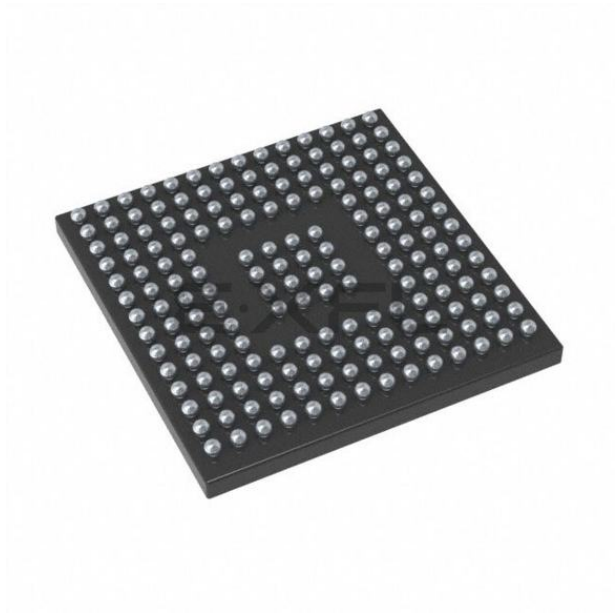

**Welcome to E-XFL.COM**

## Understanding Embedded - DSP (Digital Signal Processors)

Embedded - DSP (Digital Signal Processors) are specialized microprocessors designed to perform complex mathematical computations on digital signals in real-time. Unlike general-purpose processors, DSPs are optimized for high-speed numeric processing tasks, making them ideal for applications that require efficient and precise manipulation of digital data. These processors are fundamental in converting and processing signals in various forms, including audio, video, and communication signals, ensuring that data is accurately interpreted and utilized in embedded systems.

## Applications of Embedded - DSP (Digital Signal Processors)

| Details | |
|---|---|
| Product Status | Not For New Designs |
| Type | Audio Codec |
| Interface | I²C, SLIMbus, SPI |
| Clock Rate | 900MIPS |
| Non-Volatile Memory | - |
| On-Chip RAM | - |
| Voltage - I/O | 1.2V, 1.8V, 3.6V |
| Voltage - Core | 1.20V |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 176-UFBGA, WLCSP |
| Supplier Device Package | 176-WCSP (5.86x5.96) |
| Purchase URL | https://www.e-xfl.com/product-detail/cirrus-logic/cs47l85-cwzr |

| PIN NO | NAME | TYPE | DESCRIPTION |
|---|---|---|---|
| L13 | SLIMDAT | Digital Input / Output | SLIM Bus Data input / output |
| N12 | SPKCLK1/ GPIO37 | Digital Input / Output | Digital speaker (PDM) 1 clock output / GPIO37.<br>GPIO output is selectable CMOS or Open Drain;<br>SPKCCLK output is CMOS. |
| M11 | SPKDAT1/ GPIO39 | Digital Input / Output | Digital speaker (PDM) 1 data output / GPIO39.<br>GPIO output is selectable CMOS or Open Drain;<br>SPKDAT output is CMOS. |
| P11 | SPKCLK2/ GPIO38 | Digital Input / Output | Digital speaker (PDM) 2 clock output / GPIO38.<br>GPIO output is selectable CMOS or Open Drain;<br>SPKCLK output is CMOS. |
| N11 | SPKDAT2/ GPIO40 | Digital Input / Output | Digital speaker (PDM) 2 data output / GPIO40.<br>GPIO output is selectable CMOS or Open Drain;<br>SPKDAT output is CMOS. |
| M1 | SPKGNDLN | Supply | Left speaker driver ground (Return path for SPKVDDL).<br>See note. |
| M2 | SPKGNDLP | Supply | Left speaker driver ground (Return path for SPKVDDL).<br>See note. |
| L1 | SPKGNDRN | Supply | Right speaker driver ground (Return path for SPKVDDR).<br>See note. |
| L2 | SPKGNDRP | Supply | Right speaker driver ground (Return path for SPKVDDR).<br>See note. |
| N1 | SPKOUTLN | Analogue Output | Left speaker negative output |
| N2 | SPKOUTLP | Analogue Output | Left speaker positive output |
| K1 | SPKOUTRN | Analogue Output | Right speaker negative output |
| K2 | SPKOUTRP | Analogue Output | Right speaker positive output |
| G2 | SPKTST1 | Analogue Output | Test function (recommend no external connection) |
| G3 | SPKTST2 | Analogue Output | Test function (recommend no external connection) |
| P1, P2 | SPKVDDL | Supply | Left speaker driver supply |
| J1, J2 | SPKVDDR | Supply | Right speaker driver supply |
| D13, H1 | SUBGND | Supply | Substrate ground |
| H8 | TCK | Digital Input | JTAG clock input.<br>Internal pull-down holds this pin at logic 0 for normal operation. |
| J9 | TDI | Digital Input | JTAG data input.<br>Internal pull-down holds this pin at logic 0 for normal operation. |
| H9 | TDO | Digital Output | JTAG data output |
| J8 | TMS | Digital Input | JTAG mode select input.<br>Internal pull-down holds this pin at logic 0 for normal operation. |
| G8 | TRST | Digital Input | JTAG Test Access Port reset (active low).<br>Internal pull-down holds this pin at logic 0 for normal operation.<br>External connection to DGND is recommended, if the JTAG interface function is not required. |
| G1 | VREFC | Analogue Output | Bandgap reference external capacitor connection |

**Note:**

Separate P/N ground connections are provided for each speaker driver channel; this provides flexible support for current monitoring and output protection circuits. If this option is not used, then the respective ground connections should be tied together on the PCB.

The following table identifies the power domain and ground reference associated with each of the input / output pins.

| PIN NO | NAME | POWER DOMAIN | GROUND DOMAIN |
|---|---|---|---|
| H13 | AIF1BCLK/GPIO16 | DBVDD1 | DGND |
| H12 | AIF1LRCLK/GPIO18 | DBVDD1 | DGND |
| G12 | AIF1RXDAT/GPIO17 | DBVDD1 | DGND |
| G11 | AIF1TXDAT/GPIO15 | DBVDD1 | DGND |
| M12 | AIF2BCLK/GPIO20 | DBVDD2 | DGND |
| L10 | AIF2LRCLK/GPIO22 | DBVDD2 | DGND |
| N13 | AIF2RXDAT/GPIO21 | DBVDD2 | DGND |
| L11 | AIF2TXDAT/GPIO19 | DBVDD2 | DGND |
| M4 | AIF3BCLK/GPIO24 | DBVDD3 | DGND |
| K4 | AIF3LRCLK/GPIO26 | DBVDD3 | DGND |
| N4 | AIF3RXDAT/GPIO25 | DBVDD3 | DGND |
| L4 | AIF3TXDAT/GPIO23 | DBVDD3 | DGND |
| M7 | AIF4BCLK/GPIO28 | DBVDD3 | DGND |
| N8 | AIF4LRCLK/GPIO30 | DBVDD3 | DGND |
| P8 | AIF4RXDAT/GPIO29 | DBVDD3 | DGND |
| L7 | AIF4TXDAT/GPIO27 | DBVDD3 | DGND |
| J12 | CIF1MISO | DBVDD1 | DGND |
| J11 | CIF1MOSI | DBVDD1 | DGND |
| J13 | CIF1SCLK | DBVDD1 | DGND |
| H11 | $\overline{\text{CIF1SS}}$ | DBVDD1 | DGND |
| K11 | CIF2SCLK | DBVDD1 | DGND |
| K13 | CIF2SDA | DBVDD1 | DGND |
| L6 | CIF3MISO | DBVDD3 | DGND |
| N6 | CIF3MOSI | DBVDD3 | DGND |
| N7 | CIF3SCLK | DBVDD3 | DGND |
| M6 | $\overline{\text{CIF3SS}}$ | DBVDD3 | DGND |
| N10 | DMICCLK4/GPIO31 | DBVDD4 | DGND |
| N9 | DMICCLK5/GPIO33 | DBVDD4 | DGND |
| L8 | DMICCLK6/GPIO35 | DBVDD4 | DGND |
| M9 | DMICDAT4/GPIO32 | DBVDD4 | DGND |
| P9 | DMICDAT5/GPIO34 | DBVDD4 | DGND |
| M8 | DMICDAT6/GPIO36 | DBVDD4 | DGND |
| G9 | GPIO1 | DBVDD1 | DGND |
| J7 | GPIO2 | DBVDD2 | DGND |
| J4 | GPIO3 | DBVDD3 | DGND |
| H7 | GPIO4 | DBVDD3 | DGND |
| G6 | GPIO5 | DBVDD3 | DGND |
| N5 | GPIO6 | DBVDD3 | DGND |
| H6 | GPIO7 | DBVDD3 | DGND |
| J6 | GPIO8 | DBVDD3 | DGND |
| B13 | HPDETL | AVDD | AGND |
| B12 | HPDETR | AVDD | AGND |
| C4 | IN1ALN/<br>DMICCLK1 | MICVDD (analogue) /<br>MICVDD, MICBIAS1, MICBIAS2, MICBIAS3 (digital)<br>The DMICCLK1 power domain is selectable using IN1_DMIC_SUP | AGND |
| C3 | IN1ALP | MICVDD | AGND |
| E1 | IN1BN | MICVDD | AGND |
| E2 | IN1BP | MICVDD | AGND |
| B1 | IN1RN/<br>DMICDAT1 | MICVDD (analogue) /<br>MICVDD, MICBIAS1, MICBIAS2, MICBIAS3 (digital)<br>The DMICDAT1 power domain is selectable using IN1_DMIC_SUP | AGND |
| B2 | IN1RP | MICVDD | AGND |

## DIGITAL AUDIO INTERFACE - TDM MODE

When TDM operation is used on the AIFnTXDAT pins, it is important that two devices do not attempt to drive the AIFnTXDAT pin simultaneously. To support this requirement, the AIFnTXDAT pins can be configured to be tri-stated when not outputting data.

The timing of the AIFnTXDAT tri-stating at the start and end of the data transmission is described in Figure 7 below.

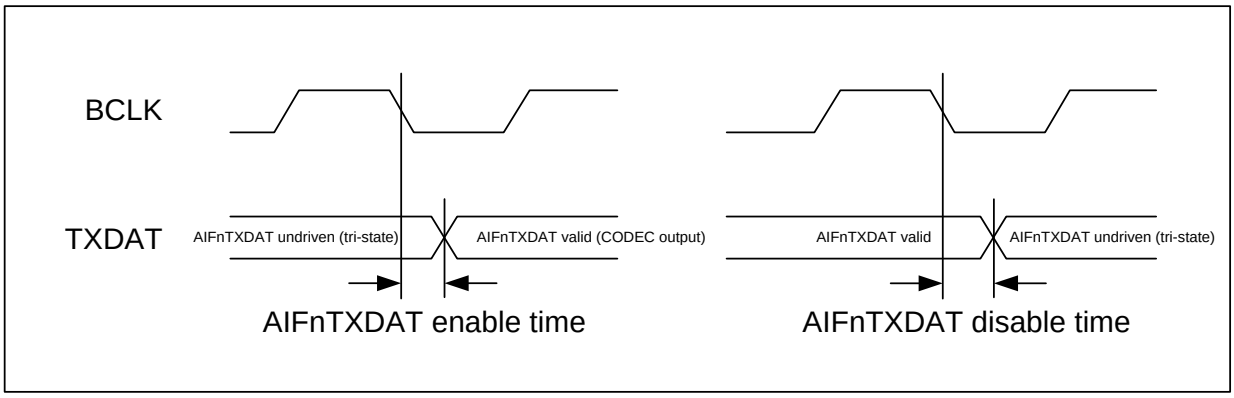


**Figure 7 Audio Interface Timing - TDM Mode**

**Test Conditions**

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

| PARAMETER | MIN | TYP | MAX | UNIT |
|---|---|---|---|---|
| **TDM Timing - Master Mode**<br>$C_{LOAD}$ (AIFnTXDAT) = 15pF to 25pF. BCLK slew (10% to 90%) = 3.7ns to 5.6ns. | | | | |
| AIFnTXDAT enable time from BCLK falling edge | 0 | | | ns |
| AIFnTXDAT disable time from BCLK falling edge | | | 6 | ns |
| **TDM Timing - Slave Mode**<br>$C_{LOAD}$ (AIFnTXDAT) = 15pF). BCLK slew (10% to 90%) = 3ns. | | | | |
| AIFnTXDAT enable time from BCLK falling edge | 2 | | | ns |
| AIFnTXDAT disable time from BCLK falling edge | | | 12.2 | ns |
| **TDM Timing - Slave Mode**<br>$C_{LOAD}$ (AIFnTXDAT) = 25pF). BCLK slew (10% to 90%) = 6ns | | | | |
| AIFnTXDAT enable time from BCLK falling edge | 2 | | | ns |
| AIFnTXDAT disable time from BCLK falling edge | | | 14.2 | ns |

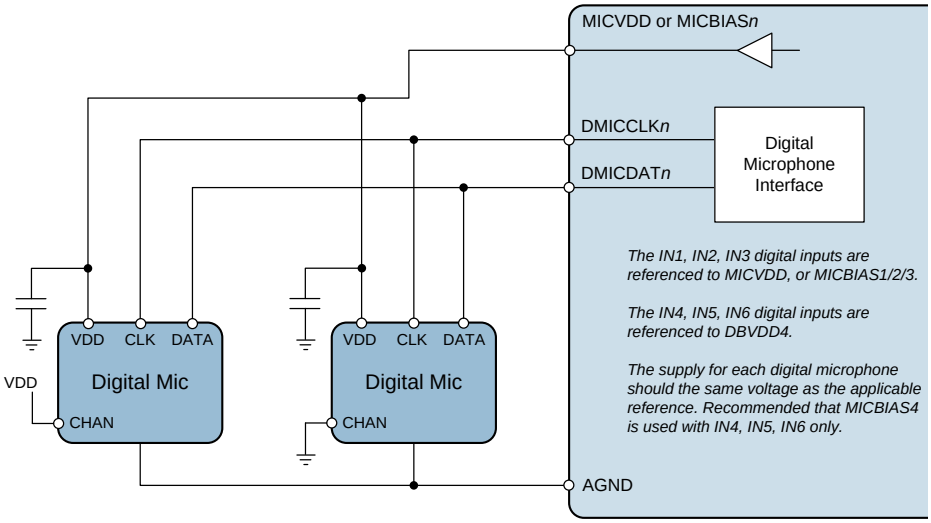


**Figure 20 Digital Microphone Input**

Two digital microphone channels are interleaved on DMICDAT*n*. The digital microphone interface timing is illustrated in Figure 21. Each microphone must tri-state its data output when the other microphone is transmitting.

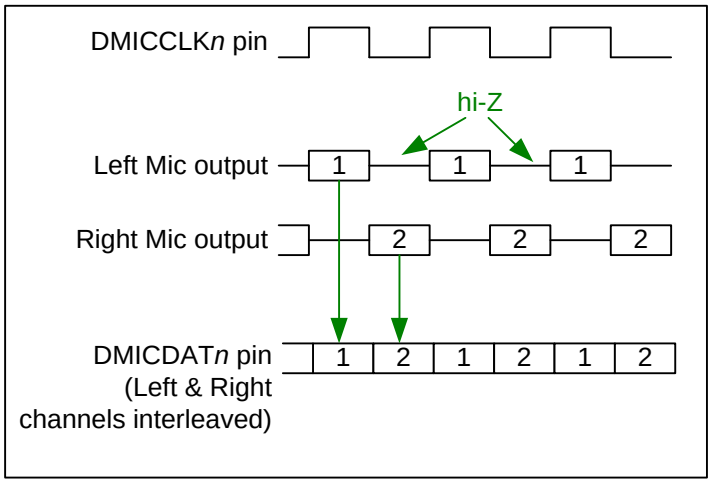


**Figure 21 Digital Microphone Interface Timing**

When digital microphone input is enabled, the CS47L85 outputs a clock signal on the applicable DMICCLK pin(s). The DMICCLK frequency is selectable, as described in Table 1.

Note that SYSCLK must be present and enabled when using the Digital Microphone inputs; see “Clocking and Sample Rates” for details of SYSCLK and the associated register control fields.

## INPUT SIGNAL PATH DIGITAL VOLUME CONTROL

A digital volume control is provided on each of the input signal paths, providing -64dB to +31.5dB gain control in 0.5dB 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 un-mute), the rate is controlled by the IN_VI_RAMP register. For decreasing gain (or mute), the rate is controlled by the IN_VD_RAMP register. Note that the IN_VI_RAMP and IN_VD_RAMP registers 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 set to 0, the digital volume and mute settings will be loaded into the respective control register, but will not actually 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 control registers provide 0.5dB steps, the internal circuits provide signal gain adjustment in 0.125dB steps. This allows a very high degree of gain control, and smooth volume ramping under all operating conditions.

The 0dBFS level of the IN1-IN6 digital input paths is not equal to the 0dBFS level of the CS47L85 digital core. The maximum digital input signal level is -6dBFS (see “Electrical Characteristics”). Under 0dBFS gain conditions, a -6dBFS input signal corresponds to a 0dBFS input to the CS47L85 digital core functions.

The digital volume control register fields are described in Table 4 and Table 5.

| REGISTER ADDRESS | BIT | LABEL | DEFAULT | DESCRIPTION |
|---|---|---|---|---|
| R777 (0309h) Input_Volume_Ramp | 6:4 | IN_VD_RAMP [2:0] | 010 | Input Volume Decreasing Ramp Rate (seconds/6dB)<br>000 = 0ms<br>001 = 0.5ms<br>010 = 1ms<br>011 = 2ms<br>100 = 4ms<br>101 = 8ms<br>110 = 15ms<br>111 = 30ms<br>This register should not be changed while a volume ramp is in progress. |
| | 2:0 | IN_VI_RAMP [2:0] | 010 | Input Volume Increasing Ramp Rate (seconds/6dB)<br>000 = 0ms<br>001 = 0.5ms<br>010 = 1ms<br>011 = 2ms<br>100 = 4ms<br>101 = 8ms<br>110 = 15ms<br>111 = 30ms<br>This register should not be changed while a volume ramp is in progress. |
| R785 (0311h) ADC_Digital_Volume_1L | 9 | IN_VU | | Input Signal Paths Volume and Mute Update<br>Writing a 1 to this bit will cause the Input Signal Paths Volume and Mute settings to be updated simultaneously |
| | 8 | IN1L_MUTE | 1 | Input Path 1 (Left) Digital Mute<br>0 = Un-mute<br>1 = Mute |

| REGISTER ADDRESS | BIT | LABEL | DEFAULT | DESCRIPTION |
|---|---|---|---|---|
| | 7:0 | IN1L_VOL [7:0] | 80h | Input Path 1 (Left) Digital Volume<br>-64dB to +31.5dB in 0.5dB steps<br>00h = -64dB<br>01h = -63.5dB<br>… (0.5dB steps)<br>80h = 0dB<br>… (0.5dB steps)<br>BFh = +31.5dB<br>C0h to FFh = Reserved<br>(See Table 5 for volume range) |
| R789 (0315h) ADC_Digital_Volume_1R | 9 | IN_VU | | Input Signal Paths Volume and Mute Update<br>Writing a 1 to this bit will cause the Input Signal Paths Volume and Mute settings to be updated simultaneously |
| | 8 | IN1R_MUTE | 1 | Input Path 1 (Right) Digital Mute<br>0 = Un-mute<br>1 = Mute |
| | 7:0 | IN1R_VOL [7:0] | 80h | Input Path 1 (Right) Digital Volume<br>-64dB to +31.5dB in 0.5dB steps<br>00h = -64dB<br>01h = -63.5dB<br>… (0.5dB steps)<br>80h = 0dB<br>… (0.5dB steps)<br>BFh = +31.5dB<br>C0h to FFh = Reserved<br>(See Table 5 for volume range) |
| R793 (0319h) ADC_Digital_Volume_2L | 9 | IN_VU | | Input Signal Paths Volume and Mute Update<br>Writing a 1 to this bit will cause the Input Signal Paths Volume and Mute settings to be updated simultaneously |
| | 8 | IN2L_MUTE | 1 | Input Path 2 (Left) Digital Mute<br>0 = Un-mute<br>1 = Mute |
| | 7:0 | IN2L_VOL [7:0] | 80h | Input Path 2 (Left) Digital Volume<br>-64dB to +31.5dB in 0.5dB steps<br>00h = -64dB<br>01h = -63.5dB<br>… (0.5dB steps)<br>80h = 0dB<br>… (0.5dB steps)<br>BFh = +31.5dB<br>C0h to FFh = Reserved<br>(See Table 5 for volume range) |
| R797 (031Dh) ADC_Digital_Volume_2R | 9 | IN_VU | | Input Signal Paths Volume and Mute Update<br>Writing a 1 to this bit will cause the Input Signal Paths Volume and Mute settings to be updated simultaneously |
| | 8 | IN2R_MUTE | 1 | Input Path 2 (Right) Digital Mute<br>0 = Un-mute<br>1 = Mute |

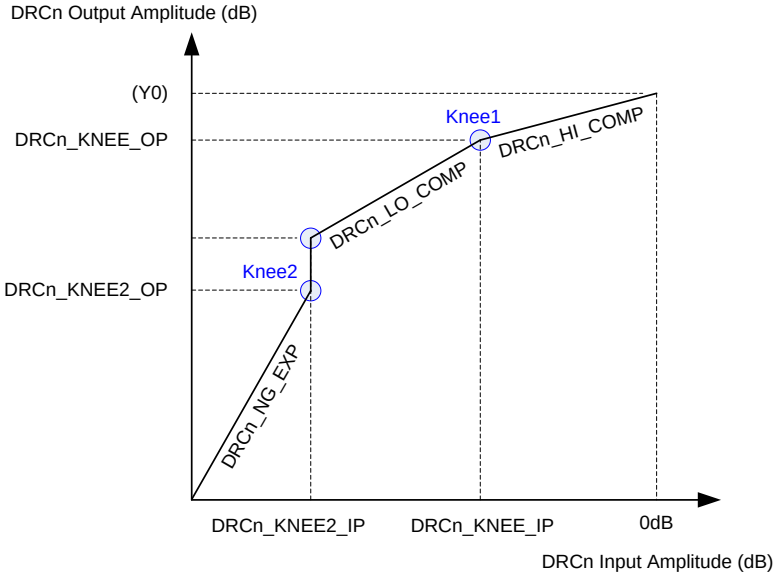


**Figure 29 DRC Response Characteristic**

The slope of the DRC response is determined by register fields DRCn_HI_COMP and DRCn_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 the DRCn_NG_EXP register. 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 DRCn_KNEE2_OP knee is enabled ("Knee2" in Figure 29), this introduces the vertical line in the response pattern illustrated, resulting in infinitely steep attenuation at this point in the response.

The DRC parameters are listed in Table 13.

| REF | PARAMETER | DESCRIPTION |
|---|---|---|
| 1 | DRCn_KNEE_IP | Input level at Knee1 (dB) |
| 2 | DRCn_KNEE_OP | Output level at Knee2 (dB) |
| 3 | DRCn_HI_COMP | Compression ratio above Knee1 |
| 4 | DRCn_LO_COMP | Compression ratio below Knee1 |
| 5 | DRCn_KNEE2_IP | Input level at Knee2 (dB) |
| 6 | DRCn_NG_EXP | Expansion ratio below Knee2 |
| 7 | DRCn_KNEE2_OP | Output level at Knee2 (dB) |

**Table 13 DRC Response Parameters**

The noise gate is enabled when the DRCn_NG_ENA register is set. When the noise gate is not enabled, parameters 5, 6, 7 above are ignored, and the DRCn_LO_COMP slope applies to all input signal levels below Knee1.

The DRCn_KNEE2_OP knee is enabled when the DRCn_KNEE2_OP_ENA register is set. When this bit is not set, then parameter 7 above is ignored, and the Knee2 position always coincides with the low end of the DRCn_LO_COMP region.

The "Knee1" point in Figure 29 is determined by register fields DRCn_KNEE_IP and DRCn_KNEE_OP.

Parameter Y0, the output level for a 0dB input, is not specified directly, but can be calculated from the other parameters, using the equation:

Y0 = DRCn_KNEE_OP - (DRCn_KNEE_IP x DRCn_HI_COMP)

## ASYNCHRONOUS SAMPLE RATE CONVERTER (ASRC)

The CS47L85 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 “Clocking and Sample Rates”. Every digital signal path must be synchronised either to SYSCLK or to ASYNCCLK.

There are two Asynchronous Sample Rate Converters (ASRCs). Each ASRC provides two stereo signal paths between the SYSCLK and ASYNCCLK domains, as illustrated in Figure 38.

The sample rate on the SYSCLK domain is selected using the ASRCn_RATE1 registers - the rate can be set equal to SAMPLE_RATE_1, SAMPLE_RATE_2 or SAMPLE_RATE_3.

The sample rate on the ASYNCCLK domain is selected using the ASRCn_RATE2 registers - the rate can be set equal to ASYNC_SAMPLE_RATE_1 or ASYNC_SAMPLE_RATE_2.

See “Clocking and Sample Rates” for details of the sample rate control registers.

The ASRCn_RATE1 and ASRCn_RATE2 registers should not be changed if any of the respective *_SRC*n* registers is non-zero. The associated *_SRC*n* registers should be cleared to 00h before writing new values to ASRCn_RATE1 or ASRCn_RATE2. A minimum delay of 125us should be allowed between clearing the *_SRC*n* registers and writing to the associated ASRCn_RATE1 or ASRCn_RATE2 registers. See Table 23 for further details.

Each ASRC supports sample rates in the range 8kHz to 192kHz. For each ASRC, the ratio of the applicable SAMPLE_RATE_n and ASYNC_SAMPLE_RATE_n registers must not exceed 6.

The ASRC*n* IN1 (Left and Right) paths convert from the SYSCLK domain to the ASYNCCLK domain. These paths are enabled using the ASRC*n*_IN1L_ENA and ASRC*n*_IN1R_ENA register bits respectively.

The ASRC*n* IN2 (Left and Right) paths convert from the ASYNCCLK domain to the SYSCLK domain. These paths are enabled using the ASRC*n*_IN2L_ENA and ASRC*n*_IN2R_ENA register bits respectively.

Synchronisation (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 “Interrupts”.

The ASRC Lock status of each ASRC path can be output directly on a GPIO pin as an external indication of ASRC Lock. See “General Purpose Input / Output” to configure a GPIO pin for this function.

The CS47L85 performs automatic checks to confirm that the SYSCLK or ASYNCCLK frequency is high enough to support the commanded ASRC and digital mixing functions. If an attempt is made to enable an ASRC signal path, and there are insufficient SYSCLK or ASYNCLK cycles to support it, then the attempt will be unsuccessful. (Note that any signal paths that are already active will not be affected under these circumstances.)

The status bits in Register R3809 indicate the status of each of the ASRC signal paths. If an Underclocked Error condition occurs, then these bits provide readback of which ASRC signal path(s) have been successfully enabled.

The status bits in Registers R1600 to R3192 indicate the status of each of the digital mixers. If an Underclocked Error condition occurs, then these bits provide readback of which mixer(s) have been successfully enabled.

The Asynchronous Sample Rate Converter (ASRC) signal paths and control registers are illustrated in Figure 38.

## DSP FIRMWARE CONTROL

The CS47L85 digital core incorporates seven DSP processing blocks, capable of running a wide range of audio enhancement functions. Different firmware configurations can be loaded onto each DSP, enabling the CS47L85 to be highly customised for specific application requirements. Full read/write access to the device register map is supported from each DSP core, including access to the firmware registers of the other DSPs. Synchronisation of different DSPs is supported, and shared data memory space is provided for the DSP2 and DSP3 blocks; these features enable enhanced processing capabilities for the associated DSPs.

Examples of the DSP functions include Virtual Surround Sound (VSS), Multiband Compressor (MBC), and the Cirrus Logic SoundClear® suite of audio processing algorithms. Note that it is possible to implement more than one type of audio enhancement function on a single DSP; the precise combination(s) of functions will vary from one firmware configuration to another.

The DSP blocks each employ the same internal architecture, and provide an equivalent processing capability. Note that the DSPs differ in terms of the firmware memory sizes associated with each. DSPs 1 to 5 can be clocked at up to 150MHz, corresponding to 150 MIPS each. DSP6 and DSP7 are designed for low power operation, clocked at up to 75MHz each. The DSP6 core is optimised for always-on (voice trigger) software functions.

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

In order to use the DSP blocks, the required firmware configuration must first be loaded onto the device by writing the appropriate files to the CS47L85 register map. The firmware configuration will comprise Program, Coefficient and Data content. In some cases, the Coefficient content must be derived using tools provided in Cirrus Logic's WISCE evaluation board control software.

Details of how to load the firmware configuration onto the CS47L85 are described below. Note that the WISCE evaluation board control software provides support for easy loading of Program, Coefficient and Data content onto the CS47L85. Please contact your local 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 register control fields.

The audio signal paths connecting to/from the DSP processing blocks are configured as described in the "Digital Core" section. Note that the DSP firmware must be loaded and enabled before audio signal paths can be enabled.

### DSP FIRMWARE MEMORY AND REGISTER MAPPING

The DSP firmware memory is programmed by writing to the registers referenced in Table 25. Note that clocking is not required for access to the firmware registers by the host processor.

The CS47L85 Program, Coefficient and Data register memory space is described in Table 25. The full register map listing is provided in a separate document - see "Register Map" for further information. The shared DSP2/DSP3 memory space is implemented at two different register address locations; reading or writing at either address will access the same memory data.

If multiple DSPs write to a shared memory address at the same time, then the address at which the collision occurred will be reported in the DSP3_DUALMEM_COLLISION_ADDR register. Note that this field is coded in 24-bit DSP data word units, and is defined relative to the base address of the applicable shared memory area.

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

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

## DSP DIRECT MEMORY ACCESS (DMA) CONTROL

Each DSP provides a multi-channel DMA function; this is configured using the registers described in Table 28.

There are 8 WDMA (DSP input) and 6 RDMA (DSP output) channels for each DSP; these are enabled using the DSPn_WDMA_CHANNEL_ENABLE and DSPn_RDMA_CHANNEL_ENABLE fields. The status of each WDMA channel is indicated in DSPn_WDMA_ACTIVE_CHANNELS.

The DMA can access the X data memory or Y data memory associated with the respective DSP. The applicable memory is selected using bit [15] of the respective *_START_ADDRESS register.

The start address of each DMA channel is configured as described in Table 28. 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 DSPn_DMA_BUFFER_LENGTH field. The selected buffer length applies to all enabled DMA channels.

Note that the start address registers, and buffer length registers, are defined in 24-bit DSP data word units. This means that the LSB of these fields represents one 24-bit DSP memory word. (Note that this differs from the CS47L85 register map layout, as described in Table 25).

The parameters of a DMA channel (i.e., Start Address or Offset Address) must not be changed whilst 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' respectively, 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 DSPn_PING_FULL bit will be asserted (set to '1'), and a 'DSP Start' signal will be generated. The 'Start' signal from the DMA is typically used to start Firmware execution, as noted in Table 27. Meanwhile, further DSP input data will be filling up the 'pong' buffer.

When the 'pong' input buffer is full, the DSPn_PONG_FULL bit will be asserted, and another 'DSP Start' signal will be generated. The DSP Firmware must take care to read the input data from the applicable buffer, in accordance with the DSPn_PING_FULL and DSPn_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 as the input 'pong' buffers are filled.

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

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

The DSPCLK system clock must be configured and enabled before any DMA channel is enabled. The DMA channels should be kept disabled (DSPn_[WDMA/RDMA]_CHANNEL_ENABLE=00h) if DSPCLK is not enabled. See "Clocking and Sample Rates" for details of the system clocks (including requirements for reconfiguring DSPCLK while DMA channels are enabled).

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

| REGISTER ADDRESS | BIT | LABEL | DEFAULT | DESCRIPTION |
|---|---|---|---|---|
| DSP1 Base Address = R1048064 (0FFE00h)<br>DSP2 Base Address = R1572352 (17FE00h)<br>DSP3 Base Address = R2096640 (1FFE00h)<br>DSP4 Base Address = R2620928 (27FE00h)<br>DSP5 Base Address = R3145216 (2FFE00h)<br>DSP6 Base Address = R3669504 (37FE00h)<br>DSP7 Base Address = R4193792 (3FFE00h) | | | | |
| base address +04h<br>DSPn_Status_1 | 31 | DSPn_PING_FULL | 0 | DSPn WDMA Ping Buffer Status<br>0 = Not Full<br>1 = Full |
| | 30 | DSPn_PONG_FULL | 0 | DSPn WDMA Pong Buffer Status<br>0 = Not Full<br>1 = Full |

| REGISTER ADDRESS | BIT | LABEL | DEFAULT | DESCRIPTION |
|---|---|---|---|---|
| base address +008h<br>MIFn_I2C_CONFIG_5 | 0 | MIFn_WDT_ENA | 0 | Watchdog Timer (WDT) control<br>0 = Disabled<br>1 = Enabled<br>When bus monitoring functions are enabled (MIFn_SCL_MON_ENA=1), the Watchdog Timer is used to detect the SCLK line being pulled low for a prolonged duration. |
| base address +080h<br>MIFn_I2C_STATUS_1 | 2 | MIFn_WDT_TIMEOUT_STS | 0 | Watchdog Timer (WDT) Error Status<br>This bit, when set, indicates that the WDT expired during the I2C transaction.<br>This bit is latched when set; it is cleared by writing '1'.<br>This bit is automatically cleared on next I2C transaction. |
| | 0 | MIFn_NACK_STS | 0 | NACK Error Status<br>This bit, when set, indicates that a NACK Error signal was received during the I2C transaction.<br>This bit is latched when set; it is cleared by writing '1'.<br>This bit is automatically cleared on next I2C transaction. |
| base address +100h<br>MIFn_CONFIG_1 | 0 | MIFn_START | 0 | Starts the I2C transaction<br>Write '1' to start. |
| base address +104h<br>MIFn_CONFIG_3 | 17:16 | MIFn_WORD_SIZE [1:0] | 00 | Selects the data word format.<br>I2C transactions are made up of 1-Byte data words; the sequence order of these words differs according to the applicable word format.<br>Correct setting of the MIFn_WORD_SIZE field ensures that each data word is transmitted/received as Most-Significant-Byte first.<br><br>00 = 8-bit (1, 2, 3, 4, 5, 6, 7, 8, etc)<br>01 = 16-bit (2, 1, 4, 3, 6, 5, 8, 7, etc)<br>10 = 32-bit (4, 3, 2, 1, 8, 7, 6, 5, etc)<br><br>The numbers in brackets describe the order in which the applicable MIFn_[TX\|RX]_BYTEx fields are transmitted/received over the I2C interface. |
| | 0 | MIFn_READ_WRITE_SEL | 0 | Selects the I2C Command type<br>0 = Master Write<br>1 = Master Read |
| base address +106h<br>MIFn_CONFIG_4 | 20:0 | MIFn_TX_LENGTH [20:0] | 00_0000h | Selects the total number of data bytes in an I2C Write operation.<br>00_0000h = 1 byte<br>00_0001h = 2 bytes<br>00_0002h = 3 bytes<br>…<br>1F_FFFFh = 2,097,152 bytes |

| REGISTER ADDRESS | BIT | LABEL | DEFAULT | DESCRIPTION |
|---|---|---|---|---|
| READ | 8:0 | EVENTLOGn_FIFO13_ID [8:0] | 000h | Event Log FIFO Index 13 source<br>*Register description is as above.* |
| base address +B6h EVENTLOGn_FIFO13_TIME | 31:0 | EVENTLOGn_FIFO13_TIME [31:0] | 0000 0000h | Event Log FIFO Index 13 Time |
| base address +B8h EVENTLOGn_FIFO14_READ | 12 | EVENTLOGn_FIFO14_POL | 0 | Event Log FIFO Index 14 polarity<br>*Register description is as above.* |
| | 8:0 | EVENTLOGn_FIFO14_ID [8:0] | 000h | Event Log FIFO Index 14 source<br>*Register description is as above.* |
| base address +BAh EVENTLOGn_FIFO14_TIME | 31:0 | EVENTLOGn_FIFO14_TIME [31:0] | 0000 0000h | Event Log FIFO Index 14 Time |
| base address +BCh EVENTLOGn_FIFO15_READ | 12 | EVENTLOGn_FIFO15_POL | 0 | Event Log FIFO Index 15 polarity<br>*Register description is as above.* |
| | 8:0 | EVENTLOGn_FIFO15_ID [8:0] | 000h | Event Log FIFO Index 15 source<br>*Register description is as above.* |
| base address +BEh EVENTLOGn_FIFO15_TIME | 31:0 | EVENTLOGn_FIFO15_TIME [31:0] | 0000 0000h | Event Log FIFO Index 15 Time |

**Table 33 Event Logger (EVENTLOGn) Control**

A list of the valid input sources for the Event Loggers is provided in Table 34.

The “EDGE” type noted is coded as “S” (single edge) or “D” (dual edge). Note that a single-edge input source will only provide valid input to the Event Logger in the default (Rising Edge Triggered) polarity.

It is advised to take care when enabling IRQ1 or IRQ2 as an input source for the Event Loggers; a recursive loop, where the IRQ signal is also an output from the same Event Logger, must be avoided.

| ID | DESCRIPTION | EDGE | ID | DESCRIPTION | EDGE | ID | DESCRIPTION | EDGE |
|---|---|---|---|---|---|---|---|---|
| 3 | irq1 | D | 259 | gpio4 | D | 354 | event3_full | S |
| 4 | irq2 | D | 260 | gpio5 | D | 355 | event4_full | S |
| 9 | sysclk_fail | S | 261 | gpio6 | D | 356 | event5_full | S |
| 24 | fll1_lock | D | 262 | gpio7 | D | 357 | event6_full | S |
| 25 | fll2_lock | D | 263 | gpio8 | D | 358 | event7_full | S |
| 26 | fll3_lock | D | 264 | gpio9 | D | 359 | event8_full | S |
| 32 | frame_start_g1r1 | S | 265 | gpio10 | D | 368 | event1_wmark | S |
| 33 | frame_start_g1r2 | S | 266 | gpio11 | D | 369 | event2_wmark | S |
| 34 | frame_start_g1r3 | S | 267 | gpio12 | D | 370 | event3_wmark | S |
| 40 | frame_start_g2r1_sys | S | 268 | gpio13 | D | 371 | event4_wmark | S |
| 41 | frame_start_g2r2_sys | S | 269 | gpio14 | D | 372 | event5_wmark | S |
| 80 | hpdet | S | 270 | gpio15 | D | 373 | event6_wmark | S |
| 88 | micdet | S | 271 | gpio16 | D | 374 | event7_wmark | S |
| 96 | jd1_rise | S | 272 | gpio17 | D | 375 | event8_wmark | S |
| 97 | jd1_fall | S | 273 | gpio18 | D | 384 | dsp1_dma | S |
| 98 | jd2_rise | S | 274 | gpio19 | D | 385 | dsp2_dma | S |
| 99 | jd2_fall | S | 275 | gpio20 | D | 386 | dsp3_dma | S |
| 100 | micd_clamp_rise | S | 276 | gpio21 | D | 387 | dsp4_dma | S |
| 101 | micd_clamp_fall | S | 277 | gpio22 | D | 388 | dsp5_dma | S |
| 128 | drc1_sig_det | D | 278 | gpio23 | D | 389 | dsp6_dma | S |
| 129 | drc2_sig_det | D | 279 | gpio24 | D | 390 | dsp7_dma | S |

## 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.

### 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 will provide Manager and Framer Device functions; the Manager has the capabilities to administer the bus, whilst 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 will be active at any time. The Framer function can be transferred between Devices when required. Generic Devices provide the basic SLIMbus functionality for the associated Port(s), and for the Transport Protocol by which audio signal paths are established on the bus.

### 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
- 4 Cells make a Slot
- 192 Slots make a Frame
- 8 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 is not fixed; it 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 will typically comprise Control space (for bus messages, synchronisation etc.) and Data space (for audio paths). The precise allocation is configurable, and can be entirely Control space, if required.

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

The SLIMbus Frame always comprises 192 Slots, regardless of the Subframe definition. A number of Slots are allocated to Control space, as noted above; the remaining Slots are allocated to Data space. Some of the Control space is required for Framing Information and for the Guide Channel (described below); the remainder of the Control space are allocated to the Message Channel.

### CONTROL SPACE

Framing Information is provided in Slots 0 and 96 of every Frame. Slot 0 contains a 4-bit synchronisation code; Slot 96 contains the 32-bit Framing Information, transmitted 4 bits at a time over the 8 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 synchronisation. 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 will depend 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.

### EVENT LOGGER FIFO BUFFER STATUS OUTPUT

GP*n*_FN = 150h, 151h, 152h, 153h, 154h, 155h, 156h, 157h.

The Event Loggers are each provided with a 16-stage FIFO buffer, in which any detected events (signal transitions) are recorded. Status outputs for each FIFO buffer are provided. See “DSP Peripheral Control” for details of the Event Loggers.

A logic signal from the Event Loggers may be output directly on a GPIO pin by setting the respective GPIO registers as described in “GPIO Control”. This logic signal is set high whenever the ‘FIFO Not Empty’ condition is true.

The Event Loggers also provide inputs to the Interrupt control circuit. An interrupt event is triggered whenever the respective FIFO condition occurs. The associated interrupt bits are latched once set; they can be polled at any time or used to control the IRQ signal. See “Interrupts” for more details of the Interrupt event handling.

### GENERAL PURPOSE SWITCH

The CS47L85 provides a General Purpose Switch, which can be used as a controllable analogue switch for external functions. The switch is implemented between the GPSWP and GPSWN pins. Note that this feature is entirely independent to the GPIOn pins.

The General Purpose Switch is configured using SW1_MODE. This register allows the switch to be disabled, enabled, or synchronised to the MICDET Clamp status, as described in Table 99.

The switch is a bi-directional analogue switch, offering flexibility in the potential circuit applications. Refer to the “Absolute Maximum Ratings” and “Electrical Characteristics” for further details.

The switch can be used in conjunction with the MICDET Clamp function, in order suppress pops and clicks associated with jack insertion and removal. An example circuit is shown in Figure 69, within the “External Accessory Detection” section. Note that the MICDET Clamp function must also be configured appropriately when using this method of pop suppression control.

| REGISTER ADDRESS | BIT | LABEL | DEFAULT | DESCRIPTION |
|---|---|---|---|---|
| R712 (02C8h) GP_Switch_1 | 1:0 | SW1_MODE [1:0] | 00 | General Purpose Switch control<br>00 = Disabled (open)<br>01 = Enabled (closed)<br>10 = Enabled when MICDET Clamp is active<br>11 = Enabled when MICDET Clamp is not active |

**Table 99 General Purpose Switch control**

## 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 100 and Table 101 for a full definition of the Interrupt Controller inputs.) Any combination of these inputs can be used to trigger an Interrupt Request (IRQ) event.

The Interrupt Controller supports two sets of interrupt registers. This allows two separate Interrupt Request (IRQ) outputs to be generated, and for each IRQ to report a different set of input or status conditions.

For each Interrupt Request (IRQ1 and IRQ2) output, there is an Interrupt register field associated with each of the interrupt inputs. These fields are asserted whenever a logic edge is detected on the respective input. Some inputs are triggered on rising edges only; some are triggered on both edges. Separate rising and falling interrupt registers are provided for the JD1 and JD2 signals. The Interrupt register fields for IRQ1 are described in Table 100. The Interrupt register fields for IRQ2 are described in Table 101. The Interrupt flags can be polled at any time, or else in response to the Interrupt Request (IRQ) output being signalled via the $\overline{\text{IRQ}}$ pin or a GPIO pin.

All of the Interrupts are edge-triggered, as noted above. Many of these are triggered on both the rising and falling edges and, therefore, the Interrupt registers cannot indicate which edge has been detected. The “Raw Status” fields described in Table 100 and Table 101 provide readback of the current value of the corresponding inputs to the Interrupt Controller. Note that the “Raw Status” bits associated with IRQ1 and IRQ2 both provide the same readback information. The status of any GPIO (or DSP GPIO) inputs can also be read using the GPIO (or DSP GPIO) control registers, as described in Table 94 and Table 36.

Individual mask bits can enable or disable different functions from the Interrupt controller. The mask bits are described in Table 100 (for IRQ1) and Table 101 (for IRQ2). Note that a masked interrupt input will not assert the corresponding interrupt register field, and will not cause the associated Interrupt Request (IRQ) output to be asserted.

The Interrupt Request (IRQ) outputs represent the logical ‘OR’ of the associated interrupt registers. (IRQ1 is derived from the _EINT1 registers; IRQ2 is derived from the _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 register bit(s). The Interrupt Request (IRQ) outputs are not reset until each of the associated interrupts has been reset.

A de-bounce circuit can be enabled on any GPIO input, to avoid false event triggers. This is enabled on each pin using the register bits described in Table 94. The GPIO de-bounce circuit uses the 32kHz clock, which must be enabled whenever the GPIO de-bounce function is required.

A de-bounce circuit is always enabled on the FLL status inputs; either the 32kHz 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 100 and Table 101), are valid without clocking, and can be used to provide FLL status readback when system clocks are not available.

The IRQ outputs can be globally masked using the IM_IRQ1 and IM_IRQ2 register bits. When not masked, the IRQ status can be read from IRQ1_STS and IRQ2_STS for the respective IRQ outputs.

The IRQ1 output is provided externally on the $\overline{\text{IRQ}}$ pin. Under default conditions, this output is ‘Active Low’. The polarity can be inverted using the IRQ_POL register. The $\overline{\text{IRQ}}$ output can be either CMOS driven or Open Drain; this is selected using the IRQ_OP_CFG register. Note that the $\overline{\text{IRQ}}$ output is referenced to the DBVDD1 power domain.

The IRQ2 status can be used to trigger DSP firmware execution - see “DSP Firmware Control”. 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 “General Purpose Input / Output”.

The CS47L85 Interrupt Controller circuit is illustrated in Figure 73. (Note that not all interrupt inputs are shown.) The control fields associated with IRQ1 and IRQ2 are described in Table 100 and Table 101 respectively. The global interrupt mask bits, status bits, and output configuration register are described in Table 102.

Note that, under default register conditions, the ‘Boot Done’ status is the only un-masked interrupt source; a falling edge on the $\overline{\text{IRQ}}$ pin will indicate completion of the Boot Sequence.

| REGISTER ADDRESS | BIT | LABEL | DEFAULT | DESCRIPTION |
|---|---|---|---|---|
| | 3:0 | ASYNC_CLK_SRC [3:0] | 0101 | ASYNCCLK Source<br>0000 = MCLK1<br>0001 = MCLK2<br>0010 = Reserved<br>0011 = Reserved<br>0100 = FLL1<br>0101 = FLL2<br>0110 = FLL3<br>0111 = Reserved<br>1000 = AIF1BCLK<br>1001 = AIF2BCLK<br>1010 = AIF3BCLK<br>1011 = AIF4BCLK<br>All other codes are Reserved |
| R275 (0113h) Async_sample_rate_1 | 4:0 | ASYNC_SAMPLE_RATE_1 [4:0] | 10001 | ASYNC Sample Rate 1 Select<br>00h = None<br>01h = 12kHz<br>02h = 24kHz<br>03h = 48kHz<br>04h = 96kHz<br>05h = 192kHz<br>09h = 11.025kHz<br>0Ah = 22.05kHz<br>0Bh = 44.1kHz<br>0Ch = 88.2kHz<br>0Dh = 176.4kHz<br>11h = 8kHz<br>12h = 16kHz<br>13h = 32kHz<br>All other codes are Reserved |
| R276 (0114h) Async_sample_rate_2 | 4:0 | ASYNC_SAMPLE_RATE_2 [4:0] | 10001 | ASYNC Sample Rate 2 Select<br>*Register coding is same as ASYNC_SAMPLE_RATE_1.* |
| R283 (011Bh) Async_sample_rate_1_status | 4:0 | ASYNC_SAMPLE_RATE_1_STS [4:0] | 00000 | ASYNC Sample Rate 1 Status<br>(Read only)<br>*Register coding is same as ASYNC_SAMPLE_RATE_1.* |
| R284 (011Ch) Async_sample_rate_2_status | 4:0 | ASYNC_SAMPLE_RATE_2_STS [4:0] | 00000 | ASYNC Sample Rate 2 Status<br>(Read only)<br>*Register coding is same as ASYNC_SAMPLE_RATE_1.* |
| R288 (0120h) DSP_Clock_1 | 10:8 | DSP_CLK_FREQ_RANGE [2:0] | 000 | DSPCLK Frequency<br>000=5.5MHz to 9.375MHz (9.216MHz)<br>001=9.375MHz to 18.75MHz (18.432MHz)<br>010=18.75MHz to 37.5MHz (36.864MHz)<br>011=37.5MHz to 75MHz (73.728MHz)<br>100=75MHz to 150MHz (147.456MHz)<br>All other codes are Reserved<br>The frequencies in brackets are the nominal (or typical) frequencies for each setting.<br>If the DSPCLK frequency is equal to one of the threshold frequencies quoted (e.g., 37.5MHz), then the higher range setting (e.g., 011) should be selected. |

| REGISTER ADDRESS | BIT | LABEL | DEFAULT | DESCRIPTION |
|---|---|---|---|---|
| R407 (0197h) FLL2_Loop_Filter_Test_1 | 15 | FLL2_FRC_INTEG_UPD | 0 | Write '1' to apply the FLL2_FRC_INTEG_VAL setting. (Only valid when FLL2_FREERUN=1) |
| | 11:0 | FLL2_FRC_INTEG_VAL [11:0] | 000h | FLL2 Forced Integrator Value |
| R408 (0198h) FLL2_NCO_Test_0 | 15 | FLL2_INTEG_VALID | 0 | FLL2 Integrator Valid<br>Indicates if the FLL2_INTEG register is valid<br>0 = Not valid<br>1 = Valid |
| | 11:0 | FLL2_INTEG [11:0] | 000h | FLL2 Integrator Value<br>(Read-only)<br>Indicates the current FLL2 integrator setting. Only valid when FLL2_INTEG_VALID = 1. |
| R433 (01B1h) FLL3_Control_1 | 1 | FLL3_FREERUN | 1 | FLL3 Free-Running Mode Enable<br>0 = Disabled<br>1 = Enabled<br>The FLL feedback mechanism is halted in Free-Running mode, and the latest integrator setting is maintained |
| R439 (01B7h) FLL3_Loop_Filter_Test_1 | 15 | FLL3_FRC_INTEG_UPD | 0 | Write '1' to apply the FLL3_FRC_INTEG_VAL setting. (Only valid when FLL3_FREERUN=1) |
| | 11:0 | FLL3_FRC_INTEG_VAL [11:0] | 000h | FLL3 Forced Integrator Value |
| R440 (01B8h) FLL3_NCO_Test_0 | 15 | FLL3_INTEG_VALID | 0 | FLL3 Integrator Valid<br>Indicates if the FLL3_INTEG register is valid<br>0 = Not valid<br>1 = Valid |
| | 11:0 | FLL3_INTEG [11:0] | 000h | FLL3 Integrator Value<br>(Read-only)<br>Indicates the current FLL3 integrator setting. Only valid when FLL3_INTEG_VALID = 1. |

**Table 112 Free-Running FLL Mode Control**

### SPREAD SPECTRUM FLL CONTROL

The CS47L85 can apply modulation to the FLL outputs, using spread spectrum techniques. This can be used to control the EMI characteristics of the circuits that are clocked via the FLLs.

Each of the FLLs can be individually configured for Triangle modulation, Zero Mean Frequency Modulation (ZMFM) or Dither. The amplitude and frequency parameters of the spread spectrum functions is also programmable, using the registers described in Table 113.

| REGISTER ADDRESS | BIT | LABEL | DEFAULT | DESCRIPTION |
|---|---|---|---|---|
| R393 (0189h) FLL1_Spread_Spectrum | 5:4 | FLL1_SS_AMPL [1:0] | 00 | FLL1 Spread Spectrum Amplitude<br>Controls the extent of the spread-spectrum modulation.<br>00 = 0.7% (triangle), 0.7% (ZMFM, dither)<br>01 = 1.1% (triangle), 1.3% (ZMFM, dither)<br>10 = 2.3% (triangle), 2.6% (ZMFM, dither)<br>11 = 4.6% (triangle), 5.2% (ZMFM, dither) |
| | 3:2 | FLL1_SS_FREQ [1:0] | 00 | FLL1 Spread Spectrum Frequency<br>Controls the spread spectrum modulation frequency in Triangle mode.<br>00 = 439kHz<br>01 = 878kHz<br>10 = 1.17MHz<br>11 = 1.76MHz |

## POWER SUPPLY / REFERENCE DECOUPLING

Electrical coupling exists particularly in digital logic systems where switching in one sub-system 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 behaviour in other components. A decoupling ('bypass') capacitor can be used as an energy storage component which will provide 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 CS47L85, 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.

The recommended power supply decoupling capacitors for CS47L85 are detailed below in Table 136.

| POWER SUPPLY | DECOUPLING CAPACITOR |
|---|---|
| LDOVDD, DBVDD1, DBVDD2, DBVDD3 | 0.1µF ceramic (see Note) |
| AVDD1, AVDD2 | 1.0µF ceramic |
| CPVDD | 4.7µF ceramic |
| CPVDD2 | 4.7µF ceramic |
| MICVDD | 4.7µF ceramic |
| DCVDD, FLLVDD | 4 x 1.0µF ceramic - one close to each pin.<br>Alternatively, a single 4.7µF ceramic. |
| SPKVDDL, SPKVDDR | 4.7µF ceramic |
| VREFC | 2.2µF ceramic |

**Table 136 Power Supply Decoupling Capacitors**

Note: 0.1µF is required with 4.7µF a guide to the total required power rail capacitance.

All decoupling capacitors should be placed as close as possible to the CS47L85 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 CS47L85.

Due to the wide tolerance of many types of ceramic capacitors, care must be taken to ensure that the selected components provide the required capacitance across the required temperature and voltage ranges in the intended application. For most application the use of ceramic capacitors with capacitor dielectric X5R is recommended.

## CHARGE PUMP COMPONENTS

The CS47L85 incorporates two Charge Pump circuits, identified as CP1 and CP2.

CP1 generates the CP1VOUTP and CP1VOUTN 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 CS47L85 are detailed below in Table 137.

| DESCRIPTION | CAPACITOR |
|---|---|
| CP1VOUT1P decoupling | Required capacitance is 2.0µF at 2V.<br>Suitable component typically 4.7µF. |
| CP1VOUT1N decoupling | Required capacitance is 2.0µF at 2V.<br>Suitable component typically 4.7µF. |
| CP1 fly-back 1<br>(connect between CP1C1A and CP1C1B) | Required capacitance is 1.0µF at 2V.<br>Suitable component typically 2.2µF. |
| CP1VOUT2P decoupling | Required capacitance is 2.0µF at 2V.<br>Suitable component typically 4.7µF. |
| CP1VOUT2N decoupling | Required capacitance is 2.0µF at 2V.<br>Suitable component typically 4.7µF. |
| CP1 fly-back 2<br>(connect between CP1C2A and CP1C2B) | Required capacitance is 1.0µF at 2V.<br>Suitable component typically 2.2µF. |
| CP2VOUT decoupling | Required capacitance is 1.0µF at 3.6V.<br>Suitable component typically 4.7µF. |
| CP2 fly-back<br>(connect between CP2CA and CP2CB) | Required capacitance is 220nF at 2V.<br>Suitable component typically 470nF. |

**Table 137 Charge Pump External Capacitors**

Ceramic capacitors are recommended for these Charge Pump requirements. Note that, due to the wide tolerance of many types of ceramic capacitors, care must be taken to ensure that the selected components provide the required capacitance across the required temperature and voltage ranges in the intended application. Ceramic capacitors with X5R dielectric are recommended.

The positioning of the Charge Pump capacitors is important, particularly the fly-back capacitors. These capacitors should be placed as close as possible to the CS47L85. 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.

## EXTERNAL ACCESSORY DETECTION COMPONENTS

The external accessory detection circuit measures jack insertion using the JACKDET1 and JACKDET2 pins. The insertion switch status is detected using an internal pull-up resistor circuit on the respective pin. The logic thresholds associated with the each of the JACKDETn pins are the same, as noted in the “Electrical Characteristics” section. Note that an external resistor (e.g., 500kΩ) can be used to lower the effective jack detection thresholds; this provides support for different jack switch configurations.

Microphone detection and key-button press detection is supported using the MICDETn pins. The applicable pin should be connected to one of the MICBIASn outputs, via a 2.2kΩ bias resistor, as described in the “Microphone Bias Circuit” section. Note that, when using the External Accessory Detection function, the MICBIASn resistor must be 2.2kΩ +/-2%.

A recommended circuit configuration, including headphone output on HPOUT1 and microphone connections, is shown in Figure 97. See “Analogue Input Paths” for details of the DC-blocking microphone input capacitor selection.

The recommended external components and connections for microphone / push-button detection are illustrated in Figure 98.

Note that, when using the Microphone Detect circuit, it is recommended to use the IN1B or IN2B analogue microphone input paths, to ensure best immunity to electrical transients arising from the external accessory.

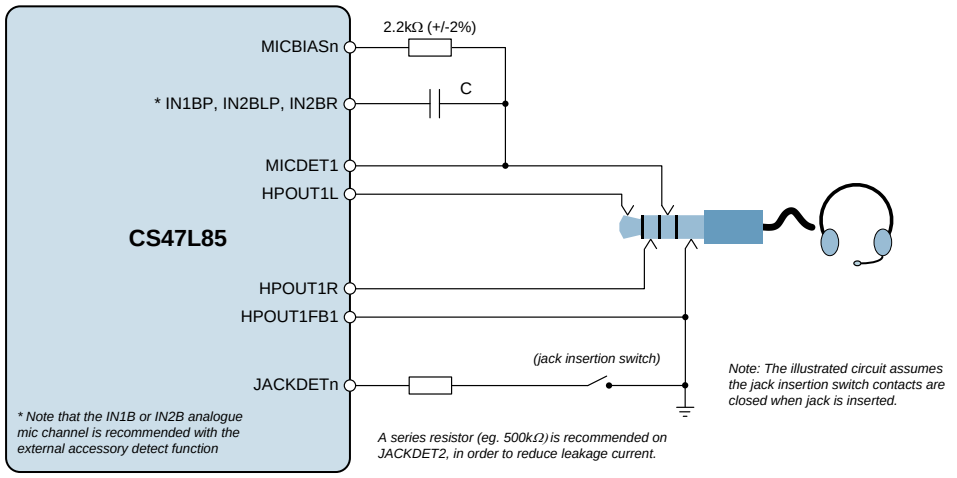


**Figure 97 External Accessory Detection**

The accessory detection circuit measures the impedance of an external load connected to one of the MICDET pins.

The microphone detection circuit uses MICVDD, MICBIAS1, MICBIAS2 or MICBIAS3 as a reference. The applicable source is configured using the MICD_BIAS_SRC register.

The CS47L85 can detect the presence of a typical microphone and up to 6 push-buttons, using the components shown in Figure 98. When the microphone detection circuit is enabled, then each of the push-buttons shown will cause a different bit within the MICD_LVL register to be set.

The microphone detect function is specifically designed to detect a video accessory (typical 75Ω) load if required. A measured external impedance of 75Ω will cause the MICD_LVL [3] bit to be set.

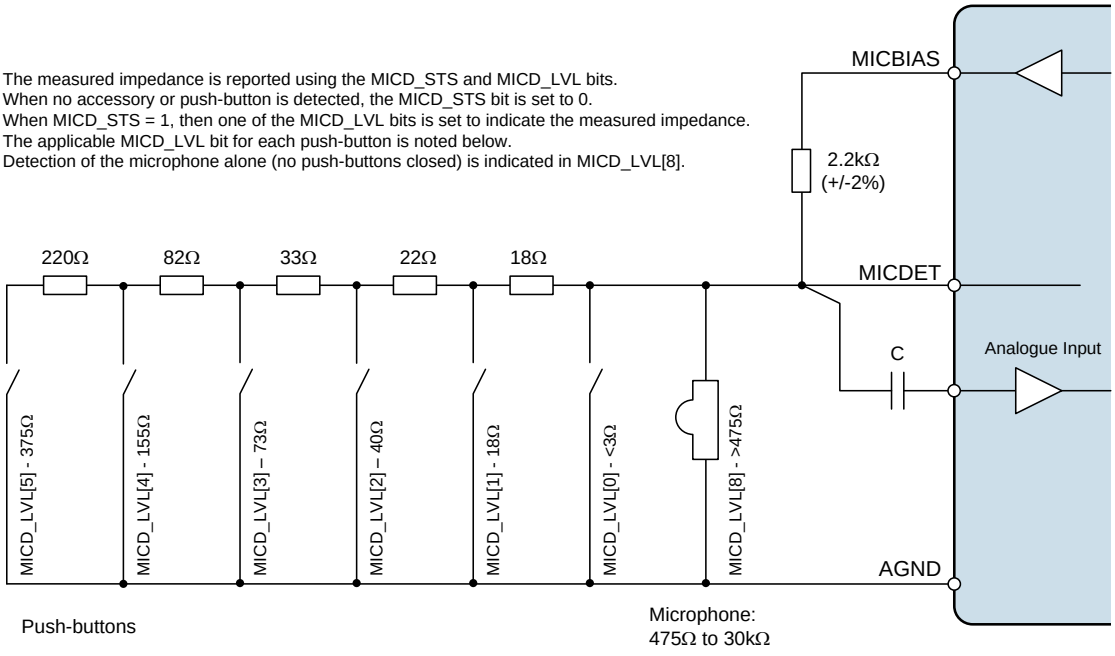


**Figure 98 External Accessory Detect Connection**