTIVATE_CHANNEL message.
- Data channels can be disconnected using the DISCONNECT_PORT or NEXT_REMOVE_CHANNEL messages. These messages provide equivalent functionality, but use different parameters (PN or CN respectively) to identify the affected signal path.

4.10 SLIMbus Interface Control

The CS47L35 features a MIPI-compliant SLIMbus interface, providing six channels of audio input and six channels of audio output. Mixed audio sample rates are supported on the SLIMbus interface. The SLIMbus interface also supports read/write access to the CS47L35 control registers.

The SLIMbus interface on CS47L35 comprises a generic device and an interface device. A maximum of 12 ports can be configured, providing up to six input (RX) channels and up to six output (TX) channels.

The audio paths associated with the SLIMbus interface are described in Section 4.3.

The SLIMbus interface supports read/write access to the CS47L35 control registers, as described in Section 4.10.6.

The SLIMbus clocking rate and channel allocations are controlled by the manager device. The message channel and data channel bandwidth may be dynamically adjusted according to the application requirements. Note that the manager device functions are not implemented on the CS47L35, and these bandwidth allocation requirements are outside the scope of this data sheet.

4.10.1 SLIMbus Device Parameters

The SLIMbus interface on the CS47L35 comprises two devices. The enumeration address of each device within the SLIMbus interface is derived from the parameters noted in Table 4-50.

Table 4-50. SLIMbus Device Parameters

Description	Manufacturer ID	Product Code	Device ID	Instance Value	Enumeration Address
Generic	0x01FA	0x6360	0x00	0x00	01FA_6360_0000
Interface	0x01FA	0x6360	0x7F	0x00	01FA_6360_7F00

4.10.2 SLIMbus Message Support

The SLIMbus interface on the CS47L35 supports bus messages as noted in Table 4-51.

Additional notes regarding SLIMbus message support are noted below, and also in Table 4-52.

Table 4-51. SLIMbus Message Support

Category	Message Code MC[6:0]	Description	Generic	Interface
Device Management Messages	0x01	REPORT_PRESENT (DC, DCV)	S	S
	0x02	ASSIGN_LOGICAL_ADDRESS (LA)	D	D
	0x04	RESET_DEVICE ()	D	D
	0x08	CHANGE_LOGICAL_ADDRESS (LA)	D	D
	0x09	CHANGE_ARBITRATION_PRIORITY (AP)	—	—
	0x0C	REQUEST_SELF_ANNOUNCEMENT ()	D	D
	0x0F	REPORT_ABSENT ()	—	—
Data Channel Management Messages	0x10	CONNECT_SOURCE (PN, CN)	D	—
	0x11	CONNECT_SINK (PN, CN)	D	—
	0x14	DISCONNECT_PORT (PN)	D	—
	0x18	CHANGE_CONTENT (CN, FL, PR, AF, DT, CL, DL)	D	—

Table 4-51. SLIMbus Message Support *(Cont.)*

Category	Message Code MC[6:0]	Description	Generic	Interface
Information Management Messages	0x20	REQUEST_INFORMATION (TID, EC)	D	D
	0x21	REQUEST_CLEAR_INFORMATION (TID, EC, CM)	D	D
	0x24	REPLY_INFORMATION (TID, IS)	S	S
	0x28	CLEAR_INFORMATION (EC, CM)	D	D
	0x29	REPORT_INFORMATION (EC, IS)	—	S
Reconfiguration Messages	0x40	BEGIN_RECONFIGURATION ()	D	D
	0x44	NEXT_ACTIVE_FRAMER (LAIF, NCo, NCi)	—	—
	0x45	NEXT_SUBFRAME_MODE (SM)	—	D
	0x46	NEXT_CLOCK_GEAR (CG)	—	—
	0x47	NEXT_ROOT_FREQUENCY (RF)	—	—
	0x4A	NEXT_PAUSE_CLOCK (RT)	—	—
	0x4B	NEXT_RESET_BUS ()	—	—
	0x4C	NEXT_SHUTDOWN_BUS ()	—	—
	0x50	NEXT_DEFINE_CHANNEL (CN, TP, SD, SL)	D	—
	0x51	NEXT_DEFINE_CONTENT (CN, FL, PR, AF, DT, CL, DL)	D	—
	0x54	NEXT_ACTIVATE_CHANNEL (CN)	D	—
	0x55	NEXT_DEACTIVATE_CHANNEL (CN)	D	—
	0x58	NEXT_REMOVE_CHANNEL (CN)	D	—
	0x5F	RECONFIGURE_NOW ()	D	D
Value Management Messages	0x60	REQUEST_VALUE (TID, EC)	—	D
	0x61	REQUEST_CHANGE_VALUE (TID, EC, VU)	—	—
	0x64	REPLY_VALUE (TID, VS)	—	S
	0x68	CHANGE_VALUE (EC, VU)	—	D

Notes:
- S = Supported as a source device only.
- D = Supported as a destination device only.

The CS47L35 SLIMbus component must be reset prior to scheduling a hardware reset or power-on reset. This can be achieved using the RESET_DEVICE message (issued to the CS47L35 interface device), or else using the NEXT_RESET_BUS message.

Table 4-52. SLIMbus Parameter Support

Parameter Code	Description	Comments
AF	Auxiliary Bits Format	—
CG	Clock Gear	—
CL	Channel Link	—
CM	Clear Mask	CS47L35 does not fully support this function. The CM bytes of the REQUEST_CLEAR_INFORMATION or CLEAR_INFORMATION messages must not be sent to CS47L35 Devices. When either of these messages is received, all bits within the specified Information Slice are cleared.
CN	Channel Number	—
DC	Device Class	—
DCV	Device Class Variation	—
DL	Data Length	—
DT	Data Type	CS47L35 supports the following DT codes: 0x0 = Not indicated 0x1 = LPCM audio Note that 2's complement PCM can be supported with DT = 0x0.
EC	Element Code	—
FL	Frequency Locked	—
IS	Information Slice	—
LA	Logical Address	—
LAIF	Logical Address, Incoming Framer	—
NCi	Number of Incoming Framer Clock Cycles	—
NCo	Number of Outgoing Framer Clock Cycles	—
PN	Port Number	Note that the Port Numbers of the CS47L35 SLIMbus paths are register-configurable, as described in Table 4-53.

Table 4-52. SLIMbus Parameter Support *(Cont.)*

Parameter Code	Description	Comments
PR	Presence Rate	Note that the Presence Rate must be the same as the sample rate selected for the associated CS47L35 SLIMbus path.
RF	Root Frequency	—
RT	Restart Time	CS47L35 supports the following RT codes: 0x0 = Fast Recovery 0x2 = Unspecified Delay When either of these values is specified, the CS47L35 resumes toggling the CLK line within four cycles of the CLK line frequency.
SD	Segment Distribution	Note that any data channels that are assigned the same SAMPLE_RATE_*n* value must also be assigned the same Segment Interval.
SL	Segment Length	—
SM	Subframe Mode	—
TID	Transaction ID	—
TP	Transport Protocol	CS47L35 supports the following TP codes for TX channels: 0x0 = Isochronous Protocol 0x1 = Pushed Protocol CS47L35 supports the following TP codes for RX channels: 0x0 = Isochronous Protocol 0x2 = Pulled Protocol
VS	Value Slice	—
VU	Value Update	—

4.10.3 SLIMbus Port Number Control

The CS47L35 SLIMbus interface supports up to six input (RX) channels and up to six output (TX) channels. The SLIMbus port numbers for these audio channels are configurable using the fields described in Table 4-53.

Table 4-53. SLIMbus Port Number Control

Register Address	Bit	Label	Default	Description
R1490 (0x05D2)	13:8	SLIMRX2_PORT_ADDR[5:0]	1	SLIMbus RX Channel n Port number
SLIMbus_RX_Ports0	5:0	SLIMRX1_PORT_ADDR[5:0]	0	Valid from 0–31
R1491 (0x05D3)	13:8	SLIMRX4_PORT_ADDR[5:0]	3	
SLIMbus_RX_Ports1	5:0	SLIMRX3_PORT_ADDR[5:0]	2	
R1492 (0x05D4)	13:8	SLIMRX6_PORT_ADDR[5:0]	5	
SLIMbus_RX_Ports2	5:0	SLIMRX5_PORT_ADDR[5:0]	4	
R1494 (0x05D6)	13:8	SLIMTX2_PORT_ADDR[5:0]	9	SLIMbus TX Channel n Port number
SLIMbus_TX_Ports0	5:0	SLIMTX1_PORT_ADDR[5:0]	8	Valid from 0–31
R1495 (0x05D7)	13:8	SLIMTX4_PORT_ADDR[5:0]	11	
SLIMbus_TX_Ports1	5:0	SLIMTX3_PORT_ADDR[5:0]	10	
R1496 (0x05D8)	13:8	SLIMTX6_PORT_ADDR[5:0]	13	
SLIMbus_TX_Ports2	5:0	SLIMTX5_PORT_ADDR[5:0]	12	

4.10.4 SLIMbus Sample-Rate Control

The SLIMbus RX inputs may be selected as input to the digital mixers or signal-processing functions within the CS47L35 digital core. The SLIMbus TX outputs are derived from the respective output mixers.

The sample rate for each SLIMbus channel is configured using SLIMRX*n*_RATE and SLIMTX*n*_RATE—see Table 4-21 within the Digital Core section.

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

4.10.5 SLIMbus Signal Path Enable

The SLIMbus interface supports up to six input (RX) channels and up to six output (TX) channels. Each of these channels can be enabled or disabled using the fields defined in Table 4-54.

Note: SLIMbus audio channels can be supported only when the corresponding ports have been enabled by the manager device (i.e., in addition to setting the respective enable bits). The status bits in Registers R1527 and R1528 indicate the status of each of the SLIMbus ports.

The system clock, SYSCLK, must be configured and enabled before any audio path is enabled. The SLIMbus audio paths should be kept disabled (SLIMRX*n*_ENA = 0, SLIMTX*n*_ENA = 0) if SYSCLK is not enabled. See Section 4.16 for details of the system clocks (including requirements for reconfiguring SYSCLK while audio paths are enabled).

Table 4-54. SLIMbus Signal Path Enable

Register Address	Bit	Label	Default	Description
R1525 (0x05F5) SLIMbus_RX_ Channel_Enable	5	SLIMRX6_ENA	0	SLIMbus RX Channel n Enable 0 = Disabled 1 = Enabled
	4	SLIMRX5_ENA	0	
	3	SLIMRX4_ENA	0	
	2	SLIMRX3_ENA	0	
	1	SLIMRX2_ENA	0	
	0	SLIMRX1_ENA	0	
R1526 (0x05F6) SLIMbus_TX_ Channel_Enable	5	SLIMTX6_ENA	0	SLIMbus TX Channel n Enable 0 = Disabled 1 = Enabled
	4	SLIMTX5_ENA	0	
	3	SLIMTX4_ENA	0	
	2	SLIMTX3_ENA	0	
	1	SLIMTX2_ENA	0	
	0	SLIMTX1_ENA	0	
R1527 (0x05F7) SLIMbus_RX_ Port_Status	5	SLIMRX6_PORT_STS	0	SLIMbus RX Channel n Port Status (Read only) 0 = Disabled 1 = Configured and active
	4	SLIMRX5_PORT_STS	0	
	3	SLIMRX4_PORT_STS	0	
	2	SLIMRX3_PORT_STS	0	
	1	SLIMRX2_PORT_STS	0	
	0	SLIMRX1_PORT_STS	0	
R1528 (0x05F8) SLIMbus_TX_ Port_Status	5	SLIMTX6_PORT_STS	0	SLIMbus TX Channel n Port Status (Read only) 0 = Disabled 1 = Configured and active
	4	SLIMTX5_PORT_STS	0	
	3	SLIMTX4_PORT_STS	0	
	2	SLIMTX3_PORT_STS	0	
	1	SLIMTX2_PORT_STS	0	
	0	SLIMTX1_PORT_STS	0	

4.10.6 SLIMbus Control Register Access

Control register access is supported via the SLIMbus interface. Full read/write access to all registers is possible, via the user value elements portion of the value map.

If the SLIMbus interface is used to access the DSP firmware memory registers, a system clocking constraint must be observed: the DSPCLK frequency, if enabled, must be greater than 1.3 x RF, where RF is the SLIMbus root frequency. Note that, if DSPCLK is disabled (DSP_CLK_ENA = 0), or if accessing other areas of the register map, the timing constraint is not applicable. See Section 4.4 for details of the DSP Firmware memory. See Section 4.16 for details of the DSPCLK signal.

Register write operations are implemented using the CHANGE_VALUE message. A maximum of two messages may be required, depending on circumstances: the first CHANGE_VALUE message selects the register page (bits [23:8] of the control register address); the second message contains the data and bits [7:0] of the register address. The first message may be omitted if the register page is unchanged from the previous read or write operation.

The required SLIMbus parameters are described in Table 4-55 and Table 4-56, for the generic case of writing the value 0xVVVV to control register address 0xYYYYZZ. Note that it is also possible to write blocks of up to 16 bytes (to consecutive register addresses), as described below.

Table 4-55. Register Write Message (1)—CHANGE_VALUE

Parameter	Value	Description
Source Address	0xSS	SS is the 8-bit logical address of the message source. This could be any active device on the bus, but is typically the manager device (0xFF).
Destination Address	0xLL	LL is the 8-bit logical address of the message destination (i.e., the CS47L35 SLIMbus interface device). The value is assigned by the SLIMbus manager device.
Access Mode	0b1	Selects byte-based access mode.
Byte Address	0x800	Identifies the user value element for selecting the control register page address.
Slice Size	0b001	Selects 2-byte slice size
Value Update	0xYYYY	YYYY is bits [23:8] of the applicable control register address.

Table 4-56. Register Write Message (2)—CHANGE_VALUE

Parameter	Value	Description
Source Address	0xSS	SS is the 8-bit logical address of the message source. This could be any active device on the bus, but is typically the manager device (0xFF).
Destination Address	0xLL	LL is the 8-bit logical address of the message destination (i.e., the CS47L35 SLIMbus interface device). The value is assigned by the SLIMbus manager device.
Access Mode	0b1	Selects byte-based access mode.
Byte Address	0xUUU	Specifies the value map address, calculated as 0xA00 + (2 x 0xZZ), where ZZ is bits [7:0] of the applicable control register address.
Slice Size	0b001	Selects 2-byte slice size
Value Update	0xVVVV	VVVV is the 16-bit data to be written.

Note: The first message may be omitted if its contents are unchanged from the previous CHANGE_VALUE message sent to the CS47L35.

Write transfers of up to 16 bytes can be configured using the slice size parameter in the second message (see Table 4-56). Additional value update words can be appended to the message in this case, with the applicable data contents. For compatibility with the CS47L35 register map, the selected number of bytes must always be an even number.

When a 2-byte transfer is selected, these bytes are written to the register address 0xYYYYZZ (using the same naming conventions as above). When more than 2 bytes are written in a single transfer, the destination register address is automatically incremented as described in Table 4-57.

Table 4-57. SLIMbus Register Write Sequence—16-Bit Register Space (< 0x3000)

Register Address (<0x3000)	Byte Sequence
Base Address (0xYYYYZZ)	Bytes 2 and 1 (0xVVVV)
Base address + 1	Bytes 4 and 3
Base address + 2	Bytes 6 and 5
Base address + 3	Bytes 8 and 7
Base address + 4	Bytes 10 and 9
Base address + 5	Bytes 12 and 11
Base address + 6	Bytes 14 and 13
Base address + 7	Bytes 16 and 15

Note: Register addresses from R12288 (0x3000) upwards are formatted as 32-bit words. When writing to these addresses, the slice size should be a multiple of 4 bytes and the byte address should be aligned with the 32-bit data word boundaries (i.e., an even number). The byte ordering for these register addresses is described in Table 4-58.

Table 4-58. SLIMbus Register Write Sequence—32-Bit Register Space (≥ 0x3000)

Register Address (≥0x3000)	Byte Sequence
Base Address (0xYYYYZZ)	Bytes 4, 3, 2, 1
Base address + 2	Bytes 8, 7, 6, 5
Base Address + 4	Bytes 12, 11, 10, 9
Base Address + 6	Bytes 16, 15, 14, 13

Register read operations are implemented using the CHANGE_VALUE and REQUEST_VALUE messages. A maximum of two messages may be required, depending on circumstances: the CHANGE_VALUE message selects the register page (bits [23:8] of the control register address); the REQUEST_VALUE message contains bits [7:0] of the register address. The first message may be omitted if the register page is unchanged from the previous read or write operation.

The required SLIMbus parameters are described in Table 4-59 and Table 4-60, for the generic case of reading the contents of control register address 0xYYYYZZ.

Table 4-59. Register Read Message (1)—CHANGE_VALUE

Parameter	Value	Description
Source Address	0xSS	SS is the 8-bit logical address of the message source. This could be any active device on the bus, but is typically the manager device (0xFF).
Destination Address	0xLL	LL is the 8-bit logical address of the message destination (i.e., the CS47L35 SLIMbus interface device). The value is assigned by the SLIMbus manager device.
Access Mode	0b1	Selects byte-based access mode.
Byte Address	0x800	Identifies the user value element for selecting the control register page address.
Slice Size	0b001	Selects 2-byte slice size
Value Update	0xYYYY	YYYY is bits [23:8] of the applicable control register address.

The CS47L35 SLIMbus interface supports register read operations of 2 bytes (i.e., one 16-bit data word) only. Register addresses from R12288 (0x3000) upwards are formatted as 32-bit words; when reading from these addresses, the 2-byte data slice represents the 2 lower bytes of the selected 32-bit word. The 2 upper bytes of the respective register can be accessed by adding ‘2’ to the Byte Address value described in Table 4-60.

Table 4-60. Register Read Message (2)—REQUEST_VALUE

Parameter	Value	Description
Source Address	0xSS	SS is the 8-bit logical address of the message source. This could be any active device on the bus, but is typically the manager device (0xFF).
Destination Address	0xLL	LL is the 8-bit logical address of the message destination (i.e., the CS47L35 SLIMbus interface device). The value is assigned by the SLIMbus manager device.
Access Mode	0b1	Selects byte-based access mode.
Byte Address	0xUUU	Specifies the value map address, calculated as 0xA00 + (2 x 0xZZ), where ZZ is bits [7:0] of the applicable control register address.
Slice Size	0b001	Selects 2-byte slice size
Transaction ID	0xTTTT	TTTT is the 16-bit transaction ID for the message. The value is assigned by the SLIMbus manager device.

Note: The first message may be omitted if its contents are unchanged from the previous CHANGE_VALUE message sent to the CS47L35.

The CS47L35 responds to the register read commands in accordance with the normal SLIMbus protocols.

Note that the CS47L35 assumes that sufficient control space slots are available in which to provide its response before the next REQUEST_VALUE message is received. The CS47L35 response is made using a REPLY_VALUE message; the SLIMbus manager should wait until the REPLY_VALUE message has been received before sending the next REQUEST_VALUE message. If additional REQUEST_VALUE messages are received before the CS47L35 response has been made, the earlier REQUEST_VALUE messages are ignored (i.e., only the last REQUEST_VALUE message is serviced).

4.10.7 SLIMbus Clocking Control

The clock frequency of the SLIMbus interface is not fixed, and may be set according to the application requirements. The clock frequency can be reconfigured dynamically as required.

The CS47L35 SLIMbus interface does not include a framer device. Accordingly, the SLIMCLK pin is always an input pin on the CS47L35. The framer function can be transferred from one device to another; this is known as framer handover, and is controlled by the manager device.

The supported root frequencies are as defined in the MIPI Alliance specification for SLIMbus.

Under normal operating conditions, the SLIMbus interface operates with a fixed root frequency (RF); dynamic updates to the bus rate are applied using a selectable clock gear (CG) function. The root frequency and the clock gear setting are controlled by the manager device; these parameters are transmitted in every SLIMbus superframe to all devices on the bus.

In Gear 10 (the highest clock gear setting), the SLIMCLK input frequency is equal to the root frequency. In lower gears, the SLIMCLK frequency is reduced by increasing powers of 2.

The clock gear definition is shown in Table 4-61.

Note: The 24.576-MHz root frequency is an example only; other frequencies are also supported.

Table 4-61. SLIMbus Clock Gear Selection

Clock Gear	Description	SLIMCLK Frequency [1]
10	Divide by 1	24.576 MHz
9	Divide by 2	12.288 MHz
8	Divide by 4	6.144 MHz
7	Divide by 8	3.072 MHz
6	Divide by 16	1.536 MHz
5	Divide by 32	768 kHz
4	Divide by 64	384 kHz
3	Divide by 128	192 kHz
2	Divide by 256	96 kHz
1	Divide by 512	48 kHz

1.Assuming 24.576-MHz root frequency

The SLIMCLK input can be used to provide a reference source for the FLL. The frequency of this reference is controlled using SLIMCLK_REF_GEAR, as described in Table 4-62.

The input clock reference for the FLL is selected by using FLL1_REFCLK_SRC. If SLIMbus is selected as the clock source, the reference signal is generated using an adaptive divider on the SLIMCLK input. The divider automatically adapts to the SLIMbus clock gear (CG). If the clock gear on the bus is lower than the SLIMCLK_REF_GEAR, the selected reference frequency cannot be supported, and the SLIMbus clock reference is disabled.

See Section 4.16 for details of system clocking and the FLL.

Table 4-62. SLIMbus Clock Reference Control

Register Address	Bit	Label	Default	Description
R1507 (0x05E3) SLIMbus_Framer_Ref_Gear	3:0	SLIMCLK_REF_GEAR[3:0]	0x0	SLIMbus Clock Reference control. Sets the SLIMbus reference clock relative to the SLIMbus Root Frequency (RF). 0x0 = Clock stopped 0x4 = Gear 4. (RF/64) 0x8 = Gear 8. (RF/4) 0x1 = Gear 1. (RF/512) 0x5 = Gear 5. (RF/32) 0x9 = Gear 9. (RF/2) 0x2 = Gear 2. (RF/256) 0x6 = Gear 6. (RF/16) 0xA = Gear 10. (RF) 0x3 = Gear 3. (RF/128) 0x7 = Gear 7. (RF/8) All other codes are reserved

4.11 Output Signal Path

The CS47L35 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 4-63.

Table 4-63. 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	SPKDAT, 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. A total of two digital output channels are provided.

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 CS47L35 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 4.5.3) can also be used as a watchdog function, to trigger a shutdown of the Class D speaker drivers; see Section 4.21.

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 one of the DSP cores. 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 4.11.8.

The CS47L35 output signal paths are shown in Fig. 4-49.

The OUT2, OUT3, and OUT4R paths are not implemented on this device.

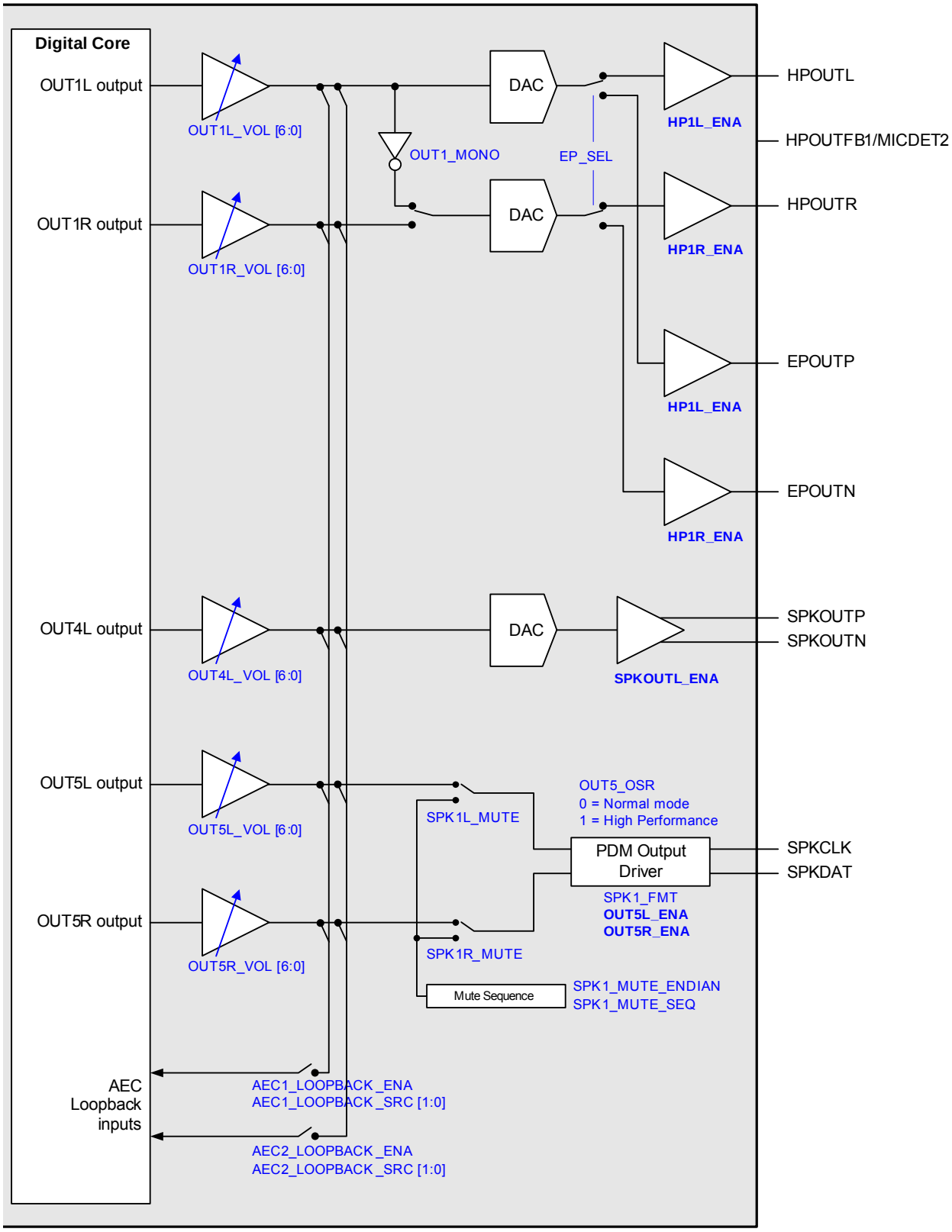


Figure 4-49. Output Signal Paths

4.11.1 Output Signal Path Enable

The output signal paths are enabled using the bits described in Table 4-64. 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 4-66 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 4-64.

The supply rails for the OUT1 outputs (HPOUT and EPOUT) are generated using an integrated dual-mode charge pump, CP1. The charge pump is enabled automatically by the CS47L35 when required by the output drivers; see Section 4.19.

The CS47L35 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 4.18 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 4.15.

The system clock, SYSCLK, must be configured and enabled before any audio path is enabled. The output signal paths should be kept disabled (HP1*x*_ENA = 0, SPKOUTL_ENA = 0, OUT5*x*_ENA = 0) if SYSCLK is not enabled. See Section 4.16 for details of the system clocks (including requirements for reconfiguring SYSCLK while audio paths are enabled).

The CS47L35 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 4-64. 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 4-64. 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

4.11.2 Output Signal Path Sample-Rate Control

The output signal paths are derived from the respective output mixers within the CS47L35 digital core. The sample rate for the output signal paths is configured using OUT_RATE—see Table 4-21.

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.

4.11.3 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 4-64.

The mono (BTL) signal paths are shown in Fig. 4-49. 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 4-64.

The SPKCLK frequency of the PDM output path (OUT5) is controlled by OUT5_OSR, as described in Table 4-65. 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 4-65 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 4-65. SPKCLK Frequency

Condition	SPKCLK Frequency
OUT5_OSR = 0	3.072 MHz
OUT5_OSR = 1	6.144 MHz

The output signal path control registers are defined in Table 4-66.

Table 4-66. 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

4.11.4 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 does not actually 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 CS47L35 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 4-67 and Table 4-68.

Table 4-67. 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 4-68 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 4-68 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 4-68 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_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	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 4-68 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 4-67. 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 4-68 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 4-68 lists the output signal path digital volume settings.

Table 4-68. 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 4-68. 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		

4.11.5 Output Signal Path Noise-Gate Control

The CS47L35 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 4-69.

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 CS47L35 gradually attenuates the respective signal path at the rate set by OUT_VD_RAMP (see Table 4-67). When the noise gate is deactivated, the output volume increases at the rate set by OUT_VI_RAMP.

Table 4-69. 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

4.11.6 Output Signal Path AEC Loop-Back

The CS47L35 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 CS47L35 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. 4-49. The AEC loop-back signals can be selected as input to any of the digital mixers within the CS47L35 digital core. The sample rate for the AEC loop-back paths is configured using OUT_RATE—see Table 4-21.

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 CS47L35 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 4-70. 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

Table 4-70. Output Signal Path AEC Loop-Back Control *(Cont.)*

Register Address	Bit	Label	Default	Description
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

4.11.7 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 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 outputs.

The ground feedback path for HPOUTL and HPOUTR is provided via the HPOUTFB1 or HPOUTFB2 pins; the applicable connection must be selected using ACCDET_SRC, as described in Table 4-71.

Note that the selected feedback pin should be connected to GND as close as possible to the respective headphone jack ground pin, as shown in Fig. 4-50. In mono (differential) mode, the feedback pin should be connected to the ground plane that is closest to the earpiece output PCB tracks.

Table 4-71. Headphone Output (HPOUT) Ground Feedback Control

Register Address	Bit	Label	Default	Description
R659 (0x0293) Accessory_ Detect_Mode_1	13	ACCDET_SRC	0	Accessory Detect/Headphone Feedback pin select 0 = Accessory detect on MICDET1; Headphone ground feedback on HPOUTFB1 1 = Accessory detect on MICDET2; Headphone ground feedback on HPOUTFB2

The headphone and earpiece connections are shown in Fig. 4-50.

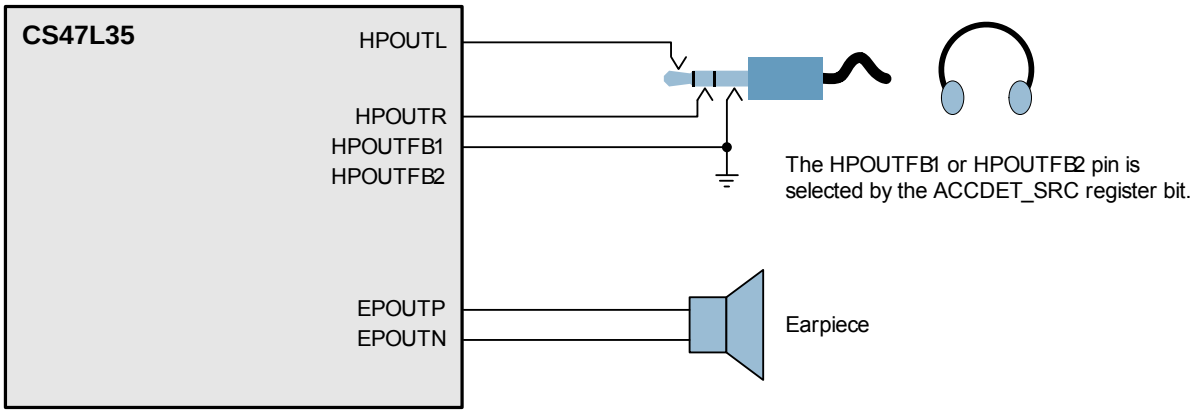


Figure 4-50. Headphone and Earpiece Connection

4.11.8 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 paths incorporate 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 4.16 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. 4-51, assuming a suitable speaker is chosen to provide the PWM filtering.

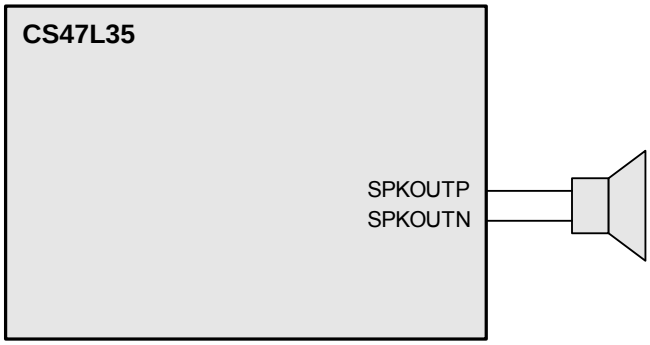


Figure 4-51. Speaker Connection

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 one of the integrated DSP cores, 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 paths, as an input to the speaker protection algorithms.

The external speaker connections, incorporating the output current monitoring requirements, are shown in Fig. 4-52. Note that, if output current monitoring is not required, these connections should be tied directly to ground on the PCB.

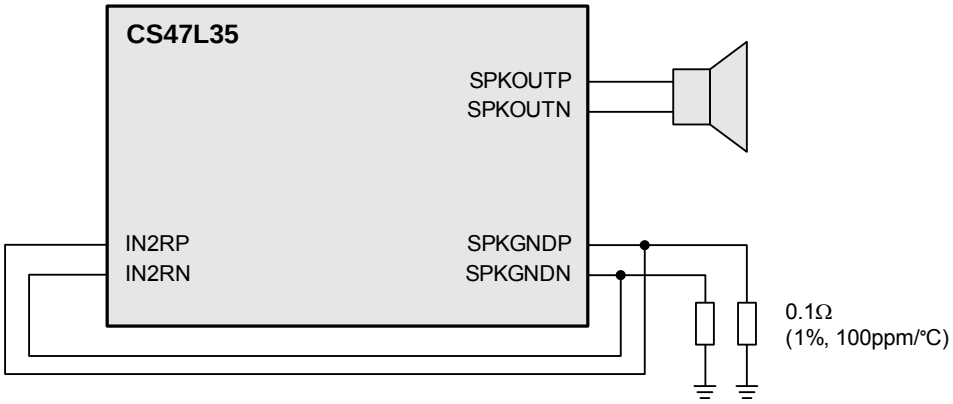


Figure 4-52. Speaker Output Current Monitoring Connections (Speaker Protection)

Please contact your Cirrus Logic representative for further information on the Speaker Protection software.

4.11.9 Speaker Outputs (Digital PDM)

The CS47L35 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 pin-specific alternative functions available on specific GPIO pins. See Section 4.14 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 SPKDAT 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. 4-53.

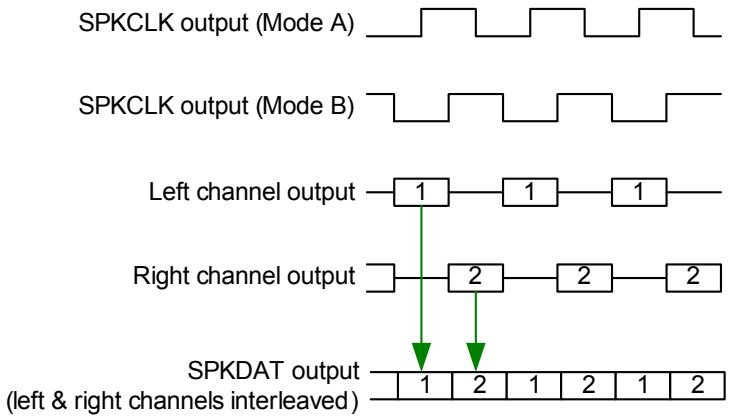


Figure 4-53. Digital Speaker (PDM) Interface Timing

Clocking for the PDM interface is derived from SYSCLK. Note that SYSCLK_ENA must also be set. See Section 4.16 for further details of the system clocks and control registers.

When 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 4.11.3. The SPKCLK frequency is set according to the operating mode of the relevant output path, as described in Table 4-72. The OUT5_OSR bit is defined in Table 4-66.

Note that the SPKCLK frequencies noted in Table 4-72 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 4-72. SPKCLK Frequency

OUT5_OSR	Description	SPKCLK Frequency
0	Normal mode	3.072 MHz
1	High Performance mode	6.144 MHz

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 4-67 for details of the OUT5L_MUTE and OUT5R_MUTE bits.

The PDM output interface registers are described in Table 4-73.

Table 4-73. 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 SPKDAT (left) or SPKDAT (right) when muted.
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 SPKDAT and SPKCLK are intended for direct connection to a compatible external speaker driver. A typical configuration is shown in Fig. 4-54.

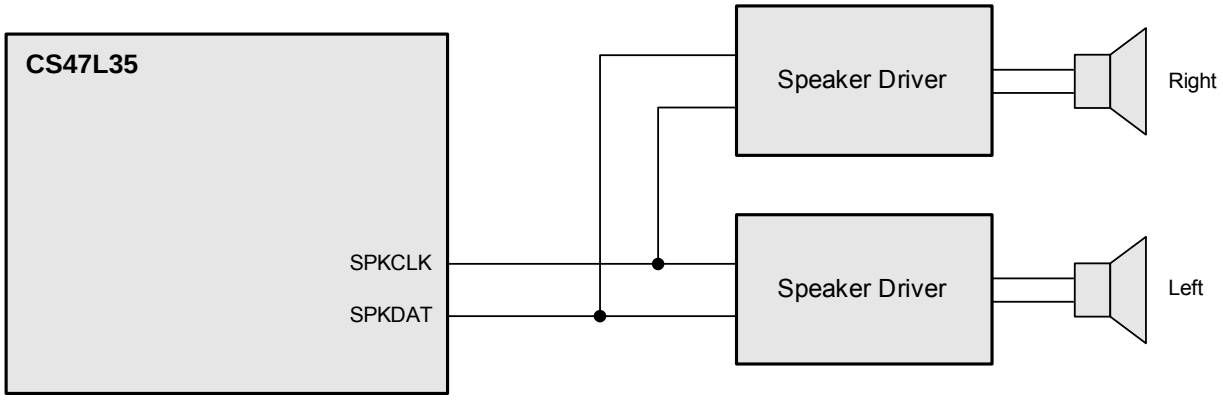


Figure 4-54. Digital Speaker (PDM) Connection

4.12 External Accessory Detection

The CS47L35 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/or 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.

Headphone impedance can be detected via the HPDETL and HPDETR 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.

The MICVDD power domain must be enabled when using the microphone detect function. (Note that MICVDD is not required for the jack detect or headphone detect functions.) The MICVDD power domain is provided using an internal charge pump (CP2) and LDO regulator (LDO2). See Section 4.19 for details of these circuits.

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 4.16 for details of the internal 32-kHz clock and associated control fields.

4.12.1 Jack Detect

The CS47L35 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 4.13). This enables a jack insertion event to be used to trigger a wake-up of the CS47L35.

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*_STS*2* 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 4-74. 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 4.15). 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 4-74.

Table 4-74. 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.)
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.

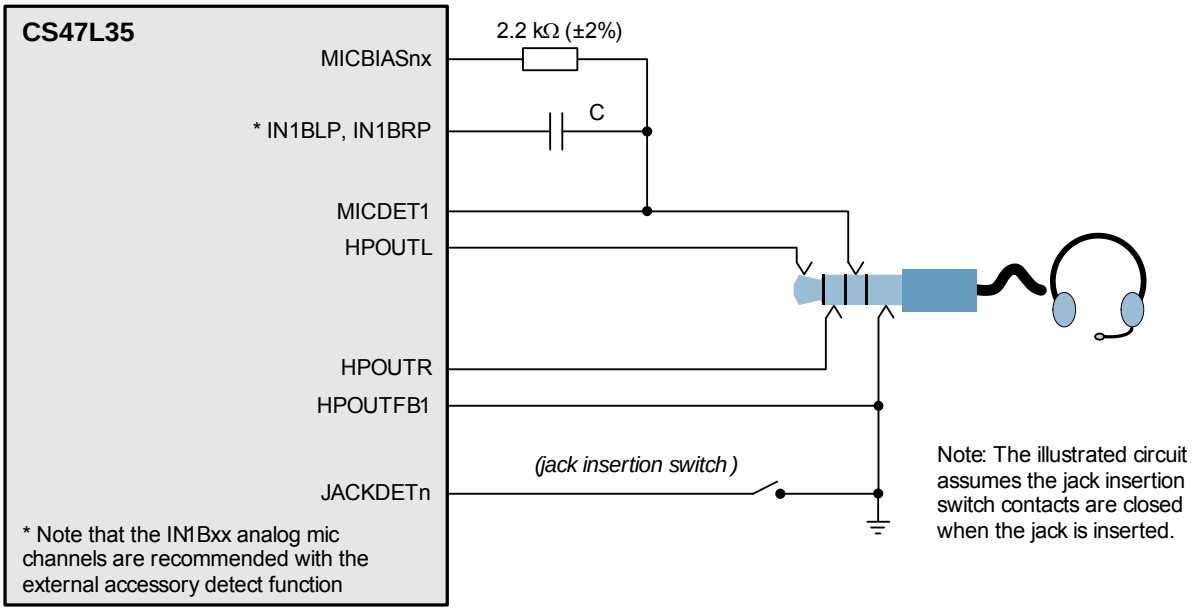


Figure 4-55. Jack Detect and External Accessory Connections

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.

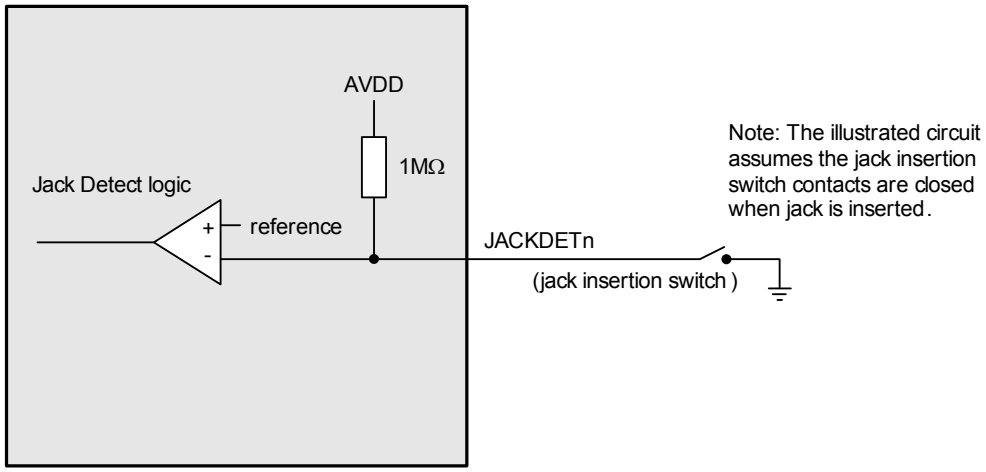


Figure 4-56. Jack Detect Comparator

4.12.2 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 when the headphone load makes momentary contact with the MICBIAS output when the jack is not fully inserted.

The CS47L35 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 external logic 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.

4.12.2.1 MICDET Clamp Control

The MICDET clamp function can be configured using the MICD_CLAMP_MODE field; the selectable logic conditions (derived from the JD1 and/or JD2 signals; see Table 4-74) provide support for different jack detect circuit configurations. The MICD_CLAMP_OVD bit, when set, activates the MICDET clamp, regardless of other conditions.

Note: The MICD_CLAMP_OVD bit is enabled by default; the MICDET clamp is always active following power-on reset, hardware reset, or software reset.

The MICDET clamp functionality (including the external IRQ) is maintained in Sleep Mode (see Section 4.13). This enables a jack insertion event to be used to trigger a wake-up of the CS47L35. A summary of the jack detect and MICDET clamp functionality, and their recommended usage in typical applications, is described in Section 4.12.2.5.

When the MICDET clamp is active, the MICDET1/HPOUTFB2 and HPOUTFB1/MICDET2 pins are short-circuited together. The grounding of the MICDET pin is achieved via the applicable HPOUTFB pin; note that it is assumed that the HPOUTFB connection is grounded externally, as shown in Fig. 4-57.

The configurable logic provides flexibility in selecting the appropriate conditions for activating the MICDET clamp. The clamp status 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 MICDET clamp debounce is selected by setting MICD_CLAMP_DB, as described in Table 4-75. 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.

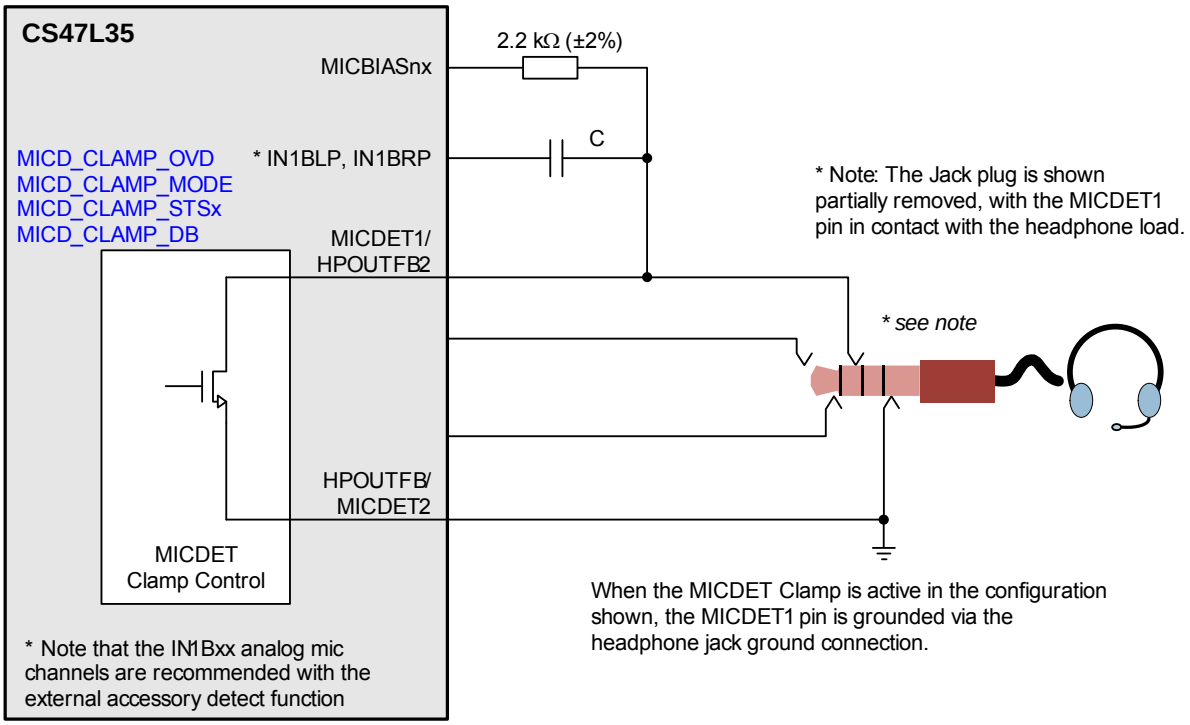


Figure 4-57. MICDET Clamp Circuit

4.12.2.2 Interrupts and Write-Sequencer Control

An interrupt request (IRQ) event is generated whenever the MICDET clamp is asserted or deasserted; see Section 4.15. Separate mask bits are provided to enable IRQ events on the rising and/or falling edge of the MICDET clamp status.

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; see Section 4.18 for further details.

4.12.2.3 Pop Suppression using General-Purpose Switch

In applications where a large decoupling capacitance is present on the MICBIAS output, the MICDET clamp function alone 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 within the CS47L35 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 4-75.

For jack pop suppression, it is recommended to set SW1_MODE = 11. In this case, the switch contacts are open whenever the MICDET clamp is active, and the switch contacts are closed whenever the MICDET clamp is inactive.

Normal accessory functions are supported when the switch contacts (GPSWP and GPSWN) are closed, and the MICDET clamp is inactive. Ground clamping of MICDET, and isolation of MICBIAS are achieved when the switch contacts are open, and the MICDET clamp is active.

Note that the MICDET clamp function must also be configured appropriately when using this method of pop suppression control.

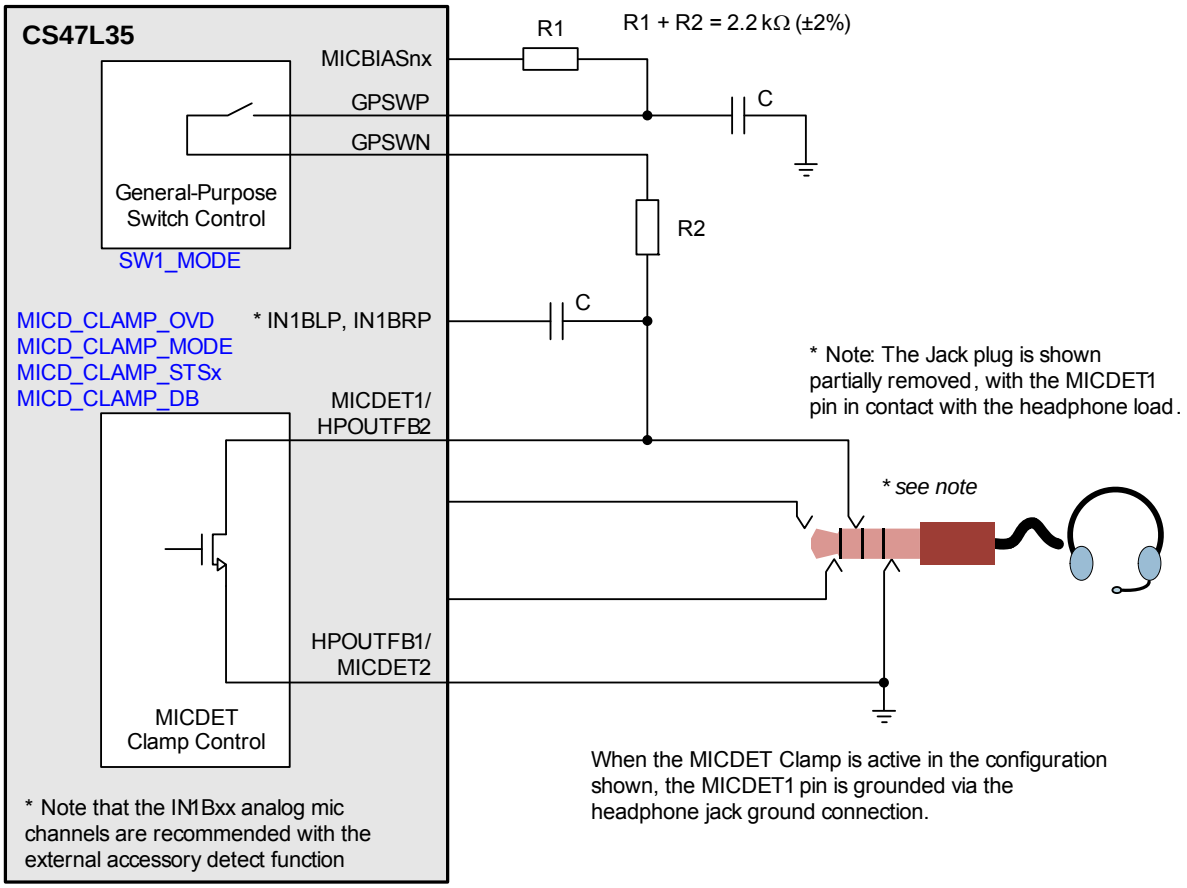


Figure 4-58. General-Purpose Switch Circuit

4.12.2.4 MICDET Clamp Control Registers

The control registers associated with the MICDET clamp and general-purpose switch functions are described in Table 4-75.

Table 4-75. 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
R710 (0x02C6) Micd_Clamp_ control	4	MICD_ CLAMP_OVD	1	MICDET Clamp Override 0 = Disabled 1 = Enabled (clamp active)
	3:0	MICD_ CLAMP_ MODE[3:0]	0000	MICDET Clamp Mode 0x0 = Disabled 0x1 = Active (MICDET1 and MICDET2 are shorted together) 0x2–0x3 = Reserved 0x4 = Active when JD1=0 0x5 = Active when JD1=1 0x6 = Active when JD2=0 0x7 = Active when JD2=1 0x8 = Active when JD1=0 or JD2=0 0x9 = Active when JD1=0 or JD2=1 0xA = Active when JD1=1 or JD2=0 0xB = Active when JD1=1 or JD2=1 0xC = Active when JD1=0 and JD2=0 0xD = Active when JD1=0 and JD2=1 0xE = Active when JD1=1 and JD2=0 0xF = Active when JD1=1 and JD2=1
R712 (0x02C8) GP_Switch_1	1:0	SW1_ MODE[1:0]	00	General-purpose Switch control 00 = Disabled (open) 10 = Enabled when MICDET clamp is active 01 = Enabled (closed) 11 = Enabled when MICDET clamp is not active
R6278 (0x1886) IRQ1_Raw_ Status_7	4	MICD_ CLAMP_STS1	0	MICDET Clamp status 0 = Clamp not active 1 = Clamp active
R6534 (0x1986) IRQ2_Raw_ Status_7	4	MICD_ CLAMP_STS2	0	MICDET Clamp status 0 = Clamp not active 1 = Clamp active
R6662 (0x1A06) Interrupt_ Debounce_7	4	MICD_ CLAMP_DB	0	MICDET Clamp debounce 0 = Disabled 1 = Enabled

4.12.2.5 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 active, 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 active (MICD_CLAMP_OVD = 1) until after a jack insertion has been detected.

  The MICDET_CLAMP_MODE field should be set according to the applicable JD1/JD2 signal configuration (configured to assert the clamp when jack is removed).
- Jack insertion is indicated using the JD1/JD2 signals (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. The JD1/JD2 status bits can be read directly, or can be unmasked as inputs to the interrupt controller. In this event, 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 should be asserted (MICD_CLAMP_OVD = 1), to make the system ready for a jack insertion.

The recommended control sequence for jack detect and MICDET clamp is summarized in Table 4-76.

Table 4-76. Control Sequence for Jack Detect and MICDET Clamp

Event	Device Actions	Recommended User Actions
Initial condition	Clamp asserted 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 asserted automatically	Disable MICBIAS and MICDET Disable I/O signal paths Enable clamp MICD_CLAMP_OVD = 1

4.12.3 Microphone Detect

The CS47L35 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.

4.12.3.1 Microphone Detect Control

The microphone detection circuit measures the impedance connected to MICDET1 or MICDET2. In the discrete measurement mode (ACCDET_MODE = 000), the function reports whether the measured impedance lies within one of eight predefined levels. In the ADC measurement mode (ACCDET_MODE = 111), a more specific result is provided in the form of a 7-bit ADC output.

The microphone detection circuit typically uses one of the MICBIAS outputs as a reference. The CS47L35 automatically enables the appropriate MICBIAS generator when required by the detection function; this allows the detection function to be supported in low-power standby operating conditions.

Note that the MICVDD power domain must be enabled when using the microphone detection function. This power domain is provided using an internal charge pump (CP2) and LDO regulator (LDO2). See Section 4.19 for details of these circuits. The internal 32-kHz clock must be present and enabled when using the microphone detection function; see Section 4.16 for details.

To select microphone detection on one of the MICDET pins, ACCDET_MODE must be set to 000 or 111 (depending on the desired measurement mode). The ACCDET_MODE field is defined in Table 4-77.

The CS47L35 can only support one headphone or microphone detection function at any time. When the detection function is not in use, it is recommended to set ACCDET_MODE = 000.

The microphone detection circuit can be enabled on the MICDET1 pin or the MICDET2 pin, selected by using the ACCDET_SRC bit.

The microphone detection circuit uses MICVDD, MICBIAS1A, MICBIAS1B, or MICBIAS2A as a reference. The applicable source is configured using the MICD_BIAS_SRC field. Note that MICBIAS2B is not a valid reference source for the microphone detection function.

When ACCDET_MODE is set to 000 or 111, microphone detection is enabled by setting MICD_ENA.

When microphone detection is enabled, the CS47L35 performs a number of measurements in order to determine the MICDET impedance. The measurement process is repeated at a cyclic rate controlled by MICD_RATE. The MICD_RATE field selects the delay between completion of one measurement and the start of the next. When the microphone detection result has settled, the CS47L35 indicates valid data by setting MICD_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 (ACCDET_MODE = 000), the measured impedance is only deemed valid after more than one successive measurement has produced the same result. The MICD_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 CS47L35 indicates valid data by setting the MICD_VALID bit. The measured impedance is indicated using the MICD_LVL and MICD_STS bits, as described in Table 4-77.

  The MICD_VALID bit, when set, remains asserted for as long as the microphone detection function is enabled (i.e., while MICD_ENA = 1). If the detected impedance changes, the MICD_LVL and MICD_STS fields change, but the MICD_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 MICD_LVL_SEL field is described in Section 4.12.3.3. The default configuration supports a maximum of four push buttons, in accordance with the Android headset specification for accessory push-button operation.

  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 (ACCDET_MODE = 111), the detection function generates two output results, contained within the MICDET_ADCVAL and MICDET_ADCVAL_DIFF fields. These fields contain the most recent measurement value (MICDET_ADCVAL) and the measurement difference value (MICDET_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 CS47L35 indicates valid data (MICD_VALID = 1), the detection must be disabled by setting MICD_ENA = 0.

  Note that MICDET_ADCVAL and MICDET_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 function is an input 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 4.15.

The fields associated with microphone detection (or other accessories) are described in Table 4-77. The external circuit configuration is shown in Fig. 4-59.

Table 4-77. Microphone Detect Control

Register Address	Bit	Label	Default	Description
R659 (0x0293) Accessory_ Detect_Mode_ 1	13	ACCDET_SRC	0	Accessory Detect/Headphone Feedback pin select 0 = Accessory detect on MICDET1; Headphone ground feedback on HPOUTFB1 1 = Accessory detect on MICDET2; Headphone ground feedback on HPOUTFB2
	2:0	ACCDET_ MODE[2:0]	000	Accessory Detect Mode Select 000 = Mic detect (MICDET*n*, discrete mode) 100 = Headphone detect (MICDET*n*) 001 = Headphone detect (HPDETL) 101 = Reserved 010 = Headphone detect (HPDETR) 110 = Reserved 011 = Reserved 111 = Mic detect (MICDET*n*, ADC mode) Note that the MICDET*n* measurements are implemented on either the MICDET1 or MICDET2 pins, depending on the ACCDET_SRC bit.

Table 4-77. Microphone Detect Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R675 (0x02A3) Mic_Detect_1	15:12	MICD_BIAS_ STARTTIME[3:0]	0001	Mic Detect Bias Start-up Delay (If MICBIAS is not enabled, this field selects the delay time allowed for MICBIAS to start-up before 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 to 1111 = 512 ms 0011 = 1 ms 1000 = 32 ms 0100 = 2 ms 1001 = 64 ms
	11:8	MICD_RATE[3:0]	0001	Mic Detect 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 to 1111 = 512 ms 0011 = 1 ms 1000 = 32 ms 0100 = 2 ms 1001 = 64 ms
	6:4	MICD_BIAS_ SRC[2:0]	000	Accessory Detect (MICDET) reference select 000 = MICVDD 010 = MICBIAS1B All other codes are reserved 001 = MICBIAS1A 011 = MICBIAS2A
	1	MICD_DBTIME	1	Mic Detect Debounce 0 = 2 measurements 1 = 4 measurements Only valid when ACCDET_MODE = 000.
	0	MICD_ENA	0	Mic Detect Enable 0 = Disabled 1 = Enabled
R676 (0x02A4) Mic_Detect_2	7:0	MICD_LVL_ SEL[7:0]	1001_ 1111	Mic Detect Level Select (enables mic/accessory detection in specific impedance ranges) [7] = Enable >1 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 when ACCDET_MODE = 000.
R677 (0x02A5) Mic_Detect_3	10:2	MICD_LVL[8:0]	0_ 0000_ 0000	Mic Detect Level (indicates the measured impedance) [8] = >475 Ω, <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 Only valid when ACCDET_MODE = 000.
	1	MICD_VALID	0	Mic Detect Data Valid 0 = Not Valid 1 = Valid
	0	MICD_STS	0	Mic Detect Status 0 = No mic/accessory present (impedance is >30 kΩ) 1 = Mic/accessory is present (impedance is <30 kΩ) Only valid when ACCDET_MODE = 000.
R683 (0x02AB) Mic_Detect_4	15:8	MICDET_ ADCVAL_ DIFF[7:0]	0x00	Mic Detect ADC Level (Difference) Only valid when ACCDET_MODE = 111.
	6:0	MICDET_ ADCVAL[6:0]	0x00	Mic Detect ADC Level Only valid when ACCDET_MODE = 111.

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.

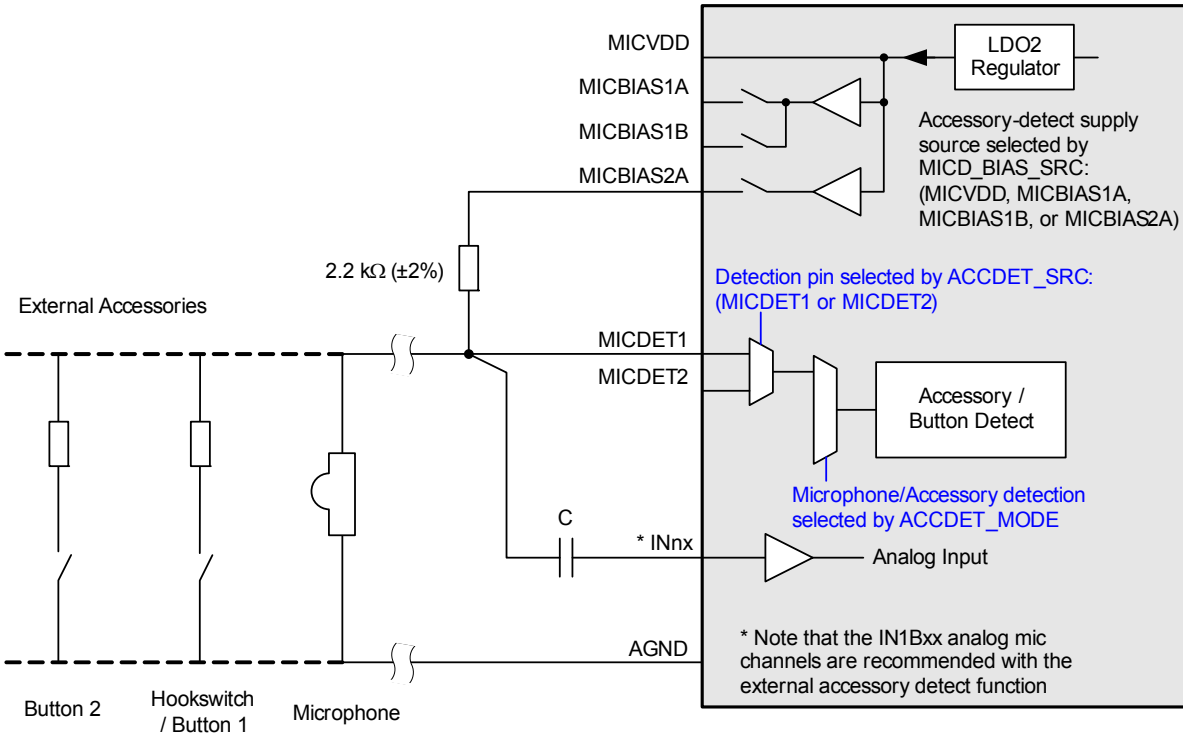


Figure 4-59. Microphone- and Accessory-Detect Interface

4.12.3.2 MICBIAS Reference Control

The voltage reference for the microphone detection is configured using the MICD_BIAS_SRC field, as described in Table 4-77. The microphone detection function automatically enables the applicable reference when required for MICDET impedance measurement.

If the selected reference (MICBIAS*nx*) 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 MICBIAS source to start-up, a time delay is applied before the measurement is performed; this is configured using MICD_BIAS_STARTTIME, as described in Table 4-77.

Note: The microphone detection automatically enables the applicable MICBIAS*x* generator (MICBIAS1 or MICBIAS2), every time the impedance measurement is scheduled. The respective MICBIAS output switches are not controlled automatically—the applicable switches must be enabled using the MICB1A_ENA, MICB1B_ENA or and MICB2A_ENA bits, as described in Table 4-112.

The MICD_BIAS_STARTTIME field should be set to 16 ms or more if MICB*n*_RATE = 1 (pop-free start-up/shutdown). MICD_BIAS_STARTTIME should be set to 0.25 ms or more if MICB*n*_RATE = 0 (fast start-up/shutdown).

The timing of the microphone detect function is shown in Fig. 4-60. Two different cases are shown, according to whether MICBIAS*nx* is enabled periodically by the impedance measurement function, or is enabled at all times.

If the selected reference (MICBIAS*nx*) is not enabled continuously, the respective MICBIAS*nx* discharge bits should be cleared. The MICBIAS control registers are described in Section 4.19.

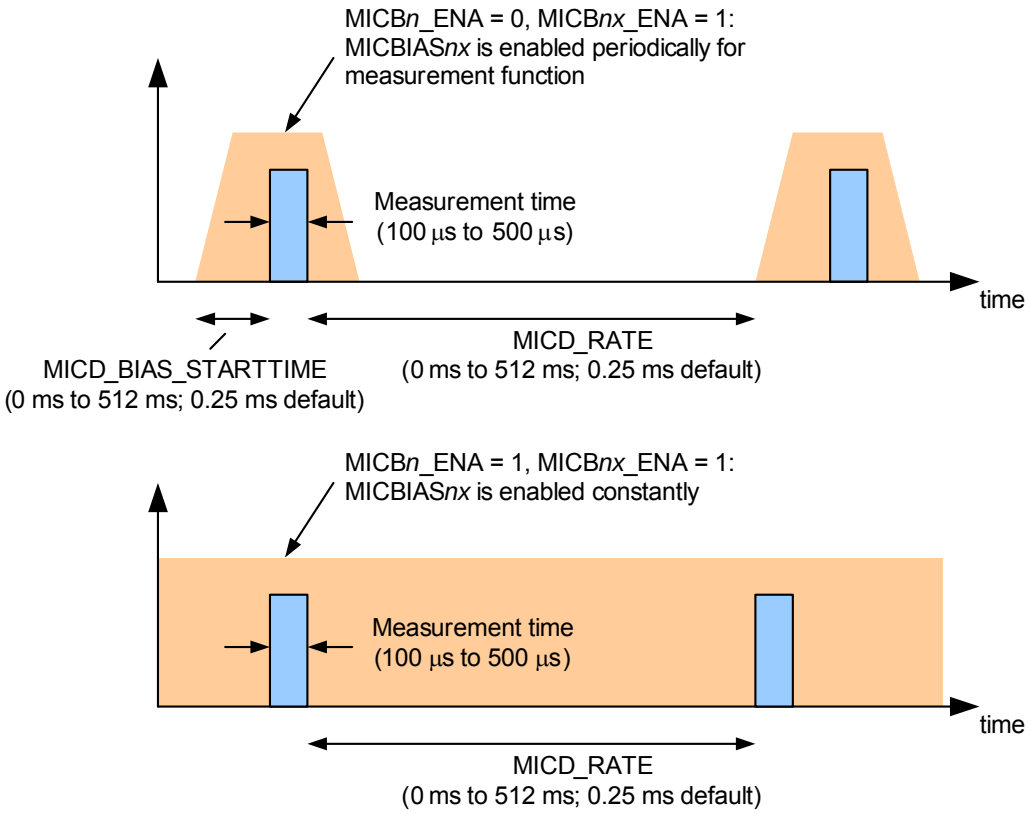


Figure 4-60. Microphone- and Accessory-Detect Timing

4.12.3.3 Measurement Range Control

When the discrete measurement mode is selected (ACCDET_MODE = 000), the MICD_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 MICD_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 MICD_LVL_SEL[2] bit enables the detection of a 360–680 Ω impedance. If MICD_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 MICD_LVL field as MICD_LVL[2] = 1.

With default register configuration, and all measurement levels enabled, the CS47L35 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.

4.12.3.4 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 MICDET 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.

4.12.4 Headphone Detect

The CS47L35 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.

4.12.4.1 Headphone Detection Control

Headphone detection can be enabled on the HPDETL pin or the HPDETR pin. Under recommended configuration, these pins provide measurement of the HPOUTL and HPOUTR loads respectively.

The headphone detect function can also be enabled on the MICDET1 pin or the MICDET2 pin. Note that, in this configuration, any MICBIAS output that is connected to the selected MICDET pin must be disabled and floating (MICB*nx*_ENA = 0, MICB*nx*_DISCH = 0).

The applicable headphone detection pin is selected using the ACCDET_MODE field. When MICDET*n* is selected (ACCDET_MODE = 100), the applicable MICDET*n* pin is determined by the ACCDET_SRC bit, as described in Table 4-80.

The CS47L35 can only support one headphone or microphone detection function at any time. When the detection function is not in use, it is recommended to set ACCDET_MODE = 000.

Headphone detection on the selected channel is commanded by writing 1 to HP_POLL.

The impedance measurement range is configured using HP_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 4.12.4.2.

Note: Setting HP_IMPEDANCE_RANGE is not required for detection on the MICDET*n* pins (ACCDET_MODE = 100). The impedance measurement range, and measurement accuracy, in this mode are different to the HPDETL and HPDETR measurement modes.

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 4-78.

Table 4-78. 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 4-64 for details of the HP1L_ENA and HP1R_ENA bits.

When headphone detection is commanded, the CS47L35 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 HP_CLK_DIV and HP_RATE.

4.12.4.2 Measurement Output

The headphone detection process typically comprises a number of separate measurements (for different impedance ranges). Completion of each measurement is indicated by HP_DONE. When this bit is set, the measurement result can be read from the HP_DACVAL and HP_DACVAL_DOWN fields, and decoded as described in Eq. 4-2.

$$\text{Impedance } (\Omega) = \frac{C_0 + (C_1 \times Offset)}{\left[\frac{(((\text{HP\_DACVAL} + \text{HP\_DACVAL\_DOWN})/2) + 0.5)}{C_2}\right] - \left[\frac{1}{C_3(1 + (C_4 \times Gradient))}\right]} - C_5$$

Equation 4-2. Headphone Impedance Calculation

The associated parameters for decoding the measurement result are defined Table 4-79. The applicable values are dependent on the HP_IMPEDANCE_RANGE setting in each case. The Offset and Gradient values are derived from register fields that are factory-calibrated for each device.

Table 4-79. Headphone Measurement Decode Parameters

Parameter	HP_IMPEDANCE_ RANGE = 00	HP_IMPEDANCE_ RANGE = 01	HP_IMPEDANCE_ RANGE = 10	HP_IMPEDANCE_ RANGE = 11
C_0	1.007	1.007	9.696	100.684
C_1	–0.0072	–0.0072	–0.0795	–0.9494
C_2	4003	7975	7300	7300
C_3	69.3	69.6	62.9	63.2
C_4	0.0055	0.0055	0.0055	0.0055
C_5	0.25	0.25	0.25	0.25
Offset	HP_OFFSET_00	HP_OFFSET_01	HP_OFFSET_10	HP_OFFSET_11
Gradient	HP_GRADIENT_0X	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 when 169 ≤ HP_DACVAL ≤ 1019. (In case of any contradiction with the HP_IMPEDANCE_RANGE description, the HP_DACVAL validity takes precedence.)

If the external impedance is entirely unknown (i.e., it could lie in any of the HP_IMPEDANCE_RANGE regions), it is recommended to test initially with HP_IMPEDANCE_RANGE = 00. If the resultant HP_DACVAL is < 169, the impedance is higher than the selected measurement range, so the test should be scheduled again, after incrementing HP_IMPEDANCE_RANGE.

Each measurement is triggered by writing 1 to HP_POLL. Completion of each measurement is indicated by HP_DONE. Note that, after HP_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 HP_LVL field. When the HP_DONE bit is set, indicating completion of a measurement, the impedance can be read directly from the HP_LVL field, provided that the value lies within the range of the applicable HP_IMPEDANCE_RANGE setting.

Note that, for detection using one of the MICDET*n* pins, the HP_LVL field is the only supported measurement output option. The HP_IMPEDANCE_RANGE field is not valid for detection on the MICDET*n* pins. See Table 4-80 for further description of the HP_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 4.15.

The fields associated with headphone detection are described in Table 4-80. 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 4-80 contain a mixture of 16- and 32-bit register addresses.

Table 4-80. Headphone Detect Control

Register Address	Bit	Label	Default	Description
R12792 (0x31F8) 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.
	7:0	HP_OFFSET_00[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.
R12794 (0x31FA) 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.
R659 (0x0293) Accessory_Detect_Mode_1	13	ACCDET_SRC	0	Accessory Detect/Headphone Feedback pin select 0 = Accessory detect on MICDET1, Headphone ground feedback on HPOUTFB1 1 = Accessory detect on MICDET2, Headphone ground feedback on HPOUTFB2
	2:0	ACCDET_MODE[2:0]	00	Accessory Detect Mode Select. 000 = Mic detect (MICDET*n*, discrete mode) 001 = Headphone detect (HPDETL) 010 = Headphone detect (HPDETR) 011 = Reserved 100 = Headphone detect (MICDET*n*) 101–110 = Reserved 111 = Mic detect (MICDET*n*, ADC mode) Note that the MICDET*n* measurements are implemented on either the MICDET1 or MICDET2 pins, depending on the ACCDET_SRC bit
R667 (0x029B) Headphone_Detect_1	10:9	HP_IMPEDANCE_RANGE[1:0]	00	Headphone Detect Range 00 = 4 Ω to 30 Ω 01 = 8 Ω to 100 Ω 10 = 100 Ω to 1 kΩ 11 = 1 kΩ to 10 kΩ Only valid when ACCDET_MODE = 001 or ACCDET_MODE = 010.
	4:3	HP_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	HP_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	HP_POLL	0	Headphone Detect Enable Write 1 to start HP Detect function

Table 4-80. Headphone Detect Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R668 (0x029C) Headphone_ Detect_2	15	HP_DONE	0	Headphone Detect Status 0 = HP Detect not complete 1 = HP Detect done
	14:0	HP_LVL[14:0]	0x0000	Headphone Detect Level LSB = 0.5 Ω 8 = 4 Ω or less 9 = 4.5 Ω 10 = 5 Ω 11 = 5.5 Ω … 20,000 = 10 kΩ or more For HPDETL or HPDETR measurement (ACCDET_MODE = 001 or 010), HPD_LVL is valid from 4 Ω to10 kΩ, within the range selected by HP_IMPEDANCE_RANGE. For MICDET*n* measurement (ACCDET_MODE = 100), HPD_LVL is valid from 400 Ω to 6 kΩ. only. If HP_LVL reports a value outside the selected 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.
R669 (0x029D) Headphone_ Detect_3	9:0	HP_ DACVAL[9:0]	0x000	Headphone Detect Level (Coded as integer, LSB = 1. See separate description for full decode information.)
R671 (0x029F) Headphone_ Detect_5	9:0	HP_DACVAL_ DOWN[9:0]	0x000	Headphone Detect Level (Coded as integer, LSB = 1. See separate description for full decode information.)

1. Default value is factory-set per device.

The external connections for the headphone detect circuit are shown in Fig. 4-61.

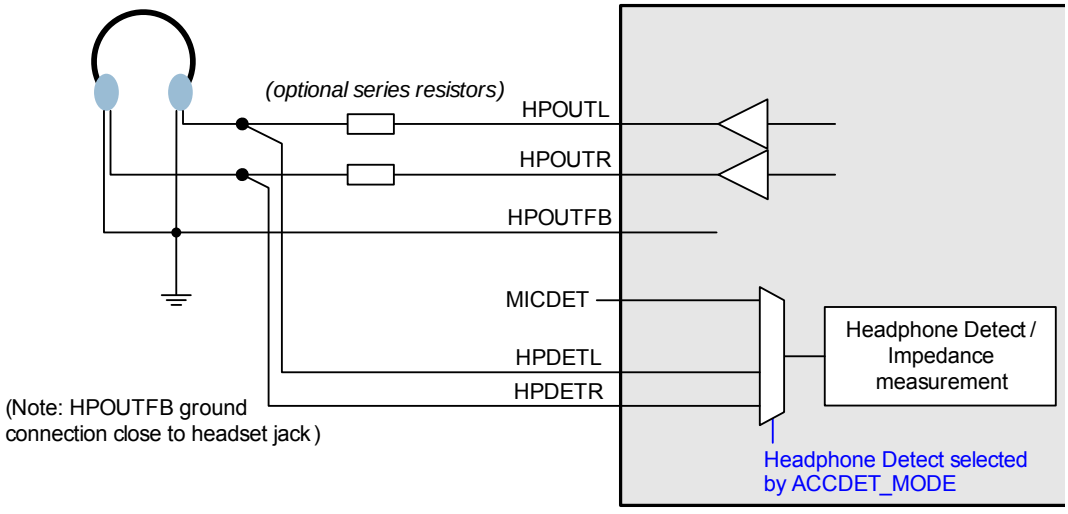


Figure 4-61. Headphone Detect Interface

Note that, where external resistors are connected in series with the headphone load, as shown, it is recommended that the HPDET*n* connection is to the headphone side of the resistors. If the HPDET*n* connection is made to the CS47L35 end of these resistors, this leads to a corresponding offset in the measured impedance.

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.

4.13 Low Power Sleep Configuration

The CS47L35 supports a low-power Sleep Mode, in which most functions are disabled and power consumption is minimized. The CS47L35 enters Sleep Mode when the DCVDD supply is removed. Note that the AVDD and DBVDD1 supplies must be present throughout the Sleep Mode duration.

In Sleep Mode, the CS47L35 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 CS47L35.

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 4-81. 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 4-81. Sleep Mode Always-On Control Registers

Register Address	Label	Reference
R710 (0x02C6)	MICD_CLAMP_OVD	See Section 4.12
	MICD_CLAMP_MODE[3:0]	
R723 (0x02D3)	JD2_ENA	
	JD1_ENA	
R6150 (0x1806)	MICD_CLAMP_FALL_EINT1	See Section 4.15
	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 4.23
	RESET_PD	

The always-on digital I/O pins are listed in Table 4-82. 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 4.14.1 for specific notes concerning the GPIO pins.

Table 4-82. Sleep Mode Always-On Digital Input/Output Pins

Pin Name	Description	Reference
$\overline{\text{IRQ}}$	Interrupt Request output	See Section 4.15
JACKDET1	Jack Detect input 1	See Section 4.12
JACKDET2	Jack Detect input 2	See Section 4.12
$\overline{\text{RESET}}$	Digital Reset input (active low)	See Section 4.23

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 4.12). 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 4-74 for details of the associated control fields.
- The MICDET clamp status is controlled by the JD1 and/or JD2 signals (see Section 4.12). 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 4-75.

The interrupt functionality associated with these signals is part of the always-on functionality, enabling the CS47L35 to provide indication of jack insertion or jack removal to the host processor in Sleep Mode; see Section 4.15.

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.

4.14 General-Purpose I/O

The CS47L35 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, DMIC, PDM, MIF)
- 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

Note that the GPIO pins are referenced to different power domains (DBVDD1 or DBVDD2), as indicated in Table 1-1.

Logic input and output (GPIO) can be supported in two different ways on the CS47L35. 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 4.5.4.

The CS47L35 also incorporates a general-purpose switch feature, which can be used as a controllable analog switch, as described in Section 4.14.16.

4.14.1 GPIO Control

For each GPIO, the selected function is determined by the GP*n*_FN field, where *n* identifies the GPIO pin (1–16). 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 4.16 for further details of the CS47L35 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 4.15 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 CS47L35 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 the GP*n*_OP_CFG bit has no effect on the respective GPIO pin—see Table 4-83 for further details. If GP*n*_FN = 0x002, the respective pin output is CMOS.

The register fields that control the GPIO pins are described in Table 4-83.

Table 4-83. GPIO Control

Register Address	Bit	Label	Default	Description
R5888 (0x1700) GPIO1_CTRL_1 to R5918 (0x171E) GPIO16_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 4-85). 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 4-84 for details)
R5889 (0x1701) GPIO1_CTRL_2 to R5919 (0x171F) GPIO16_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 4-85). 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 (1–16) 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.

4.14.2 GPIO Function Select

The available GPIO functions are described in Table 4-84. The function of each GPIO is set using GP*n*_FN, where *n* identifies the GPIO pin (1–16). Note that the respective GP*n*_DIR must also be set according to whether the function is an input or output.

Table 4-84. 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
0x018	FLL1 lock	Indicates FLL1 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 n status A pulse is output after the respective timer reaches its final count value.
0x141	Timer 2 status	
0x142	Timer 3 status	
0x143	Timer 4 status	
0x150	Event Log 1 FIFO not-empty status	Event Log *n* FIFO Not-Empty status 0 = FIFO Empty 1 = FIFO Not Empty
0x151	Event Log 2 FIFO not-empty status	
0x152	Event Log 3 FIFO not-empty status	
0x153	Event Log 4 FIFO not-empty status	

4.14.3 Pin-Specific Alternative Function—GP*n*_FN = 0x000

The CS47L35 GPIO capability is multiplexed with the pin-specific functions listed in Table 4-85. The alternate functions are selected by setting the respective GP*n*_FN fields to 0x000, as described in Section 4.14.1. 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 4-85. The respective GP*n*_DIR and GP*n*_OP_CFG bits have no effect in this case.

Table 4-85. GPIO Alternate Functions

Name	Condition	Description	Direction	Output Driver Configuration
AIF1BCLK/GPIO9	GP9_FN = 0x000	Audio Interface 1 bit clock	Digital I/O	CMOS
AIF1LRCLK/GPIO11	GP11_FN = 0x000	Audio Interface 1 left/right clock	Digital I/O	CMOS
AIF1RXDAT/GPIO8	GP8_FN = 0x000	Audio Interface 1 RX digital audio data	Digital input	CMOS
AIF1TXDAT/GPIO10	GP10_FN = 0x000	Audio Interface 1 TX digital audio data	Digital output	CMOS
AIF2BCLK/GPIO13	GP13_FN = 0x000	Audio Interface 2 bit clock	Digital I/O	CMOS
AIF2LRCLK/GPIO15	GP15_FN = 0x000	Audio Interface 2 left/right clock	Digital I/O	CMOS
AIF2RXDAT/GPIO14	GP14_FN = 0x000	Audio Interface 2 RX digital audio data	Digital input	CMOS
AIF2TXDAT/GPIO12	GP12_FN = 0x000	Audio Interface 2 TX digital audio data	Digital output	CMOS
AIF3BCLK/GPIO2	GP2_FN = 0x000	Audio Interface 3 bit clock	Digital I/O	CMOS
AIF3LRCLK/GPIO4	GP4_FN = 0x000	Audio Interface 3 left/right clock	Digital I/O	CMOS
AIF3RXDAT/GPIO3	GP3_FN = 0x000	Audio Interface 3 RX digital audio data	Digital input	CMOS
AIF3TXDAT/GPIO1	GP1_FN = 0x000	Audio Interface 3 TX digital audio data	Digital output	CMOS
MIF1SCLK/GPIO16	GP16_FN = 0x000	Master (I^2C) Interface 1 clock	Digital output	Open drain
MIF1SDA/GPIO7	GP7_FN = 0x000	Master (I^2C) Interface 1 data	Digital I/O	Open drain
SPKCLK/GPIO6	GP6_FN = 0x000	Digital speaker (PDM) clock	Digital output	CMOS
SPKDAT/GPIO5	GP5_FN = 0x000	Digital speaker (PDM) data	Digital output	CMOS

4.14.4 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 4.14.1. 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 4.15 for details of the interrupt event handling.

4.14.5 Logic 1 and Logic 0 Output (GPIO Output)—GP*n*_FN = 0x001

The CS47L35 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 4.14.1.

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.

4.14.6 DSP GPIO (Low-Latency DSP Input/Output)—GP*n*_FN = 0x002

The DSP GPIO function provides an advanced I/O capability, supporting the requirements of the CS47L35 as a multipurpose sensor hub. 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 4.14.1.

A full description of the DSP GPIO function is provided in Section 4.5.4.

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.

4.14.7 Interrupt (IRQ) Status Output—GP*n*_FN = 0x003, 0x004

The CS47L35 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 4.15 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 4.14.1.

Note that the IRQ1 status is output on the $\overline{\text{IRQ}}$ pin at all times.

4.14.8 Frequency-Locked Loop (FLL) Clock Output—GP*n*_FN = 0x010

A clock output derived from the FLL may be output on a GPIO pin. The GPIO output from the FLL is controlled by FLL1_GPCLK_DIV and FLL1_GPCLK_ENA, as described in Table 4-86.

It is recommended to disable the clock output (FLL1_GPCLK_ENA = 0) before making any change to the FLL1_GPCLK_DIV field.

Note that FLL1_GPCLK_DIV and FLL1_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 output may be output directly on a GPIO pin by setting the respective GPIO fields as described in Section 4.14.1.

See Section 4.16 for details of the CS47L35 system clocking and how to configure the FLL.

Table 4-86. FLL Clock Output Control

Register Address	Bit	Label	Default	Description
R392 (0x0188) FLL1_GPIO_Clock	7:1	FLL1_GPCLK_DIV[6:0]	0x06	FLL1 GPIO Clock Divider 0x00 to 0x05 = Reserved 0x06 = Divide by 6 0x07 = Divide by 7 0x08 = Divide by 8 0x09 = Divide by 9 … 0x7F = Divide by 127 (F_{GPIO} = F_{VCO}/FLL1_GPCLK_DIV)
	0	FLL1_GPCLK_ENA	0	FLL1 GPIO Clock Enable 0 = Disabled 1 = Enabled

4.14.9 Frequency-Locked Loop (FLL) Status Output—GP*n*_FN = 0x018

The CS47L35 provides an FLL status flag, which may be used to control other events. The FLL lock signal indicates whether FLL lock has been achieved. See Section 4.16.9 for details of the FLL.

The FLL lock signal may be output directly on a GPIO pin by setting the respective GPIO fields as described in Section 4.14.1.

The FLL lock signal is an input to the interrupt controller circuit. An interrupt event is triggered on the rising and falling edges of this signal. The associated interrupt bits are latched once set; they can be polled at any time or used to control the IRQ signal. See Section 4.15 for details of the interrupt event handling.

4.14.10 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 4-87.

It is recommended to disable the clock output (OPCLK_ENA = 0) before making any change to OPCLK_DIV or OPCLK_SEL.

The OPCLK output should be kept disabled (OPCLK_ENA = 0) if SYSCLK is not enabled. SYSCLK must be present and enabled before setting the OPCLK_ENA bit. See Section 4.16.4 for further details (including requirements for reconfiguring SYSCLK while digital core functions are enabled).

The OPCLK clock can be output directly on a GPIO pin by setting the respective GPIO fields as described in Section 4.14.1.

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 4.16 for details of the SYSCLK system clock.

Table 4-87. 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.

4.14.11 Pulse-Width Modulation (PWM) Signal Output—GP*n*_FN = 0x048, 0x049

The CS47L35 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 4.14.1.

See Section 4.3.12 for details of how to configure the PWM signal generators.

4.14.12 S/PDIF Audio Output—GP*n*_FN = 0x04C

The CS47L35 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 4.14.1.

See Section 4.3.8 for details of how to configure the S/PDIF output generator.

4.14.13 Overtemperature, Short-Circuit Protection, and Speaker Shutdown Status Output—GP*n*_FN = 0x0B6, 0x0E0, 0x0E1, 0x0E2.

The CS47L35 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 4.14.1. A GPIO pin can be used to indicate either an Overheat Warning Temperature event or an Overheat Shutdown Temperature event.

The CS47L35 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 4.14.1.

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 4.21. 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 4.15 for details of the interrupt event handling.

4.14.14 General-Purpose Timer Status Output—GP*n*_FN = 0x140–0x143

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 4.5.3 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 4.14.1. 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 4.15 for details of the interrupt event handling.

4.14.15 Event Logger FIFO Buffer Status Output—GP*n*_FN = 0x150–0x153

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 4.5.2 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 4.14.1. 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 IRQ signal. See Section 4.15 for details of the interrupt event handling.

4.14.16 General-Purpose Switch

The CS47L35 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 4-88.

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 4-88. 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

4.15 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 4-89 and Table 4-90 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 4-89. The interrupt register fields for IRQ2 are described in Table 4-90. 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 4-89 and Table 4-90 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 4-83 and Table 4-34.

Individual mask bits can enable or disable different functions from the interrupt controller. The mask bits are described in Table 4-89 (for IRQ1) and Table 4-90 (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 4-83. The GPIO debounce circuit uses the 32-kHz clock, which must be enabled whenever the GPIO debounce function is required.

A debounce circuit is always enabled on the FLL status inputs; to trigger an interrupt from the FLL status inputs, the 32-kHz clock or the SYSCLK signal must be enabled. Note that the raw status fields (described in Table 4-89 and Table 4-90) are valid without clocking, and can be used to provide FLL status indication when system clocks are not available.

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 IRQ pin. Under default conditions, this output is active low. The polarity can be inverted using IRQ_POL. The IRQ output can be either CMOS driven or open drain; this is selected using the IRQ_OP_CFG bit. The IRQ output is referenced to the DBVDD1 power domain.

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 4.14.

The CS47L35 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 4-89 and Table 4-90 respectively. The global interrupt mask bits, status bits, and output configuration fields are described Table 4-91.

Note that, under default register conditions, the boot done status is the only unmasked interrupt source; a falling edge on the IRQ pin indicates completion of the boot sequence.

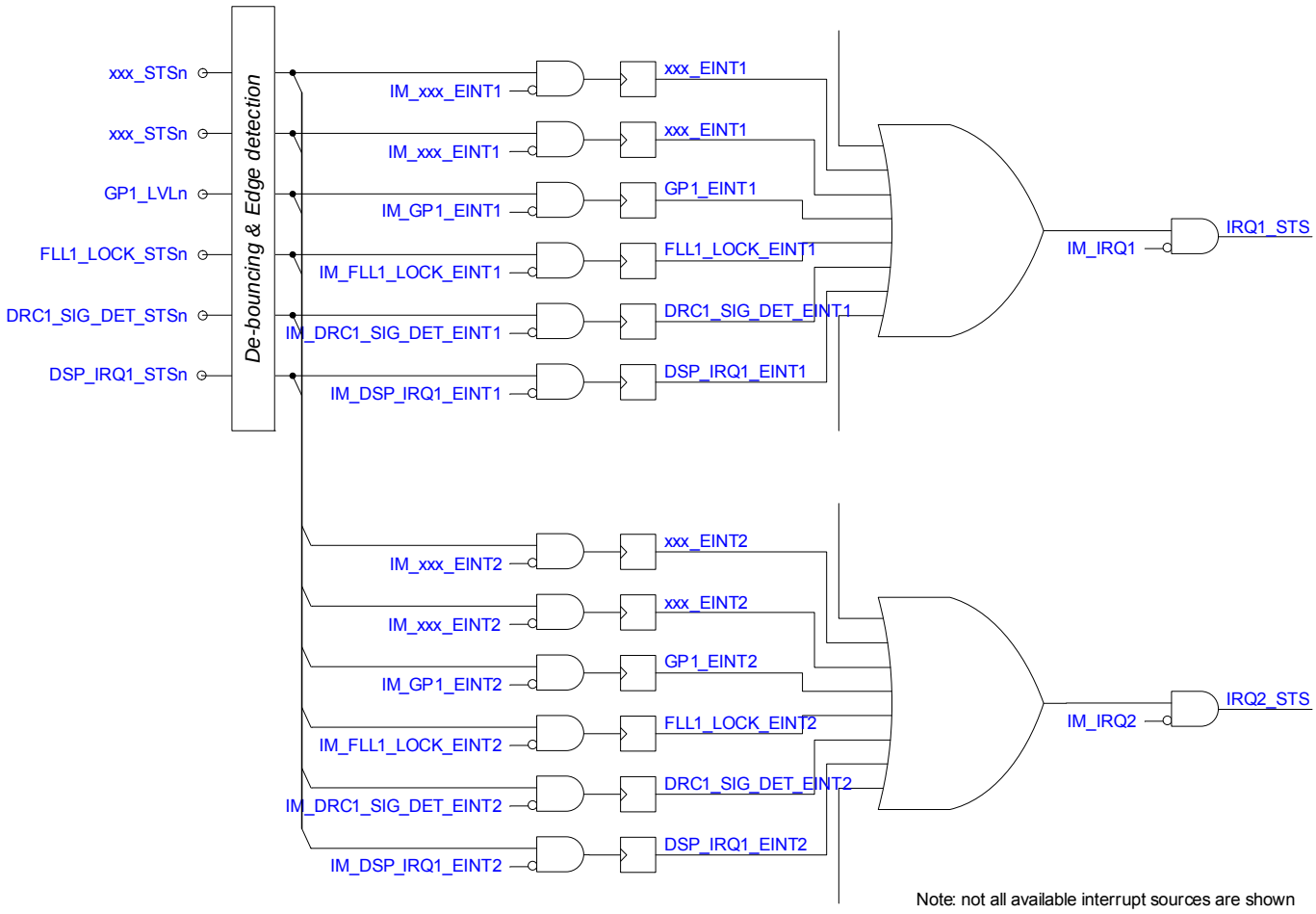


Figure 4-62. Interrupt Controller

The IRQ1 interrupt, mask, and status control registers are described in Table 4-89.

Table 4-89. Interrupt 1 Control Registers

Register Address	Bit	Label	Default	Description
R6144 (0x1800) IRQ1_Status_1	15	DSP_SHARED_WR_COLL_EINT1	0	DSP Shared Memory Collision Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	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.
R6145 (0x1801) IRQ1_Status_2	8	FLL1_LOCK_EINT1	0	FLL1 Lock Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
R6149 (0x1805) IRQ1_Status_6	8	MICDET_EINT1	0	Microphone/Accessory Detect 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	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 4-89. 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 4-89. 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	15	GP16_EINT1	0	GPIO16 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	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	3	TIMER4_EINT1	0	Timer 4 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	2	TIMER3_EINT1	0	Timer 3 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	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	3	EVENT4_NOT_EMPTY_EINT1	0	Event Log 4 FIFO Not Empty Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	2	EVENT3_NOT_EMPTY_EINT1	0	Event Log 3 FIFO Not Empty Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	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.

Table 4-89. Interrupt 1 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6166 (0x1816) IRQ1_Status_23	3	EVENT4_FULL_EINT1	0	Event Log 4 FIFO Full Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	2	EVENT3_FULL_EINT1	0	Event Log 3 FIFO Full Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	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	3	EVENT4_WMARK_EINT1	0	Event Log 4 FIFO Watermark Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	2	EVENT3_WMARK_EINT1	0	Event Log 3 FIFO Watermark Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	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	2	DSP3_DMA_EINT1	00	DSP3 DMA Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	DSP2_DMA_EINT1	00	DSP2 DMA Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	DSP1_DMA_EINT1	00	DSP1 DMA Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6170 (0x181A) IRQ1_Status_27	2	DSP3_START1_EINT1	0	DSP3 Start 1 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	DSP2_START1_EINT1	0	DSP2 Start 1 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	DSP1_START1_EINT1	0	DSP1 Start 1 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6171 (0x181B) IRQ1_Status_28	2	DSP3_START2_EINT1	0	DSP3 Start 2 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	DSP2_START2_EINT1	0	DSP2 Start 2 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	DSP1_START2_EINT1	0	DSP1 Start 2 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6173 (0x181D) IRQ1_Status_30	2	DSP3_BUSY_EINT1	0	DSP3 Busy Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	DSP2_BUSY_EINT1	0	DSP2 Busy Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	DSP1_BUSY_EINT1	0	DSP1 Busy Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6174 (0x181E) IRQ1_Status_31	0	MIF1_DONE_EINT1	0	MIF1 Done Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6175 (0x181F) IRQ1_Status_32	0	MIF1_BLOCK_EINT1	0	MIF1 Block Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6208 (0x1840) to R6239 (0x185F)		IM_*	See Footnote 1	For each x_EINT1 interrupt bit in R6144 to R6175, a corresponding mask bit (IM_*) is provided in R6208 to R6239. 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.

Table 4-89. Interrupt 1 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6273 (0x1881) IRQ1_Raw_ Status_2	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	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
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)

Table 4-89. Interrupt 1 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
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)
R6288 (0x1890) IRQ1_Raw_ Status_17	15	GP16_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).
	14	GP15_STS1	0	
	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	3	EVENT4_NOT_EMPTY_STS1	0	Event Log *n* FIFO Not Empty status 0 = FIFO Empty 1 = FIFO Not Empty
	2	EVENT3_NOT_EMPTY_STS1	0	
	1	EVENT2_NOT_EMPTY_STS1	0	
	0	EVENT1_NOT_EMPTY_STS1	0	
R6294 (0x1896) IRQ1_Raw_ Status_23	3	EVENT4_FULL_STS1	0	Event Log *n* FIFO Full status 0 = FIFO Not Full 1 = FIFO Full
	2	EVENT3_FULL_STS1	0	
	1	EVENT2_FULL_STS1	0	
	0	EVENT1_FULL_STS1	0	
R6295 (0x1897) IRQ1_Raw_ Status_24	3	EVENT4_WMARK_STS1	0	Event Log *n* FIFO Watermark status 0 = FIFO Watermark not reached 1 = FIFO Watermark reached
	2	EVENT3_WMARK_STS1	0	
	1	EVENT2_WMARK_STS1	0	
	0	EVENT1_WMARK_STS1	0	
R6296 (0x1898) IRQ1_Raw_ Status_25	2	DSP3_DMA_STS1	00	DSP*n* DMA status 0 = Normal 1 = All enabled WDMA buffers filled, and all enabled RDMA buffers emptied
	1	DSP2_DMA_STS1	00	
	0	DSP1_DMA_STS1	00	
R6301 (0x189D) IRQ1_Raw_ Status_30	2	DSP3_BUSY_STS1	0	DSP*n* Busy status 0 = DSP Idle 1 = DSP Busy
	1	DSP2_BUSY_STS1	0	
	0	DSP1_BUSY_STS1	0	

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 4-90.

Table 4-90. Interrupt 2 Control Registers

Register Address	Bit	Label	Default	Description
R6400 (0x1900) IRQ2_Status_1	15	DSP_SHARED_WR_COLL_EINT2	0	DSP Shared Memory Collision Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	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	8	FLL1_LOCK_EINT2	0	FLL1 Lock Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
R6405 (0x1905) IRQ2_Status_6	8	MICDET_EINT2	0	Microphone/Accessory Detect 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	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 4-90. 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 4-90. 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	15	GP16_EINT2	0	GPIO16 Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	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	3	TIMER4_EINT2	0	Timer 4 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	2	TIMER3_EINT2	0	Timer 3 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	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	3	EVENT4_NOT_EMPTY_EINT2	0	Event Log 4 FIFO Not Empty Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	2	EVENT3_NOT_EMPTY_EINT2	0	Event Log 3 FIFO Not Empty Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	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.

Table 4-90. Interrupt 2 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6422 (0x1916) IRQ2_Status_23	3	EVENT4_FULL_EINT2	0	Event Log 4 FIFO Full Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	2	EVENT3_FULL_EINT2	0	Event Log 3 FIFO Full Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	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	3	EVENT4_WMARK_EINT2	0	Event Log 4 FIFO Watermark Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	2	EVENT3_WMARK_EINT2	0	Event Log 3 FIFO Watermark Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	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	2	DSP3_DMA_EINT2	00	DSP3 DMA Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	DSP2_DMA_EINT2	00	DSP2 DMA Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	DSP1_DMA_EINT2	00	DSP1 DMA Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6426 (0x191A) IRQ2_Status_27	2	DSP3_START1_EINT2	0	DSP3 Start 1 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	DSP2_START1_EINT2	0	DSP2 Start 1 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	DSP1_START1_EINT2	0	DSP1 Start 1 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6427 (0x191B) IRQ2_Status_28	2	DSP3_START2_EINT2	0	DSP3 Start 2 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	DSP2_START2_EINT2	0	DSP2 Start 2 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	DSP1_START2_EINT2	0	DSP1 Start 2 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6429 (0x191D) IRQ2_Status_30	2	DSP3_BUSY_EINT2	0	DSP3 Busy Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	DSP2_BUSY_EINT2	0	DSP2 Busy Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	DSP1_BUSY_EINT2	0	DSP1 Busy Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6430 (0x191E) IRQ2_Status_31	0	MIF1_DONE_EINT2	0	MIF1 Done Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6431 (0x191F) IRQ2_Status_32	0	MIF1_BLOCK_EINT2	0	MIF1 Block Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6464 (0x1940) to R6495 (0x195F)		IM_*	1	For each x_EINT2 interrupt bit in R6400 to R6431, a corresponding mask bit (IM_*) is provided in R6464 to R6495. 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.

Table 4-90. Interrupt 2 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6529 (0x1981) IRQ2_Raw_ Status_2	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	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
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)

Table 4-90. Interrupt 2 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
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)
R6544 (0x1990) IRQ2_Raw_ Status_17	15	GP16_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).
	14	GP15_STS2	0	
	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	3	EVENT4_NOT_ EMPTY_STS2	0	Event Log *n* FIFO Not Empty status 0 = FIFO Empty 1 = FIFO Not Empty
	2	EVENT3_NOT_ EMPTY_STS2	0	
	1	EVENT2_NOT_ EMPTY_STS2	0	
	0	EVENT1_NOT_ EMPTY_STS2	0	
R6550 (0x1996) IRQ2_Raw_ Status_23	3	EVENT4_FULL_ STS2	0	Event Log *n* FIFO Full status 0 = FIFO Not Full 1 = FIFO Full
	2	EVENT3_FULL_ STS2	0	
	1	EVENT2_FULL_ STS2	0	
	0	EVENT1_FULL_ STS2	0	
R6551 (0x1997) IRQ2_Raw_ Status_24	3	EVENT4_ WMARK_STS2	0	Event Log *n* FIFO Watermark status 0 = FIFO Watermark not reached 1 = FIFO Watermark reached
	2	EVENT3_ WMARK_STS2	0	
	1	EVENT2_ WMARK_STS2	0	
	0	EVENT1_ WMARK_STS2	0	
R6552 (0x1998) IRQ2_Raw_ Status_25	2	DSP3_DMA_ STS2	00	DSP*n* DMA status 0 = Normal 1 = All enabled WDMA buffers filled, and all enabled RDMA buffers emptied
	1	DSP2_DMA_ STS2	00	
	0	DSP1_DMA_ STS2	00	

Table 4-90. Interrupt 2 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6557 (0x199D) IRQ2_Raw_ Status_30	2	DSP3_BUSY_ STS2	0	DSP*n* Busy status 0 = DSP Idle 1 = DSP Busy
	1	DSP2_BUSY_ STS2	0	
	0	DSP1_BUSY_ STS2	0	

The IRQ output and polarity control registers are described in Table 4-91.

Table 4-91. 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

4.16 Clocking and Sample Rates

The CS47L35 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 CS47L35 incorporates an FLL circuit to perform frequency conversion and filtering.

External clock signals may be connected via MCLK1 and MCLK2. These inputs are referenced to the DBVDD1 and DBVDD2 power domains respectively. In AIF Slave Modes, the BCLK signals may be used as a reference for the system clocks. The SLIMbus interface can provide the clock reference, when used as the input to the FLL. To avoid audible glitches, all clock configurations must be set up before enabling playback.

4.16.1 System Clocking Overview

The CS47L35 supports two primary clock domains—SYSCLK and DSPCLK.

The SYSCLK clock domain is the reference clock for all the audio signal paths on the CS47L35. 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 cores on the CS47L35. A wide range of DSPCLK frequencies can be supported, and a programmable clock divider is provided for each DSP core, allowing the DSP clocking (and power consumption) to be optimized according to the applicable processing requirements of each DSP core. See Section 4.4 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 cores; audio outputs from the DSP cores are synchronized to SYSCLK, regardless of the applicable DSPCLK rate.

Excluding the DSP cores, each subsystem within the CS47L35 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 cores are 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 processing requirements of each DSP core. The requirements vary, according to the particular software that is in use.

4.16.2 Sample-Rate Control

The CS47L35 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, SLIMbus), and for the input (ADC) and output (DAC) paths, but each enabled interface must still be synchronized to SYSCLK.

The CS47L35 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 4-92 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 CS47L35 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.

4.16.3 Automatic Sample-Rate Detection

The CS47L35 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 CS47L35).

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 4.18.

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 4-93.

4.16.4 SYSCLK Control

The SYSCLK clock may be provided directly from external inputs (MCLK, or Slave Mode BCLK inputs). Alternatively, SYSCLK can be derived using the integrated FLL, with MCLK, BCLK, LRCLK or SLIMCLK as a reference. The SYSCLK must be configured and enabled before any audio path is enabled.

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 4-92.

Table 4-92. SYSCLK Frequency Selection

SYSCLK Frequency (MHz)	SYSCLK_FREQ	SYSCLK_FRAC	Sample Rate (kHz)	SAMPLE_RATE_*n*
6.144 12.288 24.576 49.152 98.304	000 001 010 011 100	0	12	0x01
			24	0x02
			48	0x03
			96	0x04
			192	0x05
			8	0x11
			16	0x12
			32	0x13
5.6448 11.2896 22.5792 45.1584 90.3168	000 001 010 011 100	1	11.025	0x09
			22.05	0x0A
			44.1	0x0B
			88.2	0x0C
			176.4	0x0D

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 4-93. The source may be MCLK*n*, AIF*n*BCLK, or FLL. If the FLL circuit is selected as the source, the FLL must be enabled and configured, as described in Section 4.16.9.

Note: If FLL1 is selected as SYSCLK source, the SYSCLK frequency is F_{VCO} / 3. This enables SYSCLK frequencies of 98.304 MHz or 90.3168 MHz. See Section 4.16.9.

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 CS47L35 audio interfaces. The CS47L35 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) must be enabled before setting SYSCLK_ENA. This bit must be cleared when reconfiguring the SYSCLK source or frequency. The SYSCLK_ENA bit should also be cleared before stopping or removing the applicable clock source.

The SYSCLK signal is the reference clock for many different subsystems on the CS47L35. All of the SYSCLK-dependent subsystems should be disabled if SYSCLK is not enabled. The SYSCLK_ENA bit must be set before enabling any SYSCLK-dependent function, and all the dependent functions should be disabled before clearing the SYSCLK_ENA bit.

The SYSCLK-dependent subsystems are referenced below; if one or more of the following conditions is met, then the SYSCLK signal is required, and should not be interrupted or reconfigured.

- Input signal path enabled (IN*nx*_ENA = 1)
- OPCLK enabled for GPIO output (OPCLK_ENA = 1)
- Output signal path enabled (OUT5*x*_ENA = 1, SPKOUTL_ENA = 1, HP1*x*_ENA = 1)
- Digital audio interface path enabled (AIF*n*TX*m*_ENA = 1, AIF*n*RX*m*_ENA = 1)
- Digital audio interface clocks enabled (AIF*n*_BCLK_FRC = 1, AIF*n*_LRCLK_FRC = 1)
- SPDIF output enabled (SPD1_ENA = 1)
- Digital Core Mixer enabled (x_SRC*n* > 0x00)
- Haptic generator enabled (HAP_CTRL > 00)
- Tone generator enabled (TONE*n*_ENA = 1)
- PWM generator enabled (PWM*n*_ENA = 1)
- EQ, DRC, or LHPF processor enabled (EQ*n*_ENA = 1, DRC*nx*_ENA = 1, LHPF*n*_ENA = 1)
- ISRC channel enabled (ISRC*n*_INT*m*_ENA = 1, ISRC*n*_DEC*m*_ENA = 1)
- SLIMbus data channel enabled (SLIMTX*n*_ENA = 1, SLIMRX*n*_ENA = 1)
- Noise generator enabled (NOISE_GEN_ENA = 1)
- Timer enabled, with SYSCLK as clock source (TIMER*n*_RUNNING_STS = 1 and TIMER*n*_REFCLK_SRC = 0x8)
- DSP Core firmware requires access to registers below 0x40000

If reconfiguration of the SYSCLK source or frequency is required, and it is not possible to disable all of the SYSCLK-dependent subsystems, then the control-write sequencer must be used for the reconfiguration of SYSCLK. The control sequence should apply the following actions:

- Clear SYSCLK_ENA to 0
- Write updates to SYSCLK_SRC, SYSCLK_FREQ, and SYSCLK_FRAC
- Set SYSCLK_ENA to 1

The CS47L35 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.

4.16.5 DSPCLK Control

The DSPCLK clock may be provided directly from external inputs (MCLK, or Slave Mode BCLK inputs). Alternatively, DSPCLK can be derived using the integrated FLL, with MCLK, BCLK, LRCLK, or SLIMCLK as a reference. The DSPCLK must be configured and enabled, if running firmware applications on any of the DSP cores.

The required DSPCLK frequency depends on the requirements of firmware loaded on the DSP cores. The DSP cores are 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 processing requirements of each DSP core. The requirements vary, according to the particular software that is in use.

A configurable clock divider is provided for each DSP core, allowing the DSP clocking (and power consumption) to be optimized according to the applicable processing requirements of each DSP core; see Section 4.4 for details.

DSP_CLK_FREQ_RANGE must be configured for the applicable DSPCLK frequency. Note that, if the DSPCLK frequency is equal to one of the threshold frequencies quoted, the higher range setting should be selected. For example, if the DSPCLK frequency is 37.5 MHz, DSP_CLK_FREQ_RANGE should be set to 011.

DSP_CLK_SRC is used to select the DSPCLK source, as described in Table 4-93. The source may be MCLK*n*, AIF*n*BCLK, or FLL. If the FLL circuit is selected as the source, the FLL must be enabled and configured, as described in Section 4.16.9.

Note: If the FLL is selected as DSPCLK source, two different clock frequencies are available. For most use cases, the FLL output frequency is divided by two, when used as the DSPCLK source; this enables DSPCLK frequencies in the range 135–150 MHz. A divide-by-six option is also available, supporting low-power DSP operation with DSPCLK frequencies in the range 45–50 MHz.

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 must be cleared when reconfiguring the clock sources.

In a typical application, DSPCLK and SYSCLK are derived from a single FLL source. In this case, one of the nominal DSPCLK frequencies is likely to be applicable (see Table 4-93). Note that there is no requirement for DSPCLK to be synchronized to SYSCLK. The DSPCLK controls the software execution in the DSP cores; audio outputs from the DSP cores are synchronized to SYSCLK, regardless of the applicable DSPCLK rate.

The DSPCLK signal is the reference clock for the DSP cores and DSP peripherals on the CS47L35. All of the DSPCLK-dependent functions should be disabled if DSPCLK is not enabled. The DSPCLK_ENA bit must be set before enabling any DSPCLK-dependent function, and all the dependent functions should be disabled before clearing the DSPCLK_ENA bit.

The DSPCLK-dependent subsystems are referenced below; if one or more of the following conditions is met, then the DSPCLK signal is required, and should not be interrupted or reconfigured.

- DSP core enabled (DSPn_CORE_ENA = 1)
- DSP DMA function enabled (DSPn_[WDMA/RDMA]_CHANNEL_ENABLE > 0x00)
- DSP core in JTAG mode
- Master Interface active (MIF1_BUSY_STS = 1)
- Timer enabled (TIMERn_RUNNING_STS = 1)

If reconfiguration of the DSPCLK source or frequency is required, and it is not possible to disable all of the DSPCLK-dependent functions, then the following control requirements must be applied to reconfigure DSPCLK:

- Clear DSP_CLK_ENA to 0
- Wait 34 µs (only required if a Timer is enabled)
- Update DSP_CLK_SRC and DSP_CLK_FREQ_RANGE, and set DSP_CLK_ENA = 1. (These must be applied in a single register write operation)
- Note that, if a DSP core is enabled, DMA function is enabled, DSP core is in JTAG mode, or a Master Interface is active, then no other register read/write actions (either by Control Interface or by DSP firmware access) can be permitted during this sequence.
- If a Timer is enabled, but no DSP core, DMA, JTAG, or MIF is active, then DSPCLK can be stopped at any time. The minimum wait time of 34 µs is required before changing DSP_CLK_SRC or DSP_CLK_FREQ_RANGE, but there are no other constraints on configuring DSPCLK in these circumstances.

  If DSPCLK is the Timer clock source, the Timer pauses when DSPCLK stops, and resumes operation when DSPCLK restarts. If DSPCLK is not the clock source, the Timer operation continues when DSPCLK stops, but the Timer no longer synchronizes to DSPCLK.

4.16.6 Miscellaneous Clock Controls

The CS47L35 incorporates a 32-kHz clock circuit, which is required for input signal debounce, microphone/accessory detect, and for the Charge Pump 2 (CP2) 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 4.14 for details on configuring a GPIO pin for this function.

The CS47L35 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 CS47L35 is shown in Fig. 4-63.

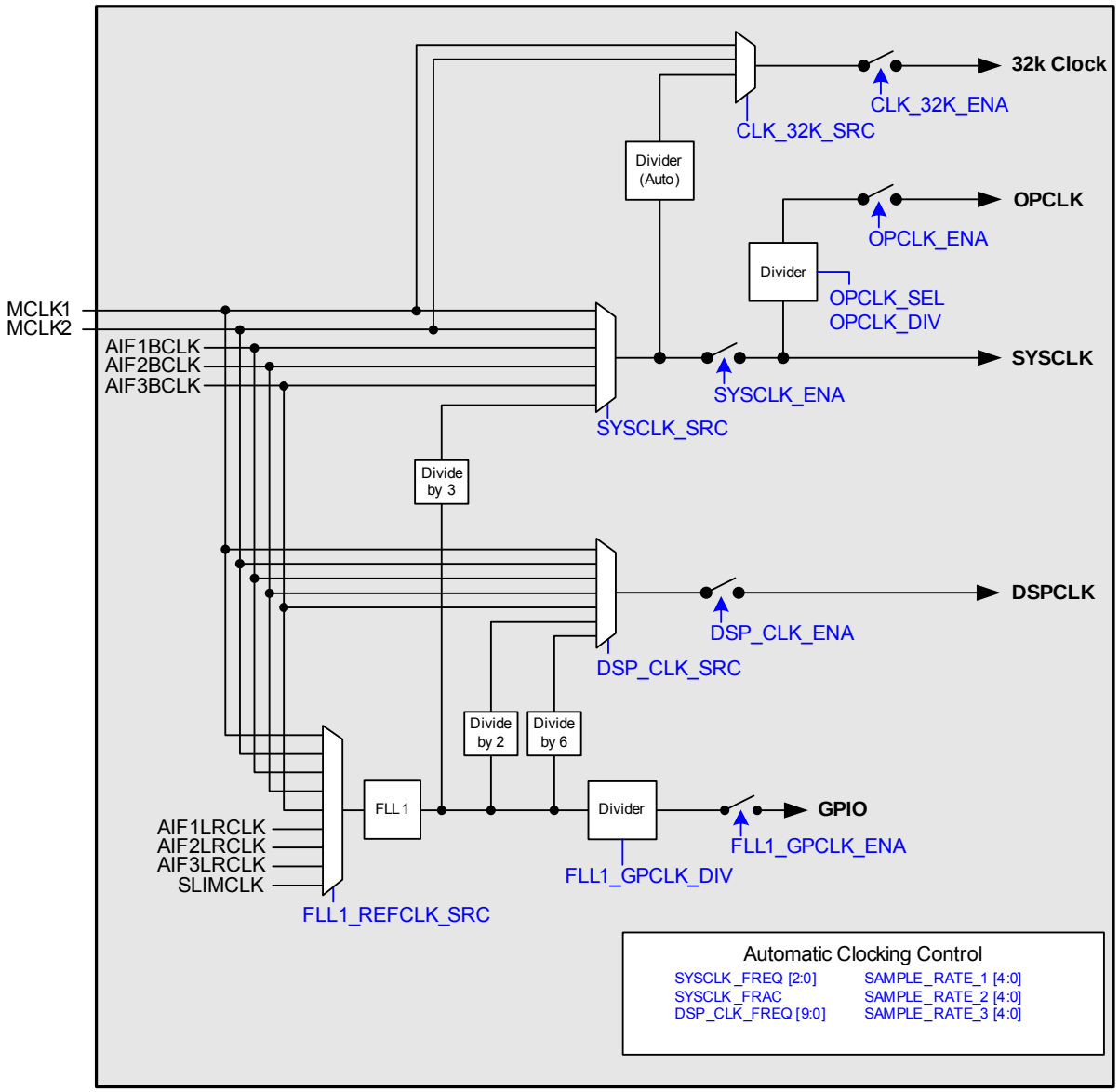


Figure 4-63. System Clocking

The CS47L35 clocking control registers are described in Table 4-93.

Table 4-93. 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 1000 = AIF1BCLK 1001 = AIF2BCLK 1010 = AIF3BCLK 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.
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.

Table 4-93. Clocking Control *(Cont.)*

Register Address	Bit	Label	Default	Description
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	10:8	DSP_CLK_ FREQ_ RANGE[2:0]	011	DSPCLK Frequency 000=5.5 MHz to 9.375 MHz (9.216 MHz) 001=9.375 MHz to 18.75 MHz (18.432 MHz) 010=18.75 MHz to 37.5 MHz (36.864 MHz) 011=37.5 MHz to 75 MHz (73.728 MHz) 100=75 MHz to 150 MHz (147.456 MHz) All other codes are reserved The frequencies in brackets are the nominal (or typical) frequencies for each setting. If the DSPCLK frequency is equal to one of the threshold frequencies quoted (e.g., 37.5 MHz), the higher range setting (e.g., 011) should be selected.
	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]	0101	DSPCLK Source 0000 = MCLK1 0001 = MCLK2 0100 = FLL1 0111 = FLL1 DIV6 1000 = AIF1BCLK 1001 = AIF2BCLK 1010 = AIF3BCLK All other codes are reserved
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.

Table 4-93. Clocking Control *(Cont.)*

Register Address	Bit	Label	Default	Description
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*.
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*.

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 the FLL, 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.

4.16.7 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 CS47L35. In Slave Mode, these are input signals to the CS47L35. 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.

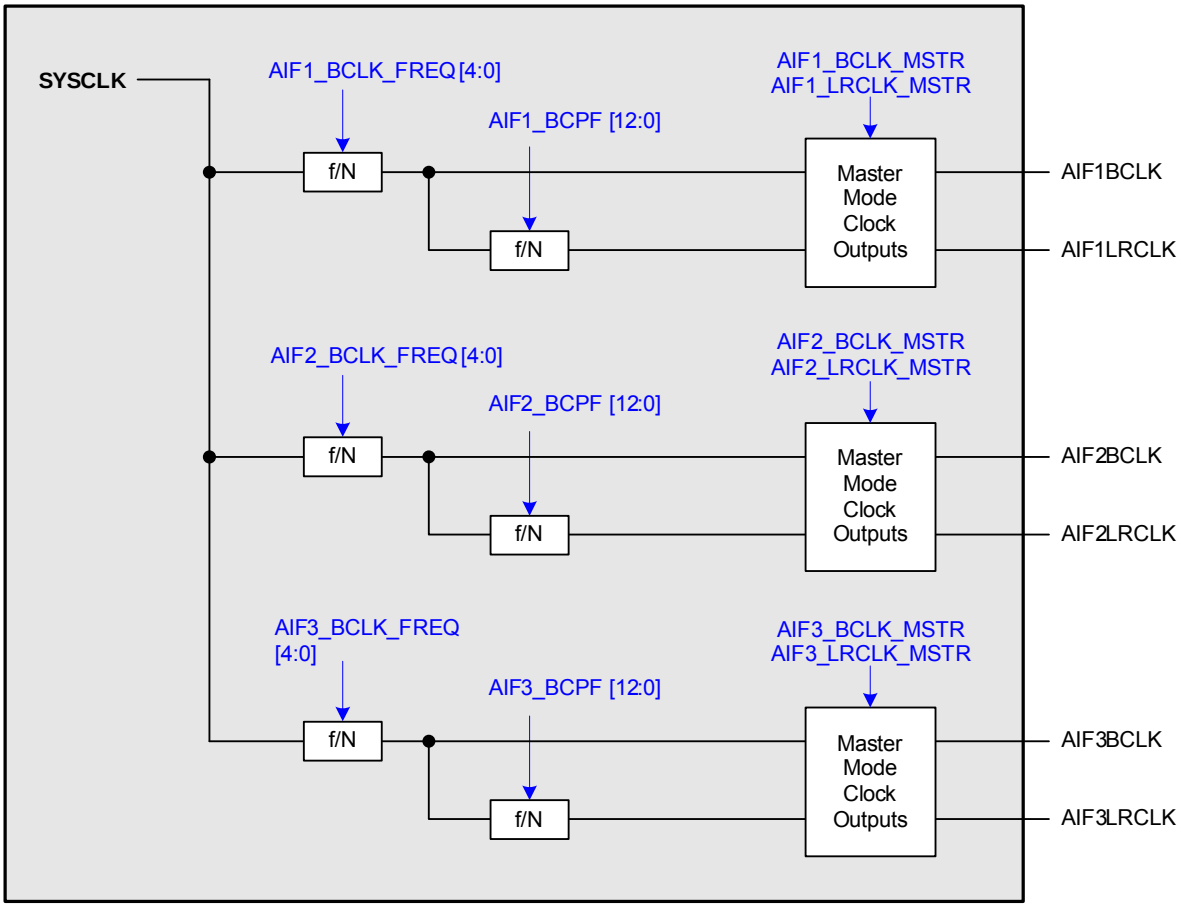


Figure 4-64. BCLK and LRCLK Control

4.16.8 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 4.17 for details of control register access.

4.16.9 Frequency-Locked Loop (FLL)

An integrated FLL is provided to support the clocking requirements of the CS47L35. This 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.

4.16.9.1 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 4.16.9.7. 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.

4.16.9.2 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 4.16.9.7. 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 any of the FLL-configuration fields, it is recommended to disable the FLL by clearing the FLL1_ENA bit before updating the other register fields; the FLL1_ENA bit should remain cleared until after the FLL has been reconfigured. If the FLL configuration is changed while the FLL is enabled, the FLL1_FREERUN bit should be set before updating any other FLL fields. A minimum delay of 32 µs should be allowed between setting FLL1_FREERUN and writing to the required FLL register fields. The FLL1_FREERUN bit should remain set until after the FLL has been reconfigured.

Note that, if the FLL1_N or FLL1_THETA fields are changed while the FLL is enabled, the FLL1_CTRL_UPD bit must also be written, as described in Section 4.16.9.4. As a general rule, however, it is recommended to configure the FLL (and FLL synchronizer, if applicable), before setting the FLL1_ENA bit.

The FLL configuration is shown in Fig. 4-65.

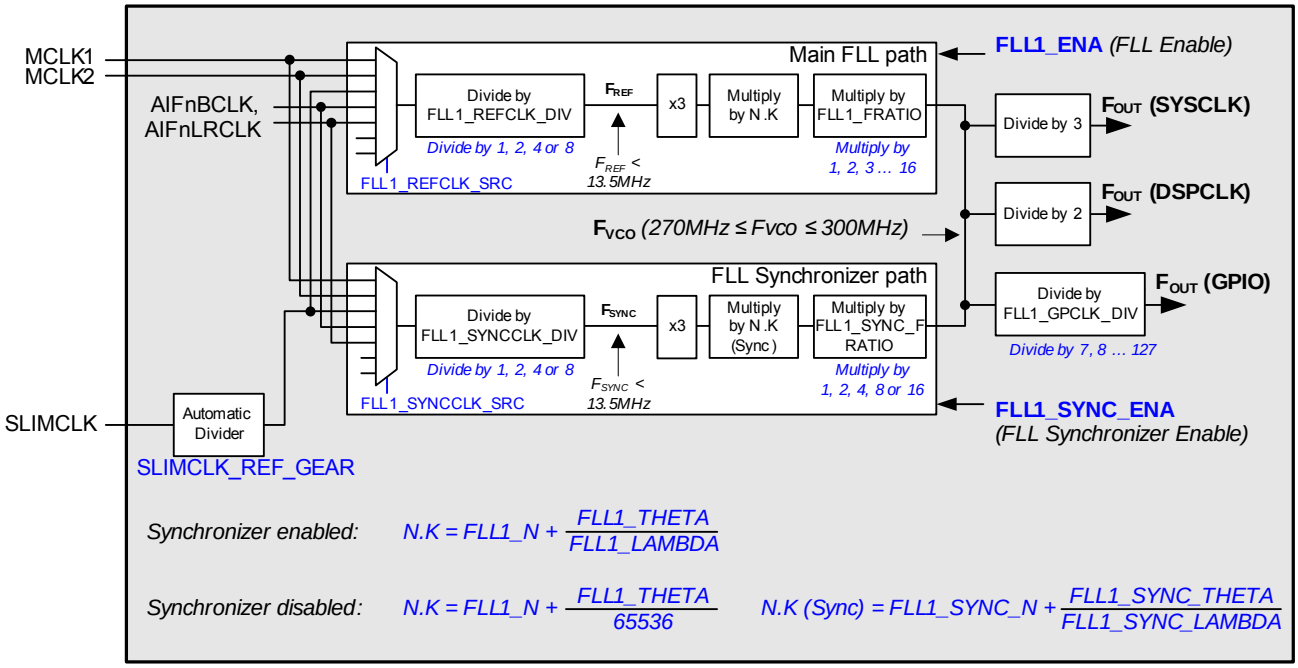


Figure 4-65. FLL Configuration

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.

4.16.9.3 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, SLIMCLK, AIF*n*BCLK, or AIF*n*LRCLK.

The SLIMCLK reference is controlled by an adaptive divider on the external SLIMCLK input. The divider automatically adapts to the SLIMbus clock gear, to provide a constant reference frequency for the FLL—see Section 4.10.7.

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.

4.16.9.4 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 270–300 MHz.

If the FLL is selected as SYSCLK source, a fixed divider set the output frequency equal to F_{VCO} / 3. Therefore, F_{VCO} must be exactly 294.912 MHz (for 48 kHz–related sample rates) or 270.9504 MHz (for 44.1 kHz–related sample rates).

If the FLL is selected as DSPCLK source, a fixed divider sets the output frequency equal to F_{VCO} / 2. This enables DSPCLK frequencies in the range 135–150 MHz. A divide-by-six option is also available, supporting low-power DSP operation with DSPCLK frequencies in the range 45–50 MHz. Note that the DSPCLK can be divided to lower clocking rates for each individual DSP.

The FLL clock can be configured as a GPIO output; a programmable divider supports division ratios in the range 7 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:

$$F_{VCO} = (F_{REF} \times 3 \times N.K \times FLL1\_FRATIO)$$

The value of N.K can thus be determined as follows:

$$N.K = F_{VCO} / (FLL1\_FRATIO \times 3 \times F_{REF})$$

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. The FLL1_LAMBDA field is not used in Integer Mode.
- In Fractional Mode (K > 0), the FLL1_THETA and FLL1_LAMBDA fields can be derived as described in Section 4.16.9.6.

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 or 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 4-94, depending on F_{REF}, FLL1_THETA, and whether the FLL synchronizer is enabled.

Table 4-94. 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} ≥ 768 kHz	0x3	
Synchronizer enabled (FLL1_SYNC_ENA = 1) or FLL Fractional Mode (FLL1_THETA > 0)	F_{REF} < 100 kHz	0x0	0
	100 kHz ≤ F_{REF} < 375 kHz	0x2	
	375 kHz ≤ F_{REF} < 1.5 MHz	0x3	
	1.5 MHz ≤ F_{REF} < 6.0 MHz	0x4	
	F_{REF} ≥ 6.0 MHz	0x5	

Note: F_{REF} is the input frequency, after division by FLL1_REFCLK_DIV, where applicable.

Other FLL fields are configured as follows:

- FLL1_PHASE_GAIN must be set to 0x2 in all cases.
- FLL1_CTRL_RATE must be set to 0x1 in all cases.

4.16.9.5 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 4-95.

Note: The FLL1_SYNC_FRATIO coding differs from that of FLL1_FRATIO.

Table 4-95. Selection of FLL1_SYNC_FRATIO, FLL1_SYNC_GAIN, FLL1_SYNC_DFSAT

Condition	FLL1_SYNC_FRATIO	FLL1_SYNC_GAIN	FLL1_SYNC_DFSAT
1 MHz ≤ F_{SYNC} < 13.5 MHz	0x0 (divide by 1)	0x4 (16x gain)	0 (wide bandwidth)
256 kHz ≤ F_{SYNC} < 1 MHz	0x1 (divide by 2)	0x2 (4x gain)	0 (wide bandwidth)
128 kHz ≤ F_{SYNC} < 256 kHz	0x2 (divide by 4)	0x0 (1x gain)	0 (wide bandwidth)
64 kHz ≤ 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 4.16.9.4.

The value of $N.K_{SYNC}$ can then be determined as follows:

$$N.K_{SYNC} = F_{VCO} / (FLL1\_SYNC\_FRATIO \times 3 \times F_{SYNC})$$

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:

- In Integer Mode (K_{SYNC} = 0), FLL1_SYNC_THETA must be set to 0. The FLL1_SYNC_THETA field is not used in Integer Mode.
- In Fractional Mode (K_{SYNC} > 0), the FLL1_SYNC_THETA and FLL1_SYNC_LAMBDA fields can be derived as described in Section 4.16.9.6.

The FLL1_SYNC_N, FLL1_SYNC_THETA, and FLL1_SYNC_LAMBDA fields are all coded as integers (LSB = 1).

4.16.9.6 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} / 3),
   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} / 3) – (FLL1_N x FLL1_FRATIO x F_{REF})) / GCD(FLL)
   FLL1_LAMBDA = (FLL1_FRATIO x F_{REF}) / GCD(FLL)

Note also that the values of FLL1_THETA and FLL1_LAMBDA must be coprime (i.e., not divisible by any common integer). 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.

4.16.9.7 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.

4.16.9.8 Spread-Spectrum FLL Control

The CS47L35 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 4.16.9.9.

4.16.9.9 FLL Control Registers

The FLL control registers are described in Table 4-96.

Example settings for a variety of reference frequencies and output frequencies are shown in Section 4.16.9.12.

Table 4-96. 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.
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 1110 = AIF3LRCLK 0001 = MCLK2 1010 = AIF3BCLK All other codes are reserved 0011 = SLIMCLK 1100 = AIF1LRCLK 1000 = AIF1BCLK 1101 = AIF2LRCLK
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

Table 4-96. FLL1 Register Map *(Cont.)*

Register Address	Bit	Label	Default	Description
R378 (0x017A) FLL1_Control_8	15:12	FLL1_PHASE_GAIN[3:0]	0010	FLL1 Phase Gain 0000 = Reserved, 0001 = Reserved, 0010 = 4, 0011 = 8, 0100 = 16, 0101 = 32, 0110 = 64, 0111 = 128, 1000–1111 = 256
	11	FLL1_PHASE_ENA	1	FLL1 Phase Integrator Control 0 = Disabled 1 = Enabled
	10:8	FLL1_CTRL_RATE[2:0]	001	FLL1 Control Ratio 000 = 1, 001 = 2, 010 = 4, 011 = 8, 100 = 16, 101 = 32, 110 = 64, 111 = 128
R383 (0x017F) 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.
R384 (0x0180) FLL1_Synchroniser_2	9:0	FLL1_SYNC_N[9:0]	0x000	FLL1 Integer multiply for F_{SYNC} (LSB = 1)
R385 (0x0181) 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.
R386 (0x0182) 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.
R387 (0x0183) FLL1_Synchroniser_5	10:8	FLL1_SYNC_FRATIO[2:0]	000	FLL1 Synchronizer F_{VCO} clock divider 000 = 1, 001 = 2, 010 = 4, 011 = 8, 1XX = 16
R388 (0x0184) FLL1_Synchroniser_6	7:6	FLL1_SYNCCLK_DIV[1:0]	00	FLL1 Synchronizer Clock Reference Divider 00 = 1, 01 = 2, 10 = 4, 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, 0001 = MCLK2, 0011 = SLIMCLK, 1000 = AIF1BCLK, 1001 = AIF2BCLK, 1010 = AIF3BCLK, 1100 = AIF1LRCLK, 1101 = AIF2LRCLK, 1110 = AIF3LRCLK, All other codes are reserved
R389 (0x0185) FLL1_Synchroniser_7	5:2	FLL1_SYNC_GAIN[3:0]	0000	FLL1 Synchronizer Gain 0000 = 1, 0001 = 2, 0010 = 4, 0011 = 8, 0100 = 16, 0101 = 32, 0110 = 64, 0111 = 128, 1000–1111 = 256
	0	FLL1_SYNC_DFSAT	1	FLL1 Synchronizer Bandwidth 0 = Wide bandwidth 1 = Narrow bandwidth
R391 (0x0187) 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) 01 = 1.1% (triangle), 1.3% (ZMFM, dither) 10 = 2.3% (triangle), 2.6% (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 01 = 878 kHz 10 = 1.17 MHz 11 = 1.76 MHz
	1:0	FLL1_SS_SEL[1:0]	00	FLL1 Spread Spectrum Select. 00 = Disabled 01 = Zero Mean Frequency (ZMFM) 10 = Triangle 11 = Dither

4.16.9.10FLL Interrupts and GPIO Output

The CS47L35 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 4.15. Note that the interrupt signal is debounced, and requires clocking to be present in order to assert the respective interrupt; either the 32-kHz clock, or the SYSCLK signal, must be enabled to trigger an interrupt from the FLL lock signal.

The FLL lock signal can be output directly on a GPIO pin as an external indication of the FLL status. See Section 4.14 to configure a GPIO pin for these functions. (This GPIO output is not debounced, and does not require clocking to be present.)

Clock output signals derived from the FLL can be output on a GPIO pin. See Section 4.14 to configure a GPIO pin for this function.

The FLL clocking configuration is shown in Fig. 4-65.

4.16.9.11Example FLL Calculation

The following example illustrates how to derive the FLL1 register fields to generate an oscillator frequency (F_{VCO}) of 294.912 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.

Note that, for the purposes of this calculation, it is assumed that the synchronizer is disabled.

1. Set FLL1_REFCLK_DIV to generate $F_{REF} \leq 13.5$ MHz:
   FLL1_REFCLK_DIV = 00 (divide by 1)
2. Calculate N.K as given by N.K = F_{VCO} / (FLL1_FRATIO × 3 × F_{REF}). Assume FLL1_FRATIO = 0x0 (divide by 1).
   N.K = 294912000 / (1 × 3× 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} / 3):
   GCD(FLL) = GCD(1 × 12000000, 294912000 / 3) = 96000
6. Determine FLL1_THETA, as given by FLL1_THETA = (F_{VCO} / 3 – (FLL1_N × FLL1_FRATIO × F_{REF})) / GCD(FLL):
   FLL1_THETA = ((294912000 / 3) – (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, FLL1_PHASE_GAIN, FLL1_PHASE_ENA, and FLL1_CTRL_RATE as specified in Section 4.16.9.4:
   FLL1_GAIN = 0x5
   FLL1_PHASE_GAIN = 0x2
   FLL1_PHASE_ENA = 1
   FLL1_CTRL_RATE = 0x1

4.16.9.12Example FLL Settings

Table 4-97 shows FLL settings for generating an oscillator frequency (F_{VCO}) of 294.912 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.

The recommended values of FLL1_PHASE_GAIN (0x2) and FLL1_CTRL_RATE (0x1) are also applicable in each case.

Note that the FLL settings in Table 4-97 assume that the synchronizer is disabled.

Table 4-97. Example FLL Settings—Synchronizer Disabled

FLL (Main Loop) Settings									
F_{SOURCE}	F_{VCO} (MHz) [1]	F_{REF} Divider [2]	FRATIO [2]	N.K [3]	FLL1_N	FLL1_THETA	FLL1_LAMBDA	FLL1_GAIN	FLL1_PHASE_ENA
32.000 kHz	294.912	1	4	768	0x300	0x0000	0x0001	0x2	1
32.768 kHz	294.912	1	3	1000	0x3E8	0x0000	0x0001	0x2	1
48 kHz	294.912	1	3	682.6667	0x2AA	0x0002	0x0003	0x0	0
128 kHz	294.912	1	1	768	0x300	0x0000	0x0001	0x2	1
512 kHz	294.912	1	1	192	0x0C0	0x0000	0x0001	0x2	1
1.536 MHz	294.912	1	1	64	0x040	0x0000	0x0001	0x3	1
3.072 MHz	294.912	1	1	32	0x020	0x0000	0x0001	0x3	1
11.2896 MHz	294.912	1	1	8.7075	0x008	0x0068	0x0093	0x5	0
12.000 MHz	294.912	1	1	8.192	0x008	0x0018	0x007D	0x5	0
12.288 MHz	294.912	1	1	8	0x008	0x0000	0x0001	0x3	1
13.000 MHz	294.912	1	1	7.5618	0x007	0x0391	0x0659	0x5	0
19.200 MHz	294.912	2	1	10.24	0x00A	0x0006	0x0019	0x5	0
24 MHz	294.912	2	1	8.192	0x008	0x0018	0x007D	0x5	0
26 MHz	294.912	2	1	7.5618	0x007	0x0391	0x0659	0x5	0
27 MHz	294.912	2	1	7.2818	0x007	0x013D	0x0465	0x5	0

1. F_{VCO} = (F_{SOURCE}/F_{REF} Divider) × 3 × N.K × FRATIO
2. See Table 4-96 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.

Table 4-98 shows example FLL settings for generating an oscillator frequency (F_{VCO}) of 294.912 MHz from a variety of low- and high-frequency reference inputs, with the synchronizer enabled. The main loop and the synchronizer loop must each be configured according to the respective input source.

Note that, if the FLL synchronizer is enabled, the recommended settings for the main loop are not the same as those described in Table 4-97.

Table 4-98. Example FLL Settings—Synchronizer Enabled

FLL (Main Loop) Settings									
F_{SOURCE}	F_{VCO} (MHz) [1]	F_{REF} Divider [2]	FRATIO [2]	N.K [3]	FLL1_N	FLL1_THETA	FLL1_LAMBDA	FLL1_GAIN	FLL1_PHASE_ENA
32.000 kHz	294.912	1	4	768	0x300	0x0000	0x0000	0x0	0
32.768 kHz	294.912	1	3	1000	0x3E8	0x0000	0x0000	0x0	0
48 kHz	294.912	1	3	682.6667	0x2AA	0xAAAA	0x0000	0x0	0
128 kHz	294.912	1	1	768	0x300	0x0000	0x0000	0x2	0
512 kHz	294.912	1	1	192	0x0C0	0x0000	0x0000	0x3	0
1.536 MHz	294.912	1	1	64	0x040	0x0000	0x0000	0x4	0
3.072 MHz	294.912	1	1	32	0x020	0x0000	0x0000	0x4	0
11.2896 MHz	294.912	1	1	8.7075	0x008	0xB51D	0x0000	0x5	0
12.000 MHz	294.912	1	1	8.192	0x008	0x3126	0x0000	0x5	0
12.288 MHz	294.912	1	1	8	0x008	0x0000	0x0000	0x5	0
13.000 MHz	294.912	1	1	7.5618	0x007	0x8FD5	0x0000	0x5	0
19.200 MHz	294.912	2	1	10.24	0x00A	0x3D70	0x0000	0x5	0
24 MHz	294.912	2	1	8.192	0x008	0x3126	0x0000	0x5	0
26 MHz	294.912	2	1	7.5618	0x007	0x8FD5	0x0000	0x5	0
27 MHz	294.912	2	1	7.2818	0x007	0x4822	0x0000	0x5	0
FLL (Synchronizer Loop) Settings									
F_{SOURCE}	F_{VCO} (MHz) [4]	F_{SYNC} Divider [5]	FRATIO [5]	N.K [6]	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) × 3 × N.K × FRATIO
2. See Table 4-96 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. F_{VCO} = (F_{SOURCE}/F_{SYNC} Divider) × 3 × N.K × FRATIO
5. See Table 4-96 for the coding of the FLL1_SYNCCLK_DIV and FLL1_SYNC_FRATIO fields.
6. N.K values are represented in the FLL1_SYNC_N, FLL1_SYNC_THETA, and FLL1_SYNC_LAMBDA fields.

4.17 Control Interface

The CS47L35 is controlled by read/write access to its control registers. Two independent control interfaces are provided, giving flexible capability as described below. Note that the SLIMbus interface also supports read/write access to the CS47L35 control registers; see Section 4.10.

Register access is possible on all of the control interfaces (including SLIMbus) simultaneously. Note that 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 CS47L35 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 4.22 for further details.

A summary of the CS47L35 control interfaces is described in Table 4-99.

Table 4-99. CS47L35 Control Interface Summary

Control Interface	Description	Pin Functions	Power Domain
CIF1	Four-wire (SPI) interface	CIF1MISO—Data output CIF1MOSI—Data input CIF1SCLK—Interface clock input CIF1$\overline{SS}$—Slave select input	DBVDD1
CIF2	Two-wire (I^2C) interface	CIF2SCLK—Interface clock input CIF2SDA—Data input/output	DBVDD1

The CS47L35 provides an integrated pull-down resistor on the CIF1MISO pin. This provides a flexible capability for interfacing with other devices. The pull-down is enabled by setting CIF1MISO_PD, as described in Table 4-100.

Table 4-100. Control Interface Pull-Down

Register Address	Bit	Label	Default	Description
R8 (0x0008) Ctrl_IF_CFG_1	7	CIF1MISO_PD	0	CIF1MISO Pull-Down Control 0 = Disabled 1 = Enabled

A detailed description of the I^2C and SPI interface modes is provided in the following sections.

4.17.1 Four-Wire (SPI) Control Mode

The SPI control interface mode is supported on CIF1 only and uses the respective $\overline{SS}$, SCLK, MOSI, and MISO pins.

In write operations (R/$\overline{W}$ = 0), the MOSI pin input is driven by the controlling device.

In read operations (R/$\overline{W}$ = 1), the MOSI pin is ignored following receipt of the valid register address.

If $\overline{SS}$ is asserted (Logic 0), the MISO output is actively driven when outputting data and is high impedance at other times. If $\overline{SS}$ 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 CIF1MISO pin, as described in Table 4-100.

Data transfers on CIF1 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. 4-66 and Fig. 4-67 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 CS47L35 automatically increments the register address at the end of each data word, for as long as $\overline{SS}$ 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. 4-66 and Fig. 4-67. Note that 16-bit data words are shown, but the equivalent protocol also applies to 32-bit data words.

Fig. 4-66 shows a single register write to a specified address.

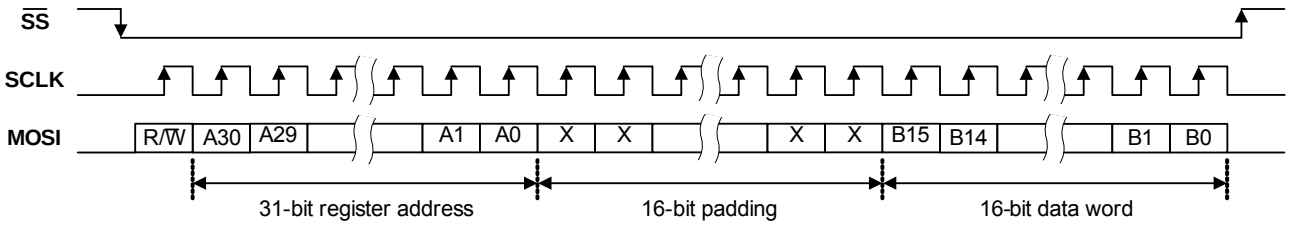


Figure 4-66. Control Interface SPI Register Write (16-Bit Data Words)

Fig. 4-67 shows a single register read from a specified address.

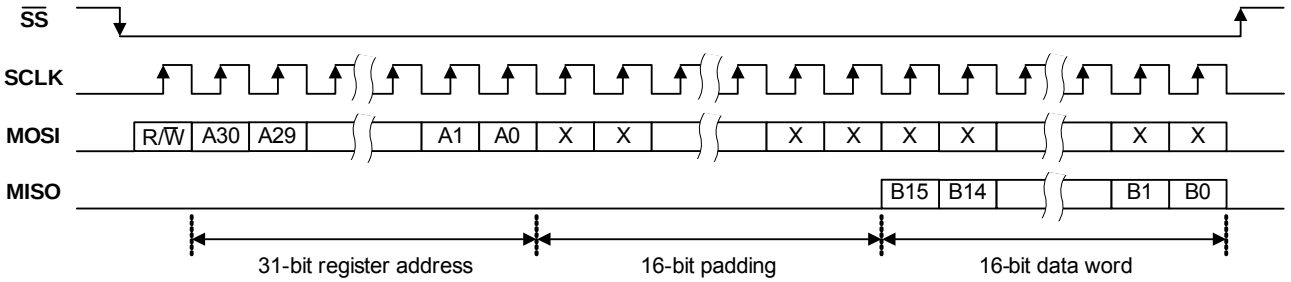


Figure 4-67. Control Interface SPI Register Read (16-Bit Data Words)

4.17.2 Two-Wire (I²C) Control Mode

The I²C control interface mode is supported on CIF2 only and uses the respective SCLK and SDA pins.

In I²C Mode, the CS47L35 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 CS47L35 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 CS47L35).

The CS47L35 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 CS47L35 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 CS47L35 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 CS47L35, the CS47L35 responds by pulling SDA low on the next clock pulse (ACK). If the device ID is not recognized or the R/$\overline{W}$ bit is set incorrectly, the CS47L35 returns to the idle condition and waits for a new start condition.

If the device ID matches the device ID of the CS47L35, 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 CS47L35 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 on CIF2 must use the applicable I²C message format, according to the register address space that is being accessed:

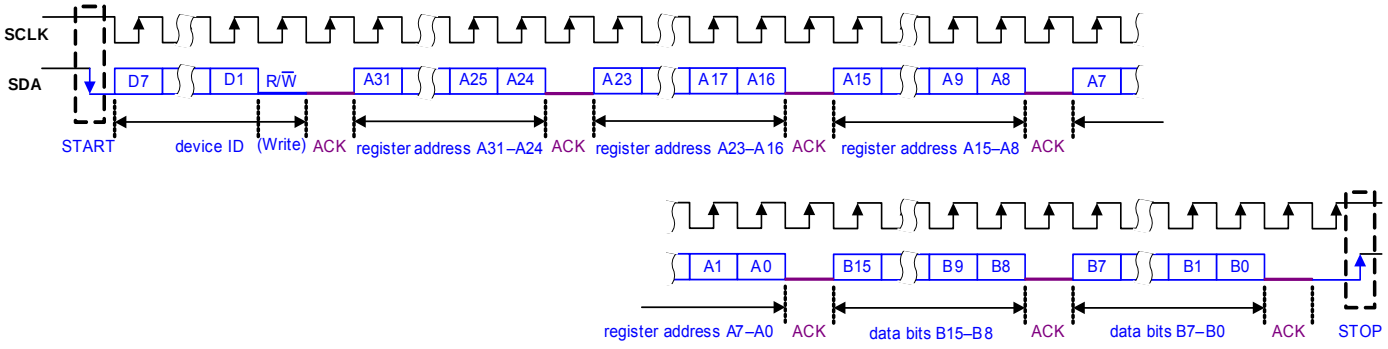
- When accessing register addresses below R12288 (0x3000), the applicable I²C protocol comprises a 32-bit register address and 16-bit data words.
- When accessing register addresses from R12888 (0x3000) upwards, the applicable I²C protocol comprises a 32-bit register address and 32-bit data words.
- 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. 4-68 and Fig. 4-69).

The CS47L35 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 CS47L35 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.

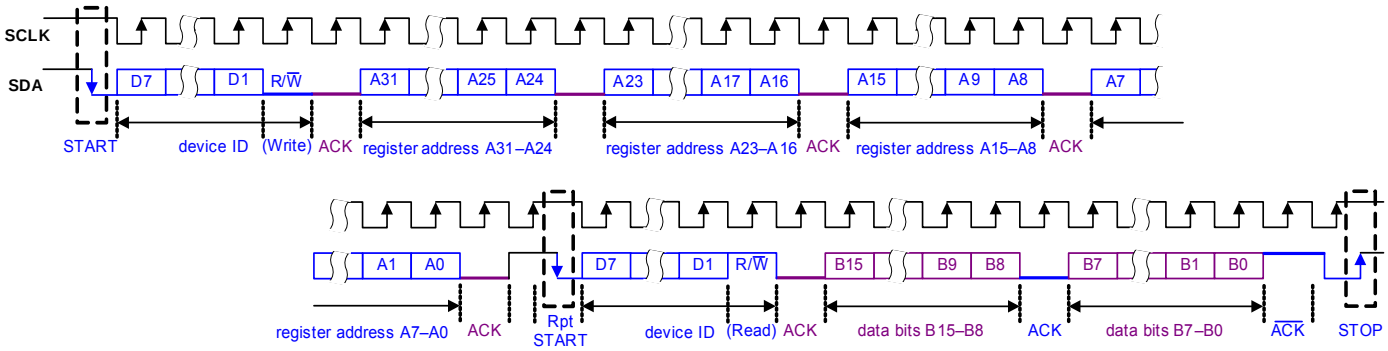
The I²C protocol for a single, 16-bit register write operation is shown in Fig. 4-68.



Note: The SDA pin is used as input for the control register address and data. SDA is pulled low by the receiving device to provide the acknowledge(ACK) response

Figure 4-68. Control Interface I²C Register Write (16-Bit Data Words)

The I²C protocol for a single, 16-bit register read operation is shown in Fig. 4-69.



Note: The SDA pin is driven by both the master and slave devices in turn to transfer device address, register address, data and ACK responses

Figure 4-69. Control Interface I²C Register Read (16-Bit Data Words)

The control interface also supports other register operations; the interface protocol for these operations is shown in Fig. 4-70 through Fig. 4-73. The terminology used in the following figures is detailed in Table 4-101.

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 4-101. Control Interface (I²C) Terminology

Terminology	Description
S	Start condition
Sr	Repeated start
A	Acknowledge (SDA low)
$\overline{A}$	Not acknowledge (SDA high)
P	Stop condition
R/$\overline{W}$	Read/not write 0 = Write; 1 = Read
[White field]	Data flow from bus master to CS47L35
[Gray field]	Data flow from CS47L35 to bus master

Fig. 4-70 shows a single register write to a specified address.

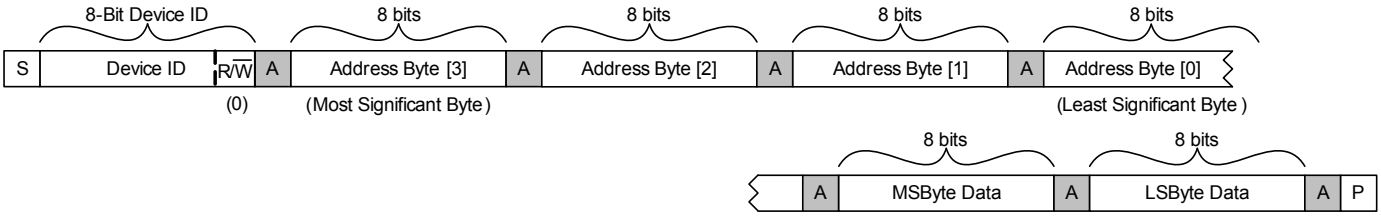


Figure 4-70. Single-Register Write to Specified Address

Fig. 4-71 shows a single register read from a specified address.

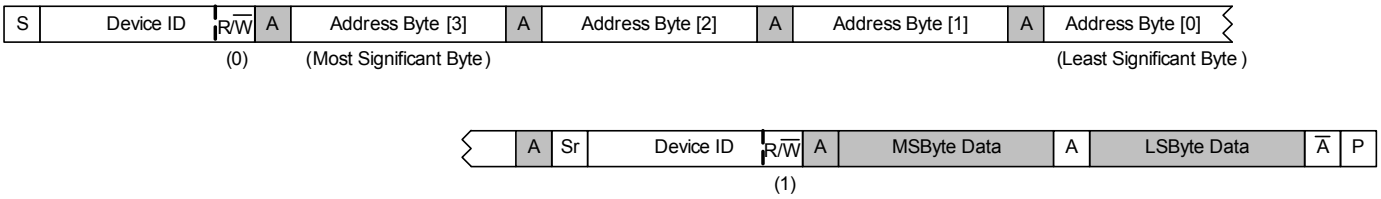


Figure 4-71. Single-Register Read from Specified Address

Fig. 4-72 shows a multiple register write to a specified address.

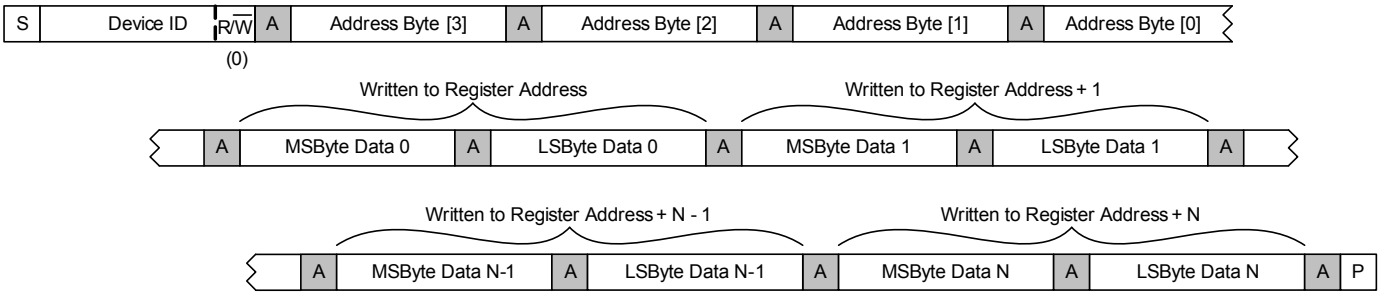


Figure 4-72. Multiple-Register Write to Specified Address

Fig. 4-73 shows a multiple register read from a specified address.

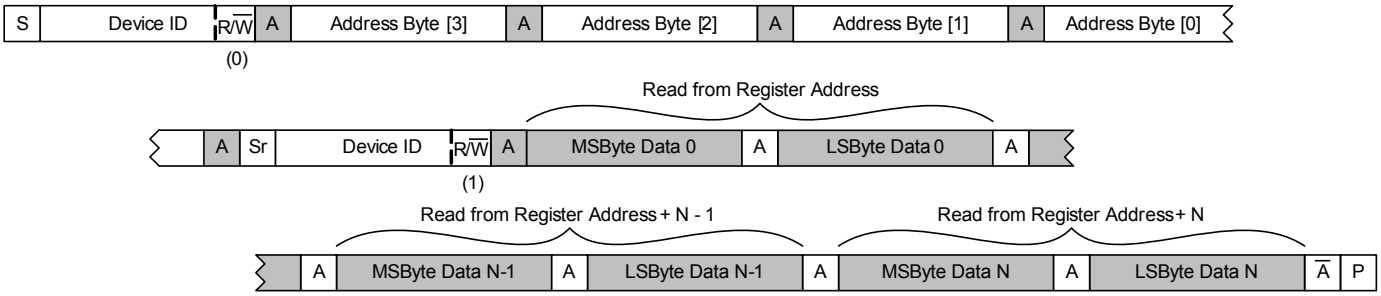


Figure 4-73. Multiple-Register Read from Specified Address

4.18 Control-Write Sequencer

The control-write sequencer is a programmable unit that forms part of the CS47L35 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.

4.18.1 Initiating a Sequence

The fields associated with running the control-write sequencer are described in Table 4-102.

The CS47L35 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 4-102. 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 4-108) 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 4-109 for a description of these registers.

Table 4-102. Write Sequencer Control—Initiating a Sequence

Register Address	Bit	Label	Default	Description
R22 (0x0016) Write_Sequencer_Ctrl_0	11	WSEQ_ABORT	0	Writing 1 to this bit aborts the current sequence.
	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).
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

Table 4-102. Write Sequencer Control—Initiating a Sequence *(Cont.)*

Register Address	Bit	Label	Default	Description
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).
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).

4.18.2 Automatic Sample-Rate Detection Sequences

The CS47L35 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 4-93.

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 4-103.

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 4.16 for further details of the automatic sample-rate detection function.

Table 4-103. 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).

4.18.3 DRC Signal-Detect Sequences

The DRC function within the CS47L35 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 4-13 and Table 4-14 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 4-13.

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 4-104. 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 4-104. 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).

4.18.4 MICDET Clamp Sequences

The CS47L35 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 4-75.

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 4-75.

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 4-105. 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 4.12 for further details of the MICDET clamp status signals.

Table 4-105. 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).

4.18.5 Event Logger Sequences

The CS47L35 provides four 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 to 4), as described in Table 4-106.

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 4.5.2 for further details of the event loggers.

Table 4-106. 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).
R122 (0x007A) Eventlog_ Sequence_ Select_3	8:0	WSEQ_ EVENTLOG3_ INDEX[8:0]	0x1FF	Event Log 3 Write Sequence start index. Contains the index location in the sequencer memory of the first command in the sequence associated with Event Log 3 FIFO Not-Empty detection. Valid from 0 to 251 (0x0FB).
R123 (0x007B) Eventlog_ Sequence_ Select_4	8:0	WSEQ_ EVENTLOG4_ INDEX[8:0]	0x1FF	Event Log 4 Write Sequence start index. Contains the index location in the sequencer memory of the first command in the sequence associated with Event Log 4 FIFO Not-Empty detection. Valid from 0 to 251 (0x0FB).

4.18.6 Boot Sequence

The CS47L35 executes a boot sequence following power-on reset, hardware reset, software reset, or wake-up from Sleep Mode. The boot sequence configures the CS47L35 with factory-set trim (calibration) data. See Section 4.22 and Section 4.23 for further details.

The start index location of the boot sequence is 224 (0x0E0). See Table 4-111 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 4-107. 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).

4.18.7 Sequencer Status Indication

The status of the write sequencer can be read using WSEQ_BUSY and WSEQ_CURRENT_INDEX, as described in Table 4-108. 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 4-108. 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.

4.18.8 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 $((2^{WSEQ\_DELAY}) - 1)$

  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 4-109. The equivalent definitions also apply to Step 1 through Step 251, in the subsequent register address locations.

Table 4-109. 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 011 = 4 bits 110 = 7 bits 001 = 2 bits 100 = 5 bits 111 = 8 bits 010 = 3 bits 101 = 6 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^{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.

4.18.9 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 CS47L35 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 4-110. 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 4-111. 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 4-102 through Table 4-106.

Table 4-111. Write Sequencer Memory Allocation

Description	Sequence Index Range
Default Sequences	0 to 114
User Space	115 to 223
Boot Sequence	224 to 251

4.19 Charge Pumps, Regulators, and Voltage Reference

The CS47L35 incorporates two charge-pump circuits and an LDO-regulator circuit to generate supply rails for internal functions and to support external microphone requirements. The CS47L35 also provides two MICBIAS generators (with four 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.

The CPVDD1 domain (1.8 V) powers the Charge Pump 1 and Charge Pump 2 circuits. The CPVDD2 power domain (1.2 V) is an additional supply used by Charge Pump 1 only.

4.19.1 Charge Pump 1

Charge Pump 1 (CP1) is used to generate the positive and negative supply rails for the analog output drivers. CP1 is enabled automatically by the CS47L35 when required by the output drivers.

The Charge Pump 1 circuit is shown in Fig. 4-74.

4.19.2 Charge Pump 2 and LDO2 Regulator

Charge Pump 2 (CP2) powers LDO2, which provides the supply rail for analog input circuits and for the MICBIAS generators. CP2 and LDO2 are enabled by setting CP2_ENA.

The 32-kHz clock must be configured and enabled when using CP2. See Section 4.16 for details of the system clocks.

When CP2 and LDO2 are enabled, the MICVDD voltage is selected using the LDO2_VSEL field. Note that, when one or more of the MICBIAS generators is operating in normal (regulator) mode, the MICVDD voltage must be at least 200 mV greater than the highest selected MICBIASn output voltages.

When CP2 and LDO2 are enabled, an internal bypass path may be selected, connecting the MICVDD pin directly to the CPVDD1 supply. This path is controlled using the CP2_BYPASS bit. Note that the bypass path is only supported when CP2 is enabled.

When CP2 is disabled, the CP2VOUT pin can be configured to be floating or to be actively discharged. This is selected using the CP2_DISCH bit.

When LDO2 is disabled, the MICVDD pin can be configured to be floating or to be actively discharged. This is selected using the LDO2_DISCH bit.

The MICVDD pin is connected to the output of LDO2. Note that the MICVDD does not support direct connection to an external supply; MICVDD is always powered internally to the CS47L35.

The Charge Pump 2 and LDO2 Regulator circuits are shown in Fig. 4-74. The associated control bits are described in Table 4-112.

Note that decoupling capacitors and flyback capacitors are required for these circuits. Refer to Section 5.1 for recommended external components.

4.19.3 Microphone Bias (MICBIAS) Control

There are two MICBIAS generators, which provide low-noise reference voltages suitable for biasing ECM-type microphones or powering digital microphones. Refer to Section 5.1.3 for recommended external components.

The MICBIAS generators are powered from MICVDD, which is generated by an internal charge pump and LDO, as shown in Fig. 4-74.

Switchable outputs from the MICBIAS generators allow four separate reference/supply outputs to be independently controlled. The MICBIAS regulators are enabled using the MICB1_ENA and MICB2_ENA bits. The MICBIAS output switches are enabled using MICB1A_ENA, MICB1B_ENA, MICB2A_ENA, and MICB2B_ENA.

Note that, to enable any of the MICBIASnx outputs, both the output switch and the respective regulator must 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 MICB*n*_DISCH bits (for the MICBIAS regulators), and the MICBnx_DISCH bits (for the switched outputs). Each discharge path is only effective when the respective regulator, or switched output, is disabled.

The MICBIAS generators can each operate in Regulator Mode or in Bypass Mode. The applicable mode is selected using the MICB*n*_BYPASS bits.

In Regulator Mode (MICB*n*_BYPASS = 0), the output voltage is selected using the MICB*n*_LVL fields. In this mode, MICVDD must be at least 200mV greater than the required MICBIAS output voltages. The MICBIAS outputs are powered from the MICVDD pin and use the internal band-gap circuit as a reference.

In Regulator Mode, the MICBIAS regulators are designed to operate without external decoupling capacitors. The regulators can be configured to support a capacitive load if required, using the MICB*n*_EXT_CAP bits. (This may be appropriate for a DMIC supply.) It is important that the external capacitance is compatible with the applicable MICB*n*_EXT_CAP setting. The compatible load conditions are detailed in Table 3-11.

In Bypass Mode (MICB*n*_BYPASS = 1), the respective outputs (MICBIAS*nx*), when enabled, are connected directly to MICVDD. This enables a low power operating state. Note that the MICB*n*_EXT_CAP settings are not applicable in Bypass Mode—there are no restrictions on the external MICBIAS capacitance in Bypass Mode.

The MICBIAS generators incorporate 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 MICB*n*_RATE bits.

The MICBIAS generators are shown in Fig. 4-74. The MICBIAS control fields are described in Table 4-112.

The maximum output current for each MICBIAS regulator is noted in Table 3-11. This limit must be observed for each pair of MICBIAS*nx* outputs, especially if more than one microphone is connected to a single regulator. Note that the maximum output current differs between Regulator Mode and Bypass Mode.

4.19.4 Voltage-Reference Circuit

The CS47L35 incorporates a voltage-reference circuit, powered by AVDD. This circuit ensures the accuracy of the LDO-regulator and MICBIAS voltage settings.

4.19.5 Block Diagram and Control Registers

The charge-pump and regulator circuits are shown in Fig. 4-74. Note that decoupling capacitors and flyback capacitors are required for these circuits. Refer to Section 5.1 for recommended external components.

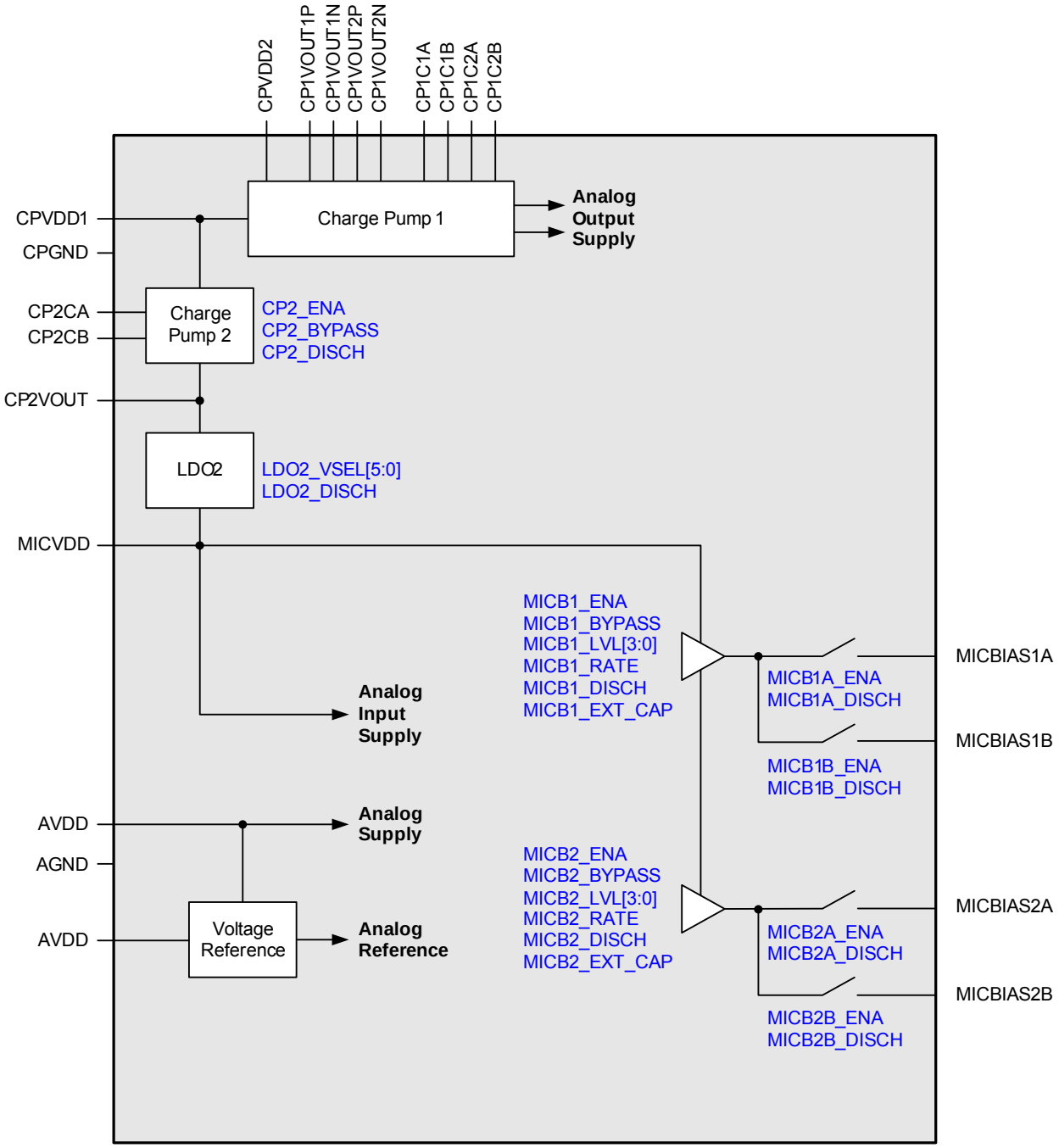


Figure 4-74. Charge Pumps and Regulators

The charge-pump and regulator control registers are described in Table 4-112.

Table 4-112. Charge-Pump and LDO Control Registers

Register Address	Bit	Label	Default	Description
R512 (0x0200) Mic_Charge_ Pump_1	2	CP2_DISCH	1	Charge Pump 2 Discharge 0 = CP2VOUT floating when disabled 1 = CP2VOUT discharged when disabled
	1	CP2_BYPASS	1	Charge Pump 2 and LDO2 Bypass Mode 0 = Normal 1 = Bypass Mode In Bypass Mode, CPVDD1 is connected directly to MICVDD. Note that CP2_ENA must also be set.
	0	CP2_ENA	1	Charge Pump 2 and LDO2 Control (Provides analog input and MICVDD supplies) 0 = Disabled 1 = Enabled
R531 (0x0213) LDO2_Control_1	10:5	LDO2_VSEL[5:0]	0x1F	LDO2 Output Voltage Select [1] 0x00 = 0.900 V; 0x13 = 1.375 V; … (100-mV steps) 0x01 = 0.925 V; 0x14 = 1.400 V; 0x26 = 3.200 V 0x02 = 0.950 V; 0x15 = 1.500 V; 0x27 to 0x3F = 3.300 V … (25-mV steps); 0x16 = 1.600 V
	2	LDO2_DISCH	1	LDO2 Discharge 0 = MICVDD floating when disabled 1 = MICVDD discharged when disabled
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 4-112. Charge-Pump and LDO Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R537 (0x0219) Mic_Bias_Ctrl_2	15	MICB2_EXT_CAP	0	Microphone Bias 2 External Capacitor (when MICB2_BYPASS = 0). Configures the MICBIAS2 regulator according to the specified capacitance connected to the MICBIAS2x outputs. 0 = No external capacitor 1 = External capacitor connected
	8:5	MICB2_LVL[3:0]	0x7	Microphone Bias 2 Voltage Control (when MICB2_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	MICB2_RATE	0	Microphone Bias 2 Rate (Bypass Mode) 0 = Fast start-up/shutdown 1 = Pop-free start-up/shutdown
	2	MICB2_DISCH	1	Microphone Bias 2 Discharge 0 = MICBIAS2 floating when disabled 1 = MICBIAS2 discharged when disabled
	1	MICB2_BYPASS	1	Microphone Bias 2 Mode 0 = Regulator Mode 1 = Bypass Mode
	0	MICB2_ENA	0	Microphone Bias 2 Enable 0 = Disabled 1 = Enabled
R540 (0x021C) Mic_Bias_Ctrl_5	5	MICB1B_DISCH	0	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	0	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
R542 (0x021E) Mic_Bias_Ctrl_6	5	MICB2B_DISCH	0	Microphone Bias 2B Discharge 0 = MICBIAS2B floating when disabled 1 = MICBIAS2B discharged when disabled
	4	MICB2B_ENA	0	Microphone Bias 2B Enable 0 = Disabled 1 = Enabled
	1	MICB2A_DISCH	0	Microphone Bias 2A Discharge 0 = MICBIAS2A floating when disabled 1 = MICBIAS2A discharged when disabled
	0	MICB2A_ENA	0	Microphone Bias 2A Enable 0 = Disabled 1 = Enabled

1.See Table 4-113 for LDO2 output voltage definition.

Table 4-113 lists the LDO2 voltage control settings.

Table 4-113. LDO2 Voltage Control

LDO2_VSEL[5:0]	LDO Output	LDO2_VSEL[5:0]	LDO Output	LDO2_VSEL[5:0]	LDO Output
0x00	0.900 V	0x10	1.300 V	0x20	2.600 V
0x01	0.925 V	0x11	1.325 V	0x21	2.700 V
0x02	0.950 V	0x12	1.350 V	0x22	2.800 V
0x03	0.975 V	0x13	1.375 V	0x23	2.900 V
0x04	1.000 V	0x14	1.400 V	0x24	3.000 V
0x05	1.025 V	0x0E	1.250 V	0x1C	2.200 V

Table 4-113. LDO2 Voltage Control *(Cont.)*

LDO2_VSEL[5:0]	LDO Output
0x06	1.050 V
0x07	1.075 V
0x08	1.100 V
0x09	1.125 V
0x0A	1.150 V
0x0B	1.175 V
0x0C	1.200 V
0x0D	1.225 V

LDO2_VSEL[5:0]	LDO Output
0x0F	1.275 V
0x15	1.500 V
0x16	1.600 V
0x17	1.700 V
0x18	1.800 V
0x19	1.900 V
0x1A	2.000 V
0x1B	2.100 V

LDO2_VSEL[5:0]	LDO Output
0x1D	2.300 V
0x1E	2.400V
0x1F	2.500 V
0x25	3.100 V
0x26	3.200 V
0x27	3.300 V
0x28 to 0x3F	3.300 V

4.20 JTAG Interface

The JTAG interface provides test and debug access to the CS47L35 DSP core. The interface comprises five pins, as detailed below.

- TCK: clock input
- TDI: data input
- TDO: data output
- TMS: mode select input
- TRST: test access port reset input (active low)

For normal operation (test and debug access disabled), the JTAG interface should be held in reset (i.e., TRST should be at Logic 0). An internal pull-down resistor holds the TRST pin low when not actively driven. External connection to DGND is recommended, if the JTAG interface function is not required.

The other JTAG input pins (TCK, TDI, TMS) should also be held at Logic 0 for normal operation. An internal pull-down resistor holds these pins low when not actively driven.

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 4.23 for further details of the CS47L35 software reset.

4.21 Thermal, Short-Circuit, and Timer-Controlled Protection

The CS47L35 incorporates thermal protection, short-circuit detection, and timer-controlled speaker disable functions; these are described in the following subsections.

4.21.1 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 4.15. A two-stage indication is provided, via the SPK_OVERHEAT_WARN_EINTn and SPK_OVERHEAT_EINTn interrupts.

If the upper temperature threshold (SPK_OVERHEAT_EINTn) 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_EINTn, is asserted.

4.21.2 Short Circuit Protection

The short-circuit detection function for the Class D speaker outputs is triggered when the respective output drivers are enabled (see Table 4-64). If a short circuit is detected at this time, the enable does not succeed, and the respective 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 4.15. If the Class D speaker short-circuit condition is detected, the respective drivers are automatically disabled in order to protect the device. When the speaker driver shutdown is complete, a further interrupt, SPK_SHUTDOWN_EINTn, 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 functions provide input to the interrupt control circuit and can be used to trigger an interrupt event when a short-circuit condition is detected; see Section 4.15.

4.21.3 Timer-Controlled Speaker Shutdown

The general-purpose timers (see Section 4.5.3) can also be used to trigger a shutdown of the Class D speaker drivers. This is configured using the SPK_SHUTDOWN_TIMER_SEL field, as described in Table 4-114.

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 drivers are automatically disabled. When the driver shutdown is complete, an interrupt event (SPK_SHUTDOWN_EINTn) is signaled.

To enable the Class D speaker outputs following a timeout condition, the host processor must disable and reenable the output drivers.

Table 4-114. 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. Unlisted codes are reserved 0x0 = Disabled 0x1 = Timer 1 0x2 = Timer 2 0x3 = Timer 3 0x4 = Timer 4 All other codes are reserved

4.21.4 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 4.14 to configure a GPIO pin for this function.

4.22 Power-On Reset (POR)

The CS47L35 remains in the reset state until AVDD, DBVDD1, 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.

After the initial power-up, the POR is rescheduled following an interruption to the DBVDD1 or AVDD supplies.

If the CS47L35 SLIMbus component is in its operational state, it must be reset before scheduling a POR. See Section 4.10 for details of the SLIMbus reset control messages.

4.22.1 Boot Sequence

Following power-on reset, a boot sequence is executed. The BOOT_DONE_STS*x* bits are asserted on completion of the boot sequence, as described in Table 4-115. Control-register writes should not be attempted until BOOT_DONE_STS*x* has been asserted. Note that the BOOT_DONE_STS1 and BOOT_DONE_STS*2* bits provide the same information.

The BOOT_DONE_STS*x* signal 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 4.15. 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 4.18.

Table 4-115. 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.

4.22.2 Digital I/O Status in Reset

Table 4-116 describes the default status of the CS47L35 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 (see Section 4.23).

The same default conditions are applicable following a wake-up transition, except for the IRQ and 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 4-116 are not valid if modified by the boot sequence or by a wake-up control sequence. See Section 4.18 for details of these functions.

Table 4-116. CS47L35 Digital I/O Status in Reset

Power Domain	Pin No	Name	Type	Reset Status
MICVDD [1]	E3	IN1ALN/DMICCLK1	Analog input/Digital output	Analog input
	C1	IN1ARN/DMICDAT1	Analog input/Digital input	Analog input
	H4	IN2LN/DMICCLK2	Analog input/Digital output	Analog input
	E1	IN2RN/DMICDAT2	Analog input/Digital input	Analog input
DBVDD1	T12	AIF1BCLK/GPIO9	Digital I/O	Digital input (GPIO), bus-keeper enabled [2]
	R11	AIF1LRCLK/GPIO11	Digital I/O	Digital input (GPIO), bus-keeper enabled [2]
	P10	AIF1RXDAT/GPIO8	Digital I/O	Digital input (GPIO), bus-keeper enabled [2]
	T10	AIF1TXDAT/GPIO10	Digital I/O	Digital input (GPIO), bus-keeper enabled [2]
	M10	CIF1MISO	Digital output	Digital output
	L9	CIF1MOSI	Digital input	Digital input
	L11	CIF1SCLK	Digital input	Digital input
	K10	CIF1SS	Digital input	Digital input
	P12	CIF2SCLK	Digital input	Digital input
	N11	CIF2SDA	Digital I/O	Digital input
	J9	IRQ	Digital output	Digital output
	M12	MCLK1	Digital input	Digital input
	T8	MIF1SCLK/GPIO16	Digital I/O	Digital input (GPIO), bus-keeper enabled [2]
	P8	MIF1SDA/GPIO7	Digital I/O	Digital input (GPIO), bus-keeper enabled [2]
	H8	RESET	Digital input	Digital input, Pull-up to DBVDD1
	R9	SLIMCLK	Digital I/O	Digital input
	N9	SLIMDAT	Digital I/O	Digital input

Table 4-116. CS47L35 Digital I/O Status in Reset *(Cont.)*

Power Domain	Pin No	Name	Type	Reset Status
DBVDD2	P6	AIF2BCLK/GPIO13	Digital I/O	Digital input (GPIO), bus-keeper enabled [2]
	M6	AIF2LRCLK/GPIO15	Digital I/O	Digital input (GPIO), bus-keeper enabled [2]
	K6	AIF2RXDAT/GPIO14	Digital I/O	Digital input (GPIO), bus-keeper enabled [2]
	L5	AIF2TXDAT/GPIO12	Digital I/O	Digital input (GPIO), bus-keeper enabled [2]
	R5	AIF3BCLK/GPIO2	Digital I/O	Digital input (GPIO), bus-keeper enabled [2]
	N5	AIF3LRCLK/GPIO4	Digital I/O	Digital input (GPIO), bus-keeper enabled [2]
	P4	AIF3RXDAT/GPIO3	Digital I/O	Digital input (GPIO), bus-keeper enabled [2]
	M4	AIF3TXDAT/GPIO1	Digital I/O	Digital input (GPIO), bus-keeper enabled [2]
	T6	MCLK2	Digital input	Digital input
	M2	SPKCLK/GPIO6	Digital I/O	Digital input (GPIO), bus-keeper enabled [2]
	N3	SPKDAT/GPIO5	Digital I/O	Digital input (GPIO), bus-keeper enabled [2]
	R7	TCK	Digital input	Digital input, Pull-down to DGND
	N7	TDI	Digital input	Digital input, Pull-down to DGND
	L7	TDO	Digital output	Digital output
	K8	TMS	Digital input	Digital input, Pull-down to DGND
	M8	TRST	Digital input	Digital input, Pull-down to DGND

1. The dual-function INnLN/DMICCLKn and INnRN/DMICDATn pins default to their respective analog input functions after power-on reset is completed. The analog input functions are referenced to the MICVDD power domain.
2. The power-up condition of the GPIO pins depends upon whether the pin is actively driven by another device when the CS47L35 starts up. 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 default input level.

4.23 Hardware Reset, Software Reset, Wake-Up, and Device ID

The CS47L35 supports hardware- and software-controlled reset functions. The reset functions, and the Sleep/Wake-Up state transitions, provide similar (but not identical) functionality. Each of these is described in the following subsections.

The CS47L35 device ID can be read from the Software_Reset (R0) control register, as described in Section 4.23.7.

4.23.1 Hardware Reset

The CS47L35 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 DBVDD1 power domain. A hardware reset causes all of the CS47L35 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 4-117. When the pull-up and pull-down resistors are both enabled, the CS47L35 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).

If the CS47L35 SLIMbus component is in its operational state, it must be reset prior to scheduling a hardware reset. See Section 4.10 for details of the SLIMbus reset control messages.

Table 4-117. Reset Pull-Up/Pull-Down Configuration

Register Address	Bit	Label	Default	Description
R6864 (0x1AD0) AOD_Pad_Ctrl	1	RESET_PU	1	$\overline{\text{RESET}}$ 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 $\overline{\text{RESET}}$ pin.
	0	RESET_PD	0	$\overline{\text{RESET}}$ 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 $\overline{\text{RESET}}$ pin.

4.23.2 Software Reset

A software reset is executed by writing any value to register R0. A software reset causes most of the CS47L35 control registers to be reset to their default states. Note that the control-write sequencer memory is retained during software reset.

Note that the first register read/write operation following a software reset may be unsuccessful, if the register access is attempted via a different control interface to the one that commanded the software reset. Note that only the first register read/write is affected, and only when using more than one control interface.

4.23.3 Wake-Up

The CS47L35 is in Sleep Mode when AVDD and DBVDD1 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 4.13.)

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 4.13 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 4.13 for details of the always-on functions.

4.23.4 Write Sequencer and DSP Firmware Memory Control in Reset and Wake-Up

The control-write sequencer memory contents reverts to its default contents 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 CS47L35 memory reset conditions.

4.23.5 Boot Sequence

Following hardware reset, software reset, or wake-up from Sleep Mode, a boot sequence is executed. The BOOT_DONE_STS*x* bits (see Table 4-115) are deasserted during hardware reset and software reset, and also in Sleep Mode. The BOOT_DONE_STS*x* bits are asserted on completion of the boot sequence. Control register writes should not be attempted until BOOT_DONE_STSx has been asserted.

The BOOT_DONE_STSx status is an input to the interrupt control circuit and can be used to trigger an interrupt event; see Section 4.15. Note that the BOOT_DONE_STS1 and BOOT_DONE_STS*2* bits provide the same information.

For details of the boot sequence, see Section 4.18.

4.23.6 Digital I/O Status in Reset

The status of the CS47L35 digital I/O pins following hardware reset, software reset, or wake-up is described in Section 4.22.2.

4.23.7 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 4-118. Device Reset and ID

Register Address	Bit	Label	Default	Description
R0 (0x0000) Software_Reset	15:0	SW_RST_DEV_ID[15:0]	0x6360	Writing to this register resets all registers to their default state. Reading from this register indicates Device ID 0x6360.
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 CS47L35.

5.1.1 Analog Input Paths

The CS47L35 supports up to six analog audio input connections. Each of these inputs is biased to the internal DC reference, VREF. (Note that this reference voltage is present on the VREFC pin.) A DC-blocking capacitor is required for each analog input pin used in the target application. 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.

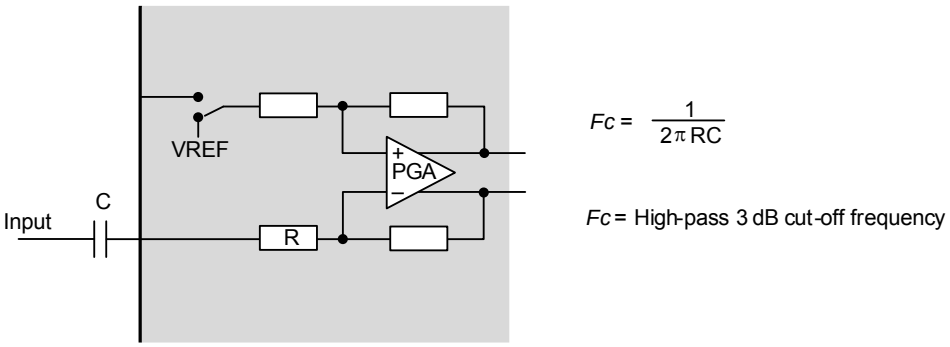


Figure 5-1. Audio Input Path DC-Blocking Capacitor

In accordance with the CS47L35 input pin resistance (see Table 3-5), a 1-µF capacitance for all input connections 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 CS47L35 microphone bias circuit, are shown in Section 5.1.3; see Fig. 5-2.

5.1.2 DMIC Input Paths

The CS47L35 supports up to four channels of DMIC input; two channels of audio data can be multiplexed on each DMICDAT*n* pin. Each of these stereo pairs is clocked using the respective DMICCLK*n* pin.

The external connections for digital microphones, incorporating the CS47L35 microphone bias circuit, are shown in Fig. 5-4. 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 a single DMICDAT pin, 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 CS47L35 samples the DMIC data at the end of each DMICCLK phase. Each microphone must tristate its data output when the other microphone is transmitting. Integrated pull-down resistors can be enabled on the DMICDAT pins if required.

The voltage reference for each DMIC interface is selectable. It is important that the selected reference for the CS47L35 interface is compatible with the applicable configuration of the external microphone.

5.1.3 Microphone Bias Circuit

The CS47L35 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 regulators on the CS47L35. Two MICBIAS generators are available; switchable outputs allow four separate reference/supply outputs to be independently controlled.

Note that the MICVDD pin can also be used (instead of MICBIAS*nx*) 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 CS47L35 is not exceeded.

A 2.2-kΩ bias resistor is recommended; this provides compatibility with a wide range of microphone components.

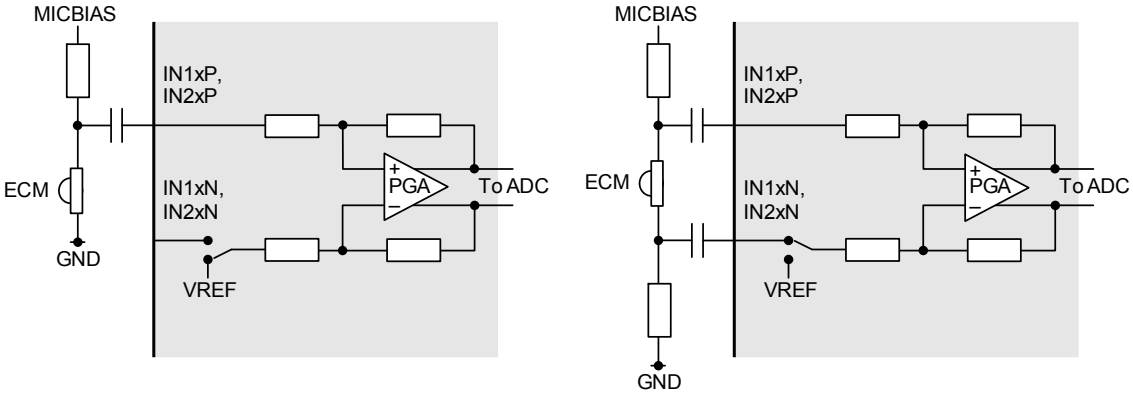


Figure 5-2. Single-Ended and Differential ECM Microphone Connections

Analog MEMS microphones can be connected to the CS47L35 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.

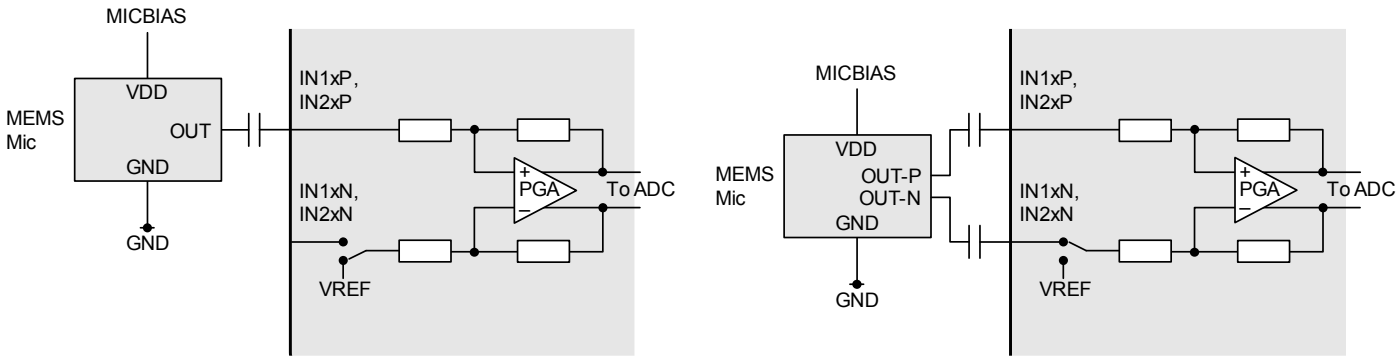


Figure 5-3. Single-Ended and Differential Analog MEMS Microphone Connections

DMIC connection to the CS47L35 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.

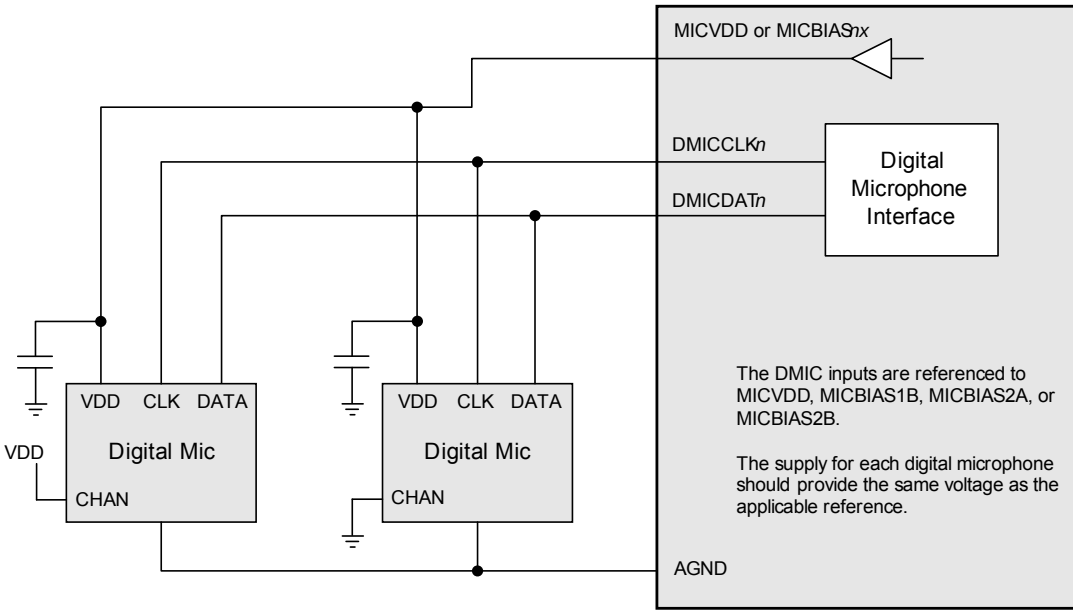


Figure 5-4. DMIC Connection

Each MICBIAS generator can operate in Regulator Mode or in Bypass Mode. See Section 4.19 for details of the MICBIAS generators.

In Regulator Mode, the MICBIAS regulators are designed to operate without external decoupling capacitors. The regulators 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 MICBIAS1 or MICBIAS2 exceeds the specified conditions for Regulator Mode (e.g., due to a decoupling capacitor or long PCB trace), the respective generator must be configured in Bypass Mode.

The maximum output current for each MICBIAS regulator is noted in Table 3-11. This limit must be observed for each pair of MICBIAS*nx* outputs, especially if more than one microphone is connected to a single regulator. 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 CS47L35 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 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 outputs. The HPOUT feedback is supported on two pins—the applicable pin is selected using ACCDET_SRC.

The selected feedback pin should be connected to GND as close as possible to the respective headphone jack ground pin, as shown in Fig. 5-5. In mono (differential) mode, the feedback pins should be connected to the ground plane that is closest to the earpiece output PCB tracks.

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.

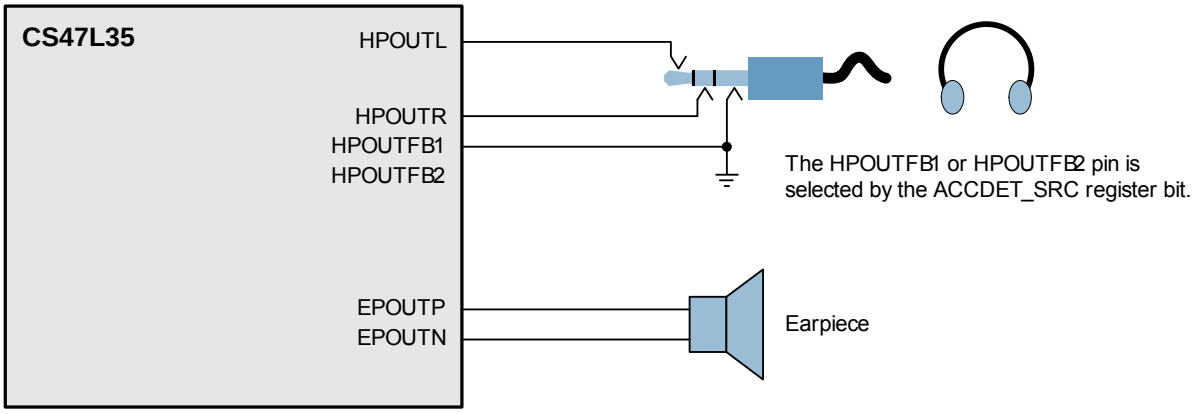


Figure 5-5. Headphone and Earpiece Connection

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 CS47L35 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.

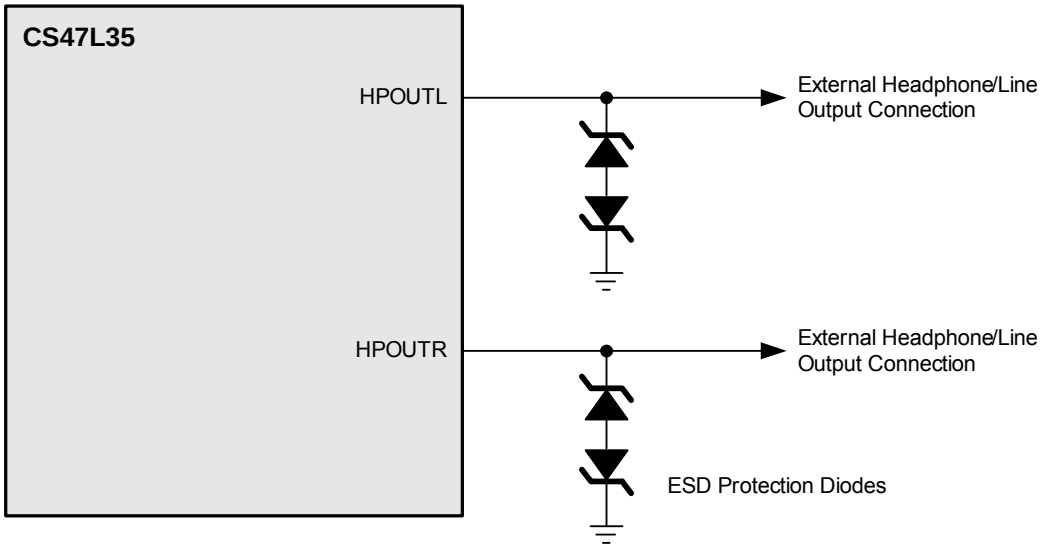


Figure 5-6. ESD Diode Configuration for External Output Connections

5.1.5 Speaker-Driver Output Path

The CS47L35 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 in this mode.

The efficiency of the speaker driver is affected by the series resistance between the CS47L35 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.

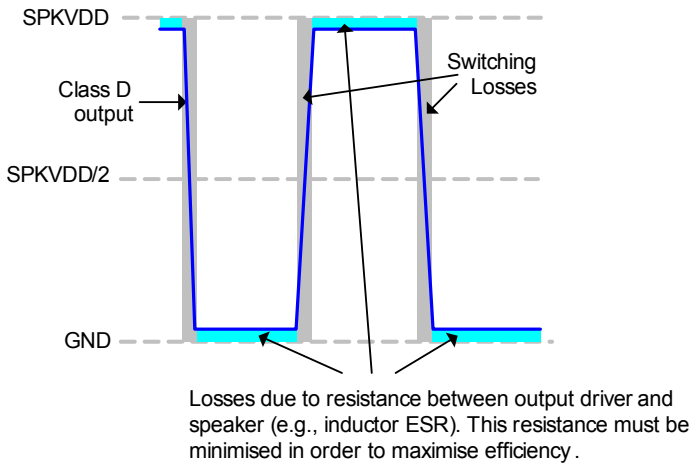


Figure 5-7. Speaker Connection Losses

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.

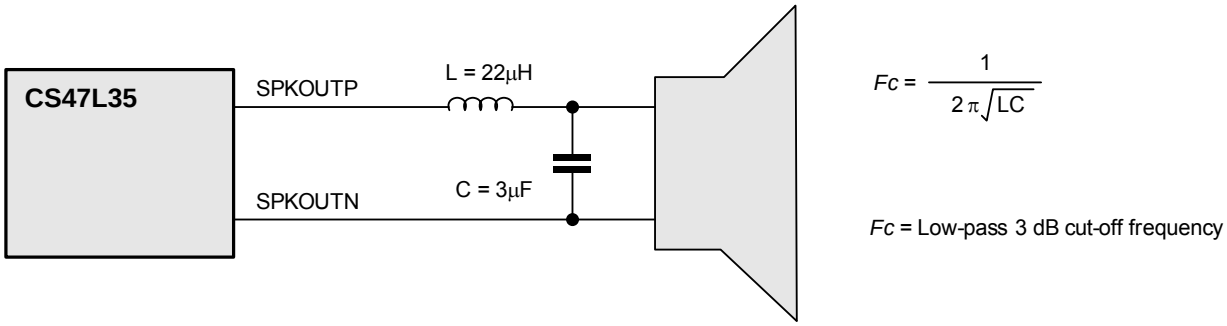


Figure 5-8. Class D Output Filter Components

A simple equivalent circuit of a loudspeaker consists of a serially-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.

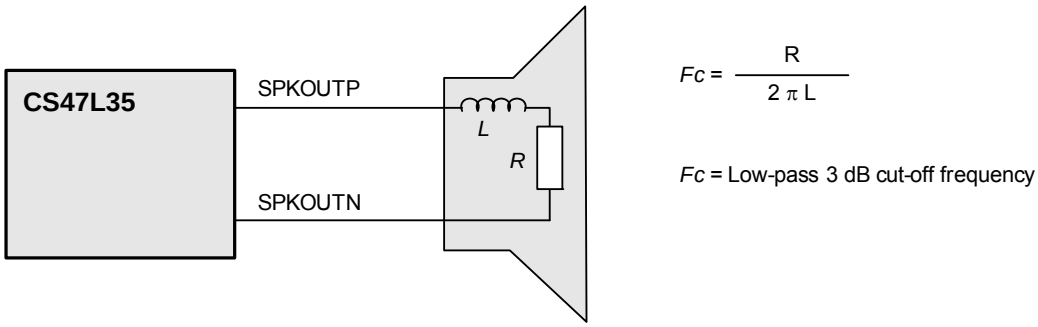


Figure 5-9. Speaker Equivalent Circuit for Filterless Operation

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 CS47L35 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.

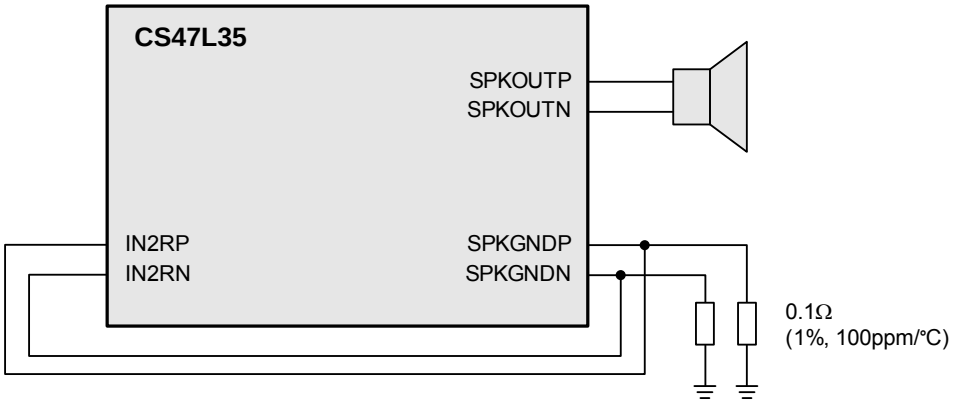


Figure 5-10. Speaker Output Current Monitoring Connections (Speaker Protection)

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 CS47L35, 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 CS47L35 are detailed in Table 5-1.

Table 5-1. Power Supply Decoupling Capacitors

Power Supply	Decoupling Capacitor
AVDD1, AVDD2	2 x 1.0 µF ceramic—one capacitor on each AVDD*n* pin
CPVDD1	4.7 µF ceramic
CPVDD2	4.7 µF ceramic
DBVDD1, DBVDD2	2 x 0.1 µF ceramic [1]—one capacitor on each DBVDD*n* pin
DCVDD	4.7 µF ceramic
FLLVDD	1.0 µF ceramic
MICVDD	4.7 µF ceramic
SPKVDD	4.7 µF ceramic
VREFC	2.2 µF ceramic

1. Total capacitance of 4.7 µF is required for each DBVDDn domain. This can be provided by dedicated DBVDDn decoupling or by other capacitors on the same power rail.

All decoupling capacitors should be placed as close as possible to the CS47L35 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 CS47L35.

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. For most application the use of ceramic capacitors with capacitor dielectric X5R is recommended.

5.1.7 Charge-Pump Components

The CS47L35 incorporates two charge-pump circuits (CP1 and CP2).

CP1 generates the CP1VOUT*nx* supply rails for the ground-referenced headphone drivers; CP2 generates the CP2VOUT supply rail for the microphone bias (MICBIAS) regulators.

Decoupling capacitors are required on each of the charge-pump outputs. Two fly-back capacitors are required for CP1; a single fly-back capacitor is required for CP2.

The recommended charge-pump capacitors for CS47L35 are detailed in Table 5-2.

Table 5-2. Charge-Pump External Capacitors

Description	Capacitor
CP1VOUT1P decoupling	Required capacitance is 2.0 µF at 2 V. Suitable component typically 4.7 µF.
CP1VOUT1N decoupling	Required capacitance is 2.0 µF at 2 V. Suitable component typically 4.7 µF.
CP1 fly-back 1 (connect between CP1C1A and CP1C1B)	Required capacitance is 1.0 µF at 2 V. Suitable component typically 2.2 µF.
CP1VOUT2P decoupling	Required capacitance is 2.0 µF at 2 V. Suitable component typically 4.7 µF.
CP1VOUT2N decoupling	Required capacitance is 2.0 µF at 2 V. Suitable component typically 4.7 µF.
CP1 fly-back 2 (connect between CP1C2A and CP1C2B)	Required capacitance is 1.0 µF at 2 V. Suitable component typically 2.2 µF.
CP2VOUT decoupling	Required capacitance is 1.0 µF at 3.6 V. Suitable component typically 4.7 µF.
CP2 fly-back (connect between CP2CA and CP2CB)	Required capacitance is 220 nF at 2 V. Suitable component typically 470 nF.

Ceramic capacitors are recommended for these charge-pump requirements. Note that, 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.

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 CS47L35. The component choice and positioning of the CP1 components are more critical than those of CP2, due to the higher output power requirements of CP1.

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 MICBIAS*nx* 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 MICBIAS*nx* 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.

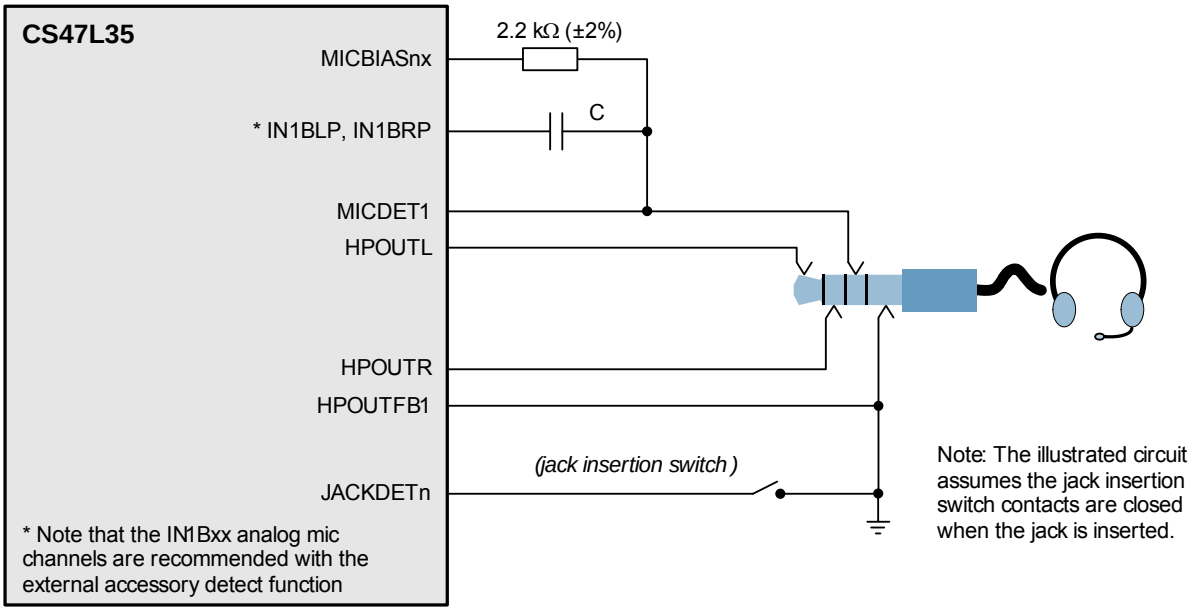


Figure 5-11. External Accessory Detection

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 MICBIAS2A as a reference. The applicable source is configured using MICD_BIAS_SRC.

The CS47L35 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 MICD_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.

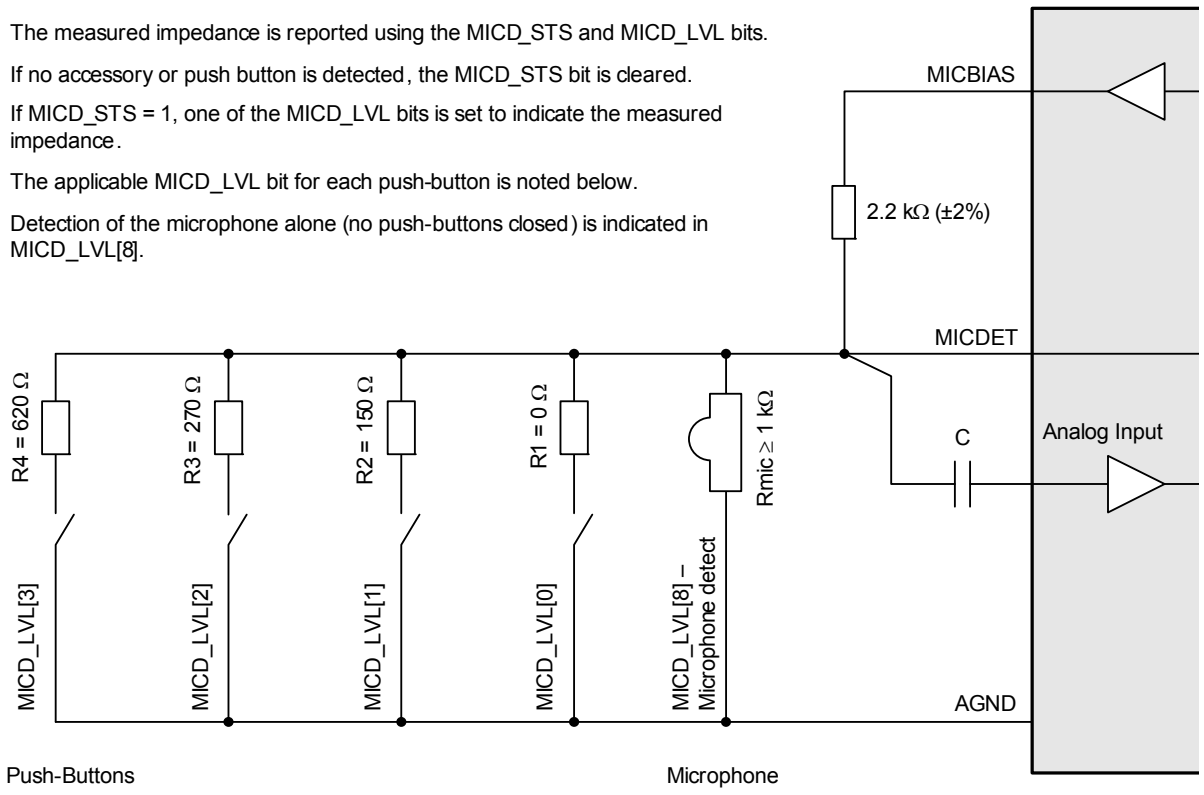


Figure 5-12. External Accessory Detect Components

5.2 Resets Summary

Table 5-3 summarizes of the CS47L35 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 DBVDD1 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 DBVDD1 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 4.13 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 CS47L35 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
R8 (0x0008) Ctrl_IF_CFG_1	8	CIF1MISO_DRV_STR	1	CIF1MISO output drive strength 0 = 4 mA 1 = 8 mA
R9 (0x0009) Ctrl_IF_CFG_2	9	CIF2SDA_DRV_STR	1	CIF2SDA output drive strength 0 = 4 mA 1 = 8 mA
R1520 (0x05F0) Slimbus_Pad_Ctrl	1	SLIMDAT_DRV_STR	0	SLIMDAT output drive strength 0 = 8 mA 1 = 12 mA
R5889 (0x1701) GPIO1_CTRL2	12	GP1_DRV_STR	1	AIF3TXDAT/GPIO1 output drive strength 0 = 4 mA 1 = 8 mA
R5891 (0x1703) GPIO2_CTRL2	12	GP2_DRV_STR	1	AIF3BCLK/GPIO2 output drive strength 0 = 4 mA 1 = 8 mA
R5893 (0x1705) GPIO3_CTRL2	12	GP3_DRV_STR	1	AIF3RXDAT/GPIO3 output drive strength 0 = 4 mA 1 = 8 mA
R5895 (0x1707) GPIO4_CTRL2	12	GP4_DRV_STR	1	AIF3LRCLK/GPIO4 output drive strength 0 = 4 mA 1 = 8 mA
R5897 (0x1709) GPIO5_CTRL2	12	GP5_DRV_STR	1	SPKDAT/GPIO5 output drive strength 0 = 4 mA 1 = 8 mA
R5899 (0x170B) GPIO6_CTRL2	12	GP6_DRV_STR	1	SPKCLK/GPIO6 output drive strength 0 = 4 mA 1 = 8 mA
R5901 (0x170D) GPIO7_CTRL2	12	GP7_DRV_STR	1	MIF1SDA/GPIO7 output drive strength 0 = 4 mA 1 = 8 mA
R5903 (0x170F) GPIO8_CTRL2	12	GP8_DRV_STR	1	AIF1RXDAT/GPIO8 output drive strength 0 = 4 mA 1 = 8 mA
R5905 (0x1711) GPIO9_CTRL2	12	GP9_DRV_STR	1	AIF1BCLK/GPIO9 output drive strength 0 = 4 mA 1 = 8 mA
R5907 (0x1713) GPIO10_CTRL2	12	GP10_DRV_STR	1	AIF1TXDAT/GPIO10 output drive strength 0 = 4 mA 1 = 8 mA
R5909 (0x1715) GPIO11_CTRL2	12	GP11_DRV_STR	1	AIF1LRCLK/GPIO11 output drive strength 0 = 4 mA 1 = 8 mA
R5911 (0x1717) GPIO12_CTRL2	12	GP12_DRV_STR	1	AIF2TXDAT/GPIO12 output drive strength 0 = 4 mA 1 = 8 mA
R5913 (0x1719) GPIO13_CTRL2	12	GP13_DRV_STR	1	AIF2BCLK/GPIO13 output drive strength 0 = 4 mA 1 = 8 mA

Table 5-4. Output Drive-Strength and Slew-Rate Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R5915 (0x171B) GPIO14_CTRL2	12	GP14_DRV_STR	1	AIF2RXDAT/GPIO14 output drive strength 0 = 4 mA 1 = 8 mA
R5917 (0x171D) GPIO15_CTRL2	12	GP15_DRV_STR	1	AIF2LRCLK/GPIO15 output drive strength 0 = 4 mA 1 = 8 mA
R5919 (0x171F) GPIO16_CTRL2	12	GP16_DRV_STR	1	MIF1SCLK/GPIO16 output drive strength 0 = 4 mA 1 = 8 mA

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 CS47L35 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, an AIF*n* or SLIMbus interface 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 CS47L35. 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 SLIMbus interface can also provide the clock reference, via the FLL, provided that the BCLK and LRCLK signals are externally synchronized with the SLIMCLK input.

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, SLIMCLK) 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 (e.g., SLIMCLK) 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.

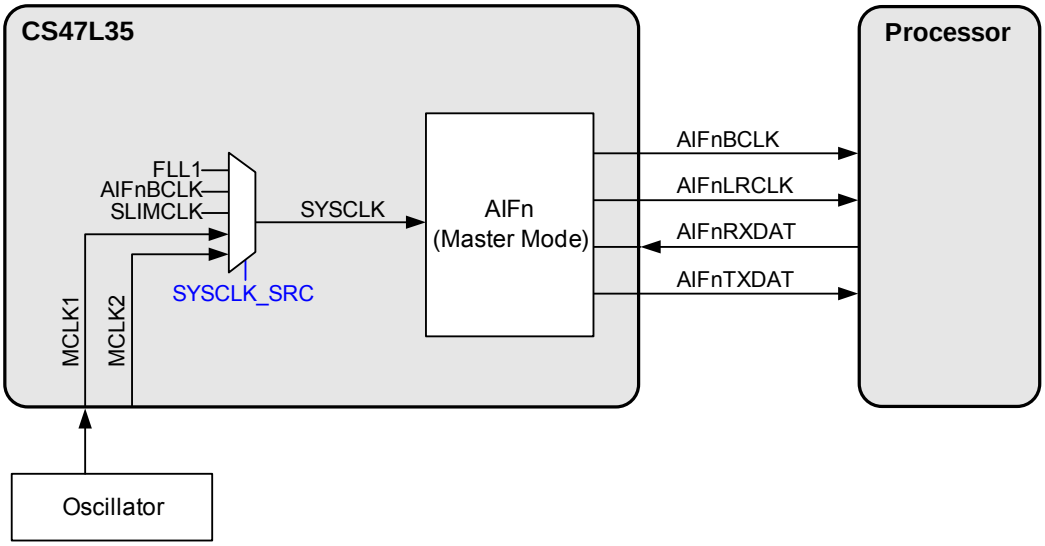


Figure 5-13. AIF Master Mode, Using MCLK as Reference

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.

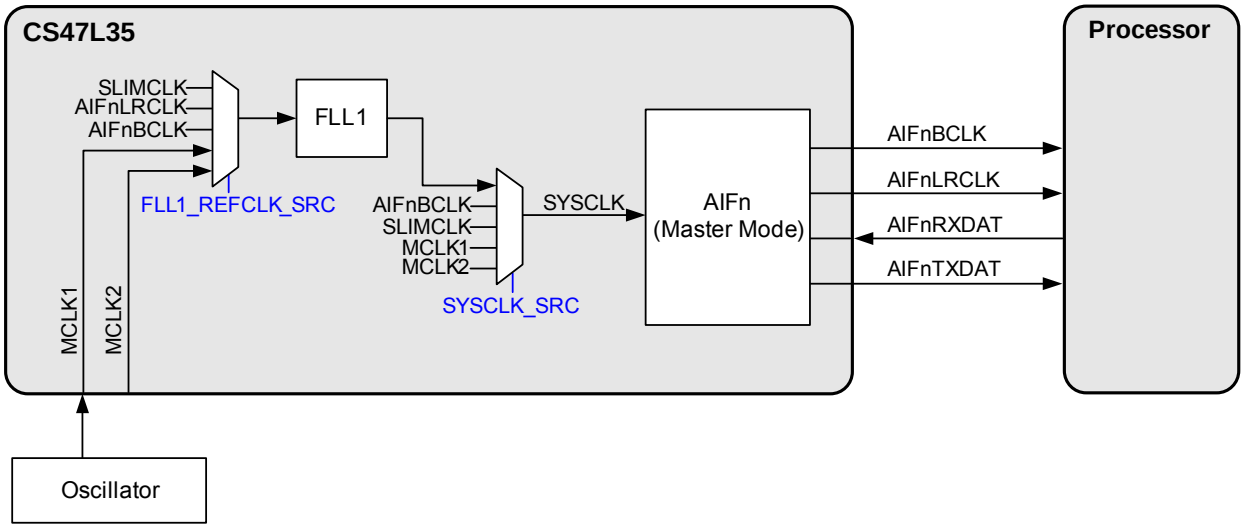


Figure 5-14. AIF Master Mode, Using MCLK and FLL as Reference

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 SLIMCLK as the reference.

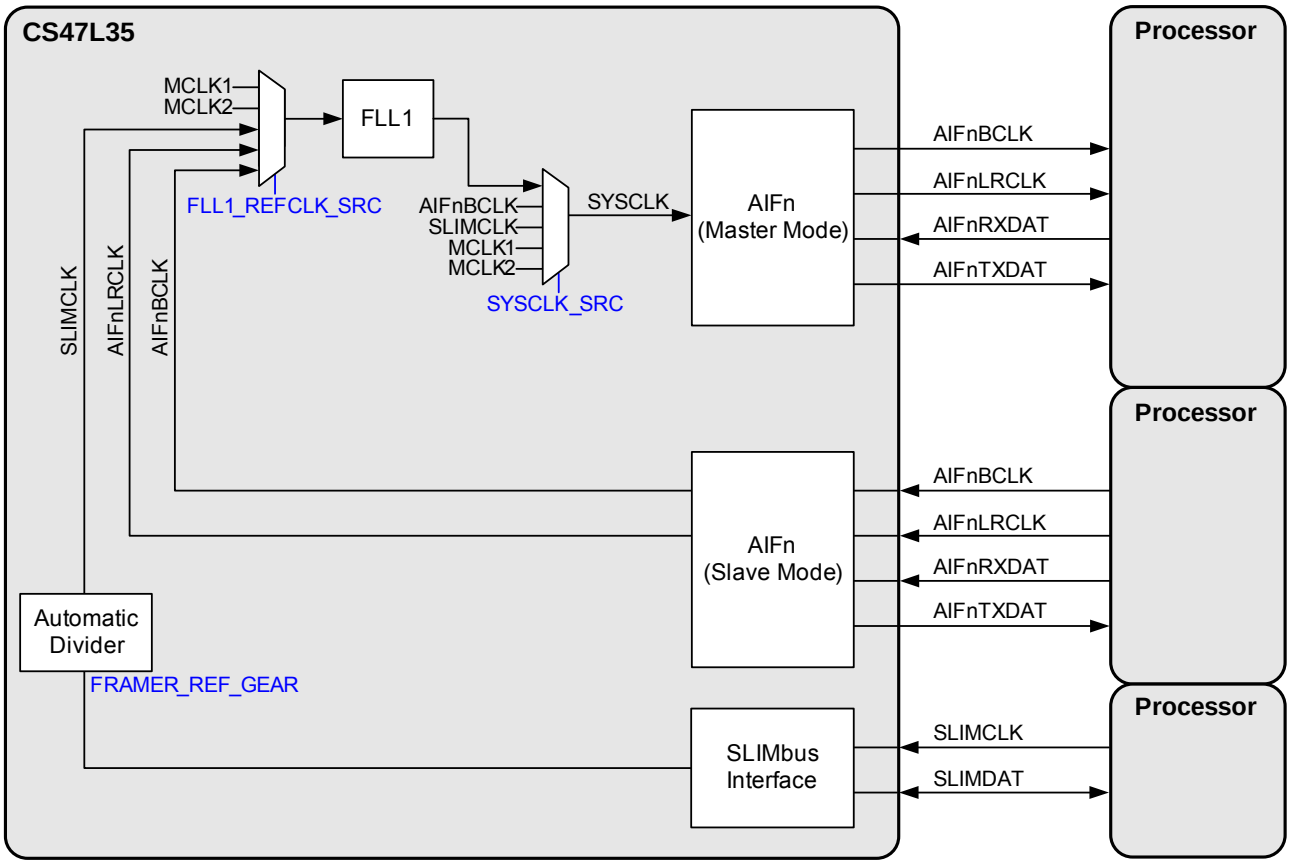


Figure 5-15. AIF Master Mode, Using Another Interface as Reference

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.

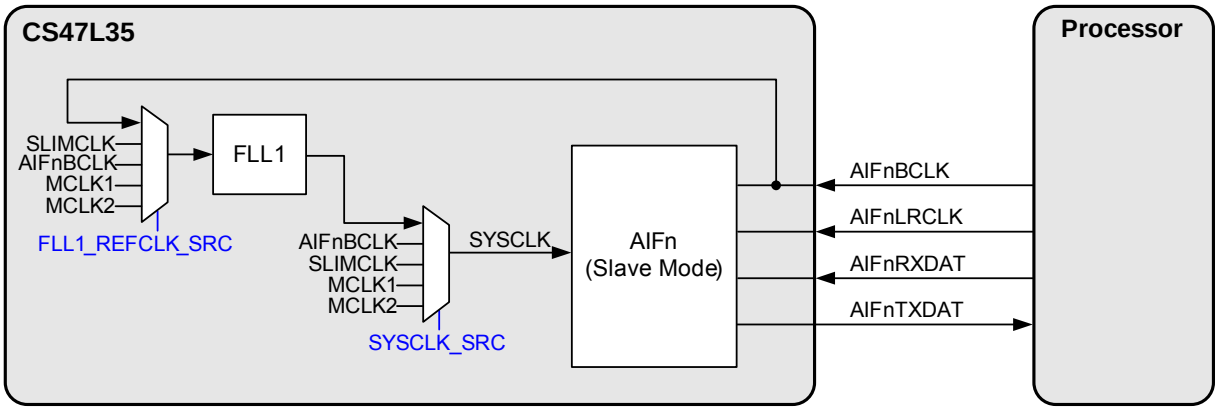


Figure 5-16. AIF Slave Mode, Using BCLK and FLL as Reference

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.

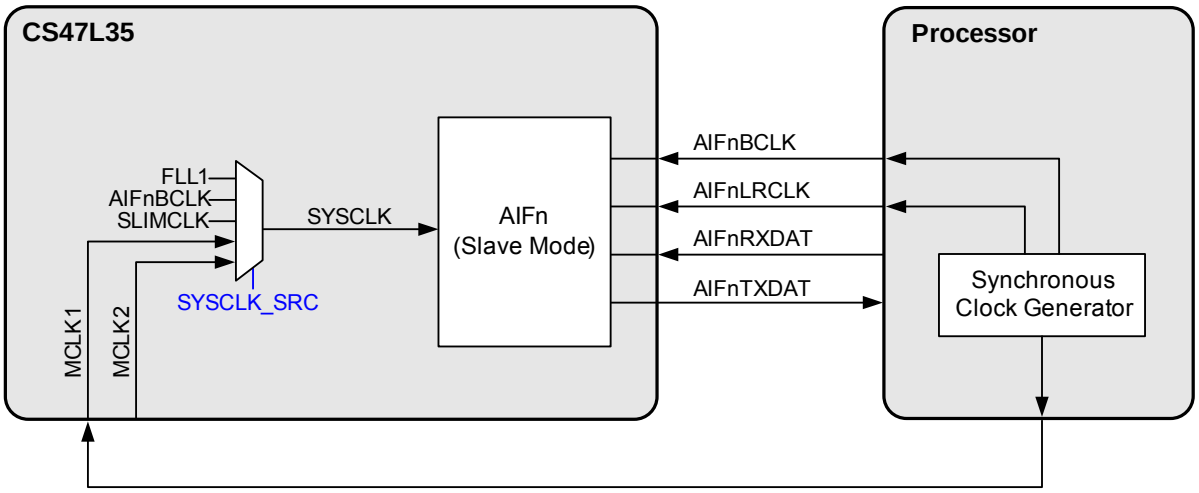


Figure 5-17. AIF Slave Mode, Using MCLK as Reference

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.

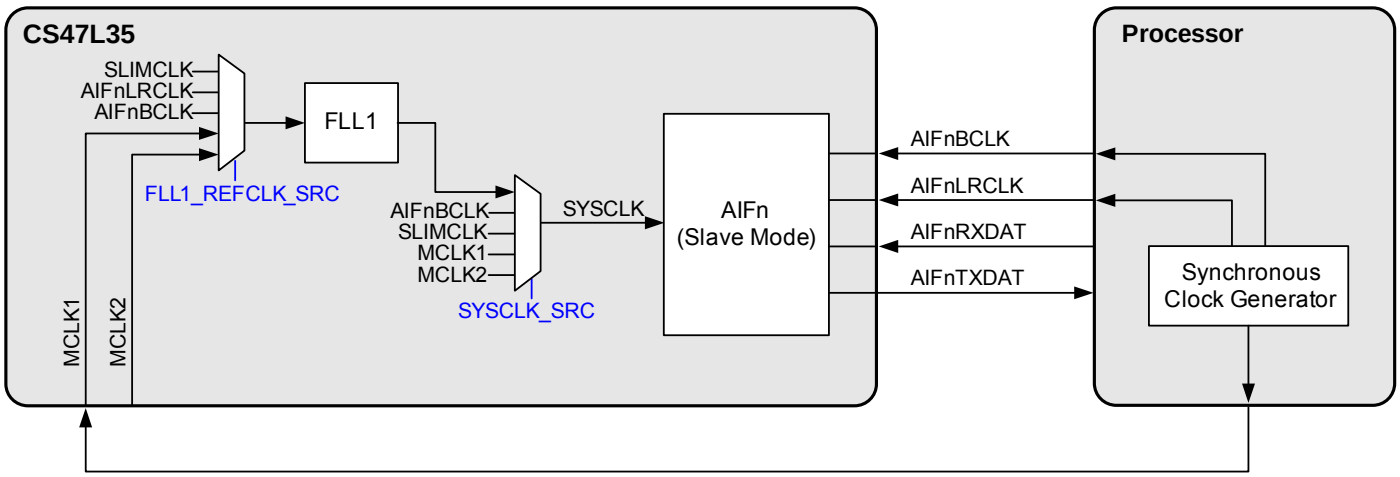


Figure 5-18. AIF Slave Mode, Using MCLK and FLL as Reference

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 SLIMCLK as the reference. For correct operation, the SLIMCLK input must be fully synchronized to the other audio interfaces.

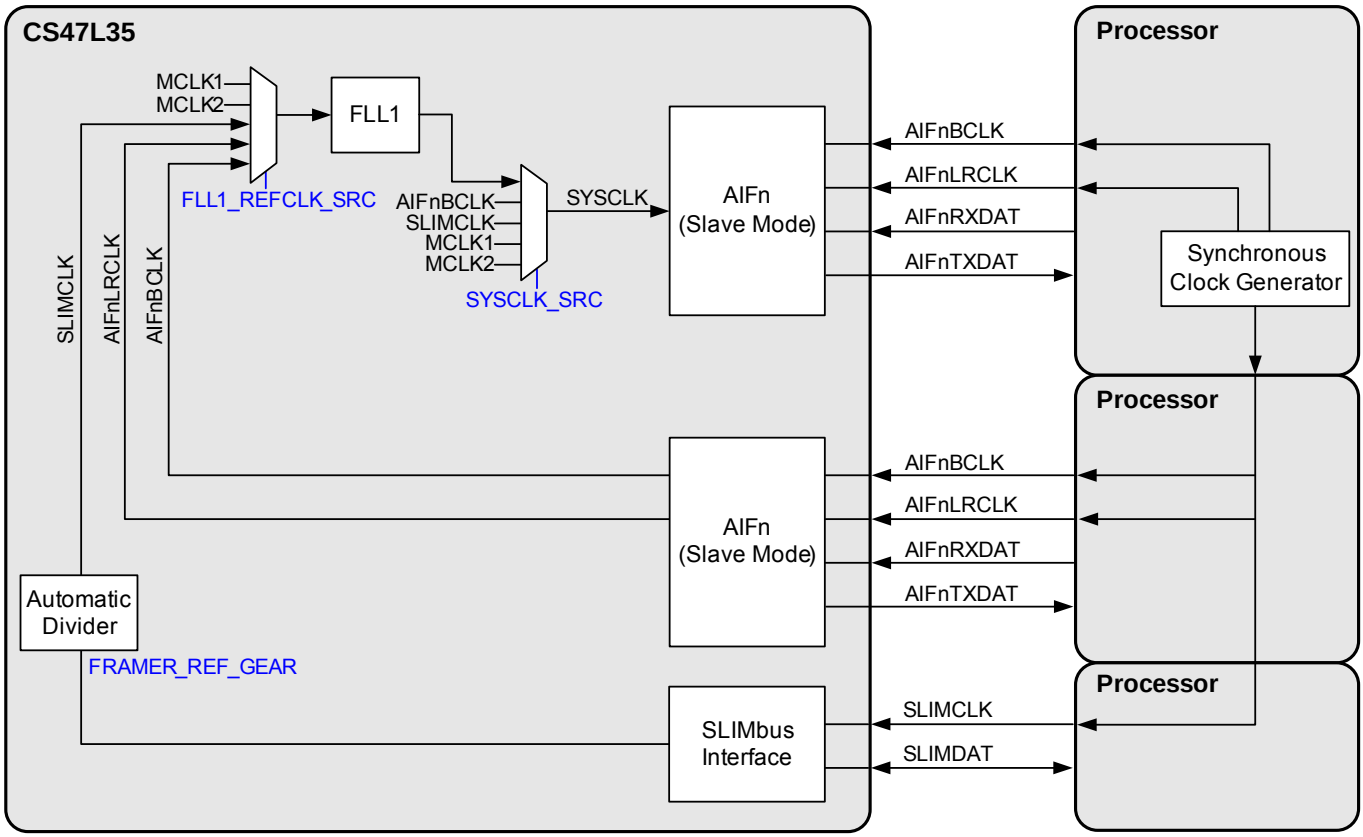


Figure 5-19. AIF Slave Mode, Using Another Interface as Reference

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 CS47L35 device as possible, with current loop areas kept as small as possible.

6 Register Map

The CS47L35 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 CS47L35 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]																6360h
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	0	0	0	0	1	CIF1MISO_DRV_STR	CIF1MISO_PD	0	0	0	1	0	0	0	0308h
R9 (9h)	Ctrl_IF_CFG_2	0	0	0	0	0	0	CIF2SDA_DRV_STR	0	0	0	0	0	0	0	0	0	0200h
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
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

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
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
R122 (7Ah)	Eventlog_Sequence_ Select_3	0	0	0	0	0	0	0	WSEQ_EVENTLOG3_INDEX [8:0]									01FFh
R123 (7Bh)	Eventlog_Sequence_ Select_4	0	0	0	0	0	0	0	WSEQ_EVENTLOG4_INDEX [8:0]									01FFh
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
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	DSP_CLK_FREQ_RANGE [2:0]			0	DSP_CLK_ENA	0	0	DSP_CLK_SRC [3:0]				0305h
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	0	0	0	MCLK2_PD	MCLK1_PD	0	0	0	0	0	0	0	0000h
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
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

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
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	FLL1_PHASE_GAIN[3:0]				FLL1_PHASE_ENA	FLL1_CTRL_RATE[2:0]			0	0	0	0	0	1	1	0	2906h
R383 (17Fh)	FLL1_Synchroniser_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLL1_SYNC_ENA	0000h
R384 (180h)	FLL1_Synchroniser_2	0	0	0	0	0	0	FLL1_SYNC_N [9:0]										0000h
R385 (181h)	FLL1_Synchroniser_3	FLL1_SYNC_THETA [15:0]																0000h
R386 (182h)	FLL1_Synchroniser_4	FLL1_SYNC_LAMBDA [15:0]																0000h
R387 (183h)	FLL1_Synchroniser_5	0	0	0	0	0	FLL1_SYNC_FRATIO [2:0]			0	0	0	0	0	0	0	0	0000h
R388 (184h)	FLL1_Synchroniser_6	0	0	0	0	0	0	0	0	FLL1_SYNCCLK_DIV [1:0]		0	0	FLL1_SYNCCLK_SRC [3:0]				0000h
R389 (185h)	FLL1_Synchroniser_7	0	0	0	0	0	0	0	0	0	0	FLL1_SYNC_GAIN [3:0]				0	FLL1_SYNC_DFSAT	0001h
R391 (187h)	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
R392 (188h)	FLL1_GPIO_Clock	0	0	0	0	0	0	0	0	FLL1_GPCLK_DIV [6:0]							FLL1_GPCLK_ENA	000Ch
R512 (200h)	Mic_Charge_Pump_1	0	0	0	0	0	0	0	0	0	0	0	0	0	CP2_DISCH	CP2_BYPASS	CP2_ENA	0006h
R531 (213h)	LDO2_Control_1	0	0	0	0	0	LDO2_VSEL [5:0]						0	0	LDO2_DISCH	0	0	03E4h
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
R537 (219h)	Mic_Bias_Ctrl_2	MICB2_EXT_CAP	0	0	0	0	0	0	MICB2_LVL [3:0]				0	MICB2_RATE	MICB2_DISCH	MICB2_BYPASS	MICB2_ENA	00E6h
R540 (21Ch)	Mic_Bias_Ctrl_5	0	0	0	0	0	0	0	0	0	0	MICB1B_DISCH	MICB1B_ENA	0	0	MICB1A_DISCH	MICB1A_ENA	0022h
R542 (21Eh)	Mic_Bias_Ctrl_6	0	0	0	0	0	0	0	0	0	0	MICB2B_DISCH	MICB2B_ENA	0	0	MICB2A_DISCH	MICB2A_ENA	0022h
R620 (26Ch)	SPK_Watchdog_1	0	0	0	0	0	0	0	0	0	0	0	0	SPK_SHUTDOWN_TIMER_SEL [3:0]				0000h
R659 (293h)	Accessory_Detect_Mode_1	0	0	ACCDET_SRC	0	0	0	0	0	1	0	0	0	0	ACCDET_MODE [2:0]			0080h
R667 (29Bh)	Headphone_Detect_1	0	0	0	0	0	HP_IMPEDANCE_RANGE [1:0]		0	0	0	0	HP_CLK_DIV [1:0]		HP_RATE [1:0]		HP_POLL *(M)*	0000h
R668 (29Ch)	Headphone_Detect_2	HP_DONE	HP_LVL [14:0]															0000h
R669 (29Dh)	Headphone_Detect_3	0	0	0	0	0	0	HP_DACVAL [9:0]										0000h
R671 (29Fh)	Headphone_Detect_5	0	0	0	0	0	0	HP_DACVAL_DOWN [9:0]										0000h
R675 (2A3h)	Mic_Detect_1	MICD_BIAS_STARTTIME [3:0]				MICD_RATE [3:0]				0	MICD_BIAS_SRC [2:0]			0	0	MICD_DBTIME	MICD_ENA	1102h
R676 (2A4h)	Mic_Detect_2	0	0	0	0	0	0	0	0	MICD_LVL_SEL [7:0]								009Fh
R677 (2A5h)	Mic_Detect_3	0	0	0	0	0	MICD_LVL [8:0]									MICD_VALID	MICD_STS	0000h
R683 (2ABh)	Mic_Detect_4	MICDET_ADCVAL_DIFF [7:0]								0	MICDET_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

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
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	0	0	0	0	0	IN1_OSR [2:0]			0	0	IN1L_DMIC_DLY [5:0]						0500h
R788 (314h)	IN1R_Control	IN1R_HPF	0	0	0	0	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	0	0	0	0	0	0	0	0	0	0	IN1R_DMIC_DLY [5:0]						0000h
R792 (318h)	IN2L_Control	IN2L_HPF	0	0	IN2_DMIC_SUP [1:0]		IN2_MODE	0	0	IN2L_PGA_VOL [6:0]							0	0080h
R793 (319h)	ADC_Digital_Volume_2L	0	IN2L_SRC [1:0]		0	0	0	IN_VU	IN2L_MUTE	IN2L_VOL [7:0]								0180h
R794 (31Ah)	DMIC2L_Control	0	0	0	0	0	IN2_OSR [2:0]			0	0	IN2L_DMIC_DLY [5:0]						0500h
R796 (31Ch)	IN2R_Control	IN2R_HPF	0	0	0	0	0	0	0	IN2R_PGA_VOL [6:0]							0	0080h
R797 (31Dh)	ADC_Digital_Volume_2R	0	IN2R_SRC [1:0]		0	0	0	IN_VU	IN2R_MUTE	IN2R_VOL [7:0]								0180h
R798 (31Eh)	DMIC2R_Control	0	0	0	0	0	0	0	0	0	0	IN2R_DMIC_DLY [5:0]						0000h
R840 (348h)	Dig_Mic_Pad_Ctrl	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DMICDAT2_PD	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
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
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

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
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	0	AIF1_LRCLK_ADV	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
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	0	AIF2_LRCLK_ADV	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
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
R1369 (559h)	AIF2_Tx_Enables	0	0	0	0	0	0	0	0	0	0	0	0	0	0	AIF2TX2_ENA	AIF2TX1_ENA	0000h
R1370 (55Ah)	AIF2_Rx_Enables	0	0	0	0	0	0	0	0	0	0	0	0	0	0	AIF2RX2_ENA	AIF2RX1_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
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	0	AIF3_LRCLK_ADV	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
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
R1490 (5D2h)	SLIMbus_RX_Ports0	0	0	SLIMRX2_PORT_ADDR [5:0]						0	0	SLIMRX1_PORT_ADDR [5:0]						0100h
R1491 (5D3h)	SLIMbus_RX_Ports1	0	0	SLIMRX4_PORT_ADDR [5:0]						0	0	SLIMRX3_PORT_ADDR [5:0]						0302h
R1492 (5D4h)	SLIMbus_RX_Ports2	0	0	SLIMRX6_PORT_ADDR [5:0]						0	0	SLIMRX5_PORT_ADDR [5:0]						0504h
R1494 (5D6h)	SLIMbus_TX_Ports0	0	0	SLIMTX2_PORT_ADDR [5:0]						0	0	SLIMTX1_PORT_ADDR [5:0]						0908h
R1495 (5D7h)	SLIMbus_TX_Ports1	0	0	SLIMTX4_PORT_ADDR [5:0]						0	0	SLIMTX3_PORT_ADDR [5:0]						0B0Ah
R1496 (5D8h)	SLIMbus_TX_Ports2	0	0	SLIMTX6_PORT_ADDR [5:0]						0	0	SLIMTX5_PORT_ADDR [5:0]						0D0Ch
R1507 (5E3h)	SLIMbus_Framer_Ref_Gear	0	0	0	0	0	0	0	0	0	0	0	0	SLIMCLK_REF_GEAR [3:0]				0000h
R1509 (5E5h)	SLIMbus_Rates_1	0	SLIMRX2_RATE [3:0]				0	0	0	0	SLIMRX1_RATE [3:0]				0	0	0	0000h
R1510 (5E6h)	SLIMbus_Rates_2	0	SLIMRX4_RATE [3:0]				0	0	0	0	SLIMRX3_RATE [3:0]				0	0	0	0000h
R1511 (5E7h)	SLIMbus_Rates_3	0	SLIMRX6_RATE [3:0]				0	0	0	0	SLIMRX5_RATE [3:0]				0	0	0	0000h
R1513 (5E9h)	SLIMbus_Rates_5	0	SLIMTX2_RATE [3:0]				0	0	0	0	SLIMTX1_RATE [3:0]				0	0	0	0000h
R1514 (5EAh)	SLIMbus_Rates_6	0	SLIMTX4_RATE [3:0]				0	0	0	0	SLIMTX3_RATE [3:0]				0	0	0	0000h
R1515 (5EBh)	SLIMbus_Rates_7	0	SLIMTX6_RATE [3:0]				0	0	0	0	SLIMTX5_RATE [3:0]				0	0	0	0000h
R1520 (5F0h)	Slimbus_Pad_Ctrl	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SLIMDAT_DRV_STR	SLIMCLK_DRV_STR	0003h
R1525 (5F5h)	SLIMbus_RX_Channel_Enable	0	0	0	0	0	0	0	0	0	0	SLIMRX6_ENA	SLIMRX5_ENA	SLIMRX4_ENA	SLIMRX3_ENA	SLIMRX2_ENA	SLIMRX1_ENA	0000h
R1526 (5F6h)	SLIMbus_TX_Channel_Enable	0	0	0	0	0	0	0	0	0	0	SLIMTX6_ENA	SLIMTX5_ENA	SLIMTX4_ENA	SLIMTX3_ENA	SLIMTX2_ENA	SLIMTX1_ENA	0000h
R1527 (5F7h)	SLIMbus_RX_Port_Status	0	0	0	0	0	0	0	0	0	0	SLIMRX6_PORT_STS	SLIMRX5_PORT_STS	SLIMRX4_PORT_STS	SLIMRX3_PORT_STS	SLIMRX2_PORT_STS	SLIMRX1_PORT_STS	0000h
R1528 (5F8h)	SLIMbus_TX_Port_Status	0	0	0	0	0	0	0	0	0	0	SLIMTX6_PORT_STS	SLIMTX5_PORT_STS	SLIMTX4_PORT_STS	SLIMTX3_PORT_STS	SLIMTX2_PORT_STS	SLIMTX1_PORT_STS	0000h
R1600 (640h)	PWM1MIX_Input_1_Source	PWM1MIX_STS1	0	0	0	0	0	0	0	PWM1MIX_SRC1 [7:0]								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
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	OUT1RMI X_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	OUT1RMI X_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	OUT1RMI X_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	OUT1RMI X_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
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

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
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
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

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
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
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

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
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
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
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
R1984 (7C0h)	SLIMTX1MIX_Input_1_Source	SLIMTX1MIX_STS1	0	0	0	0	0	0	0	SLIMTX1MIX_SRC1 [7:0]								0000h
R1985 (7C1h)	SLIMTX1MIX_Input_1_Volume	0	0	0	0	0	0	0	0	SLIMTX1MIX_VOL1 [6:0]							0	0080h
R1986 (7C2h)	SLIMTX1MIX_Input_2_Source	SLIMTX1MIX_STS2	0	0	0	0	0	0	0	SLIMTX1MIX_SRC2 [7:0]								0000h
R1987 (7C3h)	SLIMTX1MIX_Input_2_Volume	0	0	0	0	0	0	0	0	SLIMTX1MIX_VOL2 [6:0]							0	0080h
R1988 (7C4h)	SLIMTX1MIX_Input_3_Source	SLIMTX1MIX_STS3	0	0	0	0	0	0	0	SLIMTX1MIX_SRC3 [7:0]								0000h
R1989 (7C5h)	SLIMTX1MIX_Input_3_Volume	0	0	0	0	0	0	0	0	SLIMTX1MIX_VOL3 [6:0]							0	0080h
R1990 (7C6h)	SLIMTX1MIX_Input_4_Source	SLIMTX1MIX_STS4	0	0	0	0	0	0	0	SLIMTX1MIX_SRC4 [7:0]								0000h
R1991 (7C7h)	SLIMTX1MIX_Input_4_Volume	0	0	0	0	0	0	0	0	SLIMTX1MIX_VOL4 [6:0]							0	0080h
R1992 (7C8h)	SLIMTX2MIX_Input_1_Source	SLIMTX2MIX_STS1	0	0	0	0	0	0	0	SLIMTX2MIX_SRC1 [7:0]								0000h
R1993 (7C9h)	SLIMTX2MIX_Input_1_Volume	0	0	0	0	0	0	0	0	SLIMTX2MIX_VOL1 [6:0]							0	0080h
R1994 (7CAh)	SLIMTX2MIX_Input_2_Source	SLIMTX2MIX_STS2	0	0	0	0	0	0	0	SLIMTX2MIX_SRC2 [7:0]								0000h
R1995 (7CBh)	SLIMTX2MIX_Input_2_Volume	0	0	0	0	0	0	0	0	SLIMTX2MIX_VOL2 [6:0]							0	0080h
R1996 (7CCh)	SLIMTX2MIX_Input_3_Source	SLIMTX2MIX_STS3	0	0	0	0	0	0	0	SLIMTX2MIX_SRC3 [7:0]								0000h
R1997 (7CDh)	SLIMTX2MIX_Input_3_Volume	0	0	0	0	0	0	0	0	SLIMTX2MIX_VOL3 [6:0]							0	0080h
R1998 (7CEh)	SLIMTX2MIX_Input_4_Source	SLIMTX2MIX_STS4	0	0	0	0	0	0	0	SLIMTX2MIX_SRC4 [7:0]								0000h
R1999 (7CFh)	SLIMTX2MIX_Input_4_Volume	0	0	0	0	0	0	0	0	SLIMTX2MIX_VOL4 [6:0]							0	0080h
R2000 (7D0h)	SLIMTX3MIX_Input_1_Source	SLIMTX3MIX_STS1	0	0	0	0	0	0	0	SLIMTX3MIX_SRC1 [7:0]								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
R2001 (7D1h)	SLIMTX3MIX_Input_1_Volume	0	0	0	0	0	0	0	0	SLIMTX3MIX_VOL1 [6:0]							0	0080h
R2002 (7D2h)	SLIMTX3MIX_Input_2_Source	SLIMTX3MIX_STS2	0	0	0	0	0	0	0	SLIMTX3MIX_SRC2 [7:0]								0000h
R2003 (7D3h)	SLIMTX3MIX_Input_2_Volume	0	0	0	0	0	0	0	0	SLIMTX3MIX_VOL2 [6:0]							0	0080h
R2004 (7D4h)	SLIMTX3MIX_Input_3_Source	SLIMTX3MIX_STS3	0	0	0	0	0	0	0	SLIMTX3MIX_SRC3 [7:0]								0000h
R2005 (7D5h)	SLIMTX3MIX_Input_3_Volume	0	0	0	0	0	0	0	0	SLIMTX3MIX_VOL3 [6:0]							0	0080h
R2006 (7D6h)	SLIMTX3MIX_Input_4_Source	SLIMTX3MIX_STS4	0	0	0	0	0	0	0	SLIMTX3MIX_SRC4 [7:0]								0000h
R2007 (7D7h)	SLIMTX3MIX_Input_4_Volume	0	0	0	0	0	0	0	0	SLIMTX3MIX_VOL4 [6:0]							0	0080h
R2008 (7D8h)	SLIMTX4MIX_Input_1_Source	SLIMTX4MIX_STS1	0	0	0	0	0	0	0	SLIMTX4MIX_SRC1 [7:0]								0000h
R2009 (7D9h)	SLIMTX4MIX_Input_1_Volume	0	0	0	0	0	0	0	0	SLIMTX4MIX_VOL1 [6:0]							0	0080h
R2010 (7DAh)	SLIMTX4MIX_Input_2_Source	SLIMTX4MIX_STS2	0	0	0	0	0	0	0	SLIMTX4MIX_SRC2 [7:0]								0000h
R2011 (7DBh)	SLIMTX4MIX_Input_2_Volume	0	0	0	0	0	0	0	0	SLIMTX4MIX_VOL2 [6:0]							0	0080h
R2012 (7DCh)	SLIMTX4MIX_Input_3_Source	SLIMTX4MIX_STS3	0	0	0	0	0	0	0	SLIMTX4MIX_SRC3 [7:0]								0000h
R2013 (7DDh)	SLIMTX4MIX_Input_3_Volume	0	0	0	0	0	0	0	0	SLIMTX4MIX_VOL3 [6:0]							0	0080h
R2014 (7DEh)	SLIMTX4MIX_Input_4_Source	SLIMTX4MIX_STS4	0	0	0	0	0	0	0	SLIMTX4MIX_SRC4 [7:0]								0000h
R2015 (7DFh)	SLIMTX4MIX_Input_4_Volume	0	0	0	0	0	0	0	0	SLIMTX4MIX_VOL4 [6:0]							0	0080h
R2016 (7E0h)	SLIMTX5MIX_Input_1_Source	SLIMTX5MIX_STS1	0	0	0	0	0	0	0	SLIMTX5MIX_SRC1 [7:0]								0000h
R2017 (7E1h)	SLIMTX5MIX_Input_1_Volume	0	0	0	0	0	0	0	0	SLIMTX5MIX_VOL1 [6:0]							0	0080h
R2018 (7E2h)	SLIMTX5MIX_Input_2_Source	SLIMTX5MIX_STS2	0	0	0	0	0	0	0	SLIMTX5MIX_SRC2 [7:0]								0000h
R2019 (7E3h)	SLIMTX5MIX_Input_2_Volume	0	0	0	0	0	0	0	0	SLIMTX5MIX_VOL2 [6:0]							0	0080h
R2020 (7E4h)	SLIMTX5MIX_Input_3_Source	SLIMTX5MIX_STS3	0	0	0	0	0	0	0	SLIMTX5MIX_SRC3 [7:0]								0000h
R2021 (7E5h)	SLIMTX5MIX_Input_3_Volume	0	0	0	0	0	0	0	0	SLIMTX5MIX_VOL3 [6:0]							0	0080h
R2022 (7E6h)	SLIMTX5MIX_Input_4_Source	SLIMTX5MIX_STS4	0	0	0	0	0	0	0	SLIMTX5MIX_SRC4 [7:0]								0000h
R2023 (7E7h)	SLIMTX5MIX_Input_4_Volume	0	0	0	0	0	0	0	0	SLIMTX5MIX_VOL4 [6:0]							0	0080h
R2024 (7E8h)	SLIMTX6MIX_Input_1_Source	SLIMTX6MIX_STS1	0	0	0	0	0	0	0	SLIMTX6MIX_SRC1 [7:0]								0000h
R2025 (7E9h)	SLIMTX6MIX_Input_1_Volume	0	0	0	0	0	0	0	0	SLIMTX6MIX_VOL1 [6:0]							0	0080h
R2026 (7EAh)	SLIMTX6MIX_Input_2_Source	SLIMTX6MIX_STS2	0	0	0	0	0	0	0	SLIMTX6MIX_SRC2 [7:0]								0000h
R2027 (7EBh)	SLIMTX6MIX_Input_2_Volume	0	0	0	0	0	0	0	0	SLIMTX6MIX_VOL2 [6:0]							0	0080h
R2028 (7ECh)	SLIMTX6MIX_Input_3_Source	SLIMTX6MIX_STS3	0	0	0	0	0	0	0	SLIMTX6MIX_SRC3 [7:0]								0000h
R2029 (7EDh)	SLIMTX6MIX_Input_3_Volume	0	0	0	0	0	0	0	0	SLIMTX6MIX_VOL3 [6:0]							0	0080h
R2030 (7EEh)	SLIMTX6MIX_Input_4_Source	SLIMTX6MIX_STS4	0	0	0	0	0	0	0	SLIMTX6MIX_SRC4 [7:0]								0000h
R2031 (7EFh)	SLIMTX6MIX_Input_4_Volume	0	0	0	0	0	0	0	0	SLIMTX6MIX_VOL4 [6:0]							0	0080h
R2048 (800h)	SPDIF1TX1MIX_Input_1_Source	SPDIF1TX1MIX_STS	0	0	0	0	0	0	0	SPDIF1TX1MIX_SRC [7:0]								0000h
R2049 (801h)	SPDIF1TX1MIX_Input_1_Volume	0	0	0	0	0	0	0	0	SPDIF1TX1MIX_VOL [6:0]							0	0080h
R2056 (808h)	SPDIF1TX2MIX_Input_1_Source	SPDIF1TX2MIX_STS	0	0	0	0	0	0	0	SPDIF1TX2MIX_SRC [7:0]								0000h
R2057 (809h)	SPDIF1TX2MIX_Input_1_Volume	0	0	0	0	0	0	0	0	SPDIF1TX2MIX_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

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
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
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	DRC1RMI X_STS1	0	0	0	0	0	0	0	DRC1RMIX_SRC1 [7:0]								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
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
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

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
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
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	DSP1AUX1MIX_STS	0	0	0	0	0	0	0	DSP1AUX1_SRC [7:0]								0000h
R2392 (958h)	DSP1AUX2MIX_Input_1_Source	DSP1AUX2MIX_STS	0	0	0	0	0	0	0	DSP1AUX2_SRC [7:0]								0000h
R2400 (960h)	DSP1AUX3MIX_Input_1_Source	DSP1AUX3MIX_STS	0	0	0	0	0	0	0	DSP1AUX3_SRC [7:0]								0000h
R2408 (968h)	DSP1AUX4MIX_Input_1_Source	DSP1AUX4MIX_STS	0	0	0	0	0	0	0	DSP1AUX4_SRC [7:0]								0000h
R2416 (970h)	DSP1AUX5MIX_Input_1_Source	DSP1AUX5MIX_STS	0	0	0	0	0	0	0	DSP1AUX5_SRC [7:0]								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
R2424 (978h)	DSP1AUX6MIX_Input_1_Source	DSP1AUX6MIX_STS	0	0	0	0	0	0	0	DSP1AUX6_SRC [7:0]								0000h
R2432 (980h)	DSP2LMIX_Input_1_Source	DSP2LMIX_STS1	0	0	0	0	0	0	0	DSP2LMIX_SRC1 [7:0]								0000h
R2433 (981h)	DSP2LMIX_Input_1_Volume	0	0	0	0	0	0	0	0	DSP2LMIX_VOL1 [6:0]							0	0080h
R2434 (982h)	DSP2LMIX_Input_2_Source	DSP2LMIX_STS2	0	0	0	0	0	0	0	DSP2LMIX_SRC2 [7:0]								0000h
R2435 (983h)	DSP2LMIX_Input_2_Volume	0	0	0	0	0	0	0	0	DSP2LMIX_VOL2 [6:0]							0	0080h
R2436 (984h)	DSP2LMIX_Input_3_Source	DSP2LMIX_STS3	0	0	0	0	0	0	0	DSP2LMIX_SRC3 [7:0]								0000h
R2437 (985h)	DSP2LMIX_Input_3_Volume	0	0	0	0	0	0	0	0	DSP2LMIX_VOL3 [6:0]							0	0080h
R2438 (986h)	DSP2LMIX_Input_4_Source	DSP2LMIX_STS4	0	0	0	0	0	0	0	DSP2LMIX_SRC4 [7:0]								0000h
R2439 (987h)	DSP2LMIX_Input_4_Volume	0	0	0	0	0	0	0	0	DSP2LMIX_VOL4 [6:0]							0	0080h
R2440 (988h)	DSP2RMIX_Input_1_Source	DSP2RMIX_STS1	0	0	0	0	0	0	0	DSP2RMIX_SRC1 [7:0]								0000h
R2441 (989h)	DSP2RMIX_Input_1_Volume	0	0	0	0	0	0	0	0	DSP2RMIX_VOL1 [6:0]							0	0080h
R2442 (98Ah)	DSP2RMIX_Input_2_Source	DSP2RMIX_STS2	0	0	0	0	0	0	0	DSP2RMIX_SRC2 [7:0]								0000h
R2443 (98Bh)	DSP2RMIX_Input_2_Volume	0	0	0	0	0	0	0	0	DSP2RMIX_VOL2 [6:0]							0	0080h
R2444 (98Ch)	DSP2RMIX_Input_3_Source	DSP2RMIX_STS3	0	0	0	0	0	0	0	DSP2RMIX_SRC3 [7:0]								0000h
R2445 (98Dh)	DSP2RMIX_Input_3_Volume	0	0	0	0	0	0	0	0	DSP2RMIX_VOL3 [6:0]							0	0080h
R2446 (98Eh)	DSP2RMIX_Input_4_Source	DSP2RMIX_STS4	0	0	0	0	0	0	0	DSP2RMIX_SRC4 [7:0]								0000h
R2447 (98Fh)	DSP2RMIX_Input_4_Volume	0	0	0	0	0	0	0	0	DSP2RMIX_VOL4 [6:0]							0	0080h
R2448 (990h)	DSP2AUX1MIX_Input_1_Source	DSP2AUX1MIX_STS	0	0	0	0	0	0	0	DSP2AUX1_SRC [7:0]								0000h
R2456 (998h)	DSP2AUX2MIX_Input_1_Source	DSP2AUX2MIX_STS	0	0	0	0	0	0	0	DSP2AUX2_SRC [7:0]								0000h
R2464 (9A0h)	DSP2AUX3MIX_Input_1_Source	DSP2AUX3MIX_STS	0	0	0	0	0	0	0	DSP2AUX3_SRC [7:0]								0000h
R2472 (9A8h)	DSP2AUX4MIX_Input_1_Source	DSP2AUX4MIX_STS	0	0	0	0	0	0	0	DSP2AUX4_SRC [7:0]								0000h
R2480 (9B0h)	DSP2AUX5MIX_Input_1_Source	DSP2AUX5MIX_STS	0	0	0	0	0	0	0	DSP2AUX5_SRC [7:0]								0000h
R2488 (9B8h)	DSP2AUX6MIX_Input_1_Source	DSP2AUX6MIX_STS	0	0	0	0	0	0	0	DSP2AUX6_SRC [7:0]								0000h
R2496 (9C0h)	DSP3LMIX_Input_1_Source	DSP3LMIX_STS1	0	0	0	0	0	0	0	DSP3LMIX_SRC1 [7:0]								0000h
R2497 (9C1h)	DSP3LMIX_Input_1_Volume	0	0	0	0	0	0	0	0	DSP3LMIX_VOL1 [6:0]							0	0080h
R2498 (9C2h)	DSP3LMIX_Input_2_Source	DSP3LMIX_STS2	0	0	0	0	0	0	0	DSP3LMIX_SRC2 [7:0]								0000h
R2499 (9C3h)	DSP3LMIX_Input_2_Volume	0	0	0	0	0	0	0	0	DSP3LMIX_VOL2 [6:0]							0	0080h
R2500 (9C4h)	DSP3LMIX_Input_3_Source	DSP3LMIX_STS3	0	0	0	0	0	0	0	DSP3LMIX_SRC3 [7:0]								0000h
R2501 (9C5h)	DSP3LMIX_Input_3_Volume	0	0	0	0	0	0	0	0	DSP3LMIX_VOL3 [6:0]							0	0080h
R2502 (9C6h)	DSP3LMIX_Input_4_Source	DSP3LMIX_STS4	0	0	0	0	0	0	0	DSP3LMIX_SRC4 [7:0]								0000h
R2503 (9C7h)	DSP3LMIX_Input_4_Volume	0	0	0	0	0	0	0	0	DSP3LMIX_VOL4 [6:0]							0	0080h
R2504 (9C8h)	DSP3RMIX_Input_1_Source	DSP3RMIX_STS1	0	0	0	0	0	0	0	DSP3RMIX_SRC1 [7:0]								0000h
R2505 (9C9h)	DSP3RMIX_Input_1_Volume	0	0	0	0	0	0	0	0	DSP3RMIX_VOL1 [6:0]							0	0080h
R2506 (9CAh)	DSP3RMIX_Input_2_Source	DSP3RMIX_STS2	0	0	0	0	0	0	0	DSP3RMIX_SRC2 [7:0]								0000h
R2507 (9CBh)	DSP3RMIX_Input_2_Volume	0	0	0	0	0	0	0	0	DSP3RMIX_VOL2 [6:0]							0	0080h
R2508 (9CCh)	DSP3RMIX_Input_3_Source	DSP3RMIX_STS3	0	0	0	0	0	0	0	DSP3RMIX_SRC3 [7:0]								0000h
R2509 (9CDh)	DSP3RMIX_Input_3_Volume	0	0	0	0	0	0	0	0	DSP3RMIX_VOL3 [6:0]							0	0080h
R2510 (9CEh)	DSP3RMIX_Input_4_Source	DSP3RMIX_STS4	0	0	0	0	0	0	0	DSP3RMIX_SRC4 [7:0]								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
R2511 (9CFh)	DSP3RMIX_Input_4_Volume	0	0	0	0	0	0	0	0	DSP3RMIX_VOL4 [6:0]							0	0080h
R2512 (9D0h)	DSP3AUX1MIX_Input_1_Source	DSP3AUX1MIX_STS	0	0	0	0	0	0	0	DSP3AUX1_SRC [7:0]								0000h
R2520 (9D8h)	DSP3AUX2MIX_Input_1_Source	DSP3AUX2MIX_STS	0	0	0	0	0	0	0	DSP3AUX2_SRC [7:0]								0000h
R2528 (9E0h)	DSP3AUX3MIX_Input_1_Source	DSP3AUX3MIX_STS	0	0	0	0	0	0	0	DSP3AUX3_SRC [7:0]								0000h
R2536 (9E8h)	DSP3AUX4MIX_Input_1_Source	DSP3AUX4MIX_STS	0	0	0	0	0	0	0	DSP3AUX4_SRC [7:0]								0000h
R2544 (9F0h)	DSP3AUX5MIX_Input_1_Source	DSP3AUX5MIX_STS	0	0	0	0	0	0	0	DSP3AUX5_SRC [7:0]								0000h
R2552 (9F8h)	DSP3AUX6MIX_Input_1_Source	DSP3AUX6MIX_STS	0	0	0	0	0	0	0	DSP3AUX6_SRC [7:0]								0000h
R2816 (B00h)	ISRC1DEC1MIX_Input_1_Source	ISRC1DEC1MIX_STS	0	0	0	0	0	0	0	ISRC1DEC1_SRC [7:0]								0000h
R2824 (B08h)	ISRC1DEC2MIX_Input_1_Source	ISRC1DEC2MIX_STS	0	0	0	0	0	0	0	ISRC1DEC2_SRC [7:0]								0000h
R2832 (B10h)	ISRC1DEC3MIX_Input_1_Source	ISRC1DEC3MIX_STS	0	0	0	0	0	0	0	ISRC1DEC3_SRC [7:0]								0000h
R2840 (B18h)	ISRC1DEC4MIX_Input_1_Source	ISRC1DEC4MIX_STS	0	0	0	0	0	0	0	ISRC1DEC4_SRC [7:0]								0000h
R2848 (B20h)	ISRC1INT1MIX_Input_1_Source	ISRC1INT1MIX_STS	0	0	0	0	0	0	0	ISRC1INT1_SRC [7:0]								0000h
R2856 (B28h)	ISRC1INT2MIX_Input_1_Source	ISRC1INT2MIX_STS	0	0	0	0	0	0	0	ISRC1INT2_SRC [7:0]								0000h
R2864 (B30h)	ISRC1INT3MIX_Input_1_Source	ISRC1INT3MIX_STS	0	0	0	0	0	0	0	ISRC1INT3_SRC [7:0]								0000h
R2872 (B38h)	ISRC1INT4MIX_Input_1_Source	ISRC1INT4MIX_STS	0	0	0	0	0	0	0	ISRC1INT4_SRC [7:0]								0000h
R2880 (B40h)	ISRC2DEC1MIX_Input_1_Source	ISRC2DEC1MIX_STS	0	0	0	0	0	0	0	ISRC2DEC1_SRC [7:0]								0000h
R2888 (B48h)	ISRC2DEC2MIX_Input_1_Source	ISRC2DEC2MIX_STS	0	0	0	0	0	0	0	ISRC2DEC2_SRC [7:0]								0000h
R2896 (B50h)	ISRC2DEC3MIX_Input_1_Source	ISRC2DEC3MIX_STS	0	0	0	0	0	0	0	ISRC2DEC3_SRC [7:0]								0000h
R2904 (B58h)	ISRC2DEC4MIX_Input_1_Source	ISRC2DEC4MIX_STS	0	0	0	0	0	0	0	ISRC2DEC4_SRC [7:0]								0000h
R2912 (B60h)	ISRC2INT1MIX_Input_1_Source	ISRC2INT1MIX_STS	0	0	0	0	0	0	0	ISRC2INT1_SRC [7:0]								0000h
R2920 (B68h)	ISRC2INT2MIX_Input_1_Source	ISRC2INT2MIX_STS	0	0	0	0	0	0	0	ISRC2INT2_SRC [7:0]								0000h
R2928 (B70h)	ISRC2INT3MIX_Input_1_Source	ISRC2INT3MIX_STS	0	0	0	0	0	0	0	ISRC2INT3_SRC [7:0]								0000h
R2936 (B78h)	ISRC2INT4MIX_Input_1_Source	ISRC2INT4MIX_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
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

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
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
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

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
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
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

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
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
R4608 (1200h)	Clock_enable_overrides_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLL1_ENA1	DSP_CLK_ENA1	0000h
R4610 (1202h)	Clock_enable_overrides_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLL1_ENA2	DSP_CLK_ENA2	0000h
R4612 (1204h)	Clock_enable_overrides_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLL1_ENA3	DSP_CLK_ENA3	0000h
R4614 (1206h)	Clock_enable_overrides_4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLL1_ENA4	DSP_CLK_ENA4	0000h
R4616 (1208h)	Clock_enable_overrides_5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLL1_ENA5	DSP_CLK_ENA5	0000h
R4618 (120Ah)	Clock_enable_overrides_6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLL1_ENA6	DSP_CLK_ENA6	0000h
R4620 (120Ch)	Clock_enable_overrides_7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLL1_ENA7	DSP_CLK_ENA7	0000h
R4622 (120Eh)	Clock_enable_overrides_8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLL1_ENA8	DSP_CLK_ENA8	0000h
R4624 (1210h)	Clock_enable_overrides_9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLL1_ENA9	DSP_CLK_ENA9	0000h
R4626 (1212h)	Clock_enable_overrides_10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLL1_ENA10	DSP_CLK_ENA10	0000h
R4628 (1214h)	Clock_enable_overrides_11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLL1_ENA11	DSP_CLK_ENA11	0000h
R4630 (1216h)	Clock_enable_overrides_12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLL1_ENA12	DSP_CLK_ENA12	0000h
R4632 (1218h)	Clock_enable_overrides_13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLL1_ENA13	DSP_CLK_ENA13	0000h
R4634 (121Ah)	Clock_enable_overrides_14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLL1_ENA14	DSP_CLK_ENA14	0000h
R4636 (121Ch)	Clock_enable_overrides_15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLL1_ENA15	DSP_CLK_ENA15	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
R4638 (121Eh)	Clock_enable_overrides_16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	FLL1_ENA16	DSP_CLK_ENA16	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	0	0	0	GP1_FN [8:0]									2001h
R5889 (1701h)	GPIO1_CTRL_2	GP1_DIR	GP1_PU	GP1_PD	GP1_DRV_STR	0	0	0	0	0	0	0	0	0	0	0	0	F000h
R5890 (1702h)	GPIO2_CTRL_1	GP2_LVL	GP2_OP_CFG	GP2_DB	GP2_POL	0	0	0	GP2_FN [8:0]									2001h
R5891 (1703h)	GPIO2_CTRL_2	GP2_DIR	GP2_PU	GP2_PD	GP2_DRV_STR	0	0	0	0	0	0	0	0	0	0	0	0	F000h
R5892 (1704h)	GPIO3_CTRL_1	GP3_LVL	GP3_OP_CFG	GP3_DB	GP3_POL	0	0	0	GP3_FN [8:0]									2001h
R5893 (1705h)	GPIO3_CTRL_2	GP3_DIR	GP3_PU	GP3_PD	GP3_DRV_STR	0	0	0	0	0	0	0	0	0	0	0	0	F000h
R5894 (1706h)	GPIO4_CTRL_1	GP4_LVL	GP4_OP_CFG	GP4_DB	GP4_POL	0	0	0	GP4_FN [8:0]									2001h
R5895 (1707h)	GPIO4_CTRL_2	GP4_DIR	GP4_PU	GP4_PD	GP4_DRV_STR	0	0	0	0	0	0	0	0	0	0	0	0	F000h
R5896 (1708h)	GPIO5_CTRL_1	GP5_LVL	GP5_OP_CFG	GP5_DB	GP5_POL	0	0	0	GP5_FN [8:0]									2001h
R5897 (1709h)	GPIO5_CTRL_2	GP5_DIR	GP5_PU	GP5_PD	GP5_DRV_STR	0	0	0	0	0	0	0	0	0	0	0	0	F000h
R5898 (170Ah)	GPIO6_CTRL_1	GP6_LVL	GP6_OP_CFG	GP6_DB	GP6_POL	0	0	0	GP6_FN [8:0]									2001h
R5899 (170Bh)	GPIO6_CTRL_2	GP6_DIR	GP6_PU	GP6_PD	GP6_DRV_STR	0	0	0	0	0	0	0	0	0	0	0	0	F000h
R5900 (170Ch)	GPIO7_CTRL_1	GP7_LVL	GP7_OP_CFG	GP7_DB	GP7_POL	0	0	0	GP7_FN [8:0]									2001h
R5901 (170Dh)	GPIO7_CTRL_2	GP7_DIR	GP7_PU	GP7_PD	GP7_DRV_STR	0	0	0	0	0	0	0	0	0	0	0	0	F000h
R5902 (170Eh)	GPIO8_CTRL_1	GP8_LVL	GP8_OP_CFG	GP8_DB	GP8_POL	0	0	0	GP8_FN [8:0]									2001h
R5903 (170Fh)	GPIO8_CTRL_2	GP8_DIR	GP8_PU	GP8_PD	GP8_DRV_STR	0	0	0	0	0	0	0	0	0	0	0	0	F000h
R5904 (1710h)	GPIO9_CTRL_1	GP9_LVL	GP9_OP_CFG	GP9_DB	GP9_POL	0	0	0	GP9_FN [8:0]									2001h
R5905 (1711h)	GPIO9_CTRL_2	GP9_DIR	GP9_PU	GP9_PD	GP9_DRV_STR	0	0	0	0	0	0	0	0	0	0	0	0	F000h
R5906 (1712h)	GPIO10_CTRL_1	GP10_LVL	GP10_OP_CFG	GP10_DB	GP10_POL	0	0	0	GP10_FN [8:0]									2001h
R5907 (1713h)	GPIO10_CTRL_2	GP10_DIR	GP10_PU	GP10_PD	GP10_DRV_STR	0	0	0	0	0	0	0	0	0	0	0	0	F000h
R5908 (1714h)	GPIO11_CTRL_1	GP11_LVL	GP11_OP_CFG	GP11_DB	GP11_POL	0	0	0	GP11_FN [8:0]									2001h
R5909 (1715h)	GPIO11_CTRL_2	GP11_DIR	GP11_PU	GP11_PD	GP11_DRV_STR	0	0	0	0	0	0	0	0	0	0	0	0	F000h
R5910 (1716h)	GPIO12_CTRL_1	GP12_LVL	GP12_OP_CFG	GP12_DB	GP12_POL	0	0	0	GP12_FN [8:0]									2001h
R5911 (1717h)	GPIO12_CTRL_2	GP12_DIR	GP12_PU	GP12_PD	GP12_DRV_STR	0	0	0	0	0	0	0	0	0	0	0	0	F000h
R5912 (1718h)	GPIO13_CTRL_1	GP13_LVL	GP13_OP_CFG	GP13_DB	GP13_POL	0	0	0	GP13_FN [8:0]									2001h
R5913 (1719h)	GPIO13_CTRL_2	GP13_DIR	GP13_PU	GP13_PD	GP13_DRV_STR	0	0	0	0	0	0	0	0	0	0	0	0	F000h
R5914 (171Ah)	GPIO14_CTRL_1	GP14_LVL	GP14_OP_CFG	GP14_DB	GP14_POL	0	0	0	GP14_FN [8:0]									2001h
R5915 (171Bh)	GPIO14_CTRL_2	GP14_DIR	GP14_PU	GP14_PD	GP14_DRV_STR	0	0	0	0	0	0	0	0	0	0	0	0	F000h
R5916 (171Ch)	GPIO15_CTRL_1	GP15_LVL	GP15_OP_CFG	GP15_DB	GP15_POL	0	0	0	GP15_FN [8:0]									2001h

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
R5917 (171Dh)	GPIO15_CTRL_2	GP15_DIR	GP15_PU	GP15_PD	GP15_DRV_STR	0	0	0	0	0	0	0	0	0	0	0	0	F000h
R5918 (171Eh)	GPIO16_CTRL_1	GP16_LVL	GP16_OP_CFG	GP16_DB	GP16_POL	0	0	0	GP16_FN [8:0]									2001h
R5919 (171Fh)	GPIO16_CTRL_2	GP16_DIR	GP16_PU	GP16_PD	GP16_DRV_STR	0	0	0	0	0	0	0	0	0	0	0	0	F000h
R6144 (1800h)	IRQ1_Status_1	DSP_SHARED_WR_COLL_EINT1	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	0	0	0	0	0	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	0	MICDET_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	0	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	SPKOUTLSC_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
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
R6160 (1810h)	IRQ1_Status_17	GPIO16_EINT1	GPIO15_EINT1	GPIO14_EINT1	GPIO13_EINT1	GPIO12_EINT1	GPIO11_EINT1	GPIO10_EINT1	GPIO9_EINT1	GPIO8_EINT1	GPIO7_EINT1	GPIO6_EINT1	GPIO5_EINT1	GPIO4_EINT1	GPIO3_EINT1	GPIO2_EINT1	GPIO1_EINT1	0000h
R6164 (1814h)	IRQ1_Status_21	0	0	0	0	0	0	0	0	0	0	0	0	TIMER4_EINT1	TIMER3_EINT1	TIMER2_EINT1	TIMER1_EINT1	0000h
R6165 (1815h)	IRQ1_Status_22	0	0	0	0	0	0	0	0	0	0	0	0	EVENT4_NOT_EMPTY_EINT1	EVENT3_NOT_EMPTY_EINT1	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	EVENT4_FULL_EINT1	EVENT3_FULL_EINT1	EVENT2_FULL_EINT1	EVENT1_FULL_EINT1	0000h
R6167 (1817h)	IRQ1_Status_24	0	0	0	0	0	0	0	0	0	0	0	0	EVENT4_WMARK_EINT1	EVENT3_WMARK_EINT1	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	DSP3_DMA_EINT1	DSP2_DMA_EINT1	DSP1_DMA_EINT1	0000h
R6170 (181Ah)	IRQ1_Status_27	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP3_START1_EINT1	DSP2_START1_EINT1	DSP1_START1_EINT1	0000h
R6171 (181Bh)	IRQ1_Status_28	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP3_START2_EINT1	DSP2_START2_EINT1	DSP1_START2_EINT1	0000h
R6173 (181Dh)	IRQ1_Status_30	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP3_BUSY_EINT1	DSP2_BUSY_EINT1	DSP1_BUSY_EINT1	0000h
R6174 (181Eh)	IRQ1_Status_31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MIF1_DONE_EINT1	0000h
R6175 (181Fh)	IRQ1_Status_32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MIF1_BLOCK_EINT1	0000h
R6208 (1840h)	IRQ1_Mask_1	IM_DSP_SHARED_WR_COLL_EINT1	0	0	IM_CTRLIF_ERR_EINT1	0	0	IM_SYSCLK_FAIL_EINT1	1	IM_BOOT_DONE_EINT1	0	0	0	0	0	0	0	9300h
R6209 (1841h)	IRQ1_Mask_2	0	0	0	0	0	0	0	IM_FLL1_LOCK_EINT1	0	0	0	0	0	0	0	0	0100h
R6213 (1845h)	IRQ1_Mask_6	0	0	0	0	0	0	0	IM_MICDET_EINT1	0	0	0	0	0	0	0	IM_HPDET_EINT1	0101h
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	0	IM_DRC2_SIG_DET_EINT1	IM_DRC1_SIG_DET_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
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_SPKOUTLSC_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
R6224 (1850h)	IRQ1_Mask_17	IM_GPIO16_EINT1	IM_GPIO15_EINT1	IM_GPIO14_EINT1	IM_GPIO13_EINT1	IM_GPIO12_EINT1	IM_GPIO11_EINT1	IM_GPIO10_EINT1	IM_GPIO9_EINT1	IM_GPIO8_EINT1	IM_GPIO7_EINT1	IM_GPIO6_EINT1	IM_GPIO5_EINT1	IM_GPIO4_EINT1	IM_GPIO3_EINT1	IM_GPIO2_EINT1	IM_GPIO1_EINT1	FFFFh
R6228 (1854h)	IRQ1_Mask_21	0	0	0	0	0	0	0	0	0	0	0	0	IM_TIMER4_EINT1	IM_TIMER3_EINT1	IM_TIMER2_EINT1	IM_TIMER1_EINT1	000Fh
R6229 (1855h)	IRQ1_Mask_22	0	0	0	0	0	0	0	0	0	0	0	0	IM_EVENT4_NOT_EMPTY_EINT1	IM_EVENT3_NOT_EMPTY_EINT1	IM_EVENT2_NOT_EMPTY_EINT1	IM_EVENT1_NOT_EMPTY_EINT1	000Fh
R6230 (1856h)	IRQ1_Mask_23	0	0	0	0	0	0	0	0	0	0	0	0	IM_EVENT4_FULL_EINT1	IM_EVENT3_FULL_EINT1	IM_EVENT2_FULL_EINT1	IM_EVENT1_FULL_EINT1	000Fh
R6231 (1857h)	IRQ1_Mask_24	0	0	0	0	0	0	0	0	0	0	0	0	IM_EVENT4_WMARK_EINT1	IM_EVENT3_WMARK_EINT1	IM_EVENT2_WMARK_EINT1	IM_EVENT1_WMARK_EINT1	000Fh
R6232 (1858h)	IRQ1_Mask_25	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP3_DMA_EINT1	IM_DSP2_DMA_EINT1	IM_DSP1_DMA_EINT1	0007h
R6234 (185Ah)	IRQ1_Mask_27	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP3_START1_EINT1	IM_DSP2_START1_EINT1	IM_DSP1_START1_EINT1	0007h
R6235 (185Bh)	IRQ1_Mask_28	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP3_START2_EINT1	IM_DSP2_START2_EINT1	IM_DSP1_START2_EINT1	0007h
R6237 (185Dh)	IRQ1_Mask_30	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP3_BUSY_EINT1	IM_DSP2_BUSY_EINT1	IM_DSP1_BUSY_EINT1	0007h
R6238 (185Eh)	IRQ1_Mask_31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_MIF1_DONE_EINT1	0001h
R6239 (185Fh)	IRQ1_Mask_32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_MIF1_BLOCK_EINT1	0001h
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	0	0	0	0	0	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	0	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
R6288 (1890h)	IRQ1_Raw_Status_17	GPIO16_STS1	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	EVENT4_NOT_EMPTY_STS1	EVENT3_NOT_EMPTY_STS1	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	EVENT4_FULL_STS1	EVENT3_FULL_STS1	EVENT2_FULL_STS1	EVENT1_FULL_STS1	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
R6295 (1897h)	IRQ1_Raw_Status_24	0	0	0	0	0	0	0	0	0	0	0	0	EVENT4_WMARK_STS1	EVENT3_WMARK_STS1	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	DSP3_DMA_STS1	DSP2_DMA_STS1	DSP1_DMA_STS1	0000h
R6301 (189Dh)	IRQ1_Raw_Status_30	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP3_BUSY_STS1	DSP2_BUSY_STS1	DSP1_BUSY_STS1	0000h
R6400 (1900h)	IRQ2_Status_1	DSP_SHARED_WR_COLL_EINT2	0	0	CTRLIF_ERR_EINT2	0	0	SYSCLK_FAIL_EINT2	0	BOOT_DONE_EINT2	0	0	0	0	0	0	0	0000h
R6401 (1901h)	IRQ2_Status_2	0	0	0	0	0	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	0	MICDET_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	0	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	SPKOUTLSC_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
R6416 (1910h)	IRQ2_Status_17	GPIO16_EINT2	GPIO15_EINT2	GPIO14_EINT2	GPIO13_EINT2	GPIO12_EINT2	GPIO11_EINT2	GPIO10_EINT2	GPIO9_EINT2	GPIO8_EINT2	GPIO7_EINT2	GPIO6_EINT2	GPIO5_EINT2	GPIO4_EINT2	GPIO3_EINT2	GPIO2_EINT2	GPIO1_EINT2	0000h
R6420 (1914h)	IRQ2_Status_21	0	0	0	0	0	0	0	0	0	0	0	0	TIMER4_EINT2	TIMER3_EINT2	TIMER2_EINT2	TIMER1_EINT2	0000h
R6421 (1915h)	IRQ2_Status_22	0	0	0	0	0	0	0	0	0	0	0	0	EVENT4_NOT_EMPTY_EINT2	EVENT3_NOT_EMPTY_EINT2	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	EVENT4_FULL_EINT2	EVENT3_FULL_EINT2	EVENT2_FULL_EINT2	EVENT1_FULL_EINT2	0000h
R6423 (1917h)	IRQ2_Status_24	0	0	0	0	0	0	0	0	0	0	0	0	EVENT4_WMARK_EINT2	EVENT3_WMARK_EINT2	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	DSP3_DMA_EINT2	DSP2_DMA_EINT2	DSP1_DMA_EINT2	0000h
R6426 (191Ah)	IRQ2_Status_27	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP3_START1_EINT2	DSP2_START1_EINT2	DSP1_START1_EINT2	0000h
R6427 (191Bh)	IRQ2_Status_28	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP3_START2_EINT2	DSP2_START2_EINT2	DSP1_START2_EINT2	0000h
R6429 (191Dh)	IRQ2_Status_30	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP3_BUSY_EINT2	DSP2_BUSY_EINT2	DSP1_BUSY_EINT2	0000h
R6430 (191Eh)	IRQ2_Status_31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MIF1_DONE_EINT2	0000h
R6431 (191Fh)	IRQ2_Status_32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MIF1_BLOCK_EINT2	0000h
R6464 (1940h)	IRQ2_Mask_1	IM_DSP_SHARED_WR_COLL_EINT2	0	0	IM_CTRLIF_ERR_EINT2	0	0	IM_SYSCLK_FAIL_EINT2	1	IM_BOOT_DONE_EINT2	0	0	0	0	0	0	0	9380h
R6465 (1941h)	IRQ2_Mask_2	0	0	0	0	0	0	0	IM_FLL1_LOCK_EINT2	0	0	0	0	0	0	0	0	0100h
R6469 (1945h)	IRQ2_Mask_6	0	0	0	0	0	0	0	IM_MICDET_EINT2	0	0	0	0	0	0	0	IM_HPDET_EINT2	0101h
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

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
R6472 (1948h)	IRQ2_Mask_9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DRC2_SIG_DET_EINT2	IM_DRC1_SIG_DET_EINT2	0003h
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
R6475 (194Bh)	IRQ2_Mask_12	0	0	0	0	0	0	0	0	0	IM_SPKOUTLSC_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
R6480 (1950h)	IRQ2_Mask_17	IM_GPIO16_EINT2	IM_GPIO15_EINT2	IM_GPIO14_EINT2	IM_GPIO13_EINT2	IM_GPIO12_EINT2	IM_GPIO11_EINT2	IM_GPIO10_EINT2	IM_GPIO9_EINT2	IM_GPIO8_EINT2	IM_GPIO7_EINT2	IM_GPIO6_EINT2	IM_GPIO5_EINT2	IM_GPIO4_EINT2	IM_GPIO3_EINT2	IM_GPIO2_EINT2	IM_GPIO1_EINT2	FFFFh
R6484 (1954h)	IRQ2_Mask_21	0	0	0	0	0	0	0	0	0	0	0	0	IM_TIMER4_EINT2	IM_TIMER3_EINT2	IM_TIMER2_EINT2	IM_TIMER1_EINT2	000Fh
R6485 (1955h)	IRQ2_Mask_22	0	0	0	0	0	0	0	0	0	0	0	0	IM_EVENT4_NOT_EMPTY_EINT2	IM_EVENT3_NOT_EMPTY_EINT2	IM_EVENT2_NOT_EMPTY_EINT2	IM_EVENT1_NOT_EMPTY_EINT2	000Fh
R6486 (1956h)	IRQ2_Mask_23	0	0	0	0	0	0	0	0	0	0	0	0	IM_EVENT4_FULL_EINT2	IM_EVENT3_FULL_EINT2	IM_EVENT2_FULL_EINT2	IM_EVENT1_FULL_EINT2	000Fh
R6487 (1957h)	IRQ2_Mask_24	0	0	0	0	0	0	0	0	0	0	0	0	IM_EVENT4_WMARK_EINT2	IM_EVENT3_WMARK_EINT2	IM_EVENT2_WMARK_EINT2	IM_EVENT1_WMARK_EINT2	000Fh
R6488 (1958h)	IRQ2_Mask_25	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP3_DMA_EINT2	IM_DSP2_DMA_EINT2	IM_DSP1_DMA_EINT2	0007h
R6490 (195Ah)	IRQ2_Mask_27	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP3_START1_EINT2	IM_DSP2_START1_EINT2	IM_DSP1_START1_EINT2	0007h
R6491 (195Bh)	IRQ2_Mask_28	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP3_START2_EINT2	IM_DSP2_START2_EINT2	IM_DSP1_START2_EINT2	0007h
R6493 (195Dh)	IRQ2_Mask_30	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP3_BUSY_EINT2	IM_DSP2_BUSY_EINT2	IM_DSP1_BUSY_EINT2	0007h
R6494 (195Eh)	IRQ2_Mask_31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_MIF1_DONE_EINT2	0001h
R6495 (195Fh)	IRQ2_Mask_32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_MIF1_BLOCK_EINT2	0001h
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	0	0	0	0	0	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	0	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
R6544 (1990h)	IRQ2_Raw_Status_17	GPIO16_STS2	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	EVENT4_NOT_EMPTY_STS2	EVENT3_NOT_EMPTY_STS2	EVENT2_NOT_EMPTY_STS2	EVENT1_NOT_EMPTY_STS2	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
R6550 (1996h)	IRQ2_Raw_Status_23	0	0	0	0	0	0	0	0	0	0	0	0	EVENT4_FULL_STS2	EVENT3_FULL_STS2	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	EVENT4_WMARK_STS2	EVENT3_WMARK_STS2	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	DSP3_DMA_STS2	DSP2_DMA_STS2	DSP1_DMA_STS2	0000h
R6557 (199Dh)	IRQ2_Raw_Status_30	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP3_BUSY_STS2	DSP2_BUSY_STS2	DSP1_BUSY_STS2	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	0	0	0	0	0	0	0	0	0	0	0	0	0	RESET_PU	RESET_PD	0002h

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]													82253719h
		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]													C2300001h
		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]													02251301h
		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]													8225191Fh
		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]													82310B00h
		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]													E231023Bh
		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]													02313B01h
		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]													62300000h
		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]													E2314288h
		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]													02310B00h
		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]													02310B00h
		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]													02250E01h
		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]													42310C02h
		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]													E2310227h
		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]													02313B01h
		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]													E2314266h
		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]													E2315294h
		WSEQ_DELAY16 [3:0]			WSEQ_DATA_START16 [3:0]					WSEQ_DATA16 [7:0]								
R12322 (3022h)	WSEQ_Sequence_18	WSEQ_DATA_WIDTH17 [2:0]			WSEQ_ADDR17 [12:0]													02310B00h
		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]													02310B00h
		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]													E2251734h
		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]													0225F501h
		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]													0000F000h
		WSEQ_DELAY21 [3:0]			WSEQ_DATA_START21 [3:0]					WSEQ_DATA21 [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
R12332 (302Ch)	WSEQ_Sequence_23	WSEQ_DATA_WIDTH22 [2:0]			WSEQ_ADDR22 [12:0]													0000F000h
		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]													0000F000h
		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]													0000F000h
		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]													82263719h
		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]													C2300001h
		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]													02261301h
		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]													8226191Fh
		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]													82310B02h
		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]													E231023Bh
		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]													02313B01h
		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]													62300000h
		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]													E2314288h
		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]													02310B00h
		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]													02310B00h
		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]													02260E01h
		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]													42310C03h
		WSEQ_DELAY52 [3:0]				WSEQ_DATA_START52 [3:0]				WSEQ_DATA52 [7:0]								
R12394 (306Ah)	WSEQ_Sequence_54	WSEQ_DATA_WIDTH53 [2:0]			WSEQ_ADDR53 [12:0]													E2310227h
		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]													02313B01h
		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]													E2314266h
		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]													E2315294h
		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]													02310B00h
		WSEQ_DELAY57 [3:0]				WSEQ_DATA_START57 [3:0]				WSEQ_DATA57 [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
R12404 (3074h)	WSEQ_Sequence_59	WSEQ_DATA_WIDTH58 [2:0]			WSEQ_ADDR58 [12:0]													02310B00h
		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]													E2261734h
		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]													0226F501h
		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]													0000F000h
		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]													0000F000h
		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]													0000F000h
		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]													0000F000h
		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]													0000F000h
		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]													026D0101h
		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]													44B00004h
		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]													04020701h
		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]													04AE5801h
		WSEQ_DELAY88 [3:0]			WSEQ_DATA_START88 [3:0]					WSEQ_DATA88 [7:0]								
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]													A4AE201Fh
		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]													A4AE301Dh
		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]													A4AE203Ch
		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]													A4AE303Ch
		WSEQ_DELAY93 [3:0]			WSEQ_DATA_START93 [3:0]					WSEQ_DATA93 [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
R12476 (30BCh)	WSEQ_Sequence_95	WSEQ_DATA_WIDTH94 [2:0]			WSEQ_ADDR94 [12:0]													026D4F01h
		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]													026D0100h
		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]													04B00200h
		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]													04C7F101h
		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]													0000F000h
		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]													026D0101h
		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]													A4AE101Dh
		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]													A4AE0003h
		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]													04AE1800h
		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]													04024700h
		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]													A4AE0003h
		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]													026D0F00h
		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]													04C7F301h
		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]								
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]													00000000h
		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]													00000000h
		WSEQ_DELAY129 [3:0]				WSEQ_DATA_START129 [3:0]				WSEQ_DATA129 [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
R12548 (3104h)	WSEQ_Sequence_131	WSEQ_DATA_WIDTH130 [2:0]			WSEQ_ADDR130 [12:0]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		WSEQ_DELAY160 [3:0]				WSEQ_DATA_START160 [3:0]				WSEQ_DATA160 [7:0]								
R12610 (3142h)	WSEQ_Sequence_162	WSEQ_DATA_WIDTH161 [2:0]			WSEQ_ADDR161 [12:0]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		WSEQ_DELAY165 [3:0]				WSEQ_DATA_START165 [3:0]				WSEQ_DATA165 [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
R12620 (314Ch)	WSEQ_Sequence_167	WSEQ_DATA_WIDTH166 [2:0]			WSEQ_ADDR166 [12:0]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		WSEQ_DELAY196 [3:0]				WSEQ_DATA_START196 [3:0]				WSEQ_DATA196 [7:0]								
R12682 (318Ah)	WSEQ_Sequence_198	WSEQ_DATA_WIDTH197 [2:0]			WSEQ_ADDR197 [12:0]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		WSEQ_DELAY201 [3:0]				WSEQ_DATA_START201 [3:0]				WSEQ_DATA201 [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
R12692 (3194h)	WSEQ_Sequence_203	WSEQ_DATA_WIDTH202 [2:0]			WSEQ_ADDR202 [12:0]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]													00000000h
		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]								
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]								

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
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]								
R12792 (31F8h)	OTP_HPDET_Cal_1	HP_OFFSET_11 [7:0]								HP_OFFSET_10 [7:0]								00000000h
		HP_OFFSET_01 [7:0]								HP_OFFSET_00 [7:0]								
R12794 (31FAh)	OTP_HPDET_Cal_2	SPARE2 [7:0]								SPARE1 [7:0]								00000000h
		HP_GRADIENT_1X [7:0]								HP_GRADIENT_0X [7:0]								
R12796 (31FCh)	OTP_Security_1	SECURITY_VALUE [63:48]																00000000h
		SECURITY_VALUE [47:32]																
R12798 (31FEh)	OTP_Security_2	SECURITY_VALUE [31:16]																00000000h
		SECURITY_VALUE [15:0]																
R262146 (40002h)	MIF1_I2C_CONFIG_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	MIF1_SLV_ADDR [9:0]										MIF1_ADDR_MODE	
R262148 (40004h)	MIF1_I2C_CONFIG_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000004h
		0	0	0	0	0	MIF1_HS_MASTER_ID [2:0]			0	0	0	0	MIF1_NACK_RESPONSE	MIF1_SCL_MON_ENA	MIF1_RPT_START	MIF1_START_BYTE_ENA	
R262152 (40008h)	MIF1_I2C_CONFIG_5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000001h
		0	0	0	0	0	0	0	MIF1_WDT_VALUE_FRC_ENA	0	0	0	0	0	0	0	MIF1_WDT_ENA	
R262272 (40080h)	MIF1_I2C_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	MIF1_WDT_TIMEOUT_STS	MIF1_ARBIT_LOST_STS	MIF1_NACK_STS	
R262400 (40100h)	MIF1_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	MIF1_START	
R262404 (40104h)	MIF1_CONFIG_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MIF1_WORD_SIZE [1:0]		00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	MIF1_READ_WRITE_SEL	
R262406 (40106h)	MIF1_CONFIG_4	0	0	0	0	0	0	0	0	0	0	0	MIF1_TX_LENGTH [20:16]					00000000h
		MIF1_TX_LENGTH [15:0]																
R262416 (40110h)	MIF1_CONFIG_5	0	0	0	0	0	0	0	0	0	0	0	MIF1_RX_LENGTH [20:16]					00000000h
		MIF1_RX_LENGTH [15:0]																
R262418 (40112h)	MIF1_CONFIG_6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000010h
		0	0	0	0	0	0	0	0	MIF1_TX_BLOCK_LENGTH [7:0]								
R262420 (40114h)	MIF1_CONFIG_7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000010h
		0	0	0	0	0	0	0	0	MIF1_RX_BLOCK_LENGTH [7:0]								
R262422 (40116h)	MIF1_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	MIF1_RX_DONE	0	0	0	MIF1_TX_DONE	
R262528 (40180h)	MIF1_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	MIF1_BUSY_STS	0	0	0	MIF1_RX_REQUEST	0	0	0	MIF1_TX_REQUEST	

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
R262530 (40182h)	MIF1_STATUS_2	0	0	0	0	0	0	0	0	0	0	0	MIF1_BYTE_COUNT [20:16]					00000000h
		MIF1_BYTE_COUNT [15:0]																
R262656 (40200h)	MIF1_TX_1	MIF1_TX_BYTE4 [7:0]								MIF1_TX_BYTE3 [7:0]								00000000h
		MIF1_TX_BYTE2 [7:0]								MIF1_TX_BYTE1 [7:0]								
R262658 (40202h)	MIF1_TX_2	MIF1_TX_BYTE8 [7:0]								MIF1_TX_BYTE7 [7:0]								00000000h
		MIF1_TX_BYTE6 [7:0]								MIF1_TX_BYTE5 [7:0]								
R262660 (40204h)	MIF1_TX_3	MIF1_TX_BYTE12 [7:0]								MIF1_TX_BYTE11 [7:0]								00000000h
		MIF1_TX_BYTE10 [7:0]								MIF1_TX_BYTE9 [7:0]								
R262662 (40206h)	MIF1_TX_4	MIF1_TX_BYTE16 [7:0]								MIF1_TX_BYTE15 [7:0]								00000000h
		MIF1_TX_BYTE14 [7:0]								MIF1_TX_BYTE13 [7:0]								
R262912 (40300h)	MIF1_RX_1	MIF1_RX_BYTE4 [7:0]								MIF1_RX_BYTE3 [7:0]								00000000h
		MIF1_RX_BYTE2 [7:0]								MIF1_RX_BYTE1 [7:0]								
R262914 (40302h)	MIF1_RX_2	MIF1_RX_BYTE8 [7:0]								MIF1_RX_BYTE7 [7:0]								00000000h
		MIF1_RX_BYTE6 [7:0]								MIF1_RX_BYTE5 [7:0]								
R262916 (40304h)	MIF1_RX_3	MIF1_RX_BYTE12 [7:0]								MIF1_RX_BYTE11 [7:0]								00000000h
		MIF1_RX_BYTE10 [7:0]								MIF1_RX_BYTE9 [7:0]								
R262918 (40306h)	MIF1_RX_4	MIF1_RX_BYTE16 [7:0]								MIF1_RX_BYTE15 [7:0]								00000000h
		MIF1_RX_BYTE14 [7:0]								MIF1_RX_BYTE13 [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	0	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	0	EVENTLOG1_CH1_SEL [8:0]									
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	0	EVENTLOG1_CH2_SEL [8:0]									
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	0	EVENTLOG1_CH3_SEL [8:0]									
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	0	EVENTLOG1_CH4_SEL [8:0]									
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	0	EVENTLOG1_CH5_SEL [8:0]									
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	0	EVENTLOG1_CH6_SEL [8:0]									
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	0	EVENTLOG1_CH7_SEL [8:0]									
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	0	EVENTLOG1_CH8_SEL [8:0]									
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	0	EVENTLOG1_CH9_SEL [8:0]									
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	0	EVENTLOG1_CH10_SEL [8:0]									
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	0	EVENTLOG1_CH11_SEL [8:0]									

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
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	0	EVENTLOG1_CH12_SEL [8:0]									
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	0	EVENTLOG1_CH13_SEL [8:0]									
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	0	EVENTLOG1_CH14_SEL [8:0]									
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	0	EVENTLOG1_CH15_SEL [8:0]									
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	0	EVENTLOG1_CH16_SEL [8:0]									
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	0	EVENTLOG1_FIFO0_ID [8:0]									
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	0	EVENTLOG1_FIFO1_ID [8:0]									
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	0	EVENTLOG1_FIFO2_ID [8:0]									
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	0	EVENTLOG1_FIFO3_ID [8:0]									
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	0	EVENTLOG1_FIFO4_ID [8:0]									
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	0	EVENTLOG1_FIFO5_ID [8:0]									
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	0	EVENTLOG1_FIFO6_ID [8:0]									
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	0	EVENTLOG1_FIFO7_ID [8:0]									
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	0	EVENTLOG1_FIFO8_ID [8:0]									
R295074 (480A2h)	EVENTLOG1_FIFO8_TIME	EVENTLOG1_FIFO8_TIME [31:16]																00000000h
		EVENTLOG1_FIFO8_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
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	0	EVENTLOG1_FIFO9_ID [8:0]									
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	0	EVENTLOG1_FIFO10_ID [8:0]									
R295082 (480AAh)	EVENTLOG1_FIFO10_ TIME	EVENTLOG1_FIFO10_TIME [31:16]																00000000h
		EVENTLOG1_FIFO10_TIME [15:0]																
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	0	EVENTLOG1_FIFO11_ID [8:0]									
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	0	EVENTLOG1_FIFO12_ID [8:0]									
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	0	EVENTLOG1_FIFO13_ID [8:0]									
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	0	EVENTLOG1_FIFO14_ID [8:0]									
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	0	EVENTLOG1_FIFO15_ID [8:0]									
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	0	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	0	EVENTLOG2_CH1_SEL [8:0]									
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	0	EVENTLOG2_CH2_SEL [8:0]									
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	0	EVENTLOG2_CH3_SEL [8:0]									
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	0	EVENTLOG2_CH4_SEL [8:0]									

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
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	0	EVENTLOG2_CH5_SEL [8:0]									
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	0	EVENTLOG2_CH6_SEL [8:0]									
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	0	EVENTLOG2_CH7_SEL [8:0]									
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	0	EVENTLOG2_CH8_SEL [8:0]									
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	0	EVENTLOG2_CH9_SEL [8:0]									
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	0	EVENTLOG2_CH10_SEL [8:0]									
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	0	EVENTLOG2_CH11_SEL [8:0]									
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	0	EVENTLOG2_CH12_SEL [8:0]									
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	0	EVENTLOG2_CH13_SEL [8:0]									
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	0	EVENTLOG2_CH14_SEL [8:0]									
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	0	EVENTLOG2_CH15_SEL [8:0]									
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	0	EVENTLOG2_CH16_SEL [8:0]									
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	0	EVENTLOG2_FIFO0_ID [8:0]									
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	0	EVENTLOG2_FIFO1_ID [8:0]									
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	0	EVENTLOG2_FIFO2_ID [8:0]									
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	0	EVENTLOG2_FIFO3_ID [8:0]									
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	0	EVENTLOG2_FIFO4_ID [8:0]									
R295570 (48292h)	EVENTLOG2_FIFO4_TIME	EVENTLOG2_FIFO4_TIME [31:16]																00000000h
		EVENTLOG2_FIFO4_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
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	0	EVENTLOG2_FIFO5_ID [8:0]									
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	0	EVENTLOG2_FIFO6_ID [8:0]									
R295578 (4829Ah)	EVENTLOG2_FIFO6_TIME	EVENTLOG2_FIFO6_TIME [31:16]																00000000h
		EVENTLOG2_FIFO6_TIME [15:0]																
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	0	EVENTLOG2_FIFO7_ID [8:0]									
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	0	EVENTLOG2_FIFO8_ID [8:0]									
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	0	EVENTLOG2_FIFO9_ID [8:0]									
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	0	EVENTLOG2_FIFO10_ID [8:0]									
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	0	EVENTLOG2_FIFO11_ID [8:0]									
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	0	EVENTLOG2_FIFO12_ID [8:0]									
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	0	EVENTLOG2_FIFO13_ID [8:0]									
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	0	EVENTLOG2_FIFO14_ID [8:0]									
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	0	EVENTLOG2_FIFO15_ID [8:0]									
R295614 (482BEh)	EVENTLOG2_FIFO15_TIME	EVENTLOG2_FIFO15_TIME [31:16]																00000000h
		EVENTLOG2_FIFO15_TIME [15:0]																
R295936 (48400h)	EVENTLOG3_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	0	0	0	0	0	0	0	EVENTLOG3_RST	EVENTLOG3_ENA	
R295940 (48404h)	EVENTLOG3_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	EVENTLOG3_TIMER_SEL [1:0]		

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
R295948 (4840Ch)	EVENTLOG3_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	EVENTLOG3_FIFO_WMARK [3:0]				
R295950 (4840Eh)	EVENTLOG3_FIFO_POINTER1	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENTLOG3_FULL	EVENTLOG3_WMARK_STS	EVENTLOG3_NOT_EMPTY	00000000h
		0	0	0	0	EVENTLOG3_FIFO_WPTR [3:0]				0	0	0	0	EVENTLOG3_FIFO_RPTR [3:0]				
R295968 (48420h)	EVENTLOG3_CH_ENABLE1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG3_CH16_ENA	EVENTLOG3_CH15_ENA	EVENTLOG3_CH14_ENA	EVENTLOG3_CH13_ENA	EVENTLOG3_CH12_ENA	EVENTLOG3_CH11_ENA	EVENTLOG3_CH10_ENA	EVENTLOG3_CH9_ENA	EVENTLOG3_CH8_ENA	EVENTLOG3_CH7_ENA	EVENTLOG3_CH6_ENA	EVENTLOG3_CH5_ENA	EVENTLOG3_CH4_ENA	EVENTLOG3_CH3_ENA	EVENTLOG3_CH2_ENA	EVENTLOG3_CH1_ENA	
R296000 (48440h)	EVENTLOG3_CH1_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG3_CH1_DB	EVENTLOG3_CH1_POL	0	0	0	0	0	EVENTLOG3_CH1_SEL [8:0]									
R296002 (48442h)	EVENTLOG3_CH2_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG3_CH2_DB	EVENTLOG3_CH2_POL	0	0	0	0	0	EVENTLOG3_CH2_SEL [8:0]									
R296004 (48444h)	EVENTLOG3_CH3_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG3_CH3_DB	EVENTLOG3_CH3_POL	0	0	0	0	0	EVENTLOG3_CH3_SEL [8:0]									
R296006 (48446h)	EVENTLOG3_CH4_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG3_CH4_DB	EVENTLOG3_CH4_POL	0	0	0	0	0	EVENTLOG3_CH4_SEL [8:0]									
R296008 (48448h)	EVENTLOG3_CH5_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG3_CH5_DB	EVENTLOG3_CH5_POL	0	0	0	0	0	EVENTLOG3_CH5_SEL [8:0]									
R296010 (4844Ah)	EVENTLOG3_CH6_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG3_CH6_DB	EVENTLOG3_CH6_POL	0	0	0	0	0	EVENTLOG3_CH6_SEL [8:0]									
R296012 (4844Ch)	EVENTLOG3_CH7_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG3_CH7_DB	EVENTLOG3_CH7_POL	0	0	0	0	0	EVENTLOG3_CH7_SEL [8:0]									
R296014 (4844Eh)	EVENTLOG3_CH8_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG3_CH8_DB	EVENTLOG3_CH8_POL	0	0	0	0	0	EVENTLOG3_CH8_SEL [8:0]									
R296016 (48450h)	EVENTLOG3_CH9_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG3_CH9_DB	EVENTLOG3_CH9_POL	0	0	0	0	0	EVENTLOG3_CH9_SEL [8:0]									
R296018 (48452h)	EVENTLOG3_CH10_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG3_CH10_DB	EVENTLOG3_CH10_POL	0	0	0	0	0	EVENTLOG3_CH10_SEL [8:0]									
R296020 (48454h)	EVENTLOG3_CH11_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG3_CH11_DB	EVENTLOG3_CH11_POL	0	0	0	0	0	EVENTLOG3_CH11_SEL [8:0]									
R296022 (48456h)	EVENTLOG3_CH12_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG3_CH12_DB	EVENTLOG3_CH12_POL	0	0	0	0	0	EVENTLOG3_CH12_SEL [8:0]									
R296024 (48458h)	EVENTLOG3_CH13_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG3_CH13_DB	EVENTLOG3_CH13_POL	0	0	0	0	0	EVENTLOG3_CH13_SEL [8:0]									
R296026 (4845Ah)	EVENTLOG3_CH14_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG3_CH14_DB	EVENTLOG3_CH14_POL	0	0	0	0	0	EVENTLOG3_CH14_SEL [8:0]									
R296028 (4845Ch)	EVENTLOG3_CH15_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG3_CH15_DB	EVENTLOG3_CH15_POL	0	0	0	0	0	EVENTLOG3_CH15_SEL [8:0]									
R296030 (4845Eh)	EVENTLOG3_CH16_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG3_CH16_DB	EVENTLOG3_CH16_POL	0	0	0	0	0	EVENTLOG3_CH16_SEL [8:0]									
R296064 (48480h)	EVENTLOG3_FIFO0_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG3_FIFO0_POL	0	0	0	EVENTLOG3_FIFO0_ID [8:0]									
R296066 (48482h)	EVENTLOG3_FIFO0_TIME	EVENTLOG3_FIFO0_TIME [31:16]																00000000h
		EVENTLOG3_FIFO0_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
R296068 (48484h)	EVENTLOG3_FIFO1_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG3_FIFO1_POL	0	0	0	EVENTLOG3_FIFO1_ID [8:0]									
R296070 (48486h)	EVENTLOG3_FIFO1_TIME	EVENTLOG3_FIFO1_TIME [31:16]																00000000h
		EVENTLOG3_FIFO1_TIME [15:0]																
R296072 (48488h)	EVENTLOG3_FIFO2_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG3_FIFO2_POL	0	0	0	EVENTLOG3_FIFO2_ID [8:0]									
R296074 (4848Ah)	EVENTLOG3_FIFO2_TIME	EVENTLOG3_FIFO2_TIME [31:16]																00000000h
		EVENTLOG3_FIFO2_TIME [15:0]																
R296076 (4848Ch)	EVENTLOG3_FIFO3_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG3_FIFO3_POL	0	0	0	EVENTLOG3_FIFO3_ID [8:0]									
R296078 (4848Eh)	EVENTLOG3_FIFO3_TIME	EVENTLOG3_FIFO3_TIME [31:16]																00000000h
		EVENTLOG3_FIFO3_TIME [15:0]																
R296080 (48490h)	EVENTLOG3_FIFO4_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG3_FIFO4_POL	0	0	0	EVENTLOG3_FIFO4_ID [8:0]									
R296082 (48492h)	EVENTLOG3_FIFO4_TIME	EVENTLOG3_FIFO4_TIME [31:16]																00000000h
		EVENTLOG3_FIFO4_TIME [15:0]																
R296084 (48494h)	EVENTLOG3_FIFO5_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG3_FIFO5_POL	0	0	0	EVENTLOG3_FIFO5_ID [8:0]									
R296086 (48496h)	EVENTLOG3_FIFO5_TIME	EVENTLOG3_FIFO5_TIME [31:16]																00000000h
		EVENTLOG3_FIFO5_TIME [15:0]																
R296088 (48498h)	EVENTLOG3_FIFO6_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG3_FIFO6_POL	0	0	0	EVENTLOG3_FIFO6_ID [8:0]									
R296090 (4849Ah)	EVENTLOG3_FIFO6_TIME	EVENTLOG3_FIFO6_TIME [31:16]																00000000h
		EVENTLOG3_FIFO6_TIME [15:0]																
R296092 (4849Ch)	EVENTLOG3_FIFO7_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG3_FIFO7_POL	0	0	0	EVENTLOG3_FIFO7_ID [8:0]									
R296094 (4849Eh)	EVENTLOG3_FIFO7_TIME	EVENTLOG3_FIFO7_TIME [31:16]																00000000h
		EVENTLOG3_FIFO7_TIME [15:0]																
R296096 (484A0h)	EVENTLOG3_FIFO8_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG3_FIFO8_POL	0	0	0	EVENTLOG3_FIFO8_ID [8:0]									
R296098 (484A2h)	EVENTLOG3_FIFO8_TIME	EVENTLOG3_FIFO8_TIME [31:16]																00000000h
		EVENTLOG3_FIFO8_TIME [15:0]																
R296100 (484A4h)	EVENTLOG3_FIFO9_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG3_FIFO9_POL	0	0	0	EVENTLOG3_FIFO9_ID [8:0]									
R296102 (484A6h)	EVENTLOG3_FIFO9_TIME	EVENTLOG3_FIFO9_TIME [31:16]																00000000h
		EVENTLOG3_FIFO9_TIME [15:0]																
R296104 (484A8h)	EVENTLOG3_FIFO10_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG3_FIFO10_POL	0	0	0	EVENTLOG3_FIFO10_ID [8:0]									
R296106 (484AAh)	EVENTLOG3_FIFO10_TIME	EVENTLOG3_FIFO10_TIME [31:16]																00000000h
		EVENTLOG3_FIFO10_TIME [15:0]																
R296108 (484ACh)	EVENTLOG3_FIFO11_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG3_FIFO11_POL	0	0	0	EVENTLOG3_FIFO11_ID [8:0]									
R296110 (484AEh)	EVENTLOG3_FIFO11_TIME	EVENTLOG3_FIFO11_TIME [31:16]																00000000h
		EVENTLOG3_FIFO11_TIME [15:0]																
R296112 (484B0h)	EVENTLOG3_FIFO12_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG3_FIFO12_POL	0	0	0	EVENTLOG3_FIFO12_ID [8:0]									
R296114 (484B2h)	EVENTLOG3_FIFO12_TIME	EVENTLOG3_FIFO12_TIME [31:16]																00000000h
		EVENTLOG3_FIFO12_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
R296116 (484B4h)	EVENTLOG3_FIFO13_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG3_FIFO13_POL	0	0	0	EVENTLOG3_FIFO13_ID [8:0]									
R296118 (484B6h)	EVENTLOG3_FIFO13_TIME	EVENTLOG3_FIFO13_TIME [31:16]																00000000h
		EVENTLOG3_FIFO13_TIME [15:0]																
R296120 (484B8h)	EVENTLOG3_FIFO14_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG3_FIFO14_POL	0	0	0	EVENTLOG3_FIFO14_ID [8:0]									
R296122 (484BAh)	EVENTLOG3_FIFO14_TIME	EVENTLOG3_FIFO14_TIME [31:16]																00000000h
		EVENTLOG3_FIFO14_TIME [15:0]																
R296124 (484BCh)	EVENTLOG3_FIFO15_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG3_FIFO15_POL	0	0	0	EVENTLOG3_FIFO15_ID [8:0]									
R296126 (484BEh)	EVENTLOG3_FIFO15_TIME	EVENTLOG3_FIFO15_TIME [31:16]																00000000h
		EVENTLOG3_FIFO15_TIME [15:0]																
R296448 (48600h)	EVENTLOG4_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	0	0	0	0	0	0	0	EVENTLOG4_RST	EVENTLOG4_ENA	
R296452 (48604h)	EVENTLOG4_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	EVENTLOG4_TIMER_SEL [1:0]		
R296460 (4860Ch)	EVENTLOG4_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	EVENTLOG4_FIFO_WMARK [3:0]				
R296462 (4860Eh)	EVENTLOG4_FIFO_POINTER1	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENTLOG4_FULL	EVENTLOG4_WMARK_STS	EVENTLOG4_NOT_EMPTY	00000000h
		0	0	0	0	EVENTLOG4_FIFO_WPTR [3:0]			0	0	0	0	EVENTLOG4_FIFO_RPTR [3:0]					
R296480 (48620h)	EVENTLOG4_CH_ENABLE1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG4_CH16_ENA	EVENTLOG4_CH15_ENA	EVENTLOG4_CH14_ENA	EVENTLOG4_CH13_ENA	EVENTLOG4_CH12_ENA	EVENTLOG4_CH11_ENA	EVENTLOG4_CH10_ENA	EVENTLOG4_CH9_ENA	EVENTLOG4_CH8_ENA	EVENTLOG4_CH7_ENA	EVENTLOG4_CH6_ENA	EVENTLOG4_CH5_ENA	EVENTLOG4_CH4_ENA	EVENTLOG4_CH3_ENA	EVENTLOG4_CH2_ENA	EVENTLOG4_CH1_ENA	
R296512 (48640h)	EVENTLOG4_CH1_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG4_CH1_DB	EVENTLOG4_CH1_POL	0	0	0	0	0	EVENTLOG4_CH1_SEL [8:0]									
R296514 (48642h)	EVENTLOG4_CH2_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG4_CH2_DB	EVENTLOG4_CH2_POL	0	0	0	0	0	EVENTLOG4_CH2_SEL [8:0]									
R296516 (48644h)	EVENTLOG4_CH3_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG4_CH3_DB	EVENTLOG4_CH3_POL	0	0	0	0	0	EVENTLOG4_CH3_SEL [8:0]									
R296518 (48646h)	EVENTLOG4_CH4_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG4_CH4_DB	EVENTLOG4_CH4_POL	0	0	0	0	0	EVENTLOG4_CH4_SEL [8:0]									
R296520 (48648h)	EVENTLOG4_CH5_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG4_CH5_DB	EVENTLOG4_CH5_POL	0	0	0	0	0	EVENTLOG4_CH5_SEL [8:0]									
R296522 (4864Ah)	EVENTLOG4_CH6_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG4_CH6_DB	EVENTLOG4_CH6_POL	0	0	0	0	0	EVENTLOG4_CH6_SEL [8:0]									
R296524 (4864Ch)	EVENTLOG4_CH7_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG4_CH7_DB	EVENTLOG4_CH7_POL	0	0	0	0	0	EVENTLOG4_CH7_SEL [8:0]									
R296526 (4864Eh)	EVENTLOG4_CH8_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG4_CH8_DB	EVENTLOG4_CH8_POL	0	0	0	0	0	EVENTLOG4_CH8_SEL [8:0]									
R296528 (48650h)	EVENTLOG4_CH9_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG4_CH9_DB	EVENTLOG4_CH9_POL	0	0	0	0	0	EVENTLOG4_CH9_SEL [8:0]									
R296530 (48652h)	EVENTLOG4_CH10_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG4_CH10_DB	EVENTLOG4_CH10_POL	0	0	0	0	0	EVENTLOG4_CH10_SEL [8:0]									
R296532 (48654h)	EVENTLOG4_CH11_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG4_CH11_DB	EVENTLOG4_CH11_POL	0	0	0	0	0	EVENTLOG4_CH11_SEL [8:0]									

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
R296534 (48656h)	EVENTLOG4_CH12_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG4_CH12_DB	EVENTLOG4_CH12_POL	0	0	0	0	0	EVENTLOG4_CH12_SEL [8:0]									
R296536 (48658h)	EVENTLOG4_CH13_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG4_CH13_DB	EVENTLOG4_CH13_POL	0	0	0	0	0	EVENTLOG4_CH13_SEL [8:0]									
R296538 (4865Ah)	EVENTLOG4_CH14_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG4_CH14_DB	EVENTLOG4_CH14_POL	0	0	0	0	0	EVENTLOG4_CH14_SEL [8:0]									
R296540 (4865Ch)	EVENTLOG4_CH15_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG4_CH15_DB	EVENTLOG4_CH15_POL	0	0	0	0	0	EVENTLOG4_CH15_SEL [8:0]									
R296542 (4865Eh)	EVENTLOG4_CH16_DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG4_CH16_DB	EVENTLOG4_CH16_POL	0	0	0	0	0	EVENTLOG4_CH16_SEL [8:0]									
R296576 (48680h)	EVENTLOG4_FIFO0_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG4_FIFO0_POL	0	0	0	EVENTLOG4_FIFO0_ID [8:0]									
R296578 (48682h)	EVENTLOG4_FIFO0_TIME	EVENTLOG4_FIFO0_TIME [31:16]																00000000h
		EVENTLOG4_FIFO0_TIME [15:0]																
R296580 (48684h)	EVENTLOG4_FIFO1_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG4_FIFO1_POL	0	0	0	EVENTLOG4_FIFO1_ID [8:0]									
R296582 (48686h)	EVENTLOG4_FIFO1_TIME	EVENTLOG4_FIFO1_TIME [31:16]																00000000h
		EVENTLOG4_FIFO1_TIME [15:0]																
R296584 (48688h)	EVENTLOG4_FIFO2_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG4_FIFO2_POL	0	0	0	EVENTLOG4_FIFO2_ID [8:0]									
R296586 (4868Ah)	EVENTLOG4_FIFO2_TIME	EVENTLOG4_FIFO2_TIME [31:16]																00000000h
		EVENTLOG4_FIFO2_TIME [15:0]																
R296588 (4868Ch)	EVENTLOG4_FIFO3_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG4_FIFO3_POL	0	0	0	EVENTLOG4_FIFO3_ID [8:0]									
R296590 (4868Eh)	EVENTLOG4_FIFO3_TIME	EVENTLOG4_FIFO3_TIME [31:16]																00000000h
		EVENTLOG4_FIFO3_TIME [15:0]																
R296592 (48690h)	EVENTLOG4_FIFO4_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG4_FIFO4_POL	0	0	0	EVENTLOG4_FIFO4_ID [8:0]									
R296594 (48692h)	EVENTLOG4_FIFO4_TIME	EVENTLOG4_FIFO4_TIME [31:16]																00000000h
		EVENTLOG4_FIFO4_TIME [15:0]																
R296596 (48694h)	EVENTLOG4_FIFO5_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG4_FIFO5_POL	0	0	0	EVENTLOG4_FIFO5_ID [8:0]									
R296598 (48696h)	EVENTLOG4_FIFO5_TIME	EVENTLOG4_FIFO5_TIME [31:16]																00000000h
		EVENTLOG4_FIFO5_TIME [15:0]																
R296600 (48698h)	EVENTLOG4_FIFO6_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG4_FIFO6_POL	0	0	0	EVENTLOG4_FIFO6_ID [8:0]									
R296602 (4869Ah)	EVENTLOG4_FIFO6_TIME	EVENTLOG4_FIFO6_TIME [31:16]																00000000h
		EVENTLOG4_FIFO6_TIME [15:0]																
R296604 (4869Ch)	EVENTLOG4_FIFO7_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG4_FIFO7_POL	0	0	0	EVENTLOG4_FIFO7_ID [8:0]									
R296606 (4869Eh)	EVENTLOG4_FIFO7_TIME	EVENTLOG4_FIFO7_TIME [31:16]																00000000h
		EVENTLOG4_FIFO7_TIME [15:0]																
R296608 (486A0h)	EVENTLOG4_FIFO8_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG4_FIFO8_POL	0	0	0	EVENTLOG4_FIFO8_ID [8:0]									
R296610 (486A2h)	EVENTLOG4_FIFO8_TIME	EVENTLOG4_FIFO8_TIME [31:16]																00000000h
		EVENTLOG4_FIFO8_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
R296612 (486A4h)	EVENTLOG4_FIFO9_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG4_FIFO9_POL	0	0	0	EVENTLOG4_FIFO9_ID [8:0]									
R296614 (486A6h)	EVENTLOG4_FIFO9_TIME	EVENTLOG4_FIFO9_TIME [31:16]																00000000h
		EVENTLOG4_FIFO9_TIME [15:0]																
R296616 (486A8h)	EVENTLOG4_FIFO10_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG4_FIFO10_POL	0	0	0	EVENTLOG4_FIFO10_ID [8:0]									
R296618 (486AAh)	EVENTLOG4_FIFO10_TIME	EVENTLOG4_FIFO10_TIME [31:16]																00000000h
		EVENTLOG4_FIFO10_TIME [15:0]																
R296620 (486ACh)	EVENTLOG4_FIFO11_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG4_FIFO11_POL	0	0	0	EVENTLOG4_FIFO11_ID [8:0]									
R296622 (486AEh)	EVENTLOG4_FIFO11_TIME	EVENTLOG4_FIFO11_TIME [31:16]																00000000h
		EVENTLOG4_FIFO11_TIME [15:0]																
R296624 (486B0h)	EVENTLOG4_FIFO12_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG4_FIFO12_POL	0	0	0	EVENTLOG4_FIFO12_ID [8:0]									
R296626 (486B2h)	EVENTLOG4_FIFO12_TIME	EVENTLOG4_FIFO12_TIME [31:16]																00000000h
		EVENTLOG4_FIFO12_TIME [15:0]																
R296628 (486B4h)	EVENTLOG4_FIFO13_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG4_FIFO13_POL	0	0	0	EVENTLOG4_FIFO13_ID [8:0]									
R296630 (486B6h)	EVENTLOG4_FIFO13_TIME	EVENTLOG4_FIFO13_TIME [31:16]																00000000h
		EVENTLOG4_FIFO13_TIME [15:0]																
R296632 (486B8h)	EVENTLOG4_FIFO14_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG4_FIFO14_POL	0	0	0	EVENTLOG4_FIFO14_ID [8:0]									
R296634 (486BAh)	EVENTLOG4_FIFO14_TIME	EVENTLOG4_FIFO14_TIME [31:16]																00000000h
		EVENTLOG4_FIFO14_TIME [15:0]																
R296636 (486BCh)	EVENTLOG4_FIFO15_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG4_FIFO15_POL	0	0	0	EVENTLOG4_FIFO15_ID [8:0]									
R296638 (486BEh)	EVENTLOG4_FIFO15_TIME	EVENTLOG4_FIFO15_TIME [31:16]																00000000h
		EVENTLOG4_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]																
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]																

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
R311552 (4C100h)	Timer3_Control	0	0	0	0	0	0	0	0	0	0	TIMER3_CONTINUOUS	TIMER3_DIR	0	TIMER3_PRESCALE [2:0]			00000000h
		0	TIMER3_REFCLK_DIV [2:0]			0	TIMER3_REFCLK_FREQ_SEL [2:0]			0	0	0	0	TIMER3_REFCLK_SRC [3:0]				
R311554 (4C102h)	Timer3_Count_Preset	TIMER3_MAX_COUNT [31:16]																00000000h
		TIMER3_MAX_COUNT [15:0]																
R311558 (4C106h)	Timer3_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	TIMER3_STOP	0	0	0	TIMER3_START	
R311560 (4C108h)	Timer3_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	TIMER3_RUNNING_STS	
R311562 (4C10Ah)	Timer3_Count_Readback	TIMER3_CUR_COUNT [31:16]																00000000h
		TIMER3_CUR_COUNT [15:0]																
R311680 (4C180h)	Timer4_Control	0	0	0	0	0	0	0	0	0	0	TIMER4_CONTINUOUS	TIMER4_DIR	0	TIMER4_PRESCALE [2:0]			00000000h
		0	TIMER4_REFCLK_DIV [2:0]			0	TIMER4_REFCLK_FREQ_SEL [2:0]			0	0	0	0	TIMER4_REFCLK_SRC [3:0]				
R311682 (4C182h)	Timer4_Count_Preset	TIMER4_MAX_COUNT [31:16]																00000000h
		TIMER4_MAX_COUNT [15:0]																
R311686 (4C186h)	Timer4_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	TIMER4_STOP	0	0	0	TIMER4_START	
R311688 (4C188h)	Timer4_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	TIMER4_RUNNING_STS	
R311690 (4C18Ah)	Timer4_Count_Readback	TIMER4_CUR_COUNT [31:16]																00000000h
		TIMER4_CUR_COUNT [15:0]																
R315392 (4D000h)	DSPGP_Status_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSPGP16_STS	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	0000FFFFh
		DSPGP16_SET1_MASK	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	0000FFFFh
		DSPGP16_SET1_DIR	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
		DSPGP16_SET1_LVL	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	0000FFFFh
		DSPGP16_SET2_MASK	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	0000FFFFh
		DSPGP16_SET2_DIR	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
		DSPGP16_SET2_LVL	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	0000FFFFh
		DSPGP16_SET3_MASK	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	0000FFFFh
		DSPGP16_SET3_DIR	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
		DSPGP16_SET3_LVL	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	0000FFFFh
		DSPGP16_SET4_MASK	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	0000FFFFh
		DSPGP16_SET4_DIR	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
		DSPGP16_SET4_LVL	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	

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
R327680 (50000h)	RA_MIF_Thread_Ctrl_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	RA_MIF_STS	
R327684 (50004h)	RA_MIF_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_MIF_SHARE_STS	
R327688 (50008h)	RA_MIF_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_MIF_NUM [5:0]						
R327696 (50010h)	RA_MIF1_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_MIF1_IN_USE_STS	RA_MIF1_SHARE	0	0	0	0	0	0	0	0	0	RA_MIF1_OWNER [4:0]					
R327698 (50012h)	RA_MIF1_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_MIF1_IN_USE_SET [4:0]					
R327700 (50014h)	RA_MIF1_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_MIF1_IN_USE_CLR [4:0]					
R327702 (50016h)	RA_MIF1_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_MIF1_CAP_I2C	
R327704 (50018h)	RA_MIF1_Thread_Ctrl_Debug_1	RA_MIF1_IN_USE_DBG0 [31:16]																00000000h
		RA_MIF1_IN_USE_DBG0 [15:0]																
R328704 (50400h)	RA_EVENTLOG_Thread_Ctrl_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	RA_EVENTLOG_STS [3:0]				
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	RA_EVENTLOG_SHARE_STS [3:0]				
R328712 (50408h)	RA_EVENTLOG_Thread_Ctrl_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000004h
		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_IN_USE_STS	RA_EVENTLOG1_SHARE	0	0	0	0	0	0	0	0	0	RA_EVENTLOG1_OWNER [4:0]					
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]																
R328738 (50422h)R328736 (50420h)0	RA_EVENTLOG2_Thread_Ctrl_20	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	RA_EVENTLOG2_IN_USE_SET [4:0]															
R328740 (50424h)	RA_EVENTLOG2_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_EVENTLOG2_IN_USE_CLR [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]																
R328752 (50430h)	RA_EVENTLOG3_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_EVENTLOG3_IN_USE_STS	RA_EVENTLOG3_SHARE	0	0	0	0	0	0	0	0	0	RA_EVENTLOG3_OWNER [4:0]					
R328754 (50432h)	RA_EVENTLOG3_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_EVENTLOG3_IN_USE_SET [4:0]					
R328756 (50434h)	RA_EVENTLOG3_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_EVENTLOG3_IN_USE_CLR [4:0]					
R328760 (50438h)	RA_EVENTLOG3_Thread_Ctrl_Debug_1	RA_EVENTLOG3_IN_USE_DBG0 [31:16]																00000000h
		RA_EVENTLOG3_IN_USE_DBG0 [15:0]																
R328770 (50442h)R328768 (50440h)0	RA_EVENTLOG4_Thread_Ctrl_20	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	RA_EVENTLOG4_IN_USE_SET [4:0]															
R328772 (50444h)	RA_EVENTLOG4_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_EVENTLOG4_IN_USE_CLR [4:0]					
R328776 (50448h)	RA_EVENTLOG4_Thread_Ctrl_Debug_1	RA_EVENTLOG4_IN_USE_DBG0 [31:16]																00000000h
		RA_EVENTLOG4_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	00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	RA_TIMER_STS [3:0]				
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	RA_TIMER_SHARE_STS [3:0]				
R329736 (50808h)	RA_TIMER_Thread_Ctrl_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000004h
		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]					

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
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]																
R329776 (50830h)	RA_TIMER3_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_TIMER3_IN_USE_STS	RA_TIMER3_SHARE	0	0	0	0	0	0	0	0	0	RA_TIMER3_OWNER [4:0]					
R329778 (50832h)	RA_TIMER3_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_TIMER3_IN_USE_SET [4:0]					
R329780 (50834h)	RA_TIMER3_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_TIMER3_IN_USE_CLR [4:0]					
R329782 (50836h)	RA_TIMER3_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_TIMER3_CAP_EVT	
R329784 (50838h)	RA_TIMER3_Thread_Ctrl_Debug_1	RA_TIMER3_IN_USE_DBG0 [31:16]																00000000h
		RA_TIMER3_IN_USE_DBG0 [15:0]																
R329792 (50840h)	RA_TIMER4_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_TIMER4_IN_USE_STS	RA_TIMER4_SHARE	0	0	0	0	0	0	0	0	0	RA_TIMER4_OWNER [4:0]					
R329794 (50842h)	RA_TIMER4_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_TIMER4_IN_USE_SET [4:0]					
R329796 (50844h)	RA_TIMER4_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_TIMER4_IN_USE_CLR [4:0]					
R329798 (50846h)	RA_TIMER4_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_TIMER4_CAP_EVT	
R329800 (50848h)	RA_TIMER4_Thread_Ctrl_Debug_1	RA_TIMER4_IN_USE_DBG0 [31:16]																00000000h
		RA_TIMER4_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	00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	RA_DSPGP_SET_STS [3:0]				
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	0	0	0	0	RA_DSPGP_SET_SHARE_STS [3:0]				
R330760 (50C08h)	RA_DSPGP_SET_Thread_Ctrl_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000004h
		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]					

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
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]					
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]					
R330804 (50C34h)	RA_DSPGP_SET3_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_SET3_IN_USE_CLR [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]																
R331776 (51000h)	RA_SPARE_A_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_A_STS [15:0]																
R331780 (51004h)	RA_SPARE_A_Thread_Ctrl_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_A_SHARE_STS [15:0]																
R331784 (51008h)	RA_SPARE_A_Thread_Ctrl_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000010h
		0	0	0	0	0	0	0	0	0	0	RA_SPARE_A_NUM [5:0]						
R331792 (51010h)	RA_SPARE_A1_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_A1_IN_USE_STS	RA_SPARE_A1_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_A1_OWNER [4:0]					
R331794 (51012h)	RA_SPARE_A1_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_SPARE_A1_IN_USE_SET [4:0]					
R331796 (51014h)	RA_SPARE_A1_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_SPARE_A1_IN_USE_CLR [4:0]					
R331800 (51018h)	RA_SPARE_A1_Thread_Ctrl_Debug_1	RA_SPARE_A1_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_A1_IN_USE_DBG0 [15:0]																
R331808 (51020h)	RA_SPARE_A2_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_A2_IN_USE_STS	RA_SPARE_A2_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_A2_OWNER [4:0]					
R331810 (51022h)	RA_SPARE_A2_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_SPARE_A2_IN_USE_SET [4:0]					
R331812 (51024h)	RA_SPARE_A2_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_SPARE_A2_IN_USE_CLR [4:0]					
R331816 (51028h)	RA_SPARE_A2_Thread_Ctrl_Debug_1	RA_SPARE_A2_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_A2_IN_USE_DBG0 [15:0]																
R331824 (51030h)	RA_SPARE_A3_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_A3_IN_USE_STS	RA_SPARE_A3_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_A3_OWNER [4:0]					
R331826 (51032h)	RA_SPARE_A3_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_SPARE_A3_IN_USE_SET [4:0]					
R331828 (51034h)	RA_SPARE_A3_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_SPARE_A3_IN_USE_CLR [4:0]					
R331832 (51038h)	RA_SPARE_A3_Thread_Ctrl_Debug_1	RA_SPARE_A3_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_A3_IN_USE_DBG0 [15:0]																
R331840 (51040h)	RA_SPARE_A4_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_A4_IN_USE_STS	RA_SPARE_A4_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_A4_OWNER [4:0]					
R331842 (51042h)	RA_SPARE_A4_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_SPARE_A4_IN_USE_SET [4:0]					
R331844 (51044h)	RA_SPARE_A4_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_SPARE_A4_IN_USE_CLR [4:0]					
R331848 (51048h)	RA_SPARE_A4_Thread_Ctrl_Debug_1	RA_SPARE_A4_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_A4_IN_USE_DBG0 [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
R331856 (51050h)	RA_SPARE_A5_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_A5_IN_USE_STS	RA_SPARE_A5_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_A5_OWNER [4:0]					
R331858 (51052h)	RA_SPARE_A5_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_SPARE_A5_IN_USE_SET [4:0]					
R331860 (51054h)	RA_SPARE_A5_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_SPARE_A5_IN_USE_CLR [4:0]					
R331864 (51058h)	RA_SPARE_A5_Thread_Ctrl_Debug_1	RA_SPARE_A5_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_A5_IN_USE_DBG0 [15:0]																
R331872 (51060h)	RA_SPARE_A6_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_A6_IN_USE_STS	RA_SPARE_A6_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_A6_OWNER [4:0]					
R331874 (51062h)	RA_SPARE_A6_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_SPARE_A6_IN_USE_SET [4:0]					
R331876 (51064h)	RA_SPARE_A6_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_SPARE_A6_IN_USE_CLR [4:0]					
R331880 (51068h)	RA_SPARE_A6_Thread_Ctrl_Debug_1	RA_SPARE_A6_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_A6_IN_USE_DBG0 [15:0]																
R331888 (51070h)	RA_SPARE_A7_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_A7_IN_USE_STS	RA_SPARE_A7_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_A7_OWNER [4:0]					
R331890 (51072h)	RA_SPARE_A7_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_SPARE_A7_IN_USE_SET [4:0]					
R331892 (51074h)	RA_SPARE_A7_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_SPARE_A7_IN_USE_CLR [4:0]					
R331896 (51078h)	RA_SPARE_A7_Thread_Ctrl_Debug_1	RA_SPARE_A7_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_A7_IN_USE_DBG0 [15:0]																
R331904 (51080h)	RA_SPARE_A8_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_A8_IN_USE_STS	RA_SPARE_A8_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_A8_OWNER [4:0]					
R331906 (51082h)	RA_SPARE_A8_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_SPARE_A8_IN_USE_SET [4:0]					
R331908 (51084h)	RA_SPARE_A8_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_SPARE_A8_IN_USE_CLR [4:0]					
R331912 (51088h)	RA_SPARE_A8_Thread_Ctrl_Debug_1	RA_SPARE_A8_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_A8_IN_USE_DBG0 [15:0]																
R331920 (51090h)	RA_SPARE_A9_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_A9_IN_USE_STS	RA_SPARE_A9_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_A9_OWNER [4:0]					
R331922 (51092h)	RA_SPARE_A9_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_SPARE_A9_IN_USE_SET [4:0]					
R331924 (51094h)	RA_SPARE_A9_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_SPARE_A9_IN_USE_CLR [4:0]					
R331928 (51098h)	RA_SPARE_A9_Thread_Ctrl_Debug_1	RA_SPARE_A9_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_A9_IN_USE_DBG0 [15:0]																
R331936 (510A0h)	RA_SPARE_A10_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_A10_IN_USE_STS	RA_SPARE_A10_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_A10_OWNER [4:0]					
R331938 (510A2h)	RA_SPARE_A10_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_SPARE_A10_IN_USE_SET [4:0]					
R331940 (510A4h)	RA_SPARE_A10_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_SPARE_A10_IN_USE_CLR [4:0]					
R331944 (510A8h)	RA_SPARE_A10_Thread_Ctrl_Debug_1	RA_SPARE_A10_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_A10_IN_USE_DBG0 [15:0]																
R331952 (510B0h)	RA_SPARE_A11_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_A11_IN_USE_STS	RA_SPARE_A11_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_A11_OWNER [4:0]					
R331954 (510B2h)	RA_SPARE_A11_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_SPARE_A11_IN_USE_SET [4:0]					
R331956 (510B4h)	RA_SPARE_A11_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_SPARE_A11_IN_USE_CLR [4:0]					
R331960 (510B8h)	RA_SPARE_A11_Thread_Ctrl_Debug_1	RA_SPARE_A11_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_A11_IN_USE_DBG0 [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
R331968 (510C0h)	RA_SPARE_A12_ Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_A12_IN_USE_STS	RA_SPARE_A12_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_A12_OWNER [4:0]					
R331970 (510C2h)	RA_SPARE_A12_ 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_SPARE_A12_IN_USE_SET [4:0]					
R331972 (510C4h)	RA_SPARE_A12_ 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_SPARE_A12_IN_USE_CLR [4:0]					
R331976 (510C8h)	RA_SPARE_A12_ Thread_Ctrl_Debug_1	RA_SPARE_A12_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_A12_IN_USE_DBG0 [15:0]																
R331984 (510D0h)	RA_SPARE_A13_ Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_A13_IN_USE_STS	RA_SPARE_A13_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_A13_OWNER [4:0]					
R331986 (510D2h)	RA_SPARE_A13_ 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_SPARE_A13_IN_USE_SET [4:0]					
R331988 (510D4h)	RA_SPARE_A13_ 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_SPARE_A13_IN_USE_CLR [4:0]					
R331992 (510D8h)	RA_SPARE_A13_ Thread_Ctrl_Debug_1	RA_SPARE_A13_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_A13_IN_USE_DBG0 [15:0]																
R332000 (510E0h)	RA_SPARE_A14_ Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_A14_IN_USE_STS	RA_SPARE_A14_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_A14_OWNER [4:0]					
R332002 (510E2h)	RA_SPARE_A14_ 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_SPARE_A14_IN_USE_SET [4:0]					
R332004 (510E4h)	RA_SPARE_A14_ 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_SPARE_A14_IN_USE_CLR [4:0]					
R332008 (510E8h)	RA_SPARE_A14_ Thread_Ctrl_Debug_1	RA_SPARE_A14_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_A14_IN_USE_DBG0 [15:0]																
R332016 (510F0h)	RA_SPARE_A15_ Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_A15_IN_USE_STS	RA_SPARE_A15_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_A15_OWNER [4:0]					
R332018 (510F2h)	RA_SPARE_A15_ 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_SPARE_A15_IN_USE_SET [4:0]					
R332020 (510F4h)	RA_SPARE_A15_ 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_SPARE_A15_IN_USE_CLR [4:0]					
R332024 (510F8h)	RA_SPARE_A15_ Thread_Ctrl_Debug_1	RA_SPARE_A15_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_A15_IN_USE_DBG0 [15:0]																
R332032 (51100h)	RA_SPARE_A16_ Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_A16_IN_USE_STS	RA_SPARE_A16_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_A16_OWNER [4:0]					
R332034 (51102h)	RA_SPARE_A16_ 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_SPARE_A16_IN_USE_SET [4:0]					
R332036 (51104h)	RA_SPARE_A16_ 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_SPARE_A16_IN_USE_CLR [4:0]					
R332040 (51108h)	RA_SPARE_A16_ Thread_Ctrl_Debug_1	RA_SPARE_A16_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_A16_IN_USE_DBG0 [15:0]																
R332800 (51400h)	RA_SPARE_B_Thread_ Ctrl_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	RA_SPARE_B_STS [7:0]								
R332804 (51404h)	RA_SPARE_B_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_SPARE_B_SHARE_STS [7:0]								
R332808 (51408h)	RA_SPARE_B_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_SPARE_B_NUM [5:0]						
R332816 (51410h)	RA_SPARE_B1_ Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_B1_IN_USE_STS	RA_SPARE_B1_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_B1_OWNER [4:0]					
R332818 (51412h)	RA_SPARE_B1_ 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_SPARE_B1_IN_USE_SET [4:0]					
R332820 (51414h)	RA_SPARE_B1_ 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_SPARE_B1_IN_USE_CLR [4:0]					
R332824 (51418h)	RA_SPARE_B1_ Thread_Ctrl_Debug_1	RA_SPARE_B1_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_B1_IN_USE_DBG0 [15:0]																
R332832 (51420h)	RA_SPARE_B2_ Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_B2_IN_USE_STS	RA_SPARE_B2_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_B2_OWNER [4:0]					
R332834 (51422h)	RA_SPARE_B2_ 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_SPARE_B2_IN_USE_SET [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
R332836 (51424h)	RA_SPARE_B2_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_SPARE_B2_IN_USE_CLR [4:0]					
R332840 (51428h)	RA_SPARE_B2_Thread_Ctrl_Debug_1	RA_SPARE_B2_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_B2_IN_USE_DBG0 [15:0]																
R332848 (51430h)	RA_SPARE_B3_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_B3_IN_USE_STS	RA_SPARE_B3_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_B3_OWNER [4:0]					
R332850 (51432h)	RA_SPARE_B3_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_SPARE_B3_IN_USE_SET [4:0]					
R332852 (51434h)	RA_SPARE_B3_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_SPARE_B3_IN_USE_CLR [4:0]					
R332856 (51438h)	RA_SPARE_B3_Thread_Ctrl_Debug_1	RA_SPARE_B3_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_B3_IN_USE_DBG0 [15:0]																
R332864 (51440h)	RA_SPARE_B4_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_B4_IN_USE_STS	RA_SPARE_B4_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_B4_OWNER [4:0]					
R332866 (51442h)	RA_SPARE_B4_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_SPARE_B4_IN_USE_SET [4:0]					
R332868 (51444h)	RA_SPARE_B4_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_SPARE_B4_IN_USE_CLR [4:0]					
R332872 (51448h)	RA_SPARE_B4_Thread_Ctrl_Debug_1	RA_SPARE_B4_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_B4_IN_USE_DBG0 [15:0]																
R332880 (51450h)	RA_SPARE_B5_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_B5_IN_USE_STS	RA_SPARE_B5_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_B5_OWNER [4:0]					
R332882 (51452h)	RA_SPARE_B5_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_SPARE_B5_IN_USE_SET [4:0]					
R332884 (51454h)	RA_SPARE_B5_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_SPARE_B5_IN_USE_CLR [4:0]					
R332888 (51458h)	RA_SPARE_B5_Thread_Ctrl_Debug_1	RA_SPARE_B5_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_B5_IN_USE_DBG0 [15:0]																
R332896 (51460h)	RA_SPARE_B6_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_B6_IN_USE_STS	RA_SPARE_B6_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_B6_OWNER [4:0]					
R332898 (51462h)	RA_SPARE_B6_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_SPARE_B6_IN_USE_SET [4:0]					
R332900 (51464h)	RA_SPARE_B6_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_SPARE_B6_IN_USE_CLR [4:0]					
R332904 (51468h)	RA_SPARE_B6_Thread_Ctrl_Debug_1	RA_SPARE_B6_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_B6_IN_USE_DBG0 [15:0]																
R332912 (51470h)	RA_SPARE_B7_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_B7_IN_USE_STS	RA_SPARE_B7_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_B7_OWNER [4:0]					
R332914 (51472h)	RA_SPARE_B7_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_SPARE_B7_IN_USE_SET [4:0]					
R332916 (51474h)	RA_SPARE_B7_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_SPARE_B7_IN_USE_CLR [4:0]					
R332920 (51478h)	RA_SPARE_B7_Thread_Ctrl_Debug_1	RA_SPARE_B7_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_B7_IN_USE_DBG0 [15:0]																
R332928 (51480h)	RA_SPARE_B8_Thread_Ctrl_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		RA_SPARE_B8_IN_USE_STS	RA_SPARE_B8_SHARE	0	0	0	0	0	0	0	0	0	RA_SPARE_B8_OWNER [4:0]					
R332930 (51482h)	RA_SPARE_B8_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_SPARE_B8_IN_USE_SET [4:0]					
R332932 (51484h)	RA_SPARE_B8_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_SPARE_B8_IN_USE_CLR [4:0]					
R332936 (51488h)	RA_SPARE_B8_Thread_Ctrl_Debug_1	RA_SPARE_B8_IN_USE_DBG0 [31:16]																00000000h
		RA_SPARE_B8_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]																
R548858 (85FFAh)	DSP1_PMEM_12285	0	0	0	0	0	0	0	0	DSP1_PM_8190 [39:32]								00000000h
		DSP1_PM_8190 [31:16]																

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
R548860 (85FFCh)	DSP1_PMEM_12286	DSP1_PM_8190 [15:0]																00000000h
		0	0	0	0	0	0	0	0	DSP1_PM_END [39:32]								
R548862 (85FFEh)	DSP1_PMEM_12287	DSP1_PM_END [31:16]																00000000h
		DSP1_PM_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]																
R688124 (A7FFCh)	DSP1_XMEM_16382	0	0	0	0	0	0	0	0	DSP1_XM_16382 [23:16]								00000000h
		DSP1_XM_16382 [15:0]																
R688126 (A7FFEh)	DSP1_XMEM_16383	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]																
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	0	0	0	0	0	0	DSP1_CLK_SEL [2:0]			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	
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	DSP1_DUALMEM_COLLISION_ADDR [15:0]																00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	DSP1_CLK_SEL_STS [2:0]			DSP1_CLK_AVAIL	
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	DSP1_BUS_ERROR_ADDR_HI [15:0]																00000000h
		DSP1_BUS_ERROR_ADDR_LOW [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
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]																
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]																
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]					
R1048576 (100000h)	DSP2_PMEM_0	0	0	0	0	0	0	0	0	DSP2_PM_START [39:32]								00000000h
		DSP2_PM_START [31:16]																
R1048578 (100002h)	DSP2_PMEM_1	DSP2_PM_START [15:0]																00000000h
		0	0	0	0	0	0	0	0	DSP2_PM_1 [39:32]								
R1048580 (100004h)	DSP2_PMEM_2	DSP2_PM_1 [31:16]																00000000h
		DSP2_PM_1 [15:0]																
R1110010 (10EFFAh)	DSP2_PMEM_30717	0	0	0	0	0	0	0	0	DSP2_PM_20478 [39:32]								00000000h
		DSP2_PM_20478 [31:16]																
R1110012 (10EFFCh)	DSP2_PMEM_30718	DSP2_PM_20478 [15:0]																00000000h
		0	0	0	0	0	0	0	0	DSP2_PM_END [39:32]								
R1110014 (10EFFEh)	DSP2_PMEM_30719	DSP2_PM_END [31:16]																00000000h
		DSP2_PM_END [15:0]																
R1179648 (120000h)	DSP2_XMEM_0	0	0	0	0	0	0	0	0	DSP2_XM_START [23:16]								00000000h
		DSP2_XM_START [15:0]																
R1179650 (120002h)	DSP2_XMEM_1	0	0	0	0	0	0	0	0	DSP2_XM_1 [23:16]								00000000h
		DSP2_XM_1 [15:0]																
R1228796 (12BFFCh)	DSP2_XMEM_24574	0	0	0	0	0	0	0	0	DSP2_XM_24574 [23:16]								00000000h
		DSP2_XM_24574 [15:0]																
R1228798 (12BFFEh)	DSP2_XMEM_24575	0	0	0	0	0	0	0	0	DSP2_XM_END [23:16]								00000000h
		DSP2_XM_END [15:0]																
R1269760 (136000h)	DSP2_XMEM_EXT_0	0	0	0	0	0	0	0	0	DSP2_XM_EXT_START [23:16]								00000000h
		DSP2_XM_EXT_START [15:0]																
R1269762 (136002h)	DSP2_XMEM_EXT_1	0	0	0	0	0	0	0	0	DSP2_XM_EXT_1 [23:16]								00000000h
		DSP2_XM_EXT_1 [15:0]																
R1277948 (137FFCh)	DSP2_XMEM_EXT_4094	0	0	0	0	0	0	0	0	DSP2_XM_EXT_4094 [23:16]								00000000h
		DSP2_XM_EXT_4094 [15:0]																
R1277950 (137FFEh)	DSP2_XMEM_EXT_4095	0	0	0	0	0	0	0	0	DSP2_XM_EXT_END [23:16]								00000000h
		DSP2_XM_EXT_END [15:0]																
R1310720 (140000h)	DSP2_YMEM_0	0	0	0	0	0	0	0	0	DSP2_YM_START [23:16]								00000000h
		DSP2_YM_START [15:0]																
R1310722 (140002h)	DSP2_YMEM_1	0	0	0	0	0	0	0	0	DSP2_YM_1 [23:16]								00000000h
		DSP2_YM_1 [15:0]																
R1359868 (14BFFCh)	DSP2_YMEM_24574	0	0	0	0	0	0	0	0	DSP2_YM_24574 [23:16]								00000000h
		DSP2_YM_24574 [15:0]																
R1359870 (14BFFEh)	DSP2_YMEM_24575	0	0	0	0	0	0	0	0	DSP2_YM_END [23:16]								00000000h
		DSP2_YM_END [15:0]																
R1441792 (160000h)	DSP2_ZMEM_0	0	0	0	0	0	0	0	0	DSP2_ZM_START [23:16]								00000000h
		DSP2_ZM_START [15:0]																
R1441794 (160002h)	DSP2_ZMEM_1	0	0	0	0	0	0	0	0	DSP2_ZM_1 [23:16]								00000000h
		DSP2_ZM_1 [15:0]																
R1449980 (161FFCh)	DSP2_ZMEM_4094	0	0	0	0	0	0	0	0	DSP2_ZM_4094 [23:16]								00000000h
		DSP2_ZM_4094 [15:0]																
R1449982 (161FFEh)	DSP2_ZMEM_4095	0	0	0	0	0	0	0	0	DSP2_ZM_END [23:16]								00000000h
		DSP2_ZM_END [15:0]																
R1572352 (17FE00h)	DSP2_Config_1	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP2_CLK_SEL [2:0]			00000000h
		0	DSP2_RATE [3:0]				0	0	0	0	0	0	DSP2_MEM_ENA	DSP2_DBG_CLK_ENA	0	DSP2_CORE_ENA	DSP2_START	
R1572356 (17FE04h)	DSP2_Status_1	DSP2_PING_FULL	DSP2_PONG_FULL	0	0	0	0	0	0	DSP2_WDMA_ACTIVE_CHANNELS [7:0]								00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R1572358 (17FE06h)	DSP2_Status_2	DSP2_DUALMEM_COLLISION_ADDR [15:0]																00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	DSP2_CLK_SEL_STS [2:0]			DSP2_CLK_AVAIL	
R1572368 (17FE10h)	DSP2_WDMA_Buffer_1	DSP2_START_ADDRESS_WDMA_BUFFER_1 [15:0]																00000000h
		DSP2_START_ADDRESS_WDMA_BUFFER_0 [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
R1572370 (17FE12h)	DSP2_WDMA_Buffer_2	DSP2_START_ADDRESS_WDMA_BUFFER_3 [15:0]																00000000h
		DSP2_START_ADDRESS_WDMA_BUFFER_2 [15:0]																
R1572372 (17FE14h)	DSP2_WDMA_Buffer_3	DSP2_START_ADDRESS_WDMA_BUFFER_5 [15:0]																00000000h
		DSP2_START_ADDRESS_WDMA_BUFFER_4 [15:0]																
R1572374 (17FE16h)	DSP2_WDMA_Buffer_4	DSP2_START_ADDRESS_WDMA_BUFFER_7 [15:0]																00000000h
		DSP2_START_ADDRESS_WDMA_BUFFER_6 [15:0]																
R1572384 (17FE20h)	DSP2_RDMA_Buffer_1	DSP2_START_ADDRESS_RDMA_BUFFER_1 [15:0]																00000000h
		DSP2_START_ADDRESS_RDMA_BUFFER_0 [15:0]																
R1572386 (17FE22h)	DSP2_RDMA_Buffer_2	DSP2_START_ADDRESS_RDMA_BUFFER_3 [15:0]																00000000h
		DSP2_START_ADDRESS_RDMA_BUFFER_2 [15:0]																
R1572388 (17FE24h)	DSP2_RDMA_Buffer_3	DSP2_START_ADDRESS_RDMA_BUFFER_5 [15:0]																00000000h
		DSP2_START_ADDRESS_RDMA_BUFFER_4 [15:0]																
R1572400 (17FE30h)	DSP2_DMA_Config_1	0	0	0	0	0	0	0	0	DSP2_WDMA_CHANNEL_ENABLE [7:0]								00000000h
		0	0	DSP2_DMA_BUFFER_LENGTH [13:0]														
R1572402 (17FE32h)	DSP2_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	DSP2_WDMA_CHANNEL_OFFSET [7:0]								
R1572404 (17FE34h)	DSP2_DMA_Config_3	0	0	0	0	0	0	0	0	0	0	DSP2_RDMA_CHANNEL_OFFSET [5:0]						00000000h
		0	0	0	0	0	0	0	0	0	0	DSP2_RDMA_CHANNEL_ENABLE [5:0]						
R1572406 (17FE36h)	DSP2_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	DSP2_DMA_WORD_SEL	
R1572408 (17FE38h)	DSP2_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	DSP2_START_IN_SEL [4:0]					
R1572416 (17FE40h)	DSP2_Scratch_1	DSP2_SCRATCH_1 [15:0]																00000000h
		DSP2_SCRATCH_0 [15:0]																
R1572418 (17FE42h)	DSP2_Scratch_2	DSP2_SCRATCH_3 [15:0]																00000000h
		DSP2_SCRATCH_2 [15:0]																
R1572434 (17FE52h)	DSP2_Bus_Error_Addr	DSP2_BUS_ERROR_ADDR_HI [15:0]																00000000h
		DSP2_BUS_ERROR_ADDR_LOW [15:0]																
R1572436 (17FE54h)	DSP2_Ext_window_A	DSP2_EXT_A_PSIZE16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP2_EXT_A_PAGE [15:0]																
R1572438 (17FE56h)	DSP2_Ext_window_B	DSP2_EXT_B_PSIZE16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP2_EXT_B_PAGE [15:0]																
R1572440 (17FE58h)	DSP2_Ext_window_C	DSP2_EXT_C_PSIZE16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP2_EXT_C_PAGE [15:0]																
R1572442 (17FE5Ah)	DSP2_Ext_window_D	DSP2_EXT_D_PSIZE16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP2_EXT_D_PAGE [15:0]																
R1572448 (17FE60h)	DSP2_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	DSP2_CORE_NUMBER [4:0]					
R1572864 (180000h)	DSP3_PMEM_0	0	0	0	0	0	0	0	0	DSP3_PM_START [39:32]								00000000h
		DSP3_PM_START [31:16]																
R1572866 (180002h)	DSP3_PMEM_1	DSP3_PM_START [15:0]																00000000h
		0	0	0	0	0	0	0	0	DSP3_PM_1 [39:32]								
R1572868 (180004h)	DSP3_PMEM_2	DSP3_PM_1 [31:16]																00000000h
		DSP3_PM_1 [15:0]																
R1634298 (18EFFAh)	DSP3_PMEM_30717	0	0	0	0	0	0	0	0	DSP3_PM_20478 [39:32]								00000000h
		DSP3_PM_20478 [31:16]																
R1634300 (18EFFCh)	DSP3_PMEM_30718	DSP3_PM_20478 [15:0]																00000000h
		0	0	0	0	0	0	0	0	DSP3_PM_END [39:32]								
R1634302 (18EFFEh)	DSP3_PMEM_30719	DSP3_PM_END [31:16]																00000000h
		DSP3_PM_END [15:0]																
R1703936 (1A0000h)	DSP3_XMEM_0	0	0	0	0	0	0	0	0	DSP3_XM_START [23:16]								00000000h
		DSP3_XM_START [15:0]																
R1703938 (1A0002h)	DSP3_XMEM_1	0	0	0	0	0	0	0	0	DSP3_XM_1 [23:16]								00000000h
		DSP3_XM_1 [15:0]																
R1777660 (1B1FFCh)	DSP3_XMEM_36862	0	0	0	0	0	0	0	0	DSP3_XM_36862 [23:16]								00000000h
		DSP3_XM_36862 [15:0]																
R1777662 (1B1FFEh)	DSP3_XMEM_36863	0	0	0	0	0	0	0	0	DSP3_XM_END [23:16]								00000000h
		DSP3_XM_END [15:0]																
R1794048 (1B6000h)	DSP3_XMEM_EXT_0	0	0	0	0	0	0	0	0	DSP3_XM_EXT_START [23:16]								00000000h
		DSP3_XM_EXT_START [15:0]																
R1794050 (1B6002h)	DSP3_XMEM_EXT_1	0	0	0	0	0	0	0	0	DSP3_XM_EXT_1 [23:16]								00000000h
		DSP3_XM_EXT_1 [15:0]																
R1802236 (1B7FFCh)	DSP3_XMEM_EXT_4094	0	0	0	0	0	0	0	0	DSP3_XM_EXT_4094 [23:16]								00000000h
		DSP3_XM_EXT_4094 [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
R1802238 (1B7FFEh)	DSP3_XMEM_EXT_4095	0	0	0	0	0	0	0	0	DSP3_XM_EXT_END [23:16]								00000000h
		DSP3_XM_EXT_END [15:0]																
R1835008 (1C0000h)	DSP3_YMEM_0	0	0	0	0	0	0	0	0	DSP3_YM_START [23:16]								00000000h
		DSP3_YM_START [15:0]																
R1835010 (1C0002h)	DSP3_YMEM_1	0	0	0	0	0	0	0	0	DSP3_YM_1 [23:16]								00000000h
		DSP3_YM_1 [15:0]																
R1884156 (1CBFFCh)	DSP3_YMEM_24574	0	0	0	0	0	0	0	0	DSP3_YM_24574 [23:16]								00000000h
		DSP3_YM_24574 [15:0]																
R1884158 (1CBFFEh)	DSP3_YMEM_24575	0	0	0	0	0	0	0	0	DSP3_YM_END [23:16]								00000000h
		DSP3_YM_END [15:0]																
R1966080 (1E0000h)	DSP3_ZMEM_0	0	0	0	0	0	0	0	0	DSP3_ZM_START [23:16]								00000000h
		DSP3_ZM_START [15:0]																
R1966082 (1E0002h)	DSP3_ZMEM_1	0	0	0	0	0	0	0	0	DSP3_ZM_1 [23:16]								00000000h
		DSP3_ZM_1 [15:0]																
R1974268 (1E1FFCh)	DSP3_ZMEM_4094	0	0	0	0	0	0	0	0	DSP3_ZM_4094 [23:16]								00000000h
		DSP3_ZM_4094 [15:0]																
R1974270 (1E1FFEh)	DSP3_ZMEM_4095	0	0	0	0	0	0	0	0	DSP3_ZM_END [23:16]								00000000h
		DSP3_ZM_END [15:0]																
R2096640 (1FFE00h)	DSP3_Config_1	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP3_CLK_SEL [2:0]			00000000h
		0	DSP3_RATE [3:0]				0	0	0	0	0	0	DSP3_MEM_ENA	DSP3_DBG_CLK_ENA	0	DSP3_CORE_ENA	DSP3_START	
R2096644 (1FFE04h)	DSP3_Status_1	DSP3_PING_FULL	DSP3_PONG_FULL	0	0	0	0	0	0	DSP3_WDMA_ACTIVE_CHANNELS [7:0]								00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R2096646 (1FFE06h)	DSP3_Status_2	DSP3_DUALMEM_COLLISION_ADDR [15:0]																00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	DSP3_CLK_SEL_STS [2:0]			DSP3_CLK_AVAIL	
R2096656 (1FFE10h)	DSP3_WDMA_Buffer_1	DSP3_START_ADDRESS_WDMA_BUFFER_1 [15:0]																00000000h
		DSP3_START_ADDRESS_WDMA_BUFFER_0 [15:0]																
R2096658 (1FFE12h)	DSP3_WDMA_Buffer_2	DSP3_START_ADDRESS_WDMA_BUFFER_3 [15:0]																00000000h
		DSP3_START_ADDRESS_WDMA_BUFFER_2 [15:0]																
R2096660 (1FFE14h)	DSP3_WDMA_Buffer_3	DSP3_START_ADDRESS_WDMA_BUFFER_5 [15:0]																00000000h
		DSP3_START_ADDRESS_WDMA_BUFFER_4 [15:0]																
R2096662 (1FFE16h)	DSP3_WDMA_Buffer_4	DSP3_START_ADDRESS_WDMA_BUFFER_7 [15:0]																00000000h
		DSP3_START_ADDRESS_WDMA_BUFFER_6 [15:0]																
R2096672 (1FFE20h)	DSP3_RDMA_Buffer_1	DSP3_START_ADDRESS_RDMA_BUFFER_1 [15:0]																00000000h
		DSP3_START_ADDRESS_RDMA_BUFFER_0 [15:0]																
R2096674 (1FFE22h)	DSP3_RDMA_Buffer_2	DSP3_START_ADDRESS_RDMA_BUFFER_3 [15:0]																00000000h
		DSP3_START_ADDRESS_RDMA_BUFFER_2 [15:0]																
R2096676 (1FFE24h)	DSP3_RDMA_Buffer_3	DSP3_START_ADDRESS_RDMA_BUFFER_5 [15:0]																00000000h
		DSP3_START_ADDRESS_RDMA_BUFFER_4 [15:0]																
R2096688 (1FFE30h)	DSP3_DMA_Config_1	0	0	0	0	0	0	0	0	DSP3_WDMA_CHANNEL_ENABLE [7:0]								00000000h
		0	0	DSP3_DMA_BUFFER_LENGTH [13:0]														
R2096690 (1FFE32h)	DSP3_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	DSP3_WDMA_CHANNEL_OFFSET [7:0]								
R2096692 (1FFE34h)	DSP3_DMA_Config_3	0	0	0	0	0	0	0	0	0	0	DSP3_RDMA_CHANNEL_OFFSET [5:0]						00000000h
		0	0	0	0	0	0	0	0	0	0	DSP3_RDMA_CHANNEL_ENABLE [5:0]						
R2096694 (1FFE36h)	DSP3_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	DSP3_DMA_WORD_SEL	
R2096696 (1FFE38h)	DSP3_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	DSP3_START_IN_SEL [4:0]					
R2096704 (1FFE40h)	DSP3_Scratch_1	DSP3_SCRATCH_1 [15:0]																00000000h
		DSP3_SCRATCH_0 [15:0]																
R2096706 (1FFE42h)	DSP3_Scratch_2	DSP3_SCRATCH_3 [15:0]																00000000h
		DSP3_SCRATCH_2 [15:0]																
R2096722 (1FFE52h)	DSP3_Bus_Error_Addr	DSP3_BUS_ERROR_ADDR_HI [15:0]																00000000h
		DSP3_BUS_ERROR_ADDR_LOW [15:0]																
R2096724 (1FFE54h)	DSP3_Ext_window_A	DSP3_EXT_A_PSIZE16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP3_EXT_A_PAGE [15:0]																
R2096726 (1FFE56h)	DSP3_Ext_window_B	DSP3_EXT_B_PSIZE16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP3_EXT_B_PAGE [15:0]																
R2096728 (1FFE58h)	DSP3_Ext_window_C	DSP3_EXT_C_PSIZE16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP3_EXT_C_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
R2096730 (1FFE5Ah)	DSP3_Ext_window_D	DSP3_EXT_D_PSIZE16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP3_EXT_D_PAGE [15:0]																
R2096736 (1FFE60h)	DSP3_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	DSP3_CORE_NUMBER [4:0]					

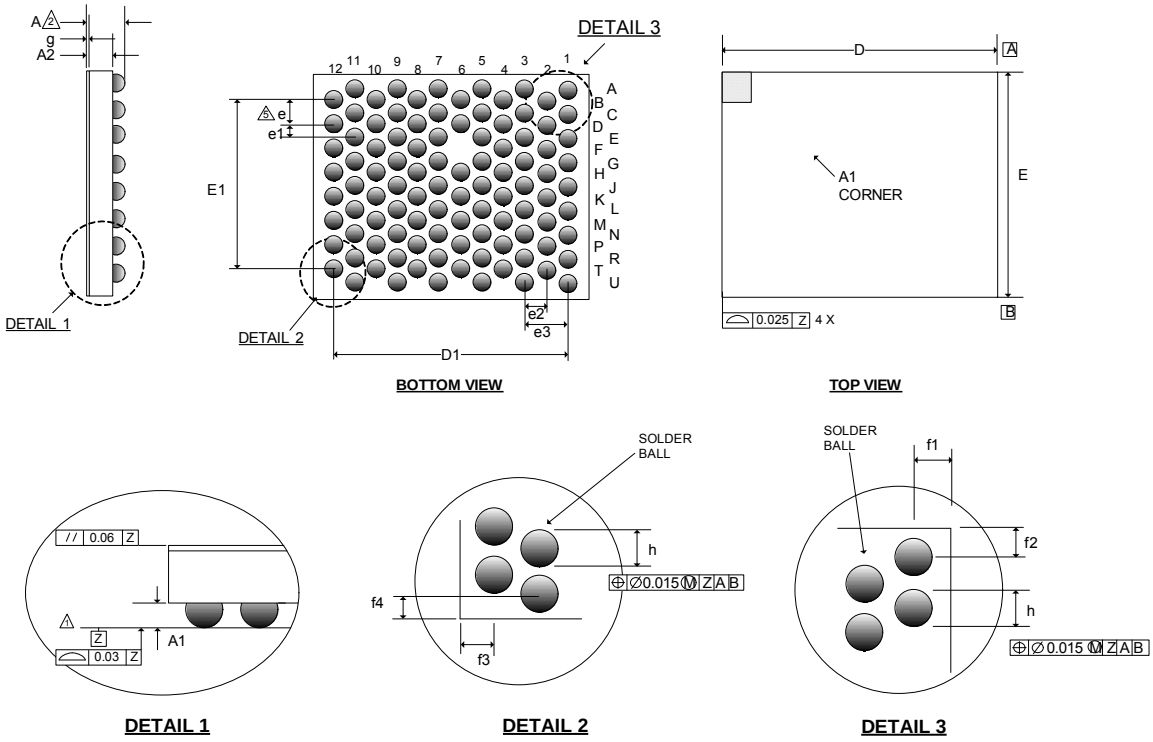
7 Thermal Characteristics

Table 7-1. Typical JEDEC Four-Layer, 2s2p Board Thermal Characteristics

Parameter	Symbol	WLCSP	Units
Junction-to-ambient thermal resistance	θ_{JA}	34.5	°C/W
Junction-to-board thermal resistance	θ_{JB}	7.6	°C/W
Junction-to-case thermal resistance	θ_{JC}	0.895	°C/W
Junction-to-board thermal-characterization parameter	Ψ_{JB}	7.6	°C/W
Junction-to-package-top thermal-characterization parameter	Ψ_{JT}	0.075	°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



Symbols	Dimensions (mm)			
	MIN	NOM	MAX	NOTE
A	0.470	0.504	0.538	2
A1	0.172	0.202	0.232	
A2	0.286	0.302	0.318	
D	4.525	4.555	4.585	
D1		3.811 BSC		
E	3.616	3.646	3.676	
E1		2.800 BSC		
e		0.400 BSC		5
e1		0.200 BSC		5
e2		0.347 BSC		5
e3		0.693 BSC		5
f1		0.372 BSC		
f2		0.223 BSC		
f3		0.372 BSC		
f4		0.223 BSC		
g		0.022		
h	0.232	0.262	0.292	

NOTES:
1. PRIMARY DATUM -Z- AND SEATING PLANE ARE DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS .
2. THIS DIMENSION INCLUDES STAND -OFF HEIGHT 'A 1'.
3. A1 CORNER IS IDENTIFIED BY INL /LASER MARK ON TOP PACKAGE .
4. BILATERAL TOLERANCE ZONE IS APPLIED TO EACH SIDE OF THE PACKAGE BODY .
5. 'e' REPRESENTS THE BASIC SOLDER BALL GRID PITCH .
6. THIS DRAWING IS SUBJECT TO CHANGE WITHOUT NOTICE .
7. FOLLOWS JEDEC DESIGN GUIDE MO -211-C.

Figure 8-1. 101-Ball WLCSP Package Drawing

9 Ordering Information

Table 9-1. Ordering Information

Product	Description	Package	Halogen Free	Pb Free	Grade	Temperature Range	Container	Order #
CS47L35	Smart Codec with Low-Power Audio DSP	101-ball WLCSP	Yes	Yes	Commercial	–40 to +85°C	Tape and Reel [1]	CS47L35-CWZR

1.Reel quantity = 7000

10 References

- MIPI Alliance, *MIPI Alliance Specification for Serial Low-Power Inter-Chip Media Bus (SLIMbus).* http://www.mipi.org
- 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
F1 JUN ‘16	• Series resistor recommended on FLLVDD connection (Section 2, Table 3-3). • THD+N test limits updated (Table 3-9). • Typical power consumption updated (Table 3-23). • DMA data word format (DSPn_DMA_WORD_SEL) control field added (Section 4.4.3). • GPn_OP_CFG and GPn_DIR field descriptions updated (Section 4.14.1, Section 4.14.3). • Deleted I²C support for multiple register read from previous register address (Section 4.17.2). • PCB layout guidelines updated (Section 5.1.6).
F2 JAN ‘17	• Clarification of PDM input/output digital signal levels (Section 4.2.7, Section 4.11.4). • Clarification of DSPn_DUALMEM_COLLISION_ADDR field (Section 4.4.1, Table 4-25). • FLL configuration and example settings updated (Section 4.16.9). • Correction to FLL synchronizer example settings (Table 4-98). • Thermal characteristics updated (Section 7).

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.