CIRRUS LOGIC®

CS42L92

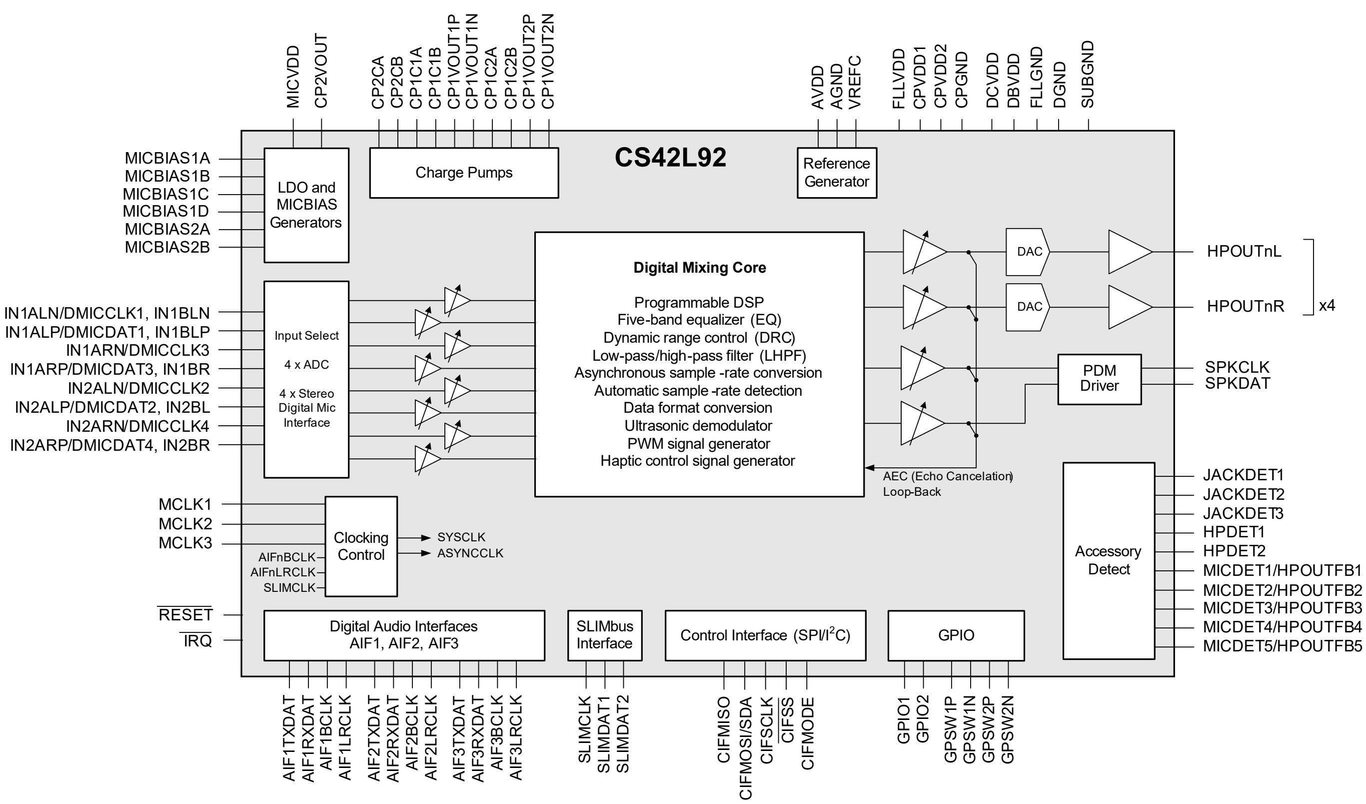
32-Bit 384-kHz Hi-Fi Audio Codec

Features

- Integrated multichannel 32-bit hi-fi audio hub codec
  - 99-dB signal-to-noise ratio (SNR) mic input (48 kHz)
  - 127-dB SNR headphone playback (48 kHz)
  - –100-dB total harmonic distortion + noise (THD+N)
  - Ultrasonic input- and output-path support
- Up to eight analog or digital microphone (DMIC) inputs
- Multipurpose headphone/earpiece/line output drivers
  - Support for balanced headphone output loads
  - 33 mW into 32-Ω load at 0.1% THD+N
  - Hi-fi filters for audiophile-quality playback
- Native audio playback up to 384 kHz sample rate, concurrent with voice and ultrasonic input signal paths
- Digital pulse-density modulation (PDM) output interface
- Multichannel asynchronous sample-rate conversion
- Multiline SLIMbus® audio and control interface
- Three multichannel digital-audio interfaces
  - Standard data formats up to 384 kHz, 32 bits
- Flexible clocking, derived from MCLK*n*, AIF*n*, or SLIMbus
  - Low-power frequency-locked loops (FLLs) support reference clocks down to 32 kHz.
- Advanced accessory detection functions
- Configurable functions on up to 16 general-purpose input/output (GPIO) pins
- Integrated regulators and charge pumps
- Small W-CSP package, 0.4-mm staggered ball array

Applications

- Smartphones, tablets, and multimedia handsets



Description

The CS42L92 is a highly integrated low-power audio system for smartphones, tablets, and other portable audio devices. Multiple input/output paths are supported by a fully flexible all-digital mixing and routing engine, incorporating sample rate converters and other signal-processing functions for wide use-case flexibility. The integrated DSP provides a general-purpose signal processing capability; this is supported by general-purpose timer and event-logger functions.

The digital audio interfaces and hi-fi DACs enable 32-bit playback through the entire signal chain. Native audio playback at sample rates up to 384 kHz is possible, concurrent with voice and ultrasonic input paths.

The CS42L92 supports up to eight analog inputs and up to eight PDM digital inputs. Low-power input modes are available for always-on (e.g., voice-trigger) functionality using either analog or digital input. A smart accessory interface, with multipurpose impedance sensing and measurement capability, supports detection of external headsets and push buttons. Dual headphone connections (e.g., 3.5-mm and USB-C™) can be detected simultaneously.

Four hi-fi quality stereo headphone drivers are provided, each supporting stereo ground-referenced or mono bridge-tied load (BTL) configurations. Multiple headphone/earpiece outputs can be supported, including balanced stereo headphone configurations. The output drivers offer noise levels as low as 0.63 μV_{RMS} into line or headphone loads. Selectable hi-fi filters support playback modes at sample rates up to 384 kHz.

Two channels of PDM output (one stereo interface) are available, and also an IEC-60958-3–compatible S/PDIF transmitter. A signal generator for controlling haptics devices is included; vibe actuators can connect directly to the PDM output interface. All inputs, outputs, and system interfaces can function concurrently.

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. Two FLLs are integrated, providing support for a wide range of system-clock frequencies.

The CS42L92 is configured using the SLIMbus, SPI™, or I^2C interfaces. The device is powered from 1.8- and 1.2-V supplies. 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

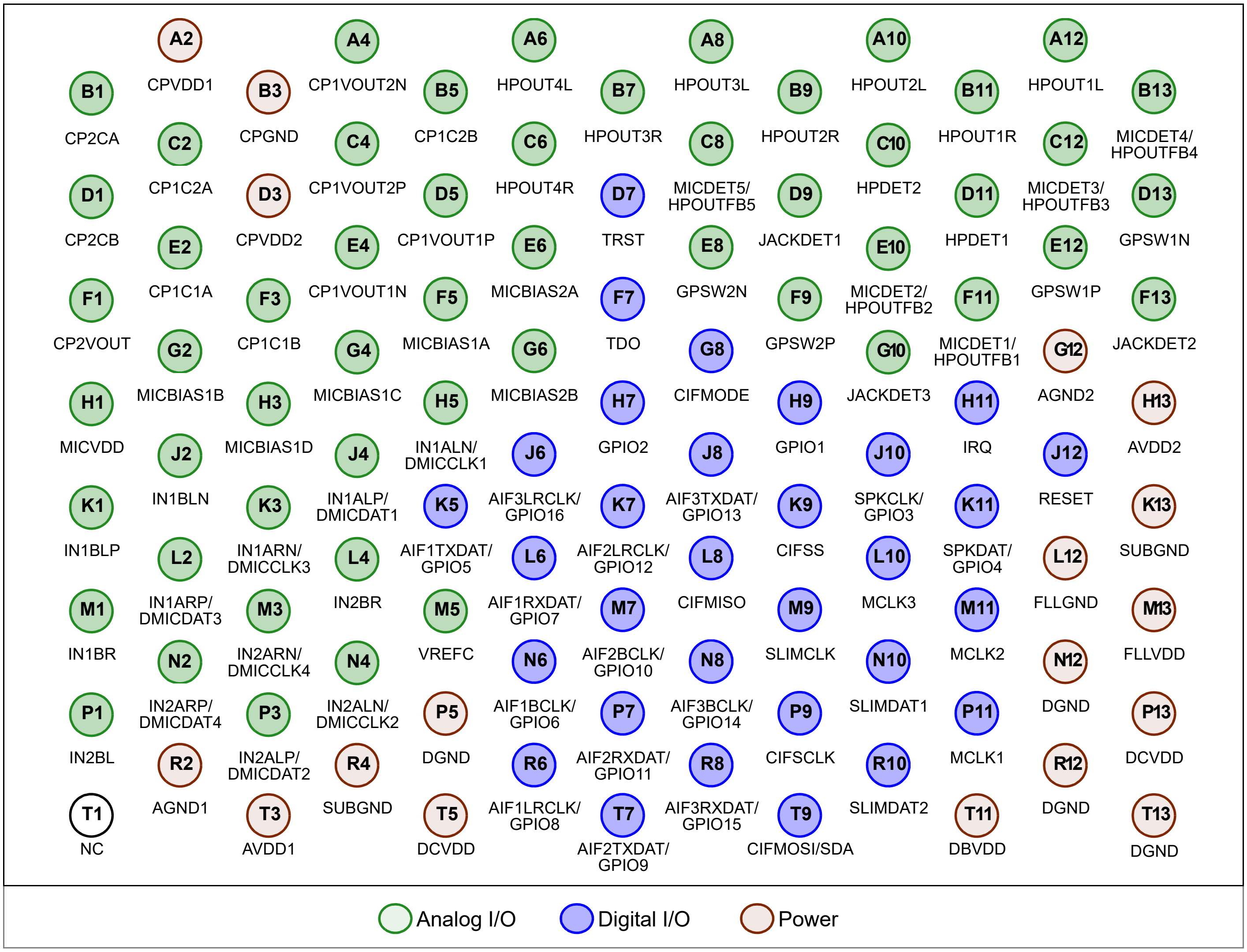


Figure 1-1. Top-Down (Through-Package) View—104-Ball WLCSP Package

1.2 Pin Descriptions

Table 1-1 describes each pin on the CS42L92. Note that pins that share a common name should be tied together on the printed circuit board (PCB).

Table 1-1. Pin Descriptions

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

Pin Name	Pin #	Power Supply	I/O	Pin Description	Digital Pad Attributes	State at Reset [1]
				Analog I/O		
CP1C1A	E2	—	O	Charge Pump 1 fly-back capacitor 1 pin	—	—
CP1C1B	F3	—	O	Charge Pump 1 fly-back capacitor 1 pin	—	—
CP1C2A	C2	—	O	Charge Pump 1 fly-back capacitor 2 pin	—	—
CP1C2B	B5	—	O	Charge Pump 1 fly-back capacitor 2 pin	—	—
CP1VOUT1N	E4	—	O	Charge Pump 1 negative output 1 decoupling pin	—	Output
CP1VOUT1P	D5	—	O	Charge Pump 1 positive output 1 decoupling pin	—	Output
CP1VOUT2N	A4	—	O	Charge Pump 1 negative output 2 decoupling pin	—	Output
CP1VOUT2P	C4	—	O	Charge Pump 1 positive output 2 decoupling pin	—	Output
CP2CA	B1	—	O	Charge Pump 2 fly-back capacitor pin	—	Output
CP2CB	D1	—	O	Charge Pump 2 fly-back capacitor pin	—	Output
CP2VOUT	F1	—	O	Charge Pump 2 output decoupling pin/supply for LDO2	—	Output
GPSW1N	D13	—	I/O	General-purpose bidirectional switch 1 contact	—	—
GPSW1P	E12	—	I/O	General-purpose bidirectional switch 1 contact	—	—
GPSW2N	E8	—	I/O	General-purpose bidirectional switch 2 contact	—	—
GPSW2P	F9	—	I/O	General-purpose bidirectional switch 2 contact	—	—
HPDET1	D11	—	I/O	Headphone sense 1 input	—	Input
HPDET2	C10	—	I/O	Headphone sense 2 input	—	Input
HPOUT1L	A12	—	O	Left headphone 1 output	—	Output
HPOUT1R	B11	—	O	Right headphone 1 output	—	Output
HPOUT2L	A10	—	O	Left headphone 2 output	—	Output
HPOUT2R	B9	—	O	Right headphone 2 output	—	Output
HPOUT3L	A8	—	O	Left headphone 3 output	—	Output
HPOUT3R	B7	—	O	Right headphone 3 output	—	Output
HPOUT4L	A6	—	O	Left headphone 4 output	—	Output
HPOUT4R	C6	—	O	Right headphone 4 output	—	Output
IN1ALN/ DMICCLK1	H5	MICVDD or MICBIASn [2]	I	Left/right-channel negative differential mic/line input/ DMIC Clock Output 1. Also suitable for connection to external accessory interfaces.	—	IN1ALN input
IN1ALP/ DMICDAT1	J4	MICVDD or MICBIASn [2]	I	Left-channel single-ended mic/line input/positive differential mic/line input/DMIC Data Input 1. Also suitable for connection to external accessory interfaces.	PD/H	IN1ALP input
IN1ARN/ DMICCLK3	K3	MICVDD or MICBIASn [2]	I	Right-channel negative differential mic/line input/DMIC Clock Output 3. Also suitable for connection to external accessory interfaces.	—	IN1ARN input
IN1ARP/ DMICDAT3	L2	MICVDD or MICBIASn [2]	I/O	Right-channel single-ended mic/line input/positive differential mic/line input/DMIC Data Input 3. Also suitable for connection to external accessory interfaces.	PD/H	IN1ARP input
IN1BLN	J2	MICVDD	I	Negative differential mic/line input. Also suitable for connection to external accessory interfaces.	—	Input
IN1BLP	K1	MICVDD	I	Single-ended mic/line input/positive differential mic/line input. Also suitable for connection to external accessory interfaces.	—	Input

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

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

Pin Name	Pin #	Power Supply	I/O	Pin Description	Digital Pad Attributes	State at Reset [1]
IN1BR	M1	MICVDD	I	Right-channel single-ended mic/line input/positive differential mic/line input.	—	Input
IN2ALN/ DMICCLK2	N4	MICVDD or MICBIASn [2]	I	Left-channel negative differential mic/line input/DMIC Clock Output 2. Also suitable for connection to external accessory interfaces.	—	IN2ALN input
IN2ALP/ DMICDAT2	P3	MICVDD or MICBIASn [2]	I/O	Left-channel single-ended mic/line input/positive differential mic/line input/DMIC Data Input 2. Also suitable for connection to external accessory interfaces.	PD/H	IN2ALP input
IN2ARN/ DMICCLK4	M3	MICVDD or MICBIASn [2]	I	Right-channel negative differential mic/line input/DMIC Clock Output 4. Also suitable for connection to external accessory interfaces.	—	IN2ARN input
IN2ARP/ DMICDAT4	N2	MICVDD or MICBIASn [2]	I/O	Right-channel single-ended mic/line input/positive differential mic/line input/DMIC Data Input 4. Also suitable for connection to external accessory interfaces.	PD/H	IN2ARP input
IN2BL	P1	MICVDD	I	Left-channel single-ended mic/line input	—	Input
IN2BR	L4	MICVDD	I	Right-channel single-ended mic/line input. Also suitable for connection to external accessory interfaces.	—	Input
JACKDET1	D9	AVDD	I	Jack detect input 1	—	Input
JACKDET2	F13	AVDD	I	Jack detect input 2	—	Input
JACKDET3	G10	AVDD	I	Jack detect input 3	—	Input
MICBIAS1A	F5	MICVDD	O	Microphone bias 1A	—	Output
MICBIAS1B	G2	MICVDD	O	Microphone bias 1B	—	Output
MICBIAS1C	G4	MICVDD	O	Microphone bias 1C	—	Output
MICBIAS1D	H3	MICVDD	O	Microphone bias 1D	—	Output
MICBIAS2A	E6	MICVDD	O	Microphone bias 2A	—	Output
MICBIAS2B	G6	MICVDD	O	Microphone bias 2B	—	Output
MICDET1/ HPOUTFB1	F11	—	I/O	Mic/accessory sense input 1/HPOUT ground feedback pin 1	—	Input
MICDET2/ HPOUTFB2	E10	—	I/O	Mic/accessory sense input 2/HPOUT ground feedback pin 2	—	Input
MICDET3/ HPOUTFB3	C12	—	I/O	Mic/accessory sense input 3/HPOUT ground feedback pin 3	—	Input
MICDET4/ HPOUTFB4	B13	—	I/O	Mic/accessory sense input 4/HPOUT ground feedback pin 4	—	Input
MICDET5/ HPOUTFB5	C8	—	I/O	Mic/accessory sense input 5/HPOUT ground feedback pin 5	—	Input
MICVDD	H1	—	O	LDO2 output decoupling pin (generated internally by CS42L92). (Can also be used as reference/supply for external microphones.)	—	Output
VREFC	M5	—	O	Band-gap reference external capacitor connection	—	Output
				Digital I/O		
AIF1BCLK/ GPIO6	N6	DBVDD	I/O	Audio interface 1 bit clock/GPIO6	PU/PD/K/H/ Z/C/OD	GPIO6 input with bus-keeper
AIF1LRCLK/ GPIO8	R6	DBVDD	I/O	Audio interface 1 left/right clock/GPIO8	PU/PD/K/H/ Z/C/OD	GPIO8 input with bus-keeper
AIF1RXDAT/ GPIO7	L6	DBVDD	I/O	Audio interface 1 RX digital audio data/GPIO7	PU/PD/K/H/ C/OD	GPIO7 input with bus-keeper
AIF1TXDAT/ GPIO5	K5	DBVDD	I/O	Audio interface 1 TX digital audio data/GPIO5	PU/PD/K/H/ Z/C/OD	GPIO5 input with bus-keeper

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

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

Pin Name	Pin #	Power Supply	I/O	Pin Description	Digital Pad Attributes	State at Reset [1]
AIF2BCLK/ GPIO10	M7	DBVDD	I/O	Audio interface 2 bit clock/GPIO10	PU/PD/K/H/ Z/C/OD	GPIO10 input with bus-keeper
AIF2LRCLK/ GPIO12	K7	DBVDD	I/O	Audio interface 2 left/right clock/GPIO12	PU/PD/K/H/ Z/C/OD	GPIO12 input with bus-keeper
AIF2RXDAT/ GPIO11	P7	DBVDD	I/O	Audio interface 2 RX digital audio data/GPIO11	PU/PD/K/H/ C/OD	GPIO11 input with bus-keeper
AIF2TXDAT/ GPIO9	T7	DBVDD	I/O	Audio interface 2 TX digital audio data/GPIO9	PU/PD/K/H/ Z/C/OD	GPIO9 input with bus-keeper
AIF3BCLK/ GPIO14	N8	DBVDD	I/O	Audio interface 3 bit clock/GPIO14. If the JTAG interface is configured, this pin provides the TCK input connection.	PU/PD/K/H/ Z/C/OD	GPIO14 input with bus-keeper
AIF3LRCLK/ GPIO16	J6	DBVDD	I/O	Audio interface 3 left/right clock/GPIO16	PU/PD/K/H/ Z/C/OD	GPIO16 input with bus-keeper
AIF3RXDAT/ GPIO15	R8	DBVDD	I/O	Audio interface 3 RX digital audio data/GPIO15. If the JTAG interface is configured, this pin provides the TDI input connection.	PU/PD/K/H/ C/OD	GPIO15 input with bus-keeper
AIF3TXDAT/ GPIO13	J8	DBVDD	I/O	Audio interface 3 TX digital audio data/GPIO13. If the JTAG interface is configured, this pin provides the TMS input connection.	PU/PD/K/H/ Z/C/OD	GPIO13 input with bus-keeper
CIFMISO	L8	DBVDD	O	Control interface (SPI) Master In Slave Out data. The CIFMISO is high impedance if $\overline{\text{CIF1SS}}$ is not asserted.	Z/C	Output
CIFMOSI/SDA	T9	DBVDD	I/O	Control interface (SPI) Master Out Slave In data/ Control interface (I^{2}C) data input/output.	H/OD	Input
CIFSCLK	P9	DBVDD	I	Control interface clock input	H	Input
$\overline{\text{CIFSS}}$	K9	DBVDD	I	Control interface (SPI) slave select (SS)	H	Input
CIFMODE	G8	DBVDD	I	Control interface mode select	H	Input
GPIO1	H9	DBVDD	I/O	GPIO1	PU/PD/K/H/ C/OD	GPIO1 input with bus-keeper
GPIO2	H7	DBVDD	I/O	GPIO2	PU/PD/K/H/ C/OD	GPIO2 input with bus-keeper
$\overline{\text{IRQ}}$	H11	DBVDD	O	Interrupt request (IRQ) output (default is active low)	C/OD	Output
MCLK1	P11	DBVDD	I	Master clock 1	H	Input
MCLK2	M11	DBVDD	I	Master clock 2	H	Input
MCLK3	L10	DBVDD	I	Master clock 3	H	Input
$\overline{\text{RESET}}$	J12	DBVDD	I	Digital reset input (active low)	PU/PD/K/H	Input with pull-up
SLIMCLK	M9	DBVDD	I/O	SLIMbus clock I/O	H/C	Input
SLIMDAT1	N10	DBVDD	I/O	SLIMbus data I/O	H/C	Input
SLIMDAT2	R10	DBVDD	I/O	SLIMbus data I/O	H/C	Input
SPKCLK/ GPIO3	J10	DBVDD	I/O	Digital speaker (PDM) clock output/GPIO3	PU/PD/K/H/ C/OD	GPIO3 input with bus-keeper
SPKDAT/ GPIO4	K11	DBVDD	I/O	Digital speaker (PDM) data output/GPIO4	PU/PD/K/H/ C/OD	GPIO4 input with bus-keeper
TDO	F7	DBVDD	O	JTAG data output	C	Output
TRST	D7	DBVDD	I	JTAG test access port reset (active low)	PD/H	Input with pull-down
				Supply		
AGND1	R2	—	—	Analog ground (return path for AVDD1)	—	—
AGND2	G12	—	—	Analog ground (return path for AVDD2)	—	—
AVDD1	T3	—	—	Analog supply	—	—
AVDD2	H13	—	—	Analog supply	—	—

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

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

Pin Name	Pin #	Power Supply	I/O	Pin Description	Digital Pad Attributes	State at Reset [1]
CPGND	B3	—	—	Charge pump ground (return path for CPVDD1, CPVDD2)	—	—
CPVDD1	A2	—	—	Supply for Charge Pump 1 and Charge Pump 2	—	—
CPVDD2	D3	—	—	Secondary supply for Charge Pump 1	—	—
DBVDD	T11	—	—	Digital buffer (I/O) supply	—	—
DCVDD	P13, T5	—	—	Digital core supply	—	—
DGND	N12, P5, R12, T13	—	—	Digital ground (return path for DCVDD and DBVDD)	—	—
FLLGND	L12	—	—	Analog ground (return path for FLLVDD)	—	—
FLLVDD	M13	—	—	Analog FLL supply	—	—
SUBGND	K13, R4	—	—	Substrate ground	—	—
No Connect						
NC	T1	—	—	—	—	—

1. Note that the default conditions described are not valid if modified by the boot sequence or by a wake-up control sequence.
2. The analog input functions on these pins are referenced to the MICVDD power domain. The digital input/output functions are referenced to the MICVDD or MICBIAS*n* power domain, as selected by the applicable IN*x*_DMIC_SUP field.

2 Typical Connection Diagram

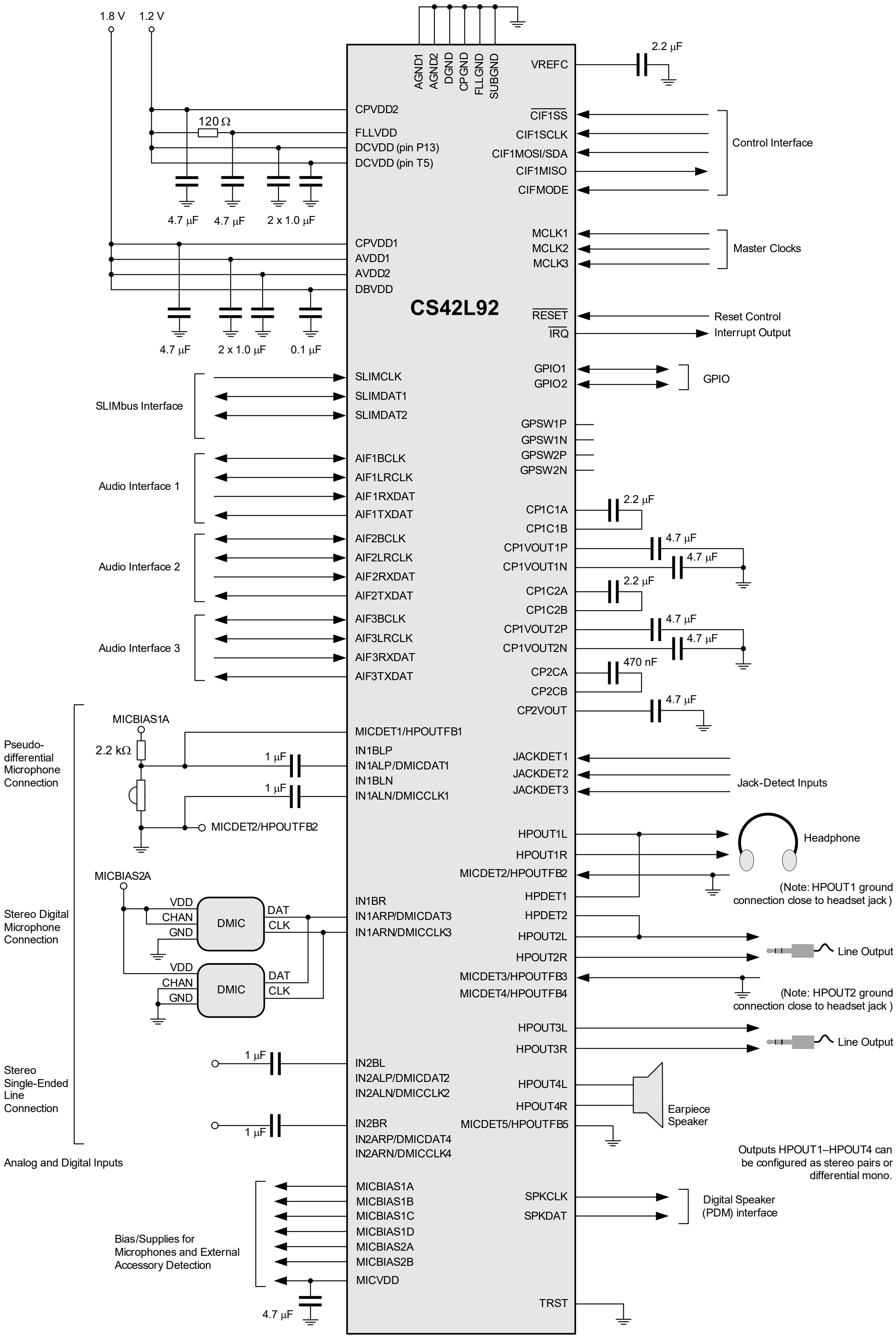


Figure 2-1. Typical Connection Diagram

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]	–0.3 V	1.6 V
	AVDD [2], CPVDD1, CPVDD2	–0.3 V	2.5 V
	DBVDD, MICVDD	–0.3 V	5.0 V
Voltage range digital inputs DBVDD domain	—	SUBGND – 0.3 V	DBVDD + 0.3 V
DMICDAT1–DMICDAT2	—	SUBGND – 0.3 V	MICVDD + 0.3 V
Voltage range analog inputs	IN1ARx, IN2Axx, IN2Bx	SUBGND – 0.3 V	MICVDD + 0.3 V
	IN1ALx, IN1BLx, IN1BR	SUBGND – 0.9 V	MICVDD + 0.3 V
	MICDET*n* [3]	SUBGND – 0.3 V	MICVDD + 0.3 V
	HPOUTFB*n* [3]	SUBGND – 0.3 V	SUBGND + 0.3 V
	JACKDET1, HPDET1, HPDET2	CP1VOUT2N – 0.3 V [5]	AVDD + 0.3 V
	JACKDET2 [4], JACKDET3 [4]	SUBGND – 0.3 V	MICVDD + 0.3 V
	GPSW*n*P, GPSW*n*N	SUBGND – 0.3 V	MICVDD + 0.3 V
Ground	AGND [6], DGND, CPGND, FLLGND	SUBGND – 0.3V	SUBGND + 0.3V
Operating temperature range	T_A	–40ºC	+85ºC
Operating junction temperature	T_J	–40ºC	+125ºC
Storage temperature after soldering	—	–65ºC	+150ºC

ESD-sensitive device. The CS42L92 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 to a common supply rail. 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 MICDET*n* and HPOUTFB*n* functions share common pins. The absolute maximum rating varies according to the applicable function of each pin. The HPOUTFB*n* ratings are applicable if any of the HP*n*_GND_SEL bits select the respective pin for HPOUT ground feedback.
4. If AVDD > MICVDD (e.g., if LDO2 is disabled), the maximum JACKDET2/JACKDET3 voltage is AVDD + 0.3 V.
5. CP1VOUT2N is an internal supply, generated by the CS42L92 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	DBVDD	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
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, FLLGND, 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 to a common supply rail. 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 DBVDD are present.
3. It is recommended to connect a 120-Ω resistor in series with the FLLVDD pin connection. Note that the minimum voltage limit applies at the supply end of the 120-Ω resistor in this case.
4. If the SLIMbus interface is enabled, the maximum DBVDD 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, CPGND, FLLGND, and SUBGND must not exceed 0.1 Ω.
9. If the DCVDD rise time exceeds 2 ms, $\overline{\text{RESET}}$ must be asserted 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—IN1xx, IN2xx

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 maximum input signal level is reduced by 6 dB if mid-power operation is selected (IN*n*_OSR = 100); the maximum signal level corresponds to –6 dBFS at the respective ADC outputs in this case.
- 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	11	—	kΩ
	Differential PGA input, All PGA gain settings	17	22	—	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—DMICDAT*n*

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

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

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

Table 3-8. Output Characteristics

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

Parameter			Minimum	Typical	Maximum	Units
Line/headphone/earpiece output driver (HPOUT*n*L, HPOUT*n*R)	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
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): DBVDD = CPVDD1 = AVDD = 1.8 V, DCVDD = FLLVDD = CPVDD2 = 1.2 V; MICVDD = 3.1 V (powered from internal LDO); 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 (HPOUT*n*L, HPOUT*n*R)	DC offset at Load	Single-ended mode	—	50	—	µV
		Differential (BTL) mode	—	75	—	µV
Analog input paths (IN1xx, IN2xx) to ADC (Differential Input Mode)	SNR (A-weighted), defined in Table 3-1	20 Hz to 20 kHz, 48 kHz sample rate	91	99	—	dB
		20 Hz to 8 kHz, 16 kHz sample rate	—	104	—	dB
	THD, defined in Table 3-1	–1 dBV input	—	–87	—	dB
	THD+N, defined in Table 3-1	–1 dBV input	—	–88	–79	dB
	Channel separation (L/R), defined in Table 3-1	100 Hz to 10 kHz	—	109	—	dB
	Input-referred noise floor	A-weighted, PGA gain = +20 dB	—	2.6	—	μV_{RMS}
	CMRR, defined in Table 3-1	PGA gain = +30 dB	—	83	—	dB
		PGA gain = 0 dB	—	72	—	dB
	PSRR (DBVDD, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	99	—	dB
		100 mV (peak-peak) 10 kHz	—	84	—	dB
	PSRR (DCVDD, FLLVDD, CPVDD2), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	100	—	dB
		100 mV (peak-peak) 10 kHz	—	82	—	dB
Analog input paths (IN1xx, IN2xx) to ADC (Single-Ended Input Mode)	SNR (A-weighted), defined in Table 3-1	20 Hz to 20 kHz, 48 kHz sample rate	87	98	—	dB
		20 Hz to 8 kHz, 16 kHz sample rate	—	108	—	dB
	THD, defined in Table 3-1	–7dB V input	—	–84	—	dB
	THD+N, defined in Table 3-1	–7dB V input	—	–83	–78	dB
	Channel separation (L/R), defined in Table 3-1	100 Hz to 10 kHz	—	107	—	dB
	Input-referred noise floor	A-weighted, PGA gain = +20 dB	—	4	—	μV_{RMS}
	PSRR (DBVDD, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	76	—	dB
		100 mV (peak-peak) 10 kHz	—	52	—	dB
	PSRR (DCVDD, FLLVDD, CPVDD2), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	96	—	dB
		100 mV (peak-peak) 10 kHz	—	87	—	dB

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

Test conditions (unless specified otherwise): DBVDD = CPVDD1 = AVDD = 1.8 V, DCVDD = FLLVDD = CPVDD2 = 1.2 V; MICVDD = 3.1 V (powered from internal LDO); 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 (IN1xx, IN2xx) to ADC (Differential Input, Mid Power Mode)	SNR, defined in Table 3-1	A-weighted	—	88	—	dB
	THD, defined in Table 3-1	–7 dBV input	—	–81	—	dB
	THD+N, defined in Table 3-1	–7 dBV input	—	–80	—	dB
	Channel separation (L/R), defined in Table 3-1	100 Hz to 10 kHz	—	96	—	dB
	Input-referred noise floor	A-weighted, PGA gain = +20 dB	—	4.79	—	μV_{RMS}
	CMRR, defined in Table 3-1	PGA gain = +30 dB PGA gain = 0 dB	— —	73 81	— —	dB dB
	PSRR (DBVDD, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	93 75	— —	dB dB
	PSRR (DCVDD, FLLVDD, CPVDD2), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	97 81	— —	dB dB
DAC to line output (HPOUT1*x*, HPOUT2*x*; Load = 10 kΩ, 50 pF)	Full-scale output signal level	0 dBFS input	— —	1 0	— —	V_{RMS} dBV
	SNR, defined in Table 3-1	A-weighted, output signal = 1 V_{RMS}	—	125	—	dB
	Dynamic range, defined in Table 3-1	A-weighted, –60 dBFS input	108	117	—	dB
	THD+N, defined in Table 3-1	0 dBFS input	—	–100	–90	dB
	Channel separation (L/R), defined in Table 3-1	100 Hz to 10 kHz	—	95	—	dB
	Output noise floor	A-weighted	—	0.7	—	μV_{RMS}
	PSRR (DBVDD, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	105 81	— —	dB dB
	PSRR (DCVDD, FLLVDD, CPVDD2), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	108 81	— —	dB dB
DAC to headphone output (HPOUT1*x*, HPOUT2*x*; R_L = 32 Ω)	Maximum output power	0.1% THD+N	—	33	—	mW
	SNR, defined in Table 3-1	A-weighted, output signal = 1 V_{RMS}	—	125	—	dB
	Dynamic range, defined in Table 3-1	A-weighted, –60 dBFS input	108	117	—	dB
	THD+N, defined in Table 3-1	P_O = 20 mW	—	–100	–90	dB
	THD+N, defined in Table 3-1	P_O = 2 mW	—	–96	—	dB
	Channel separation (L/R), defined in Table 3-1	100 Hz to 10 kHz	—	105	—	dB
	Output noise floor	A-weighted	—	0.6	—	μV_{RMS}
	PSRR (DBVDD, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	126 103	— —	dB dB
	PSRR (DCVDD, FLLVDD, CPVDD2), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	128 105	— —	dB dB
DAC to headphone output (HPOUT1*x*, HPOUT2*x*; R_L = 16 Ω)	Maximum output power	0.1% THD+N	—	46	—	mW
	SNR, defined in Table 3-1	A-weighted, output signal = 1 V_{RMS}	—	125	—	dB
	Dynamic range, defined in Table 3-1	A-weighted, –60 dBFS input	108	117	—	dB
	THD+N, defined in Table 3-1	P_O = 20 mW	—	–97	–90	dB
	THD+N, defined in Table 3-1	P_O = 2 mW	—	–95	—	dB
	Channel separation (L/R), defined in Table 3-1	100 Hz to 10 kHz	—	97	—	dB
	Output noise floor	A-weighted	—	0.6	—	μV_{RMS}
	PSRR (DBVDD, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	127 107	— —	dB dB
	PSRR (DCVDD, FLLVDD, CPVDD2), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	127 108	— —	dB dB
DAC to headphone output (HPOUT1L+HPOUT1R, HPOUT2L+HPOUT2R; Stereo differential output, R_L = 32 Ω BTL)	Maximum output power	0.1% THD+N	—	109	—	mW
	SNR, defined in Table 3-1	A-weighted, output signal = 2 V_{RMS}	—	127	—	dB
	Dynamic range, defined in Table 3-1	A-weighted, –60 dBFS input	—	115	—	dB
	THD+N, defined in Table 3-1	P_O = 75 mW	—	–100	—	dB
	THD+N, defined in Table 3-1	P_O = 5 mW	—	–97	—	dB
	Output noise floor	A-weighted	—	0.36	—	μV_{RMS}
	PSRR (DBVDD, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	120 90	— —	dB dB
	PSRR (DCVDD, FLLVDD, CPVDD2), 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): DBVDD = CPVDD1 = AVDD = 1.8 V, DCVDD = FLLVDD = CPVDD2 = 1.2 V; MICVDD = 3.1 V (powered from internal LDO); 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 line output (HPOUT3*x*, HPOUT4*x*; Load = 10 kΩ, 50 pF)	Full-scale output signal level	0 dBFS input	— —	1 0	— —	V_{RMS} dBV
	SNR, defined in Table 3-1	A-weighted, output signal = 1 V_{RMS}	115	125	—	dB
	Dynamic range, defined in Table 3-1	A-weighted, –60 dBFS input	107	115	—	dB
	THD, defined in Table 3-1	0 dBFS input	—	–92	—	dB
	THD+N, defined in Table 3-1	0 dBFS input	—	–91	–85	dB
	Channel separation (L/R), defined in Table 3-1	100 Hz to 10 kHz	—	90	—	dB
	Output noise floor	A-weighted	—	0.63	—	μV_{RMS}
	PSRR (DBVDD, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	110 78	— —	dB dB
	PSRR (DCVDD, FLLVDD, CPVDD2), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	99 69	— —	dB dB
DAC to headphone output (HPOUT3*x*, HPOUT4*x*; R_L = 32 Ω)	Maximum output power	0.1% THD+N	—	33	—	mW
	SNR, defined in Table 3-1	A-weighted, output signal = 1 V_{RMS}	—	124	—	dB
	Dynamic range, defined in Table 3-1	A-weighted, –60 dBFS input	107	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	—	–90	–85	dB
	THD, defined in Table 3-1	P_O = 2 mW	—	–94	—	dB
	THD+N, defined in Table 3-1	P_O = 2 mW	—	–92	—	dB
	Channel separation (L/R), defined in Table 3-1	100 Hz to 10 kHz	—	102	—	dB
	Output noise floor	A-weighted	—	0.63	—	μV_{RMS}
	PSRR (DBVDD, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	127 100	— —	dB dB
	PSRR (DCVDD, FLLVDD, CPVDD2), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	120 95	— —	dB dB
DAC to headphone output (HPOUT3*x*, HPOUT4*x*; R_L = 16 Ω)	Maximum output power	0.1% THD+N	—	46	—	mW
	SNR, defined in Table 3-1	A-weighted, output signal = 1 V_{RMS}	—	124	—	dB
	Dynamic range, defined in Table 3-1	A-weighted, –60 dBFS input	107	115	—	dB
	THD, defined in Table 3-1	P_O = 20 mW	—	–94	—	dB
	THD+N, defined in Table 3-1	P_O = 20 mW	—	–89	–80	dB
	THD, defined in Table 3-1	P_O = 2 mW	—	–91	—	dB
	THD+N, defined in Table 3-1	P_O = 2 mW	—	–89	—	dB
	Channel separation (L/R), defined in Table 3-1	100 Hz to 10 kHz	—	100	—	dB
	Output noise floor	A-weighted	—	0.63	—	μV_{RMS}
	PSRR (DBVDD, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	128 104	— —	dB dB
	PSRR (DCVDD, FLLVDD, CPVDD2), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	125 97	— —	dB dB
DAC to earpiece output (HPOUT3L+HPOUT3R, HPOUT4L+HPOUT4R, Mono Mode, R_L = 32 Ω BTL)	Maximum output power	0.1% THD+N	—	115	—	mW
	SNR, defined in Table 3-1	A-weighted, output signal = 2 V_{RMS}	—	129	—	dB
	Dynamic range, defined in Table 3-1	A-weighted, –60 dBFS input	110	120	—	dB
	THD, defined in Table 3-1	P_O = 75 mW	—	–99	—	dB
	THD+N, defined in Table 3-1	P_O = 75 mW	—	–97	—	dB
	THD, defined in Table 3-1	P_O = 5 mW	—	–98	—	dB
	THD+N, defined in Table 3-1	P_O = 5 mW	—	–94	—	dB
	Output noise floor	A-weighted	—	0.36	—	μV_{RMS}
	PSRR (DBVDD, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	127 106	— —	dB dB
	PSRR (DCVDD, FLLVDD, CPVDD2), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	125 101	— —	dB dB

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

Test conditions (unless specified otherwise): DBVDD = CPVDD1 = AVDD = 1.8 V, DCVDD = FLLVDD = CPVDD2 = 1.2 V; MICVDD = 3.1 V (powered from internal LDO); 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 (HPOUT3L+HPOUT3R, HPOUT4L+HPOUT4R, Mono Mode, R_L = 16 Ω BTL)	Maximum output power	0.1% THD+N	—	138	—	mW
	SNR, defined in Table 3-1	A-weighted, output signal = 2 V_{RMS}	—	128	—	dB
	Dynamic range, defined in Table 3-1	A-weighted, –60 dBFS input	110	120	—	dB
	THD, defined in Table 3-1	P_O = 75 mW	—	–97	—	dB
	THD+N, defined in Table 3-1	P_O = 75 mW	—	–95	—	dB
	THD, defined in Table 3-1	P_O = 5 mW	—	–96	—	dB
	THD+N, defined in Table 3-1	P_O = 5 mW	—	–94	—	dB
	Output noise floor	A-weighted	—	0.4	—	μV_{RMS}
	PSRR (DBVDD, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	125 106	— —	dB dB
	PSRR (DCVDD, FLLVDD, CPVDD2), defined in Table 3-1	100 mV (peak-peak) 217 Hz 100 mV (peak-peak) 10 kHz	— —	125 101	— —	dB dB

Table 3-10. Digital Input/Output

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

Parameter			Minimum	Typical	Maximum	Units
Digital I/O (except DMICDAT*n* and DMICCLK*n*) [1,2]	Input HIGH level	V_{DBVDD} = 1.71–1.98 V V_{DBVDD} = 2.5 V ±10% V_{DBVDD} = 3.3 V ±10%	0.75 × DBVDD 0.8 × DBVDD 0.7 × DBVDD	— — —	— — —	V V V
	Input LOW level	V_{DBVDD} = 1.71–1.98 V V_{DBVDD} = 2.5 V ±10% V_{DBVDD} = 3.3 V ±10%	— — —	— — —	0.3 × DBVDD 0.25 × DBVDD 0.2 × DBVDD	V V V
	Output HIGH level (I_{OH} = 1 mA)	V_{DBVDD} = 1.71–1.98 V V_{DBVDD} = 2.5 V ±10% V_{DBVDD} = 3.3 V ±10%	0.75 × DBVDD 0.65 × DBVDD 0.7 × DBVDD	— — —	— — —	V V V
	Output LOW level (I_{OL} = 1mA)	V_{DBVDDn} = 1.71–1.98 V V_{DBVDDn} = 2.5 V ±10% V_{DBVDDn} = 3.3 V ±10%	— — —	— — —	0.25 × DBVDD 0.3 × DBVDD 0.15 × DBVDD	V V V
	Input capacitance		—	—	5	pF
	Input leakage		–10	—	10	µA
	Pull-up/pull-down resistance (where applicable)		35	—	55	kΩ
DMIC I/O (DMICDAT*n* and DMICCLK*n*) [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 DBVDD.
2. Note that digital input pins should not be left unconnected or floating.
3. DMICDAT*n* and DMICCLK*n* are referenced to a selectable supply, V_{SUP}, according to the IN*n*_DMIC_SUP fields.

Table 3-11. Miscellaneous Characteristics

Test conditions (unless specified otherwise): DBVDD = CPVDD1 = AVDD = 1.8 V, DCVDD = FLLVDD = CPVDD2 = 1.2 V; MICVDD = 3.1 V (powered from internal LDO); 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 (MICBIAS*nx*) [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	—	50	—	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, CPVDD1, AVDD), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	105	—	dB
		100 mV (peak-peak) 10 kHz	—	95	—	dB
	PSRR (DCVDD, FLLVDD, CPVDD2), defined in Table 3-1	100 mV (peak-peak) 217 Hz	—	99	—	dB
		100 mV (peak-peak) 10 kHz	—	92	—	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	—	40	—	Ω
		Switch open	—	100	—	MΩ
External Accessory Detect	Headphone detection load impedance range: Detection via HPDET1 (HPD_SENSE_SEL = 0100) or HPDET2 (HPD_SENSE_SEL = 0101)	HPD_IMPEDANCE_RANGE = 00	4	—	30	Ω
		HPD_IMPEDANCE_RANGE = 01	8	—	100	Ω
		HPD_IMPEDANCE_RANGE = 10	100	—	1000	Ω
		HPD_IMPEDANCE_RANGE = 11	1000	—	10000	Ω
	Headphone detection load impedance range: Detection via MICDET*n* or JACKDET*n* pins		400	—	6000	Ω
	Headphone detection accuracy: (HPD_DACVAL, HPDET*n* pin)	HPD_IMPEDANCE_RANGE = 01 or 10	–5	—	+5	%
		HPD_IMPEDANCE_RANGE = 00 or 11	–10	—	+10	%
	Headphone detection accuracy (HPD_LVL, MICDET*n* or JACKDET*n* pin)		–20	—	+20	%
	Microphone impedance detection range: (MICD*n*_ADC_MODE = 0, 2.2 kΩ ±2% MICBIAS resistor. [5])	for MICD*n*_LVL[0] = 1	0	—	70	Ω
		for MICD*n*_LVL[1] = 1	110	—	180	Ω
		for MICD*n*_LVL[2] = 1	210	—	290	Ω
		for MICD*n*_LVL[3] = 1	360	—	680	Ω
		for MICD*n*_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 JACKDET2/3, Jack insertion	—	0.27	—	V
		Detection on JACKDET2/3, Jack removal	—	0.9	—	V
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	—	1.0	2.5	ms
Frequency-Locked Loop (FLL1, FLL2)	Output frequency		45	—	50	MHz
	Lock Time	F_{REF} = 32 kHz, F_{FLL} = 49.152 MHz	—	5	—	ms
		F_{REF} = 12 MHz, F_{FLL} = 49.152 MHz	—	1	—	ms
$\overline{\text{RESET}}$ pin input	$\overline{\text{RESET}}$ input pulse width [6]		1	—	—	µs

1. No capacitor on MICBIAS*nx*. 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 GPSW*n*N pin voltage must not exceed GPSW*n*P + 0.3 V. See Table 3-2 for voltage limits applicable to the GPSW*n*P and GPSW*n*N 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	—	0.70	V
DBVDD Reset threshold	V_{DBVDD} rising	V_{DBVDD}	—	—	1.66	V
	V_{DBVDD} 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, MCLK3) [1]	MCLK cycle time	MCLK as input to FLL, FLL*n*_REFCLK_DIV = 00	74	—	—	ns
		MCLK as input to FLL, FLL*n*_REFCLK_DIV = 01	37	—	—	ns
		MCLK as input to FLL, FLL*n*_REFCLK_DIV = 10	18	—	—	ns
		MCLK as input to FLL, FLL*n*_REFCLK_DIV = 11	12.5	—	—	ns
		MCLK as direct SYSCLK or ASYNCCLK source	40	—	—	ns
	MCLK duty cycle	MCLK as input to FLL	80:20	—	20:80	%
		MCLK as direct SYSCLK or ASYNCCLK source	60:40	—	40:60	%
Frequency-locked loop (FLL1, FLL2)	FLL input frequency	FLL*n*_REFCLK_DIV = 00	0.032	—	13	MHz
		FLL*n*_REFCLK_DIV = 01	0.064	—	26	MHz
		FLL*n*_REFCLK_DIV = 11	0.128	—	52	MHz
		FLL*n*_REFCLK_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
	ASYNCCLK frequency	ASYNC_CLK_FREQ = 000	–1%	6.144	+1%	MHz
			–1%	5.6448	+1%	MHz
		ASYNC_CLK_FREQ = 001	–1%	12.288	+1%	MHz
			–1%	11.2896	+1%	MHz
		ASYNC_CLK_FREQ = 010	–1%	24.576	+1%	MHz
			–1%	22.5792	+1%	MHz
		ASYNC_CLK_FREQ = 011	–1%	49.152	+1%	MHz
			–1%	45.1584	+1%	MHz
		ASYNC_CLK_FREQ = 100	–1%	98.304	+1%	MHz
			–1%	90.3168	+1%	MHz
	DSPCLK frequency		5	—	150	MHz

1. If MCLK1, MCLK2, or MCLK3 is selected as a source for SYSCLK or ASYNCCLK (either directly or via one of the FLLs), the frequency must be within 1% of the applicable SYSCLK_FREQ or ASYNC_CLK_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 DMIC interfaces is selectable, using the IN*n*_DMIC_SUP fields—each interface may be referenced to MICVDD, MICBIAS1, or MICBIAS2.

1. DMIC interface timing

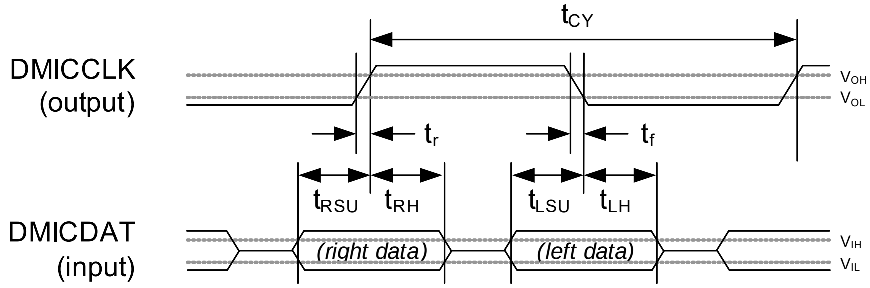
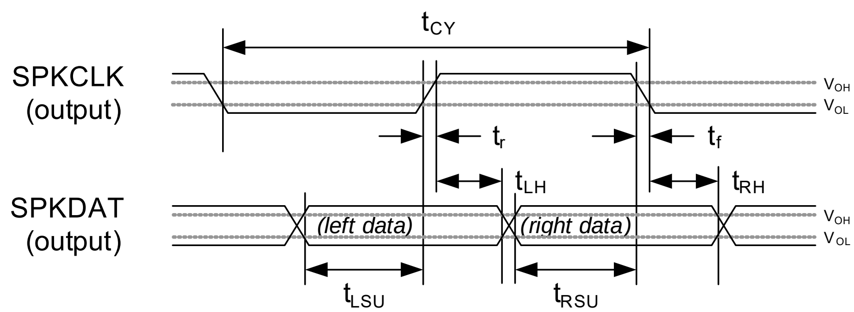


Table 3-15. Digital Speaker (PDM) Interface Timing

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

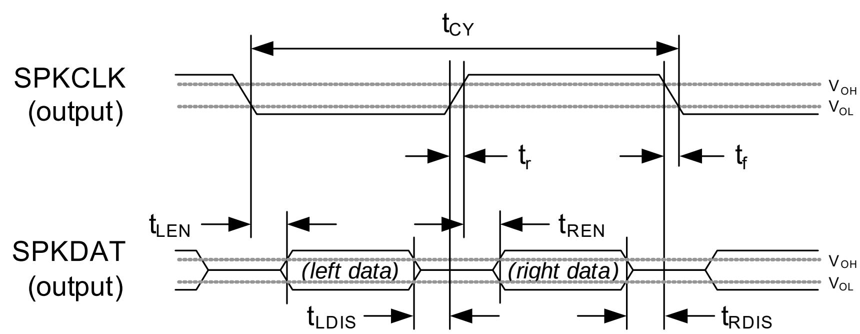


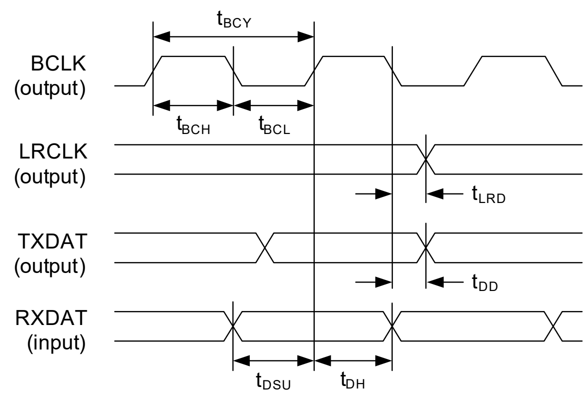
Table 3-16. Digital Audio Interface—Master Mode

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

Notes: 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.

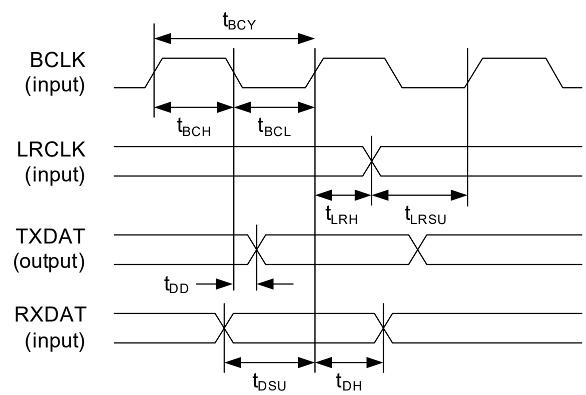
Table 3-17. Digital Audio Interface—Slave Mode

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 or ASYNCCLK source	t_{BCH}	16	—	—	ns
	All other conditions	t_{BCH}	14	—	—	ns
AIF*n*BCLK pulse width low	BCLK as direct SYSCLK or ASYNCCLK 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 or ASYNCCLK (either directly or via one of the FLLs), the frequency must be within 1% of the applicable SYSCLK_FREQ or ASYNC_CLK_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.

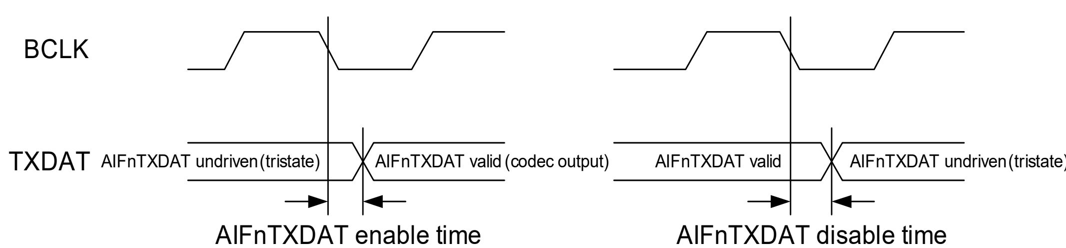


Table 3-19. Control Interface Timing—Two-Wire (I²C) Mode

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 pulse-width low	t_1	160	—	—	ns
SCLK pulse-width high	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 > 1.7 MHz	t_6	—	—	80	ns
SCLK frequency > 1 MHz	t_6	—	—	160	ns
SCLK frequency ≤ 1 MHz	t_6	—	—	2000	ns
SDA, SCLK fall time (90%–10%) SCLK frequency > 1.7 MHz	t_7	—	—	60	ns
SCLK frequency > 1 MHz	t_7	—	—	160	ns
SCLK frequency ≤ 1 MHz	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

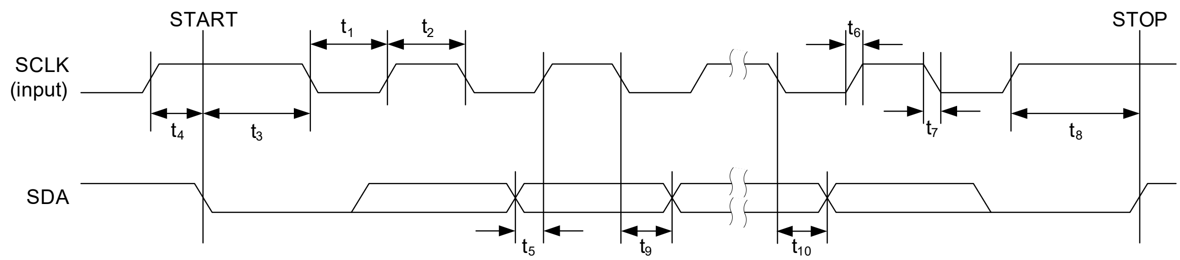
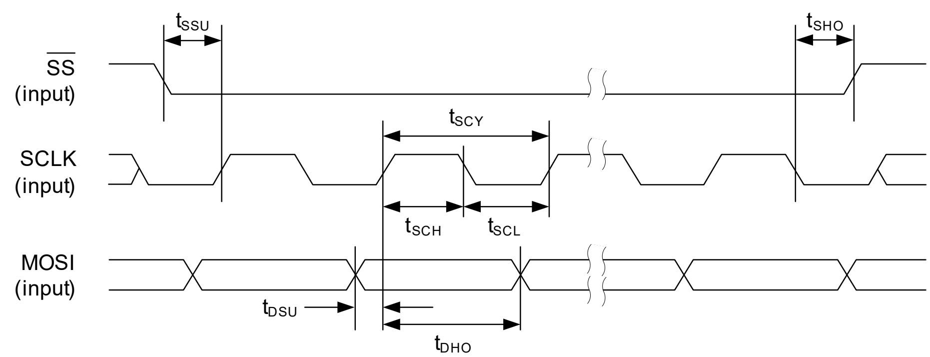


Table 3-20. Control Interface Timing—Four-Wire (SPI) Mode

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}	38.4	—	—	ns
	SYSCLK_ENA = 1, SYSCLK_FREQ = 000	t_{SCY}	76.8	—	—	ns
	SYSCLK_ENA = 1, SYSCLK_FREQ > 000	t_{SCY}	38.4	—	—	ns
SCLK pulse-width low		t_{SCL}	15.3	—	—	ns
SCLK pulse-width high		t_{SCH}	15.3	—	—	ns
MOSI to SCLK set-up time		t_{DSU}	1.5	—	—	ns
MOSI to SCLK hold time		t_{DHO}	1.7	—	—	ns
SCLK falling edge to MISO transition	SCLK slew (90%–10%) = 5 ns, C_{LOAD} (MISO) = 25 pF	t_{DL}	0	—	12.6	ns

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



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

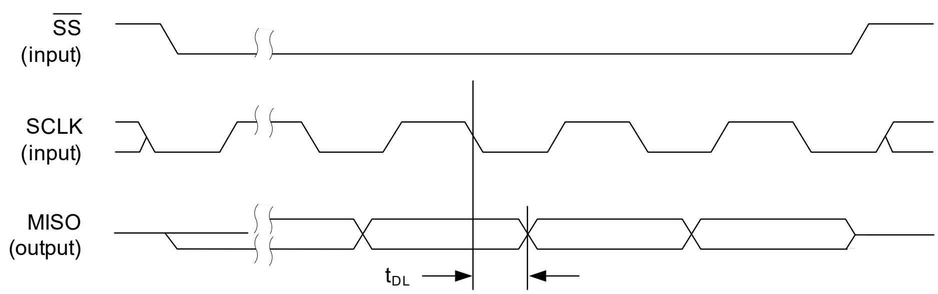


Table 3-21. SLIMbus Interface Timing

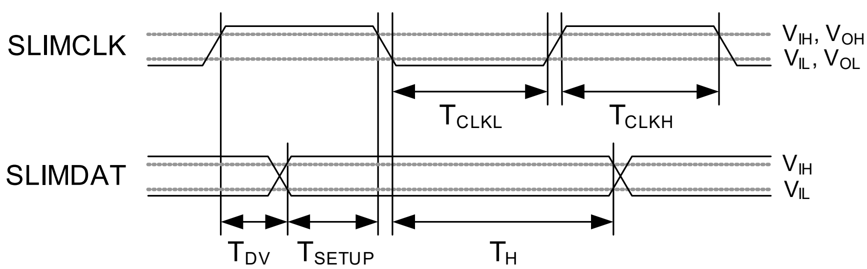
The following timing information is valid across the full range of recommended operating conditions.

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_{DBVDD}	—	0.22 x V_{DBVDD}	V/ns
		C_{LOAD} = 70 pF, SLIMCLK_DRV_STR = 0	SR_{CLK}	0.02 x V_{DBVDD}	—	0.05 x V_{DBVDD}	V/ns
		C_{LOAD} = 70 pF, SLIMCLK_DRV_STR = 1	SR_{CLK}	0.04 x V_{DBVDD}	—	0.11 x V_{DBVDD}	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, DBVDD = 1.71 V	T_{DV}	—	4.7	8.1	ns
		C_{LOAD} = 15 pF, SLIMDAT_DRV_STR = 1, DBVDD = 1.71 V	T_{DV}	—	4.3	7.3	ns
		C_{LOAD} = 30 pF, SLIMDAT_DRV_STR = 0, DBVDD = 1.71 V	T_{DV}	—	6.8	11.8	ns
		C_{LOAD} = 30 pF, SLIMDAT_DRV_STR = 1, DBVDD = 1.71 V	T_{DV}	—	5.8	10.0	ns
		C_{LOAD} = 50 pF, SLIMDAT_DRV_STR = 0, DBVDD = 1.71 V	T_{DV}	—	9.6	16.6	ns
		C_{LOAD} = 50 pF, SLIMDAT_DRV_STR = 1, DBVDD = 1.71 V	T_{DV}	—	7.9	13.7	ns
		C_{LOAD} = 70 pF, SLIMDAT_DRV_STR = 0, DBVDD = 1.71 V	T_{DV}	—	12.4	21.5	ns
		C_{LOAD} = 70 pF, SLIMDAT_DRV_STR = 1, DBVDD = 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_{DBVDD}	V/ns
		C_{LOAD} = 30 pF, SLIMDAT_DRV_STR = 0	SR_{DATA}	—	—	0.35 x V_{DBVDD}	V/ns
		C_{LOAD} = 30pF, SLIMDAT_DRV_STR = 1	SR_{DATA}	—	—	0.46 x V_{DBVDD}	V/ns
		C_{LOAD} = 70pF, SLIMDAT_DRV_STR = 0	SR_{DATA}	—	—	0.16 x V_{DBVDD}	V/ns
		C_{LOAD} = 70pF, SLIMCLK_DRV_STR = 1	SR_{DATA}	—	—	0.21 x V_{DBVDD}	V/ns
Other parameters	Driver disable time		T_{DD}	—	—	6	ns
	Bus holder output impedance	0.1 x V_{DBVDD} < V < 0.9 x V_{DBVDD}	R_{DATAS}	18	—	50	kΩ

Notes:

- The signal timing information describes the timing requirements of the SLIMbus interface as a whole, not just the CS42L92 device.
- T_{DV} is the propagation delay from the rising SLIMCLK edge (at CS42L92 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 CS42L92), relative to the falling SLIMCLK edge (at CS42L92).
- T_H is the hold time for SLIMDAT input (at CS42L92) relative to the falling SLIMCLK edge (at CS42L92).
- 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

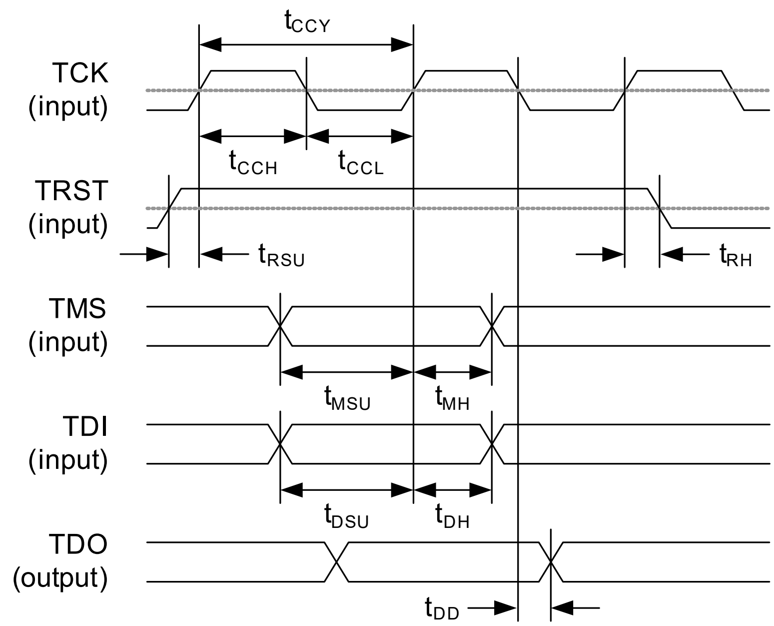


Table 3-23. Typical Power Consumption

Test conditions (unless specified otherwise): DBVDD = CPVDD1 = AVDD = 1.8 V, DCVDD = FLLVDD = CPVDD2 = 1.2 V; MICVDD = Off (CP2 and LDO2 disabled); T_A = +25ºC; 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)	P_{TOT} (mW)
Headphone playback	AIF1 to DAC to HPOUT1 (stereo), 32-Ω load.	Quiescent 1-kHz sine wave, P_O = 10 mW	1.8 37.8	0.8 1.9	3.6 48.78
Earpiece playback	AIF1 to DAC to HPOUT1 (mono), 32-Ω load (BTL).	Quiescent 1-kHz sine wave, P_O = 30 mW	1.2 61.7	0.85 1.7	2.97 77.1
Stereo line record	Analog line to ADC to AIF1, MICVDD = 1.8V (CP2 and LDO2 bypass enabled).	1-kHz sine wave, -1 dBFS output	1.1	2.2	5.28
Sleep Mode	Accessory detect enabled (JD1_ENA = 1)		0.000	0.010	0.018

Table 3-24. Typical Signal Latency

Test conditions (unless specified otherwise): DBVDD = CPVDD1 = AVDD = 1.8 V, DCVDD = FLLVDD = CPVDD2 = 1.2 V; MICVDD = Off (CP2 and LDO2 disabled); T_A = +25ºC; 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 (AIF*n*) to analog output (HPOUT*n*)	192 kHz input, 192 kHz output, Synchronous	237
	96 kHz input, 96 kHz output, Synchronous	269
	48 kHz input, 48 kHz output, Synchronous	358
	44.1 kHz input, 44.1 kHz output, Synchronous	374
	16 kHz input, 16 kHz output, Synchronous	629
	8 kHz input, 8 kHz output, Synchronous	1334
	8 kHz input, 48 kHz output, Isochronous [1]	1939
	16 kHz input, 48 kHz output, Isochronous [1]	993
	8 kHz input, 44.1 kHz output, Asynchronous [2]	1880
	16 kHz input, 44.1 kHz output, Asynchronous [2]	1084
ADC to AIF path—analog input (IN*n*) to digital output (AIF*n*) [3]	192 kHz input, 192 kHz output, Synchronous	50
	96 kHz input, 96 kHz output, Synchronous	96
	48 kHz input, 48 kHz output, Synchronous	193
	44.1 kHz input, 44.1 kHz output, Synchronous	212
	16 kHz input, 16 kHz output, Synchronous	558
	8 kHz input, 8 kHz output, Synchronous	1170
	8 kHz input, 48 kHz output, Isochronous [1]	1696
	16 kHz input, 48 kHz output, Isochronous [1]	861
	44.1 kHz input, 8 kHz output, Asynchronous [2]	1364
	44.1 kHz input, 16 kHz output, Asynchronous [2]	841

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

4 Functional Description

The CS42L92 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. The digital audio interfaces and hi-fi DACs support 32-bit playback, offering audiophile-quality playback at sample rates up to 384 kHz. Native audio playback is possible, concurrent with voice and ultrasonic input paths. Multiple analog outputs support a wide variety of the headphone/accessory configurations, including balanced stereo headphone loads.

4.1 Overview

The CS42L92 block diagram is shown in Fig. 4-1.

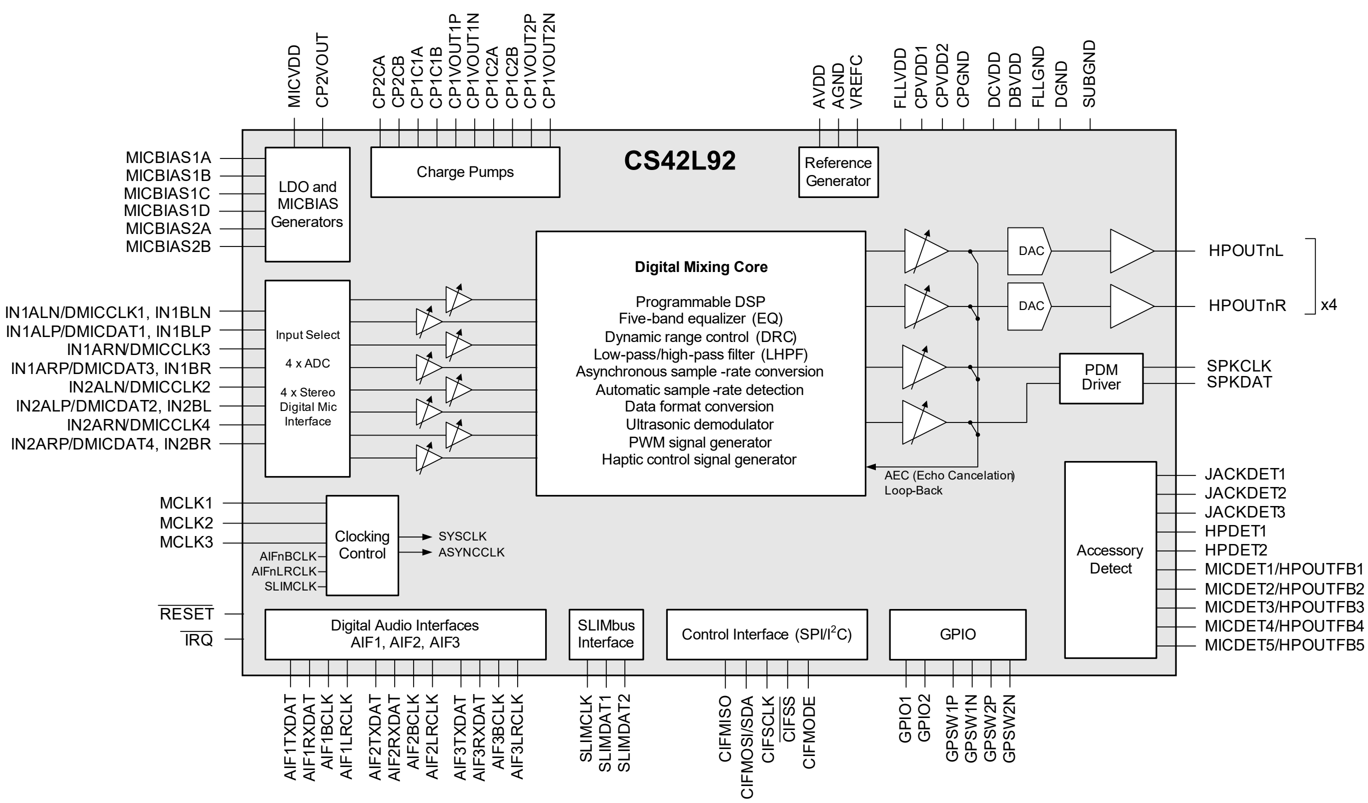


Figure 4-1. CS42L92 Block Diagram

The CS42L92 combines fixed-function signal-processing blocks with a fully-flexible, all-digital audio mixing and routing engine for extensive use-case flexibility. The signal-processing blocks include high-/low-pass filters, EQ, dynamic range control, and sample-rate converters.

The CS42L92 provides multiple digital audio interfaces, including SLIMbus, to provide independent and fully asynchronous 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. Two frequency-locked loop (FLL) circuits provide additional flexibility for system clocking, including low-power always-on operation. Seamless switching between clock sources is supported, and free-running modes are also available.

Unused circuitry can be disabled under software control to save power; low leakage currents enable extended standby/off time in portable battery-powered applications. The CS42L92 always-on circuitry can be used in conjunction with the applications 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. The integrated DSP provides a general-purpose signal processing capability; this is supported by general-purpose timer and event-logger functions.

4.1.1 Hi-Fi Audio Codec

The CS42L92 is a high-performance, low-power audio codec that uses a simple analog architecture. Four ADCs are incorporated, with multiplexers to support up to eight analog inputs. Six DACs are incorporated, with two being switchable between two separate headphone output paths.

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 CS42L92 input channels support up to eight analog inputs or up to eight digital inputs, multiplexed into four stereo input signal paths. In differential mode, the analog input path SNR is 104 dB (16-kHz sample rate, i.e., wideband voice mode). The input paths can be configured for low-power operation, ideal for analog or digital microphone input in always-on applications. Ultrasonic signal demodulation functions are provided, supporting a variety of presence-detection applications.

The analog outputs comprise four 33-mW (125 dB SNR) stereo headphone amplifiers with ground-referenced output. The output drivers are designed to support many different system architectures and are compatible with line or headphone loads in single-ended or differential (BTL) configurations. Headphone outputs HPOUT1 and HPOUT2 offer 127 dB SNR and –100 dB THD+N performance with a stereo differential headphone load. Outputs HPOUT3 and HPOUT4 are multiplexed, with the respective DACs and signal paths common to both pairs of output drivers.

Each output path supports independent mixing, equalization, filtering, gain controls. This allows each signal path to be individually tailored for the load characteristics. Selectable hi-fi filters support audiophile playback modes at sample rates up to 384 kHz. All outputs have integrated pop and click suppression features.

The headphone 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 CS42L92 is cost optimized for a wide range of mobile phone applications. External speaker amplifiers can be connected using the stereo PDM outputs; this can ease layout and electromagnetic compatibility by avoiding the need to run high-power speaker outputs over a long distance and across interconnects.

4.1.2 Digital Audio Core

The CS42L92 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. Soft mute and unmute control ensures smooth transitions between use cases without interrupting existing audio streams elsewhere.

Highly flexible digital mixing, including mixing between audio interfaces, is possible. The CS42L92 performs multichannel full-duplex asynchronous 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. Each AIF supports eight input/output channels. Bidirectional operation at sample rates up to 384 kHz is supported. Data words of up to 32 bits can be routed through AIF1 and AIF3. Data-format conversion (DFC) functions are available to support different interface standards on the input and output signal paths.

Eight digital PDM input channels are available (four 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 auxiliary PDM interface can be used to provide an audio path between an analog microphone connected to the CS42L92 and a digital input to an external audio processor. The auxiliary PDM interface operates in master or slave modes, and is configured on GPIO pins.

The CS42L92 features a SLIMbus interface, compliant with the MIPI® SLIMbus specification, providing eight 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 CS42L92 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 is supported by a configurable SPI/I2C control interface. The interface supports SPI slave operation up to 26 MHz or I2C slave operation up to 3.4 MHz. Full access to the register map is also provided via the SLIMbus port.

4.1.4 Other Features

The CS42L92 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.

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 CS42L92 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 CS42L92 provides two dedicated GPIO pins; a further 14 GPIOs are multiplexed with other functions. Comprehensive interrupt functionality is also provided for monitoring internal and external event conditions.

The integrated DSP provides a general-purpose signal processing capability; this is supported by general-purpose timer and event-logger functions.

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 using the PDM digital output path.

A smart accessory interface is included, supporting a wide variety of system configurations. Jack detection, accessory sensing, and impedance measurement is provided, for external headset and push-button detection. Dual headphone connections (e.g., 3.5 mm and USB-C) can be detected simultaneously. 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, MCLK2, or MCLK3 input pins. Alternatively, the SLIMbus interface, or the audio interfaces (configured in Slave Mode), can be used to provide a clock reference. The CS42L92 also provides two integrated FLL circuits for clock frequency conversion and stability. The flexible clocking architecture supports low-power always-on operation, with reference frequencies down to 32 kHz. Seamless switching between clock sources is supported; free-running FLL modes are also available.

The CS42L92 is powered from 1.8- and 1.2-V external supplies. 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. Power consumption is optimized across a wide variety of voice and multimedia use cases.

4.2 Input Signal Path

The CS42L92 provides flexible input channels, supporting up to eight analog inputs or up to eight digital inputs. Selectable combinations of analog (mic or line) and digital inputs are multiplexed into four stereo input signal paths. Input paths IN1 and IN2 support analog and digital inputs; input paths IN3 and IN4 support digital inputs only.

The analog input paths support single-ended and differential configurations, programmable gain control, and are digitized using a high performance sigma-delta ADC. The analog input paths can be configured for low-power operation, ideal for always-on applications. Analog inputs can be configured as input to the auxiliary PDM interface, providing an audio path between an analog microphone connected to the CS42L92 and a digital input to an external audio processor.

The digital input paths interface directly with external digital microphones; a separate microphone interface clock is provided for four separate stereo pairs of digital 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 six separate reference/supply outputs to be independently controlled.

Digital volume control is available on all inputs (analog and digital), with programmable ramp control for smooth, glitch-free operation. A configurable signal-detect function is available on each input signal path. Ultrasonic signal demodulation functions are provided on the input signal paths, supporting a variety of presence-detection applications.

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

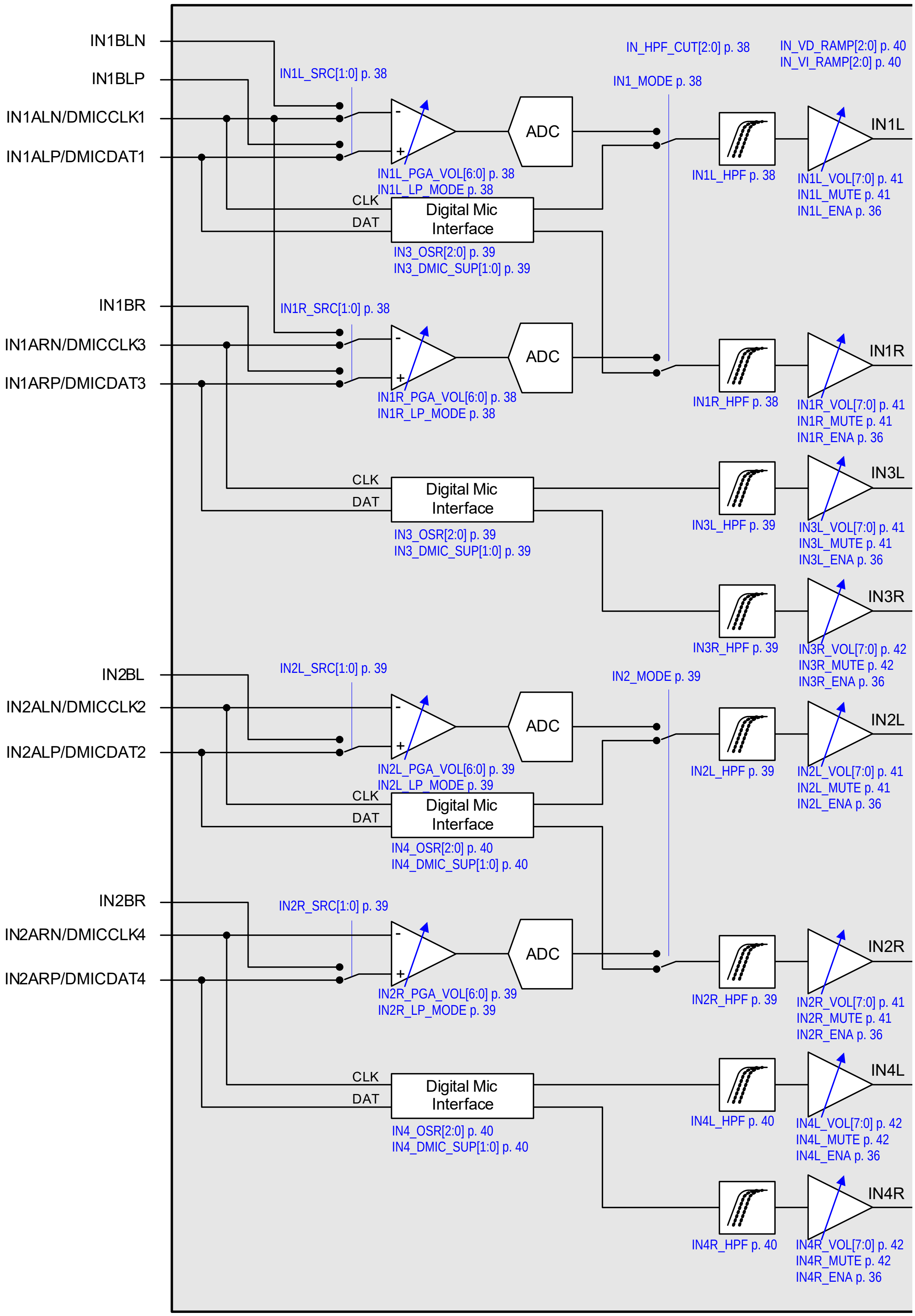


Figure 4-2. Input Signal Paths

4.2.1 Analog Microphone Input

Up to eight analog microphones can be connected to the CS42L92, either in single-ended or differential configuration. The input configuration and pin selection is controlled using IN*nx*_SRC, as described in Section 4.2.6.

The CS42L92 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 IN1AL*x*, IN1BL*x*, or IN1BR 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. 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 and the inverted (or noisy ground) signal is connected to the inverting input pins.

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.

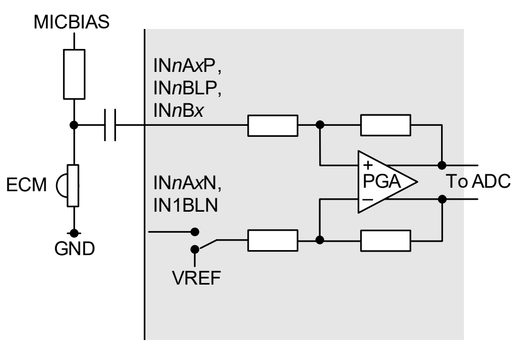


Figure 4-3. Single-Ended ECM Input

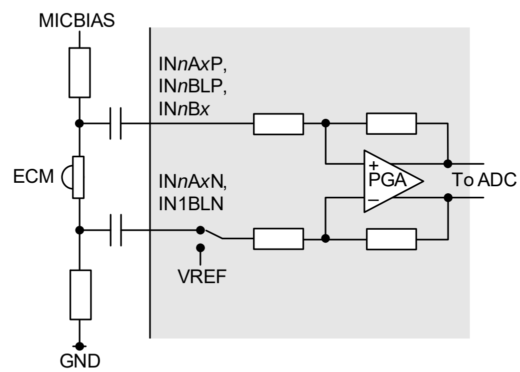


Figure 4-4. Differential ECM Input

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. The typical connections for pseudodifferential input are shown in Fig. 4-5.

Note that pseudodifferential input is the recommended configuration if the accessory-detection functions are used on this input path. The IN*nx*_SRC field settings are the same for pseudodifferential connection as for differential.

The IN1ALN pin can be used as the inverting input connection for the IN1L and IN1R paths concurrently. This allows two microphones to be supported, in pseudodifferential configuration, while minimizing the number of pin connections required—see Fig. 4-6.

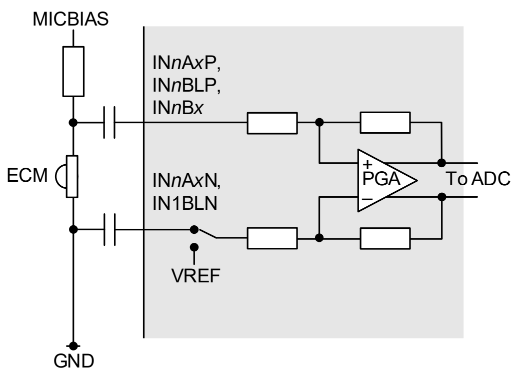


Figure 4-5. Pseudodifferential ECM Input

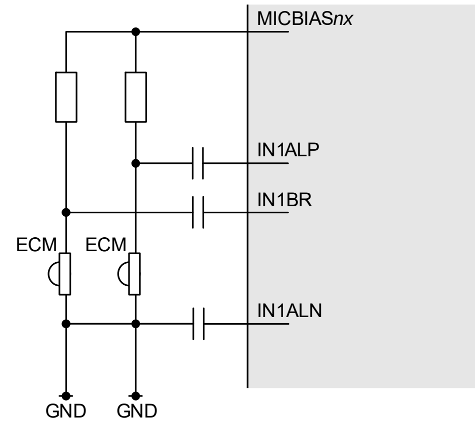


Figure 4-6. Pseudodifferential IN1L/IN1R Input

Analog MEMS microphones can be connected to the CS42L92 in a similar manner to the ECM configurations. Typical configurations are shown in Fig. 4-7 and Fig. 4-8. In this configuration, the integrated MICBIAS generators provide a low-noise power supply for the microphones.

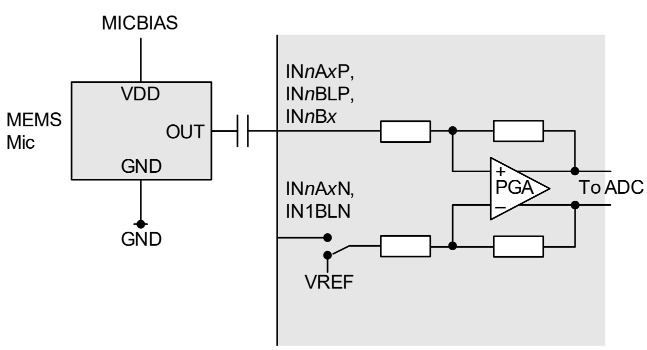


Figure 4-7. Single-Ended MEMS Input

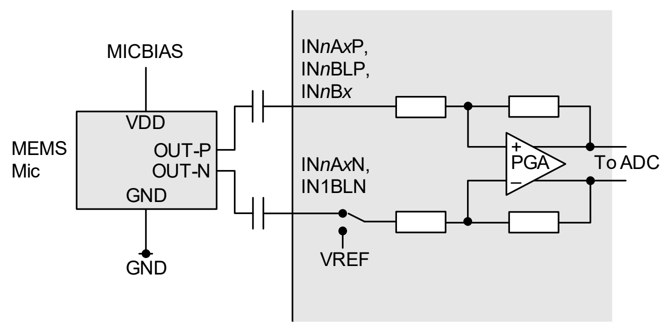


Figure 4-8. Differential MEMS Input

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 CS42L92 in a similar manner to the mic inputs. Single-ended and differential configurations are supported on each analog input path, using the IN*nx*_SRC bits as described in Section 4.2.6.

The analog line input configurations are shown in Fig. 4-9 and Fig. 4-10. Note that the microphone bias (MICBIAS) is not used for line input connections.

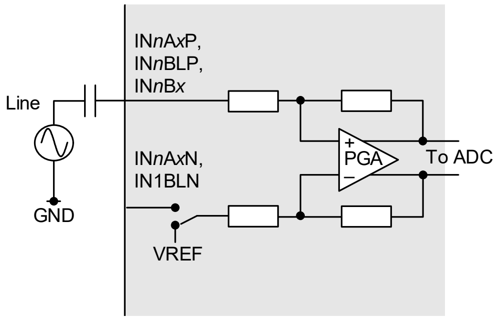


Figure 4-9. Single-Ended Line Input

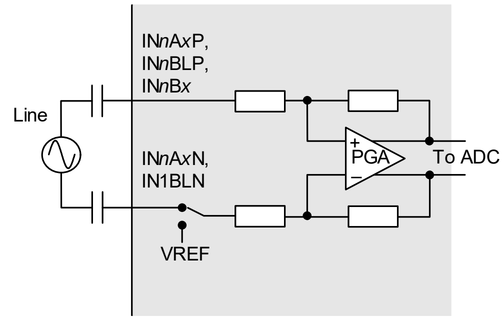


Figure 4-10. Differential Line Input

4.2.3 DMIC Input

As many as eight digital microphones can be connected to the CS42L92. DMIC operation on input paths IN1–IN4 is selected using IN1_MODE and IN2_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* output.

If DMIC input is enabled, the CS42L92 outputs the DMIC clock 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 DMICCLK*n* frequency is selected, the maximum valid sample rate for the respective paths is restricted as described in Table 4-1. If the input sample rates are set globally using IN_RATE (i.e., IN_RATE_MODE = 0), all input paths are affected similarly.

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

The DMIC clock 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 DMIC clock frequencies are scaled accordingly.

Table 4-1. DMICCLK Frequency

Condition	DMIC Clock 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 DMIC interfaces is selectable, using IN*n*_DMIC_SUP; each interface may be referenced to MICVDD, MICBIAS1, or MICBIAS2. 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. 4-11. The microphones must be configured to ensure that the left mic transmits a data bit when DMICCLK is high and the right mic transmits a data bit when DMICCLK is low. The CS42L92 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 CS42L92 provides integrated pull-down resistors on the DMICDAT*n* pins. This provides a flexible capability for interfacing with other devices.

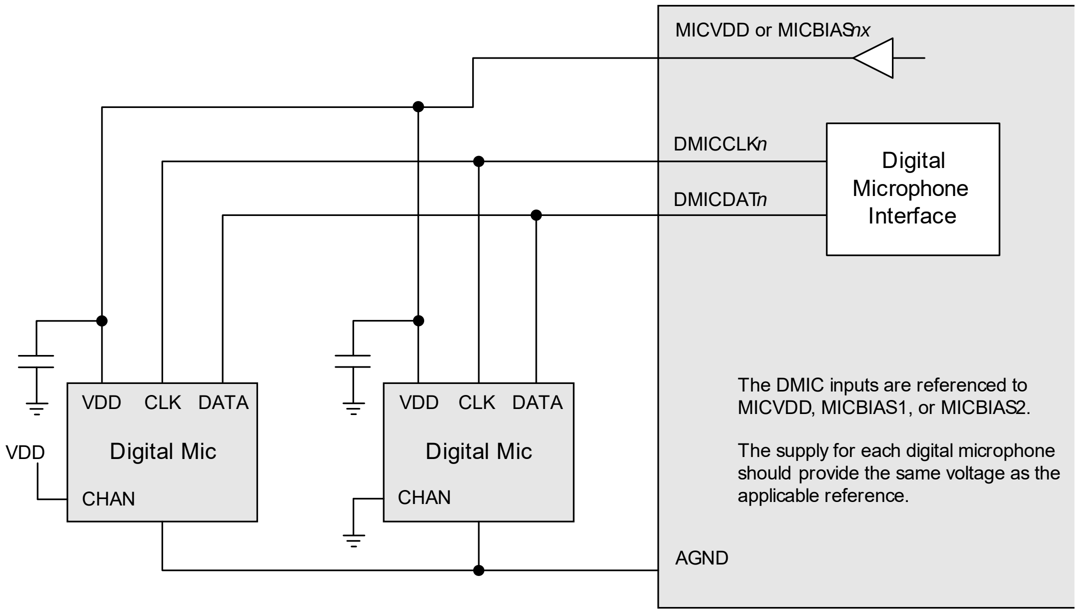


Figure 4-11. DMIC Input

Two DMIC channels are interleaved on DMICDAT*n*. The DMIC interface timing is shown in Fig. 4-12. Each microphone must tristate its data output when the other microphone is transmitting.

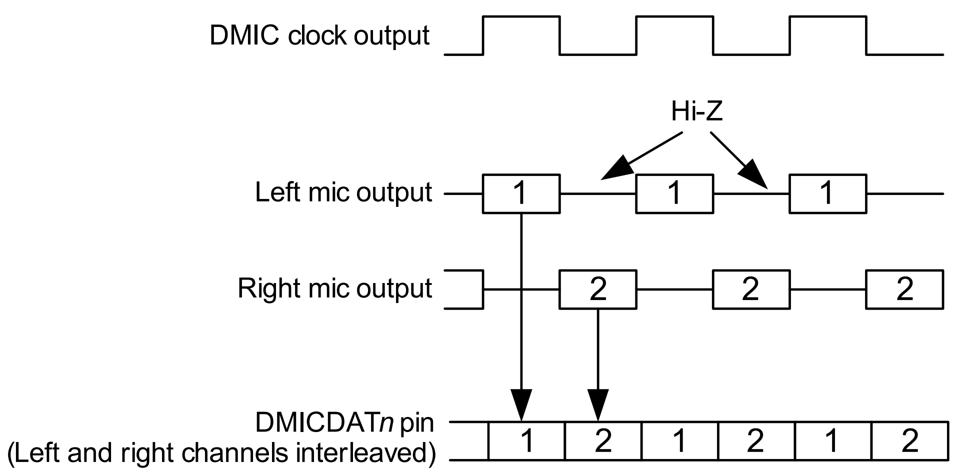


Figure 4-12. DMIC Interface Timing

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 ASYNCCLK and 32-kHz clock may also be required, depending on the path configuration. See Section 4.16 for details of the system clocks.

The CS42L92 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 input signal path. 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	7	IN4L_ENA	0	Input Path 4 (left) enable 0 = Disabled 1 = Enabled
	6	IN4R_ENA	0	Input Path 4 (right) enable 0 = Disabled 1 = Enabled
	5	IN3L_ENA	0	Input Path 3 (left) enable 0 = Disabled 1 = Enabled
	4	IN3R_ENA	0	Input Path 3 (right) enable 0 = Disabled 1 = Enabled
	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	7	IN4L_ENA_STS	0	Input Path 4 (left) enable status 0 = Disabled 1 = Enabled
	6	IN4R_ENA_STS	0	Input Path 4 (right) enable status 0 = Disabled 1 = Enabled
	5	IN3L_ENA_STS	0	Input Path 3 (left) enable status 0 = Disabled 1 = Enabled
	4	IN3R_ENA_STS	0	Input Path 3 (right) enable status 0 = Disabled 1 = Enabled
	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 CS42L92 digital core. The sample rate for the input signal paths can be set globally, or can be configured independently for each input channel.

The IN_RATE_MODE bit (defined in Table 4-3) controls whether the input sample rates are set globally using IN_RATE, or independently for each input channel using the IN*nx*_RATE fields (where *n* is 1–4 and *x* is L or R for the left/right channels respectively). The IN_RATE and IN*nx*_RATE fields are defined in Table 4-26.

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

4.2.6 Input Signal Path Configuration

The CS42L92 supports up to eight analog inputs or up to eight digital inputs. Selectable combinations of analog (mic or line) and digital inputs are multiplexed into four stereo input signal paths.

Input paths IN1 and IN2 can be configured for single-ended, differential, or DMIC operation. The analog input configuration and pin selection is controlled using the IN*nx*_SRC bits; digital input mode is selected by setting IN*n*_MODE.

Input paths IN3 and IN4 support digital inputs only. Note that the external pin connections are shared with the IN1R and IN2R analog input paths—the following restrictions apply:

- If IN3L or IN3R input paths are enabled, IN1R analog input is restricted to differential (IN1BR–IN1ALN) or single-ended (IN1BR) configurations only.
- If IN4L or IN4R input paths are enabled, IN2R analog input is restricted to single-ended (IN2BR) configuration only.

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

The 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 DMIC input is selected, the respective DMICCLK*n* frequency is controlled by the respective IN*n*_OSR field.

If a 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, MICBIAS1, or MICBIAS2. The voltage reference for each digital input path should be set equal to the applicable power supply of the respective microphones.

The CS42L92 input paths can be configured for power-saving operation, ideal for always-on applications. The low-power configurations allow the power consumption to be optimized with respect to the required audio performance characteristics.

- If a signal path is configured for analog input, low-power operation can be selected by setting the respective IN*nx*_LP_MODE bit. Mid-power operation can be configured by setting IN*n*_OSR = 100 for the respective signal paths. (Note that the IN*nx*_LP_MODE bit should be cleared in the mid-power configuration.) Note that the maximum input-signal level is reduced by 6 dB if mid-power operation is selected—see Table 3-4.
- If a signal path is configured for digital input, the respective DMICCLK*n* frequency can be configured using the IN*n*_OSR bits. Reducing the DMICCLK*n* frequency reduces power consumption at the expense of audio performance. The IN*n*_OSR field also supports high performance DMIC mode, when 6.144 MHz DMICCLK is selected.

  If 384- or 768-kHz DMICCLK*n* frequency is selected, the maximum sample rate for the respective paths is restricted as described in Table 4-1. If the input sample rates are set globally using IN_RATE (i.e., IN_RATE_MODE = 0), all input paths are affected similarly.

The MICVDD voltage is generated by an internal charge pump and LDO regulator. The MICBIAS*n* 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
R776 (0x0308) Input_Rate	10	IN_RATE_MODE	1	Input Path Sample Rate Configuration 0 = Global control (all input paths configured using IN_RATE) 1 = Individual channel control (using the respective INnx_RATE fields)
R780 (0x030C) HPF_Control	2:0	IN_HPF_CUT[2:0]	010	Input Path HPF Select. Controls the cut-off frequency of the input path HPF circuits. 000 = 2.5 Hz; 010 = 10 Hz; 100 = 40 Hz 001 = 5 Hz; 011 = 20 Hz; All other codes are reserved
R784 (0x0310) IN1L_Control	15	IN1L_HPF	0	Input Path 1 (Left) HPF Enable 0 = Disabled 1 = Enabled
	12:11	IN1_DMIC_SUP[1:0]	00	Input Path 1 DMIC Reference Select (sets the DMICDAT1 and DMICCLK1 logic levels) 00 = MICVDD; 10 = MICBIAS2 01 = MICBIAS1; 11 = Reserved
	10	IN1_MODE	0	Input Path 1 Mode 0 = Analog input 1 = Digital input
	7:1	IN1L_PGA_VOL[6:0]	0x40	Input Path 1 (Left) PGA Volume (applicable to analog input only) 0x00 to 0x3F = Reserved; 0x42 = 2 dB; 0x60 to 0x7F = Reserved 0x40 = 0 dB; … (1-dB steps) 0x41 = 1 dB; 0x5F = 31 dB
R785 (0x0311) ADC_Digital_Volume_1L	14:13	IN1L_SRC[1:0]	00	Input Path 1 (Left) Source 00 = Differential (IN1ALP–IN1ALN); 10 = Differential (IN1BLP–IN1BLN) 01 = Single-ended (IN1ALP); 11 = Single-ended (IN1BLP)
	11	IN1L_LP_MODE	0	Input Path 1 (Left) Low-Power Mode (applicable to analog input only) 0 = High Performance Mode 1 = Low Power Mode
R786 (0x0312) DMIC1L_Control	10:8	IN1_OSR[2:0]	101	Input Path 1 Oversample Rate Control If analog input is selected, this field is used to select Mid-Power Mode. 100 = Mid Power Mode; All other codes are reserved 101 = Normal If digital input is selected, this field controls the DMICCLK1 frequency. 010 = 384 kHz; 101 = 3.072 MHz 011 = 768 kHz; 110 = 6.144 MHz 100 = 1.536 MHz; All other codes are reserved
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 input 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 (IN1BR–IN1ALN) 01 = Single-ended (IN1ARP); 11 = Single-ended (IN1BR)
	11	IN1R_LP_MODE	0	Input Path 1 (Right) Low-Power Mode (applicable to analog input only) 0 = High Performance Mode 1 = Low Power Mode

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

Register Address	Bit	Label	Default	Description
R792 (0x0318) IN2L_Control	15	IN2L_HPF	0	Input Path 2 (Left) HPF Enable 0 = Disabled 1 = Enabled
	12:11	IN2_DMIC_SUP[1:0]	00	Input Path 2 DMIC Reference Select (sets the DMICDAT2 and DMICCLK2 logic levels) 00 = MICVDD 10 = MICBIAS2 01 = MICBIAS1 11 = Reserved
	10	IN2_MODE	0	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 input 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 (IN2ALP–IN2ALN) 10 = Differential (IN2BL–IN2ALN) 01 = Single-ended (IN2ALP) 11 = Single-ended (IN2BL)
	11	IN2L_LP_MODE	0	Input Path 2 (Left) Low-Power Mode (applicable to analog input only) 0 = High Performance Mode 1 = Low Power Mode
R794 (0x031A) DMIC2L_Control	10:8	IN2_OSR[2:0]	101	Input Path 2 Oversample Rate Control If analog input is selected, this field is used to select Mid-Power Mode. 100 = Mid Power Mode All other codes are reserved 101 = Normal If digital input is selected, this field controls the DMICCLK2 frequency. 010 = 384 kHz 101 = 3.072 MHz 011 = 768 kHz 110 = 6.144 MHz 100 = 1.536 MHz All other codes are reserved
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 input 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 (IN2ARP–IN2ARN) 10 = Differential (IN2BR–IN2ARN) 01 = Single-ended (IN2ARP) 11 = Single-ended (IN2BR)
	11	IN2R_LP_MODE	0	Input Path 2 (Right) Low-Power Mode (applicable to analog input only) 0 = High Performance Mode 1 = Low Power Mode
R800 (0x0320) IN3L_Control	15	IN3L_HPF	0	Input Path 3 (Left) HPF Enable 0 = Disabled 1 = Enabled
	12:11	IN3_DMIC_SUP[1:0]	00	Input Path 3 DMIC Reference Select (sets the DMICDAT3 and DMICCLK3 logic levels) 00 = MICVDD 10 = MICBIAS2 01 = MICBIAS1 11 = Reserved
R802 (0x0322) DMIC3L_Control	10:8	IN3_OSR[2:0]	101	Input Path 3 Oversample Rate Control - selects the DMICCLK3 frequency. 010 = 384 kHz 101 = 3.072 MHz 011 = 768 kHz 110 = 6.144 MHz 100 = 1.536 MHz All other codes are reserved
R804 (0x0324) IN3R_Control	15	IN3R_HPF	0	Input Path 3 (Right) HPF Enable 0 = Disabled 1 = Enabled

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

Register Address	Bit	Label	Default	Description
R808 (0x0328) IN4L_Control	15	IN4L_HPF	0	Input Path 4 (Left) HPF Enable 0 = Disabled 1 = Enabled
	12:11	IN4_DMIC_SUP[1:0]	00	Input Path 4 DMIC Reference Select (sets the DMICDAT4 and DMICCLK4 logic levels) 00 = MICVDD 10 = MICBIAS2 01 = MICBIAS1 11 = Reserved
R810 (0x032A) DMIC4L_Control	10:8	IN4_OSR[2:0]	101	Input Path 4 Oversample Rate Control - selects the DMICCLK4 frequency. 010 = 384 kHz 101 = 3.072 MHz 011 = 768 kHz 110 = 6.144 MHz 100 = 1.536 MHz All other codes are reserved
R812 (0x032C) IN4R_Control	15	IN4R_HPF	0	Input Path 4 (Right) HPF Enable 0 = Disabled 1 = Enabled

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–IN4 digital input paths is not equal to the 0 dBFS level of the CS42L92 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 CS42L92 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

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

Register Address	Bit	Label	Default	Description
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 = –64 dB; 0x80 = 0 dB; 0xC0 to 0xFF = Reserved 0x01 = –63.5 dB; … (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 = –64 dB; 0x80 = 0 dB; 0xC0 to 0xFF = Reserved 0x01 = –63.5 dB; … (0.5-dB steps) … (0.5-dB steps); 0xBF = +31.5 dB
R793 (0x0319) ADC_Digital_ Volume_2L	9	IN_VU	See Footnote 1	Input Signal Paths Volume and Mute Update. Writing 1 to this bit causes the Input Signal Paths Volume and Mute settings to be updated simultaneously
	8	IN2L_MUTE	1	Input Path 2 (Left) Digital Mute 0 = Unmute 1 = Mute
	7:0	IN2L_VOL[7:0]	0x80	Input Path 2 (Left) Digital Volume (see Table 4-5 for volume register definition). –64 dB to +31.5 dB in 0.5-dB steps 0x00 = –64 dB; 0x80 = 0 dB; 0xC0 to 0xFF = Reserved 0x01 = –63.5 dB; … (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 = –64 dB; 0x80 = 0 dB; 0xC0 to 0xFF = Reserved 0x01 = –63.5 dB; … (0.5-dB steps) … (0.5-dB steps); 0xBF = +31.5 dB
R801 (0x0321) ADC_Digital_ Volume_3L	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	IN3L_MUTE	1	Input Path 3 (Left) Digital Mute 0 = Unmute 1 = Mute
	7:0	IN3L_VOL[7:0]	0x80	Input Path 3 (Left) Digital Volume (see Table 4-5 for volume register definition). –64 dB to +31.5 dB in 0.5-dB steps 0x00 = –64 dB; 0x80 = 0 dB; 0xC0 to 0xFF = Reserved 0x01 = –63.5 dB; … (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
R805 (0x0325) ADC_Digital_ Volume_3R	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	IN3R_MUTE	1	Input Path 3 (Right) Digital Mute 0 = Unmute 1 = Mute
	7:0	IN3R_VOL[7:0]	0x80	Input Path 3 (Right) Volume (see Table 4-5 for volume register definition). –64 dB to +31.5 dB in 0.5-dB steps 0x00 = –64 dB 0x80 = 0 dB 0xC0 to 0xFF = Reserved 0x01 = –63.5 dB … (0.5-dB steps) … (0.5-dB steps) 0xBF = +31.5 dB
R809 (0x0329) ADC_Digital_ Volume_4L	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	IN4L_MUTE	1	Input Path 4 (Left) Digital Mute 0 = Unmute 1 = Mute
	7:0	IN4L_VOL[7:0]	0x80	Input Path 4 (Left) Digital Volume (see Table 4-5 for volume register definition). –64 dB to +31.5 dB in 0.5-dB steps 0x00 = –64 dB 0x80 = 0 dB 0xC0 to 0xFF = Reserved 0x01 = –63.5 dB … (0.5-dB steps) … (0.5-dB steps) 0xBF = +31.5 dB
R813 (0x032D) ADC_Digital_ Volume_4R	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	IN4R_MUTE	1	Input Path 4 (Right) Digital Mute 0 = Unmute 1 = Mute
	7:0	IN4R_VOL[7:0]	0x80	Input Path 4 (Right) Digital Volume (see Table 4-5 for volume register definition). –64 dB to +31.5 dB in 0.5-dB steps 0x00 = –64 dB 0x80 = 0 dB 0xC0 to 0xFF = Reserved 0x01 = –63.5 dB … (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

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)
0x12	–55.0	0x43	–30.5	0x74	–6.0	0xA5	18.5
0x13	–54.5	0x44	–30.0	0x75	–5.5	0xA6	19.0
0x14	–54.0	0x45	–29.5	0x76	–5.0	0xA7	19.5
0x15	–53.5	0x46	–29.0	0x77	–4.5	0xA8	20.0
0x16	–53.0	0x47	–28.5	0x78	–4.0	0xA9	20.5
0x17	–52.5	0x48	–28.0	0x79	–3.5	0xAA	21.0
0x18	–52.0	0x49	–27.5	0x7A	–3.0	0xAB	21.5
0x19	–51.5	0x4A	–27.0	0x7B	–2.5	0xAC	22.0
0x1A	–51.0	0x4B	–26.5	0x7C	–2.0	0xAD	22.5
0x1B	–50.5	0x4C	–26.0	0x7D	–1.5	0xAE	23.0
0x1C	–50.0	0x4D	–25.5	0x7E	–1.0	0xAF	23.5
0x1D	–49.5	0x4E	–25.0	0x7F	–0.5	0xB0	24.0
0x1E	–49.0	0x4F	–24.5	0x80	0.0	0xB1	24.5
0x1F	–48.5	0x50	–24.0	0x81	0.5	0xB2	25.0
0x20	–48.0	0x51	–23.5	0x82	1.0	0xB3	25.5
0x21	–47.5	0x52	–23.0	0x83	1.5	0xB4	26.0
0x22	–47.0	0x53	–22.5	0x84	2.0	0xB5	26.5
0x23	–46.5	0x54	–22.0	0x85	2.5	0xB6	27.0
0x24	–46.0	0x55	–21.5	0x86	3.0	0xB7	27.5
0x25	–45.5	0x56	–21.0	0x87	3.5	0xB8	28.0
0x26	–45.0	0x57	–20.5	0x88	4.0	0xB9	28.5
0x27	–44.5	0x58	–20.0	0x89	4.5	0xBA	29.0
0x28	–44.0	0x59	–19.5	0x8A	5.0	0xBB	29.5
0x29	–43.5	0x5A	–19.0	0x8B	5.5	0xBC	30.0
0x2A	–43.0	0x5B	–18.5	0x8C	6.0	0xBD	30.5
0x2B	–42.5	0x5C	–18.0	0x8D	6.5	0xBE	31.0
0x2C	–42.0	0x5D	–17.5	0x8E	7.0	0xBF	31.5
0x2D	–41.5	0x5E	–17.0	0x8F	7.5	0xC0–0xFF	Reserved
0x2E	–41.0	0x5F	–16.5	0x90	8.0		
0x2F	–40.5	0x60	–16.0	0x91	8.5		
0x30	–40.0	0x61	–15.5	0x92	9.0		

4.2.8 Input Signal Path Signal-Detect Control

The CS42L92 provides a digital signal-detect function for the input signal path. This enables system actions to be triggered by signal detection and allows the device to remain in a low-power state until a valid audio signal is detected. A mute function is integrated with the signal-detect circuit, ensuring the respective digital audio path remains at zero until the detection threshold level is reached. Signal detection is also indicated via the interrupt controller.

The signal-detect function is supported on input paths IN1–IN4 in analog and digital configurations. (For input paths IN1 and IN2, digital input is selected by setting the respective IN*n*_MODE bit.) Note that the valid operating conditions for this function vary, depending on the applicable signal-path configuration.

- The signal-detect function is supported on analog input paths for sample rates up to 16 kHz.
- The signal-detect function is supported on digital input paths for sample rates up to 16 kHz (if DMICCLK*n* ≥ 768kHz) and up to 48 kHz (if DMICCLK*n* ≥ 2.8224 MHz).

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

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

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

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

The signal-detect function is an input to the interrupt control circuit and can be used to trigger an interrupt event; see Section 4.15. Note that the respective interrupt event represents the logic OR of the signal detection on all input channels and does not provide indication of which input channel caused the interrupt. To avoid multiple interrupts, the signal-detect interrupt can be reasserted only after all input channels have fallen below the trigger threshold level.

The input path signal-detection control registers are described in Table 4-6.

Table 4-6. Input Signal Path Signal-Detect Control

Register Address	Bit	Label	Default	Description
R786 (0x0312) DMIC1L_Control	15	IN1L_SIG_DET_ ENA	0	Input Path 1 (Left) Signal-Detect Enable 0 = Disabled 1 = Enabled
R790 (0x0316) DMIC1R_Control	15	IN1R_SIG_DET_ ENA	0	Input Path 1 (Right) Signal-Detect Enable 0 = Disabled 1 = Enabled
R794 (0x031A) DMIC2L_Control	15	IN2L_SIG_DET_ ENA	0	Input Path 2 (Left) Signal-Detect Enable 0 = Disabled 1 = Enabled
R798 (0x031E) DMIC2R_Control	15	IN2R_SIG_DET_ ENA	0	Input Path 2 (Right) Signal-Detect Enable 0 = Disabled 1 = Enabled
R802 (0x0320) DMIC3L_Control	15	IN3L_SIG_DET_ ENA	0	Input Path 3 (Left) Signal-Detect Enable 0 = Disabled 1 = Enabled
R806 (0x0326) DMIC3R_Control	15	IN3R_SIG_DET_ ENA	0	Input Path 3 (Right) Signal-Detect Enable 0 = Disabled 1 = Enabled
R810 (0x032A) DMIC4L_Control	15	IN4L_SIG_DET_ ENA	0	Input Path 4 (Left) Signal-Detect Enable 0 = Disabled 1 = Enabled
R814 (0x032E) DMIC4R_Control	15	IN4R_SIG_DET_ ENA	0	Input Path 4 (Right) Signal-Detect Enable 0 = Disabled 1 = Enabled
R832 (0x0340) Signal_Detect_Globals	8:4	IN_SIG_DET_ THR[4:0]	0x00	Input Signal Path Signal-Detect Threshold 0x00 = –30.1 dB, 0x01 = –36.1 dB, 0x02 = –42.1 dB, 0x03 = –48.2 dB, 0x04 = –50.7 dB 0x05 = –54.2 dB, 0x06 = –56.7 dB, 0x07 = –60.2 dB, 0x08 = –66.2 dB, 0x09 = –68.7 dB 0x0A = –72.2 dB, 0x0B = –74.7 dB, 0x0C = –78.3 dB, 0x0D = –80.8 dB All other codes are reserved
	3:0	IN_SIG_DET_ HOLD[3:0]	0001	Input Signal Path Signal-Detect Hold Time (delay before signal detect indication is deasserted) 0000 = Reserved, 0001 = 4–8 ms, 0010 = 8–12 ms, 0011 = 12–16 ms ... (4-ms steps) 1001 = 36–40 ms, 1010 = 40–44 ms, 1011 = 48–52 ms 1100 = 96–100 ms, 1101 = 192–196 ms, 1110 = 384–388 ms, 1111 = 768–772 ms

4.2.9 Ultrasonic Signal Demodulation

The CS42L92 provides ultrasonic signal-processing functions on the input signal paths. Configurable filters and demodulator functions enable ultrasonic signals to be translated down to the audio band and routed through the digital mixer core. Two ultrasonic processing blocks are incorporated, with independent configuration controls for each.

The input source for the ultrasonic blocks is configured using the US*n*_SRC fields (where *n* identifies the applicable block US1 or US2). The input signal gain is set using US*n*_GAIN. The input frequency range is selected by US*n*_FREQ.

The ultrasonic functions can be supported on any of the input signal paths (IN1–IN4). The inputs to the ultrasonic functions are independent of the enable, mute, and digital-volume settings of the respective input signal paths—the ultrasonic functions do not require the selected input signal paths to be enabled, and the mute/digital-volume settings of the input signal paths have no effect on the ultrasonic functions.

The system clock, SYSCLK, must be present and enabled when using the ultrasonic functions. The SYSCLK frequency must be a multiple of 6.144 MHz (SYSCLK_FRAC = 0) in this case. See Section 4.16 for details of SYSCLK and the associated registers.

The ultrasonic demodulator function is enabled by setting US*n*_ENA. The frequency band and signal gain are selected using US*n*_FREQ and US*n*_GAIN respectively.

The output from the ultrasonic demodulator is a frequency-modulated image of the selected input frequency range. The folding frequency that characterizes the frequency modulation is set according to the US*n*_FREQ setting—see Table 4-7. The relationship between input and output frequencies is described in Eq. 4-1.

$$F_{OUT} = |F_{IN} - F_{FOLD}|$$

Equation 4-1. Ultrasonic Demodulator Characteristic

Note that, depending on the input frequency range and the folding frequency, F_{FOLD}, the modulated output in respect of certain input frequencies may overlap others. This effect arises if the folding frequency lies within the input frequency range, with the result that two different input frequencies will each be modulated to the same output frequency. This effect is limited to the outer edges of the input frequency range in all cases. Amplitude response across the input frequency range is flat to within 1.5 dB in all cases.

The demodulated ultrasonic outputs can be selected as input to the digital mixers or signal-processing functions within the digital core by setting the respective x_SRCn fields as described in Section 4.3.1.

The sample rate for the demodulated ultrasonic output is configured using US*n*_RATE—see Table 4-26. The selected sample rate must be one of the SYSCLK-related rates, and must be equal to the output rate set by US*n*_FREQ (see Table 4-7). Note that sample-rate conversion is required when routing the ultrasonic signals to any signal chain that is asynchronous or configured for a different sample rate.

The characteristics associated with the US*n*_FREQ field setting are shown in Table 4-7.

Table 4-7. Ultrasonic Frequency Control

Condition	Input Frequency Band	Output Sample Rate	Demodulator Folding Frequency (F_{FOLD})
US*n*_FREQ = 000	24.5–40.5 kHz	32 kHz	40.42 kHz
US*n*_FREQ = 001	18–22 kHz	8 kHz	18.29 kHz
US*n*_FREQ = 010	16–24 kHz	16 kHz	16 kHz
US*n*_FREQ = 011	20–28 kHz	16 kHz	20.21 kHz

The ultrasonic demodulation control registers are described in Table 4-8.

Table 4-8. Ultrasonic Signal Demodulation Control

Register Address	Bit	Label	Default	Description
R4224 (0x1080) US1_Ctrl_0	13:12	US1_GAIN[1:0]	10	Ultrasonic Demodulator 1 Gain 00 = Disabled (no signal) 10 = 1 dB 01 = –5 dB 11 = 7 dB
	11:8	US1_SRC[3:0]	0x0	Ultrasonic Demodulator 1 Source 0x0 = IN1L 0x3 = IN2R 0x6 = IN4L 0x1 = IN1R 0x4 = IN3L 0x7 = IN4R 0x2 = IN2L 0x5 = IN3R All other codes are reserved
	6:4	US1_FREQ[2:0]	011	Ultrasonic Demodulator 1 Frequency 000 = 24.5–40.5 kHz 010 = 16–24 kHz All other codes are reserved 001 = 18–22 kHz 011 = 20–28 kHz
	0	US1_ENA	0	Ultrasonic Demodulator 1 Enable 0 = Disabled 1 = Enabled
R4226 (0x1082) US2_Ctrl_0	13:12	US2_GAIN[1:0]	10	Ultrasonic Demodulator 2 Gain 00 = Disabled (no signal) 10 = 1 dB 01 = –5 dB 11 = 7 dB
	11:8	US2_SRC[3:0]	0x0	Ultrasonic Demodulator 2 Source 0x0 = IN1L 0x3 = IN2R 0x6 = IN4L 0x1 = IN1R 0x4 = IN3L 0x7 = IN4R 0x2 = IN2L 0x5 = IN3R All other codes are reserved
	6:4	US2_FREQ[2:0]	011	Ultrasonic Demodulator 2 Frequency 000 = 24.5–40.5 kHz 010 = 16–24 kHz All other codes are reserved 001 = 18–22 kHz 011 = 20–28 kHz
	0	US2_ENA	0	Ultrasonic Demodulator 2 Enable 0 = Disabled 1 = Enabled

4.2.10 Auxiliary PDM Interface

The auxiliary PDM interface supports a one-channel digital output derived from one of the CS42L92 analog input paths. This can be used to provide an audio path between an analog microphone connected to the CS42L92 and a digital input to an external audio processor.

The auxiliary PDM interface signal path is shown in Fig. 4-13.

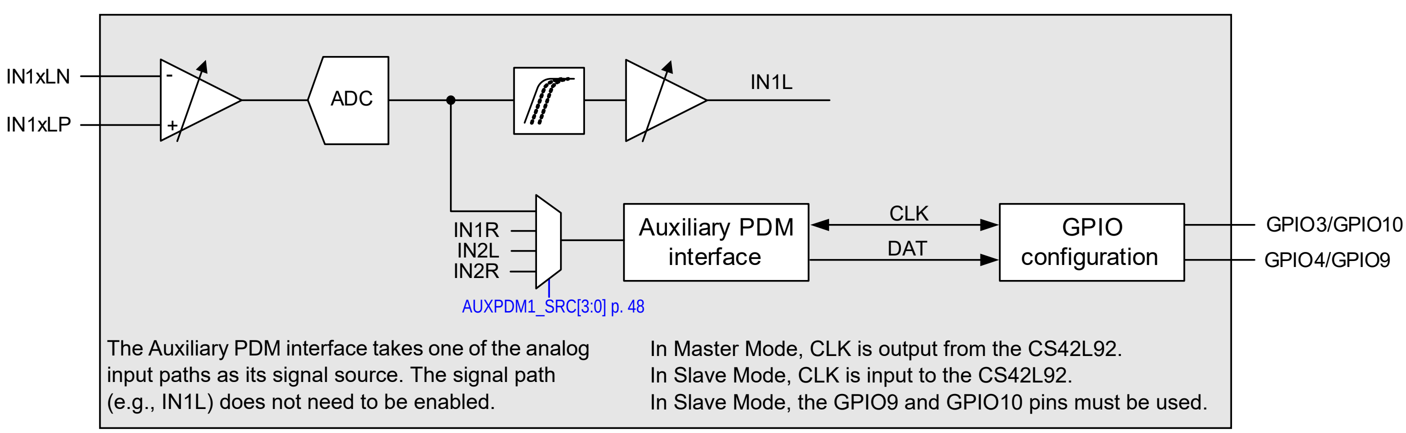


Figure 4-13. Auxiliary PDM Interface

The input source for the auxiliary PDM interface is selected using AUXPDM1_SRC. Note that the selected input path must be configured for analog input—the auxiliary PDM function is not supported for the DMICDAT*n* inputs.

Note: The input to the auxiliary PDM interface is independent of the enable, mute, and digital-volume settings of the respective input signal path—the auxiliary PDM function does not require the selected input signal paths to be enabled, and the mute/digital-volume settings of the input signal paths have no effect on the auxiliary PDM function.

The auxiliary PDM interface is enabled by setting AUXPDM1_ENA. Note that the other auxiliary PDM control fields should be configured before enabling the interface. The external connections (GPIO pins) should also be configured before enabling the interface; the AUXPDM1_ENA bit should be set as the final step of the enable sequence. The AUXPDM1_ENA bit should be cleared before changing the interface configuration.

The output signal can be muted and unmuted using AUXPDM1_MUTE.

The interface operates in master or slave modes, selected using the AUXPDM1_MSTR bit. In Master Mode, the clock (CLK) signal is generated by the CS42L92; in Slave Mode, the CLK signal is an input to the CS42L92.

The CLK frequency is selected using the AUXPDM1_CLK_FREQ field. For each setting of this field, the actual frequency depends on whether SYSCLK is configured for 48- or 44.1-kHz related sample rates. See Section 4.16 for details of the system clocks. Note that the CLK frequency must be configured in master and slave modes, using the AUXPDM1_CLK_FREQ field.

The timing of the data (DAT) output signal can be controlled using AUXPDM1_TXEDGE. This selects whether the DAT output changes on the rising or falling edge of CLK.

The auxiliary PDM interface timing is shown in Fig. 4-14. In Master Mode, the clock and data outputs are driven synchronously by the CS42L92. In Slave Mode, the timing of the output data is controlled by the external clock input.

In Slave Mode, the CS42L92 system clock (SYSCLK) must be synchronized to the CLK input. This is achieved by using one of the FLLs to generate the system clock, with AIF2BCLK configured as the FLL clock reference. (If the auxiliary PDM interface is configured in Slave Mode, the CLK input is supported on the AIF2BCLK/GPIO10 pin.) Further details of the FLLs are provided in Section 4.16.8.

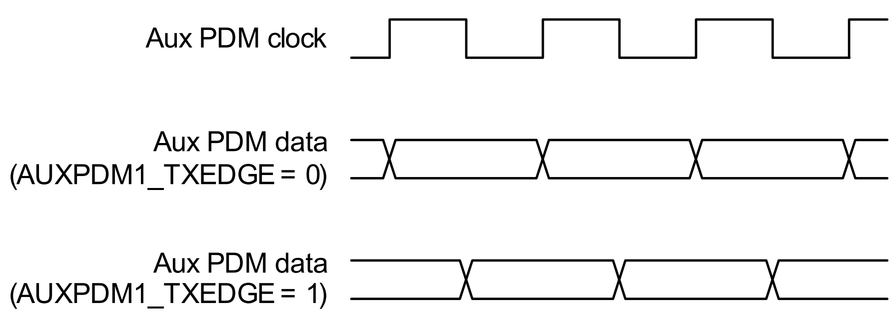


Figure 4-14. Auxiliary PDM Interface Timing

The external connections associated with the auxiliary PDM interface are implemented on GPIO pins, which must be configured for the respective CLK and DAT functions. The auxiliary PDM signals are alternative functions available on specific GPIO pins only. See Section 4.14 to configure the GPIO pins for the auxiliary PDM interface. Note that the output pins (including the CLK function, if Master Mode is selected) must be configured as outputs using the respective GP*n*_DIR fields.

- In Master Mode, the CLK output can be configured on the GPIO3 or GPIO10 pins. The DAT output can be configured on GPIO4 or GPIO9.
- In Slave Mode, the CLK input is supported on GPIO10 only. The DAT output is supported on GPIO9.

The auxiliary PDM interface control registers are described in Table 4-9.

Table 4-9. Auxiliary PDM Interface Control

Register Address	Bit	Label	Default	Description
R4288 (0x10C0) AUXPDM1_Ctrl_0	11:8	AUXPDM1_SRC[3:0]	0x0	Auxiliary PDM 1 Source 0x0 = IN1L 0x1 = IN1R 0x2 = IN2L 0x3 = IN2R All other codes are reserved Note that the selected input source must be configured for analog input—digital input paths are not supported. The Auxiliary PDM interface must be disabled when updating this field.
	4	AUXPDM1_TXEDGE	0	Auxiliary PDM 1 Timing 0 = Output data is driven on rising edge of AUXPDM1_CLK 1 = Output data is driven on falling edge of AUXPDM1_CLK The Auxiliary PDM interface must be disabled when updating this field.
	3	AUXPDM1_MSTR	1	Auxiliary PDM 1 Master Mode select 0 = AUXPDM1_CLK Slave Mode (input) 1 = AUXPDM1_CLK Master mode (output) The Auxiliary PDM interface must be disabled when updating this field.
	2	AUXPDM1_MUTE	0	Auxiliary PDM 1 Mute 0 = Unmute 1 = Mute
	0	AUXPDM1_ENA	0	Auxiliary PDM 1 Enable 0 = Disabled 1 = Enabled
R4281 (0x10C1) AUXPDM1_Ctrl_1	15:14	AUXPDM1_FREQ[1:0]	01	Auxiliary PDM 1 CLK Rate 00 = 3.072 MHz (2.8824 MHz) 01 = 2.048 MHz (1.8816 MHz) 10 = 1.536 MHz (1.4112 MHz) 11 = 768 kHz (705.6 kHz) The frequencies in brackets apply for 44.1 kHz–related sample rates only (i.e., if SYSCLK_FRAC = 1). The Auxiliary PDM interface must be disabled when updating this field.

4.2.11 DMIC Pin Configuration

DMIC operation on input paths IN1–IN4 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 CS42L92 provides integrated pull-down resistors on each DMICDAT*n* pin. This provides a flexible capability for interfacing with other devices.

The DMICDAT*n* pull-down resistors can be configured independently using the bits described in Table 4-10. Note that, if the DMICDAT*n* DMIC input paths are disabled, the pull-down is disabled on the respective pin.

Table 4-10. DMIC Interface Pull-Down Control

Register Address	Bit	Label	Default	Description
R840 (0x0348) Dig_Mic_Pad_Ctrl	3	DMICDAT4_PD	0	DMICDAT4 Pull-Down Control 0 = Disabled 1 = Enabled
	2	DMICDAT3_PD	0	DMICDAT3 Pull-Down Control 0 = Disabled 1 = Enabled
	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 CS42L92 digital core provides extensive mixing and processing capabilities for multiple signal paths. The configuration is highly flexible and supports virtually every conceivable input/output connection between the available processing blocks.

The digital core provides parametric equalization (EQ) functions, DRC, and low-/high-pass filters (LHPF).

The CS42L92 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 or referenced to asynchronous clock domains. Data-format conversion (DFC) functions are available to support different interface standards on the input and output signal paths.

The integrated DSP provides a general-purpose signal processing capability; this is supported by general-purpose timer and event-logger functions. Note that the DSP configuration data is lost whenever the DCVDD power domain is removed; the DSP configuration data must be downloaded to the CS42L92 each time the device is powered up.

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 CS42L92 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. 4-15. The control registers associated with the digital-core signal paths are shown in Fig. 4-16 through Fig. 4-33. The full list of digital mixer control registers (R1600–R3576) is provided in Section 6. Generic register field definitions are provided in Table 4-11.

The digital audio core is predominantly a 24-bit architecture, but also provides support for 32-bit signal paths. Audio data samples of up to 32 bits can be received via the AIF1, AIF3, and SLIMbus input channels and routed to the AIF1, AIF3, SLIMbus, S/PDIF, and DAC output paths. The respective output mixers provide full support for 32-bit data words.

Note that all other signal paths and signal-processing blocks within the digital core are limited to 24-bit data length; data samples are truncated to 24-bit length if they are routed through any function that does not support 32-bit data words.

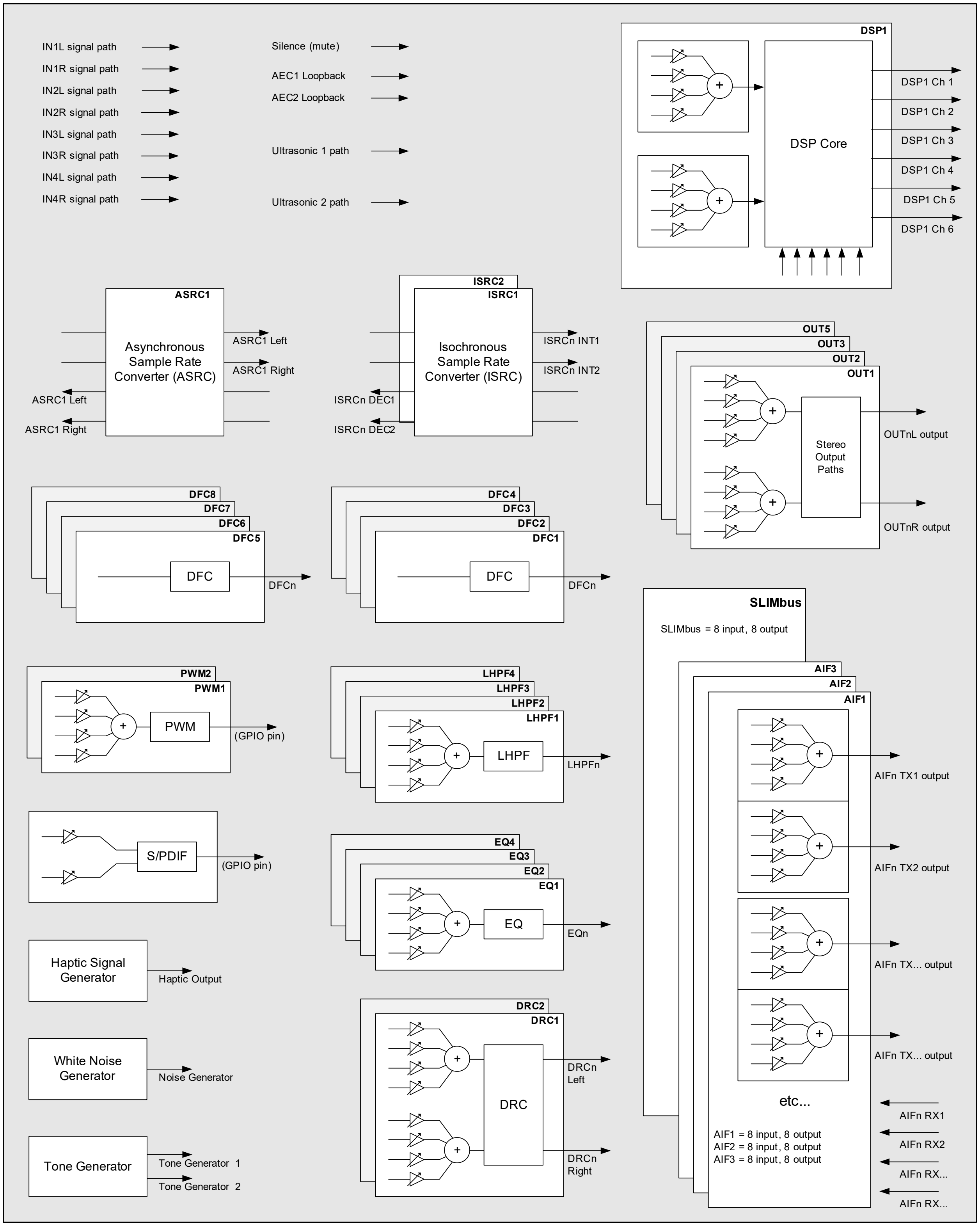


Figure 4-15. Digital Core

4.3.1 Digital-Core Mixers

The CS42L92 provides an extensive digital mixing capability. The digital-core mixing and signal-processing blocks are shown in Fig. 4-15. 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. 4-16–Fig. 4-33. The full list of digital mixer control registers (R1600–R3576) 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 4-11.

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 ASRC, ISRC, DFC, 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 4.3.15 and Section 4.3.16. The DFCs provide support for different data types, including floating point formats. Note that, if unsigned or floating point data is present within the digital core, some restrictions on the valid signal routing options apply—see Section 4.3.13.

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 4-11.

Table 4-11. Digital-Core Mixer Control Registers

Register Address	Bit	Label	Default	Description		
R1600 (0x0640) to R3576 (0x0DF8)	15	x_STS*n*	0	[Digital Core function] input *n* status 0 = Disabled 1 = Enabled		
	7:1	x_VOL*n*	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*	0x00	[Digital Core function] input *n* source select		
				0x00 = Silence (mute)	0x2E = AIF2 RX7	0x63 = LHPF4
				0x04 = Tone generator 1	0x2F = AIF2 RX8	0x68 = DSP1 Channel 1
				0x05 = Tone generator 2	0x30 = AIF3 RX1	0x69 = DSP1 Channel 2
				0x06 = Haptic generator	0x31 = AIF3 RX2	0x6A = DSP1 Channel 3
				0x08 = AEC Loop-Back 1	0x32 = AIF3 RX3	0x6B = DSP1 Channel 4
				0x09 = AEC Loop-Back 2	0x33 = AIF3 RX4	0x6C = DSP1 Channel 5
				0x0D = Noise generator	0x34 = AIF3 RX5	0x6D = DSP1 Channel 6
				0x10 = IN1L signal path	0x35 = AIF3 RX6	0x90 = ASRC1 IN1 Left
				0x11 = IN1R signal path	0x36 = AIF3 RX7	0x91 = ASRC1 IN1 Right
				0x12 = IN2L signal path	0x37 = AIF3 RX8	0x92 = ASRC1 IN2 Left
				0x13 = IN2R signal path	0x38 = SLIMbus RX1	0x93 = ASRC1 IN2 Right
				0x14 = IN3L signal path	0x39 = SLIMbus RX2	0xA0 = ISRC1 INT1
				0x15 = IN3R signal path	0x3A = SLIMbus RX3	0xA1 = ISRC1 INT2
				0x16 = IN4L signal path	0x3B = SLIMbus RX4	0xA4 = ISRC1 DEC1
				0x17 = IN4R signal path	0x3C = SLIMbus RX5	0xA5 = ISRC1 DEC2
				0x20 = AIF1 RX1	0x3D = SLIMbus RX6	0xA8 = ISRC2 INT1
				0x21 = AIF1 RX2	0x3E = SLIMbus RX7	0xA9 = ISRC2 INT2
				0x22 = AIF1 RX3	0x3F = SLIMbus RX8	0xAC = ISRC2 DEC1
				0x23 = AIF1 RX4	0x50 = EQ1	0xAD = ISRC2 DEC2
				0x24 = AIF1 RX5	0x51 = EQ2	0xF0 = US1
				0x25 = AIF1 RX6	0x52 = EQ3	0xF1 = US2
				0x26 = AIF1 RX7	0x53 = EQ4	0xF8 = DFC1
				0x27 = AIF1 RX8	0x58 = DRC1 Left	0xF9 = DFC2
				0x28 = AIF2 RX1	0x59 = DRC1 Right	0xFA = DFC3
				0x29 = AIF2 RX2	0x5A = DRC2 Left	0xFB = DFC4
				0x2A = AIF2 RX3	0x5B = DRC2 Right	0xFC = DFC5
				0x2B = AIF2 RX4	0x60 = LHPF1	0xFD = DFC6
				0x2C = AIF2 RX5	0x61 = LHPF2	0xFE = DFC7
				0x2D = AIF2 RX6	0x62 = LHPF3	0xFF = DFC8

4.3.2 Digital-Core Inputs

The digital core comprises multiple input paths, as shown in Fig. 4-16. Any of these inputs may be selected as a source to the digital mixers or signal-processing functions within the CS42L92 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 CS42L92 digital core. Those input sources, which are not shown in Fig. 4-16, are described separately throughout Section 4.3.

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 sample rate for the input signal paths is configured by using the applicable IN_RATE, AIF*n*_RATE, SLIMRX*n*_RATE, or US*n*_RATE field; see Table 4-26. Note that sample-rate conversion is required when routing the input signal paths to any signal chain that is asynchronous or configured for a different sample rate.

Figure 4-16. 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. 4-17. The output paths associated with OUT1–OUT5 are shown in Fig. 4-18. The output paths associated with the SLIMbus interface are shown in Fig. 4-19.

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. 4-17) are located at register addresses R1792–R1983 (0x0700–0x07BF). The OUT1–OUT5 output mixer control fields (see Fig. 4-18) are located at addresses R1664–R1743 (0x0680–0x06CF). The SLIMbus output mixer control fields (see Fig. 4-19) are located at addresses R1984–R2047 (0x07C0–0x07FF).

The full list of digital mixer control registers (R1600–R3576) is provided in Section 6. Generic register field definitions are provided in Table 4-11.

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 4.3.15 and Section 4.3.16.

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 4-26. Note that sample-rate conversion is required when routing the output signal paths to any signal chain that is asynchronous or 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 4-26 for details.

The AIF1, AIF3, SLIMbus, and DAC (OUT1–OUT3) output mixers provide full support for 32-bit data words. Audio data samples up to 32 bits are supported on the AIF1, AIF3, and SLIMbus input channels, which can be routed to the AIF1, AIF3, SLIMbus, and DAC (OUT1–OUT3) output paths. Note that other signal paths and signal-processing blocks within the digital core are limited to 24-bit data length; data samples are truncated to 24-bit length if they are routed through any function that does not support 32-bit data words.

The CS42L92 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–R3576 indicate the status of each digital mixer. If an underclocked error condition occurs, these bits indicate which mixers have been enabled.

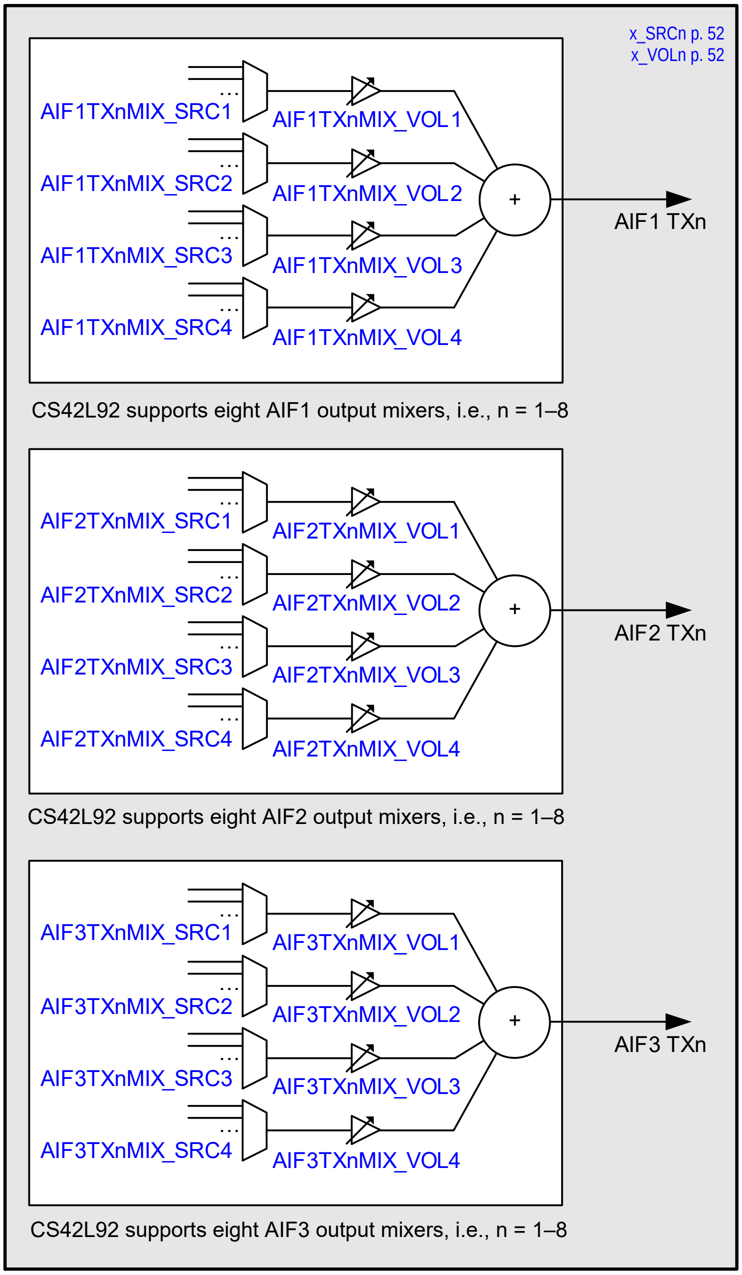


Figure 4-17. Digital-Core AIF Outputs

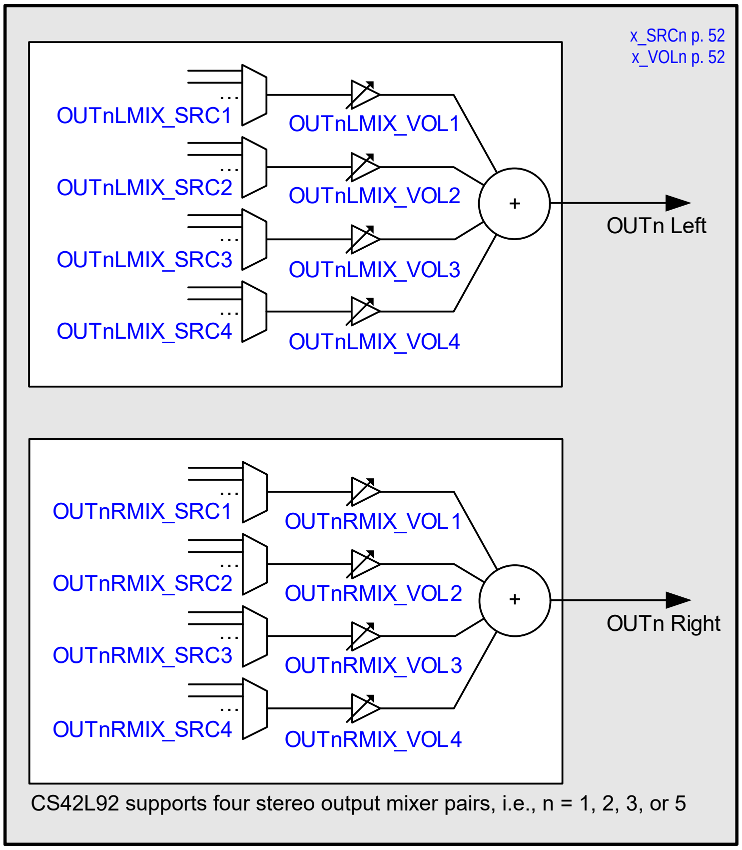


Figure 4-18. Digital-Core OUT*n* Outputs

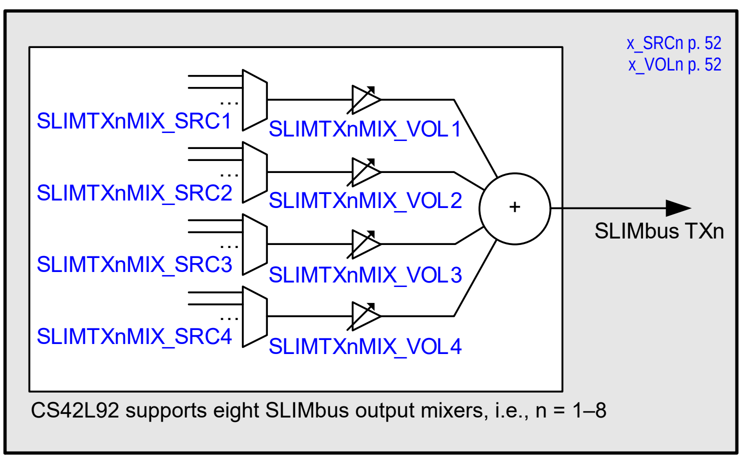


Figure 4-19. Digital-Core SLIMbus Outputs

4.3.4 Five-Band Parametric Equalizer (EQ)

The digital core provides four EQ processing blocks as shown in Fig. 4-20. 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.

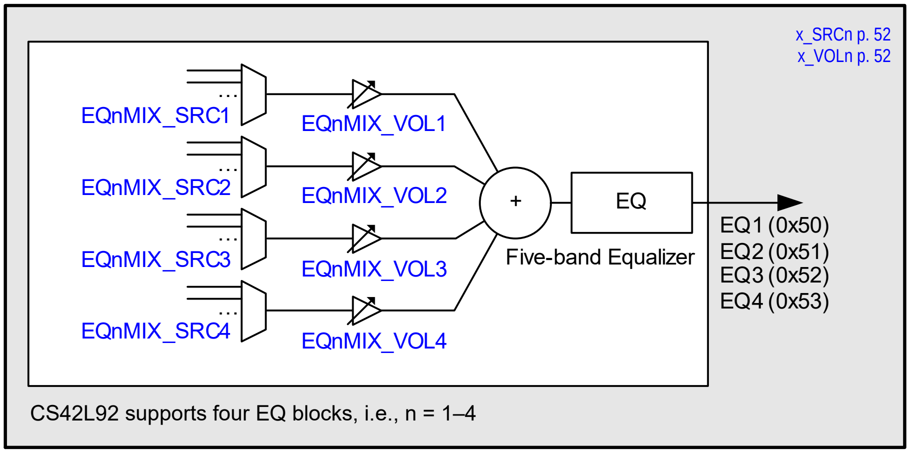


Figure 4-20. Digital-Core EQ Blocks

The EQ1–EQ4 mixer control fields (see Fig. 4-20) are located at register addresses R2176–R2207 (0x0880–0x089F).

The full list of digital-mixer control registers (R1600–R3576) is provided in Section 6. Generic register field definitions are provided in Table 4-11.

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 4.3.15 and Section 4.3.16.

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 sample rate for the EQ function is configured using FX_RATE; see Table 4-26. 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 asynchronous or 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 4-26 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 4-12. 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 4-12. 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 4-13.

Table 4-13. 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 respective signal-processing function. Each bit is coded as follows: 0 = Disabled 1 = Enabled [11] = EQ4 [7] = DRC2 (Right) [3] = LHPF4 [10] = EQ3 [6] = DRC2 (Left) [2] = LHPF3 [9] = EQ2 [5] = DRC1 (Right) [1] = LHPF2 [8] = EQ1 [4] = DRC1 (Left) [0] = LHPF1
R3600 (0x0E10) EQ1_1	15:11	EQ1_B1_GAIN[4:0]	0x0C	EQ1 Band 1 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	10:6	EQ1_B2_GAIN[4:0]	0x0C	EQ1 Band 2 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	5:1	EQ1_B3_GAIN[4:0]	0x0C	EQ1 Band 3 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	0	EQ1_ENA	0	EQ1 Enable 0 = Disabled 1 = Enabled
R3601 (0x0E11) EQ1_2	15:11	EQ1_B4_GAIN[4:0]	0x0C	EQ1 Band 4 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	10:6	EQ1_B5_GAIN[4:0]	0x0C	EQ1 Band 5 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	0	EQ1_B1_MODE	0	EQ1 Band 1 Mode 0 = Shelving filter 1 = Peak filter
R3602 (0x0E12) to R3620 (0x0E24)	15:0	EQ1_B1_* EQ1_B2_* EQ1_B3_* EQ1_B4_* EQ1_B5_*	—	EQ1 Frequency Coefficients. Refer to WISCE evaluation board control software for the derivation of these field values.
R3622 (0x0E26) EQ2_1	15:11	EQ2_B1_GAIN[4:0]	0x0C	EQ2 Band 1 Gain [1] –12 dB to +12 dB in 1-dB steps
	10:6	EQ2_B2_GAIN[4:0]	0x0C	EQ2 Band 2 Gain [1] –12 dB to +12 dB in 1-dB steps
	5:1	EQ2_B3_GAIN[4:0]	0x0C	EQ2 Band 3 Gain [1] –12 dB to +12 dB in 1-dB steps
	0	EQ2_ENA	0	EQ2 Enable 0 = Disabled 1 = Enabled
R3623 (0x0E27) EQ2_2	15:11	EQ2_B4_GAIN[4:0]	0x0C	EQ2 Band 4 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	10:6	EQ2_B5_GAIN[4:0]	0x0C	EQ2 Band 5 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	0	EQ2_B1_MODE	0	EQ2 Band 1 Mode 0 = Shelving filter 1 = Peak filter
R3624 (0x0E28) to R3642 (0x0E3A)	15:0	EQ2_B1_* EQ2_B2_* EQ2_B3_* EQ2_B4_* EQ2_B5_*	—	EQ2 Frequency Coefficients. Refer to WISCE evaluation board control software for the derivation of these field values.
R3644 (0x0E3C) EQ3_1	15:11	EQ3_B1_GAIN[4:0]	0x0C	EQ3 Band 1 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	10:6	EQ3_B2_GAIN[4:0]	0x0C	EQ3 Band 2 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	5:1	EQ3_B3_GAIN[4:0]	0x0C	EQ3 Band 3 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	0	EQ3_ENA	0	EQ3 Enable 0 = Disabled 1 = Enabled
R3645 (0x0E3D) EQ3_2	15:11	EQ3_B4_GAIN[4:0]	0x0C	EQ3 Band 4 Gain[1] (–12 dB to +12 dB in 1-dB steps)
	10:6	EQ3_B5_GAIN[4:0]	0x0C	EQ3 Band 5 Gain [1] (–12 dB to +12 dB in 1-dB steps)
	0	EQ3_B1_MODE	0	EQ3 Band 1 Mode 0 = Shelving filter 1 = Peak filter

Table 4-13. EQ Enable and Gain Control *(Cont.)*

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

1. See Table 4-14 for gain range.

Table 4-14 lists the EQ gain control settings.

Table 4-14. 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 CS42L92 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 EQ, DRC, and LHPF signal path. 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–R3576 indicate the status of each digital mixer. 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. 4-21. 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.

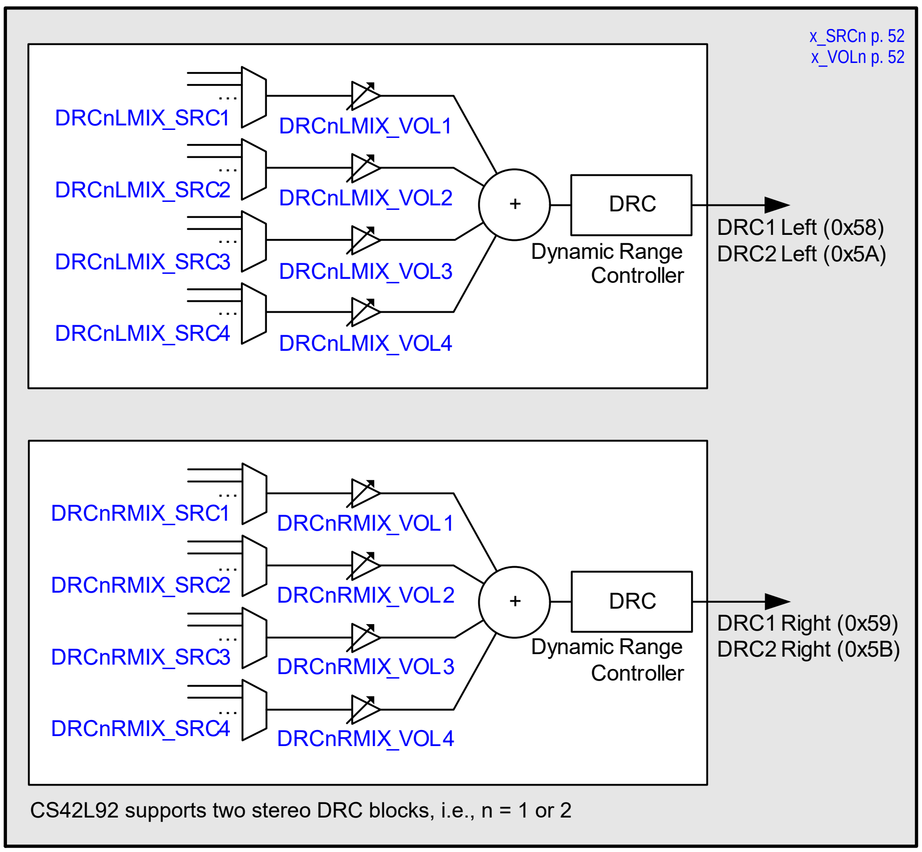


Figure 4-21. Dynamic Range Control (DRC) Block

The DRC1 and DRC2 mixer control fields (see Fig. 4-21) are located at register addresses R2240–R2271 (0x08C0–0x08DF).

The full list of digital mixer control registers (R1600–R3576) is provided in Section 6. Generic register field definitions are provided in Table 4-11.

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 4.3.15 and Section 4.3.16.

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

The sample rate for the DRC function is configured using FX_RATE; see Table 4-26. 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 asynchronous or 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 4-26 for details.

The DRC functions are enabled using the control registers described in Table 4-15.

Table 4-15. 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 DRC. The associated control fields are described in Table 4-17 and Table 4-18 for DRC1 and DRC2 respectively.

4.3.5.1 DRC Compression, Expansion, and Limiting

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

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

For additional attenuation of signals in the noise-gate region, an additional knee can be defined (shown as Knee 2 in Fig. 4-22). 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. 4-22.

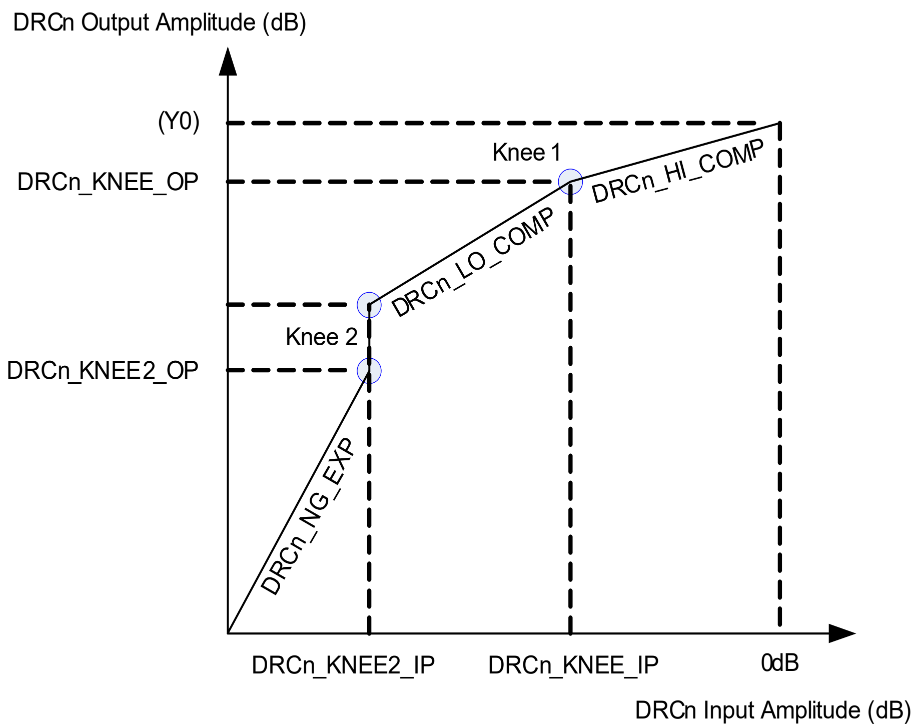


Figure 4-22. DRC Response Characteristic

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. 4-22), 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 4-16.

Table 4-16. 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 4-16) 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. 4-22 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. 4-2.

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

Equation 4-2. 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. 4-22. 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 4-17 and Table 4-18. The register defaults are suitable for general-purpose microphone use.

4.3.5.4 Anticlip Control

The DRC includes an anticlip function to avoid signal clipping when the input amplitude rises very quickly. This function 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 function is enabled using the DRC*n*_ANTICLIP bit. Note that the feed-forward processing increases the latency in the input signal path.

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.

It is recommended to disable the anticlip function if the quick-release function (see Section 4.3.5.5) is enabled.

4.3.5.5 Quick Release Control

The DRC includes a quick-release function 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 function 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 function 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.

It is recommended to disable the quick-release function if the anticlip function (see Section 4.3.5.4) is enabled.

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 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 4-17.

Table 4-17. 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 respective signal-processing function. 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

Table 4-17. DRC1 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
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
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

Table 4-17. DRC1 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description		
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	11X = Reserved
				001 = 1/2	100 = 1/16	
				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 4-18.

Table 4-18. 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 respective signal-processing function. 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

Table 4-18. DRC2 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
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) 0x1E = –75 dB 0x01 = –31.5 dB 0x1F = –76.5 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 4-18. DRC2 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R3722 (0x0E8A) DRC2_ctrl3	15:12	DRC2_NG_ MINGAIN[3:0]	0000	DRC2 Minimum gain to attenuate audio signals when the Noise Gate is active. 0000 = –36 dB 0101 = –6 dB 1010 = 24 dB 0001 = –30 dB 0110 = 0 dB 1011 = 30 dB 0010 = –24 dB 0111 = 6 dB 1100 = 36 dB 0011 = –18 dB 1000 = 12 dB 1101 to 1111 = Reserved 0100 = –12 dB 1001 = 18 dB
	11:10	DRC2_NG_ EXP[1:0]	00	DRC2 Noise-Gate slope 00 = 1 (no expansion) 10 = 4 01 = 2 11 = 8
	9:8	DRC2_QR_ THR[1:0]	00	DRC2 Quick-release threshold (crest factor in dB) 00 = 12 dB 10 = 24 dB 01 = 18 dB 11 = 30 dB
	7:6	DRC2_QR_ DCY[1:0]	00	DRC2 Quick-release decay rate (seconds/6 dB) 00 = 0.725 ms 10 = 5.8 ms 01 = 1.45 ms 11 = Reserved
	5:3	DRC2_HI_ COMP[2:0]	011	DRC2 Compressor slope (upper region) 000 = 1 (no compression) 011 = 1/8 11X = Reserved 001 = 1/2 100 = 1/16 010 = 1/4 101 = 0
	2:0	DRC2_LO_ COMP[2:0]	000	DRC2 Compressor slope (lower region) 000 = 1 (no compression) 011 = 1/8 11X = Reserved 001 = 1/2 100 = 0 010 = 1/4 101 = Reserved
R3723 (0x0E8B) DRC2_ctrl4	10:5	DRC2_KNEE_ IP[5:0]	000000	DRC2 Input signal level at the compressor knee. 0x00 = 0 dB 0x02 = –1.5 dB 0x3C = –45 dB 0x01 = –0.75 dB … (–0.75-dB steps) 0x3D–0x3F = Reserved
	4:0	DRC2_KNEE_ OP[4:0]	00000	DRC2 Output signal at the compressor knee. 0x00 = 0 dB 0x02 = –1.5 dB 0x1E = –22.5 dB 0x01 = –0.75 dB … (–0.75 dB steps) 0x1F = Reserved
R3724 (0x0E8C) DRC2_ctrl5	9:5	DRC2_KNEE2_ IP[4:0]	00000	DRC2 Input signal level at the noise-gate threshold, Knee 2. 0x00 = –36 dB 0x02 = –39 dB 0x1E = –81 dB 0x01 = –37.5 dB … (-1.5-dB steps) 0x1F = –82.5 dB Applicable only if DRC2_NG_ENA = 1.
	4:0	DRC2_KNEE2_ OP[4:0]	00000	DRC2 Output signal at the noise-gate threshold, Knee 2. 0x00 = –30 dB 0x02 = –33 dB 0x1E = –75 dB 0x01 = –31.5 dB … (–1.5dB steps) 0x1F = –76.5 dB Applicable only if DRC2_KNEE2_OP_ENA = 1.

The CS42L92 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 EQ, DRC, and LHPF signal path. 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–R3576 indicate the status of each digital mixer. 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. 4-23. 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).

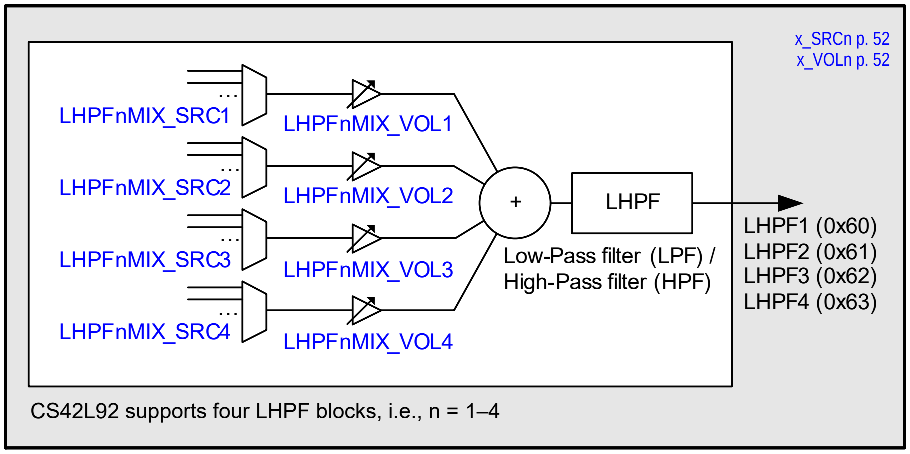


Figure 4-23. Digital-Core LPF/HPF Blocks

The LHPF1–LHPF4 mixer control fields, shown in Fig. 4-23, are located at register addresses R2304–R2335 (0x0900–0x091F).

The full list of digital mixer control registers (R1600–R3576) is provided in Section 6. Generic register field definitions are provided in Table 4-11.

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 4.3.15 and Section 4.3.16.

The hexadecimal numbers in Fig. 4-23 indicate the corresponding x_SRC*n* setting for selection of that signal as an input to another digital-core function.

The sample rate for the LHPF function is configured using FX_RATE; see Table 4-26. 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 asynchronous or 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 4-26 for details.

The control registers associated with the LHPF functions are described in Table 4-19.

The cut-off frequencies for the LHPF blocks are set by 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 4-19. 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

Table 4-19. Low-Pass Filter/High-Pass Filter *(Cont.)*

Register Address	Bit	Label	Default	Description
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 CS42L92 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 EQ, DRC, and LHPF signal path. 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–R3576 indicate the status of each digital mixer. If an underclocked error condition occurs, these bits indicate which mixers have been enabled.

4.3.7 Digital-Core DSP

The digital core provides one programmable DSP processing block as shown in Fig. 4-24. The DSP block supports eight inputs (Left, Right, Aux1, Aux2, … Aux6). A four-input mixer is associated with the left and right inputs, providing further expansion of the number of input paths. Each of the input sources is selectable, and independent volume control is provided for left and right input mixer channels. The DSP block supports six outputs.

The functionality of the DSP processing block is not fixed; application-specific algorithms can be implemented according to different customer requirements. The procedure for configuring the CS42L92 DSP functions is tailored to each customer's application; please contact your Cirrus Logic representative for details.

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

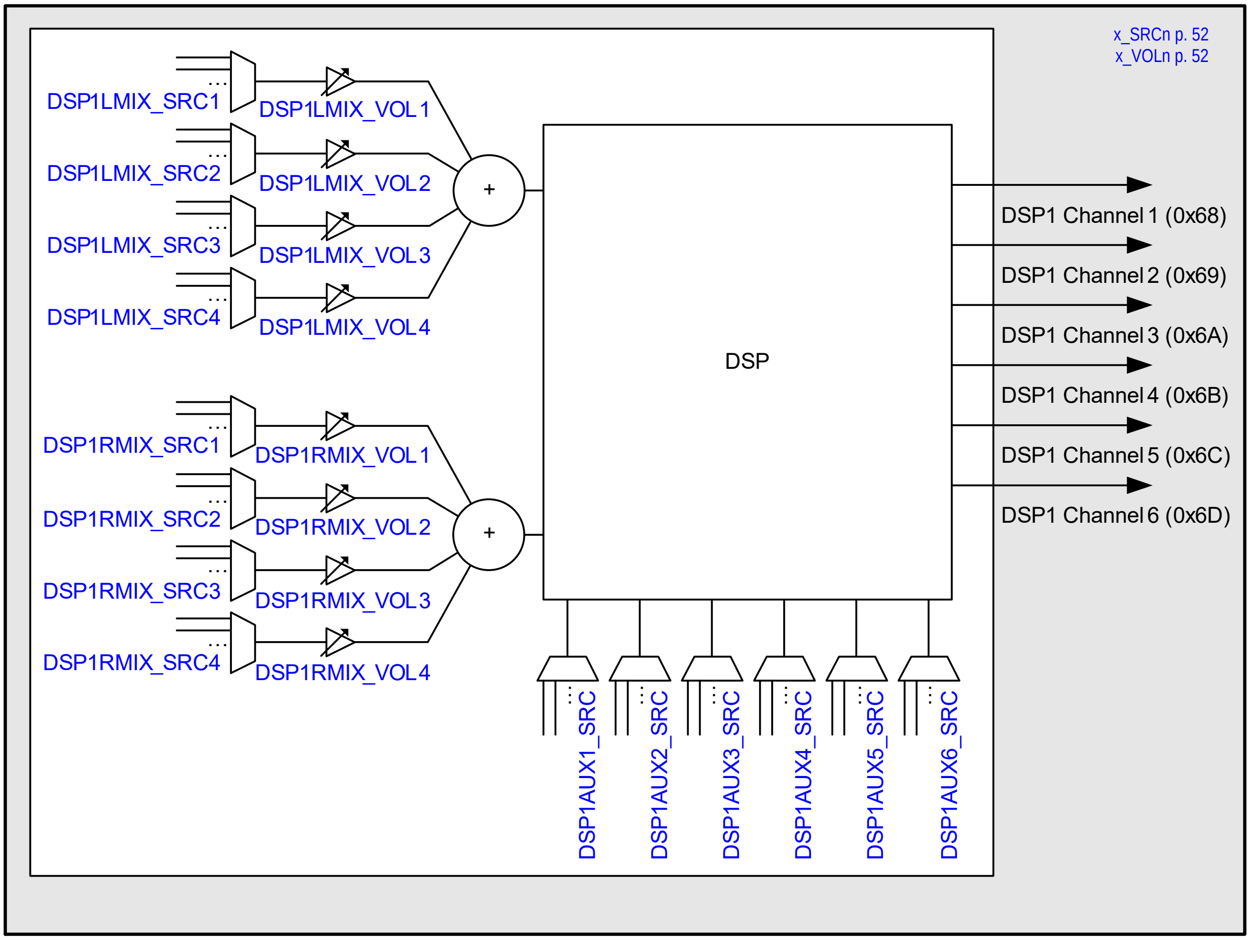


Figure 4-24. Digital-Core DSP Block

The DSP mixer input control fields (see Fig. 4-24) are located at register addresses R2368–R2424 (0x0940–0x0978).

The full list of digital mixer control registers (R1600–R3576) is provided in Section 6. Generic register field definitions are provided in Table 4-11.

The x_SRC*n* fields select the input sources for the DSP processing block. Note that the selected input sources must be configured for the same sample rate as the DSP. Sample-rate conversion functions are available to support flexible interconnectivity; see Section 4.3.16.

The hexadecimal numbers in Fig. 4-24 indicate the corresponding x_SRC*n* setting for selection of that signal as an input to another digital-core function.

The sample rate for the DSP functions is configured using the DSP1_RATE field; see Table 4-26. Sample-rate conversion is required when routing the DSP signal paths to any signal chain that is configured for a different sample rate.

The DSP1_RATE field must not be changed if any of the respective x_SRC*n* fields is nonzero. The associated x_SRC*n* fields must be cleared before writing new values to DSP1_RATE. A minimum delay of 125 µs must be allowed between clearing the x_SRC*n* fields and writing to the DSP1_RATE field. See Table 4-26 for details.

The CS42L92 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–R3576 indicate the status of each digital mixer. If an underclocked error condition occurs, these bits indicate which mixers have been enabled.

4.3.8 S/PDIF Output Generator

The CS42L92 incorporates an IEC-60958-3–compatible S/PDIF output generator, as shown in Fig. 4-25; 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 CS42L92 digital core.

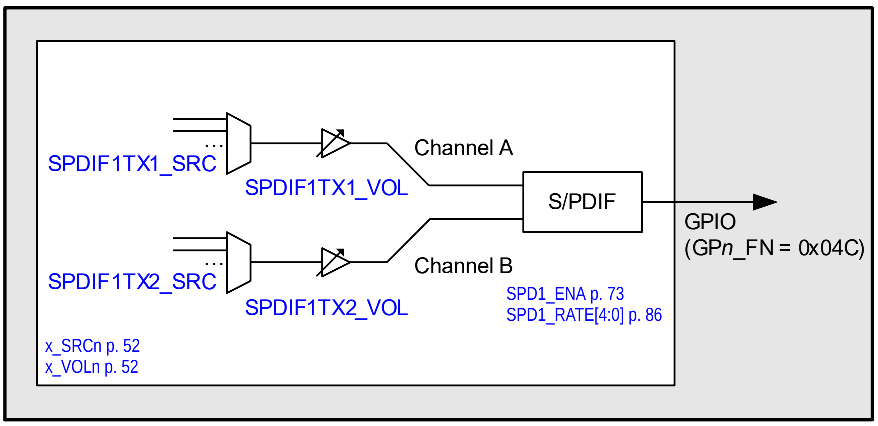


Figure 4-25. Digital-Core S/PDIF Output Generator

The S/PDIF input control fields (see Fig. 4-25) are located at register addresses R2048–R2057 (0x0800–0x0809).

The full list of digital mixer control registers (R1600–R3576) is provided in Section 6. Generic register field definitions are provided in Table 4-11.

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 4.3.15 and Section 4.3.16.

The sample rate of the S/PDIF generator is configured using SPD1_RATE; see Table 4-26. 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 asynchronous or 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 4-26 for details.

The S/PDIF generator is enabled by setting SPD1_ENA, as described in Table 4-20.

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 (IEC60958-3 Digital Audio Interface - Consumer) for full details of the S/PDIF protocol and configuration parameters.

Table 4-20. 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 S/PDIF output generator provides full support for 32-bit data words. Audio data samples up to 32 bits are supported on the AIF1, AIF3, and SLIMbus input channels, which can be routed to the S/PDIF output. Note that other signal paths and signal-processing blocks within the digital core are limited to 24-bit data length; data samples are truncated to 24-bit length if they are routed through any function that does not support 32-bit data words.

The CS42L92 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–R3576 indicate the status of each digital mixer. If an underclocked error condition occurs, these bits indicate which mixers have been enabled.

4.3.9 Tone Generator

The CS42L92 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.

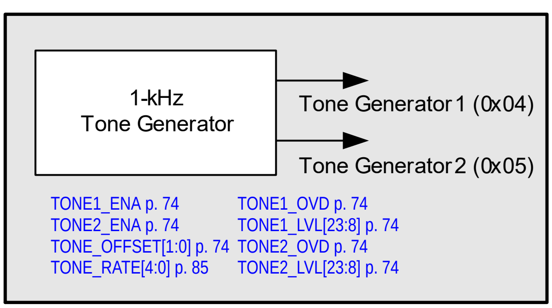


Figure 4-26. Digital-Core Tone Generator

The tone generator outputs can be selected as input to any of the digital mixers or signal-processing functions within the CS42L92 digital core. The hexadecimal numbers in Fig. 4-26 indicate the corresponding x_SRC*n* setting for selection of that signal as an input to another digital-core function.

The sample rate for the tone generator is configured using TONE_RATE. See Table 4-26. Note that sample-rate conversion is required when routing the tone generator outputs to any signal chain that is asynchronous or configured for a different sample rate.

The tone generator outputs are enabled by setting the TONE1_ENA and TONE2_ENA bits as described in Table 4-21. The phase relationship is configured using TONE_OFFSET.

The tone generator outputs can also provide a configurable DC signal level, for use as a test signal. The DC output is selected using the TONE*n*_OVD bits, and the DC signal amplitude is configured using the TONE*n*_LVL fields, as described in Table 4-21.

Table 4-21. 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° (in phase) 01 = 90° ahead 10 = 180° ahead 11 = 270° ahead
	5	TONE2_ OVD	0	Tone Generator 2 Override 0 = Disabled (1-kHz tone output) 1 = Enabled (DC signal output) The DC signal level, when selected, is configured using TONE2_LVL[23:0]
	4	TONE1_ OVD	0	Tone Generator 1 Override 0 = Disabled (1-kHz tone output) 1 = Enabled (DC signal output) The DC signal level, when selected, is configured using TONE1_LVL[23:0]
	1	TONE2_ENA	0	Tone Generator 2 Enable 0 = Disabled 1 = Enabled
	0	TONE1_ENA	0	Tone Generator 1 Enable 0 = Disabled 1 = Enabled
R33 (0x0021) Tone_Generator_2	15:0	TONE1_ LVL[23:8]	0x1000	Tone Generator 1 DC output level TONE1_LVL[23:8] is coded as 2’s complement. Bits [23:20] contain the integer portion; bits [19:0] contain the fractional portion. The digital core 0 dBFS level corresponds to 0x10_0000 (+1) or 0xF0_0000 (–1).
R34 (0x0022) Tone_Generator_3	7:0	TONE1_ LVL[7:0]	0x00	Tone Generator 1 DC output level TONE1_LVL[23:8] is coded as 2’s complement. Bits [23:20] contain the integer portion; bits [19:0] contain the fractional portion. The digital core 0 dBFS level corresponds to 0x10_0000 (+1) or 0xF0_0000 (–1).
R35 (0x0023) Tone_Generator_4	15:0	TONE2_ LVL[23:8]	0x1000	Tone Generator 2 DC output level TONE2_LVL[23:8] is coded as 2’s complement. Bits [23:20] contain the integer portion; bits [19:0] contain the fractional portion. The digital core 0 dBFS level corresponds to 0x10_0000 (+1) or 0xF0_0000 (–1).
R36 (0x0024) Tone_Generator_5	7:0	TONE2_ LVL[7:0]	0x00	Tone Generator 2 DC output level TONE2_LVL[23:8] is coded as 2’s complement. Bits [23:20] contain the integer portion; bits [19:0] contain the fractional portion. The digital core 0 dBFS level corresponds to 0x10_0000 (+1) or 0xF0_0000 (–1).

4.3.10 Noise Generator

The CS42L92 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.

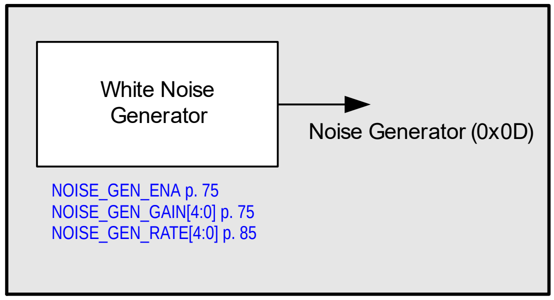


Figure 4-27. Digital-Core Noise Generator

The noise generator can be selected as input to any of the digital mixers or signal-processing functions within the CS42L92 digital core. The hexadecimal number (0x0D) in Fig. 4-27 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 NOISE_GEN_RATE. See Table 4-26. Note that sample-rate conversion is required when routing the noise generator output to any signal chain that is asynchronous or configured for a different sample rate.

The noise generator is enabled by setting NOISE_GEN_ENA, described in Table 4-22. The signal level is configured using NOISE_GEN_GAIN.

Table 4-22. 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 CS42L92 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 CS42L92. In a typical use case the haptic signal may be routed, via one of the digital-core output mixers, to the digital PDM output. An external amplifier can be used to drive the haptic device, as shown in Fig. 4-28.

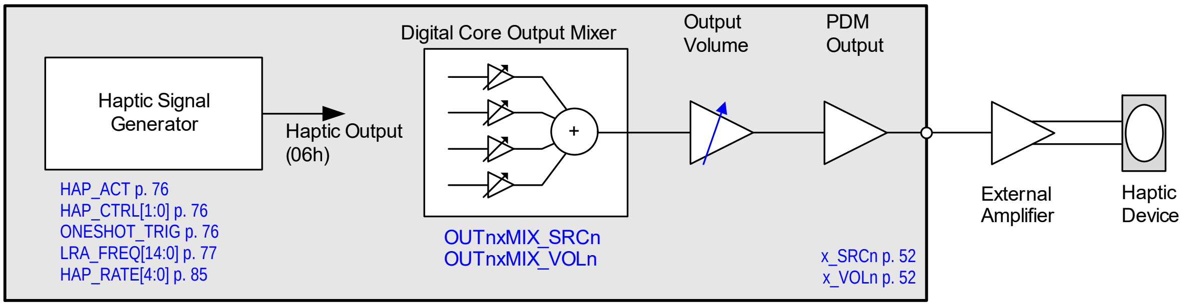


Figure 4-28. Digital-Core Haptic Signal Generator

The hexadecimal number (0x06) in Fig. 4-28 indicates the corresponding x_SRC*n* setting for selection of the haptic signal generator as an input to another digital-core function.

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

The sample rate for the haptic signal generator is configured using the HAP_RATE field. See Table 4-26. Note that sample-rate conversion is required when routing the haptic signal generator output to any signal chain that is asynchronous or 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 4-23. 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 4-23. 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)

Table 4-23. Haptic Signal Generator Control *(Cont.)*

Register Address	Bit	Label	Default	Description
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 or ASYNCCLK. If HAP_RATE < 1000, SYSCLK is the clock source, and the applicable System Clock frequency is determined by SYSCLK. If HAP_RATE ≥ 1000, ASYNCCLK is the clock source, and the applicable System Clock frequency is determined by ASYNCCLK. Valid for haptic frequency in the range 100–250 Hz For 6.144-MHz System Clock: 0x77FF = 100 Hz; 0x4491 = 175 Hz; 0x2FFF = 250 Hz For 5.6448-MHz System Clock: 0x6E3F = 100 Hz; 0x3EFF = 175 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 0x001 = 0.625 ms 0x002 = 1.25 ms … (0.625-ms steps) 0x1FF = 319.375 ms
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 0x001 = 0.625 ms 0x002 = 1.25 ms … (0.625-ms steps) 0x7FF = 1279.375 ms
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 0x001 = 0.625 ms 0x002 = 1.25 ms … (0.625-ms steps) 0x1FF = 319.375 ms
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 CS42L92 incorporates two PWM signal generators as shown in Fig. 4-29. 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 CS42L92 digital core.

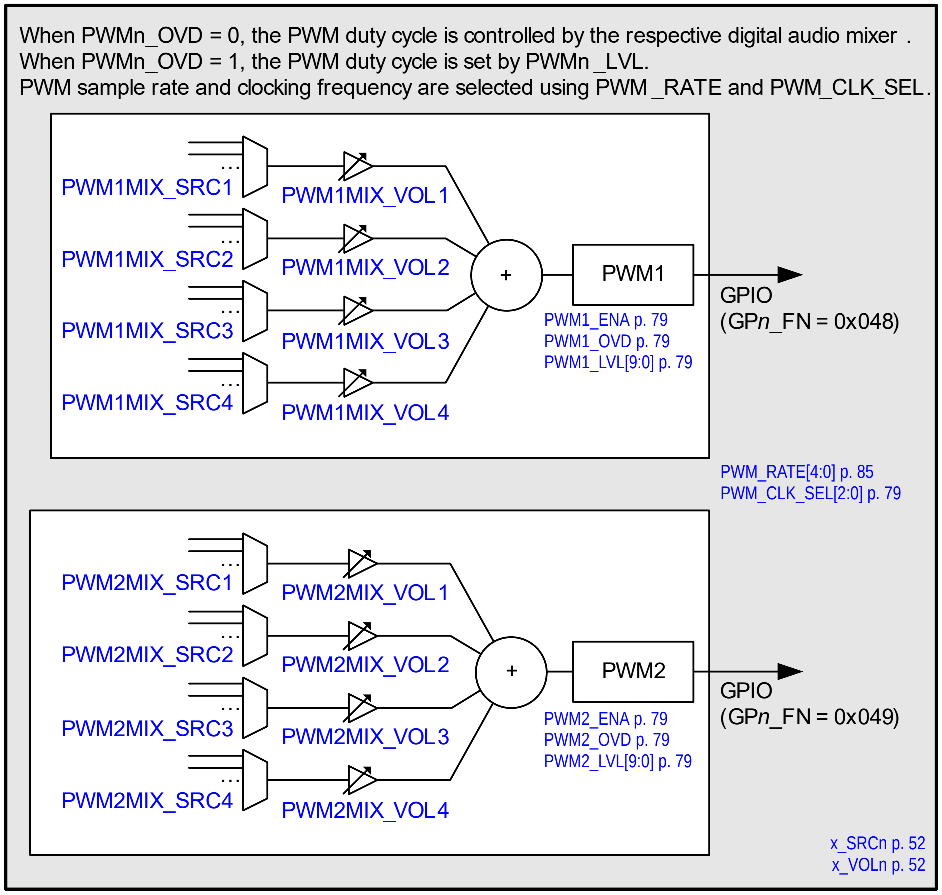


Figure 4-29. Digital-Core PWM Generator

The PWM1 and PWM2 mixer control fields (see Fig. 4-29) are located at register addresses R1600–R1615 (0x0640–0x064F).

The full list of digital mixer control registers (R1600–R3576) is provided in Section 6. Generic register field definitions are provided in Table 4-11.

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 4.3.15 and Section 4.3.16.

The PWM sample rate (cycle time) is configured using PWM_RATE. See Table 4-26. Note that sample-rate conversion is required when linking the PWM generators to any signal chain that is asynchronous or 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 4-26 for details.

The PWM generators are enabled by setting PWM1_ENA and PWM2_ENA, respectively, as described in Table 4-24.

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. 4-29.

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 (if PWM_RATE < 1000) or ASYNCCLK (if PWM_RATE ≥ 1000).

Table 4-24. 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 (if PWM_RATE < 1000) or less than or equal to ASYNCCLK (if PWM_RATE ≥ 1000).
	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 CS42L92 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–R3576 indicate the status of each digital mixer. If an underclocked error condition occurs, these bits indicate which mixers have been enabled.

4.3.13 Data Format Conversion

The digital mixing and signal-processing functions on the CS42L92 are designed to route audio data in signed fixed point format. Data format converter (DFC) blocks are incorporated in the digital core, with the capability to convert audio data between signed, unsigned, and floating-point formats. The DFCs enable the flexibility to support many different interface standards on the input and output signal paths. They can also be used to apply dithering to digital audio data.

The digital core provides eight DFC blocks as shown in Fig. 4-30. Each DFC supports one input and one output path only.

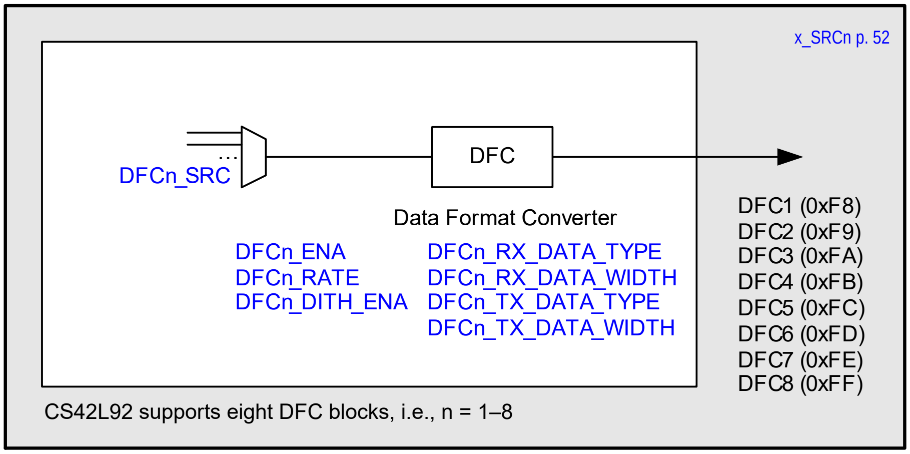


Figure 4-30. Digital-Core DFC Blocks

The DFC1–DFC8 input control fields (see Fig. 4-30) are located at register addresses R3520–R3576 (0x0DC0–0x0DF8).

The full list of digital-mixer control registers (R1600–R3576) is provided in Section 6. Generic register field definitions are provided in Table 4-11.

The x_SRC*n* fields select the input sources for the respective DFCs. Note that the selected input sources must be configured for the same sample rate as the DFC to which they are connected. Sample-rate conversion functions are available to support flexible interconnectivity; see Section 4.3.15 and Section 4.3.16.

The hexadecimal numbers in Fig. 4-30 indicate the corresponding x_SRC*n* setting for selection of that signal as an input to another digital-core function.

The sample rate for each converter DFC*n* is configured using the respective DFC*n*_RATE field; see Table 4-26. Note that sample-rate conversion is required when routing the DFC paths to any signal chain that is asynchronous or configured for a different sample rate.

The DFC*n*_RATE fields must not be changed if the associated x_SRC*n* field is nonzero. The associated x_SRC*n* field must be cleared before writing a new value to DFC*n*_RATE. A minimum delay of 125 µs must be allowed between clearing the x_SRC*n* field and writing to the associated DFC*n*_RATE field. See Table 4-26 for details.

The DFC is enabled by setting DFC*n*_ENA.

The input data format is configured using the DFC*n*_RX_DATA_TYPE and DFC*n*_RX_DATA_WIDTH fields. Valid data types are signed fixed point, unsigned fixed point, and three different floating point configurations. If a fixed point data type is selected, the data width (number of data bits) is selected using DFC*n*_RX_DATA_WIDTH.

The DFC input can be any of the digital core inputs or signal processing blocks. If the input data type is unsigned or floating point format, one of the AIF or SLIMbus RX channels must be selected as the DFC input source—unsigned and floating point data types are not valid selections with any other source within the digital core.

The output data format is configured using the DFC*n*_TX_DATA_TYPE and DFC*n*_TX_DATA_WIDTH fields. Valid data types are signed fixed point, unsigned fixed point, and floating point formats. If a fixed point data type is selected, the data width (number of data bits) is selected using DFC*n*_TX_DATA_WIDTH.

The DFC output can be selected as input to any of the digital mixers or signal-processing functions within the CS42L92 digital core. If the DFC output data type is unsigned fixed point or floating point format, it must be routed directly to the AIF or SLIMbus TX channels using the respective digital-core output mixers—unsigned fixed point or floating point data is not valid as input to any other digital core functions.

Note: If unsigned fixed point or floating point data is routed from a DFC output to an AIF or SLIMbus TX channel, the DFC must be the only enabled signal path in the respective output mixer, and the associated volume selection must be 0 dB.

The DFC can apply dithering to its output data; this is enabled by setting DFC*n*_DITH_ENA. Note that dithering is only valid if the output data is in fixed point format (signed or unsigned). If the output data type is floating point, the DFC*n*_DITH_ENA bit must be cleared.

The dither function can be used to improve the noise characteristics of signals routed in the digital core. Dithering is particularly recommended if truncating audio data (e.g., from 32- to 24-bit format) as it converts the truncation/quantization errors into benign background noise.

The DFCs provide input to the interrupt control circuit and can be used to trigger an interrupt event if saturation (arithmetic error) is detected; see Section 4.15.

The control registers associated with the DFCs are described in Table 4-25.

Table 4-25. Digital-Core DFC Control

Register Address	Bit	Label	Default	Description
R5248 (0x1480) DFC1_CTRL_W0 R5254 (0x1486) DFC2_CTRL_W0 R5260 (0x148C) DFC3_CTRL_W0 R5266 (0x1492) DFC4_CTRL_W0 R5272 (0x1498) DFC5_CTRL_W0 R5278 (0x149E) DFC6_CTRL_W0 R5284 (0x14A4) DFC7_CTRL_W0 R5290 (0x14AA) DFC8_CTRL_W0	1	DFC*n*_DITH_ENA	0	DFC*n* dither enable (valid for fixed point output data only) 0 = Disabled 1 = Enabled
	0	DFC*n*_ENA	0	DFC*n* enable 0 = Disabled 1 = Enabled
R5250 (0x1482) DFC1_RX_W0 R5256 (0x1488) DFC2_RX_W0 R5262 (0x148E) DFC3_RX_W0 R5268 (0x1494) DFC4_RX_W0 R5274 (0x149A) DFC5_RX_W0 R5280 (0x14A0) DFC6_RX_W0 R5286 (0x14A6) DFC7_RX_W0 R5292 (0x14AC) DFC8_RX_W0	12:8	DFC*n*_RX_DATA_WIDTH[4:0]	0x1F	DFC*n* input data width (valid for fixed point data types only) 0x00 to 0x06 = Reserved 0x07 = 8 bits 0x08 = 9 bits 0x09 = 10 bits ... 0x1F = 32 bits
	2:0	DFC*n*_RX_DATA_TYPE[2:0]	000	DFC*n* input data type 000 = Signed, fixed point 001 = Unsigned, fixed point 010 = Single-precision floating point (binary32) 100 = Half-precision floating point (binary16) 101 = ARM-alternative half-precision floating point All other codes are reserved.
R5252 (0x1488) DFC1_TX_W0 R5258 (0x148A) DFC2_TX_W0 R5264 (0x1490) DFC3_TX_W0 R5270 (0x1494) DFC4_TX_W0 R5276 (0x149C) DFC5_TX_W0 R5282 (0x14A2) DFC6_TX_W0 R5288 (0x14A8) DFC7_TX_W0 R5294 (0x14AE) DFC8_TX_W0	12:8	DFC*n*_TX_DATA_WIDTH[4:0]	0x1F	DFC*n* output data width (valid for fixed point data types only) 0x00 to 0x06 = Reserved 0x07 = 8 bits 0x08 = 9 bits 0x09 = 10 bits ... 0x1F = 32 bits
	2:0	DFC*n*_TX_DATA_TYPE[2:0]	000	DFC*n* output data type 000 = Signed, fixed point 001 = Unsigned, fixed point 010 = Single-precision floating point (binary32) 100 = Half-precision floating point (binary16) 101 = ARM-alternative half-precision floating point All other codes are reserved.

The CS42L92 automatically checks to confirm whether the SYSCLK frequency is high enough to support the commanded DFC and digital mixing functions. If an attempt is made to enable DFC 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 status bits in registers R1600–R3576 indicate the status of each digital mixer. If an underclocked error condition occurs, these bits indicate which mixers have been enabled.

4.3.14 Sample-Rate Control

The CS42L92 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 and/or referenced to asynchronous clock domains.

Two independent clock domains are supported for the audio signal paths, referenced to SYSCLK and ASYNCCLK respectively, as described in Section 4.16. Every digital signal path must be synchronized either to SYSCLK or to ASYNCCLK.

Up to five different sample rates may be in use at any time on the CS42L92. Three of these sample rates must be synchronized to SYSCLK; the remaining two, where required, must be synchronized to ASYNCCLK.

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

The asynchronous sample-rate converter (ASRC) supports two-way stereo conversion paths between the SYSCLK and ASYNCCLK domains. The ASRC is described in Section 4.3.15.

There are two isochronous sample-rate converters (ISRCs). Each ISRC supports two-way, two-channel conversion paths between sample rates on the SYSCLK domain, or between sample rates on the ASYNCCLK domain. The ISRCs are described in Section 4.3.16.

The sample rate of different blocks within the CS42L92 digital core are controlled as shown in Fig. 4-31. 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 4-26 for details.

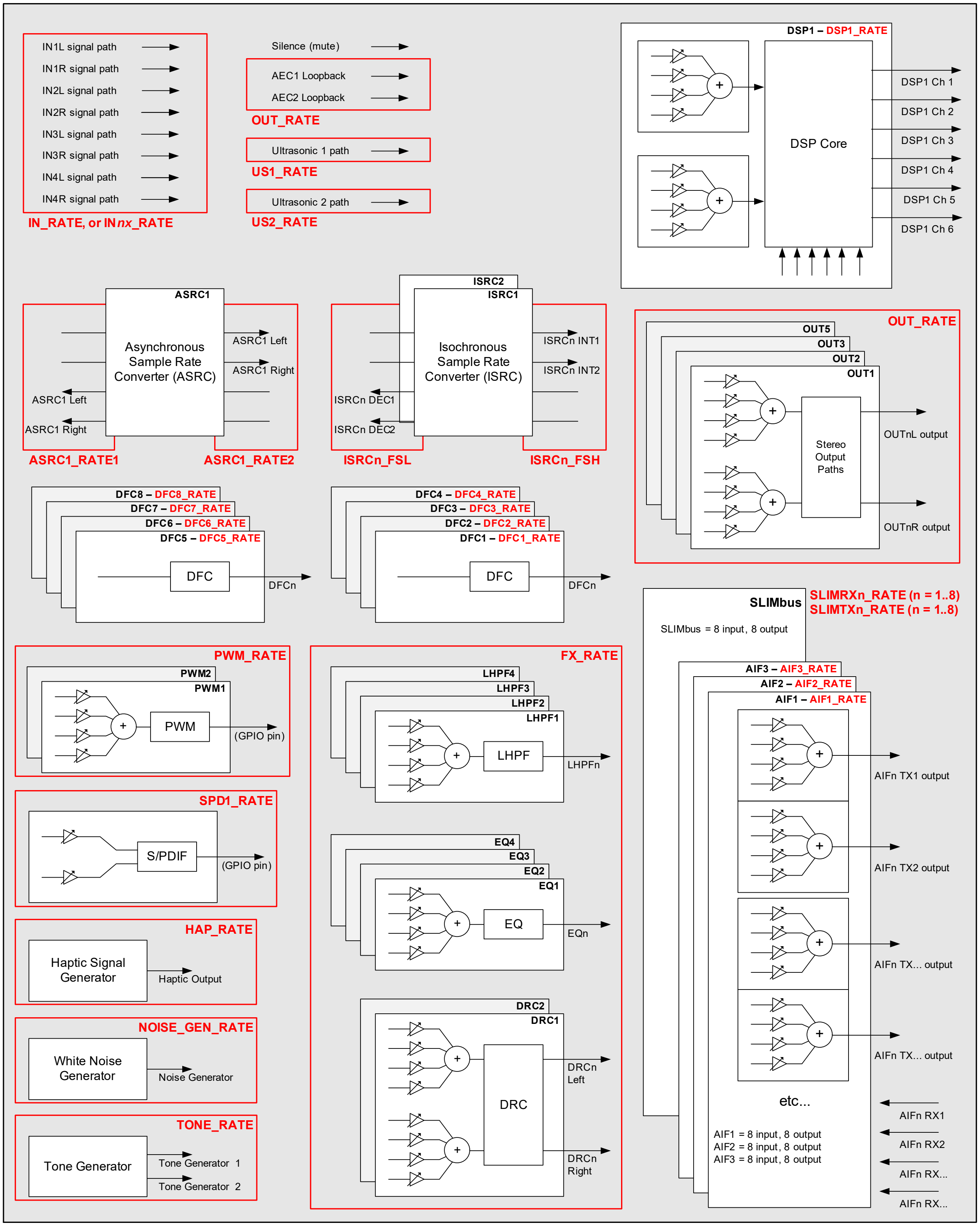


Figure 4-31. Digital-Core Sample-Rate Control

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 can either be set globally (using IN_RATE), or can be configured independently for each input channel (using the respective IN*nx*_RATE fields). The applicable mode depends on IN_RATE_MODE, as described in

Table 4-3.

The ultrasonic demodulator circuits can be selected as input to the digital mixers or signal-processing functions. The sample rate for these signals are configured using US1_RATE and US2_RATE. The selected sample rate must be equal to the output rate of the demodulator function, set by the respective US*n*_FREQ field—see Section 4.2.9.

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. Clocking for the DACs and output signal path circuits must also be configured using the OUT_CLK_SRC field—see Section 4.11.2.

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 eight input channels and eight output channels. The sample rate of each channel can be configured independently, using SLIMTX*n*_RATE and SLIMRX*n*_RATE.

Note that the SLIMbus interface provides simultaneous support for SYSCLK-referenced and ASYNCCLK-referenced sample rates on different channels. For example, 48-kHz and 44.1-kHz SLIMbus audio paths can be simultaneously supported.

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

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

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

The tone generator 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. In a typical use case the haptic signal may be routed, via one of the digital-core output mixers, to the digital PDM output (OUT5). 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 DFCn blocks can be enabled in signal paths within the digital core. The applicable sample rates are configured using the DFC*n*_RATE fields (where *n* identifies the applicable DFC block).

The sample-rate control registers are described in Table 4-26. Refer to the field descriptions for details of the valid selections in each case—note that the input (ADC) and output (DAC) signal paths must always be associated with the SYSCLK clocking domain; different sample rates may be selected concurrently, but both these rates must be synchronized to SYSCLK.

The control registers associated with the ASRC and ISRCs are described in Table 4-27 and Table 4-28.

Note that 32-bit register addressing is used from R12888 (0x3000) upwards; 16-bit format is used otherwise. The registers noted in Table 4-26 contain a mixture of 16-bit and 32-bit register addresses.

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

Register Address	Bit	Label	Default	Description
R32 (0x0020) Tone_Generator_1	15:11	TONE_RATE[4:0]	0x00	Tone Generator Sample Rate 0x00 = SAMPLE_RATE_1 0x01 = SAMPLE_RATE_2 0x02 = SAMPLE_RATE_3 0x08 = ASYNC_SAMPLE_RATE_1 0x09 = ASYNC_SAMPLE_RATE_2 All other codes are reserved. The selected sample rate is valid in the range 8–192 kHz.
R48 (0x0030) PWM_Drive_1	15:11	PWM_RATE[4:0]	0x00	PWM Frequency (sample rate) 0x00 = SAMPLE_RATE_1 0x01 = SAMPLE_RATE_2 0x02 = SAMPLE_RATE_3 0x08 = ASYNC_SAMPLE_RATE_1 0x09 = ASYNC_SAMPLE_RATE_2 All other codes are reserved. The selected sample rate is valid in the range 8–192 kHz. All PWM*n*MIX_SRC*m* fields must be cleared before changing PWM_RATE.
R144 (0x0090) Haptics_Control_1	15:11	HAP_RATE[4:0]	0x00	Haptic Signal Generator Sample Rate 0x00 = SAMPLE_RATE_1 0x01 = SAMPLE_RATE_2 0x02 = SAMPLE_RATE_3 0x08 = ASYNC_SAMPLE_RATE_1 0x09 = ASYNC_SAMPLE_RATE_2 All other codes are reserved. The selected sample rate is valid in the range 8–192 kHz.
R160 (0x00A0) Comfort_Noise_ Generator	15:11	NOISE_GEN_ RATE[4:0]	0x00	Noise Generator Sample Rate 0x00 = SAMPLE_RATE_1 0x01 = SAMPLE_RATE_2 0x02 = SAMPLE_RATE_3 0x08 = ASYNC_SAMPLE_RATE_1 0x09 = ASYNC_SAMPLE_RATE_2 All other codes are reserved. The selected sample rate is valid in the range 8–192 kHz.
R776 (0x0308) Input_Rate	15:11	IN_RATE[4:0]	0x00	Input Signal Paths Sample Rate (only valid if IN_RATE_MODE = 0) 0x00 = SAMPLE_RATE_1 0x01 = SAMPLE_RATE_2 0x02 = SAMPLE_RATE_3 All other codes are reserved. The selected sample rate is valid in the range 8–192 kHz. If 384 kHz/768 kHz DMIC rate is selected on any of the input paths (INn_ OSR = 01X), the input paths sample rate is valid up to 48 kHz/96 kHz respectively.

Table 4-26. Digital-Core Sample-Rate Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R787 (0x0313) IN1L_Rate_ Control	15:11	IN1L_RATE[4:0]	0x00	Input Path *n* (Left/Right) Sample Rate (only valid if IN_RATE_MODE = 1) 0x00 = SAMPLE_RATE_1 0x01 = SAMPLE_RATE_2 0x02 = SAMPLE_RATE_3 All other codes are reserved. The selected sample rate is valid in the range 8–192 kHz. If 384 kHz/768 kHz DMIC rate is selected (IN*n*_OSR = 01X), the IN*n*L/IN*n*R sample rate is valid up to 48 kHz/96 kHz respectively.
R791 (0x0317) IN1R_Rate_ Control	15:11	IN1R_RATE[4:0]	0x00	
R795 (0x031B) IN2L_Rate_ Control	15:11	IN2L_RATE[4:0]	0x00	
R799 (0x031F) IN2R_Rate_ Control	15:11	IN2R_RATE[4:0]	0x00	
R803 (0x0323) IN3L_Rate_ Control	15:11	IN3L_RATE[4:0]	0x00	
R807 (0x0327) IN3R_Rate_ Control	15:11	IN3R_RATE[4:0]	0x00	
R811 (0x032B) IN4L_Rate_ Control	15:11	IN4L_RATE[4:0]	0x00	
R815 (0x032F) IN4R_Rate_ Control	15:11	IN4R_RATE[4:0]	0x00	
R1032 (0x0408) Output_Rate_1	15:11	OUT_RATE[4:0]	0x00	Output Signal Paths Sample Rate 0x00 = SAMPLE_RATE_1 0x01 = SAMPLE_RATE_2 0x02 = SAMPLE_RATE_3 0x08 = ASYNC_SAMPLE_RATE_1 0x09 = ASYNC_SAMPLE_RATE_2 All other codes are reserved. The selected sample rate is valid in the range 8–384 kHz. All OUT*nx*MIX_SRC*m* fields must be cleared before changing OUT_RATE.
R1283 (0x0503) AIF1_Rate_Ctrl	15:11	AIF1_RATE[4:0]	0x00	AIF*n* Audio Interface Sample Rate 0x00 = SAMPLE_RATE_1 0x01 = SAMPLE_RATE_2 0x02 = SAMPLE_RATE_3 0x08 = ASYNC_SAMPLE_RATE_1 0x09 = ASYNC_SAMPLE_RATE_2 All other codes are reserved. The selected sample rate is valid in the range 8–384 kHz. All AIF*n*TXMIX_SRC*m* fields must be cleared before changing AIF*n*_RATE.
R1347 (0x0543) AIF2_Rate_Ctrl	15:11	AIF2_RATE[4:0]	0x00	
R1411 (0x0583) AIF3_Rate_Ctrl	15:11	AIF3_RATE[4:0]	0x00	
R1474 (0x05C2) SPD1_TX_Control	8:4	SPD1_RATE[4:0]	0x00	S/PDIF Transmitter Sample Rate 0x00 = SAMPLE_RATE_1 0x01 = SAMPLE_RATE_2 0x02 = SAMPLE_RATE_3 0x08 = ASYNC_SAMPLE_RATE_1 0x09 = ASYNC_SAMPLE_RATE_2 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.

Table 4-26. Digital-Core Sample-Rate Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R1509 (0x05E5) SLIMbus_Rates_1	15:11	SLIMRX2_RATE[4:0]	0x00	SLIMbus RX Channel *n* Sample Rate 0x00 = SAMPLE_RATE_1 0x01 = SAMPLE_RATE_2 0x02 = SAMPLE_RATE_3 0x08 = ASYNC_SAMPLE_RATE_1 0x09 = ASYNC_SAMPLE_RATE_2 All other codes are reserved. The selected sample rate is valid in the range 8–384 kHz.
	7:3	SLIMRX1_RATE[4:0]	0x00	
R1510 (0x05E6) SLIMbus_Rates_2	15:11	SLIMRX4_RATE[4:0]	0x00	
	7:3	SLIMRX3_RATE[4:0]	0x00	
R1511 (0x05E7) SLIMbus_Rates_3	15:11	SLIMRX6_RATE[4:0]	0x00	
	7:3	SLIMRX5_RATE[4:0]	0x00	
R1512 (0x05E8) SLIMbus_Rates_4	14:15	SLIMRX8_RATE[4:0]	0x00	
	7:3	SLIMRX7_RATE[4:0]	0x00	
R1513 (0x05E9) SLIMbus_Rates_5	15:11	SLIMTX2_RATE[4:0]	0x00	SLIMbus TX Channel *n* Sample Rate 0x00 = SAMPLE_RATE_1 0x01 = SAMPLE_RATE_2 0x02 = SAMPLE_RATE_3 0x08 = ASYNC_SAMPLE_RATE_1 0x09 = ASYNC_SAMPLE_RATE_2 All other codes are reserved. The selected sample rate is valid in the range 8–384 kHz. All SLIMTX*n*MIX_SRC*m* fields must be cleared before changing SLIMTX*n*_RATE.
	7:3	SLIMTX1_RATE[4:0]	0x00	
R1514 (0x05EA) SLIMbus_Rates_6	15:11	SLIMTX4_RATE[4:0]	0x00	
	7:3	SLIMTX3_RATE[4:0]	0x00	
R1515 (0x05EB) SLIMbus_Rates_7	15:11	SLIMTX6_RATE[4:0]	0x00	
	7:3	SLIMTX5_RATE[4:0]	0x00	
R1516 (0x05EC) SLIMbus_Rates_8	15:11	SLIMTX8_RATE[4:0]	0x00	
	7:3	SLIMTX7_RATE[4:0]	0x00	
R3584 (0x0E00) FX_Ctrl1	15:11	FX_RATE[4:0]	0x00	FX Sample Rate (EQ, LHPF, DRC) 0x00 = SAMPLE_RATE_1 0x01 = SAMPLE_RATE_2 0x02 = SAMPLE_RATE_3 0x08 = ASYNC_SAMPLE_RATE_1 0x09 = ASYNC_SAMPLE_RATE_2 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.
R4225 (0x1081) US1_Ctrl_1	15:11	US1_RATE[4:0]	0x00	Ultrasonic Demodulator 1 Sample Rate 0x00 = SAMPLE_RATE_1 0x01 = SAMPLE_RATE_2 0x02 = SAMPLE_RATE_3 All other codes are reserved. The selected sample rate must be the same as the output rate set by US1_FREQ (i.e., 8, 16, or 32 kHz).
R4227 (0x1083) US2_Ctrl_1	15:11	US2_RATE[4:0]	0x00	Ultrasonic Demodulator 2 Sample Rate 0x00 = SAMPLE_RATE_1 0x01 = SAMPLE_RATE_2 0x02 = SAMPLE_RATE_3 All other codes are reserved. The selected sample rate must be the same as the output rate set by US2_FREQ (i.e., 8, 16, or 32 kHz).

Table 4-26. Digital-Core Sample-Rate Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R5248 (0x1480) DFC1_CTRL_W0	6:2	DFC1_RATE[4:0]	0x00	DFC*n* Sample Rate 0x00 = SAMPLE_RATE_1 0x01 = SAMPLE_RATE_2 0x02 = SAMPLE_RATE_3 0x08 = ASYNC_SAMPLE_RATE_1 0x09 = ASYNC_SAMPLE_RATE_2 All other codes are reserved. The selected sample rate is valid in the range 8–192 kHz. The DFC*n*_SRC field must be cleared before changing DFC*n*_RATE.
R5254 (0x1486) DFC2_CTRL_W0	6:2	DFC2_RATE[4:0]	0x00	
R5260 (0x148C) DFC3_CTRL_W0	6:2	DFC3_RATE[4:0]	0x00	
R5266 (0x1492) DFC4_CTRL_W0	6:2	DFC4_RATE[4:0]	0x00	
R5272 (0x1498) DFC5_CTRL_W0	6:2	DFC5_RATE[4:0]	0x00	
R5278 (0x149E) DFC6_CTRL_W0	6:2	DFC6_RATE[4:0]	0x00	
R5284 (0x14A4) DFC7_CTRL_W0	6:2	DFC7_RATE[4:0]	0x00	
R5290 (0x14AA) DFC8_CTRL_W0	6:2	DFC8_RATE[4:0]	0x00	
R1048064 (0xF_FE00) DSP1_Config_1	15:11	DSP1_RATE[4:0]	0x00	DSP1 Sample Rate 0x00 = SAMPLE_RATE_1 0x01 = SAMPLE_RATE_2 0x02 = SAMPLE_RATE_3 0x08 = ASYNC_SAMPLE_RATE_1 0x09 = ASYNC_SAMPLE_RATE_2 All other codes are reserved. The selected sample rate is valid in the range 8–384 kHz. All DSP1xMIX_SRC*m* fields must be cleared before changing DSP1_RATE.

4.3.15 Asynchronous Sample-Rate Converter (ASRC)

The CS42L92 supports multiple signal paths through the digital core. Two independent clock domains are supported for the audio signal paths, referenced to SYSCLK and ASYNCCLK respectively, as described in Section 4.16. Every digital signal path must be synchronized either to SYSCLK or to ASYNCCLK.

The ASRC provides stereo, two-way signal paths between two sample rates, as shown in Fig. 4-32. Each of the sample rates may be referenced to the SYSCLK or ASYNCCLK domain. The clock domains and sample rates associated with the ASRC signal paths are selected using the ASRC1_RATE1 and ASRC1_RATE2 fields.

- ASRC1_RATE1 selects the clock domain and sample rate of the inputs to the ASRC1 IN1x paths, and the outputs from the ASRC1 IN2x paths.
- ASRC1_RATE2 selects the clock domain and sample rate of the inputs to the ASRC1 IN2x paths, and the outputs from the ASRC1 IN1x paths.

Note that it is possible to select two sample rates for the ASRC that are each referenced to the same clock domain. This provides flexibility to switch between synchronous and asynchronous use cases without changing the signal routing configuration of the affected audio paths.

See Section 4.16 for details of the sample-rate control registers.

The ASRC supports sample rates from 8–192 kHz. The ratio of the applicable SAMPLE_RATE_*n* and ASYNC_SAMPLE_RATE_*n* fields must not exceed 6.

The ASRC1_RATE1 and ASRC1_RATE2 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 ASRC1_RATE1 or ASRC1_RATE2. A minimum delay of 125 µs must be allowed between clearing the x_SRC*n* fields and writing to the associated ASRC1_RATE1 or ASRC1_RATE2 fields. See Table 4-27 for details.

The ASRC signal paths are enabled using the ASRC1_IN*nx*_ENA bits, as follows:

- The ASRC1 IN1 (left and right) paths convert from the ASRC1_RATE1 sample rate to the ASRC1_RATE2 sample rate. These paths are enabled by setting the ASRC1_IN1L_ENA and ASRC1_IN1R_ENA bits, respectively.
- The ASRC1 IN2 (left and right) paths convert from the ASRC1_RATE2 sample rate to the ASRC1_RATE1 sample rate. These paths are enabled by setting the ASRC1_IN2L_ENA and ASRC1_IN2R_ENA bits, respectively.

Synchronization (lock) between different clock domains is not instantaneous when the clocking or sample rate configurations are updated. The lock status of each ASRC path is an input to the interrupt control circuit and can be used to trigger an interrupt event; see Section 4.15.

The ASRC lock status of each ASRC path can be output directly on a GPIO pin as an external indication of ASRC lock. See Section 4.14 to configure a GPIO pin for this function.

The CS42L92 performs automatic checks to confirm that the SYSCLK or ASYNCCLK frequency is high enough to support the commanded ASRC and digital mixing functions. If the frequency is too low, an attempt to enable an ASRC signal path fails. Note that active signal paths are not affected under such circumstances.

The status bits in register R3809 indicate the status of each ASRC signal path. If an underclocked error condition occurs, these bits indicate which ASRC signal paths have been enabled.

The status bits in registers R1600–R3576 indicate the status of each digital mixer. If an underclocked error condition occurs, these bits indicate which mixers have been enabled.

The ASRC signal paths and control registers are shown in Fig. 4-32.

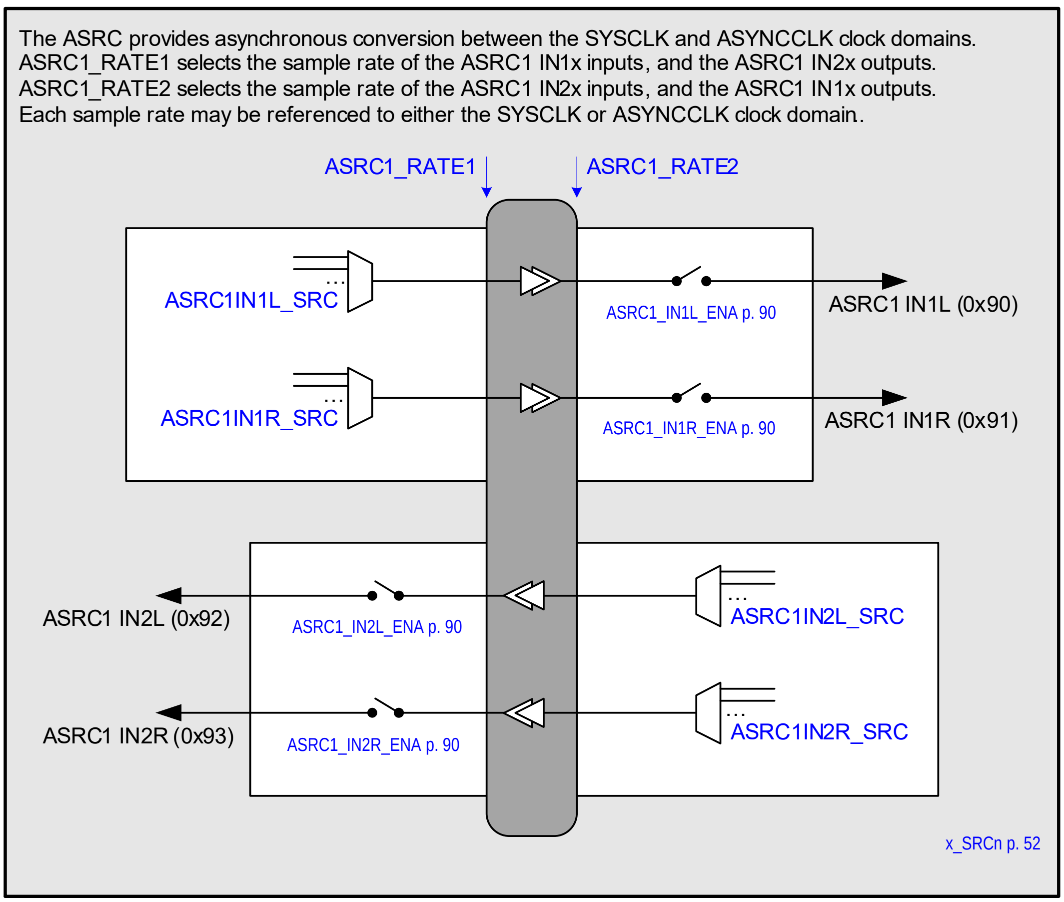


Figure 4-32. Asynchronous Sample-Rate Converters (ASRCs)

The ASRC1 input control fields (see Fig. 4-32) are located at register addresses R2688–R2712 (0x0A80–0x0A98).

The full list of digital mixer control registers (R1600–R3576) is provided in Section 6. Generic register field definitions are provided in Table 4-11.

The x_SRC*n* fields select the input sources for the ASRC paths. Note that the selected input sources must be configured for the same sample rate as the ASRC to which they are connected.

The hexadecimal numbers in Fig. 4-32 indicate the corresponding x_SRC*n* setting for selection of that signal as an input to another digital-core function.

The fields associated with the ASRC are described in Table 4-27.

Table 4-27. Digital-Core ASRC Control

Register Address	Bit	Label	Default	Description
R3808 (0x0EE0) ASRC1_ENABLE	3	ASRC1_IN2L_ ENA	0	ASRC1 IN2 (left) enable (Left channel from ASRC1_RATE2 sample rate to ASRC1_RATE1 sample rate) 0 = Disabled 1 = Enabled
	2	ASRC1_IN2R_ ENA	0	ASRC1 IN2 (right) enable (Right channel from ASRC1_RATE2 sample rate to ASRC1_RATE1 sample rate) 0 = Disabled 1 = Enabled
	1	ASRC1_IN1L_ ENA	0	ASRC1 IN1 (left) enable (Left channel from ASRC1_RATE1 sample rate to ASRC1_RATE2 sample rate) 0 = Disabled 1 = Enabled
	0	ASRC1_IN1R_ ENA	0	ASRC1 IN1 (right) enable (Right channel from ASRC1_RATE1 sample rate to ASRC1_RATE2 sample rate) 0 = Disabled 1 = Enabled

Table 4-27. Digital-Core ASRC Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R3809 (0x0EE1) ASRC1_STATUS	3	ASRC1_IN2L_ ENA_STS	0	ASRC1 IN2 (left) enable status (Left channel from ASRC1_RATE2 sample rate to ASRC1_RATE1 sample rate) 0 = Disabled 1 = Enabled
	2	ASRC1_IN2R_ ENA_STS	0	ASRC1 IN2 (right) enable status (Right channel from ASRC1_RATE2 sample rate to ASRC1_RATE1 sample rate) 0 = Disabled 1 = Enabled
	1	ASRC1_IN1L_ ENA_STS	0	ASRC1 IN1 (left) enable status (Left channel from ASRC1_RATE1 sample rate to ASRC1_RATE2 sample rate) 0 = Disabled 1 = Enabled
	0	ASRC1_IN1R_ ENA_STS	0	ASRC1 IN1 (right) enable status (Right channel from ASRC1_RATE1 sample rate to ASRC1_RATE2 sample rate) 0 = Disabled 1 = Enabled
R3810 (0x0EE2) ASRC1_RATE1	15:11	ASRC1_ RATE1[4:0]	0x00	ASRC1 Sample Rate select for ASRC1 IN1x inputs and ASRC1 IN2x outputs 0x00 = SAMPLE_RATE_1 0x01 = SAMPLE_RATE_2 0x02 = SAMPLE_RATE_3 0x08 = ASYNC_SAMPLE_RATE_1 0x09 = ASYNC_SAMPLE_RATE_2 All other codes are reserved. The selected sample rate is valid in the range 8–192 kHz. All ASRC1_IN1*x*_SRC fields must be cleared before changing ASRC1_RATE1.
R3811 (0x0EE3) ASRC1_RATE2	15:11	ASRC1_ RATE2[4:0]	0x08	ASRC1 Sample Rate select for ASRC1 IN2x inputs and ASRC1 IN1x outputs 0x00 = SAMPLE_RATE_1 0x01 = SAMPLE_RATE_2 0x02 = SAMPLE_RATE_3 0x08 = ASYNC_SAMPLE_RATE_1 0x09 = ASYNC_SAMPLE_RATE_2 All other codes are reserved. The selected sample rate is valid in the range 8–192 kHz. All ASRC1_IN1*x*_SRC fields must be cleared before changing ASRC1_RATE1.

4.3.16 Isochronous Sample-Rate Converter (ISRC)

The CS42L92 supports multiple signal paths through the digital core. The ISRCs provide sample-rate conversion between synchronized sample rates on the SYSCLK clock domain, or between synchronized sample rates on the ASYNCCLK clock domain.

There are two ISRCs on the CS42L92. Each ISRC provides two stereo signal paths between two different sample rates, as shown in Fig. 4-33.

The sample rates associated with each ISRC can be set independently. Note that the two sample rates associated with any single ISRC must both be referenced to the same clock domain (SYSCLK or ASYNCCLK).

- When an ISRC is used on the SYSCLK domain, the associated sample rates may be selected from SAMPLE_RATE_1, SAMPLE_RATE_2, or SAMPLE_RATE_3.
- When an ISRC is used on the ASYNCCLK domain, the associated sample rates are ASYNC_SAMPLE_RATE_1 and ASYNC_SAMPLE_RATE_2.

See Section 4.16 for details of the sample-rate control registers.

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). The higher of the two sample rates must be selected by ISRC*n*_FSH in each case.

The ISRCs support sample rates in the range 8–384 kHz. For each ISRC, the ratio of the applicable SAMPLE_RATE_*n* or ASYNC_SAMPLE_RATE_*n* fields must not exceed 24. The sample-rate conversion ratio must be an integer (1–24) or equal to 1.5.

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 4-28 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 CS42L92 performs automatic checks to confirm that the SYSCLK or ASYNCCLK 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–R3576 indicate the status of each digital mixer. 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. 4-33.

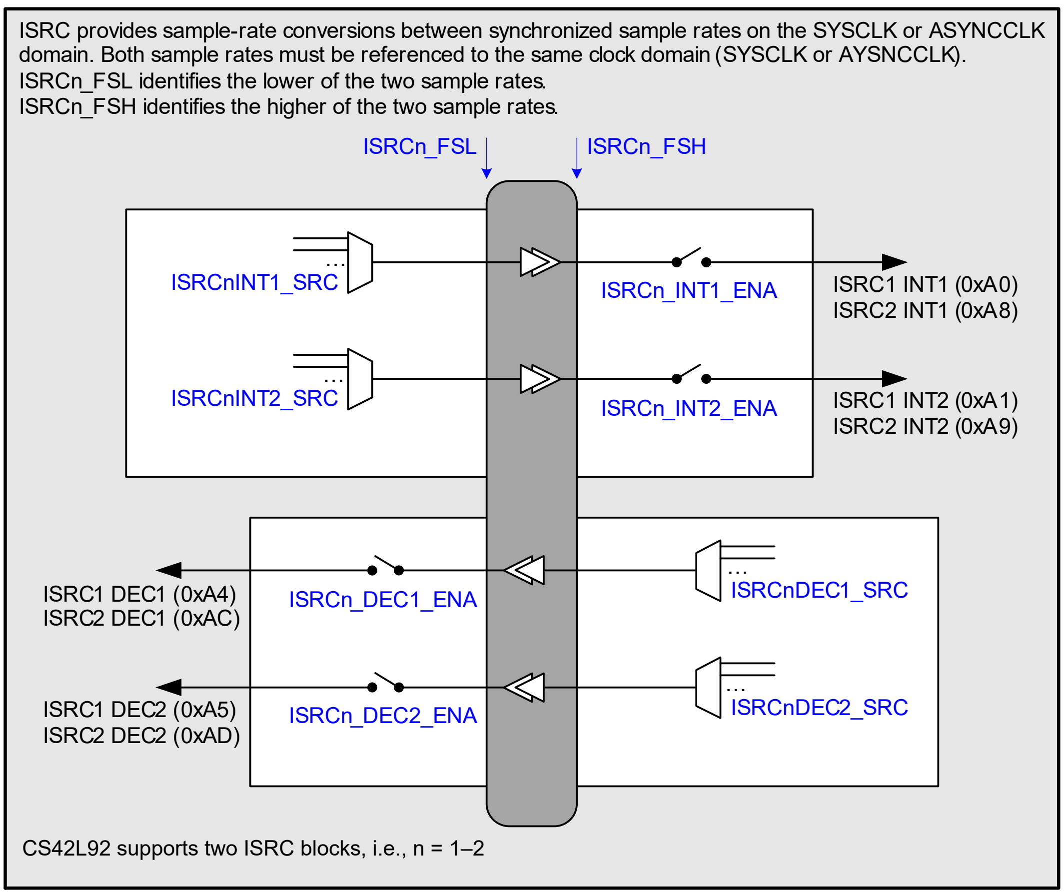


Figure 4-33. Isochronous Sample-Rate Converters (ISRCs)

The ISRC input control fields (see Fig. 4-33) are located at register addresses R2816–R2920 (0x0B00–0x0B68).

The full list of digital mixer control registers (R1600–R3576) is provided in Section 6. Generic register field definitions are provided in Table 4-11.

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. 4-33 indicate the corresponding x_SRC setting for selection of that signal as an input to another digital-core function.

The fields associated with the ISRCs are described in Table 4-28.

Table 4-28. Digital-Core ISRC Control

Register Address	Bit	Label	Default	Description
R3824 (0x0EF0) ISRC1_CTRL_1	15:11	ISRC1_FSH[4:0]	0x00	ISRC1 High Sample Rate (Sets the higher of the ISRC1 sample rates) 0x00 = SAMPLE_RATE_1 0x01 = SAMPLE_RATE_2 0x02 = SAMPLE_RATE_3 0x08 = ASYNC_SAMPLE_RATE_1 0x09 = ASYNC_SAMPLE_RATE_2 All other codes are reserved. The selected sample rate is valid in the range 8–192 kHz. The ISRC1_FSH and ISRC1_FSL fields must both select sample rates referenced to the same clock domain (SYSCLK or ASYNCCLK). All ISRC1_DEC*n*_SRC fields must be cleared before changing ISRC1_FSH.
R3825 (0x0EF1) ISRC1_CTRL_2	15:11	ISRC1_FSL[4:0]	0x00	ISRC1 Low Sample Rate (Sets the lower of the ISRC1 sample rates) 0x00 = SAMPLE_RATE_1 0x01 = SAMPLE_RATE_2 0x02 = SAMPLE_RATE_3 0x08 = ASYNC_SAMPLE_RATE_1 0x09 = ASYNC_SAMPLE_RATE_2 All other codes are reserved. The selected sample rate is valid in the range 8–192 kHz. The ISRC1_FSH and ISRC1_FSL fields must both select sample rates referenced to the same clock domain (SYSCLK or ASYNCCLK). All ISRC1_INT*n*_SRC fields must be cleared before changing ISRC1_FSL.
R3826 (0x0EF2) ISRC1_CTRL_3	15	ISRC1_INT1_ENA	0	ISRC1 INT1 Enable (Interpolation Channel 1 path from ISRC1_FSL rate to ISRC1_FSH rate) 0 = Disabled 1 = Enabled
	14	ISRC1_INT2_ENA	0	ISRC1 INT2 Enable (Interpolation Channel 2 path from ISRC1_FSL rate to ISRC1_FSH rate) 0 = Disabled 1 = Enabled
	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
R3827 (0x0EF3) ISRC2_CTRL_1	15:11	ISRC2_FSH[4:0]	0x00	ISRC2 High Sample Rate (Sets the higher of the ISRC2 sample rates) 0x00 = SAMPLE_RATE_1 0x01 = SAMPLE_RATE_2 0x02 = SAMPLE_RATE_3 0x08 = ASYNC_SAMPLE_RATE_1 0x09 = ASYNC_SAMPLE_RATE_2 All other codes are reserved. The selected sample rate is valid in the range 8–192 kHz. The ISRC2_FSH and ISRC2_FSL fields must both select sample rates referenced to the same clock domain (SYSCLK or ASYNCCLK). All ISRC2_DEC*n*_SRC fields must be cleared before changing ISRC2_FSH.

Table 4-28. Digital-Core ISRC Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R3828 (0x0EF4) ISRC2_CTRL_2	15:11	ISRC2_FSL[4:0]	0x00	ISRC2 Low Sample Rate (Sets the lower of the ISRC2 sample rates) 0x00 = SAMPLE_RATE_1 0x01 = SAMPLE_RATE_2 0x02 = SAMPLE_RATE_3 0x08 = ASYNC_SAMPLE_RATE_1 0x09 = ASYNC_SAMPLE_RATE_2 All other codes are reserved. The selected sample rate is valid in the range 8–192 kHz. The ISRC2_FSH and ISRC2_FSL fields must both select sample rates referenced to the same clock domain (SYSCLK or ASYNCCLK). All ISRC2_INT*n*_SRC fields must be cleared before changing ISRC2_FSL.
R3829 (0x0EF5) ISRC2_CTRL_3	15	ISRC2_INT1_ENA	0	ISRC2 INT1 Enable (Interpolation Channel 1 path from ISRC2_FSL rate to ISRC2_FSH rate) 0 = Disabled 1 = Enabled
	14	ISRC2_INT2_ENA	0	ISRC2 INT2 Enable (Interpolation Channel 2 path from ISRC2_FSL rate to ISRC2_FSH rate) 0 = Disabled 1 = Enabled
	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

4.4 DSP Firmware Control

The CS42L92 digital core incorporates one programmable DSP processing block, capable of running a range of application-specific algorithms. Different firmware configurations can be loaded onto the DSP, enabling the CS42L92 to be customized for specific application requirements. Full read/write access to the device register map is supported from the DSP core.

The DSP can be clocked at up to 75 MHz, corresponding to 75 MIPS. A software programming guide can be provided to assist users in developing their own software algorithms—please contact your Cirrus Logic representative for further information.

To use the programmable DSP, the required firmware configuration must first be loaded onto the device by writing the appropriate files to the CS42L92 register map. The firmware configuration comprises program, data, and coefficient content.

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 CS42L92. Please contact your Cirrus Logic representative for more details of the WISCE evaluation board control software.

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

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

4.4.1 DSP Firmware Memory and Register Mapping

The DSP firmware memory is programmed by writing to the registers referenced in Table 4-29. Note that clocking is not required for access to the firmware registers by the host processor.

The CS42L92 program, data, and coefficient register memory space is described in Table 4-29. The full register map listing is provided in Section 6.

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

Table 4-29. DSP Program, Data, and Coefficient Registers

DSP Number	Description	Register Address	Number of Registers	DSP Memory Size
DSP1	Program memory	0x08_0000–0x08_2FFE	6144	4k x 40-bit words
	X-Data memory	0x0A_0000–0x0A_1FFE	4096	4k x 24-bit words
	Y-Data memory	0x0C_0000–0x0C_1FFE	4096	4k x 24-bit words
	Coefficient memory	0x0E_0000–0x0E_1FFE	4096	4k x 24-bit words

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

Note that the register window space is additional to the X-data memory size described in Table 4-29.

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

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

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

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

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

The register map window functions are shown in Fig. 4-34. Further information on the definition and usage of the DSP firmware memories is provided in the software programming guide; contact your Cirrus Logic representative if required.

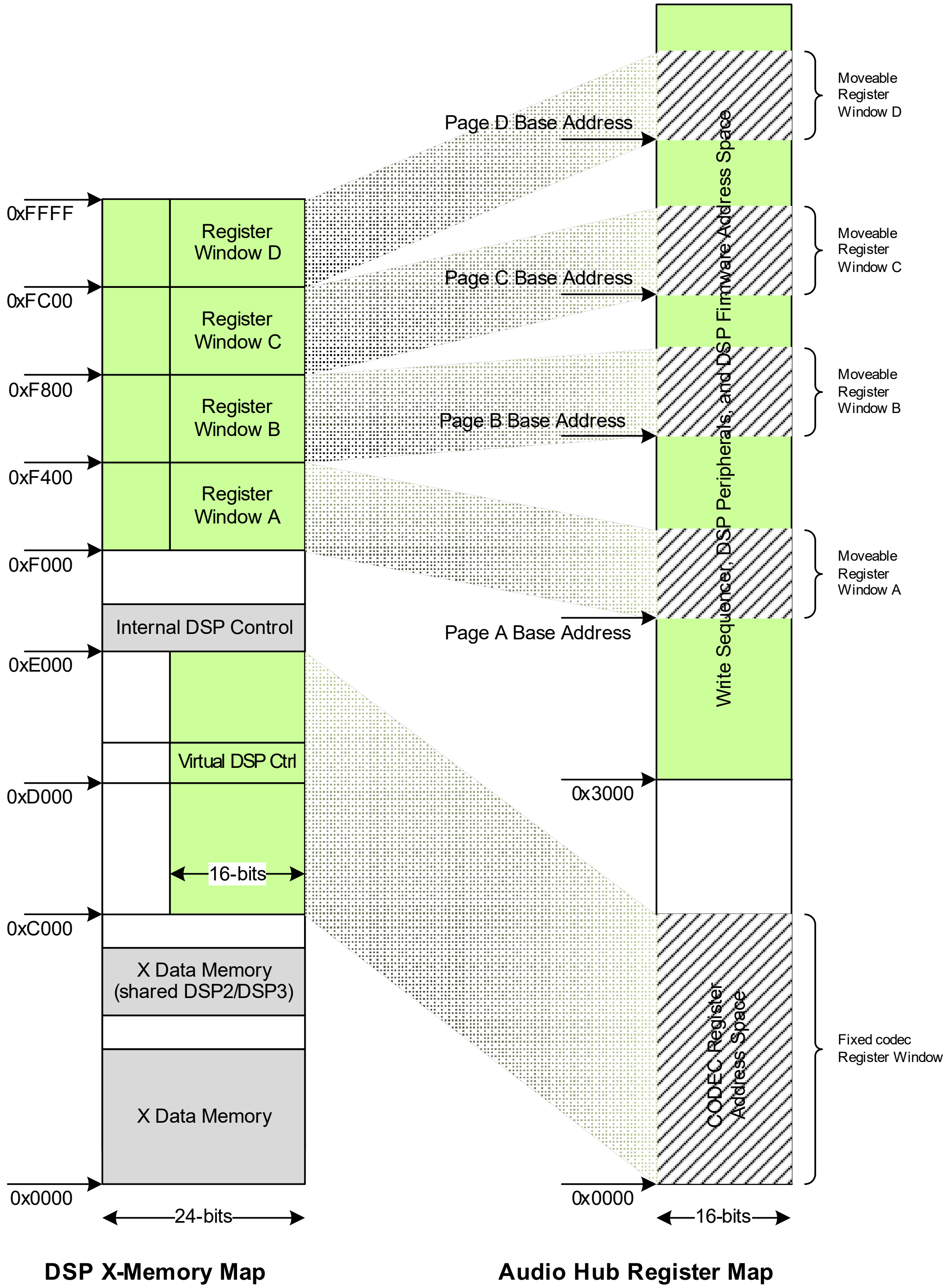


Figure 4-34. X-Data Memory Map

Note that the full CS42L92 register space is shown here as 16-bit width. (SPI/I2C register access uses 32-bit data width at 0x3000 and above.) However, the window base address fields (DSP1_EXT_[A/B/C/D]_PAGE) are referenced to 16-bit width, and 16-bit register mapping is shown. Hence, the device register map is shown here entirely as 16-bit width for ease of explanation.

The control registers associated with the register map window functions are described in Table 4-30.

Table 4-30. X-Data Memory and Clocking Control

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

4.4.2 DSP Memory Locking

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

The CS42L92 provides a register-locking feature that blocks DSP register-write attempts to invalid register regions, preventing the firmware from making unintentional changes to register and memory contents. An interrupt event and associated debug information are generated if any write-access attempt is blocked; this can be used to assist software development and debug.

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

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

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

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

The lockable register/memory regions are defined in Table 4-31.

Table 4-31. DSP Memory Locking Regions

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

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

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

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

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

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

The control registers associated with the DSP memory locking functions are described in Table 4-32.

Table 4-32. DSP Memory Locking Control

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

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

4.4.3 DSP Firmware Control

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

4.4.3.1 DSP Memory

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

The DSP1_RAM_RDY status bit indicates whether the DSP memory is ready for read/write access. The DSP memory should not be accessed until this bit has been set.

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

The DSP firmware memory is always cleared under power-on reset, hardware reset, and Sleep Mode conditions. See Section 5.2 for a summary of the CS42L92 reset behavior.

4.4.3.2 DSP Clocking

Clocking is required for the DSP processing block, when executing software or when supporting DMA functions. (Note that clocking is not required for access to the firmware registers by the host processor.) Clocking within the DSP is enabled and disabled automatically, as required by the DSP core and DMA channel status.

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

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

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

The DSPCLK configuration provides input to the interrupt control circuit and can be used to trigger an interrupt event when the DSP1 clock frequency is less than the requested frequency; see Section 4.15.

4.4.3.3 DSP Code Execution

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

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

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

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

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

4.4.3.4 DSP Watchdog Timer

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

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

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

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

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

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

4.4.3.5 DSP Control Registers

The DSP memory, clocking, code-execution, and watchdog control registers are described in Table 4-33.

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

Table 4-33. DSP Control Registers

Register Address	Bit	Label	Default	Description
R1048064 (0xF_ FE00) DSP1_Config_1	4	DSP1_MEM_ENA	0	DSP1 memory control 0 = Disabled 1 = Enabled The DSP1 memory contents are lost when DSP1_MEM_ENA =0. Note that this bit is not affected by software reset; it remains in its previous condition.
	1	DSP1_CORE_ENA	0	DSP1 enable. Controls the DSP1 firmware execution 0 = Disabled 1 = Enabled
	0	DSP1_START	—	DSP1 start Write 1 to start DSP1 firmware execution
R1048066 (0xF_ FE02) DSP1_Config_2	15:0	DSP1_CLK_FREQ_ SEL[15:0]	0x0000	DSP1 clock frequency select Coded as LSB = 1/64 MHz, Valid from 5.6 to 75 MHz. The DSP1 clock must be less than or equal to the DSPCLK frequency. The DSP1 clock is generated by division of DSPCLK, and may differ from the selected frequency. The DSP1 clock frequency can be read from DSP1_ CLK_FREQ_STS.
R1048070 (0xF_ FE06) DSP1_Status_2	0	DSP1_CLK_AVAIL	0	DSP1 clock availability (read only) 0 = No Clock 1 = Clock Available This bit exists for legacy software support only; it is not recommended for future designs—it may be unreliable on the latest device architectures.
R1048072 (0xF_ FE08) DSP1_Status_3	15:0	DSP1_CLK_FREQ_ STS[15:0]	0x0000	DSP1 clock frequency (read only). Valid only when the respective DSP core is enabled. Coded as LSB = 1/64 MHz.
R1048074 (0xF_ FE0A) DSP1_Watchdog_1	4:1	DSP1_WDT_MAX_ COUNT[3:0]	0x0	DSP1 watchdog timeout value. 0x0 = 2 ms 0x5 = 64 ms 0xA = 2 s 0x1 = 4 ms 0x6 = 128 ms 0xB = 4 s 0x2 = 8 ms 0x7 = 256 ms 0xC = 8 s 0x3 = 16 ms 0x8 = 512 ms 0xD–0xF = reserved 0x4 = 32 ms 0x9 = 1 s
	0	DSP1_WDT_ENA	0	DSP1 watchdog enable 0 = Disabled 1 = Enabled
R1048120 (0xF_ FE38) DSP1_External_Start	4:0	DSP1_START_IN_ SEL[4:0]	0x00	DSP1 firmware execution control. Selects the trigger for DSP1 firmware execution. 0x00 = DMA 0x0B = IRQ2 0x10 = Event Logger 1 All other codes are reserved. Note that the DSP1_START bit also starts the DSP1 firmware execution, regardless of this field setting.
R1048158 (0xF_ FE5E) DSP1_Watchdog_2	15:0	DSP1_WDT_ RESET[15:0]	0x0000	DSP1 watchdog reset. Write 0x5555, followed by 0xAAAA, to reset the watchdog.
R1048186 (0xF_ FE7A) DSP1_Region_lock_ ctrl_0	13	DSP1_WDT_ TIMEOUT_STS	0	DSP1 watchdog timeout status This bit, when set, indicates that the watchdog timeout has occurred. This bit is latched when set; it is cleared when the watchdog is disabled or reset.

4.4.4 DSP Direct Memory Access (DMA) Control

The DSP provides a multichannel DMA function; this is configured using the registers described in Table 4-34.

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

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

The start address of each DMA channel is configured as described in Table 4-34. Note that the required address is defined relative to the base address of the selected (X-data or Y-data) memory.

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

Note that the start-address fields and buffer-length fields are defined in 24-bit DSP data word units. This means that the LSB of these fields represents one 24-bit DSP memory word. This differs from the CS42L92 register map layout described in Table 4-29.

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

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

When the ping input data buffer is full, the DSP1_PING_FULL bit is set, and a DSP start signal is generated. The start signal from the DMA is typically used to start firmware execution, as noted in Table 4-33. Meanwhile, further DSP input data fills up the pong buffer.

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

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

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

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

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

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

Table 4-34. DMA Control

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

Table 4-34. DMA Control *(Cont.)*

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

4.4.5 DSP Interrupts

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

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

The CS42L92 also provides 16 control bits that allow the DSP core to generate programmable interrupt events. When a 1 is written to these bits (see Table 4-35), 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 4-35. DSP Interrupts

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

Table 4-35. DSP Interrupts *(Cont.)*

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

4.4.6 DSP Debug Support

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

The JTAG interface provides test and debug access to the CS42L92, as described in Section 4.20. The JTAG interface clock can be enabled for the DSP core using DSP1_DBG_CLK_ENA. Note that, when the JTAG interface is used to access the DSP core, the DSP1_CORE_ENA bit must also be set.

The DSP1_LOCK_ERR_STS bit indicates that the DSP attempted to write to a locked register address. The DSP1_ADDR_ERR_STS bit indicates that the DSP attempted to access an invalid memory address (i.e., an address whose contents are undefined). Once set, these bits remain set until a 1 is written to DSP1_ERR_CLEAR.

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

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

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

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

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

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

Table 4-36. DSP Debug Support

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

4.4.7 Virtual DSP Registers

The DSP control registers are described throughout Section 4.4. Each control register has a unique location within the CS42L92 register map.

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

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

The virtual DSP registers are defined at register addresses R4096–R4192 (0x1000–0x1060) in the device register map. Note that these registers cannot be accessed directly at the addresses shown; they can be only accessed through DSP firmware code, using the register window function shown in Fig. 4-34. The virtual DSP registers are located at address 0xD000 in the X-data memory map.

4.5 DSP Peripheral Control

The CS42L92 incorporates a suite of DSP peripheral functions that can be integrated together to provide an enhanced capability for DSP applications. Configurable event log functions provide multichannel monitoring of internal and external signals. The general-purpose timer provides time-stamp data for the event logger; it also supports the watchdog and other miscellaneous time-based functions. Maskable GPIO provides an efficient mechanism for the DSP core to access the required input and output signals.

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

4.5.1 Event Logger

The CS42L92 provides an event log function, supporting multichannel, edge-sensitive monitoring and recording of internal or external signals.

4.5.1.1 Overview

The event logger allows 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 logger, 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 the general-purpose timer, is associated and recorded with each FIFO index, to provide a comprehensive record of the detected events.

The event logger must be associated with the general-purpose timer. The 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 are queued, and are 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 queued events are not logged.

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

4.5.1.2 Event Logger Control

The event logger is enabled by setting EVENTLOG1_ENA. The event logger can be reset by writing 1 to EVENTLOG1_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 EVENTLOG1_TIME_SEL. Note that the event logger must be disabled (EVENTLOG1_ENA = 0) when selecting the timer source.

4.5.1.3 Input Channel Configuration

The event logger allows up to 16 input channels to be configured for detection and logging. The EVENTLOG1_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 EVENTLOG1_CH*x*_POL and EVENTLOG1_CH*x*_DB bits respectively.

To avoid filling the FIFO buffer with repeated instances of any event, a selectable filter is provided for each input channel. If the EVENTLOG1_CH*x*_FILT bit is set, new events on the respective channel are ignored by the event logger if the unread entries in the FIFO buffer indicate a previous event of the same type (i.e., same input source and same polarity). The read/write pointers of the FIFO buffer (see Section 4.5.1.4) are used to determine which FIFO entries are unread (i.e., have not yet been read by the host processor).

The input channels can be enabled or disabled freely, using EVENTLOG1_CH*x*_ENA, without having to disable the event logger entirely. An input channel must be disabled whenever the associated x_SEL, x_FILT, x_POL, or x_DB fields are written. It is possible to reconfigure input channels while the event logger is enabled, provided the channels being reconfigured are disabled when doing so.

The available input sources include GPIO inputs, external accessory status (jack, mic, sensors), and signals generated by the integrated DSP core. A list of the valid input sources for the event logger is provided in Table 4-38. 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 (EVENTLOG1_CH*x*_POL = 0), and the corresponding input signal is asserted (Logic 1) at the time when the event logger is enabled, an event is logged in respect of this initial state. Similarly, if an input channel is configured for falling edge detection, and is deasserted (Logic 0) when the event logger is enabled, a corresponding event is logged. If rising and falling edges are both configured for detection, an event is always logged in respect of the initial condition.

4.5.1.4 FIFO Buffer

Each event (signal transition) that meets the criteria of an enabled channel is written to the 16-stage FIFO buffer. The buffer is filled cyclically, but does not overwrite unread data when full. A status bit is provided to indicate 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 EVENTLOG1_FIFO*y*_ID (identifying the source signal of the associated log event), the EVENTLOG1_FIFO*y*_POL (the polarity of the respective event transition), and the EVENTLOG1_FIFO*y*_TIME field (containing the 32-bit time stamp from the timer).

The FIFO buffer is managed using EVENTLOG1_FIFO_WPTR and EVENTLOG1_FIFO_RPTR. The write pointer (WPTR) field identifies the index location (0 to 15) in which the next event is logged. The read pointer (RPTR) field identifies the index location of the first set of unread data, if any exists. Both of these fields are initialized to 0 when the event logger is reset.

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

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

4.5.1.5 Status Bits

The EVENTLOG1_CHx_STS bits indicate the status of the source signal for the respective input channel. Note that the status indication is not valid for all input source selections—it is not possible to provide indication of transitory events (e.g., microphone accessory detection).

The EVENTLOG1_NOT_EMPTY bit indicates whether the FIFO buffer is empty. If this bit is set, it indicates one or more new sets of data in the FIFO.

The EVENTLOG1_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 EVENTLOG1_FIFO_WMARK field.

The EVENTLOG1_FULL bit indicates when the FIFO buffer is full. If this bit is set, it indicates that there are 16 sets of new event data in the FIFO. Note that this does not mean that a buffer overflow condition has occurred, but further events are not logged or indicated until the buffer has been cleared.

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

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

The control-write sequencer is automatically triggered whenever the NOT_EMPTY status of the event log buffer is asserted. 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.

4.5.1.7 Event Logger Control Registers

The event logger control registers are described in Table 4-37.

Table 4-37. Event Logger (EVENTLOG1) Control

Register Address	Bit	Label	Default	Description
R294912 (0x4_8000) EVENTLOG1_CONTROL	1	EVENTLOG1_RST	0	Event Log Reset Write 1 to reset the status outputs and clear the FIFO buffer.
	0	EVENTLOG1_ENA	0	Event Log Enable 0 = Disabled 1 = Enabled
R294916 (0x4_8004) EVENTLOG1_TIMER_SEL	1:0	EVENTLOG1_TIMER_ SEL[1:0]	00	Event Log Timer Source Select 00 = Timer 1 All other codes are reserved. Note that the event log must be disabled when updating this field.
R294924 (0x4_800C) EVENTLOG1_FIFO_ CONTROL1	3:0	EVENTLOG1_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.
R294926 (0x4_800E) EVENTLOG1_FIFO_ POINTER1	18	EVENTLOG1_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	EVENTLOG1_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	EVENTLOG1_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	EVENTLOG1_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	EVENTLOG1_FIFO_ RPTR[3:0]	0x0	Event Log FIFO Read Pointer. Indicates the FIFO index location of the first set of unread data, if any exists. For the intended FIFO behavior, this field must be incremented after the respective data has been read.

Table 4-37. Event Logger (EVENTLOG1) Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R294944 (0x4_8020) EVENTLOG1_CH_ENABLE	15	EVENTLOG1_CH16_ENA	0	Event Log Channel 16 Enable 0 = Disabled, 1 = Enabled
	14	EVENTLOG1_CH15_ENA	0	Event Log Channel 15 Enable 0 = Disabled, 1 = Enabled
	13	EVENTLOG1_CH14_ENA	0	Event Log Channel 14 Enable 0 = Disabled, 1 = Enabled
	12	EVENTLOG1_CH13_ENA	0	Event Log Channel 13 Enable 0 = Disabled, 1 = Enabled
	11	EVENTLOG1_CH12_ENA	0	Event Log Channel 12 Enable 0 = Disabled, 1 = Enabled
	10	EVENTLOG1_CH11_ENA	0	Event Log Channel 11 Enable 0 = Disabled, 1 = Enabled
	9	EVENTLOG1_CH10_ENA	0	Event Log Channel 10 Enable 0 = Disabled, 1 = Enabled
	8	EVENTLOG1_CH9_ENA	0	Event Log Channel 9 Enable 0 = Disabled, 1 = Enabled
	7	EVENTLOG1_CH8_ENA	0	Event Log Channel 8 Enable 0 = Disabled, 1 = Enabled
	6	EVENTLOG1_CH7_ENA	0	Event Log Channel 7 Enable 0 = Disabled, 1 = Enabled
	5	EVENTLOG1_CH6_ENA	0	Event Log Channel 6 Enable 0 = Disabled, 1 = Enabled
	4	EVENTLOG1_CH5_ENA	0	Event Log Channel 5 Enable 0 = Disabled, 1 = Enabled
	3	EVENTLOG1_CH4_ENA	0	Event Log Channel 4 Enable 0 = Disabled, 1 = Enabled
	2	EVENTLOG1_CH3_ENA	0	Event Log Channel 3 Enable 0 = Disabled, 1 = Enabled
	1	EVENTLOG1_CH2_ENA	0	Event Log Channel 2 Enable 0 = Disabled, 1 = Enabled
	0	EVENTLOG1_CH1_ENA	0	Event Log Channel 1 Enable 0 = Disabled, 1 = Enabled
R294948 (0x4_8024) EVENTLOG1_CH_STATUS	15	EVENTLOG1_CH16_STS	0	Event Log Channel 16 Status
	14	EVENTLOG1_CH15_STS	0	Event Log Channel 15 Status
	13	EVENTLOG1_CH14_STS	0	Event Log Channel 14 Status
	12	EVENTLOG1_CH13_STS	0	Event Log Channel 13 Status
	11	EVENTLOG1_CH12_STS	0	Event Log Channel 12 Status
	10	EVENTLOG1_CH11_STS	0	Event Log Channel 11 Status
	9	EVENTLOG1_CH10_STS	0	Event Log Channel 10 Status
	8	EVENTLOG1_CH9_STS	0	Event Log Channel 9 Status
	7	EVENTLOG1_CH8_STS	0	Event Log Channel 8 Status
	6	EVENTLOG1_CH7_STS	0	Event Log Channel 7 Status
	5	EVENTLOG1_CH6_STS	0	Event Log Channel 6 Status
	4	EVENTLOG1_CH5_STS	0	Event Log Channel 5 Status
	3	EVENTLOG1_CH4_STS	0	Event Log Channel 4 Status
	2	EVENTLOG1_CH3_STS	0	Event Log Channel 3 Status
	1	EVENTLOG1_CH2_STS	0	Event Log Channel 2 Status
	0	EVENTLOG1_CH1_STS	0	Event Log Channel 1 Status

Table 4-37. Event Logger (EVENTLOG1) Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R294976 (0x4_8040) EVENTLOG1_CH1_DEFINE to R295006 (0x4_805E) EVENTLOG1_CH16_DEFINE	15	EVENTLOG1_CH*n*_DB	0	Event Log Channel *n* debounce 0 = Disabled, 1 = Enabled Note that channel must be disabled when updating this field.
	14	EVENTLOG1_CH*n*_POL	0	Event Log Channel *n* polarity 0 = Rising edge triggered, 1 = Falling edge triggered Note that channel must be disabled when updating this field.
	13	EVENTLOG1_CH*n*_FILT	0	Event Log Channel *n* filter 0 = Disabled, 1 = Enabled If the filter is enabled, the channel is ignored if the FIFO contains unread events of the same source/polarity. Note that channel must be disabled when updating this field.
	9:0	EVENTLOG1_CH*n*_SEL[9:0]	0x000	Event Log Channel *n* source [1] Note that channel must be disabled when updating this field.
R295040 (0x4_8080) EVENTLOG1_FIFO0_READ	12	EVENTLOG1_FIFO0_POL	0	Event Log FIFO Index 0 polarity 0 = Rising edge, 1 = Falling edge
	9:0	EVENTLOG1_FIFO0_ID[9:0]	0x000	Event Log FIFO Index 0 source [1]
R295042 (0x4_8082) EVENTLOG1_FIFO0_TIME	31:0	EVENTLOG1_FIFO0_ TIME[31:0]	0x0000 _0000	Event Log FIFO Index 0 Time
R295044 (0x4_8084) EVENTLOG1_FIFO1_READ	12	EVENTLOG1_FIFO1_POL	0	Event Log FIFO Index 1 polarity 0 = Rising edge, 1 = Falling edge
	9:0	EVENTLOG1_FIFO1_ID[9:0]	0x000	Event Log FIFO Index 1 source [1]
R295046 (0x4_8086) EVENTLOG1_FIFO1_TIME	31:0	EVENTLOG1_FIFO1_ TIME[31:0]	0x0000 _0000	Event Log FIFO Index 1 Time
R295048 (0x4_8088) EVENTLOG1_FIFO2_READ	12	EVENTLOG1_FIFO2_POL	0	Event Log FIFO Index 2 polarity 0 = Rising edge, 1 = Falling edge
	9:0	EVENTLOG1_FIFO2_ID[9:0]	0x000	Event Log FIFO Index 2 source [1]
R295050 (0x4_808A) EVENTLOG1_FIFO2_TIME	31:0	EVENTLOG1_FIFO2_ TIME[31:0]	0x0000 _0000	Event Log FIFO Index 2 Time
R295052 (0x4_808C) EVENTLOG1_FIFO3_READ	12	EVENTLOG1_FIFO3_POL	0	Event Log FIFO Index 3 polarity 0 = Rising edge, 1 = Falling edge
	9:0	EVENTLOG1_FIFO3_ID[9:0]	0x000	Event Log FIFO Index 3 source [1]
R295054 (0x4_808E) EVENTLOG1_FIFO3_TIME	31:0	EVENTLOG1_FIFO3_ TIME[31:0]	0x0000 _0000	Event Log FIFO Index 3 Time
R295056 (0x4_8090) EVENTLOG1_FIFO4_READ	12	EVENTLOG1_FIFO4_POL	0	Event Log FIFO Index 4 polarity 0 = Rising edge, 1 = Falling edge
	9:0	EVENTLOG1_FIFO4_ID[9:0]	0x000	Event Log FIFO Index 4 source [1]
R295058 (0x4_8092) EVENTLOG1_FIFO4_TIME	31:0	EVENTLOG1_FIFO4_ TIME[31:0]	0x0000 _0000	Event Log FIFO Index 4 Time
R295060 (0x4_8094) EVENTLOG1_FIFO5_READ	12	EVENTLOG1_FIFO5_POL	0	Event Log FIFO Index 5 polarity Field description is as above.
	9:0	EVENTLOG1_FIFO5_ID[9:0]	0x000	Event Log FIFO Index 5 source [1]
R295062 (0x4_8096) EVENTLOG1_FIFO5_TIME	31:0	EVENTLOG1_FIFO5_ TIME[31:0]	0x0000 _0000	Event Log FIFO Index 5 Time
R295064 (0x4_8098) EVENTLOG1_FIFO6_READ	12	EVENTLOG1_FIFO6_POL	0	Event Log FIFO Index 6 polarity 0 = Rising edge, 1 = Falling edge
	9:0	EVENTLOG1_FIFO6_ID[9:0]	0x000	Event Log FIFO Index 6 source [1]
R295066 (0x4_809A) EVENTLOG1_FIFO6_TIME	31:0	EVENTLOG1_FIFO6_ TIME[31:0]	0x0000 _0000	Event Log FIFO Index 6 Time
R295068 (0x4_809C) EVENTLOG1_FIFO7_READ	12	EVENTLOG1_FIFO7_POL	0	Event Log FIFO Index 7 polarity 0 = Rising edge, 1 = Falling edge
	9:0	EVENTLOG1_FIFO7_ID[9:0]	0x000	Event Log FIFO Index 7 source [1]
R295070 (0x4_809E) EVENTLOG1_FIFO7_TIME	31:0	EVENTLOG1_FIFO7_ TIME[31:0]	0x0000 _0000	Event Log FIFO Index 7 Time
R295072 (0x4_80A0) EVENTLOG1_FIFO8_READ	12	EVENTLOG1_FIFO8_POL	0	Event Log FIFO Index 8 polarity 0 = Rising edge, 1 = Falling edge
	9:0	EVENTLOG1_FIFO8_ID[9:0]	0x000	Event Log FIFO Index 8 source [1]

Table 4-37. Event Logger (EVENTLOG1) Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R295074 (0x4_80A2) EVENTLOG1_FIFO8_TIME	31:0	EVENTLOG1_FIFO8_ TIME[31:0]	0x0000 _0000	Event Log FIFO Index 8 Time
R295076 (0x4_80A4) EVENTLOG1_FIFO9_READ	12	EVENTLOG1_FIFO9_POL	0	Event Log FIFO Index 9 polarity 0 = Rising edge, 1 = Falling edge
	9:0	EVENTLOG1_FIFO9_ID[9:0]	0x000	Event Log FIFO Index 9 source [1]
R295078 (0x4_80A6) EVENTLOG1_FIFO9_TIME	31:0	EVENTLOG1_FIFO9_ TIME[31:0]	0x0000 _0000	Event Log FIFO Index 9 Time
R295080 (0x4_80A8) EVENTLOG1_FIFO10_READ	12	EVENTLOG1_FIFO10_POL	0	Event Log FIFO Index 10 polarity 0 = Rising edge, 1 = Falling edge
	9:0	EVENTLOG1_FIFO10_ID[9:0]	0x000	Event Log FIFO Index 10 source [1]
R295082 (0x4_80AA) EVENTLOG1_FIFO10_TIME	31:0	EVENTLOG1_FIFO10_ TIME[31:0]	0x0000 _0000	Event Log FIFO Index 10 Time
R295084 (0x4_80AC) EVENTLOG1_FIFO11_READ	12	EVENTLOG1_FIFO11_POL	0	Event Log FIFO Index 11 polarity 0 = Rising edge, 1 = Falling edge
	9:0	EVENTLOG1_FIFO11_ID[9:0]	0x000	Event Log FIFO Index 11 source [1]
R295086 (0x4_80AE) EVENTLOG1_FIFO11_TIME	31:0	EVENTLOG1_FIFO11_ TIME[31:0]	0x0000 _0000	Event Log FIFO Index 11 Time
R295088 (0x4_80B0) EVENTLOG1_FIFO12_READ	12	EVENTLOG1_FIFO12_POL	0	Event Log FIFO Index 12 polarity 0 = Rising edge, 1 = Falling edge
	9:0	EVENTLOG1_FIFO12_ID[9:0]	0x000	Event Log FIFO Index 12 source [1]
R295090 (0x4_80B2) EVENTLOG1_FIFO12_TIME	31:0	EVENTLOG1_FIFO12_ TIME[31:0]	0x0000 _0000	Event Log FIFO Index 12 Time
R295092 (0x4_80B4) EVENTLOG1_FIFO13_READ	12	EVENTLOG1_FIFO13_POL	0	Event Log FIFO Index 13 polarity 0 = Rising edge, 1 = Falling edge
	9:0	EVENTLOG1_FIFO13_ID[9:0]	0x000	Event Log FIFO Index 13 source [1]
R295094 (0x4_80B6) EVENTLOG1_FIFO13_TIME	31:0	EVENTLOG1_FIFO13_ TIME[31:0]	0x0000 _0000	Event Log FIFO Index 13 Time
R295096 (0x4_80B8) EVENTLOG1_FIFO14_READ	12	EVENTLOG1_FIFO14_POL	0	Event Log FIFO Index 14 polarity 0 = Rising edge, 1 = Falling edge
	9:0	EVENTLOG1_FIFO14_ID[9:0]	0x000	Event Log FIFO Index 14 source [1]
R295098 (0x4_80BA) EVENTLOG1_FIFO14_TIME	31:0	EVENTLOG1_FIFO14_ TIME[31:0]	0x0000 _0000	Event Log FIFO Index 14 Time
R295100 (0x4_80BC) EVENTLOG1_FIFO15_READ	12	EVENTLOG1_FIFO15_POL	0	Event Log FIFO Index 15 polarity 0 = Rising edge, 1 = Falling edge
	9:0	EVENTLOG1_FIFO15_ID[9:0]	0x000	Event Log FIFO Index 15 source [1]
R295102 (0x4_80BE) EVENTLOG1_FIFO15_TIME	31:0	EVENTLOG1_FIFO15_ TIME[31:0]	0x0000 _0000	Event Log FIFO Index 15 Time

1. See Table 4-38 for valid channel source selections

4.5.1.8 Event Logger Input Sources

A list of the valid input sources for the event logger is provided in Table 4-38.

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 logger; a recursive loop, where the IRQ*n* signal is also an output from the event logger, must be avoided.

Table 4-38. Event Logger Input Sources

ID	Description	Edge
3	irq1	D
4	irq2	D
9	sysclk_fail	S
24	fll1_lock	D
25	fll2_lock	D

ID	Description	Edge
137	asrc1_in2_lock	D
160	dsp_irq1	S
161	dsp_irq2	S
162	dsp_irq3	S
163	dsp_irq4	S

ID	Description	Edge
258	gpio3	D
259	gpio4	D
260	gpio5	D
261	gpio6	D
262	gpio7	D

Table 4-38. Event Logger Input Sources *(Cont.)*

ID	Description	Edge	ID	Description	Edge	ID	Description	Edge
28	sysclk_err	D	164	dsp_irq5	S	263	gpio8	D
29	asyncclk_err	D	165	dsp_irq6	S	264	gpio9	D
30	dspclk_err	D	166	dsp_irq7	S	265	gpio10	D
32	frame_start_g1r1	S	167	dsp_irq8	S	266	gpio11	D
33	frame_start_g1r2	S	168	dsp_irq9	S	267	gpio12	D
34	frame_start_g1r3	S	169	dsp_irq10	S	268	gpio13	D
40	frame_start_g2r1_sys	S	170	dsp_irq11	S	269	gpio14	D
41	frame_start_g2r2_sys	S	171	dsp_irq12	S	270	gpio15	D
80	hpdet	S	172	dsp_irq13	S	271	gpio16	D
88	micdet1	S	173	dsp_irq14	S	320	Timer1	S
89	micdet2	S	174	dsp_irq15	S	336	event1_not_empty	S
96	jd1_rise	S	175	dsp_irq16	S	352	event1_full	S
97	jd1_fall	S	176	hp1l_sc	S	368	event1_wmark	S
98	jd2_rise	S	177	hp1r_sc	S	384	dsp1_dma	S
99	jd2_fall	S	178	hp2l_sc	S	416	dsp1_start1	S
100	micd_clamp_rise	S	179	hp2r_sc	S	432	dsp1_start2	S
101	micd_clamp_fall	S	180	hp3l_sc	S	448	dsp1_start	S
104	jd3_rise	S	181	hp3r_sc	S	464	dsp1_busy	D
105	jd3_fall	S	184	hp4l_sc	S	512	dsp1_bus_err	S
128	drc1_sig_det	D	185	hp4r_sc	S	560	alarm1_ch1	S
129	drc2_sig_det	D	236	dfc_saturate	S	561	alarm1_ch2	S
130	inputs_sig_det	D	256	gpio1	D	562	alarm1_ch3	S
136	asrc1_in1_lock	D	257	gpio2	D	563	alarm1_ch4	S

4.5.2 Alarm Generators

The CS42L92 provides four alarm-generator circuits are associated with the general-purpose timer. These can be used to generate interrupt events according to the count value of the timer. The alarm interrupts can be either one-off events, or can be configured for cyclic (repeated) triggers.

4.5.2.1 Alarm Control

The alarm is enabled by writing 1 to the ALM1_CH*n*_START bit (where *n* identifies the respective alarm, 1–4). The alarm is disabled by writing 1 to ALM1_CH*n*_STOP.

The operating mode of each alarm is configured using ALM1_CH*n*_TRIG_MODE. In each mode, the alarm events are controlled by the alarm-trigger value, ALM1_CH*n*_TRIG_VAL.

- In Absolute Mode, the alarm output is triggered when the timer count value is equal to the alarm trigger value.
- In Relative Mode, the alarm output is triggered when the timer count value has incremented by a number equal to the alarm trigger value—this mode counts the number of clock cycles after the ALM1_CH*n*_START bit is written.
- In Combination Mode, the alarm output is initially triggered as described for the Absolute Mode; the alarm then operates as described for the Relative Mode.

When the alarm output is triggered, an output signal is asserted for the respective alarm. The output is asserted for a duration that is configured using ALM1_CH*n*_PULSE_DUR. The resulting signal can be output directly on a GPIO pin.

If an alarm is enabled and an update is written to ALM1_CH*n*_TRIG_VAL or ALM1_CH*n*_PULSE_DUR, the new value is loaded into the respective control register, but does not reconfigure the alarm immediately. If the ALM1_CH*n*_UPD bit is set, the alarm-trigger and pulse-duration values are updated when the alarm is next triggered. The alarm-trigger and pulse-duration settings can also be updated by writing 1 to ALM1_CH*n*_START.

Note that, if an alarm is enabled, the general-purpose timer must be configured for continuous, count-up operation. The TIMER1_MAX_COUNT value must be greater than the respective ALM1_CH*n*_TRIG_VAL setting.

4.5.2.2 Interrupts and GPIO Output

The alarm generators provide input to the interrupt control circuit and can be used to trigger an interrupt event when the alarm-trigger conditions are met. An interrupt event is triggered on the rising edge of the alarm output signal.

The alarm output status bits, TIMER_ALM1_CH*n*_STS*x*, are asserted for a duration that is configured using ALM1_CH*n*_PULSE_DUR. Note that the TIMER_ALM1_CH*n*_STS1 and TIMER_ALM1_CH*n*_STS2 bits provide the same information.

See Section 4.15 for details of the CS42L92 interrupt controller.

The alarm status can be output directly on a GPIO pin as an external indication of the alarm events. See Section 4.14 to configure a GPIO pin for this function.

4.5.2.3 Alarm Control Registers

The alarm control registers are described in Table 4-39.

Table 4-39. Alarm (ALM1) Control

Register Address	Bit	Label	Default	Description
R303104 (0x4A000) ALM1_CFG	0	ALM1_TIMER_SEL	0	Alarm block ALM1 timer source select 0 = Timer 1 All other codes are reserved. All ALM1 channels must be disabled when updating this register.
R303120 (0x4A010) ALM1_CONFIG1	4	ALM1_CH1_CONT	0	Channel 1 continuous mode select 0 = Single mode 1 = Continuous mode Channel 1 must be disabled (ALM1_CH1_STS = 0) when updating this field.
	1:0	ALM1_CH1_TRIG_MODE[1:0]	00	Channel 1 trigger mode select 00 = Absolute Mode: Alarm is triggered when the count value of the timer source is equal to ALM1_CH1_TRIG_VAL. 01 = Relative Mode: Alarm is triggered when the count value has incremented by a number equal to ALM1_CH1_TRIG_VAL. 10 = Combination Mode: Alarm is initially triggered as described for Absolute Mode; the alarm then operates as described for Relative Mode. 11 = Reserved Channel 1 must be disabled (ALM1_CH1_STS = 0) when updating this field.
R303122 (0x4A012) ALM1_CTRL1	15	ALM1_CH1_UPD	0	Channel 1 update control—Write 1 to indicate a new trigger value or pulse duration is ready to be applied. If Channel 1 is enabled and ALM1_CH1_UPD is set, the ALM1_CH1_TRIG_VAL and ALM1_CH1_PULSE_DUR settings are updated when the alarm is next triggered or by writing 1 to ALM1_CH1_START. If Channel 1 is disabled, the ALM1_CH1_UPD bit has no effect, and the ALM1_CH1_TRIG_VAL and ALM1_CH1_PULSE_DUR settings are updated immediately when writing to the respective fields.
	4	ALM1_CH1_STOP	—	Channel 1 stop control—Write 1 to disable Channel 1
	0	ALM1_CH1_START	—	Channel 1 start control—Write 1 to enable or restart Channel 1
R303124 (0x4A014) ALM1_TRIG_VAL1	31:0	ALM1_CH1_TRIG_VAL[31:0]	0x0000_0000	Channel 1 alarm trigger value
R303126 (0x4A016) ALM1_PULSE_DUR1	31:0	ALM1_CH1_PULSE_DUR[31:0]	0x0000_0000	Channel 1 alarm output pulse duration The pulse duration is referenced to the count rate of the selected timer source
R303128 (0x4A018) ALM1_STATUS1	0	ALM1_CH1_STS	0	Channel 1 status 0 = Disabled 1 = Enabled

Table 4-39. Alarm (ALM1) Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R303136 (0x4A020) ALM1_CONFIG2	4	ALM1_CH2_ CONT	0	Channel 2 continuous mode select 0 = Single mode 1 = Continuous mode Channel 2 must be disabled (ALM1_CH2_STS = 0) when updating this field.
	1:0	ALM1_CH2_ TRIG_MODE[1:0]	00	Channel 2 trigger mode select 00 = Absolute Mode: Alarm is triggered when the count value of the timer source is equal to ALM1_CH2_TRIG_VAL. 01 = Relative Mode: Alarm is triggered when the count value has incremented by a number equal to ALM1_CH2_TRIG_VAL. 10 = Combination Mode: Alarm is initially triggered as described for Absolute Mode; the alarm then operates as described for Relative Mode. 11 = Reserved Channel 2 must be disabled (ALM1_CH2_STS = 0) when updating this field.
R303138 (0x4A022) ALM1_CTRL2	15	ALM1_CH2_UPD	0	Channel 2 update control—Write 1 to indicate a new trigger value or pulse duration is ready to be applied. If Channel 2 is enabled and ALM1_CH2_UPD is set, the ALM1_CH2_TRIG_VAL and ALM1_CH2_PULSE_DUR settings are updated when the alarm is next triggered or by writing 1 to ALM1_CH2_START. If Channel 2 is disabled, the ALM1_CH2_UPD bit has no effect, and the ALM1_ CH2_TRIG_VAL and ALM1_CH2_PULSE_DUR settings are updated immediately when writing to the respective fields.
	4	ALM1_CH2_ STOP	—	Channel 2 stop control—Write 1 to disable Channel 2
	0	ALM1_CH2_ START	—	Channel 2 start control—Write 1 to enable or restart Channel 2
R303140 (0x4A024) ALM1_TRIG_VAL2	31:0	ALM1_CH2_ TRIG_VAL[31:0]	0x0000 _0000	Channel 2 alarm trigger value
R303142 (0x4A026) ALM1_PULSE_DUR2	31:0	ALM1_CH2_ PULSE_ DUR[31:0]	0x0000 _0000	Channel 2 alarm output pulse duration The pulse duration is referenced to the count rate of the selected timer source
R303144 (0x4A028) ALM1_STATUS2	0	ALM1_CH2_STS	0	Channel 2 status 0 = Disabled 1 = Enabled
R303152 (0x4A030) ALM1_CONFIG3	4	ALM1_CH3_ CONT	0	Channel 3 continuous mode select 0 = Single mode 1 = Continuous mode Channel 3 must be disabled (ALM1_CH3_STS = 0) when updating this field.
	1:0	ALM1_CH3_ TRIG_MODE[1:0]	00	Channel 3 trigger mode select 00 = Absolute Mode: Alarm is triggered when the count value of the timer source is equal to ALM1_CH3_TRIG_VAL. 01 = Relative Mode: Alarm is triggered when the count value has incremented by a number equal to ALM1_CH3_TRIG_VAL. 10 = Combination Mode: Alarm is initially triggered as described for Absolute Mode; the alarm then operates as described for Relative Mode. 11 = Reserved Channel 3 must be disabled (ALM1_CH3_STS = 0) when updating this field.
R303154 (0x4A032) ALM1_CTRL3	15	ALM1_CH3_UPD	0	Channel 3 update control—Write 1 to indicate a new trigger value or pulse duration is ready to be applied. If Channel 3 is enabled and ALM1_CH3_UPD is set, the ALM1_CH3_TRIG_VAL and ALM1_CH3_PULSE_DUR settings are updated when the alarm is next triggered or by writing 1 to ALM1_CH3_START. If Channel 3 is disabled, the ALM1_CH3_UPD bit has no effect, and the ALM1_ CH3_TRIG_VAL and ALM1_CH3_PULSE_DUR settings are updated immediately when writing to the respective fields.
	4	ALM1_CH3_ STOP	—	Channel 3 stop control—Write 1 to disable Channel 3
	0	ALM1_CH3_ START	—	Channel 3 start control—Write 1 to enable or restart Channel 3
R303156 (0x4A034) ALM1_TRIG_VAL3	31:0	ALM1_CH3_ TRIG_VAL[31:0]	0x0000 _0000	Channel 3 alarm trigger value

Table 4-39. Alarm (ALM1) Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R303158 (0x4A036) ALM1_PULSE_DUR3	31:0	ALM1_CH3_ PULSE_ DUR[31:0]	0x0000 _0000	Channel 3 alarm output pulse duration The pulse duration is referenced to the count rate of the selected timer source
R303160 (0x4A038) ALM1_STATUS3	0	ALM1_CH3_STS	0	Channel 3 status 0 = Disabled 1 = Enabled
R303168 (0x4A040) ALM1_CONFIG4	4	ALM1_CH4_ CONT	0	Channel 4 continuous mode select 0 = Single mode 1 = Continuous mode Channel 4 must be disabled (ALM1_CH4_STS = 0) when updating this field.
	1:0	ALM1_CH4_ TRIG_MODE[1:0]	00	Channel 4 trigger mode select 00 = Absolute Mode: Alarm is triggered when the count value of the timer source is equal to ALM1_CH4_TRIG_VAL. 01 = Relative Mode: Alarm is triggered when the count value has incremented by a number equal to ALM1_CH4_TRIG_VAL. 10 = Combination Mode: Alarm is initially triggered as described for Absolute Mode; the alarm then operates as described for Relative Mode. 11 = Reserved Channel 4 must be disabled (ALM1_CH4_STS = 0) when updating this field.
R303170 (0x4A042) ALM1_CTRL4	15	ALM1_CH4_UPD	0	Channel 4 update control—Write 1 to indicate a new trigger value or pulse duration is ready to be applied. If Channel 4 is enabled and ALM1_CH4_UPD is set, the ALM1_CH4_TRIG_VAL and ALM1_CH4_PULSE_DUR settings are updated when the alarm is next triggered or by writing 1 to ALM1_CH4_START. If Channel 4 is disabled, the ALM1_CH4_UPD bit has no effect, and the ALM1_ CH4_TRIG_VAL and ALM1_CH4_PULSE_DUR settings are updated immediately when writing to the respective fields.
	4	ALM1_CH4_ STOP	—	Channel 4 stop control—Write 1 to disable Channel 4
	0	ALM1_CH4_ START	—	Channel 4 start control—Write 1 to enable or restart Channel 4
R303172 (0x4A044) ALM1_TRIG_VAL4	31:0	ALM1_CH4_ TRIG_VAL[31:0]	0x0000 _0000	Channel 4 alarm trigger value
R303174 (0x4A046) ALM1_PULSE_DUR4	31:0	ALM1_CH4_ PULSE_ DUR[31:0]	0x0000 _0000	Channel 4 alarm output pulse duration The pulse duration is referenced to the count rate of the selected timer source
R303176 (0x4A048) ALM1_STATUS4	0	ALM1_CH4_STS	0	Channel 4 status 0 = Disabled 1 = Enabled

4.5.3 General-Purpose Timer

The CS42L92 incorporates a general-purpose timer, which supports a wide variety of uses. The general-purpose timer provides time-stamp data for the event logger; it also supports the watchdog and other miscellaneous time-based functions, providing additional capability for signal-processing applications.

4.5.3.1 Overview

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

The timer 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 the timer is selected using TIMER1_REFCLK_SRC.

If SYSCLK, ASYNCCLK, or DSPCLK is selected, a lower clock frequency, derived from the applicable system clock, can be selected using the TIMER1_REFCLK_FREQ_SEL field (for SYSCLK or ASYNCCLK source) or the TIMER1_DSPCLK_FREQ_SEL field (for DSPCLK source). The applicable division ratio is determined automatically, assuming the respective clock source has been correctly configured as described in Section 4.16.

Note that, depending on the DSPCLK frequency and the available clock dividers, the timer reference clock may differ from the selected clock if DSPCLK is the selected source. In most cases, the reference clock frequency equals or exceeds the requested frequency. A lower frequency is implemented if limited by either the DSPCLK frequency or the maximum TIMER1 clocking frequency.

If any source other than DSPCLK is selected, the clock can be further divided using TIMER1_REFCLK_DIV. Division ratios in the range 1 to 128 can be selected.

Note that, if DSPCLK is enabled, the CS42L92 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 TIMER1_REFCLK_FREQ_SEL and after TIMER1_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 TIMER1_REFCLK_DIV field can be used to ensure that these criteria are met.

One final division, controlled by TIMER1_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 TIMER1_COUNT field is incremented (or decremented).

The maximum count value of the timer is determined by the TIMER1_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 TIMER1_CUR_COUNT field.

The timer is started by writing 1 to TIMER1_START. Note that, if the timer is already running, it restarts from its initial value. The timer is stopped by writing 1 to TIMER1_STOP. The count direction (up or down) is selected using the TIMER1_DIR bit.

The TIMER1_CONTINUOUS bit selects whether the timer automatically restarts after the end-of-count condition has been reached. The TIMER1_RUNNING_STS indicates whether the timer is running, or if it has stopped.

Note that the timer should be stopped before making any changes to the timer control registers. The timer configuration should only be changed if TIMER1_RUNNING_STS = 0.

4.5.3.3 Interrupts and GPIO Output

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.

4.5.3.4 Timer Block Diagram and Control Registers

The timer block is shown in Fig. 4-35.

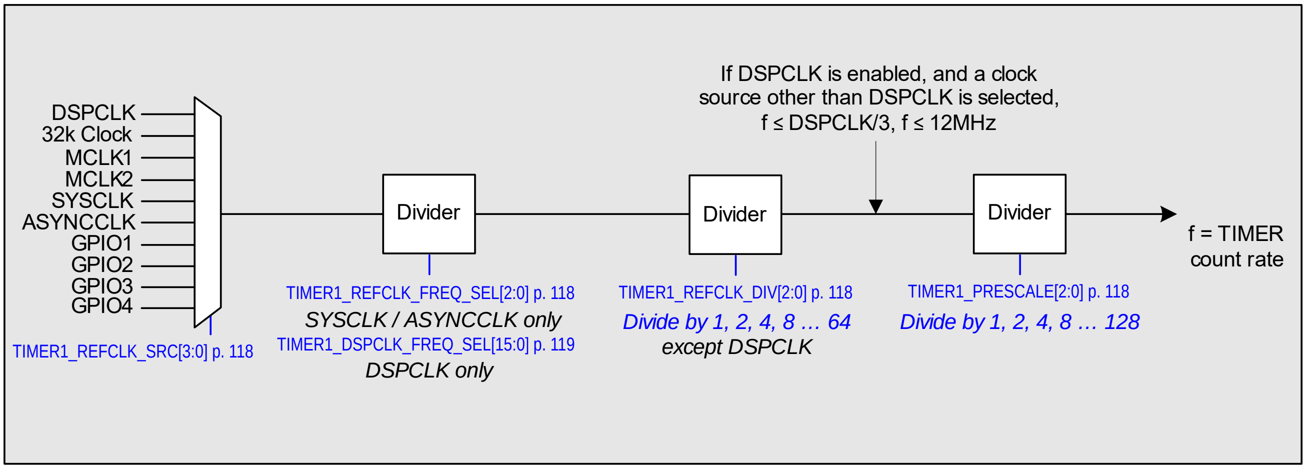


Figure 4-35. General-Purpose Timer

The timer control registers are described in Table 4-40.

Table 4-40. General-Purpose Timer (TIMER1) Control

Register Address	Bit	Label	Default	Description
R311296 (0x4_C000) Timer1_Control	21	TIMER1_ CONTINUOUS	0	Timer Continuous Mode select 0 = Single mode 1 = Continuous mode Timer must be stopped (TIMER1_RUNNING_STS = 0) when updating this field
	20	TIMER1_DIR	0	Timer Count Direction 0 = Down 1 = Up Timer must be stopped (TIMER1_RUNNING_STS = 0) when updating this field
	18:16	TIMER1_ 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 (TIMER1_RUNNING_STS = 0) when updating this field
	14:12	TIMER1_ 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 the event logger. In this case, the divider output corresponds to the frequency of event logging opportunities on the respective modules. Timer must be stopped (TIMER1_RUNNING_STS = 0) when updating this field
	10:8	TIMER1_ REFCLK_FREQ_ SEL[2:0]	000	Timer Reference Frequency Select (SYSCLK or ASYNCCLK source) 000 = 6.144 MHz (5.6448 MHz); 010 = 24.576 MHz (22.5792 MHz) 001 = 12.288 MHz (11.2896 MHz); 011 = 49.152 MHz (45.1584 MHz) All other codes are reserved. The selected frequency must be less than or equal to the frequency of the source. Timer must be stopped (TIMER1_RUNNING_STS = 0) when updating this field.
	3:0	TIMER1_ REFCLK_ SRC[3:0]	0000	Timer Reference Source Select. Timer must be stopped (TIMER1_RUNNING_STS=0) when updating this field. 0000 = DSPCLK; 1000 = SYSCLK; 1110 = GPIO3 0001 = 32-kHz clock; 1001 = ASYNCCLK; 1111 = GPIO4 0100 = MCLK1; 1100 = GPIO1; All other codes are reserved. 0101 = MCLK2; 1101 = GPIO2
R311298 (0x4_C002) Timer1_Count_Preset	31:0	TIMER1_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 (TIMER1_RUNNING_STS = 0) when updating this field.
R311302 (0x4_C006) Timer1_Start_and_ Stop	4	TIMER1_STOP	0	Timer Stop Control Write 1 to stop.
	0	TIMER1_START	0	Timer Start Control Write 1 to start. If the timer is already running, it restarts from its initial value.
R311304 (0x4_C008) Timer1_Status	0	TIMER1_ RUNNING_STS	0	Timer Running Status 0 = Timer stopped 1 = Timer running

Table 4-40. General-Purpose Timer (TIMER1) Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R311306 (0x4_C00A) Timer1_Count_ Readback	31:0	TIMER1_CUR_ COUNT[31:0]	0x0000	Timer Current Count value
R311308 (0x4_C00C) Timer1_DSP_Clock_ Config	15:0	TIMER1_ DSPCLK_FREQ_ SEL[15:0]	0x0000	Timer Reference Frequency Select (DSPCLK source) Coded as LSB = 1/64 MHz, Valid from 5.6 MHz to 148 MHz. The timer reference frequency must be less than or equal to the DSPCLK frequency. The timer reference is generated by division of DSPCLK, and may differ from the selected frequency. The timer reference frequency can be read from TIMER1_DSPCLK_FREQ_STS. Timer must be stopped (TIMER1_RUNNING_STS=0) when updating this field.
R311310 (0x4_C00E) Timer1_DSP_Clock_ Status	15:0	TIMER1_ DSPCLK_FREQ_ STS[15:0]	0x0000	Timer Reference Frequency (Read only) Only valid when DSPCLK is the selected clock source. Coded as LSB = 1/64 MHz.

4.5.4 DSP GPIO

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

4.5.4.1 Overview

The CS42L92 supports up to 16 GPIO pins, which can be assigned to application-specific functions. There are 2 dedicated GPIO pins; the remaining 14 GPIOs 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 function; this provides a highly efficient mechanism for the 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 eight 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*_SET*x*_DIR) and level control bits (DSPGP*n*_SET*x*_LVL) is only valid when the channel (DSPGP*n*) 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*_SET*x*_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. 4-36, 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 GPIO pin, 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 4-93 for details of the GPIO pull-up and pull-down control bits.

4.5.4.4 DSP GPIO Block Diagram and Control Registers

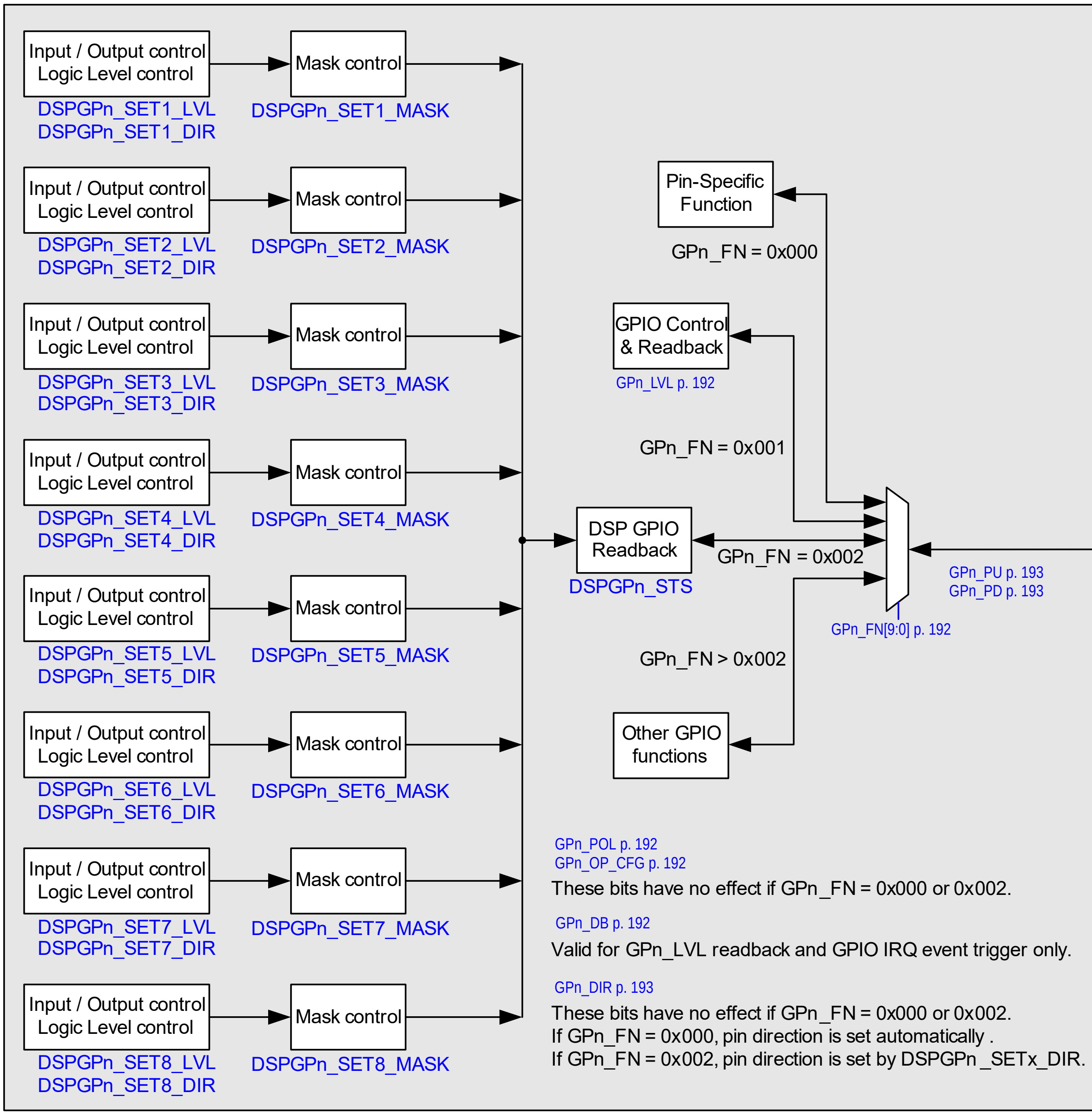


Figure 4-36. DSP GPIO Control

The control registers associated with the DSP GPIO are described in Table 4-41.

Table 4-41. 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 R315552 (0x4_D0A0) DSPGP_SET5_Mask_1 R315584 (0x4_D0C0) DSPGP_SET6_Mask_1 R315616 (0x4_D0E0) DSPGP_SET7_Mask_1 R315648 (0x4_D100) DSPGP_SET8_Mask_1	15	DSPGP16_SETn_MASK	1	DSP SETn GPIO16 Mask Control 0 = Unmasked, 1 = Masked A GPIO pin should be unmasked in a maximum of one SET at any time.
	14	DSPGP15_SETn_MASK	1	DSP SETn GPIO15 Mask Control
	13	DSPGP14_SETn_MASK	1	DSP SETn GPIO14 Mask Control
	12	DSPGP13_SETn_MASK	1	DSP SETn GPIO13 Mask Control
	11	DSPGP12_SETn_MASK	1	DSP SETn GPIO12 Mask Control
	10	DSPGP11_SETn_MASK	1	DSP SETn GPIO11 Mask Control
	9	DSPGP10_SETn_MASK	1	DSP SETn GPIO10 Mask Control
	8	DSPGP9_SETn_MASK	1	DSP SETn GPIO9 Mask Control
	7	DSPGP8_SETn_MASK	1	DSP SETn GPIO8 Mask Control
	6	DSPGP7_SETn_MASK	1	DSP SETn GPIO7 Mask Control
	5	DSPGP6_SETn_MASK	1	DSP SETn GPIO6 Mask Control
	4	DSPGP5_SETn_MASK	1	DSP SETn GPIO5 Mask Control
	3	DSPGP4_SETn_MASK	1	DSP SETn GPIO4 Mask Control
	2	DSPGP3_SETn_MASK	1	DSP SETn GPIO3 Mask Control
	1	DSPGP2_SETn_MASK	1	DSP SETn GPIO2 Mask Control
	0	DSPGP1_SETn_MASK	1	DSP SETn 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 R315560 (0x4_D0A8) DSPGP_SET5_Direction_1 R315592 (0x4_D0C8) DSPGP_SET6_Direction_1 R315624 (0x4_D0E8) DSPGP_SET7_Direction_1 R315656 (0x4_D108) DSPGP_SET8_Direction_1	15	DSPGP16_SETn_DIR	1	DSP SETn GPIO16 Direction Control 0 = Output, 1 = Input
	14	DSPGP15_SETn_DIR	1	DSP SETn GPIO15 Direction Control
	13	DSPGP14_SETn_DIR	1	DSP SETn GPIO14 Direction Control
	12	DSPGP13_SETn_DIR	1	DSP SETn GPIO13 Direction Control
	11	DSPGP12_SETn_DIR	1	DSP SETn GPIO12 Direction Control
	10	DSPGP11_SETn_DIR	1	DSP SETn GPIO11 Direction Control
	9	DSPGP10_SETn_DIR	1	DSP SETn GPIO10 Direction Control
	8	DSPGP9_SETn_DIR	1	DSP SETn GPIO9 Direction Control
	7	DSPGP8_SETn_DIR	1	DSP SETn GPIO8 Direction Control
	6	DSPGP7_SETn_DIR	1	DSP SETn GPIO7 Direction Control
	5	DSPGP6_SETn_DIR	1	DSP SETn GPIO6 Direction Control
	4	DSPGP5_SETn_DIR	1	DSP SETn GPIO5 Direction Control
	3	DSPGP4_SETn_DIR	1	DSP SETn GPIO4 Direction Control
	2	DSPGP3_SETn_DIR	1	DSP SETn GPIO3 Direction Control
	1	DSPGP2_SETn_DIR	1	DSP SETn GPIO2 Direction Control
	0	DSPGP1_SETn_DIR	1	DSP SETn GPIO1 Direction Control

Table 4-41. 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 R315568 (0x4_D0B0) DSPGP_SET5_Level_1 R315600 (0x4_D0D0) DSPGP_SET6_Level_1 R315632 (0x4_D0F0) DSPGP_SET7_Level_1 R315664 (0x4_D110) DSPGP_SET8_Level_1	15	DSPGP16_SETn_LVL	0	DSP SETn GPIO16 Output Level 0 = Logic 0, 1 = Logic 1
	14	DSPGP15_SETn_LVL	0	DSP SETn GPIO15 Output Level
	13	DSPGP14_SETn_LVL	0	DSP SETn GPIO14 Output Level
	12	DSPGP13_SETn_LVL	0	DSP SETn GPIO13 Output Level
	11	DSPGP12_SETn_LVL	0	DSP SETn GPIO12 Output Level
	10	DSPGP11_SETn_LVL	0	DSP SETn GPIO11 Output Level
	9	DSPGP10_SETn_LVL	0	DSP SETn GPIO10 Output Level
	8	DSPGP9_SETn_LVL	0	DSP SETn GPIO9 Output Level
	7	DSPGP8_SETn_LVL	0	DSP SETn GPIO8 Output Level
	6	DSPGP7_SETn_LVL	0	DSP SETn GPIO7 Output Level
	5	DSPGP6_SETn_LVL	0	DSP SETn GPIO6 Output Level
	4	DSPGP5_SETn_LVL	0	DSP SETn GPIO5 Output Level
	3	DSPGP4_SETn_LVL	0	DSP SETn GPIO4 Output Level
	2	DSPGP3_SETn_LVL	0	DSP SETn GPIO3 Output Level
	1	DSPGP2_SETn_LVL	0	DSP SETn GPIO2 Output Level
	0	DSPGP1_SETn_LVL	0	DSP SETn GPIO1 Output Level

4.6 Digital Audio Interface

The CS42L92 provides three audio interfaces, AIF1–AIF3. Each interface is independently configurable on the respective transmit (TX) and receive (RX) paths. Each AIF supports up to eight channels of input and output signal paths.

The data sources for the audio interface transmit (TX) paths can be selected from any of the CS42L92 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. 4-37.

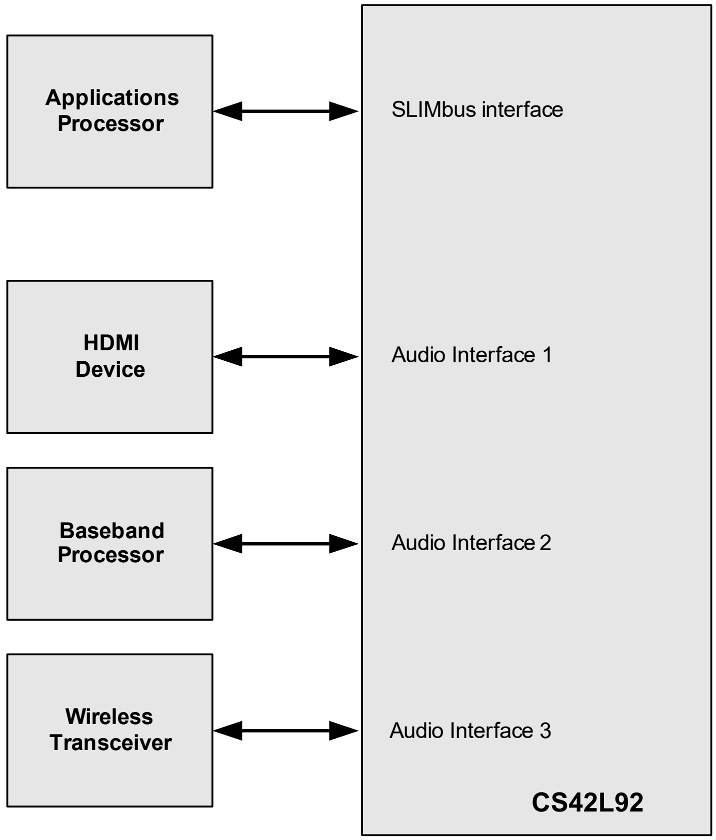


Figure 4-37. Typical AIF Connections

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 CS42L92. 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
- I²S
- Left-justified

The left-justified and DSP-B formats are valid in Master Mode only (i.e., BCLK and LRCLK are outputs from the CS42L92). 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. Mono PCM operation can be supported using the DSP modes. Refer to Table 3-16 through Table 3-18 for signal timing information.

For typical applications, AIF data is encoded in 2's complement (signed, fixed-point) format. This format is compatible with all of the digital mixing and signal-processing functions on the CS42L92. Other data types, including floating point formats, can be supported using the DFCs. Note that, if unsigned or floating point data is present within the digital core, some restrictions on the valid signal routing options apply—see Section 4.3.13.

4.6.1 Master and Slave Mode Operation

The CS42L92 digital audio interfaces can operate as a master or slave, as shown in Fig. 4-38 and Fig. 4-39. The associated control bits are described in Section 4.7.

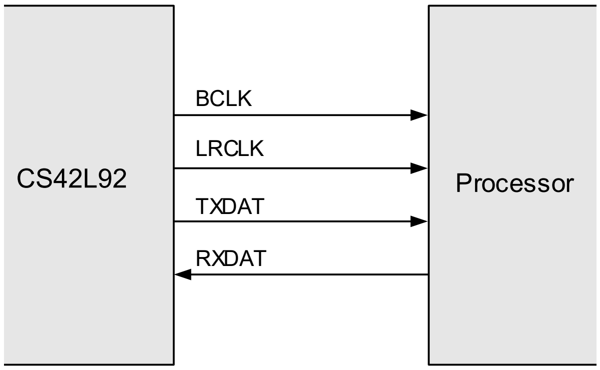


Figure 4-38. Master Mode

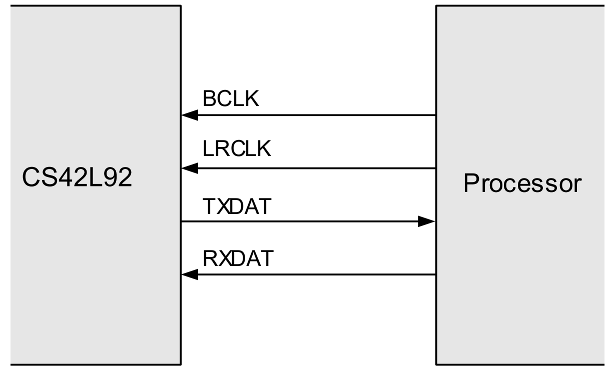


Figure 4-39. Slave Mode

4.6.2 Audio Data Formats

The CS42L92 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 CS42L92).

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 CS42L92 are described as follows. Note that the polarity of the BCLK and LRCLK signals can be inverted if required; the following descriptions all assume the default, noninverted polarity 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. 4-40 and Fig. 4-41. 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. CS42L92 data that is output on the left channel is read as mono data by the receiving equipment. Mono PCM data received by the CS42L92 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. 4-40.

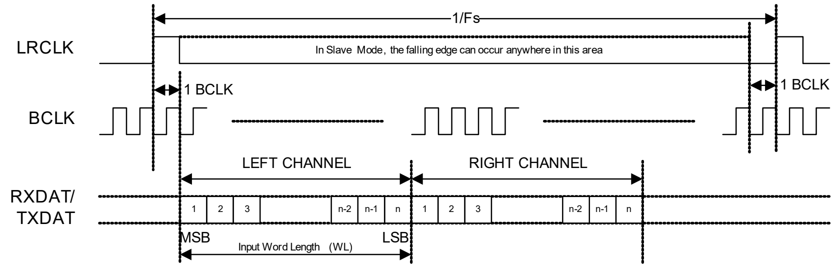


Figure 4-40. DSP Mode A Data Format

DSP Mode B data format is shown in Fig. 4-41.

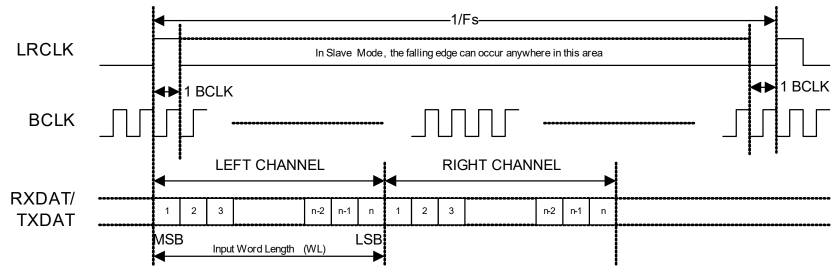


Figure 4-41. DSP Mode B Data Format

- 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. 4-42.

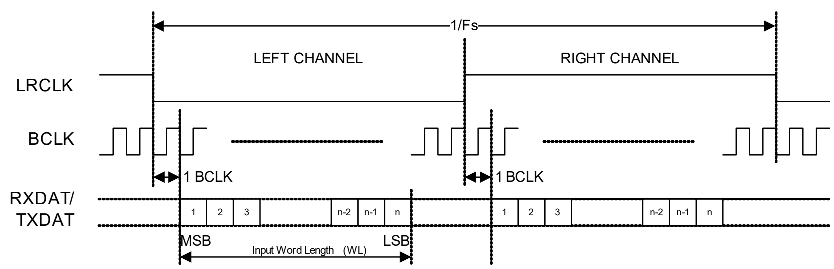


Figure 4-42. I2S Data Format (Assuming n-Bit Word Length)

- 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. 4-43.

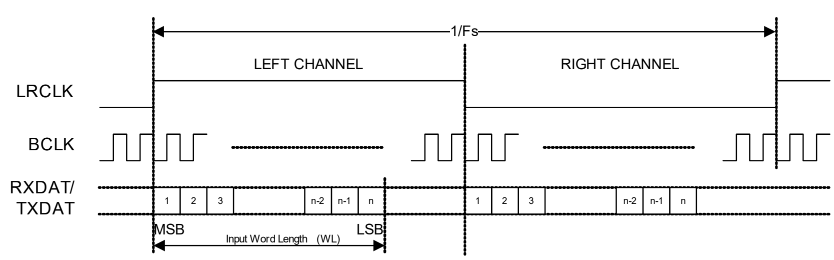


Figure 4-43. Left-Justified Data Format (Assuming n-Bit Word Length)

4.6.3 AIF Time-Slot Configuration

Multichannel operation is supported on AIF1–AIF3, with up to eight channels of input and output on each. 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. Note that, on each interface, all input and output channels must operate at the same sample rate (Fs).

Each audio channel 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 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 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. 4-44 through Fig. 4-47. One example is shown for each of the four possible data formats.

Fig. 4-44 shows an example of DSP Mode A format. Four enabled audio channels are shown, allocated to time slots 0 through 3.

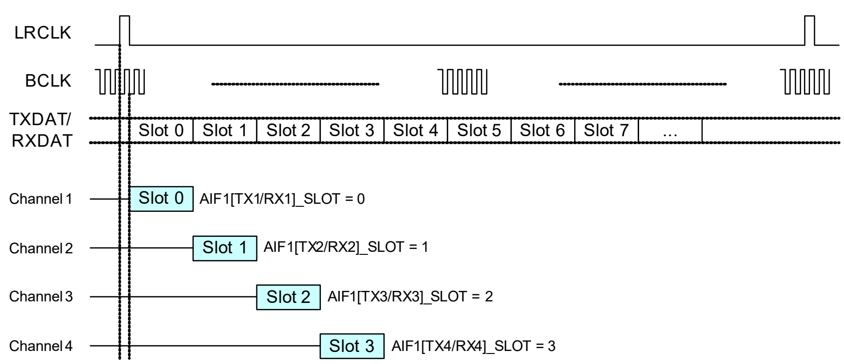


Figure 4-44. DSP Mode A Example

Fig. 4-45 shows an example of DSP Mode B format. Six enabled audio channels are shown, with time slots 4 and 5 unused.

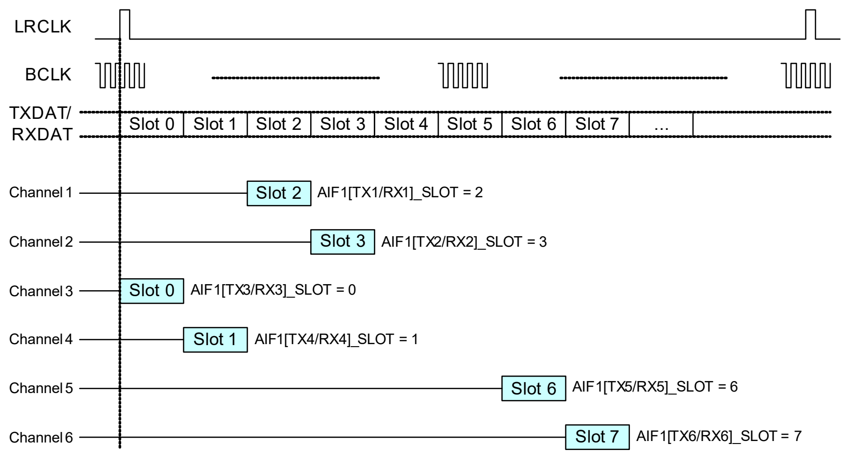


Figure 4-45. DSP Mode B Example

Fig. 4-46 shows an example of I2S format. Four enabled channels are shown, allocated to time slots 0 through 3.

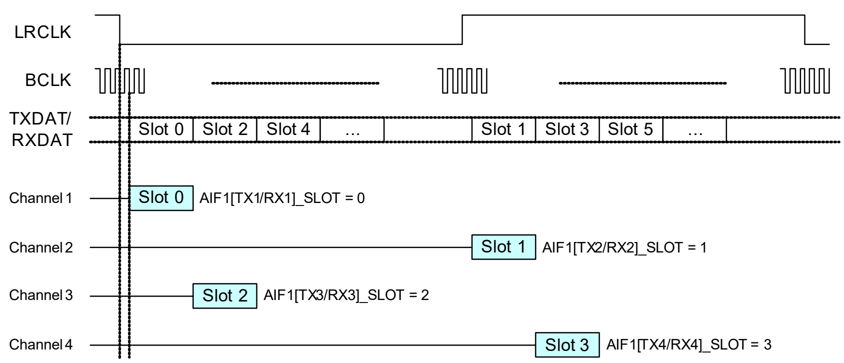


Figure 4-46. I2S Example

Fig. 4-47 shows an example of left-justified format. Six enabled channels are shown.

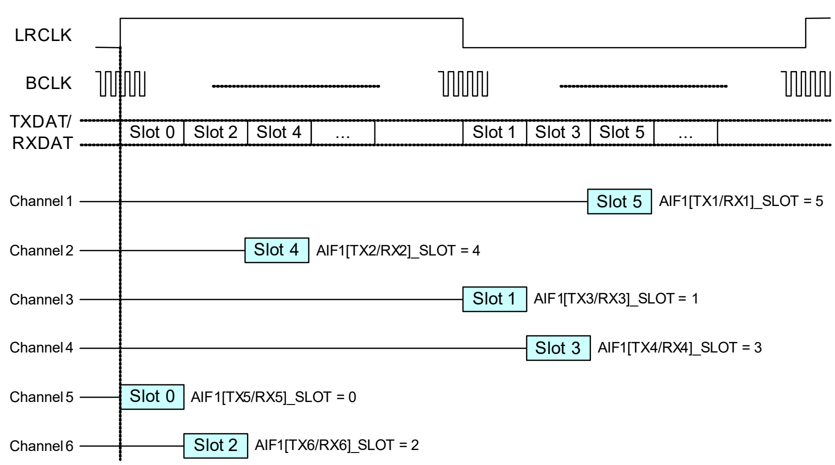


Figure 4-47. Left-Justified Example

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 audio channel in turn.

This form of TDM is implemented between two devices, using the electrical connections shown Fig. 4-38 or Fig. 4-39.

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. 4-48, Fig. 4-49, and Fig. 4-50.

The CS42L92 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. 4-48, Fig. 4-49, and Fig. 4-50.

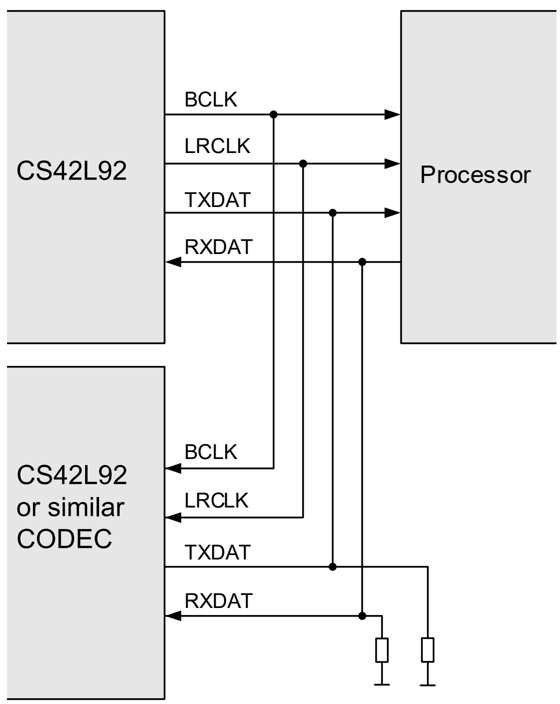


Figure 4-48. TDM with CS42L92 as Master

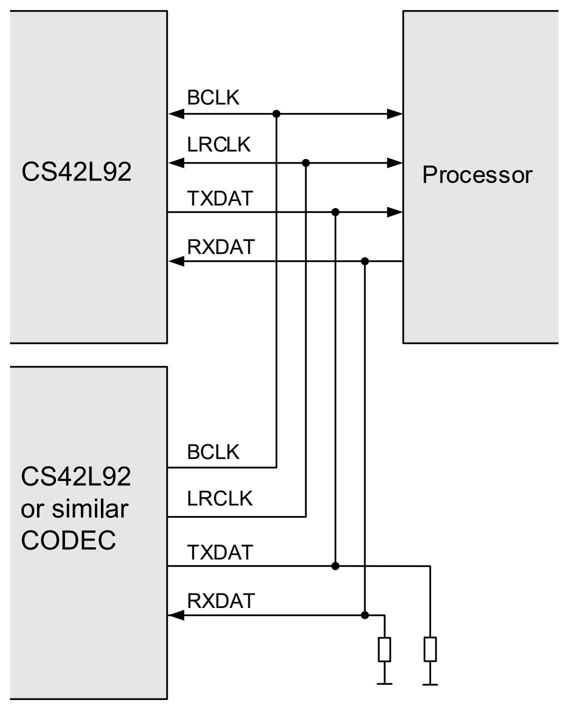


Figure 4-49. TDM with Other Codec as Master

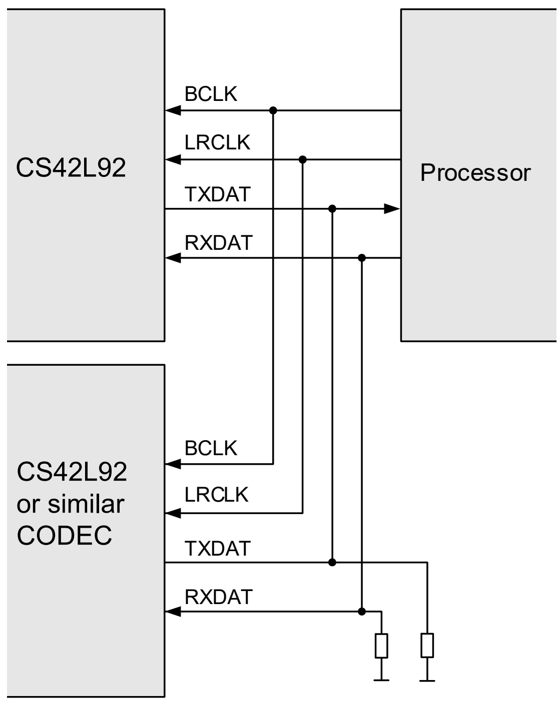


Figure 4-50. TDM with Processor as Master

4.7 Digital Audio Interface Control

This section describes the configuration of the CS42L92 digital audio interface paths.

Each AIF supports up to eight input signal paths and up to eight 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 AIF1 and AIF3 interfaces provide full support for 32-bit data words (input and output). Audio data samples up to 32 bits can be routed to the AIF1, AIF3, SLIMbus, S/PDIF, and DAC output paths. Note that other signal paths and signal-processing blocks within the digital core are limited to 24-bit data length; data samples are truncated to 24-bit length if they are routed through any function that does not support 32-bit data words.

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 CS42L92 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 4-26.

Note that sample-rate conversion is required when routing the AIF paths to any signal chain that is asynchronous or 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 multifunction 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 pull-up and pull-down resistor can be configured independently using the fields described in Table 4-93.

If the pull-up and pull-down resistors are both enabled, the CS42L92 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 CS42L92 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 CS42L92 when one or more AIF*n* channels is enabled.

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

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

Master Mode is selected on the AIF*n*LRCLK pin by setting AIF*n*_LRCLK_MSTR. In Master Mode, the AIF*n*LRCLK signal is generated by the CS42L92 when one or more AIF*n* channels is enabled.

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

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

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

The AIF1 master/slave control registers are described in Table 4-42.

Table 4-42. 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 4-42. AIF1 Master/Slave Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R1282 (0x0502) AIF1_Rx_Pin_Ctrl	4	AIF1_ LRCLK_ ADV	0	AIF1 Audio Interface LRCLK Advance 0 = Normal 1 = AIF1LRCLK transition is advanced to the previous BCLK phase
	2	AIF1_ LRCLK_INV	0	AIF1 Audio Interface LRCLK Invert 0 = AIF1LRCLK not inverted 1 = AIF1LRCLK inverted
	1	AIF1_ LRCLK_ FRC	0	AIF1 Audio Interface LRCLK Output Control 0 = Normal 1 = AIF1LRCLK always enabled in Master Mode
	0	AIF1_ LRCLK_ MSTR	0	AIF1 Audio Interface LRCLK Master Select 0 = AIF1LRCLK Slave Mode 1 = AIF1LRCLK Master Mode

The AIF2 master/slave control registers are described in Table 4-43.

Table 4-43. AIF2 Master/Slave Control

Register Address	Bit	Label	Default	Description
R1344 (0x0540) AIF2_BCLK_Ctrl	7	AIF2_BCLK_ INV	0	AIF2 Audio Interface BCLK Invert 0 = AIF2BCLK not inverted 1 = AIF2BCLK inverted
	6	AIF2_BCLK_ FRC	0	AIF2 Audio Interface BCLK Output Control 0 = Normal 1 = AIF2BCLK always enabled in Master Mode
	5	AIF2_BCLK_ MSTR	0	AIF2 Audio Interface BCLK Master Select 0 = AIF2BCLK Slave Mode 1 = AIF2BCLK Master Mode
R1346 (0x0542) AIF2_Rx_Pin_Ctrl	4	AIF2_ LRCLK_ADV	0	AIF2 Audio Interface LRCLK Advance 0 = Normal 1 = AIF2LRCLK transition is advanced to the previous BCLK phase
	2	AIF2_ LRCLK_INV	0	AIF2 Audio Interface LRCLK Invert 0 = AIF2LRCLK not inverted 1 = AIF2LRCLK inverted
	1	AIF2_ LRCLK_FRC	0	AIF2 Audio Interface LRCLK Output Control 0 = Normal 1 = AIF2LRCLK always enabled in Master Mode
	0	AIF2_ LRCLK_ MSTR	0	AIF2 Audio Interface LRCLK Master Select 0 = AIF2LRCLK Slave Mode 1 = AIF2LRCLK Master Mode

The AIF3 master/slave control registers are described in Table 4-44.

Table 4-44. 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 4-44. AIF3 Master/Slave Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R1410 (0x0582) AIF3_Rx_Pin_Ctrl	4	AIF3_ LRCLK_ADV	0	AIF3 Audio Interface LRCLK Advance 0 = Normal 1 = AIF3LRCLK transition is advanced to the previous BCLK phase
	2	AIF3_ LRCLK_INV	0	AIF3 Audio Interface LRCLK Invert 0 = AIF3LRCLK not inverted 1 = AIF3LRCLK inverted
	1	AIF3_ LRCLK_FRC	0	AIF3 Audio Interface LRCLK Output Control 0 = Normal 1 = AIF3LRCLK always enabled in Master Mode
	0	AIF3_ LRCLK_ MSTR	0	AIF3 Audio Interface LRCLK Master Select 0 = AIF3LRCLK Slave Mode 1 = AIF3LRCLK Master Mode

4.7.3.1 LRCLK Advance

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

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

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

The adjusted interface timing (AIF*n*_LRCLK_ADV = 1), is shown in Fig. 4-51. 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.

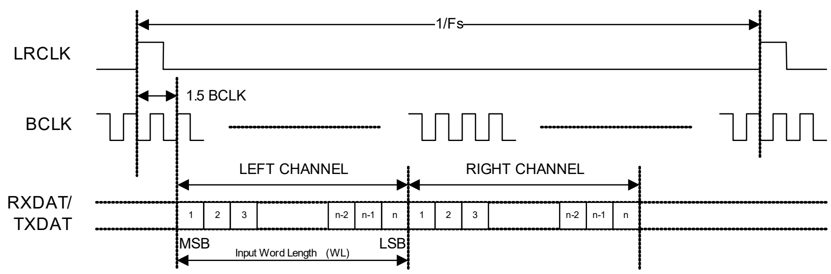


Figure 4-51. LRCLK advance—DSP-A Master Mode

4.7.4 AIF Signal Path Enable

The AIF1–AIF3 interfaces support up to eight input (RX) channels and up to eight output (TX) channels. Each channel is enabled or disabled using the bits defined in Table 4-45, Table 4-46 and Table 4-47.

The system clock, SYSCLK, must be configured and enabled before any audio path is enabled. The ASYNCCLK may also be required, depending on the path configuration. See Section 4.16 for details of the system clocks.

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

The CS42L92 performs automatic checks to confirm that the SYSCLK and ASYNCCLK frequencies are 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 4-45.

Table 4-45. AIF1 Signal Path Enable

Register Address	Bit	Label	Default	Description
R1305 (0x0519) AIF1_Tx_Enables	7	AIF1TX8_ENA	0	AIF1 Audio Interface TX Channel *n* Enable 0 = Disabled 1 = Enabled
	6	AIF1TX7_ENA	0	
	5	AIF1TX6_ENA	0	
	4	AIF1TX5_ENA	0	
	3	AIF1TX4_ENA	0	
	2	AIF1TX3_ENA	0	
	1	AIF1TX2_ENA	0	
	0	AIF1TX1_ENA	0	
R1306 (0x051A) AIF1_Rx_Enables	7	AIF1RX8_ENA	0	AIF1 Audio Interface RX Channel *n* Enable 0 = Disabled 1 = Enabled
	6	AIF1RX7_ENA	0	
	5	AIF1RX6_ENA	0	
	4	AIF1RX5_ENA	0	
	3	AIF1RX4_ENA	0	
	2	AIF1RX3_ENA	0	
	1	AIF1RX2_ENA	0	
	0	AIF1RX1_ENA	0	

The AIF2 signal-path-enable bits are described in Table 4-46.

Table 4-46. AIF2 Signal Path Enable

Register Address	Bit	Label	Default	Description
R1369 (0x0559) AIF2_Tx_Enables	7	AIF2TX8_ENA	0	AIF2 Audio Interface TX Channel *n* Enable 0 = Disabled 1 = Enabled
	6	AIF2TX7_ENA	0	
	5	AIF2TX6_ENA	0	
	4	AIF2TX5_ENA	0	
	3	AIF2TX4_ENA	0	
	2	AIF2TX3_ENA	0	
	1	AIF2TX2_ENA	0	
	0	AIF2TX1_ENA	0	
R1370 (0x055A) AIF2_Rx_Enables	7	AIF2RX8_ENA	0	AIF2 Audio Interface RX Channel *n* Enable 0 = Disabled 1 = Enabled
	6	AIF2RX7_ENA	0	
	5	AIF2RX6_ENA	0	
	4	AIF2RX5_ENA	0	
	3	AIF2RX4_ENA	0	
	2	AIF2RX3_ENA	0	
	1	AIF2RX2_ENA	0	
	0	AIF2RX1_ENA	0	

The AIF3 signal-path-enable bits are described in Table 4-47.

Table 4-47. AIF3 Signal Path Enable

Register Address	Bit	Label	Default	Description
R1433 (0x0599) AIF3_Tx_Enables	7	AIF3TX8_ENA	0	AIF3 Audio Interface TX Channel *n* Enable 0 = Disabled 1 = Enabled
	6	AIF3TX7_ENA	0	
	5	AIF3TX6_ENA	0	
	4	AIF3TX5_ENA	0	
	3	AIF3TX4_ENA	0	
	2	AIF3TX3_ENA	0	
	1	AIF3TX2_ENA	0	
	0	AIF3TX1_ENA	0	

Table 4-47. AIF3 Signal Path Enable *(Cont.)*

Register Address	Bit	Label	Default	Description
R1434 (0x059A) AIF3_Rx_Enables	7	AIF3RX8_ENA	0	AIF3 Audio Interface RX Channel *n* Enable 0 = Disabled 1 = Enabled
	6	AIF3RX7_ENA	0	
	5	AIF3RX6_ENA	0	
	4	AIF3RX5_ENA	0	
	3	AIF3RX4_ENA	0	
	2	AIF3RX3_ENA	0	
	1	AIF3RX2_ENA	0	
	0	AIF3RX1_ENA	0	

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, as described in Table 4-48 through Table 4-50.

- If AIFn_RATE < 1000 (Table 4-26), AIFn is referenced to the SYSCLK clocking domain and the applicable frequency depends upon the SAMPLE_RATE_1, SAMPLE_RATE_2 or SAMPLE_RATE_3 fields.
- If AIFn_RATE ≥ 1000, AIFn is referenced to the ASYNCCLK clocking domain and the applicable frequency depends upon the ASYNC_SAMPLE_RATE_1 or ASYNC_SAMPLE_RATE_2 fields.

The selected AIF*n*BCLK rate must be less than or equal to SYSCLK/2, or ASYNCCLK/2, as applicable. See Section 4.16 for details of SYSCLK and ASYNCCLK clock domains, 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 and 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 4-48.

Table 4-48. 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; 0x07 = 384 kHz (352.8 kHz); 0x0D = 3.072 MHz (2.8824 MHz) 0x02 = 64 kHz (58.8 kHz); 0x08 = 512 kHz (470.4 kHz); 0x0E = 4.096 MHz (3.7632 MHz) 0x03 = 96 kHz (88.2 kHz); 0x09 = 768 kHz (705.6 kHz); 0x0F = 6.144 MHz (5.6448 MHz) 0x04 = 128 kHz (117.6 kHz); 0x0A = 1.024 MHz (940.8 kHz); 0x10 = 8.192 MHz (7.5264 MHz) 0x05 = 192 kHz (176.4 kHz); 0x0B = 1.536 MHz (1.4112 MHz); 0x11 = 12.288 MHz (11.2896 MHz) 0x06 = 256 kHz (235.2 kHz); 0x0C = 2.048 MHz (1.8816 MHz); 0x12 = 24.576 MHz (22.5792 MHz) The frequencies in brackets apply for 44.1 kHz–related sample rates only. If AIF1_RATE < 1000, AIF1 is referenced to the SYSCLK clock domain. In this case, the 44.1 kHz–related frequencies apply if SAMPLE_RATE_n = 01XXX. If AIF1_RATE ≥ 1000, AIF1 is referenced to the ASYNCCLK clock domain. In this case, the 44.1 kHz–related frequencies apply if ASYNC_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 4-49.

Table 4-49. 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; 0x07 = 384 kHz (352.8 kHz); 0x0D = 3.072 MHz (2.8824 MHz) 0x02 = 64 kHz (58.8 kHz); 0x08 = 512 kHz (470.4 kHz); 0x0E = 4.096 MHz (3.7632 MHz) 0x03 = 96 kHz (88.2 kHz); 0x09 = 768 kHz (705.6 kHz); 0x0F = 6.144 MHz (5.6448 MHz) 0x04 = 128 kHz (117.6 kHz); 0x0A = 1.024 MHz (940.8 kHz); 0x10 = 8.192 MHz (7.5264 MHz) 0x05 = 192 kHz (176.4 kHz); 0x0B = 1.536 MHz (1.4112 MHz); 0x11 = 12.288 MHz (11.2896 MHz) 0x06 = 256 kHz (235.2 kHz); 0x0C = 2.048 MHz (1.8816 MHz); 0x12 = 24.576 MHz (22.5792 MHz) The frequencies in brackets apply for 44.1 kHz–related sample rates only. If AIF2_RATE < 1000, AIF2 is referenced to the SYSCLK clock domain. In this case, the 44.1 kHz–related frequencies apply if SAMPLE_RATE_n = 01XXX. If AIF2_RATE ≥ 1000, AIF2 is referenced to the ASYNCCLK clock domain. In this case, the 44.1 kHz–related frequencies apply if ASYNC_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 4-50.

Table 4-50. 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; 0x07 = 384 kHz (352.8 kHz); 0x0D = 3.072 MHz (2.8824 MHz) 0x02 = 64 kHz (58.8 kHz); 0x08 = 512 kHz (470.4 kHz); 0x0E = 4.096 MHz (3.7632 MHz) 0x03 = 96 kHz (88.2 kHz); 0x09 = 768 kHz (705.6 kHz); 0x0F = 6.144 MHz (5.6448 MHz) 0x04 = 128 kHz (117.6 kHz); 0x0A = 1.024 MHz (940.8 kHz); 0x10 = 8.192 MHz (7.5264 MHz) 0x05 = 192 kHz (176.4 kHz); 0x0B = 1.536 MHz (1.4112 MHz); 0x11 = 12.288 MHz (11.2896 MHz) 0x06 = 256 kHz (235.2 kHz); 0x0C = 2.048 MHz (1.8816 MHz); 0x12 = 24.576 MHz (22.5792 MHz) The frequencies in brackets apply for 44.1 kHz–related sample rates only. If AIF3_RATE < 1000, AIF3 is referenced to the SYSCLK clock domain. In this case, the 44.1 kHz–related frequencies apply if SAMPLE_RATE_n = 01XXX. If AIF3_RATE ≥ 1000, AIF3 is referenced to the ASYNCCLK clock domain. In this case, the 44.1 kHz–related frequencies apply if ASYNC_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–AIF3 are described in Table 4-51 through Table 4-53 respectively.

Note that left-justified and DSP-B modes are valid in Master Mode only (i.e., BCLK and LRCLK are outputs from the CS42L92).

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 are 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. 4-44 through Fig. 4-47.

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

The AIF1 data control fields are described in Table 4-51.

Table 4-51. AIF1 Digital Audio Data Control

Register Address	Bit	Label	Default	Description
R1284 (0x0504) AIF1_Format	2:0	AIF1_FMT[2:0]	000	AIF1 Audio Interface Format 000 = DSP Mode A 001 = DSP Mode B 010 = I^2S mode 011 = Left-Justified mode Other codes are reserved.
R1287 (0x0507) AIF1_Frame_Ctrl_1	13:8	AIF1TX_WL[5:0]	0x18	AIF1 TX Word Length (Number of valid data bits per slot) Integer (LSB = 1); Valid from 16 to 32
	7:0	AIF1TX_SLOT_ LEN[7:0]	0x18	AIF1 TX Slot Length (Number of BCLK cycles per slot) Integer (LSB = 1); Valid from 16 to 128
R1288 (0x0508) AIF1_Frame_Ctrl_2	13:8	AIF1RX_WL[5:0]	0x18	AIF1 RX Word Length (Number of valid data bits per slot) Integer (LSB = 1); Valid from 16 to 32
	7:0	AIF1RX_SLOT_ LEN[7:0]	0x18	AIF1 RX Slot Length (Number of BCLK cycles per slot) Integer (LSB = 1); Valid from 16 to 128
R1289 (0x0509) to R1296 (0x0510)	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	
	5:0	AIF1TX7_SLOT[5:0]	0x6	
	5:0	AIF1TX8_SLOT[5:0]	0x7	
R1297 (0x0511) to R1304 (0x0518)	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	
	5:0	AIF1RX7_SLOT[5:0]	0x6	
	5:0	AIF1RX8_SLOT[5:0]	0x7	

The AIF2 data control fields are described in Table 4-52.

Table 4-52. 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 = I^2S 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

Table 4-52. AIF2 Digital Audio Data Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R1352 (0x0548) AIF2_Frame_Ctrl_2	13:8	AIF2RX_WL[5:0]	0x18	AIF2 RX Word Length (Number of valid data bits per slot) Integer (LSB = 1); Valid from 16 to 32
	7:0	AIF2RX_SLOT_ LEN[7:0]	0x18	AIF2 RX Slot Length (Number of BCLK cycles per slot) Integer (LSB = 1); Valid from 16 to 128
R1353 (0x0549) to R1360 (0x0550)	5:0	AIF2TX1_SLOT[5:0]	0x0	AIF2 TX Channel n Slot position Defines the TX time slot position of the Channel n audio sample Integer (LSB=1); Valid from 0 to 63
	5:0	AIF2TX2_SLOT[5:0]	0x1	
	5:0	AIF2TX3_SLOT[5:0]	0x2	
	5:0	AIF2TX4_SLOT[5:0]	0x3	
	5:0	AIF2TX5_SLOT[5:0]	0x4	
	5:0	AIF2TX6_SLOT[5:0]	0x5	
	5:0	AIF2TX7_SLOT[5:0]	0x6	
	5:0	AIF2TX8_SLOT[5:0]	0x7	
R1361 (0x0551) to R1368 (0x0558)	5:0	AIF2RX1_SLOT[5:0]	0x0	AIF2 RX Channel n Slot position Defines the RX time slot position of the Channel n audio sample Integer (LSB=1); Valid from 0 to 63
	5:0	AIF2RX2_SLOT[5:0]	0x1	
	5:0	AIF2RX3_SLOT[5:0]	0x2	
	5:0	AIF2RX4_SLOT[5:0]	0x3	
	5:0	AIF2RX5_SLOT[5:0]	0x4	
	5:0	AIF2RX6_SLOT[5:0]	0x5	
	5:0	AIF2RX7_SLOT[5:0]	0x6	
	5:0	AIF2RX8_SLOT[5:0]	0x7	

The AIF3 data control fields are described in Table 4-53.

Table 4-53. 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) to R1424 (0x0590)	5:0	AIF3TX1_SLOT[5:0]	0x0	AIF3 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	AIF3TX2_SLOT[5:0]	0x1	
	5:0	AIF3TX3_SLOT[5:0]	0x2	
	5:0	AIF3TX4_SLOT[5:0]	0x3	
	5:0	AIF3TX5_SLOT[5:0]	0x4	
	5:0	AIF3TX6_SLOT[5:0]	0x5	
	5:0	AIF3TX7_SLOT[5:0]	0x6	
	5:0	AIF3TX8_SLOT[5:0]	0x7	

Table 4-53. AIF3 Digital Audio Data Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R1425 (0x0591)	5:0	AIF3RX1_SLOT[5:0]	0x0	AIF3 RX Channel n Slot position
to	5:0	AIF3RX2_SLOT[5:0]	0x1	Defines the RX time slot position of the Channel n audio sample
R1432 (0x0598)	5:0	AIF3RX3_SLOT[5:0]	0x2	Integer (LSB=1); Valid from 0 to 63
	5:0	AIF3RX4_SLOT[5:0]	0x3	
	5:0	AIF3RX5_SLOT[5:0]	0x4	
	5:0	AIF3RX6_SLOT[5:0]	0x5	
	5:0	AIF3RX7_SLOT[5:0]	0x6	
	5:0	AIF3RX8_SLOT[5:0]	0x7	

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 CS42L92 is not transmitting data (i.e., during time slots that are not enabled for output by the CS42L92). If the AIF*n*TX_DAT_TRI bit is set, the CS42L92 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 4-54.

Table 4-54. 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 4-55.

Table 4-55. 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 4-56.

Table 4-56. 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) 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.

Multiline SLIMbus comprises one or more secondary data line, supporting additional bandwidth and flexibility for data transfer over the bus. All data lines are synchronized to the bus clock; the RF and CG parameters are common to all data lines. Note that control space is allocated on the primary data line only—secondary lines are used exclusively for data space. Accordingly, the SM parameter is defined for the primary line only.

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.

To transfer data between devices on the SLIMbus interface, a data channel connection is established between a source and one or more destination (sink) ports. A unique port number (PN) address is defined for every active TX/RX port. Multiple data channels can share the same port address by assigning different end point (EP) values to each channel.

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, SL, and data line (LN) parameters. For multiline operation, the LN value identifies the data line on which the channel is present. Note that the mapping of secondary data lines (1–7) with respect to the secondary data pins is configurable on each multiline SLIMbus component, using value elements associated with the respective interface device. See Section 4.9.3 for details of value elements.

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 CS42L92.

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 CS42L92 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 CS42L92 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 CS42L92 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, EP, CN) and CONNECT_SINK (PN, EP, CN) messages are issued to the respective devices, defining the ports between which a data channel is to be established. The end point parameter allows up to eight channels to share the same port number. Multiple destinations (sinks) can be configured for a channel, if required. The payload of each message contains the port number (PN), end point (EP), 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, LN) message informs the associated devices of the structure of the data channel. The payload of this message contains the channel number (CN), TP, SD, SL, and LN 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 CS42L92 features a MIPI-compliant SLIMbus interface. It supports multichannel audio input/output and control register read/write access.

The SLIMbus interface on CS42L92 comprises a generic device, framer device, interface device, and 16 data ports, providing up to 8 input (RX) channels and up to 8 output (TX) channels. Multiline capability is supported, offering additional bandwidth and system-level flexibility.

The interface supports up to eight audio input channels and up to eight audio output channels. Mixed sample rates can be supported simultaneously. The audio signal paths associated with the SLIMbus interface are described in Section 4.3.

The SLIMbus interface also supports read/write access to the CS42L92 control registers via the value map of the interface device, as described in Section 4.10.5.

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 CS42L92, and these bandwidth allocation requirements are outside the scope of this data sheet.

The SLIMbus interface provides full support for 32-bit data words (input and output). Audio data samples up to 32 bits can be routed to the AIF1, AIF3, SLIMbus, S/PDIF, and DAC output paths. Note that other signal paths and signal-processing blocks within the digital core are limited to 24-bit data length; data samples are truncated to 24-bit length if they are routed through any function that does not support 32-bit data words.

4.10.1 SLIMbus Device Parameters

The SLIMbus interface on the CS42L92 comprises three devices. The enumeration address of each device within the SLIMbus interface is derived from the parameters noted in Table 4-57.

Table 4-57. SLIMbus Device Parameters

Description	Manufacturer ID	Product Code	Device ID	Instance Value	Enumeration Address
Generic	0x01FA	0x6371	0x00	0x00	01FA_6371_0000
Framer	0x01FA	0x6371	0x55	0x00	01FA_6371_5500
Interface	0x01FA	0x6371	0x7F	0x00	01FA_6371_7F00

4.10.2 SLIMbus Message Support

The SLIMbus interface on the CS42L92 supports bus messages as described in Table 4-58 and Table 4-59.

Table 4-58. SLIMbus Message Support

Category	Message Code MC[6:0]	Description	Generic	Framer	Interface
Device Management Messages	0x01	REPORT_PRESENT (DC, DCV)	S	S	S
	0x02	ASSIGN_LOGICAL_ADDRESS (LA)	D	D	D
	0x04	RESET_DEVICE ()	D	D	D
	0x08	CHANGE_LOGICAL_ADDRESS (LA)	D	D	D
	0x09	CHANGE_ARBITRATION_PRIORITY (AP)	—	—	—
	0x0C	REQUEST_SELF_ANNOUNCEMENT ()	D	D	D
	0x0F	REPORT_ABSENT ()	—	—	—
Data Channel Management Messages	0x10	CONNECT_SOURCE (PN, EP, CN)	D	—	—
	0x11	CONNECT_SINK (PN, EP, CN)	D	—	—
	0x14	DISCONNECT_PORT (PN)	D	—	—
	0x18	CHANGE_CONTENT (CN, FL, PR, AF, DT, CL, DL)	D	—	—
Information Management Messages	0x20	REQUEST_INFORMATION (TID, EC)	D	D	D
	0x21	REQUEST_CLEAR_INFORMATION (TID, EC, CM)	D	D	D
	0x24	REPLY_INFORMATION (TID, IS)	S	S	S
	0x28	CLEAR_INFORMATION (EC, CM)	D	D	D
	0x29	REPORT_INFORMATION (EC, IS)	—	—	S

Table 4-58. SLIMbus Message Support *(Cont.)*

Category	Message Code MC[6:0]	Description	Generic	Framer	Interface
Reconfiguration Messages	0x40	BEGIN_RECONFIGURATION ()	D	D	D
	0x44	NEXT_ACTIVE_FRAMER (LAIF, NCo, NCi)	—	D	—
	0x45	NEXT_SUBFRAME_MODE (SM)	—	D	D
	0x46	NEXT_CLOCK_GEAR (CG)	—	D	—
	0x47	NEXT_ROOT_FREQUENCY (RF)	—	D	—
	0x4A	NEXT_PAUSE_CLOCK (RT)	—	D	—
	0x4B	NEXT_RESET_BUS ()	—	D	—
	0x4C	NEXT_SHUTDOWN_BUS ()	—	D	—
	0x50	NEXT_DEFINE_CHANNEL (CN, TP, SD, SL, LN)	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	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 CS42L92 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 CS42L92 interface device), or else using the NEXT_RESET_BUS message.

Additional notes on specific SLIMbus parameters are described in Table 4-59.

Table 4-59. SLIMbus Parameter Support

Parameter Code	Description	Comments
AF	Auxiliary Bits Format	—
CG	Clock Gear	—
CL	Channel Link	—
CM	Clear Mask	CS42L92 does not fully support this function. The CM bytes of the REQUEST_CLEAR_INFORMATION or CLEAR_INFORMATION messages must not be sent to CS42L92 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	CS42L92 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	—
EP	End Point	—
FL	Frequency Locked	—
IS	Information Slice	—
LA	Logical Address	—
LAIF	Logical Address, Incoming Framer	—
LN	Data Line	All LN codes (0–7) are supported. The LN value must be equal to one of the data lines that is mapped to one of the CS42L92 SLIMDAT*n* pins. The mapping of the secondary SLIMDAT*n* pins onto the SLIMbus data lines is configurable, using the value elements of the interface device. (Note that the primary data pin, SLIMDAT1, is always mapped to SLIMbus data line DATA0.)
NCi	Number of Incoming Framer Clock Cycles	—
NCo	Number of Outgoing Framer Clock Cycles	—

Table 4-59. SLIMbus Parameter Support *(Cont.)*

Parameter Code	Description	Comments
PN	Port Number	Note that the Port Numbers of the CS42L92 SLIMbus paths are register-configurable, as described in Table 4-62.
PR	Presence Rate	Note that the Presence Rate must be the same as the sample rate selected for the associated CS42L92 SLIMbus path.
RF	Root Frequency	CS42L92 supports the following RF codes as Active Framer: 0x1 = 24.576 MHz 0x2 = 22.5792 MHz All codes are supported when CS42L92 is not the Active Framer.
RT	Restart Time	CS42L92 supports the following RT codes: 0x0 = Fast Recovery 0x2 = Unspecified Delay When either of these values is specified, the CS42L92 resumes toggling the CLK line within four cycles of the CLK line frequency.
SD	Segment Distribution	Note that any audio channels that are assigned the same SAMPLE_RATE_*n* or ASYNC_SAMPLE_RATE_*n* value must also be assigned the same Segment Interval.
SL	Segment Length	Note that any active data channels that are assigned the same Port Number must also be assigned the same Segment Length.
SM	Subframe Mode	—
TID	Transaction ID	—
TP	Transport Protocol	CS42L92 supports the following TP codes, according to the applicable audio channel port: Audio TX channel ports: 0x0 (Isochronous Protocol) or 0x1 (Pushed Protocol) Audio RX channel ports: 0x0 (Isochronous Protocol) or 0x2 (Pulled Protocol) Note that any active data channels that are assigned the same Port Number must also be assigned the same Transport Protocol.
VS	Value Slice	—
VU	Value Update	—

4.10.3 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 CS42L92 SLIMbus interface includes a framer device. When configured as the active framer, the SLIMbus clock (SLIMCLK) is an output from the CS42L92. At other times, SLIMCLK is an input. 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 in active framer mode are 24.576 or 22.5792 MHz only. At other times, 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 (or output) 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 4-60.

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

Table 4-60. 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

Table 4-60. SLIMbus Clock Gear Selection *(Cont.)*

Clock Gear	Description	SLIMCLK Frequency [1]
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

If the CS42L92 is the active framer, the SLIMCLK output is synchronized to the SYSCLK or ASYNCCLK system clock, as selected by the SLIMCLK_SRC bit. The applicable system clock must be enabled and configured at the SLIMbus root frequency, whenever the CS42L92 is the active framer.

If the CS42L92 is not the active framer, the SLIMCLK input can be used to provide a reference source for the FLLs. The frequency of this reference is controlled using SLIMCLK_REF_GEAR, as described in Table 4-61.

The input clock reference for the FLLs is selected by using FLL*n*_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 FLLs.

Table 4-61. SLIMbus Clock Reference Control

Register Address	Bit	Label	Default	Description
R1507 (0x05E3) SLIMbus_Framer_ Ref_Gear	4	SLIMCLK_SRC	0	SLIMbus Clock source Selects the SLIMbus reference clock in Active Framer mode. 0 = SYSCLK 1 = ASYNCCLK Note that the applicable clock must be enabled, and configured at the SLIMbus Root Frequency, in Active Framer mode.
	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

4.10.4 SLIMbus Audio Channel Control

4.10.4.1 Port Number Control

The CS42L92 SLIMbus interface supports up to eight audio input (RX) channels and up to eight audio output (TX) channels. The port number and end point number for each channel is configurable using the fields described in Table 4-62.

Table 4-62. SLIMbus Audio Port Number Control

Register Address	Bit	Label	Default	Description
R1490 (0x05D2)	15:8	SLIMRX2_PORT_ADDR[7:0]	0x01	SLIMbus RX Channel n Port number Bits [7:5] specify the End Point (valid from 0–7). Bits [4:0] specify the Port Number (valid from 0–31)
SLIMbus_RX_Ports0	7:0	SLIMRX1_PORT_ADDR[7:0]	0x00	
R1491 (0x05D3)	15:8	SLIMRX4_PORT_ADDR[7:0]	0x03	
SLIMbus_RX_Ports1	7:0	SLIMRX3_PORT_ADDR[7:0]	0x02	
R1492 (0x05D4)	15:8	SLIMRX6_PORT_ADDR[7:0]	0x05	
SLIMbus_RX_Ports2	7:0	SLIMRX5_PORT_ADDR[7:0]	0x04	
R1493 (0x05D5)	15:8	SLIMRX8_PORT_ADDR[7:0]	0x07	
SLIMbus_RX_Ports3	7:0	SLIMRX7_PORT_ADDR[7:0]	0x06	

Table 4-62. SLIMbus Audio Port Number Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R1494 (0x05D6) SLIMbus_TX_Ports0	15:8	SLIMTX2_PORT_ADDR[7:0]	0x09	SLIMbus TX Channel n Port number Valid from 0–31 Bits [7:5] specify the End Point (valid from 0–7). Bits [4:0] specify the Port Number (valid from 0–31)
	7:0	SLIMTX1_PORT_ADDR[7:0]	0x08	
R1495 (0x05D7) SLIMbus_TX_Ports1	15:8	SLIMTX4_PORT_ADDR[7:0]	0x0B	
	7:0	SLIMTX3_PORT_ADDR[7:0]	0x0A	
R1496 (0x05D8) SLIMbus_TX_Ports2	15:8	SLIMTX6_PORT_ADDR[7:0]	0x0D	
	7:0	SLIMTX5_PORT_ADDR[7:0]	0x0C	
R1497 (0x05D9) SLIMbus_TX_Ports3	15:8	SLIMTX8_PORT_ADDR[7:0]	0x0F	
	7:0	SLIMTX7_PORT_ADDR[7:0]	0x0E	

4.10.4.2 Sample-Rate Control

The SLIMbus audio inputs may be selected as input to the digital mixers or signal-processing functions within the CS42L92 digital core. The SLIMbus audio 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 4-26.

Note that the SLIMbus interface provides simultaneous support for SYSCLK-referenced and ASYNCCLK-referenced sample rates on different channels. For example, 48-kHz and 44.1-kHz SLIMbus audio paths can be simultaneously supported.

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

4.10.4.3 Signal Path Enable

The SLIMbus interface supports up to eight audio input (RX) channels and up to eight audio output (TX) channels. Each channel can be enabled or disabled using the fields defined in Table 4-63.

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 SLIMbus port.

The system clock, SYSCLK, must be configured and enabled before any audio path is enabled. The ASYNCCLK may also be required, depending on the path configuration. See Section 4.16 for details of the system clocks.

Table 4-63. SLIMbus Signal Path Enable

Register Address	Bit	Label	Default	Description
R1525 (0x05F5) SLIMbus_RX_ Channel_Enable	7	SLIMRX8_ENA	0	SLIMbus RX Channel n Enable 0 = Disabled 1 = Enabled
	6	SLIMRX7_ENA	0	
	5	SLIMRX6_ENA	0	
	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	7	SLIMTX8_ENA	0	SLIMbus TX Channel n Enable 0 = Disabled 1 = Enabled
	6	SLIMTX7_ENA	0	
	5	SLIMTX6_ENA	0	
	4	SLIMTX5_ENA	0	
	3	SLIMTX4_ENA	0	
	2	SLIMTX3_ENA	0	
	1	SLIMTX2_ENA	0	
	0	SLIMTX1_ENA	0	

Table 4-63. SLIMbus Signal Path Enable *(Cont.)*

Register Address	Bit	Label	Default	Description
R1527 (0x05F7) SLIMbus_RX_ Port_Status	7	SLIMRX8_PORT_STS	0	SLIMbus RX Channel n Port Status (Read only) 0 = Disabled 1 = Configured and active
	6	SLIMRX7_PORT_STS	0	
	5	SLIMRX6_PORT_STS	0	
	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	7	SLIMTX8_PORT_STS	0	SLIMbus TX Channel n Port Status (Read only) 0 = Disabled 1 = Configured and active
	6	SLIMTX7_PORT_STS	0	
	5	SLIMTX6_PORT_STS	0	
	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	

4.10.5 SLIMbus Control Register Access

The SLIMbus interface supports read/write access to the CS42L92 control registers via the value map of the interface device. Full read/write access to all registers is possible, via the user value elements portion of the value map.

Note that if multibyte transfers of more than 8 bytes are scheduled (see Section 4.10.5.3), system clocking constraints must be observed to ensure control interface limits are not exceeded. Full details of the applicable clocking requirements are provided in Section 4.16.7.

4.10.5.1 Control Register Write

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 4-64 and Table 4-65, 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 in Section 4.10.5.3.

Table 4-64. 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 CS42L92 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 4-65. 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 CS42L92 SLIMbus interface device). The value is assigned by the SLIMbus manager device.
Access Mode	0b1	Selects byte-based access mode.

Table 4-65. Register Write Message (2)—CHANGE_VALUE *(Cont.)*

Parameter	Value	Description
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 CS42L92.

4.10.5.2 Control Register Read

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 4-66 and Table 4-67, for the generic case of reading the contents of control register address 0xYYYYZZ. Note that it is also possible to read blocks of up to 8 bytes (to consecutive register addresses), as described in Section 4.10.5.3.

Table 4-66. 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 CS42L92 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 4-67. 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 CS42L92 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 CS42L92.

The CS42L92 responds to the register read commands in accordance with the normal SLIMbus protocols.

Note that the CS42L92 assumes that sufficient control space slots are available in which to provide its response before the next REQUEST_VALUE message is received. The CS42L92 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 CS42L92 response has been made, the earlier REQUEST_VALUE messages are ignored (i.e., only the last REQUEST_VALUE message is serviced).

4.10.5.3 Multibyte Control Register Access

Register data transfers of up to 16 bytes can be configured using the slice size parameter in the second message of the applicable protocol (see Table 4-65 or Table 4-67). Additional value update words are appended to respective data message in this case, with the applicable data contents.

Multibyte register read/write access is supported with slice size of 2, 4, 8, 12, or 16 bytes. Note that if multibyte transfers of more than 8 bytes are scheduled, system clocking constraints must be observed to ensure control interface limits are not exceeded. See Section 4.16.7.

When a 2-byte transfer is selected, the register address 0xYYYYZZ is used (using the same naming conventions as above). When more than 2 bytes are transferred, the register address is automatically incremented as described in Table 4-68.

Table 4-68. SLIMbus Register Read/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 accessing 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 4-69.

Table 4-69. SLIMbus Register Read/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

4.11 Output Signal Path

The CS42L92 provides four stereo pairs of audio output signal paths. These outputs comprise ground-referenced headphone drivers, and a digital output interface suitable for external speaker drivers. The output signal paths are summarized in Table 4-70.

Table 4-70. Output Signal Path Summary

Signal Path	Descriptions	Output Pins
OUT1L, OUT1R	Ground-referenced headphone/earpiece output	HPOUT1L, HPOUT1R
OUT2L, OUT2R	Ground-referenced headphone/earpiece output	HPOUT2L, HPOUT2R
OUT3L, OUT3R	Ground-referenced headphone/earpiece output	HPOUT3L, HPOUT3R or HPOUT4L, HPOUT4R
OUT5L, OUT5R	Digital speaker (PDM) output	SPKDAT, SPKCLK

The analog output paths incorporate high performance 32-bit sigma-delta DACs.

Under default conditions, the headphone drivers provide a stereo, single-ended output. A mono mode is also supported, providing a differential (BTL) output configuration. The ground-referenced headphone output paths incorporate a common mode feedback path for rejection of system-related noise. These outputs support direct connection to headphone loads, with no requirement for AC coupling capacitors.

The OUT1 and OUT2 paths can be configured in High Performance Mode, offering exceptional performance on the respective headphone outputs. Balanced stereo headphone loads can be supported by assigning the OUT1 and OUT2 signal paths for the left and right channels respectively, with the associated drivers configured for differential output.

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 filters can be enabled in the output signal paths, supporting audiophile-quality DAC playback options. These hi-fi filters allow user selection of the preferred characteristics, (e.g., linear phase, antialiasing, or apodizing filter responses).

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 output signal path. Any two of the output signal paths may be selected as input to the AEC loop-back paths.

The CS42L92 provides short-circuit detection on the headphone output paths. See Section 4.21 for further details.

The CS42L92 output signal paths are shown in Fig. 4-52. Note that the OUT4 path is not implemented on this device.

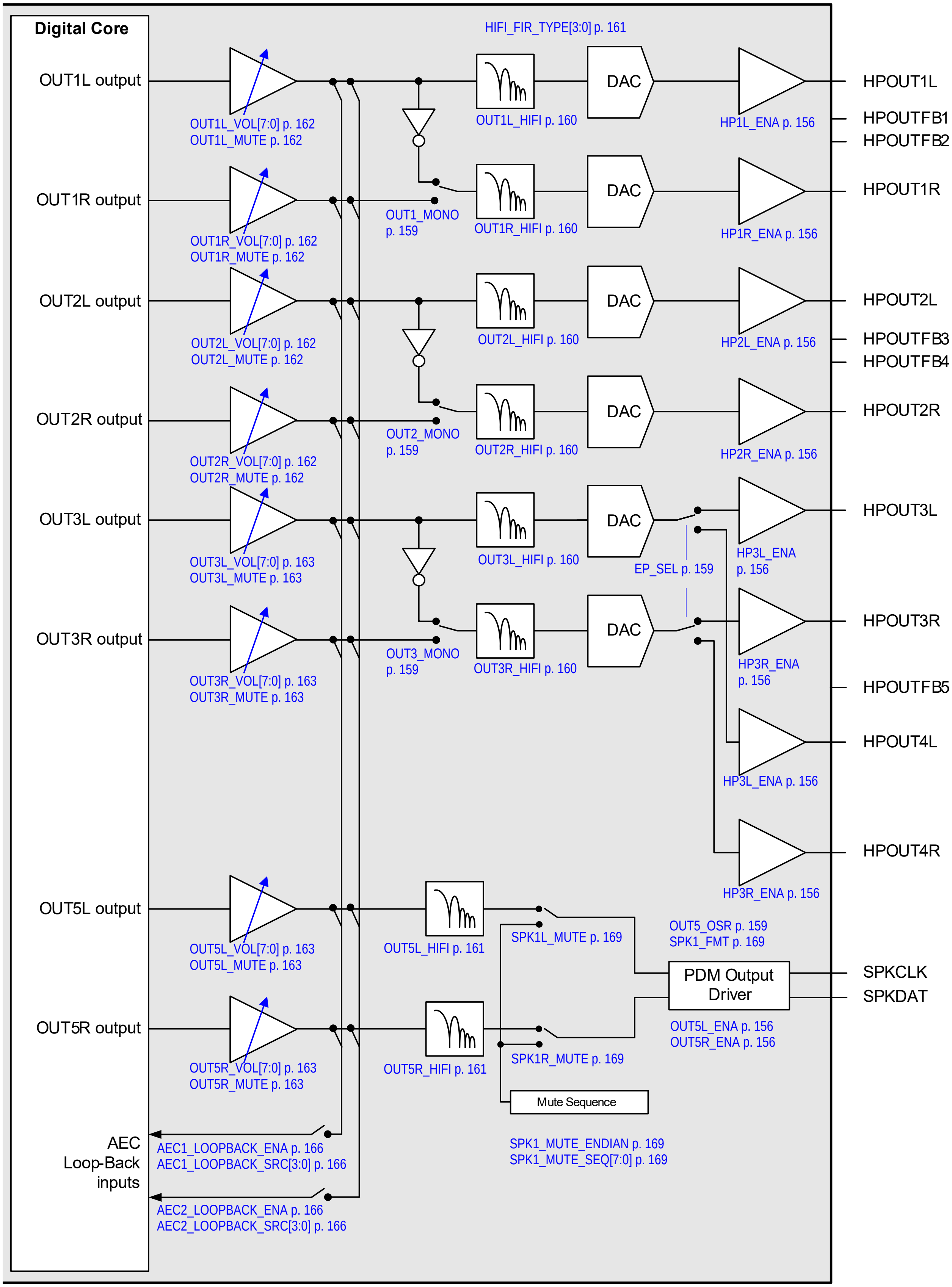


Figure 4-52. Output Signal Paths

4.11.1 Output Signal Path Enable

The output signal paths are enabled using the bits described in Table 4-71. The respective bits must be enabled for analog or digital output on the respective output paths.

The OUT3 path is associated with the HPOUT3 and HPOUT4 output drivers. The HP3L_ENA and HP3R_ENA bits control either the HPOUT3 or HPOUT4 drivers, depending on the EP_SEL register bit selection. See Table 4-74 for details of the EP_SEL register bit.

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 4-71.

The supply rails for the analog outputs (HPOUT1–HPOUT4) are generated using an integrated dual-mode charge pump, CP1. The charge pump is enabled automatically by the CS42L92 when required by the output drivers; see Section 4.19.

The CS42L92 schedules a pop-suppressed control sequence to enable or disable the HPOUT1–HPOUT4 signal paths. This is automatically managed by the control-write sequencer in response to setting the respective HP*nx*_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 ASYNCCLK may also be required, depending on the path configuration. See Section 4.16 for details of the system clocks.

The CS42L92 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 output signal path. If an underclocked error condition occurs, these bits indicate which signal paths have been enabled.

Table 4-71. 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
	5	HP3L_ENA	0	Output Path 3 (left) enable If EP_SEL = 0, this bit controls the HPOUT3L output driver. If EP_SEL = 1, this bit controls the HPOUT4L output driver. 0 = Disabled 1 = Enabled
	4	HP3R_ENA	0	Output Path 3 (right) enable If EP_SEL = 0, this bit controls the HPOUT3R output driver. If EP_SEL = 1, this bit controls the HPOUT4R output driver. 0 = Disabled 1 = Enabled
	3	HP2L_ENA	0	Output Path 2 (left) enable 0 = Disabled 1 = Enabled
	2	HP2R_ENA	0	Output Path 2 (right) enable 0 = Disabled 1 = Enabled
	1	HP1L_ENA	0	Output Path 1 (left) enable 0 = Disabled 1 = Enabled
	0	HP1R_ENA	0	Output Path 1 (right) enable 0 = Disabled 1 = Enabled
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
R1030 (0x0406) Raw_Output_Status_1	5	OUT3L_ENA_STS	0	Output Path 3 (left) enable status 0 = Disabled 1 = Enabled
	4	OUT3R_ENA_STS	0	Output Path 3 (right) enable status 0 = Disabled 1 = Enabled
	3	OUT2L_ENA_STS	0	Output Path 2 (left) enable status 0 = Disabled 1 = Enabled
	2	OUT2R_ENA_STS	0	Output Path 2 (right) enable status 0 = Disabled 1 = Enabled
	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 CS42L92 digital core. The sample rate for the output signal paths is configured using OUT_RATE—see Table 4-26.

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

A clocking signal is required by the DACs and output signal path circuits. This clock can be provided internally from the SYSCLK or ASYNCCLK system clocks, or can be provided externally via the MCLK*n* pins. The clock source is selected using OUT_CLK_SRC.

The output signal path clock must be synchronized with whichever clock domain is associated with the OUT_RATE setting.

- If OUT_RATE is configured for one of the SYSCLK-related sample rates (SAMPLE_RATE_*n*), then ASYNCCLK is not a valid selection for OUT_CLK_SRC.
- If OUT_RATE is configured for one of the ASYNCCLK-related sample rates (ASYNC_SAMPLE_RATE_*n*), then SYSCLK is not a valid selection for OUT_CLK_SRC.
- If MCLK*n* is selected as the clock source (OUT_CLK_SRC = 100–110), the respective MCLK frequency must be an integer multiple of the sample rate selected by OUT_RATE. The required clock frequency is either 6.144 MHz (for 48-kHz-related sample rates) or 5.6448 MHz (44.1-kHz-related sample rates). A configurable divider is provided in order to provide flexibility in matching the MCLK input with the required frequency.

If a suitable external MCLK source is available, then the choice exists between selecting the internal SYSCLK/ASYNCCLK or the external MCLK as the output path clock. If the external MCLK source is high quality, then MCLK is the recommended source. If the MCLK source is not high quality, then the SYSCLK/ASYNCCLK source (generated by one of the FLLs) may provide a better clock.

The clocking configuration for the output signal path is shown in Fig. 4-53.

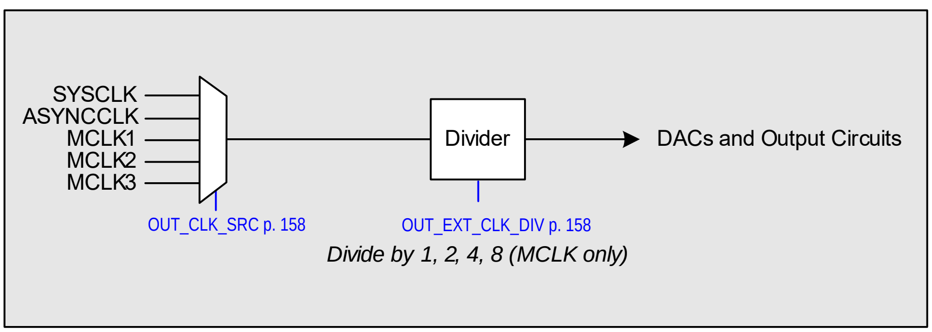


Figure 4-53. Output Signal Path Clocking Control

The output signal path clocking control registers are defined in Table 4-72. The OUT_RATE field is defined in Table 4-26.

Table 4-72. Output Signal Path Clocking Control

Register Address	Bit	Label	Default	Description
R1032 (0x0408) Output_Rate_1	5:4	OUT_EXT_CLK_DIV	00	Output Signal Path Clock Divider (only valid if OUT_CLK_SRC selects MCLK*n* as the clock source) 00 = Divide by 1 01 = Divide by 2 10 = Divide by 4 11 = Divide by 8
	2:0	OUT_CLK_SRC	000	Output Signal Path Clock Source 000 = SYSCLK 001 = ASYNCCLK 100 = MCLK1 101 = MCLK2 110 = MCLK3 All other codes are reserved Note that the selected clock must be synchronized with whichever clock domain is associated with the OUT_RATE setting.

4.11.3 Output Signal Path Control

The OUT1 and OUT2 paths support selectable configuration for low-power or high-performance operation. The applicable mode is selected using the CP_DAC_MODE bit, as described in Table 4-74. The High Performance Mode is configured by default.

Under default register conditions, the headphone output (HPOUT1–HPOUT4) paths are configured for stereo output. The headphone paths can be configured for mono differential (BTL) output using the OUT*n*_MONO bits; this is ideal for driving an earpiece or hearing aid coil.

If the OUT*n*_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 bits described in Table 4-71.

The mono (BTL) signal paths are shown in Fig. 4-52. Note that, in Mono Mode, the effective gain of the signal path is increased by 6 dB.

Note that the EP_SEL and OUT3_MONO bits should not be changed if the HPOUT3 or HPOUT4 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 HPOUT3 and HPOUT4 drivers are enabled using the HP3L_ENA and HP3R_ENA bits, as described in Table 4-71.

The SPKCLK frequency of the PDM output path (OUT5) is controlled by OUT5_OSR, as described in Table 4-73. 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 4-73 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 4-73. 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 4-74.

Table 4-74. Output Signal Path Control

Register Address	Bit	Label	Default	Description
R1024 (0x0400) Output_Enables_1	15	EP_SEL	0	Output Path 3 Output Driver select 0 = HPOUT3L and HPOUT3R 1 = HPOUT4L and HPOUT4R
R1032 (0x408) Output_Rate_1	6	CP_DAC_MODE	1	Output Path 1 and 2 Performance Mode select 0 = Low Power Mode 1 = High Performance Mode
R1040 (0x0410) Output_Path_ Config_1L	12	OUT1_MONO	0	Output Path 1 Mono Mode (Configures HPOUT1L and HPOUT1R as a mono differential output.) 0 = Disabled 1 = Enabled The gain of the signal path is increased by 6 dB in differential (mono) mode.
R1048 (0x0418) Output_Path_ Config_2L	12	OUT2_MONO	0	Output Path 2 Mono Mode (Configures HPOUT2L and HPOUT2R as a mono differential output.) 0 = Disabled 1 = Enabled The gain of the signal path is increased by 6 dB in differential (mono) mode.
R1056 (0x0420) Output_Path_ Config_3L	12	OUT3_MONO	0	Output Path 3 Mono Mode (Configures HPOUT3 and HPOUT4 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 Filter Control

An integrated signal-processing engine on the CS42L92 supports digital filter requirements for range of hi-fi applications. Preset filter coefficients are held in on-board ROM, and can be configured and enabled as required. The hi-fi filters are tailored to specific sample rates, and provide options relating to passband frequency, stopband attenuation, and phase-response characteristics.

The filter type is selected using the HIFI_FIR_TYPE field. The available filters are each described in Table 4-75. The digital filter can be enabled in any output path, using the respective OUT*nx*_HIFI bits.

Note that only one filter type can be selected at any time, but the applicable filter can be enabled on any number of output paths simultaneously. The supported sample rates for each filter type are noted in Table 4-75—the selected filter must be consistent with the output sample rate (OUT_RATE) setting.

The digital filter can be enabled or disabled independently in any output path. A short interruption to the playback if the filter type is changed while the hi-fi filters are enabled on any output path.

The hi-fi digital filters are described in Table 4-75.

Table 4-75. Output Signal Path Digital Filter Types

Type	Sample Rate	Description	Passband	Stopband	Passband Ripple	Stopband Attenuation
0	48 kHz, 96 kHz, 192 kHz, 44.1 kHz, 88.2 kHz, 176.4 kHz	Deep stopband, linear phase	0.454 fs	0.546 fs	0.001 dB	120 dB
1		Deep stopband, minimum phase	0.454 fs	0.546 fs	0.001 dB	120 dB
2	48 kHz, 44.1 kHz	Antialias, linear phase	0.417 fs	0.5 fs	0.001 dB	110 dB
3		Antialias, minimum phase	0.417 fs	0.5 fs	0.001 dB	110 dB
4	96 kHz, 192 kHz, 384 kHz, 88.2 kHz, 176.4 kHz, 352.8 kHz	Non-apodizing, linear phase	0.227 fs	0.5 fs	0.001 dB	120 dB
5		Non-apodizing, minimum phase	0.227 fs	0.5 fs	0.001 dB	120 dB
6		Apodizing, linear phase	0.227 fs	0.454 fs	0.001 dB	120 dB
7		Apodizing, minimum phase	0.227 fs	0.454 fs	0.001 dB	120 dB

Table 4-75. Output Signal Path Digital Filter Types *(Cont.)*

Type	Sample Rate	Description	Passband	Stopband	Passband Ripple	Stopband Attenuation
8	192 kHz, 384 kHz, 176.4 kHz, 352.8 kHz	Non-apodizing, linear phase	0.114 fs	0.5 fs	0.001 dB	125 dB
9		Non-apodizing, minimum phase	0.114 fs	0.5 fs	0.001 dB	125 dB
10		Apodizing, linear phase	0.114 fs	0.454 fs	0.001 dB	125 dB
11		Apodizing, minimum phase	0.114 fs	0.454 fs	0.001 dB	125 dB
12	192 kHz, 384 kHz, 176.4 kHz, 352.8 kHz	Non-apodizing, non-interpolating linear phase	0.114 fs	0.499 fs	0.001 dB	125 dB
13		Non-apodizing, non-interpolating minimum phase	0.114 fs	0.499 fs	0.001 dB	125 dB
14		Apodizing, non-interpolating linear phase	0.114 fs	0.499 fs	0.001 dB	125 dB
15		Apodizing, non-interpolating minimum phase	0.114 fs	0.499 fs	0.001 dB	125 dB

At 48 kHz (or 44.1 kHz) sample rate, the key parameters of the hi-fi digital filters are the stopband attenuation and phase response. The stopband characteristics are noted in Table 4-75.

The choice between linear phase and minimum phase filters determines time-domain effects of the filter transfer function. Linear phase offers zero group delay, and equal levels of pre- and postringing. Minimum phase offers minimum preringing and latency, but higher levels of postringing and group delay distortion.

For sample rates above 48kHz, a choice between apodizing and non-apodizing filters is available. Apodizing filters have the capability to reduce time-domain distortion—this can be used to eliminate time smear effects introduced by other filters in the signal chain, provided the other filters have a flat frequency response throughout the apodizing filter's cut-off region. The cut-off (stopband) frequency of the apodizing filters are slightly lower than the respective non-apodizing filter response.

The hi-fi digital filter control registers are described in Table 4-76.

Table 4-76. Output Signal Path Digital Filter Control

Register Address	Bit	Label	Default	Description
R1040 (0x0410) Output_Path_Config_1L	14	OUT1L_HIFI	0	Output Path 1 (Left) Hi-Fi Filter Enable 0 = Disabled 1 = Enabled
R1044 (0x0414) Output_Path_Config_1R	14	OUT1R_HIFI	0	Output Path 1 (Right) Hi-Fi Filter Enable 0 = Disabled 1 = Enabled
R1048 (0x0418) Output_Path_Config_2L	14	OUT2L_HIFI	0	Output Path 2 (Left) Hi-Fi Filter Enable 0 = Disabled 1 = Enabled
R1052 (0x041C) Output_Path_Config_2R	14	OUT2R_HIFI	0	Output Path 2 (Right) Hi-Fi Filter Enable 0 = Disabled 1 = Enabled
R1056 (0x0420) Output_Path_Config_3L	14	OUT3L_HIFI	0	Output Path 3 (Left) Hi-Fi Filter Enable 0 = Disabled 1 = Enabled
R1060 (0x0424) Output_Path_Config_3R	14	OUT3R_HIFI	0	Output Path 3 (Right) Hi-Fi Filter Enable 0 = Disabled 1 = Enabled

Table 4-76. Output Signal Path Digital Filter Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R1072 (0x0430) Output_Path_ Config_5L	14	OUT5L_HIFI	0	Output Path 5 (Left) Hi-Fi Filter Enable 0 = Disabled 1 = Enabled
R1076 (0x0434) Output_Path_ Config_5R	14	OUT5R_HIFI	0	Output Path 5 (Right) Hi-Fi Filter Enable 0 = Disabled 1 = Enabled
R1102 (0x044E) Filter_Control	3:0	HIFI_FIR_ TYPE[3:0]	0x0	Output Path Hi-Fi Filter Select 0x0 = 192 kHz deep stopband, linear phase 0x1 = 192 kHz deep stopband, minimum phase 0x2 = 48 kHz antialias, linear phase 0x3 = 48 kHz antialias, minimum phase 0x4 = 384 kHz non-apodizing, linear phase 0x5 = 384 kHz non-apodizing, minimum phase 0x6 = 384 kHz apodizing, linear phase 0x7 = 384 kHz apodizing, minimum phase 0x8 = 384 kHz non-apodizing, linear phase 0x9 = 384 kHz non-apodizing, minimum phase 0xA = 384 kHz apodizing, linear phase 0xB = 384 kHz apodizing, minimum phase 0xC = 384 kHz non-apodizing, non-interpolating linear phase 0xD = 384 kHz non-apodizing, non-interpolating minimum phase 0xE = 384 kHz apodizing, non-interpolating linear phase 0xF = 384 kHz apodizing, non-interpolating minimum phase

4.11.5 Output Signal Path Digital Volume Control

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

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

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

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

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

Note: The 0 dBFS level of the OUT5 digital output path is not equal to the 0 dBFS level of the CS42L92 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 4-77 and Table 4-78.

Table 4-77. 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 4-78 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 4-78 for volume register definition). –64 dB to +31.5 dB in 0.5-dB steps 0x00 = –64 dB 0x80 = 0 dB 0xC0 to 0xFF = Reserved 0x01 = –63.5 dB … (0.5-dB steps) … (0.5-dB steps) 0xBF = +31.5 dB
R1049 (0x0419) DAC_Digital_Volume_2L	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	OUT2L_MUTE	1	Output Path 2 (Left) Digital Mute 0 = Unmute 1 = Mute
	7:0	OUT2L_VOL[7:0]	0x80	Output Path 2 (Left) Digital Volume (see Table 4-78 for volume register definition). –64 dB to +31.5 dB in 0.5-dB steps 0x00 = –64 dB 0x80 = 0 dB 0xC0 to 0xFF = Reserved 0x01 = –63.5 dB … (0.5-dB steps) … (0.5-dB steps) 0xBF = +31.5 dB
R1053 (0x041D) DAC_Digital_Volume_2R	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	OUT2R_MUTE	1	Output Path 2 (Right) Digital Mute 0 = Unmute 1 = Mute
	7:0	OUT2R_VOL[7:0]	0x80	Output Path 2 (Right) Digital Volume (see Table 4-78 for volume register definition). –64 dB to +31.5 dB in 0.5-dB steps 0x00 = –64 dB 0x80 = 0 dB 0xC0 to 0xFF = Reserved 0x01 = –63.5 dB … (0.5-dB steps) … (0.5-dB steps) 0xBF = +31.5 dB

Table 4-77. Output Signal Path Digital Volume Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R1057 (0x0421) DAC_Digital_ Volume_3L	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	OUT3L_MUTE	1	Output Path 3 (Left) Digital Mute 0 = Unmute 1 = Mute
	7:0	OUT3L_VOL[7:0]	0x80	Output Path 3 (Left) Digital Volume (see Table 4-78 for volume register definition). –64 dB to +31.5 dB in 0.5-dB steps 0x00 = –64 dB 0x80 = 0 dB 0xC0 to 0xFF = Reserved 0x01 = –63.5 dB … (0.5-dB steps) … (0.5-dB steps) 0xBF = +31.5 dB
R1061 (0x0425) DAC_Digital_ Volume_3R	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	OUT3R_MUTE	1	Output Path 3 (Right) Digital Mute 0 = Unmute 1 = Mute
	7:0	OUT3R_VOL[7:0]	0x80	Output Path 3 (Right) Digital Volume (see Table 4-78 for volume register definition). –64 dB to +31.5 dB in 0.5-dB steps 0x00 = –64 dB 0x80 = 0 dB 0xC0 to 0xFF = Reserved 0x01 = –63.5 dB … (0.5-dB steps) … (0.5-dB steps) 0xBF = +31.5 dB
R1073 (0x0431) DAC_Digital_ Volume_5L	9	OUT_VU	See Footnote 1	Output Signal Paths Volume Update. Writing 1 to this bit causes the Output Signal Paths Volume and Mute settings to be updated simultaneously
	8	OUT5L_MUTE	1	Output Path 5 (Left) Digital Mute 0 = Unmute 1 = Mute
	7:0	OUT5L_VOL[7:0]	0x80	Output Path 5 (Left) Digital Volume (see Table 4-78 for volume register definition). –64 dB to +31.5 dB in 0.5-dB steps 0x00 = –64 dB 0x80 = 0 dB 0xC0 to 0xFF = Reserved 0x01 = –63.5 dB … (0.5-dB steps) … (0.5-dB steps) 0xBF = +31.5 dB
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 4-78 for volume register definition). –64 dB to +31.5 dB in 0.5-dB steps 0x00 = –64 dB 0x80 = 0 dB 0xC0 to 0xFF = Reserved 0x01 = –63.5 dB … (0.5-dB steps) … (0.5-dB steps) 0xBF = +31.5 dB

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

Table 4-78 lists the output signal path digital volume settings.

Table 4-78. 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

Table 4-78. 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)
0x08	–60.0	0x39	–35.5	0x6A	–11.0	0x9B	13.5
0x09	–59.5	0x3A	–35.0	0x6B	–10.5	0x9C	14.0
0x0A	–59.0	0x3B	–34.5	0x6C	–10.0	0x9D	14.5
0x0B	–58.5	0x3C	–34.0	0x6D	–9.5	0x9E	15.0
0x0C	–58.0	0x3D	–33.5	0x6E	–9.0	0x9F	15.5
0x0D	–57.5	0x3E	–33.0	0x6F	–8.5	0xA0	16.0
0x0E	–57.0	0x3F	–32.5	0x70	–8.0	0xA1	16.5
0x0F	–56.5	0x40	–32.0	0x71	–7.5	0xA2	17.0
0x10	–56.0	0x41	–31.5	0x72	–7.0	0xA3	17.5
0x11	–55.5	0x42	–31.0	0x73	–6.5	0xA4	18.0
0x12	–55.0	0x43	–30.5	0x74	–6.0	0xA5	18.5
0x13	–54.5	0x44	–30.0	0x75	–5.5	0xA6	19.0
0x14	–54.0	0x45	–29.5	0x76	–5.0	0xA7	19.5
0x15	–53.5	0x46	–29.0	0x77	–4.5	0xA8	20.0
0x16	–53.0	0x47	–28.5	0x78	–4.0	0xA9	20.5
0x17	–52.5	0x48	–28.0	0x79	–3.5	0xAA	21.0
0x18	–52.0	0x49	–27.5	0x7A	–3.0	0xAB	21.5
0x19	–51.5	0x4A	–27.0	0x7B	–2.5	0xAC	22.0
0x1A	–51.0	0x4B	–26.5	0x7C	–2.0	0xAD	22.5
0x1B	–50.5	0x4C	–26.0	0x7D	–1.5	0xAE	23.0
0x1C	–50.0	0x4D	–25.5	0x7E	–1.0	0xAF	23.5
0x1D	–49.5	0x4E	–25.0	0x7F	–0.5	0xB0	24.0
0x1E	–49.0	0x4F	–24.5	0x80	0.0	0xB1	24.5
0x1F	–48.5	0x50	–24.0	0x81	0.5	0xB2	25.0
0x20	–48.0	0x51	–23.5	0x82	1.0	0xB3	25.5
0x21	–47.5	0x52	–23.0	0x83	1.5	0xB4	26.0
0x22	–47.0	0x53	–22.5	0x84	2.0	0xB5	26.5
0x23	–46.5	0x54	–22.0	0x85	2.5	0xB6	27.0
0x24	–46.0	0x55	–21.5	0x86	3.0	0xB7	27.5
0x25	–45.5	0x56	–21.0	0x87	3.5	0xB8	28.0
0x26	–45.0	0x57	–20.5	0x88	4.0	0xB9	28.5
0x27	–44.5	0x58	–20.0	0x89	4.5	0xBA	29.0
0x28	–44.0	0x59	–19.5	0x8A	5.0	0xBB	29.5
0x29	–43.5	0x5A	–19.0	0x8B	5.5	0xBC	30.0
0x2A	–43.0	0x5B	–18.5	0x8C	6.0	0xBD	30.5
0x2B	–42.5	0x5C	–18.0	0x8D	6.5	0xBE	31.0
0x2C	–42.0	0x5D	–17.5	0x8E	7.0	0xBF	31.5
0x2D	–41.5	0x5E	–17.0	0x8F	7.5	0xC0–0xFF	Reserved
0x2E	–41.0	0x5F	–16.5	0x90	8.0		
0x2F	–40.5	0x60	–16.0	0x91	8.5		
0x30	–40.0	0x61	–15.5	0x92	9.0		

4.11.6 Output Signal Path Noise-Gate Control

The CS42L92 provides a digital noise-gate function for each output signal path. 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 4-79.

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 output-path noise gate 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 CS42L92 gradually attenuates the respective signal path at the rate set by OUT_VD_RAMP (see Table 4-77). When the noise gate is deactivated, the output volume increases at the rate set by OUT_VI_RAMP.

Table 4-79. 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 [10] = Reserved [9] = OUT5R [8] = OUT5L [7] = Reserved [6] = Reserved [5] = OUT3R [4] = OUT3L [3] = OUT2R [2] = OUT2L [1] = OUT1R [0] = OUT1L
R1047 (0x0417) Noise_Gate_Select_1R	11:0	OUT1R_NGATE_SRC[11:0]	0x002	
R1051 (0x041B) Noise_Gate_Select_2L	11:0	OUT2L_NGATE_SRC[11:0]	0x004	
R1055 (0x041F) Noise_Gate_Select_2R	11:0	OUT2R_NGATE_SRC[11:0]	0x008	
R1059 (0x0423) Noise_Gate_Select_3L	11:0	OUT3L_NGATE_SRC[11:0]	0x010	
R1063 (0x0427) Noise_Gate_Select_3R	11:0	OUT3R_NGATE_SRC[11:0]	0x020	
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

4.11.7 Output Signal Path AEC Loop-Back

The CS42L92 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 CS42L92 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. 4-52. The AEC loop-back signals can be selected as input to any of the digital mixers within the CS42L92 digital core. The sample rate for the AEC loop-back paths is configured using OUT_RATE—see Table 4-26.

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 CS42L92 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 4-80. 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 0100 = OUT3L All other codes are reserved 0001 = OUT1R 0101 = OUT3R 0010 = OUT2L 1000 = OUT5L 0011 = OUT2R 1001 = OUT5R
	1	AEC1_ENA_STS	0	Transmit (Tx) Path AEC1 Control Status 0 = Disabled 1 = Enabled
	0	AEC1_LOOPBACK_ ENA	0	Transmit (Tx) Path AEC1 Control 0 = Disabled 1 = Enabled
R1105 (0x0451) DAC_AEC_ Control_2	5:2	AEC2_LOOPBACK_ SRC[3:0]	0000	Input source for Tx AEC2 function 0000 = OUT1L 0100 = OUT3L All other codes are reserved 0001 = OUT1R 0101 = OUT3R 0010 = OUT2L 1000 = OUT5L 0011 = OUT2R 1001 = OUT5R
	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

4.11.8 Headphone Outputs

The headphone/earpiece driver outputs, HPOUT1–HPOUT4, are suitable for direct connection to external headphones and earpieces. The outputs are ground referenced, eliminating any requirement for AC coupling capacitors.

The headphone outputs incorporate a common mode, or ground loop, feedback path that provides rejection of system-related ground noise. The feedback connection is configurable for each output path. Five feedback pins are provided (HPOUTFB*n*). The selected feedback pins must be connected to ground for normal operation of the headphone outputs.

Note that the feedback pins should be connected to GND as close as possible to the respective headphone jack ground pin, as shown in Fig. 4-54. In mono (differential) mode, the feedback pins should be connected to the ground plane that is closest to the earpiece output PCB tracks.

The ground feedback path for HPOUT1 and HPOUT2 headphone outputs is selected using the HP1_GND_SEL and HP2_GND_SEL register fields respectively—see Table 4-81. The ground feedback path for HPOUT3 and HPOUT4 headphone outputs is selected using the HP3_GND_SEL field.

If none of the HPOUTFB*n* pins are available for use as a ground feedback connection, an internal ground (AGND) can be selected by setting the respective HP*n*_GND_SEL field to 101.

Table 4-81. Headphone Output (HPOUT) Ground Feedback Control

Register Address	Bit	Label	Default	Description
R1042 (0x0412) Output_Path_ Config_1	2:0	HP1_GND_ SEL[2:0]	000	HPOUT1 ground feedback pin select 000 = HPOUTFB1 001 = HPOUTFB2 010 = HPOUTFB3 011 = HPOUTFB4 100 = HPOUTFB5 101 = AGND (internal connection) All other codes are reserved
R1050 (0x041A) Output_Path_ Config_2	2:0	HP2_GND_ SEL[2:0]	010	HPOUT2 ground feedback pin select 000 = HPOUTFB1 001 = HPOUTFB2 010 = HPOUTFB3 011 = HPOUTFB4 100 = HPOUTFB5 101 = AGND (internal connection) All other codes are reserved
R1058 (0x0422) Output_Path_ Config_3	2:0	HP3_GND_ SEL[2:0]	010	HPOUT3/HPOUT4 ground feedback pin select 000 = HPOUTFB1 001 = HPOUTFB2 010 = HPOUTFB3 011 = HPOUTFB4 100 = HPOUTFB5 101 = AGND (internal connection) All other codes are reserved

The headphone and earpiece connections are shown in Fig. 4-54.

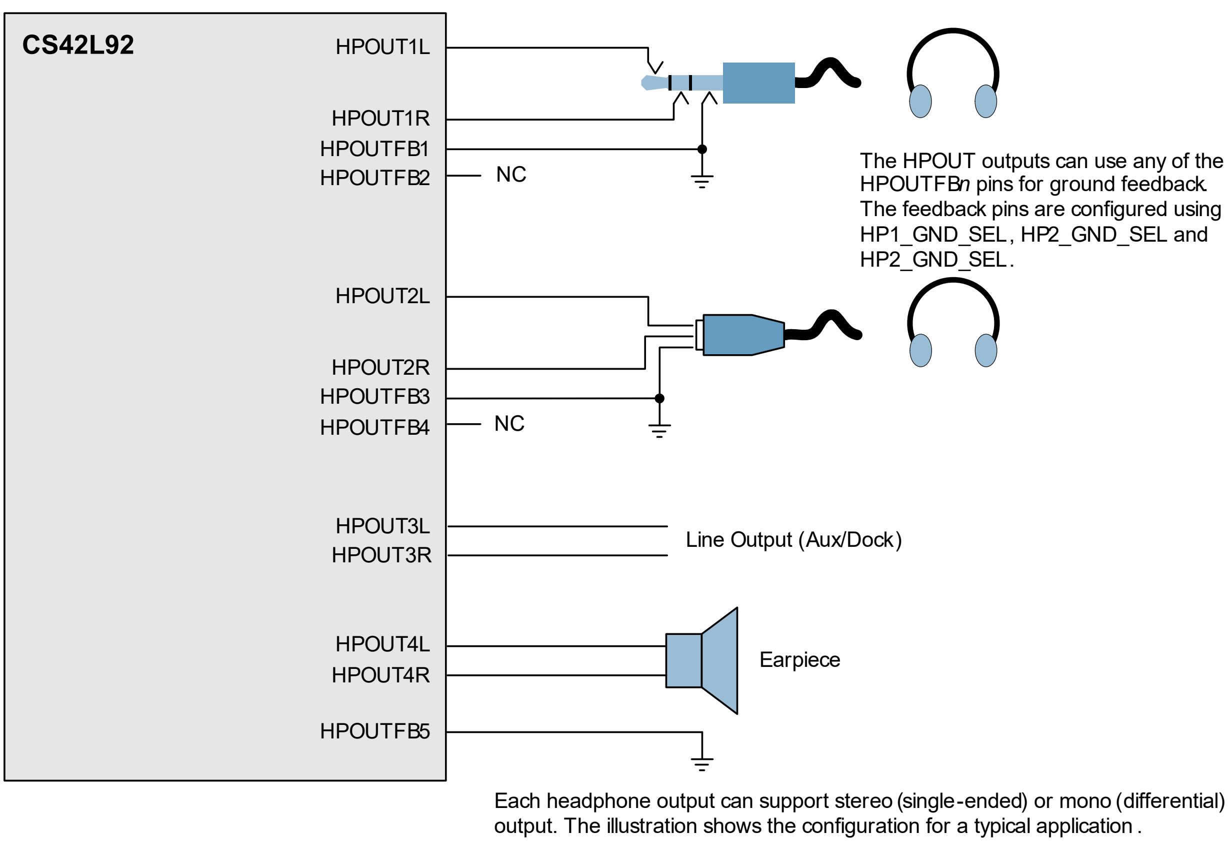


Figure 4-54. Headphone and Earpiece Connection

4.11.9 Speaker Outputs (Digital PDM)

The CS42L92 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 multifunction 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. 4-55.

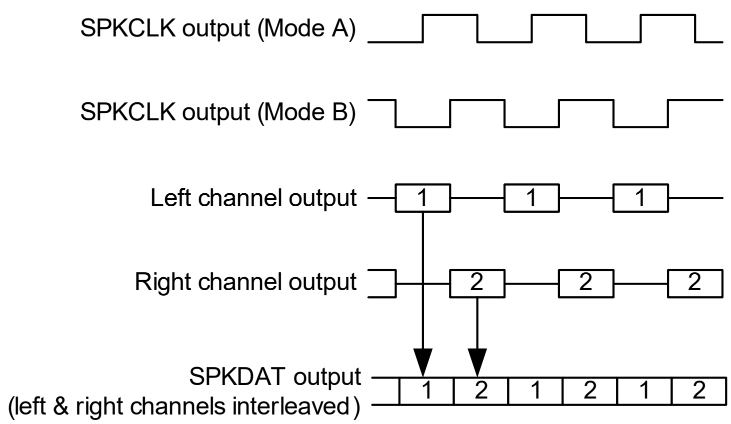


Figure 4-55. Digital Speaker (PDM) Interface Timing

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.

If the OUT5L or OUT5R output signal path is enabled, the PDM interface clock signal is output on the SPKCLK pin.

The output signal paths support normal and high performance operating modes, as described in Section 4.11.3. The SPKCLK frequency is set according to the operating mode of the relevant output path, as described in Table 4-82. The OUT5_OSR bit is defined in Table 4-74.

Note that the SPKCLK frequencies noted in Table 4-82 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 4-82. 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 4-77 for details of the OUT5L_MUTE and OUT5R_MUTE bits.

The PDM output interface registers are described in Table 4-83.

Table 4-83. 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. 4-56.

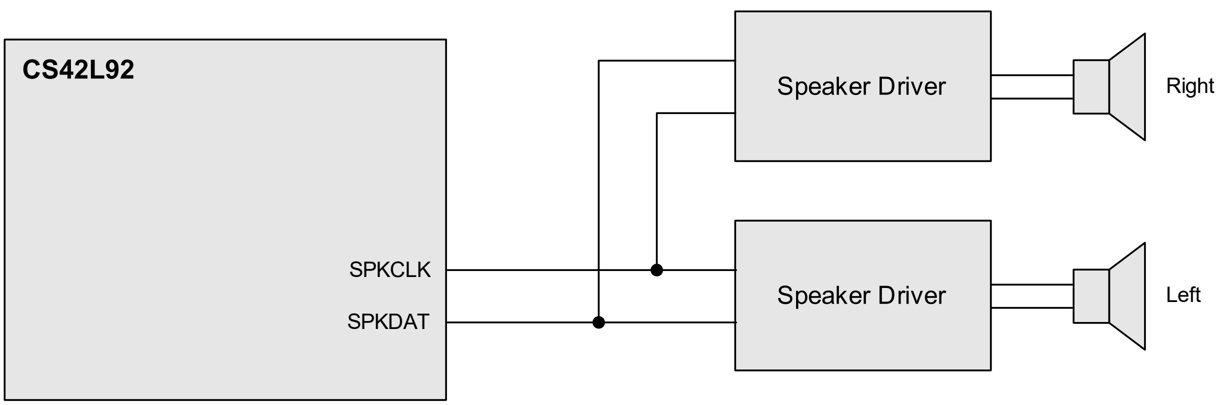


Figure 4-56. Digital Speaker (PDM) Connection

4.12 External Accessory Detection

The CS42L92 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 JACKDET*n* pins (where *n* = 1, 2, or 3), which are typically 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. (The CS42L92 incorporates two MICDET clamps.) This function can also be used to trigger interrupt events, and to trigger the control-write sequencer. The integrated general-purpose switches can be synchronized with the MICDET clamps, to provide additional pop-suppression capability.

Microphones, push buttons and other accessories can be detected via the MICDET*n* pins. The presence of a microphone, and the status of a hook switch can be detected. This feature can also be used to detect push-button operation. (Note that accessory detection is also possible via the HPDET*n* and JACKDET*n* pins, subject to some additional constraints.)

Headphone impedance can be measured via the HPDET1 and HPDET2 pins; this can be used to set different gain levels or other configuration settings according to the type of load connected. For example, different settings may be applicable to headphone or line output loads. (Note that impedance measurement is also possible via the MICDET*n* and JACKDET*n* pins, subject to some additional constraints.)

The 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 CS42L92 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 CS42L92.

Jack insertion and removal is detected using the JACKDET*n* pins. The recommended external connections are shown in Fig. 4-57. Note that the logic thresholds associated with the JACKDET*n* pins 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, 2, or 3 for JACKDET1, JACKDET2, or JACKDET3 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 4-84. 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 (JD*n*) 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 JD*n* signals.

The control registers associated with the jack detect function are described in Table 4-84.

Table 4-84. Jack Detect Control

Register Address	Bit	Label	Default	Description
R723 (0x02D3) Jack_detect_analog	2	JD3_ENA	0	JACKDET3 enable 0 = Disabled 1 = Enabled
	1	JD2_ENA	0	JACKDET2 enable 0 = Disabled 1 = Enabled
	0	JD1_ENA	0	JACKDET1 enable 0 = Disabled 1 = Enabled

Table 4-84. Jack Detect Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R6278 (0x1886) IRQ1_Raw_ Status_7	8	JD3_STS1	0	JACKDET3 input status 0 = Jack not detected 1 = Jack is detected (Assumes the JACKDET3 pin is pulled low on jack insertion.)
	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	8	JD3_STS2	0	JACKDET3 input status 0 = Jack not detected 1 = Jack is detected (Assumes the JACKDET3 pin is pulled low on jack insertion.)
	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	8	JD3_DB	0	JACKDET3 input debounce 0 = Disabled 1 = Enabled
	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 HPOUT1 and microphone connections, is shown in Fig. 4-57. See Section 5.1 for details of recommended external components.

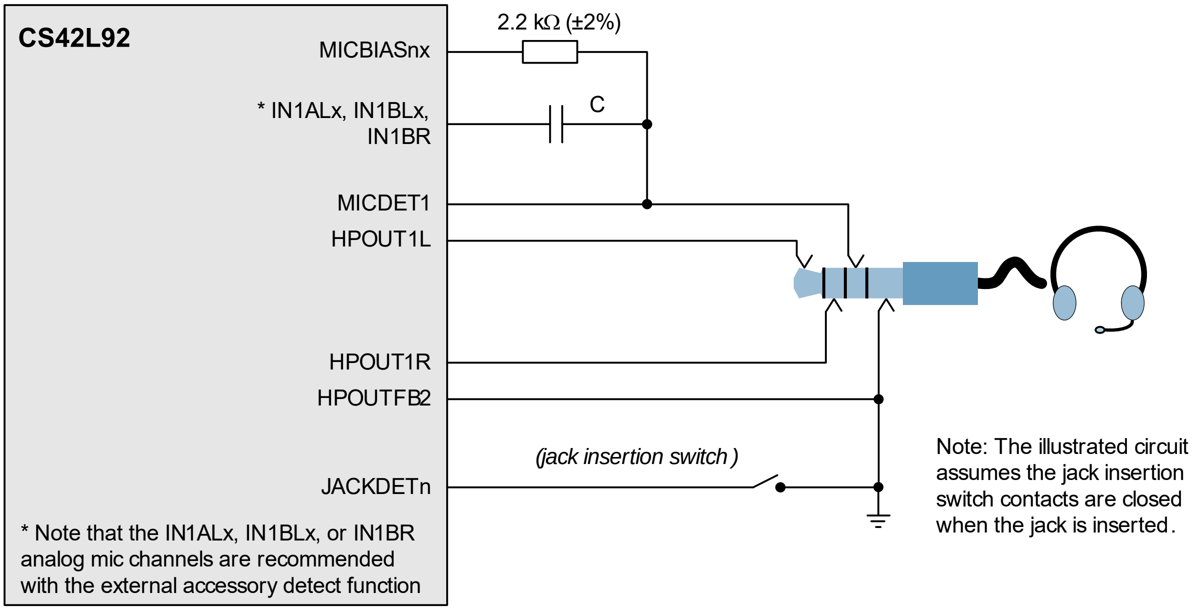


Figure 4-57. Jack Detect and External Accessory Connections

The internal comparator circuit used to detect the JACKDET*n* status is shown in Fig. 4-58. 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.

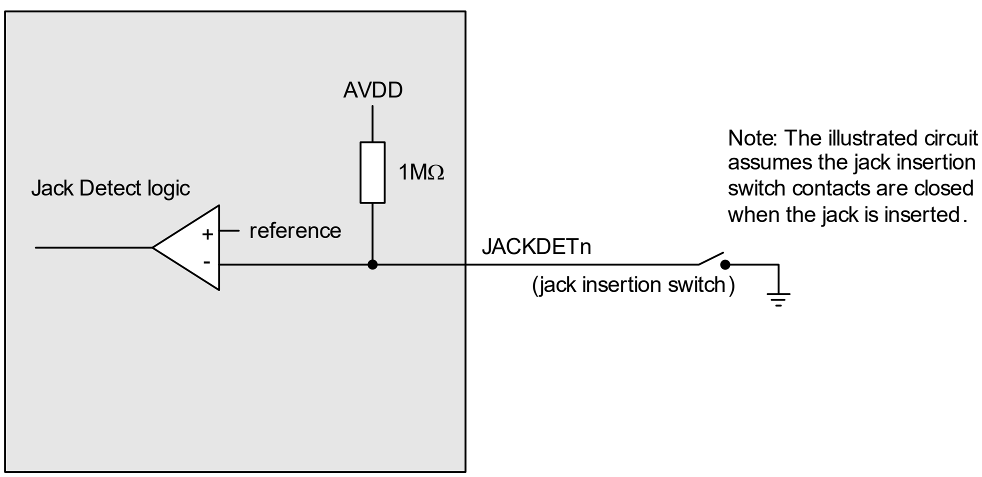


Figure 4-58. Jack Detect Comparator

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

The CS42L92 provides a MICDET clamp function to suppress pops and clicks caused by jack insertion or removal. Two MICDET clamps are incorporated; each clamp can be controlled directly, or can be activated by a logic function derived from the JACKDET*n* inputs. The clamp status can be read using the relevant register status bits. The clamp status can also be used to trigger an interrupt (IRQ) event or to trigger the control-write sequencer.

Two general-purpose analog switches are incorporated; these can be configured to augment the MICDET clamp functions and used to support the pop-suppression circuits, as described in Section 4.12.2.3.

Note that, due to control logic that is shared between the two MICDET clamps, some restrictions regarding the operation of the clamps must be observed, as described in Section 4.12.2.1.

4.12.2.1 MICDET Clamp Control

The MICDET clamp function can be configured using the MICD_CLAMP*n*_MODE fields (where *n* = 1 or 2 for Clamp 1 or Clamp 2 respectively). Selectable logic conditions (derived from the JD1, JD2, or JD3 signals—see Table 4-84) provide support for different jack detect circuit configurations. Setting the MICD_CLAMP*n*_OVD bit enables the respective MICDET clamp, regardless of other conditions.

Note: The MICD_CLAMP*n*_OVD bits are set by default. Accordingly, the MICDET clamps are always enabled following power-on reset, hardware reset, or software reset.

The MICDET clamp functionality (including the external IRQ) is maintained in Sleep Mode (see Section 4.13). This enables a jack insertion event to be used to trigger a wake-up of the CS42L92. The recommended control sequence for the jack detect and MICDET clamp control is described in Section 4.12.2.5.

The MICDET clamps are effective on pins MICDET1–MICDET4, as follows:

- If MICDET Clamp 1 is enabled, the MICDET1/HPOUTFB1 and MICDET2/HPOUTFB2 pins are shorted together.
- If MICDET Clamp 2 is enabled, the MICDET3/HPOUTFB3 and MICDET4/HPOUTFB4 pins are shorted together.

If either clamp is enabled, the grounding of the applicable MICDET is achieved via the HPOUTFB function of the other pin associated with the respective clamp. It is assumed that the HPOUTFB connection is grounded externally, as shown in Fig. 4-59.

The selectable logic conditions supported by the MICD_CLAMP*n*_MODE fields provide flexibility in selecting the appropriate conditions for controlling each MICDET clamp. The MICD_CLAMP1_MODE field allows Clamp 1 to be controlled by the JD1 or JD2 signals. The MICD_CLAMP2_MODE field allows Clamp 2 to be controlled by the JD3 signal.

Note that, due to control logic that is shared between the two clamps, the option to control both clamps in response to the JD*n* signals cannot be supported at the same time. If automatic (JACKDET-triggered) operation is selected on one of the clamps, the other clamp must be held in a fixed state. It is recommended that the unused clamp should be enabled by setting the respective MICD_CLAMP*n*_OVD bit.

The status of the clamps 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 MICD_CLAMP_STS*x* bits indicate the status of the active clamp only (i.e., the clamp that is not in the overridden state). The status of a clamp in the overridden (MICD_CLAMP*n*_OVD = 1) state is not indicated. It is assumed that a maximum of one clamp is active at any time.

The MICDET clamp debounce is selected by setting MICD_CLAMP_DB bit, as described in Table 4-85. 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. 4-59. Note that the jack plug is shown partially removed, with the MICDET1 pin in contact with the headphone load.

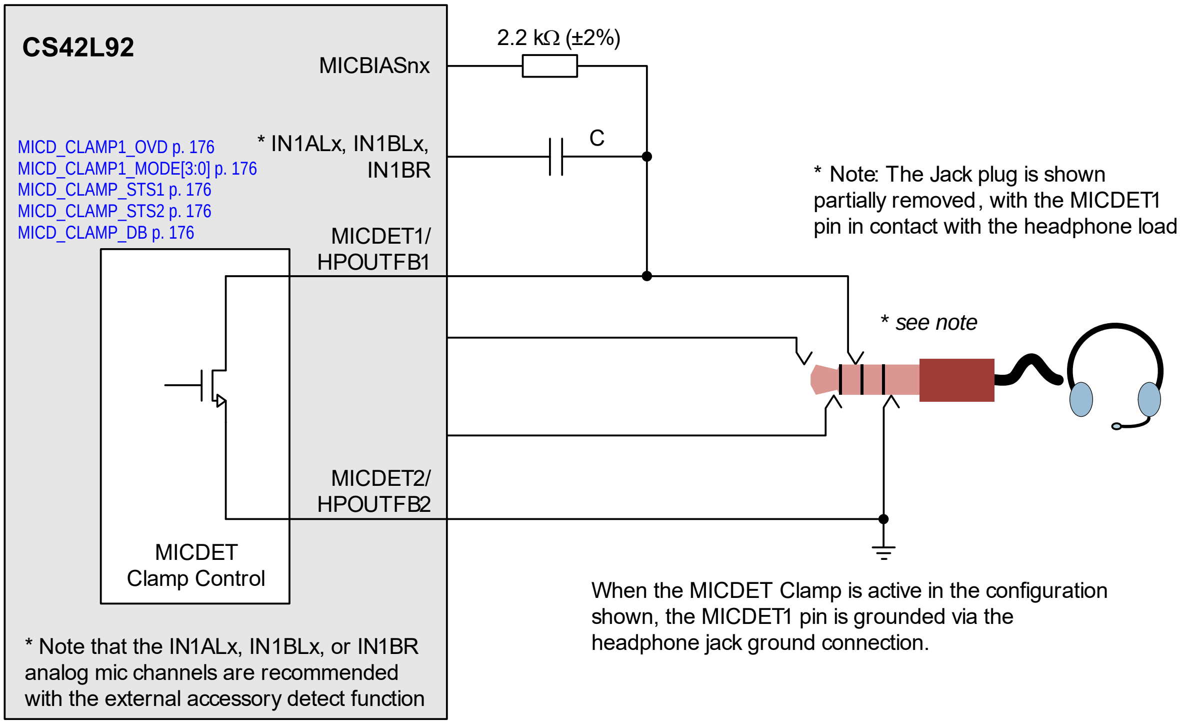


Figure 4-59. MICDET Clamp Circuit

4.12.2.2 Interrupts and Write-Sequencer Control

An interrupt request (IRQ) event can be generated in response to the MICDET clamp status. A MICDET clamp interrupt is generated whenever the logic condition of the JD*n* signals cause a change in the clamp status. Separate maskable interrupts are provided for the rising and falling edges of the MICDET clamp status—see Section 4.15.

The control-write sequencer can be triggered by the MICDET clamp status. This is enabled using the WSEQ_ENA_MICD_CLAMP_FALL and WSEQ_ENA_MICD_CLAMP_RISE bits. Note that these control-sequencer events are only valid if the clamp status changed in response to the JD*n* signals. See Section 4.18 for details of the control-write sequencer.

Note that, due to control logic that is shared between the two MICDET clamps, the option to control both clamps in response to the JD*n* signals cannot be supported at the same time. It is assumed that a maximum of one clamp is active at any time. Accordingly, only the active clamp is capable of generating an interrupt event or triggering the control-write sequencer.

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 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 one of the general-purpose switches on the CS42L92 to provide isolation from the MICBIAS output.

There are two general-purpose switches, configured using the respective SW*n*_MODE fields (where n = 1 or 2). The SW*n*_MODE fields allow the switches to be disabled, enabled, or synchronized to the MICDET clamp status, as described in Table 4-85.

For jack pop suppression, it is recommended to set SW*n*_MODE = 11 for the applicable switch. In this case, the switch contacts are open whenever the MICDET clamp status bits are set (clamp enabled), and the switch contacts are closed whenever the MICDET clamp status bits are clear (clamp disabled).

Note that the MICDET clamp status (MICD_CLAMP_STS*x*) is shared between the two clamps. Under recommended operating conditions, a maximum of one clamp should be active at any time (see Section 4.12.2.1). In this case, the MICD_CLAMP_STS*x* bits indicate the status of the active clamp only.

A typical pop-suppression circuit, incorporating the general-purpose switch and MICDET clamp function, is shown in Fig. 4-60. Normal accessory functions are supported when the switch contacts (GPSW*n*P and GPSW*n*N) are closed, and the MICDET clamp is disabled. Ground clamping of MICDET, and isolation of MICBIAS, are achieved when the switch contacts are open and the MICDET clamp is enabled.

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

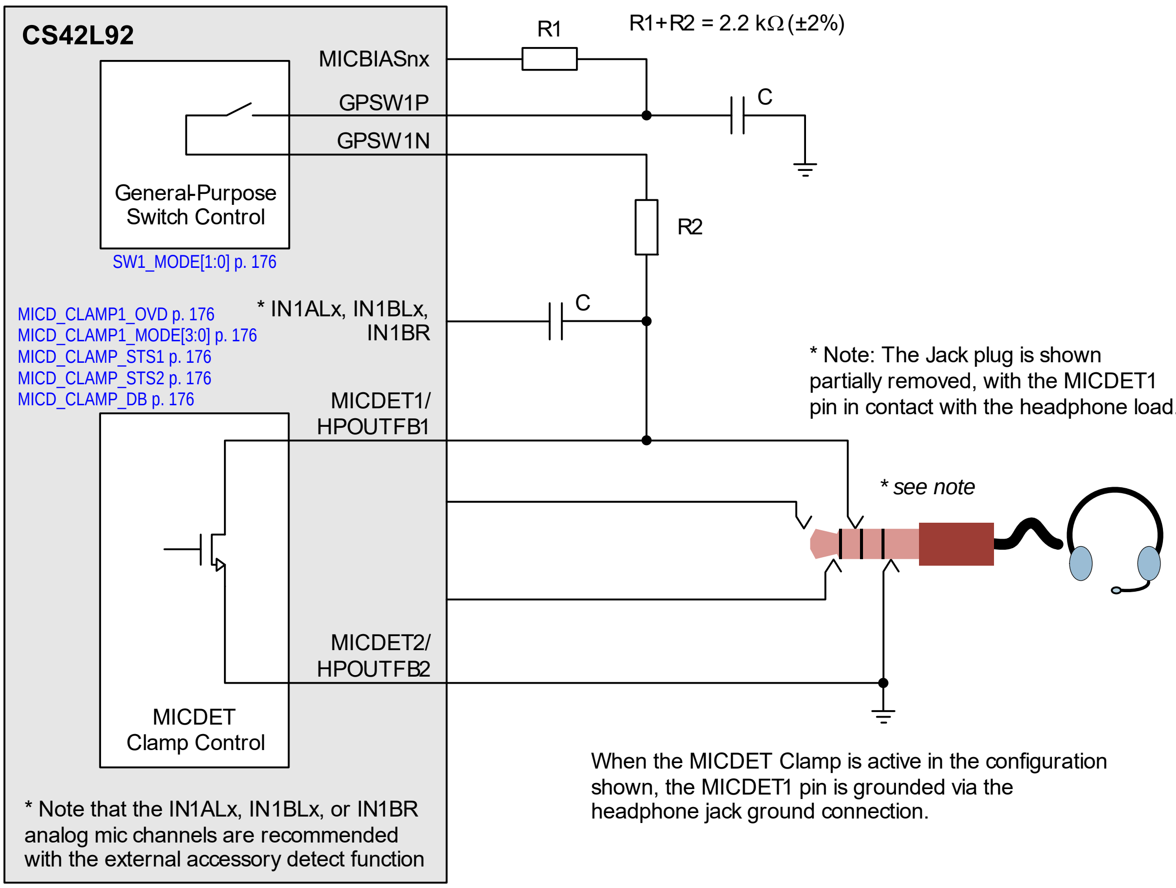


Figure 4-60. General-Purpose Switch Circuit

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 4-85.

Table 4-85. MICDET Clamp and General-Purpose Switch Control

Register Address	Bit	Label	Default	Description
R65 (0x0041) Sequence_control	7	WSEQ_ENA_ MICD_CLAMP_ FALL	0	MICDET Clamp (Falling) Write Sequencer Select 0 = Disabled 1 = Enabled
	6	WSEQ_ENA_ MICD_CLAMP_ RISE	0	MICDET Clamp (Rising) Write Sequencer Select 0 = Disabled 1 = Enabled

Table 4-85. MICDET Clamp and General-Purpose Switch Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R710 (0x02C6) Micd_Clamp_control	9	MICD_CLAMP2_OVD	1	MICDET Clamp 2 Override 0 = Disabled (clamp is controlled by MICD_CLAMP2_MODE) 1 = Enabled (clamp is enabled)
	8:6	MICD_CLAMP2_MODE[2:0]	000	MICDET Clamp 2 Mode 0x0 = Disabled 0x1 = Enabled (MICDET3/MICDET4 shorted together) 0x2 = Enabled if JD3=0 0x3 = Enabled if JD3=1
	4	MICD_CLAMP1_OVD	1	MICDET Clamp 1Override 0 = Disabled (clamp is controlled by MICD_CLAMP1_MODE) 1 = Enabled (clamp is enabled)
	3:0	MICD_CLAMP1_MODE[3:0]	0000	MICDET Clamp 1 Mode 0x0 = Disabled 0x1 = Enabled (MICDET1/MICDET2 shorted together) 0x2–0x3 = Reserved 0x4 = Enabled if JD1=0 0x5 = Enabled if JD1=1 0x6 = Enabled if JD2=0 0x7 = Enabled if JD2=1 0x8 = Enabled if JD1=0 or JD2=0 0x9 = Enabled if JD1=0 or JD2=1 0xA = Enabled if JD1=1 or JD2=0 0xB = Enabled if JD1=1 or JD2=1 0xC = Enabled if JD1=0 and JD2=0 0xD = Enabled if JD1=0 and JD2=1 0xE = Enabled if JD1=1 and JD2=0 0xF = Enabled if JD1=1 and JD2=1
R712 (0x02C8) GP_Switch_1	3:2	SW2_MODE[1:0]	00	General-purpose Switch 2 control 00 = Disabled (switch open) 01 = Enabled (switch closed) 10 = Enabled if MICDET clamp status is set 11 = Enabled if MICDET clamp status is clear
	1:0	SW1_MODE[1:0]	00	General-purpose Switch 1 control 00 = Disabled (switch open) 01 = Enabled (switch closed) 10 = Enabled if MICDET clamp status is set 11 = Enabled if MICDET clamp status is clear
R6278 (0x1886) IRQ1_Raw_Status_7	4	MICD_CLAMP_STS1	0	MICDET Clamp status 0 = Clamp disabled 1 = Clamp enabled **Note:** Separate _STS bits are not provided for each clamp—it is assumed that a maximum of one clamp is active at any time. The clamp override condition (MICD_CLAMP*n*_OVD = 1) is not indicated.
R6534 (0x1986) IRQ2_Raw_Status_7	4	MICD_CLAMP_STS2	0	MICDET Clamp status 0 = Clamp disabled 1 = Clamp enabled **Note:** Separate _STS bits are not provided for each clamp—it is assumed that a maximum of one clamp is active at any time. The clamp override condition (MICD_CLAMP*n*_OVD = 1) is not indicated.
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 clamps are enabled due to the default setting of the MICD_CLAMP*n*_OVD bits; this ensures no spurious output can occur during jack insertion. It is recommended to keep the MICDET clamps enabled (MICD_CLAMP*n*_OVD = 1) until after a jack insertion has been detected.

  To control MICDET Clamp 1 according to the JD1/JD2 signals, the MICDET_CLAMP1_MODE field should be set according to the required logic condition (configured to enable the clamp when jack is removed).

  To control MICDET Clamp 2 according to the JD3 signal, the MICDET_CLAMP2_MODE field should be set according to the required logic condition (configured to enable the clamp when jack is removed).

  Note that, due to control logic that is shared between the two MICDET clamps, the option to control both clamps in response to the JD*n* signals cannot be supported at the same time. It is assumed that a maximum of one clamp is active at any time.

- Jack insertion is indicated using the JD*n* signals or MICDET clamp interrupts (assuming that the MICD_CLAMP*n*_MODE field has been correctly set for the applicable JD*n* 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 associated MICDET clamp should be disabled (MICD_CLAMP*n*_OVD = 0) immediately before enabling the MICBIAS (or immediately before enabling MICD*n*_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 JD*n* signals or MICDET clamp interrupts. The associated status bits can be read directly, or can be unmasked as inputs to the interrupt controller. The MICDET clamp ensures fast and automatic silencing of the jack outputs.

  Under typical use cases, the respective MICBIAS generator and headset audio paths should all be disabled following jack removal.

  After jack removal has been detected, the MICDET clamp override bit (MICD_CLAMP*n*_OVD) should be set, to make the system ready for a jack insertion.

The recommended control sequence for jack detect and MICDET clamp is summarized in Table 4-86.

Table 4-86. Control Sequence for Jack Detect and MICDET Clamp

Event	Device Actions	Recommended User Actions
Initial condition	Clamp enabled by default	Configure MICD_CLAMP*n*_MODE
Jack insertion	Jack insertion signaled via IRQ	For headphone-only operation: Enable output signal paths For other use cases: Disable clamp, MICD_CLAMP*n*_OVD = 0 Enable MICBIAS and MICDET Enable I/O signal paths
Jack removal	Jack removal signaled via IRQ, Clamp enabled automatically	Disable MICBIAS and MICDET Disable I/O signal paths Enable clamp MICD_CLAMP*n*_OVD = 1

4.12.3 Microphone Detect

The CS42L92 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 external impedance connected to the MICDET*n* pins. In the discrete measurement mode, the function reports whether the measured impedance lies within one of eight predefined levels. In the ADC measurement mode, a more specific result is provided in the form of a 7-bit ADC output.

Note that microphone/accessory detection is also possible via the HPDET*n* and JACKDET*n* pins, subject to some additional constraints. If the measurement (sense) pin is connected to MICVDD or MICBIAS*nx* (typically via a 2.2-kΩ bias resistor), MICDET*n* must always be used.

The microphone detection circuit typically uses one of the MICBIAS outputs as a reference. The CS42L92 automatically enables the appropriate MICBIAS output when required in order to perform the detection function; this allows the detection function to be supported in low-power standby operating conditions.

The 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.

The CS42L92 provides two microphone detection circuits, which are independently configurable. Detection can be enabled on both circuits simultaneously.

To configure the microphone detection circuit, the applicable pin connections for the intended measurement must be written to the MICD*n*_SENSE_SEL and MICD*n*_GND_SEL fields (where *n* identifies the respective detection circuit, 1 or 2). The respective detection circuit measures the external impedance between the pins selected by these two fields; the valid selections for each are defined in Table 4-87.

Note: There is no requirement for the SENSE and GND pin selections to be uniquely assigned between the microphone detect and headphone detect functions—the same pin may be used as a SENSE or GND connection for more than one of the detection functions. If multiple microphone/headphone detections are enabled, the respective measurements are automatically scheduled in isolation to each other. See Section 4.12.4 for details of the headphone detect function.

The microphone detection circuit uses MICVDD, or any one of the MICBIAS*xy* sources, as a reference. The applicable source is configured using the MICD*n*_BIAS_SRC field. If HPDET*n* or JACKDET*n* is selected as the measurement pin, MICD*n*_BIAS_SRC should be set to 1111.

The microphone detection function is enabled by setting MICD*n*_ENA.

When microphone detection is enabled, the CS42L92 performs a number of measurements in order to determine the external impedance between the selected pins. The measurement process is repeated at a cyclic rate controlled by MICD*n*_RATE. The MICD*n*_RATE field selects the delay between completion of one measurement and the start of the next. When the microphone detection result has settled, the CS42L92 indicates valid data by setting MICD*n*_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 (MICD*n*_ADC_MODE = 0), the measured impedance is only deemed valid after more than one successive measurement has produced the same result. The MICD*n*_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 CS42L92 indicates valid data by setting the MICD*n*_VALID bit. The measured impedance is indicated using the MICD*n*_LVL and MICD*n*_STS bits, as described in Table 4-87.

  The MICD*n*_VALID bit, when set, remains asserted for as long as the microphone detection function is enabled (i.e., while MICD*n*_ENA = 1). If the detected impedance changes, the MICD*n*_LVL and MICD*n*_STS fields change, but the MICD*n*_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*n*_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™ wired headset specification.

  Note that, for typical headset detection, the choice of external resistance values must take into account the impedance of the microphone—the detected impedance corresponds to the combined parallel resistance of the microphone and any asserted push button. Examples of suitable external components are described in Section 5.1.7.

- In the ADC measurement mode (MICD*n*_ADC_MODE = 1), the detection function generates two output results, contained within the MICD*n*_ADCVAL and MICD*n*_ADCVAL_DIFF fields. These fields contain the most recent measurement value (MICD*n*_ADCVAL) and the measurement difference value (MICD*n*_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 CS42L92 indicates valid data (MICD*n*_VALID = 1), the detection must be disabled by setting MICD*n*_ENA = 0.

  Note that MICD*n*_ADCVAL and MICD*n*_ADCVAL_DIFF do not follow a linear coding. The appropriate test condition for accepting the measurement value (or for rescheduling the measurement) varies depending on the application requirements, and depending on the expected impedance value.

The microphone detection functions are inputs to the interrupt control circuit and can be used to trigger an interrupt event every time an accessory insertion, removal, or impedance change is detected; see Section 4.15.

The fields associated with microphone detection (or other accessories) are described in Table 4-87. The external circuit configuration is shown in Fig. 4-61.

Table 4-87. Microphone Detect Control

Register Address	Bit	Label	Default	Description
R674 (0x02A2) Mic_Detect_1_ Control_0	15	MICD1_ADC_ MODE	0	Mic Detect 1 Measurement Mode 0 = Discrete Mode 1 = ADC Mode
	7:4	MICD1_SENSE_ SEL[3:0]	0001	Mic Detect 1 Sense Select 0000 = MICDET1 0001 = MICDET2 0010 = MICDET3 0011 = MICDET4 0100 = HPDET1 0101 = HPDET2 0110 = JACKDET1 0111 = JACKDET2 1000 = MICDET5 1001 = JACKDET3 All other codes are reserved
	2:0	MICD1_GND_ SEL[2:0]	000	Mic Detect 1 Ground Select 000 = MICDET1/HPOUTFB1 001 = MICDET2/HPOUTFB2 010 = MICDET3/HPOUTFB3 011 = MICDET4/HPOUTFB4 100 = MICDET5/HPOUTFB5 All other codes are reserved
R675 (0x02A3) Mic_Detect_1_ Control_1	15:12	MICD1_BIAS_ STARTTIME[3:0]	0001	Mic Detect 1 Bias Start-up Delay (Selects the delay time between enabling the MICBIASnx reference and performing the MICDET function.) 0000 = 0 ms (continuous) 0001 = 0.25 ms 0010 = 0.5 ms 0011 = 1 ms 0100 = 2 ms 0101 = 4 ms 0110 = 8 ms 0111 = 16 ms 1000 = 32 ms 1001 = 64 ms 1010 = 128 ms 1011 = 256 ms 1100 = 512 ms 1101 = 24 ms 1110 to 1111 = 512 ms
	11:8	MICD1_ RATE[3:0]	0001	Mic Detect 1 Rate (Selects the delay between successive MICDET measurements.) 0000 = 0 ms (continuous) 0001 = 0.25 ms 0010 = 0.5 ms 0011 = 1 ms 0100 = 2 ms 0101 = 4 ms 0110 = 8 ms 0111 = 16 ms 1000 = 32 ms 1001 = 64 ms 1010 = 128 ms 1011 = 256 ms 1100 = 512 ms 1101 = 24 ms 1110 to 1111 = 512 ms
	7:4	MICD1_BIAS_ SRC[3:0]	0000	Mic Detect 1 Reference Select 0000 = MICBIAS1A 0001 = MICBIAS1B 0010 = MICBIAS1C 0011 = MICBIAS1D 0100 = MICBIAS2A 0101 = MICBIAS2B 1111 = MICVDD All other codes are reserved
	1	MICD1_DBTIME	1	Mic Detect 1 Debounce 0 = 2 measurements 1 = 4 measurements Only valid when MICD1_ADC_MODE = 0.
	0	MICD1_ENA	0	Mic Detect 1 Enable 0 = Disabled 1 = Enabled
R676 (0x02A4) Mic_Detect_1_ Control_2	7:0	MICD1_LVL_ SEL[7:0]	1001_ 1111	Mic Detect 1 Level Select (enables mic/accessory detection in specific impedance ranges) [7] = Enable 1–30 kΩ detection [6] = Not used [5] = Not used [4] = Not used [3] = Not used [2] = Enable 360–680 Ω detection [1] = Enable 210–290 Ω detection [0] = Enable 110–180 Ω detection Only valid when MICD1_ADC_MODE = 0.

Table 4-87. Microphone Detect Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R677 (0x02A5) Mic_Detect_1_ Control_3	10:2	MICD1_LVL[8:0]	0_0000_ 0000	Mic Detect 1 Level (indicates the measured impedance) [8] = 1–30 kΩ [7] = Not used [6] = Not used [5] = Not used [4] = Not used [3] = 360–680 Ω [2] = 210–290 Ω [1] = 110–180 Ω [0] = 0–70 Ω Accessory detection is assured within the specified impedance limits. Note that other impedance conditions, including loads >30 kΩ, may also be indicated using these bits. Only valid when MICD1_ADC_MODE = 0.
	1	MICD1_VALID	0	Mic Detect 1 Data Valid 0 = Not Valid 1 = Valid
	0	MICD1_STS	0	Mic Detect 1 Status 0 = Mic/accessory not detected 1 = Mic/accessory detected Mic/accessory detection is assured for load impedance up to 30 kΩ. Only valid when MICD1_ADC_MODE = 0.
R683 (0x02AB) Mic_Detect_1_ Control_4	15:8	MICD1_ ADCVAL_ DIFF[7:0]	0x00	Mic Detect 1 ADC Level (Difference) Only valid when MICD1_ADC_MODE = 1.
	6:0	MICD1_ ADCVAL[6:0]	0x00	Mic Detect 1 ADC Level Only valid when MICD1_ADC_MODE = 1.
R690 (0x02B2) Mic_Detect_2_ Control_0	15	MICD2_ADC_ MODE	0	Mic Detect 2 Measurement Mode 0 = Discrete Mode 1 = ADC Mode
	7:4	MICD2_SENSE_ SEL[3:0]	0001	Mic Detect 2 Sense Select 0000 = MICDET1 0001 = MICDET2 0010 = MICDET3 0011 = MICDET4 0100 = HPDET1 0101 = HPDET2 0110 = JACKDET1 0111 = JACKDET2 1000 = MICDET5 1001 = JACKDET3 All other codes are reserved
	2:0	MICD2_GND_ SEL[2:0]	000	Mic Detect 2 Ground Select 000 = MICDET1/HPOUTFB1 001 = MICDET2/HPOUTFB2 010 = MICDET3/HPOUTFB3 011 = MICDET4/HPOUTFB4 100 = MICDET5/HPOUTFB5 All other codes are reserved

Table 4-87. Microphone Detect Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R691 (0x02B3) Mic_Detect_2_ Control_1	15:12	MICD2_BIAS_ STARTTIME[3:0]	0001	Mic Detect 2 Bias Start-up Delay (Selects the delay time between enabling the MICBIASnx reference and performing the MICDET function.) 0000 = 0 ms (continuous) 0101 = 4 ms 1010 = 128 ms 0001 = 0.25 ms 0110 = 8 ms 1011 = 256 ms 0010 = 0.5 ms 0111 = 16 ms 1100 = 512 ms 0011 = 1 ms 1000 = 32 ms 1101 = 24 ms 0100 = 2 ms 1001 = 64 ms 1110 to 1111 = 512 ms
	11:8	MICD2_ RATE[3:0]	0001	Mic Detect 2 Rate (Selects the delay between successive MICDET measurements.) 0000 = 0 ms (continuous) 0101 = 4 ms 1010 = 128 ms 0001 = 0.25 ms 0110 = 8 ms 1011 = 256 ms 0010 = 0.5 ms 0111 = 16 ms 1100 = 512 ms 0011 = 1 ms 1000 = 32 ms 1101 = 24 ms 0100 = 2 ms 1001 = 64 ms 1110 to 1111 = 512 ms
	7:4	MICD2_BIAS_ SRC[3:0]	0000	Mic Detect 2 Reference Select 0000 = MICBIAS1A 0011 = MICBIAS1D 1111 = MICVDD 0001 = MICBIAS1B 0100 = MICBIAS2A All other codes are reserved 0010 = MICBIAS1C 0101 = MICBIAS2B
	1	MICD2_DBTIME	1	Mic Detect 2 Debounce 0 = 2 measurements 1 = 4 measurements Only valid when MICD2_ADC_MODE = 0.
	0	MICD2_ENA	0	Mic Detect 2 Enable 0 = Disabled 1 = Enabled
R692 (0x02B4) Mic_Detect_2_ Control_2	7:0	MICD2_LVL_ SEL[7:0]	1001_ 1111	Mic Detect 2 Level Select (enables mic/accessory detection in specific impedance ranges) [7] = Enable 1–30 kΩ detection [3] = Not used [6] = Not used [2] = Enable 360–680 Ω detection [5] = Not used [1] = Enable 210–290 Ω detection [4] = Not used [0] = Enable 110–180 Ω detection Only valid when MICD2_ADC_MODE = 0.
R693 (0x02B5) Mic_Detect_2_ Control_3	10:2	MICD2_LVL[8:0]	0_0000_ 0000	Mic Detect 2 Level (indicates the measured impedance) [8] = 1–30 kΩ [3] = 360–680 Ω [7] = Not used [2] = 210–290 Ω [6] = Not used [1] = 110–180 Ω [5] = Not used [0] = 0–70 Ω [4] = Not used Accessory detection is assured within the specified impedance limits. Note that other impedance conditions, including loads >30 kΩ, may also be indicated using these bits. Only valid when MICD2_ADC_MODE = 0.
	1	MICD2_VALID	0	Mic Detect 2 Data Valid 0 = Not Valid 1 = Valid
	0	MICD2_STS	0	Mic Detect 2 Status 0 = Mic/accessory not detected 1 = Mic/accessory detected Mic/accessory detection is assured for load impedance up to 30 kΩ. Only valid when MICD2_ADC_MODE = 0.
R699 (0x02BB) Mic_Detect_2_ Control_4	15:8	MICD2_ ADCVAL_ DIFF[7:0]	0x00	Mic Detect 2 ADC Level (Difference) Only valid when MICD2_ADC_MODE = 1.
	6:0	MICD2_ ADCVAL[6:0]	0x00	Mic Detect 2 ADC Level Only valid when MICD2_ADC_MODE = 1.

The external connections for the microphone detect circuit are shown in Fig. 4-61. 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 IN1AL*x*, IN1BL*x*, or IN1BR analog microphone input paths to ensure best immunity to electrical transients arising from the external accessory.

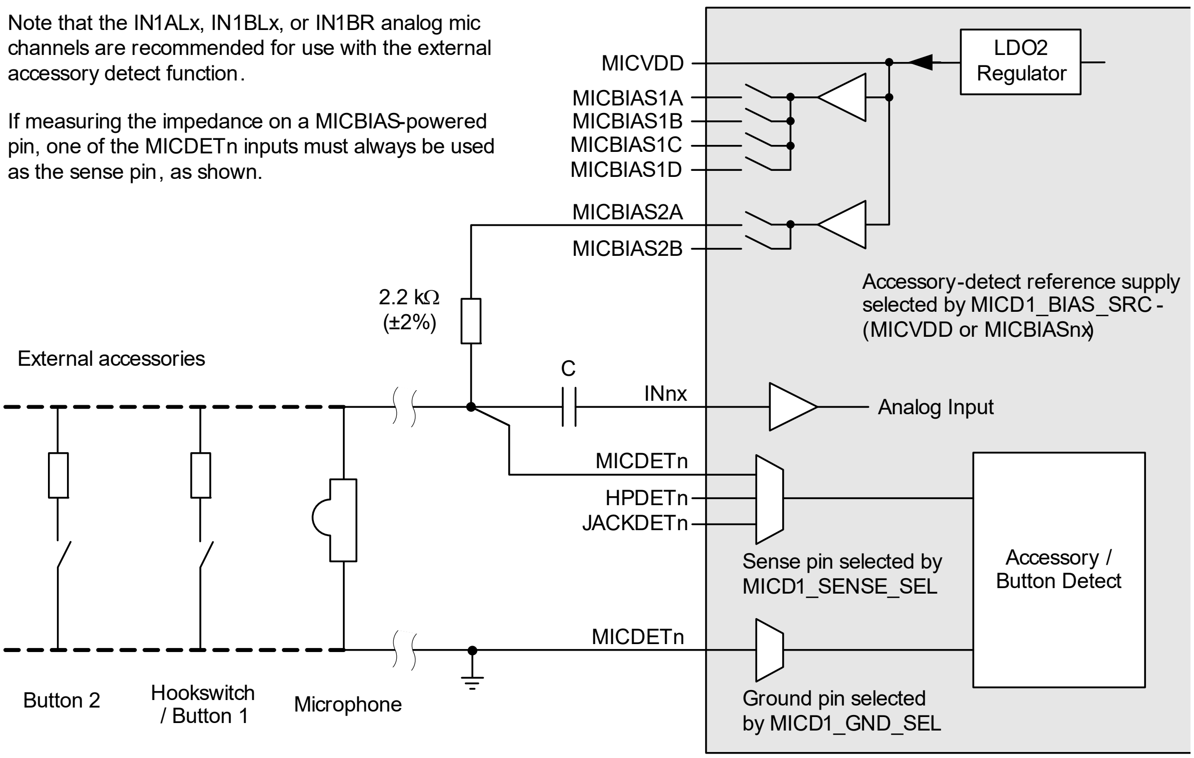


Figure 4-61. Microphone- and Accessory-Detect Interface

4.12.3.2 MICBIAS Reference Control

The voltage reference for the microphone detection is configured using the MICD*n*_BIAS_SRC field, as described in Table 4-87. The microphone detection function automatically enables the applicable reference when required for impedance measurement.

If the selected reference (MICBIAS*xy*) is not already enabled, the microphone detect circuit automatically enables the respective MICBIAS output for short periods of time only, every time the impedance measurement is scheduled. To allow time for the associated circuitry to stabilize, a time delay is applied before the measurement is performed; this is configured using MICD*n*_BIAS_STARTTIME, as described in Table 4-87. If the measurement rate setting (MICD*n*_RATE) is greater than 0x0, the delay (MICD*n*_BIAS_STARTTIME) should be set to 0.25 ms or more.

Note: The microphone detection automatically enables the applicable MICBIAS*xy* output switch, every time the impedance measurement is scheduled. The respective MICBIAS generator (MICBIAS1 or MICBIAS2) is not controlled automatically—the applicable generators must be enabled using the MICB1_ENA and MICB2_ENA bits, as described in Table 4-124.

The timing of the microphone detect function is shown in Fig. 4-62. Two different cases are shown, according to whether MICBIAS*xy* is enabled periodically by the impedance measurement function, or is enabled at all times.

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

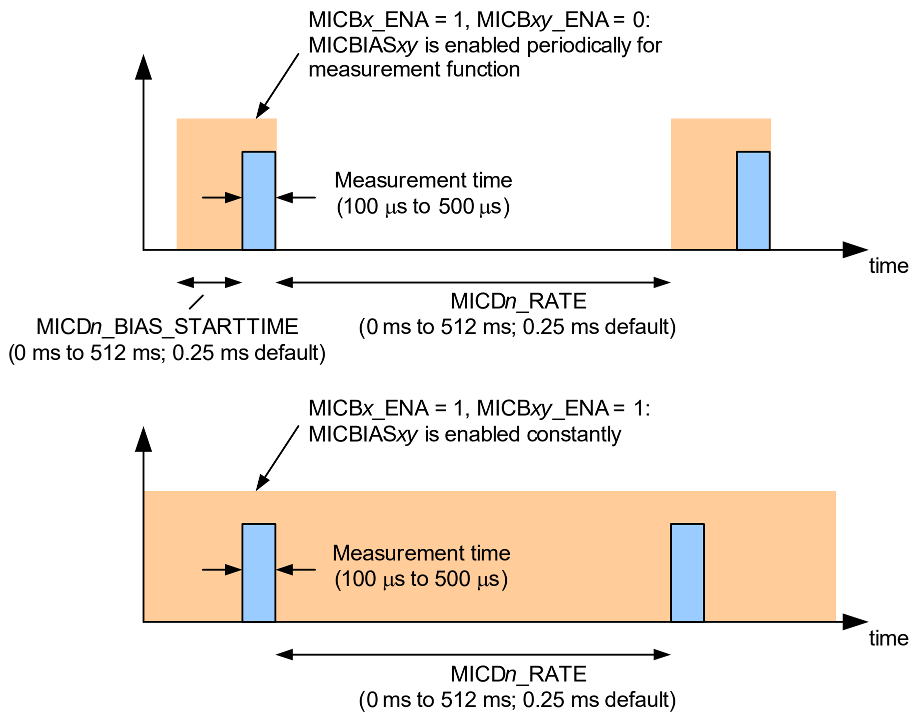


Figure 4-62. Microphone- and Accessory-Detect Timing

4.12.3.3 Measurement Range Control

When the discrete measurement mode is selected (MICD*n*_ADC_MODE = 0), the MICD*n*_LVL_SEL[7:0] bits allow each impedance measurement level to be enabled or disabled independently. This allows the function to be tailored to the particular application requirements.

If one or more bits within MICD*n*_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*n*_LVL_SEL[2] bit enables the detection of a 360–680 Ω impedance. If MICD*n*_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*n*_LVL field as MICD*n*_LVL[2] = 1.

With default register configuration, and all measurement levels enabled, the CS42L92 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 data sheet—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. 4-61. Examples of suitable external components are described in Section 5.1.7.

The accuracy of the microphone detect function is assured whenever the connected load is within the applicable limits specified in Table 3-11. It is required that a 2.2-kΩ (2%) resistor must also be connected between the measurement (SENSE) pin and the selected MICBIAS reference—different resistor values lead to inaccuracy in the impedance measurement.

Note that, for typical headset detection, the choice of external resistance values must take into account the impedance of the microphone—the detected impedance corresponds to the combined parallel resistance of the microphone and any asserted push button.

4.12.4 Headphone Detect

The CS42L92 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*n*.

4.12.4.1 Headphone Detection Control

The headphone detection circuit measures the external impedance connected to the HPDET*n* pins. In typical usage, this provides measurement of the load impedance on one or more of the headphone outputs (HPOUT1–4).

Note that impedance measurement is also possible via the MICDET*n* and JACKDET*n* pins, subject to some additional constraints. If the measurement (sense) pin is connected to one of the headphone outputs, then HPDET1, HPDET2, or JACKDET1 must always be used. The valid measurement range and the measurement accuracy are reduced, if using the MICDET*n* or JACKDET*n* pins.

To configure the headphone detection circuit, the applicable pin connections for the intended measurement must be written to the HPD_SENSE_SEL and HPD_GND_SEL fields. The headphone detection circuit measures the external impedance between the pins selected by these two fields; the valid selections for each are defined in Table 4-90.

When measuring the load impedance on one the HPOUT*n* output paths, the HPD_GND_SEL selection should be the same MICDET*n*/HPOUTFB*n* pin as the ground feedback pin for the applicable headphone output. See Section 4.11.8 to configure the ground feedback pin for the HPOUT*n* outputs.

The HPD_FRC_SEL field must also be configured, to select where the measurement current is applied. As a general rule, this should be the same as the HPD_SENSE_SEL pin. Other configurations can be used if required—for example, to improve measurement accuracy in cases where the SENSE input path includes significant unwanted resistance.

Note: There is no requirement for the SENSE and GND pin selections to be uniquely assigned between the microphone detect and headphone detect functions—the same pin may be used as a SENSE or GND connection for more than one of the detection functions. If multiple microphone/headphone detections are enabled, the respective measurements are automatically scheduled in isolation to each other. See Section 4.12.3 for details of the microphone detect function.

Headphone detection is commanded by writing 1 to HPD_POLL.

The impedance measurement range is configured using HPD_IMPEDANCE_RANGE. This field should be set in accordance with the expected load impedance. Note that a number of separate measurements are typically required to determine the load impedance; the recommended control requirements are described in Section 4.12.4.2.

Note: Setting HPD_IMPEDANCE_RANGE is not required for detection on the MICDET*n* or JACKDET*n* pins. The impedance measurement range, and measurement accuracy, in these cases are different to the HPDET1 and HPDET2 measurements.

If headphone detection is performed using a measurement pin connected to one of the headphone outputs, the respective output driver must be disabled before the measurement is commanded. The required settings are shown in Table 4-88.

Table 4-88. Output Configuration for Headphone Detect

Description	Requirement
HPOUT1L Impedance measurement	HPD_OVD_ENA = 1, HPD_OUT_SEL = 000, HP1L_ENA = 0
HPOUT1R Impedance measurement	HPD_OVD_ENA = 1, HPD_OUT_SEL = 001, HP1R_ENA = 0
HPOUT2L Impedance measurement	HPD_OVD_ENA = 1, HPD_OUT_SEL = 010, HP2L_ENA = 0
HPOUT2R Impedance measurement	HPD_OVD_ENA = 1, HPD_OUT_SEL = 011, HP2R_ENA = 0
HPOUT3L or HPOUT4L Impedance measurement	HPD_OVD_ENA = 1, HPD_OUT_SEL = 100, HP3L_ENA = 0
HPOUT3R or HPOUT4R Impedance measurement	HPD_OVD_ENA = 1, HPD_OUT_SEL = 101, HP3R_ENA = 0

Note: The applicable headphone outputs configuration must be maintained until after the headphone detection has completed. See Table 4-71 for details of the HP*nx*_ENA bits.

If headphone detection is performed using a measurement pin that is not connected to one of the headphone outputs, the HPD_OVD_ENA bit should be cleared.

If headphone detection is performed using a measurement pin that is also connected to one of the MICBIAS outputs, the respective MICBIAS output must be disabled and floating (MICB*nx*_ENA = 0, MICB*nx*_DISCH = 0).

When headphone detection is commanded, the CS42L92 uses an adjustable current source to determine the connected impedance. A sweep of measurement currents is applied. The rate of this sweep can be adjusted using HPD_CLK_DIV and HPD_RATE.

4.12.4.2 Measurement Output

The headphone detection process typically comprises a number of separate measurements (for different impedance ranges). Completion of each measurement is indicated by HPD_DONE. When this bit is set, the measurement result can be read from the HPD_DACVAL field, and decoded as described in Eq. 4-3.

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

Equation 4-3. Headphone Impedance Calculation

The associated parameters for decoding the measurement result are defined Table 4-89. The applicable values are dependent on the HPD_IMPEDANCE_RANGE setting in each case. The *Offset* and *Gradient* values are derived from register fields that are factory-calibrated for each device.

Table 4-89. Headphone Measurement Decode Parameters

Parameter	HPD_IMPEDANCE_ RANGE = 00	HPD_IMPEDANCE_ RANGE = 01	HPD_IMPEDANCE_ RANGE = 10	HPD_IMPEDANCE_ RANGE = 11
C_0	1.007	1.007	9.744	100.684
C_1	–0.0072	–0.0072	–0.0795	–0.9494
C_2	4005	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	0.6	0.6	0.6	0.6
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 if 169 $\leq$ HPD_DACVAL $\leq$ 1019. (In case of any contradiction with the HPD_ IMPEDANCE_RANGE description, the HPD_DACVAL validity takes precedence.)

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

Each measurement is triggered by writing 1 to HPD_POLL. Completion of each measurement is indicated by HPD_DONE. Note that, after HPD_DONE bit has been asserted, it remains asserted until the next measurement has been commanded.

Note: A simpler, but less accurate, procedure for headphone impedance measurement is also supported, using the HPD_LVL field. When the HPD_DONE bit is set, indicating completion of a measurement, the impedance can be read directly from the HPD_LVL field, provided that the value lies within the range of the applicable HPD_ IMPEDANCE_RANGE setting.

Note that, for detection using the MICDET*n* or JACKDET*n* pins, the HPD_LVL field is the only supported measurement output option. The HPD_IMPEDANCE_RANGE field is not valid for detection on the MICDET*n* or JACKDET*n* pins. See Table 4-90 for further description of the HPD_LVL field.

The headphone detection function is an input to the interrupt control circuit and can be used to trigger an interrupt event on completion of the headphone detection; see Section 4.15.

The fields associated with headphone detection are described in Table 4-90. The external circuit configuration is shown Fig. 4-63.

Note that 32-bit register addressing is used from R12888 (0x3000) upwards; 16-bit format is used otherwise. The registers noted in Table 4-90 contain a mixture of 16- and 32-bit register addresses.

Table 4-90. Headphone Detect Control

Register Address	Bit	Label	Default	Description
R665 (0x0299) Headphone_ Detect_0	15	HPD_OVD_ENA	0	Headphone Detect Output Override Enable This bit, when set, causes the HPD_OUT_SEL headphone output channel to be automatically configured for headphone detection each time headphone detection is scheduled. Note that the respective output driver must also be disabled (HPnx_ENA = 0) for the duration of a headphone output impedance measurement. 0 = Disabled 1 = Enabled
	14:12	HPD_OUT_SEL[2:0]	000	Headphone Detect Output Channel Select 000 = HPOUT1L 001 = HPOUT1R 010 = HPOUT2L 011 = HPOUT2R 100 = HPOUT3L 101 = HPOUT3R All other codes are reserved
	11:8	HPD_FRC_SEL[3:0]	0000	Headphone Detect Measurement Current Pin Select 0000 = MICDET1 0001 = MICDET2 0010 = MICDET3 0011 = MICDET4 0100 = HPDET1 0101 = HPDET2 0110 = JACKDET1 0111 = JACKDET2 1000 = MICDET5 1001 = JACKDET3 All other codes are reserved
	7:4	HPD_SENSE_SEL[3:0]	0000	Headphone Detect Sense Pin Select 0000 = MICDET1 0001 = MICDET2 0010 = MICDET3 0011 = MICDET4 0100 = HPDET1 0101 = HPDET2 0110 = JACKDET1 0111 = JACKDET2 1000 = MICDET5 1001 = JACKDET3 All other codes are reserved
	2:0	HPD_GND_SEL[2:0]	000	Headphone Detect Ground Pin Select 000 = MICDET1/HPOUTFB1 001 = MICDET2/HPOUTFB2 010 = MICDET3/HPOUTFB3 011 = MICDET4/HPOUTFB4 100 = MICDET5/HPOUTFB5 All other codes are reserved

Table 4-90. Headphone Detect Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R667 (0x029B) Headphone_ Detect_1	10:9	HPD_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 HPD_SENSE_SEL = 0100 or 0101.
	4:3	HPD_CLK_DIV[1:0]	00	Headphone Detect Clock Rate (Selects the clocking rate of the headphone detect adjustable current source. Decreasing the clock rate gives a slower measurement time.) 00 = 32 kHz 01 = 16 kHz 10 = 8 kHz 11 = 4 kHz
	2:1	HPD_RATE[1:0]	00	Headphone Detect Sweep Rate (Selects the step size between successive measurements. Increasing the step size gives a faster measurement time.) 00 = 1 01 = 2 10 = 4 11 = Reserved
	0	HPD_POLL	0	Headphone Detect Enable Write 1 to start HP Detect function
R668 (0x029C) Headphone_ Detect_2	15	HPD_DONE	0	Headphone Detect Status 0 = HP Detect not complete 1 = HP Detect done
	14:0	HPD_LVL[14:0]	0x0000	Headphone Detect Level 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 HPDET1 or HPDET2 measurement (HPD_SENSE_SEL = 0100 or 0101), HPD_LVL is valid from 4 Ω to10 kΩ, within the range selected by HPD_IMPEDANCE_RANGE. For other measurements, HPD_LVL is valid from 400 Ω to 6 kΩ only. If HPD_LVL reports a value outside the valid range, the range should be adjusted and the measurement repeated. A 0-Ω result may be reported if the measurement is less than the minimum value for the selected range.
R669 (0x029D) Headphone_ Detect_3	9:0	HPD_DACVAL[9:0]	0x000	Headphone Detect Level (Coded as integer, LSB = 1). See separate description for full decode information.

Table 4-90. Headphone Detect Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R131076 (0x20004) OTP_HPDET_Cal_1	31:24	HP_OFFSET_11[7:0]	See Footnote 1	Headphone Detect Calibration field. Signed number, LSB = 0.25. Range is –31.75 to +31.75. Default value is factory-set per device.
	23:16	HP_OFFSET_10[7:0]	See Footnote 1	Headphone Detect Calibration field. Signed number, LSB = 0.25. Range is –31.75 to +31.75. Default value is factory-set per device.
	15:8	HP_OFFSET_01[7:0]	See Footnote 1	Headphone Detect Calibration field. Signed number, LSB = 0.25. Range is –31.75 to +31.75. Default value is factory-set per device.
	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.
R131078 (0x20006) OTP_HPDET_Cal_2	15:8	HP_GRADIENT_1X[7:0]	See Footnote 1	Headphone Detect Calibration field. Signed number, LSB = 0.25. Range is –31.75 to +31.75. Default value is factory-set per device.
	7:0	HP_GRADIENT_0X[7:0]	See Footnote 1	Headphone Detect Calibration field. Signed number, LSB = 0.25. Range is –31.75 to +31.75. Default value is factory-set per device.

1. Default value is factory-set per device.

The external connections for the headphone detect circuit are shown in Fig. 4-63.

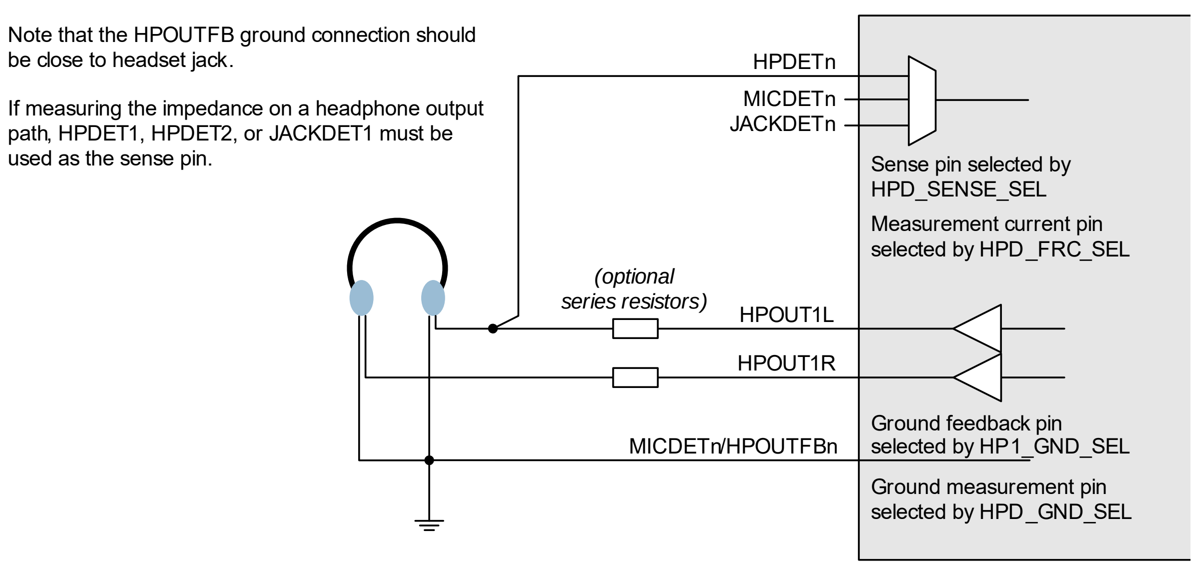


Figure 4-63. Headphone Detect Interface

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 CS42L92 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 CS42L92 supports a low-power Sleep Mode, in which most functions are disabled and power consumption is minimized. The CS42L92 enters Sleep Mode when the DCVDD supply is removed. Note that the AVDD and DBVDD supplies must be present throughout the Sleep Mode duration.

In Sleep Mode, the CS42L92 can generate an interrupt event in response to a change in voltage on the JACKDET1, JACKDET2, or JACKDET3 pins. This enables a jack insertion event (or other digital logic transition) to be used to trigger a wake-up of the CS42L92.

In Sleep Mode, the CS42L92 can provide an unregulated voltage output on the MICBIAS1A pin. This can be used to power an external microphone during Sleep Mode—see Section 4.19.

The system clocks (SYSCLK, ASYNCCLK, 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 4-91. 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 4-91. Sleep Mode Always-On Control Registers

Register Address	Label	Reference
R710 (0x02C6)	MICD_CLAMP2_OVD	See Section 4.12
	MICD_CLAMP2_MODE[2:0]	
	MICD_CLAMP1_OVD	
	MICD_CLAMP1_MODE[3:0]	
R723 (0x02D3)	MICB1A_AOD_ENA	See Section 4.19
	JD3_ENA	See Section 4.12
	JD2_ENA	
	JD1_ENA	
R6150 (0x1806)	MICD_CLAMP2_FALL_EINT1	See Section 4.15
	MICD_CLAMP2_RISE_EINT1	
	JD3_FALL_EINT1	
	JD3_RISE_EINT1	
	MICD_CLAMP1_FALL_EINT1	
	MICD_CLAMP1_RISE_EINT1	
	JD2_FALL_EINT1	
	JD2_RISE_EINT1	
	JD1_FALL_EINT1	
	JD1_RISE_EINT1	
R6214 (0x1846)	IM_MICD_CLAMP2_FALL_EINT1	
	IM_MICD_CLAMP2_RISE_EINT1	
	IM_JD3_FALL_EINT1	
	IM_JD3_RISE_EINT1	
	IM_MICD_CLAMP1_FALL_EINT1	
	IM_MICD_CLAMP1_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 4-92. All other digital input pins have no effect in Sleep Mode; all other digital output pins are undriven (floating).

Note: The IN*nx*/DMIC*x* connections are isolated from the CS42L92 circuits in Sleep Mode. This enables a microphone that is connected to the CS42L92 to be used by another circuit while Sleep Mode is selected.

The $\overline{IRQ}$ output is normally deasserted in Sleep Mode. In Sleep Mode, the $\overline{IRQ}$ output can be asserted only in response to the JACKDET1, JACKDET2, or JACKDET3 inputs. If the $\overline{IRQ}$ 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 4-92. Sleep Mode Always-On Digital Input/Output Pins

Pin Name	Description	Reference
$\overline{IRQ}$	Interrupt Request output	See Section 4.15
JACKDET1	Jack Detect input 1	See Section 4.12
JACKDET2	Jack Detect input 2	
JACKDET3	Jack Detect input 3	
$\overline{RESET}$	Digital Reset input (active low)	See Section 4.23

The always-on functionality includes the JD1, JD2, and JD3 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, JD2, or JD3.

The JD1, JD2, JD3, and MICDET clamp status signals are derived from the JACKDET1, JACKDET2, and JACKDET3 inputs, and can be used to trigger the interrupt controller.

- The JD1, JD2, and JD3 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.

  If these signals are enabled, the JD1, JD2, and JD3 signals indicate the status of the JACKDET1, JACKDET2, and JACKDET3 input pins respectively. See Table 4-84 for details of the associated control fields.
- The MICDET clamp status is controlled by the JD1, JD2, or JD3 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_CLAMP1_MODE and MICD_CLAMP2_MODE, as described in Table 4-85. Note that, due to control logic that is shared between the two clamps, the option to control both clamps in response to the JD*n* signals cannot be supported at the same time. It is assumed that a maximum of one clamp is active at any time—the MICDET clamp status provides an indication for the active clamp only.

The interrupt functionality associated with these signals is part of the always-on functionality, enabling the CS42L92 to provide indication of jack insertion or jack removal to the host processor in Sleep Mode; see Section 4.15.

Note that the JACKDET*n* 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 $\overline{IRQ}$ 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 CS42L92 supports up to 16 GPIO pins, which can be assigned to application-specific functions. The GPIOs enable interfacing and detection of external hardware and can provide logic outputs to other devices. The GPIO input functions can be used to generate an interrupt (IRQ) event.

There are 2 dedicated GPIO pins; the remaining 14 GPIOs are implemented as alternate functions to a pin-specific capability. The GPIO and interrupt circuits support the following functions:

- Pin-specific alternative functions for external interfaces (AIF, PDM)
- Logic input/button detect (GPIO input)
- Logic 1 and Logic 0 output (GPIO output)
- Interrupt (IRQ) status output
- Clock output
- Frequency-locked loop (FLL) status output
- FLL clock output
- IEC-60958-3–compatible S/PDIF output
- Pulse-width modulation (PWM) signal output
- ASRC lock status
- General-purpose timer status output
- Event logger FIFO buffer status output
- Alarm generator status output
- Auxiliary PDM interface

Logic input and output (GPIO) can be supported in two different ways on the CS42L92. 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 CS42L92 also incorporates two general-purpose switches; these are analog switches, described in Section 4.14.18.

If the JTAG interface is enabled, the GPIO13–15 pins are configured as a JTAG interface that provides test and debug access to the CS42L92. The respective GPIO configuration registers have no effect in this case, and the GPIO pins cannot be assigned any other function. See Section 4.20 for details of the JTAG interface.

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 CS42L92 clocking configuration.

Each GPIO pin 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 $\overline{\text{IRQ}}$ signal. See Section 4.15 for details of the interrupt event handling.

Integrated pull-up and pull-down resistors are provided on each GPIO pin; 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 CS42L92 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 4-93 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 4-93.

Table 4-93. GPIO Control

Register Address	Bit	Label	Default	Description
R5888 (0x1700) GPIO1_CTRL_1 to R5918 (0x171E) GPIO16_CTRL1	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 4-95). 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.
	9:0	GP*n*_FN[9:0]	0x001	GPIO*n* Pin Function (see Table 4-94 for details)

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

Register Address	Bit	Label	Default	Description
R5889 (0x1701) GPIO1_CTRL_2 to R5919 (0x171F) GPIO16_CTRL2	15	GP*n*_DIR	1	GPIO*n* Pin Direction 0 = Output 1 = Input Note that, if GP*n*_FN = 0x000 or 0x002, this bit has no effect on the GPIO*n* pin. If GPn_FN = 0x000, the pin direction is set according to the applicable pin-specific function (see Table 4-95). 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]	0x0	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 4-94. 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 4-94. GPIO Function Select

GP*n*_FN	Valid On	Description	Comments
0x000	GPIO3–16 only	Pin-specific alternate function	Alternate functions supporting digital microphone, digital audio interface, master control interface, and PDM output functions.
0x001	All GPIOs (1–16)	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	All GPIOs (1–16)	DSP GPIO	Low latency input/output for DSP functions.
0x003	All GPIOs (1–16)	IRQ1 output	Interrupt (IRQ1) output 0 = IRQ1 not asserted 1 = IRQ1 asserted
0x004	All GPIOs (1–16)	IRQ2 output	Interrupt (IRQ2) output 0 = IRQ2 not asserted 1 = IRQ2 asserted
0x010	GPIO1–4 only	FLL1 clock	Clock output from FLL1
0x011	GPIO1–4 only	FLL2 clock	Clock output from FLL2

Table 4-94. GPIO Function Select *(Cont.)*

GP*n*_FN	Valid On	Description	Comments
0x018	GPIO1–4 only	FLL1 lock	Indicates FLL1 lock status 0 = Not locked 1 = Locked
0x019	GPIO1–4 only	FLL2 lock	Indicates FLL2 lock status 0 = Not locked 1 = Locked
0x040	GPIO1–4 only	OPCLK clock output	Configurable clock output derived from SYSCLK
0x041	GPIO1–4 only	OPCLK async clock output	Configurable clock output derived from ASYNCCLK
0x048	All GPIOs (1–16)	PWM1 output	Configurable PWM output PWM1
0x049	All GPIOs (1–16)	PWM2 output	Configurable PWM output PWM2
0x04C	All GPIOs (1–16)	S/PDIF output	IEC-60958-3–compatible S/PDIF output
0x088	GPIO1–4 only	ASRC1 IN1 lock	Indicates ASRC1 IN1 Lock status (ASRC IN1 paths convert from the ASRC1_RATE1 sample rate to the ASRC1_RATE2 sample rate.) 0 = Not locked 1 = Locked
0x089	GPIO1–4 only	ASRC1 IN2 lock	Indicates ASRC1 IN2 Lock status (ASRC IN2 paths convert from the ASRC1_RATE2 sample rate to the ASRC1_RATE1 sample rate.) 0 = Not locked 1 = Locked
0x140	All GPIOs (1–16)	Timer 1 status	Timer 1 Status A pulse is output after the respective timer reaches its final count value.
0x150	GPIO1–4 only	Event Log 1 FIFO not-empty status	Event Log 1 FIFO Not-Empty status 0 = FIFO Empty 1 = FIFO Not Empty
0x230	All GPIOs (1–16)	Alarm 1 Channel 1 status	Alarm 1 Channel 1 Status A pulse is output when the respective alarm-trigger conditions are met. The pulse duration is configurable.
0x231	All GPIOs (1–16)	Alarm 1 Channel 2 status	Alarm 1 Channel 2 Status A pulse is output when the respective alarm-trigger conditions are met. The pulse duration is configurable.
0x232	All GPIOs (1–16)	Alarm 1 Channel 3 status	Alarm 1 Channel 3 Status A pulse is output when the respective alarm-trigger conditions are met. The pulse duration is configurable.
0x233	All GPIOs (1–16)	Alarm 1 Channel 4 status	Alarm 1 Channel 4 Status A pulse is output when the respective alarm-trigger conditions are met. The pulse duration is configurable.
0x280	GPIO3 or GPIO10	Auxiliary PDM clock input/output	Auxiliary PDM interface clock
0x281	GPIO4 or GPIO9	Auxiliary PDM data output	Auxiliary PDM interface data

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

The CS42L92 provides two dedicated GPIO pins (1–2). The remaining 14 GPIOs are multiplexed with the pin-specific functions listed in Table 4-95. 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 4-95. The respective GP*n*_DIR and GP*n*_OP_CFG bits have no effect in this case.

Table 4-95. GPIO Alternate Functions

Name	Condition	Description	Direction	Output Driver Configuration
AIF1BCLK/GPIO6	GP6_FN = 0x000	Audio Interface 1 bit clock	Digital I/O	CMOS
AIF1LRCLK/GPIO8	GP8_FN = 0x000	Audio Interface 1 left/right clock	Digital I/O	CMOS
AIF1RXDAT/GPIO7	GP7_FN = 0x000	Audio Interface 1 RX digital audio data	Digital input	—

Table 4-95. GPIO Alternate Functions *(Cont.)*

Name	Condition	Description	Direction	Output Driver Configuration
AIF1TXDAT/GPIO5	GP5_FN = 0x000	Audio Interface 1 TX digital audio data	Digital output	CMOS
AIF2BCLK/GPIO10	GP10_FN = 0x000	Audio Interface 2 bit clock	Digital I/O	CMOS
AIF2LRCLK/GPIO12	GP12_FN = 0x000	Audio Interface 2 left/right clock	Digital I/O	CMOS
AIF2RXDAT/GPIO11	GP11_FN = 0x000	Audio Interface 2 RX digital audio data	Digital input	—
AIF2TXDAT/GPIO9	GP9_FN = 0x000	Audio Interface 2 TX digital audio data	Digital output	CMOS
AIF3BCLK/GPIO14	GP14_FN = 0x000	Audio Interface 3 bit clock	Digital I/O	CMOS
AIF3LRCLK/GPIO16	GP16_FN = 0x000	Audio Interface 3 left/right clock	Digital I/O	CMOS
AIF3RXDAT/GPIO15	GP15_FN = 0x000	Audio Interface 3 RX digital audio data	Digital input	—
AIF3TXDAT/GPIO13	GP13_FN = 0x000	Audio Interface 3 TX digital audio data	Digital output	CMOS
SPKCLK/GPIO3	GP3_FN = 0x000	Digital speaker (PDM) clock	Digital output	CMOS
SPKDAT/GPIO4	GP4_FN = 0x000	Digital speaker (PDM) data	Digital output	CMOS

Note that if the JTAG interface is enabled, the GPIO13–15 pins are configured as a JTAG interface. Under these conditions, the respective GPIO configuration registers have no effect, and the GPIO pins cannot be assigned any other function. See Section 4.20 for details of the JTAG interface.

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 $\overline{IRQ}$ 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 CS42L92 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 for signal-processing applications. The DSP GPIO pins are accessed using maskable sets of I/O control registers; this allows the selected combinations of GPIOs to be controlled with ease, regardless of how the allocation of GPIO pins has been implemented in hardware.

The DSP GPIO function is selected by setting the respective GPIO fields as described in Section 4.14.1.

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 CS42L92 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 a 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, 0x011

Clock outputs derived from the FLLs may be output on a GPIO pin. The GPIO output from each FLL*n* (FLL1 or FLL2) is controlled by the respective FLL*n*_GPCLK_DIV and FLL*n*_GPCLK_ENA fields, as described in Table 4-96.

It is recommended to disable the clock output (FLL*n*_GPCLK_ENA = 0) before making any change to the respective FLL*n*_GPCLK_DIV field.

Note that FLL*n*_GPCLK_DIV and FLL*n*_GPCLK_ENA affect the GPIO outputs only; they do not affect the FLL frequency. The maximum output frequency supported for GPIO output is noted in Table 3-10.

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

See Section 4.16 for details of the CS42L92 system clocking and how to configure the FLLs.

Table 4-96. FLL Clock Output Control

Register Address	Bit	Label	Default	Description
R398 (0x018E) FLL1_GPIO_Clock	7:1	FLL1_GPCLK_ DIV[6:0]	0x02	FLL1 GPIO Clock Divider 0x00 = Reserved 0x01 = Reserved 0x02 = Divide by 2 0x03 = Divide by 3 0x04 = Divide by 4 … 0x7F = Divide by 127 (F_{GPIO} = F_{FLL}/FLL1_GPCLK_DIV)
	0	FLL1_GPCLK_ ENA	0	FLL1 GPIO Clock Enable 0 = Disabled 1 = Enabled
R430 (0x01AE) FLL2_GPIO_Clock	7:1	FLL2_GPCLK_ DIV[6:0]	0x02	FLL2 GPIO Clock Divider 0x00 = Reserved 0x01 = Reserved 0x02 = Divide by 2 0x03 = Divide by 3 0x04 = Divide by 4 … 0x7F = Divide by 127 (F_{GPIO} = F_{FLL}/FLL2_GPCLK_DIV)
	0	FLL2_GPCLK_ ENA	0	FLL2 GPIO Clock Enable 0 = Disabled 1 = Enabled

4.14.9 Frequency-Locked Loop (FLL) Status Output—GP*n*_FN = 0x018, 0x019

The CS42L92 provides FLL status flags, which may be used to control other events. The FLL lock signals indicate whether FLL lock has been achieved. See Section 4.16.8 for details of the FLLs.

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

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

4.14.10 OPCLK and OPCLK_ASYNC Clock Output—GP*n*_FN = 0x040, 0x041

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 4-97.

A clock output (OPCLK_ASYNC) derived from ASYNCCLK can be output on a GPIO pin. The OPCLK_ASYNC frequency is controlled by OPCLK_ASYNC_DIV and OPCLK_ASYNC_SEL. The OPCLK_ASYNC output is enabled by setting OPCLK_ASYNC_ENA.

It is recommended to disable the clock output (OPCLK_ENA = 0 or OPCLK_ASYNC_ENA = 0) before making any change to the respective OPCLK_DIV, OPCLK_SEL, OPCLK_ASYNC_DIV, or OPCLK_ASYNC_SEL fields.

The OPCLK or OPCLK_ASYNC 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 OPCLK_ASYNC source frequency cannot be higher than the ASYNCCLK frequency. The maximum output frequency supported for GPIO output is noted in Table 3-10.

See Section 4.16 for details of the system clocks (SYSCLK and ASYNCCLK).

Table 4-97. 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.

Table 4-97. OPCLK Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R330 (0x014A) Output_async_ lock	15	OPCLK_ASYNC_ ENA	0	OPCLK_ASYNC Enable 0 = Disabled 1 = Enabled
	7:3	OPCLK_ASYNC_ DIV[4:0]	0x00	OPCLK_ASYNC 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_ASYNC signal is enabled.
	2:0	OPCLK_ASYNC_ SEL[2:0]	000	OPCLK_ASYNC 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 ASYNCCLK rates only (i.e., ASYNC_SAMPLE_RATE_n = 01XXX). The OPCLK_ASYNC Source Frequency must be less than or equal to the ASYNCCLK frequency.

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

The CS42L92 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 CS42L92 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 ASRC Lock Status Output—GPn_FN = 0x088, 0x089

The CS42L92 provides ASRC status flags, which may be used to control other events. The ASRC-lock signals indicate whether ASRC lock has been achieved. See Section 4.3.15 for details of the ASRCs.

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

The ASRC lock signals are inputs to the interrupt control circuit. An interrupt event is triggered on the rising and falling edges of the ASRC lock signals. The associated interrupt bits are latched once set; they can be polled at any time or used to control the $\overline{\text{IRQ}}$ signal. See Section 4.15 for details of the interrupt event handling.

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

The general-purpose timer can count up or down, and supports continuous or single count modes. A status output, indicating the progress of the timer, is provided. See Section 4.5.3 for details of the general-purpose timer.

A logic signal from the general-purpose timer 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 timer reaches its final count value.

The general-purpose timer also provides input to the interrupt control circuit. An interrupt event is triggered whenever the timer reaches its final count value. The associated interrupt bit is latched once set; it can be polled at any time or used to control the $\overline{\text{IRQ}}$ 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

The event logger incorporates a 16-stage FIFO buffer, in which any detected events (signal transitions) are recorded. A status output for the FIFO buffer is provided. See Section 4.5.1 for details of the event logger.

A logic signal from the event logger 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 logger also provides input to the interrupt control circuit. An interrupt event is triggered whenever the FIFO condition occurs. The associated interrupt bit is latched once set; it can be polled at any time or used to control the $\overline{\text{IRQ}}$ signal. See Section 4.15 for details of the interrupt event handling.

4.14.16 Alarm Generator Status Output—GPn_FN = 0x230, 0x231, 0x232, 0x233

The CS42L92 incorporates four alarm-generator circuits that are associated with the general-purpose timer. A status output is provided by each alarm; these can be used to indicate one-off events, or can be configured for cyclic (repeated) triggers. See Section 4.5.2 for details of the alarm-control circuits.

The alarm status may be output directly on a GPIO pin by setting the respective GPIO fields as described in Section 4.14.1. The alarm status is asserted when the respective alarm-trigger conditions are met. The signal is asserted for a duration that is configurable as described in Section 4.5.2.1.

The alarm generators also provide input to the interrupt control circuit. An interrupt event is triggered whenever the alarm-trigger conditions are met. The associated interrupt bit is latched once set; it can be polled at any time or used to control the $\overline{\text{IRQ}}$ signal. See Section 4.15 for details of the interrupt event handling.

4.14.17 Auxiliary PDM Interface—GP*n*_FN = 0x280, 0x281

The CS42L92 provides an auxiliary PDM interface that can be used to provide an audio path between an analog microphone connected to the CS42L92 and a digital input to an external audio processor. The external connections to the auxiliary PDM interface are supported on GPIO pins as follows:

- In Master Mode (AUXPDM1_MSTR = 1), the CLK output can be configured on the GPIO3 or GPIO10 pins. The DAT output can be configured on GPIO4 or GPIO9.
- In Slave Mode (AUXPDM1_MSTR = 0), the CLK input is supported on GPIO10 only. The DAT output is supported on GPIO9 only.

The applicable GPIO pins are configured by setting the respective GPIO fields as described in Section 4.14.1.

See Section 4.2.10 for details of how to configure the auxiliary PDM interface.

4.14.18 General-Purpose Switch

The CS42L92 provides two general-purpose switches, which can be used as controllable analog switches for external functions. The switches support bidirectional analog operation, offering flexibility in the potential circuit applications. Refer to Table 3-2 and Table 3-10 for further details. Note that this feature is entirely independent of the GPIO*n* pins.

- The GP1 switch is implemented between the GPSW1P and GPSW1N pins; it is configured using SW1_MODE.
- The GP2 switch is implemented between the GPSW2P and GPSW2N pins; it is configured using SW2_MODE.

The SW*n*_MODE fields allow the switches to be disabled, enabled, or synchronized to the MICDET clamp status, as described in Table 4-98.

The switches 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. 4-60 within Section 4.12.2. Note that the MICDET clamp function must also be configured appropriately when using this method of pop suppression.

Table 4-98. General-Purpose Switch Control

Register Address	Bit	Label	Default	Description
R712 (0x02C8) GP_Switch_1	3:2	SW2_MODE[1:0]	00	General-purpose Switch 2 control 00 = Disabled (switch open) 01 = Enabled (switch closed) 10 = Enabled if MICDET clamp status is set 11 = Enabled if MICDET clamp status is clear
	1:0	SW1_MODE[1:0]	00	General-purpose Switch 1 control 00 = Disabled (switch open) 01 = Enabled (switch closed) 10 = Enabled if MICDET clamp status is set 11 = Enabled if MICDET clamp status is clear

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/ASRC lock detection, and status flags from DSP peripheral functions. See Table 4-99 and Table 4-100 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 interrupt input. 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 4-99. The interrupt register fields for IRQ2 are described in Table 4-100. 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 4-99 and Table 4-100 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 4-93 and Table 4-41.

Individual mask bits can enable or disable different functions from the interrupt controller. The mask bits are described in Table 4-99 (for IRQ1) and Table 4-100 (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 4-93. 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—either the 32-kHz clock or the SYSCLK signal must be enabled to trigger an interrupt from the FLL status inputs. Note that the raw status fields (described in Table 4-99 and Table 4-100) are valid without clocking; these fields can be used to provide FLL status readback if 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 active low and is referenced to the DBVDD 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 CS42L92 interrupt controller circuit is shown in Fig. 4-64. (Note that not all interrupt inputs are shown.) The control fields associated with IRQ1 and IRQ2 are described in Table 4-99 and Table 4-100 respectively. The global interrupt mask bits, status bits, and output configuration fields are described Table 4-101.

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.

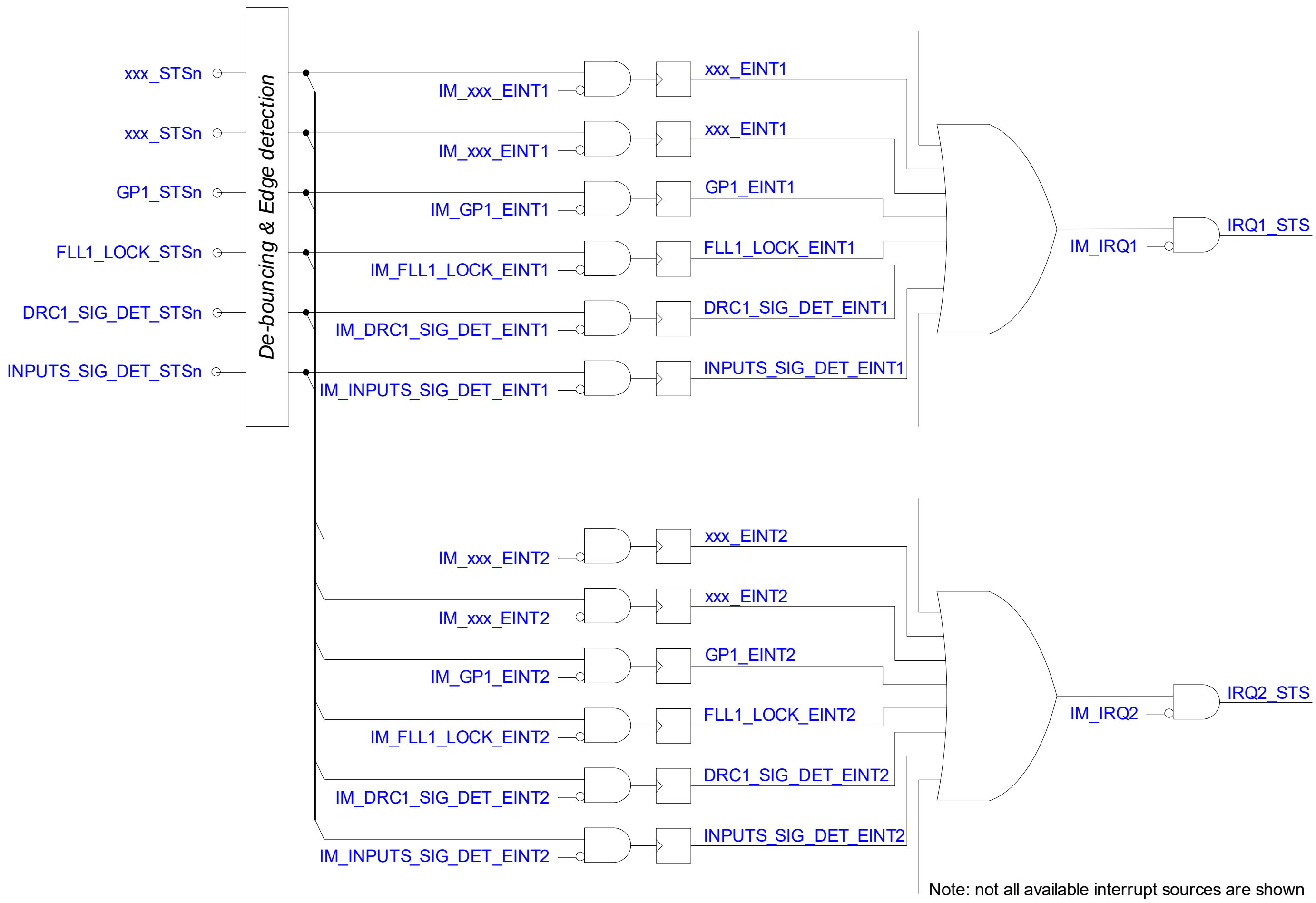


Figure 4-64. Interrupt Controller

The IRQ1 interrupt, mask, and status control registers are described in Table 4-99.

Table 4-99. Interrupt 1 Control Registers

Register Address	Bit	Label	Default	Description
R6144 (0x1800) IRQ1_Status_1	12	CTRLIF_ERR_EINT1	0	Control Interface Error Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	9	SYSCLK_FAIL_EINT1	0	SYSCLK Fail Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	7	BOOT_DONE_EINT1	0	Boot Done Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6145 (0x1801) IRQ1_Status_2	14	DSPCLK_ERR_EINT1	0	DSPCLK Error Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	13	ASYNCCLK_ERR_EINT1	0	ASYNCCLK Error Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	12	SYSCLK_ERR_EINT1	0	SYSCLK Error Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	9	FLL2_LOCK_EINT1	0	FLL2 Lock Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	8	FLL1_LOCK_EINT1	0	FLL1 Lock Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	6	FLL2_REF_LOST_EINT1	0	FLL2 Reference Lost Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	5	FLL1_REF_LOST_EINT1	0	FLL1 Reference Lost Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6149 (0x1805) IRQ1_Status_6	9	MICDET2_EINT1	0	Mic/Accessory Detect 2 Interrupt (Detection event triggered) Note: Cleared when a 1 is written.
	8	MICDET1_EINT1	0	Mic/Accessory Detect 1 Interrupt (Detection event triggered) Note: Cleared when a 1 is written.
	0	HPDET_EINT1	0	Headphone Detect Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6150 (0x1806) IRQ1_Status_7	11	MICD_CLAMP2_FALL_EINT1	0	MICDET Clamp 2 Interrupt (Falling edge triggered) Note: Cleared when a 1 is written.
	10	MICD_CLAMP2_RISE_EINT1	0	MICDET Clamp 2 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	9	JD3_FALL_EINT1	0	JD3 Interrupt (Falling edge triggered) Note: Cleared when a 1 is written.
	8	JD3_RISE_EINT1	0	JD3 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	5	MICD_CLAMP1_FALL_EINT1	0	MICDET Clamp 1 Interrupt (Falling edge triggered) Note: Cleared when a 1 is written.
	4	MICD_CLAMP1_RISE_EINT1	0	MICDET Clamp 1 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.

Table 4-99. Interrupt 1 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6152 (0x1808) IRQ1_Status_9	9	ASRC1_IN2_LOCK_EINT1	0	ASRC1 IN2 Lock Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	8	ASRC1_IN1_LOCK_EINT1	0	ASRC1 IN1 Lock Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	2	INPUTS_SIG_DET_EINT1	0	Input Path Signal-Detect Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	1	DRC2_SIG_DET_EINT1	0	DRC2 Signal-Detect Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	0	DRC1_SIG_DET_EINT1	0	DRC1 Signal-Detect Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
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.

Table 4-99. Interrupt 1 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6155 (0x180B) IRQ1_Status_12	9	HP4R_SC_EINT1	0	HPOUT4R Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	8	HP4L_SC_EINT1	0	HPOUT4L Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	5	HP3R_SC_EINT1	0	HPOUT3R Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	4	HP3L_SC_EINT1	0	HPOUT3L Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	3	HP2R_SC_EINT1	0	HPOUT2R Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	2	HP2L_SC_EINT1	0	HPOUT2L Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	HP1R_SC_EINT1	0	HPOUT1R Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	HP1L_SC_EINT1	0	HPOUT1L Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6156 (0x180C) IRQ1_Status_13	5	HP3R_ENABLE_DONE_EINT1	0	HPOUT3R/HPOUT4R Enable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	4	HP3L_ENABLE_DONE_EINT1	0	HPOUT3L/HPOUT4L Enable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	3	HP2R_ENABLE_DONE_EINT1	0	HPOUT2R Enable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	2	HP2L_ENABLE_DONE_EINT1	0	HPOUT2L Enable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	HP1R_ENABLE_DONE_EINT1	0	HPOUT1R Enable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	HP1L_ENABLE_DONE_EINT1	0	HPOUT1L Enable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6157 (0x180D) IRQ1_Status_14	5	HP3R_DISABLE_DONE_EINT1	0	HPOUT3R/HPOUT4R Disable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	4	HP3L_DISABLE_DONE_EINT1	0	HPOUT3L/HPOUT4L Disable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	3	HP2R_DISABLE_DONE_EINT1	0	HPOUT2R Disable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	2	HP2L_DISABLE_DONE_EINT1	0	HPOUT2L Disable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	HP1R_DISABLE_DONE_EINT1	0	HPOUT1R Disable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	HP1L_DISABLE_DONE_EINT1	0	HPOUT1L Disable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6158 (0x180E) IRQ1_Status_15	12	DFC_SATURATE_EINT1	0	DFC Saturate Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.

Table 4-99. Interrupt 1 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
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	0	TIMER1_EINT1	0	Timer 1 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6165 (0x1815) IRQ1_Status_22	0	EVENT1_NOT_EMPTY_EINT1	0	Event Log 1 FIFO Not Empty Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6166 (0x1816) IRQ1_Status_23	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	0	EVENT1_WMARK_EINT1	0	Event Log 1 FIFO Watermark Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6168 (0x1818) IRQ1_Status_25	0	DSP1_DMA_EINT1	0	DSP1 DMA Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6170 (0x181A) IRQ1_Status_27	0	DSP1_START1_EINT1	0	DSP1 Start 1 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6171 (0x181B) IRQ1_Status_28	0	DSP1_START2_EINT1	0	DSP1 Start 2 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6173 (0x181D) IRQ1_Status_30	0	DSP1_BUSY_EINT1	0	DSP1 Busy Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6176 (0x1820) IRQ1_Status_33	0	DSP1_BUS_ERR_EINT1	0	DSP1 Bus Error Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.

Table 4-99. Interrupt 1 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6179 (0x1823) IRQ1_Status_36	3	TIMER_ALM1_CH4_EINT1	0	Alarm 1 Channel 4 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	2	TIMER_ALM1_CH3_EINT1	0	Alarm 1 Channel 3 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	TIMER_ALM1_CH2_EINT1	0	Alarm 1 Channel 2 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	TIMER_ALM1_CH1_EINT1	0	Alarm 1 Channel 1 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6208 (0x1840) to R6243 (0x1863)		IM_*	See Footnote [1]	For each x_EINT1 interrupt bit in R6144 to R6179, a corresponding mask bit (IM_*) is provided in R6208 to R6243. The mask bits are coded as follows: 0 = Do not mask interrupt 1 = Mask interrupt
R6272 (0x1880) IRQ1_Raw_Status_1	12	CTRLIF_ERR_STS1	0	Control Interface Error Status 0 = Normal 1 = Control Interface Error
	7	BOOT_DONE_STS1	0	Boot Status 0 = Busy (boot sequence in progress) 1 = Idle (boot sequence completed) Control register writes should not be attempted until Boot Sequence has completed.
R6273 (0x1881) IRQ1_Raw_Status_2	14	DSPCLK_ERR_STS1	0	DSPCLK Error Interrupt Status 0 = Normal 1 = Insufficient DSPCLK cycles for one or more of the requested DSP1 clock frequencies
	13	ASYNCCLK_ERR_STS1	0	ASYNCCLK Error Interrupt Status 0 = Normal 1 = Insufficient ASYNCCLK cycles for the requested signal path functionality
	12	SYSCLK_ERR_STS1	0	SYSCLK Error Interrupt Status 0 = Normal 1 = Insufficient SYSCLK cycles for the requested signal path functionality
	9	FLL2_LOCK_STS1	0	FLL2 Lock Status 0 = Not locked 1 = Locked
	8	FLL1_LOCK_STS1	0	FLL1 Lock Status 0 = Not locked 1 = Locked
	6	FLL2_REF_LOST_STS1	0	FLL2 Reference Lost Status 0 = Normal 1 = Reference Lost
	5	FLL1_REF_LOST_STS1	0	FLL1 Reference Lost Status 0 = Normal 1 = Reference Lost

Table 4-99. Interrupt 1 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6278 (0x1886) IRQ1_Raw_ Status_7	8	JD3_STS1	0	JACKDET3 input status 0 = Jack not detected 1 = Jack is detected (Assumes the JACKDET3 pin is pulled low on jack insertion.)
	4	MICD_CLAMP_STS1	0	MICDET Clamp status 0 = Clamp disabled 1 = Clamp enabled Separate _STS bits are not provided for each clamp—it is assumed that a maximum of one clamp is active at any time. The clamp override condition (MICD_CLAMP*n*_OVD = 1) is not indicated.
	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	9	ASRC1_IN2_LOCK_STS1	0	ASRC1 IN2 Lock Status 0 = Not locked 1 = Locked
	8	ASRC1_IN1_LOCK_STS1	0	ASRC1 IN1 Lock Status 0 = Not locked 1 = Locked
	2	INPUTS_SIG_DET_STS1	0	Input Path Signal-Detect Status 0 = Normal 1 = Signal detected
	1	DRC2_SIG_DET_STS1	0	DRC2 Signal-Detect Status 0 = Normal 1 = Signal detected
	0	DRC1_SIG_DET_STS1	0	DRC1 Signal-Detect Status 0 = Normal 1 = Signal detected

Table 4-99. Interrupt 1 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6283 (0x188B) IRQ1_Raw_ Status_12	9	HP4R_SC_STS1	0	HPOUT4R Short Circuit Status 0 = Normal 1 = Short Circuit detected
	8	HP4L_SC_STS1	0	HPOUT4L Short Circuit Status 0 = Normal 1 = Short Circuit detected
	5	HP3R_SC_STS1	0	HPOUT3R Short Circuit Status 0 = Normal 1 = Short Circuit detected
	4	HP3L_SC_STS1	0	HPOUT3L Short Circuit Status 0 = Normal 1 = Short Circuit detected
	3	HP2R_SC_STS1	0	HPOUT2R Short Circuit Status 0 = Normal 1 = Short Circuit detected
	2	HP2L_SC_STS1	0	HPOUT2L Short Circuit Status 0 = Normal 1 = Short Circuit detected
	1	HP1R_SC_STS1	0	HPOUT1R Short Circuit Status 0 = Normal 1 = Short Circuit detected
	0	HP1L_SC_STS1	0	HPOUT1L Short Circuit Status 0 = Normal 1 = Short Circuit detected
R6284 (0x188C) IRQ1_Raw_ Status_13	5	HP3R_ENABLE_DONE_STS1	0	HPOUT3R/HPOUT4R Enable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	4	HP3L_ENABLE_DONE_STS1	0	HPOUT3L/HPOUT4L Enable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	3	HP2R_ENABLE_DONE_STS1	0	HPOUT2R Enable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	2	HP2L_ENABLE_DONE_STS1	0	HPOUT2L Enable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	1	HP1R_ENABLE_DONE_STS1	0	HPOUT1R Enable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	0	HP1L_ENABLE_DONE_STS1	0	HPOUT1L Enable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)

Table 4-99. Interrupt 1 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6285 (0x188D) IRQ1_Raw_ Status_14	5	HP3R_DISABLE_DONE_STS1	0	HPOUT3R/HPOUT4R Disable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	4	HP3L_DISABLE_DONE_STS1	0	HPOUT3L/HPOUT4L Disable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	3	HP2R_DISABLE_DONE_STS1	0	HPOUT2R Disable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	2	HP2L_DISABLE_DONE_STS1	0	HPOUT2L Disable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	1	HP1R_DISABLE_DONE_STS1	0	HPOUT1R Disable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	0	HP1L_DISABLE_DONE_STS1	0	HPOUT1L Disable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
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	0	EVENT1_NOT_EMPTY_STS1	0	Event Log 1 FIFO Not-Empty status 0 = FIFO Empty 1 = FIFO Not Empty
R6294 (0x1896) IRQ1_Raw_ Status_23	0	EVENT1_FULL_STS1	0	Event Log 1 FIFO Full status 0 = FIFO Not Full 1 = FIFO Full
R6295 (0x1897) IRQ1_Raw_ Status_24	0	EVENT1_WMARK_STS1	0	Event Log 1 FIFO Watermark status 0 = FIFO Watermark not reached 1 = FIFO Watermark reached
R6296 (0x1898) IRQ1_Raw_ Status_25	0	DSP1_DMA_STS1	0	DSP1 DMA status 0 = Normal 1 = All enabled WDMA buffers filled, and all enabled RDMA buffers emptied
R6301 (0x189D) IRQ1_Raw_ Status_30	0	DSP1_BUSY_STS1	0	DSP1 Busy status 0 = DSP Idle 1 = DSP Busy

Table 4-99. Interrupt 1 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6307 (0x18A3) IRQ1_Raw_ Status_36	3	TIMER_ALM1_CH4_STS1	0	Alarm 1 Channel 4 status 0 = Alarm idle 1 = Alarm output asserted
	2	TIMER_ALM1_CH3_STS1	0	Alarm 1 Channel 3 status 0 = Alarm idle 1 = Alarm output asserted
	1	TIMER_ALM1_CH2_STS1	0	Alarm 1 Channel 2 status 0 = Alarm idle 1 = Alarm output asserted
	0	TIMER_ALM1_CH1_STS1	0	Alarm 1 Channel 1 status 0 = Alarm idle 1 = Alarm output asserted

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 4-100.

Table 4-100. Interrupt 2 Control Registers

Register Address	Bit	Label	Default	Description
R6400 (0x1900) IRQ2_Status_1	12	CTRLIF_ERR_EINT2	0	Control Interface Error Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	9	SYSCLK_FAIL_EINT2	0	SYSCLK Fail Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	7	BOOT_DONE_EINT2	0	Boot Done Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6401 (0x1901) IRQ2_Status_2	14	DSPCLK_ERR_EINT2	0	DSPCLK Error Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	13	ASYNCCLK_ERR_EINT2	0	ASYNCCLK Error Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	12	SYSCLK_ERR_EINT2	0	SYSCLK Error Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	9	FLL2_LOCK_EINT2	0	FLL2 Lock Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	8	FLL1_LOCK_EINT2	0	FLL1 Lock Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	6	FLL2_REF_LOST_EINT2	0	FLL2 Reference Lost Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	5	FLL1_REF_LOST_EINT2	0	FLL1 Reference Lost Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6405 (0x1905) IRQ2_Status_6	9	MICDET2_EINT2	0	Mic/Accessory Detect 2 Interrupt (Detection event triggered) Note: Cleared when a 1 is written.
	8	MICDET1_EINT2	0	Mic/Accessory Detect 1 Interrupt (Detection event triggered) Note: Cleared when a 1 is written.
	0	HPDET_EINT2	0	Headphone Detect Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.

Table 4-100. Interrupt 2 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6406 (0x1906) IRQ2_Status_7	3	JD3_FALL_EINT2	0	JD3 Interrupt (Falling edge triggered) Note: Cleared when a 1 is written.
	2	JD3_RISE_EINT2	0	JD3 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	5	MICD_CLAMP_FALL_EINT2	0	MICDET Clamp Interrupt (Falling edge triggered) Indicates a falling edge transition on MICDET Clamp 1 or MICDET Clamp 2. Separate x_EINT2 bits are not provided—it is assumed that a maximum of one clamp is active at any time. Note: Cleared when a 1 is written.
	4	MICD_CLAMP_RISE_EINT2	0	MICDET Clamp Interrupt (Rising edge triggered) Indicates a rising edge transition on MICDET Clamp 1 or MICDET Clamp 2. Separate x_EINT2 bits are not provided—it is assumed that a maximum of one clamp is active at any time. 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	9	ASRC1_IN2_LOCK_EINT2	0	ASRC1 IN2 Lock Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	8	ASRC1_IN1_LOCK_EINT2	0	ASRC1 IN1 Lock Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	2	INPUTS_SIG_DET_EINT2	0	Input Path Signal-Detect Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	1	DRC2_SIG_DET_EINT2	0	DRC2 Signal-Detect Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.
	0	DRC1_SIG_DET_EINT2	0	DRC1 Signal-Detect Interrupt (Rising and falling edge triggered) Note: Cleared when a 1 is written.

Table 4-100. 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	9	HP4R_SC_EINT2	0	HPOUT4R Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	8	HP4L_SC_EINT2	0	HPOUT4L Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	5	HP3R_SC_EINT2	0	HPOUT3R Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	4	HP3L_SC_EINT2	0	HPOUT3L Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	3	HP2R_SC_EINT2	0	HPOUT2R Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	2	HP2L_SC_EINT2	0	HPOUT2L Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	HP1R_SC_EINT2	0	HPOUT1R Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	HP1L_SC_EINT2	0	HPOUT1L Short Circuit Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.

Table 4-100. Interrupt 2 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6412 (0x190C) IRQ2_Status_13	5	HP3R_ENABLE_DONE_EINT2	0	HPOUT3R/HPOUT4R Enable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	4	HP3L_ENABLE_DONE_EINT2	0	HPOUT3L/HPOUT4L Enable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	3	HP2R_ENABLE_DONE_EINT2	0	HPOUT2R Enable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	2	HP2L_ENABLE_DONE_EINT2	0	HPOUT2L Enable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	HP1R_ENABLE_DONE_EINT2	0	HPOUT1R Enable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	HP1L_ENABLE_DONE_EINT2	0	HPOUT1L Enable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6413 (0x190D) IRQ2_Status_14	5	HP3R_DISABLE_DONE_EINT2	0	HPOUT3R/HPOUT4R Disable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	4	HP3L_DISABLE_DONE_EINT2	0	HPOUT3L/HPOUT4L Disable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	3	HP2R_DISABLE_DONE_EINT2	0	HPOUT2R Disable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	2	HP2L_DISABLE_DONE_EINT2	0	HPOUT2L Disable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	HP1R_DISABLE_DONE_EINT2	0	HPOUT1R Disable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	HP1L_DISABLE_DONE_EINT2	0	HPOUT1L Disable Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6414 (0x190E) IRQ2_Status_15	12	DFC_SATURATE_EINT2	0	DFC Saturate Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.

Table 4-100. Interrupt 2 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
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	0	TIMER1_EINT2	0	Timer 1 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6421 (0x1915) IRQ2_Status_22	0	EVENT1_NOT_EMPTY_EINT2	0	Event Log 1 FIFO Not Empty Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6422 (0x1916) IRQ2_Status_23	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	0	EVENT1_WMARK_EINT2	0	Event Log 1 FIFO Watermark Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6424 (0x1918) IRQ2_Status_25	0	DSP1_DMA_EINT2	0	DSP1 DMA Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6426 (0x191A) IRQ2_Status_27	0	DSP1_START1_EINT2	0	DSP1 Start 1 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6427 (0x191B) IRQ2_Status_28	0	DSP1_START2_EINT2	0	DSP1 Start 2 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6429 (0x191D) IRQ2_Status_30	0	DSP1_BUSY_EINT2	0	DSP1 Busy Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6432 (0x1920) IRQ2_Status_33	0	DSP1_BUS_ERR_EINT2	0	DSP1 Bus Error Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.

Table 4-100. Interrupt 2 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6435 (0x1923) IRQ2_Status_36	3	TIMER_ALM1_CH4_EINT2	0	Alarm 1 Channel 4 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	2	TIMER_ALM1_CH3_EINT2	0	Alarm 1 Channel 3 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	1	TIMER_ALM1_CH2_EINT2	0	Alarm 1 Channel 2 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
	0	TIMER_ALM1_CH1_EINT2	0	Alarm 1 Channel 1 Interrupt (Rising edge triggered) Note: Cleared when a 1 is written.
R6464 (0x1940) to R6499 (0x1963)		IM_*	1	For each x_EINT2 interrupt bit in R6400 to R6435, a corresponding mask bit (IM_*) is provided in R6464 to R6499. The mask bits are coded as follows: 0 = Do not mask interrupt 1 = Mask interrupt
R6528 (0x1980) IRQ2_Raw_Status_1	12	CTRLIF_ERR_STS2	0	Control Interface Error Status 0 = Normal 1 = Control Interface Error
	7	BOOT_DONE_STS2	0	Boot Status 0 = Busy (boot sequence in progress) 1 = Idle (boot sequence completed) Control register writes should not be attempted until Boot Sequence has completed.
R6529 (0x1981) IRQ2_Raw_Status_2	14	DSPCLK_ERR_STS2	0	DSPCLK Error Interrupt Status 0 = Normal 1 = Insufficient DSPCLK cycles for one or more of the requested DSP1 clock frequencies
	13	ASYNCCLK_ERR_STS2	0	ASYNCCLK Error Interrupt Status 0 = Normal 1 = Insufficient ASYNCCLK cycles for the requested signal path functionality
	12	SYSCLK_ERR_STS2	0	SYSCLK Error Interrupt Status 0 = Normal 1 = Insufficient SYSCLK cycles for the requested signal path functionality
	9	FLL2_LOCK_STS2	0	FLL2 Lock Status 0 = Not locked 1 = Locked
	8	FLL1_LOCK_STS2	0	FLL1 Lock Status 0 = Not locked 1 = Locked
	6	FLL2_REF_LOST_STS2	0	FLL2 Reference Lost Status 0 = Normal 1 = Reference Lost
	5	FLL1_REF_LOST_STS2	0	FLL1 Reference Lost Status 0 = Normal 1 = Reference Lost

Table 4-100. Interrupt 2 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6534 (0x1986) IRQ2_Raw_ Status_7	8	JD3_STS2	0	JACKDET3 input status 0 = Jack not detected 1 = Jack is detected (Assumes the JACKDET3 pin is pulled low on jack insertion.)
	4	MICD_CLAMP_STS2	0	MICDET Clamp status 0 = Clamp disabled 1 = Clamp enabled 0 = Clamp disabled 1 = Clamp enabled Separate _STS bits are not provided for each clamp—it is assumed that a maximum of one clamp is active at any time. The clamp override condition (MICD_CLAMP*n*_OVD = 1) is not indicated.
	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	9	ASRC1_IN2_LOCK_STS2	0	ASRC1 IN2 Lock Status 0 = Not locked 1 = Locked
	8	ASRC1_IN1_LOCK_STS2	0	ASRC1 IN1 Lock Status 0 = Not locked 1 = Locked
	2	INPUTS_SIG_DET_STS2	0	Input Path Signal-Detect Status 0 = Normal 1 = Signal detected
	1	DRC2_SIG_DET_STS2	0	DRC2 Signal-Detect Status 0 = Normal 1 = Signal detected
	0	DRC1_SIG_DET_STS2	0	DRC1 Signal-Detect Status 0 = Normal 1 = Signal detected

Table 4-100. Interrupt 2 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6539 (0x198B) IRQ2_Raw_ Status_12	9	HP4R_SC_STS2	0	HPOUT4R Short Circuit Status 0 = Normal 1 = Short Circuit detected
	8	HP4L_SC_STS2	0	HPOUT4L Short Circuit Status 0 = Normal 1 = Short Circuit detected
	5	HP3R_SC_STS2	0	HPOUT3R Short Circuit Status 0 = Normal 1 = Short Circuit detected
	4	HP3L_SC_STS2	0	HPOUT3L Short Circuit Status 0 = Normal 1 = Short Circuit detected
	3	HP2R_SC_STS2	0	HPOUT2R Short Circuit Status 0 = Normal 1 = Short Circuit detected
	2	HP2L_SC_STS2	0	HPOUT2L Short Circuit Status 0 = Normal 1 = Short Circuit detected
	1	HP1R_SC_STS2	0	HPOUT1R Short Circuit Status 0 = Normal 1 = Short Circuit detected
	0	HP1L_SC_STS2	0	HPOUT1L Short Circuit Status 0 = Normal 1 = Short Circuit detected
R6540 (0x198C) IRQ2_Raw_ Status_13	5	HP3R_ENABLE_DONE_STS2	0	HPOUT3R/HPOUT4R Enable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	4	HP3L_ENABLE_DONE_STS2	0	HPOUT3L/HPOUT4L Enable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	3	HP2R_ENABLE_DONE_STS2	0	HPOUT2R Enable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	2	HP2L_ENABLE_DONE_STS2	0	HPOUT2L Enable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	1	HP1R_ENABLE_DONE_STS2	0	HPOUT1R Enable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	0	HP1L_ENABLE_DONE_STS2	0	HPOUT1L Enable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)

Table 4-100. Interrupt 2 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6541 (0x198D) IRQ2_Raw_ Status_14	5	HP3R_DISABLE_DONE_STS2	0	HPOUT3R/HPOUT4R Disable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	4	HP3L_DISABLE_DONE_STS2	0	HPOUT3L/HPOUT4L Disable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	3	HP2R_DISABLE_DONE_STS2	0	HPOUT2R Disable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	2	HP2L_DISABLE_DONE_STS2	0	HPOUT2L Disable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	1	HP1R_DISABLE_DONE_STS2	0	HPOUT1R Disable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
	0	HP1L_DISABLE_DONE_STS2	0	HPOUT1L Disable Status 0 = Busy (sequence in progress) 1 = Idle (sequence completed)
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	0	EVENT1_NOT_EMPTY_STS2	0	Event Log 1 FIFO Not-Empty status 0 = FIFO Empty 1 = FIFO Not Empty
R6550 (0x1996) IRQ2_Raw_ Status_23	0	EVENT1_FULL_STS2	0	Event Log *n* FIFO Full status 0 = FIFO Not Full 1 = FIFO Full
R6551 (0x1997) IRQ2_Raw_ Status_24	0	EVENT1_WMARK_STS2	0	Event Log 1 FIFO Watermark status 0 = FIFO Watermark not reached 1 = FIFO Watermark reached
R6552 (0x1998) IRQ2_Raw_ Status_25	0	DSP1_DMA_STS2	0	DSP1 DMA status 0 = Normal 1 = All enabled WDMA buffers filled, and all enabled RDMA buffers emptied
R6557 (0x199D) IRQ2_Raw_ Status_30	0	DSP1_BUSY_STS2	0	DSP1 Busy status 0 = DSP Idle 1 = DSP Busy

Table 4-100. Interrupt 2 Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R6563 (0x19A3) IRQ2_Raw_ Status_36	3	TIMER_ALM1_CH4_STS2	0	Alarm 1 Channel 4 status 0 = Alarm idle 1 = Alarm output asserted
	2	TIMER_ALM1_CH3_STS2	0	Alarm 1 Channel 3 status 0 = Alarm idle 1 = Alarm output asserted
	1	TIMER_ALM1_CH2_STS2	0	Alarm 1 Channel 2 status 0 = Alarm idle 1 = Alarm output asserted
	0	TIMER_ALM1_CH1_STS2	0	Alarm 1 Channel 1 status 0 = Alarm idle 1 = Alarm output asserted

The IRQ output and polarity control registers are described in Table 4-101.

Table 4-101. 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 CS42L92 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 CS42L92 incorporates two FLL circuits to perform frequency conversion and filtering.

External clock signals may be connected via the MCLK1, MCLK2, and MCLK3 input pins. 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 one of the FLLs. To avoid audible glitches, all clock configurations must be set up before enabling playback.

4.16.1 System Clocking Overview

The CS42L92 supports three primary clock domains—SYSCLK, ASYNCCLK, and DSPCLK.

The SYSCLK and ASYNCCLK clock domains are the reference clocks for all the audio signal paths on the CS42L92. Up to five different sample rates may be independently selected for specific audio interfaces and other input/output signal paths; each selected sample rate must be synchronized either to SYSCLK or to ASYNCCLK, as described in Section 4.16.2.

The SYSCLK and ASYNCCLK clock domains are independent (i.e., not synchronized). Stereo full-duplex sample-rate conversion is supported, allowing asynchronous audio data to be mixed and to be routed between independent interfaces. See Section 4.3 for further details.

The DSPCLK clock domain is the reference clock for the programmable DSP core on the CS42L92. A wide range of DSPCLK frequencies can be supported, and a programmable clock divider is provided for the DSP core, allowing the DSP clocking (and power consumption) to be optimized according to the applicable processing requirements. See Section 4.3 for further details.

Note that there is no requirement for DSPCLK to be synchronized to SYSCLK or ASYNCCLK. The DSPCLK controls the software execution in the DSP core; audio outputs from the DSP are synchronized either to SYSCLK or ASYNCCLK, regardless of the applicable DSPCLK rate.

Excluding the DSP core, each subsystem within the CS42L92 digital core is clocked at a dynamically controlled rate, limited by the SYSCLK (or ASYNCCLK) frequency, as applicable. For maximum signal mixing and processing capacity, it is recommended that the highest possible SYSCLK and ASYNCCLK frequencies are configured.

The DSP core is clocked at the DSPCLK rate (or supported divisions of the DSPCLK frequency). The DSPCLK configuration must ensure that sufficient clock cycles are available for the processing requirements of the DSP core. The requirements vary, according to the particular software that is in use.

4.16.2 Sample-Rate Control

The CS42L92 supports two independent clock domains for the audio signal paths, referenced to SYSCLK and ASYNCCLK respectively.

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. Each of these must be referenced either to SYSCLK or to ASYNCCLK. (Note that the SLIMbus interface supports multiple sample rates, selected independently for each input or output channel.)

Up to three different sample rates can be 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 4-102 and the accompanying text).

The remaining two sample rates can be selected using ASYNC_SAMPLE_RATE_1 and ASYNC_SAMPLE_RATE_2. These sample rates must be numerically related to each other and to the ASYNCCLK frequency (further details of these requirements are provided in Table 4-103 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 or ASYNC_SAMPLE_RATE_n fields.

Note that if any two interfaces are operating at the same sample rate, but are not synchronized, one of these must be referenced to the ASYNCCLK domain, and the other to the SYSCLK domain.

When any of the SAMPLE_RATE_*n* or ASYNC_SAMPLE_RATE_*n* fields is written to, the activation of the new setting is automatically synchronized by the CS42L92 to ensure continuity of all active signal paths. The SAMPLE_RATE_*n*_STS and ASYNC_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:

- Unless otherwise noted, the sample rate selection for all functions is valid from 8–192 kHz.
- The input (ADC/DMIC) signal paths must always be associated with the SYSCLK clocking domain.
- If 384- or 768-kHz DMICCLK clock rate is selected, the supported sample rate for the respective input paths is restricted as described in Table 4-1. The sample rate for the input signal paths can be set globally, or can be configured independently for each input channel—see Section 4.2.5.
- The S/PDIF sample rate is valid from 32–192 kHz. The output (DAC), digital audio interface (AIF), DSP core, and SLIMbus input/output sample rates are valid from 8–384 kHz.
- The asynchronous sample-rate converter (ASRC) supports sample rates 8–192 kHz. The ratio of the two sample rates must not exceed 6.
- The isochronous sample-rate converters (ISRCs) support sample rates 8–384 kHz. For each ISRC, the ratio of the applicable SAMPLE_RATE_*n* or ASYNC_SAMPLE_RATE_*n* fields must not exceed 24. The sample-rate conversion ratio must be an integer (1–24) or equal to 1.5.
- All external clock references (MCLK input or Slave Mode AIF input) must be within 1% of the applicable register field settings.

4.16.3 Automatic Sample-Rate Detection

The CS42L92 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 CS42L92).

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 detected sample rate. 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 384 kHz and 352.8 kHz must not be selected concurrently.
- 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 4-104.

4.16.4 System Clock Configuration

The system clocks (SYSCLK, ASYNCCLK and DSPCLK) may be provided directly from external inputs (MCLK, or Slave Mode BCLK inputs). Alternatively, these clocks can be derived using the integrated FLLs, with MCLK, BCLK, LRCLK or SLIMCLK as a reference. Each clock is configured independently, as described in the following sections.

The SYSCLK (and ASYNCCLK, when applicable) clocks must be configured and enabled before any audio path is enabled. The DSPCLK clock must be configured and enabled, if running firmware applications on the DSP core.

4.16.4.1 SYSCLK Configuration

The required SYSCLK frequency is dependent on the SAMPLE_RATE_*n* fields. Table 4-102 illustrates the valid SYSCLK frequencies for every supported sample rate.

The SYSCLK frequency must be valid for all of the SAMPLE_RATE_*n* fields. It follows that all of the SAMPLE_RATE_*n* fields must select numerically-related values, that is, all from the same group of sample rates as represented in Table 4-102.

Table 4-102. 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
			384	0x06
			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
			352.8	0x0E

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 4-104. The source may be MCLK*n*, AIF*n*BCLK, or FLL*n*. If an FLL circuit is selected as the source, the relevant FLL must be enabled and configured, as described in Section 4.16.8.

Notes: If FLL1 is selected as SYSCLK source, two different clock frequencies are available. Typical use cases should select a SYSCLK frequency equal to F_{FLL1} × 2 (i.e., in the range 90–100 MHz). A lower frequency selection, equal to F_{FLL1}, is provided to support low-power always-on use cases.

If FLL2 is selected as SYSCLK source, the SYSCLK frequency is F_{FLL2} × 2 (i.e., 90–100 MHz).

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 this 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 CS42L92 audio interfaces. The CS42L92 supports sample rates ranging from 8–384 kHz. See Section 4.16.2 for further details of the supported sample rates for each of the digital-core functions.

The SYSCLK signal is enabled by setting SYSCLK_ENA. The applicable clock source (MCLK*n*, AIF*n*BCLK, or FLL*n*) must be enabled before setting SYSCLK_ENA. This bit should be cleared before stopping or removing the applicable clock source.

The CS42L92 supports seamless switching between clock sources. To change the SYSCLK configuration while SYSCLK is enabled, the SYSCLK_FRAC, SYSCLK_FREQ, and SYSCLK_SRC fields must be updated together in one register write operation. Note that, if changing the frequency only (not the source), SYSCLK_ENA should be cleared before the clock frequency is updated. The current SYSCLK frequency and source can be read from the SYSCLK_FREQ_STS and SYSCLK_SRC_STS fields respectively.

The CS42L92 performs automatic checks to confirm that the SYSCLK frequency is high enough to support the commanded signal paths and processing functions. If the frequency is too low, an attempt to enable a signal path or processing function fails. Note that active signal paths are not affected under such circumstances.

The SYSCLK frequency check provides input to the interrupt-control circuit and can be used to trigger an interrupt event if the frequency is not high enough to support the commanded functionality; see Section 4.15.

4.16.4.2 ASYNCCLK Configuration

The required ASYNCCLK frequency is dependent on the ASYNC_SAMPLE_RATE_*n* fields. Table 4-103 illustrates the valid ASYNCCLK frequencies for every supported sample rate.

Note that, if all the sample rates in the system are synchronized to SYSCLK, the ASYNCCLK should be disabled (see Table 4-104). The associated register field values are not important in this case.

Table 4-103. ASYNCCLK Frequency Selection

ASYNCCLK Frequency (MHz)	ASYNC_CLK_FREQ	Sample Rate (kHz)	ASYNC_SAMPLE_RATE_*n*
6.144 12.288 24.576 49.152 98.304	000 001 010 011 100	12	0x01
		24	0x02
		48	0x03
		96	0x04
		192	0x05
		384	0x06
		8	0x11
		16	0x12
		32	0x13
5.6448 11.2896 22.5792 45.1584 90.3168	000 001 010 011 100	11.025	0x09
		22.05	0x0A
		44.1	0x0B
		88.2	0x0C
		176.4	0x0D
		352.8	0x0E

Note: The ASYNC_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 ASYNCCLK frequency.

ASYNC_CLK_SRC is used to select the ASYNCCLK source, as described in Table 4-104. The source may be MCLK*n*, AIF*n*BCLK, or FLL*n*. If an FLL circuit is selected as the source, the relevant FLL must be enabled and configured, as described in Section 4.16.8.

Notes: If FLL1 is selected as ASYNCCLK source, two different clock frequencies are available. Typical use cases should select a SYSCLK frequency equal to F_{FLL1} × 2 (i.e., in the range 90–100 MHz). A lower frequency selection, equal to F_{FLL1}, is provided to support low-power always-on use cases.

If FLL2 is selected as ASYNCCLK source, the ASYNCCLK frequency is F_{FLL2} × 2 (i.e., 90–100 MHz).

ASYNC_CLK_FREQ is set according to the frequency of the selected ASYNCCLK source.

The ASYNCCLK-referenced circuits within the digital core are clocked at a dynamically controlled rate that is limited by the ASYNCCLK frequency. For maximum signal mixing and processing capacity, the highest possible ASYNCCLK frequency should be used.

The ASYNC_SAMPLE_RATE_n fields are set according to the sample rates of any audio interface that is not synchronized to the SYSCLK clock domain.

The ASYNCCLK signal is enabled by setting ASYNC_CLK_ENA. The applicable clock source (MCLK*n*, AIF*n*BCLK, or FLL*n*) must be enabled before setting ASYNC_CLK_ENA. This bit should be cleared before stopping or removing the applicable clock source.

The CS42L92 supports seamless switching between clock sources. To change the ASYNCCLK configuration while ASYNCCLK is enabled, the ASYNC_CLK_FREQ and ASYNC_CLK_SRC fields must be updated together in one register write operation. Note that, if changing the frequency only (not the source), ASYNC_CLK_ENA should be cleared before the clock frequency is updated. The current ASYNCCLK frequency and source can be read from the ASYNC_CLK_FREQ_STS and ASYNC_CLK_SRC_STS fields respectively.

The CS42L92 performs automatic checks to confirm that the ASYNCCLK frequency is high enough to support the commanded signal paths and processing functions. If the frequency is too low, an attempt to enable a signal path or processing function fails. Note that active signal paths are not affected under such circumstances.

The ASYNCCLK frequency check provides input to the interrupt-control circuit and can be used to trigger an interrupt event if the frequency is not high enough to support the commanded functionality; see Section 4.15.

4.16.4.3 DSPCLK Configuration

The required DSPCLK frequency depends on the requirements of firmware loaded on the DSP core. The DSP is clocked at the DSPCLK rate or at supported divisions of the DSPCLK frequency; the DSPCLK configuration must ensure that sufficient clock cycles are available for the processing requirements. The requirements vary, according to the particular firmware that is in use.

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

DSP_CLK_FREQ must be configured for the applicable DSPCLK frequency. This field is coded in LSB units of 1/64 MHz. Note that, if the field coding cannot represent the DSPCLK frequency exactly, the DSPCLK frequency must be rounded down in the DSP_CLK_FREQ field.

The suggested method for calculating DSP_CLK_FREQ is to multiply the DSPCLK frequency by 64, round down to the nearest integer, and use the resulting integer as DSP_CLK_FREQ (LSB = 1).

DSP_CLK_SRC is used to select the DSPCLK source, as described in Table 4-104. The source may be MCLK*n*, AIF*n*BCLK, or FLL*n*. If an FLL circuit is selected as the source, the relevant FLL must be enabled and configured, as described in Section 4.16.8.

Notes: If FLL1 is selected as DSPCLK source, two different clock frequencies are available. Typical use cases should select a DSPCLK frequency equal to $F_{FLL1} \times 3$ (i.e., in the range 135–150 MHz). A lower frequency selection, equal to F_{FLL1}, is provided to support low-power always-on use cases.

If FLL2 is selected as DSPCLK source, the DSPCLK frequency is $F_{FLL2} \times 3$ (i.e., 135–150 MHz).

The DSPCLK signal is enabled by setting DSP_CLK_ENA. The applicable clock source (MCLK*n*, AIF*n*BCLK, or FLL*n*) must be enabled before setting DSP_CLK_ENA. This bit should be cleared when reconfiguring the clock sources.

The CS42L92 supports seamless switching between clock sources. To change the DSPCLK configuration while DSPCLK is enabled, the DSP_CLK_FREQ field must be updated before DSP_CLK_SRC. The new configuration becomes effective when the DSP_CLK_SRC field is written. Note that, if changing the frequency only (not the source), the DSP_CLK_ENA bit should be cleared before the clock frequency is updated. The current DSPCLK frequency and source can be read from the DSP_CLK_FREQ_STS and DSP_CLK_SRC_STS fields respectively.

In a typical application, DSPCLK and SYSCLK are derived from a single FLL source. Note that there is no requirement for DSPCLK to be synchronized to SYSCLK or ASYNCCLK. The DSPCLK controls the software execution in the DSP core; audio outputs from the DSP core are synchronized either to SYSCLK or ASYNCCLK, regardless of the applicable DSPCLK rate.

4.16.5 Miscellaneous Clock Controls

The CS42L92 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 used.

The 32-kHz clock can be generated automatically from SYSCLK, or may be provided externally via the MCLK1 or MCLK2 input pins. 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. A clock output (OPCLK_ASYNC) derived from ASYNCCLK can be output on a GPIO pin. See Section 4.14 for details on configuring a GPIO pin for these functions.

The CS42L92 provides integrated pull-down resistors on the MCLK1, MCLK2, and MCLK3 pins. This provides a flexible capability for interfacing with other devices.

The clocking scheme for the CS42L92 is shown in Fig. 4-65.

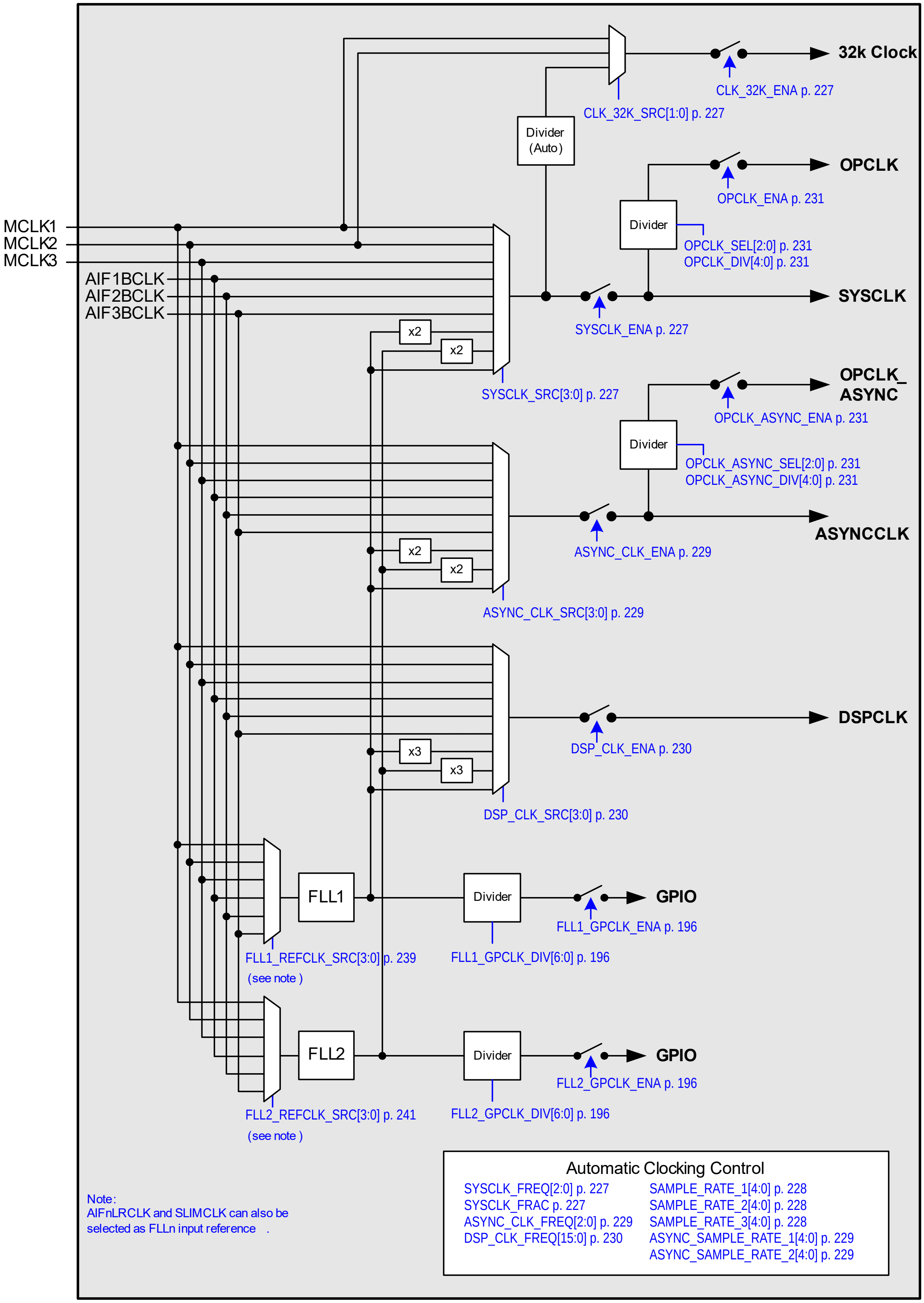


Figure 4-65. System Clocking

The CS42L92 clocking control registers are described in Table 4-104.

Table 4-104. 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. Clear this bit before stopping the reference clock or changing the reference clock frequency. Note that the SYSCLK frequency can be changed without disabling, provided the clock source is also changed at the same time.
	3:0	SYSCLK_SRC[3:0]	0100	SYSCLK Source 0000 = MCLK1 0001 = MCLK2 0010 = MCLK3 0100 = FLL1 × 2 0101 = FLL2 × 2 1000 = AIF1BCLK 1001 = AIF2BCLK 1010 = AIF3BCLK 1111 = FLL1 All other codes are reserved

Table 4-104. Clocking Control *(Cont.)*

Register Address	Bit	Label	Default	Description
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 0x06 = 384 kHz 0x09 = 11.025 kHz 0x0A = 22.05 kHz 0x0B = 44.1 kHz 0x0C = 88.2 kHz 0x0D = 176.4 kHz 0x0E = 352.8 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.
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.

Table 4-104. Clocking Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R274 (0x0112) Async_clock_1	10:8	ASYNC_CLK_ FREQ[2:0]	011	ASYNCCLK 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., ASYNC_SAMPLE_RATE_n = 01XXX).
	6	ASYNC_CLK_ ENA	0	ASYNCCLK Control 0 = Disabled 1 = Enabled ASYNCCLK should only be enabled after the applicable clock source has been configured and enabled. Clear this bit before stopping the reference clock or changing the reference clock frequency. Note that the ASYNCCLK frequency can be changed without disabling, provided the clock source is also changed at the same time.
	3:0	ASYNC_CLK_ SRC[3:0]	0101	ASYNCCLK Source 0000 = MCLK1 0001 = MCLK2 0010 = MCLK3 0100 = FLL1 × 2 0101 = FLL2 × 2 1000 = AIF1BCLK 1001 = AIF2BCLK 1010 = AIF3BCLK 1111 = FLL1 All other codes are reserved
R275 (0x0113) Async_sample_ rate_1	4:0	ASYNC_ SAMPLE_RATE_ 1[4:0]	0x11	ASYNC Sample Rate 1 Select 0x00 = None 0x01 = 12 kHz 0x02 = 24 kHz 0x03 = 48 kHz 0x04 = 96 kHz 0x05 = 192 kHz 0x06 = 384 kHz 0x09 = 11.025 kHz 0x0A = 22.05 kHz 0x0B = 44.1 kHz 0x0C = 88.2 kHz 0x0D = 176.4 kHz 0x0E = 352.8 kHz 0x11 = 8 kHz 0x12 = 16 kHz 0x13 = 32 kHz All other codes are reserved
R276 (0x0114) Async_sample_ rate_2	4:0	ASYNC_ SAMPLE_RATE_ 2[4:0]	0x11	ASYNC Sample Rate 2 Select Field coding is same as ASYNC_SAMPLE_RATE_1.
R283 (0x011B) Async_sample_ rate_1_status	4:0	ASYNC_ SAMPLE_RATE_ 1_STS[4:0]	0x00	ASYNC Sample Rate 1 Status (Read only) Field coding is same as ASYNC_SAMPLE_RATE_1.
R284 (0x011C) Async_sample_ rate_2_status	4:0	ASYNC_ SAMPLE_RATE_ 2_STS[4:0]	0x00	ASYNC Sample Rate 2 Status (Read only) Field coding is same as ASYNC_SAMPLE_RATE_1.

Table 4-104. Clocking Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R288 (0x0120) DSP_Clock_1	6	DSP_CLK_ENA	0	DSPCLK Control 0 = Disabled 1 = Enabled DSPCLK should only be enabled after the applicable clock source has been configured and enabled. Clear this bit before stopping the reference clock or changing the reference clock frequency. Note that the DSPCLK frequency can be changed without disabling, provided the clock source is also changed at the same time.
	3:0	DSP_CLK_SRC[3:0]	0101	DSPCLK Source 0000 = MCLK1 0001 = MCLK2 0010 = MCLK3 0100 = FLL1 × 3 0101 = FLL2 × 3 1000 = AIF1BCLK 1001 = AIF2BCLK 1010 = AIF3BCLK 1111 = FLL1 All other codes are reserved
R290 (0x0122) DSP_Clock_2	15:0	DSP_CLK_FREQ[15:0]	0x0000	DSPCLK Frequency Coded as LSB = 1/64 MHz, Valid from 5.6 MHz to 148 MHz. Note that, if this field is written while DSPCLK is enabled, the new frequency does not become effective until DSP_CLK_SRC is updated. To reconfigure DSPCLK while DSPCLK is enabled, the DSP_CLK_FREQ field must be updated before DSP_CLK_SRC.
R294 (0x0126) DSP_Clock_4	15:0	DSP_CLK_FREQ_STS[15:0]	0x0000	DSPCLK Frequency (Read only) Coded as LSB = 1/64 MHz.
R295 (0x0127) DSP_Clock_5	3:0	DSP_CLK_SRC_STS[3:0]	0101	DSPCLK Source (Read only) 0000 = MCLK1 0001 = MCLK2 0010 = MCLK3 0100 = FLL1 × 3 0101 = FLL2 × 3 1000 = AIF1BCLK 1001 = AIF2BCLK 1010 = AIF3BCLK 1111 = FLL1 All other codes are reserved

Table 4-104. Clocking Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R329 (0x0149) Output_system_clock	15	OPCLK_ENA	0	OPCLK Enable 0 = Disabled 1 = Enabled
	7:3	OPCLK_DIV[4:0]	0x00	OPCLK Divider 0x02 = Divide by 2 0x04 = Divide by 4 0x06 = Divide by 6 … (even numbers only) 0x1E = Divide by 30 Note that only even numbered divisions (2, 4, 6, etc.) are valid selections. All other codes are reserved when the OPCLK signal is enabled.
	2:0	OPCLK_SEL[2:0]	000	OPCLK Source Frequency 000 = 6.144 MHz (5.6448 MHz) 001 = 12.288 MHz (11.2896 MHz) 010 = 24.576 MHz (22.5792 MHz) 011 = 49.152 MHz (45.1584 MHz) All other codes are reserved The frequencies in brackets apply for 44.1 kHz–related SYSCLK rates only (i.e., SAMPLE_RATE_*n* = 01XXX). The OPCLK Source Frequency must be less than or equal to the SYSCLK frequency.
R330 (0x014A) Output_async_clock	15	OPCLK_ASYNC_ENA	0	OPCLK_ASYNC Enable 0 = Disabled 1 = Enabled
	7:3	OPCLK_ASYNC_DIV[4:0]	0x00	OPCLK_ASYNC 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_ASYNC signal is enabled.
	2:0	OPCLK_ASYNC_SEL[2:0]	000	OPCLK_ASYNC 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 ASYNCCLK rates only (i.e., ASYNC_SAMPLE_RATE_n = 01XXX). The OPCLK_ASYNC Source Frequency must be less than or equal to the ASYNCCLK frequency.
R334 (0x014E) Clock_Gen_Pad_Ctrl	9	MCLK3_PD	0	MCLK3 Pull-Down Control 0 = Disabled 1 = Enabled
	8	MCLK2_PD	0	MCLK2 Pull-Down Control 0 = Disabled 1 = Enabled
	7	MCLK1_PD	0	MCLK1 Pull-Down Control 0 = Disabled 1 = Enabled

Table 4-104. Clocking Control *(Cont.)*

Register Address	Bit	Label	Default	Description
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*.

Table 4-104. Clocking Control *(Cont.)*

Register Address	Bit	Label	Default	Description
R352 (0x0160) Clocking_debug_5	15:13	ASYNC_CLK_ FREQ_STS[2:0]	000	ASYNCCLK Frequency (Read only) 000 = 6.144 MHz (5.6448 MHz) 001 = 12.288 MHz (11.2896 MHz) 010 = 24.576 MHz (22.5792 MHz) 011 = 49.152 MHz (45.1584 MHz) 100 = 98.304 MHz (90.3168 MHz) All other codes are reserved The frequencies in brackets apply for 44.1 kHz–related sample rates only (i.e., ASYNC_SAMPLE_RATE_n = 01XXX).
	12:9	ASYNC_CLK_ SRC_STS[3:0]	0000	ASYNCCLK Source (Read only) 0000 = MCLK1 0001 = MCLK2 0010 = MCLK3 0100 = FLL1 × 2 0101 = FLL2 × 2 1000 = AIF1BCLK 1001 = AIF2BCLK 1010 = AIF3BCLK 1111 = FLL1 All other codes are reserved
	6:4	SYSCLK_FREQ_ STS[2:0]	000	SYSCLK Frequency (Read only) 000 = 6.144 MHz (5.6448 MHz) 001 = 12.288 MHz (11.2896 MHz) 010 = 24.576 MHz (22.5792 MHz) 011 = 49.152 MHz (45.1584 MHz) 100 = 98.304 MHz (90.3168 MHz) All other codes are reserved The frequencies in brackets apply for 44.1 kHz–related sample rates only (i.e., SAMPLE_RATE_*n* = 01XXX).
	3:0	SYSCLK_SRC_ STS[3:0]	0000	SYSCLK Source (Read only) 0000 = MCLK1 0001 = MCLK2 0010 = MCLK3 0100 = FLL1 × 2 0101 = FLL2 × 2 1000 = AIF1BCLK 1001 = AIF2BCLK 1010 = AIF3BCLK 1111 = FLL1 All other codes are reserved

In AIF Slave Modes, it is important to ensure that the applicable clock domain (SYSCLK or ASYNCCLK) is synchronized with the associated external LRCLK. This can be achieved by selecting an MCLK*n* input that is derived from the same reference as the LRCLK, or can be achieved by selecting the external BCLK or LRCLK signal as a reference input to one of the FLLs, as a source for SYSCLK or ASYNCCLK.

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.

4.16.6 BCLK and LRCLK Control

The digital audio interfaces (AIF1–AIF3) use BCLK and LRCLK signals for synchronization. In Master Mode, these are output signals, generated by the CS42L92. In Slave Mode, these are input signals to the CS42L92. It is also possible to support mixed master/slave operation.

The BCLK and LRCLK signals are controlled as shown in Fig. 4-66. See Section 4.7 for details of the associated control fields.

Note that the BCLK and LRCLK signals are synchronized to SYSCLK or ASYNCCLK, depending upon the applicable clock domain for the respective interface. See Section 4.3.14 for further details.

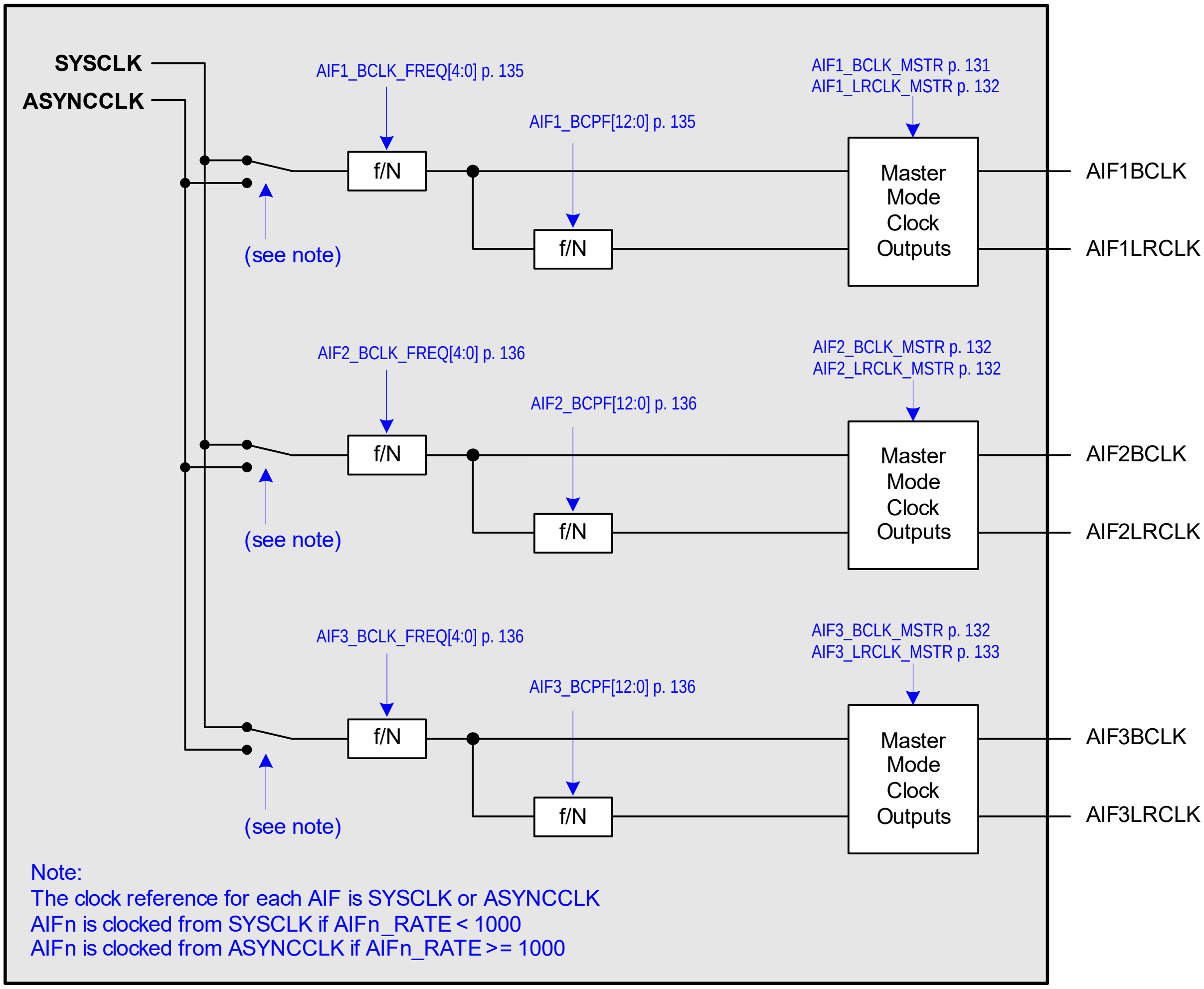


Figure 4-66. BCLK and LRCLK Control

4.16.7 Control Interface Clocking

Register map access is possible with or without a system clock—there is no requirement for SYSCLK, or any other system clock, to be enabled when accessing the register map. See Section 4.17 for details of control register access.

Timing specifications for each of the control interfaces are provided in Table 3-19–Table 3-21. In some applications, additional system-wide constraints must be observed to ensure control interface limits are not exceeded. These constraints need to be considered if any of the following conditions is true.

- SYSCLK is enabled and is < 22.5792 MHz
- Control-register access is scheduled at register address 0x8_0000 or above
- Control-register access is scheduled on more than one of the control interfaces simultaneously

The control interface limits vary depending on the system clock (SYSCLK or DSPCLK) configuration, the address of the control register access, and on which control interfaces are being used.

Table 4-105 describes valid system conditions for accessing the codec registers (below 0x8_0000). The control interfaces must operate within the limits represented by one of the permitted configurations shown, in accordance with the applicable SYSCLK frequency.

Table 4-105. Maximum SPI/SLIMbus Clock Speeds—Codec Register Access

SYSCLK Condition	SPI	SLIMbus	Description
SYSCLK is disabled, or SYSCLK ≥ 22.5792 MHz	26 MHz	27 MHz	Full concurrent SPI/SLIMbus capability for codec register access.
SYSCLK = 12.288 MHz	26 MHz	—	SPI and SLIMbus operating in isolation.
	—	27 MHz	
	13 MHz	24.576 MHz	SPI and SLIMbus operating concurrently.
	12 MHz	27 MHz	
SYSCLK = 11.2896 MHz	26 MHz	—	SPI and SLIMbus operating in isolation.
	—	27 MHz	
	12 MHz	22.5792 MHz	SPI and SLIMbus operating concurrently.
	9 MHz	27 MHz	
SYSCLK < 11.2896 MHz	13 MHz	—	SPI and SLIMbus operating in isolation.
	—	27 MHz	

Notes:
- If SYSCLK < 11.2896 MHz, simultaneous register access via SPI/SLIMbus control interfaces should not be attempted.
- If SYSCLK is disabled, full concurrent SPI/SLIMbus capability for codec register access is supported.
- The SPI interface limits noted above are only applicable if the SPI interface is accessing codec registers. Options shown with "—" in the SPI column represent use cases where the SPI interface is either unused, or is being used to access the DSP registers.
- The SLIMbus interface limits noted above are only applicable if multibyte burst transfers of more than 8 bytes are scheduled. Options shown with "—" in the SLIMbus column represent use cases where SLIMbus is either unused, or is configured to support any combination of audio channels, and burst transfers ≤ 8 bytes.
- Register access via the I^2C interface is supported at all times, regardless of the SLIMbus loading.

Table 4-106 describes valid system conditions for accessing the DSP firmware registers (0x8_0000 and above). The control interfaces must operate within the limits represented by one of the permitted configurations shown, in accordance with the applicable DSPCLK frequency.

Table 4-106. Maximum SPI/SLIMbus Clock Speeds—DSP Firmware Register Access

DSPCLK Condition	SPI	SLIMbus	Description
DSPCLK is disabled, or DSPCLK ≥ 45 MHz	26 MHz	—	One high speed interface.
	—	27 MHz	
	26 MHz	24.576 MHz	SPI and SLIMbus operating concurrently.
	21 MHz	27 MHz	
24.576 MHz ≤ DSPCLK < 45 MHz	26 MHz	—	SPI and SLIMbus operating in isolation.
	—	24.576 MHz	
12.288 MHz ≤ DSPCLK < 24.576 MHz	13 MHz	—	SPI and SLIMbus operating in isolation.
	—	12.288 MHz	

Notes:
- If DSPCLK < 24.576MHz, simultaneous register access via SPI/SLIMbus control interfaces should not be attempted.
- If DSPCLK is disabled, the valid configurations are the same as for DSPCLK ≥ 45MHz.
- The SPI interface limits noted above are only applicable if the SPI interface is accessing DSP registers. Options shown with "—" in the SPI column represent use cases where the SPI interface is either unused, or is being used to access the codec registers.
- The SLIMbus interface limits noted above are only applicable if multibyte burst transfers of more than 8 bytes are scheduled. Options shown with "—" in the SLIMbus column represent use cases where SLIMbus is either unused, or is configured to support any combination of audio channels, and burst transfers ≤ 8 bytes.
- Register access via the I^2C interface is supported at all times, regardless of the SLIMbus loading.

4.16.8 Frequency-Locked Loop (FLL1, FLL2)

Two integrated FLLs are provided to support the clocking requirements of the CS42L92. These can be configured according to the available reference clocks and the application requirements. The reference clock may use a high frequency (e.g., 12.288 MHz) or low frequency (e.g., 32.768 kHz). The FLL is tolerant of jitter and may be used to generate a stable output clock from a less stable input reference.

4.16.8.1 Overview

The FLL characteristics are summarized in Table 3-11. In normal operation, the FLL output is frequency locked to an input clock reference. The FLL can be used to generate a free-running clock in the absence of any external reference, as described in Section 4.16.8.6.

4.16.8.2 FLL Enable

The FLL is enabled by setting FLL*n*_ENA (where *n* = 1 or 2 for the corresponding FLL). Note that the other FLL fields should be configured before enabling the FLL; the FLL*n*_ENA bit should be set as the final step of the FLL*n* enable sequence.

The FLL supports configurable free-running operation in FLL Hold Mode, using the FLL*n*_HOLD bit described in Section 4.16.8.6. If the FLL is enabled and FLL Hold Mode is selected, the configured output frequency is maintained without any input reference required. Note that, once the FLL output has been established, the FLL is always free running if the input reference clock is stopped, regardless of the FLL*n*_HOLD bit.

Note that, to disable the FLL while the input reference clock has stopped, FLL*n*_HOLD must be set before clearing FLL*n*_ENA.

When changing FLL settings, it is recommended to disable the FLL by clearing FLL*n*_ENA before updating the other register fields. It is possible to configure the FLL while the FLL is enabled, as described in Section 4.16.8.4. As a general rule, however, it is recommended to configure the FLL before setting FLL*n*_ENA.

The procedure for configuring the FLL is described in the following subsections. The description is applicable to FLL1 and FLL2; the associated control fields are described in Table 4-108 and Table 4-109 respectively.

4.16.8.3 Input Frequency Control

The main input reference is selected using FLL*n*_REFCLK_SRC. The available options are MCLK*n*, 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 4.10.3.

The FLL*n*_REFCLK_DIV field controls a programmable divider on the input reference. The input can be divided by 1, 2, 4 or 8. The divider should be configured to bring each reference down to 13 MHz or below. For best performance, it is recommended that the highest possible frequency—within the 13 MHz limit—should be selected.

The FLL incorporates a reference-detection circuit for the main input clock. This ensures best FLL performance in the event of the main input clock being interrupted. If there is a possibility of the main input being interrupted while the FLL is enabled, then the reference-detection circuit must be enabled by setting FLL*n*_REFDET. The reference detection also provides input to the interrupt control circuit and can be used to trigger an interrupt event when the input reference is stopped—see Section 4.15.

4.16.8.4 Output Frequency Control

The FLL output frequency, F_{FLL}, relative to the main input reference F_{REF}, is a function of:

- The frequency ratio set by FLL*n*_FB_DIV
- The real number represented by N.K. (N = integer; K = fractional portion, i.e., < 1)

The output frequency must be in the range 45–50 MHz.

If the FLL is selected as SYSCLK or ASYNCCLK source, the respective F_{FLL} frequency must be exactly 49.152 MHz (for 48 kHz–related sample rates) or 45.1584 MHz (for 44.1 kHz–related sample rates).

- If FLL2 is selected as SYSCLK or ASYNCCLK source, the respective clock frequency is equal to $F_{FLL} \times 2$.
- If FLL1 is selected as SYSCLK or ASYNCCLK source, two different frequencies are available. Typical use cases should select the higher frequency ($F_{FLL} \times 2$); a lower frequency (F_{FLL}) is available to support low-power always-on use cases.

If the FLL is selected as DSPCLK source, the following frequency options are supported:

- If FLL2 is selected as DSPCLK source, the DSPCLK frequency is in the range 135–150 MHz. The frequency is equal to $F_{FLL} \times 3$.
- If FLL1 is selected as DSPCLK source, two different frequencies are available. Typical use cases should select the higher frequency ($F_{FLL} \times 3$); a lower frequency (F_{FLL}) is available to support low-power always-on use cases.
- Note that the DSPCLK can be divided to lower frequencies for clocking the DSP core.

The FLL clock can be configured as a GPIO output; a programmable divider supports division ratios in the range 2 through 127, enabling a wide range of GPIO clock output frequencies.

Note: The chosen F_{FLL} frequency can be used to support multiple outputs simultaneously (e.g., SYSCLK, DSPCLK, and GPIO).

To configure the FLL output frequency, it must be determined whether Integer Mode or Fractional Mode is required.

- If the ratio F_{FLL} / F_{REF} is an integer, then Integer Mode applies
- If the ratio F_{FLL} / F_{REF} is not an integer, then Fractional Mode applies

The input reference must be identified in one of three frequency ranges.

- If F_{REF} < 192 kHz, this is *low* clock frequency
- If $F_{REF} \geq$ 192 kHz and FREF < 1.152 MHz, this is *mid* clock frequency
- If $F_{REF} \geq$ 1.152 MHz, this is *high* clock frequency

Note: F_{REF} is the input frequency, after division by FLL*n*_REFCLK_DIV, where applicable.

The FLL oscillator frequency, F_{OSC}, is set according to the applicable mode and input reference frequency.

- If Fractional Mode is used and F_{REF} is high frequency, then $F_{OSC} = F_{FLL} \times 6$
- Otherwise, $F_{OSC} = F_{FLL}$

The FLL oscillator frequency, F_{OSC}, is set according to the following equation:

$$F_{OSC} = (F_{REF} \times N.K \times FLLn\_FB\_DIV)$$

The FLLn_FB_DIV value should be configured according to the applicable mode and input reference frequency.

- If Integer Mode is used and F_{REF} is low frequency, then FLL*n*_FB_DIV should be set to 4
- If Fractional Mode is used and F_{REF} is low frequency, then FLL*n*_FB_DIV should be set to 256
- Otherwise, FLL*n*_FB_DIV should be set to 1

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

$$N.K = F_{OSC} / (FLLn\_FB\_DIV \times F_{REF})$$

The calculated value of N must lie within a valid range, according to the applicable mode.

- If Integer Mode is used, N is valid in the range 1–1023
- If Fractional Mode is used, N is valid in the range 4–255

If the calculated value of N is too high, a higher FLL*n*_FB_DIV is required. If the calculated value of N is too low, a lower FLL*n*_FB_DIV is required. It is recommended to adjust the FLL*n*_FB_DIV value by multiplying or dividing by 2 until a valid N is achieved.

The value of N is held in FLL*n*_N.

The value of K is determined by the ratio FLL*n*_THETA / FLL*n*_LAMBDA. In Fractional Mode, the FLL*n*_THETA and FLL*n*_LAMBDA fields can be derived as described in Section 4.16.8.5.

The FLL*n*_N, FLL*n*_THETA, and FLL*n*_LAMBDA fields are all coded as integers (LSB = 1).

When changing FLL settings, it is recommended to disable the FLL by clearing FLL*n*_ENA before updating the other register fields. If the FLL settings or input reference are changed without disabling the FLL, the FLL Hold Mode must be selected before writing to any other FLL control fields. FLL Hold Mode is selected by setting FLL*n*_HOLD.

If the FLL control fields are written while the FLL is enabled (FLL*n*_ENA = 1), the new values are only effective when a 1 is written to FLL*n*_CTRL_UPD. This makes it possible to update the FLL configuration fields simultaneously, without disabling the FLL.

To change FLL settings without disabling the FLL, the recommended control sequence is:

- Select FLL Hold Mode (FLL*n*_HOLD = 1)
- Write to the FLL control fields
- Update the FLL control registers (write 1 to FLL*n*_CTRL_UPD)
- Disable FLL Hold Mode (FLL*n*_HOLD = 0)

Note that, if the FLL is disabled, the FLL control fields can be updated without writing to FLL*n*_CTRL_UPD.

The FLL*n*_PD_GAIN_FINE, FLL*n*_PD_GAIN_COARSE, FLL*n*_FD_GAIN_FINE, FLL*n*_FD_GAIN_COARSE, FLL*n*_HP, and FLLn_CLK_VCO_FAST_SRC fields should be configured as described in Table 4-107.

Note: When writing to the FLL*n*_CLK_VCO_FAST_SRC or FLL*n*_HP fields, take care not to change other nonzero bits that are configured at the same register address.

Table 4-107. FLL*n* Control Field Settings

Condition	FLL*n*_PD_GAIN_FINE	FLL*n*_PD_GAIN_COARSE	FLL*n*_FD_GAIN_FINE	FLL*n*_FD_GAIN_COARSE	FLL*n*_LOCKDET_THR	FLL*n*_CLK_VCO_FAST_SRC	FLL*n*_HP
Low clock frequency Mid clock frequency High clock frequency	0x2 0x2 0x2	0x3 0x2 0x1	0xF 0xF 0xF	0x0 0x2 0x0	0x2 0x8 0x8	—	—
Integer Mode Fractional Mode, Low clock frequency Fractional Mode, Mid clock frequency Fractional Mode, High clock frequency	—	—	—	—	—	0x0 0x0 0x0 0x3	—
Integer Mode Fractional Mode	—	—	—	—	—	—	0x0 0x3

4.16.8.5 Calculation of Theta and Lambda

In Fractional Mode, FLL*n*_THETA and FLL*n*_LAMBDA are calculated with the following steps:

1. Calculate GCD(FLL) using the Greatest Common Denominator function:

   GCD(FLL) = GCD(FLL*n*_FB_DIV × F_{REF}, F_{OSC}),

   where GCD(x, y) is the greatest common denominator of x and y.

   F_{REF} is the input frequency, after division by FLL*n*_REFCLK_DIV, where applicable.

2. Calculate FLL*n*_THETA and FLL*n*_LAMBDA using the following equations:

   FLL*n*_THETA = (F_{OSC} – (FLL_N × FLL*n*_FB_DIV × F_{REF})) / GCD(FLL)

   FLL*n*_LAMBDA = (FLL*n*_FB_DIV × F_{REF}) / GCD(FLL)

Notes: The values of GCD(FLL), FLL*n*_THETA, and FLL*n*_LAMBDA should be calculated using the applicable frequency values in Hz (i.e., not kHz or MHz).

In Fractional Mode, the values of FLL*n*_THETA and FLL*n*_LAMBDA must be coprime (i.e., not divisible by any common integer). The calculation above ensures that the values are coprime.

The value of K must be less than 1 (i.e., FLL*n*_THETA must be less than FLL*n*_LAMBDA).

4.16.8.6 FLL Hold Mode

FLL Hold Mode enables the FLL to generate a clock signal even if no external reference is available, such as when the normal input reference has been interrupted during a standby or start-up period. FLL Hold Mode is selected by setting FLL*n*_HOLD.

If the FLL is enabled and FLL Hold Mode is selected, the normal feedback mechanism of the FLL is halted and the FLL oscillates independently of the external input references—the FLL output frequency remains unchanged if FLL Hold Mode is enabled.

If the FLL is enabled and the input reference clock is stopped, the loop always runs freely, regardless of the FLL*n*_HOLD setting. If FLL*n*_HOLD = 0, the FLL relocks to the input reference whenever it is available.

If the FLL configuration or input reference are changed without disabling the FLL, the FLL Hold Mode must be selected before writing to any other FLL control fields—see Section 4.16.8.4.

The free-running FLL clock may be selected as the SYSCLK, ASYNCCLK, or DSPCLK source, as shown in Fig. 4-65.

4.16.8.7 FLL Control Registers

The FLL1 control registers are described in Table 4-108.

Example settings for a variety of reference frequencies and output frequencies are shown in Section 4.16.8.10.

Table 4-108. FLL1 Register Map

Register Address	Bit	Label	Default	Description
R369 (0x0171) FLL1_Control_1	3:0	FLL1_REFCLK_SRC[3:0]	0000	FLL1 Clock source 0000 = MCLK1 0001 = MCLK2 0010 = MCLK3 0011 = SLIMCLK 1000 = AIF1BCLK 1001 = AIF2BCLK 1010 = AIF3BCLK 1100 = AIF1LRCLK 1101 = AIF2LRCLK 1110 = AIF3LRCLK All other codes are reserved
	2	FLL1_HOLD	1	FLL1 Hold Mode Enable 0 = Disabled 1 = Enabled The FLL feedback mechanism is halted in FLL Hold 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.
R370 (0x0172) FLL1_Control_2	15	FLL1_CTRL_UPD	0	FLL1 Control Update Write 1 to apply the FLL1 configuration field settings. (Only valid if FLL1_ENA = 1)
	9:0	FLL1_N[9:0]	0x004	FLL1 Integer multiply for F_{REF} Coded as LSB = 1.
R371 (0x0173) FLL1_Control_3	15:0	FLL1_THETA[15:0]	0x0000	FLL1 Fractional multiply for F_{REF}. Sets the numerator (multiply) part of the FLL1_THETA/FLL1_LAMBDA ratio. Coded as LSB = 1.
R372 (0x0174) FLL1_Control_4	15:0	FLL1_LAMBDA[15:0]	0x0000	FLL1 Fractional multiply for F_{REF}. Sets the denominator (dividing) part of the FLL1_THETA/FLL1_LAMBDA ratio. Coded as LSB = 1.
R373 (0x0175) FLL1_Control_5	9:0	FLL1_FB_DIV[9:0]	0x0001	FLL1 Clock Feedback ratio Coded as LSB = 1.
R374 (0x0176) FLL1_Control_6	15	FLL1_REFDET	1	FLL1 Reference Detect control 0 = Disabled 1 = Enabled
	7:6	FLL1_REFCLK_DIV[1:0]	00	FLL1 Clock Reference divider 00 = 1 01 = 2 10 = 4 11 = 8 MCLK (or other input reference) must be divided down to ≤ 13 MHz.

Table 4-108. FLL1 Register Map *(Cont.)*

Register Address	Bit	Label	Default	Description
R376 (0x0178) FLL1_Control_8	15:12	FLL1_PD_GAIN_ FINE[3:0]	0x2	FLL1 Phase Detector Gain 2 Gain is 2^{-X}, where X is FLL1_PD_GAIN_FINE in 2’s complement coding. 0000 = 1 0110 = 2^{-6} 1100 = 16 0001 = 0.5 0111 = 2^{-7} 1101 = 8 0010 = 0.25 1000 = 256 1110 = 4 0011 = 0.125 1001 = 128 1111 = 2 0100 = 2^{-4} 1010 = 64 0101 = 2^{-5} 1011 = 32
	11:8	FLL1_PD_GAIN_ COARSE[3:0]	0x1	FLL1 Phase Detector Gain 1 Gain is 2^{-X}, where X is FLL1_PD_GAIN_COARSE in 2’s complement coding. 0000 = 1 0110 = 2^{-6} 1100 = 16 0001 = 0.5 0111 = 2^{-7} 1101 = 8 0010 = 0.25 1000 = 256 1110 = 4 0011 = 0.125 1001 = 128 1111 = 2 0100 = 2^{-4} 1010 = 64 0101 = 2^{-5} 1011 = 32
	7:4	FLL1_FD_GAIN_ FINE[3:0]	0xF	FLL1 Frequency Detector Gain 2 Gain is 2^{-X}, where X is FLL1_FD_GAIN_FINE in integer coding. 0000 = 1 0011 = 0.125 1110 = 2^{-14} 0001 = 0.5 ... 1111 = Reserved 0010 = 0.25 1101 = 2^{-13}
	3:0	FLL1_FD_GAIN_ COARSE[3:0]	0x0	FLL1 Frequency Detector Gain 1 Gain is 2^{-X}, where X is FLL1_FD_GAIN_COARSE in 2’s complement coding. 0000 = 1 0110 = 2^{-6} 1100 = 16 0001 = 0.5 0111 = 2^{-7} 1101 = 8 0010 = 0.25 1000 = 256 1110 = 4 0011 = 0.125 1001 = 128 1111 = 2 0100 = 2^{-4} 1010 = 64 0101 = 2^{-5} 1011 = 32
R378 (0x017A) FLL1_Control_10	15:14	FLL1_HP[1:0]	00	FLL1 Fractional Mode control 00 = Integer mode 10 = Reserved 01 = Reserved 11 = Fractional Mode
R379 (0x017B) FLL1_Control_11	4:1	FLL1_LOCKDET_ THR[3:0]	0x8	FLL1 Lock Detect threshold Valid from 0x0 (low threshold) to 0xF (high threshold)
	0	FLL1_LOCKDET	1	FLL1 Lock Detect enabled 0 = Disabled 1 = Enabled
R381 (0x017D) FLL1_Digital_ Test_1	1:0	FLL1_CLK_VCO_ FAST_SRC[1:0]	0x0	FLL1 Oscillator Frequency Control 00 = 45–50 MHz 10 = Reserved 01 = Reserved 11 = 270–300 MHz

The FLL2 control registers are described in Table 4-109.

Table 4-109. FLL2 Register Map

Register Address	Bit	Label	Default	Description
R401 (0x0191) FLL2_Control_1	3:0	FLL2_REFCLK_SRC[3:0]	0111	FLL2 Clock source 0000 = MCLK1; 0001 = MCLK2; 0010 = MCLK3; 0011 = SLIMCLK; 1000 = AIF1BCLK; 1001 = AIF2BCLK; 1010 = AIF3BCLK; 1100 = AIF1LRCLK; 1101 = AIF2LRCLK; 1110 = AIF3LRCLK; All other codes are reserved
	2	FLL2_HOLD	1	FLL2 Hold Mode Enable 0 = Disabled 1 = Enabled The FLL feedback mechanism is halted in FLL Hold Mode, and the latest integrator setting is maintained.
	0	FLL2_ENA	0	FLL2 Enable 0 = Disabled 1 = Enabled This should be set as the final step of the FLL2 enable sequence.
R402 (0x0192) FLL2_Control_2	15	FLL2_CTRL_UPD	0	FLL2 Control Update Write 1 to apply the FLL2 configuration field settings. (Only valid if FLL2_ENA = 1)
	9:0	FLL2_N[9:0]	0x004	FLL2 Integer multiply for F_{REF} Coded as LSB = 1.
R403 (0x0193) FLL2_Control_3	15:0	FLL2_THETA[15:0]	0x0000	FLL2 Fractional multiply for F_{REF}. Sets the numerator (multiply) part of the FLL2_THETA/FLL2_LAMBDA ratio. Coded as LSB = 1.
R404 (0x0194) FLL2_Control_4	15:0	FLL2_LAMBDA[15:0]	0x0000	FLL2 Fractional multiply for F_{REF}. Sets the denominator (dividing) part of the FLL2_THETA/FLL2_LAMBDA ratio. Coded as LSB = 1.
R405 (0x0195) FLL2_Control_5	9:0	FLL2_FB_DIV[9:0]	0x0001	FLL2 Clock Feedback ratio Coded as LSB = 1.
R406 (0x0196) FLL2_Control_6	15	FLL2_REFDET	1	FLL2 Reference Detect control 0 = Disabled 1 = Enabled
	7:6	FLL2_REFCLK_DIV[1:0]	00	FLL2 Clock Reference divider 00 = 1; 01 = 2; 10 = 4; 11 = 8 MCLK (or other input reference) must be divided down to ≤ 13 MHz.

Table 4-109. FLL2 Register Map *(Cont.)*

Register Address	Bit	Label	Default	Description
R408 (0x0198) FLL2_Control_8	15:12	FLL2_PD_GAIN_ FINE[3:0]	0x2	FLL2 Phase Detector Gain 2 Gain is 2^{-X}, where X is FLL2_PD_GAIN_FINE in 2's complement coding. 0000 = 1 0110 = 2^{-6} 1100 = 16 0001 = 0.5 0111 = 2^{-7} 1101 = 8 0010 = 0.25 1000 = 256 1110 = 4 0011 = 0.125 1001 = 128 1111 = 2 0100 = 2^{-4} 1010 = 64 0101 = 2^{-5} 1011 = 32
	11:8	FLL2_PD_GAIN_ COARSE[3:0]	0x1	FLL2 Phase Detector Gain 1 Gain is 2^{-X}, where X is FLL2_PD_GAIN_COARSE in 2's complement coding. 0000 = 1 0110 = 2^{-6} 1100 = 16 0001 = 0.5 0111 = 2^{-7} 1101 = 8 0010 = 0.25 1000 = 256 1110 = 4 0011 = 0.125 1001 = 128 1111 = 2 0100 = 2^{-4} 1010 = 64 0101 = 2^{-5} 1011 = 32
	7:4	FLL2_FD_GAIN_ FINE[3:0]	0xF	FLL2 Frequency Detector Gain 2 Gain is 2^{-X}, where X is FLL2_FD_GAIN_FINE in integer coding. 0000 = 1 0011 = 0.125 1110 = 2^{-14} 0001 = 0.5 ... 1111 = Reserved 0010 = 0.25 1101 = 2^{-13}
	3:0	FLL2_FD_GAIN_ COARSE[3:0]	0x0	FLL2 Frequency Detector Gain 1 Gain is 2^{-X}, where X is FLL2_FD_GAIN_COARSE in 2's complement coding. 0000 = 1 0110 = 2^{-6} 1100 = 16 0001 = 0.5 0111 = 2^{-7} 1101 = 8 0010 = 0.25 1000 = 256 1110 = 4 0011 = 0.125 1001 = 128 1111 = 2 0100 = 2^{-4} 1010 = 64 0101 = 2^{-5} 1011 = 32
R410 (0x019A) FLL2_Control_10	15:14	FLL2_HP[1:0]	00	FLL2 Fractional Mode control 00 = Integer mode 10 = Reserved 01 = Reserved 11 = Fractional Mode
R411 (0x019B) FLL2_Control_11	4:1	FLL2_LOCKDET_ THR[3:0]	0x8	FLL2 Lock Detect threshold Valid from 0x0 (low threshold) to 0xF (high threshold)
	0	FLL2_LOCKDET	1	FLL2 Lock Detect enabled 0 = Disabled 1 = Enabled
R413 (0x019D) FLL2_Digital_ Test_1	1:0	FLL2_CLK_VCO_ FAST_SRC[1:0]	0x0	FLL2 Oscillator Frequency Control 00 = 45–50 MHz 10 = Reserved 01 = Reserved 11 = 270–300 MHz

4.16.8.8 FLL Interrupts and GPIO Output

For each FLL, the CS42L92 provides status signals that indicate whether the input reference is present and whether FLL lock has been achieved (i.e., the FLL is locked to the input reference signal).

To enable the FLL lock indication, the FLL*n*_LOCKDET bit must be set. The FLL lock condition is measured with respect to a configurable threshold that is set using FLL*n*_LOCKDET_THR. Note that the FLL*n*_LOCKDET_THR field controls the lock indication only—it does not control the behavior of the FLL.

To enable the FLL input reference indication, the FLL*n*_REFDET bit must be set.

The FLL status signals are inputs to the interrupt control circuit and can be used to trigger an interrupt event when the input reference is stopped or when the FLL lock status changes. See Section 4.15. Note that these interrupt signals are debounced and require 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 signals.

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. (These GPIO outputs are not debounced and do 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.

4.16.8.9 Example FLL Calculation

The following example illustrates how to derive the FLL1 register fields to generate an FLL output frequency (F_{FLL}) of 49.152 MHz from a 12.000-MHz reference clock (F_{REF}). This is suitable for generating SYSCLK at 98.304 MHz and DSPCLK at 147.456 MHz.

1. Set FLL1_REFCLK_DIV to generate $F_{REF} \leq 13$ MHz:
   FLL1_REFCLK_DIV = 00 (divide by 1)
2. Determine if Integer Mode or Fractional Mode is required:
   F_{FLL} / F_{REF} is 4.096. Therefore, Fractional Mode applies.
3. Identify the input clock frequency range:
   $F_{REF} \geq 1.152$ MHz. This is *high* clock frequency.
4. Calculate the FLL oscillator frequency, F_{OSC}:
   In Fractional Mode, with high clock frequency input, $F_{OSC} = F_{FLL} \times 6 = 294.912$ MHz
5. Select the required value of FLL1_FB_DIV:
   In Fractional Mode, with high clock frequency input, FLL1_FB_DIV = 1
6. Calculate N.K as given by N.K = F_{OSC} / (FLL1_FB_DIV × F_{REF}):
   N.K = 294912000 / (1 × 12000000) = 24.576
7. Confirm that the calculated value of N is within the valid range for fractional mode (4–255).
8. Determine FLL1_N from the integer portion of N.K:
   FLL1_N = 24 (0x018)
9. Determine GCD(FLL), as given by GCD(FLL) = GCD(FLL1_FB_DIV × F_{REF}, F_{OSC}):
   GCD(FLL) = GCD(1 × 12000000, 294912000) = 96000
10. Determine FLL1_THETA, as given by FLL1_THETA = (F_{OSC} – (FLL1_N × FLL1_FB_DIV × F_{REF})) / GCD(FLL):
    FLL1_THETA = (294912000 – (24 × 1 × 12000000)) / 96000
    FLL1_THETA = 72 (0x0048)
11. Determine FLL1_LAMBDA, as given by FLL1_LAMBDA = (FLL1_FB_DIV x F_{REF}) / GCD(FLL):
    FLL1_LAMBDA = (1 × 12000000) / 96000
    FLL1_LAMBDA = 125 (0x007D)
12. Determine other FLL settings as specified in Table 4-107 for Fractional Mode and high clock frequency input:
    FLL1_PD_GAIN_FINE = 0x2
    FLL1_PD_GAIN_COARSE = 0x1
    FLL1_FD_GAIN_FINE = 0xF
    FLL1_FD_GAIN_COARSE = 0x0
    FLL1_CLK_VCO_FAST_SRC = 0x3
    FLL1_HP = 0x3

4.16.8.10 Example FLL Settings

Table 4-110 shows FLL settings for generating an output frequency (F_{FLL}) of 49.152 MHz from a variety of low- and high-frequency reference inputs. This is suitable for generating SYSCLK at 98.304 MHz and DSPCLK at 147.456 MHz.

Table 4-110. Example FLL Settings

F_{SOURCE}	F_{FLL} (MHz)	F_{REF} Divider[1]	FB_DIV[1]	N.K [2]	FLL*n*_N	FLL*n*_ THETA	FLL*n*_ LAMBDA
32.000 kHz	49.152	1	4	384	0x180	0x0000	0x0001
32.768 kHz	49.152	1	4	375	0x177	0x0000	0x0001
44.100 kHz	49.152	1	256	4.3537415	0x004	0x0034	0x0093
48 kHz	49.152	1	4	256	0x100	0x0000	0x0001
128 kHz	49.152	1	4	96	0x060	0x0000	0x0001
9.6 MHz	49.152	1	1	30.72	0x01E	0x0012	0x0019
10 MHz	49.152	1	1	29.4912	0x01D	0x0133	0x0271
11.2896 MHz	49.152	1	1	26.12245	0x01A	0x0006	0x0031
12.000 MHz	49.152	1	1	24.576	0x018	0x0048	0x007D
12.288 MHz	49.152	1	1	4	0x004	0x0000	0x0001
13.000 MHz	49.152	1	1	22.68554	0x016	0x045A	0x0659
19.200 MHz	49.152	2	1	30.72	0x01E	0x0012	0x0019
22.5792 MHz	49.152	2	1	26.12245	0x01A	0x0006	0x0031
24 MHz	49.152	2	1	24.576	0x018	0x0048	0x007D
24.576 MHz	49.152	2	1	4	0x004	0x0000	0x0001
26 MHz	49.152	2	1	22.68554	0x016	0x045A	0x0659

1.See Table 4-108 and Table 4-109 for the coding of the FLL*n*_REFCLK_DIV and FLL*n*_FB_DIV fields.
2.N.K values are represented in the FLL*n*_N, FLL*n*_THETA, and FLL*n*_LAMBDA fields.

4.17 Control Interface

The CS42L92 is controlled by read/write access to its control registers. The control interface supports 2-wire (I^2C) and 4-wire (SPI) modes. Note that the SLIMbus interface also supports read/write access to the CS42L92 control registers; see Section 4.10.

The CS42L92 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.

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.

Timing specifications for each of the control interfaces are provided in Table 3-19–Table 3-21. In some applications, additional system-wide constraints must be observed to ensure control interface limits are not exceeded. Full details of these requirements are provided in Section 4.16.7. These constraints need to be considered if any of the following conditions is true.

- SYSCLK is enabled and is < 22.5792 MHz
- Control-register access is scheduled at register address 0x80000 or above
- Control-register access is scheduled on more than one of the control interfaces simultaneously

The control interface can be configured as a 2-wire (I^2C) or 4-wire (SPI) interface. The mode is determined by the logic level on the CIFMODE pin, as described in Table 4-111.

Table 4-111. CS42L92 Control Interface Summary

CIFMODE	Interface Mode	Pin Functions
Logic 1	Four-wire (SPI) interface	CIFMISO—Data output CIFMOSI—Data input CIFSCLK—Interface clock input $\overline{\text{CIFSS}}$—Slave select input
Logic 0	Two-wire (I²C) interface	CIFSCLK—Interface clock input CIFSDA—Data input/output

Note: The CIFMOSI and CIFSDA functions are multiplexed on a dual-function pin.

An integrated pull-down resistor is provided on the CIF1MISO pin. This provides a flexible capability for interfacing with other devices. The pull-down is configured using the CIF1MISO_PD bit, as described in Table 4-112.

Table 4-112. Control Interface Pull-Down

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

A detailed description of the I²C 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 using the $\overline{\text{CIFSS}}$, CIFSCLK, CIFMOSI, and CIFMISO pins.

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

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

If $\overline{\text{SS}}$ is asserted (Logic 0), the MISO output is actively driven when outputting data and is high impedance at other times. If $\overline{\text{SS}}$ 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 MISO pin, as described in Table 4-112.

Data transfers on the SPI interface must use the applicable 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. 4-67 and Fig. 4-68 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 CS42L92 automatically increments the register address at the end of each data word, for as long as $\overline{\text{SS}}$ 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. 4-67 and Fig. 4-68. Note that 16-bit data words are shown, but the equivalent protocol also applies to 32-bit data words.

Fig. 4-67 shows a single register write to a specified address.

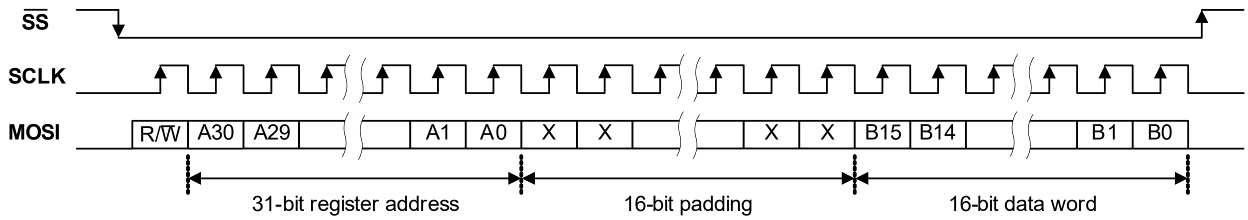


Figure 4-67. Control Interface SPI Register Write (16-Bit Data Words)

Fig. 4-68 shows a single register read from a specified address.

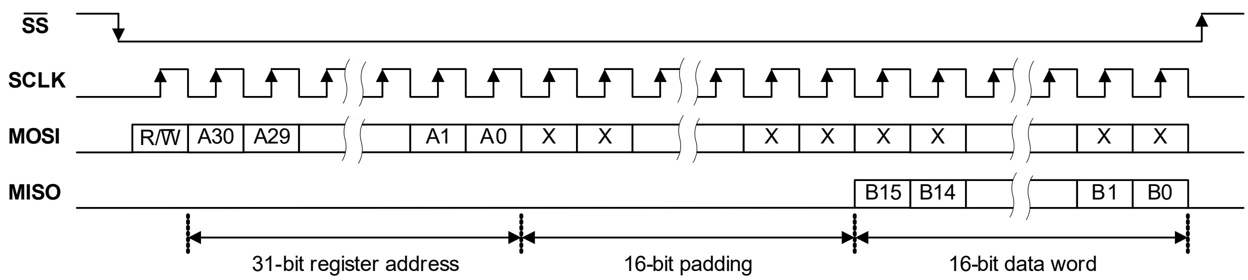


Figure 4-68. Control Interface SPI Register Read (16-Bit Data Words)

4.17.2 Two-Wire (I2C) Control Mode

The I2C control interface mode is supported using the CIFSCLK and CIFSDA pins.

In I2C Mode, the CS42L92 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 CS42L92 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 CS42L92).

The CS42L92 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 CS42L92 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 CS42L92 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 CS42L92, the CS42L92 responds by pulling SDA low on the next clock pulse (ACK). If the device ID is not recognized or the R/W bit is set incorrectly, the CS42L92 returns to the idle condition and waits for a new start condition.

If the device ID matches the device ID of the CS42L92, 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 CS42L92 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 the I2C interface must use the applicable 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. 4-69 and Fig. 4-70).

The CS42L92 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 CS42L92 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. 4-69.

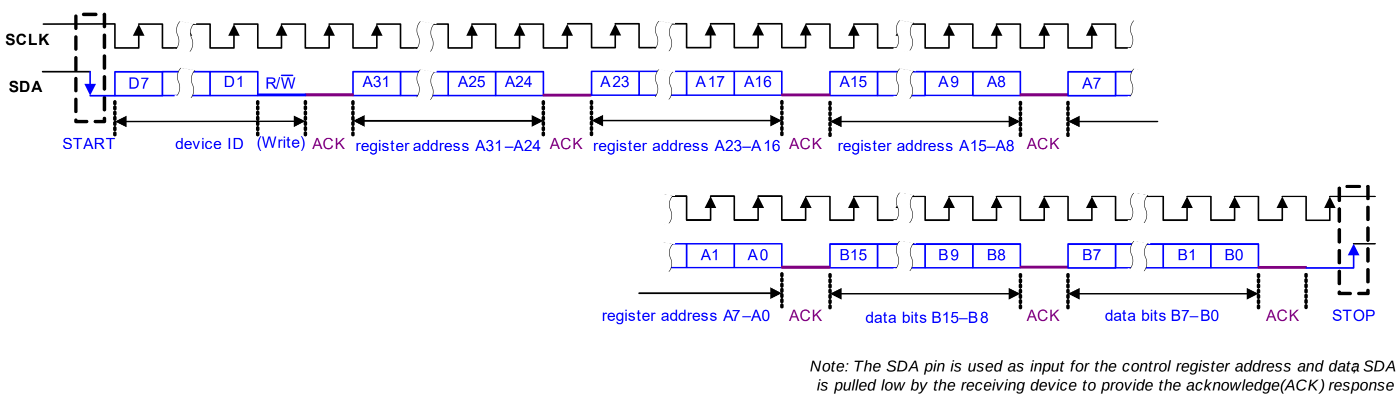


Figure 4-69. Control Interface I2C Register Write (16-Bit Data Words)

The I2C protocol for a single, 16-bit register read operation is shown in Fig. 4-70.

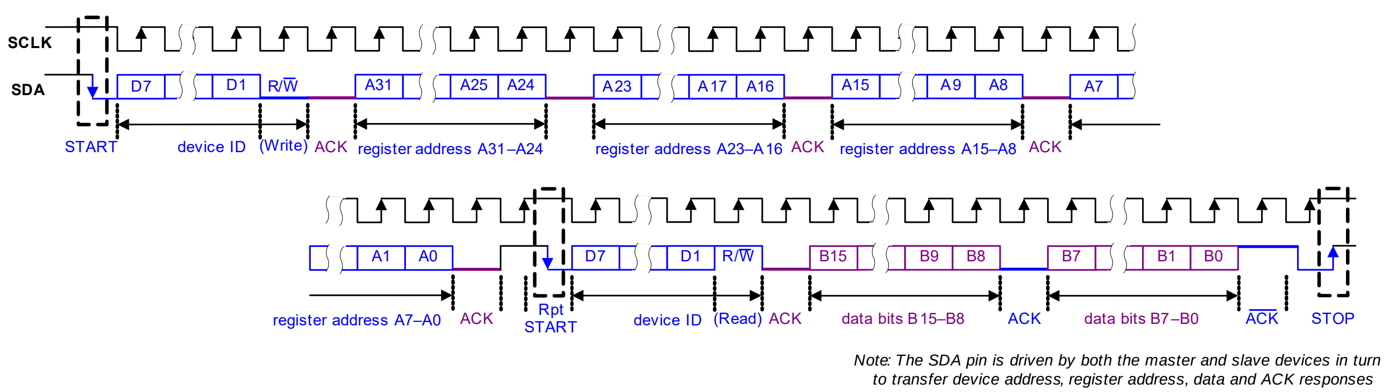


Figure 4-70. Control Interface I2C Register Read (16-Bit Data Words)

The control interface also supports other register operations; the interface protocol for these operations is shown in Fig. 4-71 through Fig. 4-74. The terminology used in the following figures is detailed in Table 4-113.

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 4-113. 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 CS42L92
[Gray field]	Data flow from CS42L92 to bus master

Fig. 4-71 shows a single register write to a specified address.

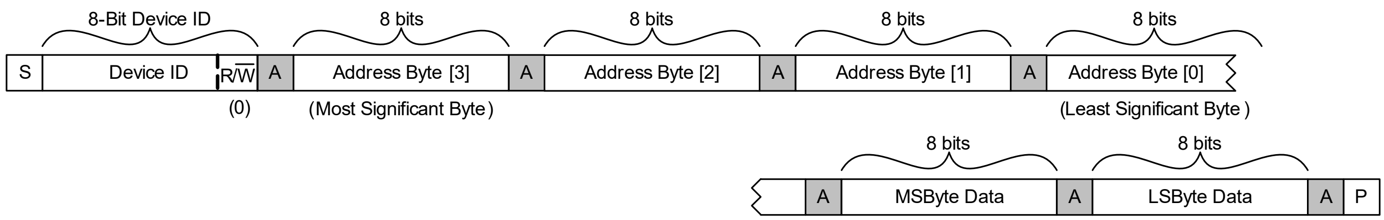


Figure 4-71. Single-Register Write to Specified Address

Fig. 4-72 shows a single register read from a specified address.

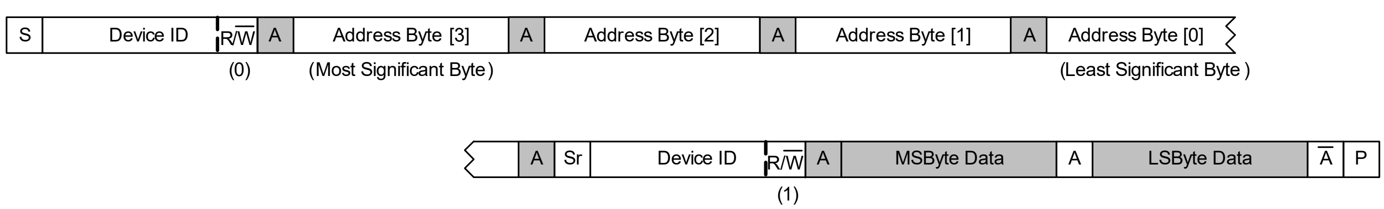


Figure 4-72. Single-Register Read from Specified Address

Fig. 4-73 shows a multiple register write to a specified address.

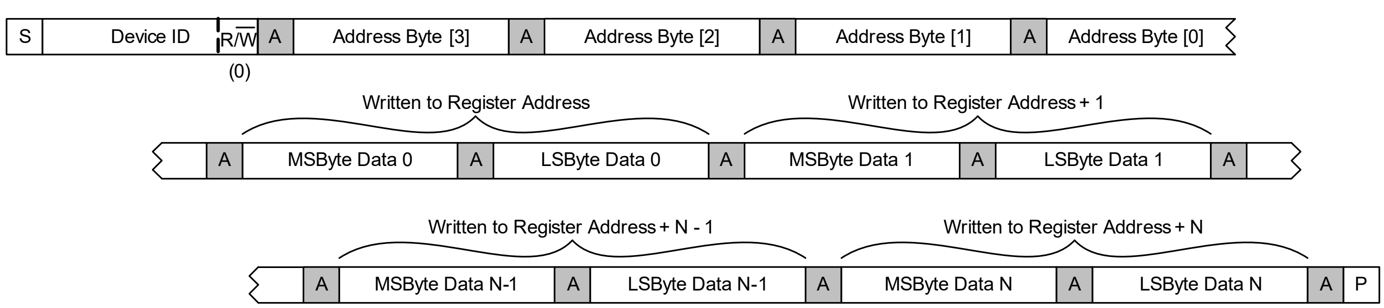


Figure 4-73. Multiple-Register Write to Specified Address

Fig. 4-74 shows a multiple register read from a specified address.

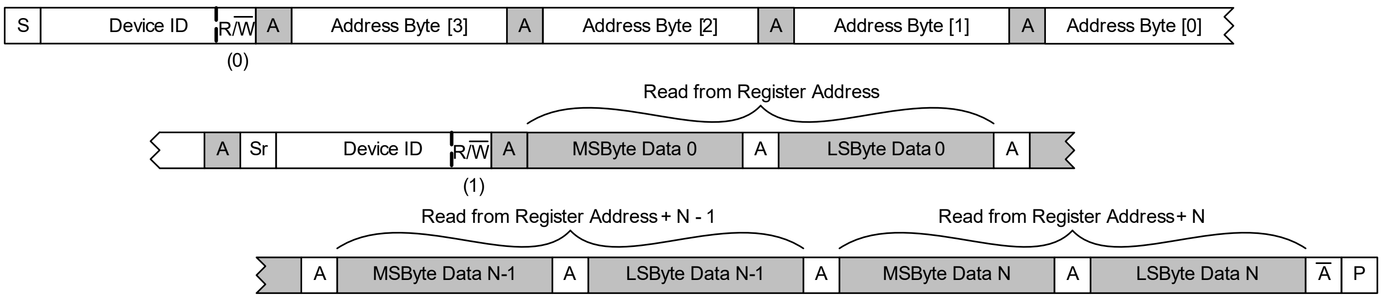


Figure 4-74. Multiple-Register Read from Specified Address

4.18 Control-Write Sequencer

The control-write sequencer is a programmable unit that forms part of the CS42L92 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, 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 4-114.

The CS42L92 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 trigger control bit, as described in Table 4-114. 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 4-120) 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 508 register-write commands. These are defined in registers R12288 (0x3000) through R13302 (0x33F6). See Table 4-121 for a description of these registers.

Table 4-114. 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 507 (0x1FB).
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 4-114. 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 507 (0x1FB).
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 507 (0x1FB).
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 507 (0x1FB).
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 507 (0x1FB).
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 507 (0x1FB).
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 507 (0x1FB).
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 507 (0x1FB).
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 507 (0x1FB).
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 507 (0x1FB).
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 507 (0x1FB).
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 507 (0x1FB).
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 507 (0x1FB).
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 507 (0x1FB).
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 507 (0x1FB).
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 507 (0x1FB).
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 507 (0x1FB).

4.18.2 Automatic Sample-Rate Detection Sequences

The CS42L92 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 4-104.

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 4-115.

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 4-115. 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 507 (0x1FB).
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 507 (0x1FB).
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 507 (0x1FB).
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 507 (0x1FB).

4.18.3 DRC Signal-Detect Sequences

The DRC function within the CS42L92 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 4-17 and Table 4-18 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 4-17.

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 4-116. 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 4-116. 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 507 (0x1FB).
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 507 (0x1FB).

4.18.4 MICDET Clamp Sequences

The CS42L92 supports external accessory detection functions, including the MICDET clamp circuits. The MICDET clamp status can be used to trigger the control-write sequencer. The MICDET clamps can be controlled using selectable logic conditions in respect of the JD*n* signals, as described in Table 4-85.

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 WSEQ_ENA_MICD_CLAMP_RISE and WSEQ_ENA_MICD_CLAMP_FALL bits, as described in Table 4-85.

If one of the selected JD*n* logic conditions is detected, the control-write sequencer is triggered. Note that these control-sequencer events are only valid if the clamp status changed in response to the JD*n* signals. The applicable start index location within the sequencer memory is separately configurable for the rising and falling edge conditions.

Note that, due to control logic that is shared between the two MICDET clamps, the option to control both clamps in response to the JD*n* signals cannot be supported at the same time. It is assumed that a maximum of one clamp is active at any time. Accordingly, only the active clamp is capable of triggering the control-write sequencer.

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 4-117. 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 4-117. 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 507 (0x1FB).
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 507 (0x1FB).

4.18.5 Event Logger Sequences

The CS42L92 provides an event log function, 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.

The WSEQ_EVENTLOG1_INDEX field defines the sequencer start index corresponding to the event logger, as described in Table 4-118.

Note that a sequencer start index of 0x1FF causes the respective sequence to be aborted.

The event logger control sequence is undefined following power-on reset, a hardware reset, or a Sleep Mode transition. The event logger control sequence must be reconfigured by the host processor following any of these events. Note that all control sequences are maintained in the sequencer memory through software reset.

See Section 4.5.1 for further details of the event logger.

Table 4-118. Write Sequencer Control—Event Logger

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 507 (0x1FB).

4.18.6 Boot Sequence

The CS42L92 executes a boot sequence following power-on reset, hardware reset, software reset, or wake-up from Sleep Mode. The boot sequence configures the CS42L92 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 384 (0x180). See Table 4-123 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 4-119. 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 384 (0x180).

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 4-120. 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 4-120. 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 five fields, contained within a single 32-bit register. An extended instruction set is also defined; the associated actions make use of alternate definitions of the 32-bit registers.

The sequencer instruction fields are replicated 508 times, defining each of the sequencer's 508 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 507):

- WSEQ_DATA_WIDTH*n* is a 3-bit field that identifies the width of the data block to be written. Note that the maximum value of this field selects a width of 8 bits; writes to fields that are larger than 8 bits wide must be performed using two separate operations of the write sequencer.
- WSEQ_ADDR*n* is a 12-bit field containing the register address in which the data should be written. The applicable register address is referenced to the base address currently configured for the sequencer—it is calculated as: (base address * 512) + WSEQ_ADDR*n*. Note that the base address is configured using the sequencer's extended instruction set.
- WSEQ_DELAY*n* is a 4-bit field that controls the waiting time between the current step and the next step in the sequence (i.e., the delay occurs after the write in which it was called). The total delay time per step (including execution) is defined below, giving a useful range of execution/delay times from 3.3 µs up to 1 s per step.

  If WSEQ_DELAYn = 0x0 or 0xF, the step execution time is 3.3 µs

  For all other values, the step execution time is 61.44 µs x $((2^{\text{WSEQ\_DELAY}}) – 1)$

  Setting this field to 0xF identifies the step as the last in the sequence
- WSEQ_DATA_START*n* is a 4-bit field that identifies the LSB position within the selected control register to which the data should be written. For example, setting WSEQ_DATA_START*n* = 0100 selects bit [4] as the LSB position of the data to be written.
- WSEQ_DATA*n* is an 8-bit field that contains the data to be written to the selected control register. The WSEQ_DATA_WIDTH*n* field determines how many of these bits are written to the selected control register; the most significant bits (above the number indicated by WSEQ_DATA_WIDTH*n*) are ignored.

The extended instruction set for the write sequencer is accessed by setting WSEQ_MODE*n* (bit [28]) in the respective sequencer definition register. The extended instruction set comprises the following functions:

- If bits [31:24] = 0x11, the register base address is set equal to the value contained in bits [23:0].
- If bits [31:16] = 0x12FF, the sequencer performs an unconditional jump to the index location defined in bits [15:0]. The index location is valid in the range 0 to 507 (0x1FB).
- All other settings within the extended instruction set are reserved.

The control field definitions for Step 0 are described in Table 4-121. The equivalent definitions also apply to Step 1 through Step 507, in the subsequent register address locations.

Table 4-121. Write Sequencer Control—Programming a Sequence

Register Address	Bit	Label	Default	Description
R12288 (0x3000) WSEQ_ Sequence_1	31:29	WSEQ_DATA_ WIDTH0[2:0]	000	Width of the data block written in this sequence step. 000 = 1 bit; 001 = 2 bits; 010 = 3 bits; 011 = 4 bits; 100 = 5 bits; 101 = 6 bits; 110 = 7 bits; 111 = 8 bits
	28	WSEQ_MODE0	0	Extended Sequencer Instruction select 0 = Basic instruction set 1 = Extended instruction set
	27:16	WSEQ_ADDR0[11:0]	0x000	Control Register Address to be written to in this sequence step. The register address is calculated as: (Base Address * 512) + WSEQ_ADDRn. Base Address is 0x00_0000 by default, and is configured using the sequencer's extended instruction set.
	15:12	WSEQ_DELAY0[3:0]	0000	Time delay after executing this step. 0x0 = 3.3 µs 0x1 to 0xE = 61.44 µs x $((2^{WSEQ\_DELAY})–1)$ 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 508 write operations; these are indexed as 0 to 507 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–OUT3 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 CS42L92 memory reset conditions.

The default control sequences can be overwritten in the sequencer memory, if required. Note that the headphone/earpiece output path enable bits (HP*nx*_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 4-122. 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 4-123. 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 4-114 through Table 4-118.

Table 4-123. Write Sequencer Memory Allocation

Description	Sequence Index Range
Default Sequences	0 to 302
User Space	303 to 383
Boot Sequence	384 to 507

4.19 Charge Pumps, Regulators, and Voltage Reference

The CS42L92 incorporates two charge-pump circuits and an LDO-regulator circuit to generate supply rails for internal functions and to support external microphone requirements. The CS42L92 also provides two MICBIAS generators (with six switchable outputs), which provide low noise reference voltages suitable for biasing ECM-type microphones or powering digital microphones.

In Sleep Mode, the CS42L92 can provide an unregulated voltage output that can be used to power an external microphone. See Section 4.13 for details of Sleep Mode.

The CPVDD 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. Refer to Section 5.1 for recommended external components.

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 CS42L92 when required by the output drivers.

The Charge Pump 1 circuit is shown in Fig. 4-75.

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.

If 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.

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

Note: The 32-kHz clock must be configured and enabled if CP2 is enabled in its normal operating mode. The 32-kHz clock is not required in bypass mode (CP2_BYPASS = 1). See Section 4.16 for details of the system clocks.

If CP2 is disabled, the CP2VOUT pin can be either floating or actively discharged. The behavior is configured using the CP2_DISCH bit.

If LDO2 is disabled, the MICVDD pin can be either floating or actively discharged. The behavior is configured 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 CS42L92.

The Charge Pump 2 and LDO2 Regulator circuits are shown in Fig. 4-75. The associated control bits are described in Table 4-124.

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. 4-75.

Switchable outputs from the MICBIAS generators allow six 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 the MICB1*x*_ENA and MICB2*x*_ENA (where *x* is A, B, C, or D).

The MICBIAS1 generator supports four switchable outputs (MICBIAS1A–MICBIAS1D). The MICBIAS2 generator supports two switchable outputs (MICBIAS2A–MICBIAS2B).

Note that, to enable any of the MICBIAS*nx* 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 MICB*nx*_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 200 mV 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. 4-75. The MICBIAS control fields are described in Table 4-124.

The maximum output current for each MICBIAS regulator is noted in Table 3-11. This limit must be observed for each set 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.

4.19.3.1 MICBIAS output in Sleep Mode

The CS42L92 supports a low-power Sleep Mode, in which most functions are disabled and power consumption is minimized. The CS42L92 can maintain an unregulated voltage output in Sleep Mode, suitable for powering an external microphone in always-on applications. The INnx/DMICx connections are isolated from the CS42L92 circuits in Sleep Mode, to allow any connected microphone to be used by another circuit while Sleep Mode is selected.

If the MICB1A_AOD_ENA bit is set, the normal register-field configuration of CP2, LDO2, and MICBIAS1 are overridden. Under these conditions, the CP2, LDO2 and MICBIAS1 circuits are bypassed and the MICBIAS1A output is connected directly to the CPVDD supply.

The unregulated MICBIAS1A output can be maintained in Sleep Mode by setting MICB1A_AOD_ENA before the DCVDD supply is removed.

To minimize any transient effects during the Sleep Mode transition, the MICBIAS1A should be configured in bypass mode before Sleep Mode is enabled. The following control sequence is recommended:

- CP_BYPASS = 1
- MICB1_BYPASS = 1
- MICB1A_AOD_ENA = 1
- Remove DCVDD

To minimize any transient effects following a wake-up transition, the MICBIAS1A should be configured in bypass mode before clearing MICB1A_AOD_ENA. The following control sequence is recommended:

- Apply DCVDD
- LDO2_VSEL = (set as required)
- CP2_BYPASS = 0
- MICB1_LVL = (set as required)
- MICB1_BYPASS = 0
- MICB1_ENA = 1
- MICB1A_ENA = 1
- MICB1A_AOD_ENA = 0

4.19.4 Voltage-Reference Circuit

The CS42L92 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. 4-75. Note that decoupling capacitors and flyback capacitors are required for these circuits. Refer to Section 5.1 for recommended external components.

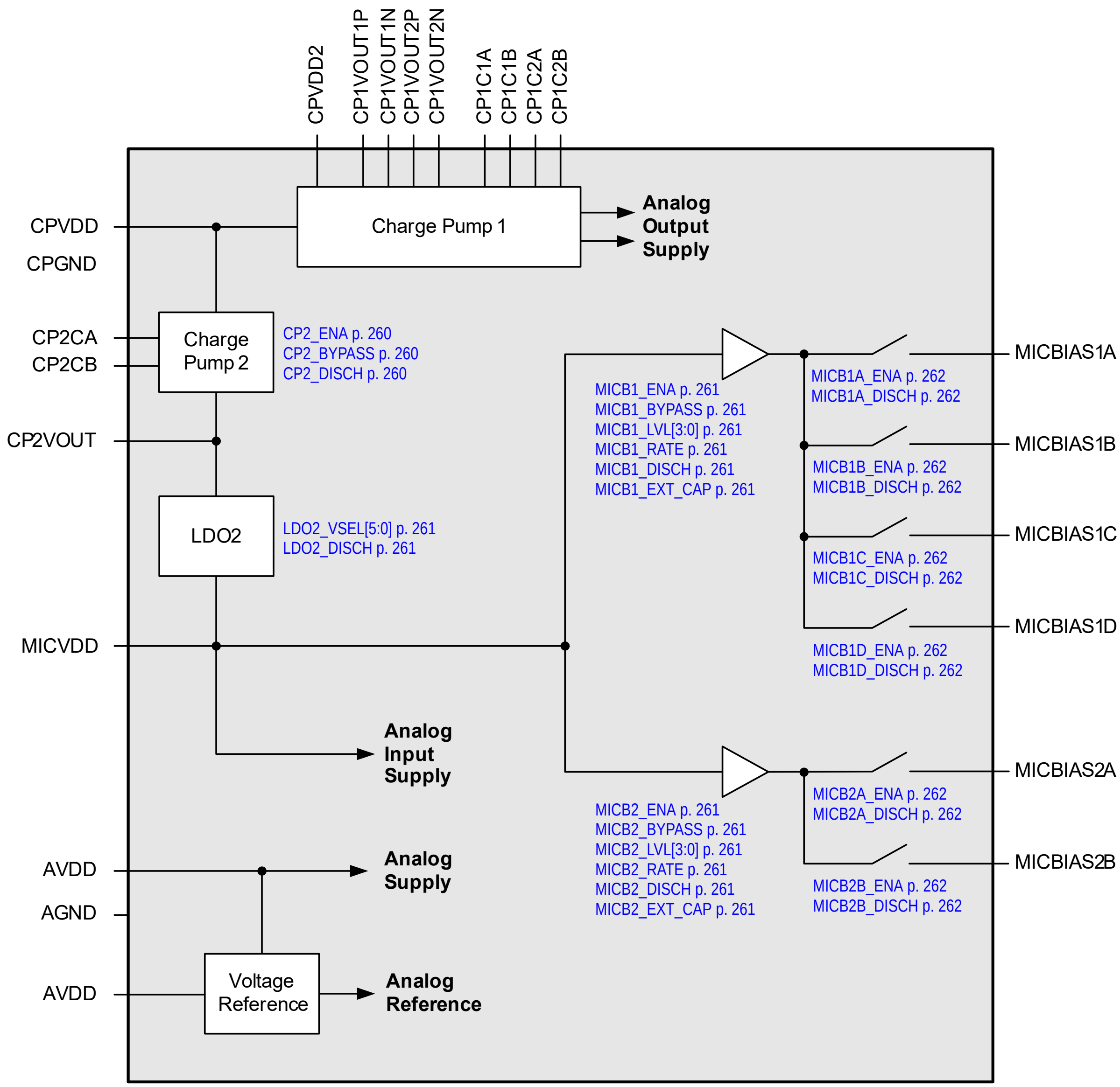


Figure 4-75. Charge Pumps and Regulators

The charge-pump and regulator control registers are described in Table 4-124.

Table 4-124. 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, CPVDD 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

Table 4-124. Charge-Pump and LDO Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
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
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

Table 4-124. Charge-Pump and LDO Control Registers *(Cont.)*

Register Address	Bit	Label	Default	Description
R540 (0x021C) Mic_Bias_Ctrl_5	13	MICB1D_DISCH	1	Microphone Bias 1D Discharge 0 = MICBIAS1D floating when disabled 1 = MICBIAS1D discharged when disabled
	12	MICB1D_ENA	0	Microphone Bias 1D Enable 0 = Disabled 1 = Enabled
	9	MICB1C_DISCH	1	Microphone Bias 1C Discharge 0 = MICBIAS1C floating when disabled 1 = MICBIAS1C discharged when disabled
	8	MICB1C_ENA	0	Microphone Bias 1C Enable 0 = Disabled 1 = Enabled
	5	MICB1B_DISCH	1	Microphone Bias 1B Discharge 0 = MICBIAS1B floating when disabled 1 = MICBIAS1B discharged when disabled
	4	MICB1B_ENA	0	Microphone Bias 1B Enable 0 = Disabled 1 = Enabled
	1	MICB1A_DISCH	1	Microphone Bias 1A Discharge 0 = MICBIAS1A floating when disabled 1 = MICBIAS1A discharged when disabled
	0	MICB1A_ENA	0	Microphone Bias 1A Enable 0 = Disabled 1 = Enabled
R542 (0x021E) Mic_Bias_Ctrl_6	5	MICB2B_DISCH	1	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	1	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
R723 (0x02D3) Jack_detect_ analogue	14	MICB1A_AOD_ ENA	0	Microphone Bias 1A Always-On Enable 0 = Disabled 1 = Enabled

1.See Table 4-125 for LDO2 output voltage definition.

Table 4-125 lists the LDO2 voltage control settings.

Table 4-125. LDO2 Voltage Control

LDO2_VSEL[5:0]	LDO Output	LDO2_VSEL[5:0]	LDO Output
0x00	0.900 V	0x15	1.500 V
0x01	0.925 V	0x16	1.600 V
0x02	0.950 V	0x17	1.700 V
0x03	0.975 V	0x18	1.800 V
0x04	1.000 V	0x19	1.900 V
0x05	1.025 V	0x1A	2.000 V
0x06	1.050 V	0x1B	2.100 V
0x07	1.075 V	0x1C	2.200 V
0x08	1.100 V	0x1D	2.300 V
0x09	1.125 V	0x1E	2.400V
0x0A	1.150 V	0x1F	2.500 V

Table 4-125. LDO2 Voltage Control *(Cont.)*

LDO2_VSEL[5:0]	LDO Output	LDO2_VSEL[5:0]	LDO Output
0x0B	1.175 V	0x20	2.600 V
0x0C	1.200 V	0x21	2.700 V
0x0D	1.225 V	0x22	2.800 V
0x0E	1.250 V	0x23	2.900 V
0x0F	1.275 V	0x24	3.000 V
0x10	1.300 V	0x25	3.100 V
0x11	1.325 V	0x26	3.200 V
0x12	1.350 V	0x27	3.300 V
0x13	1.375 V	0x28 to 0x3F	3.300 V
0x14	1.400 V		

4.20 JTAG Interface

The JTAG interface provides test and debug access to the CS42L92. The interface comprises five connections, some of which are multiplexed with the GPIO13–GPIO15 pins, as noted in Table 4-126.

Table 4-126. JTAG Interface Connections

Pin No	Pin Name	JTAG Function	JTAG Description
N8	AIF3BCLK/GPIO14	TCK	Clock input
R8	AIF3RXDAT/GPIO15	TDI	Data input
F7	TDO	TDO	Data output
J8	AIF3TXDAT/GPIO13	TMS	Mode select input
D7	TRST	TRST	Test access port reset input (active low)

The JTAG interface is supported whenever the GPIO13–GPIO15 are configured as GPIO inputs. To allow JTAG operation, the following conditions must be met:

- GP*n*_FN = 0x001 (*n* = 13–15)
- GP*n*_DIR = 0x1 (*n* = 13–15)

The GP*n*_FN and GP*n*_DIR fields are defined in Table 4-93. Note that the above conditions are also the default values of these control fields.

For normal operation (test and debug access disabled), the JTAG interface should be held in reset. An internal pull-down resistor holds the TRST pin low (i.e., JTAG interface is held in reset) when not actively driven. It is recommended to connect the TRST pin to DGND, if the JTAG interface function is not required.

Integrated pull-up and pull-down resistors can be enabled on the TCK, TDI, and TMS pins. This is provided as part of the GPIO functionality, and provides a flexible capability for interfacing with other devices. The pull-up and pull-down resistors can be configured independently using the fields described in Table 4-93.

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 CS42L92 software reset.

4.21 Short-Circuit Protection

The CS42L92 provides short-circuit protection on the headphone output drivers.

The short-circuit protection function for the headphone output paths operates continuously if the respective output driver is enabled. If a short circuit is detected on the headphone 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 short-circuit protection function provides input to the interrupt control circuit and can be used to trigger an interrupt event when a short-circuit condition is detected; see Section 4.15.

4.22 Power-On Reset (POR)

The CS42L92 remains in the reset state until AVDD, DBVDD, and DCVDD are above their respective reset thresholds. Note that specified device performance is not assured outside the voltage ranges defined in Table 3-3.

After the initial power-up, the POR is rescheduled following an interruption to the DBVDD or AVDD supplies.

If the CS42L92 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 4-127. 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 $\overline{\text{IRQ}}$ pin indicates completion of the boot sequence.

For details of the boot sequence, see Section 4.18.

Table 4-127. 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 1-1 describes the default status of the CS42L92 digital I/O pins on completion of power-on reset, prior to 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 $\overline{\text{IRQ}}$ and $\overline{\text{RESET}}$ pins. These are always-on pins whose configuration is unchanged in Sleep Mode and during a wake-up transition.

Note that the default conditions described in Table 1-1 are not valid if modified by the boot sequence or by a wake-up control sequence. See Section 4.18 for details of these functions.

4.23 Hardware Reset, Software Reset, Wake-Up, and Device ID

The CS42L92 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 CS42L92 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 CS42L92 provides a hardware reset function, which is executed whenever the RESET input is asserted (Logic 0). The RESET input is active low and is referenced to the DBVDD power domain. A hardware reset causes all of the CS42L92 control registers to be reset to their default states.

An internal pull-up resistor is enabled by default on the RESET pin; this can be configured using the RESET_PU bit. A pull-down resistor is also available, as described in Table 4-128. When the pull-up and pull-down resistors are both enabled, the CS42L92 provides a bus keeper function on the 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 CS42L92 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 4-128. Reset Pull-Up/Pull-Down Configuration

Register Address	Bit	Label	Default	Description
R6864 (0x1AD0) AOD_Pad_Ctrl	1	RESET_PU	1	RESET Pull-up enable 0 = Disabled 1 = Enabled **Note:** If RESET_PD and RESET_PU are both set, a bus keeper function is enabled on the RESET pin.
	0	RESET_PD	0	RESET Pull-down enable 0 = Disabled 1 = Enabled **Note:** If RESET_PD and RESET_PU are both set, a bus keeper function is enabled on the RESET pin.

4.23.2 Software Reset

A software reset is executed by writing any value to register R0. A software reset causes most of the CS42L92 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 CS42L92 is in Sleep Mode when AVDD and DBVDD are present, and DCVDD is below its reset threshold. (Note that specific control requirements are also applicable for entering Sleep Mode, as described in Section 4.13.)

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 CS42L92 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 4-127) 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_STS*x* has been asserted.

The BOOT_DONE_STS*x* 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 CS42L92 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 4-129. Device Reset and ID

Register Address	Bit	Label	Default	Description
R0 (0x0000) Software_Reset	15:0	SW_RST_DEV_ID[15:0]	0x6371	Writing to this register resets all registers to their default state. Reading from this register indicates Device ID 0x6371.
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 CS42L92.

5.1.1 Analog Input Paths

The CS42L92 supports up to eight analog audio input connections. Each input 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.

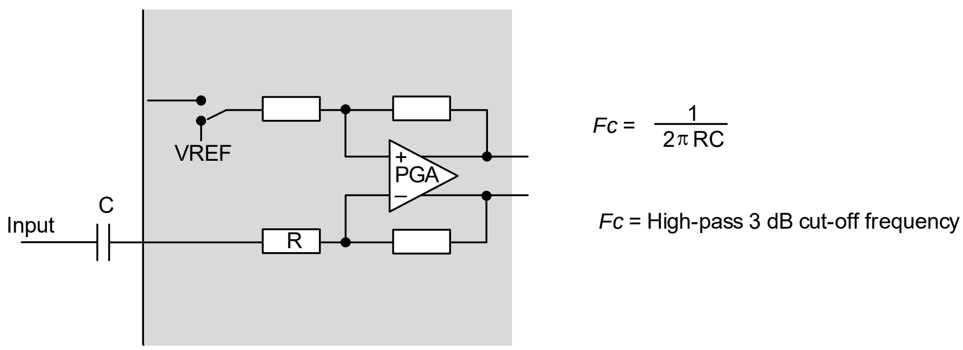


Figure 5-1. Audio Input Path DC-Blocking Capacitor

In accordance with the CS42L92 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 CS42L92 microphone bias circuit, are shown in Fig. 5-2.

5.1.2 DMIC Input Paths

The CS42L92 supports up to eight channels of DMIC input; two channels of audio data can be multiplexed on each DMICDAT*n* pin. Each stereo pair is clocked using the respective DMICCLK*n* pin.

The external connections for digital microphones, incorporating the CS42L92 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 CS42L92 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 CS42L92 interface is compatible with the applicable configuration of the external microphone.

5.1.3 Microphone Bias Circuit

The CS42L92 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 CS42L92. Two MICBIAS generators are available; switchable outputs allow six 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 CS42L92 is not exceeded.

A 2.2-kΩ bias resistor is recommended; this provides compatibility with a wide range of microphone components.

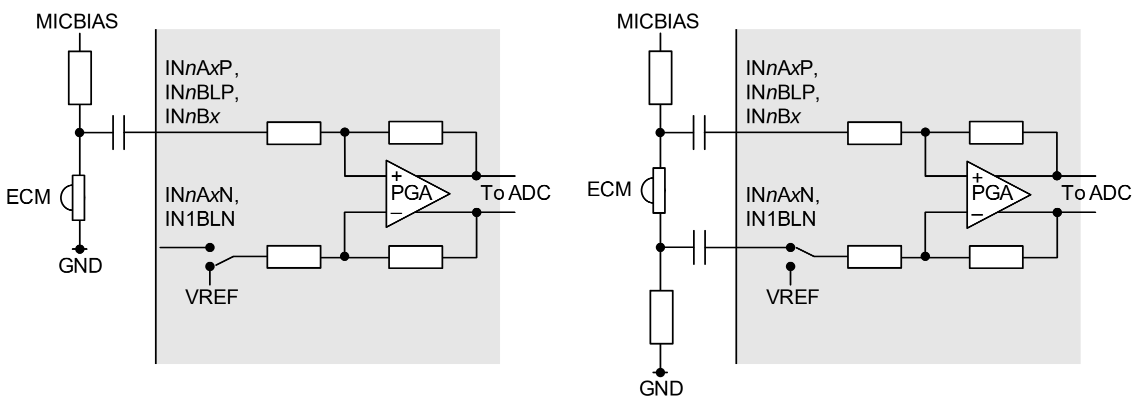


Figure 5-2. Single-Ended and Differential Analog Microphone Connections

Analog MEMS microphones can be connected to the CS42L92 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.

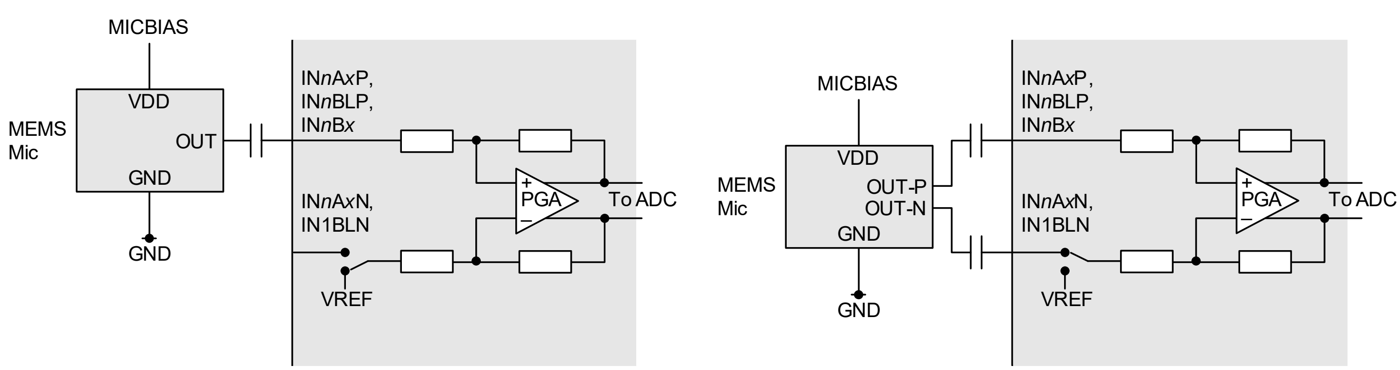


Figure 5-3. Single-Ended and Differential Analog Microphone Connections

DMIC connection to the CS42L92 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.

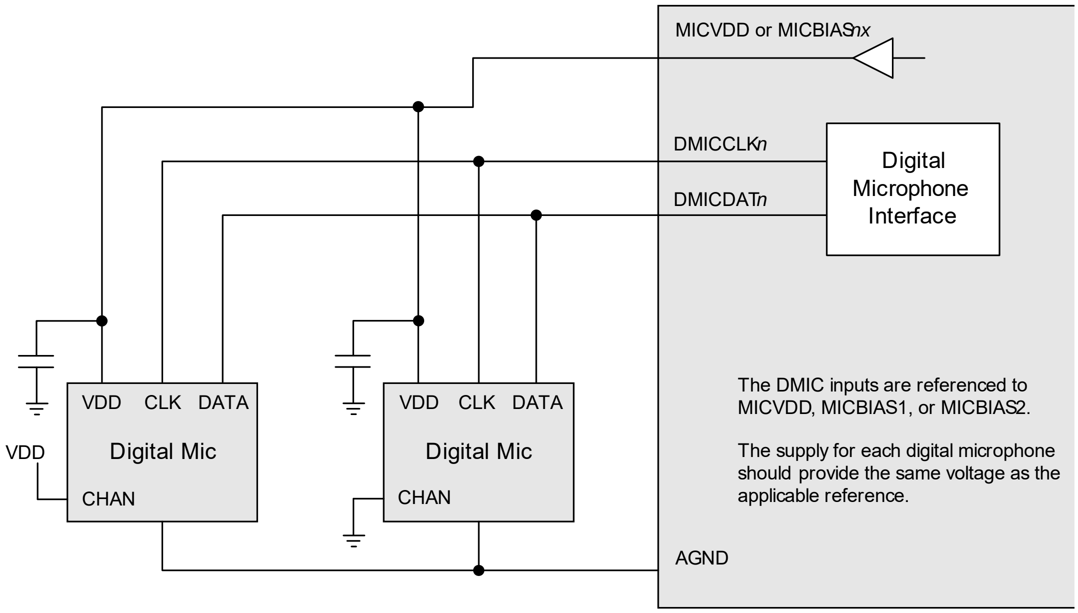


Figure 5-4. DMIC Connection

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 set 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 Driver Output Path

The CS42L92 provides four stereo headphone output drivers. These outputs are all ground referenced, allowing direct connection to the external loads. There is no requirement for DC-blocking capacitors.

In single-ended (default) configuration, the headphone outputs comprise eight independently controlled output channels, for up to four stereo headphone or line outputs. In mono (BTL) mode, the headphone drivers support up to four differential outputs, suitable for a mono earpiece or hearing coil load.

Note that the OUT3 signal path is common to the HPOUT3 and HPOUT4 drivers—only one of these stereo drivers may be enabled at any time.

The headphone outputs incorporate a common mode, or ground loop, feedback path that provides rejection of system-related ground noise. The feedback pins must be connected to ground for normal operation of the headphone outputs. Five feedback pins are provided (HPOUTFB1–HPOUTFB5). The ground feedback path for each HPOUT path is selected using the HP*n*_GND_SEL bits as follows:

- The ground feedback path for HPOUT1 and HPOUT2 headphone outputs is selected using the HP1_GND_SEL and HP2_GND_SEL register fields respectively—see Table 4-81.
- The ground feedback path for HPOUT3 and HPOUT4 headphone outputs is selected using the HP3_GND_SEL field.

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.

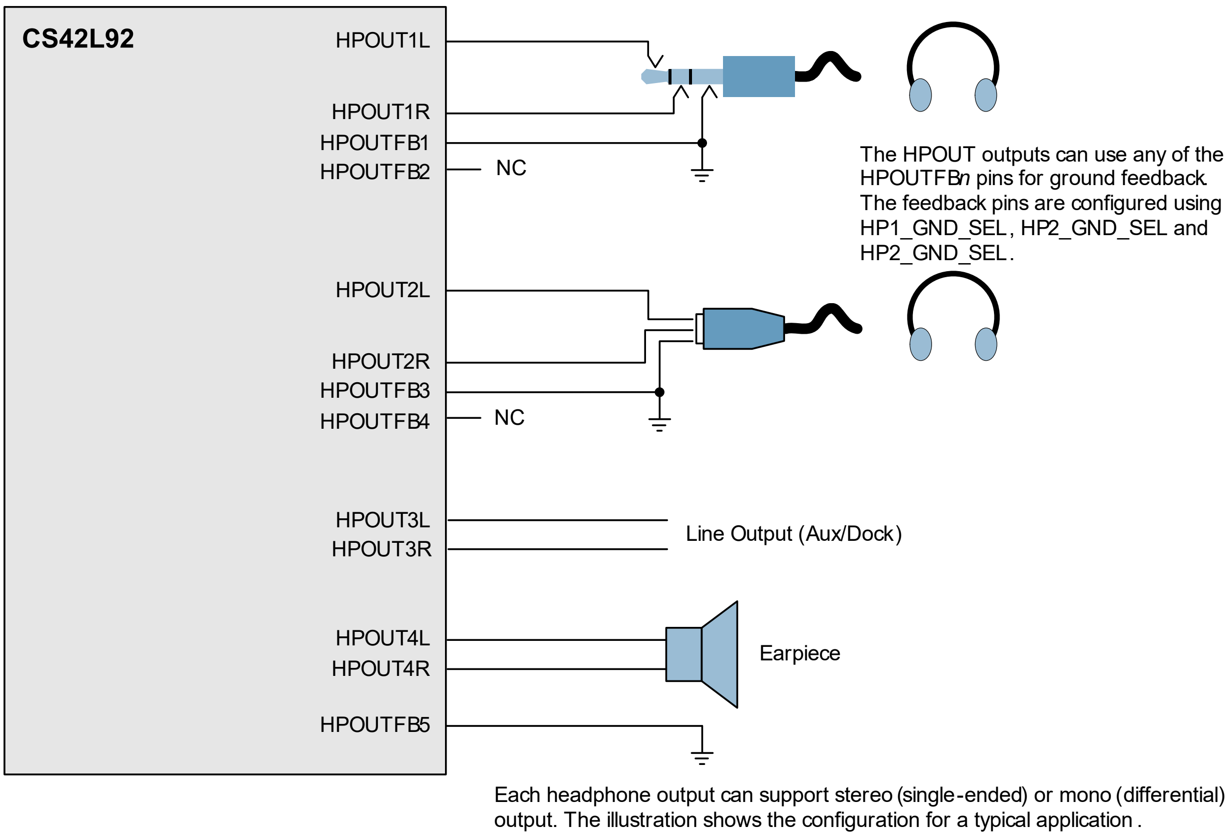


Figure 5-5. Headphone and Earpiece Connection

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 would be recommended if the headphone paths (HPOUT1–HPOUT4) are used for external headphone or line output.

The HPOUT*n* 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 CS42L92 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.

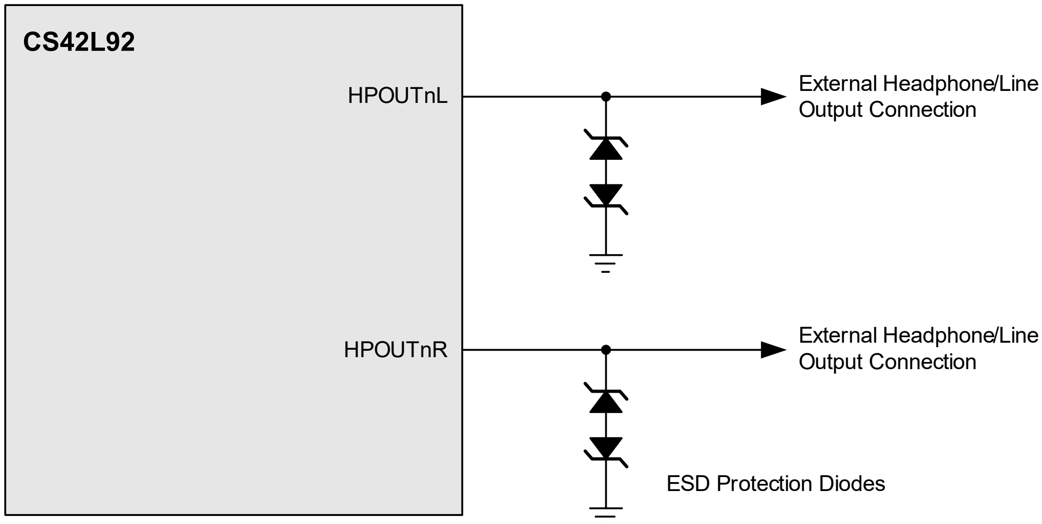


Figure 5-6. ESD Diode Configuration for External Output Connections

5.1.5 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 CS42L92, 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 CS42L92 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
DBVDD	1 x 0.1 µF ceramic [1]
DCVDD	2 x 1.0 µF ceramic—one capacitor on each DCVDD pin
FLLVDD	4.7 µF ceramic
MICVDD	4.7 µF ceramic
VREFC	2.2 µF ceramic

1. Total capacitance of 4.7 µF is required for the DBVDD domain. This can be provided by dedicated DBVDD decoupling or by other capacitors on the same power rail.

All decoupling capacitors should be placed as close as possible to the CS42L92 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 CS42L92.

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 X7R is recommended.

5.1.6 Charge-Pump Components

The CS42L92 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 CS42L92 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 X7R 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 CS42L92. 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.

5.1.7 External Accessory Detection Components

The external accessory detection circuit measures jack insertion using the JACKDET*n* 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 JACKDET*n* pins 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 MICDET*n* 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 HPOUT1 and microphone connections, is shown in Fig. 5-7. 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. 5-7.

Note that, when using the microphone detect circuit, it is recommended to use the IN1AL*x*, IN1BL*x*, or IN1BR analog microphone input paths to ensure best immunity to electrical transients arising from the external accessory.

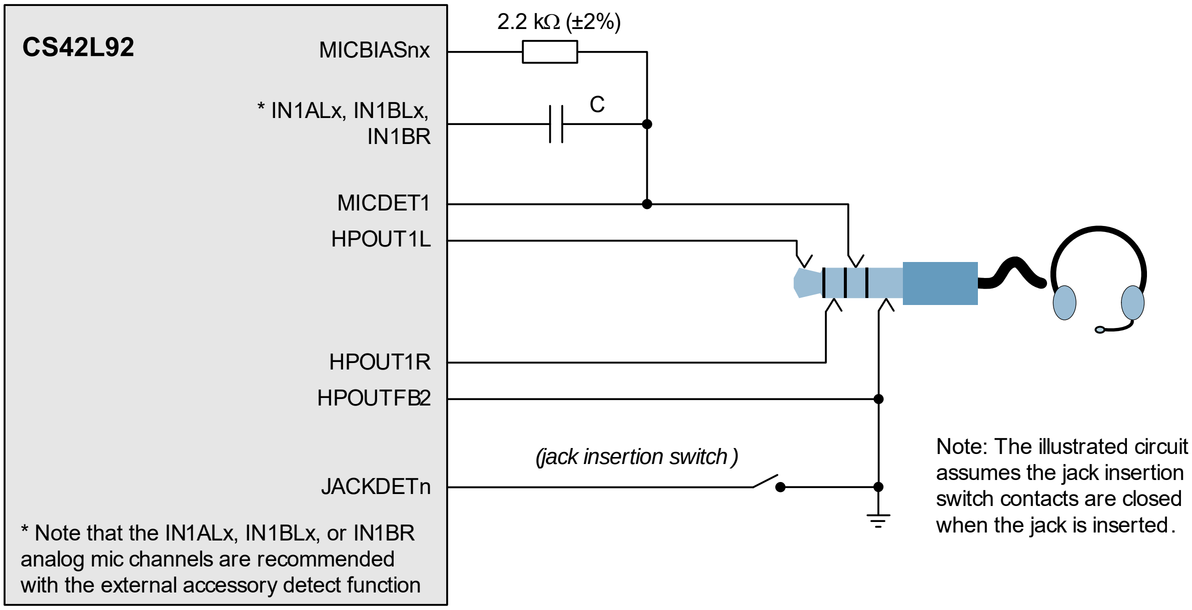


Figure 5-7. External Accessory Detection

The accessory detection circuit measures the impedance of an external load connected to one of the MICDET pins.

The microphone-detection circuit uses MICVDD, or any one of the MICBIAS*nx* sources, as a reference. The applicable source is configured using MICD*n*_BIAS_SRC.

With default register configuration, the CS42L92 can detect the presence of a typical microphone and up to four push buttons, using the components shown in Fig. 5-8. When the microphone detection circuit is enabled, each of the push buttons shown causes a different bit in the MICD*n*_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. 5-8 are examples only, assuming default impedance measurement ranges and a microphone impedance of 1 kΩ or higher.

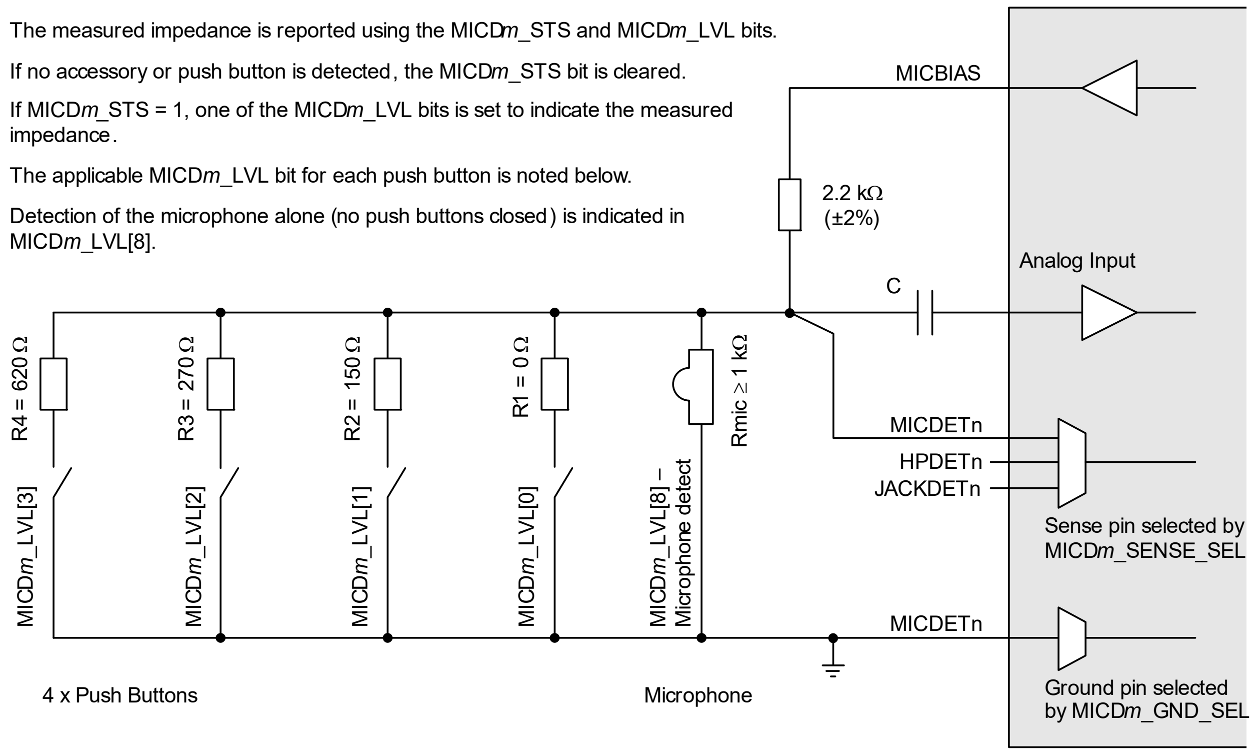


Figure 5-8. External Accessory Detect Components

5.2 Resets Summary

Table 5-3 summarizes the CS42L92 registers and other programmable memory under different reset conditions. The associated events and conditions are listed as follows:

- A power-on reset occurs when AVDD or DBVDD is below its respective reset threshold. Note that DCVDD is also required for initial start-up; subsequent interruption to DCVDD should only be permitted as part of a control sequence for entering Sleep Mode.
- A hardware reset occurs when the $\overline{\text{RESET}}$ input is asserted (Logic 0).
- A software reset occurs when register R0 is written to.
- Sleep Mode is selected when DCVDD is removed. Note that the AVDD and DBVDD supplies must be present throughout the Sleep Mode duration.

Table 5-3. Memory Reset Summary

Reset Type	Always-On Registers [1]	Other Registers	Control-Write Sequencer Memory	DSP Firmware Memory
Power-on reset	Reset	Reset	Reset	Reset
Hardware reset	Reset	Reset	Reset	Reset
Software reset	Reset	Reset	Retained	Retained [2]
Sleep Mode	Retained	Reset	Reset	Reset

1. See Section 4.13 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 CS42L92 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
R1520 (0x05F0) Slimbus_Pad_Ctrl	1	SLIMDAT2_DRV_ STR	1	SLIMDAT2 output drive strength 0 = 8 mA 1 = 12 mA
	1	SLIMDAT1_DRV_ STR	1	SLIMDAT1 output drive strength 0 = 8 mA 1 = 12 mA
	0	SLIMCLK_DRV_ STR	1	SLIMCLK output drive strength 0 = 2 mA 1 = 4 mA
R5889 (0x1701) GPIO1_CTRL2	12	GP1_DRV_STR	1	GPIO1 output drive strength 0 = 4 mA 1 = 8 mA
R5891 (0x1703) GPIO2_CTRL2	12	GP2_DRV_STR	1	GPIO2 output drive strength 0 = 4 mA 1 = 8 mA
R5893 (0x1705) GPIO3_CTRL2	12	GP3_DRV_STR	1	SPKCLK/GPIO3 output drive strength 0 = 4 mA 1 = 8 mA
R5895 (0x1707) GPIO4_CTRL2	12	GP4_DRV_STR	1	SPKDAT/GPIO4 output drive strength 0 = 4 mA 1 = 8 mA
R5897 (0x1709) GPIO5_CTRL2	12	GP5_DRV_STR	1	AIF1TXDAT/GPIO5 output drive strength 0 = 4 mA 1 = 8 mA
R5899 (0x170B) GPIO6_CTRL2	12	GP6_DRV_STR	1	AIF1BCLK/GPIO6 output drive strength 0 = 4 mA 1 = 8 mA
R5901 (0x170D) GPIO7_CTRL2	12	GP7_DRV_STR	1	AIF1RXDAT/GPIO7 output drive strength 0 = 4 mA 1 = 8 mA
R5903 (0x170F) GPIO8_CTRL2	12	GP8_DRV_STR	1	AIF1LRCLK/GPIO8 output drive strength 0 = 4 mA 1 = 8 mA
R5905 (0x1711) GPIO9_CTRL2	12	GP9_DRV_STR	1	AIF2TXDAT/GPIO9 output drive strength 0 = 4 mA 1 = 8 mA
R5907 (0x1713) GPIO10_CTRL2	12	GP10_DRV_STR	1	AIF2BCLK/GPIO10 output drive strength 0 = 4 mA 1 = 8 mA
R5909 (0x1715) GPIO11_CTRL2	12	GP11_DRV_STR	1	AIF2RXDAT/GPIO11 output drive strength 0 = 4 mA 1 = 8 mA
R5911 (0x1717) GPIO12_CTRL2	12	GP12_DRV_STR	1	AIF2LRCLK/GPIO12 output drive strength 0 = 4 mA 1 = 8 mA
R5913 (0x1719) GPIO13_CTRL2	12	GP13_DRV_STR	1	AIF3TXDAT/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	AIF3BCLK/GPIO14 output drive strength 0 = 4 mA 1 = 8 mA
R5917 (0x171D) GPIO15_CTRL2	12	GP15_DRV_STR	1	AIF3RXDAT/GPIO15 output drive strength 0 = 4 mA 1 = 8 mA
R5919 (0x171F) GPIO16_CTRL2	12	GP16_DRV_STR	1	AIF3LRCLK/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 (or ASYNCCLK, where applicable).

In AIF Master Mode, the external BCLK and LRCLK signals are generated by the CS42L92 and synchronization of these signals with SYSCLK (or ASYNCCLK) is ensured. In this case, clocking of the AIF is typically derived from the MCLK*n* inputs, either directly or via one of the FLL circuits. 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 CS42L92. In this case, the system clock (SYSCLK or ASYNCCLK) 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, MCLK2, or MCLK3 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 one of the FLLs, 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.

The applicable system clock (SYSCLK or ASYNCCLK) depends on the AIFn_RATE setting for the relevant digital audio interface; if AIFn_RATE < 1000, SYSCLK is applicable; if AIFn_RATE ≥ 1000, ASYNCCLK is applicable.

Table 5-5. AIF Clocking Configurations

AIF Mode	Clocking Configuration
AIF Master Mode	SYSCLK_SRC (ASYNCCLK_SRC) selects MCLK1, MCLK2, or MCLK3 as SYSCLK (ASYNCCLK) source.
	SYSCLK_SRC (ASYNCCLK_SRC) selects FLL*n* as SYSCLK (ASYNCCLK) source; FLL*n*_REFCLK_SRC selects MCLK1, MCLK2, or MCLK3 as FLL*n* source.
	SYSCLK_SRC (ASYNCCLK_SRC) selects FLL*n* as SYSCLK (ASYNCCLK) source; FLL*n*_REFCLK_SRC selects a different interface (BCLK, LRCLK, SLIMCLK) as FLL*n* source.
AIF Slave Mode	SYSCLK_SRC (ASYNCCLK_SRC) selects FLL*n* as SYSCLK (ASYNCCLK) source; FLL*n*_REFCLK_SRC selects BCLK as FLL*n* source.
	SYSCLK_SRC (ASYNCCLK_SRC) selects MCLK1, MCLK2, or MCLK3 as SYSCLK (ASYNCCLK) source, provided MCLK is externally synchronized to the BCLK input.
	SYSCLK_SRC (ASYNCCLK_SRC) selects FLL*n* as SYSCLK (ASYNCCLK) source; FLL*n*_REFCLK_SRC selects MCLK1, MCLK2, or MCLK3 as FLL*n* source, provided MCLK is externally synchronized to the BCLK input.
	SYSCLK_SRC (ASYNCCLK_SRC) selects FLL*n* as SYSCLK (ASYNCCLK) source; FLL*n*_REFCLK_SRC selects a different interface (e.g., SLIMCLK) as FLL*n* source, provided the other interface is externally synchronized to the BCLK input.

In each case, the SYSCLK (ASYNCCLK) frequency must be a valid ratio to the LRCLK frequency; the supported clocking rates are defined by the SYSCLK_FREQ (ASYNC_CLK_FREQ) and SAMPLE_RATE_*n* (ASYNC_SAMPLE_RATE_*n*) fields.

The valid AIF clocking configurations are shown in Fig. 5-9 to Fig. 5-15. Note that, where MCLK1 is shown as the clock source, it is equally possible to select MCLK2 or MCLK3 as the clock source. Similarly, in cases where FLL1 is shown, it is equally possible to select FLL2.

Fig. 5-9 shows AIF Master Mode operation, using MCLK as the clock reference.

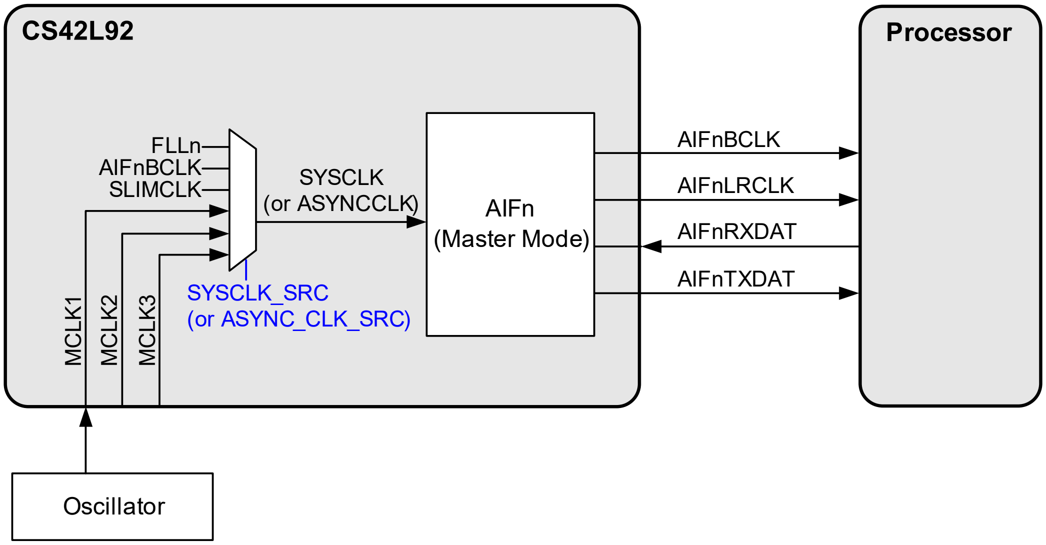


Figure 5-9. AIF Master Mode, Using MCLK as Reference

Fig. 5-10 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.

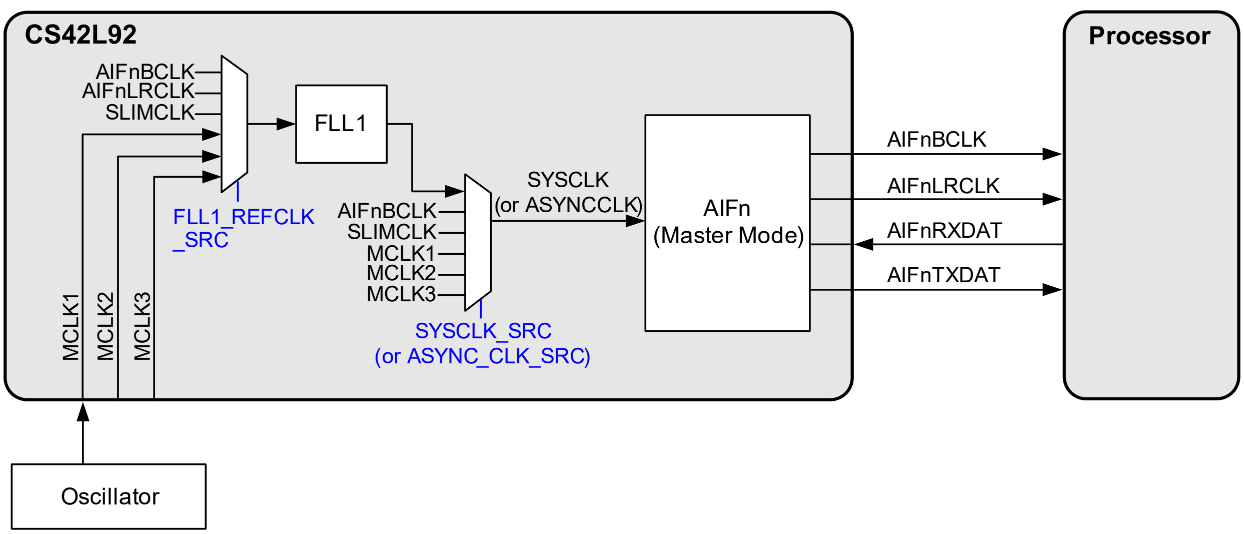


Figure 5-10. AIF Master Mode, Using MCLK and FLL as Reference

Fig. 5-11 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.

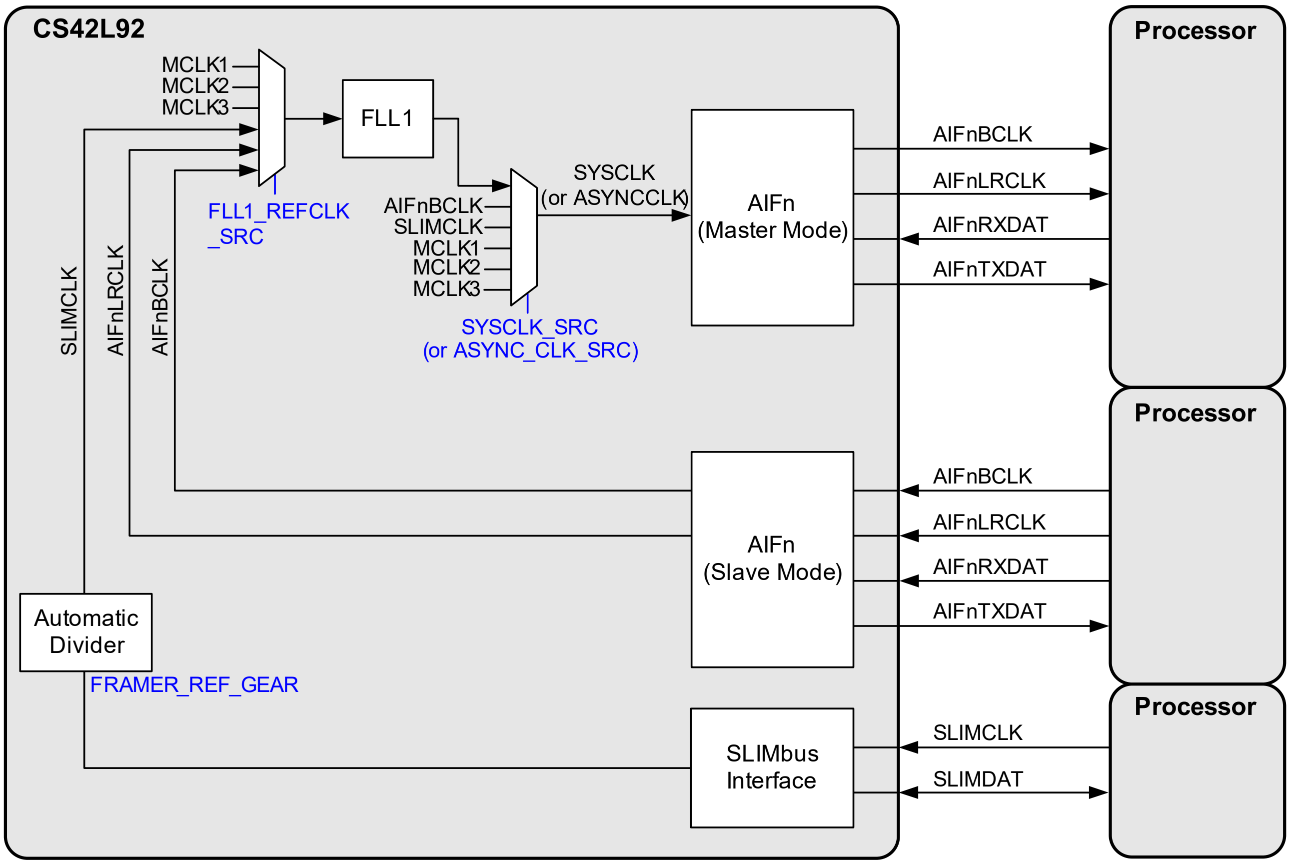


Figure 5-11. AIF Master Mode, Using Another Interface as Reference

Fig. 5-12 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.

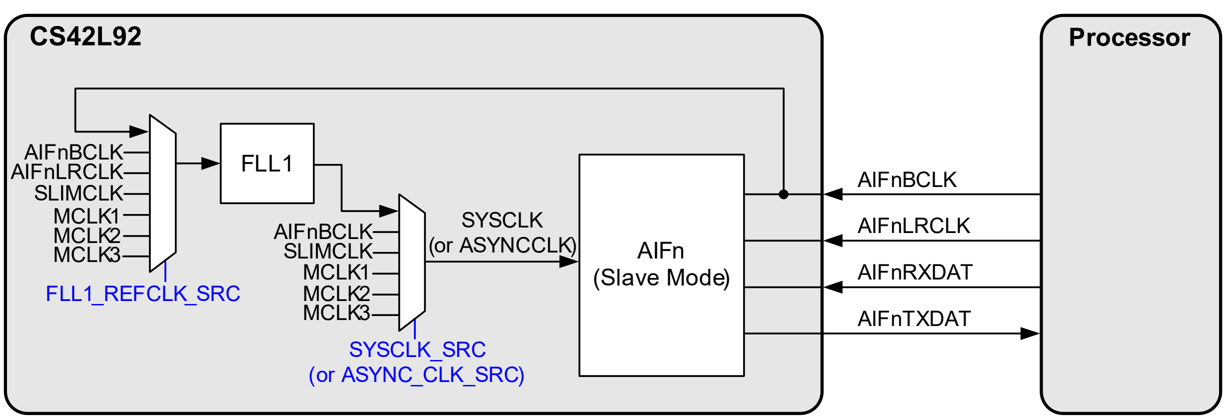


Figure 5-12. AIF Slave Mode, Using BCLK and FLL as Reference

Fig. 5-13 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.

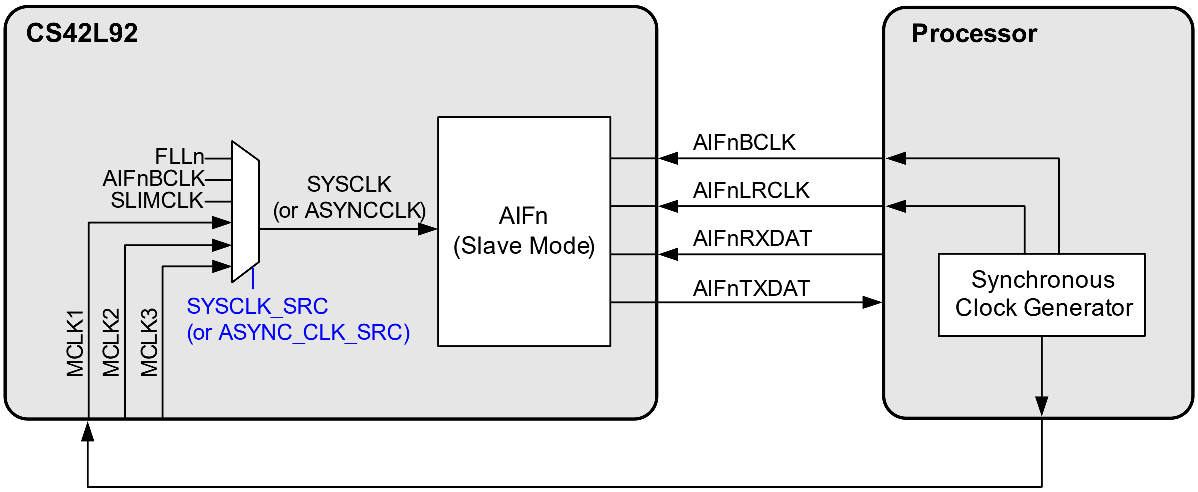


Figure 5-13. AIF Slave Mode, Using MCLK as Reference

Fig. 5-14 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.

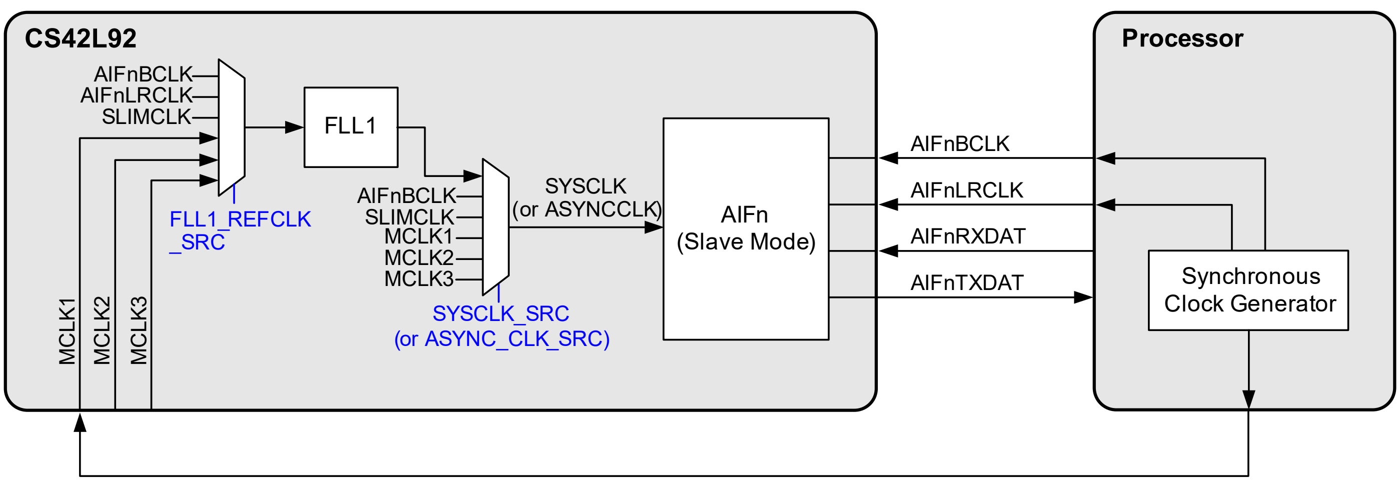


Figure 5-14. AIF Slave Mode, Using MCLK and FLL as Reference

Fig. 5-15 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.

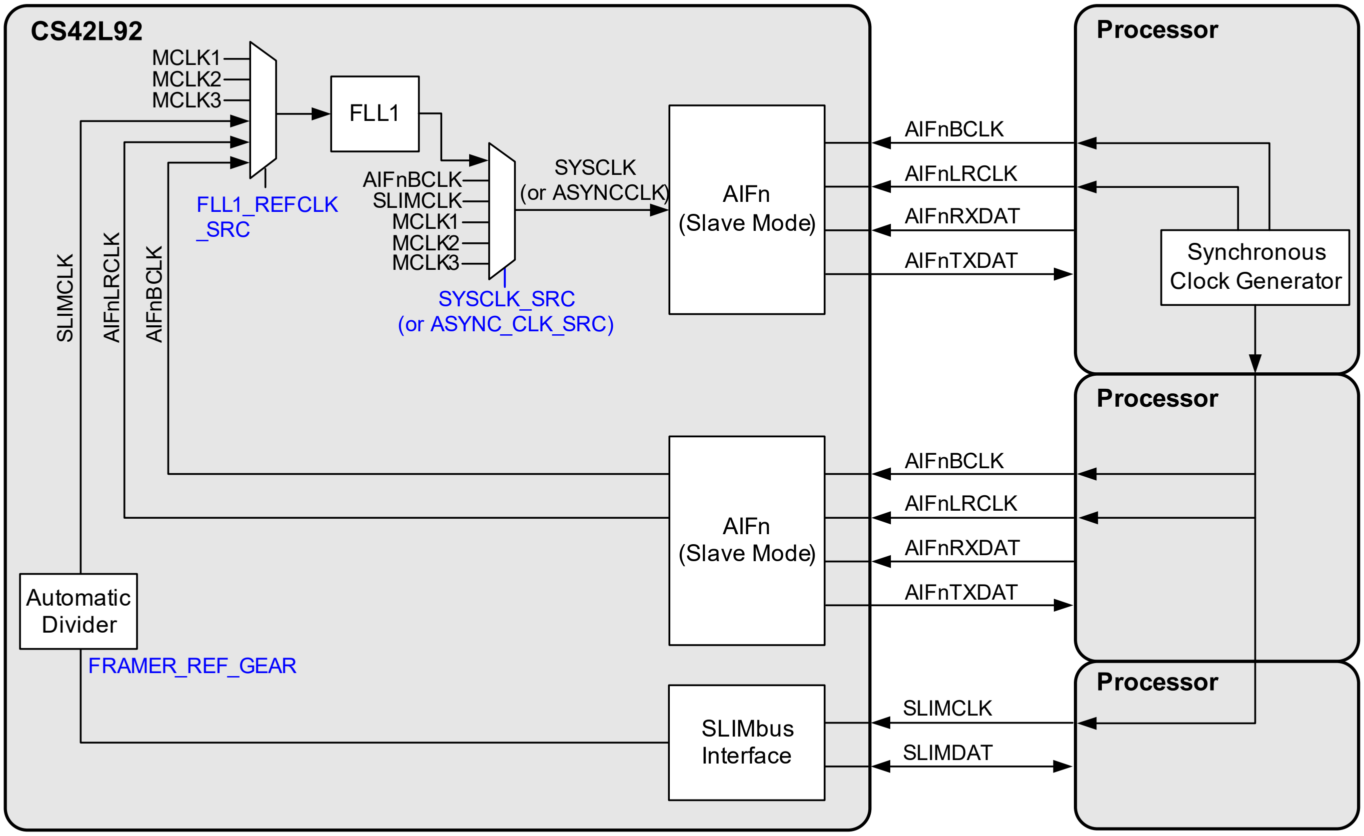


Figure 5-15. AIF Slave Mode, Using Another Interface as Reference

5.5 PCB Layout Considerations

PCB layout should be carefully considered, to ensure optimum performance of the CS42L92. Poor PCB layout degrades the performance and is a contributory factor in EMI, ground bounce, and resistive voltage losses. All external components should be placed close to the CS42L92, with current loop areas kept as small as possible. The following specific considerations should be noted:

- Placement of the charge pump capacitors is a high priority requirement—these capacitors (particularly the fly-back capacitors) must be placed as close as possible to the CS42L92. 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.
- Decoupling capacitors should be placed as close as possible to the CS42L92. 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 CS42L92.
- The VREFC capacitor should be placed as close as possible to the CS42L92. The ground connection to the VREFC capacitor should be as close as possible to the AGND1 ball of the CS42L92.
- If multiple power supply rails are connected to a single supply source, it is recommended to provide separate PCB tracks connecting each rail to the supply. This configuration is also known as *star connection*.
- If power supply rails are routed between different layers of the PCB, it is recommended to use several track vias, in order to minimize resistive voltage losses.
- Differential input signal tracks should be routed as a pair, ensuring similar length/width dimensions on each track. Input signal paths should be kept away from high frequency digital signals.

- Differential output signal tracks should be routed as a pair, ensuring similar length/width dimensions on each track. The tracks should provide a low resistance path from the device output pin to the load (< 1% of the minimum load).
- The headphone output ground-feedback pins should be connected to GND as close as possible to the respective headphone jack ground pin. The ground-feedback PCB track should follow the same route as the respective output signal paths.

6 Register Map

The CS42L92 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 CS42L92 register map is defined in two regions:

- The register space below 0x3000 is defined in 16-bit word format
- The 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]																6371h
R1 (1h)	Hardware_Revision	0	0	0	0	0	0	0	0	HW_REVISION [7:0]								0001h
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
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	TONE_RATE [4:0]					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	PWM_RATE [4:0]					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
R66 (42h)R65 (41h)	Spare_Triggers	WSEQ_TRG16	WSEQ_TRG15	WSEQ_TRG14	WSEQ_TRG13	WSEQ_TRG12	WSEQ_TRG11	WSEQ_TRG10	WSEQ_TRG9	WSEQ_TRG8	WSEQ_TRG7	WSEQ_TRG6	WSEQ_TRG5	WSEQ_TRG4	WSEQ_TRG3	WSEQ_TRG2	WSEQ_TRG1	0000h
R75 (4Bh)	Spare_Sequence_Select_1	0	0	0	0	0	0	0	WSEQ_TRG1_INDEX [8:0]									01FFh
R76 (4Ch)	Spare_Sequence_Select_2	0	0	0	0	0	0	0	WSEQ_TRG2_INDEX [8:0]									01FFh
R77 (4Dh)	Spare_Sequence_Select_3	0	0	0	0	0	0	0	WSEQ_TRG3_INDEX [8:0]									01FFh
R78 (4Eh)	Spare_Sequence_Select_4	0	0	0	0	0	0	0	WSEQ_TRG4_INDEX [8:0]									01FFh
R79 (4Fh)	Spare_Sequence_Select_5	0	0	0	0	0	0	0	WSEQ_TRG5_INDEX [8:0]									01FFh
R80 (50h)	Spare_Sequence_Select_6	0	0	0	0	0	0	0	WSEQ_TRG6_INDEX [8:0]									01FFh

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R89 (59h)	Spare_Sequence_Select_7	0	0	0	0	0	0	0	WSEQ_TRG7_INDEX [8:0]									01FFh
R90 (5Ah)	Spare_Sequence_Select_8	0	0	0	0	0	0	0	WSEQ_TRG8_INDEX [8:0]									01FFh
R91 (5Bh)	Spare_Sequence_Select_9	0	0	0	0	0	0	0	WSEQ_TRG9_INDEX [8:0]									01FFh
R92 (5Ch)	Spare_Sequence_Select_10	0	0	0	0	0	0	0	WSEQ_TRG10_INDEX [8:0]									01FFh
R93 (5Dh)	Spare_Sequence_Select_11	0	0	0	0	0	0	0	WSEQ_TRG11_INDEX [8:0]									01FFh
R94 (5Eh)	Spare_Sequence_Select_12	0	0	0	0	0	0	0	WSEQ_TRG12_INDEX [8:0]									01FFh
R97 (61h)	Sample_Rate_Sequence_Select_1	0	0	0	0	0	0	0	WSEQ_SAMPLE_RATE_DETECT_A_INDEX [8:0]									01FFh
R98 (62h)	Sample_Rate_Sequence_Select_2	0	0	0	0	0	0	0	WSEQ_SAMPLE_RATE_DETECT_B_INDEX [8:0]									01FFh
R99 (63h)	Sample_Rate_Sequence_Select_3	0	0	0	0	0	0	0	WSEQ_SAMPLE_RATE_DETECT_C_INDEX [8:0]									01FFh
R100 (64h)	Sample_Rate_Sequence_Select_4	0	0	0	0	0	0	0	WSEQ_SAMPLE_RATE_DETECT_D_INDEX [8:0]									01FFh
R102 (66h)	Always_On_Triggers_Sequence_Select_1	0	0	0	0	0	0	0	WSEQ_MICD_CLAMP_RISE_INDEX [8:0]									01FFh
R103 (67h)	Always_On_Triggers_Sequence_Select_2	0	0	0	0	0	0	0	WSEQ_MICD_CLAMP_FALL_INDEX [8:0]									01FFh
R104 (68h)	Spare_Sequence_Select_13	0	0	0	0	0	0	0	WSEQ_TRG13_INDEX [8:0]									01FFh
R105 (69h)	Spare_Sequence_Select_14	0	0	0	0	0	0	0	WSEQ_TRG14_INDEX [8:0]									01FFh
R106 (6Ah)	Spare_Sequence_Select_15	0	0	0	0	0	0	0	WSEQ_TRG15_INDEX [8:0]									01FFh
R107 (6Bh)	Spare_Sequence_Select_16	0	0	0	0	0	0	0	WSEQ_TRG16_INDEX [8:0]									01FFh
R110 (6Eh)	Trigger_Sequence_Select_32	0	0	0	0	0	0	0	WSEQ_DRC1_SIG_DET_RISE_INDEX [8:0]									01FFh
R111 (6Fh)	Trigger_Sequence_Select_33	0	0	0	0	0	0	0	WSEQ_DRC1_SIG_DET_FALL_INDEX [8:0]									01FFh
R120 (78h)	Eventlog_Sequence_Select_1	0	0	0	0	0	0	0	WSEQ_EVENTLOG1_INDEX [8:0]									01FFh
R144 (90h)	Haptics_Control_1	HAP_RATE [4:0]					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	NOISE_GEN_RATE [4:0]					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
R274 (112h)	Async_clock_1	0	0	0	0	0	ASYNC_CLK_FREQ [2:0]			0	ASYNC_CLK_ENA	0	0	ASYNC_CLK_SRC [3:0]				0305h

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
R275 (113h)	Async_sample_rate_1	0	0	0	0	0	0	0	0	0	0	0	ASYNC_SAMPLE_RATE_1 [4:0]					0011h
R276 (114h)	Async_sample_rate_2	0	0	0	0	0	0	0	0	0	0	0	ASYNC_SAMPLE_RATE_2 [4:0]					0011h
R283 (11Bh)	Async_sample_rate_1_status	0	0	0	0	0	0	0	0	0	0	0	ASYNC_SAMPLE_RATE_1_STS [4:0]					0000h
R284 (11Ch)	Async_sample_rate_2_status	0	0	0	0	0	0	0	0	0	0	0	ASYNC_SAMPLE_RATE_2_STS [4:0]					0000h
R329 (149h)	Output_system_clock	OPCLK_ENA	0	0	0	0	0	0	0	OPCLK_DIV [4:0]					OPCLK_SEL [2:0]			0000h
R330 (14Ah)	Output_async_clock	OPCLK_ASYNC_ENA	0	0	0	0	0	0	0	OPCLK_ASYNC_DIV [4:0]					OPCLK_ASYNC_SEL [2:0]			0000h
R334 (14Eh)	Clock_Gen_Pad_Ctrl	0	0	0	0	0	0	MCLK3_PD	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
R352 (160h)	Clocking_debug_5	ASYNC_CLK_FREQ_STS [2:0]			ASYNC_CLK_SRC_STS [3:0]				0	0	SYSCLK_FREQ_STS [2:0]			SYSCLK_SRC_STS [3:0]				0000h
R369 (171h)	FLL1_Control_1	FLL1_REFCLK_SRC [3:0]				0	0	0	0	0	0	0	0	0	FLL1_HOLD	0	FLL1_ENA	7004h
R370 (172h)	FLL1_Control_2	FLL1_CTRL_UPD	0	0	0	0	0	FLL1_N [9:0]										0004h
R371 (173h)	FLL1_Control_3	FLL1_THETA [15:0]																0000h
R372 (174h)	FLL1_Control_4	FLL1_LAMBDA [15:0]																0000h
R373 (175h)	FLL1_Control_5	0	0	0	0	0	0	FLL1_FB_DIV [9:0]										0001h
R374 (176h)	FLL1_Control_6	FLL1_REFDET	0	0	0	0	0	0	0	FLL1_REFCLK_DIV [1:0]		0	0	0	0	0	0	8000h
R376 (178h)	FLL1_Control_8	FLL1_PD_GAIN_FINE [3:0]				FLL1_PD_GAIN_COARSE [3:0]				FLL1_FD_GAIN_FINE [3:0]				FLL1_FD_GAIN_COARSE [3:0]				21F0h
R378 (17Ah)	FLL1_Control_10	FLL1_HP [1:0]		0	1	0	0	0	0	0	0	0	0	0	0	0	0	1000h
R379 (17Bh)	FLL1_Control_11	0	0	0	0	0	0	0	0	0	0	0	FLL1_LOCKDET_THR [3:0]				FLL1_LOCKDET	0011h
R381 (17Dh)	FLL1_Digital_Test_1	0	0	1	1	0	0	1	1	0	1	1	1	1	0	FLL1_CLK_VCO_FAST_SRC[1:0]		33E8h
R398 (18Eh)	FLL1_GPIO_Clock	0	0	0	0	FLL1_GPDIV_SRC [1:0]		0	0	FLL1_GPCLK_DIV [6:0]							FLL1_GPCLK_ENA	0C04h
R401 (191h)	FLL2_Control_1	FLL2_REFCLK_SRC [3:0]				0	0	0	0	0	0	0	0	0	FLL2_HOLD	0	FLL2_ENA	7000h
R402 (192h)	FLL2_Control_2	FLL2_CTRL_UPD	0	0	0	0	0	FLL2_N [9:0]										0004h
R403 (193h)	FLL2_Control_3	FLL2_THETA [15:0]																0000h
R404 (194h)	FLL2_Control_4	FLL2_LAMBDA [15:0]																0000h
R405 (195h)	FLL2_Control_5	0	0	0	0	0	0	FLL2_FB_DIV [9:0]										0001h
R406 (196h)	FLL2_Control_6	FLL2_REFDET	0	0	0	0	0	0	0	FLL2_REFCLK_DIV [1:0]		0	0	0	0	0	0	8000h
R408 (198h)	FLL2_Control_8	FLL2_PD_GAIN_FINE [3:0]				FLL2_PD_GAIN_COARSE [3:0]				FLL2_FD_GAIN_FINE [3:0]				FLL2_FD_GAIN_COARSE [3:0]				21F0h
R410 (19Ah)	FLL2_Control_10	FLL2_HP [1:0]		0	1	0	0	0	0	0	0	0	0	0	0	0	0	1000h
R411 (19Bh)	FLL2_Control_11	0	0	0	0	0	0	0	0	0	0	0	FLL2_LOCKDET_THR [3:0]				FLL2_LOCKDET	0011h
R413 (19Dh)	FLL2_Digital_Test_1	0	0	1	1	0	0	1	1	0	1	1	1	1	0	FLL2_CLK_VCO_FAST_SRC[1:0]		33E8h
R430 (1AEh)	FLL2_GPIO_Clock	0	0	0	0	FLL2_GPDIV_SRC [1:0]		0	0	FLL2_GPCLK_DIV [6:0]							FLL2_GPCLK_ENA	0C04h
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

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R536 (218h)	Mic_Bias_Ctrl_1	MICB1_EXT_CAP	0	0	0	0	0	0	MICB1_LVL [3:0]				0	MICB1_RATE	MICB1_DISCH	MICB1_BYPASS	MICB1_ENA	00E6h
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	MICB1D_DISCH	MICB1D_ENA	0	0	MICB1C_DISCH	MICB1C_ENA	0	0	MICB1B_DISCH	MICB1B_ENA	0	0	MICB1A_DISCH	MICB1A_ENA	2222h
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
R665 (299h)	Headphone_Detect_0	HPD_OVD_ENA	HPD_OUT_SEL [2:0]			HPD_FRC_SEL [3:0]				HPD_SENSE_SEL [3:0]				0	HPD_GND_SEL [2:0]			0000h
R667 (29Bh)	Headphone_Detect_1	0	0	0	0	0	HPD_IMPEDANCE_RANGE [1:0]		0	0	0	0	HPD_CLK_DIV [1:0]		HPD_RATE [1:0]		HPD_POLL (M)	0000h
R668 (29Ch)	Headphone_Detect_2	HPD_DONE	HPD_LVL [14:0]															0000h
R669 (29Dh)	Headphone_Detect_3	0	0	0	0	0	0	HPD_DACVAL [9:0]										0000h
R674 (2A2h)	Mic_Detect_1_Control_0	MICD1_ADC_MODE	0	0	0	0	0	0	0	MICD1_SENSE_SEL [3:0]				0	MICD1_GND_SEL [2:0]			0010h
R675 (2A3h)	Mic_Detect_1_Control_1	MICD1_BIAS_STARTTIME [3:0]				MICD1_RATE [3:0]				MICD1_BIAS_SRC [3:0]				0	0	MICD1_DBTIME	MICD1_ENA	1102h
R676 (2A4h)	Mic_Detect_1_Control_2	0	0	0	0	0	0	0	0	MICD1_LVL_SEL [7:0]								009Fh
R677 (2A5h)	Mic_Detect_1_Control_3	0	0	0	0	0	MICD1_LVL [8:0]									MICD1_VALID	MICD1_STS	0000h
R683 (2ABh)	Mic_Detect_1_Control_4	MICD1_ADCVAL_DIFF [7:0]								0	MICD1_ADCVAL [6:0]							0000h
R690 (2B2h)	Mic_Detect_2_Control_0	MICD2_ADC_MODE	0	0	0	0	0	0	0	MICD2_SENSE_SEL [3:0]				0	MICD2_GND_SEL [2:0]			0010h
R691 (2B3h)	Mic_Detect_2_Control_1	MICD2_BIAS_STARTTIME [3:0]				MICD2_RATE [3:0]				MICD2_BIAS_SRC [3:0]				0	0	MICD2_DBTIME	MICD2_ENA	1102h
R692 (2B4h)	Mic_Detect_2_Control_2	0	0	0	0	0	0	0	0	MICD2_LVL_SEL [7:0]								009Fh
R693 (2B5h)	Mic_Detect_2_Control_3	0	0	0	0	0	MICD2_LVL [8:0]									MICD2_VALID	MICD2_STS	0000h
R699 (2BBh)	Mic_Detect_2_Control_4	MICD2_ADCVAL_DIFF [7:0]								0	MICD2_ADCVAL [6:0]							0000h
R710 (2C6h)	Micd_Clamp_control	0	0	0	0	0	0	MICD_CLAMP2_OVD	MICD_CLAMP2_MODE [2:0]			0	MICD_CLAMP1_OVD	MICD_CLAMP1_MODE [3:0]				0210h
R712 (2C8h)	GP_Switch_1	0	0	0	0	0	0	0	0	0	0	0	0	SW2_MODE [1:0]		SW1_MODE [1:0]		0000h
R723 (2D3h)	Jack_detect_analogue	0	MICB1A_AOD_ENA	0	0	0	0	0	0	0	0	0	0	0	JD3_ENA	JD2_ENA	JD1_ENA	0000h
R768 (300h)	Input_Enables	0	0	0	0	0	0	0	0	IN4L_ENA	IN4R_ENA	IN3L_ENA	IN3R_ENA	IN2L_ENA	IN2R_ENA	IN1L_ENA	IN1R_ENA	0000h
R769 (301h)	Input_Enables_Status	0	0	0	0	0	0	0	0	IN4L_ENA_STS	IN4R_ENA_STS	IN3L_ENA_STS	IN3R_ENA_STS	IN2L_ENA_STS	IN2R_ENA_STS	IN1L_ENA_STS	IN1R_ENA_STS	0000h
R776 (308h)	Input_Rate	IN_RATE [4:0]					IN_RATE_MODE	0	0	0	0	0	0	0	0	0	0	0400h
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	IN1L_LP_MODE	0	IN_VU	IN1L_MUTE	IN1L_VOL [7:0]								0180h
R786 (312h)	DMIC1L_Control	IN1L_SIG_DET_ENA	0	0	0	0	IN1_OSR [2:0]			0	0	0	0	0	0	0	0	0500h
R787 (313h)	IN1L_Rate_Control	IN1L_RATE [4:0]					0	0	0	0	0	0	0	0	0	0	0	0000h
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	IN1R_LP_MODE	0	IN_VU	IN1R_MUTE	IN1R_VOL [7:0]								0180h
R790 (316h)	DMIC1R_Control	IN1R_SIG_DET_ENA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h
R791 (317h)	IN1R_Rate_Control	IN1R_RATE [4:0]					0	0	0	0	0	0	0	0	0	0	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	IN2L_LP_MODE	0	IN_VU	IN2L_MUTE	IN2L_VOL [7:0]								0180h
R794 (31Ah)	DMIC2L_Control	IN2L_SIG_DET_ENA	0	0	0	0	IN2_OSR [2:0]			0	0	0	0	0	0	0	0	0500h

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
R795 (31Bh)	IN2L_Rate_Control	IN2L_RATE [4:0]					0	0	0	0	0	0	0	0	0	0	0	0000h
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	IN2R_LP_MODE	0	IN_VU	IN2R_MUTE	IN2R_VOL [7:0]								0180h
R798 (31Eh)	DMIC2R_Control	IN2R_SIG_DET_ENA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h
R799 (31Fh)	IN2R_Rate_Control	IN2R_RATE [4:0]					0	0	0	0	0	0	0	0	0	0	0	0000h
R800 (320h)	IN3L_Control	IN3L_HPF	0	0	IN3_DMIC_SUP [1:0]		0	0	0	0	0	0	0	0	0	0	0	0000h
R801 (321h)	ADC_Digital_Volume_3L	0	0	0	0	0	0	IN_VU	IN3L_MUTE	IN3L_VOL [7:0]								0180h
R802 (322h)	DMIC3L_Control	IN3L_SIG_DET_ENA	0	0	0	0	IN3_OSR [2:0]			0	0	0	0	0	0	0	0	0500h
R803 (323h)	IN3L_Rate_Control	IN3L_RATE [4:0]					0	0	0	0	0	0	0	0	0	0	0	0000h
R804 (324h)	IN3R_Control	IN3R_HPF	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h
R805 (325h)	ADC_Digital_Volume_3R	0	0	0	0	0	0	IN_VU	IN3R_MUTE	IN3R_VOL [7:0]								0180h
R806 (326h)	DMIC3R_Control	IN3R_SIG_DET_ENA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h
R807 (327h)	IN3R_Rate_Control	IN3R_RATE [4:0]					0	0	0	0	0	0	0	0	0	0	0	0000h
R808 (328h)	IN4L_Control	IN4L_HPF	0	0	IN4_DMIC_SUP [1:0]		0	0	0	0	0	0	0	0	0	0	0	0000h
R809 (329h)	ADC_Digital_Volume_4L	0	0	0	0	0	0	IN_VU	IN4L_MUTE	IN4L_VOL [7:0]								0180h
R810 (32Ah)	DMIC4L_Control	IN4L_SIG_DET_ENA	0	0	0	0	IN4_OSR [2:0]			0	0	0	0	0	0	0	0	0500h
R811 (32Bh)	IN4L_Rate_Control	IN4L_RATE [4:0]					0	0	0	0	0	0	0	0	0	0	0	0000h
R812 (32Ch)	IN4R_Control	IN4R_HPF	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h
R813 (32Dh)	ADC_Digital_Volume_4R	0	0	0	0	0	0	IN_VU	IN4R_MUTE	IN4R_VOL [7:0]								0180h
R814 (32Eh)	DMIC4R_Control	IN4R_SIG_DET_ENA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h
R815 (32Fh)	IN4R_Rate_Control	IN4R_RATE [4:0]					0	0	0	0	0	0	0	0	0	0	0	0000h
R832 (340h)	Signal_Detect_Globals	0	0	0	0	0	0	0	IN_SIG_DET_THR [4:0]					IN_SIG_DET_HOLD [3:0]				0001h
R840 (348h)	Dig_Mic_Pad_Ctrl	0	0	0	0	0	0	0	0	0	0	0	0	DMICDAT4_PD	DMICDAT3_PD	DMICDAT2_PD	DMICDAT1_PD	0000h
R1024 (400h)	Output_Enables_1	EP_SEL	0	0	0	0	0	OUT5L_ENA	OUT5R_ENA	0	0	HP3L_ENA	HP3R_ENA	HP2L_ENA	HP2R_ENA	HP1L_ENA	HP1R_ENA	0000h
R1025 (401h)	Output_Status_1	0	0	0	0	0	0	OUT5L_ENA_STS	OUT5R_ENA_STS	0	0	0	0	0	0	0	0	0000h
R1030 (406h)	Raw_Output_Status_1	0	0	0	0	0	0	0	0	0	0	OUT3L_ENA_STS	OUT3R_ENA_STS	OUT2L_ENA_STS	OUT2R_ENA_STS	OUT1L_ENA_STS	OUT1R_ENA_STS	0000h
R1032 (408h)	Output_Rate_1	OUT_RATE [4:0]					0	0	0	0	CP_DAC_MODE	OUT_EXT_CLK_DIV [1:0]		0	OUT_CLK_SRC [2:0]			0040h
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	OUT1L_HIFI	0	OUT1_MONO	0	0	0	0	1	0	0	0	0	0	0	0	0080h
R1041 (411h)	DAC_Digital_Volume_1L	0	0	0	0	0	0	OUT_VU	OUT1L_MUTE	OUT1L_VOL [7:0]								0180h
R1042 (412h)	Output_Path_Config_1	0	0	0	0	0	0	0	0	0	0	0	0	0	HP1_GND_SEL [2:0]			0000h
R1043 (413h)	Noise_Gate_Select_1L	0	0	0	0	OUT1L_NGATE_SRC [11:0]												0001h
R1044 (414h)	Output_Path_Config_1R	0	OUT1R_HIFI	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0080h
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
R1048 (418h)	Output_Path_Config_2L	0	OUT2L_HIFI	0	OUT2_MONO	0	0	0	0	1	0	0	0	0	0	0	0	0080h
R1049 (419h)	DAC_Digital_Volume_2L	0	0	0	0	0	0	OUT_VU	OUT2L_MUTE	OUT2L_VOL [7:0]								0180h
R1050 (41Ah)	Output_Path_Config_2	0	0	0	0	0	0	0	0	0	0	0	0	0	HP2_GND_SEL [2:0]			0002h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R1051 (41Bh)	Noise_Gate_Select_2L	0	0	0	0	OUT2L_NGATE_SRC [11:0]												0004h
R1052 (41Ch)	Output_Path_Config_2R	0	OUT2R_HIFI	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0080h
R1053 (41Dh)	DAC_Digital_Volume_2R	0	0	0	0	0	0	OUT_VU	OUT2R_MUTE	OUT2R_VOL [7:0]								0180h
R1055 (41Fh)	Noise_Gate_Select_2R	0	0	0	0	OUT2R_NGATE_SRC [11:0]												0008h
R1056 (420h)	Output_Path_Config_3L	0	OUT3L_HIFI	0	OUT3_MONO	0	0	0	0	1	0	0	0	0	0	0	0	0080h
R1057 (421h)	DAC_Digital_Volume_3L	0	0	0	0	0	0	OUT_VU	OUT3L_MUTE	OUT3L_VOL [7:0]								0180h
R1058 (422h)	Output_Path_Config_3	0	0	0	0	0	0	0	0	0	0	0	0	0	HP3_GND_SEL [2:0]			0002h
R1059 (423h)	Noise_Gate_Select_3L	0	0	0	0	OUT3L_NGATE_SRC [11:0]												0010h
R1060 (424h)	Output_Path_Config_3R	0	OUT3R_HIFI	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0080h
R1061 (425h)	DAC_Digital_Volume_3R	0	0	0	0	0	0	OUT_VU	OUT3R_MUTE	OUT3R_VOL [7:0]								0180h
R1063 (427h)	Noise_Gate_Select_3R	0	0	0	0	OUT3R_NGATE_SRC [11:0]												0020h
R1072 (430h)	Output_Path_Config_5L	0	OUT5L_HIFI	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
R1076 (434h)	Output_Path_Config_5R	0	OUT5R_HIFI	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h
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
R1102 (44Eh)	Filter_Control	0	0	0	0	0	0	0	0	0	0	0	0	HIFI_FIR_TYPE [3:0]				0000h
R1104 (450h)	DAC_AEC_Control_1	0	0	0	0	0	0	0	0	0	0	AEC1_LOOPBACK_SRC [3:0]				AEC1_ENA_STS	AEC1_LOOPBACK_ENA	0000h
R1105 (451h)	DAC_AEC_Control_2	0	0	0	0	0	0	0	0	0	0	AEC2_LOOPBACK_SRC [3:0]				AEC2_ENA_STS	AEC2_LOOPBACK_ENA	0000h
R1112 (458h)	Noise_Gate_Control	0	0	0	0	0	0	0	0	0	0	NGATE_HOLD [1:0]		NGATE_THR [2:0]			NGATE_ENA	0000h
R1168 (490h)	PDM_SPK1_CTRL_1	0	0	SPK1R_MUTE	SPK1L_MUTE	0	0	0	SPK1_MUTE_ENDIAN	SPK1_MUTE_SEQ [7:0]								0069h
R1169 (491h)	PDM_SPK1_CTRL_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SPK1_FMT	0000h
R1280 (500h)	AIF1_BCLK_Ctrl	0	0	0	0	0	0	0	0	AIF1_BCLK_INV	AIF1_BCLK_FRC	AIF1_BCLK_MSTR	AIF1_BCLK_FREQ [4:0]					000Ch
R1281 (501h)	AIF1_Tx_Pin_Ctrl	0	0	0	0	0	0	0	0	0	0	AIF1TX_DAT_TRI	0	0	0	0	0	0000h
R1282 (502h)	AIF1_Rx_Pin_Ctrl	0	0	0	0	0	0	0	0	0	0	0	AIF1_LRCLK_MODE	0	AIF1_LRCLK_INV	AIF1_LRCLK_FRC	AIF1_LRCLK_MSTR	0000h
R1283 (503h)	AIF1_Rate_Ctrl	AIF1_RATE [4:0]					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

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
R1295 (50Fh)	AIF1_Frame_Ctrl_9	0	0	0	0	0	0	0	0	0	0	AIF1TX7_SLOT [5:0]						0006h
R1296 (510h)	AIF1_Frame_Ctrl_10	0	0	0	0	0	0	0	0	0	0	AIF1TX8_SLOT [5:0]						0007h
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
R1303 (517h)	AIF1_Frame_Ctrl_17	0	0	0	0	0	0	0	0	0	0	AIF1RX7_SLOT [5:0]						0006h
R1304 (518h)	AIF1_Frame_Ctrl_18	0	0	0	0	0	0	0	0	0	0	AIF1RX8_SLOT [5:0]						0007h
R1305 (519h)	AIF1_Tx_Enables	0	0	0	0	0	0	0	0	AIF1TX8_ENA	AIF1TX7_ENA	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	AIF1RX8_ENA	AIF1RX7_ENA	AIF1RX6_ENA	AIF1RX5_ENA	AIF1RX4_ENA	AIF1RX3_ENA	AIF1RX2_ENA	AIF1RX1_ENA	0000h
R1344 (540h)	AIF2_BCLK_Ctrl	0	0	0	0	0	0	0	0	AIF2_BCLK_INV	AIF2_BCLK_FRC	AIF2_BCLK_MSTR	AIF2_BCLK_FREQ [4:0]					000Ch
R1345 (541h)	AIF2_Tx_Pin_Ctrl	0	0	0	0	0	0	0	0	0	0	AIF2TX_DAT_TRI	0	0	0	0	0	0000h
R1346 (542h)	AIF2_Rx_Pin_Ctrl	0	0	0	0	0	0	0	0	0	0	0	AIF2_LRCLK_MODE	0	AIF2_LRCLK_INV	AIF2_LRCLK_FRC	AIF2_LRCLK_MSTR	0000h
R1347 (543h)	AIF2_Rate_Ctrl	AIF2_RATE [4:0]					0	0	0	0	AIF2_TRI	0	0	0	0	0	0	0000h
R1348 (544h)	AIF2_Format	0	0	0	0	0	0	0	0	0	0	0	0	0	AIF2_FMT [2:0]			0000h
R1350 (546h)	AIF2_Rx_BCLK_Rate	0	0	0	AIF2_BCPF [12:0]													0040h
R1351 (547h)	AIF2_Frame_Ctrl_1	0	0	AIF2TX_WL [5:0]						AIF2TX_SLOT_LEN [7:0]								1818h
R1352 (548h)	AIF2_Frame_Ctrl_2	0	0	AIF2RX_WL [5:0]						AIF2RX_SLOT_LEN [7:0]								1818h
R1353 (549h)	AIF2_Frame_Ctrl_3	0	0	0	0	0	0	0	0	0	0	AIF2TX1_SLOT [5:0]						0000h
R1354 (54Ah)	AIF2_Frame_Ctrl_4	0	0	0	0	0	0	0	0	0	0	AIF2TX2_SLOT [5:0]						0001h
R1355 (54Bh)	AIF2_Frame_Ctrl_5	0	0	0	0	0	0	0	0	0	0	AIF2TX3_SLOT [5:0]						0002h
R1356 (54Ch)	AIF2_Frame_Ctrl_6	0	0	0	0	0	0	0	0	0	0	AIF2TX4_SLOT [5:0]						0003h
R1357 (54Dh)	AIF2_Frame_Ctrl_7	0	0	0	0	0	0	0	0	0	0	AIF2TX5_SLOT [5:0]						0004h
R1358 (54Eh)	AIF2_Frame_Ctrl_8	0	0	0	0	0	0	0	0	0	0	AIF2TX6_SLOT [5:0]						0005h
R1359 (54Fh)	AIF2_Frame_Ctrl_9	0	0	0	0	0	0	0	0	0	0	AIF2TX7_SLOT [5:0]						0006h
R1360 (550h)	AIF2_Frame_Ctrl_10	0	0	0	0	0	0	0	0	0	0	AIF2TX8_SLOT [5:0]						0007h
R1361 (551h)	AIF2_Frame_Ctrl_11	0	0	0	0	0	0	0	0	0	0	AIF2RX1_SLOT [5:0]						0000h
R1362 (552h)	AIF2_Frame_Ctrl_12	0	0	0	0	0	0	0	0	0	0	AIF2RX2_SLOT [5:0]						0001h
R1363 (553h)	AIF2_Frame_Ctrl_13	0	0	0	0	0	0	0	0	0	0	AIF2RX3_SLOT [5:0]						0002h
R1364 (554h)	AIF2_Frame_Ctrl_14	0	0	0	0	0	0	0	0	0	0	AIF2RX4_SLOT [5:0]						0003h
R1365 (555h)	AIF2_Frame_Ctrl_15	0	0	0	0	0	0	0	0	0	0	AIF2RX5_SLOT [5:0]						0004h
R1366 (556h)	AIF2_Frame_Ctrl_16	0	0	0	0	0	0	0	0	0	0	AIF2RX6_SLOT [5:0]						0005h
R1367 (557h)	AIF2_Frame_Ctrl_17	0	0	0	0	0	0	0	0	0	0	AIF2RX7_SLOT [5:0]						0006h
R1368 (558h)	AIF2_Frame_Ctrl_18	0	0	0	0	0	0	0	0	0	0	AIF2RX8_SLOT [5:0]						0007h
R1369 (559h)	AIF2_Tx_Enables	0	0	0	0	0	0	0	0	AIF2TX8_ENA	AIF2TX7_ENA	AIF2TX6_ENA	AIF2TX5_ENA	AIF2TX4_ENA	AIF2TX3_ENA	AIF2TX2_ENA	AIF2TX1_ENA	0000h
R1370 (55Ah)	AIF2_Rx_Enables	0	0	0	0	0	0	0	0	AIF2RX8_ENA	AIF2RX7_ENA	AIF2RX6_ENA	AIF2RX5_ENA	AIF2RX4_ENA	AIF2RX3_ENA	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	AIF3_LRCLK_MODE	0	AIF3_LRCLK_INV	AIF3_LRCLK_FRC	AIF3_LRCLK_MSTR	0000h
R1411 (583h)	AIF3_Rate_Ctrl	AIF3_RATE [4:0]					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
R1419 (58Bh)	AIF3_Frame_Ctrl_5	0	0	0	0	0	0	0	0	0	0	AIF3TX3_SLOT [5:0]						0002h
R1420 (58Ch)	AIF3_Frame_Ctrl_6	0	0	0	0	0	0	0	0	0	0	AIF3TX4_SLOT [5:0]						0003h
R1421 (58Dh)	AIF3_Frame_Ctrl_7	0	0	0	0	0	0	0	0	0	0	AIF3TX5_SLOT [5:0]						0004h
R1422 (58Eh)	AIF3_Frame_Ctrl_8	0	0	0	0	0	0	0	0	0	0	AIF3TX6_SLOT [5:0]						0005h
R1423 (58Fh)	AIF3_Frame_Ctrl_9	0	0	0	0	0	0	0	0	0	0	AIF3TX7_SLOT [5:0]						0006h
R1424 (590h)	AIF3_Frame_Ctrl_10	0	0	0	0	0	0	0	0	0	0	AIF3TX8_SLOT [5:0]						0007h
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
R1427 (593h)	AIF3_Frame_Ctrl_13	0	0	0	0	0	0	0	0	0	0	AIF3RX3_SLOT [5:0]						0002h
R1428 (594h)	AIF3_Frame_Ctrl_14	0	0	0	0	0	0	0	0	0	0	AIF3RX4_SLOT [5:0]						0003h
R1429 (595h)	AIF3_Frame_Ctrl_15	0	0	0	0	0	0	0	0	0	0	AIF3RX5_SLOT [5:0]						0004h
R1430 (596h)	AIF3_Frame_Ctrl_16	0	0	0	0	0	0	0	0	0	0	AIF3RX6_SLOT [5:0]						0005h
R1431 (597h)	AIF3_Frame_Ctrl_17	0	0	0	0	0	0	0	0	0	0	AIF3RX7_SLOT [5:0]						0006h
R1432 (598h)	AIF3_Frame_Ctrl_18	0	0	0	0	0	0	0	0	0	0	AIF3RX8_SLOT [5:0]						0007h
R1433 (599h)	AIF3_Tx_Enables	0	0	0	0	0	0	0	0	AIF3TX8_ENA	AIF3TX7_ENA	AIF3TX6_ENA	AIF3TX5_ENA	AIF3TX4_ENA	AIF3TX3_ENA	AIF3TX2_ENA	AIF3TX1_ENA	0000h
R1434 (59Ah)	AIF3_Rx_Enables	0	0	0	0	0	0	0	0	AIF3RX8_ENA	AIF3RX7_ENA	AIF3RX6_ENA	AIF3RX5_ENA	AIF3RX4_ENA	AIF3RX3_ENA	AIF3RX2_ENA	AIF3RX1_ENA	0000h
R1474 (5C2h)	SPD1_TX_Control	0	0	SPD1_VAL2	SPD1_VAL1	0	0	0	SPD1_RATE [4:0]					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	SLIMRX2_PORT_ADDR [7:0]								SLIMRX1_PORT_ADDR [7:0]								0100h
R1491 (5D3h)	SLIMbus_RX_Ports1	SLIMRX4_PORT_ADDR [7:0]								SLIMRX3_PORT_ADDR [7:0]								0302h
R1492 (5D4h)	SLIMbus_RX_Ports2	SLIMRX6_PORT_ADDR [7:0]								SLIMRX5_PORT_ADDR [7:0]								0504h
R1493 (5D5h)	SLIMbus_RX_Ports3	SLIMRX8_PORT_ADDR [7:0]								SLIMRX7_PORT_ADDR [7:0]								0706h
R1494 (5D6h)	SLIMbus_TX_Ports0	SLIMTX2_PORT_ADDR [7:0]								SLIMTX1_PORT_ADDR [7:0]								0908h
R1495 (5D7h)	SLIMbus_TX_Ports1	SLIMTX4_PORT_ADDR [7:0]								SLIMTX3_PORT_ADDR [7:0]								0B0Ah
R1496 (5D8h)	SLIMbus_TX_Ports2	SLIMTX6_PORT_ADDR [7:0]								SLIMTX5_PORT_ADDR [7:0]								0D0Ch
R1497 (5D9h)	SLIMbus_TX_Ports3	SLIMTX8_PORT_ADDR [7:0]								SLIMTX7_PORT_ADDR [7:0]								0F0Eh

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
R1507 (5E3h)	SLIMbus_Framer_Ref_Gear	0	0	0	0	0	0	0	0	0	0	0	SLIMCLK_SRC	SLIMCLK_REF_GEAR [3:0]				0000h
R1509 (5E5h)	SLIMbus_Rates_1	SLIMRX2_RATE [4:0]					0	0	0	SLIMRX1_RATE [4:0]					0	0	0	0000h
R1510 (5E6h)	SLIMbus_Rates_2	SLIMRX4_RATE [4:0]					0	0	0	SLIMRX3_RATE [4:0]					0	0	0	0000h
R1511 (5E7h)	SLIMbus_Rates_3	SLIMRX6_RATE [4:0]					0	0	0	SLIMRX5_RATE [4:0]					0	0	0	0000h
R1512 (5E8h)	SLIMbus_Rates_4	SLIMRX8_RATE [4:0]					0	0	0	SLIMRX7_RATE [4:0]					0	0	0	0000h
R1513 (5E9h)	SLIMbus_Rates_5	SLIMTX2_RATE [4:0]					0	0	0	SLIMTX1_RATE [4:0]					0	0	0	0000h
R1514 (5EAh)	SLIMbus_Rates_6	SLIMTX4_RATE [4:0]					0	0	0	SLIMTX3_RATE [4:0]					0	0	0	0000h
R1515 (5EBh)	SLIMbus_Rates_7	SLIMTX6_RATE [4:0]					0	0	0	SLIMTX5_RATE [4:0]					0	0	0	0000h
R1516 (5ECh)	SLIMbus_Rates_8	SLIMTX8_RATE [4:0]					0	0	0	SLIMTX7_RATE [4:0]					0	0	0	0000h
R1520 (5F0h)	Slimbus_Pad_Ctrl	0	0	0	0	0	0	0	0	0	0	0	0	0	SLIMDAT2_DRV_STR	SLIMDAT1_DRV_STR	SLIMCLK_DRV_STR	0007h
R1525 (5F5h)	SLIMbus_RX_Channel_Enable	0	0	0	0	0	0	0	0	SLIMRX8_ENA	SLIMRX7_ENA	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	SLIMTX8_ENA	SLIMTX7_ENA	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	SLIMRX8_PORT_STS	SLIMRX7_PORT_STS	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	SLIMTX8_PORT_STS	SLIMTX7_PORT_STS	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_STS	0	0	0	0	0	0	0	PWM1MIX_SRC1 [7:0]								0000h
R1601 (641h)	PWM1MIX_Input_1_Volume	0	0	0	0	0	0	0	0	PWM1MIX_VOL1 [6:0]							0	0080h
R1602 (642h)	PWM1MIX_Input_2_Source	PWM1MIX_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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

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
R1672 (688h)	OUT1RMIX_Input_1_Source	OUT1RMIX_STS	0	0	0	0	0	0	0	OUT1RMIX_SRC1 [7:0]								0000h
R1673 (689h)	OUT1RMIX_Input_1_Volume	0	0	0	0	0	0	0	0	OUT1RMIX_VOL1 [6:0]							0	0080h
R1674 (68Ah)	OUT1RMIX_Input_2_Source	OUT1RMIX_STS	0	0	0	0	0	0	0	OUT1RMIX_SRC2 [7:0]								0000h
R1675 (68Bh)	OUT1RMIX_Input_2_Volume	0	0	0	0	0	0	0	0	OUT1RMIX_VOL2 [6:0]							0	0080h
R1676 (68Ch)	OUT1RMIX_Input_3_Source	OUT1RMIX_STS	0	0	0	0	0	0	0	OUT1RMIX_SRC3 [7:0]								0000h
R1677 (68Dh)	OUT1RMIX_Input_3_Volume	0	0	0	0	0	0	0	0	OUT1RMIX_VOL3 [6:0]							0	0080h
R1678 (68Eh)	OUT1RMIX_Input_4_Source	OUT1RMIX_STS	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
R1680 (690h)	OUT2LMIX_Input_1_Source	OUT2LMIX_STS	0	0	0	0	0	0	0	OUT2LMIX_SRC1 [7:0]								0000h
R1681 (691h)	OUT2LMIX_Input_1_Volume	0	0	0	0	0	0	0	0	OUT2LMIX_VOL1 [6:0]							0	0080h
R1682 (692h)	OUT2LMIX_Input_2_Source	OUT2LMIX_STS	0	0	0	0	0	0	0	OUT2LMIX_SRC2 [7:0]								0000h
R1683 (693h)	OUT2LMIX_Input_2_Volume	0	0	0	0	0	0	0	0	OUT2LMIX_VOL2 [6:0]							0	0080h
R1684 (694h)	OUT2LMIX_Input_3_Source	OUT2LMIX_STS	0	0	0	0	0	0	0	OUT2LMIX_SRC3 [7:0]								0000h
R1685 (695h)	OUT2LMIX_Input_3_Volume	0	0	0	0	0	0	0	0	OUT2LMIX_VOL3 [6:0]							0	0080h
R1686 (696h)	OUT2LMIX_Input_4_Source	OUT2LMIX_STS	0	0	0	0	0	0	0	OUT2LMIX_SRC4 [7:0]								0000h
R1687 (697h)	OUT2LMIX_Input_4_Volume	0	0	0	0	0	0	0	0	OUT2LMIX_VOL4 [6:0]							0	0080h
R1688 (698h)	OUT2RMIX_Input_1_Source	OUT2RMIX_STS	0	0	0	0	0	0	0	OUT2RMIX_SRC1 [7:0]								0000h
R1689 (699h)	OUT2RMIX_Input_1_Volume	0	0	0	0	0	0	0	0	OUT2RMIX_VOL1 [6:0]							0	0080h
R1690 (69Ah)	OUT2RMIX_Input_2_Source	OUT2RMIX_STS	0	0	0	0	0	0	0	OUT2RMIX_SRC2 [7:0]								0000h
R1691 (69Bh)	OUT2RMIX_Input_2_Volume	0	0	0	0	0	0	0	0	OUT2RMIX_VOL2 [6:0]							0	0080h
R1692 (69Ch)	OUT2RMIX_Input_3_Source	OUT2RMIX_STS	0	0	0	0	0	0	0	OUT2RMIX_SRC3 [7:0]								0000h
R1693 (69Dh)	OUT2RMIX_Input_3_Volume	0	0	0	0	0	0	0	0	OUT2RMIX_VOL3 [6:0]							0	0080h
R1694 (69Eh)	OUT2RMIX_Input_4_Source	OUT2RMIX_STS	0	0	0	0	0	0	0	OUT2RMIX_SRC4 [7:0]								0000h
R1695 (69Fh)	OUT2RMIX_Input_4_Volume	0	0	0	0	0	0	0	0	OUT2RMIX_VOL4 [6:0]							0	0080h
R1696 (6A0h)	OUT3LMIX_Input_1_Source	OUT3LMIX_STS	0	0	0	0	0	0	0	OUT3LMIX_SRC1 [7:0]								0000h
R1697 (6A1h)	OUT3LMIX_Input_1_Volume	0	0	0	0	0	0	0	0	OUT3LMIX_VOL1 [6:0]							0	0080h
R1698 (6A2h)	OUT3LMIX_Input_2_Source	OUT3LMIX_STS	0	0	0	0	0	0	0	OUT3LMIX_SRC2 [7:0]								0000h
R1699 (6A3h)	OUT3LMIX_Input_2_Volume	0	0	0	0	0	0	0	0	OUT3LMIX_VOL2 [6:0]							0	0080h
R1700 (6A4h)	OUT3LMIX_Input_3_Source	OUT3LMIX_STS	0	0	0	0	0	0	0	OUT3LMIX_SRC3 [7:0]								0000h
R1701 (6A5h)	OUT3LMIX_Input_3_Volume	0	0	0	0	0	0	0	0	OUT3LMIX_VOL3 [6:0]							0	0080h
R1702 (6A6h)	OUT3LMIX_Input_4_Source	OUT3LMIX_STS	0	0	0	0	0	0	0	OUT3LMIX_SRC4 [7:0]								0000h
R1703 (6A7h)	OUT3LMIX_Input_4_Volume	0	0	0	0	0	0	0	0	OUT3LMIX_VOL4 [6:0]							0	0080h
R1704 (6A8h)	OUT3RMIX_Input_1_Source	OUT3RMIX_STS	0	0	0	0	0	0	0	OUT3RMIX_SRC1 [7:0]								0000h
R1705 (6A9h)	OUT3RMIX_Input_1_Volume	0	0	0	0	0	0	0	0	OUT3RMIX_VOL1 [6:0]							0	0080h
R1706 (6AAh)	OUT3RMIX_Input_2_Source	OUT3RMIX_STS	0	0	0	0	0	0	0	OUT3RMIX_SRC2 [7:0]								0000h
R1707 (6ABh)	OUT3RMIX_Input_2_Volume	0	0	0	0	0	0	0	0	OUT3RMIX_VOL2 [6:0]							0	0080h
R1708 (6ACh)	OUT3RMIX_Input_3_Source	OUT3RMIX_STS	0	0	0	0	0	0	0	OUT3RMIX_SRC3 [7:0]								0000h
R1709 (6ADh)	OUT3RMIX_Input_3_Volume	0	0	0	0	0	0	0	0	OUT3RMIX_VOL3 [6:0]							0	0080h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R1710 (6AEh)	OUT3RMIX_Input_4_Source	OUT3RMIX_STS	0	0	0	0	0	0	0	OUT3RMIX_SRC4 [7:0]								0000h
R1711 (6AFh)	OUT3RMIX_Input_4_Volume	0	0	0	0	0	0	0	0	OUT3RMIX_VOL4 [6:0]							0	0080h
R1728 (6C0h)	OUT5LMIX_Input_1_Source	OUT5LMIX_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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

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
R1812 (714h)	AIF1TX3MIX_Input_3_Source	AIF1TX3MIX_STS	0	0	0	0	0	0	0	AIF1TX3MIX_SRC3 [7:0]								0000h
R1813 (715h)	AIF1TX3MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF1TX3MIX_VOL3 [6:0]							0	0080h
R1814 (716h)	AIF1TX3MIX_Input_4_Source	AIF1TX3MIX_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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
R1840 (730h)	AIF1TX7MIX_Input_1_Source	AIF1TX7MIX_STS	0	0	0	0	0	0	0	AIF1TX7MIX_SRC1 [7:0]								0000h
R1841 (731h)	AIF1TX7MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF1TX7MIX_VOL1 [6:0]							0	0080h
R1842 (732h)	AIF1TX7MIX_Input_2_Source	AIF1TX7MIX_STS	0	0	0	0	0	0	0	AIF1TX7MIX_SRC2 [7:0]								0000h
R1843 (733h)	AIF1TX7MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF1TX7MIX_VOL2 [6:0]							0	0080h
R1844 (734h)	AIF1TX7MIX_Input_3_Source	AIF1TX7MIX_STS	0	0	0	0	0	0	0	AIF1TX7MIX_SRC3 [7:0]								0000h
R1845 (735h)	AIF1TX7MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF1TX7MIX_VOL3 [6:0]							0	0080h
R1846 (736h)	AIF1TX7MIX_Input_4_Source	AIF1TX7MIX_STS	0	0	0	0	0	0	0	AIF1TX7MIX_SRC4 [7:0]								0000h
R1847 (737h)	AIF1TX7MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF1TX7MIX_VOL4 [6:0]							0	0080h
R1848 (738h)	AIF1TX8MIX_Input_1_Source	AIF1TX8MIX_STS	0	0	0	0	0	0	0	AIF1TX8MIX_SRC1 [7:0]								0000h
R1849 (739h)	AIF1TX8MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF1TX8MIX_VOL1 [6:0]							0	0080h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R1850 (73Ah)	AIF1TX8MIX_Input_2_ Source	AIF1TX8MI X_STS	0	0	0	0	0	0	0	AIF1TX8MIX_SRC2 [7:0]								0000h
R1851 (73Bh)	AIF1TX8MIX_Input_2_ Volume	0	0	0	0	0	0	0	0	AIF1TX8MIX_VOL2 [6:0]							0	0080h
R1852 (73Ch)	AIF1TX8MIX_Input_3_ Source	AIF1TX8MI X_STS	0	0	0	0	0	0	0	AIF1TX8MIX_SRC3 [7:0]								0000h
R1853 (73Dh)	AIF1TX8MIX_Input_3_ Volume	0	0	0	0	0	0	0	0	AIF1TX8MIX_VOL3 [6:0]							0	0080h
R1854 (73Eh)	AIF1TX8MIX_Input_4_ Source	AIF1TX8MI X_STS	0	0	0	0	0	0	0	AIF1TX8MIX_SRC4 [7:0]								0000h
R1855 (73Fh)	AIF1TX8MIX_Input_4_ Volume	0	0	0	0	0	0	0	0	AIF1TX8MIX_VOL4 [6:0]							0	0080h
R1856 (740h)	AIF2TX1MIX_Input_1_ Source	AIF2TX1MI X_STS	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	AIF2TX1MI X_STS	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	AIF2TX1MI X_STS	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	AIF2TX1MI X_STS	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	AIF2TX2MI X_STS	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	AIF2TX2MI X_STS	0	0	0	0	0	0	0	AIF2TX2MIX_SRC2 [7:0]								0000h
R1867 (74Bh)	AIF2TX2MIX_Input_2_ Volume	0	0	0	0	0	0	0	0	AIF2TX2MIX_VOL2 [6:0]							0	0080h
R1868 (74Ch)	AIF2TX2MIX_Input_3_ Source	AIF2TX2MI X_STS	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	AIF2TX2MI X_STS	0	0	0	0	0	0	0	AIF2TX2MIX_SRC4 [7:0]								0000h
R1871 (74Fh)	AIF2TX2MIX_Input_4_ Volume	0	0	0	0	0	0	0	0	AIF2TX2MIX_VOL4 [6:0]							0	0080h
R1872 (750h)	AIF2TX3MIX_Input_1_ Source	AIF2TX3MI X_STS	0	0	0	0	0	0	0	AIF2TX3MIX_SRC1 [7:0]								0000h
R1873 (751h)	AIF2TX3MIX_Input_1_ Volume	0	0	0	0	0	0	0	0	AIF2TX3MIX_VOL1 [6:0]							0	0080h
R1874 (752h)	AIF2TX3MIX_Input_2_ Source	AIF2TX3MI X_STS	0	0	0	0	0	0	0	AIF2TX3MIX_SRC2 [7:0]								0000h
R1875 (753h)	AIF2TX3MIX_Input_2_ Volume	0	0	0	0	0	0	0	0	AIF2TX3MIX_VOL2 [6:0]							0	0080h
R1876 (754h)	AIF2TX3MIX_Input_3_ Source	AIF2TX3MI X_STS	0	0	0	0	0	0	0	AIF2TX3MIX_SRC3 [7:0]								0000h
R1877 (755h)	AIF2TX3MIX_Input_3_ Volume	0	0	0	0	0	0	0	0	AIF2TX3MIX_VOL3 [6:0]							0	0080h
R1878 (756h)	AIF2TX3MIX_Input_4_ Source	AIF2TX3MI X_STS	0	0	0	0	0	0	0	AIF2TX3MIX_SRC4 [7:0]								0000h
R1879 (757h)	AIF2TX3MIX_Input_4_ Volume	0	0	0	0	0	0	0	0	AIF2TX3MIX_VOL4 [6:0]							0	0080h
R1880 (758h)	AIF2TX4MIX_Input_1_ Source	AIF2TX4MI X_STS	0	0	0	0	0	0	0	AIF2TX4MIX_SRC1 [7:0]								0000h
R1881 (759h)	AIF2TX4MIX_Input_1_ Volume	0	0	0	0	0	0	0	0	AIF2TX4MIX_VOL1 [6:0]							0	0080h
R1882 (75Ah)	AIF2TX4MIX_Input_2_ Source	AIF2TX4MI X_STS	0	0	0	0	0	0	0	AIF2TX4MIX_SRC2 [7:0]								0000h
R1883 (75Bh)	AIF2TX4MIX_Input_2_ Volume	0	0	0	0	0	0	0	0	AIF2TX4MIX_VOL2 [6:0]							0	0080h
R1884 (75Ch)	AIF2TX4MIX_Input_3_ Source	AIF2TX4MI X_STS	0	0	0	0	0	0	0	AIF2TX4MIX_SRC3 [7:0]								0000h
R1885 (75Dh)	AIF2TX4MIX_Input_3_ Volume	0	0	0	0	0	0	0	0	AIF2TX4MIX_VOL3 [6:0]							0	0080h
R1886 (75Eh)	AIF2TX4MIX_Input_4_ Source	AIF2TX4MI X_STS	0	0	0	0	0	0	0	AIF2TX4MIX_SRC4 [7:0]								0000h
R1887 (75Fh)	AIF2TX4MIX_Input_4_ Volume	0	0	0	0	0	0	0	0	AIF2TX4MIX_VOL4 [6:0]							0	0080h

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
R1888 (760h)	AIF2TX5MIX_Input_1_Source	AIF2TX5MIX_STS	0	0	0	0	0	0	0	AIF2TX5MIX_SRC1 [7:0]								0000h
R1889 (761h)	AIF2TX5MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF2TX5MIX_VOL1 [6:0]							0	0080h
R1890 (762h)	AIF2TX5MIX_Input_2_Source	AIF2TX5MIX_STS	0	0	0	0	0	0	0	AIF2TX5MIX_SRC2 [7:0]								0000h
R1891 (763h)	AIF2TX5MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF2TX5MIX_VOL2 [6:0]							0	0080h
R1892 (764h)	AIF2TX5MIX_Input_3_Source	AIF2TX5MIX_STS	0	0	0	0	0	0	0	AIF2TX5MIX_SRC3 [7:0]								0000h
R1893 (765h)	AIF2TX5MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF2TX5MIX_VOL3 [6:0]							0	0080h
R1894 (766h)	AIF2TX5MIX_Input_4_Source	AIF2TX5MIX_STS	0	0	0	0	0	0	0	AIF2TX5MIX_SRC4 [7:0]								0000h
R1895 (767h)	AIF2TX5MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF2TX5MIX_VOL4 [6:0]							0	0080h
R1896 (768h)	AIF2TX6MIX_Input_1_Source	AIF2TX6MIX_STS	0	0	0	0	0	0	0	AIF2TX6MIX_SRC1 [7:0]								0000h
R1897 (769h)	AIF2TX6MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF2TX6MIX_VOL1 [6:0]							0	0080h
R1898 (76Ah)	AIF2TX6MIX_Input_2_Source	AIF2TX6MIX_STS	0	0	0	0	0	0	0	AIF2TX6MIX_SRC2 [7:0]								0000h
R1899 (76Bh)	AIF2TX6MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF2TX6MIX_VOL2 [6:0]							0	0080h
R1900 (76Ch)	AIF2TX6MIX_Input_3_Source	AIF2TX6MIX_STS	0	0	0	0	0	0	0	AIF2TX6MIX_SRC3 [7:0]								0000h
R1901 (76Dh)	AIF2TX6MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF2TX6MIX_VOL3 [6:0]							0	0080h
R1902 (76Eh)	AIF2TX6MIX_Input_4_Source	AIF2TX6MIX_STS	0	0	0	0	0	0	0	AIF2TX6MIX_SRC4 [7:0]								0000h
R1903 (76Fh)	AIF2TX6MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF2TX6MIX_VOL4 [6:0]							0	0080h
R1904 (770h)	AIF2TX7MIX_Input_1_Source	AIF2TX7MIX_STS	0	0	0	0	0	0	0	AIF2TX7MIX_SRC1 [7:0]								0000h
R1905 (771h)	AIF2TX7MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF2TX7MIX_VOL1 [6:0]							0	0080h
R1906 (772h)	AIF2TX7MIX_Input_2_Source	AIF2TX7MIX_STS	0	0	0	0	0	0	0	AIF2TX7MIX_SRC2 [7:0]								0000h
R1907 (773h)	AIF2TX7MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF2TX7MIX_VOL2 [6:0]							0	0080h
R1908 (774h)	AIF2TX7MIX_Input_3_Source	AIF2TX7MIX_STS	0	0	0	0	0	0	0	AIF2TX7MIX_SRC3 [7:0]								0000h
R1909 (775h)	AIF2TX7MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF2TX7MIX_VOL3 [6:0]							0	0080h
R1910 (776h)	AIF2TX7MIX_Input_4_Source	AIF2TX7MIX_STS	0	0	0	0	0	0	0	AIF2TX7MIX_SRC4 [7:0]								0000h
R1911 (777h)	AIF2TX7MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF2TX7MIX_VOL4 [6:0]							0	0080h
R1912 (778h)	AIF2TX8MIX_Input_1_Source	AIF2TX8MIX_STS	0	0	0	0	0	0	0	AIF2TX8MIX_SRC1 [7:0]								0000h
R1913 (779h)	AIF2TX8MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF2TX8MIX_VOL1 [6:0]							0	0080h
R1914 (77Ah)	AIF2TX8MIX_Input_2_Source	AIF2TX8MIX_STS	0	0	0	0	0	0	0	AIF2TX8MIX_SRC2 [7:0]								0000h
R1915 (77Bh)	AIF2TX8MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF2TX8MIX_VOL2 [6:0]							0	0080h
R1916 (77Ch)	AIF2TX8MIX_Input_3_Source	AIF2TX8MIX_STS	0	0	0	0	0	0	0	AIF2TX8MIX_SRC3 [7:0]								0000h
R1917 (77Dh)	AIF2TX8MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF2TX8MIX_VOL3 [6:0]							0	0080h
R1918 (77Eh)	AIF2TX8MIX_Input_4_Source	AIF2TX8MIX_STS	0	0	0	0	0	0	0	AIF2TX8MIX_SRC4 [7:0]								0000h
R1919 (77Fh)	AIF2TX8MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF2TX8MIX_VOL4 [6:0]							0	0080h
R1920 (780h)	AIF3TX1MIX_Input_1_Source	AIF3TX1MIX_STS	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_STS	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_STS	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

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
R1926 (786h)	AIF3TX1MIX_Input_4_Source	AIF3TX1MIX_STS	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_STS	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_STS	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_STS	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_STS	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
R1936 (790h)	AIF3TX3MIX_Input_1_Source	AIF3TX3MIX_STS	0	0	0	0	0	0	0	AIF3TX3MIX_SRC1 [7:0]								0000h
R1937 (791h)	AIF3TX3MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF3TX3MIX_VOL1 [6:0]							0	0080h
R1938 (792h)	AIF3TX3MIX_Input_2_Source	AIF3TX3MIX_STS	0	0	0	0	0	0	0	AIF3TX3MIX_SRC2 [7:0]								0000h
R1939 (793h)	AIF3TX3MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF3TX3MIX_VOL2 [6:0]							0	0080h
R1940 (794h)	AIF3TX3MIX_Input_3_Source	AIF3TX3MIX_STS	0	0	0	0	0	0	0	AIF3TX3MIX_SRC3 [7:0]								0000h
R1941 (795h)	AIF3TX3MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF3TX3MIX_VOL3 [6:0]							0	0080h
R1942 (796h)	AIF3TX3MIX_Input_4_Source	AIF3TX3MIX_STS	0	0	0	0	0	0	0	AIF3TX3MIX_SRC4 [7:0]								0000h
R1943 (797h)	AIF3TX3MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF3TX3MIX_VOL4 [6:0]							0	0080h
R1944 (798h)	AIF3TX4MIX_Input_1_Source	AIF3TX4MIX_STS	0	0	0	0	0	0	0	AIF3TX4MIX_SRC1 [7:0]								0000h
R1945 (799h)	AIF3TX4MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF3TX4MIX_VOL1 [6:0]							0	0080h
R1946 (79Ah)	AIF3TX4MIX_Input_2_Source	AIF3TX4MIX_STS	0	0	0	0	0	0	0	AIF3TX4MIX_SRC2 [7:0]								0000h
R1947 (79Bh)	AIF3TX4MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF3TX4MIX_VOL2 [6:0]							0	0080h
R1948 (79Ch)	AIF3TX4MIX_Input_3_Source	AIF3TX4MIX_STS	0	0	0	0	0	0	0	AIF3TX4MIX_SRC3 [7:0]								0000h
R1949 (79Dh)	AIF3TX4MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF3TX4MIX_VOL3 [6:0]							0	0080h
R1950 (79Eh)	AIF3TX4MIX_Input_4_Source	AIF3TX4MIX_STS	0	0	0	0	0	0	0	AIF3TX4MIX_SRC4 [7:0]								0000h
R1951 (79Fh)	AIF3TX4MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF3TX4MIX_VOL4 [6:0]							0	0080h
R1952 (7A0h)	AIF3TX5MIX_Input_1_Source	AIF3TX5MIX_STS	0	0	0	0	0	0	0	AIF3TX5MIX_SRC1 [7:0]								0000h
R1953 (7A1h)	AIF3TX5MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF3TX5MIX_VOL1 [6:0]							0	0080h
R1954 (7A2h)	AIF3TX5MIX_Input_2_Source	AIF3TX5MIX_STS	0	0	0	0	0	0	0	AIF3TX5MIX_SRC2 [7:0]								0000h
R1955 (7A3h)	AIF3TX5MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF3TX5MIX_VOL2 [6:0]							0	0080h
R1956 (7A4h)	AIF3TX5MIX_Input_3_Source	AIF3TX5MIX_STS	0	0	0	0	0	0	0	AIF3TX5MIX_SRC3 [7:0]								0000h
R1957 (7A5h)	AIF3TX5MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF3TX5MIX_VOL3 [6:0]							0	0080h
R1958 (7A6h)	AIF3TX5MIX_Input_4_Source	AIF3TX5MIX_STS	0	0	0	0	0	0	0	AIF3TX5MIX_SRC4 [7:0]								0000h
R1959 (7A7h)	AIF3TX5MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF3TX5MIX_VOL4 [6:0]							0	0080h
R1960 (7A8h)	AIF3TX6MIX_Input_1_Source	AIF3TX6MIX_STS	0	0	0	0	0	0	0	AIF3TX6MIX_SRC1 [7:0]								0000h
R1961 (7A9h)	AIF3TX6MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF3TX6MIX_VOL1 [6:0]							0	0080h
R1962 (7AAh)	AIF3TX6MIX_Input_2_Source	AIF3TX6MIX_STS	0	0	0	0	0	0	0	AIF3TX6MIX_SRC2 [7:0]								0000h
R1963 (7ABh)	AIF3TX6MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF3TX6MIX_VOL2 [6:0]							0	0080h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R1964 (7ACh)	AIF3TX6MIX_Input_3_Source	AIF3TX6MIX_STS	0	0	0	0	0	0	0	AIF3TX6MIX_SRC3 [7:0]								0000h
R1965 (7ADh)	AIF3TX6MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF3TX6MIX_VOL3 [6:0]							0	0080h
R1966 (7AEh)	AIF3TX6MIX_Input_4_Source	AIF3TX6MIX_STS	0	0	0	0	0	0	0	AIF3TX6MIX_SRC4 [7:0]								0000h
R1967 (7AFh)	AIF3TX6MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF3TX6MIX_VOL4 [6:0]							0	0080h
R1968 (7B0h)	AIF3TX7MIX_Input_1_Source	AIF3TX7MIX_STS	0	0	0	0	0	0	0	AIF3TX7MIX_SRC1 [7:0]								0000h
R1969 (7B1h)	AIF3TX7MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF3TX7MIX_VOL1 [6:0]							0	0080h
R1970 (7B2h)	AIF3TX7MIX_Input_2_Source	AIF3TX7MIX_STS	0	0	0	0	0	0	0	AIF3TX7MIX_SRC2 [7:0]								0000h
R1971 (7B3h)	AIF3TX7MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF3TX7MIX_VOL2 [6:0]							0	0080h
R1972 (7B4h)	AIF3TX7MIX_Input_3_Source	AIF3TX7MIX_STS	0	0	0	0	0	0	0	AIF3TX7MIX_SRC3 [7:0]								0000h
R1973 (7B5h)	AIF3TX7MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF3TX7MIX_VOL3 [6:0]							0	0080h
R1974 (7B6h)	AIF3TX7MIX_Input_4_Source	AIF3TX7MIX_STS	0	0	0	0	0	0	0	AIF3TX7MIX_SRC4 [7:0]								0000h
R1975 (7B7h)	AIF3TX7MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF3TX7MIX_VOL4 [6:0]							0	0080h
R1976 (7B8h)	AIF3TX8MIX_Input_1_Source	AIF3TX8MIX_STS	0	0	0	0	0	0	0	AIF3TX8MIX_SRC1 [7:0]								0000h
R1977 (7B9h)	AIF3TX8MIX_Input_1_Volume	0	0	0	0	0	0	0	0	AIF3TX8MIX_VOL1 [6:0]							0	0080h
R1978 (7BAh)	AIF3TX8MIX_Input_2_Source	AIF3TX8MIX_STS	0	0	0	0	0	0	0	AIF3TX8MIX_SRC2 [7:0]								0000h
R1979 (7BBh)	AIF3TX8MIX_Input_2_Volume	0	0	0	0	0	0	0	0	AIF3TX8MIX_VOL2 [6:0]							0	0080h
R1980 (7BCh)	AIF3TX8MIX_Input_3_Source	AIF3TX8MIX_STS	0	0	0	0	0	0	0	AIF3TX8MIX_SRC3 [7:0]								0000h
R1981 (7BDh)	AIF3TX8MIX_Input_3_Volume	0	0	0	0	0	0	0	0	AIF3TX8MIX_VOL3 [6:0]							0	0080h
R1982 (7BEh)	AIF3TX8MIX_Input_4_Source	AIF3TX8MIX_STS	0	0	0	0	0	0	0	AIF3TX8MIX_SRC4 [7:0]								0000h
R1983 (7BFh)	AIF3TX8MIX_Input_4_Volume	0	0	0	0	0	0	0	0	AIF3TX8MIX_VOL4 [6:0]							0	0080h
R1984 (7C0h)	SLIMTX1MIX_Input_1_Source	SLIMTX1MIX_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	0	0	0	0	0	0	0	SLIMTX3MIX_SRC1 [7:0]								0000h
R2001 (7D1h)	SLIMTX3MIX_Input_1_Volume	0	0	0	0	0	0	0	0	SLIMTX3MIX_VOL1 [6:0]							0	0080h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R2002 (7D2h)	SLIMTX3MIX_Input_2_Source	SLIMTX3MIX_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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
R2032 (7F0h)	SLIMTX7MIX_Input_1_Source	SLIMTX7MIX_STS	0	0	0	0	0	0	0	SLIMTX7MIX_SRC1 [7:0]								0000h
R2033 (7F1h)	SLIMTX7MIX_Input_1_Volume	0	0	0	0	0	0	0	0	SLIMTX7MIX_VOL1 [6:0]							0	0080h
R2034 (7F2h)	SLIMTX7MIX_Input_2_Source	SLIMTX7MIX_STS	0	0	0	0	0	0	0	SLIMTX7MIX_SRC2 [7:0]								0000h
R2035 (7F3h)	SLIMTX7MIX_Input_2_Volume	0	0	0	0	0	0	0	0	SLIMTX7MIX_VOL2 [6:0]							0	0080h
R2036 (7F4h)	SLIMTX7MIX_Input_3_Source	SLIMTX7MIX_STS	0	0	0	0	0	0	0	SLIMTX7MIX_SRC3 [7:0]								0000h
R2037 (7F5h)	SLIMTX7MIX_Input_3_Volume	0	0	0	0	0	0	0	0	SLIMTX7MIX_VOL3 [6:0]							0	0080h
R2038 (7F6h)	SLIMTX7MIX_Input_4_Source	SLIMTX7MIX_STS	0	0	0	0	0	0	0	SLIMTX7MIX_SRC4 [7:0]								0000h
R2039 (7F7h)	SLIMTX7MIX_Input_4_Volume	0	0	0	0	0	0	0	0	SLIMTX7MIX_VOL4 [6:0]							0	0080h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R2040 (7F8h)	SLIMTX8MIX_Input_1_Source	SLIMTX8MIX_STS	0	0	0	0	0	0	0	SLIMTX8MIX_SRC1 [7:0]								0000h
R2041 (7F9h)	SLIMTX8MIX_Input_1_Volume	0	0	0	0	0	0	0	0	SLIMTX8MIX_VOL1 [6:0]							0	0080h
R2042 (7FAh)	SLIMTX8MIX_Input_2_Source	SLIMTX8MIX_STS	0	0	0	0	0	0	0	SLIMTX8MIX_SRC2 [7:0]								0000h
R2043 (7FBh)	SLIMTX8MIX_Input_2_Volume	0	0	0	0	0	0	0	0	SLIMTX8MIX_VOL2 [6:0]							0	0080h
R2044 (7FCh)	SLIMTX8MIX_Input_3_Source	SLIMTX8MIX_STS	0	0	0	0	0	0	0	SLIMTX8MIX_SRC3 [7:0]								0000h
R2045 (7FDh)	SLIMTX8MIX_Input_3_Volume	0	0	0	0	0	0	0	0	SLIMTX8MIX_VOL3 [6:0]							0	0080h
R2046 (7FEh)	SLIMTX8MIX_Input_4_Source	SLIMTX8MIX_STS	0	0	0	0	0	0	0	SLIMTX8MIX_SRC4 [7:0]								0000h
R2047 (7FFh)	SLIMTX8MIX_Input_4_Volume	0	0	0	0	0	0	0	0	SLIMTX8MIX_VOL4 [6:0]							0	0080h
R2048 (800h)	SPDIF1TX1MIX_Input_1_Source	SPDIF1TX1_STS	0	0	0	0	0	0	0	SPDIF1TX1_SRC [7:0]								0000h
R2049 (801h)	SPDIF1TX1MIX_Input_1_Volume	0	0	0	0	0	0	0	0	SPDIF1TX1_VOL [6:0]							0	0080h
R2056 (808h)	SPDIF1TX2MIX_Input_1_Source	SPDIF1TX2_STS	0	0	0	0	0	0	0	SPDIF1TX2_SRC [7:0]								0000h
R2057 (809h)	SPDIF1TX2MIX_Input_1_Volume	0	0	0	0	0	0	0	0	SPDIF1TX2_VOL [6:0]							0	0080h
R2176 (880h)	EQ1MIX_Input_1_Source	EQ1MIX_STS	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_STS	0	0	0	0	0	0	0	EQ1MIX_SRC2 [7:0]								0000h
R2179 (883h)	EQ1MIX_Input_2_Volume	0	0	0	0	0	0	0	0	EQ1MIX_VOL2 [6:0]							0	0080h
R2180 (884h)	EQ1MIX_Input_3_Source	EQ1MIX_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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

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
R2202 (89Ah)	EQ4MIX_Input_2_Source	EQ4MIX_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	0	0	0	0	0	0	0	DRC1LMIX_SRC4 [7:0]								0000h
R2247 (8C7h)	DRC1LMIX_Input_4_Volume	0	0	0	0	0	0	0	0	DRC1LMIX_VOL4 [6:0]							0	0080h
R2248 (8C8h)	DRC1RMIX_Input_1_Source	DRC1RMIX_STS	0	0	0	0	0	0	0	DRC1RMIX_SRC1 [7:0]								0000h
R2249 (8C9h)	DRC1RMIX_Input_1_Volume	0	0	0	0	0	0	0	0	DRC1RMIX_VOL1 [6:0]							0	0080h
R2250 (8CAh)	DRC1RMIX_Input_2_Source	DRC1RMIX_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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

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
R2304 (900h)	HPLP1MIX_Input_1_Source	LHPF1MIX_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	0	0	0	0	0	0	0	LHPF2MIX_SRC4 [7:0]								0000h
R2319 (90Fh)	HPLP2MIX_Input_4_Volume	0	0	0	0	0	0	0	0	LHPF2MIX_VOL4 [6:0]							0	0080h
R2320 (910h)	HPLP3MIX_Input_1_Source	LHPF3MIX_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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_STS	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

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
R2374 (946h)	DSP1LMIX_Input_4_Source	DSP1LMIX_STS	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_STS	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_STS	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_STS	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_STS	0	0	0	0	0	0	0	DSP1RMIX_SRC4 [7:0]								0000h
R2383 (94Fh)	DSP1RMIX_Input_4_Volume	0	0	0	0	0	0	0	0	DSP1RMIX_VOL4 [6:0]							0	0080h
R2384 (950h)	DSP1AUX1MIX_Input_1_Source	DSP1AUX1_STS	0	0	0	0	0	0	0	DSP1AUX1_SRC [7:0]								0000h
R2392 (958h)	DSP1AUX2MIX_Input_1_Source	DSP1AUX2_STS	0	0	0	0	0	0	0	DSP1AUX2_SRC [7:0]								0000h
R2400 (960h)	DSP1AUX3MIX_Input_1_Source	DSP1AUX3_STS	0	0	0	0	0	0	0	DSP1AUX3_SRC [7:0]								0000h
R2408 (968h)	DSP1AUX4MIX_Input_1_Source	DSP1AUX4_STS	0	0	0	0	0	0	0	DSP1AUX4_SRC [7:0]								0000h
R2416 (970h)	DSP1AUX5MIX_Input_1_Source	DSP1AUX5_STS	0	0	0	0	0	0	0	DSP1AUX5_SRC [7:0]								0000h
R2424 (978h)	DSP1AUX6MIX_Input_1_Source	DSP1AUX6_STS	0	0	0	0	0	0	0	DSP1AUX6_SRC [7:0]								0000h
R2688 (A80h)	ASRC1_1LMIX_Input_1_Source	ASRC1_IN1L_STS	0	0	0	0	0	0	0	ASRC1_IN1L_SRC [7:0]								0000h
R2696 (A88h)	ASRC1_1RMIX_Input_1_Source	ASRC1_IN1R_STS	0	0	0	0	0	0	0	ASRC1_IN1R_SRC [7:0]								0000h
R2704 (A90h)	ASRC1_2LMIX_Input_1_Source	ASRC1_IN2L_STS	0	0	0	0	0	0	0	ASRC1_IN2L_SRC [7:0]								0000h
R2712 (A98h)	ASRC1_2RMIX_Input_1_Source	ASRC1_IN2R_STS	0	0	0	0	0	0	0	ASRC1_IN2R_SRC [7:0]								0000h
R2816 (B00h)	ISRC1DEC1MIX_Input_1_Source	ISRC1DEC1_STS	0	0	0	0	0	0	0	ISRC1DEC1_SRC [7:0]								0000h
R2824 (B08h)	ISRC1DEC2MIX_Input_1_Source	ISRC1DEC2_STS	0	0	0	0	0	0	0	ISRC1DEC2_SRC [7:0]								0000h
R2848 (B20h)	ISRC1INT1MIX_Input_1_Source	ISRC1INT1_STS	0	0	0	0	0	0	0	ISRC1INT1_SRC [7:0]								0000h
R2856 (B28h)	ISRC1INT2MIX_Input_1_Source	ISRC1INT2_STS	0	0	0	0	0	0	0	ISRC1INT2_SRC [7:0]								0000h
R2880 (B40h)	ISRC2DEC1MIX_Input_1_Source	ISRC2DEC1_STS	0	0	0	0	0	0	0	ISRC2DEC1_SRC [7:0]								0000h
R2888 (B48h)	ISRC2DEC2MIX_Input_1_Source	ISRC2DEC2_STS	0	0	0	0	0	0	0	ISRC2DEC2_SRC [7:0]								0000h
R2912 (B60h)	ISRC2INT1MIX_Input_1_Source	ISRC2INT1_STS	0	0	0	0	0	0	0	ISRC2INT1_SRC [7:0]								0000h
R2920 (B68h)	ISRC2INT2MIX_Input_1_Source	ISRC2INT2_STS	0	0	0	0	0	0	0	ISRC2INT2_SRC [7:0]								0000h
R3520 (DC0h)	DFC1MIX_Input_1_Source	DFC1_STS	0	0	0	0	0	0	0	DFC1_SRC [7:0]								0000h
R3528 (DC8h)	DFC2MIX_Input_1_Source	DFC2_STS	0	0	0	0	0	0	0	DFC2_SRC [7:0]								0000h
R3536 (DD0h)	DFC3MIX_Input_1_Source	DFC3_STS	0	0	0	0	0	0	0	DFC3_SRC [7:0]								0000h
R3544 (DD8h)	DFC4MIX_Input_1_Source	DFC4_STS	0	0	0	0	0	0	0	DFC4_SRC [7:0]								0000h
R3552 (DE0h)	DFC5MIX_Input_1_Source	DFC5_STS	0	0	0	0	0	0	0	DFC5_SRC [7:0]								0000h
R3560 (DE8h)	DFC6MIX_Input_1_Source	DFC6_STS	0	0	0	0	0	0	0	DFC6_SRC [7:0]								0000h
R3568 (DF0h)	DFC7MIX_Input_1_Source	DFC7_STS	0	0	0	0	0	0	0	DFC7_SRC [7:0]								0000h
R3576 (DF8h)	DFC8MIX_Input_1_Source	DFC8_STS	0	0	0	0	0	0	0	DFC8_SRC [7:0]								0000h
R3584 (E00h)	FX_Ctrl1	FX_RATE [4:0]					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

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
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
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

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
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
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

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
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
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
R3808 (EE0h)	ASRC1_ENABLE	0	0	0	0	0	0	0	0	0	0	0	0	ASRC1_IN2L_ENA	ASRC1_IN2R_ENA	ASRC1_IN1L_ENA	ASRC1_IN1R_ENA	0000h
R3809 (EE1h)	ASRC1_STATUS	0	0	0	0	0	0	0	0	0	0	0	0	ASRC1_IN2L_ENA_STS	ASRC1_IN2R_ENA_STS	ASRC1_IN1L_ENA_STS	ASRC1_IN1R_ENA_STS	0000h
R3810 (EE2h)	ASRC1_RATE1	ASRC1_RATE1 [4:0]					0	0	0	0	0	0	0	0	0	0	0	0000h
R3811 (EE3h)	ASRC1_RATE2	ASRC1_RATE2 [4:0]					0	0	0	0	0	0	0	0	0	0	0	4000h
R3824 (EF0h)	ISRC1_CTRL_1	ISRC1_FSH [4:0]					0	0	0	0	0	0	0	0	0	0	0	0000h
R3825 (EF1h)	ISRC1_CTRL_2	ISRC1_FSL [4:0]					0	0	0	0	0	0	0	0	0	0	1	0001h
R3826 (EF2h)	ISRC1_CTRL_3	ISRC1_INT1_ENA	ISRC1_INT2_ENA	0	0	0	0	ISRC1_DEC1_ENA	ISRC1_DEC2_ENA	0	0	0	0	0	0	0	0	0000h
R3827 (EF3h)	ISRC2_CTRL_1	ISRC2_FSH [4:0]					0	0	0	0	0	0	0	0	0	0	0	0000h
R3828 (EF4h)	ISRC2_CTRL_2	ISRC2_FSL [4:0]					0	0	0	0	0	0	0	0	0	0	1	0001h
R3829 (EF5h)	ISRC2_CTRL_3	ISRC2_INT1_ENA	ISRC2_INT2_ENA	0	0	0	0	ISRC2_DEC1_ENA	ISRC2_DEC2_ENA	0	0	0	0	0	0	0	0	0000h
R4224 (1080h)	US1_Ctrl_0	0	0	US1_GAIN [1:0]		US1_SRC [3:0]				0	US1_FREQ [2:0]			0	0	0	US1_ENA	2030h

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
R4225 (1081h)	US1_Ctrl_1	US1_RATE [4:0]					0	0	0	0	0	0	0	0	0	0	0	0000h
R4226 (1082h)	US2_Ctrl_0	0	0	US2_GAIN [1:0]		US2_SRC [3:0]				0	US2_FREQ [2:0]			0	0	0	US2_ENA	2030h
R4227 (1083h)	US2_Ctrl_1	US2_RATE [4:0]					0	0	0	0	0	0	0	0	0	0	0	0000h
R4288 (10C0h)	AUXPDM1_Ctrl_0	0	0	0	0	AUXPDM1_SRC[3:0]				0	0	0	AUXPDM1_TXEDGE	AUXPDM1_MSTR	AUXPDM1_MUTE	0	AUXPDM1_ENA	0008h
R4289 (10C1h)	AUXPDM1_Ctrl_1	AUXPDM1_CLK_FREQ[1:0]		0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000h
R5248 (1480h)	DFC1_CTRL_W0	0	0	0	0	0	0	0	0	0	DFC1_RATE [4:0]					DFC1_DITH_ENA	DFC1_ENA	0000h
R5250 (1482h)	DFC1_RX_W0	0	0	0	DFC1_RX_DATA_WIDTH [4:0]					0	0	0	0	0	DFC1_RX_DATA_TYPE [2:0]			1F00h
R5252 (1484h)	DFC1_TX_W0	0	0	0	DFC1_TX_DATA_WIDTH [4:0]					0	0	0	0	0	DFC1_TX_DATA_TYPE [2:0]			1F00h
R5254 (1486h)	DFC2_CTRL_W0	0	0	0	0	0	0	0	0	0	DFC2_RATE [4:0]					DFC2_DITH_ENA	DFC2_ENA	0000h
R5256 (1488h)	DFC2_RX_W0	0	0	0	DFC2_RX_DATA_WIDTH [4:0]					0	0	0	0	0	DFC2_RX_DATA_TYPE [2:0]			1F00h
R5258 (148Ah)	DFC2_TX_W0	0	0	0	DFC2_TX_DATA_WIDTH [4:0]					0	0	0	0	0	DFC2_TX_DATA_TYPE [2:0]			1F00h
R5260 (148Ch)	DFC3_CTRL_W0	0	0	0	0	0	0	0	0	0	DFC3_RATE [4:0]					DFC3_DITH_ENA	DFC3_ENA	0000h
R5262 (148Eh)	DFC3_RX_W0	0	0	0	DFC3_RX_DATA_WIDTH [4:0]					0	0	0	0	0	DFC3_RX_DATA_TYPE [2:0]			1F00h
R5264 (1490h)	DFC3_TX_W0	0	0	0	DFC3_TX_DATA_WIDTH [4:0]					0	0	0	0	0	DFC3_TX_DATA_TYPE [2:0]			1F00h
R5266 (1492h)	DFC4_CTRL_W0	0	0	0	0	0	0	0	0	0	DFC4_RATE [4:0]					DFC4_DITH_ENA	DFC4_ENA	0000h
R5268 (1494h)	DFC4_RX_W0	0	0	0	DFC4_RX_DATA_WIDTH [4:0]					0	0	0	0	0	DFC4_RX_DATA_TYPE [2:0]			1F00h
R5270 (1496h)	DFC4_TX_W0	0	0	0	DFC4_TX_DATA_WIDTH [4:0]					0	0	0	0	0	DFC4_TX_DATA_TYPE [2:0]			1F00h
R5272 (1498h)	DFC5_CTRL_W0	0	0	0	0	0	0	0	0	0	DFC5_RATE [4:0]					DFC5_DITH_ENA	DFC5_ENA	0000h
R5274 (149Ah)	DFC5_RX_W0	0	0	0	DFC5_RX_DATA_WIDTH [4:0]					0	0	0	0	0	DFC5_RX_DATA_TYPE [2:0]			1F00h
R5276 (149Ch)	DFC5_TX_W0	0	0	0	DFC5_TX_DATA_WIDTH [4:0]					0	0	0	0	0	DFC5_TX_DATA_TYPE [2:0]			1F00h
R5278 (149Eh)	DFC6_CTRL_W0	0	0	0	0	0	0	0	0	0	DFC6_RATE [4:0]					DFC6_DITH_ENA	DFC6_ENA	0000h
R5280 (14A0h)	DFC6_RX_W0	0	0	0	DFC6_RX_DATA_WIDTH [4:0]					0	0	0	0	0	DFC6_RX_DATA_TYPE [2:0]			1F00h
R5282 (14A2h)	DFC6_TX_W0	0	0	0	DFC6_TX_DATA_WIDTH [4:0]					0	0	0	0	0	DFC6_TX_DATA_TYPE [2:0]			1F00h
R5284 (14A4h)	DFC7_CTRL_W0	0	0	0	0	0	0	0	0	0	DFC7_RATE [4:0]					DFC7_DITH_ENA	DFC7_ENA	0000h
R5286 (14A6h)	DFC7_RX_W0	0	0	0	DFC7_RX_DATA_WIDTH [4:0]					0	0	0	0	0	DFC7_RX_DATA_TYPE [2:0]			1F00h
R5288 (14A8h)	DFC7_TX_W0	0	0	0	DFC7_TX_DATA_WIDTH [4:0]					0	0	0	0	0	DFC7_TX_DATA_TYPE [2:0]			1F00h
R5290 (14AAh)	DFC8_CTRL_W0	0	0	0	0	0	0	0	0	0	DFC8_RATE [4:0]					DFC8_DITH_ENA	DFC8_ENA	0000h
R5292 (14ACh)	DFC8_RX_W0	0	0	0	DFC8_RX_DATA_WIDTH [4:0]					0	0	0	0	0	DFC8_RX_DATA_TYPE [2:0]			1F00h
R5294 (14AEh)	DFC8_TX_W0	0	0	0	DFC8_TX_DATA_WIDTH [4:0]					0	0	0	0	0	DFC8_TX_DATA_TYPE [2:0]			1F00h
R5302 (14B6h)	DFC_STATUS_W0	0	0	0	0	0	0	0	0	DFC_ERR_CHAN [7:0]								0000h
R5303 (14B7h)	DFC_STATUS_W1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DFC_DITH_TYPE [1:0]		0000h
R5632 (1600h)	ADSP2_IRQ0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP_IRQ2	DSP_IRQ1	0000h
R5633 (1601h)	ADSP2_IRQ1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP_IRQ4	DSP_IRQ3	0000h
R5634 (1602h)	ADSP2_IRQ2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP_IRQ6	DSP_IRQ5	0000h
R5635 (1603h)	ADSP2_IRQ3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP_IRQ8	DSP_IRQ7	0000h
R5636 (1604h)	ADSP2_IRQ4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP_IRQ10	DSP_IRQ9	0000h
R5637 (1605h)	ADSP2_IRQ5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP_IRQ12	DSP_IRQ11	0000h
R5638 (1606h)	ADSP2_IRQ6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP_IRQ14	DSP_IRQ13	0000h

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
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	GP1_FN [9:0]										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	GP2_FN [9:0]										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	GP3_FN [9:0]										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	GP4_FN [9:0]										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	GP5_FN [9:0]										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	GP6_FN [9:0]										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	GP7_FN [9:0]										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	GP8_FN [9:0]										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	GP9_FN [9:0]										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	GP10_FN [9:0]										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	GP11_FN [9:0]										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	GP12_FN [9:0]										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	GP13_FN [9:0]										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	GP14_FN [9:0]										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	GP15_FN [9:0]										2001h
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	GP16_FN [9:0]										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	0	0	0	CTRLIF_ERR_EINT1	0	0	SYSCLK_FAIL_EINT1	0	BOOT_DONE_EINT1	0	0	0	0	0	0	0	0000h
R6145 (1801h)	IRQ1_Status_2	0	0	CLK_ASYNC_ERR_EINT1	CLK_SYS_ERR_EINT1	0	0	FLL2_LOCK_EINT1	FLL1_LOCK_EINT1	0	FLL2_REF_LOST_EINT1	FLL1_REF_LOST_EINT1	0	0	0	0	0	0000h
R6149 (1805h)	IRQ1_Status_6	0	0	0	0	0	0	MICDET2_EINT1	MICDET1_EINT1	0	0	0	0	0	0	0	HPDET_EINT1	0000h
R6150 (1806h)	IRQ1_Status_7	0	0	0	0	MICD_CLAMP2_FALL_EINT1	MICD_CLAMP2_RISE_EINT1	JD3_FALL_EINT1	JD3_RISE_EINT1	0	0	MICD_CLAMP1_FALL_EINT1	MICD_CLAMP1_RISE_EINT1	JD2_FALL_EINT1	JD2_RISE_EINT1	JD1_FALL_EINT1	JD1_RISE_EINT1	0000h

Table 6-1. Register Map Definition—16-bit region *(Cont.)*

Register	Name	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Default
R6152 (1808h)	IRQ1_Status_9	0	0	0	0	0	0	ASRC1_IN2_LOCK_EINT1	ASRC1_IN1_LOCK_EINT1	0	0	0	0	0	INPUTS_SIG_DET_EINT1	DRC2_SIG_DET_EINT1	DRC1_SIG_DET_EINT1	0000h
R6154 (180Ah)	IRQ1_Status_11	DSP_IRQ16_EINT1	DSP_IRQ15_EINT1	DSP_IRQ14_EINT1	DSP_IRQ13_EINT1	DSP_IRQ12_EINT1	DSP_IRQ11_EINT1	DSP_IRQ10_EINT1	DSP_IRQ9_EINT1	DSP_IRQ8_EINT1	DSP_IRQ7_EINT1	DSP_IRQ6_EINT1	DSP_IRQ5_EINT1	DSP_IRQ4_EINT1	DSP_IRQ3_EINT1	DSP_IRQ2_EINT1	DSP_IRQ1_EINT1	0000h
R6155 (180Bh)	IRQ1_Status_12	0	0	0	0	0	0	HP4R_SC_EINT1	HP4L_SC_EINT1	0	0	HP3R_SC_EINT1	HP3L_SC_EINT1	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	0	HP3R_ENABLE_DONE_EINT1	HP3L_ENABLE_DONE_EINT1	HP2R_ENABLE_DONE_EINT1	HP2L_ENABLE_DONE_EINT1	HP1R_ENABLE_DONE_EINT1	HP1L_ENABLE_DONE_EINT1	0000h
R6157 (180Dh)	IRQ1_Status_14	0	0	0	0	0	0	0	0	0	0	HP3R_DISABLE_DONE_EINT1	HP3L_DISABLE_DONE_EINT1	HP2R_DISABLE_DONE_EINT1	HP2L_DISABLE_DONE_EINT1	HP1R_DISABLE_DONE_EINT1	HP1L_DISABLE_DONE_EINT1	0000h
R6158 (180Eh)	IRQ1_Status_15	0	0	0	DFC_SATURATE_EINT1	0	0	0	0	0	0	0	0	0	0	0	0	0000h
R6160 (1810h)	IRQ1_Status_17	GP16_EINT1	GP15_EINT1	GP14_EINT1	GP13_EINT1	GP12_EINT1	GP11_EINT1	GP10_EINT1	GP9_EINT1	GP8_EINT1	GP7_EINT1	GP6_EINT1	GP5_EINT1	GP4_EINT1	GP3_EINT1	GP2_EINT1	GP1_EINT1	0000h
R6164 (1814h)	IRQ1_Status_21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TIMER1_EINT1	0000h
R6165 (1815h)	IRQ1_Status_22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT1_NOT_EMPTY_EINT1	0000h
R6166 (1816h)	IRQ1_Status_23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT1_FULL_EINT1	0000h
R6167 (1817h)	IRQ1_Status_24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT1_WMARK_EINT1	0000h
R6168 (1818h)	IRQ1_Status_25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_DMA_EINT1	0000h
R6170 (181Ah)	IRQ1_Status_27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_START1_EINT1	0000h
R6171 (181Bh)	IRQ1_Status_28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_START2_EINT1	0000h
R6173 (181Dh)	IRQ1_Status_30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_BUSY_EINT1	0000h
R6176 (1820h)	IRQ1_Status_33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_BUS_ERR_EINT1	0000h
R6179 (1823h)	IRQ1_Status_36	0	0	0	0	0	0	0	0	0	0	0	0	TIMER_ALM1_CH4_EINT1	TIMER_ALM1_CH3_EINT1	TIMER_ALM1_CH2_EINT1	TIMER_ALM1_CH1_EINT1	0000h
R6183 (1827h)	IRQ1_Status_40	0	0	0	0	0	0	FLL2_SYNC_ACTIVE_EINT1	FLL1_SYNC_ACTIVE_EINT1	0	0	0	0	0	0	0	0	0000h
R6208 (1840h)	IRQ1_Mask_1	0	0	0	IM_CTRLIF_ERR_EINT1	0	0	IM_SYSCLK_FAIL_EINT1	0	IM_BOOT_DONE_EINT1	0	0	0	0	0	0	0	1200h
R6209 (1841h)	IRQ1_Mask_2	0	1	IM_CLK_ASYNC_ERR_EINT1	IM_CLK_SYS_ERR_EINT1	0	0	IM_FLL2_LOCK_EINT1	IM_FLL1_LOCK_EINT1	0	IM_FLL2_REF_LOST_EINT1	IM_FLL1_REF_LOST_EINT1	0	0	0	0	0	77E0h
R6213 (1845h)	IRQ1_Mask_6	0	0	0	0	0	0	IM_MICDET2_EINT1	IM_MICDET1_EINT1	0	0	0	0	0	0	0	IM_HPDET_EINT1	0301h
R6214 (1846h)	IRQ1_Mask_7	0	0	0	0	IM_MICD_CLAMP2_FALL_EINT1	IM_MICD_CLAMP2_RISE_EINT1	IM_JD3_FALL_EINT1	IM_JD3_RISE_EINT1	0	0	IM_MICD_CLAMP1_FALL_EINT1	IM_MICD_CLAMP1_RISE_EINT1	IM_JD2_FALL_EINT1	IM_JD2_RISE_EINT1	IM_JD1_FALL_EINT1	IM_JD1_RISE_EINT1	0F3Fh
R6216 (1848h)	IRQ1_Mask_9	0	0	0	0	0	0	IM_ASRC1_IN2_LOCK_EINT1	IM_ASRC1_IN1_LOCK_EINT1	0	0	0	1	1	IM_INPUTS_SIG_DET_EINT1	IM_DRC2_SIG_DET_EINT1	IM_DRC1_SIG_DET_EINT1	031Fh
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	IM_HP4R_SC_EINT1	IM_HP4L_SC_EINT1	0	0	IM_HP3R_SC_EINT1	IM_HP3L_SC_EINT1	IM_HP2R_SC_EINT1	IM_HP2L_SC_EINT1	IM_HP1R_SC_EINT1	IM_HP1L_SC_EINT1	033Fh
R6220 (184Ch)	IRQ1_Mask_13	0	0	0	0	0	0	0	0	0	0	IM_HP3R_ENABLE_DONE_EINT1	IM_HP3L_ENABLE_DONE_EINT1	IM_HP2R_ENABLE_DONE_EINT1	IM_HP2L_ENABLE_DONE_EINT1	IM_HP1R_ENABLE_DONE_EINT1	IM_HP1L_ENABLE_DONE_EINT1	003Fh
R6221 (184Dh)	IRQ1_Mask_14	0	0	0	0	0	0	0	0	0	0	IM_HP3R_DISABLE_DONE_EINT1	IM_HP3L_DISABLE_DONE_EINT1	IM_HP2R_DISABLE_DONE_EINT1	IM_HP2L_DISABLE_DONE_EINT1	IM_HP1R_DISABLE_DONE_EINT1	IM_HP1L_DISABLE_DONE_EINT1	003Fh
R6222 (184Eh)	IRQ1_Mask_15	0	0	0	IM_DFC_SATURATE_EINT1	0	0	0	0	0	0	0	0	0	0	0	0	1000h

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
R6224 (1850h)	IRQ1_Mask_17	IM_GP16_EINT1	IM_GP15_EINT1	IM_GP14_EINT1	IM_GP13_EINT1	IM_GP12_EINT1	IM_GP11_EINT1	IM_GP10_EINT1	IM_GP9_EINT1	IM_GP8_EINT1	IM_GP7_EINT1	IM_GP6_EINT1	IM_GP5_EINT1	IM_GP4_EINT1	IM_GP3_EINT1	IM_GP2_EINT1	IM_GP1_EINT1	FFFFh
R6228 (1854h)	IRQ1_Mask_21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_TIMER1_EINT1	0001h
R6229 (1855h)	IRQ1_Mask_22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_EVENT1_NOT_EMPTY_EINT1	0001h
R6230 (1856h)	IRQ1_Mask_23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_EVENT1_FULL_EINT1	0001h
R6231 (1857h)	IRQ1_Mask_24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_EVENT1_WMARK_EINT1	0001h
R6232 (1858h)	IRQ1_Mask_25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP1_DMA_EINT1	0001h
R6234 (185Ah)	IRQ1_Mask_27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP1_START1_EINT1	0001h
R6235 (185Bh)	IRQ1_Mask_28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP1_START2_EINT1	0001h
R6237 (185Dh)	IRQ1_Mask_30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP1_BUSY_EINT1	0001h
R6240 (1860h)	IRQ1_Mask_33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP1_BUS_ERR_EINT1	0001h
R6243 (1863h)	IRQ1_Mask_36	0	0	0	0	0	0	0	0	0	0	0	0	IM_TIMER_ALM1_CH4_EINT1	IM_TIMER_ALM1_CH3_EINT1	IM_TIMER_ALM1_CH2_EINT1	IM_TIMER_ALM1_CH1_EINT1	000Fh
R6247 (1867h)	IRQ1_Mask_40	0	0	0	0	0	0	IM_FLL2_SYNC_ACTIVE_EINT1	IM_FLL1_SYNC_ACTIVE_EINT1	0	0	0	0	0	0	0	0	0300h
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	CLK_ASYNC_ERR_STS1	CLK_SYS_ERR_STS1	0	0	FLL2_LOCK_STS1	FLL1_LOCK_STS1	0	FLL2_REF_LOST_STS1	FLL1_REF_LOST_STS1	0	0	0	0	0	0000h
R6278 (1886h)	IRQ1_Raw_Status_7	0	0	0	0	0	0	0	JD3_STS1	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	ASRC1_IN2_LOCK_STS1	ASRC1_IN1_LOCK_STS1	0	0	0	0	0	INPUTS_SIG_DET_STS1	DRC2_SIG_DET_STS1	DRC1_SIG_DET_STS1	0000h
R6283 (188Bh)	IRQ1_Raw_Status_12	0	0	0	0	0	0	HP4R_SC_STS1	HP4L_SC_STS1	0	0	HP3R_SC_STS1	HP3L_SC_STS1	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	0	HP3R_ENABLE_DONE_STS1	HP3L_ENABLE_DONE_STS1	HP2R_ENABLE_DONE_STS1	HP2L_ENABLE_DONE_STS1	HP1R_ENABLE_DONE_STS1	HP1L_ENABLE_DONE_STS1	0000h
R6285 (188Dh)	IRQ1_Raw_Status_14	0	0	0	0	0	0	0	0	0	0	HP3R_DISABLE_DONE_STS1	HP3L_DISABLE_DONE_STS1	HP2R_DISABLE_DONE_STS1	HP2L_DISABLE_DONE_STS1	HP1R_DISABLE_DONE_STS1	HP1L_DISABLE_DONE_STS1	0000h
R6288 (1890h)	IRQ1_Raw_Status_17	GP16_STS1	GP15_STS1	GP14_STS1	GP13_STS1	GP12_STS1	GP11_STS1	GP10_STS1	GP9_STS1	GP8_STS1	GP7_STS1	GP6_STS1	GP5_STS1	GP4_STS1	GP3_STS1	GP2_STS1	GP1_STS1	0000h
R6292 (1894h)	IRQ1_Raw_Status_21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TIMER1_STS1	0000h
R6293 (1895h)	IRQ1_Raw_Status_22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT1_NOT_EMPTY_STS1	0000h
R6294 (1896h)	IRQ1_Raw_Status_23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT1_FULL_STS1	0000h
R6295 (1897h)	IRQ1_Raw_Status_24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT1_WMARK_STS1	0000h
R6296 (1898h)	IRQ1_Raw_Status_25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_DMA_STS1	0000h
R6301 (189Dh)	IRQ1_Raw_Status_30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_BUSY_STS1	0000h
R6304 (18A0h)	IRQ1_Raw_Status_33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_BUS_ERR_STS1	0000h
R6307 (18A3h)	IRQ1_Raw_Status_36	0	0	0	0	0	0	0	0	0	0	0	0	TIMER_ALM1_CH4_STS1	TIMER_ALM1_CH3_STS1	TIMER_ALM1_CH2_STS1	TIMER_ALM1_CH1_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
R6311 (18A7h)	IRQ1_Raw_Status_40	0	0	0	0	0	0	FLL2_SYNC_ACTIVE_STS1	FLL1_SYNC_ACTIVE_STS1	0	0	0	0	0	0	0	0	0000h
R6400 (1900h)	IRQ2_Status_1	0	0	0	CTRLIF_ERR_EINT2	0	0	SYSCLK_FAIL_EINT2	0	BOOT_DONE_EINT2	0	0	0	0	0	0	0	0000h
R6401 (1901h)	IRQ2_Status_2	0	0	CLK_ASYNC_ERR_EINT2	CLK_SYS_ERR_EINT2	0	0	FLL2_LOCK_EINT2	FLL1_LOCK_EINT2	0	FLL2_REF_LOST_EINT2	FLL1_REF_LOST_EINT2	0	0	0	0	0	0000h
R6405 (1905h)	IRQ2_Status_6	0	0	0	0	0	0	MICDET2_EINT2	MICDET1_EINT2	0	0	0	0	0	0	0	HPDET_EINT2	0000h
R6406 (1906h)	IRQ2_Status_7	0	0	0	0	0	0	JD3_FALL_EINT2	JD3_RISE_EINT2	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	ASRC1_IN2_LOCK_EINT2	ASRC1_IN1_LOCK_EINT2	0	0	0	0	0	INPUTS_SIG_DET_EINT2	DRC2_SIG_DET_EINT2	DRC1_SIG_DET_EINT2	0000h
R6410 (190Ah)	IRQ2_Status_11	DSP_IRQ16_EINT2	DSP_IRQ15_EINT2	DSP_IRQ14_EINT2	DSP_IRQ13_EINT2	DSP_IRQ12_EINT2	DSP_IRQ11_EINT2	DSP_IRQ10_EINT2	DSP_IRQ9_EINT2	DSP_IRQ8_EINT2	DSP_IRQ7_EINT2	DSP_IRQ6_EINT2	DSP_IRQ5_EINT2	DSP_IRQ4_EINT2	DSP_IRQ3_EINT2	DSP_IRQ2_EINT2	DSP_IRQ1_EINT2	0000h
R6411 (190Bh)	IRQ2_Status_12	0	0	0	0	0	0	HP4R_SC_EINT2	HP4L_SC_EINT2	0	0	HP3R_SC_EINT2	HP3L_SC_EINT2	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	0	HP3R_ENABLE_DONE_EINT2	HP3L_ENABLE_DONE_EINT2	HP2R_ENABLE_DONE_EINT2	HP2L_ENABLE_DONE_EINT2	HP1R_ENABLE_DONE_EINT2	HP1L_ENABLE_DONE_EINT2	0000h
R6413 (190Dh)	IRQ2_Status_14	0	0	0	0	0	0	0	0	0	0	HP3R_DISABLE_DONE_EINT2	HP3L_DISABLE_DONE_EINT2	HP2R_DISABLE_DONE_EINT2	HP2L_DISABLE_DONE_EINT2	HP1R_DISABLE_DONE_EINT2	HP1L_DISABLE_DONE_EINT2	0000h
R6414 (190Eh)	IRQ2_Status_15	0	0	0	DFC_SATURATE_EINT2	0	0	0	0	0	0	0	0	0	0	0	0	0000h
R6416 (1910h)	IRQ2_Status_17	GP16_EINT2	GP15_EINT2	GP14_EINT2	GP13_EINT2	GP12_EINT2	GP11_EINT2	GP10_EINT2	GP9_EINT2	GP8_EINT2	GP7_EINT2	GP6_EINT2	GP5_EINT2	GP4_EINT2	GP3_EINT2	GP2_EINT2	GP1_EINT2	0000h
R6420 (1914h)	IRQ2_Status_21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TIMER1_EINT2	0000h
R6421 (1915h)	IRQ2_Status_22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT1_NOT_EMPTY_EINT2	0000h
R6422 (1916h)	IRQ2_Status_23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT1_FULL_EINT2	0000h
R6423 (1917h)	IRQ2_Status_24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT1_WMARK_EINT2	0000h
R6424 (1918h)	IRQ2_Status_25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_DMA_EINT2	0000h
R6426 (191Ah)	IRQ2_Status_27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_START1_EINT2	0000h
R6427 (191Bh)	IRQ2_Status_28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_START2_EINT2	0000h
R6429 (191Dh)	IRQ2_Status_30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_BUSY_EINT2	0000h
R6432 (1920h)	IRQ2_Status_33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_BUS_ERR_EINT2	0000h
R6435 (1923h)	IRQ2_Status_36	0	0	0	0	0	0	0	0	0	0	0	0	TIMER_ALM1_CH4_EINT2	TIMER_ALM1_CH3_EINT2	TIMER_ALM1_CH2_EINT2	TIMER_ALM1_CH1_EINT2	0000h
R6439 (1927h)	IRQ2_Status_40	0	0	0	0	0	0	FLL2_SYNC_ACTIVE_EINT2	FLL1_SYNC_ACTIVE_EINT2	0	0	0	0	0	0	0	0	0000h
R6464 (1940h)	IRQ2_Mask_1	0	0	0	IM_CTRLIF_ERR_EINT2	0	0	IM_SYSCLK_FAIL_EINT2	0	IM_BOOT_DONE_EINT2	0	0	0	0	0	0	0	1280h
R6465 (1941h)	IRQ2_Mask_2	0	1	IM_CLK_ASYNC_ERR_EINT2	IM_CLK_SYS_ERR_EINT2	0	0	IM_FLL2_LOCK_EINT2	IM_FLL1_LOCK_EINT2	0	IM_FLL2_REF_LOST_EINT2	IM_FLL1_REF_LOST_EINT2	0	0	0	0	0	77E0h
R6469 (1945h)	IRQ2_Mask_6	0	0	0	0	0	0	IM_MICDET2_EINT2	IM_MICDET1_EINT2	0	0	0	0	0	0	0	IM_HPDET_EINT2	0301h
R6470 (1946h)	IRQ2_Mask_7	0	0	0	0	0	0	IM_JD3_FALL_EINT2	IM_JD3_RISE_EINT2	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	033Fh

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	IM_ASRC1_IN2_LOCK_EINT2	IM_ASRC1_IN1_LOCK_EINT2	0	0	0	1	1	IM_INPUTS_SIG_DET_EINT2	IM_DRC2_SIG_DET_EINT2	IM_DRC1_SIG_DET_EINT2	031Fh
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	IM_HP4R_SC_EINT2	IM_HP4L_SC_EINT2	0	0	IM_HP3R_SC_EINT2	IM_HP3L_SC_EINT2	IM_HP2R_SC_EINT2	IM_HP2L_SC_EINT2	IM_HP1R_SC_EINT2	IM_HP1L_SC_EINT2	033Fh
R6476 (194Ch)	IRQ2_Mask_13	0	0	0	0	0	0	0	0	0	0	IM_HP3R_ENABLE_DONE_EINT2	IM_HP3L_ENABLE_DONE_EINT2	IM_HP2R_ENABLE_DONE_EINT2	IM_HP2L_ENABLE_DONE_EINT2	IM_HP1R_ENABLE_DONE_EINT2	IM_HP1L_ENABLE_DONE_EINT2	003Fh
R6477 (194Dh)	IRQ2_Mask_14	0	0	0	0	0	0	0	0	0	0	IM_HP3R_DISABLE_DONE_EINT2	IM_HP3L_DISABLE_DONE_EINT2	IM_HP2R_DISABLE_DONE_EINT2	IM_HP2L_DISABLE_DONE_EINT2	IM_HP1R_DISABLE_DONE_EINT2	IM_HP1L_DISABLE_DONE_EINT2	003Fh
R6478 (194Eh)	IRQ2_Mask_15	0	0	0	IM_DFC_SATURATE_EINT2	0	0	0	0	0	0	0	0	0	0	0	0	1000h
R6480 (1950h)	IRQ2_Mask_17	IM_GP16_EINT2	IM_GP15_EINT2	IM_GP14_EINT2	IM_GP13_EINT2	IM_GP12_EINT2	IM_GP11_EINT2	IM_GP10_EINT2	IM_GP9_EINT2	IM_GP8_EINT2	IM_GP7_EINT2	IM_GP6_EINT2	IM_GP5_EINT2	IM_GP4_EINT2	IM_GP3_EINT2	IM_GP2_EINT2	IM_GP1_EINT2	FFFFh
R6484 (1954h)	IRQ2_Mask_21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_TIMER1_EINT2	0001h
R6485 (1955h)	IRQ2_Mask_22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_EVENT1_NOT_EMPTY_EINT2	0001h
R6486 (1956h)	IRQ2_Mask_23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_EVENT1_FULL_EINT2	0001h
R6487 (1957h)	IRQ2_Mask_24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_EVENT1_WMARK_EINT2	0001h
R6488 (1958h)	IRQ2_Mask_25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP1_DMA_EINT2	0001h
R6490 (195Ah)	IRQ2_Mask_27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP1_START1_EINT2	0001h
R6491 (195Bh)	IRQ2_Mask_28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP1_START2_EINT2	0001h
R6493 (195Dh)	IRQ2_Mask_30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP1_BUSY_EINT2	0001h
R6496 (1960h)	IRQ2_Mask_33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IM_DSP1_BUS_ERR_EINT2	0001h
R6499 (1963h)	IRQ2_Mask_36	0	0	0	0	0	0	0	0	0	0	0	0	IM_TIMER_ALM1_CH4_EINT2	IM_TIMER_ALM1_CH3_EINT2	IM_TIMER_ALM1_CH2_EINT2	IM_TIMER_ALM1_CH1_EINT2	000Fh
R6503 (1967h)	IRQ2_Mask_40	0	0	0	0	0	0	IM_FLL2_SYNC_ACTIVE_EINT2	IM_FLL1_SYNC_ACTIVE_EINT2	0	0	0	0	0	0	0	0	0300h
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	CLK_ASYNC_ERR_STS2	CLK_SYS_ERR_STS2	0	0	FLL2_LOCK_STS2	FLL1_LOCK_STS2	0	FLL2_REF_LOST_STS2	FLL1_REF_LOST_STS2	0	0	0	0	0	0000h
R6534 (1986h)	IRQ2_Raw_Status_7	0	0	0	0	0	0	0	JD3_STS2	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	ASRC1_IN2_LOCK_STS2	ASRC1_IN1_LOCK_STS2	0	0	0	0	0	INPUTS_SIG_DET_STS2	DRC2_SIG_DET_STS2	DRC1_SIG_DET_STS2	0000h
R6539 (198Bh)	IRQ2_Raw_Status_12	0	0	0	0	0	0	HP4R_SC_STS2	HP4L_SC_STS2	0	0	HP3R_SC_STS2	HP3L_SC_STS2	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	0	HP3R_ENABLE_DONE_STS2	HP3L_ENABLE_DONE_STS2	HP2R_ENABLE_DONE_STS2	HP2L_ENABLE_DONE_STS2	HP1R_ENABLE_DONE_STS2	HP1L_ENABLE_DONE_STS2	0000h
R6541 (198Dh)	IRQ2_Raw_Status_14	0	0	0	0	0	0	0	0	0	0	HP3R_DISABLE_DONE_STS2	HP3L_DISABLE_DONE_STS2	HP2R_DISABLE_DONE_STS2	HP2L_DISABLE_DONE_STS2	HP1R_DISABLE_DONE_STS2	HP1L_DISABLE_DONE_STS2	0000h
R6544 (1990h)	IRQ2_Raw_Status_17	GP16_STS2	GP15_STS2	GP14_STS2	GP13_STS2	GP12_STS2	GP11_STS2	GP10_STS2	GP9_STS2	GP8_STS2	GP7_STS2	GP6_STS2	GP5_STS2	GP4_STS2	GP3_STS2	GP2_STS2	GP1_STS2	0000h
R6548 (1994h)	IRQ2_Raw_Status_21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TIMER1_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
R6549 (1995h)	IRQ2_Raw_Status_22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT1_NOT_EMPTY_STS2	0000h
R6550 (1996h)	IRQ2_Raw_Status_23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT1_FULL_STS2	0000h
R6551 (1997h)	IRQ2_Raw_Status_24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	EVENT1_WMARK_STS2	0000h
R6552 (1998h)	IRQ2_Raw_Status_25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_DMA_STS2	0000h
R6557 (199Dh)	IRQ2_Raw_Status_30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_BUSY_STS2	0000h
R6560 (19A0h)	IRQ2_Raw_Status_33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_BUS_ERR_STS2	0000h
R6563 (19A3h)	IRQ2_Raw_Status_36	0	0	0	0	0	0	0	0	0	0	0	0	TIMER_ALM1_CH4_STS2	TIMER_ALM1_CH3_STS2	TIMER_ALM1_CH2_STS2	TIMER_ALM1_CH1_STS2	0000h
R6567 (19A7h)	IRQ2_Raw_Status_40	0	0	0	0	0	0	FLL2_SYNC_ACTIVE_STS2	FLL1_SYNC_ACTIVE_STS2	0	0	0	0	0	0	0	0	0000h
R6662 (1A06h)	Interrupt_Debounce_7	0	0	0	0	0	0	0	JD3_DB	0	0	0	MICD_CLAMP_DB	0	JD2_DB	0	JD1_DB	0000h
R6784 (1A80h)	IRQ1_CTRL	0	1	0	0	IM_IRQ1	IRQ_POL	IRQ_OP_CFG	0	0	0	0	0	0	0	0	0	4400h
R6786 (1A82h)	IRQ2_CTRL	0	0	0	0	IM_IRQ2	0	0	0	0	0	0	0	0	0	0	0	0000h
R6816 (1AA0h)	Interrupt_Raw_Status_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	IRQ2_STS	IRQ1_STS	0000h
R6848 (1AC0h)	GPIO_Debounce_Config	0	0	0	0	0	0	0	0	0	0	0	0	GP_DBTIME [3:0]				0000h
R6864 (1AD0h)	AOD_Pad_Ctrl	0	1	0	0	0	0	0	0	0	0	0	0	0	0	RESET_PU	RESET_PD	4002h

The 32-bit register space is described in Table 6-2.

Table 6-2. Register Map Definition—32-bit region

Register	Name	31 15	30 14	29 13	28 12	27 11	26 10	25 9	24 8	23 7	22 6	21 5	20 4	19 3	18 2	17 1	16 0	Default
R12288 (3000h)	WSEQ_Sequence_1	WSEQ_DATA_WIDTH0 [2:0]			WSEQ_ADDR0 [12:0]													0000F000h
		WSEQ_DELAY0 [3:0]				WSEQ_DATA_START0 [3:0]				WSEQ_DATA0 [7:0]								
R12290 (3002h)	WSEQ_Sequence_2	WSEQ_DATA_WIDTH1 [2:0]			WSEQ_ADDR1 [12:0]													0000F000h
		WSEQ_DELAY1 [3:0]				WSEQ_DATA_START1 [3:0]				WSEQ_DATA1 [7:0]								
R12292 (3004h)	WSEQ_Sequence_3	WSEQ_DATA_WIDTH2 [2:0]			WSEQ_ADDR2 [12:0]													0000F000h
		WSEQ_DELAY2 [3:0]				WSEQ_DATA_START2 [3:0]				WSEQ_DATA2 [7:0]								
R12294 (3006h)	WSEQ_Sequence_4	WSEQ_DATA_WIDTH3 [2:0]			WSEQ_ADDR3 [12:0]													0000F000h
		WSEQ_DELAY3 [3:0]				WSEQ_DATA_START3 [3:0]				WSEQ_DATA3 [7:0]								
R12296 (3008h)	WSEQ_Sequence_5	WSEQ_DATA_WIDTH4 [2:0]			WSEQ_ADDR4 [12:0]													82253719h
		WSEQ_DELAY4 [3:0]				WSEQ_DATA_START4 [3:0]				WSEQ_DATA4 [7:0]								
R12298 (300Ah)	WSEQ_Sequence_6	WSEQ_DATA_WIDTH5 [2:0]			WSEQ_ADDR5 [12:0]													C2300001h
		WSEQ_DELAY5 [3:0]				WSEQ_DATA_START5 [3:0]				WSEQ_DATA5 [7:0]								
R12300 (300Ch)	WSEQ_Sequence_7	WSEQ_DATA_WIDTH6 [2:0]			WSEQ_ADDR6 [12:0]													02251301h
		WSEQ_DELAY6 [3:0]				WSEQ_DATA_START6 [3:0]				WSEQ_DATA6 [7:0]								
R12302 (300Eh)	WSEQ_Sequence_8	WSEQ_DATA_WIDTH7 [2:0]			WSEQ_ADDR7 [12:0]													8225191Fh
		WSEQ_DELAY7 [3:0]				WSEQ_DATA_START7 [3:0]				WSEQ_DATA7 [7:0]								
R12304 (3010h)	WSEQ_Sequence_9	WSEQ_DATA_WIDTH8 [2:0]			WSEQ_ADDR8 [12:0]													82310B00h
		WSEQ_DELAY8 [3:0]				WSEQ_DATA_START8 [3:0]				WSEQ_DATA8 [7:0]								
R12306 (3012h)	WSEQ_Sequence_10	WSEQ_DATA_WIDTH9 [2:0]			WSEQ_ADDR9 [12:0]													E231023Bh
		WSEQ_DELAY9 [3:0]				WSEQ_DATA_START9 [3:0]				WSEQ_DATA9 [7:0]								
R12308 (3014h)	WSEQ_Sequence_11	WSEQ_DATA_WIDTH10 [2:0]			WSEQ_ADDR10 [12:0]													02313B01h
		WSEQ_DELAY10 [3:0]				WSEQ_DATA_START10 [3:0]				WSEQ_DATA10 [7:0]								
R12310 (3016h)	WSEQ_Sequence_12	WSEQ_DATA_WIDTH11 [2:0]			WSEQ_ADDR11 [12:0]													62300000h
		WSEQ_DELAY11 [3:0]				WSEQ_DATA_START11 [3:0]				WSEQ_DATA11 [7:0]								
R12312 (3018h)	WSEQ_Sequence_13	WSEQ_DATA_WIDTH12 [2:0]			WSEQ_ADDR12 [12:0]													E2314288h
		WSEQ_DELAY12 [3:0]				WSEQ_DATA_START12 [3:0]				WSEQ_DATA12 [7:0]								
R12314 (301Ah)	WSEQ_Sequence_14	WSEQ_DATA_WIDTH13 [2:0]			WSEQ_ADDR13 [12:0]													02310B00h
		WSEQ_DELAY13 [3:0]				WSEQ_DATA_START13 [3:0]				WSEQ_DATA13 [7:0]								
R12316 (301Ch)	WSEQ_Sequence_15	WSEQ_DATA_WIDTH14 [2:0]			WSEQ_ADDR14 [12:0]													02310B00h
		WSEQ_DELAY14 [3:0]				WSEQ_DATA_START14 [3:0]				WSEQ_DATA14 [7:0]								
R12318 (301Eh)	WSEQ_Sequence_16	WSEQ_DATA_WIDTH15 [2:0]			WSEQ_ADDR15 [12:0]													02250E01h
		WSEQ_DELAY15 [3:0]				WSEQ_DATA_START15 [3:0]				WSEQ_DATA15 [7:0]								

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
R12320 (3020h)	WSEQ_Sequence_17	WSEQ_DATA_WIDTH16 [2:0]			WSEQ_ADDR16 [12:0]													42310C02h
		WSEQ_DELAY16 [3:0]				WSEQ_DATA_START16 [3:0]				WSEQ_DATA16 [7:0]								
R12322 (3022h)	WSEQ_Sequence_18	WSEQ_DATA_WIDTH17 [2:0]			WSEQ_ADDR17 [12:0]													E2310227h
		WSEQ_DELAY17 [3:0]				WSEQ_DATA_START17 [3:0]				WSEQ_DATA17 [7:0]								
R12324 (3024h)	WSEQ_Sequence_19	WSEQ_DATA_WIDTH18 [2:0]			WSEQ_ADDR18 [12:0]													02313B01h
		WSEQ_DELAY18 [3:0]				WSEQ_DATA_START18 [3:0]				WSEQ_DATA18 [7:0]								
R12326 (3026h)	WSEQ_Sequence_20	WSEQ_DATA_WIDTH19 [2:0]			WSEQ_ADDR19 [12:0]													E2314266h
		WSEQ_DELAY19 [3:0]				WSEQ_DATA_START19 [3:0]				WSEQ_DATA19 [7:0]								
R12328 (3028h)	WSEQ_Sequence_21	WSEQ_DATA_WIDTH20 [2:0]			WSEQ_ADDR20 [12:0]													E2315294h
		WSEQ_DELAY20 [3:0]				WSEQ_DATA_START20 [3:0]				WSEQ_DATA20 [7:0]								
R12330 (302Ah)	WSEQ_Sequence_22	WSEQ_DATA_WIDTH21 [2:0]			WSEQ_ADDR21 [12:0]													02310B00h
		WSEQ_DELAY21 [3:0]				WSEQ_DATA_START21 [3:0]				WSEQ_DATA21 [7:0]								
R12332 (302Ch)	WSEQ_Sequence_23	WSEQ_DATA_WIDTH22 [2:0]			WSEQ_ADDR22 [12:0]													02310B00h
		WSEQ_DELAY22 [3:0]				WSEQ_DATA_START22 [3:0]				WSEQ_DATA22 [7:0]								
R12334 (302Eh)	WSEQ_Sequence_24	WSEQ_DATA_WIDTH23 [2:0]			WSEQ_ADDR23 [12:0]													E2251734h
		WSEQ_DELAY23 [3:0]				WSEQ_DATA_START23 [3:0]				WSEQ_DATA23 [7:0]								
R12336 (3030h)	WSEQ_Sequence_25	WSEQ_DATA_WIDTH24 [2:0]			WSEQ_ADDR24 [12:0]													0225F501h
		WSEQ_DELAY24 [3:0]				WSEQ_DATA_START24 [3:0]				WSEQ_DATA24 [7:0]								
R12338 (3032h)	WSEQ_Sequence_26	WSEQ_DATA_WIDTH25 [2:0]			WSEQ_ADDR25 [12:0]													0000F000h
		WSEQ_DELAY25 [3:0]				WSEQ_DATA_START25 [3:0]				WSEQ_DATA25 [7:0]								
R12340 (3034h)	WSEQ_Sequence_27	WSEQ_DATA_WIDTH26 [2:0]			WSEQ_ADDR26 [12:0]													0000F000h
		WSEQ_DELAY26 [3:0]				WSEQ_DATA_START26 [3:0]				WSEQ_DATA26 [7:0]								
R12342 (3036h)	WSEQ_Sequence_28	WSEQ_DATA_WIDTH27 [2:0]			WSEQ_ADDR27 [12:0]													0000F000h
		WSEQ_DELAY27 [3:0]				WSEQ_DATA_START27 [3:0]				WSEQ_DATA27 [7:0]								
R12344 (3038h)	WSEQ_Sequence_29	WSEQ_DATA_WIDTH28 [2:0]			WSEQ_ADDR28 [12:0]													0000F000h
		WSEQ_DELAY28 [3:0]				WSEQ_DATA_START28 [3:0]				WSEQ_DATA28 [7:0]								
R12346 (303Ah)	WSEQ_Sequence_30	WSEQ_DATA_WIDTH29 [2:0]			WSEQ_ADDR29 [12:0]													0000F000h
		WSEQ_DELAY29 [3:0]				WSEQ_DATA_START29 [3:0]				WSEQ_DATA29 [7:0]								
R12348 (303Ch)	WSEQ_Sequence_31	WSEQ_DATA_WIDTH30 [2:0]			WSEQ_ADDR30 [12:0]													0000F000h
		WSEQ_DELAY30 [3:0]				WSEQ_DATA_START30 [3:0]				WSEQ_DATA30 [7:0]								
R12350 (303Eh)	WSEQ_Sequence_32	WSEQ_DATA_WIDTH31 [2:0]			WSEQ_ADDR31 [12:0]													02253A01h
		WSEQ_DELAY31 [3:0]				WSEQ_DATA_START31 [3:0]				WSEQ_DATA31 [7:0]								
R12352 (3040h)	WSEQ_Sequence_33	WSEQ_DATA_WIDTH32 [2:0]			WSEQ_ADDR32 [12:0]													C2251300h
		WSEQ_DELAY32 [3:0]				WSEQ_DATA_START32 [3:0]				WSEQ_DATA32 [7:0]								
R12354 (3042h)	WSEQ_Sequence_34	WSEQ_DATA_WIDTH33 [2:0]			WSEQ_ADDR33 [12:0]													02250B00h
		WSEQ_DELAY33 [3:0]				WSEQ_DATA_START33 [3:0]				WSEQ_DATA33 [7:0]								
R12356 (3044h)	WSEQ_Sequence_35	WSEQ_DATA_WIDTH34 [2:0]			WSEQ_ADDR34 [12:0]													0225FF01h
		WSEQ_DELAY34 [3:0]				WSEQ_DATA_START34 [3:0]				WSEQ_DATA34 [7:0]								
R12358 (3046h)	WSEQ_Sequence_36	WSEQ_DATA_WIDTH35 [2:0]			WSEQ_ADDR35 [12:0]													0000F000h
		WSEQ_DELAY35 [3:0]				WSEQ_DATA_START35 [3:0]				WSEQ_DATA35 [7:0]								
R12360 (3048h)	WSEQ_Sequence_37	WSEQ_DATA_WIDTH36 [2:0]			WSEQ_ADDR36 [12:0]													0000F000h
		WSEQ_DELAY36 [3:0]				WSEQ_DATA_START36 [3:0]				WSEQ_DATA36 [7:0]								
R12362 (304Ah)	WSEQ_Sequence_38	WSEQ_DATA_WIDTH37 [2:0]			WSEQ_ADDR37 [12:0]													0000F000h
		WSEQ_DELAY37 [3:0]				WSEQ_DATA_START37 [3:0]				WSEQ_DATA37 [7:0]								
R12364 (304Ch)	WSEQ_Sequence_39	WSEQ_DATA_WIDTH38 [2:0]			WSEQ_ADDR38 [12:0]													0000F000h
		WSEQ_DELAY38 [3:0]				WSEQ_DATA_START38 [3:0]				WSEQ_DATA38 [7:0]								
R12366 (304Eh)	WSEQ_Sequence_40	WSEQ_DATA_WIDTH39 [2:0]			WSEQ_ADDR39 [12:0]													0000F000h
		WSEQ_DELAY39 [3:0]				WSEQ_DATA_START39 [3:0]				WSEQ_DATA39 [7:0]								
R12368 (3050h)	WSEQ_Sequence_41	WSEQ_DATA_WIDTH40 [2:0]			WSEQ_ADDR40 [12:0]													0000F000h
		WSEQ_DELAY40 [3:0]				WSEQ_DATA_START40 [3:0]				WSEQ_DATA40 [7:0]								
R12370 (3052h)	WSEQ_Sequence_42	WSEQ_DATA_WIDTH41 [2:0]			WSEQ_ADDR41 [12:0]													0000F000h
		WSEQ_DELAY41 [3:0]				WSEQ_DATA_START41 [3:0]				WSEQ_DATA41 [7:0]								
R12372 (3054h)	WSEQ_Sequence_43	WSEQ_DATA_WIDTH42 [2:0]			WSEQ_ADDR42 [12:0]													0000F000h
		WSEQ_DELAY42 [3:0]				WSEQ_DATA_START42 [3:0]				WSEQ_DATA42 [7:0]								
R12374 (3056h)	WSEQ_Sequence_44	WSEQ_DATA_WIDTH43 [2:0]			WSEQ_ADDR43 [12:0]													0000F000h
		WSEQ_DELAY43 [3:0]				WSEQ_DATA_START43 [3:0]				WSEQ_DATA43 [7:0]								
R12376 (3058h)	WSEQ_Sequence_45	WSEQ_DATA_WIDTH44 [2:0]			WSEQ_ADDR44 [12:0]													82263719h
		WSEQ_DELAY44 [3:0]				WSEQ_DATA_START44 [3:0]				WSEQ_DATA44 [7:0]								
R12378 (305Ah)	WSEQ_Sequence_46	WSEQ_DATA_WIDTH45 [2:0]			WSEQ_ADDR45 [12:0]													C2300001h
		WSEQ_DELAY45 [3:0]				WSEQ_DATA_START45 [3:0]				WSEQ_DATA45 [7:0]								
R12380 (305Ch)	WSEQ_Sequence_47	WSEQ_DATA_WIDTH46 [2:0]			WSEQ_ADDR46 [12:0]													02261301h
		WSEQ_DELAY46 [3:0]				WSEQ_DATA_START46 [3:0]				WSEQ_DATA46 [7:0]								
R12382 (305Eh)	WSEQ_Sequence_48	WSEQ_DATA_WIDTH47 [2:0]			WSEQ_ADDR47 [12:0]													8226191Fh
		WSEQ_DELAY47 [3:0]				WSEQ_DATA_START47 [3:0]				WSEQ_DATA47 [7:0]								
R12384 (3060h)	WSEQ_Sequence_49	WSEQ_DATA_WIDTH48 [2:0]			WSEQ_ADDR48 [12:0]													82310B02h
		WSEQ_DELAY48 [3:0]				WSEQ_DATA_START48 [3:0]				WSEQ_DATA48 [7:0]								
R12386 (3062h)	WSEQ_Sequence_50	WSEQ_DATA_WIDTH49 [2:0]			WSEQ_ADDR49 [12:0]													E231023Bh
		WSEQ_DELAY49 [3:0]				WSEQ_DATA_START49 [3:0]				WSEQ_DATA49 [7:0]								
R12388 (3064h)	WSEQ_Sequence_51	WSEQ_DATA_WIDTH50 [2:0]			WSEQ_ADDR50 [12:0]													02313B01h
		WSEQ_DELAY50 [3:0]				WSEQ_DATA_START50 [3:0]				WSEQ_DATA50 [7:0]								
R12390 (3066h)	WSEQ_Sequence_52	WSEQ_DATA_WIDTH51 [2:0]			WSEQ_ADDR51 [12:0]													62300000h
		WSEQ_DELAY51 [3:0]				WSEQ_DATA_START51 [3:0]				WSEQ_DATA51 [7:0]								

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
R12392 (3068h)	WSEQ_Sequence_53	WSEQ_DATA_WIDTH52 [2:0]			WSEQ_ADDR52 [12:0]													E2314288h
		WSEQ_DELAY52 [3:0]				WSEQ_DATA_START52 [3:0]				WSEQ_DATA52 [7:0]								
R12394 (306Ah)	WSEQ_Sequence_54	WSEQ_DATA_WIDTH53 [2:0]			WSEQ_ADDR53 [12:0]													02310B00h
		WSEQ_DELAY53 [3:0]				WSEQ_DATA_START53 [3:0]				WSEQ_DATA53 [7:0]								
R12396 (306Ch)	WSEQ_Sequence_55	WSEQ_DATA_WIDTH54 [2:0]			WSEQ_ADDR54 [12:0]													02310B00h
		WSEQ_DELAY54 [3:0]				WSEQ_DATA_START54 [3:0]				WSEQ_DATA54 [7:0]								
R12398 (306Eh)	WSEQ_Sequence_56	WSEQ_DATA_WIDTH55 [2:0]			WSEQ_ADDR55 [12:0]													02260E01h
		WSEQ_DELAY55 [3:0]				WSEQ_DATA_START55 [3:0]				WSEQ_DATA55 [7:0]								
R12400 (3070h)	WSEQ_Sequence_57	WSEQ_DATA_WIDTH56 [2:0]			WSEQ_ADDR56 [12:0]													42310C03h
		WSEQ_DELAY56 [3:0]				WSEQ_DATA_START56 [3:0]				WSEQ_DATA56 [7:0]								
R12402 (3072h)	WSEQ_Sequence_58	WSEQ_DATA_WIDTH57 [2:0]			WSEQ_ADDR57 [12:0]													E2310227h
		WSEQ_DELAY57 [3:0]				WSEQ_DATA_START57 [3:0]				WSEQ_DATA57 [7:0]								
R12404 (3074h)	WSEQ_Sequence_59	WSEQ_DATA_WIDTH58 [2:0]			WSEQ_ADDR58 [12:0]													02313B01h
		WSEQ_DELAY58 [3:0]				WSEQ_DATA_START58 [3:0]				WSEQ_DATA58 [7:0]								
R12406 (3076h)	WSEQ_Sequence_60	WSEQ_DATA_WIDTH59 [2:0]			WSEQ_ADDR59 [12:0]													E2314266h
		WSEQ_DELAY59 [3:0]				WSEQ_DATA_START59 [3:0]				WSEQ_DATA59 [7:0]								
R12408 (3078h)	WSEQ_Sequence_61	WSEQ_DATA_WIDTH60 [2:0]			WSEQ_ADDR60 [12:0]													E2315294h
		WSEQ_DELAY60 [3:0]				WSEQ_DATA_START60 [3:0]				WSEQ_DATA60 [7:0]								
R12410 (307Ah)	WSEQ_Sequence_62	WSEQ_DATA_WIDTH61 [2:0]			WSEQ_ADDR61 [12:0]													02310B00h
		WSEQ_DELAY61 [3:0]				WSEQ_DATA_START61 [3:0]				WSEQ_DATA61 [7:0]								
R12412 (307Ch)	WSEQ_Sequence_63	WSEQ_DATA_WIDTH62 [2:0]			WSEQ_ADDR62 [12:0]													02310B00h
		WSEQ_DELAY62 [3:0]				WSEQ_DATA_START62 [3:0]				WSEQ_DATA62 [7:0]								
R12414 (307Eh)	WSEQ_Sequence_64	WSEQ_DATA_WIDTH63 [2:0]			WSEQ_ADDR63 [12:0]													E2261734h
		WSEQ_DELAY63 [3:0]				WSEQ_DATA_START63 [3:0]				WSEQ_DATA63 [7:0]								
R12416 (3080h)	WSEQ_Sequence_65	WSEQ_DATA_WIDTH64 [2:0]			WSEQ_ADDR64 [12:0]													0226F501h
		WSEQ_DELAY64 [3:0]				WSEQ_DATA_START64 [3:0]				WSEQ_DATA64 [7:0]								
R12418 (3082h)	WSEQ_Sequence_66	WSEQ_DATA_WIDTH65 [2:0]			WSEQ_ADDR65 [12:0]													0000F000h
		WSEQ_DELAY65 [3:0]				WSEQ_DATA_START65 [3:0]				WSEQ_DATA65 [7:0]								
R12420 (3084h)	WSEQ_Sequence_67	WSEQ_DATA_WIDTH66 [2:0]			WSEQ_ADDR66 [12:0]													0000F000h
		WSEQ_DELAY66 [3:0]				WSEQ_DATA_START66 [3:0]				WSEQ_DATA66 [7:0]								
R12422 (3086h)	WSEQ_Sequence_68	WSEQ_DATA_WIDTH67 [2:0]			WSEQ_ADDR67 [12:0]													0000F000h
		WSEQ_DELAY67 [3:0]				WSEQ_DATA_START67 [3:0]				WSEQ_DATA67 [7:0]								
R12424 (3088h)	WSEQ_Sequence_69	WSEQ_DATA_WIDTH68 [2:0]			WSEQ_ADDR68 [12:0]													0000F000h
		WSEQ_DELAY68 [3:0]				WSEQ_DATA_START68 [3:0]				WSEQ_DATA68 [7:0]								
R12426 (308Ah)	WSEQ_Sequence_70	WSEQ_DATA_WIDTH69 [2:0]			WSEQ_ADDR69 [12:0]													0000F000h
		WSEQ_DELAY69 [3:0]				WSEQ_DATA_START69 [3:0]				WSEQ_DATA69 [7:0]								
R12428 (308Ch)	WSEQ_Sequence_71	WSEQ_DATA_WIDTH70 [2:0]			WSEQ_ADDR70 [12:0]													0000F000h
		WSEQ_DELAY70 [3:0]				WSEQ_DATA_START70 [3:0]				WSEQ_DATA70 [7:0]								
R12430 (308Eh)	WSEQ_Sequence_72	WSEQ_DATA_WIDTH71 [2:0]			WSEQ_ADDR71 [12:0]													02263A01h
		WSEQ_DELAY71 [3:0]				WSEQ_DATA_START71 [3:0]				WSEQ_DATA71 [7:0]								
R12432 (3090h)	WSEQ_Sequence_73	WSEQ_DATA_WIDTH72 [2:0]			WSEQ_ADDR72 [12:0]													C2261300h
		WSEQ_DELAY72 [3:0]				WSEQ_DATA_START72 [3:0]				WSEQ_DATA72 [7:0]								
R12434 (3092h)	WSEQ_Sequence_74	WSEQ_DATA_WIDTH73 [2:0]			WSEQ_ADDR73 [12:0]													02260B00h
		WSEQ_DELAY73 [3:0]				WSEQ_DATA_START73 [3:0]				WSEQ_DATA73 [7:0]								
R12436 (3094h)	WSEQ_Sequence_75	WSEQ_DATA_WIDTH74 [2:0]			WSEQ_ADDR74 [12:0]													0226FF01h
		WSEQ_DELAY74 [3:0]				WSEQ_DATA_START74 [3:0]				WSEQ_DATA74 [7:0]								
R12438 (3096h)	WSEQ_Sequence_76	WSEQ_DATA_WIDTH75 [2:0]			WSEQ_ADDR75 [12:0]													0000F000h
		WSEQ_DELAY75 [3:0]				WSEQ_DATA_START75 [3:0]				WSEQ_DATA75 [7:0]								
R12440 (3098h)	WSEQ_Sequence_77	WSEQ_DATA_WIDTH76 [2:0]			WSEQ_ADDR76 [12:0]													0000F000h
		WSEQ_DELAY76 [3:0]				WSEQ_DATA_START76 [3:0]				WSEQ_DATA76 [7:0]								
R12442 (309Ah)	WSEQ_Sequence_78	WSEQ_DATA_WIDTH77 [2:0]			WSEQ_ADDR77 [12:0]													0000F000h
		WSEQ_DELAY77 [3:0]				WSEQ_DATA_START77 [3:0]				WSEQ_DATA77 [7:0]								
R12444 (309Ch)	WSEQ_Sequence_79	WSEQ_DATA_WIDTH78 [2:0]			WSEQ_ADDR78 [12:0]													0000F000h
		WSEQ_DELAY78 [3:0]				WSEQ_DATA_START78 [3:0]				WSEQ_DATA78 [7:0]								
R12446 (309Eh)	WSEQ_Sequence_80	WSEQ_DATA_WIDTH79 [2:0]			WSEQ_ADDR79 [12:0]													0000F000h
		WSEQ_DELAY79 [3:0]				WSEQ_DATA_START79 [3:0]				WSEQ_DATA79 [7:0]								
R12448 (30A0h)	WSEQ_Sequence_81	WSEQ_DATA_WIDTH80 [2:0]			WSEQ_ADDR80 [12:0]													0000F000h
		WSEQ_DELAY80 [3:0]				WSEQ_DATA_START80 [3:0]				WSEQ_DATA80 [7:0]								
R12450 (30A2h)	WSEQ_Sequence_82	WSEQ_DATA_WIDTH81 [2:0]			WSEQ_ADDR81 [12:0]													0000F000h
		WSEQ_DELAY81 [3:0]				WSEQ_DATA_START81 [3:0]				WSEQ_DATA81 [7:0]								
R12452 (30A4h)	WSEQ_Sequence_83	WSEQ_DATA_WIDTH82 [2:0]			WSEQ_ADDR82 [12:0]													0000F000h
		WSEQ_DELAY82 [3:0]				WSEQ_DATA_START82 [3:0]				WSEQ_DATA82 [7:0]								
R12454 (30A6h)	WSEQ_Sequence_84	WSEQ_DATA_WIDTH83 [2:0]			WSEQ_ADDR83 [12:0]													0000F000h
		WSEQ_DELAY83 [3:0]				WSEQ_DATA_START83 [3:0]				WSEQ_DATA83 [7:0]								
R12456 (30A8h)	WSEQ_Sequence_85	WSEQ_DATA_WIDTH84 [2:0]			WSEQ_ADDR84 [12:0]													82273719h
		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]													C2400001h
		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]													02271301h
		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]													8227191Fh
		WSEQ_DELAY87 [3:0]				WSEQ_DATA_START87 [3:0]				WSEQ_DATA87 [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
R12464 (30B0h)	WSEQ_Sequence_89	WSEQ_DATA_WIDTH88 [2:0]			WSEQ_ADDR88 [12:0]													82410B00h
		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]													E241023Bh
		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]													02413B01h
		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]													62400000h
		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]													E2414288h
		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]													02410B00h
		WSEQ_DELAY93 [3:0]				WSEQ_DATA_START93 [3:0]				WSEQ_DATA93 [7:0]								
R12476 (30BCh)	WSEQ_Sequence_95	WSEQ_DATA_WIDTH94 [2:0]			WSEQ_ADDR94 [12:0]													02410B00h
		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]													02270E01h
		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]													42410C02h
		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]													E2410227h
		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]													02413B01h
		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]													E2414266h
		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]													E2415294h
		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]													02410B00h
		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]													02410B00h
		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]													E2271734h
		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]													0227F501h
		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]													0000F000h
		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]													0000F000h
		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]													0000F000h
		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]													0000F000h
		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]													02273A01h
		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]													C2271300h
		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]													02270B00h
		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]													0227FF01h
		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]								

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
R12536	WSEQ_Sequence_125	WSEQ_DATA_WIDTH124 [2:0]			WSEQ_ADDR124 [12:0]													82283719h
(30F8h)		WSEQ_DELAY124 [3:0]				WSEQ_DATA_START124 [3:0]				WSEQ_DATA124 [7:0]								
R12538	WSEQ_Sequence_126	WSEQ_DATA_WIDTH125 [2:0]			WSEQ_ADDR125 [12:0]													C2400001h
(30FAh)		WSEQ_DELAY125 [3:0]				WSEQ_DATA_START125 [3:0]				WSEQ_DATA125 [7:0]								
R12540	WSEQ_Sequence_127	WSEQ_DATA_WIDTH126 [2:0]			WSEQ_ADDR126 [12:0]													02281301h
(30FCh)		WSEQ_DELAY126 [3:0]				WSEQ_DATA_START126 [3:0]				WSEQ_DATA126 [7:0]								
R12542	WSEQ_Sequence_128	WSEQ_DATA_WIDTH127 [2:0]			WSEQ_ADDR127 [12:0]													8228191Fh
(30FEh)		WSEQ_DELAY127 [3:0]				WSEQ_DATA_START127 [3:0]				WSEQ_DATA127 [7:0]								
R12544	WSEQ_Sequence_129	WSEQ_DATA_WIDTH128 [2:0]			WSEQ_ADDR128 [12:0]													82410B02h
(3100h)		WSEQ_DELAY128 [3:0]				WSEQ_DATA_START128 [3:0]				WSEQ_DATA128 [7:0]								
R12546	WSEQ_Sequence_130	WSEQ_DATA_WIDTH129 [2:0]			WSEQ_ADDR129 [12:0]													E241023Bh
(3102h)		WSEQ_DELAY129 [3:0]				WSEQ_DATA_START129 [3:0]				WSEQ_DATA129 [7:0]								
R12548	WSEQ_Sequence_131	WSEQ_DATA_WIDTH130 [2:0]			WSEQ_ADDR130 [12:0]													02413B01h
(3104h)		WSEQ_DELAY130 [3:0]				WSEQ_DATA_START130 [3:0]				WSEQ_DATA130 [7:0]								
R12550	WSEQ_Sequence_132	WSEQ_DATA_WIDTH131 [2:0]			WSEQ_ADDR131 [12:0]													62400000h
(3106h)		WSEQ_DELAY131 [3:0]				WSEQ_DATA_START131 [3:0]				WSEQ_DATA131 [7:0]								
R12552	WSEQ_Sequence_133	WSEQ_DATA_WIDTH132 [2:0]			WSEQ_ADDR132 [12:0]													E2414288h
(3108h)		WSEQ_DELAY132 [3:0]				WSEQ_DATA_START132 [3:0]				WSEQ_DATA132 [7:0]								
R12554	WSEQ_Sequence_134	WSEQ_DATA_WIDTH133 [2:0]			WSEQ_ADDR133 [12:0]													02410B00h
(310Ah)		WSEQ_DELAY133 [3:0]				WSEQ_DATA_START133 [3:0]				WSEQ_DATA133 [7:0]								
R12556	WSEQ_Sequence_135	WSEQ_DATA_WIDTH134 [2:0]			WSEQ_ADDR134 [12:0]													02410B00h
(310Ch)		WSEQ_DELAY134 [3:0]				WSEQ_DATA_START134 [3:0]				WSEQ_DATA134 [7:0]								
R12558	WSEQ_Sequence_136	WSEQ_DATA_WIDTH135 [2:0]			WSEQ_ADDR135 [12:0]													02280E01h
(310Eh)		WSEQ_DELAY135 [3:0]				WSEQ_DATA_START135 [3:0]				WSEQ_DATA135 [7:0]								
R12560	WSEQ_Sequence_137	WSEQ_DATA_WIDTH136 [2:0]			WSEQ_ADDR136 [12:0]													42410C03h
(3110h)		WSEQ_DELAY136 [3:0]				WSEQ_DATA_START136 [3:0]				WSEQ_DATA136 [7:0]								
R12562	WSEQ_Sequence_138	WSEQ_DATA_WIDTH137 [2:0]			WSEQ_ADDR137 [12:0]													E2410227h
(3112h)		WSEQ_DELAY137 [3:0]				WSEQ_DATA_START137 [3:0]				WSEQ_DATA137 [7:0]								
R12564	WSEQ_Sequence_139	WSEQ_DATA_WIDTH138 [2:0]			WSEQ_ADDR138 [12:0]													02413B01h
(3114h)		WSEQ_DELAY138 [3:0]				WSEQ_DATA_START138 [3:0]				WSEQ_DATA138 [7:0]								
R12566	WSEQ_Sequence_140	WSEQ_DATA_WIDTH139 [2:0]			WSEQ_ADDR139 [12:0]													E2414266h
(3116h)		WSEQ_DELAY139 [3:0]				WSEQ_DATA_START139 [3:0]				WSEQ_DATA139 [7:0]								
R12568	WSEQ_Sequence_141	WSEQ_DATA_WIDTH140 [2:0]			WSEQ_ADDR140 [12:0]													E2415294h
(3118h)		WSEQ_DELAY140 [3:0]				WSEQ_DATA_START140 [3:0]				WSEQ_DATA140 [7:0]								
R12570	WSEQ_Sequence_142	WSEQ_DATA_WIDTH141 [2:0]			WSEQ_ADDR141 [12:0]													02410B00h
(311Ah)		WSEQ_DELAY141 [3:0]				WSEQ_DATA_START141 [3:0]				WSEQ_DATA141 [7:0]								
R12572	WSEQ_Sequence_143	WSEQ_DATA_WIDTH142 [2:0]			WSEQ_ADDR142 [12:0]													02410B00h
(311Ch)		WSEQ_DELAY142 [3:0]				WSEQ_DATA_START142 [3:0]				WSEQ_DATA142 [7:0]								
R12574	WSEQ_Sequence_144	WSEQ_DATA_WIDTH143 [2:0]			WSEQ_ADDR143 [12:0]													E2281734h
(311Eh)		WSEQ_DELAY143 [3:0]				WSEQ_DATA_START143 [3:0]				WSEQ_DATA143 [7:0]								
R12576	WSEQ_Sequence_145	WSEQ_DATA_WIDTH144 [2:0]			WSEQ_ADDR144 [12:0]													0228F501h
(3120h)		WSEQ_DELAY144 [3:0]				WSEQ_DATA_START144 [3:0]				WSEQ_DATA144 [7:0]								
R12578	WSEQ_Sequence_146	WSEQ_DATA_WIDTH145 [2:0]			WSEQ_ADDR145 [12:0]													0000F000h
(3122h)		WSEQ_DELAY145 [3:0]				WSEQ_DATA_START145 [3:0]				WSEQ_DATA145 [7:0]								
R12580	WSEQ_Sequence_147	WSEQ_DATA_WIDTH146 [2:0]			WSEQ_ADDR146 [12:0]													0000F000h
(3124h)		WSEQ_DELAY146 [3:0]				WSEQ_DATA_START146 [3:0]				WSEQ_DATA146 [7:0]								
R12582	WSEQ_Sequence_148	WSEQ_DATA_WIDTH147 [2:0]			WSEQ_ADDR147 [12:0]													0000F000h
(3126h)		WSEQ_DELAY147 [3:0]				WSEQ_DATA_START147 [3:0]				WSEQ_DATA147 [7:0]								
R12584	WSEQ_Sequence_149	WSEQ_DATA_WIDTH148 [2:0]			WSEQ_ADDR148 [12:0]													0000F000h
(3128h)		WSEQ_DELAY148 [3:0]				WSEQ_DATA_START148 [3:0]				WSEQ_DATA148 [7:0]								
R12586	WSEQ_Sequence_150	WSEQ_DATA_WIDTH149 [2:0]			WSEQ_ADDR149 [12:0]													0000F000h
(312Ah)		WSEQ_DELAY149 [3:0]				WSEQ_DATA_START149 [3:0]				WSEQ_DATA149 [7:0]								
R12588	WSEQ_Sequence_151	WSEQ_DATA_WIDTH150 [2:0]			WSEQ_ADDR150 [12:0]													0000F000h
(312Ch)		WSEQ_DELAY150 [3:0]				WSEQ_DATA_START150 [3:0]				WSEQ_DATA150 [7:0]								
R12590	WSEQ_Sequence_152	WSEQ_DATA_WIDTH151 [2:0]			WSEQ_ADDR151 [12:0]													02283A01h
(312Eh)		WSEQ_DELAY151 [3:0]				WSEQ_DATA_START151 [3:0]				WSEQ_DATA151 [7:0]								
R12592	WSEQ_Sequence_153	WSEQ_DATA_WIDTH152 [2:0]			WSEQ_ADDR152 [12:0]													C2281300h
(3130h)		WSEQ_DELAY152 [3:0]				WSEQ_DATA_START152 [3:0]				WSEQ_DATA152 [7:0]								
R12594	WSEQ_Sequence_154	WSEQ_DATA_WIDTH153 [2:0]			WSEQ_ADDR153 [12:0]													02280B00h
(3132h)		WSEQ_DELAY153 [3:0]				WSEQ_DATA_START153 [3:0]				WSEQ_DATA153 [7:0]								
R12596	WSEQ_Sequence_155	WSEQ_DATA_WIDTH154 [2:0]			WSEQ_ADDR154 [12:0]													0228FF01h
(3134h)		WSEQ_DELAY154 [3:0]				WSEQ_DATA_START154 [3:0]				WSEQ_DATA154 [7:0]								
R12598	WSEQ_Sequence_156	WSEQ_DATA_WIDTH155 [2:0]			WSEQ_ADDR155 [12:0]													0000F000h
(3136h)		WSEQ_DELAY155 [3:0]				WSEQ_DATA_START155 [3:0]				WSEQ_DATA155 [7:0]								
R12600	WSEQ_Sequence_157	WSEQ_DATA_WIDTH156 [2:0]			WSEQ_ADDR156 [12:0]													0000F000h
(3138h)		WSEQ_DELAY156 [3:0]				WSEQ_DATA_START156 [3:0]				WSEQ_DATA156 [7:0]								
R12602	WSEQ_Sequence_158	WSEQ_DATA_WIDTH157 [2:0]			WSEQ_ADDR157 [12:0]													0000F000h
(313Ah)		WSEQ_DELAY157 [3:0]				WSEQ_DATA_START157 [3:0]				WSEQ_DATA157 [7:0]								
R12604	WSEQ_Sequence_159	WSEQ_DATA_WIDTH158 [2:0]			WSEQ_ADDR158 [12:0]													0000F000h
(313Ch)		WSEQ_DELAY158 [3:0]				WSEQ_DATA_START158 [3:0]				WSEQ_DATA158 [7:0]								
R12606	WSEQ_Sequence_160	WSEQ_DATA_WIDTH159 [2:0]			WSEQ_ADDR159 [12:0]													0000F000h
(313Eh)		WSEQ_DELAY159 [3:0]				WSEQ_DATA_START159 [3:0]				WSEQ_DATA159 [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
R12608 (3140h)	WSEQ_Sequence_161	WSEQ_DATA_WIDTH160 [2:0]			WSEQ_ADDR160 [12:0]													0000F000h
		WSEQ_DELAY160 [3:0]				WSEQ_DATA_START160 [3:0]				WSEQ_DATA160 [7:0]								
R12610 (3142h)	WSEQ_Sequence_162	WSEQ_DATA_WIDTH161 [2:0]			WSEQ_ADDR161 [12:0]													0000F000h
		WSEQ_DELAY161 [3:0]				WSEQ_DATA_START161 [3:0]				WSEQ_DATA161 [7:0]								
R12612 (3144h)	WSEQ_Sequence_163	WSEQ_DATA_WIDTH162 [2:0]			WSEQ_ADDR162 [12:0]													0000F000h
		WSEQ_DELAY162 [3:0]				WSEQ_DATA_START162 [3:0]				WSEQ_DATA162 [7:0]								
R12614 (3146h)	WSEQ_Sequence_164	WSEQ_DATA_WIDTH163 [2:0]			WSEQ_ADDR163 [12:0]													0000F000h
		WSEQ_DELAY163 [3:0]				WSEQ_DATA_START163 [3:0]				WSEQ_DATA163 [7:0]								
R12616 (3148h)	WSEQ_Sequence_165	WSEQ_DATA_WIDTH164 [2:0]			WSEQ_ADDR164 [12:0]													82293719h
		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]													C2500001h
		WSEQ_DELAY165 [3:0]				WSEQ_DATA_START165 [3:0]				WSEQ_DATA165 [7:0]								
R12620 (314Ch)	WSEQ_Sequence_167	WSEQ_DATA_WIDTH166 [2:0]			WSEQ_ADDR166 [12:0]													02291301h
		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]													8229191Fh
		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]													82510B00h
		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]													E251023Bh
		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]													02513B01h
		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]													62500000h
		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]													E2514288h
		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]													02510B00h
		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]													02510B00h
		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]													02290E01h
		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]													42510C02h
		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]													E2510227h
		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]													02513B01h
		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]													E2514266h
		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]													E2515294h
		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]													02510B00h
		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]													02510B00h
		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]													E2291734h
		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]													0229F501h
		WSEQ_DELAY184 [3:0]				WSEQ_DATA_START184 [3:0]				WSEQ_DATA184 [7:0]								
R12658 (3172h)	WSEQ_Sequence_186	WSEQ_DATA_WIDTH185 [2:0]			WSEQ_ADDR185 [12:0]													0000F000h
		WSEQ_DELAY185 [3:0]				WSEQ_DATA_START185 [3:0]				WSEQ_DATA185 [7:0]								
R12660 (3174h)	WSEQ_Sequence_187	WSEQ_DATA_WIDTH186 [2:0]			WSEQ_ADDR186 [12:0]													0000F000h
		WSEQ_DELAY186 [3:0]				WSEQ_DATA_START186 [3:0]				WSEQ_DATA186 [7:0]								
R12662 (3176h)	WSEQ_Sequence_188	WSEQ_DATA_WIDTH187 [2:0]			WSEQ_ADDR187 [12:0]													0000F000h
		WSEQ_DELAY187 [3:0]				WSEQ_DATA_START187 [3:0]				WSEQ_DATA187 [7:0]								
R12664 (3178h)	WSEQ_Sequence_189	WSEQ_DATA_WIDTH188 [2:0]			WSEQ_ADDR188 [12:0]													0000F000h
		WSEQ_DELAY188 [3:0]				WSEQ_DATA_START188 [3:0]				WSEQ_DATA188 [7:0]								
R12666 (317Ah)	WSEQ_Sequence_190	WSEQ_DATA_WIDTH189 [2:0]			WSEQ_ADDR189 [12:0]													0000F000h
		WSEQ_DELAY189 [3:0]				WSEQ_DATA_START189 [3:0]				WSEQ_DATA189 [7:0]								
R12668 (317Ch)	WSEQ_Sequence_191	WSEQ_DATA_WIDTH190 [2:0]			WSEQ_ADDR190 [12:0]													0000F000h
		WSEQ_DELAY190 [3:0]				WSEQ_DATA_START190 [3:0]				WSEQ_DATA190 [7:0]								
R12670 (317Eh)	WSEQ_Sequence_192	WSEQ_DATA_WIDTH191 [2:0]			WSEQ_ADDR191 [12:0]													02293A01h
		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]													C2291300h
		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]													02290B00h
		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]													0229FF01h
		WSEQ_DELAY194 [3:0]				WSEQ_DATA_START194 [3:0]				WSEQ_DATA194 [7:0]								
R12678 (3186h)	WSEQ_Sequence_196	WSEQ_DATA_WIDTH195 [2:0]			WSEQ_ADDR195 [12:0]													0000F000h
		WSEQ_DELAY195 [3:0]				WSEQ_DATA_START195 [3:0]				WSEQ_DATA195 [7:0]								

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
R12680 (3188h)	WSEQ_Sequence_197	WSEQ_DATA_WIDTH196 [2:0]			WSEQ_ADDR196 [12:0]													0000F000h
		WSEQ_DELAY196 [3:0]				WSEQ_DATA_START196 [3:0]				WSEQ_DATA196 [7:0]								
R12682 (318Ah)	WSEQ_Sequence_198	WSEQ_DATA_WIDTH197 [2:0]			WSEQ_ADDR197 [12:0]													0000F000h
		WSEQ_DELAY197 [3:0]				WSEQ_DATA_START197 [3:0]				WSEQ_DATA197 [7:0]								
R12684 (318Ch)	WSEQ_Sequence_199	WSEQ_DATA_WIDTH198 [2:0]			WSEQ_ADDR198 [12:0]													0000F000h
		WSEQ_DELAY198 [3:0]				WSEQ_DATA_START198 [3:0]				WSEQ_DATA198 [7:0]								
R12686 (318Eh)	WSEQ_Sequence_200	WSEQ_DATA_WIDTH199 [2:0]			WSEQ_ADDR199 [12:0]													0000F000h
		WSEQ_DELAY199 [3:0]				WSEQ_DATA_START199 [3:0]				WSEQ_DATA199 [7:0]								
R12688 (3190h)	WSEQ_Sequence_201	WSEQ_DATA_WIDTH200 [2:0]			WSEQ_ADDR200 [12:0]													0000F000h
		WSEQ_DELAY200 [3:0]				WSEQ_DATA_START200 [3:0]				WSEQ_DATA200 [7:0]								
R12690 (3192h)	WSEQ_Sequence_202	WSEQ_DATA_WIDTH201 [2:0]			WSEQ_ADDR201 [12:0]													0000F000h
		WSEQ_DELAY201 [3:0]				WSEQ_DATA_START201 [3:0]				WSEQ_DATA201 [7:0]								
R12692 (3194h)	WSEQ_Sequence_203	WSEQ_DATA_WIDTH202 [2:0]			WSEQ_ADDR202 [12:0]													0000F000h
		WSEQ_DELAY202 [3:0]				WSEQ_DATA_START202 [3:0]				WSEQ_DATA202 [7:0]								
R12694 (3196h)	WSEQ_Sequence_204	WSEQ_DATA_WIDTH203 [2:0]			WSEQ_ADDR203 [12:0]													0000F000h
		WSEQ_DELAY203 [3:0]				WSEQ_DATA_START203 [3:0]				WSEQ_DATA203 [7:0]								
R12696 (3198h)	WSEQ_Sequence_205	WSEQ_DATA_WIDTH204 [2:0]			WSEQ_ADDR204 [12:0]													822A3719h
		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]													C2500001h
		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]													022A1301h
		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]													822A191Fh
		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]													82510B02h
		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]													E251023Bh
		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]													02513B01h
		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]													62500000h
		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]													E2514288h
		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]													02510B00h
		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]													02510B00h
		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]													022A0E01h
		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]													42510C03h
		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]													E2510227h
		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]													02513B01h
		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]													E2514266h
		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]													E2515294h
		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]													02510B00h
		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]													02510B00h
		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]													E22A1734h
		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]													022AF501h
		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]													0000F000h
		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]													0000F000h
		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]													0000F000h
		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]													0000F000h
		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]													0000F000h
		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]													0000F000h
		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]													022A3A01h
		WSEQ_DELAY231 [3:0]				WSEQ_DATA_START231 [3:0]				WSEQ_DATA231 [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
R12752 (31D0h)	WSEQ_Sequence_233	WSEQ_DATA_WIDTH232 [2:0]			WSEQ_ADDR232 [12:0]													C22A1300h
		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]													022A0B00h
		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]													022AFF01h
		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]													0000F000h
		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]													0000F000h
		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]													0000F000h
		WSEQ_DELAY237 [3:0]				WSEQ_DATA_START237 [3:0]				WSEQ_DATA237 [7:0]								
R12764 (31DCh)	WSEQ_Sequence_239	WSEQ_DATA_WIDTH238 [2:0]			WSEQ_ADDR238 [12:0]													0000F000h
		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]													0000F000h
		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]													0000F000h
		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]													0000F000h
		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]													0000F000h
		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]													0000F000h
		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]													0000F000h
		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]													0000F000h
		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]													0000F000h
		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]													0000F000h
		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]													0000F000h
		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]													0000F000h
		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]													0000F000h
		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]													0000F000h
		WSEQ_DELAY251 [3:0]				WSEQ_DATA_START251 [3:0]				WSEQ_DATA251 [7:0]								
R12792 (31F8h)	WSEQ_Sequence_253	WSEQ_DATA_WIDTH252 [2:0]			WSEQ_ADDR252 [12:0]													0000F000h
		WSEQ_DELAY252 [3:0]				WSEQ_DATA_START252 [3:0]				WSEQ_DATA252 [7:0]								
R12794 (31FAh)	WSEQ_Sequence_254	WSEQ_DATA_WIDTH253 [2:0]			WSEQ_ADDR253 [12:0]													0000F000h
		WSEQ_DELAY253 [3:0]				WSEQ_DATA_START253 [3:0]				WSEQ_DATA253 [7:0]								
R12796 (31FCh)	WSEQ_Sequence_255	WSEQ_DATA_WIDTH254 [2:0]			WSEQ_ADDR254 [12:0]													0000F000h
		WSEQ_DELAY254 [3:0]				WSEQ_DATA_START254 [3:0]				WSEQ_DATA254 [7:0]								
R12798 (31FEh)	WSEQ_Sequence_256	WSEQ_DATA_WIDTH255 [2:0]			WSEQ_ADDR255 [12:0]													0000F000h
		WSEQ_DELAY255 [3:0]				WSEQ_DATA_START255 [3:0]				WSEQ_DATA255 [7:0]								
R12800 (3200h)	WSEQ_Sequence_257	WSEQ_DATA_WIDTH256 [2:0]			WSEQ_ADDR256 [12:0]													0000F000h
		WSEQ_DELAY256 [3:0]				WSEQ_DATA_START256 [3:0]				WSEQ_DATA256 [7:0]								
R12802 (3202h)	WSEQ_Sequence_258	WSEQ_DATA_WIDTH257 [2:0]			WSEQ_ADDR257 [12:0]													0000F000h
		WSEQ_DELAY257 [3:0]				WSEQ_DATA_START257 [3:0]				WSEQ_DATA257 [7:0]								
R12804 (3204h)	WSEQ_Sequence_259	WSEQ_DATA_WIDTH258 [2:0]			WSEQ_ADDR258 [12:0]													0000F000h
		WSEQ_DELAY258 [3:0]				WSEQ_DATA_START258 [3:0]				WSEQ_DATA258 [7:0]								
R12806 (3206h)	WSEQ_Sequence_260	WSEQ_DATA_WIDTH259 [2:0]			WSEQ_ADDR259 [12:0]													0000F000h
		WSEQ_DELAY259 [3:0]				WSEQ_DATA_START259 [3:0]				WSEQ_DATA259 [7:0]								
R12808 (3208h)	WSEQ_Sequence_261	WSEQ_DATA_WIDTH260 [2:0]			WSEQ_ADDR260 [12:0]													0000F000h
		WSEQ_DELAY260 [3:0]				WSEQ_DATA_START260 [3:0]				WSEQ_DATA260 [7:0]								
R12810 (320Ah)	WSEQ_Sequence_262	WSEQ_DATA_WIDTH261 [2:0]			WSEQ_ADDR261 [12:0]													0000F000h
		WSEQ_DELAY261 [3:0]				WSEQ_DATA_START261 [3:0]				WSEQ_DATA261 [7:0]								
R12812 (320Ch)	WSEQ_Sequence_263	WSEQ_DATA_WIDTH262 [2:0]			WSEQ_ADDR262 [12:0]													0000F000h
		WSEQ_DELAY262 [3:0]				WSEQ_DATA_START262 [3:0]				WSEQ_DATA262 [7:0]								
R12814 (320Eh)	WSEQ_Sequence_264	WSEQ_DATA_WIDTH263 [2:0]			WSEQ_ADDR263 [12:0]													0000F000h
		WSEQ_DELAY263 [3:0]				WSEQ_DATA_START263 [3:0]				WSEQ_DATA263 [7:0]								
R12816 (3210h)	WSEQ_Sequence_265	WSEQ_DATA_WIDTH264 [2:0]			WSEQ_ADDR264 [12:0]													0000F000h
		WSEQ_DELAY264 [3:0]				WSEQ_DATA_START264 [3:0]				WSEQ_DATA264 [7:0]								
R12818 (3212h)	WSEQ_Sequence_266	WSEQ_DATA_WIDTH265 [2:0]			WSEQ_ADDR265 [12:0]													0000F000h
		WSEQ_DELAY265 [3:0]				WSEQ_DATA_START265 [3:0]				WSEQ_DATA265 [7:0]								
R12820 (3214h)	WSEQ_Sequence_267	WSEQ_DATA_WIDTH266 [2:0]			WSEQ_ADDR266 [12:0]													0000F000h
		WSEQ_DELAY266 [3:0]				WSEQ_DATA_START266 [3:0]				WSEQ_DATA266 [7:0]								
R12822 (3216h)	WSEQ_Sequence_268	WSEQ_DATA_WIDTH267 [2:0]			WSEQ_ADDR267 [12:0]													0000F000h
		WSEQ_DELAY267 [3:0]				WSEQ_DATA_START267 [3:0]				WSEQ_DATA267 [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
R12824 (3218h)	WSEQ_Sequence_269	WSEQ_DATA_WIDTH268 [2:0]			WSEQ_ADDR268 [12:0]													0000F000h
		WSEQ_DELAY268 [3:0]				WSEQ_DATA_START268 [3:0]				WSEQ_DATA268 [7:0]								
R12826 (321Ah)	WSEQ_Sequence_270	WSEQ_DATA_WIDTH269 [2:0]			WSEQ_ADDR269 [12:0]													0000F000h
		WSEQ_DELAY269 [3:0]				WSEQ_DATA_START269 [3:0]				WSEQ_DATA269 [7:0]								
R12828 (321Ch)	WSEQ_Sequence_271	WSEQ_DATA_WIDTH270 [2:0]			WSEQ_ADDR270 [12:0]													0000F000h
		WSEQ_DELAY270 [3:0]				WSEQ_DATA_START270 [3:0]				WSEQ_DATA270 [7:0]								
R12830 (321Eh)	WSEQ_Sequence_272	WSEQ_DATA_WIDTH271 [2:0]			WSEQ_ADDR271 [12:0]													0000F000h
		WSEQ_DELAY271 [3:0]				WSEQ_DATA_START271 [3:0]				WSEQ_DATA271 [7:0]								
R12832 (3220h)	WSEQ_Sequence_273	WSEQ_DATA_WIDTH272 [2:0]			WSEQ_ADDR272 [12:0]													0000F000h
		WSEQ_DELAY272 [3:0]				WSEQ_DATA_START272 [3:0]				WSEQ_DATA272 [7:0]								
R12834 (3222h)	WSEQ_Sequence_274	WSEQ_DATA_WIDTH273 [2:0]			WSEQ_ADDR273 [12:0]													0000F000h
		WSEQ_DELAY273 [3:0]				WSEQ_DATA_START273 [3:0]				WSEQ_DATA273 [7:0]								
R12836 (3224h)	WSEQ_Sequence_275	WSEQ_DATA_WIDTH274 [2:0]			WSEQ_ADDR274 [12:0]													0000F000h
		WSEQ_DELAY274 [3:0]				WSEQ_DATA_START274 [3:0]				WSEQ_DATA274 [7:0]								
R12838 (3226h)	WSEQ_Sequence_276	WSEQ_DATA_WIDTH275 [2:0]			WSEQ_ADDR275 [12:0]													0000F000h
		WSEQ_DELAY275 [3:0]				WSEQ_DATA_START275 [3:0]				WSEQ_DATA275 [7:0]								
R12840 (3228h)	WSEQ_Sequence_277	WSEQ_DATA_WIDTH276 [2:0]			WSEQ_ADDR276 [12:0]													0000F000h
		WSEQ_DELAY276 [3:0]				WSEQ_DATA_START276 [3:0]				WSEQ_DATA276 [7:0]								
R12842 (322Ah)	WSEQ_Sequence_278	WSEQ_DATA_WIDTH277 [2:0]			WSEQ_ADDR277 [12:0]													0000F000h
		WSEQ_DELAY277 [3:0]				WSEQ_DATA_START277 [3:0]				WSEQ_DATA277 [7:0]								
R12844 (322Ch)	WSEQ_Sequence_279	WSEQ_DATA_WIDTH278 [2:0]			WSEQ_ADDR278 [12:0]													0000F000h
		WSEQ_DELAY278 [3:0]				WSEQ_DATA_START278 [3:0]				WSEQ_DATA278 [7:0]								
R12846 (322Eh)	WSEQ_Sequence_280	WSEQ_DATA_WIDTH279 [2:0]			WSEQ_ADDR279 [12:0]													0000F000h
		WSEQ_DELAY279 [3:0]				WSEQ_DATA_START279 [3:0]				WSEQ_DATA279 [7:0]								
R12848 (3230h)	WSEQ_Sequence_281	WSEQ_DATA_WIDTH280 [2:0]			WSEQ_ADDR280 [12:0]													0000F000h
		WSEQ_DELAY280 [3:0]				WSEQ_DATA_START280 [3:0]				WSEQ_DATA280 [7:0]								
R12850 (3232h)	WSEQ_Sequence_282	WSEQ_DATA_WIDTH281 [2:0]			WSEQ_ADDR281 [12:0]													0000F000h
		WSEQ_DELAY281 [3:0]				WSEQ_DATA_START281 [3:0]				WSEQ_DATA281 [7:0]								
R12852 (3234h)	WSEQ_Sequence_283	WSEQ_DATA_WIDTH282 [2:0]			WSEQ_ADDR282 [12:0]													0000F000h
		WSEQ_DELAY282 [3:0]				WSEQ_DATA_START282 [3:0]				WSEQ_DATA282 [7:0]								
R12854 (3236h)	WSEQ_Sequence_284	WSEQ_DATA_WIDTH283 [2:0]			WSEQ_ADDR283 [12:0]													0000F000h
		WSEQ_DELAY283 [3:0]				WSEQ_DATA_START283 [3:0]				WSEQ_DATA283 [7:0]								
R12856 (3238h)	WSEQ_Sequence_285	WSEQ_DATA_WIDTH284 [2:0]			WSEQ_ADDR284 [12:0]													0000F000h
		WSEQ_DELAY284 [3:0]				WSEQ_DATA_START284 [3:0]				WSEQ_DATA284 [7:0]								
R12858 (323Ah)	WSEQ_Sequence_286	WSEQ_DATA_WIDTH285 [2:0]			WSEQ_ADDR285 [12:0]													0000F000h
		WSEQ_DELAY285 [3:0]				WSEQ_DATA_START285 [3:0]				WSEQ_DATA285 [7:0]								
R12860 (323Ch)	WSEQ_Sequence_287	WSEQ_DATA_WIDTH286 [2:0]			WSEQ_ADDR286 [12:0]													0000F000h
		WSEQ_DELAY286 [3:0]				WSEQ_DATA_START286 [3:0]				WSEQ_DATA286 [7:0]								
R12862 (323Eh)	WSEQ_Sequence_288	WSEQ_DATA_WIDTH287 [2:0]			WSEQ_ADDR287 [12:0]													0000F000h
		WSEQ_DELAY287 [3:0]				WSEQ_DATA_START287 [3:0]				WSEQ_DATA287 [7:0]								
R12864 (3240h)	WSEQ_Sequence_289	WSEQ_DATA_WIDTH288 [2:0]			WSEQ_ADDR288 [12:0]													0000F000h
		WSEQ_DELAY288 [3:0]				WSEQ_DATA_START288 [3:0]				WSEQ_DATA288 [7:0]								
R12866 (3242h)	WSEQ_Sequence_290	WSEQ_DATA_WIDTH289 [2:0]			WSEQ_ADDR289 [12:0]													0000F000h
		WSEQ_DELAY289 [3:0]				WSEQ_DATA_START289 [3:0]				WSEQ_DATA289 [7:0]								
R12868 (3244h)	WSEQ_Sequence_291	WSEQ_DATA_WIDTH290 [2:0]			WSEQ_ADDR290 [12:0]													0000F000h
		WSEQ_DELAY290 [3:0]				WSEQ_DATA_START290 [3:0]				WSEQ_DATA290 [7:0]								
R12870 (3246h)	WSEQ_Sequence_292	WSEQ_DATA_WIDTH291 [2:0]			WSEQ_ADDR291 [12:0]													0000F000h
		WSEQ_DELAY291 [3:0]				WSEQ_DATA_START291 [3:0]				WSEQ_DATA291 [7:0]								
R12872 (3248h)	WSEQ_Sequence_293	WSEQ_DATA_WIDTH292 [2:0]			WSEQ_ADDR292 [12:0]													0000F000h
		WSEQ_DELAY292 [3:0]				WSEQ_DATA_START292 [3:0]				WSEQ_DATA292 [7:0]								
R12874 (324Ah)	WSEQ_Sequence_294	WSEQ_DATA_WIDTH293 [2:0]			WSEQ_ADDR293 [12:0]													0000F000h
		WSEQ_DELAY293 [3:0]				WSEQ_DATA_START293 [3:0]				WSEQ_DATA293 [7:0]								
R12876 (324Ch)	WSEQ_Sequence_295	WSEQ_DATA_WIDTH294 [2:0]			WSEQ_ADDR294 [12:0]													0000F000h
		WSEQ_DELAY294 [3:0]				WSEQ_DATA_START294 [3:0]				WSEQ_DATA294 [7:0]								
R12878 (324Eh)	WSEQ_Sequence_296	WSEQ_DATA_WIDTH295 [2:0]			WSEQ_ADDR295 [12:0]													0000F000h
		WSEQ_DELAY295 [3:0]				WSEQ_DATA_START295 [3:0]				WSEQ_DATA295 [7:0]								
R12880 (3250h)	WSEQ_Sequence_297	WSEQ_DATA_WIDTH296 [2:0]			WSEQ_ADDR296 [12:0]													0000F000h
		WSEQ_DELAY296 [3:0]				WSEQ_DATA_START296 [3:0]				WSEQ_DATA296 [7:0]								
R12882 (3252h)	WSEQ_Sequence_298	WSEQ_DATA_WIDTH297 [2:0]			WSEQ_ADDR297 [12:0]													0000F000h
		WSEQ_DELAY297 [3:0]				WSEQ_DATA_START297 [3:0]				WSEQ_DATA297 [7:0]								
R12884 (3254h)	WSEQ_Sequence_299	WSEQ_DATA_WIDTH298 [2:0]			WSEQ_ADDR298 [12:0]													0000F000h
		WSEQ_DELAY298 [3:0]				WSEQ_DATA_START298 [3:0]				WSEQ_DATA298 [7:0]								
R12886 (3256h)	WSEQ_Sequence_300	WSEQ_DATA_WIDTH299 [2:0]			WSEQ_ADDR299 [12:0]													0000F000h
		WSEQ_DELAY299 [3:0]				WSEQ_DATA_START299 [3:0]				WSEQ_DATA299 [7:0]								
R12888 (3258h)	WSEQ_Sequence_301	WSEQ_DATA_WIDTH300 [2:0]			WSEQ_ADDR300 [12:0]													0000F000h
		WSEQ_DELAY300 [3:0]				WSEQ_DATA_START300 [3:0]				WSEQ_DATA300 [7:0]								
R12890 (325Ah)	WSEQ_Sequence_302	WSEQ_DATA_WIDTH301 [2:0]			WSEQ_ADDR301 [12:0]													0000F000h
		WSEQ_DELAY301 [3:0]				WSEQ_DATA_START301 [3:0]				WSEQ_DATA301 [7:0]								
R12892 (325Ch)	WSEQ_Sequence_303	WSEQ_DATA_WIDTH302 [2:0]			WSEQ_ADDR302 [12:0]													0000F000h
		WSEQ_DELAY302 [3:0]				WSEQ_DATA_START302 [3:0]				WSEQ_DATA302 [7:0]								
R12894 (325Eh)	WSEQ_Sequence_304	WSEQ_DATA_WIDTH303 [2:0]			WSEQ_ADDR303 [12:0]													0000F000h
		WSEQ_DELAY303 [3:0]				WSEQ_DATA_START303 [3:0]				WSEQ_DATA303 [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
R12896 (3260h)	WSEQ_Sequence_305	WSEQ_DATA_WIDTH304 [2:0]			WSEQ_ADDR304 [12:0]													0000F000h
		WSEQ_DELAY304 [3:0]				WSEQ_DATA_START304 [3:0]				WSEQ_DATA304 [7:0]								
R12898 (3262h)	WSEQ_Sequence_306	WSEQ_DATA_WIDTH305 [2:0]			WSEQ_ADDR305 [12:0]													0000F000h
		WSEQ_DELAY305 [3:0]				WSEQ_DATA_START305 [3:0]				WSEQ_DATA305 [7:0]								
R12900 (3264h)	WSEQ_Sequence_307	WSEQ_DATA_WIDTH306 [2:0]			WSEQ_ADDR306 [12:0]													0000F000h
		WSEQ_DELAY306 [3:0]				WSEQ_DATA_START306 [3:0]				WSEQ_DATA306 [7:0]								
R12902 (3266h)	WSEQ_Sequence_308	WSEQ_DATA_WIDTH307 [2:0]			WSEQ_ADDR307 [12:0]													0000F000h
		WSEQ_DELAY307 [3:0]				WSEQ_DATA_START307 [3:0]				WSEQ_DATA307 [7:0]								
R12904 (3268h)	WSEQ_Sequence_309	WSEQ_DATA_WIDTH308 [2:0]			WSEQ_ADDR308 [12:0]													0000F000h
		WSEQ_DELAY308 [3:0]				WSEQ_DATA_START308 [3:0]				WSEQ_DATA308 [7:0]								
R12906 (326Ah)	WSEQ_Sequence_310	WSEQ_DATA_WIDTH309 [2:0]			WSEQ_ADDR309 [12:0]													0000F000h
		WSEQ_DELAY309 [3:0]				WSEQ_DATA_START309 [3:0]				WSEQ_DATA309 [7:0]								
R12908 (326Ch)	WSEQ_Sequence_311	WSEQ_DATA_WIDTH310 [2:0]			WSEQ_ADDR310 [12:0]													0000F000h
		WSEQ_DELAY310 [3:0]				WSEQ_DATA_START310 [3:0]				WSEQ_DATA310 [7:0]								
R12910 (326Eh)	WSEQ_Sequence_312	WSEQ_DATA_WIDTH311 [2:0]			WSEQ_ADDR311 [12:0]													0000F000h
		WSEQ_DELAY311 [3:0]				WSEQ_DATA_START311 [3:0]				WSEQ_DATA311 [7:0]								
R12912 (3270h)	WSEQ_Sequence_313	WSEQ_DATA_WIDTH312 [2:0]			WSEQ_ADDR312 [12:0]													0000F000h
		WSEQ_DELAY312 [3:0]				WSEQ_DATA_START312 [3:0]				WSEQ_DATA312 [7:0]								
R12914 (3272h)	WSEQ_Sequence_314	WSEQ_DATA_WIDTH313 [2:0]			WSEQ_ADDR313 [12:0]													0000F000h
		WSEQ_DELAY313 [3:0]				WSEQ_DATA_START313 [3:0]				WSEQ_DATA313 [7:0]								
R12916 (3274h)	WSEQ_Sequence_315	WSEQ_DATA_WIDTH314 [2:0]			WSEQ_ADDR314 [12:0]													0000F000h
		WSEQ_DELAY314 [3:0]				WSEQ_DATA_START314 [3:0]				WSEQ_DATA314 [7:0]								
R12918 (3276h)	WSEQ_Sequence_316	WSEQ_DATA_WIDTH315 [2:0]			WSEQ_ADDR315 [12:0]													0000F000h
		WSEQ_DELAY315 [3:0]				WSEQ_DATA_START315 [3:0]				WSEQ_DATA315 [7:0]								
R12920 (3278h)	WSEQ_Sequence_317	WSEQ_DATA_WIDTH316 [2:0]			WSEQ_ADDR316 [12:0]													0000F000h
		WSEQ_DELAY316 [3:0]				WSEQ_DATA_START316 [3:0]				WSEQ_DATA316 [7:0]								
R12922 (327Ah)	WSEQ_Sequence_318	WSEQ_DATA_WIDTH317 [2:0]			WSEQ_ADDR317 [12:0]													0000F000h
		WSEQ_DELAY317 [3:0]				WSEQ_DATA_START317 [3:0]				WSEQ_DATA317 [7:0]								
R12924 (327Ch)	WSEQ_Sequence_319	WSEQ_DATA_WIDTH318 [2:0]			WSEQ_ADDR318 [12:0]													0000F000h
		WSEQ_DELAY318 [3:0]				WSEQ_DATA_START318 [3:0]				WSEQ_DATA318 [7:0]								
R12926 (327Eh)	WSEQ_Sequence_320	WSEQ_DATA_WIDTH319 [2:0]			WSEQ_ADDR319 [12:0]													0000F000h
		WSEQ_DELAY319 [3:0]				WSEQ_DATA_START319 [3:0]				WSEQ_DATA319 [7:0]								
R12928 (3280h)	WSEQ_Sequence_321	WSEQ_DATA_WIDTH320 [2:0]			WSEQ_ADDR320 [12:0]													0000F000h
		WSEQ_DELAY320 [3:0]				WSEQ_DATA_START320 [3:0]				WSEQ_DATA320 [7:0]								
R12930 (3282h)	WSEQ_Sequence_322	WSEQ_DATA_WIDTH321 [2:0]			WSEQ_ADDR321 [12:0]													0000F000h
		WSEQ_DELAY321 [3:0]				WSEQ_DATA_START321 [3:0]				WSEQ_DATA321 [7:0]								
R12932 (3284h)	WSEQ_Sequence_323	WSEQ_DATA_WIDTH322 [2:0]			WSEQ_ADDR322 [12:0]													0000F000h
		WSEQ_DELAY322 [3:0]				WSEQ_DATA_START322 [3:0]				WSEQ_DATA322 [7:0]								
R12934 (3286h)	WSEQ_Sequence_324	WSEQ_DATA_WIDTH323 [2:0]			WSEQ_ADDR323 [12:0]													0000F000h
		WSEQ_DELAY323 [3:0]				WSEQ_DATA_START323 [3:0]				WSEQ_DATA323 [7:0]								
R12936 (3288h)	WSEQ_Sequence_325	WSEQ_DATA_WIDTH324 [2:0]			WSEQ_ADDR324 [12:0]													0000F000h
		WSEQ_DELAY324 [3:0]				WSEQ_DATA_START324 [3:0]				WSEQ_DATA324 [7:0]								
R12938 (328Ah)	WSEQ_Sequence_326	WSEQ_DATA_WIDTH325 [2:0]			WSEQ_ADDR325 [12:0]													0000F000h
		WSEQ_DELAY325 [3:0]				WSEQ_DATA_START325 [3:0]				WSEQ_DATA325 [7:0]								
R12940 (328Ch)	WSEQ_Sequence_327	WSEQ_DATA_WIDTH326 [2:0]			WSEQ_ADDR326 [12:0]													0000F000h
		WSEQ_DELAY326 [3:0]				WSEQ_DATA_START326 [3:0]				WSEQ_DATA326 [7:0]								
R12942 (328Eh)	WSEQ_Sequence_328	WSEQ_DATA_WIDTH327 [2:0]			WSEQ_ADDR327 [12:0]													0000F000h
		WSEQ_DELAY327 [3:0]				WSEQ_DATA_START327 [3:0]				WSEQ_DATA327 [7:0]								
R12944 (3290h)	WSEQ_Sequence_329	WSEQ_DATA_WIDTH328 [2:0]			WSEQ_ADDR328 [12:0]													0000F000h
		WSEQ_DELAY328 [3:0]				WSEQ_DATA_START328 [3:0]				WSEQ_DATA328 [7:0]								
R12946 (3292h)	WSEQ_Sequence_330	WSEQ_DATA_WIDTH329 [2:0]			WSEQ_ADDR329 [12:0]													0000F000h
		WSEQ_DELAY329 [3:0]				WSEQ_DATA_START329 [3:0]				WSEQ_DATA329 [7:0]								
R12948 (3294h)	WSEQ_Sequence_331	WSEQ_DATA_WIDTH330 [2:0]			WSEQ_ADDR330 [12:0]													0000F000h
		WSEQ_DELAY330 [3:0]				WSEQ_DATA_START330 [3:0]				WSEQ_DATA330 [7:0]								
R12950 (3296h)	WSEQ_Sequence_332	WSEQ_DATA_WIDTH331 [2:0]			WSEQ_ADDR331 [12:0]													0000F000h
		WSEQ_DELAY331 [3:0]				WSEQ_DATA_START331 [3:0]				WSEQ_DATA331 [7:0]								
R12952 (3298h)	WSEQ_Sequence_333	WSEQ_DATA_WIDTH332 [2:0]			WSEQ_ADDR332 [12:0]													0000F000h
		WSEQ_DELAY332 [3:0]				WSEQ_DATA_START332 [3:0]				WSEQ_DATA332 [7:0]								
R12954 (329Ah)	WSEQ_Sequence_334	WSEQ_DATA_WIDTH333 [2:0]			WSEQ_ADDR333 [12:0]													0000F000h
		WSEQ_DELAY333 [3:0]				WSEQ_DATA_START333 [3:0]				WSEQ_DATA333 [7:0]								
R12956 (329Ch)	WSEQ_Sequence_335	WSEQ_DATA_WIDTH334 [2:0]			WSEQ_ADDR334 [12:0]													0000F000h
		WSEQ_DELAY334 [3:0]				WSEQ_DATA_START334 [3:0]				WSEQ_DATA334 [7:0]								
R12958 (329Eh)	WSEQ_Sequence_336	WSEQ_DATA_WIDTH335 [2:0]			WSEQ_ADDR335 [12:0]													0000F000h
		WSEQ_DELAY335 [3:0]				WSEQ_DATA_START335 [3:0]				WSEQ_DATA335 [7:0]								
R12960 (32A0h)	WSEQ_Sequence_337	WSEQ_DATA_WIDTH336 [2:0]			WSEQ_ADDR336 [12:0]													0000F000h
		WSEQ_DELAY336 [3:0]				WSEQ_DATA_START336 [3:0]				WSEQ_DATA336 [7:0]								
R12962 (32A2h)	WSEQ_Sequence_338	WSEQ_DATA_WIDTH337 [2:0]			WSEQ_ADDR337 [12:0]													0000F000h
		WSEQ_DELAY337 [3:0]				WSEQ_DATA_START337 [3:0]				WSEQ_DATA337 [7:0]								
R12964 (32A4h)	WSEQ_Sequence_339	WSEQ_DATA_WIDTH338 [2:0]			WSEQ_ADDR338 [12:0]													0000F000h
		WSEQ_DELAY338 [3:0]				WSEQ_DATA_START338 [3:0]				WSEQ_DATA338 [7:0]								
R12966 (32A6h)	WSEQ_Sequence_340	WSEQ_DATA_WIDTH339 [2:0]			WSEQ_ADDR339 [12:0]													0000F000h
		WSEQ_DELAY339 [3:0]				WSEQ_DATA_START339 [3:0]				WSEQ_DATA339 [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
R12968 (32A8h)	WSEQ_Sequence_341	WSEQ_DATA_WIDTH340 [2:0]			WSEQ_ADDR340 [12:0]													0000F000h
		WSEQ_DELAY340 [3:0]				WSEQ_DATA_START340 [3:0]				WSEQ_DATA340 [7:0]								
R12970 (32AAh)	WSEQ_Sequence_342	WSEQ_DATA_WIDTH341 [2:0]			WSEQ_ADDR341 [12:0]													0000F000h
		WSEQ_DELAY341 [3:0]				WSEQ_DATA_START341 [3:0]				WSEQ_DATA341 [7:0]								
R12972 (32ACh)	WSEQ_Sequence_343	WSEQ_DATA_WIDTH342 [2:0]			WSEQ_ADDR342 [12:0]													0000F000h
		WSEQ_DELAY342 [3:0]				WSEQ_DATA_START342 [3:0]				WSEQ_DATA342 [7:0]								
R12974 (32AEh)	WSEQ_Sequence_344	WSEQ_DATA_WIDTH343 [2:0]			WSEQ_ADDR343 [12:0]													0000F000h
		WSEQ_DELAY343 [3:0]				WSEQ_DATA_START343 [3:0]				WSEQ_DATA343 [7:0]								
R12976 (32B0h)	WSEQ_Sequence_345	WSEQ_DATA_WIDTH344 [2:0]			WSEQ_ADDR344 [12:0]													0000F000h
		WSEQ_DELAY344 [3:0]				WSEQ_DATA_START344 [3:0]				WSEQ_DATA344 [7:0]								
R12978 (32B2h)	WSEQ_Sequence_346	WSEQ_DATA_WIDTH345 [2:0]			WSEQ_ADDR345 [12:0]													0000F000h
		WSEQ_DELAY345 [3:0]				WSEQ_DATA_START345 [3:0]				WSEQ_DATA345 [7:0]								
R12980 (32B4h)	WSEQ_Sequence_347	WSEQ_DATA_WIDTH346 [2:0]			WSEQ_ADDR346 [12:0]													0000F000h
		WSEQ_DELAY346 [3:0]				WSEQ_DATA_START346 [3:0]				WSEQ_DATA346 [7:0]								
R12982 (32B6h)	WSEQ_Sequence_348	WSEQ_DATA_WIDTH347 [2:0]			WSEQ_ADDR347 [12:0]													0000F000h
		WSEQ_DELAY347 [3:0]				WSEQ_DATA_START347 [3:0]				WSEQ_DATA347 [7:0]								
R12984 (32B8h)	WSEQ_Sequence_349	WSEQ_DATA_WIDTH348 [2:0]			WSEQ_ADDR348 [12:0]													0000F000h
		WSEQ_DELAY348 [3:0]				WSEQ_DATA_START348 [3:0]				WSEQ_DATA348 [7:0]								
R12986 (32BAh)	WSEQ_Sequence_350	WSEQ_DATA_WIDTH349 [2:0]			WSEQ_ADDR349 [12:0]													0000F000h
		WSEQ_DELAY349 [3:0]				WSEQ_DATA_START349 [3:0]				WSEQ_DATA349 [7:0]								
R12988 (32BCh)	WSEQ_Sequence_351	WSEQ_DATA_WIDTH350 [2:0]			WSEQ_ADDR350 [12:0]													0000F000h
		WSEQ_DELAY350 [3:0]				WSEQ_DATA_START350 [3:0]				WSEQ_DATA350 [7:0]								
R12990 (32BEh)	WSEQ_Sequence_352	WSEQ_DATA_WIDTH351 [2:0]			WSEQ_ADDR351 [12:0]													0000F000h
		WSEQ_DELAY351 [3:0]				WSEQ_DATA_START351 [3:0]				WSEQ_DATA351 [7:0]								
R12992 (32C0h)	WSEQ_Sequence_353	WSEQ_DATA_WIDTH352 [2:0]			WSEQ_ADDR352 [12:0]													0000F000h
		WSEQ_DELAY352 [3:0]				WSEQ_DATA_START352 [3:0]				WSEQ_DATA352 [7:0]								
R12994 (32C2h)	WSEQ_Sequence_354	WSEQ_DATA_WIDTH353 [2:0]			WSEQ_ADDR353 [12:0]													0000F000h
		WSEQ_DELAY353 [3:0]				WSEQ_DATA_START353 [3:0]				WSEQ_DATA353 [7:0]								
R12996 (32C4h)	WSEQ_Sequence_355	WSEQ_DATA_WIDTH354 [2:0]			WSEQ_ADDR354 [12:0]													0000F000h
		WSEQ_DELAY354 [3:0]				WSEQ_DATA_START354 [3:0]				WSEQ_DATA354 [7:0]								
R12998 (32C6h)	WSEQ_Sequence_356	WSEQ_DATA_WIDTH355 [2:0]			WSEQ_ADDR355 [12:0]													0000F000h
		WSEQ_DELAY355 [3:0]				WSEQ_DATA_START355 [3:0]				WSEQ_DATA355 [7:0]								
R13000 (32C8h)	WSEQ_Sequence_357	WSEQ_DATA_WIDTH356 [2:0]			WSEQ_ADDR356 [12:0]													0000F000h
		WSEQ_DELAY356 [3:0]				WSEQ_DATA_START356 [3:0]				WSEQ_DATA356 [7:0]								
R13002 (32CAh)	WSEQ_Sequence_358	WSEQ_DATA_WIDTH357 [2:0]			WSEQ_ADDR357 [12:0]													0000F000h
		WSEQ_DELAY357 [3:0]				WSEQ_DATA_START357 [3:0]				WSEQ_DATA357 [7:0]								
R13004 (32CCh)	WSEQ_Sequence_359	WSEQ_DATA_WIDTH358 [2:0]			WSEQ_ADDR358 [12:0]													0000F000h
		WSEQ_DELAY358 [3:0]				WSEQ_DATA_START358 [3:0]				WSEQ_DATA358 [7:0]								
R13006 (32CEh)	WSEQ_Sequence_360	WSEQ_DATA_WIDTH359 [2:0]			WSEQ_ADDR359 [12:0]													0000F000h
		WSEQ_DELAY359 [3:0]				WSEQ_DATA_START359 [3:0]				WSEQ_DATA359 [7:0]								
R13008 (32D0h)	WSEQ_Sequence_361	WSEQ_DATA_WIDTH360 [2:0]			WSEQ_ADDR360 [12:0]													0000F000h
		WSEQ_DELAY360 [3:0]				WSEQ_DATA_START360 [3:0]				WSEQ_DATA360 [7:0]								
R13010 (32D2h)	WSEQ_Sequence_362	WSEQ_DATA_WIDTH361 [2:0]			WSEQ_ADDR361 [12:0]													0000F000h
		WSEQ_DELAY361 [3:0]				WSEQ_DATA_START361 [3:0]				WSEQ_DATA361 [7:0]								
R13012 (32D4h)	WSEQ_Sequence_363	WSEQ_DATA_WIDTH362 [2:0]			WSEQ_ADDR362 [12:0]													0000F000h
		WSEQ_DELAY362 [3:0]				WSEQ_DATA_START362 [3:0]				WSEQ_DATA362 [7:0]								
R13014 (32D6h)	WSEQ_Sequence_364	WSEQ_DATA_WIDTH363 [2:0]			WSEQ_ADDR363 [12:0]													0000F000h
		WSEQ_DELAY363 [3:0]				WSEQ_DATA_START363 [3:0]				WSEQ_DATA363 [7:0]								
R13016 (32D8h)	WSEQ_Sequence_365	WSEQ_DATA_WIDTH364 [2:0]			WSEQ_ADDR364 [12:0]													0000F000h
		WSEQ_DELAY364 [3:0]				WSEQ_DATA_START364 [3:0]				WSEQ_DATA364 [7:0]								
R13018 (32DAh)	WSEQ_Sequence_366	WSEQ_DATA_WIDTH365 [2:0]			WSEQ_ADDR365 [12:0]													0000F000h
		WSEQ_DELAY365 [3:0]				WSEQ_DATA_START365 [3:0]				WSEQ_DATA365 [7:0]								
R13020 (32DCh)	WSEQ_Sequence_367	WSEQ_DATA_WIDTH366 [2:0]			WSEQ_ADDR366 [12:0]													0000F000h
		WSEQ_DELAY366 [3:0]				WSEQ_DATA_START366 [3:0]				WSEQ_DATA366 [7:0]								
R13022 (32DEh)	WSEQ_Sequence_368	WSEQ_DATA_WIDTH367 [2:0]			WSEQ_ADDR367 [12:0]													0000F000h
		WSEQ_DELAY367 [3:0]				WSEQ_DATA_START367 [3:0]				WSEQ_DATA367 [7:0]								
R13024 (32E0h)	WSEQ_Sequence_369	WSEQ_DATA_WIDTH368 [2:0]			WSEQ_ADDR368 [12:0]													0000F000h
		WSEQ_DELAY368 [3:0]				WSEQ_DATA_START368 [3:0]				WSEQ_DATA368 [7:0]								
R13026 (32E2h)	WSEQ_Sequence_370	WSEQ_DATA_WIDTH369 [2:0]			WSEQ_ADDR369 [12:0]													0000F000h
		WSEQ_DELAY369 [3:0]				WSEQ_DATA_START369 [3:0]				WSEQ_DATA369 [7:0]								
R13028 (32E4h)	WSEQ_Sequence_371	WSEQ_DATA_WIDTH370 [2:0]			WSEQ_ADDR370 [12:0]													0000F000h
		WSEQ_DELAY370 [3:0]				WSEQ_DATA_START370 [3:0]				WSEQ_DATA370 [7:0]								
R13030 (32E6h)	WSEQ_Sequence_372	WSEQ_DATA_WIDTH371 [2:0]			WSEQ_ADDR371 [12:0]													0000F000h
		WSEQ_DELAY371 [3:0]				WSEQ_DATA_START371 [3:0]				WSEQ_DATA371 [7:0]								
R13032 (32E8h)	WSEQ_Sequence_373	WSEQ_DATA_WIDTH372 [2:0]			WSEQ_ADDR372 [12:0]													0000F000h
		WSEQ_DELAY372 [3:0]				WSEQ_DATA_START372 [3:0]				WSEQ_DATA372 [7:0]								
R13034 (32EAh)	WSEQ_Sequence_374	WSEQ_DATA_WIDTH373 [2:0]			WSEQ_ADDR373 [12:0]													0000F000h
		WSEQ_DELAY373 [3:0]				WSEQ_DATA_START373 [3:0]				WSEQ_DATA373 [7:0]								
R13036 (32ECh)	WSEQ_Sequence_375	WSEQ_DATA_WIDTH374 [2:0]			WSEQ_ADDR374 [12:0]													0000F000h
		WSEQ_DELAY374 [3:0]				WSEQ_DATA_START374 [3:0]				WSEQ_DATA374 [7:0]								
R13038 (32EEh)	WSEQ_Sequence_376	WSEQ_DATA_WIDTH375 [2:0]			WSEQ_ADDR375 [12:0]													0000F000h
		WSEQ_DELAY375 [3:0]				WSEQ_DATA_START375 [3:0]				WSEQ_DATA375 [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
R13040 (32F0h)	WSEQ_Sequence_377	WSEQ_DATA_WIDTH376 [2:0]			WSEQ_ADDR376 [12:0]													0000F000h
		WSEQ_DELAY376 [3:0]				WSEQ_DATA_START376 [3:0]				WSEQ_DATA376 [7:0]								
R13042 (32F2h)	WSEQ_Sequence_378	WSEQ_DATA_WIDTH377 [2:0]			WSEQ_ADDR377 [12:0]													0000F000h
		WSEQ_DELAY377 [3:0]				WSEQ_DATA_START377 [3:0]				WSEQ_DATA377 [7:0]								
R13044 (32F4h)	WSEQ_Sequence_379	WSEQ_DATA_WIDTH378 [2:0]			WSEQ_ADDR378 [12:0]													0000F000h
		WSEQ_DELAY378 [3:0]				WSEQ_DATA_START378 [3:0]				WSEQ_DATA378 [7:0]								
R13046 (32F6h)	WSEQ_Sequence_380	WSEQ_DATA_WIDTH379 [2:0]			WSEQ_ADDR379 [12:0]													0000F000h
		WSEQ_DELAY379 [3:0]				WSEQ_DATA_START379 [3:0]				WSEQ_DATA379 [7:0]								
R13048 (32F8h)	WSEQ_Sequence_381	WSEQ_DATA_WIDTH380 [2:0]			WSEQ_ADDR380 [12:0]													0000F000h
		WSEQ_DELAY380 [3:0]				WSEQ_DATA_START380 [3:0]				WSEQ_DATA380 [7:0]								
R13050 (32FAh)	WSEQ_Sequence_382	WSEQ_DATA_WIDTH381 [2:0]			WSEQ_ADDR381 [12:0]													0000F000h
		WSEQ_DELAY381 [3:0]				WSEQ_DATA_START381 [3:0]				WSEQ_DATA381 [7:0]								
R13052 (32FCh)	WSEQ_Sequence_383	WSEQ_DATA_WIDTH382 [2:0]			WSEQ_ADDR382 [12:0]													0000F000h
		WSEQ_DELAY382 [3:0]				WSEQ_DATA_START382 [3:0]				WSEQ_DATA382 [7:0]								
R13054 (32FEh)	WSEQ_Sequence_384	WSEQ_DATA_WIDTH383 [2:0]			WSEQ_ADDR383 [12:0]													0000F000h
		WSEQ_DELAY383 [3:0]				WSEQ_DATA_START383 [3:0]				WSEQ_DATA383 [7:0]								
R13056 (3300h)	WSEQ_Sequence_385	WSEQ_DATA_WIDTH384 [2:0]			WSEQ_ADDR384 [12:0]													FFFFFFFFh
		WSEQ_DELAY384 [3:0]				WSEQ_DATA_START384 [3:0]				WSEQ_DATA384 [7:0]								
R13058 (3302h)	WSEQ_Sequence_386	WSEQ_DATA_WIDTH385 [2:0]			WSEQ_ADDR385 [12:0]													FFFFFFFFh
		WSEQ_DELAY385 [3:0]				WSEQ_DATA_START385 [3:0]				WSEQ_DATA385 [7:0]								
R13060 (3304h)	WSEQ_Sequence_387	WSEQ_DATA_WIDTH386 [2:0]			WSEQ_ADDR386 [12:0]													FFFFFFFFh
		WSEQ_DELAY386 [3:0]				WSEQ_DATA_START386 [3:0]				WSEQ_DATA386 [7:0]								
R13062 (3306h)	WSEQ_Sequence_388	WSEQ_DATA_WIDTH387 [2:0]			WSEQ_ADDR387 [12:0]													FFFFFFFFh
		WSEQ_DELAY387 [3:0]				WSEQ_DATA_START387 [3:0]				WSEQ_DATA387 [7:0]								
R13064 (3308h)	WSEQ_Sequence_389	WSEQ_DATA_WIDTH388 [2:0]			WSEQ_ADDR388 [12:0]													FFFFFFFFh
		WSEQ_DELAY388 [3:0]				WSEQ_DATA_START388 [3:0]				WSEQ_DATA388 [7:0]								
R13066 (330Ah)	WSEQ_Sequence_390	WSEQ_DATA_WIDTH389 [2:0]			WSEQ_ADDR389 [12:0]													FFFFFFFFh
		WSEQ_DELAY389 [3:0]				WSEQ_DATA_START389 [3:0]				WSEQ_DATA389 [7:0]								
R13068 (330Ch)	WSEQ_Sequence_391	WSEQ_DATA_WIDTH390 [2:0]			WSEQ_ADDR390 [12:0]													FFFFFFFFh
		WSEQ_DELAY390 [3:0]				WSEQ_DATA_START390 [3:0]				WSEQ_DATA390 [7:0]								
R13070 (330Eh)	WSEQ_Sequence_392	WSEQ_DATA_WIDTH391 [2:0]			WSEQ_ADDR391 [12:0]													FFFFFFFFh
		WSEQ_DELAY391 [3:0]				WSEQ_DATA_START391 [3:0]				WSEQ_DATA391 [7:0]								
R13072 (3310h)	WSEQ_Sequence_393	WSEQ_DATA_WIDTH392 [2:0]			WSEQ_ADDR392 [12:0]													FFFFFFFFh
		WSEQ_DELAY392 [3:0]				WSEQ_DATA_START392 [3:0]				WSEQ_DATA392 [7:0]								
R13074 (3312h)	WSEQ_Sequence_394	WSEQ_DATA_WIDTH393 [2:0]			WSEQ_ADDR393 [12:0]													FFFFFFFFh
		WSEQ_DELAY393 [3:0]				WSEQ_DATA_START393 [3:0]				WSEQ_DATA393 [7:0]								
R13076 (3314h)	WSEQ_Sequence_395	WSEQ_DATA_WIDTH394 [2:0]			WSEQ_ADDR394 [12:0]													FFFFFFFFh
		WSEQ_DELAY394 [3:0]				WSEQ_DATA_START394 [3:0]				WSEQ_DATA394 [7:0]								
R13078 (3316h)	WSEQ_Sequence_396	WSEQ_DATA_WIDTH395 [2:0]			WSEQ_ADDR395 [12:0]													FFFFFFFFh
		WSEQ_DELAY395 [3:0]				WSEQ_DATA_START395 [3:0]				WSEQ_DATA395 [7:0]								
R13080 (3318h)	WSEQ_Sequence_397	WSEQ_DATA_WIDTH396 [2:0]			WSEQ_ADDR396 [12:0]													FFFFFFFFh
		WSEQ_DELAY396 [3:0]				WSEQ_DATA_START396 [3:0]				WSEQ_DATA396 [7:0]								
R13082 (331Ah)	WSEQ_Sequence_398	WSEQ_DATA_WIDTH397 [2:0]			WSEQ_ADDR397 [12:0]													FFFFFFFFh
		WSEQ_DELAY397 [3:0]				WSEQ_DATA_START397 [3:0]				WSEQ_DATA397 [7:0]								
R13084 (331Ch)	WSEQ_Sequence_399	WSEQ_DATA_WIDTH398 [2:0]			WSEQ_ADDR398 [12:0]													FFFFFFFFh
		WSEQ_DELAY398 [3:0]				WSEQ_DATA_START398 [3:0]				WSEQ_DATA398 [7:0]								
R13086 (331Eh)	WSEQ_Sequence_400	WSEQ_DATA_WIDTH399 [2:0]			WSEQ_ADDR399 [12:0]													FFFFFFFFh
		WSEQ_DELAY399 [3:0]				WSEQ_DATA_START399 [3:0]				WSEQ_DATA399 [7:0]								
R13088 (3320h)	WSEQ_Sequence_401	WSEQ_DATA_WIDTH400 [2:0]			WSEQ_ADDR400 [12:0]													FFFFFFFFh
		WSEQ_DELAY400 [3:0]				WSEQ_DATA_START400 [3:0]				WSEQ_DATA400 [7:0]								
R13090 (3322h)	WSEQ_Sequence_402	WSEQ_DATA_WIDTH401 [2:0]			WSEQ_ADDR401 [12:0]													FFFFFFFFh
		WSEQ_DELAY401 [3:0]				WSEQ_DATA_START401 [3:0]				WSEQ_DATA401 [7:0]								
R13092 (3324h)	WSEQ_Sequence_403	WSEQ_DATA_WIDTH402 [2:0]			WSEQ_ADDR402 [12:0]													FFFFFFFFh
		WSEQ_DELAY402 [3:0]				WSEQ_DATA_START402 [3:0]				WSEQ_DATA402 [7:0]								
R13094 (3326h)	WSEQ_Sequence_404	WSEQ_DATA_WIDTH403 [2:0]			WSEQ_ADDR403 [12:0]													FFFFFFFFh
		WSEQ_DELAY403 [3:0]				WSEQ_DATA_START403 [3:0]				WSEQ_DATA403 [7:0]								
R13096 (3328h)	WSEQ_Sequence_405	WSEQ_DATA_WIDTH404 [2:0]			WSEQ_ADDR404 [12:0]													FFFFFFFFh
		WSEQ_DELAY404 [3:0]				WSEQ_DATA_START404 [3:0]				WSEQ_DATA404 [7:0]								
R13098 (332Ah)	WSEQ_Sequence_406	WSEQ_DATA_WIDTH405 [2:0]			WSEQ_ADDR405 [12:0]													FFFFFFFFh
		WSEQ_DELAY405 [3:0]				WSEQ_DATA_START405 [3:0]				WSEQ_DATA405 [7:0]								
R13100 (332Ch)	WSEQ_Sequence_407	WSEQ_DATA_WIDTH406 [2:0]			WSEQ_ADDR406 [12:0]													FFFFFFFFh
		WSEQ_DELAY406 [3:0]				WSEQ_DATA_START406 [3:0]				WSEQ_DATA406 [7:0]								
R13102 (332Eh)	WSEQ_Sequence_408	WSEQ_DATA_WIDTH407 [2:0]			WSEQ_ADDR407 [12:0]													FFFFFFFFh
		WSEQ_DELAY407 [3:0]				WSEQ_DATA_START407 [3:0]				WSEQ_DATA407 [7:0]								
R13104 (3330h)	WSEQ_Sequence_409	WSEQ_DATA_WIDTH408 [2:0]			WSEQ_ADDR408 [12:0]													FFFFFFFFh
		WSEQ_DELAY408 [3:0]				WSEQ_DATA_START408 [3:0]				WSEQ_DATA408 [7:0]								
R13106 (3332h)	WSEQ_Sequence_410	WSEQ_DATA_WIDTH409 [2:0]			WSEQ_ADDR409 [12:0]													FFFFFFFFh
		WSEQ_DELAY409 [3:0]				WSEQ_DATA_START409 [3:0]				WSEQ_DATA409 [7:0]								
R13108 (3334h)	WSEQ_Sequence_411	WSEQ_DATA_WIDTH410 [2:0]			WSEQ_ADDR410 [12:0]													FFFFFFFFh
		WSEQ_DELAY410 [3:0]				WSEQ_DATA_START410 [3:0]				WSEQ_DATA410 [7:0]								
R13110 (3336h)	WSEQ_Sequence_412	WSEQ_DATA_WIDTH411 [2:0]			WSEQ_ADDR411 [12:0]													FFFFFFFFh
		WSEQ_DELAY411 [3:0]				WSEQ_DATA_START411 [3:0]				WSEQ_DATA411 [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
R13112 (3338h)	WSEQ_Sequence_413	WSEQ_DATA_WIDTH412 [2:0]			WSEQ_ADDR412 [12:0]													FFFFFFFFh
		WSEQ_DELAY412 [3:0]				WSEQ_DATA_START412 [3:0]				WSEQ_DATA412 [7:0]								
R13114 (333Ah)	WSEQ_Sequence_414	WSEQ_DATA_WIDTH413 [2:0]			WSEQ_ADDR413 [12:0]													FFFFFFFFh
		WSEQ_DELAY413 [3:0]				WSEQ_DATA_START413 [3:0]				WSEQ_DATA413 [7:0]								
R13116 (333Ch)	WSEQ_Sequence_415	WSEQ_DATA_WIDTH414 [2:0]			WSEQ_ADDR414 [12:0]													FFFFFFFFh
		WSEQ_DELAY414 [3:0]				WSEQ_DATA_START414 [3:0]				WSEQ_DATA414 [7:0]								
R13118 (333Eh)	WSEQ_Sequence_416	WSEQ_DATA_WIDTH415 [2:0]			WSEQ_ADDR415 [12:0]													FFFFFFFFh
		WSEQ_DELAY415 [3:0]				WSEQ_DATA_START415 [3:0]				WSEQ_DATA415 [7:0]								
R13120 (3340h)	WSEQ_Sequence_417	WSEQ_DATA_WIDTH416 [2:0]			WSEQ_ADDR416 [12:0]													FFFFFFFFh
		WSEQ_DELAY416 [3:0]				WSEQ_DATA_START416 [3:0]				WSEQ_DATA416 [7:0]								
R13122 (3342h)	WSEQ_Sequence_418	WSEQ_DATA_WIDTH417 [2:0]			WSEQ_ADDR417 [12:0]													FFFFFFFFh
		WSEQ_DELAY417 [3:0]				WSEQ_DATA_START417 [3:0]				WSEQ_DATA417 [7:0]								
R13124 (3344h)	WSEQ_Sequence_419	WSEQ_DATA_WIDTH418 [2:0]			WSEQ_ADDR418 [12:0]													FFFFFFFFh
		WSEQ_DELAY418 [3:0]				WSEQ_DATA_START418 [3:0]				WSEQ_DATA418 [7:0]								
R13126 (3346h)	WSEQ_Sequence_420	WSEQ_DATA_WIDTH419 [2:0]			WSEQ_ADDR419 [12:0]													FFFFFFFFh
		WSEQ_DELAY419 [3:0]				WSEQ_DATA_START419 [3:0]				WSEQ_DATA419 [7:0]								
R13128 (3348h)	WSEQ_Sequence_421	WSEQ_DATA_WIDTH420 [2:0]			WSEQ_ADDR420 [12:0]													FFFFFFFFh
		WSEQ_DELAY420 [3:0]				WSEQ_DATA_START420 [3:0]				WSEQ_DATA420 [7:0]								
R13130 (334Ah)	WSEQ_Sequence_422	WSEQ_DATA_WIDTH421 [2:0]			WSEQ_ADDR421 [12:0]													FFFFFFFFh
		WSEQ_DELAY421 [3:0]				WSEQ_DATA_START421 [3:0]				WSEQ_DATA421 [7:0]								
R13132 (334Ch)	WSEQ_Sequence_423	WSEQ_DATA_WIDTH422 [2:0]			WSEQ_ADDR422 [12:0]													FFFFFFFFh
		WSEQ_DELAY422 [3:0]				WSEQ_DATA_START422 [3:0]				WSEQ_DATA422 [7:0]								
R13134 (334Eh)	WSEQ_Sequence_424	WSEQ_DATA_WIDTH423 [2:0]			WSEQ_ADDR423 [12:0]													FFFFFFFFh
		WSEQ_DELAY423 [3:0]				WSEQ_DATA_START423 [3:0]				WSEQ_DATA423 [7:0]								
R13136 (3350h)	WSEQ_Sequence_425	WSEQ_DATA_WIDTH424 [2:0]			WSEQ_ADDR424 [12:0]													FFFFFFFFh
		WSEQ_DELAY424 [3:0]				WSEQ_DATA_START424 [3:0]				WSEQ_DATA424 [7:0]								
R13138 (3352h)	WSEQ_Sequence_426	WSEQ_DATA_WIDTH425 [2:0]			WSEQ_ADDR425 [12:0]													FFFFFFFFh
		WSEQ_DELAY425 [3:0]				WSEQ_DATA_START425 [3:0]				WSEQ_DATA425 [7:0]								
R13140 (3354h)	WSEQ_Sequence_427	WSEQ_DATA_WIDTH426 [2:0]			WSEQ_ADDR426 [12:0]													FFFFFFFFh
		WSEQ_DELAY426 [3:0]				WSEQ_DATA_START426 [3:0]				WSEQ_DATA426 [7:0]								
R13142 (3356h)	WSEQ_Sequence_428	WSEQ_DATA_WIDTH427 [2:0]			WSEQ_ADDR427 [12:0]													FFFFFFFFh
		WSEQ_DELAY427 [3:0]				WSEQ_DATA_START427 [3:0]				WSEQ_DATA427 [7:0]								
R13144 (3358h)	WSEQ_Sequence_429	WSEQ_DATA_WIDTH428 [2:0]			WSEQ_ADDR428 [12:0]													FFFFFFFFh
		WSEQ_DELAY428 [3:0]				WSEQ_DATA_START428 [3:0]				WSEQ_DATA428 [7:0]								
R13146 (335Ah)	WSEQ_Sequence_430	WSEQ_DATA_WIDTH429 [2:0]			WSEQ_ADDR429 [12:0]													FFFFFFFFh
		WSEQ_DELAY429 [3:0]				WSEQ_DATA_START429 [3:0]				WSEQ_DATA429 [7:0]								
R13148 (335Ch)	WSEQ_Sequence_431	WSEQ_DATA_WIDTH430 [2:0]			WSEQ_ADDR430 [12:0]													FFFFFFFFh
		WSEQ_DELAY430 [3:0]				WSEQ_DATA_START430 [3:0]				WSEQ_DATA430 [7:0]								
R13150 (335Eh)	WSEQ_Sequence_432	WSEQ_DATA_WIDTH431 [2:0]			WSEQ_ADDR431 [12:0]													FFFFFFFFh
		WSEQ_DELAY431 [3:0]				WSEQ_DATA_START431 [3:0]				WSEQ_DATA431 [7:0]								
R13152 (3360h)	WSEQ_Sequence_433	WSEQ_DATA_WIDTH432 [2:0]			WSEQ_ADDR432 [12:0]													FFFFFFFFh
		WSEQ_DELAY432 [3:0]				WSEQ_DATA_START432 [3:0]				WSEQ_DATA432 [7:0]								
R13154 (3362h)	WSEQ_Sequence_434	WSEQ_DATA_WIDTH433 [2:0]			WSEQ_ADDR433 [12:0]													FFFFFFFFh
		WSEQ_DELAY433 [3:0]				WSEQ_DATA_START433 [3:0]				WSEQ_DATA433 [7:0]								
R13156 (3364h)	WSEQ_Sequence_435	WSEQ_DATA_WIDTH434 [2:0]			WSEQ_ADDR434 [12:0]													FFFFFFFFh
		WSEQ_DELAY434 [3:0]				WSEQ_DATA_START434 [3:0]				WSEQ_DATA434 [7:0]								
R13158 (3366h)	WSEQ_Sequence_436	WSEQ_DATA_WIDTH435 [2:0]			WSEQ_ADDR435 [12:0]													FFFFFFFFh
		WSEQ_DELAY435 [3:0]				WSEQ_DATA_START435 [3:0]				WSEQ_DATA435 [7:0]								
R13160 (3368h)	WSEQ_Sequence_437	WSEQ_DATA_WIDTH436 [2:0]			WSEQ_ADDR436 [12:0]													FFFFFFFFh
		WSEQ_DELAY436 [3:0]				WSEQ_DATA_START436 [3:0]				WSEQ_DATA436 [7:0]								
R13162 (336Ah)	WSEQ_Sequence_438	WSEQ_DATA_WIDTH437 [2:0]			WSEQ_ADDR437 [12:0]													FFFFFFFFh
		WSEQ_DELAY437 [3:0]				WSEQ_DATA_START437 [3:0]				WSEQ_DATA437 [7:0]								
R13164 (336Ch)	WSEQ_Sequence_439	WSEQ_DATA_WIDTH438 [2:0]			WSEQ_ADDR438 [12:0]													FFFFFFFFh
		WSEQ_DELAY438 [3:0]				WSEQ_DATA_START438 [3:0]				WSEQ_DATA438 [7:0]								
R13166 (336Eh)	WSEQ_Sequence_440	WSEQ_DATA_WIDTH439 [2:0]			WSEQ_ADDR439 [12:0]													FFFFFFFFh
		WSEQ_DELAY439 [3:0]				WSEQ_DATA_START439 [3:0]				WSEQ_DATA439 [7:0]								
R13168 (3370h)	WSEQ_Sequence_441	WSEQ_DATA_WIDTH440 [2:0]			WSEQ_ADDR440 [12:0]													FFFFFFFFh
		WSEQ_DELAY440 [3:0]				WSEQ_DATA_START440 [3:0]				WSEQ_DATA440 [7:0]								
R13170 (3372h)	WSEQ_Sequence_442	WSEQ_DATA_WIDTH441 [2:0]			WSEQ_ADDR441 [12:0]													FFFFFFFFh
		WSEQ_DELAY441 [3:0]				WSEQ_DATA_START441 [3:0]				WSEQ_DATA441 [7:0]								
R13172 (3374h)	WSEQ_Sequence_443	WSEQ_DATA_WIDTH442 [2:0]			WSEQ_ADDR442 [12:0]													FFFFFFFFh
		WSEQ_DELAY442 [3:0]				WSEQ_DATA_START442 [3:0]				WSEQ_DATA442 [7:0]								
R13174 (3376h)	WSEQ_Sequence_444	WSEQ_DATA_WIDTH443 [2:0]			WSEQ_ADDR443 [12:0]													FFFFFFFFh
		WSEQ_DELAY443 [3:0]				WSEQ_DATA_START443 [3:0]				WSEQ_DATA443 [7:0]								
R13176 (3378h)	WSEQ_Sequence_445	WSEQ_DATA_WIDTH444 [2:0]			WSEQ_ADDR444 [12:0]													FFFFFFFFh
		WSEQ_DELAY444 [3:0]				WSEQ_DATA_START444 [3:0]				WSEQ_DATA444 [7:0]								
R13178 (337Ah)	WSEQ_Sequence_446	WSEQ_DATA_WIDTH445 [2:0]			WSEQ_ADDR445 [12:0]													FFFFFFFFh
		WSEQ_DELAY445 [3:0]				WSEQ_DATA_START445 [3:0]				WSEQ_DATA445 [7:0]								
R13180 (337Ch)	WSEQ_Sequence_447	WSEQ_DATA_WIDTH446 [2:0]			WSEQ_ADDR446 [12:0]													FFFFFFFFh
		WSEQ_DELAY446 [3:0]				WSEQ_DATA_START446 [3:0]				WSEQ_DATA446 [7:0]								
R13182 (337Eh)	WSEQ_Sequence_448	WSEQ_DATA_WIDTH447 [2:0]			WSEQ_ADDR447 [12:0]													FFFFFFFFh
		WSEQ_DELAY447 [3:0]				WSEQ_DATA_START447 [3:0]				WSEQ_DATA447 [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
R13184 (3380h)	WSEQ_Sequence_449	WSEQ_DATA_WIDTH448 [2:0]			WSEQ_ADDR448 [12:0]													FFFFFFFFh
		WSEQ_DELAY448 [3:0]				WSEQ_DATA_START448 [3:0]				WSEQ_DATA448 [7:0]								
R13186 (3382h)	WSEQ_Sequence_450	WSEQ_DATA_WIDTH449 [2:0]			WSEQ_ADDR449 [12:0]													FFFFFFFFh
		WSEQ_DELAY449 [3:0]				WSEQ_DATA_START449 [3:0]				WSEQ_DATA449 [7:0]								
R13188 (3384h)	WSEQ_Sequence_451	WSEQ_DATA_WIDTH450 [2:0]			WSEQ_ADDR450 [12:0]													FFFFFFFFh
		WSEQ_DELAY450 [3:0]				WSEQ_DATA_START450 [3:0]				WSEQ_DATA450 [7:0]								
R13190 (3386h)	WSEQ_Sequence_452	WSEQ_DATA_WIDTH451 [2:0]			WSEQ_ADDR451 [12:0]													FFFFFFFFh
		WSEQ_DELAY451 [3:0]				WSEQ_DATA_START451 [3:0]				WSEQ_DATA451 [7:0]								
R13192 (3388h)	WSEQ_Sequence_453	WSEQ_DATA_WIDTH452 [2:0]			WSEQ_ADDR452 [12:0]													FFFFFFFFh
		WSEQ_DELAY452 [3:0]				WSEQ_DATA_START452 [3:0]				WSEQ_DATA452 [7:0]								
R13194 (338Ah)	WSEQ_Sequence_454	WSEQ_DATA_WIDTH453 [2:0]			WSEQ_ADDR453 [12:0]													FFFFFFFFh
		WSEQ_DELAY453 [3:0]				WSEQ_DATA_START453 [3:0]				WSEQ_DATA453 [7:0]								
R13196 (338Ch)	WSEQ_Sequence_455	WSEQ_DATA_WIDTH454 [2:0]			WSEQ_ADDR454 [12:0]													FFFFFFFFh
		WSEQ_DELAY454 [3:0]				WSEQ_DATA_START454 [3:0]				WSEQ_DATA454 [7:0]								
R13198 (338Eh)	WSEQ_Sequence_456	WSEQ_DATA_WIDTH455 [2:0]			WSEQ_ADDR455 [12:0]													FFFFFFFFh
		WSEQ_DELAY455 [3:0]				WSEQ_DATA_START455 [3:0]				WSEQ_DATA455 [7:0]								
R13200 (3390h)	WSEQ_Sequence_457	WSEQ_DATA_WIDTH456 [2:0]			WSEQ_ADDR456 [12:0]													FFFFFFFFh
		WSEQ_DELAY456 [3:0]				WSEQ_DATA_START456 [3:0]				WSEQ_DATA456 [7:0]								
R13202 (3392h)	WSEQ_Sequence_458	WSEQ_DATA_WIDTH457 [2:0]			WSEQ_ADDR457 [12:0]													FFFFFFFFh
		WSEQ_DELAY457 [3:0]				WSEQ_DATA_START457 [3:0]				WSEQ_DATA457 [7:0]								
R13204 (3394h)	WSEQ_Sequence_459	WSEQ_DATA_WIDTH458 [2:0]			WSEQ_ADDR458 [12:0]													FFFFFFFFh
		WSEQ_DELAY458 [3:0]				WSEQ_DATA_START458 [3:0]				WSEQ_DATA458 [7:0]								
R13206 (3396h)	WSEQ_Sequence_460	WSEQ_DATA_WIDTH459 [2:0]			WSEQ_ADDR459 [12:0]													FFFFFFFFh
		WSEQ_DELAY459 [3:0]				WSEQ_DATA_START459 [3:0]				WSEQ_DATA459 [7:0]								
R13208 (3398h)	WSEQ_Sequence_461	WSEQ_DATA_WIDTH460 [2:0]			WSEQ_ADDR460 [12:0]													FFFFFFFFh
		WSEQ_DELAY460 [3:0]				WSEQ_DATA_START460 [3:0]				WSEQ_DATA460 [7:0]								
R13210 (339Ah)	WSEQ_Sequence_462	WSEQ_DATA_WIDTH461 [2:0]			WSEQ_ADDR461 [12:0]													FFFFFFFFh
		WSEQ_DELAY461 [3:0]				WSEQ_DATA_START461 [3:0]				WSEQ_DATA461 [7:0]								
R13212 (339Ch)	WSEQ_Sequence_463	WSEQ_DATA_WIDTH462 [2:0]			WSEQ_ADDR462 [12:0]													FFFFFFFFh
		WSEQ_DELAY462 [3:0]				WSEQ_DATA_START462 [3:0]				WSEQ_DATA462 [7:0]								
R13214 (339Eh)	WSEQ_Sequence_464	WSEQ_DATA_WIDTH463 [2:0]			WSEQ_ADDR463 [12:0]													FFFFFFFFh
		WSEQ_DELAY463 [3:0]				WSEQ_DATA_START463 [3:0]				WSEQ_DATA463 [7:0]								
R13216 (33A0h)	WSEQ_Sequence_465	WSEQ_DATA_WIDTH464 [2:0]			WSEQ_ADDR464 [12:0]													FFFFFFFFh
		WSEQ_DELAY464 [3:0]				WSEQ_DATA_START464 [3:0]				WSEQ_DATA464 [7:0]								
R13218 (33A2h)	WSEQ_Sequence_466	WSEQ_DATA_WIDTH465 [2:0]			WSEQ_ADDR465 [12:0]													FFFFFFFFh
		WSEQ_DELAY465 [3:0]				WSEQ_DATA_START465 [3:0]				WSEQ_DATA465 [7:0]								
R13220 (33A4h)	WSEQ_Sequence_467	WSEQ_DATA_WIDTH466 [2:0]			WSEQ_ADDR466 [12:0]													FFFFFFFFh
		WSEQ_DELAY466 [3:0]				WSEQ_DATA_START466 [3:0]				WSEQ_DATA466 [7:0]								
R13222 (33A6h)	WSEQ_Sequence_468	WSEQ_DATA_WIDTH467 [2:0]			WSEQ_ADDR467 [12:0]													FFFFFFFFh
		WSEQ_DELAY467 [3:0]				WSEQ_DATA_START467 [3:0]				WSEQ_DATA467 [7:0]								
R13224 (33A8h)	WSEQ_Sequence_469	WSEQ_DATA_WIDTH468 [2:0]			WSEQ_ADDR468 [12:0]													FFFFFFFFh
		WSEQ_DELAY468 [3:0]				WSEQ_DATA_START468 [3:0]				WSEQ_DATA468 [7:0]								
R13226 (33AAh)	WSEQ_Sequence_470	WSEQ_DATA_WIDTH469 [2:0]			WSEQ_ADDR469 [12:0]													FFFFFFFFh
		WSEQ_DELAY469 [3:0]				WSEQ_DATA_START469 [3:0]				WSEQ_DATA469 [7:0]								
R13228 (33ACh)	WSEQ_Sequence_471	WSEQ_DATA_WIDTH470 [2:0]			WSEQ_ADDR470 [12:0]													FFFFFFFFh
		WSEQ_DELAY470 [3:0]				WSEQ_DATA_START470 [3:0]				WSEQ_DATA470 [7:0]								
R13230 (33AEh)	WSEQ_Sequence_472	WSEQ_DATA_WIDTH471 [2:0]			WSEQ_ADDR471 [12:0]													FFFFFFFFh
		WSEQ_DELAY471 [3:0]				WSEQ_DATA_START471 [3:0]				WSEQ_DATA471 [7:0]								
R13232 (33B0h)	WSEQ_Sequence_473	WSEQ_DATA_WIDTH472 [2:0]			WSEQ_ADDR472 [12:0]													FFFFFFFFh
		WSEQ_DELAY472 [3:0]				WSEQ_DATA_START472 [3:0]				WSEQ_DATA472 [7:0]								
R13234 (33B2h)	WSEQ_Sequence_474	WSEQ_DATA_WIDTH473 [2:0]			WSEQ_ADDR473 [12:0]													FFFFFFFFh
		WSEQ_DELAY473 [3:0]				WSEQ_DATA_START473 [3:0]				WSEQ_DATA473 [7:0]								
R13236 (33B4h)	WSEQ_Sequence_475	WSEQ_DATA_WIDTH474 [2:0]			WSEQ_ADDR474 [12:0]													FFFFFFFFh
		WSEQ_DELAY474 [3:0]				WSEQ_DATA_START474 [3:0]				WSEQ_DATA474 [7:0]								
R13238 (33B6h)	WSEQ_Sequence_476	WSEQ_DATA_WIDTH475 [2:0]			WSEQ_ADDR475 [12:0]													FFFFFFFFh
		WSEQ_DELAY475 [3:0]				WSEQ_DATA_START475 [3:0]				WSEQ_DATA475 [7:0]								
R13240 (33B8h)	WSEQ_Sequence_477	WSEQ_DATA_WIDTH476 [2:0]			WSEQ_ADDR476 [12:0]													FFFFFFFFh
		WSEQ_DELAY476 [3:0]				WSEQ_DATA_START476 [3:0]				WSEQ_DATA476 [7:0]								
R13242 (33BAh)	WSEQ_Sequence_478	WSEQ_DATA_WIDTH477 [2:0]			WSEQ_ADDR477 [12:0]													FFFFFFFFh
		WSEQ_DELAY477 [3:0]				WSEQ_DATA_START477 [3:0]				WSEQ_DATA477 [7:0]								
R13244 (33BCh)	WSEQ_Sequence_479	WSEQ_DATA_WIDTH478 [2:0]			WSEQ_ADDR478 [12:0]													FFFFFFFFh
		WSEQ_DELAY478 [3:0]				WSEQ_DATA_START478 [3:0]				WSEQ_DATA478 [7:0]								
R13246 (33BEh)	WSEQ_Sequence_480	WSEQ_DATA_WIDTH479 [2:0]			WSEQ_ADDR479 [12:0]													FFFFFFFFh
		WSEQ_DELAY479 [3:0]				WSEQ_DATA_START479 [3:0]				WSEQ_DATA479 [7:0]								
R13248 (33C0h)	WSEQ_Sequence_481	WSEQ_DATA_WIDTH480 [2:0]			WSEQ_ADDR480 [12:0]													FFFFFFFFh
		WSEQ_DELAY480 [3:0]				WSEQ_DATA_START480 [3:0]				WSEQ_DATA480 [7:0]								
R13250 (33C2h)	WSEQ_Sequence_482	WSEQ_DATA_WIDTH481 [2:0]			WSEQ_ADDR481 [12:0]													FFFFFFFFh
		WSEQ_DELAY481 [3:0]				WSEQ_DATA_START481 [3:0]				WSEQ_DATA481 [7:0]								
R13252 (33C4h)	WSEQ_Sequence_483	WSEQ_DATA_WIDTH482 [2:0]			WSEQ_ADDR482 [12:0]													FFFFFFFFh
		WSEQ_DELAY482 [3:0]				WSEQ_DATA_START482 [3:0]				WSEQ_DATA482 [7:0]								
R13254 (33C6h)	WSEQ_Sequence_484	WSEQ_DATA_WIDTH483 [2:0]			WSEQ_ADDR483 [12:0]													FFFFFFFFh
		WSEQ_DELAY483 [3:0]				WSEQ_DATA_START483 [3:0]				WSEQ_DATA483 [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
R13256 (33C8h)	WSEQ_Sequence_485	WSEQ_DATA_WIDTH484 [2:0]			WSEQ_ADDR484 [12:0]													FFFFFFFFh
		WSEQ_DELAY484 [3:0]				WSEQ_DATA_START484 [3:0]				WSEQ_DATA484 [7:0]								
R13258 (33CAh)	WSEQ_Sequence_486	WSEQ_DATA_WIDTH485 [2:0]			WSEQ_ADDR485 [12:0]													FFFFFFFFh
		WSEQ_DELAY485 [3:0]				WSEQ_DATA_START485 [3:0]				WSEQ_DATA485 [7:0]								
R13260 (33CCh)	WSEQ_Sequence_487	WSEQ_DATA_WIDTH486 [2:0]			WSEQ_ADDR486 [12:0]													FFFFFFFFh
		WSEQ_DELAY486 [3:0]				WSEQ_DATA_START486 [3:0]				WSEQ_DATA486 [7:0]								
R13262 (33CEh)	WSEQ_Sequence_488	WSEQ_DATA_WIDTH487 [2:0]			WSEQ_ADDR487 [12:0]													FFFFFFFFh
		WSEQ_DELAY487 [3:0]				WSEQ_DATA_START487 [3:0]				WSEQ_DATA487 [7:0]								
R13264 (33D0h)	WSEQ_Sequence_489	WSEQ_DATA_WIDTH488 [2:0]			WSEQ_ADDR488 [12:0]													FFFFFFFFh
		WSEQ_DELAY488 [3:0]				WSEQ_DATA_START488 [3:0]				WSEQ_DATA488 [7:0]								
R13266 (33D2h)	WSEQ_Sequence_490	WSEQ_DATA_WIDTH489 [2:0]			WSEQ_ADDR489 [12:0]													FFFFFFFFh
		WSEQ_DELAY489 [3:0]				WSEQ_DATA_START489 [3:0]				WSEQ_DATA489 [7:0]								
R13268 (33D4h)	WSEQ_Sequence_491	WSEQ_DATA_WIDTH490 [2:0]			WSEQ_ADDR490 [12:0]													FFFFFFFFh
		WSEQ_DELAY490 [3:0]				WSEQ_DATA_START490 [3:0]				WSEQ_DATA490 [7:0]								
R13270 (33D6h)	WSEQ_Sequence_492	WSEQ_DATA_WIDTH491 [2:0]			WSEQ_ADDR491 [12:0]													FFFFFFFFh
		WSEQ_DELAY491 [3:0]				WSEQ_DATA_START491 [3:0]				WSEQ_DATA491 [7:0]								
R13272 (33D8h)	WSEQ_Sequence_493	WSEQ_DATA_WIDTH492 [2:0]			WSEQ_ADDR492 [12:0]													FFFFFFFFh
		WSEQ_DELAY492 [3:0]				WSEQ_DATA_START492 [3:0]				WSEQ_DATA492 [7:0]								
R13274 (33DAh)	WSEQ_Sequence_494	WSEQ_DATA_WIDTH493 [2:0]			WSEQ_ADDR493 [12:0]													FFFFFFFFh
		WSEQ_DELAY493 [3:0]				WSEQ_DATA_START493 [3:0]				WSEQ_DATA493 [7:0]								
R13276 (33DCh)	WSEQ_Sequence_495	WSEQ_DATA_WIDTH494 [2:0]			WSEQ_ADDR494 [12:0]													FFFFFFFFh
		WSEQ_DELAY494 [3:0]				WSEQ_DATA_START494 [3:0]				WSEQ_DATA494 [7:0]								
R13278 (33DEh)	WSEQ_Sequence_496	WSEQ_DATA_WIDTH495 [2:0]			WSEQ_ADDR495 [12:0]													FFFFFFFFh
		WSEQ_DELAY495 [3:0]				WSEQ_DATA_START495 [3:0]				WSEQ_DATA495 [7:0]								
R13280 (33E0h)	WSEQ_Sequence_497	WSEQ_DATA_WIDTH496 [2:0]			WSEQ_ADDR496 [12:0]													FFFFFFFFh
		WSEQ_DELAY496 [3:0]				WSEQ_DATA_START496 [3:0]				WSEQ_DATA496 [7:0]								
R13282 (33E2h)	WSEQ_Sequence_498	WSEQ_DATA_WIDTH497 [2:0]			WSEQ_ADDR497 [12:0]													FFFFFFFFh
		WSEQ_DELAY497 [3:0]				WSEQ_DATA_START497 [3:0]				WSEQ_DATA497 [7:0]								
R13284 (33E4h)	WSEQ_Sequence_499	WSEQ_DATA_WIDTH498 [2:0]			WSEQ_ADDR498 [12:0]													FFFFFFFFh
		WSEQ_DELAY498 [3:0]				WSEQ_DATA_START498 [3:0]				WSEQ_DATA498 [7:0]								
R13286 (33E6h)	WSEQ_Sequence_500	WSEQ_DATA_WIDTH499 [2:0]			WSEQ_ADDR499 [12:0]													FFFFFFFFh
		WSEQ_DELAY499 [3:0]				WSEQ_DATA_START499 [3:0]				WSEQ_DATA499 [7:0]								
R13288 (33E8h)	WSEQ_Sequence_501	WSEQ_DATA_WIDTH500 [2:0]			WSEQ_ADDR500 [12:0]													FFFFFFFFh
		WSEQ_DELAY500 [3:0]				WSEQ_DATA_START500 [3:0]				WSEQ_DATA500 [7:0]								
R13290 (33EAh)	WSEQ_Sequence_502	WSEQ_DATA_WIDTH501 [2:0]			WSEQ_ADDR501 [12:0]													FFFFFFFFh
		WSEQ_DELAY501 [3:0]				WSEQ_DATA_START501 [3:0]				WSEQ_DATA501 [7:0]								
R13292 (33ECh)	WSEQ_Sequence_503	WSEQ_DATA_WIDTH502 [2:0]			WSEQ_ADDR502 [12:0]													FFFFFFFFh
		WSEQ_DELAY502 [3:0]				WSEQ_DATA_START502 [3:0]				WSEQ_DATA502 [7:0]								
R13294 (33EEh)	WSEQ_Sequence_504	WSEQ_DATA_WIDTH503 [2:0]			WSEQ_ADDR503 [12:0]													FFFFFFFFh
		WSEQ_DELAY503 [3:0]				WSEQ_DATA_START503 [3:0]				WSEQ_DATA503 [7:0]								
R13296 (33F0h)	WSEQ_Sequence_505	WSEQ_DATA_WIDTH504 [2:0]			WSEQ_ADDR504 [12:0]													FFFFFFFFh
		WSEQ_DELAY504 [3:0]				WSEQ_DATA_START504 [3:0]				WSEQ_DATA504 [7:0]								
R13298 (33F2h)	WSEQ_Sequence_506	WSEQ_DATA_WIDTH505 [2:0]			WSEQ_ADDR505 [12:0]													FFFFFFFFh
		WSEQ_DELAY505 [3:0]				WSEQ_DATA_START505 [3:0]				WSEQ_DATA505 [7:0]								
R13300 (33F4h)	WSEQ_Sequence_507	WSEQ_DATA_WIDTH506 [2:0]			WSEQ_ADDR506 [12:0]													FFFFFFFFh
		WSEQ_DELAY506 [3:0]				WSEQ_DATA_START506 [3:0]				WSEQ_DATA506 [7:0]								
R13302 (33F6h)	WSEQ_Sequence_508	WSEQ_DATA_WIDTH507 [2:0]			WSEQ_ADDR507 [12:0]													FFFFFFFFh
		WSEQ_DELAY507 [3:0]				WSEQ_DATA_START507 [3:0]				WSEQ_DATA507 [7:0]								
R131076 (20004h)	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]								
R131078 (20006h)	OTP_HPDET_Cal_2	SPARE2 [7:0]								SPARE1 [7:0]								00000000h
		HP_GRADIENT_1X [7:0]								HP_GRADIENT_0X [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	
R294948 (48024h)	EVENTLOG1_EVENT_ STATUS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH16_ STS	EVENTLOG1_CH15_ STS	EVENTLOG1_CH14_ STS	EVENTLOG1_CH13_ STS	EVENTLOG1_CH12_ STS	EVENTLOG1_CH11_ STS	EVENTLOG1_CH10_ STS	EVENTLOG1_CH9_ STS	EVENTLOG1_CH8_ STS	EVENTLOG1_CH7_ STS	EVENTLOG1_CH6_ STS	EVENTLOG1_CH5_ STS	EVENTLOG1_CH4_ STS	EVENTLOG1_CH3_ STS	EVENTLOG1_CH2_ STS	EVENTLOG1_CH1_ STS	

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
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	EVENTLOG1_CH1_FILT	0	0	0	EVENTLOG1_CH1_SEL [9:0]										
R294978 (48042h)	EVENTLOG1_CH2_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH2_DB	EVENTLOG1_CH2_POL	EVENTLOG1_CH2_FILT	0	0	0	EVENTLOG1_CH2_SEL [9:0]										
R294980 (48044h)	EVENTLOG1_CH3_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH3_DB	EVENTLOG1_CH3_POL	EVENTLOG1_CH3_FILT	0	0	0	EVENTLOG1_CH3_SEL [9:0]										
R294982 (48046h)	EVENTLOG1_CH4_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH4_DB	EVENTLOG1_CH4_POL	EVENTLOG1_CH4_FILT	0	0	0	EVENTLOG1_CH4_SEL [9:0]										
R294984 (48048h)	EVENTLOG1_CH5_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH5_DB	EVENTLOG1_CH5_POL	EVENTLOG1_CH5_FILT	0	0	0	EVENTLOG1_CH5_SEL [9:0]										
R294986 (4804Ah)	EVENTLOG1_CH6_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH6_DB	EVENTLOG1_CH6_POL	EVENTLOG1_CH6_FILT	0	0	0	EVENTLOG1_CH6_SEL [9:0]										
R294988 (4804Ch)	EVENTLOG1_CH7_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH7_DB	EVENTLOG1_CH7_POL	EVENTLOG1_CH7_FILT	0	0	0	EVENTLOG1_CH7_SEL [9:0]										
R294990 (4804Eh)	EVENTLOG1_CH8_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH8_DB	EVENTLOG1_CH8_POL	EVENTLOG1_CH8_FILT	0	0	0	EVENTLOG1_CH8_SEL [9:0]										
R294992 (48050h)	EVENTLOG1_CH9_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH9_DB	EVENTLOG1_CH9_POL	EVENTLOG1_CH9_FILT	0	0	0	EVENTLOG1_CH9_SEL [9:0]										
R294994 (48052h)	EVENTLOG1_CH10_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH10_DB	EVENTLOG1_CH10_POL	EVENTLOG1_CH10_FILT	0	0	0	EVENTLOG1_CH10_SEL [9:0]										
R294996 (48054h)	EVENTLOG1_CH11_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH11_DB	EVENTLOG1_CH11_POL	EVENTLOG1_CH11_FILT	0	0	0	EVENTLOG1_CH11_SEL [9:0]										
R294998 (48056h)	EVENTLOG1_CH12_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH12_DB	EVENTLOG1_CH12_POL	EVENTLOG1_CH12_FILT	0	0	0	EVENTLOG1_CH12_SEL [9:0]										
R295000 (48058h)	EVENTLOG1_CH13_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH13_DB	EVENTLOG1_CH13_POL	EVENTLOG1_CH13_FILT	0	0	0	EVENTLOG1_CH13_SEL [9:0]										
R295002 (4805Ah)	EVENTLOG1_CH14_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH14_DB	EVENTLOG1_CH14_POL	EVENTLOG1_CH14_FILT	0	0	0	EVENTLOG1_CH14_SEL [9:0]										
R295004 (4805Ch)	EVENTLOG1_CH15_ DEFINE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		EVENTLOG1_CH15_DB	EVENTLOG1_CH15_POL	EVENTLOG1_CH15_FILT	0	0	0	EVENTLOG1_CH15_SEL [9:0]										
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	EVENTLOG1_CH16_FILT	0	0	0	EVENTLOG1_CH16_SEL [9:0]										
R295040 (48080h)	EVENTLOG1_FIFO0_ READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO0_POL	0	0	EVENTLOG1_FIFO0_ID [9:0]										
R295042 (48082h)	EVENTLOG1_FIFO0_ TIME	EVENTLOG1_FIFO0_TIME [31:16]																00000000h
		EVENTLOG1_FIFO0_TIME [15:0]																
R295044 (48084h)	EVENTLOG1_FIFO1_ READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO1_POL	0	0	EVENTLOG1_FIFO1_ID [9:0]										
R295046 (48086h)	EVENTLOG1_FIFO1_ TIME	EVENTLOG1_FIFO1_TIME [31:16]																00000000h
		EVENTLOG1_FIFO1_TIME [15:0]																
R295048 (48088h)	EVENTLOG1_FIFO2_ READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO2_POL	0	0	EVENTLOG1_FIFO2_ID [9:0]										
R295050 (4808Ah)	EVENTLOG1_FIFO2_ TIME	EVENTLOG1_FIFO2_TIME [31:16]																00000000h
		EVENTLOG1_FIFO2_TIME [15:0]																

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
R295052 (4808Ch)	EVENTLOG1_FIFO3_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO3_POL	0	0	EVENTLOG1_FIFO3_ID [9:0]										
R295054 (4808Eh)	EVENTLOG1_FIFO3_TIME	EVENTLOG1_FIFO3_TIME [31:16]																00000000h
		EVENTLOG1_FIFO3_TIME [15:0]																
R295056 (48090h)	EVENTLOG1_FIFO4_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO4_POL	0	0	EVENTLOG1_FIFO4_ID [9:0]										
R295058 (48092h)	EVENTLOG1_FIFO4_TIME	EVENTLOG1_FIFO4_TIME [31:16]																00000000h
		EVENTLOG1_FIFO4_TIME [15:0]																
R295060 (48094h)	EVENTLOG1_FIFO5_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO5_POL	0	0	EVENTLOG1_FIFO5_ID [9:0]										
R295062 (48096h)	EVENTLOG1_FIFO5_TIME	EVENTLOG1_FIFO5_TIME [31:16]																00000000h
		EVENTLOG1_FIFO5_TIME [15:0]																
R295064 (48098h)	EVENTLOG1_FIFO6_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO6_POL	0	0	EVENTLOG1_FIFO6_ID [9:0]										
R295066 (4809Ah)	EVENTLOG1_FIFO6_TIME	EVENTLOG1_FIFO6_TIME [31:16]																00000000h
		EVENTLOG1_FIFO6_TIME [15:0]																
R295068 (4809Ch)	EVENTLOG1_FIFO7_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO7_POL	0	0	EVENTLOG1_FIFO7_ID [9:0]										
R295070 (4809Eh)	EVENTLOG1_FIFO7_TIME	EVENTLOG1_FIFO7_TIME [31:16]																00000000h
		EVENTLOG1_FIFO7_TIME [15:0]																
R295072 (480A0h)	EVENTLOG1_FIFO8_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO8_POL	0	0	EVENTLOG1_FIFO8_ID [9:0]										
R295074 (480A2h)	EVENTLOG1_FIFO8_TIME	EVENTLOG1_FIFO8_TIME [31:16]																00000000h
		EVENTLOG1_FIFO8_TIME [15:0]																
R295076 (480A4h)	EVENTLOG1_FIFO9_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO9_POL	0	0	EVENTLOG1_FIFO9_ID [9:0]										
R295078 (480A6h)	EVENTLOG1_FIFO9_TIME	EVENTLOG1_FIFO9_TIME [31:16]																00000000h
		EVENTLOG1_FIFO9_TIME [15:0]																
R295080 (480A8h)	EVENTLOG1_FIFO10_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO10_POL	0	0	EVENTLOG1_FIFO10_ID [9:0]										
R295082 (480AAh)	EVENTLOG1_FIFO10_TIME	EVENTLOG1_FIFO10_TIME [31:16]																00000000h
		EVENTLOG1_FIFO10_TIME [15:0]																
R295084 (480ACh)	EVENTLOG1_FIFO11_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO11_POL	0	0	EVENTLOG1_FIFO11_ID [9:0]										
R295086 (480AEh)	EVENTLOG1_FIFO11_TIME	EVENTLOG1_FIFO11_TIME [31:16]																00000000h
		EVENTLOG1_FIFO11_TIME [15:0]																
R295088 (480B0h)	EVENTLOG1_FIFO12_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO12_POL	0	0	EVENTLOG1_FIFO12_ID [9:0]										
R295090 (480B2h)	EVENTLOG1_FIFO12_TIME	EVENTLOG1_FIFO12_TIME [31:16]																00000000h
		EVENTLOG1_FIFO12_TIME [15:0]																
R295092 (480B4h)	EVENTLOG1_FIFO13_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO13_POL	0	0	EVENTLOG1_FIFO13_ID [9:0]										
R295094 (480B6h)	EVENTLOG1_FIFO13_TIME	EVENTLOG1_FIFO13_TIME [31:16]																00000000h
		EVENTLOG1_FIFO13_TIME [15:0]																
R295096 (480B8h)	EVENTLOG1_FIFO14_READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO14_POL	0	0	EVENTLOG1_FIFO14_ID [9:0]										
R295098 (480BAh)	EVENTLOG1_FIFO14_TIME	EVENTLOG1_FIFO14_TIME [31:16]																00000000h
		EVENTLOG1_FIFO14_TIME [15:0]																

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
R295100 (480BCh)	EVENTLOG1_FIFO15_ READ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	EVENTLOG1_FIFO15_POL	0	0	EVENTLOG1_FIFO15_ID [9:0]										
R295102 (480BEh)	EVENTLOG1_FIFO15_ TIME	EVENTLOG1_FIFO15_TIME [31:16]																00000000h
		EVENTLOG1_FIFO15_TIME [15:0]																
R303104 (4A000h)	ALM1_CFG	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	ALM1_TIMER_SEL	
R303120 (4A010h)	ALM1_CONFIG1	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	ALM1_CH1_CONT	0	0	ALM1_CH1_TRIG_MODE[1:0]		
R303122 (4A012h)	ALM1_CTRL1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		ALM1_CH1_UPD	0	0	0	0	0	0	0	0	0	0	ALM1_CH1_STOP	0	0	0	ALM1_CH1_START	
R303124 (4A0124h)	ALM1_TRIG_VAL1	ALM1_CH1_TRIG_VAL[31:16]																00000000h
		ALM1_CH1_TRIG_VAL[15:0]																
R303126 (4A016h)	ALM1_PULSE_DUR1	ALM1_CH1_PULSE_DUR[31:16]																00000000h
		ALM1_CH1_PULSE_DUR[15:0]																
R303128 (4A018h)	ALM1_STATUS1	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	ALM1_CH1_STS	
R303136 (4A020h)	ALM1_CONFIG2	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	ALM1_CH2_CONT	0	0	ALM1_CH2_TRIG_MODE[1:0]		
R303138 (4A022h)	ALM1_CTRL2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		ALM1_CH2_UPD	0	0	0	0	0	0	0	0	0	0	ALM1_CH2_STOP	0	0	0	ALM1_CH2_START	
R303140 (4A024h)	ALM1_TRIG_VAL2	ALM1_CH2_TRIG_VAL[31:16]																00000000h
		ALM1_CH2_TRIG_VAL[15:0]																
R303142 (4A026h)	ALM1_PULSE_DUR2	ALM1_CH2_PULSE_DUR[31:16]																00000000h
		ALM1_CH2_PULSE_DUR[15:0]																
R303144 (4A028h)	ALM1_STATUS2	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	ALM1_CH2_STS	
R303152 (4A030h)	ALM1_CONFIG3	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	ALM1_CH3_CONT	0	0	ALM1_CH3_TRIG_MODE[1:0]		
R303154 (4A032h)	ALM1_CTRL3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		ALM1_CH3_UPD	0	0	0	0	0	0	0	0	0	0	ALM1_CH3_STOP	0	0	0	ALM1_CH3_START	
R303156 (4A034h)	ALM1_TRIG_VAL3	ALM1_CH3_TRIG_VAL[31:16]																00000000h
		ALM1_CH3_TRIG_VAL[15:0]																
R303158 (4A036h)	ALM1_PULSE_DUR3	ALM1_CH3_PULSE_DUR[31:16]																00000000h
		ALM1_CH3_PULSE_DUR[15:0]																
R303160 (4A038h)	ALM1_STATUS3	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	ALM1_CH3_STS	
R303168 (4A040h)	ALM1_CONFIG4	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	ALM1_CH4_CONT	0	0	ALM1_CH4_TRIG_MODE[1:0]		
R303170 (4A042h)	ALM1_CTRL4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		ALM1_CH4_UPD	0	0	0	0	0	0	0	0	0	0	ALM1_CH4_STOP	0	0	0	ALM1_CH4_START	
R303172 (4A044h)	ALM1_TRIG_VAL4	ALM1_CH4_TRIG_VAL[31:16]																00000000h
		ALM1_CH4_TRIG_VAL[15:0]																
R303174 (4A046h)	ALM1_PULSE_DUR4	ALM1_CH4_PULSE_DUR[31:16]																00000000h
		ALM1_CH4_PULSE_DUR[15:0]																
R303176 (4A048h)	ALM1_STATUS4	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	ALM1_CH4_STS	
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	

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
R311304 (4C008h)	Timer1_Status	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	TIMER1_RUNNING_STS	
R311306 (4C00Ah)	Timer1_Count_Readback	TIMER1_CUR_COUNT [31:16]																00000000h
		TIMER1_CUR_COUNT [15:0]																
R311308 (4C00Ch)	Timer1_DSP_Clock_Config	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		TIMER1_DSPCLK_FREQ_SEL [15:0]																
R311310 (4C00Eh)	Timer1_DSP_Clock_Status	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		TIMER1_DSPCLK_FREQ_STS [15:0]																
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	
R315552 (4D0A0h)	DSPGP_SET5_Mask_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000FFFFh
		DSPGP16_SET5_MASK	DSPGP15_SET5_MASK	DSPGP14_SET5_MASK	DSPGP13_SET5_MASK	DSPGP12_SET5_MASK	DSPGP11_SET5_MASK	DSPGP10_SET5_MASK	DSPGP9_SET5_MASK	DSPGP8_SET5_MASK	DSPGP7_SET5_MASK	DSPGP6_SET5_MASK	DSPGP5_SET5_MASK	DSPGP4_SET5_MASK	DSPGP3_SET5_MASK	DSPGP2_SET5_MASK	DSPGP1_SET5_MASK	
R315560 (4D0A8h)	DSPGP_SET5_Direction_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000FFFFh
		DSPGP16_SET5_DIR	DSPGP15_SET5_DIR	DSPGP14_SET5_DIR	DSPGP13_SET5_DIR	DSPGP12_SET5_DIR	DSPGP11_SET5_DIR	DSPGP10_SET5_DIR	DSPGP9_SET5_DIR	DSPGP8_SET5_DIR	DSPGP7_SET5_DIR	DSPGP6_SET5_DIR	DSPGP5_SET5_DIR	DSPGP4_SET5_DIR	DSPGP3_SET5_DIR	DSPGP2_SET5_DIR	DSPGP1_SET5_DIR	
R315568 (4D0B0h)	DSPGP_SET5_Level_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSPGP16_SET5_LVL	DSPGP15_SET5_LVL	DSPGP14_SET5_LVL	DSPGP13_SET5_LVL	DSPGP12_SET5_LVL	DSPGP11_SET5_LVL	DSPGP10_SET5_LVL	DSPGP9_SET5_LVL	DSPGP8_SET5_LVL	DSPGP7_SET5_LVL	DSPGP6_SET5_LVL	DSPGP5_SET5_LVL	DSPGP4_SET5_LVL	DSPGP3_SET5_LVL	DSPGP2_SET5_LVL	DSPGP1_SET5_LVL	
R315584 (4D0C0h)	DSPGP_SET6_Mask_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000FFFFh
		DSPGP16_SET6_MASK	DSPGP15_SET6_MASK	DSPGP14_SET6_MASK	DSPGP13_SET6_MASK	DSPGP12_SET6_MASK	DSPGP11_SET6_MASK	DSPGP10_SET6_MASK	DSPGP9_SET6_MASK	DSPGP8_SET6_MASK	DSPGP7_SET6_MASK	DSPGP6_SET6_MASK	DSPGP5_SET6_MASK	DSPGP4_SET6_MASK	DSPGP3_SET6_MASK	DSPGP2_SET6_MASK	DSPGP1_SET6_MASK	
R315592 (4D0C8h)	DSPGP_SET6_Direction_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000FFFFh
		DSPGP16_SET6_DIR	DSPGP15_SET6_DIR	DSPGP14_SET6_DIR	DSPGP13_SET6_DIR	DSPGP12_SET6_DIR	DSPGP11_SET6_DIR	DSPGP10_SET6_DIR	DSPGP9_SET6_DIR	DSPGP8_SET6_DIR	DSPGP7_SET6_DIR	DSPGP6_SET6_DIR	DSPGP5_SET6_DIR	DSPGP4_SET6_DIR	DSPGP3_SET6_DIR	DSPGP2_SET6_DIR	DSPGP1_SET6_DIR	
R315600 (4D0D0h)	DSPGP_SET6_Level_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSPGP16_SET6_LVL	DSPGP15_SET6_LVL	DSPGP14_SET6_LVL	DSPGP13_SET6_LVL	DSPGP12_SET6_LVL	DSPGP11_SET6_LVL	DSPGP10_SET6_LVL	DSPGP9_SET6_LVL	DSPGP8_SET6_LVL	DSPGP7_SET6_LVL	DSPGP6_SET6_LVL	DSPGP5_SET6_LVL	DSPGP4_SET6_LVL	DSPGP3_SET6_LVL	DSPGP2_SET6_LVL	DSPGP1_SET6_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
R315616 (4D0E0h)	DSPGP_SET7_Mask_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000FFFFh
		DSPGP16_SET7_MASK	DSPGP15_SET7_MASK	DSPGP14_SET7_MASK	DSPGP13_SET7_MASK	DSPGP12_SET7_MASK	DSPGP11_SET7_MASK	DSPGP10_SET7_MASK	DSPGP9_SET7_MASK	DSPGP8_SET7_MASK	DSPGP7_SET7_MASK	DSPGP6_SET7_MASK	DSPGP5_SET7_MASK	DSPGP4_SET7_MASK	DSPGP3_SET7_MASK	DSPGP2_SET7_MASK	DSPGP1_SET7_MASK	
R315624 (4D0E8h)	DSPGP_SET7_Direction_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000FFFFh
		DSPGP16_SET7_DIR	DSPGP15_SET7_DIR	DSPGP14_SET7_DIR	DSPGP13_SET7_DIR	DSPGP12_SET7_DIR	DSPGP11_SET7_DIR	DSPGP10_SET7_DIR	DSPGP9_SET7_DIR	DSPGP8_SET7_DIR	DSPGP7_SET7_DIR	DSPGP6_SET7_DIR	DSPGP5_SET7_DIR	DSPGP4_SET7_DIR	DSPGP3_SET7_DIR	DSPGP2_SET7_DIR	DSPGP1_SET7_DIR	
R315632 (4D0F0h)	DSPGP_SET7_Level_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSPGP16_SET7_LVL	DSPGP15_SET7_LVL	DSPGP14_SET7_LVL	DSPGP13_SET7_LVL	DSPGP12_SET7_LVL	DSPGP11_SET7_LVL	DSPGP10_SET7_LVL	DSPGP9_SET7_LVL	DSPGP8_SET7_LVL	DSPGP7_SET7_LVL	DSPGP6_SET7_LVL	DSPGP5_SET7_LVL	DSPGP4_SET7_LVL	DSPGP3_SET7_LVL	DSPGP2_SET7_LVL	DSPGP1_SET7_LVL	
R315648 (4D100h)	DSPGP_SET8_Mask_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000FFFFh
		DSPGP16_SET8_MASK	DSPGP15_SET8_MASK	DSPGP14_SET8_MASK	DSPGP13_SET8_MASK	DSPGP12_SET8_MASK	DSPGP11_SET8_MASK	DSPGP10_SET8_MASK	DSPGP9_SET8_MASK	DSPGP8_SET8_MASK	DSPGP7_SET8_MASK	DSPGP6_SET8_MASK	DSPGP5_SET8_MASK	DSPGP4_SET8_MASK	DSPGP3_SET8_MASK	DSPGP2_SET8_MASK	DSPGP1_SET8_MASK	
R315656 (4D108h)	DSPGP_SET8_Direction_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000FFFFh
		DSPGP16_SET8_DIR	DSPGP15_SET8_DIR	DSPGP14_SET8_DIR	DSPGP13_SET8_DIR	DSPGP12_SET8_DIR	DSPGP11_SET8_DIR	DSPGP10_SET8_DIR	DSPGP9_SET8_DIR	DSPGP8_SET8_DIR	DSPGP7_SET8_DIR	DSPGP6_SET8_DIR	DSPGP5_SET8_DIR	DSPGP4_SET8_DIR	DSPGP3_SET8_DIR	DSPGP2_SET8_DIR	DSPGP1_SET8_DIR	
R315664 (4D110h)	DSPGP_SET8_Level_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSPGP16_SET8_LVL	DSPGP15_SET8_LVL	DSPGP14_SET8_LVL	DSPGP13_SET8_LVL	DSPGP12_SET8_LVL	DSPGP11_SET8_LVL	DSPGP10_SET8_LVL	DSPGP9_SET8_LVL	DSPGP8_SET8_LVL	DSPGP7_SET8_LVL	DSPGP6_SET8_LVL	DSPGP5_SET8_LVL	DSPGP4_SET8_LVL	DSPGP3_SET8_LVL	DSPGP2_SET8_LVL	DSPGP1_SET8_LVL	
R524288 (80000h)	DSP1_PMEM_0	0	0	0	0	0	0	0	0	DSP1_PM_START [39:32]								00000000h
		DSP1_PM_START [31:16]																
R524290 (80002h)	DSP1_PMEM_1	DSP1_PM_START [15:0]																00000000h
		0	0	0	0	0	0	0	0	DSP1_PM_1 [39:32]								
R524292 (80004h)	DSP1_PMEM_2	DSP1_PM_1 [31:16]																00000000h
		DSP1_PM_1 [15:0]																
R561146 (88FFAh)	DSP1_PMEM_18429	0	0	0	0	0	0	0	0	DSP1_PM_12286 [39:32]								00000000h
		DSP1_PM_12286 [31:16]																
R561148 (88FFCh)	DSP1_PMEM_18430	DSP1_PM_12286 [15:0]																00000000h
		0	0	0	0	0	0	0	0	DSP1_PM_END [39:32]								
R561150 (88FFEh)	DSP1_PMEM_18431	DSP1_PM_END [31:16]																00000000h
		DSP1_PM_END [15:0]																
R655360 (A0000h)	DSP1_XMEM_0	0	0	0	0	0	0	0	0	DSP1_XM_START [23:16]								00000000h
		DSP1_XM_START [15:0]																
R655362 (A0002h)	DSP1_XMEM_1	0	0	0	0	0	0	0	0	DSP1_XM_1 [23:16]								00000000h
		DSP1_XM_1 [15:0]																
R696316 (A9FFCh)	DSP1_XMEM_20478	0	0	0	0	0	0	0	0	DSP1_XM_20478 [23:16]								00000000h
		DSP1_XM_20478 [15:0]																
R696318 (A9FFEh)	DSP1_XMEM_20479	0	0	0	0	0	0	0	0	DSP1_XM_END [23:16]								00000000h
		DSP1_XM_END [15:0]																
R786432 (C0000h)	DSP1_YMEM_0	0	0	0	0	0	0	0	0	DSP1_YM_START [23:16]								00000000h
		DSP1_YM_START [15:0]																
R786434 (C0002h)	DSP1_YMEM_1	0	0	0	0	0	0	0	0	DSP1_YM_1 [23:16]								00000000h
		DSP1_YM_1 [15:0]																
R802812 (C3FFCh)	DSP1_YMEM_8190	0	0	0	0	0	0	0	0	DSP1_YM_8190 [23:16]								00000000h
		DSP1_YM_8190 [15:0]																
R802814 (C3FFEh)	DSP1_YMEM_8191	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	0	0	0	00000000h
		0	DSP1_RATE [3:0]				0	0	0	0	0	0	DSP1_MEM_ENA	DSP1_DBG_CLK_ENA	0	DSP1_CORE_ENA	DSP1_START	
R1048066 (FFE02h)	DSP1_Config_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP1_CLK_FREQ_SEL [15:0]																
R1048068 (FFE04h)	DSP1_Status_1	DSP1_PING_FULL	DSP1_PONG_FULL	0	0	0	0	0	0	DSP1_WDMA_ACTIVE_CHANNELS [7:0]								00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
R1048070 (FFE06h)	DSP1_Status_2	DSP1_DUALMEM_COLLISION_ADDR [15:0]																00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_CLK_AVAIL	
R1048072 (FFE08h)	DSP1_Status_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP1_CLK_FREQ_STS [15:0]																
R1048074 (FFE0Ah)	DSP1_Watchdog_1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	DSP1_WDT_MAX_COUNT [3:0]				DSP1_WDT_ENA	

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
R1048080 (FFE10h)	DSP1_WDMA_Buffer_1	DSP1_START_ADDRESS_WDMA_BUFFER_1 [15:0]																00000000h
		DSP1_START_ADDRESS_WDMA_BUFFER_0 [15:0]																
R1048082 (FFE12h)	DSP1_WDMA_Buffer_2	DSP1_START_ADDRESS_WDMA_BUFFER_3 [15:0]																00000000h
		DSP1_START_ADDRESS_WDMA_BUFFER_2 [15:0]																
R1048084 (FFE14h)	DSP1_WDMA_Buffer_3	DSP1_START_ADDRESS_WDMA_BUFFER_5 [15:0]																00000000h
		DSP1_START_ADDRESS_WDMA_BUFFER_4 [15:0]																
R1048086 (FFE16h)	DSP1_WDMA_Buffer_4	DSP1_START_ADDRESS_WDMA_BUFFER_7 [15:0]																00000000h
		DSP1_START_ADDRESS_WDMA_BUFFER_6 [15:0]																
R1048096 (FFE20h)	DSP1_RDMA_Buffer_1	DSP1_START_ADDRESS_RDMA_BUFFER_1 [15:0]																00000000h
		DSP1_START_ADDRESS_RDMA_BUFFER_0 [15:0]																
R1048098 (FFE22h)	DSP1_RDMA_Buffer_2	DSP1_START_ADDRESS_RDMA_BUFFER_3 [15:0]																00000000h
		DSP1_START_ADDRESS_RDMA_BUFFER_2 [15:0]																
R1048100 (FFE24h)	DSP1_RDMA_Buffer_3	DSP1_START_ADDRESS_RDMA_BUFFER_5 [15:0]																00000000h
		DSP1_START_ADDRESS_RDMA_BUFFER_4 [15:0]																
R1048112 (FFE30h)	DSP1_DMA_Config_1	0	0	0	0	0	0	0	0	DSP1_WDMA_CHANNEL_ENABLE [7:0]								00000000h
		0	0	DSP1_DMA_BUFFER_LENGTH [13:0]														
R1048114 (FFE32h)	DSP1_DMA_Config_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	DSP1_WDMA_CHANNEL_OFFSET [7:0]								
R1048116 (FFE34h)	DSP1_DMA_Config_3	0	0	0	0	0	0	0	0	0	0	DSP1_RDMA_CHANNEL_OFFSET [5:0]						00000000h
		0	0	0	0	0	0	0	0	0	0	DSP1_RDMA_CHANNEL_ENABLE [5:0]						
R1048118 (FFE36h)	DSP1_DMA_Config_4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DSP1_DMA_WORD_SEL	
R1048120 (FFE38h)	DSP1_External_Start	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	DSP1_START_IN_SEL [4:0]					
R1048128 (FFE40h)	DSP1_Scratch_1	DSP1_SCRATCH_1 [15:0]																00000000h
		DSP1_SCRATCH_0 [15:0]																
R1048130 (FFE42h)	DSP1_Scratch_2	DSP1_SCRATCH_3 [15:0]																00000000h
		DSP1_SCRATCH_2 [15:0]																
R1048146 (FFE52h)	DSP1_Bus_Error_Addr	0	0	0	0	0	0	0	0	DSP1_BUS_ERR_ADDR [23:16]								00000000h
		DSP1_BUS_ERR_ADDR [15:0]																
R1048148 (FFE54h)	DSP1_Ext_window_A	DSP1_EXT_A_PSIZE16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP1_EXT_A_PAGE [15:0]																
R1048150 (FFE56h)	DSP1_Ext_window_B	DSP1_EXT_B_PSIZE16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP1_EXT_B_PAGE [15:0]																
R1048152 (FFE58h)	DSP1_Ext_window_C	DSP1_EXT_C_PSIZE16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP1_EXT_C_PAGE [15:0]																
R1048154 (FFE5Ah)	DSP1_Ext_window_D	DSP1_EXT_D_PSIZE16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP1_EXT_D_PAGE [15:0]																
R1048158 (FFE5Eh)	DSP1_Watchdog_2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP1_WDT_RESET [15:0]																
R1048160 (FFE60h)	DSP1_Identity	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	0	0	0	0	0	DSP1_CORE_NUMBER [4:0]					
R1048164 (FFE64h)	DSP1_Region_lock_sts_0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		0	0	0	0	0	0	DSP1_CTRL_REGION9_LOCK_STS	DSP1_CTRL_REGION8_LOCK_STS	DSP1_CTRL_REGION7_LOCK_STS	DSP1_CTRL_REGION6_LOCK_STS	DSP1_CTRL_REGION5_LOCK_STS	DSP1_CTRL_REGION4_LOCK_STS	DSP1_CTRL_REGION3_LOCK_STS	DSP1_CTRL_REGION2_LOCK_STS	DSP1_CTRL_REGION1_LOCK_STS	DSP1_CTRL_REGION0_LOCK_STS	
R1048166 (FFE66h)	DSP1_Region_lock_1__DSP1_Region_lock_0	DSP1_CTRL_REGION1_LOCK [15:0]																00000000h
		DSP1_CTRL_REGION0_LOCK [15:0]																
R1048168 (FFE68h)	DSP1_Region_lock_3__DSP1_Region_lock_2	DSP1_CTRL_REGION3_LOCK [15:0]																00000000h
		DSP1_CTRL_REGION2_LOCK [15:0]																
R1048170 (FFE6Ah)	DSP1_Region_lock_5__DSP1_Region_lock_4	DSP1_CTRL_REGION5_LOCK [15:0]																00000000h
		DSP1_CTRL_REGION4_LOCK [15:0]																
R1048172 (FFE6Ch)	DSP1_Region_lock_7__DSP1_Region_lock_6	DSP1_CTRL_REGION7_LOCK [15:0]																00000000h
		DSP1_CTRL_REGION6_LOCK [15:0]																
R1048174 (FFE6Eh)	DSP1_Region_lock_9__DSP1_Region_lock_8	DSP1_CTRL_REGION9_LOCK [15:0]																00000000h
		DSP1_CTRL_REGION8_LOCK [15:0]																
R1048186 (FFE7Ah)	DSP1_Region_lock_ctrl_0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000000h
		DSP1_LOCK_ERR_STS	DSP1_ADDR_ERR_STS	DSP1_WDT_TIMEOUT_STS	0	0	0	0	0	0	0	0	0	0	0	DSP1_ERR_PAUSE	DSP1_ERR_CLEAR	
R1048188 (FFE7Ch)	DSP1_PMEM_Err_Addr__XMEM_Err_Addr	0	DSP1_PMEM_ERR_ADDR [14:0]															00000000h
		DSP1_XMEM_ERR_ADDR [15:0]																

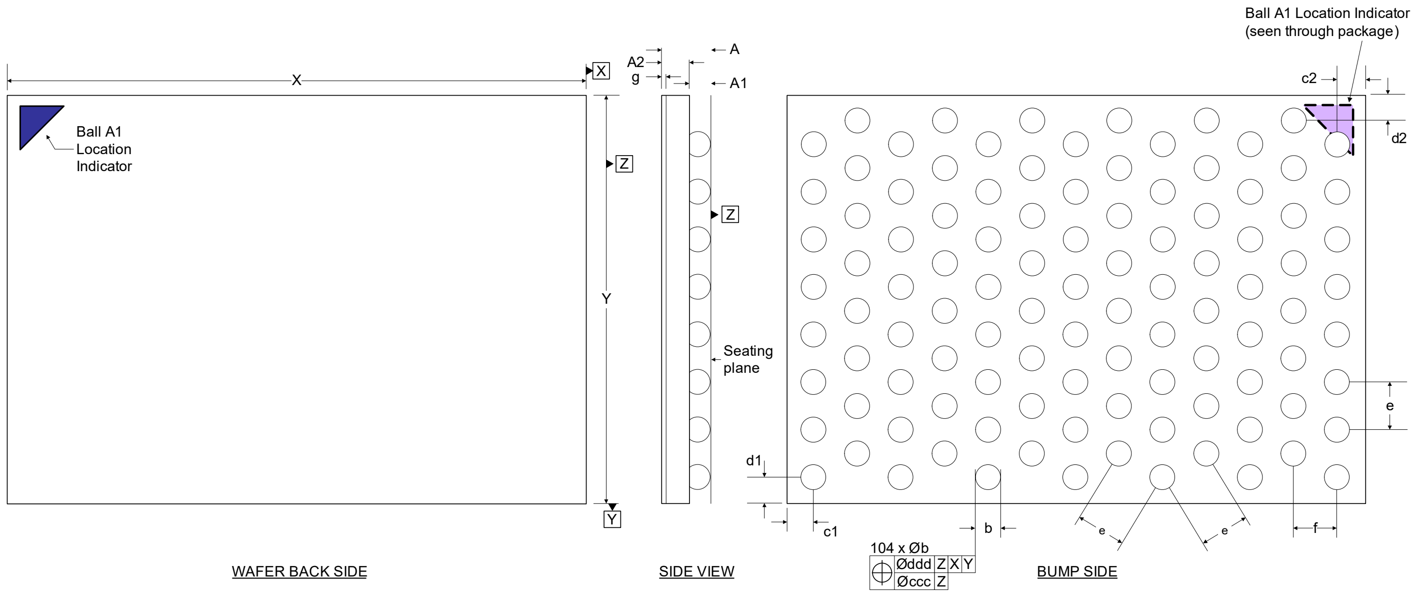
7 Thermal Characteristics

Table 7-1. Typical JEDEC Four-Layer, 2s2p Board Thermal Characteristics

Parameter	Symbol	WLCSP	Units
Junction-to-ambient thermal resistance	θ_{JA}	38.5	°C/W
Junction-to-board thermal resistance	θ_{JB}	9.3	°C/W
Junction-to-case thermal resistance	θ_{JC}	2.11	°C/W
Junction-to-board thermal-characterization parameter	Ψ_{JB}	9.3	°C/W
Junction-to-package-top thermal-characterization parameter	Ψ_{JT}	0.10	°C/W

Notes:
- Natural convection at the maximum recommended operating temperature T_A (see Table 3-3)
- Four-layer, 2s2p PCB as specified by JESD51-9 and JESD51-11; dimensions: 101.5 x 114.5 x 1.6 mm
- Thermal parameters as defined by JESD51-12

8 Package Dimensions



Notes:
- Dimensioning and tolerances per ASME Y 14.5M–2009.
- The Ball A1 position indicator is for illustration purposes only and may not be to scale.
- Dimension "b" applies to the solder sphere diameter and is measured at the midpoint between the package body and the seating plane Datum Z.

Table 8-1. WLCSP Package Dimensions

Dimension	Millimeters		
	Minimum	Nominal	Maximum
A	0.464	0.494	0.524
A1	0.161	0.19	0.219
A2	0.289	0.304	0.319
b	0.24	0.27	0.3
c1	0.3279	0.3579	0.3879
d1	0.2062	0.2362	0.2662
c2	0.2307	0.2607	0.2907
d2	0.206	0.236	0.266
e	BSC	0.4	BSC
f	BSC	0.3464	BSC
g	REF	0.3464	BSC
X	4.7504	4.7754	4.8004
Y	3.4472	3.4722	3.4972
ccc = 0.05 ddd = 0.15			

Note: Controlling dimension is millimeters.

9 Ordering Information

Table 9-1. Ordering Information

Product	Description	Package	Halogen Free	Pb Free	Grade	Temperature Range	Container	Order #
CS42L92	32-Bit 384-kHz Hi-Fi Audio Codec	104-ball WLCSP	Yes	Yes	Commercial	–40 to +85°C	Tape and Reel [1]	CS42L92-CWZR

1. Reel quantity = 6000 units.

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.* http://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 NOV '17	• AVDD maximum rating amended (Table 3-2). • DSP memory sizes amended (Table 4-29). • Package dimension tolerances added (Section 8).

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.