# Oceanside
# CX8400 (Redbud Isle) and CX8450 (Starke)

# Table of Contents


1 Device Overview ..... 9
1.1 General Description ..... 9
1.2 Changes from Deep Eddy ..... 11
1.2.1 Overview of changes ..... 11
1.2.2 OPEN Questions: ..... 12
1.2.3 Engineering Innovations ..... 12
1.2.4 New Use Cases ..... 12
1.2.5 Improved Multi-Channel Stream Support ..... 13
1.2.6 Power Off Hum Killer ..... 13
1.2.7 Improvements for operation without AVDD5V ..... 14
1.2.8 Additional Bond Options and Device IDs ..... 14
1.2.9 Part Variations: ..... 15
1.2.10 Four Digital Micorphone Array Support ..... 16
1.2.11 Synchronous Serial Digital Audio Interfaces (SSDAI) ..... 16
1.2.12 I2C master ..... 17
1.2.13 Write Sequencer ..... 17
1.2.14 Data packing ..... 18
1.2.15 Error reporting mechanism for smart amplifiers? ..... 18
1.2.16 PC_Beep support for I2S port ..... 18
1.2.17 Docking support ..... 18
1.3 Block Diagram CX8400/CX8450 ..... 19
1.3.1 Pin Assignments 60-pin CX8400/CX8450 ..... 20
1.4 Performance Requirements ..... 23
1.4.1 DAC Performance ..... 25
1.4.2 ADC Performance ..... 26
1.4.3 Digital Pad Characteristics ..... 26
1.4.4 SPDIF timing ..... 27
1.4.5 Analog Characteristics ..... 27
1.5 Temperature and Power ..... 29
1.6 GS Mark support ..... 30
1.7 Speaker Reference Return Stream ..... 30
1.8 Dual Mode Host Interface ..... 30
1.8.1 Overview ..... 30
1.8.2 HD Audio mode ..... 31
1.8.3 I2S Mode ..... 31
1.9 Clocks and PLL ..... 32
1.10 GPIO ..... 34
1.11 Jack Sense ..... 35
1.12 Headset Inline Control Support ..... 35
1.13 4 button support for Android ..... 36
1.13.1 Button Detect Process ..... 37
1.13.2 Headphone Tip/Ring Impedance Detection ..... 37
1.14 PC BEEP ..... 37
1.14.1 Case 1: Power up & Beep signal sequencing ..... 39
1.14.2 Case 2: Beep S0 and device in D0 ..... 39
1.14.3 Case 3: Beep during S0 with the device in a lower power state ..... 39
1.14.4 Case 4: Beep during resume from suspend or hibernate ..... 39
1.14.5 Power down signals ..... 40
1.15 EAPD and Speaker Mute ..... 41
1.16 Class-D Amplifier Gain ..... 44
1.17 Battery Monitoring ..... 45
1.18 Port C Analog Input Operation ..... 45
1.19 Microphone Bias Pin ..... 45
1.20 Power-off hum elimination ..... 46

2 HD Audio Electrical Interface ..... 47
2.1 Overview ..... 47
2.2 Signaling Environment ..... 47
2.2.1 DC Specifications ..... 47
2.2.2 AC Specifications ..... 48
2.2.3 Maximum AC Ratings and Device Protection ..... 52
2.3 Measurements and test conditions ..... 54
2.4 Timing Specification ..... 55
2.5 System Timing Budget ..... 58
2.6 Frequency Switching ..... 59
2.6.1 Low Frequency Mode ..... 59
2.6.2 Static Frequency Switching ..... 59
2.6.3 Software-visible Resources & Actions ..... 59
3 PCBEEP ..... 61
3.1 Case 1: Power up & Beep ..... 61
3.2 Case 2: Beep S0 and device completely at D0 ..... 61
3.3 Case 3: Beep during S0 but the device is in mixed power states ..... 61
3.4 Case 4: Beep during S0 but the device is in Idle 2 ..... 62
3.5 Case 5: Beep during resume from suspend or hibernate ..... 62
3.6 Power down signals ..... 62
4 Analog Macro ..... 63
4.1 Analog Macro Block Diagram ..... 63
4.2 AFE Interface Signal List ..... 64
4.3 Analog Controller Specification ..... 69
4.3.1 Clocking Signals ..... 69
4.3.2 REF control ..... 70
4.3.3 MICBIAS control ..... 70
4.3.4 RXPGA, AAF, and ADC control and data ..... 71
4.3.5 Multi-button and Tip & Ring detection control ..... 72
4.3.6 CSDAC control and data ..... 72
4.3.7 PORT control ..... 73
4.3.8 CLASSD control ..... 78
4.3.9 CPUMP control ..... 81
4.3.10 PCBEEP control ..... 82
4.3.11 JACKSENSE control ..... 83
4.3.12 CLOCK's ..... 83
4.3.13 POR control ..... 83
5 Chip Level Overview ..... 84
5.1 System Overview ..... 84
5.1.1 Intel HDA (High Definition Audio) Host Interface ..... 84
5.1.2 I2C Host Control Interface ..... 85
5.1.3 I2S Host Audio Data Interface ..... 85
5.1.4 I2C Peripheral Control Interface ..... 85
5.1.5 I2S Peripheral Audio Data Interface ..... 85
5.1.6 SPDIF Peripheral Audio Data Output Interface ..... 85
5.1.7 Digital Microphone Inputs ..... 86
5.1.8 Miscellaneous Digital Signals ..... 86
5.1.9 Cap-less Headphone Output Driver ..... 86
5.1.10 Class-D Speaker Driver ..... 86
5.1.11 Microphone / Line-Level Input Receivers ..... 86
5.1.12 HDA Port Definitions ..... 86
5.1.13 Multi-Button Inline Support and Tip-Ring Impedance detection ..... 87
5.1.14 Smart Amp 1.0 ..... 87
5.1.15 System Diagram ..... 88
5.1.16 Data Flow Topology ..... 89
5.1.17 SoC Block Diagram ..... 92

5.1.18 Clock Relations ..... 92
5.1.19 Oceanside Clock Diagrams ..... 92
5.1.20 Speaker Protection Clocks Block diagram ..... 96
5.2 Device Power Supplies ..... 97
5.2.1 Power Block Diagram ..... 98
5.2.2 Power supplies Specification ..... 98
5.3 Power management ..... 99
5.4 Device Power Consumption ..... 99
5.5 Oceanside Floor Plan ..... 99
5.5.1 Oceanside Standard Cell Utilization ..... 100
5.6 Pin/Pad Descriptions ..... 101
5.7 PAD Record ..... 104
6 PLLs, Clocks and Resets ..... 108
6.1 PLLs ..... 108
6.1.1 PLL Macro IO Table ..... 108
6.1.2 PLL Block Diagram ..... 109
6.1.3 PLL Input Output Operating Frequency ..... 109
6.1.4 PLL Adjustable Variables and Loop Stability Coefficients Table ..... 110
6.1.5 Fractional N – Digital Delta Sigma Modulator ..... 110
6.1.6 Electrical Specification ..... 111
6.2 PLL Tracking Loop ..... 113
6.2.1 General Theory of Operation ..... 113
6.2.2 Detailed Operation ..... 114
6.2.3 Tracking Loop Registers ..... 118
6.3 Clocks ..... 118
6.3.1 Main Clocks ..... 118
6.3.2 Generated/Derived Clocks ..... 118
6.3.3 Internal Gated Clocks ..... 120
6.3.4 Clock Relations ..... 122
6.3.5 Oceanside Clock Diagrams ..... 122
6.3.6 Speaker Protection Clocks Block diagram ..... 126
6.4 Resets ..... 127
6.4.1 Soft-POR operation ..... 128
6.4.2 SnM Reset (Known as D3live reset) ..... 128
6.4.3 Link Reset ..... 129
7 Dual Mode Host Interface ..... 131
7.1 Overview ..... 131
7.1.1 HD Audio mode ..... 131
7.1.2 I2S Mode ..... 131
7.1.3 Functional Blocks ..... 132
7.2 Host Control Interface ..... 132
7.2.1 Control Access – HDA Interface ..... 133
7.2.2 Control Access – I2C Interface ..... 135
7.2.3 Register Addressing – HDA Compliant Verbs ..... 136
7.2.4 Register Addressing – Vendor Defined Verbs ..... 137
8 I2S_PCM Host Interface ..... 140
8.1 General description ..... 140
8.2 Programmer's Reference ..... 141
8.3 Theory of Operation ..... 141
8.4 I2S Mode ..... 141
8.4.1 Left Justified Mode ..... 143
8.4.2 Right Justified Mode ..... 144
8.4.3 Intel Mode ..... 145
8.4.4 PCM Mode ..... 145
8.5 Module Usage ..... 148
8.5.1 I2s interface Design Update ..... 149

8.5.2 Half cycle timing constraint ..... 150
8.5.3 Half cycle timing numbers from timing report ..... 151
8.6 Register Spec ..... 152
9 I2S_PCM Peripheral Interfaces ..... 153
9.1 General description ..... 153
9.2 Macro Definition ..... 154
9.3 Registers ..... 154
9.4 IV Stream Packing ..... 175
9.4.1 Supporting Falcon Format ..... 175
9.4.2 Supporting Neptune Format ..... 176
9.4.3 Re-packing the IV Stream ..... 177
9.4.4 Channel Assignment ..... 178
9.5 IV Packing Register Description ..... 179
9.6 Sample rate conversion ..... 181
9.6.1 Features ..... 181
9.6.2 Path ..... 181
9.6.3 Application in Oceanside ..... 182
9.6.4 Module Interface ..... 183
9.6.5 Memory ..... 189
9.6.6 State Machines ..... 191
9.6.7 Registers ..... 194
9.6.8 FIFO RAM Access ..... 197
9.6.9 Cycle Operation ..... 198
9.6.10 Other Details ..... 200
10 I2C Host Slave Interface ..... 202
10.1 General Description ..... 202
10.2 Theory of Operation ..... 202
10.2.1 I2C Bus Protocol ..... 202
10.2.2 I2C Clock Speed ..... 203
10.2.3 I2C Device ID ..... 203
10.3 SoC Integration and Micro Architecture ..... 203
10.3.1 Module Usage ..... 203
10.3.2 I/O Specifications ..... 204
11 I2C Peripheral Master Interface ..... 206
11.1 General Description ..... 206
11.2 Theory of Operation ..... 206
11.3 Operating Flow ..... 213
11.4 FSM ..... 213
12 I2C Write Sequencer ..... 215
12.1 Operating Principle ..... 215
12.2 Operating Flow ..... 216
12.3 FSM ..... 217
13 DAC Envelope Detector ..... 218
13.1 DAC Envelope Detector Architecture ..... 218
13.1.1 Class-H Amplifier Design Notes ..... 220
13.2 DAC Envelope Detector Implementation ..... 221
13.3 SoC Integration ..... 223
14 OOB Digital Design ..... 224
14.1 General Description ..... 224
14.2 Programmer’s Reference ..... 224
14.2.1 Theory of Operation ..... 224
14.3 Register Spec ..... 226
14.3.1 CODEC_TEST11 ..... 226
15 Multi-Button/TR impedance Design ..... 228
15.1 Overview ..... 228
15.2 FSM for TR Sense and Multi-Button Detection ..... 228

15.3 Programmer's Reference ..... 230
15.3.1 Theory of Operation ..... 230
15.3.2 Data Flow Diagram ..... 231
15.3.3 Module Usage ..... 232
15.4 Testability ..... 234
15.5 Register Spec ..... 234
16 EAPD and Speaker Mute ..... 236
16.1 EAPD Pin ..... 236
16.2 Speaker Mute Pin (SPKR_MUTE#) ..... 238
16.3 Amplifier Control ..... 238
17 Power Averaging (PA-1.1) Speaker Protection ..... 242
17.1 PA-2 System Diagram ..... 242
17.2 Overview ..... 242
17.3 Deep Eddy (updated) Design ..... 243
17.3.1 3rd Order CIC Filter ..... 243
17.3.2 Voltage to Power Conversion (squaring) ..... 246
17.3.3 Low Pass (Averaging) Filter ..... 246
17.3.4 Threshold Comparison ..... 248
17.3.5 Attenuation Control ..... 249
18 Monitor ADC ..... 254
18.1 Overview ..... 254
18.2 Block I/O Signals ..... 254
18.3 Electrical Spec and Area ..... 254
18.4 Analog to Digital Interface Timing ..... 255
18.5 Battery Monitor ADC Structure Overview ..... 255
18.6 Testability ..... 256
19 Port C Analog Input Operation ..... 257
19.1 Port C Analog Input Operation ..... 257
19.2 Microphone Bias Pin ..... 257
19.3 Micbias Circuit Diagram ..... 257
20 EQ and DRC ..... 259
20.1 EQ Coefficient Read Back ..... 259
20.1.1 Register Spec ..... 259
20.2 Critical Configuration Locking ..... 260
20.2.1 2.7.1 Register Locking ..... 260
21 Record Mixer ..... 261
21.1 Overview ..... 261
21.2 Micro Architecture ..... 261
21.2.1 Multiply Path and Control ..... 261
21.3 Pre-Mixer and Post Mixer Data Path Diagram ..... 262
21.4 Node 15: Mixer Widget ..... 263
21.5 Mixer Gain ..... 264
21.6 Register Spec ..... 264
21.6.1 CODEC_TEST32 ..... 264
22 SnM (Music Request Mode) ..... 266
22.1 SnM Overview ..... 266
22.2 Entering SnM Mode ..... 267
23 HD Audio Widgets and Verbs ..... 269
23.1 Widgets/verbs (registers) Removed ..... 269
23.2 Widgets/verbs (registers) Additions or changes ..... 269
23.3 HD Audio verb to I2C Address Mapping ..... 269
23.4 General notes on verbs in this document ..... 271
23.5 Low Power Music Mode ..... 272
23.6 Node 0: Root Node ..... 273
23.6.1 Package and device options ..... 273
23.7 Node 1: AFG Widget ..... 276

23.8 Nodes 10 and 11: DAC 1 and 2 Widgets ..... 282
23.9 Node 12: PC Beep Generator Widget ..... 286
23.10 Node 13: ADC 1 Widget ..... 288
23.11 Node 14: ADC 2 Widget ..... 293
23.12 Node 15: Mixer Widget ..... 298
23.13 Node 16: PORT A / Vendor Widget ..... 301
23.14 Node 17: PORT G BTL Pin Widget ..... 305
23.15 Node 18: PORT B Widget ..... 309
23.16 Node 19: PORT D Widget ..... 314
23.17 Node 1A: PORT C Widget ..... 318
23.18 Node 1D, PORT E Widget ..... 322
23.19 Node 1E, PORT F Widget ..... 326
23.20 Node 1F: PORT H Widget ..... 330
23.21 Node 1B: Vendor Specific Widget ..... 333
23.21.1 EQ Registers ..... 333
23.21.2 DRC Registers ..... 333
23.21.3 EQ Lock Registers ..... 333
23.21.4 BIOS Registers ..... 334
23.21.5 MBIST Registers ..... 334
23.21.6 I2S Registers ..... 334
23.22 Node 1C: Vendor Specific Widget ..... 335
23.22.1 Codec Test Registers ..... 335
23.22.2 Analog Test registers ..... 335
23.22.3 Digital Test registers ..... 335
23.23 Node 20: DAC 3 – SPDIF OUTPUT Widget ..... 336
23.24 Node 21: PORT I SPDIF OUTPUT Widget ..... 340
23.25 Nodes 22 and 23: DAC 4 and 5 Widgets ..... 344
23.26 Nodes 24 and 25: ADC 3 and 4 Widgets ..... 348
23.27 Node 26, PORT J Widget ..... 353
23.28 Node 27, PORT K Widget ..... 357
23.29 Node 28, Vendor Specific Widget ..... 361
23.30 Audio Unsolicited Messages: ..... 362
24 Vender Defined Registers ..... 363
24.1 Oceanside (A0) Register Change List from Deep Eddy (A1) ..... 363
24.2 Vendor 1B Registers ..... 363
24.2.1 SPKR DRC and EQ registers ..... 363
24.2.2 EQ Lock registers ..... 369
24.2.3 Digital Bios Test Registers ..... 369
24.2.4 I2C Master Registers ..... 377
24.2.5 MBIST Registers ..... 382
24.2.6 I2SPCM Registers ..... 383
24.2.7 Peripheral I2SPCM Registers ..... 398
24.3 Vendor 1C Registers ..... 416
24.3.1 Codec Test Registers ..... 416
24.3.2 Multi-Button/TR Registers ..... 429
24.3.3 Digital test registers ..... 432
24.3.4 Analog Codec Test registers ..... 442
25 Design for Test ..... 452
25.1 General Description ..... 452
25.2 Scan ..... 452
25.2.1 Oceanside scan pin mapping ..... 452
25.2.2 IO timing (TBD) ..... 453
25.2.3 Compressed Scan ..... 454
25.2.4 Scan coverage (TBD) ..... 454
25.3 Memory BIST ..... 455
25.3.1 MBIST overview ..... 455

25.3.2 MBIST Groups ..... 456

# 1 Device Overview

## 1.1 General Description

Oceanside is a multi-channel device compliant with the Intel HD Audio bus Rev. 1.0a specification, ECR HDA048-A, Microsoft UAA specification and Windows 8.1/10 requirements. It features two stereo analog DACs, one analog stereo ADC, 1 stereo digital DAC assigned to SPDIF out, one stereo digital ADC assigned to 4 digital microphone ports. It also features a line input, line output, subwoofer output, universal headset jack, and internal stereo 2 watt class-D, with a stereo ADC loop to class-D power amp while in sleep mode.

Features:

- Two analog/four digital ADCs, 24-bit, 48 KHz,96 KHz, or 192 KHz sample rate (limited 44.1 KHz support), with an SNR of 90 dB, a THD of –75 dB, PSRR of 86db.
- Four analog DACS/four digital DACs, 24-bit, 48 kHz, 96 KHz, 192KHz sample rate (limited 44.1 KHz support), with an SNR of 97 dB, a THD of –75 dB, PSRR of 86db.
- 11 HD audio ports:
    - A stereo SPDIF output port {Port I}
        - SPDIF out supports 1.8V digital signaling.
    - One universal stereo DAC/ADC port supporting {Port A, B & D}
        - Stereo Headphone/Headset output
            - Output may be limited in analog to comply with GS MARK
        - Auto detect and switch OMTP vs. CTIA microphone and ground support
        - Stereo line output
        - Stereo line input
        - Stereo microphone support
        - Mono microphone support
        - Detection of and wake from center button presses on a headset.
        - Jack sense with headset/headphone determination
    - Two stereo digital microphone interfaces {Port C, H}
        - Port C is configurable for analog or digital microphones
        - Switchable three digital microphones to single stereo stream
        - 4 digital microphones in a single 4-channel stream
        - Low power modes for VAD (768KHz, 1.5MHz, 3.072MHz)
    - 2W Class-D speaker output with **AudioSmart**™{Port G}
    - Hardware processing
        - 7-biquad filters in the Class-D and stereo line output path
        - DRC for Class-D and stereo line output path
    - Two digital audio interfaces
        - Sample rates up to 192KHz
        - Two input ports (Port F and Port K)
        - Two output ports (Port E and Port J)
        - I2S, LJ, RJ, and TDM modes.
- Supply for bus signaling 1.5V +/- 5%, 1.8V +/-5% and 3.3V +/- 5%.
- Supply for digital 1.8V +/- 5%.
- Supply for analog 1.8V +/- 5%
- Supply for Speaker Amplifier 2.7V to 5.5V

- Internal regulator for microphone bias generation.
- Internal 1.8V to 1.2V regulator for digital core voltage
- Internal 1.8V to 1.6V regulator for analog core voltage
- 4 general purpose GPIOs shared with other functions.
    - GPIO 0/SPDIF/D3 Live (reset function is D3 live)/ DMIC_CLK2
    - GPIO 1/SPKR_MUTE/SPDIF (reset function is SPKR_MUTE)/DMIC_DAT2
    - GPIO 2/DMIC_CLK1
    - GPIO 3/DMIC_DAT1
- 2 additional GPIO on 60-pin package
- Any input port can be secured, forced to silence, by the BIOS
- Headphone output can be hardware limited to comply with GS MARK
- 4 button headset inline control support (Android)
- HDA (CX8400) or I2S/I2C (CX8450) main bus depending on bond option (default state) and I2C/HD Audio command

## 1.2 Changes from Deep Eddy

### 1.2.1 Overview of changes

Deep Eddy needs some bug fixes and enhancements. In addition, I2S bridging capability is requested to support external smart amplifiers and docking applications. The changes are summarized below.

#### 1.2.1.1 Needed Deep Eddy changes:

1. Correct Multi-channel stream support for DACs (especially 96K and 192K support)
2. Add Multi-channel stream support for ADCs
3. Power off hum killer
4. 5V AVDD fix for proper HS detection when returning from AVDD off operation.
5. Correct wake (SDI) behavior from jack sense when AVDD5 off.
6. ~~Additional bond options and IDs~~
    a. ~~Deep Eddy (HD CODEC)(42 and 50pin options)~~
    b. ~~Hunter (HD Jack CODEC)~~
    c. ~~Hornsby (I2S CODEC)(42 and 50pin options)~~
    d. ~~Zilker (I2S Jack CODEC~~
    e. Redbud Isle (HD CODEC with I2S bridge and amp)
    f. ~~Harbor (HD CODEC with I2S bridge w/o AMP)~~
    g. Starke (I2S CODEC with I2S bridge and AMP)
    h. ~~Breakwater (I2S CODEC with I2S bridge w/o AMP)~~

#### 1.2.1.2 Needed additions:

1) 2 I2S ports
    a. Each has input and output capability (independent pin complexes)
    b. I2S and TDM formats
    c. Up to 192KHz sample rates (48KHz, 96KHz, and 192KHz supported)
    d. >24.576MHz BCLK support
2) 2 more DMICs to support new 4-mic SSP and arrays
3) I2C master
4) I2C write sequencer
5) SRC filters for converting between 1,2,4x rates (for all I2S channels!)
6) Support for at least 4 output and 12 input channels in TDM mode
    a. 4 speaker output channels
    b. 4 Reference channels (AEC)
    c. 4 I channels
    d. 4 V channels
7) Data packing/unpacking for SoftAware Sensor Link
8) Error reporting mechanism for smart amplifiers
    a. Interrupt input
    b. Inband signaling (part of I2S stream)?(Not implemented)
9) PC_Beep support for I2S port

10) Docking support
    a. Dock jack sense (2 GPIO)?

#### 1.2.1.3 Not supported

1) 44.1KHz based rates on I2S port
    a. PLL for 44.1KHz I2S rates (For MCLK, BCLK, and LRCLK generation)
    b. 44.1KHz conversion supported to provide host side support for 44.1Khz
2) Local EEPROM for faster boot
    a. How to program smart amp faster?
3) Asynchronous clocking on I2S ports when in slave mode. (No ASRC)

#### 1.2.1.4 Target Package for full featured part

60-pin 7x7mm QFN

### 1.2.2 OPEN Questions:

1) Is D3 Live needed?
2) Is a separate Speaker Mute pin needed?
3) Is it OK that SPDIF is not supported when 4 DMICs are used?
4) How to map multi-channel TDM mode on peripheral I2S to HD audio and host I2S modes?

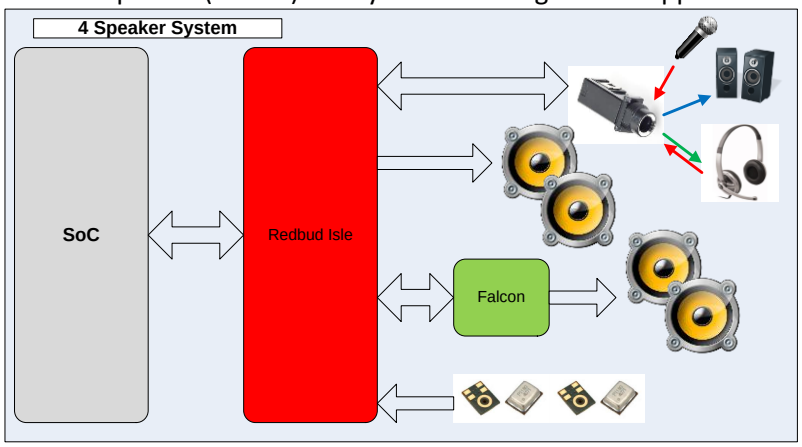
### 1.2.3 Engineering Innovations

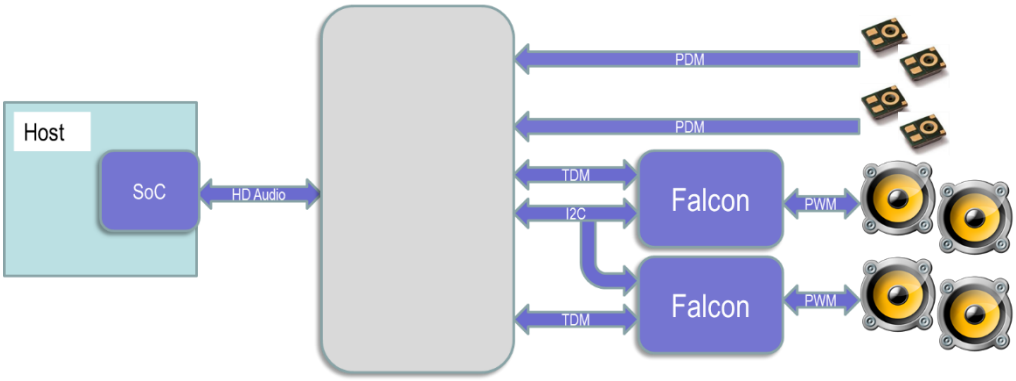
Data packing/repacking – aggregate and manage bandwidth needs from external smart amplifiers and host interface.

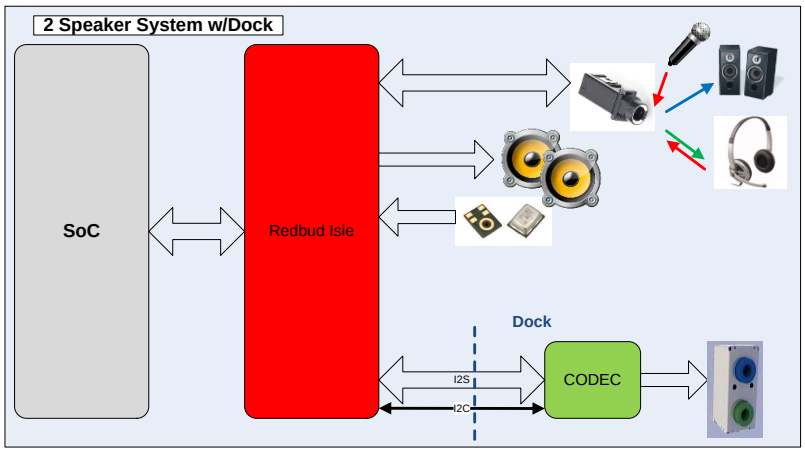
~~Proprietary in-band signaling technique for reporting error conditions from a Conexant smart amplifier to the CODEC using the digital audio interface.~~

### 1.2.4 New Use Cases

The addition of 2 digital audio interfaces and a master I2C control interface allows for supporting 1 or 2 smart amplifiers (Falcon) in a system. Docking is also supported






## 1.2.5 Improved Multi-Channel Stream Support

Cancun, Solana, Neptune, and Deep Eddy have limited support for sample rates above 48KHz. In particular, supporting 4 channels out of an 8-channel stream may work at 48KHz but not 192KHz. Also, selecting the Front and Rear, but not CTR/LFE of a 6-channel stream may not work depending on the order that the 6 channels are transmitted.
The improvements needed are known and have been implemented and verified in RTL but not in an actual product.

In addition, allowing more than one ADC to share an input stream on the HD Audio interface has not previously been supported. Again, the capability has been implemented in RTL and verified but not implemented in a shipping product.

## 1.2.6 Power Off Hum Killer

Some customers (Lenovo) have responded to a competitor's addition of this feature by demanding that all new products support the ability to prevent floating headset jack return pins when power is removed from the system. The existing capability to connect the headset return pin(s) to local ground requires an always on supply. Using Depeltion mode FETs solves this issue and has been implemented in a previous product.

### 1.2.7 Improvements for operation without AVDD5V

Headset detection requires an active AVDD5 supply. Under some low power modes (system sleep or standby) this supply may be turned off. The insertion of a plug will be properly detected with the existing circuitry but the headset detection will always return the same value due to the lack of AVDD5 and this value is likely incorrect.

It is acceptably for the initial sense cycle to be incorrect as the system is in a state where audio is not supported and is transitioning into a higher power state where audio is supported.
The plug event that triggers the headset detect cycle will also request a power state change from the host. As long as a new headset sense cycle is performed as soon as AVDD5 is turned on then the problem is reduced.

The ideal behavior would be to suppress the initial unsolicited response after the wake event has been sent and send an unsolicited response only after AVDD5 is active. This however would be problematic if AVDD5 is not restored. The suggested approach is to allow the initial wake and unsolicited response events to occur and generate a new unsolicited response after AVDD5 is active. The application of AVDD5 causing an automatic headset detect cycle if the jack detect indicates the presence of a plug in the jack. Also recommended is forcing the return device type to be headphone instead of CTIA headset if AVDD5 is off.

Related to headset detection is the jack sense (port presence detection) and wake function. As previously stated, the insertion or removal of a plug is properly detected. The following behavior is required:

1) After the debounce period, a wake event is triggered to notfy the host that AVDD5 and other supplies that are currently off need to be turned on.
   a) Currently an event is reliably provided when Ports A and B are in merge mode but not when Port A is used by itself.
   b) When Port A is not in merge mode, the headset detect state machine should be modified to send a wake event before proceeding with the headset detection to allow time for the system to react and turn on AVDD5.
2) The CODEC proceeds with the headset detection routine
   a) An unsolicited response is sent notifying the system that the presence status has changed for the headset jack (Port A or Port B when in merge mode)
   b) The pin complex presence status bit(s) are updated.
   c) The headset type is updated
   d) Port D presence unsolicited response is sent after a programmable delay.
3) If AVDD5 was previously off and is turned on during or after the headset detect cycle.
   a) A new headset detect cycle is automatically triggered when AVDD5 is turned on
   b) The headset type value is updated
   c) The port presence detect values are updated
   d) Unsolicited responses are sent to update the system regarding the port presence

### 1.2.8 Additional Bond Options and Device IDs

~~It is necessary to support the existing bond options and the new feature sets.~~
~~New options and IDs are needed for the I2S bridge chips. It is recommended that a device ID is added for the “Zilker” and “Hunter” part variants that currently reuses an existing ID.~~

Only the Redbud Isle (CX8400) and Starke (CX8450) will be supported for HP. Other possible bond options will be ignored.

### 1.2.9 Part Variations:

1) ~~Deep Eddy (HD CODEC) 42-pin package CX8070~~
2) ~~Deep Eddy (HD CODEC) 50-pin package CX8071~~
3) ~~Hornsby (I2S CODEC) 42pin package CX8120~~
4) ~~Hornsby (I2S CODEC) 50pin package CX8121~~
5) ~~Hunter (HD Jack CODEC) 42-pin package~~
6) ~~Zilker (I2S Jack CODEC) 42pin package CX8125~~
7) Redbud Isle (HD CODEC) with I2S bridge and amp)
8) ~~Harbor (HD CODEC with I2S bridge w/o AMP)~~
9) Starke (I2S CODEC with I2S bridge and AMP)
10) ~~Breakwater (I2S CODEC with I2S bridge w/o AMP)~~

| Device Name | HD Audio | I2S | AudioSmart Amp | I2S bridge | Software Option | Alternate I2C Addr | Part Number | PNP_ID | PKG |
|---|---|---|---|---|---|---|---|---|---|
| ~~Deep Eddy~~ | ~~✓~~ | | ~~✓~~ | | | | ~~CX8070-??Z~~ | ~~0x14F11F86~~ | ~~42 QFN~~ |
| | ~~✓~~ | | ~~✓~~ | | ~~✓~~ | | ~~CX8071-??Z~~ | ~~0x14F11F87~~ | ~~50 QFN~~ |
| ~~Hornsby~~ | | ~~✓~~ | ~~✓~~ | | | | ~~CX8120-??Z~~ | ~~0x14F11FB8~~ | ~~42 QFN~~ |
| | | ~~✓~~ | ~~✓~~ | | ~~✓~~ | | ~~CX8121-??Z~~ | ~~0x14F11FB9~~ | ~~50 QFN~~ |
| Redbud Isle | ✓ | | ✓ | ✓ | | | CX8400-??Z | 0x14F120D0 | 60 QFN |
| | ~~✓~~ | | ~~✓~~ | ~~✓~~ | ~~✓~~ | | ~~CX8401-??Z~~ | ~~0x14F120D1~~ | ~~60 QFN~~ |
| Starke | | ✓ | ✓ | ✓ | | | CX8450-??Z | 0x14F12102 | 60 QFN |
| | | ~~✓~~ | ~~✓~~ | ~~✓~~ | ~~✓~~ | | ~~CX8451-??Z~~ | ~~0x14F12103~~ | ~~60 QFN~~ |
| ~~Hunter~~ | ~~✓~~ | | | | | | ~~CX8075-??Z~~ | ~~0x14F1206C~~ | ~~42 QFN~~ |
| | ~~✓~~ | | | | ~~✓~~ | | ~~CX8076-??Z~~ | ~~0x14F1206D~~ | ~~42 QFN~~ |
| ~~Zilker~~ | | ~~✓~~ | | | | ~~✓~~ | ~~CX8125-??Z~~ | ~~0x14F1206E~~ | ~~42 QFN~~ |
| | | ~~✓~~ | | | ~~✓~~ | | ~~CX8126-??Z~~ | ~~0x14F1206F~~ | ~~42 QFN~~ |
| ~~Harbor~~ | ~~✓~~ | | | ~~✓~~ | | | ~~CX8405-??Z~~ | ~~0x14F120D5~~ | ~~50 QFN~~ |
| | ~~✓~~ | | | ~~✓~~ | ~~✓~~ | | ~~CX8406-??Z~~ | ~~0x14F120D6~~ | ~~50 QFN~~ |
| ~~Breakwater~~ | | ~~✓~~ | | ~~✓~~ | | | ~~CX8455-??Z~~ | ~~0x14F12107~~ | ~~50 QFN~~ |
| | | ~~✓~~ | | ~~✓~~ | ~~✓~~ | | ~~CX8456-??Z~~ | ~~0x14F12108~~ | ~~50 QFN~~ |

| Device Name | Bond Pad values BOP{5:1} | STD part# | PNP_ID | PKG |
|---|---|---|---|---|
| Deep Eddy | 11001 | CX8070 | 0x14F11F86 | 42 QFN |
| Deep Eddy (Neptune Lite) | 11000 | CX8071 | 0x14F11F87 | 50 QFN |
| Hornsby | 11011 | CX8120 | 0x14F11FB8 | 42 QFN |
| Hornsby (Atlantic lite) | 11010 | CX8121 | 0x14F11FB9 | 50 QFN |
| Redbud Isle | 10001 | CX8400 | 0x14F120D0 | 60QFN |
| Redbud Isle Alternate ID | 10000 | CX8401 | 0x14F120D1 | 60QFN |
| Starke | 10011 | CX8450 | 0x14F12102 | 60QFN |
| Starke Alternate ID | 10110 | CX8451 | 0x14F12103 | 60QFN |

| | | | | |
|---|---|---|---|---|
| Hunter | 01001 | CX8075 | 0x14F11F8B | 42 QFN |
| Hunter Alternate ID | 01000 | CX8076 | 0x14F1206D | 42 QFN |
| Zilker* | 01111 | CX8125 | 0x14F11FBD | 42 QFN |
| Zilker Alternate ID* | 01010 | CX8126 | 0x14F1206F | 42 QFN |
| Harbor | 00001 | CX8405 | 0x14F120D5 | 50QFN |
| Harbor Alternate ID | 00000 | CX8406 | 0x14F120D6 | 50QFN |
| Breakwater | 00011 | CX8455 | 0x14F12107 | 50QFN |
| Breakwater Alternate ID | 00110 | CX8456 | 0x14F12108 | 50QFN |

Note: Zilker previously used the same Bond Option and ID as Hornsby(42-pin) but used the 0x37 I2C address.

Bond options decoded as follows:

| Bond Option | Net Name | 0 | 1 | Notes |
|---|---|---|---|---|
| BOP5 | OPT5 | No Speaker Amp | AudioSmart Amp | Polarity so Deep Eddy has no downbond (pad unconnected) |
| BOP4 | OPT4 | I2S bridge support | No bridge | Polarity so Deep Eddy has no downbond (pad unconnected) |
| BOP3 | I2C_DEV_ID0 | 0110011R/nW | 0110111R/nW | I2C Address D2 follows bond pad |
| BOP2 | HBUS_MODE | HD Audio | I2S/I2C | |
| BOP1 | OPT1 | Base DEVID (50pin Deep Eddy) | Alternate DEVID (42pin Deep Eddy) | Alternate Device ID (and package indicator for Deep Eddy / Hornsby) |

### 1.2.10 Four Digital Micorphone Array Support

To support 4 mic SSP and microphone arrays wirh 4 microphones, support for an additional 2 digital microphones is required.
Port H functionality will be added back to the design (removed from Solana/Neptune to make Deep Eddy).
Since GPIO4 and GPIO5 are not available, GPIO0 and GPIO1 will be used to support Port H. However, this prevents the use of those pins for alternate functions such as SPDIF output, interrupt output, mice mute input, SnM mode request, or speaker mute.

To fully support microhone arrays, the 4 digital microphones must be aligned in phase (all samples occurring at the same point in time). Further, the CODEC must be able to transport the samples as part of a 4 channel stream (in HD Audio mode) or in separate 2 channel streams. This is addressed in a separate section.

### 1.2.11 Synchronous Serial Digital Audio Interfaces (SSDAI)

Two Digital Audio Interfaces are added to the base design to support external amplifiers and docking applications.
Each interface may be independently configured for rate and format. Each has its own MCLK, BCLK, and LRCLK pins.
Sample rates of 48KHz, 96KHz, and 192KHz are supported.
Master and Slave modes are supported
MCLK operates to at least 24.576Mhz
BCLK operates up to 24.576MHz to support 8, 16-bit channels at 192KHz.

LRCLK operates up to 192KHz
I2S, Left Justified, Right Justified, and TDM modes are supported.
To support smart amplifiers, multi-channel TDM mode is needed.
Supporting 2 amplifiers requires:
4 channels of output to the amp at up to 192KHz
4 channels of AEC reference
4 channels of I data
4 channels of V data
This would require 12 channels of data input
I/V data may be handled in 2 modes

1) Smart Amp handles data as is currently done on Neptune to pack into 2 or 4 channels depending on rate and format.
   a) We would be limited to 1amp per interface in some format/rate combinations
   b) The data channels would map 1-1 between the peripheral interface and the host interface unless additional aggreagation/packing is used (see Packing section)
2) Smart Amp provides new packing modes
   a) Data may be directly transferred from peripheral interface to host interface
   b) Data from peripheral interfaces may be aggregated into a single stream to the host
   c) Data from the peripheral interfaces may be unpacked and repacked for transmission to the host (see Packing section)

The peripheral interfaces and the host interface may be configured for different rates. HD audio mode specifically requires that we support the same set of sample rates for all DACs (except SPDIF) and for all ADCs (though the DACs and ADCs don't have to support the same set of rates.) Therefore it is required that sample rate conversion is provided for all channels. In theory this would require at least 12 channels of processing for input and 4 for output.
More evaluation is needed.

### 1.2.12 I2C master

An I2C master interface is needed to control external peripherals.
This interface operates at least up to 400KHz though faster is better.
The controller may be accessed from either the HD Audio or host I2C interface.

It is desired to have efficient methods for bridging from HD Audio or I2C to I2C so that the bus bandwidth is maximized. System start-up delay is critical and external smart amplifiers may require considerable configuration.

### 1.2.13 Write Sequencer

To reduce the burden on the host to program and reprogram external peripherals (smart amps) during power state changes, a write sequencer may be used to store one or more configuration sequences that write a series of commands to the external device in response to a single command from the host (or autonomously). The expectation is that the host will program the sequencer initially which will require a delay but after initial programming the host need only to start the sequencer and may proceed with other resume activities while the CODEC configures the external device.

### 1.2.14 Data packing

Smart amplifiers require many channels of data to support host based processing of their “smart” algorithms and to implement acoustic echo cancellation (AEC).
4 channels of smart amplifier output can require 12 channels of data retuned to the host.
Not all host interfaces can support more than 4 channels of input. In addition, DMA and other factors may cause the data to become skewed releative to each channel.
To help conform to a channel limited infrastructure, maintain sample alignment, and conserve bandwidth, data packing techniques are used.

### 1.2.15 Error reporting mechanism for smart amplifiers?

Error conditions encountered by external amplifiers should be visible to the host. Internal error conditions can use the HD audio unsolicited response reporting mechanism.
When the host interface is in I2C mode, a GPIO pin may be used as an interrupt output. (WARNING – Only 4 GPIO and all are needed for DMIC!)

An external amplifier should have a means of notifying the CODEC that an error has occurred. There are 2 options:

1) use a GPIO to provide an interrupt input (Not enough GPIO!)
2) ~~Implement a proprietary in-band signaling technique on the Digital Audio Interface~~
   a) ~~Requires that the interface is active … or~~
   b) ~~Requires a means of signaling when the interface is inactive in both master and slave modes~~

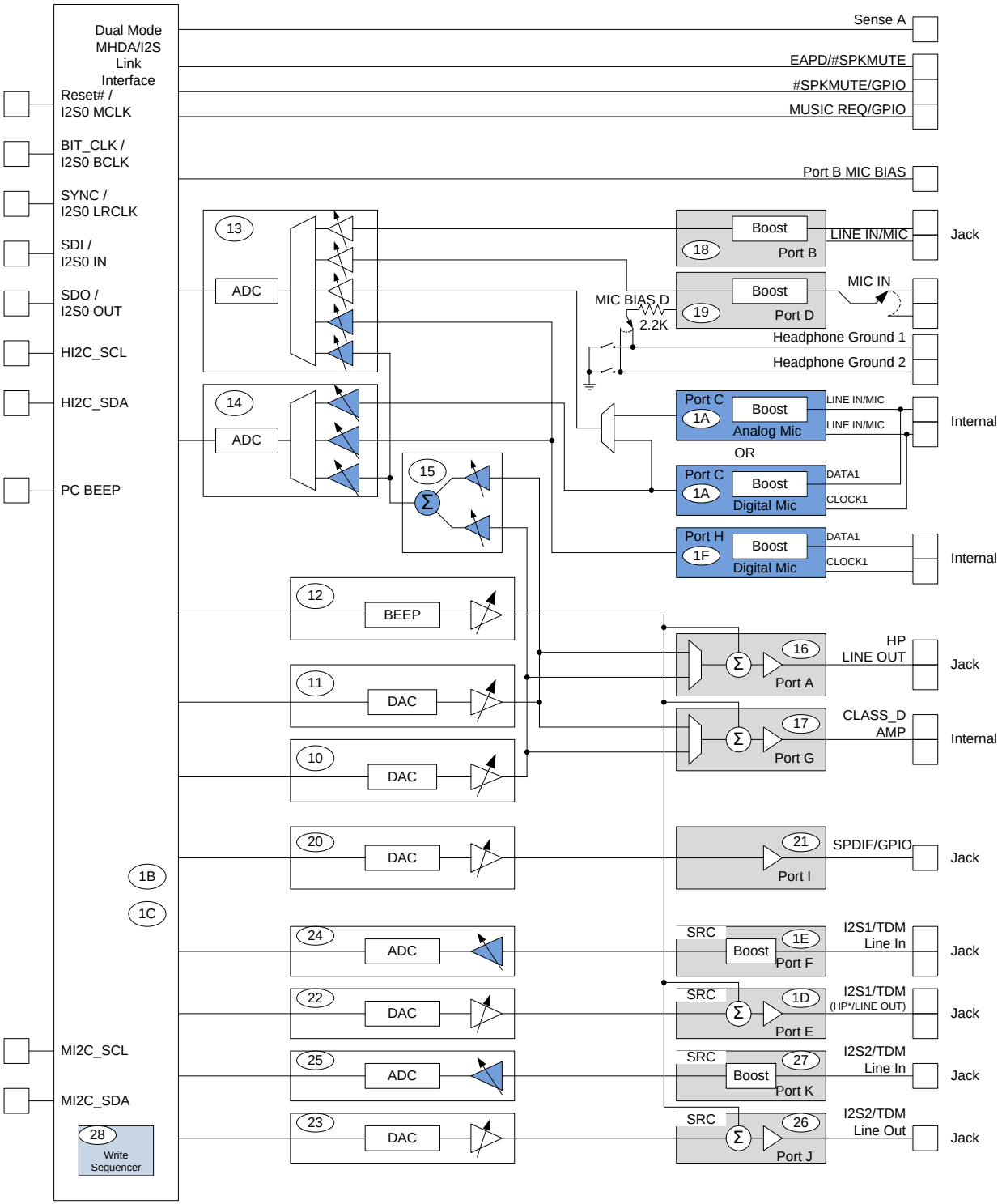
### 1.2.16 PC_Beep support for I2S port

The option to digitize the analog PC_BEEP and digital PC_BEEP signals and send across the digital audio interfaces is required.

### 1.2.17 Docking support

Dock jack sense (2 GPIO) for ports E/F using GPIO 4/5
Not enough available pins when all resources active. Would have to sacrifice DMIC or other functions.

## 1.3 Block Diagram CX8400/CX8450

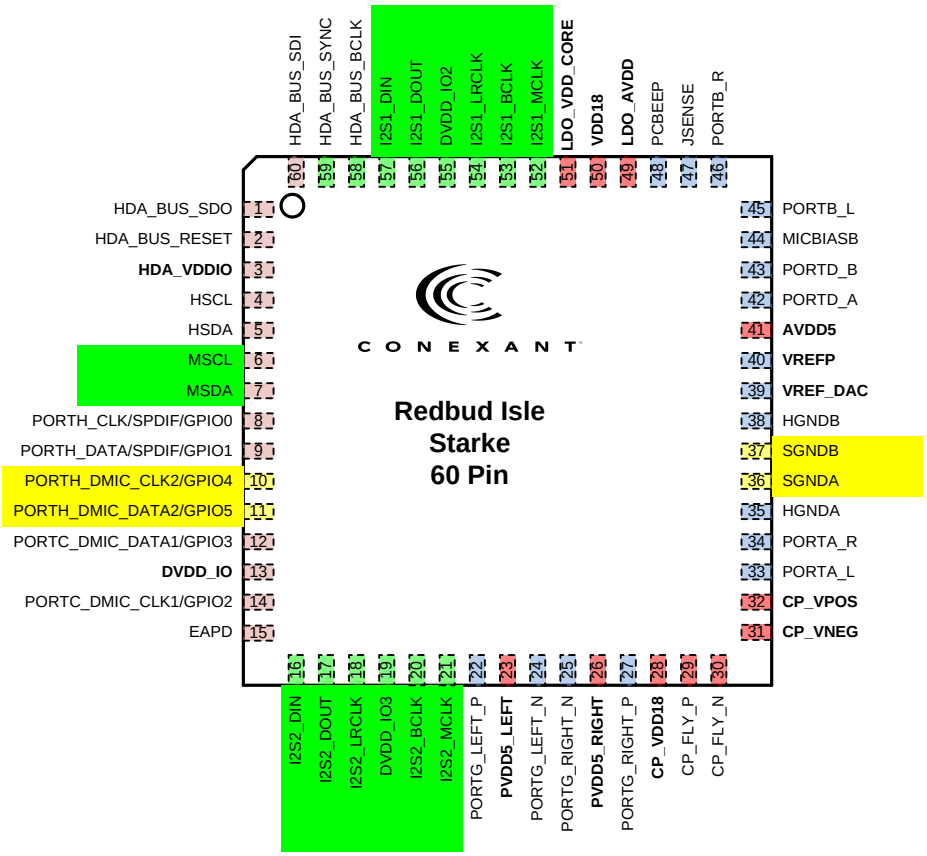


### 1.3.1 Pin Assignments 60-pin CX8400/CX8450

| Pin description | Pin # | Pin description | Pin # |
|---|---|---|---|
| HDA bus SDO | 1 | CP_VNEG | 31 |
| HDA bus reset | 2 | CP_VPOS | 32 |
| HDA VDDIO | 3 | PORTA LEFT (HEADPHONE/MIC BIAS) | 33 |
| Slave I2C SCLK | 4 | PORTA RIGHT (HEADPHONE/MIC BIAS) | 34 |
| Slave I2C SDATA | 5 | HEADSET GND(CTIA) / MIC BIAS(OMTP) | 35 |
| MSCL | 6 | SGNDA | 36 |
| MSDA | 7 | SGNDB | 37 |
| MUSIC_REQ/SPDIF/GPIO0/PORTH_CLK | 8 | HEADSET GND(OMTP) / MIC BIAS(CTIA) | 38 |
| SPKR_MUTE#/SPDIF/GPIO1/PORTH_DATA | 9 | VREF_DAC | 39 |
| PORTH_CLK/GPIO4 | 10 | VREFP (analog reference) | 40 |
| PORTH_DATA/GPIO5 | 11 | AVDD5 | 41 |
| PORTC_DATA/GPIO3 | 12 | PortD A (HEADSET MIC INPUT FOR CTIA) | 42 |
| DVDDIO | 13 | PortD B (HEADSET MIC INPUT FOR OMTP) | 43 |
| PORTC_CLK/GPIO2 | 14 | MICBIAS_B | 44 |
| EAPD | 15 | PORTB LEFT | 45 |
| I2S2_DIN | 16 | PORTB RIGHT | 46 |
| I2S2_DOUT | 17 | JSENSE | 47 |
| I2S2_LRCLK | 18 | PC BEEP | 48 |
| DVDDIO3 | 19 | LDO_AVDD | 49 |
| I2S2_BCLK | 20 | VDD18 | 50 |
| I2S2_MCLK | 21 | LDO_VDD_Core | 51 |
| PORTG LEFT + | 22 | I2S1_MCLK | 52 |
| PVDD LEFT (class-D voltage) | 23 | I2S1_BCLK | 53 |
| PORTG LEFT - | 24 | I2S1_LRCLK | 54 |
| PORTG RIGHT - | 25 | DVDDIO2 | 55 |
| PVDD RIGHT (class-D voltage) | 26 | I2S1_DOUT | 56 |
| PORTG RIGHT + | 27 | I2S1_DIN | 57 |
| CP_AVDD18 | 28 | HDA bus bclk | 58 |
| FLY_P | 29 | HDA bus sync | 59 |
| FLY_N | 30 | HDA bus SDI | 60 |

## Redbud Isle (60 Pin Package)

**grounds all downbonded to a common paddle**




Conexant Systems Inc
Proprietary and Confidential
ALL INFORMATION IN THIS DOCUMENT IS PRELIMINARY AND SUBJECT TO CHANGE WITHOUT NOTICE

Pin Reset Default and State

| pin name | Default function | Secondary functions | Default options | Secondary options |
|---|---|---|---|---|
| **HSCL** | Host_SCL | | Hi-Z | - |
| **HSDA** | Host_SDA | | Hi-Z | - |
| **MSCL** | Peripheral_SCL | | Hi-Z | |
| **MSDA** | Peripheral_SDA | | Hi-Z | |
| **EAPD** | EAPD | | Pull-Up | Hi-Z |
| **Music_Req** | D3 LIVE | GPIO 0, SPDIF, Mic_Mute, INT, DMIC2_CLK | weak pull-down | Hi-Z |
| **SPKR_MUTE#** | SPDIF | GPIO 1, SPDIF, Mic_Mute, INT, DMIC2_DATA | weak pull-down | Hi-Z |
| **DMIC 1 CLK** | DMIC 1 CLK | GPIO 2, Mic_Mute, INT | weak pull-down | Hi-Z |
| **DMIC 1 DATA** | DMIC 1 DATA | GPIO 3, Mic_Mute, INT | weak pull-down | Hi-Z |
| **GPIO4** | DMIC 2 CLK | GPIO 4, Port E/F jack detect | weak pull-down | Hi-Z |
| **GPIO5** | DMIC 2 DATA | GPIO 5, Secondary EAPD, Port E/F jack detect | weak pull-down | Hi-Z |

## 1.4 Performance Requirements

PSRR >85dB for all analog inputs and outputs (Intel)

Note that THD+N and crosstalk targets are listed as positive numbers in the table below in alignment with Microsoft's convention rather than the, more correct, convention of a negative value. All measurements are relative (ratio) to full-scale.

| Device Type | Requirement | Min | Typ | Max | Frequency range at 48KHz and above |
|---|---|---|---|---|---|
| Analog Line Output Jack | THD+N @-3dBFS | 75dB FS | 84 | | [20 Hz, 20 KHz] |
| | Dynamic range with signal present | 90dB FS Awt | 96 | | [20 Hz, 20 KHz] |
| | Magnitude Response | | | +- .25 dB ripple (.5dB peak to peak delta), 1 dB at upper band edge, 3dB at lower band edge | [20 Hz, 20 KHz] |
| | Sampling frequency accuracy | | | 0.02%. | |
| | Line output cross-talk | 80dB | | | [20 Hz, 15KHz] |
| | Full scale output voltage | 1Vrms | | | |
| | Noise level during system activity | 95 dB FS Awt | | | |
| | Interchannel phase delay | | | 30 degrees or 12.5 microseconds, whichever is greater | [20 Hz, 20 KHz] |
| | | | | | |
| Speaker Amplifier | THD+N @ rated output power (2W 4Ω) | 40dB | | | [20 Hz, 20 KHz] |
| | THD+N @ -3dB FS (1W 4Ω) | 65dB | | | [20 Hz, 20 KHz] |
| | Dynamic range with signal present | 90dB FS Awt | | | [20 Hz, 20 KHz] |
| | Magnitude Response | | | +-.25 dB ripple (.5dB peak to peak delta), 1 dB at upper band edge, 3dB at lower band edge | [20 Hz, 20 KHz] |
| | Sampling frequency accuracy | | | 0.02%. | |
| | cross-talk | 65dB | | | [20 Hz, 15 KHz] |
| | Rated Output Power PVDD=5V, 4ohm, 1% THD | 2W | | | |
| | Rated Output Power PVDD=5V, 4ohm, 10% THD | 2.8W | | | |
| | Noise level during system activity | 90dB FS Awt | | | |
| | Interchannel phase delay | | | 30° or 12.5 µs, whichever is greater | [20 Hz, 20 KHz] |
| | | | | | |
| | THD+N @ -3dBFS / 10KΩ | 75dB FS | | | [100 Hz, 20 KHz] |

| Device Type | Requirement | Min | Typ | Max | Frequency range at 48KHz and above |
|---|---|---|---|---|---|
| Analog Headphone Out Jack | THD+N @ -3dBFS / 300Ω | 75dB FS | | | [100 Hz, 20 KHz] |
| | THD+N @ -3dBFS / 32Ω | 60dB FS | | | [100 Hz, 20 KHz] |
| | THD+N @ -3dBFS / 16Ω | 60dB FS | | | [100 Hz, 20 KHz] |
| | Dynamic range with signal present, all loads | 90dB FS Awt | 96dB FS Awt | | [100 Hz, 20 KHz] |
| | Magnitude Response | | | +-.25 dB ripple (.5dB p-p delta), 1 dB at upper band edge, 3dB at lower band edge | [100 Hz ,20 KHz] |
| | Sampling frequency accuracy | | | 0.02%. | |
| | Headphone output cross-talk (at jack) | 75dB (50dB 32Ω) | | | [100 Hz, 15 KHz] |
| | Full scale output voltage, 10KΩ load | 1V rms | | | (3) |
| | Full scale output voltage, 300Ω load | 800mV rms | | | |
| | Full scale output voltage, 32Ω load | 500mV rms | | | |
| | Noise level during system activity 300Ω | 95 dB FS Awt | | | |
| | Noise level during system activity 32 Ω | 80 dB FS Awt | | | |
| | Interchannel phase delay | | | 30° or 12.5 µs, whichever is greater | [100 Hz, 20 KHz] |
| | | | | | |
| Analog Line In Jack | THD+N @ -3dB FS | 75dB FS | | | [20 Hz, 20 KHz] |
| | Dynamic range with signal present | 90dB FS Awt | | | [20 Hz, 20 KHz] |
| | Magnitude Response | | | <= +-.25 dB ripple (.5dB peak to peak delta), 1 dB at upper band edge, 3dB at lower band edge | [20 Hz, 20 KHz] |
| | Sampling frequency accuracy | | | 0.02%. | |
| | Full scale input voltage | 0.707V rms | | | |
| | | | | | |
| Analog Microphone In Jack | THD+N @ -3dB FS (20dB boost) | 75 dB FS | | | [100 Hz, 20 KHz] |
| | Dynamic range with signal present (20dB boost) | 85 dB FS Awt | | | [100 Hz, 20 KHz] |
| | Magnitude Response | | | <= +-.25 dB ripple (.5dB peak to peak delta), 1 dB at upper band edge, 3dB at lower band edge | [100 Hz, 20 KHz] |
| | Sampling frequency accuracy | | | 0.02%. | |

| Device Type | Requirement | Min | Typ | Max | Frequency range at 48KHz and above |
|---|---|---|---|---|---|
| | Full scale input voltage (20dB Boost) | 0.071 Vrms | | | |

### 1.4.1 DAC Performance

Test Conditions: $T_{ambient}$ = 25C, DVDD = 1.8V±5%, AVDD = 1.8V±5%
1kHz sine wave, Sampling Frequency = 48kHz,
10kΩ/50pF load, BW = 10Hz - 20kHz

| Parameter | Min | Typical | Max | Units |
|---|---|---|---|---|
| Full Scale output Voltage – Headphone/Line | | 1.0 | | Vrms |
| S/N (A weighted) – Headphone/Line | | 97 | | dBr |
| THD+N – Headphone/Line | | -75 | | dBr |
| Frequency Response | 20 | | 20,000 | Hz |
| PSRR | | -85 | | dB |
| Total out of band noise (28.8kHz – 100kHz) | | -60 | | dB |
| Amplifier Gain Step | | 1 | | dB |
| Output Impedance<br>Headphone amp<br>Line amp | | <br>1<br>200 | | <br>Ω<br>Ω |
| Power consumption | | | | mW |
| Channel Separation | | 75 | | dB |
| Inter-channel gain mismatch | | .2 | | dB |

Digital Filters for DAC

| Parameters | Min | Typical | Max | Units |
|---|---|---|---|---|
| Pass band | 20 | | 20,000 | Hz |
| Stop band | 28,800 | | 100,000 | Hz |
| Stop band rejection | | -76 | | dB |
| Pass band ripple | | +-0.25 | | dB |

DAC filter

| Parameter | 48kHz | Units |
|---|---|---|
| Passband (-0.2dB point) | 20 | Hz |
| Passband (-2.0dB point) | 10 | Hz |
| Passband (-0.2dB point) | 20.3 | kHz |
| Passband (-0.7dB point) | 23 | kHz |
| Trasition (-0.7dB to -76dB) | 23.0 – 28.8 | kHz |
| Stopband Rejection | -76 | dB |

### 1.4.2 ADC Performance

Test Conditions: $T_{ambient}$ = 25C, DVDD = 3.3V±5%, AVDD = 3.3V±5%
1kHz sine wave, Sampling Frequency = 48kHz,
10kΩ/50pF load, BW = 10Hz - 20kHz

| Parameter | Min | Typical | Max | Units |
|---|---|---|---|---|
| Full Scale input Voltage | | 1.0 | | Vrms |
| S/N (A weighted) | | 90 | | dBFSA |
| THD+N | | -80 | | dBFS |
| Frequency Response | 20 | | 20,000 | Hz |
| PSRR | | -85 | | dB |
| Total out of band noise (28.8kHz – 100kHz) | | -60 | | dB |
| Amplifier Gain Step | | 1 | | dB |
| Input Impedance | | 15 | | kΩ |
| Crosstalk between input channels | | -75 | | dB |
| Anti-aliasing cut-off | | | | kHz |
| Cut off at Fs | | | | dB |
| Power consumption | | | | mW |

| Symbol | 48kHz | Units |
|---|---|---|
| Passband (-0.2dB point) | 20.3 | kHz |
| Transition (-3dB to -74dB) | 23.0 – 28.8 | kHz |
| Stopband Rejection | -74 | dB |

Digital Filters for ADC

| Parameters | Min | Typical | Max | Units |
|---|---|---|---|---|
| Pass band | 0 | | 19.2 | kHz |
| Stop band | 28.8 | | 100k | kHz |
| Stop band rejection | | -76 | | dB |
| Pass band ripple | | ±0.2 | | dB |

### 1.4.3 Digital Pad Characteristics

| Parameter | Symbol | Min | Max | Units |
|---|---|---|---|---|
| Digital Power supply | $DV_{DD}$ | 1.7 | 1.9 | V |
| Bus Signaling supply | $V_{DDIO}$ | 1.4 | 3.46 | V |
| High-level Input Voltage (PC-BEEP) | $V_{IH}$ | $0.4*V_{DD}$ | $V_{DD}+0.3$ | V |
| Low-Level Input Voltage (PC_BEEP) | $V_{IL}$ | -0.3 | $0.2*V_{DD}$ | V |
| High-Level Output Voltage (SPDIF) | $V_{OH}$ | $0.8*V_{DD}$ | $V_{DD}$ | V |
| Low-Level Output Voltage (SPDIF) | $V_{OL}$ | | 0.4 | V |

| | | | | |
|---|---|---|---|---|
| High-level Input Voltage (Bus Interface) | $V_{IH}$ | | $V_{DDIO}$+0.3 | V |
| Low-Level Input Voltage (Bus Interface) | $V_{IL}$ | -0.3 | | V |
| High-Level Output Voltage (Bus Interface) | $V_{OH}$ | | $V_{DDIO}$ | V |
| Low-Level Output Voltage (Bus Interface) | $V_{OL}$ | | | V |
| Input Leakage Current (PC-BEEP) | | -10 | 10 | μA |

### 1.4.4 SPDIF timing

| Parameter | Min | Typical | Max | Units |
|---|---|---|---|---|
| Output frequency | 5.6448 | 6.144 | 12.288 | MHz |
| Output period | 177.1 | 162.8 | 81.38 | nS |
| Output jitter | | | 4 | nS |
| Output high level width | 48 | 50 | 52 | % |
| Output low level width | 48 | 50 | 52 | % |
| Output rise time | | 2 | | nS |
| Output fall time | | 2 | | nS |

### 1.4.5 Analog Characteristics

Input/output ports

| Parameter | Min | Typical | Max | Units |
|---|---|---|---|---|
| Resistive load (headphone) | | 32 | | Ω |
| Resistive load (line out) | | 10k | | Ω |
| Capacitive load (headphone) | | | 500 | pF |
| Capacitive load (line out) | | | 100 | pF |
| AC-coupling capacitor | | | 4.7 | uF |
| Output voltage | | | 1 | Vrms |
| Input voltage | | | 1 | Vrms |
| Power consumption (headphone) | | 50 | | mW |
| Power consumption (line out) | | 1.3 | | mW |
| SNR @ 0dB | | 120 | | dB |
| Crosstalk | | -80 | | dB |
| Input Impedance | | 15k | | Ω |

Microphone Input and Bias

| Parameter | Min | Typical | Max | Units |
|---|---|---|---|---|
| Input voltage | | 0.1 | | Vrms |
| SNR | | -110 | | dBFS |
| THD_N | | -80 | | dBFS |
| Crosstalk | | -110 | | dB |
| Bias voltage 80% | 2.6 | 2.75 | | V |

| | | | | |
|---|---|---|---|---|
| Bias voltage 50% | | 1.95 | 2.1 | V |
| Bias noise | | -106 | | dBV |

Gain Range and Step Size

| Parameter | Min | Typical | Max | Units |
|---|---|---|---|---|
| Gain range (output) | -74 | | 6 | dB |
| Gain range (input) | 0 | | 36 | dB |
| Gain range (ADC) | -74 | | 6 | dB |
| Gain range (PC-Beep) | -32 | | -4 | dB |
| Step size (output) | | 1 | | dB |
| Step size (input) | | 12[1] | | dB |
| Step size (ADC) | | 1 | | dB |
| Step size (PC-Beep) | | 4 | | dB |
| Step size Accuracy | -0.3 | | 0.3 | dB |
| Inter-channel gain mismatch | | .2 | | dB |

Reference Voltage

| Parameter | Min | Typical | Max | Units |
|---|---|---|---|---|
| Vref_A | | AVDD /2 | | V |
| Output noise | | TBD | | NV/sqrt(Hz) |

Jack Sensing

| Parameter | Min | Typical | Max | Units |
|---|---|---|---|---|
| Resistor tolerance | | 1 | | % |
| Pull-Up | | 5.1 | | kΩ |
| PORT A | | 39.2 | | kΩ |
| PORT B | | 20.0 | | kΩ |
| Port C | | 10.0 | | kΩ |
| | | 5.1 | | kΩ |
| De-bounce time | 10 | 50 | 80 | mS |

Phase-lock Loop

| Parameters | Min | Typical | Max | Units |
|---|---|---|---|---|
| Input Frequency | 6.0 | 24.0 | 24.0 | MHz |
| Input Frequency accuracy | -75 | | 75 | ppm |
| Input signal duty cycle | 42 | | 58 | % |
| Input signal jitter | | 150 | 500 | pS |
| PLL Output Frequency | | 24.576 | | MHz |
| Jitter | | 750 | | pS, rms |
| Lock time | | | 80 | uS |

1 6dB steps implemented by PGA and available through test bit(s)

## 1.5 Temperature and Power

| Parameter | Min | Typical | Max | Units |
|---|---|---|---|---|
| Temperature Range (storage) | | | 125 | C |
| Temperature Range (operating) | 0 | | 70 | C |
| Analog Supply voltage (AVDD) | 1.7 | 1.8 | 1.9 | Volts |
| Digital Supply voltage (DVDD) | 1.7 | 1.8 | 1.9 | Volts |
| Host interface Supply voltage (DVDDIO) | 1.418 | 1.8 | 3.465 | Volts |
| Speaker Amplifier Supply voltage (PVDD) | 2.7 | 5 | 5.5 | Volts |
| Power consumption (analog) | | TBD | | mW |
| Power consumption (digital) | | TBD | | mW |
| Audio digital supply current (Normal operation) | | TBD | | mA |
| Audio digital supply current (Power down) | | TBD | | mA |
| Audio analog supply current (Normal operation) | | TBD | | mA |
| Audio analog supply current (Power down) | | TBD | | mA |
| Analog reference voltage | | TBD | | Volts |
| | | | | |

**Power Targets**

| Mode Description | Desired | Required |
|---|---|---|
| Idle power with no headphone, no clock | <200uW | |
| Idle power with no headphone, MHDA clock | <500uW | |
| Idle power with headphone inserted, no clock | <200uW | |
| Idle power with headphone jack inserted, MHDA clock | <500uW | |
| Non active with jack inserted, no microphone or button detection enabled. | <850uW | |
| Non active with jack inserted, button detection | <5mW | |
| Non active, lowest power state | | <500uW |
| D0 MHDA to headphone 32 ohm load (quiescent) | | 15mW |
| D0 MHDA to headphone 32 ohm load (1mW/ch 32 ohm load) | | 45mW |
| D0 MHDA to/from Headset, mic & headphone (1mW into 32 ohm load) | 60mW | |

**Notes on system voltages:**
1.8 for digital is always present
VDDIO for host signaling always provided from the system.
Vbat ranges from 3.3 to 19 volts depending on the platform.
It is okay to request that 5.0V or 12V volts be generated for the chip
In that case the 5.0V or 12V generated will always be present

When not directly connected to the system battery, AVDD5 and PVDD are not guaranteed to be active in low power modes. Typically only the 1.8V supply for the digital core and host interface (interrupts and control communication) will be active with all other supplies off.

## 1.6 GS Mark support

Oceanside provides programmable digital hardware limiting approach implemented in Palm and Vilano for the headphone output path. The headphone may be configured for a setting where a full-scale DAC output will be slightly less than 450mV at the output into 32 ohms, about 7dB lower than the nominal target of 1Vrms. The limit may be set between -3dBFS and -10dBFS. This gain setting is not reported in the pin widget capabilities, it is controlled by a vendor defined verb, and may be locked to prevent adjustment after BIOS configuration.

## 1.7 Speaker Reference Return Stream

When implementing Acoustic Echo Cancellation (AEC) it is important to have access to the output audio stream as close to the speaker as possible. Non-linear processing such as DRC prevents us from simply using the same audio data that was sent to the CODEC for any host based processing. Therefore a special reference stream is provided.

The return stream has the same rate and format as the chosen output stream and has all of the processing applied that the output pin would experience. A method for programming an independent input stream tag (HD Audio host interface mode) or slot numbers (I2S/I2C interface mode programmed for TDM format) is required but the rate and format may be taken from the source stream. It is important to reference the output port rather than the input stream because one stream may be sent to multiple ports with different processing. For example, a stereo stream may be selected for playback on the headphone and speaker ports but the speaker port has EQ/DRC processing and not the headphone port. Any output capable port may be selected, and not just the BTL speaker port, because an external power amplifier may be connected to any output port.

The needed verbs should be implemented in the vendor defined widget (Node 1B: Vendor Specific Widget) associated with output processing.

## 1.8 Dual Mode Host Interface

### 1.8.1 Overview

The host interface can operate in two distinct modes; HD Audio mode, and I2S mode. The default interface mode is selected by a bond option. The mode may be changed by command using the HD Audio interface or the I2C interface. It is expected that the bus mode will be changed only when the bus is idle (no active audio streams). Clocks may or may not be active when the request is made.

### 1.8.2 HD Audio mode

In HD Audio mode, the host interface will be compliant with the HD Audio specification 1.0a and ECR HDA048-A. The bus interface is comprised of BCLK, SYNC, SDI, SDO, and an optional RESET# pin. The I2C host interface is available but command and control is expected on the HD Audio interface. Although not expected, in the case that both the I2C interface and the HD Audio interface access the CODEC at the same time, both interfaces shall meet their respective timing requirements.
I2S mode

### 1.8.3 I2S Mode

In I2S mode, the host interface will use the I2S or TDM protocol to transfer audio data to and from the host. A slave I2C compatible interface is used for control of the CODEC.
The host interface is comprised of MCLK, BCLK, LRCLK, SDI, SDO, SDA, and SCL pins. I2S, Left Justified, Right Justified, and Intel TDM formats are supported. Master and Slave modes are supported. The default is 48KHz I2S format with 16-bit data.

MCLK is the master, high speed clock provided by the host. A range of about 5.5 to 50 Mhz is desired. It is usually, but not always, an integer multiple of the sample rate. See the PLL section for more detail.

BCLK is the bit, or shift, clock used to for data recovery. This is typically 64x the sample rate and is configurable. BCLK is an input in slave mode and an output in master mode.

LRCLK is the frame clock and is typically at the sample rate of the stream. TDM modes may allow multiple samples per channel per frame. LRCLK is an input in slave mode and an output in master mode. Both short and long sync profiles are supported. The start of frame may be programmed to be on the rising or falling edge. In addition, the sync, at a minimum, is programmable for a 0 or 1 BCLK delay to support I2S and TDM modes.

**Intel requirements.**
For some Intel platforms, the Master clock will be 25, 24, or 19.2MHz. The CODEC will be in slave mode. BCLK will be 2.40MHz. There will be 50 clocks per frame and the frame rate will be 48KHz. The LRCLK and BCLK will be an exact multiple of 48KHz when MCLK is 25MHz. Intel Haswell had about an 8ppm error on the sample rate. (At 25MHz MCLK, we believe that the competition uses the 2.4MHz BCLK for their PLL. For 19.2MHz, 24MHz, and 25MHz the PLL is set to 1.2MHz for the PFD and the VCO is 98.4MHz)

**I2C Host interface**
The I2C host interface is a slave interface comprised of the HSCL (Host I2C SCL) and HSDA (Host I2C SDA) pins. These are dedicated pins that interface with the host SOC or other controller.
The interface is compatible with I2C and SMBus and supports at least 100KHz and 400KHz data rates. The default address may be re-programmed by using a register/verb access that takes effect in the next bus cycle. The default address is 7h33 (0110011R/nW) but may be changed to 7h37 (0110111R/nW) through a bond option.

1) The I2C slave interface is always active
   a) This allows a guaranteed means of switching bus modes.
   b) It may be difficult to manage disabling access where we won't create a condition where we get stuck in one mode or another.

2) All commands are available to the I2C and HD Audio interfaces.
3) Typically, one control interface is active while the other is idle or off.
   a) In I2S mode, only the I2C interface can be active considering that we have not defined a means of transferring control data across the TDM/I2S interface.
   b) In HD Audio mode both can be active.
4) If a request comes from both the I2C and HD Audio interfaces at the same time:
   a) Both interfaces will meet their timing requirements. Valid data must be read.
   b) No attempt will be taken to prevent one interface from overwriting data from the other. Accesses to a register are on a first come first served basis. In the case of an absolute tie, it is expected that the I2C access will be held off until the HD Audio access is complete due to the more restrictive HD Audio timing requirements.

## 1.9 Clocks and PLL

An internal PLL provides a high-speed master clock that is used for processing and conversion. The nominal VCO frequency is 147.456MHz (3072x 48KHz) to reduce power consumption. Control interfaces such as the I2C and HD Audio blocks should not depend on the PLL output if possible to allow shutting off the PLL in low power modes and still allow resuming from a low power state by using a command. When the host interface is in a low power state, there will be no active clock. Therefore, an internal oscillator is provided. This oscillator is imprecise but may be calibrated when a clock input is active.

In HD Audio mode, the BCLK input supplies the clock to the CODEC and may be 24MHz, 12MHz, or 6MHz. SYNC will be exactly 48KHz (24MHz/500, 12MHz/250, 6MHz/125). In I2S mode, the MCLK input supplies the clock to the CODEC and may be one of many possible rates.

For Intel based systems, the most common rates are: 25MHz, 24MHz, and 19.2MHz. For systems providing an MCLK of 24MHz or 19.2MHz, BCLK will be 2.4MHz (MCLK/10 or MCLK/8, respectively) and LRCLK will be exactly 48 KHz (BCLK/50).

For other SOCs, the clock may be a common multiple of the sample rate (256, 384, 512 times Fs, for example) or a common communications based rate (48MHz, 26MHz, 25MHz, 19.2MHz, etc.

Although desirable, it is not possible to convert all input rates to an exact multiple of 48KHz. The rate error should be less than 20ppm when an exact rate can't be provided.

| | *Input Ref Freq (MHz)* | *Pre-Div* | *Pre Scaler* | *Loop Div* | *Effective Loop Div* | *Phase Comp Freq (MHz)* | *VCO Frequency (MHz)* | *MODE OF OPERATION* |
|---|---|---|---|---|---|---|---|---|
| ***HD Audio Mode (PLL Works as a Frac PLL)*** | *24* | *4* | *1* | *24.576* | *24.576* | *6* | *147.456* | *FRAC_MODE* |
| | *12* | *2* | *1* | *24.576* | *24.576* | *6* | *147.456* | *FRAC_MODE* |
| | *6* | *1* | *1* | *24.576* | *24.576* | *6* | *147.456* | *FRAC_MODE* |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | *24* | *4* | *1* | *22.5792* | *22.5792* | *6* | *135.4752* | ***FRAC_MODE*** |
| | *12* | *2* | *1* | *22.5792* | *22.5792* | *6* | *135.4752* | ***FRAC_MODE*** |
| | *6* | *1* | *1* | *22.5792* | *22.5792* | *6* | *135.4752* | ***FRAC_MODE*** |
| ***I2S Mode (PLL Works as a Frac PLL)*** | *19.2* | *3* | *1* | *23.04* | *23.04* | *6.4* | *147.456* | ***FRAC_MODE*** |
| | *24* | *4* | *1* | *24.576* | *24.576* | *6* | *147.456* | ***FRAC_MODE*** |
| | *25* | *4* | *1* | *23.59296* | *23.59296* | *6.25* | *147.456* | ***FRAC_MODE*** |
| | *26* | *4* | *1* | *22.685538* | *22.68554* | *6.5* | *147.456* | ***FRAC_MODE*** |
| | *27* | *5* | *1* | *27.306667* | *27.30667* | *5.4* | *135.4752* | ***FRAC_MODE*** |
| | *36* | *6* | *1* | *24.576* | *24.576* | *6* | *147.456* | ***FRAC_MODE*** |
| | *48* | *8* | *1* | *24.576* | *24.576* | *6* | *147.456* | ***FRAC_MODE*** |
| ***I2S Mode (PLL Works in Integer Mode)*** | *1.536* | *1* | *2* | *48* | *96* | *1.536* | *147.456* | ***INT_MODE*** |
| | *2.304* | *1* | *2* | *32* | *64* | *2.304* | *147.456* | ***INT_MODE*** |
| | *3.072* | *1* | *1* | *48* | *48* | *3.072* | *147.456* | ***INT_MODE*** |
| | *4.608* | *1* | *1* | *32* | *32* | *4.608* | *147.456* | ***INT_MODE*** |
| | *6.144* | *1* | *1* | *24* | *24* | *6.144* | *147.456* | ***INT_MODE*** |
| | *6.912* | *3* | *2* | *32* | *64* | *2.304* | *147.456* | ***INT_MODE*** |
| | *9.216* | *2* | *1* | *32* | *32* | *4.608* | *147.456* | ***INT_MODE*** |
| | *12.288* | *2* | *1* | *24* | *24* | *6.144* | *147.456* | ***INT_MODE*** |
| | *13.824* | *3* | *1* | *32* | *32* | *4.608* | *147.456* | ***INT_MODE*** |
| | *16.384* | *3* | *1* | *27* | *27* | *5.4613* | *147.456* | ***INT_MODE*** |
| | *18.432* | *4* | *1* | *32* | *32* | *4.608* | *147.456* | ***INT_MODE*** |
| | *24.576* | *4* | *1* | *24* | *24* | *6.144* | *147.456* | ***INT_MODE*** |
| | *27.648* | *6* | *1* | *32* | *32* | *4.608* | *147.456* | ***INT_MODE*** |
| | *36.864* | *6* | *1* | *24* | *24* | *6.144* | *147.456* | ***INT_MODE*** |
| | *49.152* | *8* | *1* | *24* | *24* | *6.144* | *147.456* | ***INT_MODE*** |
| | | | | | | | | |
| | *1.4112* | *1* | *2* | *48* | *96* | *1.4112* | *135.4752* | ***INT_MODE*** |
| | *2.1168* | *1* | *2* | *32* | *64* | *2.1168* | *135.4752* | ***INT_MODE*** |
| | *2.8224* | *1* | *1* | *48* | *48* | *2.8224* | *135.4752* | ***INT_MODE*** |
| | *4.2336* | *1* | *1* | *32* | *32* | *4.2336* | *135.4752* | ***INT_MODE*** |
| | *5.6448* | *1* | *1* | *24* | *24* | *5.6448* | *135.4752* | ***INT_MODE*** |
| | *6.3504* | *3* | *2* | *32* | *64* | *2.1168* | *135.4752* | ***INT_MODE*** |
| | *8.4672* | *2* | *1* | *32* | *32* | *4.2336* | *135.4752* | ***INT_MODE*** |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | *11.2896* | *2* | *1* | *24* | *24* | *5.6448* | *135.4752* | *INT_MODE* |
| | *22.5792* | *4* | *1* | *24* | *24* | *5.6448* | *135.4752* | *INT_MODE* |
| | *28.224* | *5* | *1* | *24* | *24* | *5.6448* | *135.4752* | *INT_MODE* |
| | *45.1584* | *8* | *1* | *24* | *24* | *5.6448* | *135.4752* | *INT_MODE* |

## 1.10 GPIO

The GPIO pins have multiple functions. Most pins have at least two functions.

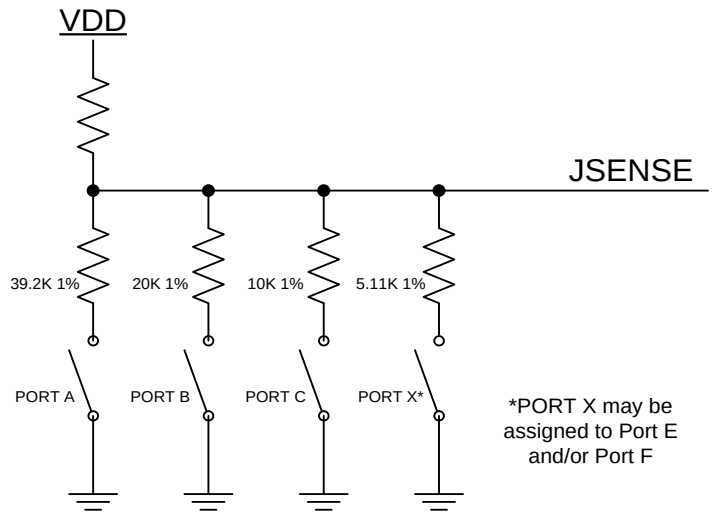
| Pin | DMIC Data | DMIC CLK | D3Live | SPDIF Out | SPKR_MUTE# | Mic Mute | INT Out |
|---|---|---|---|---|---|---|---|
| GPIO 0 | | DMIC2 | default | opt | | opt | opt |
| GPIO 1 | DMIC2 | | | opt | default | opt | opt |
| GPIO 2 | | Default (DMIC1) | | opt | | opt | opt |
| GPIO 3 | Default (DMIC1) | | | opt | | opt | opt |
| GPIO4 | | Default (DMIC2) | | | | | opt |
| GPIO5 | Default (DMIC2) | | | | | | opt |

General rules on GPIO use:

1) GPIO functionality is disabled by default
2) If a GPIO is enabled, the default function is disabled unless compatible with the GPIO setting or critical for safety or security.
    a. SPKR_MUTE#, INT, and D3Live may coexist with the GPIO configured as an input.
    b. Mic Mute, if enabled will override all other functions. The pin status is provided in a separate register (verb) so the use of a GPI is not required.
    c. DMIC and SPDIF functions are not considered compatible with GPIO.
3) If GPIO is enabled on a DMIC data pin, that DMIC path will be disabled and the record stream will be mute (zero data)
4) If a GPIO is enabled on a DMIC clock pin, any DMIC path using that clock pin will be disabled and the stream will be mute (zero data) (on parts with more than one clock pin, all DMIC data inputs may be configured to use independent clocks or share a common clock)
5) If an attempt is made to configure the same pin as more than one function other than GPIO, the following rules of precedence are suggested:
    a. Mic Mute has first priority
    b. D3 Live and SPKR_MUTE# have second priority
    c. INT has third priority
    d. SPDIF out has fourth priority

## 1.11 Jack Sense

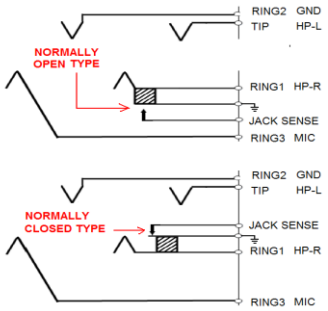
The multi-level jack sense circuit allows sensing of up to 4 jacks using one pin. Each jack is assigned a unique resistor value and must implement an electrically isolated switch (isolated from the audio pins) that is normally open. A pull-up resistor is used to bias the input. When a plug is inserted into a jack, the switch inside the jack closes and connects a resistor attached to the JSENSE pin to ground. An ADC inside the CODEC senses the change in voltage at the pin and determines how many switches are closed. To avoid erratic behavior during plug insertion or removal, a debounce filter is implemented to ignore glitches. An option to support normally closed as opposed to normally open switches is provided on a per jack basis.




Headset jacks with JS switches

NO – NORMALLY OPEN

NC – NORMALLY CLOSED




## 1.12 Headset Inline Control Support

Communications headsets commonly provide control buttons integrated into the headset. The most common implementation is a single button that shorts the microphone input to ground. This button is typically used for call answer, call end, mute, and unmute depending on the duration of the button press. Many headsets provide two additional buttons for volume up/down and other functions. These buttons connect a resistance across the microphone to change the voltage measured on the mic pin. These resistances may be linear devices such as resistors or nonlinear devices such as diodes.

The CX82xx/CX83xx will implement support for single button detection or multi-button detection. The single-button detection circuit will detect short and long switch closures and generate unsolicited responses (including wake events) for button press and release events.

Multiple button support is targeted for 4 buttons as described by Google. The multi-button detection scheme should also support 3 buttons as an option. The detection circuit will detect short and long switch closures and generate unsolicited responses (including wake events) for button press and release events.

## 1.13 4 button support for Android

Google defined a 4 button interface for Andriod phones, tablets, and Chrome notebooks. (http://source.android.com/accessories/headset-spec.html)

The microphone itself is modeled as a 1K ohm or greater resistor. This is a reasonable model for common ECM microphones used in PCs and handsets. The CODEC is expected to provide a mic bias voltage between 1.8V and 2.9V for compatibility with common microphone cartridges.

Each button has a unique resistance that is placed in parallel with the microphone when the button is pressed.

| Resistor | Min | Nominal | Max |
|---|---|---|---|
| Microphone | 1KΩ | | |
| Button 1 | 0 | 1 | 70Ω |
| Button 2 | 210Ω | 240Ω | 290Ω |
| Button 3 | 360Ω | 470Ω | 680Ω |
| Button 4 | 110Ω | 135Ω | 180Ω |

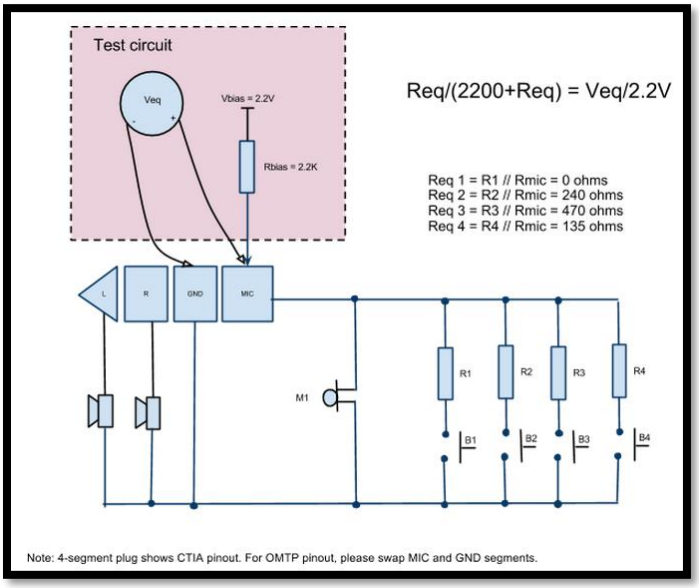
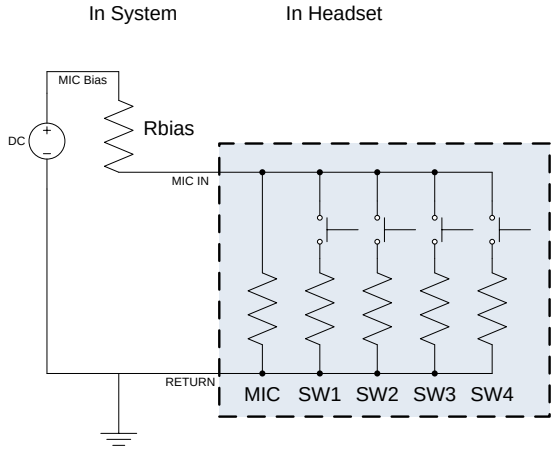



Figure 1-From Google Specification




### 1.13.1 Button Detect Process

The microphone input is monitored for a change in voltage that is larger than what is typically expected from audio. When a button is pressed or released, a message is sent to the host to tell software of the event if unsolicited responses or interrupts are enabled. Both button press and release events generate messages so that the host can distinguish between short and long button press, button hold, and multiple rapid press/release events.

The button# flag is only updated when a button press is detected and the detected value is stable for more than the debounce time. If the button is pressed longer than the long-press timout setting, the LP status bit is set. This allows software to distinguish if a button has not been released yet (DN flag set but UP flag not set); if a short press has occurred (DN and UP flags set but not LP); a long press has occurred (DN, UP, and LP flags set); or an error has occurred (UP flag set but no button flags set)

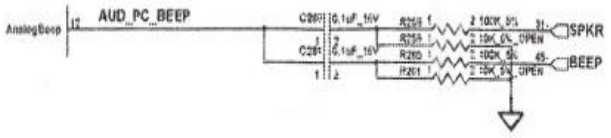
### 1.13.2 HEADPHONE TIP/RING IMPEDANCE DETECTION

In addition to monitoring buttons, the CODEC is required to determine the impedance of the speakers on the tip and ring. A value of less than 1K ohms is reported by software as headphones and a value greater than 1K ohms is reported as a line output.
It is assumed that headsets have low impedance speakers and we only need to detect impedance when a 3-conductor plug is inserted. After we have determined that a 3-conductor plug has been inserted, we sense the impedance. (Note, Google does not support input capabilities on the headset jack. Inserting a plug that is an output from another device will be detected as a low impedance load (headphones) even though it is an audio source and the port should be configured as a line input. This is OK for Google.)

## 1.14 PC BEEP

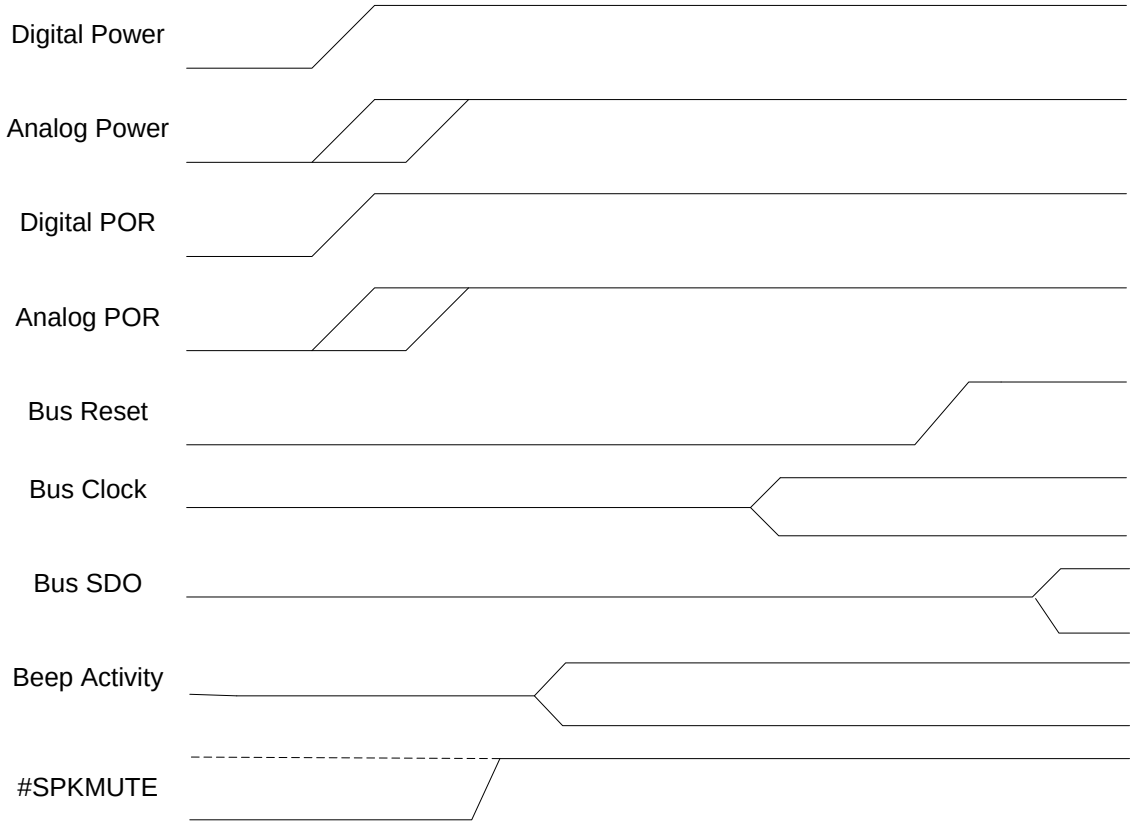
PC beep requirements.

1) At the beep pin be able to mix two external sources of beep. An example of a typical analog beep mixer is included. This functionality must be maintained.



2) If the input of PC beep is not toggling, set high or low, no noise will be injected into the output ports, output ports will return to their pre-beep power state.
3) If a coupling cap is used then at power up when the coupling cap is charging to the internal reference voltage no noise will be injected in the ports and no false activation of output ports will occur.
4) If an output port is in a low power state, unable to pass beep, but its enable bit is set, activity on the PC beep pin, input crossing both a high and low threshold, must "wake" the port and beep must pass through. A signal input of at least 200mVpp is needed to trigger the activity detector.
5) A PC beep level is exposed to the OS from the beep generator widget for user control. This level has the range -4db to -32db, 8 steps of -4db each. This level shall be used to determine the beep level for I2S and for the Class-D. Default is -20db
6) A second level control, in the BIOS configuration area, takes the output of the exposed beep level and generates the beep level for the headphone port. This level has the range of 0db to -32db, 8 steps of -4db each. This allows the headphone level to be equivalent to the Class-D setting for line out and below the class-D setting for headphone. Default is -24db below the class-D setting. Because the output of the exposed beep level is used as the starting point the headphone beep level will track the user exposed setting. It is suggested that this level be done in analog at the headphone port so that DAC tracking and multiple ports connected to the same DAC don't cause beep level conflicts or confusion.
7) Since the beep path to the analog ports is through analog the digital beep generator signal must be sent down to analog and mixed with the PC beep input pin.
8) The signal from the beep input pin must also be sent to the digital beep path for sending on to I2S. This should be done post mixing of the beep pin and the digital beep generator so that there is only one unified beep signal used by both the analog and digital paths.
9) PC beep should be sent to the class-D during sleep and music mode.
10) The BIOS selectable modes associated with PC beep are as follows.
    a. The ability to replace an existing stream with beep at the output port, I.E. mux and the ability to add beep to the existing stream, I.E. mix, must be BIOS controllable. Either setting may be used as POR default because no streams are active during boot; therefore there is no difference between mix and mux at boot. The BIOS can switch modes later if desired.
    b. A control option to have the DAC level influence the PC beep level at the port or have the beep level fixed by the beep gain setting.
    c. A control option to force the beep input pin to be disabled while the HD audio bus is out of reset. This bit must not affect the behavior of the digital beep generator – only the exposed beep pin is disabled. This may be called the "compliance mode" as this is the required behavior by HD audio specification. This mode is not the reset default however.
    d. A control option to disable the beep path in analog, I.E. power down. There may be cases where no beep is wanted in any state. In this case a beep path power down will allow the most power savings.

### 1.14.1 Case 1: Power up & Beep signal sequencing




At digital POR the port enable bit for both the Class-D amplifier and headphone out will be set. All widgets, this includes muxes, mixers, DACs, ADCs and ports, are placed in the D3 hot (10ms recovery) state. Based on the presence of the PC-Beep signal and EAPD the enabled ports should quickly move to a state that can support PC-beep pass through. In this mode PC-Beep is routed through analog.

### 1.14.2 Case 2: Beep S0 and device in D0

In this case the system is fully up. Digital and analog voltages are available. Any assigned DACs and ports are powered up. PC-beep shall be routed through analog to all enabled ports and routed through digital to I2S out if I2S output is enabled.

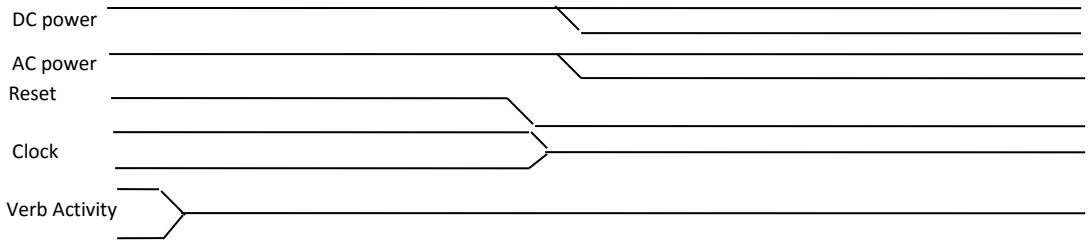
### 1.14.3 Case 3: Beep during S0 with the device in a lower power state

In this case the system is fully up. Digital and analog voltages are available the HD audio clock is present, reset is not active. Any output port that has its enable bit set must pass PC-beep; this includes I2S as long as the device is in D3 hot or higher power state.

### 1.14.4 Case 4: Beep during resume from suspend or hibernate

This looks exactly like the power up case accept that analog and digital POR may or may not occur. This is because analog and digital power may or may not have gone away.

### 1.14.5 Power down signals



## 1.15 EAPD and Speaker Mute

EAPD and Speaker Mute are two functions that enable the control of internal and external amplifiers. Speaker Mute can provide a way for an external controller to mute the internal amplifier while EAPD is a signal originally intended to control external amplifiers attached to an HD Audio CODEC. These two signals, however, may be configured in several different modes to provide flexibility in system implementation.

**EAPD Pin**

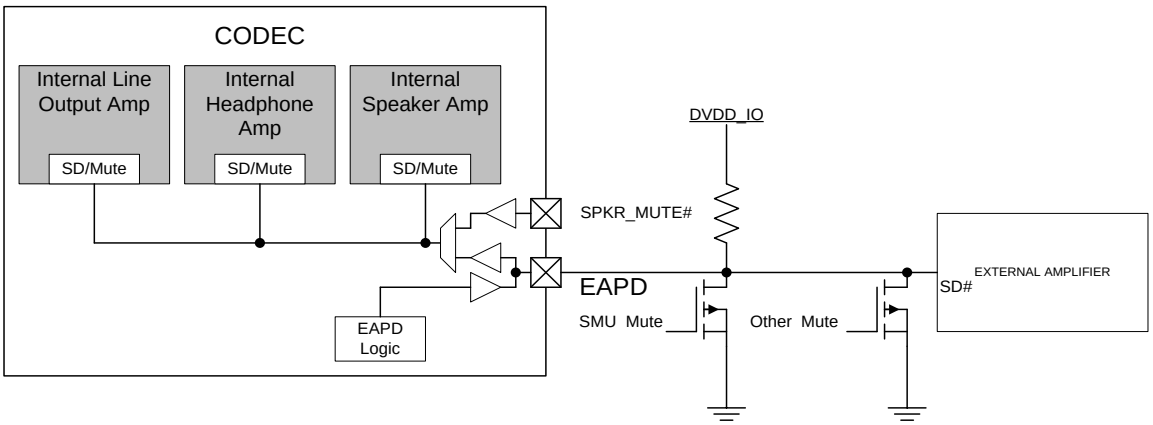
The EAPD pin is a dedicated, bi-directional control pin. As described in the HD Audio specification, EAPD (External Amplifier Power Down) is a logical construct that allows the control of an external amplifier. The EAPD value is reflected on the EAPD pin; a 1 causes the external amplifier to power up (equivalent to D0), and a 0 causes it to power down (equivalent to D3.) When the EAPD value = 1, the EAPD pin must be placed in a state appropriate to the current power state of the associated Pin Widget even though the EAPD value (in the register) may remain 1. The default state of this pin is 0 (driving low.) The pin defaults to an open-drain configuration (an external pull-up is recommended.)

Per the HD Audio specification and HDA015-B, multiple ports may control EAPD. The EAPD pin assumes the highest power state of all the EAPD bits in all of the pin complexes. The default value of EAPD is 1 (powered on), but the FG power state will override and the pin will be low. A port will request External Amp Power Up when its power state is active (FG and pin widget power state is D1 or D0) or (Analog PC_Beep is enabled and port is enabled as an output) and the port's EAPD bit is set to 1. The state of the EAPD pin (unless configured as an input or held low by an external circuit when configured as an open drain output) will be the logical OR of the external amp power up requests from all ports.

By default, the EAPD pin also functions as the shutdown input for the internal BTL amplifier. In this mode, a low value at the pin (either due to internal EAPD being 0, or to an external entity forcing the pin low) will cause the internal BTL amplifier to mute or enter a low power state depending on the amplifier configuration. (See below)

Vendor specific verbs are available to configure this pin. These verbs retain their values across link and single function group resets but are set to their default values by a power on reset:

| EAPD MODE1 | EAPD MODE0 | EAPD Pin Function | Description |
|---|---|---|---|
| 0 | 0 | Open Drain I/O | Value at pin is wired-AND of EAPD bit and external signal. (default) |
| 0 | 1 | CMOS Output | Pin assumes value of EAPD bit in pin widget. |
| 1 | 0 | CMOS Input | External signal controls internal amps. EAPD bit in pin widget is ignored. |
| 1 | 1 | CMOS Input | External signal controls internal amps. EAPD bit in pin widget is ignored. |




**Speaker Mute Pin (SPKR_MUTE#)**

The SPKR_MUTE# pin may be used as an input to disable internal amplifiers or as a general purpose I/O pin. When configured as a GPIO, the logical state of SPKR_MUTE# is inactive. Any amplifier referencing the SPKR_MUTE# pin will behave as if the SPKR_MUTE# input were in a logical state that allows playback.

By default, the SPKR_MUTE# pin is configured as the Speaker Mute input but, to avoid problems when the system is designed to use the pin as a GPIO, no amplifiers are controlled by this pin. (See implementation details for how to use SPKR_MUTE# with EAPD to work around this limitation.)

**Amplifier Control**

The internal amplifiers, including the speaker amplifier, may be controlled by their port power state and enable. As described by the HD Audio specification and Microsoft Windows Logo Program documents, the output enable in the pin widget may act as a mute but is not intended to control the power state of the port. The power state is controlled by dedicated verbs. However, sometimes it is desirable to control the mute or power state of an internal amplifier by an external means. An example is Music Mode (D3 Live). In this mode, the host interface is inactive but simple audio playback from an external source is desired. Another example is analog PC beep. Analog PC Beep may be enabled before the source of PC Beep has been configured. To avoid pops or odd sounds, the host may use EAPD or SPKR_MUTE# to mute outputs.

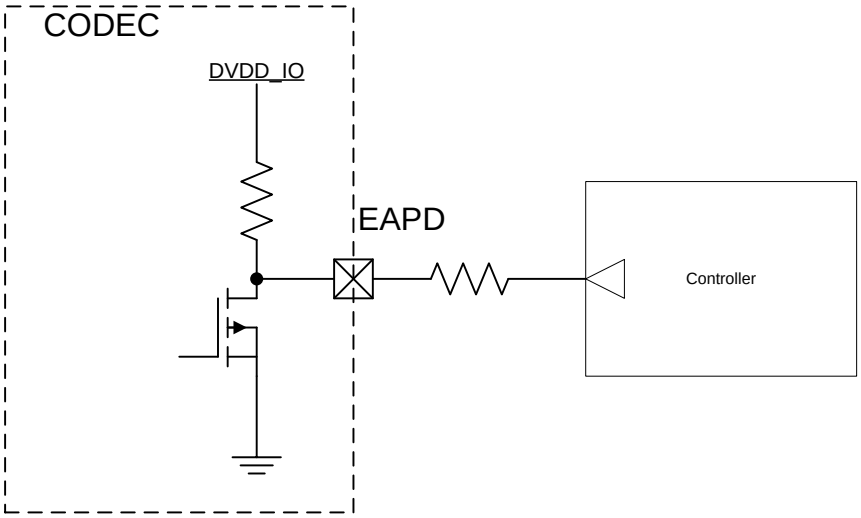
Each output amplifier may be controlled by their port controls, SPKR_MUTE#, or EAPD. This includes Port A (headphone amplifier), and Port G (Speaker amplifier).When responding to an external shutdown signal (EAPD or SPKR_MUTE#) the amplifier may be configured to mute or to turn off with either an active high or low signal. This is useful for situations when, for example, one desires to turn off the speaker amp (internal or external) when headphones on and turn off the headphones when the speakers are on.

| **Amp Control (replicated for each amp)** | **Description** |
|---|---|
| SD SEL | Selects how the port amplifier is controlled:<br>0 = Amp controlled by EAPD or SPKR_MUTE# pin (default) |

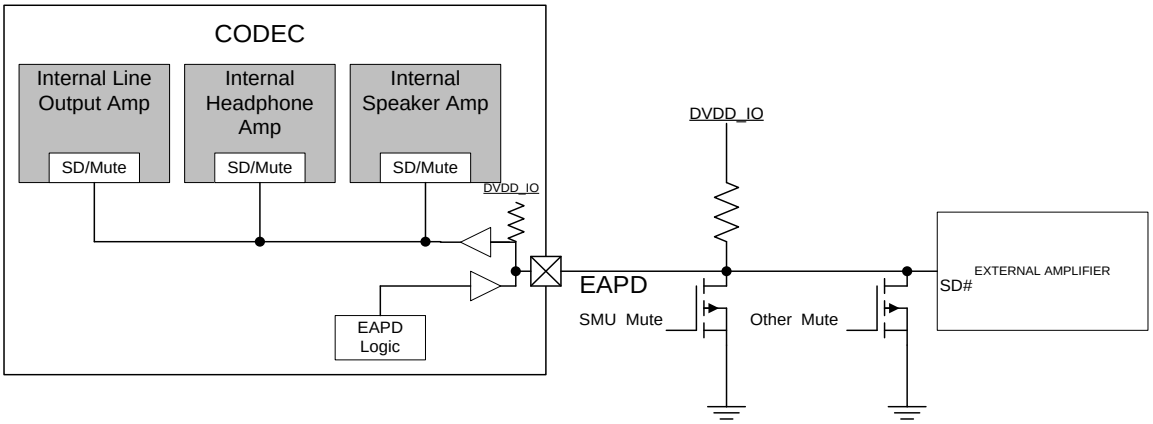
| | |
|---|---|
| | 1 = Amp controlled by pin widget control and power state |
| SD INV | Polarity when amp controlled by EAPD/SPKR_MUTE#<br>0 = Amp is disabled when input low (default)<br>1 = Amp is disabled when input is high |

System Implementation
Neither the EAPD pin nor the SPKR_MUTE# pin is needed for proper operation of the device. SPKR_MUTE# may be programmed for other functions and EAPD may be left unconnected.

When using the EAPD pin, it is important to note that it is configured as an open drain output by default. If configuring the system for EAPD as a CMOS input, use a resistor to prevent damage due to bus contention that might arise before the EAPD pin can be configured as an input.

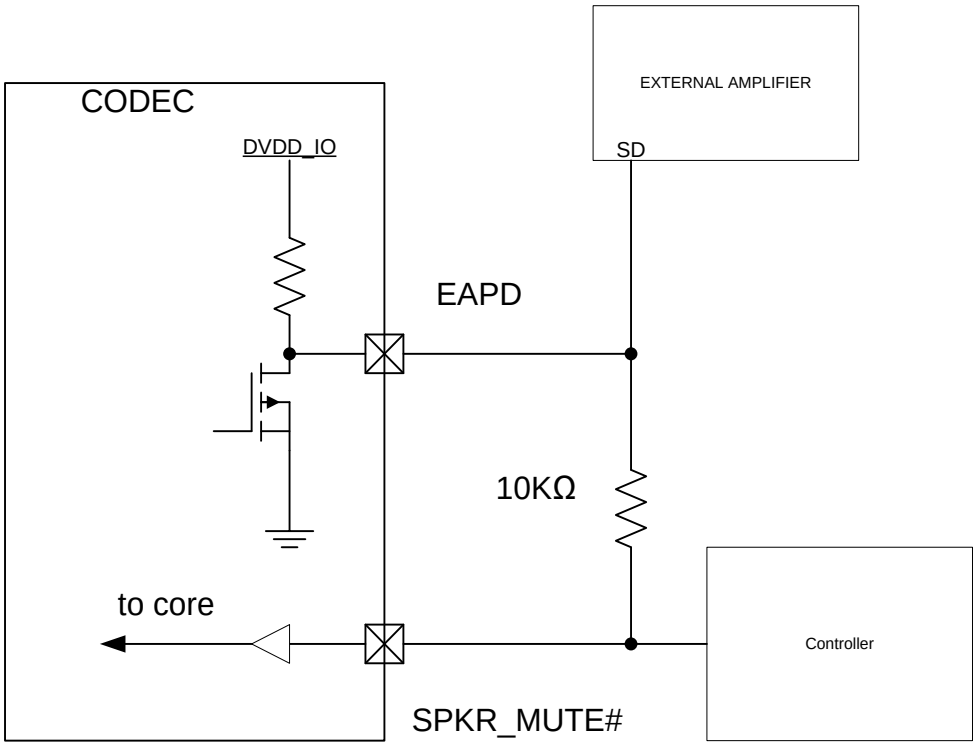



Multiple controllers may be connected in a wired AND configuration when the EAPD pin is in open drain or input mode. The state of the pin will be low unless all sources release the bus.




When implementing a system where SPKR_MUTE acts like a dedicated input and EAPD is a dedicated output (when replacing the CX20952 for example) the power-on default conditions of bi-directional EAPD and the SPKR_MUTE pin being ignored by the internal amplifiers present a challenge. To support

this use case, the SPKR_MUTE and EAPD pins may be connect together using a resistor as shown. After the application of power, the EAPD pin will control the external and internal amplifiers and take precedence over the external SPKR_MUTE controller. After configuration, the EAPD pin controls the external amplifier and SPKR_MUTE may control the internal amplifier.




## 1.16 Class-D Amplifier Gain

The class-D amplifier is designed to deliver more than 2W into a 4 ohm load. Typically, the gain for the amplifier is set to deliver 2W at low distortion (less than 1%) when the DAC PCM data is near full-scale. However, it is often desirable to set the effective gain for the DAC/class-D path higher so that average pre-recorded media will attain the rated output even though a full scale input will clip. At least 3 settings are required (gain relative to a digital full scale providing 1Vrms): ~9dB (2W), ~12dB (4W), and ~16dB (10W).

These gain settings are specifically required by Dell and desired by other customers. Dell testing procedures specify a known noise source with an average level of -10dB FS. A gain of 12dB will provide an average power of about 0.4W and is used with 0.5W speakers. A gain of 16dB will provide about 1W and is used with 1W speakers.

The following table represents the desired gain settings for the amplifier. These settings are not exposed to the operating system and are accessed using vendor defined verbs.

| Gain setting | Gain (dB) | Pout / 4 ohm load (max theoretical) |
|---|---|---|
| 0 | 16 | 10 |

| | | |
|---|---|---|
| 1 | 15 | 8 |
| 2 | 14.5 | 7 |
| 3 | 13.8 | 6 |
| 4 | 13 | 5 |
| 5 | 12.5 | 4.5 |
| 6 | 12 | 4 |
| 7 | 11 | 3.25 |
| 8 | 10 | 2.5 |
| 9 | 9 | 2 |
| 10 | 8.5 | 1.75 |
| 11 | 7.8 | 1.5 |
| 12 | 6 | 1 |
| 13 | 4.8 | 0.75 |
| 14 | 3 | 0.5 |
| 15 | 0 | 0.25 |

## 1.17 Battery Monitoring

Oceanside actively monitors the power supply to the class-D amplifier to compensate for direct connection to the system battery. As the battery voltage drops from the fully charged value to the discharged value, the battery monitor will adjust the Smart Amplifier Power Averaging circuit to accurately predict the power to the load (the circuit does not measure the voltage across the load directly. Rather, it uses the modulation depth and the measured PVDD voltage to determine the approximate power delivered to the load.)
The battery monitor circuit will measure the PVDD voltage and adjust the signal so that the power delivered to the load is within +/-50mW of the target value (100mW granularity).

## 1.18 Port C Analog Input Operation

By default, Port C is a stereo digital microphone interface. If desired, the port may be configured as an analog input port and used for analog microphone or line level inputs.
The port configuration, however, is not intended to be dynamic. The system firmware or driver should configure this port at power-on and it should remain in that mode during normal operation.

When in digital input mode, the port may connect to one or two digital microphones. The port is visible to both the ADC1 and ADC2 converter widgets.Microphone bias is not supported. Due to the nature of this configurable port, the microphone data input is a nominal 10K ohms.

When in analog input mode, the port may be used for both analog microphone or line inputs. The port is only supported by the ADC1 converter widget connection list and is not available to ADC2. Microphone bias is only available when port A and Port B are in merge mode. Otherwise, an external supply is required to power the analog microphone. (see next section.)

## 1.19 Microphone Bias Pin

The MICBIAS output pin services both Port B and Port C
The default condition for PortB is an independent input port with microphone capability while the default configuration for Port C is a digital microphone input. Port B may also be tied to port A using the “merge mode” and port C may be configured as an analog input. Since there is only one external microphone bias pin, this can create a conflict.

By default, the external microphone bias pin shall provide the bias for Port B (Port C being a digital microphone input). If Port B is configured for merge mode, the bias for port B will be supplied internally on resistors connected to the headphone Left and Right outputs. Port C may then use the micbias pin if it is configured as an analog input. If port B is not configured in merge mode, and port C is in analog input mode, the mic bias pin shall be controlled by the port B controls.

| Port B state | Port C state | Port B bias | Port C bias | MICBIAS PIN Controlled by |
|---|---|---|---|---|
| Independent | Digital mic | Pin | NA | PortB |
| Independent | Analog mic | Pin | Not supported (external supply needed) | PortB |
| Merged with PortA | Digital mic | Internal/Pin* | NA | PortC (if Port B not in use**)/PortB* |
| Merged with PortA | Analog mic | Internal/Pin* | Pin/External* | PortC (if Port B not in use**)/PortB* |

Note: * Test bit provided to force use of external pin for port B in merge mode.
Note: ** Port C controls pin unless Port B is selected for record by ADC1.

## 1.20 Power-off hum elimination

A common problem with headphone ports supporting both CTIA and OMTP configurations is hum when power is removed. This is due to the need for a supply to bias the grounding FETs and anti-pop FETs on. When there is no supply, the headphone left and right outputs, as well as the return pin, will float allowing noise to couple into any attached cable. To prevent hum, devices that are normally conducting (depletion mode FETs) are used to ensure a connection to ground for the headset signals (return at a minimum and left, right, and mic signals ideally.)

# 2 HD Audio Electrical Interface

## 2.1 Overview

This chapter defines the electrical characteristics and constraints of HD Audio link. It is divided into sections covering integrated circuit component & system specification and physical requirements. Each section contains the requirements that must be met by the respective product, as well as the assumptions it may make about the environment provided.

## 2.2 Signaling Environment

This section defines the electrical characteristics of HD Audio components for 3.3V, 1.8V, and 1.5V signaling schemes. The HD Audio components can be designed with standard CMOS I/O technology. Unless specifically stated otherwise, component parameters apply at the package pins; not at bare silicon pads nor at card edge connectors.

### 2.2.1 DC Specifications

The DC specifications of the 3.3V/1.8V/1.5V HD Audio components are summarized in Table 2-1.

**Table 2-1: 3.3V DC specification**

| Symbol | Parameter | Condition | Min | Max | Units | Notes |
|---|---|---|---|---|---|---|
| Vcc | Supply Voltage | 3.3V mode | 3.135 | 3.465 | V | |
| | | 1.8V mode | 1.70 | 1.90 | | |
| | | 1.5V mode | 1.418 | 1.583 | | |
| Vih | Input High Voltage | 3.3V mode | 0.65xVcc | | V | |
| | | 1.8/1.5V mode | 0.60xVcc | | | |
| Vil | Input Low Voltage | 3.3V mode | | 0.35xVcc | V | |
| | | 1.8/1.5V mode | | 0.40xVcc | | |
| Voh | Output High Voltage | Iout =-500uA | 0.9xVcc | | V | |
| Vol | Output Low Voltage | Iout =1500uA | | 0.10xVcc | V | |
| Iil | Input Leakage Current | 0<Vin<Vcc | | ±10 | uA | 1 |
| Cin | Input Pin Capacitance | | | 7. 5 | pF | |
| Lpin | Pin Inductance | | | 20 | nH | 2 |

NOTES:

1. For SDI buffers (or in general any bi-directional buffer with tri-state output), input leakage current also includes hi-Z output leakage.
2. This is a recommendation, not an absolute requirement. The actual value should be provided with the component data sheet.

### 2.2.2 AC Specifications

The output driver on the HD Audio electrical link must be able to deliver an initial voltage of at least Vil or Vih respectively at the receiver through the bus with known characteristic impedance and at the same time meeting signal quality requirements.

The minimum and maximum drive characteristics of HD Audio output buffers are defined by the V/I curves. Figure 2-1, Figure 2-2 and Table 2-2 describe the SDO buffer AC drive specification whereas Figure 2-3, Figure 2-4 and Table 2-3 describe the AC drive specification of the SDI buffers. The AC drive specification for SYNC, RST# and BCLK buffers is same as that of SDO. These curves should be interpreted as traditional 'DC' transistor curves with the following exceptions: 'DC drive point' is the only position on the curves at which steady state operation is intended, while the higher currents are only reached momentarily during bus switching transients. The 'AC drive point' (real definition of buffer strength) defines the minimum instantaneous current required to switch the bus.

Adherence to these curves should be evaluated at worst case conditions. Minimum pull up curve is evaluated at minimum Vcc and high temperature. Minimum pull down curve is evaluated at minimum Vcc and high temperature. The maximum curve test points are evaluated at maximum Vcc and low temperature. Inputs must be clamped to both ground and power rails. The clamp diode characteristics are also listed here for reference.

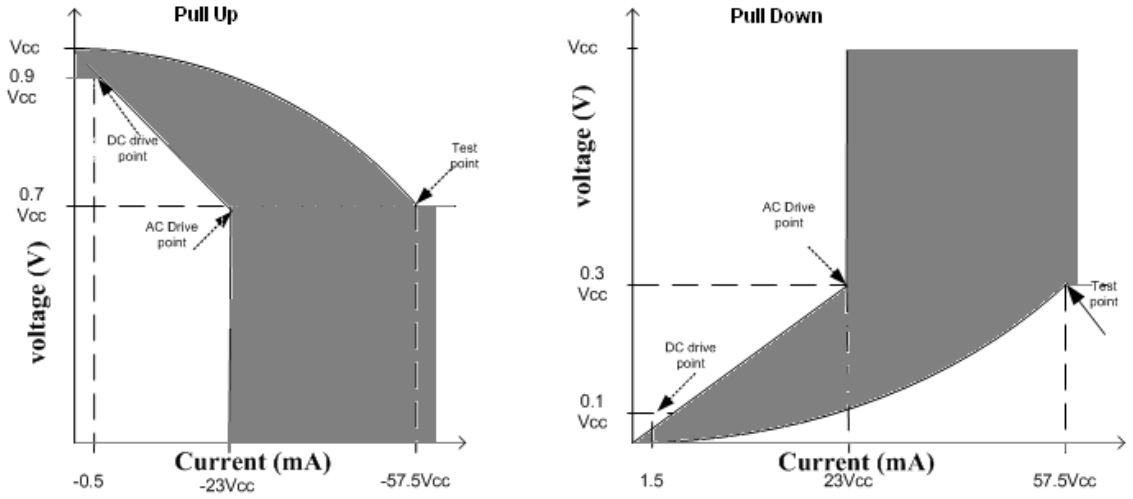



**Figure 2-1: V/I Curves for SDO buffers at 3.3V**

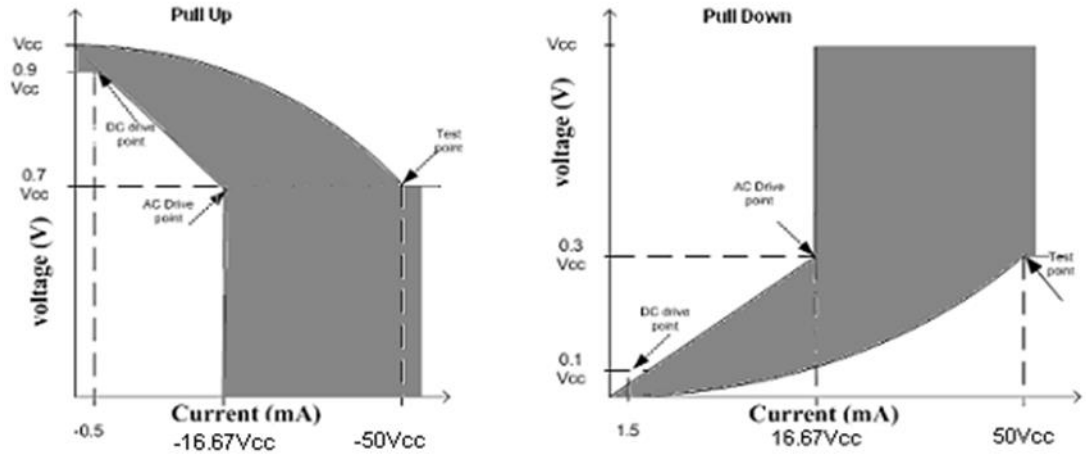



**Figure 2-2: V/I Curves for SDO buffers at 1.8V or 1.5V**

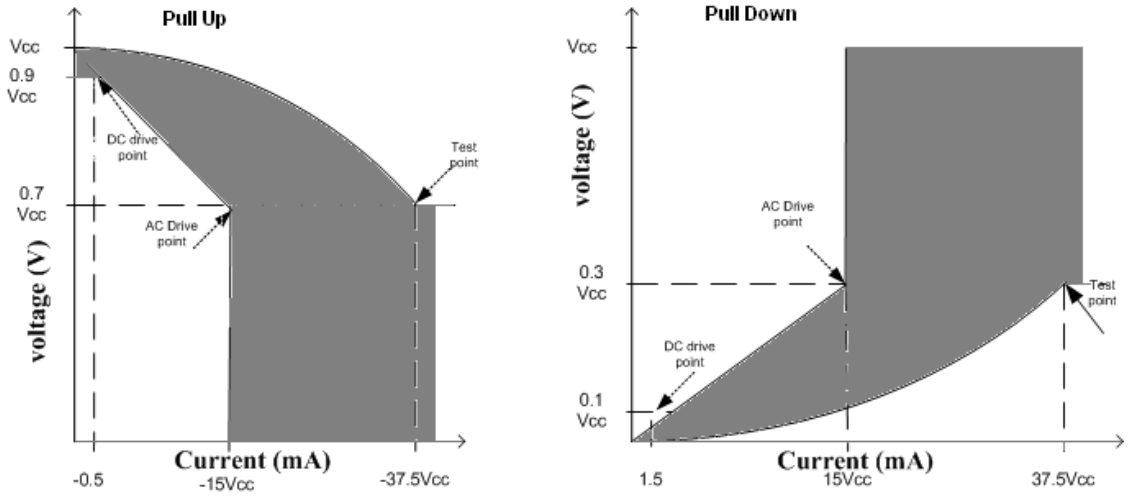



**Figure 2-3: V/I Curves for SDI buffers at 3.3V**

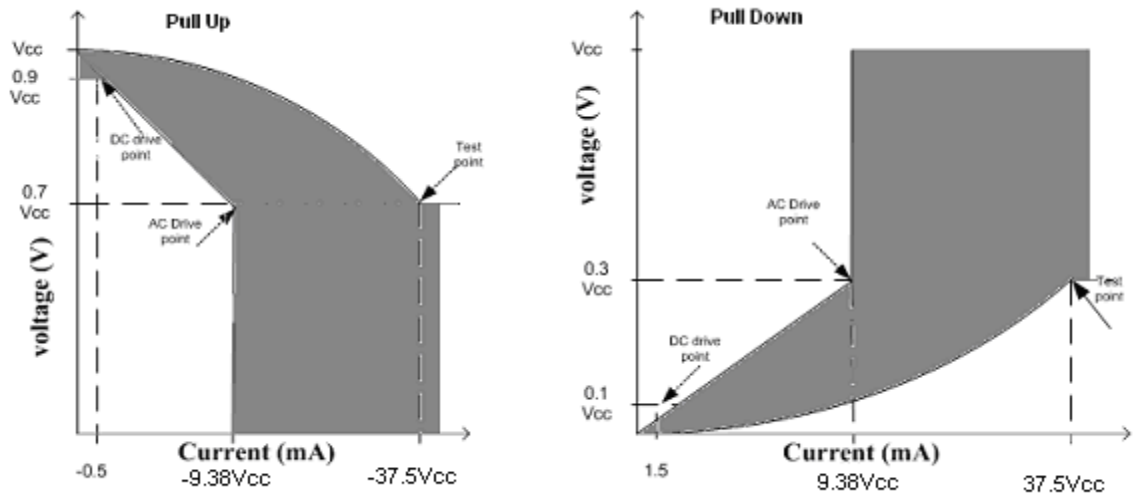



**Figure 2-4: V/I Curve for SDI buffer at 1.8V or 1.5V**

**Equation A**
Ioh = (174.2/Vcc)*(Vout-Vcc)*(Vout+0.4Vcc); for Vcc > Vout > 0.7 Vcc
**Equation B**
Iol = (273.8/Vcc)*Vout*(Vcc-Vout); for 0v < Vout < 0.3 Vcc
**Equation C**
Ioh = (113.6/Vcc)*(Vout-Vcc)*(Vout+0.4Vcc); for Vcc > Vout > 0.7 Vcc
**Equation D**
Iol = (178.6/Vcc)*Vout*(Vcc-Vout); for 0v < Vout < 0.3 Vcc
**Equation E**
Ioh = (151.52/Vcc)*(Vout-Vcc)*(Vout+0.4Vcc); for Vcc > Vout > 0.7 Vcc
**Equation F**
Iol = (238.1/Vcc)*Vout*(Vcc-Vout); for 0v < Vout < 0.3 Vcc
**Equation G**
Ioh = (113.64/Vcc)*(Vout-Vcc)*(Vout+0.4Vcc); for Vcc > Vout > 0.7 Vcc
**Equation H**
Iol = (178.57/Vcc)*Vout*(Vcc-Vout); for 0v < Vout < 0.3 Vcc

**Table 2-2: SDO buffer AC specification**

| Symbol | | Parameter | Condition | Min | Max | Units |
|---|---|---|---|---|---|---|
| Ioh (AC) | 3.3V | Switching Current high | 0 < Vout < 0.7vcc | -23Vcc | Eqt'n A | mA |
| | | | 0.7Vcc < Vout < 0.9Vcc | -76.7(Vcc-Vout) | | |
| | | | 0.7Vcc < Vout < Vcc | | | |
| | | (Test Point) | Vout = 0.7Vcc | | -57.5Vcc | |
| | 1.8V 1.5V | Switching Current high | 0 < Vout < 0.7vcc | -16.67Vcc | Eqt'n E | mA |
| | | | 0.7Vcc < Vout < 0.9Vcc | -55.57(Vcc-Vout) | | |
| | | | 0.7Vcc < Vout < Vcc | | | |
| | | (Test Point) | Vout = 0.7Vcc | | -50Vcc | |
| Iol (AC) | 3.3V | Switching Current Low | Vcc > Vout >0.3Vcc | 23Vcc | Eqt'n B | mA |
| | | | 0.3Vcc > Vout > 0.1Vcc | 76.7Vout | | mA |
| | | | 0.3Vcc > Vout > 0 | | | mA |
| | | (Test Point) | Vout = 0.3Vcc | | 57.5Vcc | mA |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 1.8V<br>1.5V | Switching<br>Current Low | Vcc > Vout >0.3Vcc | 16.67Vcc | | mA |
| | | | 0.3Vcc > Vout > 0.1Vcc | 55.57Vout | | mA |
| | | | 0.3Vcc > Vout > 0 | | Eqt'n F | mA |
| | | (Test Point) | Vout = 0.3Vcc | | 50Vcc | mA |
| Icl | All | Low Clamp<br>Current | -3 < Vin < -1 | -25+(Vin+1)/0.015 | | mA |
| Ich | All | High Clamp<br>Current | Vcc+4 > Vin > Vcc+1 | 25+(Vin-Vcc-1)/0.015 | | mA |
| slew_r | 3.3V | Output rise<br>Slew rate | 0.25Vcc to 0.75Vcc | 1 | 3 | V/ns<br>(note1) |
| | 1.8V<br>1.5V | Output rise<br>Slew rate | 0.25Vcc to 0.75Vcc | 0.5 | 1.5 | V/ns<br>(note1) |
| slew_f | 3.3V | Output rise<br>Slew rate | 0.75Vcc to 0.25Vcc | 1 | 3 | V/ns<br>(note1) |
| | 1.8V<br>1.5V | Output rise<br>Slew rate | 0.75Vcc to 0.25Vcc | 0.5 | 1.5 | V/ns<br>(note1) |

**Table 2-3: SDI buffer AC specification**

| Symbol | | Parameter | Condition | Min | Max | Units |
|---|---|---|---|---|---|---|
| Ioh (AC) | 3.3V | Switching<br>Current high | 0 < Vout < 0.7vcc | -15Vcc | | |
| | | | 0.7Vcc < Vout < 0.9Vcc | -50(Vcc-Vout) | | |
| | | | 0.7Vcc < Vout < Vcc | | Eqt'n C | |
| | | (Test Point) | Vout = 0.7Vcc | | -37.5Vcc | mA |
| Ioh (AC) | 1.8V<br>1.5V | Switching<br>Current high | 0 < Vout < 0.7vcc | -9.38Vcc | | |
| | | | 0.7Vcc < Vout < 0.9Vcc | -31.27(Vcc-Vout) | | |
| | | | 0.7Vcc < Vout < Vcc | | Eqt'n G | |
| | | (Test Point) | Vout = 0.7Vcc | | -37.5Vcc | mA |
| Iol (AC) | 3.3V | Switching<br>Current Low | Vcc > Vout >0.3Vcc | 15Vcc | | |
| | | | 0.3Vcc > Vout > 0.1Vcc | 50Vout | | |
| | | | 0.3Vcc > Vout > 0 | | Eqt'n D | |
| | | (Test Point) | Vout = 0.3Vcc | | 37.5Vcc | mA |
| Iol (AC) | 1.8V<br>1.5V | Switching<br>Current Low | Vcc > Vout >0.3Vcc | 9.38Vcc | | |
| | | | 0.3Vcc > Vout > 0.1Vcc | 31.27Vout | | |
| | | | 0.3Vcc > Vout > 0 | | Eqt'n H | |
| | | (Test Point) | Vout = 0.3Vcc | | 37.5Vcc | mA |
| Icl | | Low Clamp<br>Current | -3 < Vin < -1 | -25+(Vin+1)/0.015 | | mA |
| Ich | | High Clamp<br>Current | Vcc+4 > Vin > Vcc+1 | 25+(Vin-Vcc-1)/0.015 | | mA |
| slew_r | 3.3V | Output rise<br>Slew rate | 0.25Vcc to 0.75Vcc | 1 | 3 | V/ns<br>(note 1) |
| | 1.8V<br>1.5V | Output rise<br>Slew rate | 0.25Vcc to 0.75Vcc | 0.5 | 1.5 | V/ns<br>(note 1) |
| slew_f | 3.3V | Output rise | 0.75Vcc to 0.25Vcc | 1 | 3 | V/ns |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | Slew rate | | | | (note 1) |
| | 1.8V<br>1.5V | Output rise<br>Slew rate | 0.75Vcc to 0.25Vcc | 0.5 | 1.5 | V/ns<br>(note 1) |

**NOTES:**

1. Slew rate is to be interpreted as the cumulative edge rate across the specified range, (0.25Vcc to 0.75Vcc load for rise and 0.75Vcc to 0.25Vcc load for fall), rather than instantaneous rate at any point within the transition range.

## 2.2.3 Maximum AC Ratings and Device Protection

All HD Audio buffers should be capable of withstanding continuous exposure to the waveform shown in Figure 2-5 (3.3V signaling) or Figure 2-6 (1.8V or 1.5V signaling). It is recommended that these waveforms be used as qualification criteria against which the long-term reliability of each device is evaluated. Table 2-4 lists the parameters of the waveform. This level of robustness should be guaranteed by design; it is not intended that this waveform should be used as a production test.

These waveforms are applied with the equivalent of a zero-impedance voltage source, driving through a series resistor directly into each HD Audio input or tri-stated output pin. The open-circuit voltage of the voltage source is shown in Figure 2-5 and Figure 2-6, which is based on the worst-case overshoot and undershoot expected in actual HD Audio buses. The resistor values are calculated to produce the worst case current into an effective internal clamp diode.

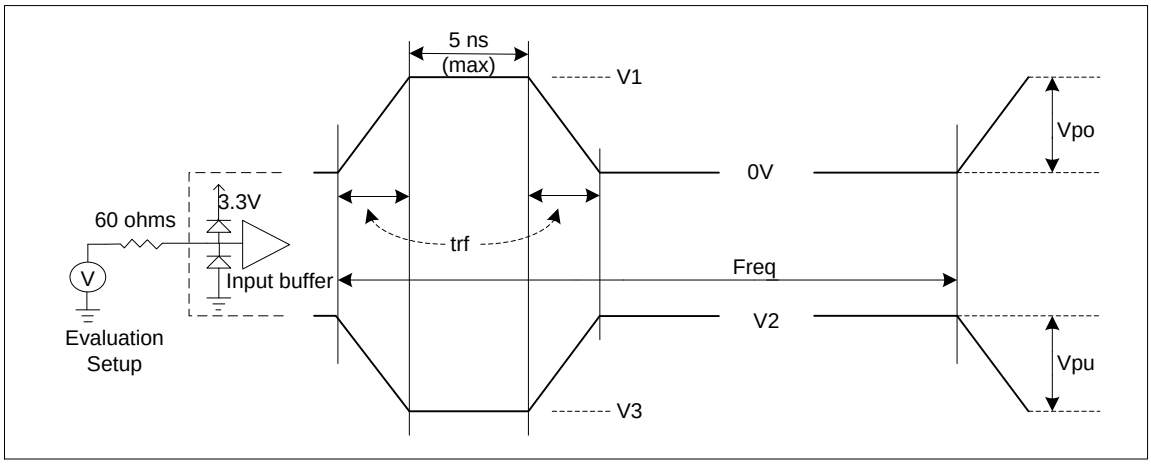



**Figure 2-5: Maximum AC waveforms for 3.3V Signaling**

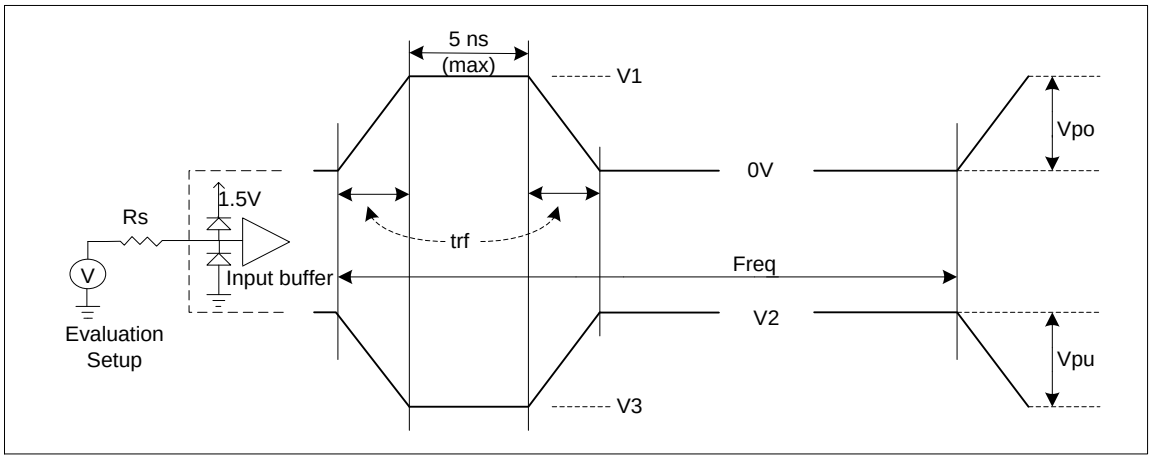



**Figure 2-6: Maximum AC waveforms for 1.5V Signaling**

**Table 2-4: Parameters for Maximum AC signaling waveforms**

| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| V1 | Overshoot Voltage 3.3V | | 6.3 | V |
| | Overshoot Voltage 1.8V | | TBD | |
| | Overshoot Voltage 1.5V | | 3.25 | |
| V2 | Undershoot initial Voltage 3.3V | | 3.465 | V |
| | Undershoot initial Voltage 1.8V | | TBD | |
| | Undershoot initial Voltage 1.5V | | | |
| V3 | Undershoot Voltage 3.3V | -3 | | V |
| | Undershoot Voltage 1.8V | TBD | | |
| | Undershoot Voltage 1.5V | -1.6 | | |
| Vpu | Waveform peak-to-peak 3.3V | | 6.465 | V |
| | Waveform peak-to-peak 1.8V | | TBD | |
| | Waveform peak-to-peak 1.5V | | 3.25 | |
| Vpo | Waveform peak-to-peak 3.3V | | 6.3 | V |
| | Waveform peak-to-peak 1.8V | | TBD | |
| | Waveform peak-to-peak 1.5V | | 3.25 | |
| trf | Rise/fall time 3.3V | 1 | 3 | V/ns |
| | Rise/fall time 1.8V or 1.5V | 0.5 | 1.5 | |
| Freq | Frequency of AC rating waveform as applied to SDI input buffers | 6 | 24 | MHz |
| Freq | Frequency of AC rating waveform as applied to SDO input buffers | 6 | 24 | MHz |

**Note that:**

- The voltage waveform is supplied at the resistor shown in the evaluation setup, not the package pin.
- Any internal clamping in the device being tested will greatly reduce the voltage levels seen at the package pin.

**Table 2-5: Resistance value for the AC rating waveform 1.8V and 1.5V**

| Rs | Condition | Value |
|---|---|---|
| | Overshoot waveform at the Codec | 65 ohms |
| | Undershoot waveform at the Codec | 101 ohms |
| | Overshoot waveform at the Controller | 108 ohms |
| | Undershoot waveform at the Controller | 133 ohms |

## 2.3 Measurements and test conditions

Figure 2-9 and Figure 2-10 define the timing parameters as measured at the controller and the codec respectively. The component test guarantees that all timings are met with minimum clock slew rate (slowest edge) and minimum voltage swing. The design must guarantee that minimum timings are also met with maximum clock slew rate (fastest edge) and maximum voltage swing. In addition, the design must guarantee proper input operation for input voltage swings and slew rates that exceed the specified test conditions. The measurement conditions are summarized in Table 2-6. Figure 2-7 and Figure 2-8 specify the load used to characterize the slew rates and delay time. The reference point for all delay timing measurements is 0.5Vcc. The load used to characterize the slew rate and the delay is the same for both 1.5V, 1.8V and 3.3V signaling.

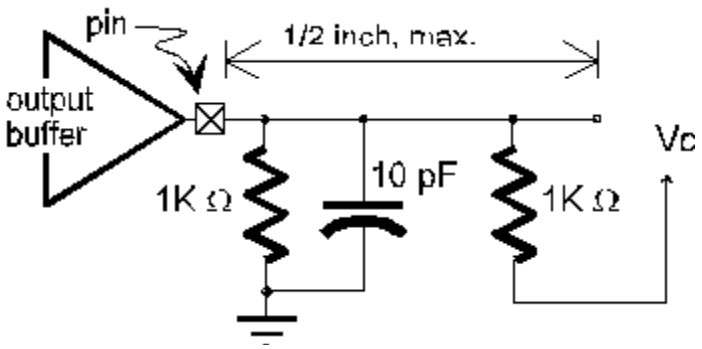



**Figure 2-7: Slew rate and min valid delay**

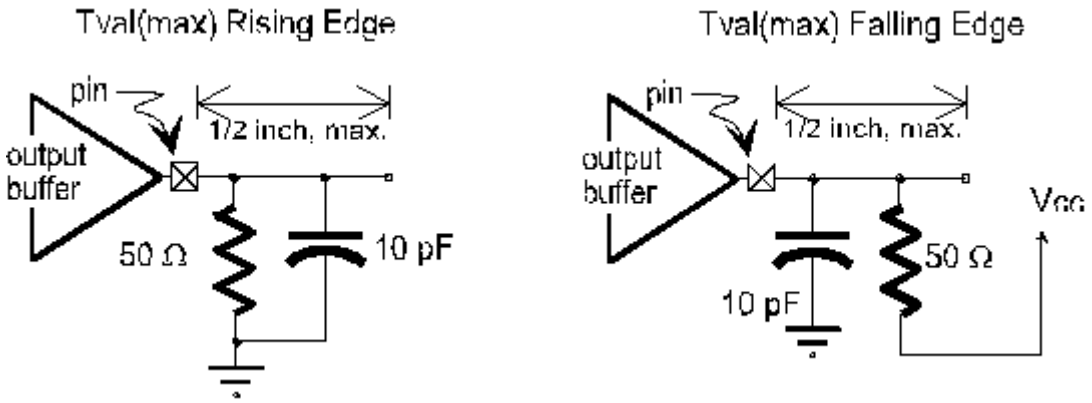



**Figure 2-8: Max valid delay**

**Table 2-6: Measurement condition parameters**

| Symbol | 1.5V signaling | 1.8V signaling | 3.3V signaling | Units |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| V_h | 0.7Vcc | 0.7Vcc | 0.75Vcc | V (note 1) |
| V_l | 0.3Vcc | 0.3Vcc | 0.25Vcc | V (note1) |
| V_test | 0.5Vcc | 0.5Vcc | 0.5Vcc | V |
| V_max | 0.5Vcc | 0.5Vcc | 0.5Vcc | V (note 2) |

**NOTES:**

1. Input test is done with 0.1Vcc of overdrive. V_h = Vih + 0.1 Vcc; V_l = Vil - 0.1 Vcc
2. V_max specifies the maximum peak to peak waveform allowed for measuring input timing

## 2.4 Timing Specification

Table 2-7 describes the timing parameters at the controller interface and Table 2-8 describes the timing parameters at the codec interface. All the timing numbers are defined at the package pins of the corresponding interface. Rise and fall time, flight and delay time, setup and hold time listed here should be used together in the worst-case scenario for modeling of the drivers on the HD Audio link. Setup and hold timing numbers for SDO are defined at every edge of the BCLK while those for SDI will be defined only at the rising edge of BCLK. This is because SDO is double pumped while SDI is not. SYNC and RST# should be treated same as SDO and hence have the same timing definitions as that of SDO. Section 2.3 describes the method and the loads used to characterize these timing parameters. Timing parameters and their values as defined here applies for 3.3V as well as 1.5V and 1.8V signaling.

**Table 2-7: Timing parameters at the controller**

| Symbol | Definition | Min | Typ | Max | Units | Notes |
|---|---|---|---|---|---|---|
| **24.0 MHz BCLK Mode (default, all devices must support this)** | | | | | | |
| | BCLK frequency | 23.9976 | 24.0 | 24.0024 | MHz | 3 |
| T_cyc | Total period of BCLK | 41.363 | 41.67 | 41.971 | ns | 3 |
| T_high | High phase of BCLK | 18.75 | | 22.91 | ns | 1 |
| T_low | Low phase of BCLK | 18.75 | | 22.91 | ns | 1 |
| | BCLK jitter | | 150 | 300 | ps | 5 |
| T_ival | Time duration for which SDO is valid before the BCLK edge | 7 | | | ns | 4,5 |
| T_val | Time duration for which SDO is valid after the BCLK edge | 7 | | | ns | 4,5 |
| T_su | Setup for SDI at rising edge of the BCLK | 15 | | | ns | 4,5 |
| T_h | Hold for SDI at rising edge of the BCLK | 0 | | | ns | 4,5 |
| | | | | | | |
| **12.0 MHz BCLK Mode (optional)** | | | | | | |
| | BCLK frequency | 11.9988 | 12.0 | 12.0012 | MHz | 3 |
| T_cyc | Total period of BCLK | 83.025 | 83.33 | 83.642 | ns | 3 |
| T_high | High phase of BCLK | 37.50 | | 45.83 | ns | 1 |
| T_low | Low phase of BCLK | 37.50 | | 45.83 | ns | 1 |
| | | | | | | |
| **6.0 MHz BCLK Mode (optional)** | | | | | | |
| | BCLK frequency | 5.9994 | 6.0 | 6.0006 | MHz | 3 |

| Symbol | Definition | Min | Typ | Max | Units | Notes |
|---|---|---|---|---|---|---|
| 24.0 MHz BCLK Mode (default, all devices must support this) | | | | | | |
| | Total period of BCLK | 166.350 | 166.67 | 166.983 | ns | 3 |
| | High phase of BCLK | 75.00 | | 91.67 | ns | 1 |
| | Low phase of BCLK | 75.00 | | 91.67 | ns | 1 |

**Table 2-8: Timing parameters at the codec**

| Symbol | Definition | Min | Typ | Max | Units | Notes |
|---|---|---|---|---|---|---|
| 24.0 MHz BCLK Mode (default, all devices must support this) | | | | | | |
| | BCLK frequency | 23.9976 | 24.0 | 24.0024 | MHz | 3 |
| T_cyc | Total period of BCLK | 41.163 | 41.67 | 42.171 | ns | 3 |
| T_high | High phase of BCLK | 17.5 | | 24.16 | ns | 2 |
| T_low | Low phase of BCLK | 17.5 | | 24.16 | ns | 2 |
| | BCLK jitter | | 150 | 500 | ps | 5 |
| T_tco | Time after rising edge of the BCLK that SDI becomes valid | 3 | | 11 | ns | 4,5 |
| T_su | Setup for SDO at both rising and falling edge of the BCLK | 5 | | | ns | 4,5 |
| T_h | Hold for SDO at both rising and falling edge of the BCLK | 5 | | | ns | 4,5 |
| | | | | | | |
| 12.0 MHz BCLK Mode (optional) | | | | | | |
| | BCLK frequency | 11.9988 | 12.0 | 12.0012 | MHz | |
| T_cyc | Total period of BCLK | 82.825 | 83.33 | 83.842 | ns | 3 |
| T_high | High phase of BCLK | 35 | | 48.33 | ns | 3 |
| T_low | Low phase of BCLK | 35 | | 48.33 | ns | 2 |
| | | | | | | 2 |
| 6.0 MHz BCLK Mode (optional) | | | | | | |
| | BCLK frequency | 5.9994 | 6.0 | 6.0006 | MHz | 3 |
| T_cyc | Total period of BCLK | 166.150 | 166.67 | 167.183 | ns | 3 |
| T_high | High phase of BCLK | 70 | | 96.67 | ns | 2 |
| T_low | Low phase of BCLK | 70 | | 96.67 | ns | 2 |

**NOTES:**

1. 45 / 55 % is the worst-case duty cycle at the controller, as measured at v_test in Figure 2-9.
2. 42 / 58 % is the worst-case duty cycle at the codec, as measured at v_test in Figure 2-10.
3. This is the long-term average frequency measured over 1 ms. Clock has a 100ppm tolerance in HD Audio interface.
4. The design should meet the timing requirements with the slew rate of the inputs in the range 1V/ns to 3 V/ns.
5. Controller BCLK Jitter, T_ival, T_val, T_su, T_h and codec BCLK Jitter, T_tco, T_su and T_h values are common for BCLK=24MHz, 12MHz and 6MHz.

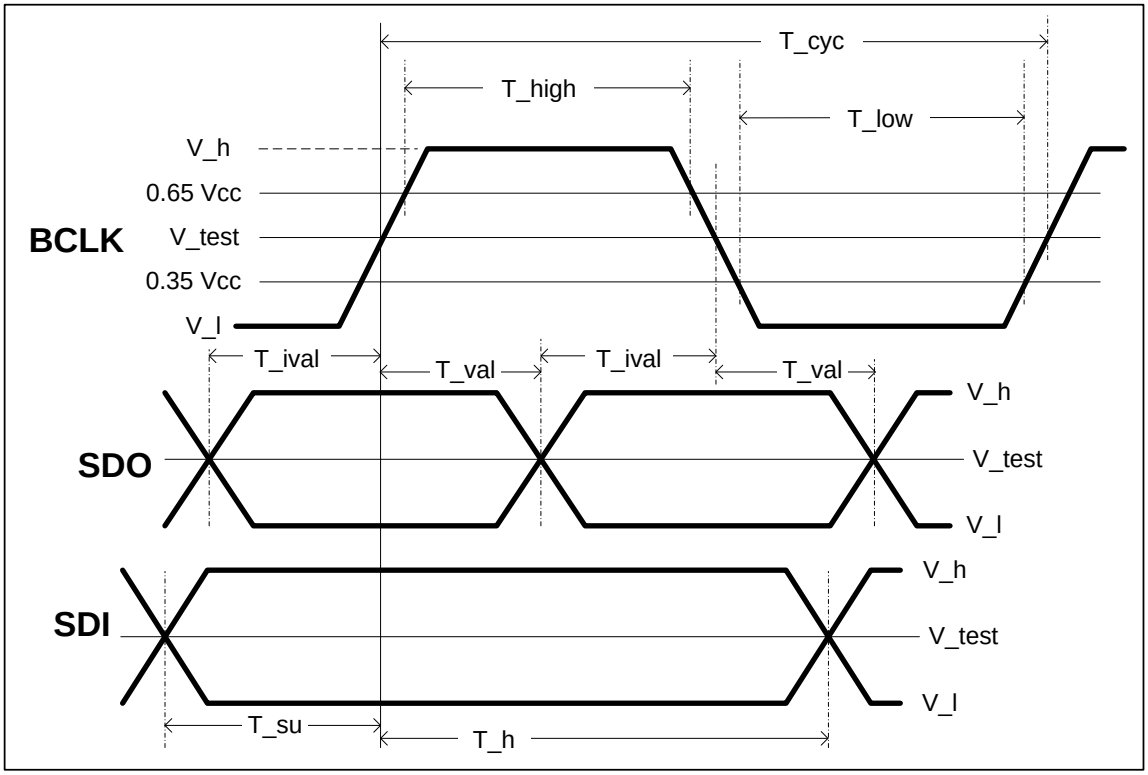



**Figure 2-9: Timing parameters as measured at the controller**

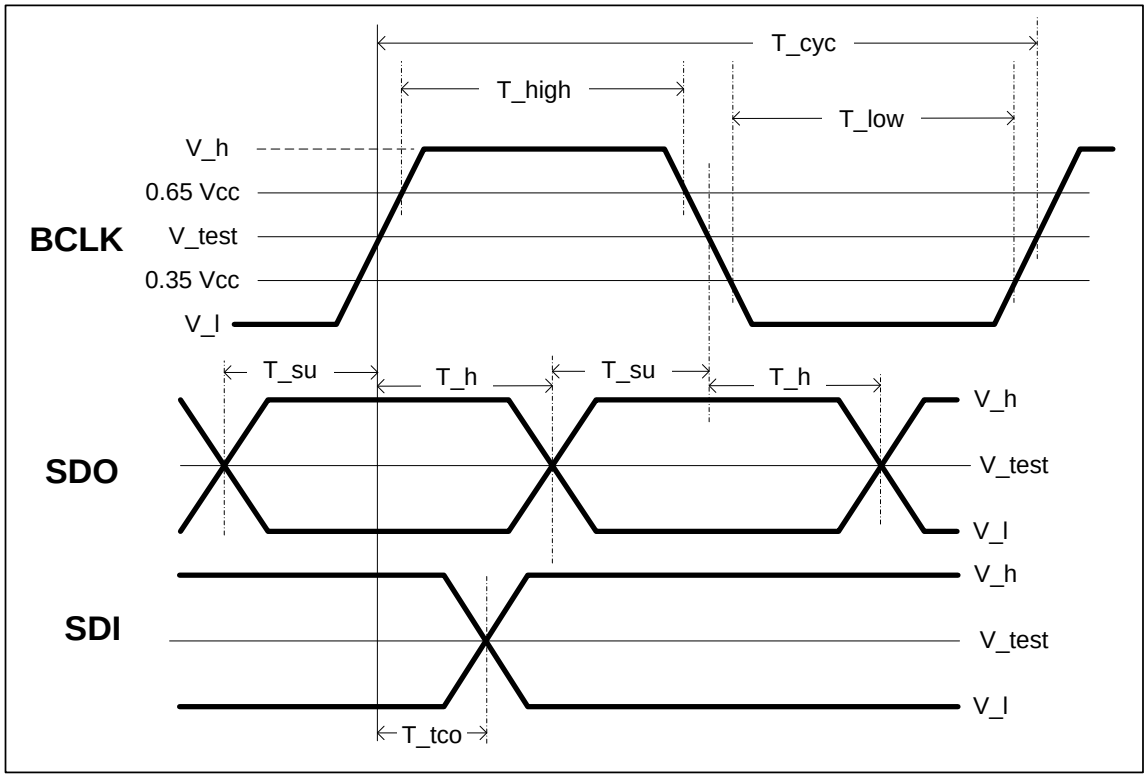



**Figure 2-10: Timing parameters as measured at the codec**

## 2.5 System Timing Budget

The setup and hold numbers at the codec are dependent on the total skew introduced between BCLK and SDO. This should include signaling induced skews as well as skews due to trace mismatches on the board. Table 2-9 describes the limits for the total skew allowed on the system. Total skew allowed between BCLK & SYNC and BCLK & RST# is the same as that allowed for BCLK & SDO, which is described below.

**Table 2-9: Total trace mismatch**

| Description | Min | Max | Units | Notes |
|---|---|---|---|---|
| Total skew allowed between BCLK and SDO | -2 | 2 | ns | 1 |

**NOTES:**

1. This should be the total mismatch in the system including the motherboard and any applicable daughter cards or connectors.

Setup and hold numbers for SDI at the controller are dependent on the flight time from the controller to the receiver and vice-versa. Table 2-10 defines the maximum flight time allowed on the system.

**Table 2-10: Maximum trace lengths**

| Description | Min | Max | Units | Notes |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| Flight time for BCLK from controller to the codec | 0 | 7 | ns | 2 |
| Flight time for SDI from the codec to the controller | 0 | 7 | ns | 2 |

**NOTES:**

2. The maximum flight times in this table will dictate the total maximum allowed trace lengths between the codec and the controller.

## 2.6 Frequency Switching

### 2.6.1 Low Frequency Mode

The High Definition Audio Architecture is designed to support static frequency switching. This is an optional feature which will help to save power when the audio sub-system is configured for lower-bandwidth operations making it feasible to run a slower BCLK. So, in addition to supporting the mandatory default 24.00-MHz BCLK, High Definition Audio Controllers and codecs may optionally support a set of lower operating frequencies i.e. 6.00-MHz and 12.00-MHz. Note that 24.00-MHz support is required for backward-compatibility.

### 2.6.2 Static Frequency Switching

This is only applicable if both the controller and all the codecs in the system advertise support for multiple frequencies. In all cases, frequency switching is fully done under software control. The process is considered “static” because it is only applicable when all audio streams are inactive and when there is never the possibility of dynamically needing to revert to a different frequency.

### 2.6.3 Software-visible Resources & Actions

The default BCLK frequency is 24MHz but it can be preconfigured to operate at 12MHz as well as 6MHz if the following conditions on the relevant link segment are met:

1) controller supports low frequency mode
2) all codecs support low frequency mode (note that the frequency switch capability is implied if more than one operating frequency is advertised)
3) all codecs’ Max Required Bandwidth can be satisfied by a lower BCLK frequency
4) there is no possibility of needing to revert in mid-stream to a different frequency
5) System software or BIOS is responsible for pre-configuring the HD Audio system to the appropriate frequency based on its innate knowledge about how much maximum audio link bandwidth is required by the specific platform. This is done while there are no active streams running. Please refer to the relevant BIOS or System Software programming guide for details.

#### 2.6.3.1 Host Controller Hardware Frequency Switching Actions

The controller hardware cannot switch frequency without explicit instructions from software. A controller exposes its supported range of frequencies through a set of capability register fields (LCTLx.BCAP). Note that the frequency switch capability is implied if more than one operating frequency is advertised.

When software issues a SET_CURRENT_FREQUENCY command. The controller hardware is responsible for decoding this command & adjusting its internal bus divider ratios to achieve the new BCLK frequency.
Note that the static frequency change is only executed when the link is in reset. It is software's responsibility to instruct the controller on when to enter and exit Link Reset. Hardware must update the LCTLx.CPA status whenever it has completed the Link Reset Entry and Exit operations. This is the indicator to software that the hardware-assisted portion of the reset sequence is complete and control is handed back to software.

### 2.6.3.2 Codec Frequency Switching Actions

Codecs advertise the range of supported frequencies through the Switchable BCLK Clock Frequency parameter (16h).
The codec is responsible for responding to the frequency switching flow initiated by the controller. This is illustrated by the figure below. Upon exiting Link-Reset, the codec will count the number of BCLKs in between the first two frame syncs which it receives from the controller. Since frames occur at fixed 48kHz (20.83µs) intervals, it can deduce the frequency from the number of BCLKs contained in the frame. For example, there are 125, 250 and 500 BCLKs per frame for 6MHz, 12MHz and 24MHz operation respectively.
The codec is also responsible for reporting the current BCLK frequency which it has detected. From RESET# de-assertion, the codec must be ready within 25 frames to report the correct frequency in response to a GET (Current BCLK Frequency) from the controller. Refer to Sections 7.3.3.44 and 7.3.4.17 of Intel ECR HDA048A for additional details.

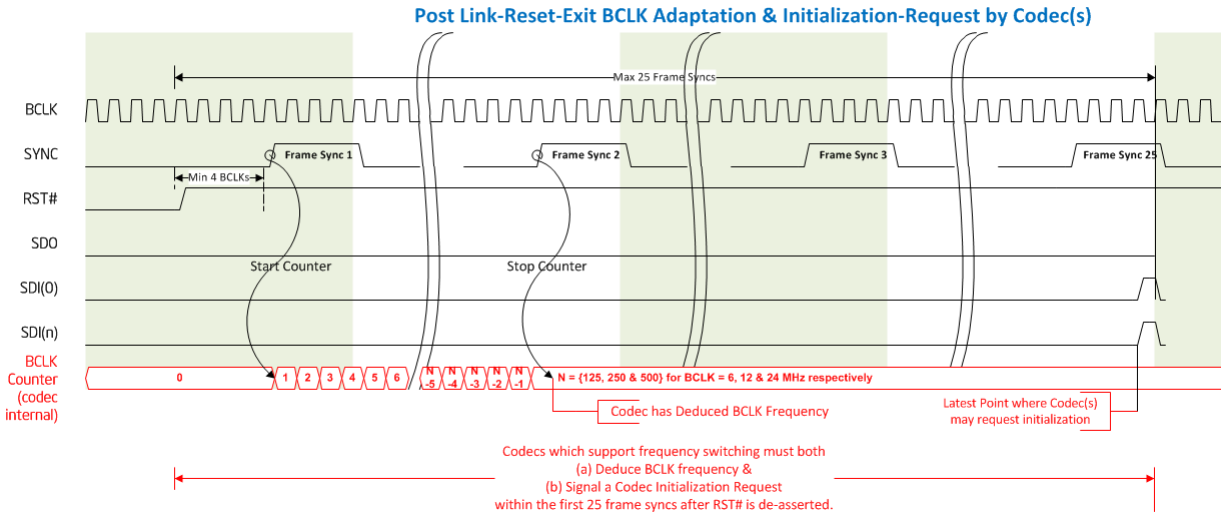
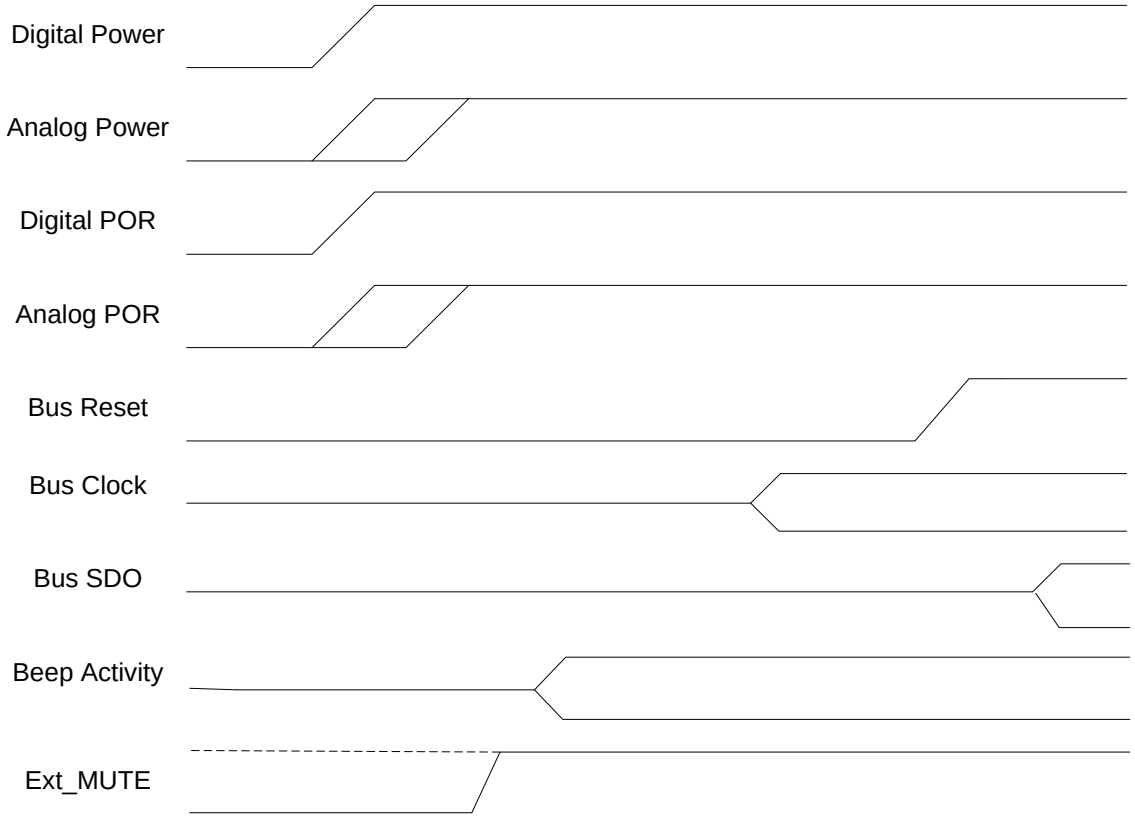



Figure 2-11: Codec deduces new BCLK frequency by counting the number of BCLKs in between the first two frame syncs upon exiting link-reset

# 3 PCBEEP

## 3.1 Case 1: Power up & Beep




At digital POR the port enable bit for both the Port G (Class-D amplifier) and Port E (line out) will be set. All widgets, this includes muxes, mixers, DACs, ADCs and ports, are placed in the Idle 2 (10ms recovery) state. Based on the state of Ext_Mute and the presence of the PC-Beep signal the enabled ports should quickly move to a state that can support PC-beep pass through. In this mode PC-Beep is routed through analog.

## 3.2 Case 2: Beep S0 and device completely at D0

In this case the system is fully up. Digital and analog voltages are available. The DACs and ports are powered up. PC-beep shall be routed through digital to all enabled ports.

## 3.3 Case 3: Beep during S0 but the device is in mixed power states

In this case the system is fully up. Digital and analog voltages are available. The system clock will be present if the audio function group is above the idle 2 state. If the audio function group is in the idle 2 state the clock may or may not be present but the hardware should switch over to the RC oscillator. Reset is not active. Any port that has its enable bit set must pass PC-Beep. The preferred method is to

power up any digital and analog blocks connected to the enabled port(s) and pass beep through the digital path. Alternately the port could be powered and beep passed through the analog path. The analog path is less preferred because the user is unaware of this hardware state and could be surprised by a randomly loud beep when the DAC volume is set to low levels.

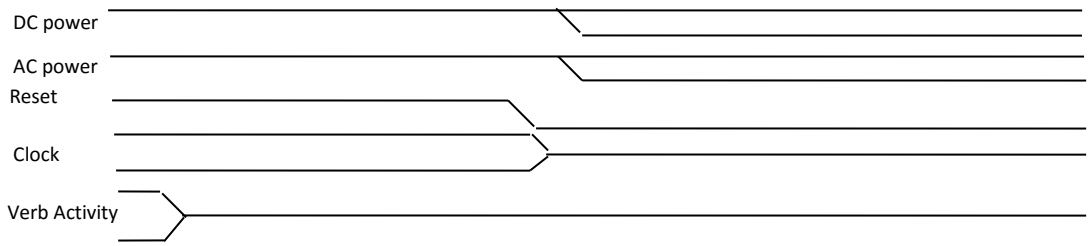
## 3.4 Case 4: Beep during S0 but the device is in Idle 2

In this case the system is fully up. Digital and analog voltages are available. The system clock may or may not be present but the hardware should switch over to the RC oscillator. Reset is not active. Any port that has its enable bit set must pass PC-Beep. The preferred method is to power up any digital and analog blocks connected to the enabled port(s) and pass beep through the digital path. Alternately the port could be powered and beep passed through the analog path. The analog path is less preferred because the user is unaware of this hardware state and could be surprised by a randomly loud beep when the DAC volume is set to low levels.

## 3.5 Case 5: Beep during resume from suspend or hibernate

This looks exactly like the power up case accept that analog and digital POR may or may not occur. This is because analog and digital power may or may not have gone away.

## 3.6 Power down signals



# 4 Analog Macro

## 4.1 Analog Macro Block Diagram

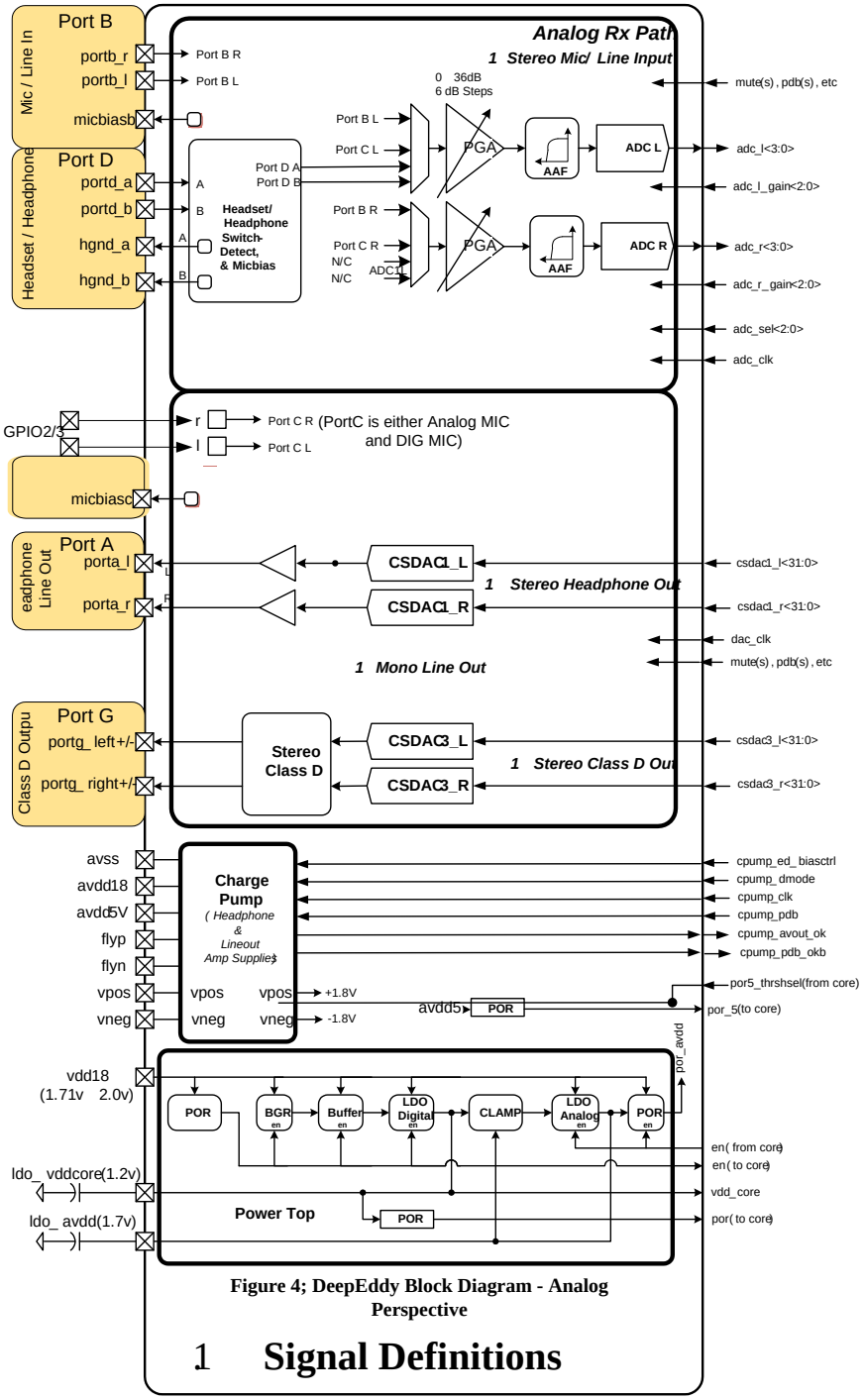



**Figure 4; DeepEddy Block Diagram - Analog Perspective**

## 1 Signal Definitions

## 4.2 AFE Interface Signal List

Signal I/O type is with respect to AFE.

| Pin | Description | Type |
|---|---|---|
| vdd_core | 1.2V core supply from digital. | inout |
| vss_core | Core ground from digital. | inout |
| por_core | Core supply por signal | output |
| por_5 | AVDD5V supply por signal | output |
| por5_thrshsel | 5V POR threshold select<br>0 = 0.875V<br>1= 1.52V | |
| scanmode | Scanmode output to Digital from AFE | Output |
| **Reference Control Signals** | | |
| vref_ok | State of analog vref<br>0 = vref is below 1.2V<br>1 = vref is above 1.2V | Output |
| lp_en | 1 = The system is in lowest power mode | Input |
| **Micbias and Headset Detection Controls** | | |
| micbias_int_en | Enable micbiasb internally.<br>0 = micbiasb is high-Z<br>1 = micbiasb is connected to porta_l/r with internal 2.2K resistor | Input |
| micbias_int_set | Selects micbiasb voltage level<br>0 = 100% (2.6V min voltage)<br>1 = 80% (2.1V max voltage) | Input |
| micbias_pin_en | Enable micbias on the pin.<br>0 = micbias pad is high-Z<br>1 = micbias pad is biased | Input |
| micbias_pin_set | Selects micbias voltage level<br>0 = 100% (2.6V min voltage)<br>1 = 80% (2.1V max voltage) | Input |
| micbiasd_en | Enable micbiasd.<br>0 = micbiasd pad is high-Z<br>1 = micbiasd pad is biased | Input |
| micbiasd_set | Selects micbiasd voltage level<br>0 = 100% (2.6V min voltage)<br>1 = 80% (2.1V max voltage) | Input |
| micbiasd_dtct | Micbias Comparator output on micbiasd<br>0 = Microphone not detected micbiasd<br>1 = microphone detected on micbiasd | Output |
| headset_sel_a | 1 = portd_a headset is detected | Input |
| headset_sel_b | 1 = portd_b headset is detected | Input |
| headset_detection | 1 = when headset dectection statemachine is running. | Input |
| headset_tip_detect | 1 = enabled when headset is inserted completely | Ouput |
| **Multibutton and Tip&Ring Detection Controls** | | |

| | | |
|---|---|---|
| mb_dac<7:0> | 8 bit SAR ADC data to AFE | Input |
| mb_adc_sample | SAR ADC sample output to AFE<br>1 = debounced micbiasd_dtct indicates button push | Input |
| mb_en | Enable multibutton detection<br>1 = Enabled<br>0 = Disabled | Input |
| mb_adc_comp | ADC single bit SAR comparator output | Output |
| tr_detect_en | Enables Tip and Ring sensing feature<br>1 = Tip and ring sensing on<br>0 = Tip and ring sensing off | Input |
| tr_compout | Tip and ring sensing comparator output<br>0 = >1K load<br>1 = <1K load | Output |
| tr_sense_switch | Change the tip and run sensing threshold<br>0 = 1KOhm<br>1 = 5KOhm | Input |
| **ADC Controls** | | |
| adc_pdb | Enable of ADC1 path (PGA+AAF+DSM), both left and right.<br>1 = ADC1 enabled | Input |
| adc_mute | Mute of PGA input of ADC1.<br>1 = enable mute. | Input |
| adc_l_gain[2:0] | ADC1 left channel input PGA gain select.<br>3'b000 = 0dB<br>3'b001 = 6db<br>3'b010 = 12dB<br>3'b011 = 18dB<br>3'b100 = 24dB<br>3'b101 = 30db<br>3'b110 = 36dB<br>3'b111 = 42dB | Input |
| adc_r_gain[2:0] | ADC1 right channel input PGA gain select.<br>3'b000 = 0dB<br>3'b001 = 6db<br>3'b010 = 12dB<br>3'b011 = 18dB<br>3'b100 = 24dB<br>3'b101 = 30db<br>3'b110 = 36dB<br>3'b111 = 42dB | Input |
| adc_sel[1:0] | ADC1 PGA input mux select.<br>2'd0 = input is port B<br>2'd1 = input is port C<br>2'd2 = input is port D_a<br>2'd3 = input is port D_b | Input |
| adc_clk | 3.072MHz ADC Sampling Clock | Input |
| adc_l[3:0] | Bitstream from ADC1 left channel modulator. | Output |

| | | |
|---|---|---|
| adc_r[3:0] | Bitstream from ADC1 right channel modulator. | Output |
| **DAC Controls** | | |
| csdac_clk | 3.072MHz DAC sampling clock | Input |
| csdac1_l[31:0] | Bitstream to DAC1 left channel CS DAC. | Input |
| csdac1_r[31:0] | Bitstream to DAC1 right channel CS DAC. | Input |
| csdac1_l_oob[4:0] | OOB DAC stream for DAC1 left channel | Input |
| csdac1_r_oob[4:0] | OOB DAC stream for DAC1 right channel | Input |
| csdac1_pdb | Enable of CS DAC 1<br>1 = CS DAC enabled | Input |
| csdac1_mute | Mute of CS DAC 1 (opamp in unity-gain feedback, clock halted).<br>1 = Enable CS DAC mute | Input |
| csdac3_l[31:0] | Bitstream to DAC3 left channel CS DAC. | Input |
| csdac3_r[31:0] | Bitstream to DAC3 right channel CS DAC. | Input |
| csdac3_pdb | Enable of CS DAC 3<br>1 = CS DAC enabled | Input |
| csdac3_mute | Mute of CS DAC 3 (opamp in unity-gain feedback, clock halted).<br>1 = Enable CS DAC mute | Input |
| **Chargepump Controls** | | |
| cpmp_pdb | Enable for charge pump.<br>1 = enabled | Input |
| cpmp_clk | 6.144MHz Clock for charge pump. | Input |
| cpmp_ctrl_dmode | Charge-pump voltage mode input. Driven from Enevelop detector<br>0 = D1 mode<br>1 = D2 mode | Input |
| cpmp_pdb_okb | Charge pump power down status signal.<br>0 = charge pump has discharged vnos and vpeg voltages to 0. | Output |
| cpmp_out_ok | Charge pump power up status signal.<br>1 = charge pump has charged vpos and vneg voltages to final value | Output |
| **Output Drivers Controls** | | |
| **PortA capless Lineout/Headphone** | | |
| porta_mute | Enable mute for port A.<br>1 = Mute enabled | Input |
| porta_highz | Sets porta output to highz for HP_merge mode<br>1 = Enale highz mode | Input |
| porta_pdb | Enable for port A.<br>1 = Port is powered on | Input |
| porta_outsw | Output switch enable for PORT A driver.<br>1 = Output grounded | Input |
| porta_hpmode | Set drive strength of port A driver.<br>0 = Support min 10 kOhm load.<br>1 = Support min 32 Ohm load. | Input |

| | | |
|---|---|---|
| porta_offcal_en | Enable PORT A offset calibration<br>1 = Calibration enabled | Input |
| porta_offcal_l<6:0> | Calibration code for PORT A left channel | Input |
| porta_offcal_r<6:0> | Calibration code for PORT A right channel | Input |
| porta_offcal_cmp_l | Comparator output for PORT A left channel offset calibration | Output |
| porta_offcal_cmp_r | Comparator output for PORT A right channel offset calibration | Output |
| **ClassD** | | |
| madc_code<4:0> | 5 bit Monitor ADC output | Output |
| spkr_clk | 3.072MHz Clock for Class D | Input |
| spkr_pdb | Enable for both left and right channels Class D.<br>1 = enabled | input |
| spkr_mute | Mute for both left and right channels of ClassD<br>1 = unmute classD | Input |
| spkr_monomode | Disable for right channel Class D.<br>1 = R channel disabled (monomode) | input |
| spkr_beepen | Beep enable for Class D buffer (allows analog pc beep if present)<br>1 = Enable beep path if analog beep present | Input |
| spkr_offcal_en | Enable PORT G offset calibration<br>1 = Calibration enabled | Input |
| spkr_offcal_cmp_l | Offset calibration comparator output for PORT G left channel | Output |
| spkr_offcal_cmp_r | Offset calibration comparator output for PORT G right channel | Output |
| spkr_offcal_l<6:0> | Calibration code for PORT G left channel | Input |
| spkr_offcal_r<6:0> | Calibration code for PORT G right channel | Input |
| spkr_outp_l | Level shifted left channel hbridge output signal fed to power averaging block | Output |
| spkr_outn_l | Level shifted left channel hbridge output signal fed to power averaging block | Output |
| spkr_outp_r | Level shifted right channel hbridge output signal fed to power averaging block | Output |
| spkr_outn_r | Level shifted right channel hbridge output signal fed to power averaging block | Output |
| spkr_test_in[1:0] | Input path to feed in pwm signal directly to hbridge. | Input |
| spkr_test_out[3:0] | Digital test outputs from class D PWM | Output |
| spkr_sc_status | ClassD hbridge Filtered short circuits protection system status<br>1=sc detected | Output |
| spkr_oc_l | Left channel ClassD hbridge internal lower threshold over current protection signal | Output |

| | | |
|---|---|---|
| spkr_oc_r | Right channel ClassD hbridge internal lower threshold short circuit protection signal | Output |
| temp_error | Error signal to indicate over-temperature detection | Output |
| spkr_por_status | ClassD POR mux out for testing purpose | Output |
| **Beep Controls** | | |
| beep_ana_en | Enable analog beep path.<br>1 = analog beep enabled | Input |
| beepgain_porta[3:0] | Gain select for analog beep, portA path<br>output beep level (dBFS) = -64 + beep1_gain*4 | Input |
| beepgain_spkr[3:0] | Gain select for analog beep, ClassD path<br>output beep level (dBFS) = -64 + beep3_gain*4 | Input |
| beep_anamux_en | Enables for beep amplifiers and port input mux during analog beep<br>1 = beep amaplifiers and port input mux are enabled<br>0 = beep amplifiers and port input mux are disabled during analog beep | Input |
| beepgen_dig_en | Enable for input of beep signal from digital beep generator to analog path (active-high) | Input |
| beepgen_dig | Beep signal from digital beep generator | Input |
| beep_dig_en | PC beep status signal for digital path.<br>1 = PC beep present and analog path disabled | Output |
| beep_dig | Digital PC beep signal. | Output |
| **Jack Sense Controls** | | |
| jack_dsw[3:0] | DAC digital data for jacksense SAR ADC. | Input |
| jack_sample | Sampling clock for jacksense SAR ADC. | Input |
| jack_comp | Comparator output for jacksense SAR ADC. | Output |
| Jack_pdb | 1 = Enable Jacksense | Input |
| **Clock outputs** | | |
| rcosc_cal[4:0] | RCOSC calibration input | Input |
| rcosc_pdb | 6MHz rcoscillator pwoerdown signal | Input |
| rcosc_clk | High frequency RCoscillator clock | Output |
| lf_rcosc_clk | Low frequency RCoscillator clock | Output |
| pll_clkout | Audio pll output | Output |
| pll_clkin | Audio pll input clock | Input |
| pll_en | Audio pll enable signal | Input |
| pll_rstn | Audio pll DSM reset signal | Input |
| clk_ref_on | MCLK status<br>0 = MCLK is not present<br>1 = MCLK is present | Input |
| pll_clktsta | PLL testmode clock a going to GPIO1 | Output |
| pll_clktstb | PLL testmode clock a going to GPIO2 | Output |
| testctrl0[11:0] | Test bits. | Input |

| | | |
|---|---|---|
| testctrl1[11:0] | Test bits. | Input |
| testctrl2[11:0] | Test bits. | Input |
| testctrl5[11:0] | Test bits. | Input |
| testctrl6[11:0] | Test bits. | Input |
| testctrl7[11:0] | Test bits. | Input |
| testctrl8[11:0] | Test bits. | Input |
| testctrl9[11:0] | Test bits. | Input |
| testctrl10[11:0] | Test bits | Input |
| testctrl11[11:0] | Test bits | Input |
| testctrl12[11:0] | Test bits | Input |
| testctrl13[11:0] | Test bits | Input |
| testctrl14[11:0] | Test bits | Input |
| testctrl15[11:0] | Test bits | Input |
| testctrl16[11:0] | Test bits | Input |
| testctrl17[11:0] | Test bits | Input |

## 4.3 Analog Controller Specification

### 4.3.1 Clocking Signals

#### 4.3.1.1 rcosc_clk

6MHz rcosc_clk output to digital

#### 4.3.1.2 rcosc_cal[4:0]

rcosc calibration code coming from digital calibration SAR statemachine

#### 4.3.1.3 rcosc_pdb

(rcosc_pdb == 0) → 6MHz rcosc is off
(rcosc_pdb == 1) → RCOSC on

- 6MHz rcosc power down signal is driven by (<650uW lowpower mode signal) !d3_lp_en.
- This rcosc is powered down only when we are in lowest power state that is D3 no bclk mode (<650uW mode). Once we are out lower power mode rcosc needs ot be enabled.
- Must also be testbit controllable.

#### 4.3.1.4 lf_rcosc_clk

200KHz low frequency rcosc_clk used for I2C, wake of jack, beep blocks

#### 4.3.1.5 pll_clkout

Audio pll clock output. This is clock is 147.456MHz VCO clock which is divided down in digital for further use.

#### 4.3.1.6 pll_clkin

Audio pll clock input which is assumed to be 6MHz in HD Audio mode.

#### 4.3.1.7 pll_clktsta, pll_clktstb

- Pll clock outputs to gpio's in testmode
- Make sure both the clocks are going to different GPIO's in testmode.

- Pll_clktsta → GPIO1
- Pll_clktstb → GPIO2

#### 4.3.1.8 clk_ref_on

- Driven by "I2S_MCLK_ON" status from digital
- Must be testbit controllable

#### 4.3.1.9 pll_en, pll_rstn

- pll_en and pll_rstn are signals which will enable the audio pll and reset audio pll DSM. Both the signals are driven by the same source.
- PLL is enabled once we
- Must also be testbit controllable

### 4.3.2 REF control

#### 4.3.2.1 vref_ok

Vref_ok is an output from analog that shows when vref is above 75% of its max value. (maybe not needed)

#### 4.3.2.2 lp_en

Analog input to show we are in D3 low power mode
Must have a testbit control to overwrite the logic

### 4.3.3 MICBIAS control

The micbiasb, micbiase and micbiasd are part of portB, port E and portD, respectively

#### 4.3.3.1 micbiasd_dtct

micbiasd_dtct an output from analog which goes to high when there is a headset connected.

#### 4.3.3.2 micbias_pin_en, micbias_int_en, micbiasd_en

These signals are derived from the portx_vref_en[2:0] azalia signals. Assign a testbit to control the micbias pin assignment to portB and portC. Default test bit setting assigns micbias pin to portC and uses internal micbias for portB.

- micbias_pin_en = !hp_merge ? ((portb_vref_en == 3'd4 || 3'd1) && ( portb_power == IDLE1 || FULL))
  : portC is analog mic ? ((testbit contrl) ? (portb_vref_en == 3'd4 || 3'd1) && ( portb_power == IDLE1 || FULL) : (portc_vref_en == 3'd4 || 3'd1) && ( portc_power == IDLE1 || FULL)))
  : (portb_vref_en == 3'd4 || 3'd1) && ( portb_power == IDLE1 || FULL);
- micbias_int_en = hp_merge ? (testbit control) ? 0 : ((portb_vref_en == 3'd4 || 3'd1) && ( portb_power == IDLE1 || FULL))
- *micbiasd_en = (portd_vref_en == 3'd4 || 3'd1) && (portd_power == IDLE1 || FULL)

#### 4.3.3.3 micbiasint_set, micbiaspin_set, micbiasd_set

These signals are derived from the portx_vref_en[2:0] azalia signals.

- micbiasint_set = (portb_vref_en == 3'd1)
  micbiaspin_set = !hp_merge ? (portb_vref_en == 3'd1)

: portC is analog mic ? ((testbit ctrl) ? (portb_vref_en == 3'd1) : (portc_vref_en == 3'd1)
: (portb_vref_en == 3'd1);

- micbiasd_set = (portd_vref_en == 3'd1)

### 4.3.4 RXPGA, AAF, and ADC control and data

The RXPGA, AAF and ADC blocks are part of node 14 (RXPGA1, AAF1 and ADC1), and node 13 (RPGA2, AAF2, and ADC2).

#### 4.3.4.1 adc_sel[1:0],adc_l_gain[2:0], adc_r_gain[2:0]

These signals are derived from the portb_gain[1:0], portc_gain[1:0], portd_gain[1:0], adc1_con_indx[1:0], adc2_con_indx[1:0] azalia signals. The adcX_con_indx[1:0] signals supports values 0,1, 2, 3 and 4.

Add a test register to the LSB of the ADC gain to provide higher gain controllability per port.

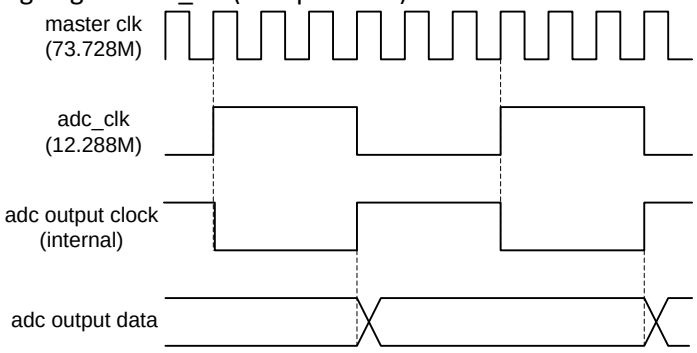
- adc_sel = (adc1/2_con_indx == portB) ? 2'b00 :
  ( adc1/2_con_indx == portC) && portC is analog mic ? 2'b01 :
  (adc1/2_con_indx == portD && headset_sel_a) ? 2'b10 :
  (adc1/2_con_indx == portD && headset_sel_b) ? 2'b11 : 2'b00
- adc_l_gain = (adc1/2_con_indx == portB) ? [portb_l_gain,testctrl] :
  (adc1/2_con_indx == portD) ? [portd_gain,testctrl] :
  (adc1/2_con_indx == portC) && portC is analog mic ?
  [portc_l_gain,testctrl]:
- adc_r_gain = (adc1/2_con_indx == portB) ? [portb_r_gain,testctrl] :
  (adc1/2_con_indx == portD) ? [portd_gain,testctrl] :
  (adc1/2_con_indx == portC) && portC is analog mic? [ portc_r_gain,testctrl]

#### 4.3.4.2 adc_pdb, adc_mute

1) adc_pdb = (node 18 || node 19 || node 1A(if analog mic) ) in full power state && (node 18 || node 19 || node 1A(if analog mic) ) conn index is connected to analog inputs && vref_ok == 1
   a) low to high transition should happen 20us after pdb_refs goes high (if pdb_refs was low).
2) adc_mute = (connected port on (node 18 || node 19 || node 1A(analog mic)) is not input enabled or not in D0) || (adc_pdb == 0)
   a) high to low transition should happen 20us after adc1_pdb goes high (if adc_pdb was low).

#### 4.3.4.3 adc_l[3:0], adc_r[3:0]

The analog ADC outputs its data on the falling edge of adc_clk. This data should be re-sampled on the rising edge of adc_clk (or equivalent).



When we have headset connected. We power off right channel and provide the mono mic data on adc_l. Need to copy the data onto right channel in digital.

#### 4.3.4.4 headset_sel_a, headset_sel_b

headset_sel _a and headset_sel_b describes which type of headset is connected. It is set high when we have a_type and b_type headset respectively.

#### 4.3.4.5 headset_detection

headset_detection is set high when we have a headset detected. It is input to the analog.

#### 4.3.4.6 headset_tip_detect

It is analog output signal which is set high only when headset is inserted completely.

### 4.3.5 Multi-button and Tip & Ring detection control

#### 4.3.5.1 mb_en

mb_en enables the multibutton detection circuitry in AFE. This signal needs to be driven by *combo_multibutton_en* in digital.

#### 4.3.5.2 mb_adc_comp, mb_adc_sample, mb_dac<7:0>

- multibutton SAR ADC control signals at the interface
- Flexibility to read mb_dac<7:0> is needed.

#### 4.3.5.3 tr_detect_en

Enables tip and ring detection and gets set by Jacksense detection statemachine. This signal should be enabled before micbiasd_en and disabled after micbiasd_en when Tip and ring sensing is enabled.

#### 4.3.5.4 tr_compout

- Tip and Ring sensing comparator output 0 → load is >1KOhms and 1 → load is >1KOhms.
- This signal should be filtered through micbiasd_dtct filter when headphone is inserted and tip & ring sensing feature is enabled.
- Bringout mb_en, mb_adc_comp, mb_adc_sample,tr_detect_en,tr_compout on GPIO's in testmode

#### 4.3.5.5 tr_sense_switch

- Tip and Ring sensing comparator output 0 → load is >1KOhms and 1 → load is >1KOhms.
- Bring out mb_en, mb_adc_comp, mb_adc_sample,tr_detect_en, tr_compout, tr_sense_switch on GPIO's in testmode

### 4.3.6 CSDAC control and data

The CSDAC blocks are part of node 16 (portA) and node 17 (portG)

#### 4.3.6.1 csdac1_pdb, csdac1_mute, csdac3_pdb, csdac3_mute.

- **csdac1_pdb** = (node 16(portA)) is full power state) &&
  vref_ok == 1 || ((DAC1 || DAC2) is connected to (node 16 (portA) headphone port

- low to high transition should happen 20us after pdb_refs goes high (if pdb_refs was low).
- high to low transition must happen at least 40us after all signal and offset ramping has completed (such that port controller has entered mute state).
- must also be testbit controllable.

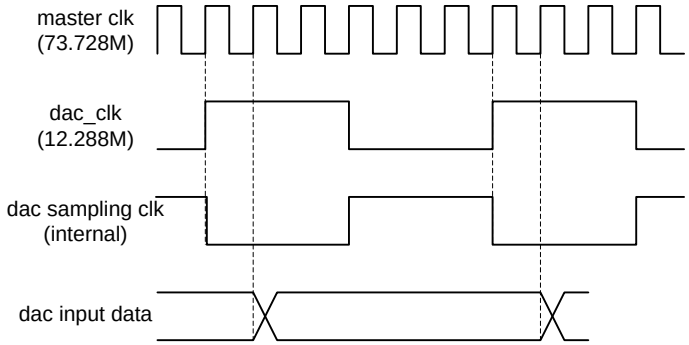
- **csdac1_mute** = (!(node 16 (portA) in full power state) && ((DAC1 || DAC2) is connected to headphone port))
  - low to high transition should happen after signal ramping and offset ramping has completed.
  - must also be testbit controllable.
- **csdac3_pdb** = (node 17 (portG) is full power state) && vref_ok == 1 || ((DAC1 || DAC2) is connected to (portG speaker port)
  - low to high transition should happen 20us after pdb_refs goes high (if pdb_refs was low).
  - high to low transition must happen at least 40us after all signal and offset ramping has completed (such that port controller has entered mute state).
  - must also be testbit controllable.
- **csdac3_mute** = (!(node 17(portG) in full power state) && ((DAC1 || DAC2) is connected to speaker port))
  - low to high transition should happen after signal ramping and offset ramping has completed.
  - must also be testbit controllable.

#### 4.3.6.2 csdac1_l_oob[4:0], csdac1_r_oob[4:0

The 5 bit OOB DAC binary inputs should be in sign and magnitude binary format. The binary to thermometer conversion happens inside the AFE. The OOB DAC data should follow same clocking structure as the actual DAC input data stream.

#### 4.3.6.3 csdac1_l[31:0], csdac1_r[31:0], csdac3_l[31:0], csdac3_r[31:0].

The DAC input data should be delayed one master clk cycle from the rising edge of CSDAC _clk.




### 4.3.7 PORT control

The port32 block is part of nodes 16 (PORT32v2_A) and port10k block is part of node 1D (PORT10K_E) and node 22 (port10K_M)

#### 4.3.7.1 porta_hpmode

This signal is derived from the porta_hphn_en azalia signals and controlled by testbit register.

porta_hpmode = porta_hphn_enable

#### 4.3.7.2 porta_offcal_en, spkr_offcal_en

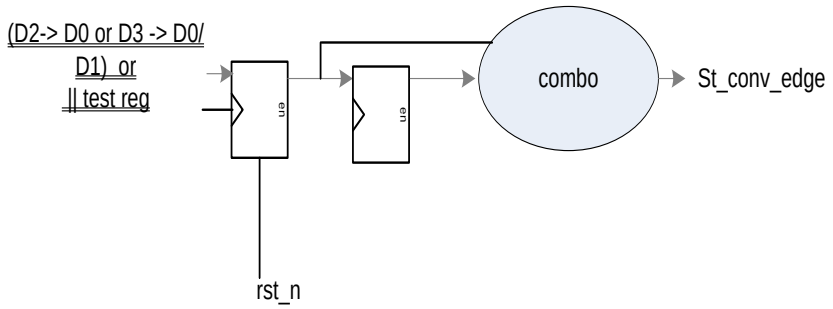
porta_offcal_en, is directly driven by enable signal from offset calibration state machine depending on the "conn_indx_path" parameter from the statemachine. We have CSDAC1 tied to portA, and CSDAC3 tied to portG. So, in total we must have 4 paths

- porta_offcal_en = (conn_indx_paths == 1) || (conn_indx_paths == 2 ) ? busy : 0;
    - porta_offcal_en runs on one SAR ADC with a 500KHz clock derived from charge-pump clock.
    - spkr_offcal_en should have separate SAR logic from other ports and runs at 15KHz clock and the clock might not be dependent on cpump clock. The specifications of the portG SAR logic will be same as porta,porte and portm. Spkr_offcal is directly driven by enable signal from the offset calibration state machine
    - spkr_offcal_en = busy;

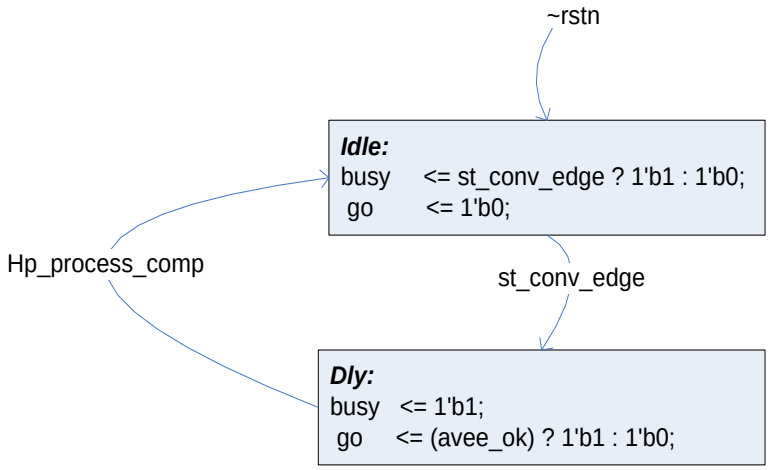
**Specification for offset calibration:**

1) Automatic offset calibration should be enabled during first good cad in HDA mode.
2) Automatic offset calibration should be enabled during first MCLK detection in I2C Host mode
3) Need flexibility to run offset calibration through I2C
4) Need flexibilty to run automatic offset calibration every D3/D2 to D1/D0 transition.
5) Need flexibilty to run manual offset calibration.
6) Testability to observe the offset calibrated offcal value and comparator output and flexibilty to force a offcal code should be available.
7) Chargepump should be enabled for every offset calibration mode of headphone, lineout and mono port. (first good cad, every D3/D2 → D1/D0 transition and manual calibration mode).
8) CSDAC should be enabled and put in mute during offset calibration.
9) CSDAC_porta_pdb and CSDAC_porta_mute is controlled through logic in analog during offset calibration.
10) Block beep signal (analog and digital) during the automatic calibration mode and in the path on which manual calibration is enabled
    a) Analog beep path:
        i) We need to block analog beep path when offcal_en is high this is done in analog by adding "offcal_en" to the mux selection control.
    b) Digital beep path:
        i) Digital beep path is blocked in analog as CSDAC is put in mute during offset calibration as mentioned in point #6.
11) "STATE_RESET" is the right state to be in when we are doing offset calibration we need to bring the statemachine back to "STATE_RESET" whenever porta_offcal_en is high.

The start of the state machine happens based on the rising edge of azalia control signals, I2S control signals or test register. Here below.

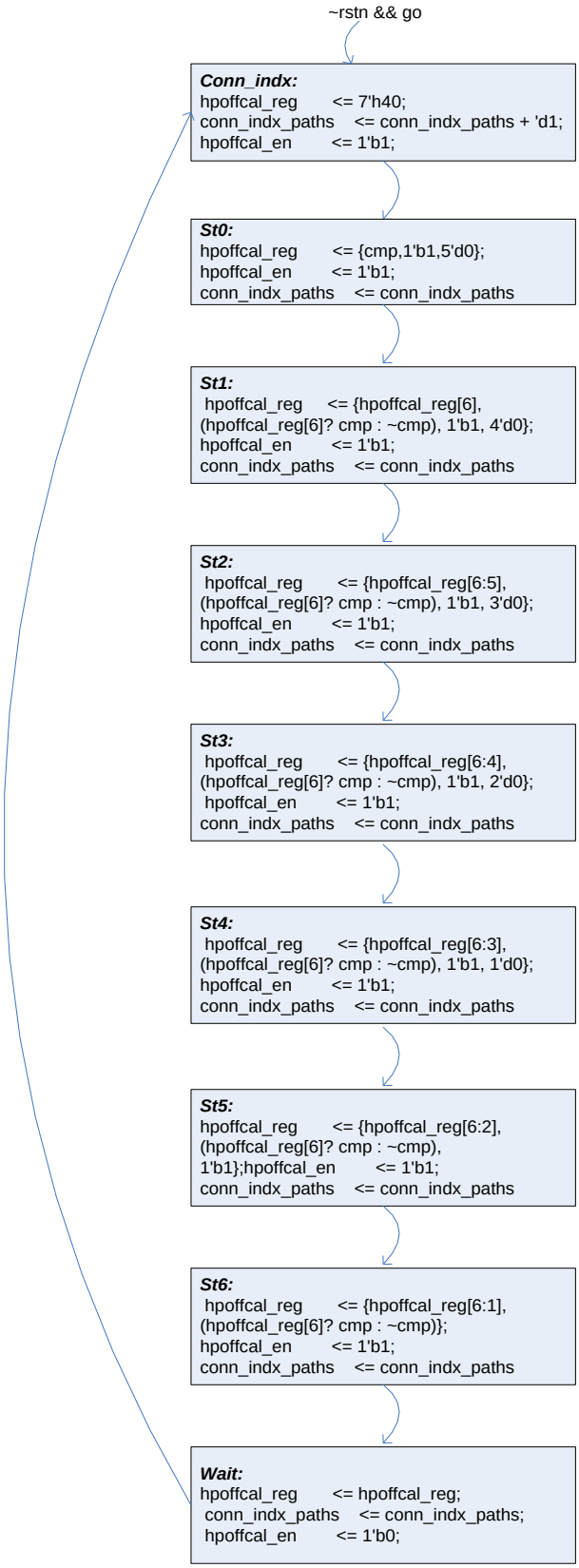



Enabling calibration SAR state machine for each of the DAC & Port combinations are done in the below states.




hp_process_comp = disable_calib || ((conn_indx_paths == 'd12) && (hp_sar_state == hp_sar_wait))

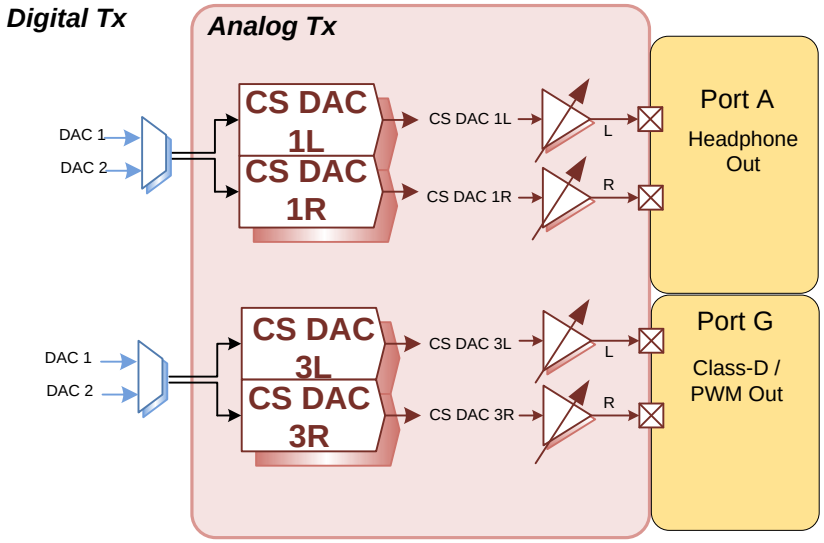
Offset Calibration SAR State machine:




Conexant Systems Inc
Proprietary and Confidential
ALL INFORMATION IN THIS DOCUMENT IS PRELIMINARY AND SUBJECT TO CHANGE WITHOUT NOTICE

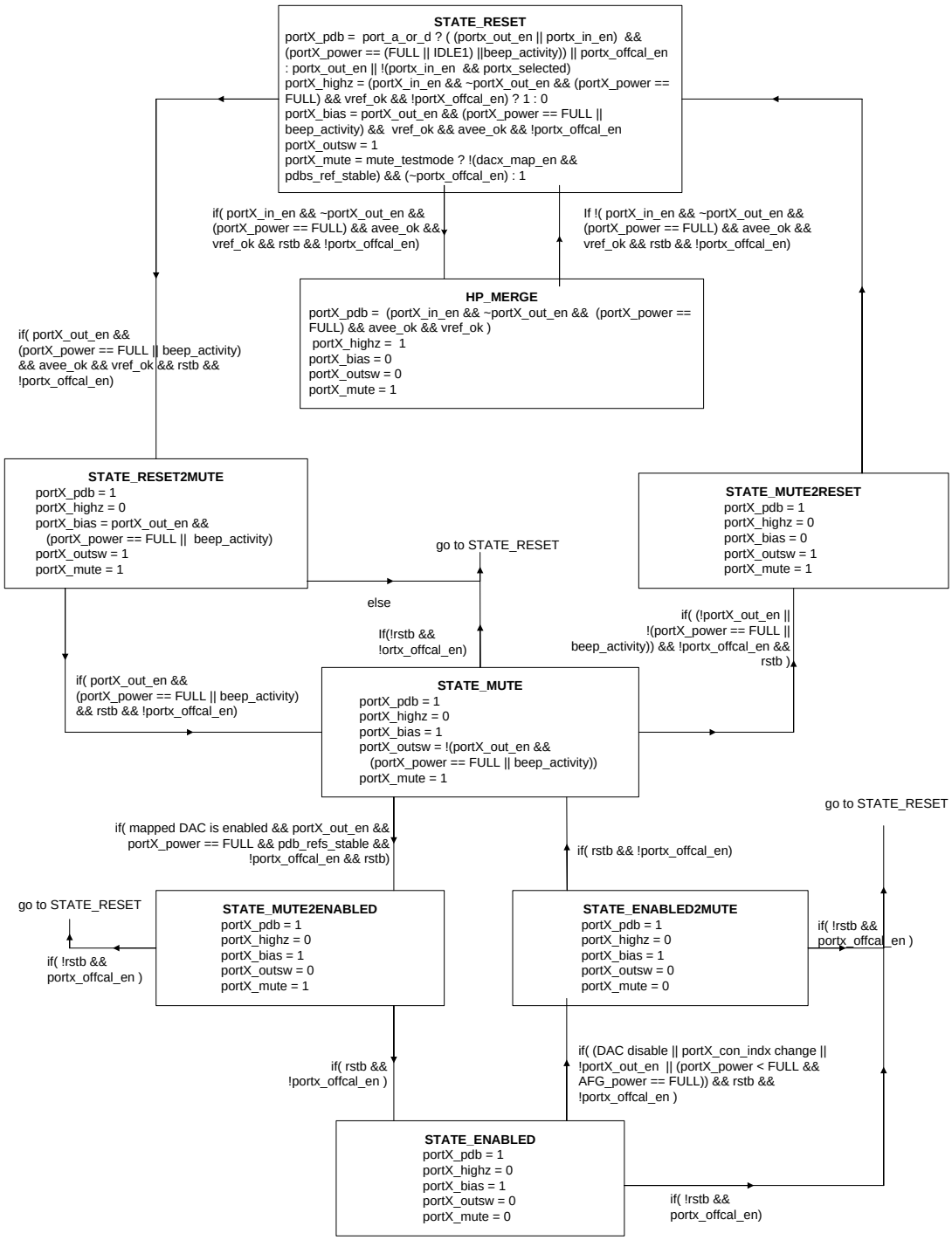
#### 4.3.7.3 porta_offcal_l[6:0], porta_offcal_r[6:0], spkr_offcal_l[6:0], spkr_offcal_r[6:0]

- we fixed one DAC per hp in analog so we have 1 DAC and 2 calibration registers are needed for porta.
- ClassD is connected to both DAC3 hence we have 2 calibration registers
- Total 4 Calibration registers is needed




#### 4.3.7.4 porta_pdb, porta_mute, porta_bias, porta_highz, porta_outsw

- These signals are controlled by a state machine called a port controller. Porta_out_en is the AND of the azalia output enable signal and extmuteb_dig, and shutdown_ana is the sw_vref output from the analog macro.
- mute_testmode signal used for porta_mute in port_controller state machine should be enabled by default.
- There is a test mode of operation that bypasses the entire state machine and the outputs are driven from test register.
- Whichever state the port is in make sure it comes back to state_reset when headset_detection is “high”




## 4.3.8 CLASSD control

The CLASSD blocks are part of node 17, the port G widget.

### 4.3.8.1 madc_code<4:0>

Monitor ADC code goes to PA input.

#### 4.3.8.2 temp_error

The temp_error is the status signals that indicated when an over temperature condition is detected.

- When temp_error goes high and we are not in self-recovery mode, the spkr port node (port G) should go to shutdown power state and output disabled. If the self-recovery mode is set the spkr port node must remain on (an analog local control will Hi-z the output as soon as the event is detected and release it after a while).
- Testbits are needed to bypass the shutdown action (testbits enabled by default. The bit is not protected)
- A status register that is cleared only when written to is needed to show any error event.
- The temp_error signal should be visible during a testmode on a gpio, in parallel with spkr_error.

#### 4.3.8.3 spkr_clk,

The 3.072MHz spkr_clk input is from the ClassD clock generator.

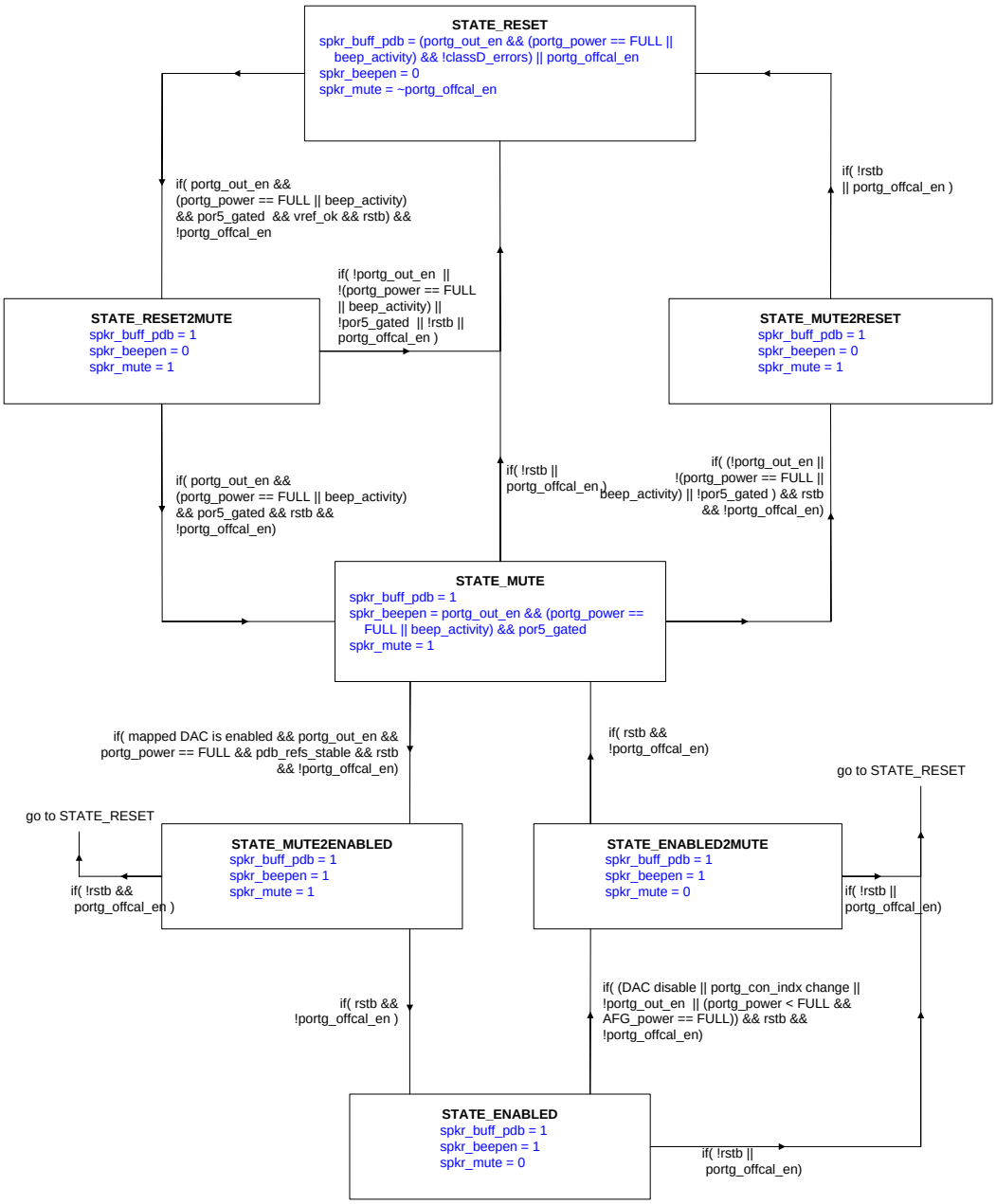
- The clock must be present in beep mode and D3 LIVE mode as well, so an RC clock should be muxed to the clock generator.

#### 4.3.8.4 spkr_monomode

Spkr_monomode is driven directly from non_azalia register. When spkr_monomode = 1 the output the right channel's internal pdb are overwritten to keep on only the left channel.
The mixing is done in analog.

#### 4.3.8.5 spkr_beepen,spkr_pdb, spkr_mute

- These signals are controlled by a state machine called a port controller. Portg_out_en is the AND of the azalia output enable signal and SPKR_MUTE(pin), and shutdown_ana is the sw_vref output from the analog macro.
- There is a test mode of operation that bypasses the entire state machine and the outputs are driven from test register.
- Single spkr_pdb control for both L and R channel. If spkr_monomode = 1 then only L channel will be tuned on in AFE. These signals are derived from the portg_out_en azalia signal.




### 4.3.8.6 spkr_test_out[1:0]

These are test outputs so that the ClassD comparator outputs can be observed on gpios.

### 4.3.8.7 spkr_test_in[1:0]

The spkr_test_pwm bus is a test signal used to feed in a pwm signal from off-chip. This should come from digital input pads.

### 4.3.8.8 spkr_oc_l,spkr_oc_r

spkr_oc_l/r is the over current detect signals. These signals are from the analog comparators and need to be filtered.

#### 4.3.8.9 spkr_sc_status

spkr_sc_status is the short circuit detect status signal. Short circuit is handled locally

- When spkr_sc_status goes high and we are not in self-recovery mode, the spkr port node (port G) should go to shutdown power state and output disabled. If the self-recovery mode is set the spkr port node must remain on (an analog local control will Hi-z the output as soon as the event is detected and release it after a while).
- Testbits are needed to bypass the shutdown action (testbits enabled by default. The bit is not protected)
- A status register that is cleared only when written to is needed to show any error event.
- The spkr_sc_status signal should be visible during a testmode on a gpio, in parallel with temp_error.

#### 4.3.8.10 spkr_offcal_en

enable the offset calibration system same as headphone port offcal enable

#### 4.3.8.11 spkr_offcal_l[6:0], spkr_offcal_r[6:0]

DAC control bits driven by the common SAR logic

#### 4.3.8.12 spkr_offcal_cmp_l, spkr_offcal_cmp_r

Left channel comparator output

### 4.3.9 CPUMP control

The CPUMP block is not directly part of any node, but is a support block for nodes 16 (port A).

#### 4.3.9.1 cpump_ctrl_dmode

This signal is driven by envelop detector output. A envelop detector needs to be added on dac1l/r data path inputs to AFE to control the chargepump supply output.

#### 4.3.9.2 cpump_pdb

The cpump_pdb signal is derived from the AFG power state, porta/portd_out_en azalia signals, porta/portb_offcal_en signals and needs to be driven by the offset cancelation system enable bit to enable the clock.

- cpump_pdb = ( (porta_out_en || porta_in_en) && (porta power is FULL or IDLE1 || beep_activity)) || (portd_out_en && (portd power is FULL or IDLE1 || beep_activity)) || cpmp_clk_en
- cpmp_clk_en enables charge pump for offset calibration state machine to start during power state change from D3/D2 to D0/D1,first good cad and manual calibration mode in HAD mode
- In I2S mode
- There should also be a testbit control.

#### 4.3.9.3 cpmp_clk

The cpmp_clk is 6.144MHz. This is derived from the cpmp clock generator. The enabled state of the clockgen derived from the cpump_pdb signal, and the cpump_pdn_ok.

- enabled if (cpmp_pdb == 1 || cpmp_pdb_okb == 1).
- The clock must be present in beep modes as well, so an RC clock should be muxed to the clock generator.

#### 4.3.9.4 cpmp_pdb_okb

- This output is a status signal that indicates when the charge-pump has discharged avee to 0V during disable.
- cpmp_pdb_okb = !(avee has reached 0V during cpump power-down && portA has completed power-down sequence).

#### 4.3.9.5 cpmp_vout_ok

- This output is a status signal that indicates that the both charge pump rails have reached its desired value during start-up.
- cpmp_vout_ok = vpos and vneg have reached its desired value during start-up && portA has completed power-up sequence.

### 4.3.10 PCBEEP control

#### 4.3.10.1 beep_ana_en

Beep_ana_en enables the analog beep path out of ports A, C, D, and E. These will play beep if they are output enabled. The analog beep path is enabled if the digital is not capable of playing beep (e.g. codec clock is disabled).

- beep_ana_en = codec clock disabled

#### 4.3.10.2 beep_gain_porta[3:0], beep_gain_spkr[3:0]

The beep_gain[3:0] control values are calculated from the beep_gain[2:0] register, second level BIO configurated beep level used for headphone and the corresponding DAC volume level.

- beep_gain_porta = -10 + (porta_beep_gain + spkr_beep_gain) + (porta_sel ? floor(dac1_volume/4) : floor(dac2_volume/4))
- beep_gain_spkr = -10 + spkr_beep_gain + (spkr_sel ? floor(dac1_volume/4) : floor(dac2_volume/4))

There should be a test bit to bypass the above equations and allow register control of beepX_gain. The 3-bit register of digital beep_gain_X can be reused, requiring only 1 extra bit for the MSB of beep_gain_X. In this bypass mode, send the same 4-bit value to beep_gain_hp and beep_gain_spkr. where *beep_gain* is the 3-bit register value that controls overall beep gain, and dacX_volume is the average of DACX left and right volume settings (post any classD gain boost). The min value of beep_gain_X is 0, and the max of beep_gain_X is 15.
Also, porta_beep_gain and porte_beep_gain are headphone beep gain 3bit gain value set by the bios for headphone. And spkr_beep_gain is the spkearker beep gain 3-bit value set by the OS.

#### 4.3.10.3 beep_pdb

The beep_pdb is used to toggle power to the analog beep PGAs. When there is no beep present, the PGAs are disabled to save power.

- beep_pdb = debounce (analog beep path enable)

#### 4.3.10.4 beep_dig and beep_dig_en

- Beep_dig and beep_dig_en are output of the beep circuitry.
- Beep_dig_en indicates when a beep is present (i.e. when the signal at the pad is less than avdd/3 or above avdd*2/3. This can toggle on every edge of a beep signal, so should be debounced after three consecutive edges. Beep_dig_en will toggle irrespective of beep_ana_en. Use the debounced signal to enable the clocking to wakeup circuit

- Beep_dig is the actual beep signal. This will toggle irrespective of beep_ana_en.

#### 4.3.10.5 beepgen_dig and beepgen_dig_en

- beepgen_dig and beepgen_dig_en are directly derived from beep_gen and beep_gen_en respectively.

### 4.3.11 JACKSENSE control

#### 4.3.11.1 jack_pdb

Analog jacksense comparator should be enabled 25us before the SAR ADC in D3_lp_mode.

#### 4.3.11.2 jack_dsw[3:0], jack_sample, and jack_comp,

The jac_dsw[3:0], jack_sample, and jack_comp are the signals that connect the DAC and comparator to the SAR logic to form the complete successive-approximation ADC for jack sensing. SAR logic is running on 200KHz clock generated from lf_rcosc_clk.

**In multi-jack sensing applications**

- jack_pdb = every 10ms in D3_lowpower mode.
- The clock to SAR logic should be gated and enabled after 25us of toggling jack_pdb signal.

**In single jack applications**

- Jack_pdb = every 10ms in D3 low power mode
- The clock to SAR logic is completely gated and SAR logic is power down in this mode
- The deglitched jack_comp signal is used as port presence signal.

### 4.3.12 CLOCK's

#### 4.3.12.1 adc_clk, dac_clk → 3.071MHz

#### 4.3.12.2 spkr_clk → 3.072MHz

#### 4.3.12.3 cpmp_clk → 6.144MHz

### 4.3.13 POR control

#### 4.3.13.1 por_core

The por_vdd_core signal is a 1.2V signal from the 1.8V to 1.2V digital regulator that indicates the state of the 1.2V supply. It is an active-low reset signal.

#### 4.3.13.2 por_5

por_5 signal is a 1.2V signal that indicates the state of 5V supply. It is an active-low reset signal

#### 4.3.13.3 por5_thrshsel

por5_thrshsel is a input signal which is the por_v threshold voltage . when it is "0" por threshold is set to 0.875V and "1" the threshold is set to 1.52V. The signal is controlled by below mentioned conditions

- madc voltage control is >4.7V. madc[4:0] → >= 22
- must have testbit control to overwrite the above madc control.

# 5 Chip Level Overview

## 5.1 System Overview

The Oceanside device provides audio codec functionality for portable PC and tablet systems. At a high level, the feature set includes the following.

- Intel HDA (High Definition Audio) interface to hosting system
- I2C / I2S interfaces to hosting system
- SPDIF output port
- I2C peripheral control interface
- Dual digital microphone input
- Two stereo DACs
- One stereo ADC
- One stereo cap-less headphone/line driver
- One stereo line level driver
- One stereo class-D speaker driver
- One stereo microphone / line level input receiver (shared by 4 input ports)
- Microphone bias generation
- Jack insertion sensing
- Apple/Nokia plug detection
- Universal jack support
- 7-Band Equalizer and Dynamic Range Control for outbound audio streams
- D3Live support
- Five GPIO pins, with interrupt generation capability.
- Smart Amp 1.0 (Smart Speaker Protection Support).
- Multi-Button Inline Support with Tip-Ring Impedance Detection.
- 2 peripheral I2S/TDM interfaces (each with input and output ports)
- Master I2C interface for controlling peripherals.

### 5.1.1 Intel HDA (High Definition Audio) Host Interface

The Intel HDA interface is one of two mechanisms available for a hosting device to interface with the Oceanside device. The other mechanism consists of the combined operation of the I2C and I2S host interfaces (below). At power-on, HDA vs. I2S interface operation is selected with the HOST_BUS_MODE die pad (0=HDA, 1=I2S). This die pad may be bonded internally to the package, and therefore unavailable at the system level.

The device pins associated with the HDA interface are re-used for the host I2S interface when that interface is selected. However, the pins associated with the I2C interface are independent of the HDA interface, and may be used for register access, even if the HDA bus mode is selected. An internal register (accessible through either HDA or I2C) can be used to override the host bus mode selected by the HOST_BUS_MODE die pad.

As per the HDA specification, the hosting device always masters (and provides clocking for) the HDA interface. The HDA bit clock is used as the primary audio clock reference within the Oceanside device. However, and internal RC oscillator is also provided, for low power states, in which the HDA bit clock is not available.

The HDA interface is compliant with the Intel High Definition Audio Specification, revision 1.0a. It also supports the low clock frequency and reduced pin count options found in ECR HDA048-A.
Additional details are provided in the Solana/Neptune share-point, as well as in the Intel HDA specification document.

### 5.1.2 I2C Host Control Interface

The I2C host control interface is the second of the two mechanisms available for a hosting device to control the operation of the Oceanside device. The other mechanism is the HDA interface (above). The I2C host control interface may be active simultaneously with the HDA interface. However, writing to the same register simultaneously through both interfaces will produce unpredictable results.
The hosting device always masters (provides clocking for) the I2C host interface.
Additional details are provided in the chapter of this document, as well as in the NXP I2C specification document, revision 4, dated February 13, 2012.

### 5.1.3 I2S Host Audio Data Interface

The I2S host audio data interface is the second of two mechanisms available for a hosting device to transfer audio sample data to and from the Oceanside device. The other mechanism is the HDA interface (above).
Unlike with the HDA interface, Oceanside may be the clock master or slave. Either the I2S bit clock or the MCLK input may be used as the primary audio clock reference within the Oceanside device.
The I2S host interface supports several operating modes, including a multi-channel mode (referred to as the TDM mode).
Additional details are provided in the chapter of this document.

### 5.1.4 I2C Peripheral Control Interface

The I2C Peripheral control interface allows the host device to control the operation of a peripheral device connected to Oceanside.
The I2C Peripheral Control Interface is the master (provides the clock). Multiple masters are not supported.
Additional details are provided in the chapter of this document, as well as in the NXP I2C specification document, revision 4, dated February 13, 2012.

### 5.1.5 I2S Peripheral Audio Data Interface

The 2 I2S peripheral audio data interfaces allow for bridging for the host device to one or more I2S peripheral devices such as smart amplifiers or additional audio CODECs (as in a docking implementation.)
The I2S host interface supports several operating modes, including a multi-channel mode (referred to as the TDM mode). The peripheral audio interfaces may operate as clock masters or slaves.
Additional details are provided in the chapter of this document.

### 5.1.6 SPDIF Peripheral Audio Data Output Interface

Oceanside provides an SPDIF peripheral audio output interface.
Additional details are provided in the **Spdif** TX document in Reno share-point, as well as the IEC 60958 standard.

### 5.1.7 Digital Microphone Inputs

Oceanside supports up to 4 pulse-density-modulated digital microphones. A pair of microphone data input pins support up to 2 microphones each. A pair of microphone clock output pins allow flexible on/off control of the attached microphones.
Additional details are provided in the Solana Digital MIC document in Solana share-point.

### 5.1.8 Miscellaneous Digital Signals

Oceanside provides the following digital I/O, in addition to the interfaces described above:

- EAPD (External Amplifier Power Down) output.
- D3Live request input
- Digital PC beep input
- Speaker Mute input
- Up to **5** GPIOs (some GPIO share pins with other functions).

Additional detail on EAPD and Speaker mute is provided later in this document.

### 5.1.9 Cap-less Headphone Output Driver

Oceanside provides one stereo cap-less headphone output drivers. An on-chip charge pump is used to generate a negative power supply, allowing the cap-less output driver to drive symmetrically around the chip ground potential, and eliminating the need for DC blocking capacitors.
Additional details are provided by analog documents in the Solana/Neptune analog share-point.

### 5.1.10 Class-D Speaker Driver

Oceanside provides one stereo Class-D speaker driver.
Additional details are provided by analog documents in Solana/Neptune share-point.

### 5.1.11 Microphone / Line-Level Input Receivers

Oceanside provides one stereo receiver for microphone or line-level inputs.
It should be noted that the input receiver is multiplexed across three separate input ports (B, C, D, see below), any one of which may be active at any point in time.
It can also be noted that while port D provides two separate input pins, only one may be active at any given time, providing a mono, rather than stereo, input. Selection between these two input pins is made automatically, depending on whether an “Apple” or “Nokia” style plug is detected.
Additional details are provided by analog documents in the Solana/Neptune share-point.

### 5.1.12 HDA Port Definitions

It is customary that HDA codecs refer to external audio interfaces as “Ports”.
The following table summarizes the HDA ports defined within the Oceanside device

| Port Name | Primary Function | Re-Taskable Function |
|---|---|---|
| A | Stereo Cap-less Headphone/Line Level Output | |
| B | Stereo Analog Microphone/Line Level Input | |
| C | Stereo Digital or Analog Microphone Input | |

| | |
|---|---|
| D | Mono Microphone/Line Level Input |
| E | Stereo I2S1 output |
| F | Stereo I2S1 input |
| G | Stereo Class-D Output |
| H | Stereo Digital Microphone Input |
| I | SPDIF Output |
| J | Stereo I2S2 output |
| K | Stereo I2S2 input |

NOTES:
Ports A and D are used together to provide a 4-conductor headset jack function.
Port D (mono mic input) is available on 2 pins. The selection of which pin is active is automatic, based on the detection of an 'Apple' vs. 'Nokia' style plug inserted into the associated headset jack. It should be noted that port D identifies itself as stereo, although only 1 of the 2 available pins may be active at any given time.
Port B is used along with Ports A and D to provide a 4 conductor 'universal' jack function. In this situation, the Port A pin widget is merged into the Port B pin widget, to provide an Input/Output pin widget. (Port A pin widget becomes 'vendor specific'.)

### 5.1.13 Multi-Button Inline Support and Tip-Ring Impedance detection

Communications headsets commonly provide control buttons integrated into the headset. The most common implementation is a single button that shorts the microphone input to ground. This button is typically used for call answer, call end, mute, and unmute depending on the duration of the button press. Many headsets provide two additional buttons for volume up/down and other functions. These buttons connect a resistance across the microphone to change the voltage measured on the mic pin. These resistances may be linear devices such as resistors or nonlinear devices such as diodes.

Oceanside will implement support for single button detection or multi-button detection. The single-button detection circuit will detect short and long switch closures and generate unsolicited responses (including wake events) for button press and release events.

Multiple button support is targeted for 4 buttons as described by Google. The multi-button detection scheme should also support 3 buttons as an option. The detection circuit will detect short and long switch closures and generate unsolicited responses (including wake events) for button press and release events.
Additional design detail provided in Solana/Neptune share-point and later in this document.

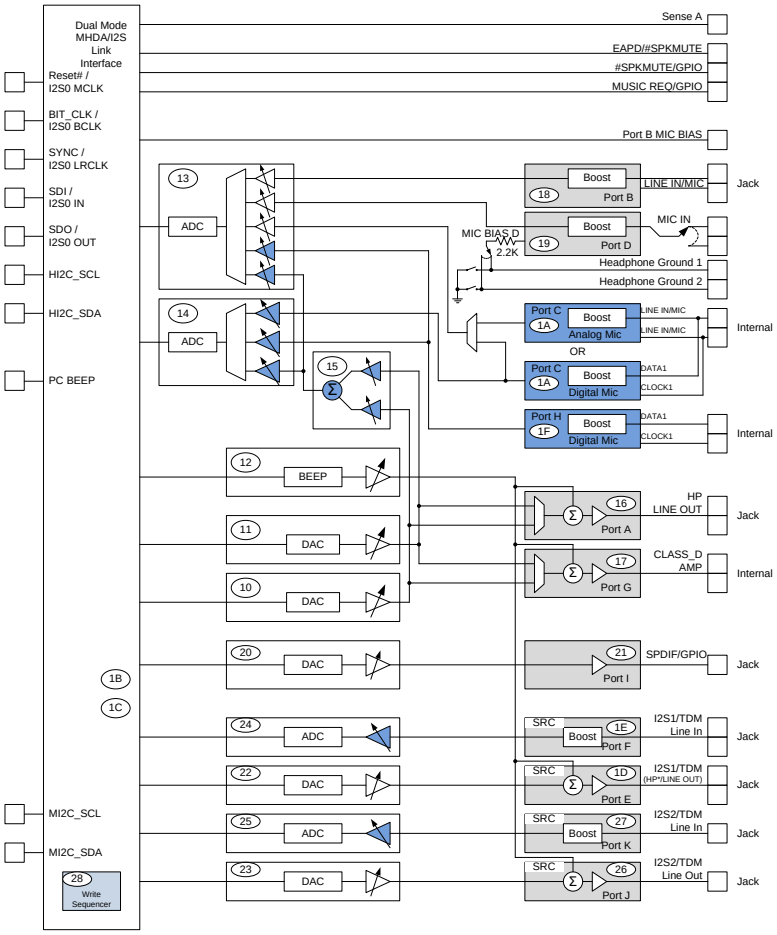
### 5.1.14 Smart Amp 1.0

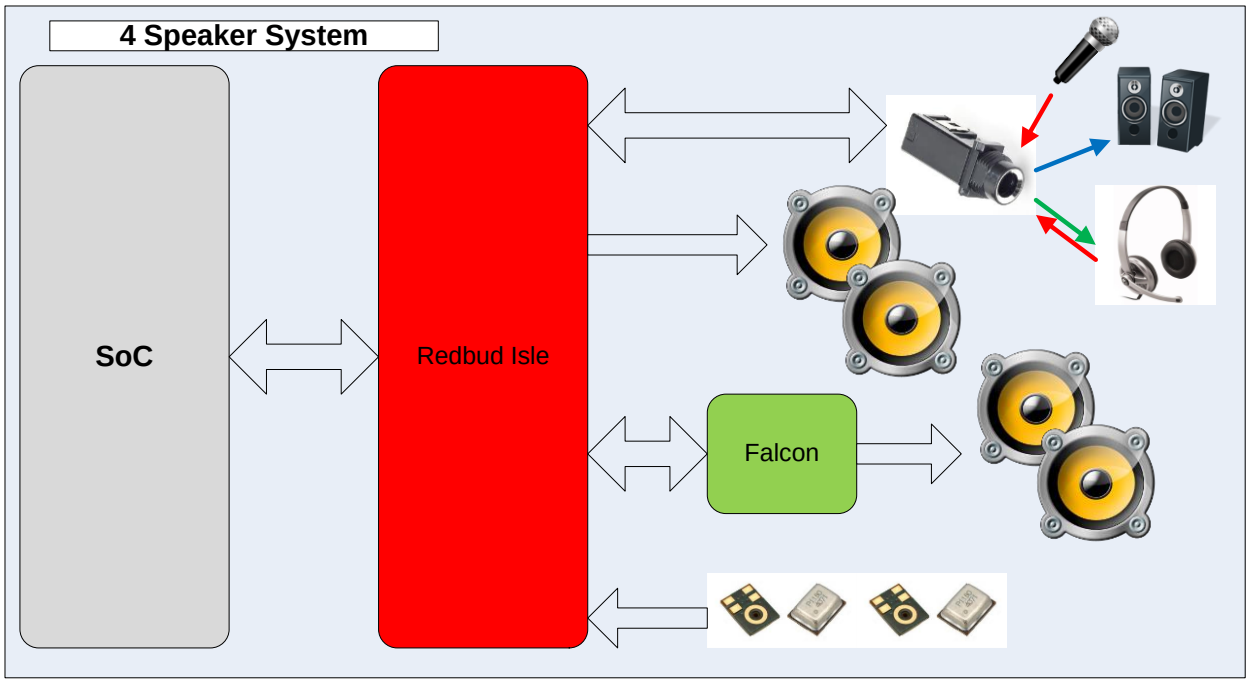
The Smart Amp function provides real-time protection against speaker damage due to high signal levels, by detecting conditions which might cause damage, and automatically lowering the power level driven into the speaker. This, in turn, allows average volume levels to be increased, resulting in an overall increase in perceived loudness. Smart Amp functionality has been provided in previous devices. Those implementations are referred to here as SmartAmp-1 (or SA-1). SA-1 uses the on/off state of the Class-D differential output drivers, instead of the measured current and voltage.

Additional design details provided later in this document.

## 5.1.15 System Diagram

The diagrams on the following page represent the logical block diagram and a High-Level System Diagram example of Oceanside.






### 5.1.16 Data Flow Topology

The diagram on the following page represents data flow through the various functions of the Oceanside device. This diagram does not represent the device design hierarchy

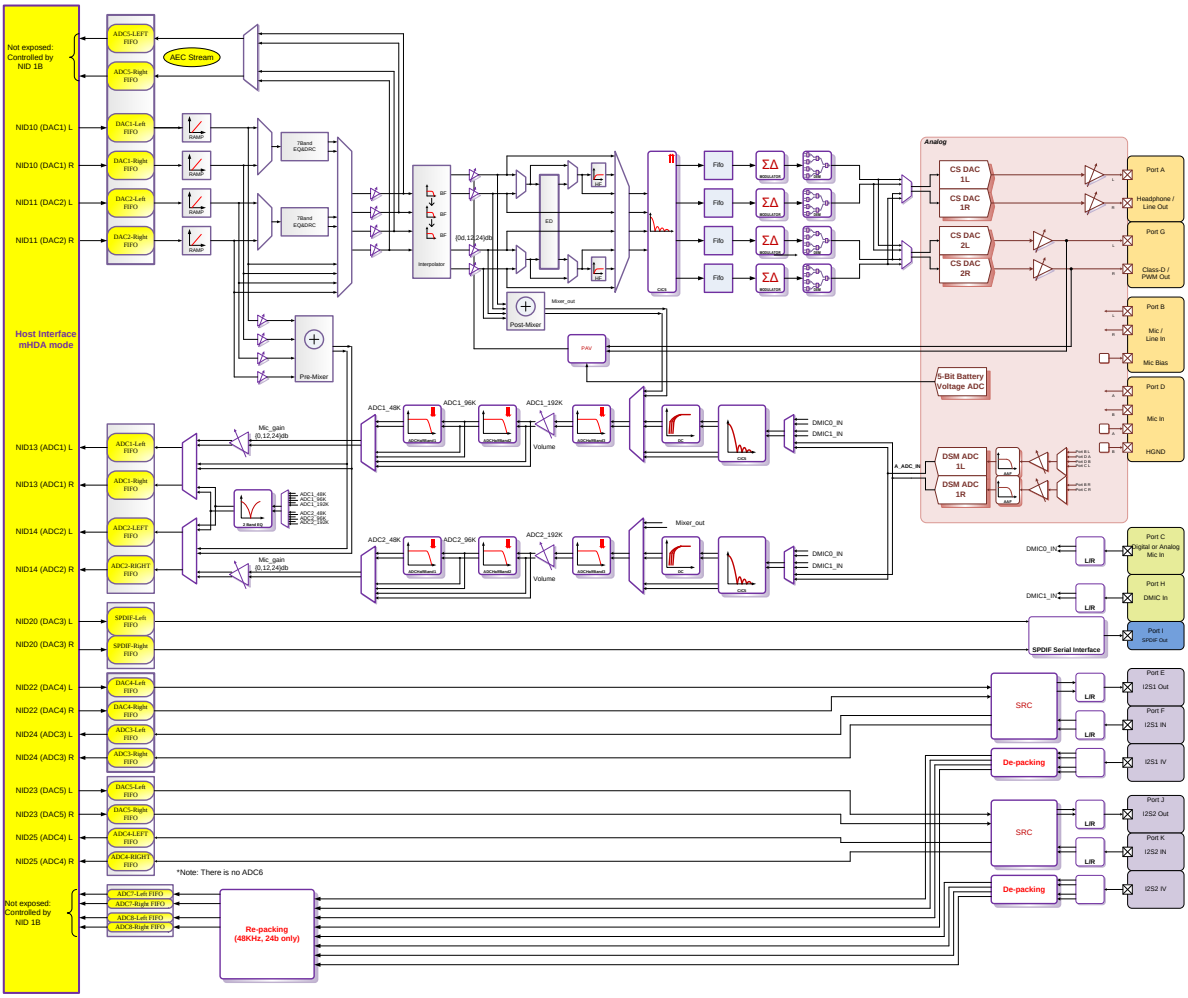



**Figure 5.1-1 Functional Block Diagram HDA host mode (CX8400)**

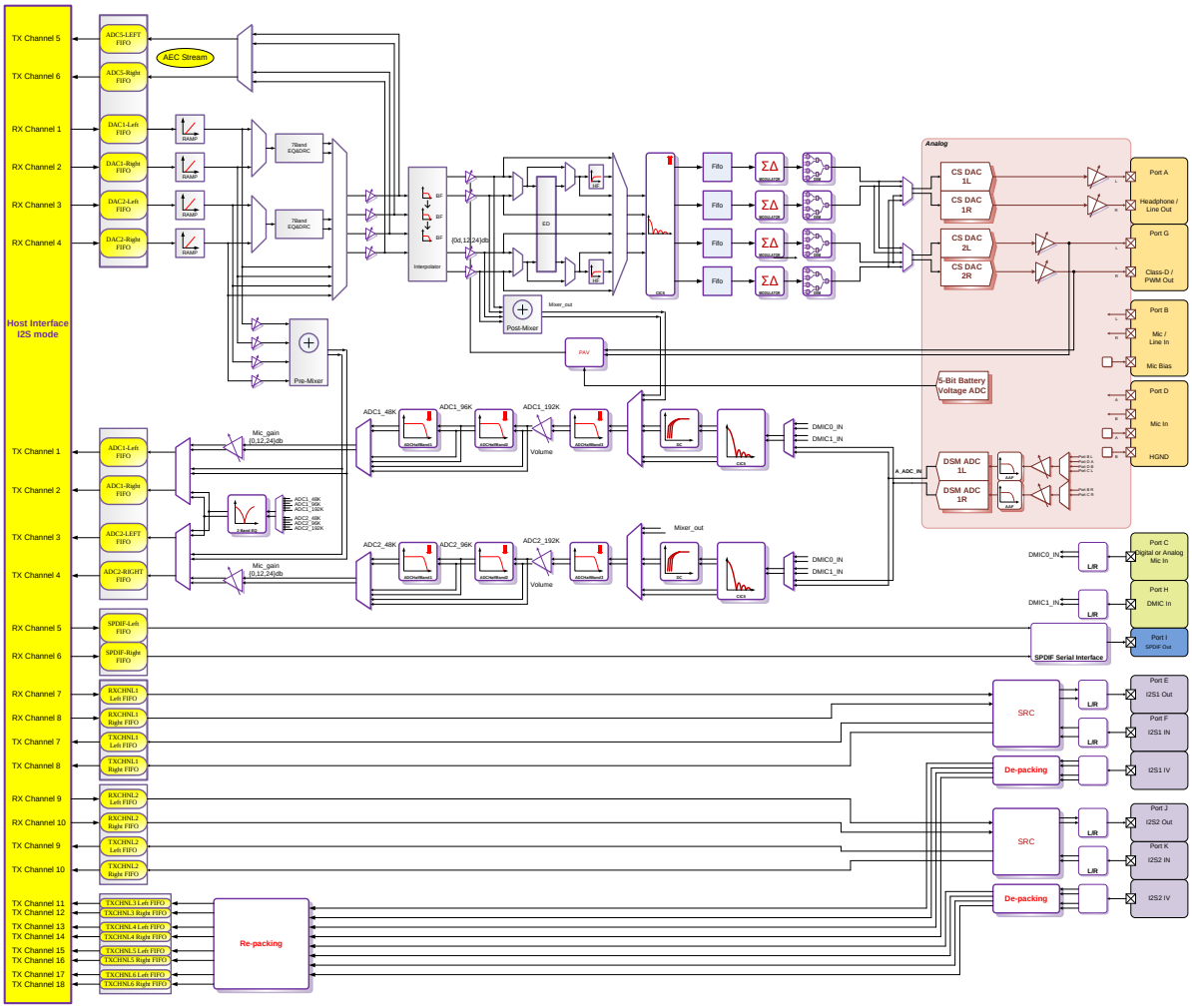



**Figure 5.1-2 Functional Block Diagram I2S host mode (CX8450)**

## 5.1.17 SoC Block Diagram

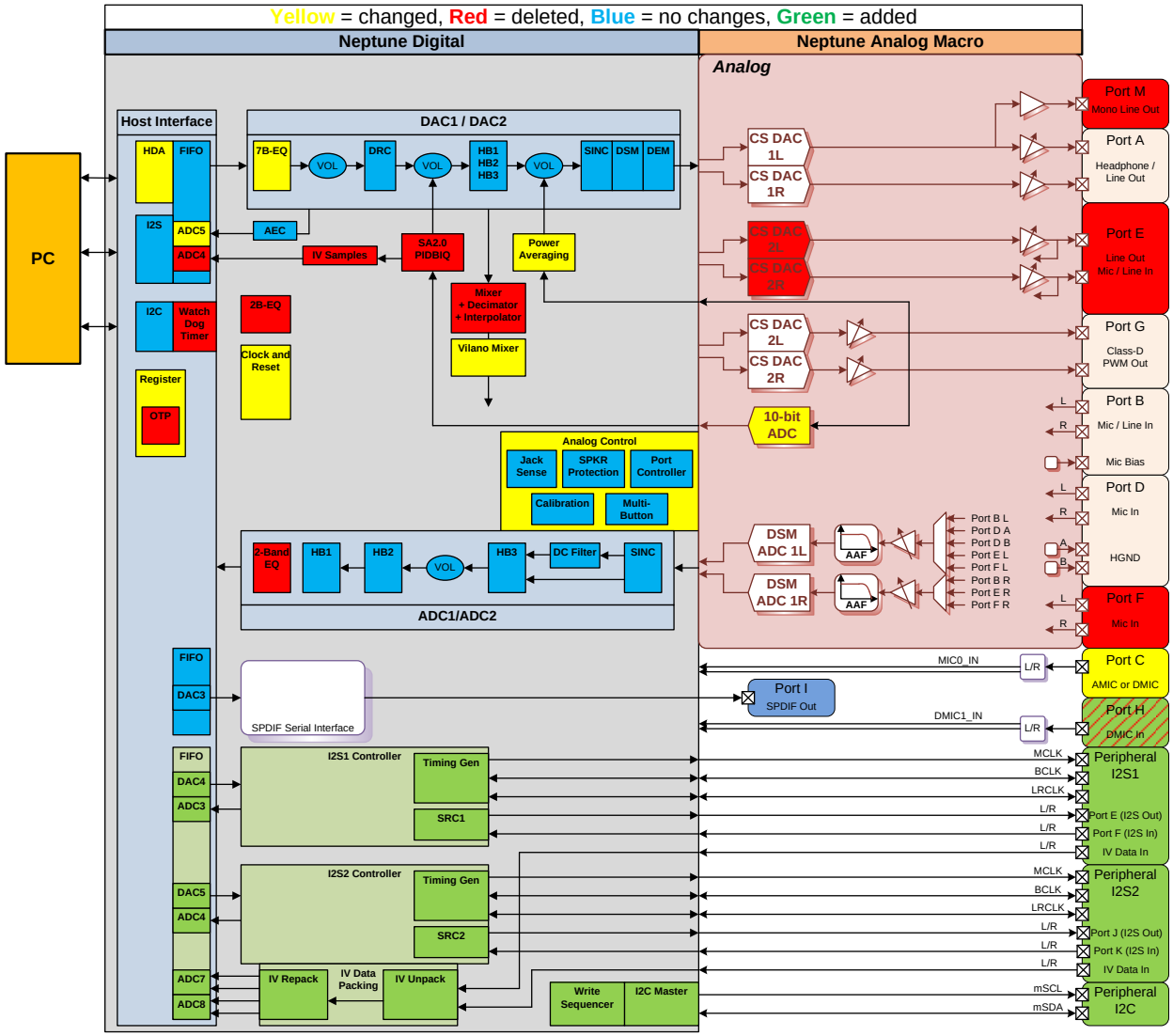



**Figure 5.1-3 SoC Block Diagram (Showing Changes from Neptune Block Diagram)**

## 5.1.18 Clock Relations

Sync group 1: {HSCL I2C_INT_CLK}
Sync group 2: {CLK_24P5 CLK_12M CLK_6M CLK_3M CLK_48K CLK_768K}
Sync group 3: {CLK_DC_DET OC_CLK}

## 5.1.19 Oceanside Clock Diagrams

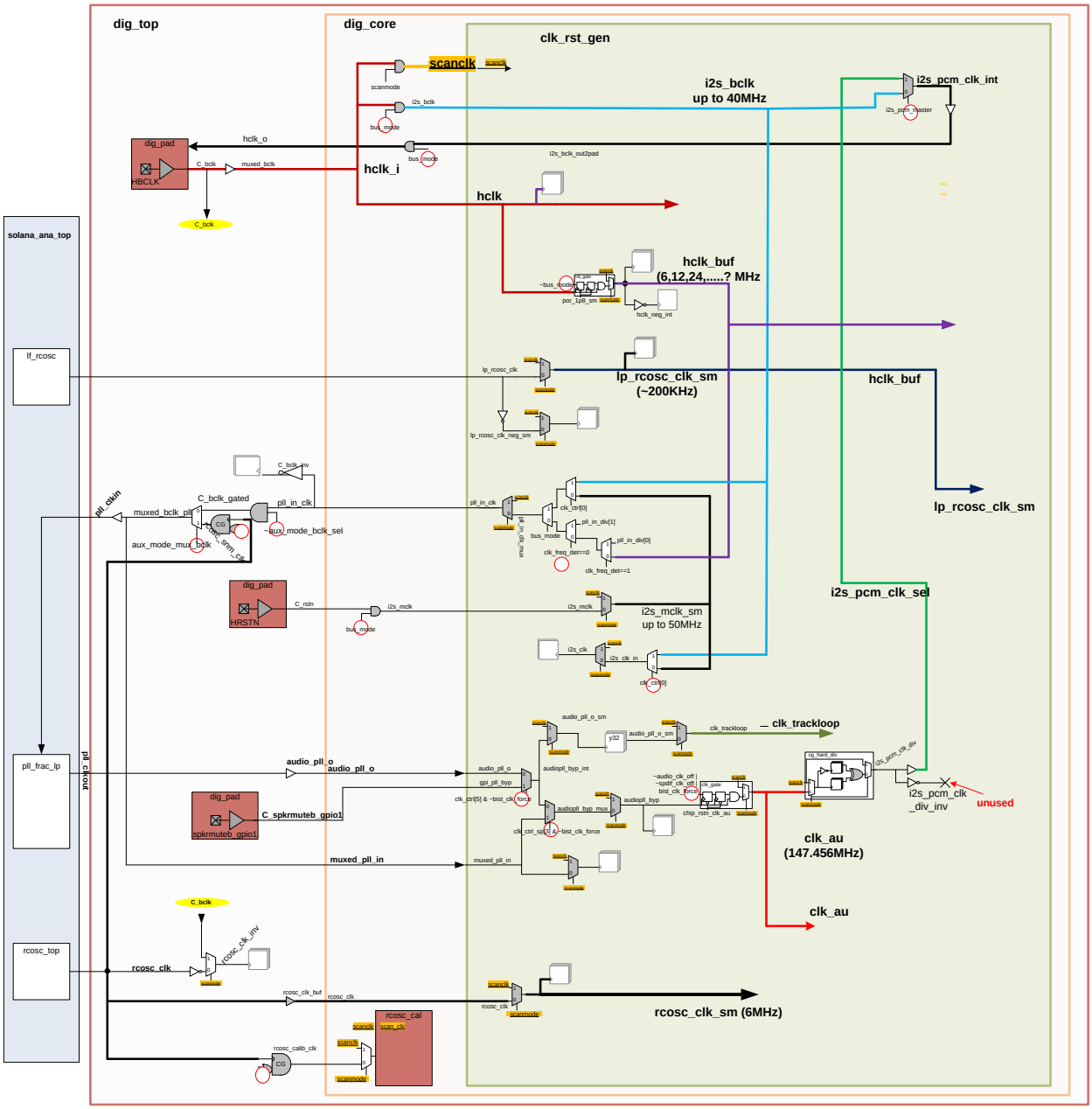



**Figure 5.1-4 Main Clock Diagram (CLK_SRC)**

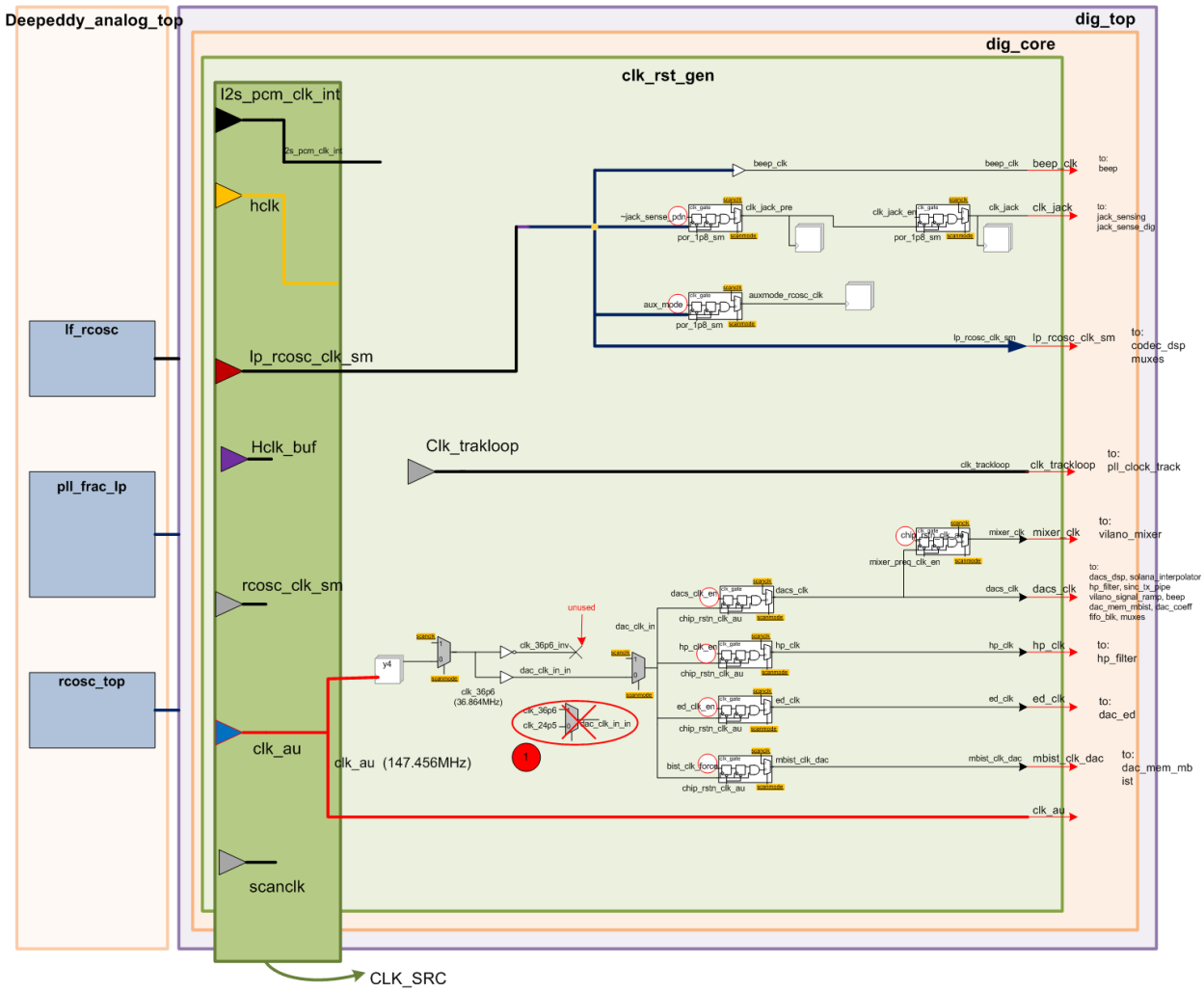



**Figure 5.1-5 Host Clocks, (Register, FIFO, Azalia, I2S)**

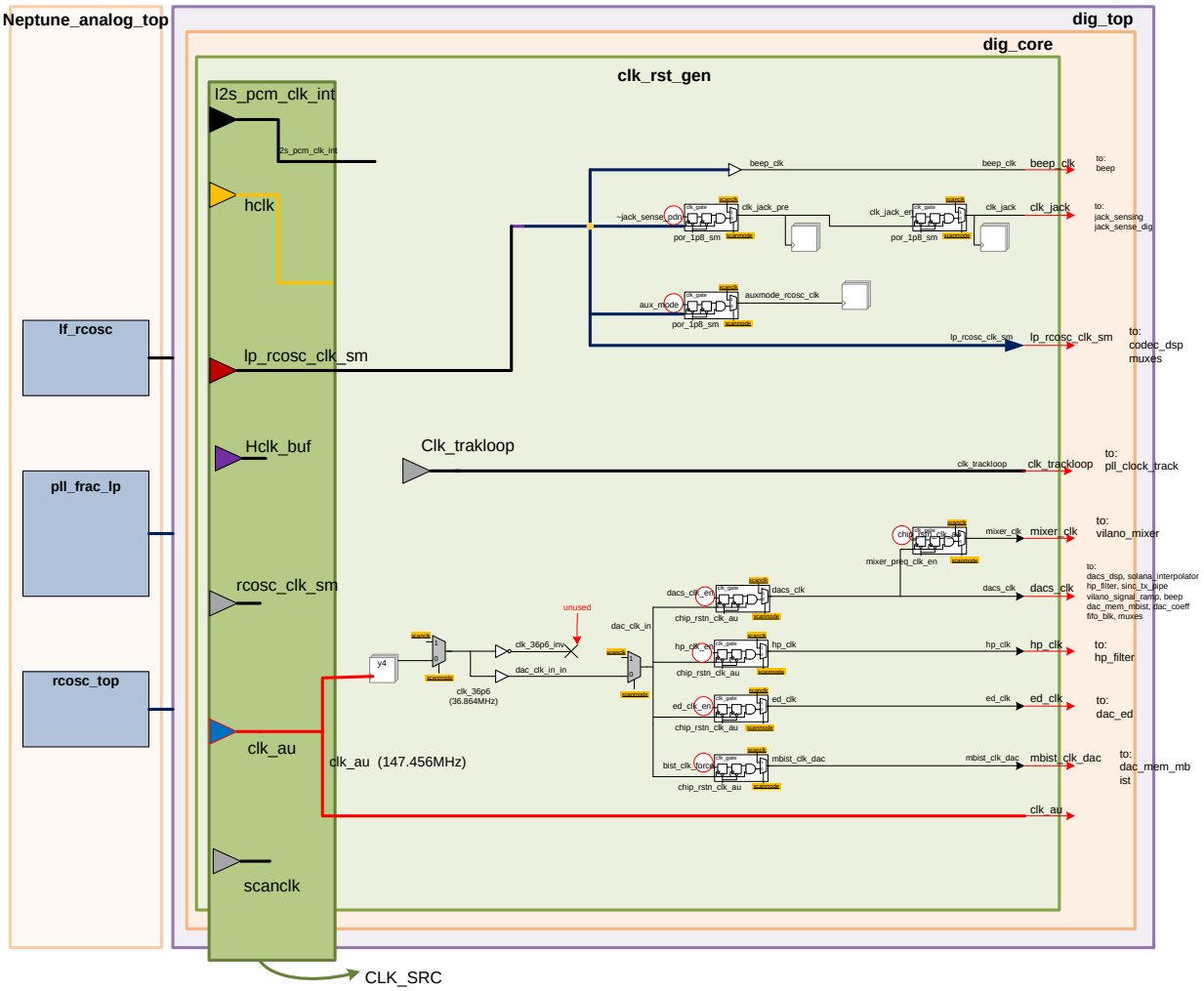



**Figure 5.1-6 Clocks Generated from clk_36p6, lp_rcosc_clk**

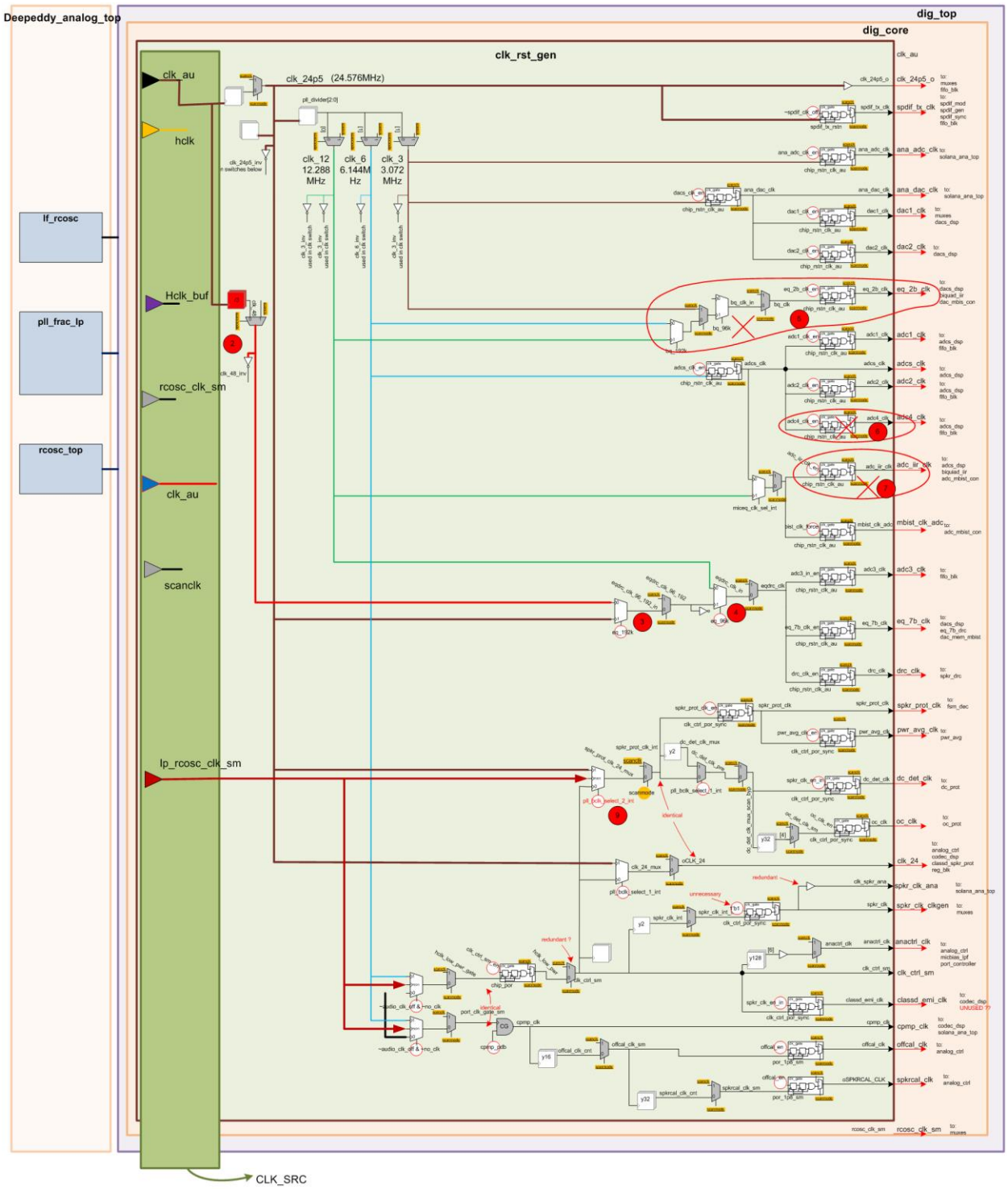
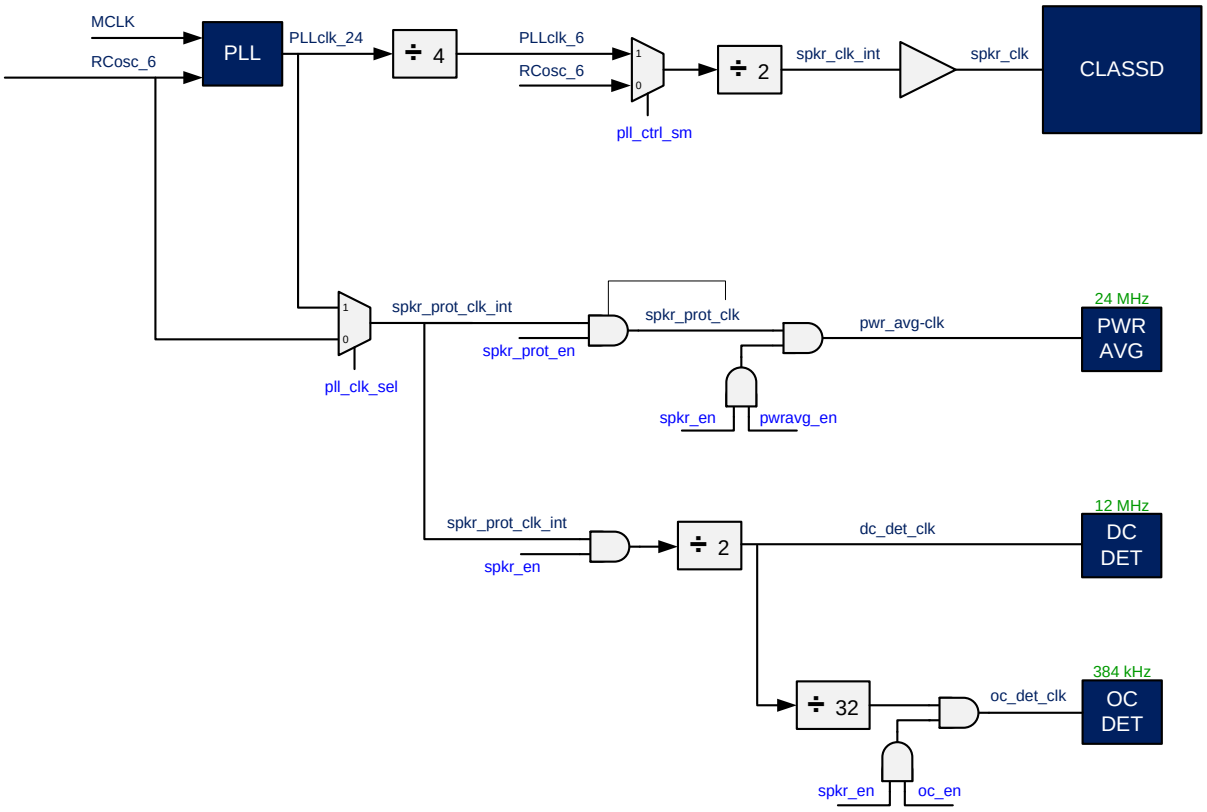


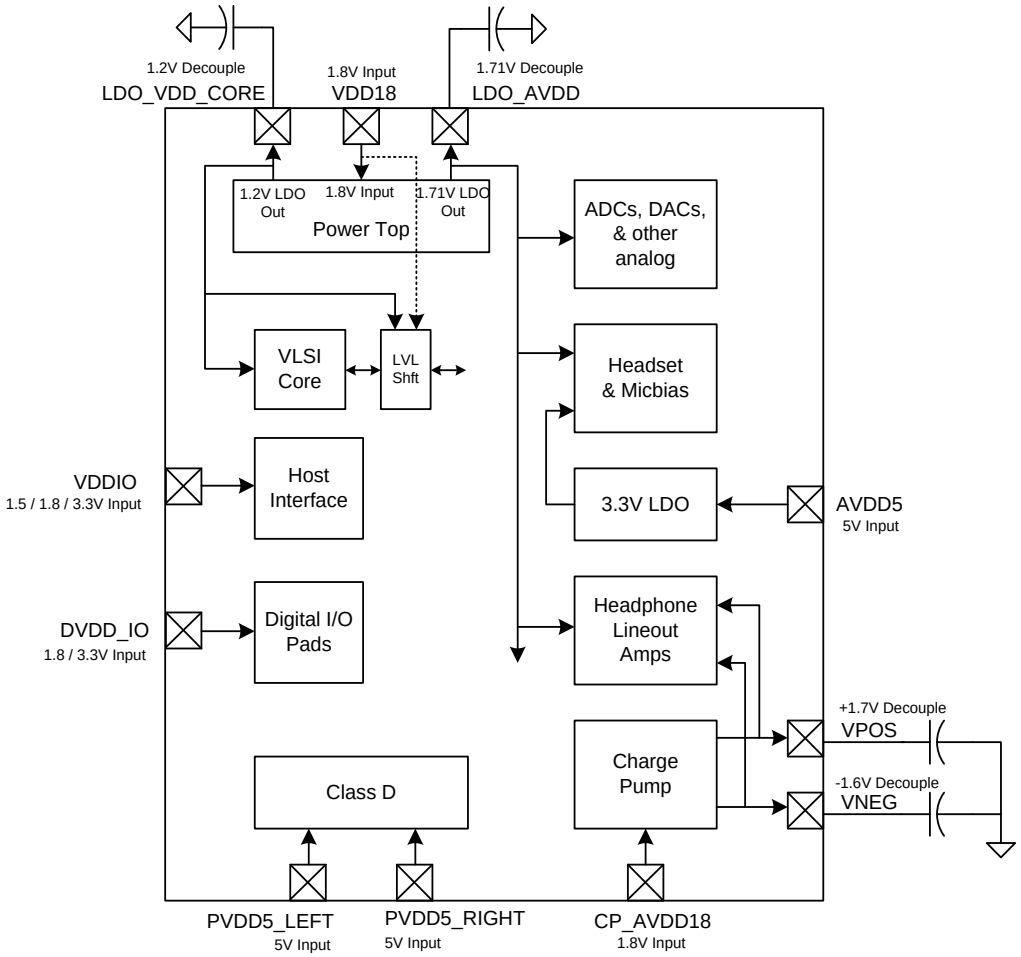

**Figure 5.1-7 Clocks Generated from clk_au (clk_24p5, clk_3 clk_6, clk_12).**

## 5.1.20 Speaker Protection Clocks Block diagram




## 5.2 Device Power Supplies

### 5.2.1 Power Block Diagram




### 5.2.2 Power supplies Specification

- VDDIO - Sets the host interface signaling level.
  - 1.5V / 1.8V / 3.3V, +/- 5%
  - Connect to the same power supply as the host bus controller.
- DVDD_IO - Digital I/O Level.
  - Connect to 1.8V / 3.3V, +/- 5%
  - Signaling Level for the DMIC, SPDIF, EAPD, & GPIO
- PVDD5_LEFT, PVDD5_RIGHT – Class D Speaker Amp power supply voltage inputs
  - Connect to 2.7 to 5.5V system power supply
- CP_AVDD18 – Charge Pump power supply input
  - Connect to 1.8V +/-5% system power supply
- AVDD5 – Internal 5V→3.3V LDO supply voltage input
  - Connect to 2.7 to 5.5V system power supply
- VDD18 – Main Power_Top supply voltage input.
  - Power_Top contains the main LDO's that power most of the analog mixed signal circuits as well as the LDO for power the VLSI Core
  - Connect to 1.8V +/-5% system power supply

## 5.3 Power management

This section describes the Power state definition and Headset, push button detection in low power.

| Power State | Description |
|---|---|
| D0 | Power fully On, Individual ADC's and DAC's can be turned on and off (Highest Power) |
| D1 | Allowing for some reduction, ADC's and DAC's will be off, Analog pass thru, Analog references on. |
| D2 | Allows For the lowest possible power consuming state from which it can return to "fully on" state (D0) within 2ms. Analog reference off. |
| D3 (hot) | All internal clocks are stopped (only a Low power oscillator is on). This is the lowest power state from the device while power is applied. |
| D3(cold) | Power is removed and all register data lost (lowest power). |
| D3.5 | Sometimes Called D4 , is the transition state between D3hot and D3cold |

**Figure 5.3-1 Device Power State Definition**

| Function | S3/D3 | S0/D3 | S3/D3.5 | S0/D3.5 | S3/D4 | S0/D4 |
|---|---|---|---|---|---|---|
| Plug HS | OK | OK | OK | OK | OK | OK |
| Unplug HS | OK | OK | OK | OK | OK | OK |
| Push Button | OK | OK | OK | OK | OK | OK |

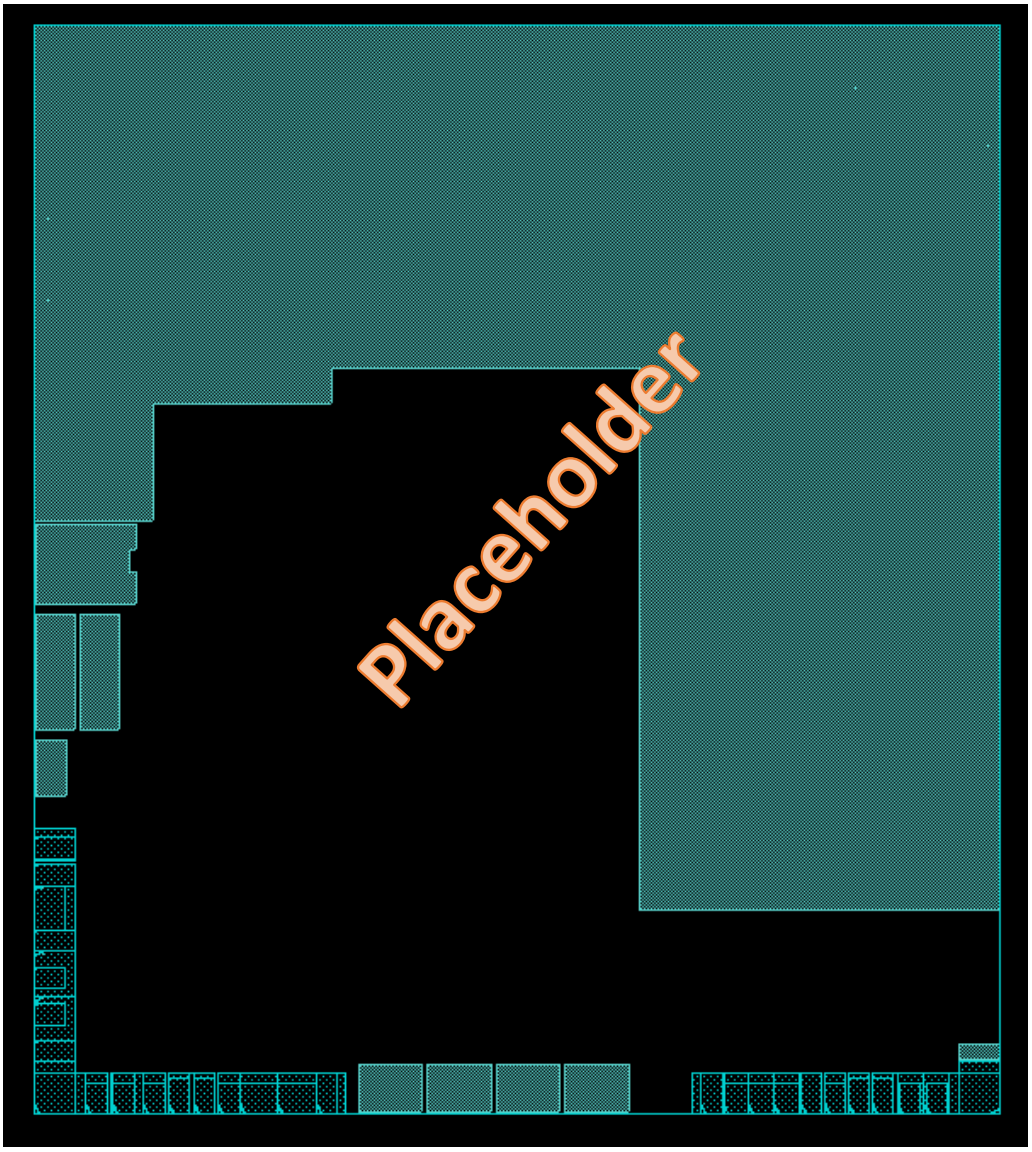
**Figure 5.3-2 Headset/Button detection in low power modes**

## 5.4 Device Power Consumption

| 1 mw HP Playback with ED (Nominal) | Digital Power EPS Estimate (mW) | Analog Power Simulated Estimate (mW) | Total Power (mW) |
|---|---|---|---|
| HD-A Stereo Playback | 13 | 15 | 28 |
| I2S Stereo Playback | 11 | 15 | 26 |

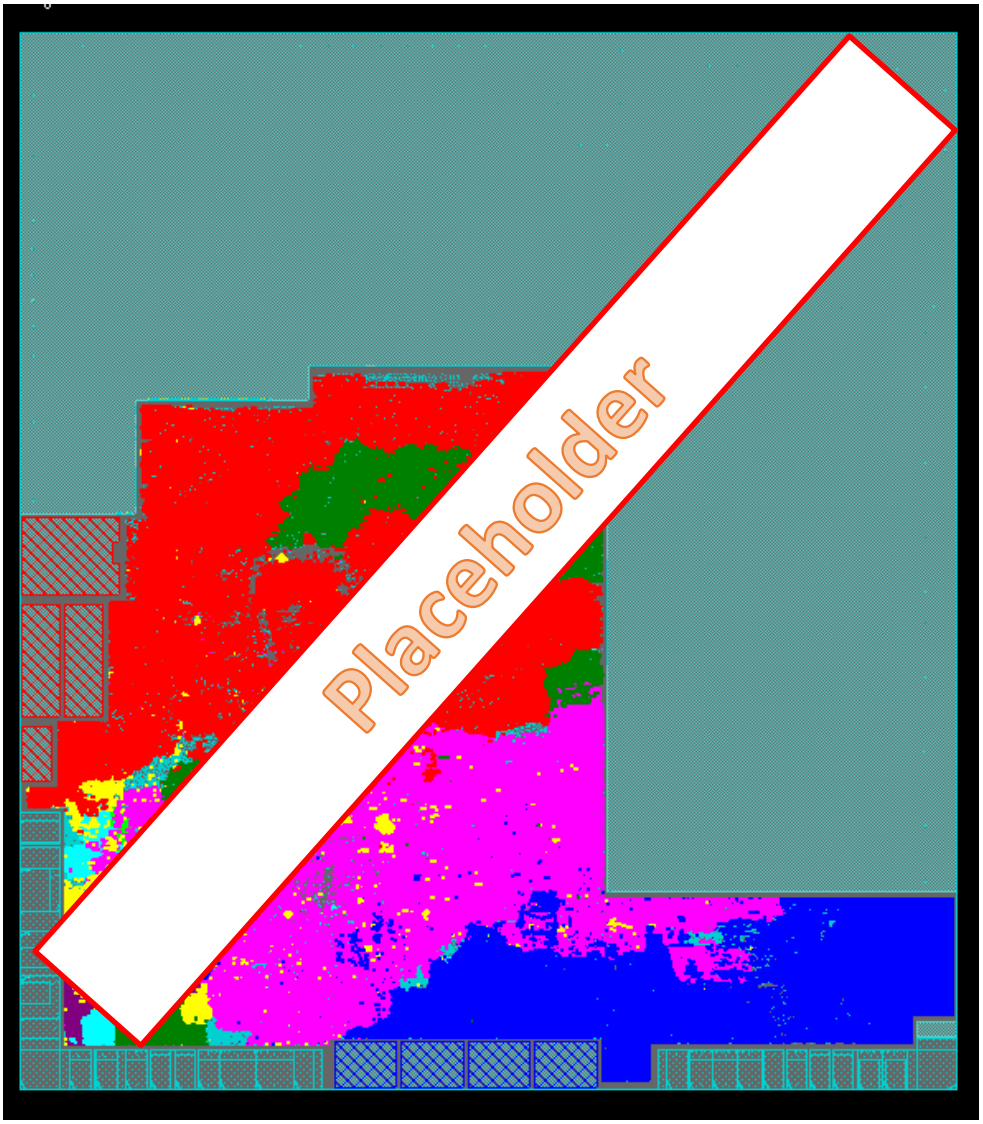
## 5.5 Oceanside Floor Plan

Oceanside floor plan diagram below.
Pre-Shrink Die Size: 3626 x 3630




### 5.5.1 Oceanside Standard Cell Utilization

Standard Cells Utilization = ????%.




## 5.6 Pin/Pad Descriptions

| Pin Label | PAD Type | Pin Number 60-Pin | I/O Type | Signal Name/Description |
|---|---|---|---|---|
| HDA_BUS_SDO | pd_hda2i | 1 | Digital In | -HD audio bus serial data input to the CODEC<br>- I2S_PCM RX data |
| HDA_BUS_RESETN | Pd_hda2i | 2 | Digital In | HD audio bus master hardware reset<br>I2S_PCM LRCK Frame Clock |
| VDDIO | | 3 | Power | 1.5V / 1.8V / 3.3V, +/- 5%. Sets the host interface signaling level. Connect to the same power supply as the host bus controller. |
| HSCL | Pd_i2c3 | 4 | Digital In | I2C Clock |

| Pin Label | PAD Type | Pin Number 60-Pin | I/O Type | Signal Name/Description |
|---|---|---|---|---|
| HSDA | Pd_i2c3 | 5 | Digital I/O | I2C Data |
| MSCL | Pd_i2c3 | 6 | Digital In | I2C Clock |
| MSDA | Pd_i2c3 | 7 | Digital I/O | I2C Data |
| MUSIC_REQ/SPDIF/GPIO0 | pd_gpioh_vseqdc00e0 | 8 | Digital I/O | Multi-function pin. **MUSIC_REQ** Pull high by external controller to enable D3 Live Mode**.** **SPDIF** SPDIF output. **GPIO0** GPIO |
| SPKR_MUTE/GPIO1 | pd_gpioh_vseqdc00e0 | 9 | Digital I/O | Multi-function pin. **SPKR_MUTE** When pulled low, holds Class-D on Mute. Instead, Conexant recommends using a bi-directional EAPD pin.<br>**GPIO1** GPIO |
| GPIO4 | pd_gpioh_vseqdc00e0 | 10 | Digital I/O | **GPIO4** GPIO |
| GPIO5 | pd_gpioh_vseqdc00e0 | 11 | Digital I/O | **GPIO5** GPIO |
| PORTC_DMIC_DATA1_GPIO3 | pd_gpioh_vseqdc00e0 | 12 | Digital I/O | Multi-function pin. **DMIC DATA1** input to the CODEC. The digital MIC data from 1 or 2 DMICS.<br>**GPIO3** GPIO |
| DVDD_IO | | 13 | Power | Sets the Digital I/O Level. Connect to 1.8V / 3.3V, +/- 5%<br>Signaling Level for the DMIC, SPDIF, EAPD, & GPIO |
| PORTC_DMIC_CLK1_GPIO2/ ANA_TST1 (ENG package) | pd_gpioh_vseqdc00e0 | 14 | Digital I/O | Multi-function pin. **DMIC CLK1** Output from the CODEC. The digital MIC clock.<br>**GPIO2** GPIO |
| EAPD | pd_gpioh_vseqdc00e0 | 15 | Digital I/O | Bi-directional pin**. Output** Controls external amplifiers power up/down per the HD Audio Spec.<br>**Input** Can be driven to mute Class D and headphones |
| I2S2_DIN | pd_hda2io | 16 | Digital I/O | Peripheral I2S2 data input |
| I2S2_DOUT | pd_hda2io | 17 | Digital I/O | Peripheral I2S2 data output |
| I2S2_LRCLK | pd_hda2io | 18 | Digital I/O | Peripheral I2S2 frame clock in/out |
| VDDO3 | | 19 | Power | Sets the Digital I/O Level. Connect to 1.8V / 3.3V, +/- 5%<br>Signaling Level for the I2S2 |
| I2S2_BCLK | pd_hda2io | 20 | Digital I/O | Peripheral I2S2 shift clock in/out |
| I2S2_MCLK | pd_hda2io | 18 | Digital I/O | Peripheral I2S2 master clock in/out |
| PORTG_LEFT_P | | 22 | Analog Out | PortG, Class D output, Left, (+) |
| PVDD5_LEFT | | 23 | Power | PortG, Class D Left, 5 Volt Power Supply Input |
| PORTG_LEFT_N | | 24 | Analog Out | PortG, Class D output, Left, (-) |
| PORTG_RIGHT_N | | 25 | Analog Out | PortG, Class D output, Right, (-) |

| Pin Label | PAD Type | Pin Number 60-Pin | I/O Type | Signal Name/Description |
|---|---|---|---|---|
| PVDD5_RIGHT | | 26 | Power | PortG, Class D Right, 5 V Power Supply Input |
| PORTG_RIGHT_P | | 27 | Analog Out | PortG, Class D output, Right, (+) |
| CP_VDD18 | | 28 | Power | Charge Pump, 1.8V Power Supply Input |
| CP_FLYP | | 29 | Analog I/O | Charge Pump, Fly Cap (+) pin |
| CP_FLYN | | 30 | Analog I/O | Charge Pump, Fly Cap (-) pin |
| CP_VNEG | | 31 | Analog Out | Charge Pump, Negative -1.7V Output |
| CP_VPOS | | 32 | Analog Out | Charge Pump, Positive +1.7V Output |
| PORTA_L | | 33 | Analog Out | Port A, Capless Headphone Out, Left |
| PORTA_R | | 34 | Analog Out | Port A, Capless Headphone Out, Right |
| HGNDA | | 35 | Analog I/O | Port D, Headset GND or Micbias<br>• Ground for CTIA-style headsets<br>• Micbias for OMTP-style headsets |
| SGNDA | | 36 | Analog I/O | Headphone reference for CTIA-style headsets |
| SGNDB | | 37 | Analog I/O | Headphone reference for OMTP-style headsets |
| HGNDB | | 38 | Analog I/O | Port D, Headset GND or Micbias<br>• Ground for OMTP-style headsets<br>• Micbias for CTIA-style headsets |
| VREF_DAC | | 39 | Analog I/O | Analog Ground reference (AGND) |
| VREFP | | 40 | Analog I/O | Analog hi-z reference decoupling |
| AVDD5 | | 41 | Power | Analog 5V Supply Input |
| PORTD_A | | 42 | Analog I/O | Port D, Apple (CTIA) Input or OMTP Ref Decouple |
| PORTD_B | | 43 | Analog I/O | Port D, Nokia (OMTP) Input or CTIA Ref Decouple |
| MICBIASC | | 44 | Analog Out | Port C MicBias Voltage Source |
| PORTB_L | | 45 | Analog In | Port B, Analog Line/Mic Input, Left |
| PORTB_R | | 46 | Analog In | Port B, Analog Line/Mic Input, Right |
| JSENSE | | 47 | Analog In | Jack sense input. Connect to the resistor network per the HD Audio Spec. |
| PCBEEP | | 48 | Analog In | PC speaker beep input. Needs to be AC-coupled with 0.1uF cap. Max input level is 1.8V. |
| LDO_AVDD | | 49 | Power | 1.6 to 1.8V LDO Output. Connect to AGND w decoupling capacitor. |

| Pin Label | PAD Type | Pin Number 60-Pin | I/O Type | Signal Name/Description |
|---|---|---|---|---|
| VDD18 | | 50 | Power | Analog Power Supply Input to the CODEC. Apply 1.8V +/- 5% |
| LDO_VDD_CORE | | 51 | Power | 1.2V LDO Output for VLSI. Connect to digital GND w a decoupling capacitor |
| I2S1_MCLK | pd_hda2io | 52 | Digital I/O | Peripheral I2S2 master clock in/out |
| I2S1_BCLK | pd_hda2io | 53 | Digital I/O | Peripheral I2S2 shift clock in/out |
| I2S1_LRCLK | pd_hda2io | 54 | Digital I/O | Peripheral I2S2 frame clock in/out |
| VDDO2 | | 55 | Power | Sets the Digital I/O Level. Connect to 1.8V / 3.3V, +/- 5%<br>Signaling Level for the I2S2 |
| I2S1_DOUT | pd_hda2io | 56 | Digital I/O | Peripheral I2S2 data output |
| I2S1_DIN | pd_hda2io | 57 | Digital I/O | Peripheral I2S2 data input |
| HDA_BUS_BCLK | pd_hda2io | 58 | Digital I/O | HD audio bus bit clock input to the CODEC |
| HDA_BUS_SYNC | pd_hda2io | 59 | Digital I/O | HD audio bus SYNC input to the CODEC |
| HDA_BUS_SDI | pd_hda2io | 60 | Digital I/O | HD audio bus serial data output from the CODEC |

## 5.7 PAD Record

| Oceanside | 60-Pin QFN | Die size = 3670um x 3140um (with scribe&seal) | | | |
|---|---|---|---|---|---|
| Pad # | **Pad Name** | **Pkg Pin** | **#Bond Wires** | **Flags** | **Comments** |
| 1 | PORTB_L | 45 | 1 | | |
| 2 | PORTB_R | 46 | 1 | | |
| 3 | JSENSE | 47 | 1 | | |
| 4 | PCBEEP | 48 | 1 | | |
| 5 | DVDD17 | 49 | 1 | | Three pads bond to package pin 49 |
| 6 | AVDD17 | 49 | 1 | | Three pads bond to package pin 49 |
| 7 | LDO_AVDD | 49 | 1 | | Three pads bond to package pin 49 |
| 8 | VDD18 | 50 | 1 | | |
| 9 | LDO_VDD_CORE | 51 | 1 | | Three pads bond to package pin 51 |
| 10 | VDD | 51 | 1 | | Three pads bond to package pin 51 |
| 11 | VDD | 51 | 1 | | Three pads bond to package pin 51 |
| 12 | VSS | PADDLE | 1 | | |
| 13 | I2S1_MCLK | 52 | 1 | | |
| 14 | I2S1_BCLK | 53 | 1 | | |

| | | | | | |
|---|---|---|---|---|---|
| 15 | I2S1_LRCLK | 54 | 1 | | |
| 16 | VDDO2 | 55 | 1 | | Two pads bond to package pin 55 |
| 17 | VDDO2 | 55 | 1 | | Two pads bond to package pin 55 |
| 18 | VSS | PADDLE | 1 | | |
| 19 | I2S1_DOUT | 56 | 1 | | |
| 20 | I2S1_DIN | 57 | 1 | | |
| 21 | HBCLK | 58 | 1 | | |
| 22 | VSS | PADDLE | 1 | | |
| 23 | HSYNC | 59 | 1 | | |
| 24 | HSDI | 60 | 1 | | |
| 25 | HSDO | 1 | 1 | | |
| 26 | VSS | PADDLE | 1 | | |
| 27 | HRSTN | 2 | 1 | | |
| 28 | VIO | 3 | 1 | | Two pads bond to package pin 3 |
| 29 | VIO | 3 | 1 | | Two pads bond to package pin 3 |
| 30 | HSCL | 4 | 1 | | |
| 31 | I2C_DEV_ID0 | See table | See table | BONDID | Will either down bond to PADDLE or be unbonded |
| 32 | HBUS_MODE | See table | See table | BONDID | Will either down bond to PADDLE or be unbonded |
| 33 | HSDA | 5 | 1 | | |
| 34 | VSS | PADDLE | 1 | | |
| 35 | VSS | PADDLE | 1 | | |
| 36 | MSCL | 6 | 1 | | |
| 37 | OPT0 | See table | See table | BONDID | Will either down bond to PADDLE or be unbonded |
| 38 | OPT1 | See table | See table | BONDID | Will either down bond to PADDLE or be unbonded |
| 39 | MSDA | 7 | 1 | | |
| 40 | MUSIC_REQ_GPIO0 | 8 | 1 | | |
| 41 | OPT2 | See table | See table | BONDID | Will either down bond to PADDLE or be unbonded |
| 42 | SPKRMUTEB_GPIO1 | 9 | 1 | | |
| 43 | GPIO4 | 10 | 1 | | |
| 44 | GPIO5 | 11 | 1 | | |
| 45 | VSS | PADDLE | 1 | | |
| 46 | DMIC_DAT1_GPIO3 | 12 | 1 | | |
| 47 | VDDO | 13 | 1 | | Two pads bond to package pin 13 |
| 48 | VDDO | 13 | 1 | | Two pads bond to package pin 13 |
| 49 | DMIC_CLK1_GPIO2 | 14 | 1 | | |
| 50 | EAPD | 15 | 1 | | |
| 51 | I2S2_DIN | 16 | 1 | | |
| 52 | I2S2_DOUT | 17 | 1 | | |
| 53 | I2S2_LRCLK | 18 | 1 | | |
| 54 | VDDO3 | 19 | 1 | | |

| | | | | | |
|---|---|---|---|---|---|
| 55 | I2S2_BCLK | 20 | 1 | | |
| 56 | VSS | PADDLE | 1 | | |
| 57 | I2S2_MCLK | 21 | 1 | | |
| 58 | PVSS_LEFT_P | PADDLE | 1 | | |
| 59 | PVSS_LEFT_P | PADDLE | 1 | | |
| 60 | PVSS_LEFT_P | PADDLE | 1 | | |
| 61 | PORTG_LEFT_P | 22 | 1 | | Three pads bond to package pin 22 |
| 62 | PORTG_LEFT_P | 22 | 1 | | Three pads bond to package pin 22 |
| 63 | PORTG_LEFT_P | 22 | 1 | | Three pads bond to package pin 22 |
| 64 | PVDD5_LEFT | 23 | 1 | | Three pads bond to package pin 23 |
| 65 | PVDD5_LEFT | 23 | 1 | | Three pads bond to package pin 23 |
| 66 | PVDD5_LEFT | 23 | 1 | | Three pads bond to package pin 23 |
| 67 | PORTG_LEFT_N | 24 | 1 | | Three pads bond to package pin 24 |
| 68 | PORTG_LEFT_N | 24 | 1 | | Three pads bond to package pin 24 |
| 69 | PORTG_LEFT_N | 24 | 1 | | Three pads bond to package pin 24 |
| 70 | PVSS_LEFT_N | PADDLE | 1 | | |
| 71 | PVSS_LEFT_N | PADDLE | 1 | | |
| 72 | PVSS_LEFT_N | PADDLE | 1 | | |
| 73 | PVSS_RIGHT_N | PADDLE | 1 | | |
| 74 | PVSS_RIGHT_N | PADDLE | 1 | | |
| 75 | PVSS_RIGHT_N | PADDLE | 1 | | |
| 76 | PORTG_RIGHT_N | 25 | 1 | | Three pads bond to package pin 25 |
| 77 | PORTG_RIGHT_N | 25 | 1 | | Three pads bond to package pin 25 |
| 78 | PORTG_RIGHT_N | 25 | 1 | | Three pads bond to package pin 25 |
| 79 | PVDD5_RIGHT | 26 | 1 | | Three pads bond to package pin 26 |
| 80 | PVDD5_RIGHT | 26 | 1 | | Three pads bond to package pin 26 |
| 81 | PVDD5_RIGHT | 26 | 1 | | Three pads bond to package pin 26 |
| 82 | PORTG_RIGHT_P | 27 | 1 | | Three pads bond to package pin 27 |
| 83 | PORTG_RIGHT_P | 27 | 1 | | Three pads bond to package pin 27 |
| 84 | PORTG_RIGHT_P | 27 | 1 | | Three pads bond to package pin 27 |
| 85 | PVSS_RIGHT_P | PADDLE | 1 | | |
| 86 | PVSS_RIGHT_P | PADDLE | 1 | | |
| 87 | PVSS_RIGHT_P | PADDLE | 1 | | |
| 88 | CP_AVDD18 | 28 | 1 | | Three pads bond to package pin 28 |
| 89 | CP_AVSS | PADDLE | 1 | | |
| 90 | CP_PVSS | PADDLE | 1 | | |
| 91 | CP_PVDD18 | 28 | 1 | | Three pads bond to package pin 28 |

| | | | | | |
|---|---|---|---|---|---|
| 92 | CP_PVDD18 | 28 | 1 | | Three pads bond to package pin 28 |
| 93 | CP_PVSS | PADDLE | 1 | | |
| 94 | CP_FLYP | 29 | 1 | | Two pads bond to package pin 29 |
| 95 | CP_FLYP | 29 | 1 | | Two pads bond to package pin 29 |
| 96 | CP_PVSS | PADDLE | 1 | | |
| 97 | CP_FLYN | 30 | 1 | | Two pads bond to package pin 30 |
| 98 | CP_PVSS | PADDLE | 1 | | |
| 99 | CP_FLYN | 30 | 1 | | Two pads bond to package pin 30 |
| 100 | CP_VNEG | 31 | 1 | | Three pads bond to package pin 31 |
| 101 | CP_VNEG | 31 | 1 | | Three pads bond to package pin 31 |
| 102 | VNEG | 31 | 1 | | Three pads bond to package pin 31 |
| 103 | CP_VPOS | 32 | 1 | | Three pads bond to package pin 32 |
| 104 | CP_VPOS | 32 | 1 | | Three pads bond to package pin 32 |
| 105 | VPOS | 32 | 1 | | Three pads bond to package pin 32 |
| 106 | PORTA_L | 33 | 1 | | |
| 107 | PORTA_R | 34 | 1 | | |
| 108 | SWGND | PADDLE | 2 | | Double-pad |
| 109 | HGND_A | 35 | 2 | | Double-pad |
| 110 | SGND_A | 36 | 1 | | |
| 111 | SGND_B | 37 | 1 | | |
| 112 | SWGND | PADDLE | 2 | | Double-pad |
| 113 | HGND_B | 38 | 2 | | Double-pad |
| 114 | VREF_DAC | 39 | 1 | | |
| 115 | VREFP | 40 | 1 | | |
| 116 | AVDD5 | 41 | 1 | | |
| 117 | DVSS17 | PADDLE | 1 | | |
| 118 | AVSS17 | PADDLE | 1 | | |
| 119 | VSUB | PADDLE | 1 | | |
| 120 | PORTD_A | 42 | 1 | | |
| 121 | PORTD_B | 43 | 1 | | |
| 122 | VSS18 | PADDLE | 1 | | |
| 123 | MICBIAS | 44 | 1 | | |

# 6 PLLs, Clocks and Resets

## 6.1 PLLs

Oceanside has a low power fractional N Phase Lock Loop (FracNPLL). The PLL accepts an input reference clock from an external source such as a crystal oscillator, and outputs clocks to be used in a digital domain and or as the sampling clock source for codec data converters. There are three modes of operation of the PLL.

- HD Audio Mode (Input **clk_ref** = 6 MHz – PLL Operates in Fractional Mode)
- I2S Input Mode (Input **clk_ref** comes from BCLK – PLL Operates in both Integer and fractional Mode)
- Low Power Mode (PLL is overridden with PLL output passing the RC Oscillator)

When the PLL Operates in the Fractional Mode, the PLL Output Frequency has a PPM error from the actual audio sampling rate. This creates a situation of FIFO Buffer overflow/underflow. There needs to be a correction loop incorporating an UP/DWN Counter that measures the Input frequency and the output frequency and generates a code that modifies the programmed **frac_lv[19:0] values.**

### 6.1.1 PLL Macro IO Table

| Name | Type | Default | Description |
|---|---|---|---|
| clk_ref | CMOS Input | 3 ~ 36 MHz | Input Reference Clock Input |
| clk_out | CMOS Output | 147.456/<br>135.4752 MHz | PLL Output Clock |
| ctrl_div_test[1:0] | CMOS Input | 2'b00 | Clock test point control bits |
| ck_testa | CMOS Output | Up to 150 MHz | Test point # 1, mux to view reference, feedback and VCO clocks |
| ck_testb | CMOS Output | Up to 150 MHz | Test point #2, mux to view reference, feedback and VCO clocks |
| ctrl_r[2:0] | CMOS Input | 3'b000 | Input Divider value control bits |
| ctrl_byp_pll_ckdet | CMOS Input | 1'b0 | Bypass Control bit for the Clock Detector Circuit in the PLL. |
| ctrl_byp_rc3 | CMOS Input | 1'b0 | Bypass Control bit, shorts the RC3 resistor. |
| dis_ynkbst | CMOS input | 1'b0 | 1 = Disables yank boost block logic |
| ctrl_rc3_mm_en | CMOS Input | 1'b0 | Control bit to overwrite the ctrl_rc3[1:0] controls with ctrl_bw[1:0] |
| ctrl_pd_up_dly[1:0] | CMOS Input | 2'b00 | Phase Detector, 'Up', reset delay, control bits |
| ctrl_pd_dn_dly[1:0] | CMOS Input | 2'b00 | Phase Detector, 'Dn', reset delay, control bits |
| ctrl_cp_iref[1:0] | CMOS Input | 2'b00 | Charge Pump Reference Current Adjustment |
| ctrl_cp_bias_mode | CMOS Input | 1'b0 | Charge Pump Reference, fixed or tracking |
| ctrl_cp_scale[1:0] | CMOS Input | 2'b00 | Charge Pump Reference Current Scale |
| ctrl_bw[1:0] | CMOS Input | 2'b00 | Bandwidth control bits |
| ctrl_rc3[1:0] | CMOS Input | 2'b00 | Third parasitic pole control bits |
| enable_pll | CMOS Input | 1'b0 | PLL enable |
| resetn_dsm_lv | CMOS Input | 1'b0 | Reset the DSM |
| dither_en_lv | CMOS Input | 1'b0 | 0 = DSM dither off<br>1 = DSM dither on |
| intmode_lv | CMOS Input | 1'b0 | 0 = normal fractional operation<br>1 = integer mode enabled |
| integ_lv[6:0] | CMOS Input | 7'b000_0000 | Feedback Divider, Integer part, Divide ratio = decimal value + 1 |
| frac_lv[19:0] | CMOS Input | 20'h0_0_0_0_0 | Feedback Divider, Fractional part |
| scan_clk_lv | CMOS Input | 1'b0 | Scan clock input |
| scan_en_lv | CMOS Input | 1'b0 | Scan enable |
| scan_in_lv | CMOS Input | 1'b0 | Scan Input |
| scanmode_lv | CMOS Input | 1'b0 | When High Puts the DSM in Scan Mode of Operation. Need the Scan Enable for it run the scan |
| scan_out_lv | CMOS Output | 1'b0 | Indicates the status of the scan |
| DVDD | Digital Supply | 1.0V | 1.0V LDO (output from 1.0V LDO / input to PLL) |

| | | | |
|---|---|---|---|
| DVSS | Digital Ground | 0 | |
| avddh | Analog Supply | 1.7V | 1.7V LDO (output from 1.7V LDO / input to PLL) |
| avss | Analog Ground | 0 | |
| vsub | Substrate | 0 | |

## 6.1.2 PLL Block Diagram

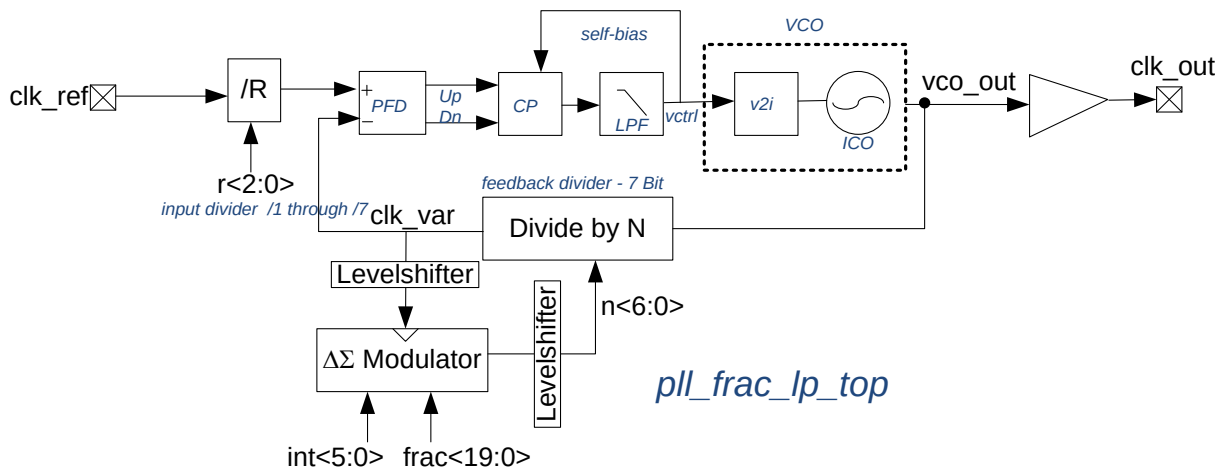



**Figure 6-1: PLL Block Diagram**

## 6.1.3 PLL Input Output Operating Frequency

The PLL shall operate with the Following Input Output Frequencies.

| | Input Ref Freq (MHz) | Pre-Div (done in digital) | Loop Div | Phase Comp Freq (MHz) | VCO Frequency (@clk_out pin )(MHz) | MODE OF OPERATION |
|---|---|---|---|---|---|---|
| HD Audio Mode (PLL Works as a Frac PLL) | 24 | 4 | 24.576 | 6 | 147.456 | FRAC_MODE |
| | 12 | 2 | 24.576 | 6 | 147.456 | FRAC_MODE |
| | 6 | 1 | 24.576 | 6 | 147.456 | FRAC_MODE |
| | | | | | | |
| | 24 | 4 | 22.5792 | 6 | 135.4752 | FRAC_MODE |
| | 12 | 2 | 22.5792 | 6 | 135.4752 | FRAC_MODE |
| | 6 | 1 | 22.5792 | 6 | 135.4752 | FRAC_MODE |
| I2S Mode (PLL Works as a Frac PLL) | 19.2 | 3 | 23.04 | 6.4 | 147.456 | FRAC_MODE |
| | 24 | 4 | 24.576 | 6 | 147.456 | FRAC_MODE |
| | 25 | 4 | 23.59296 | 6.25 | 147.456 | FRAC_MODE |
| | 26 | 4 | 22.68554 | 6.5 | 147.456 | FRAC_MODE |
| | 27 | 5 | 27.30667 | 5.4 | 135.4752 | FRAC_MODE |
| | 36 | 6 | 24.576 | 6 | 147.456 | FRAC_MODE |
| | 48 | 8 | 24.576 | 6 | 147.456 | FRAC_MODE |
| | | | | | | |
| I2S Mode (PLL Works in | 1.536 | 1 | 96 | 1.536 | 147.456 | INT_MODE |
| | 2.304 | 1 | 64 | 2.304 | 147.456 | INT_MODE |
| | 3.072 | 1 | 48 | 3.072 | 147.456 | INT_MODE |

| | | | | | | |
|---|---|---|---|---|---|---|
| **Integer Mode)** | 4.608 | 1 | 32 | 4.608 | 147.456 | INT_MODE |
| | 6.144 | 1 | 24 | 6.144 | 147.456 | INT_MODE |
| | 6.912 | 3 | 64 | 2.304 | 147.456 | INT_MODE |
| | 9.216 | 2 | 32 | 4.608 | 147.456 | INT_MODE |
| | 12.288 | 2 | 24 | 6.144 | 147.456 | INT_MODE |
| | 13.824 | 3 | 32 | 4.608 | 147.456 | INT_MODE |
| | 16.384 | 3 | 27 | 5.4613 | 147.456 | INT_MODE |
| | 18.432 | 4 | 32 | 4.608 | 147.456 | INT_MODE |
| | | | | | | |
| | 24.576 | 4 | 24 | 6.144 | 147.456 | INT_MODE |
| | 27.648 | 6 | 32 | 4.608 | 147.456 | INT_MODE |
| | 36.864 | 6 | 24 | 6.144 | 147.456 | INT_MODE |
| | 49.152 | 8 | 24 | 6.144 | 147.456 | INT_MODE |
| | | | | | | |
| | | | | | | |
| | 1.4112 | 1 | 96 | 1.4112 | 135.4752 | INT_MODE |
| | 2.1168 | 1 | 64 | 2.1168 | 135.4752 | INT_MODE |
| | 2.8224 | 1 | 48 | 2.8224 | 135.4752 | INT_MODE |
| | 4.2336 | 1 | 32 | 4.2336 | 135.4752 | INT_MODE |
| | 5.6448 | 1 | 24 | 5.6448 | 135.4752 | INT_MODE |
| | 6.3504 | 3 | 64 | 2.1168 | 135.4752 | INT_MODE |
| | 8.4672 | 2 | 32 | 4.2336 | 135.4752 | INT_MODE |
| | 11.2896 | 2 | 24 | 5.6448 | 135.4752 | INT_MODE |
| | 22.5792 | 4 | 24 | 5.6448 | 135.4752 | INT_MODE |
| | 28.224 | 5 | 24 | 5.6448 | 135.4752 | INT_MODE |
| | 45.1584 | 8 | 24 | 5.6448 | 135.4752 | INT_MODE |

*Table 6.1-1 PLL Operating Frequencies*

*NOTES:*
*The VCO does operate at the same 147.456 MHz/ 135.4752 MHz. The loop divider ratio is increased for the VCO to always operate at that frequency*

### 6.1.4 PLL Adjustable Variables and Loop Stability Coefficients Table

*The columns in Blue (Ip, R1 and R3) are the variables that need to be programmed to maintain constant loop stability dynamics, while the PLL closed loop BW is varied in proportion to the $F_{ref}$ (reference frequency). The columns for R (input divider), P (output divider) and N (the integer and fractional values) are shown for each set of Fref and Fout settings… The value N is the fractional feedback divider value which results from the ΔΣ Modulator. The value 'fo' is the PLL Loop BW and is a constant ratio of 24 times lower than the Fref reference frequency. Kvco is the VCO Gain coefficient and is shown at a nominal value of 500MHz/V. Likewise the values of the loop filter C2, C1, C3 are fixed values shown with nominal design targets.*
*The new Loop calculations table should come here*

### 6.1.5 Fractional N – Digital Delta Sigma Modulator

*The 3rd order delta-sigma modulator (ΔΣ Modulator) takes a 20-bit value (frac[19:0]) value as its input and modulates this to a 3-bit 2's complement output. This 3-bit number varies within the range -4 to +3 (decimal), and has an average (DC) value of (where $N_{FRAC}$ is the decimal value of **frac_lv** [19:0])*

$$7 \cdot \left( \frac{N_{FRAC}}{2^{20} - 4} \right) - 4$$

*This is added to the integer code n[8:0] to give the feedback division ratio (where $N_{INTEGER}$ is the decimal value of n[8:0])*

$$N_{INTEGER} + \left[ 7 \cdot \left( \frac{N_{FRAC}}{2^{20} - 4} \right) - 4 \right]$$

***Note:*** *Although the ΔΣ Modulator can output a range -4 to +3, it has a limited stable range for the DC output. The DC output must lie between -2.19 and +1.19. This sets the limit on* ***frac_lv*** *to the range 271130 – 777441.*

*The ΔΣ Modulator can be disabled by setting* ***intmode_lv*** *high. This sets the ΔΣ Modulator to output 0, so the feedback division ratio is equal to the value set by n [8:0]. Also, setting* ***intmode_lv*** *high disables the clock to the DSM, reducing power consumption.*

*The following Table shows the programming required for the ΔΣ Modulator for all the fractional values of N shown in the previous table*

| **N** | N_int (dec) | **N_int (hex)** | N_int (bin) | N_frac | N_frac LL | **N_frac_dec** | N_frac_UL | N_frac hex | N_frac_bin_20b |
|---|---|---|---|---|---|---|---|---|---|
| 22.579167 | 22 | 16 | 000010110 | 0.579167 | 271131 | 685941 | 777441 | A7775 | 1010 0111 0111 0111 0101 |
| 24.576000 | 24 | 18 | 000011000 | 0.576000 | 271131 | 685466 | 777441 | A759A | 1010 0111 0101 1001 1010 |
| 19.353571 | 19 | 13 | 000010011 | 0.353571 | 271131 | 652148 | 777441 | 9F374 | 1001 1111 0011 0111 0100 |
| 21.065143 | 21 | 15 | 000010101 | 0.065143 | 271131 | 608942 | 777441 | 94AAE | 1001 0100 1010 1010 1110 |
| 16.934375 | 16 | 10 | 000010000 | 0.934375 | 271131 | 739150 | 777441 | B474E | 1011 0100 0111 0100 1110 |
| 18.432000 | 18 | 12 | 000010010 | 0.432000 | 271131 | 663896 | 777441 | A2158 | 1010 0010 0001 0101 1000 |
| 15.052778 | 15 | F | 000001111 | 0.052778 | 271131 | 607090 | 777441 | 94372 | 1001 0100 0011 0111 0010 |
| 16.384000 | 16 | 10 | 000010000 | 0.384000 | 271131 | 656706 | 777441 | A0542 | 1010 0000 0101 0100 0010 |
| 13.547500 | 13 | D | 000001101 | 0.547500 | 271131 | 681197 | 777441 | A64ED | 1010 0110 0100 1110 1101 |
| 14.745600 | 14 | E | 000001110 | 0.745600 | 271131 | 710872 | 777441 | AD8D8 | 1010 1101 1000 1101 1000 |
| 12.315909 | 12 | C | 000001100 | 0.315909 | 271131 | 646506 | 777441 | 9DD6A | 1001 1101 1101 0110 1010 |
| 13.405091 | 13 | D | 000001101 | 0.405091 | 271131 | 659865 | 777441 | A1199 | 1010 0001 0001 1001 1001 |
| 22.579167 | 22 | 16 | 000010110 | 0.579167 | 271131 | 685941 | 777441 | A7775 | 1010 0111 0111 0111 0101 |
| 24.576000 | 24 | 18 | 000011000 | 0.576000 | 271131 | 685466 | 777441 | A759A | 1010 0111 0101 1001 1010 |
| 20.842308 | 20 | 14 | 000010100 | 0.842308 | 271131 | 725358 | 777441 | B116E | 1011 0001 0001 0110 1110 |
| 22.685538 | 22 | 16 | 000010110 | 0.685538 | 271131 | 701875 | 777441 | AB5B3 | 1010 1011 0101 1011 0011 |
| 18.815972 | 18 | 12 | 000010010 | 0.815972 | 271131 | 721413 | 777441 | B0205 | 1011 0000 0010 0000 0101 |
| 20.480000 | 20 | 14 | 000010100 | 0.480000 | 271131 | 671086 | 777441 | A3D6E | 1010 0011 1101 0110 1110 |
| 21.167969 | 21 | 15 | 000010101 | 0.167969 | 271131 | 624345 | 777441 | 986D9 | 1001 1000 0110 1101 1001 |
| 23.040000 | 23 | 17 | 000010111 | 0.040000 | 271131 | 605176 | 777441 | 93BF8 | 1001 0011 1011 1111 1000 |
| 20.651677 | 20 | 14 | 000010100 | 0.651677 | 271131 | 696803 | 777441 | AA1E3 | 1010 1010 0001 1110 0011 |
| 22.478049 | 22 | 16 | 000010110 | 0.478049 | 271131 | 670794 | 777441 | A3C4A | 1010 0011 1100 0100 1010 |
| 20.526515 | 20 | 14 | 000010100 | 0.526515 | 271131 | 678054 | 777441 | A58A6 | 1010 0101 1000 1010 0110 |
| 22.341818 | 22 | 16 | 000010110 | 0.341818 | 271131 | 650387 | 777441 | 9EC93 | 1001 1110 1100 1001 0011 |
| 22.579167 | 22 | 16 | 000010110 | 0.579167 | 271131 | 685941 | 777441 | A7775 | 1010 0111 0111 0111 0101 |
| 24.576000 | 24 | 18 | 000011000 | 0.576000 | 271131 | 685466 | 777441 | A759A | 1010 0111 0101 1001 1010 |
| 20.842308 | 20 | 14 | 000010100 | 0.842308 | 271131 | 725358 | 777441 | B116E | 1011 0001 0001 0110 1110 |
| 22.685538 | 22 | 16 | 000010110 | 0.685538 | 271131 | 701875 | 777441 | AB5B3 | 1010 1011 0101 1011 0011 |
| 20.070370 | 20 | 14 | 000010100 | 0.070370 | 271131 | 609725 | 777441 | 94DBD | 1001 0100 1101 1011 1101 |
| 21.845333 | 21 | 15 | 000010101 | 0.845333 | 271131 | 725812 | 777441 | B1334 | 1011 0001 0011 0011 0100 |

## 6.1.6 Electrical Specification

**Key Modules**

**Fractional N PLL, Key Performance Specifications, and Operation Conditions**

| Parameter Description | Specification | Design Result (Schematic) | Design Result (Extracted) | Comments/Conditions |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| Junction Temperature [ºC] | Min: -40 C<br>Typ: 65 C<br>Max: 125 C | Simulation Condition | Simulation Condition | |
| Analog Power Supply [V] | Min: 1.6V<br>Typ: 1.8V<br>Max: 2.0V | Simulation Condition | Simulation Condition | |
| Digital Power Supply [V] | Min: 0.9V<br>Typ: 1.0V<br>Max: 1.1V | Simulation Condition | Simulation Condition | DSM runs at this voltage |
| Reference Clock Frequency [MHz] | 6MHz /27MHz | Simulation Condition | Simulation Condition | |
| Loop Phase Margin | Min: 45 degrees | Min: | Min: | |
| VCO Frequency Range | Min: 135.475 MHz<br>Max: 147.456 MHz | Min:<br>Max: | Min:<br>Max: | Re-evaluate with new strategy for reference freq. See if we can lower VCO Freq..for power |
| Output Frequency | VCO Freq divided by 2,4,6,8,12,16,24, or 48 | Min:<br>Max: | Min<br>Max: | |
| Lock Time | Max: 200 uS | Max: | Max: | |
| Closed Loop Bandwidth | Min: 250KHz<br>Max: 500KHz | Min:<br>Max: | Min:<br>Max: | |
| PLL deterministic jitter in locked condition (peak to peak) due to ripple on control voltage and mismatches in charge pump. | Max: 100pS | Max: | Max: | TBD… 200 ps Total for deterministic plus rms Jitter |
| Fractional mode RMS jitter @12.288MHZ (integrated from 100Hz – 2MHz) | Max: 100pS | Max: | Max: | TBD… 200 ps Total for deterministic plus rms Jitter |
| Total RMS Jitter integrated from 3.072MHz +/- 20Hz→20Khz | TBD | | | SIMULATE AND compare with IBIZA PLL |
| PLL output duty cycle | Min: 49.99%<br>Max: 50.01% | Min:<br>Max: | Min:<br>Max: | |
| PLL Current Consumption [mA] | Max: 500uA (TBD) | Max: | Max | |
| Power Down current | Max :100nA | Max: | Max: | |
| Area | TBD | | | **Smaller than IBIZA PLL** |

## VCO Key Performance Specifications and Operating Conditions

| Parameter Description | Specification | Design Result (Schematic) | Design Result (Extracted) | Comments/Conditions |
|---|---|---|---|---|
| Junction Temperature [ºC] | Min: -40 C<br>Typ: 65 C<br>Max: 125 C | Simulation Condition | Simulation Condition | |
| AVDD18 Power Supply [V] | Min: 1.6V<br>Typ: 1.8V<br>Max: 2.0V | Simulation Condition | Simulation Condition | |
| VCO Gain ($K_{VCO}$) @ 147.456MHz | Min:<br>Typ: 500MHz /V<br>Max: | Min:<br>Typ:<br>Max: | Min:<br>Typ:<br>Max: | |
| VCO Duty Cycle @ 147.456MHz | Min: 49.99%<br>Max: 50.01% | Min:<br>Typ:<br>Max: | Min:<br>Typ:<br>Max: | |

| Parameter Description | Specification | Design Result (Schematic) | Design Result (Extracted) | Comments/Conditions |
|---|---|---|---|---|
| VCO Phase Noise For 147.456MHz<br>@10KHz offset<br>@ 100KHz offset<br>@ 1MHz offset | <br><br>Typ: -50dBc/Hz<br>Typ: -80dBc/Hz<br>Typ: -100dBc/Hz | <br><br>Typ:<br>Typ:<br>Typ: | <br><br>Typ:<br>Typ:<br>Typ: | Bottom Line is Total Jitter and power dissipation |

## 6.2 PLL Tracking Loop

Oceanside uses a "Fractional-N" PLL to generate internal clocks from an incoming reference clock. For proper operation, the internal clocks must be an exact multiple of the incoming sample rate.
More specifically, for incoming sample rates of 48KHz, 96KHz or 192KHz, the PLL VCO must generate 147.456MHz, which is equal to 48KHz * 3072. For an incoming sample rate of 44.1KHz, the VCO must generate 44.1KHz * 3072 = 135.4752MHz.
However, depending on the frequency of the incoming reference clock, as compared with the incoming sample rate, the Fractional-N PLL may not be able to generate the precise internal clock frequency required. In many cases, the PLL will only be able to come within a few PPM of the required frequency. An example of this situation would be an incoming reference clock frequency of 24MHz, with an incoming sample rate of 48KHz. The ratio of the required VCO frequency (147.456MHz) to the reference frequency (24MHz) is a fractional quantity (6.144) which cannot be exactly realized by the Fractional-N PLL. Instead, the closest programmable ratio supported by the PLL is approximately 6.1439991722, which introduces an error of approximately 0.135ppm.
To compensate for this limitation, a tracking loop is placed outside the PLL. The tracking loop monitors the incoming sample rate as well as the internally generated clock rate. It then makes repeated, small adjustments to the programming of the Fractional-N PLL, over time, to coordinate the internal clocks with the incoming sample rate.

### 6.2.1 General Theory of Operation

The PLL Tracking Loop compares the PLL generated clock rate with the incoming sample rate, and adjusts the Fractional-N PLL programming to keep the two coordinated. Over the long term, the internal logic of the device (using the PLL generated clocks) must process exactly the same number of samples as is transported by the host side interface. Over the short term, slight discrepancies between the incoming sample rate and the internal clocking will be absorbed by the host interface FIFOs. It is important, though, that these slight discrepancies, and PLL adjustments, do not introduce audible artifacts into the DAC/ADC performance.
For purposes of comparing clock rates, the incoming SYNC signal is used to represent the incoming sample rate.
To compare the incoming SYNC signal with the PLL generated clock, the SYNC signal is first synchronized into the PLL clock domain. The duration of the SYNC period (from rising edge to rising edge) is then measured in the PLL clock domain. Differences from the ideal measurement are accumulated over time, and it is this accumulated value which determines the adjustment to be applied to the PLL programming.

#### 6.2.1.1 HDA Host Interface Mode

When the host interface is operating in HDA mode, the incoming reference clock is the serial bit clock (BCLK). The HDA interface supports just 3-bit clock rates (6MHz, 12MHz and 24MHz).
It should be noted that the HDA SYNC signal is always 48KHz, for all supported sample rates (48Ksps, 96Ksps, 192Ksps and 44.1Ksps). It should also be noted that Oceanside can support 44.1Ksps only if all incoming data streams are using 44.1Ksps. (Mixtures of 44.1Ksps streams with streams of other sample rates are not supported.)
Because the SYNC signal is always at 48KHz, the ratio of the VCO frequency to the SYNC frequency can take on only 2 values:

*(1) 147.456MHz / 48KHz = 3072 (for sample rates of 48, 96 or 192Ksps)*
*(2) 135.4752MHz / 48KHz = 2822.4 (for a sample rate of 44.1Ksps)*

*It can be noted that in the case of a 44.1Ksps sample rate, the ratio between the VCO clock and the incoming SYNC rate is a fractional quantity. The TBD test register is used to select 44.1Ksps support within Oceanside. As such, it is also used to select between the above ratios.*
*It should be noted that the incoming HDA BCLK rate (6, 12 or 24MHz) is automatically detected by logic outside the tracking loop, and that the PLL is automatically configured to produce the required 147.456MHz or 135.4752MHz frequency from the incoming BCLK.*

#### 6.2.1.2 I2S Host Interface Mode

When the host interface is operating in I2S mode, the incoming reference clock can be either MCLK or BCLK. Unlike the HDA BCLK, I2S mode MCLK and BCLK are not restricted to specific frequencies.
Also unlike HDA mode, the I2S SYNC input frequency is always at the sample rate, rather than a fixed 48KHz.
Like HDA mode, however, the PLL VCO frequencies can still take on only the same two values (147.456MHz or 135.4752MHz).
Because of the above, the ratio of the VCO frequency to the SYNC frequency can take on one of the following 4 values:

*(1) 147.456MHz / 48KHz = 3072*
*(2) 135.4752MHz / 44.1KHz = 3072*
*(3) 147.456MHz / 96KHz = 1536*
*(4) 147.456MHz / 192KHz = 768*

Since the MCLK frequency is not restricted to any specific values, the user is required to program the Fractional-N PLL directly. It is necessary that the user selects the setting which most closely approximates the required frequency ratio.
In the same way, as with HDA, the TBD test bit (also accessible through the I2C host interface) is used to select between the 48KHz and 44.1KHz base rates. Additionally, however, it is necessary to differentiate between 48, 96 and 192Ksps. To do this, the PLL tracking loop looks at the “Sample Base Rate Multiple” value found in the “Converter Format” control for DAC1 (node 10).

### 6.2.2 Detailed Operation

#### 6.2.2.1 Tracking Loop Functional Clock

The Tracking Loop function operates in the PLL clock domain. Since the PLL VCO operates at a very high speed, it is first divided to provide a convenient clock for the Tracking Loop function. Specifically, the Tracking Loop function receives a clock which is the PLL VCO output divided by 32. Since the PLL VCO can take on two frequencies (147.456MHz and 135.4752MHz), the tracking loop clock can take on either of two frequencies (4.608MHz and 4.2336MHz).

**Tracking Loop Units of Measurement**

While the PLL VCO output can take on only two frequencies, the sample rate can take on several. The possible relationships between the sample rate and tracking loop clock are summarized in the following *table:*

| Interface Mode | Sample Rate | SYNC Freq | VCO Freq | T. L. Clock Divisor | T. L. Clock Freq | T. L. Clock cycles per SYNC period | T.L Clock cycles per Sample Period |
|---|---|---|---|---|---|---|---|
| HD Audio | 48 | 48KHz | 147.456MHz | 32 | 4.608MHz | 96 | 96 |
| | 96 | 48KHz | 147.456MHz | 32 | 4.608MHz | 96 | 48 |
| | 192 | 48KHz | 147.456MHz | 32 | 4.608MHz | 96 | 24 |
| | 44.1 | 48KHz | 135.4752MHz | 32 | 4.2336MHz | 88.2 | 96 |
| I2S | 48 | 48KHz | 147.456MHz | 32 | 4.608MHz | 96 | 96 |
| | 44.1 | 44.1KHz | 135.4752MHz | 32 | 4.2336MHz | 96 | 96 |
| | 96 | 96KHz | 147.456MHz | 32 | 4.608MHz | 48 | 48 |
| | 192 | 192KHz | 147.456MHz | 32 | 4.608MHz | 24 | 24 |

#### 6.2.2.2 SYNC Signal Re-Synchronization and Error Measurement

As indicated above, the duration of the incoming SYNC signal is measured in the PLL clock domain. However, the PLL clock and the SYNC signal are typically asynchronous to each other. Therefore, the SYNC signal is first synchronized into the PLL clock domain. The synchronization circuit is designed in such a way that it also detects the rising edges of the incoming SYNC signal, to be used to measure the SYNC signal period.

The time between each rising edge of the SYNC signal (as detected by the above circuit) is measured, in units of the Tracking Loop clock. For most modes of operation, there should be an integer number of Tracking Loop clock cycles per each SYNC period (24, 48 or 96). In one operating mode (44.1Ksps using the HDA interface mode), however, there are, on average, 88.2 Tracking Loop cycles per SYNC period. Since the PLL may not precisely generate the required clock frequency, the measured number of Tracking Loop clock cycles per SYNC period will vary from the “ideal” target value. This error (measured SYNC period – ideal SYNC period) is accumulated over time, and the accumulated error is used to adjust the PLL programming.

##### 6.2.2.2.1 Numeric Representation

The counter used to measure the number of Tracking Loop clock cycles per SYNC period uses a 7-bit unsigned representation (representing values from 0 to 127).

The error accumulator uses a 9-bit signed representation, with saturation to avoid rollover (representing values from -256 to +255).

##### 6.2.2.2.2 Special Case – 44.1Ksps in HDA mode

For most modes of operation, the ideal SYNC period consists of an integer number of Tracking Loop clocks (24, 48 or 96). The error is calculated by simply subtracting the target number of clock cycles from the counted number of clock cycles.
In the case of HDA mode with a 44.1Ksps sample rate, however, the "ideal" SYNC period consists of (on average) 88.2 Tracking Loop clocks. When computing the error in this mode, the 'ideal' value (which is subtracted from the counted number of clock cycles) is varied between 88 and 89, in such a way that the average is 88.2. This is accomplished by simply using a repeating sequence, in which the value 88 is used for 4 consecutive calculations, and the value 89 is used for 1 calculation.

#### 6.2.2.3 PLL Adjustment

The Fractional-N PLL accepts a 20-bit fractional quantity, which, when combined with a 6-bit integer quantity, establishes the feedback divider ratio used by the PLL. Both the fractional and integer quantities are established through register settings.
The output of the error accumulator (above) is used to select an adjustment value, which is either added to or subtracted from the fractional portion of the PLL feedback divider ratio.
Given the error calculation and accumulation (described above), if the PLL generated clock frequency is greater than the target frequency, the accumulated error will tend to be positive. On the other hand, if the PLL generated clock frequency is less than the target, the accumulated error will be negative.
When the PLL generated clock frequency is fast (and the accumulated error is positive) subtracting an adjustment value to the fractional portion of the PLL feedback divider ratio will cause the PLL output to be slowed. Conversely, when the PLL generated clock frequency is slow, adding an adjustment value will cause the PLL output to be sped up.
For flexibility, the Tracking Loop provides two programmable adjustment values (ADJ0 and ADJ1), which can be added or subtracted from the fractional portion of the PLL feedback divider ratio, as appropriate. It also provides two programmable threshold values (TH0 and TH1), which determine when the adjustment values are to be applied. Note that for proper operation, the value of TH0 must always be less than or equal to the value of TH1.
Application of an adjustment value is determined by the accumulated error, and how it compares to the programmed threshold values. There are also two modes of operation, which further determine the selection of the adjustment value.

##### 6.2.2.3.1 Adjustment with Dead-Band

The use of a dead-band in the PLL adjustment is established through register programming (below). When the dead-band is enabled, accumulated error values between –Threshold0 and +Threshold0 will result in no adjustment to the PLL fractional programming.
Adjustment is therefore made as described in the following table:

| Accumulated Error vs. Thresholds | Applied Adjustment |
|---|---|
| Accum Error < -TH1 <= -TH0 <= +TH0 <= +TH1 | +ADJ1 |
| -TH1 <= Accum Error < -TH0 <= +TH0 <= +TH1 | +ADJ0 |
| -TH1 <= -TH0 <= Accum Error <= +TH0 <= +TH1 | No Adjustment |
| -TH1 <= -TH0 <= +TH0 < Accum Error <= +TH1 | -ADJ0 |

##### 6.2.2.3.2 Adjustment Without Dead-Band

Disabling the use of a dead-band (which is the default mode of operation) results in operation like that with the dead-band, except when the accumulated error lies between –Threshold0 and +Threshold0. Operation in this region is as follows:

- If the accumulated error exceeds +Threshold0, the adjustment -ADJ0 will be applied. This adjustment will continue to be applied until:
    - The accumulated error drops below -Threshold0, in which case +ADJ0 will be applied.
    - The accumulated error continues to increase beyond +Threshold1, in which case -ADJ1 will be applied.
- If the accumulated error drops below -Threshold0, the adjustment +ADJ0 will be applied. This adjustment will continue to be applied until:
    - The accumulated rises to above +Threshold0, in which case -ADJ0 will be applied.
    - The accumulated error continues to decrease beyond -Threshold1, in which case +ADJ1 will be applied.

The result of the above is a 'hysteresis' effect when the accumulated error lies between –Threshold0 and +Threshold0.

#### 6.2.2.3.3 Numeric Representation

Each adjustment value is represented by an 8-bit, unsigned value. The least significant bit of the adjustment value is aligned with the least significant bit of the fractional portion of the PLL feedback divider ratio.
As described above, the accumulated error is represented by a signed, 9-bit value, taking on values from -256 to +255. The two programmable thresholds are each represented by an unsigned 7-bit value, capable of taking on values from 0 to +255.

#### 6.2.2.3.4 Notes on Programming

##### 6.2.2.3.4.1 Adjustment Granularity

Given the numeric representation of the programmable adjustment values, the smallest adjustment that can be made to the fractional portion of the PLL feedback divider ratio is approximately +/- 6.676e-6, while the largest possible adjustment is approximately +/- 0.0017.

#### 6.2.2.3.5 Accumulated Error vs. Sample Period

From the table, above, the number of Tracking Loop clock cycles per sample period can vary be 24, 48 or 96, depending on operating mode. It should be noted that an accumulated error matching this figure represents an accumulated error of one sample period. As such, the associated data FIFO will have gained or lost one sample.
In HDA mode, it is possible to have data streams of differing sample rates running simultaneously. In this situation, the streams with the higher sample rates will tend to fill or empty their associated data FIFOs more quickly than streams with lower sample rates. Thus, it is advisable to establish threshold programming that will prevent overflow/underflow for the higher sample rate streams. Specifically, a threshold of 24 will represent an accumulated error of one sample period for a data stream at 192Ksps.

##### 6.2.2.3.5.1 Audible Artifacts

Noting the interaction between the threshold values and the accumulated error, it can be recognized that programming TH0 and TH1 to very small values will result in the applied adjustment being changed very frequently. This is likely to result in unwanted artifacts in audio performance. Conversely, programming TH0 and TH1 to larger values will result in slower modifications to the PLL adjustment, reducing audible artifacts.

Similarly, it can also be recognized that programming the adjustment values (ADJ0 and ADJ1) to larger values will increase the rate at which the PLL adjustment value is modified, potentially resulting in unwanted artifacts.
A recommended starting point for programming is:

- *TH0 = 24 (representing 1 sample period at 192Ksps)*
- *TH1 = 36 (representing 1.5 sample periods at 192Ksps)*
- *ADJ0 = 1 (representing an adjustment of +/- 6.676e-6)*
- *ADJ1 = 2 (representing an adjustment of +/- 13.34e-6)*

### 6.2.3 Tracking Loop Registers

See PLL Trackloop registers

## 6.3 Clocks

### 6.3.1 Main Clocks

Below Table Shows Oceanside main clock sources.

| Clock Name | Maximum Frequency | Description |
|---|---|---|
| HDA_BCLK (cnxt_dt_hda_bclk) | (24,12,6) Mhz | HD Audio interface clock input to the chip. (source = C_BCLK) |
| I2S_BCLK (cnxt_dt_i2s_bclk) | Max clock freq < 25MHZ (1/2 cycle mode)<br>Max Clock Freq < 40 Mhz in Full cycle mode. | I2S Bit clock input in slave mode. (source = C_BCLK) |
| SCAN_BCLK (cnxt_dt_scan_bclk) | Scan Clock | Scan clock buffer (source = C_BCLK) |
| rcosc_clk | 6 MHZ clock (typical) | Internal RC Oscillator clock (Source= Oceanside_ana_top.rcosc_clk) |
| lp_rcosc_clk | 200 Khz (typical) | Low power rcosc clock (Source= Oceanside_ana_top.lf_rcosc_clk) |
| Clk_au | 147.47 Mhz | Audio PLL output (Oceanside_ana_top.pll_clkout) |
| I2s_mclk | Max clock freq < 25MHZ | Mclk for i2s mode to syn por_1p8_sm (source = C_HRSTN) |
| I2c_clk | 400 KHZ | C_HSCL |
| CLK_PI2S_A_SLV | 12MHz | i2s1_bclk (C_i2s1_bclk) |
| CLK_PI2S_B_SLV | 12MHz | i2s2_bclk (C_i2s2_bclk) |

### 6.3.2 Generated/Derived Clocks

Below table consists of generated/derived clocks in the design

| Clock Name | Maximum Frequency | Description |
|---|---|---|
| clk_24p5 | 24.576Mhz | Audio PLL divide by 6 (clk_au/6) |
| Clk_36p6 | 36.6MHz | Audio PLL div by 4 (clk_au/4) |
| Clk_12 | 12.28Mhz | Clk_24p5 /2 |
| Clk_6 | 6.144MHZ | Clk24p5/4 |
| Clk_3 | 3.072 | Clk24p5/8 |
| CLK_48K | 48K | Clk_6/128 |
| CLK_768K | 768K | Clk_6/8 |

| Clock Name | Maximum Frequency | Description |
|---|---|---|
| CLK_48M | 48M | Clk_au/3 |
| Clk_trackloop | 147.47/32 | Pll track loop clock |
| Spkr_clk | 3.072Mhz | Clk_6/2 |
| Offcal_clk | 384KHZ | Clk_6/16 (cpmp_clk/16) |
| Spkrcal_clk | 384 or 48 KHZ | (Clk_6/32) (clk_6/8) (internally state machine has a divided by 4.) |
| Spkr_prot_clk | 24.576 or 6MHZ | Clk_24p5 or (CLK_CTRL_SM) |
| Dc_det_clk | 12.28MHZ | Spkr_prot_clk/2 |
| Pwr_avg_clk | 24.576 | Spkr_prot_clk |
| Oc_clk | 384KHZ | Spkr_prot_clk/32 |
| Clk_jack | 200khz | Lp_rcosc |
| Beep_clk | 200KHZ | Lp_rcosc |
| i2s_bclk (master mode) | Aud_pll_clk/6 | I2S Master mode clk, i2s_pcm_hard_clk_div |
| Auxmode_rcosc_clk | 200KHZ | Lp_rcosc_clk |
| CLK_I2C_ACC | 14.7456MHz | I2C master module clock |
| CLK_PI2S_WFIFO | 3.072MHz | PI2S WFIFO clock |
| I2C_MST_CLK | 1MHz | I2C master output clock to another chip (C_mscl) |
| CLK_PI2S_A_MST | 18MHz | PI2S A master mode clock<br>(source = clk_au) |
| CLK_PI2S_B_MST | 18MHz | PI2S B master mode clock<br>(source = clk_au) |

### 6.3.3 Internal Gated Clocks

| Clock Name | Source Clock | Clk_Gate Control | Other Controls | Frequency |
|---|---|---|---|---|
| **Hclk_or_lp_rcosc** | Hclk or lp_rcosc | select= hd_clk_off \| bus_mode<br>reset=por_1p8_sm | Lp_rcosc if hd_clk_off or i2s_mode | 200KHz or 24Mhz. |
| **I2s_bclk_out (gated)** | I2S_BCLK (slave mode)<br>I2s_pcm_hard_clk_div (master mode) | Enable: i2s_bclk_en<br>Reset: i2s_rstn | Digital_bios_test[2]=polarity<br>I2s_pcm_master | < 25MhZ |
| **hclk_buf (gated) (hd_interface block)** | HDA_BCLK | Enable: ~bus_mode<br>Reset: por_1p8_sm | | 24/12/6 Mhz |
| **Hclk_azalia (azalea block)** | Hclk_buf | En: hclk_gate_en<br>Reset: chip_por | Digital_bios_test0[7] (defaut=1) enable azalea clock for state machime when in snm mode. | 24/12/6<br>This clock always on if hclk_buf on |
| **Hclk_low_pwr_gate / port_clk_gate_sm** | Clk_6<br>or<br>Rcosc_clk | select=<br>~audio_clk_off &<br>~no_clk<br>Reset=por_1p8_sm | Clk_6 if (audio_clk_off=0 and no_clk=0)<br>Otherwise rcosc_clk. | 6Mhz |
| **Hclk_low_pwr/clk_ctrl_sm** | Hclk_low_pwr_gate | En=clk_ctrl_sm_en<br>Reset = por_1p8_sm | Clk_ctrl_sm_en=<br>low_pwr_mode_en?<br>(beep_act\|aux_mode_clk_off\|snm_jsense\|~codec_clk_off\|jack_valid_en\|<br>offcal_en\|combo_jack_det) | 6mhz |
| **Cpmp_clk** | Port_clk_gate_sm | En=cpmp_pdb \|<br>cpmp_pdb_okb<br>Reset=por_1p8_sm | See cpmp-pdb term in RTL.<br>=portaEn\|porte_sn\|portm_en…<br>\|beep_en \|cpmp_clk_en | 6mhz |
| **Fifo_clk** | I2s_bclk_out (i2s mode)<br>or<br>Hclk_azalia (hd mode) | Bus_mode=1<br>i2s_bclk_out else<br>hclk_azalia | | <25MHZ |

| Register Block | | | |
|---|---|---|---|
| **Clock Name** | **Source Clock** | **Clock_Gate Control** | **Frequency** |
| Hclk_reg | Hclk_or_lp_rcosc | No gating | 200khz or 24Mhz |
| Hclk_reg_gated | Hclk_or_lp_rcosc | En= hclk_reg_gate_en (write access)<br>\| aux_mode | 200KHZ or 24Mhz |

| Azalia Block | | | |
|---|---|---|---|
| **Clock Name** | **Source Clock** | **Clock_Gate Control** | |
| Hclk_azalia | Hclk_azalia | | |
| Hclk_neg | ~hclk_azalia | | |
| Hclk_lp | Hclk_or_lp_rcosc | | |

| Fifo Block | | | |
|---|---|---|---|
| **Clock Name** | **Source Clock** | **Clock_Gate Control** | |
| Hclk_dac1, hclk_dac2, hclk_dac3 | Fifo_clk | En: Hclk_dacx_clk_en<br>Reset: chip_por | |
| Hclk_adc1, hclk_adc2, hclk_adc3 | Fifo_clk | En: hclk_adcX_clk_en<br>Reset: chip_por | |
| Hclk_spdif | Fifo_clk | Hclk_spdif_clk_en<br>Chip_por. | |

| Audio ADCs/DACs/Other Blocks | | | |
|---|---|---|---|
| **Clock Name** | **Source Clock** | **Clock_Gate Control** | **Block Name** |
| Dacs_clk | Clk_36p6 | En: Dacs_clk_en<br>Reset:chip_rstn_clk_au | Dacs_dsp, Oceanside_interpolator,vilano_signal_ramp, beep,dac_mem_mbist,dac_coeff, fifo_blk, muxes |
| Hp_clk | Clk_36p6 | En: hp_clk_en<br>Reset: chip_stn_clk_au | Hp_filter |
| Ed_clk | Clk_36p6 | En: ed_clk_en<br>Reset: chip_stn_clk_au | ED |
| Mbist_clk_dac | Clk_36p6 | En: bist_clk_force<br>Reset: chip_rstn_clk_au | Dac_mem_bist |
| Ana_adc_clk | Clk_3 | En: ana_sadc-clk_en<br>Chip_rstn_clk_au | Oceanside_ana_top |
| Ana_dac_clk | Clk_3 (3MHZ) | Dacs_clk_en<br>Chip_rstn_clk_au | Oceanside_ana_top |
| Spdif_tx_clk | Clk_24p5 | ~spdif_clk_off,<br>Chip_rstn_clk_au | Spdif_mod, spdif_gen, fifo_blk |
| Dac1_clk, dac2_clk | Ana_dac_clk (3MHZ) | DacX_clk_en,<br>chip_rstn_clk_au | Dacs_dsp |
| Bq_clk | Clk_3(48KHZ),<br>clk_6(96KHZ),<br>clk_12(192KHZ) | | Biquad EQ |
| Eq_2b_clk | Bq_clk | Eq_2b_clk_en,<br>chip_rstn_clk-au | Biquad iir,dacs_dsp,dac_mbist_con |
| Adcs_clk | Clk_6 | Adcs_clk_en<br>Chip_rstn_clk-au | Adcs_dsp |
| Adc1_clk, adc2_clk | Adcs_clk | adcX_clk_en<br>chip_rstn_clk_au | Adcs_dsp, fifo_blk |
| Mic_eq_clk_in | Adcs_clk or<br>Clk_12 | Select: mic_eq_clk_sel=1<br>clk_12 | |
| Adc_iir_clk | Mic_eq_clk_in | En: adc_iir_clk_en<br>Chip_rstn_clk_au | Adcs_adc,biquad_iir,adc_mbist_con |
| Mbist_clk_adc | Mic_eq_clk_in | En: bits_clk_force | Adc_mbits_con |
| Eqdrc_clk/ | Clk_24p5(192KHZ)<br>Clk_12(96KHZ), clk_6<br>(otherwise) | | |
| Drc_clk | Eqdrc_clk | Drc_clk_en | Spkr_drc |
| Eq_7b_clk | Eqdrc_clk | Eq_7b_clk_en | Dacs_dsp, eq_7b_drc, dac_mem_mbist |
| Adc3_clk | Eqdrc_clk | Adc3_clk_en | Fifo_blk |
| Clk_24 (redundant)<br>=spkr_prot_clk | Lp_rcosc / clk | | Codec_dsp,analog_ctrl,classd_spkr_prot,reg_blk |

### 6.3.4 Clock Relations

Sync group 1: {HSCL I2C_INT_CLK}
Sync group 2: {CLK_24P5 CLK_12M CLK_6M CLK_3M CLK_48K CLK_768K}
Sync group 3: {CLK_DC_DET OC_CLK}

### 6.3.5 Oceanside Clock Diagrams

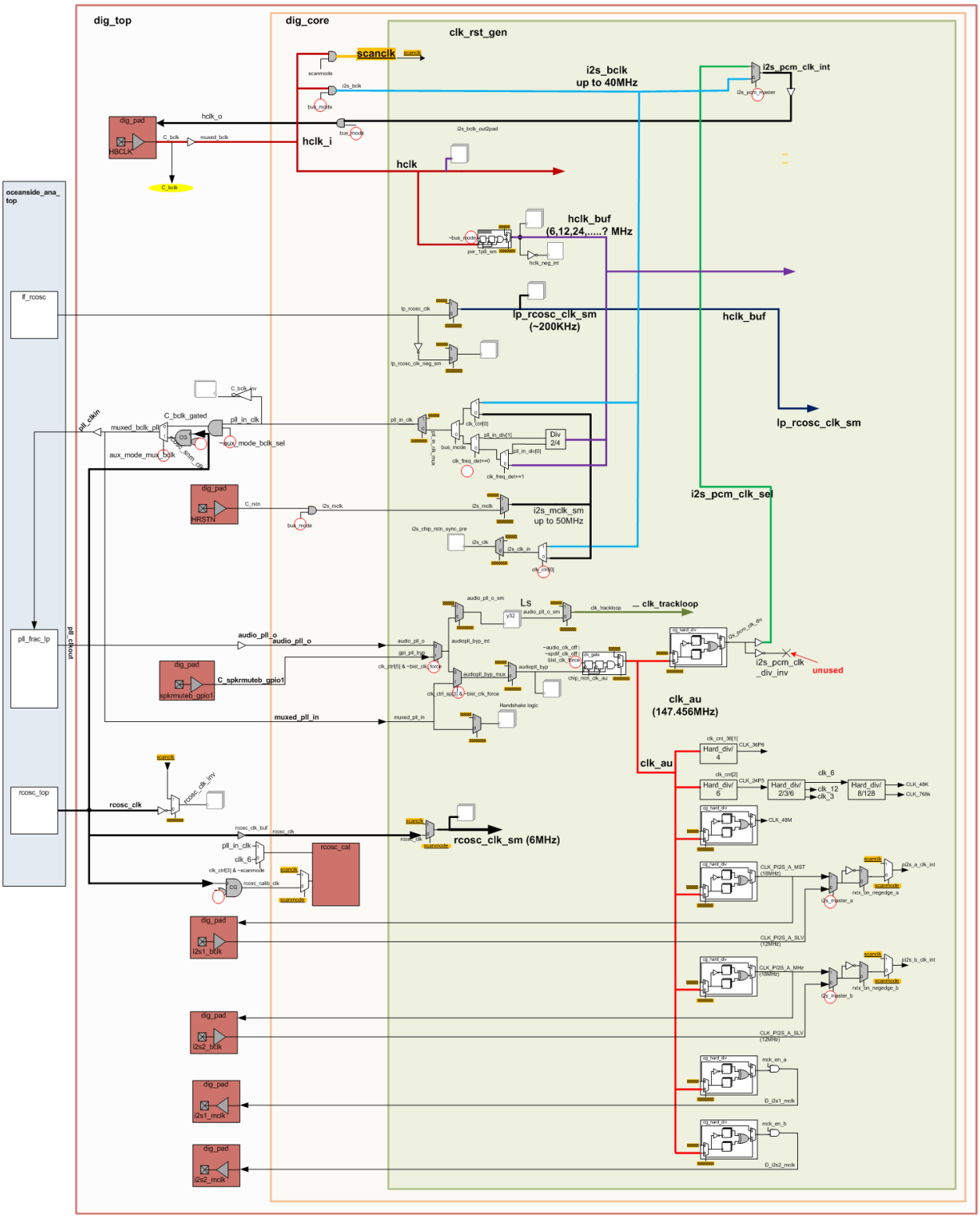



**Figure 6-2 Main Clock Diagram (CLK_SRC)**

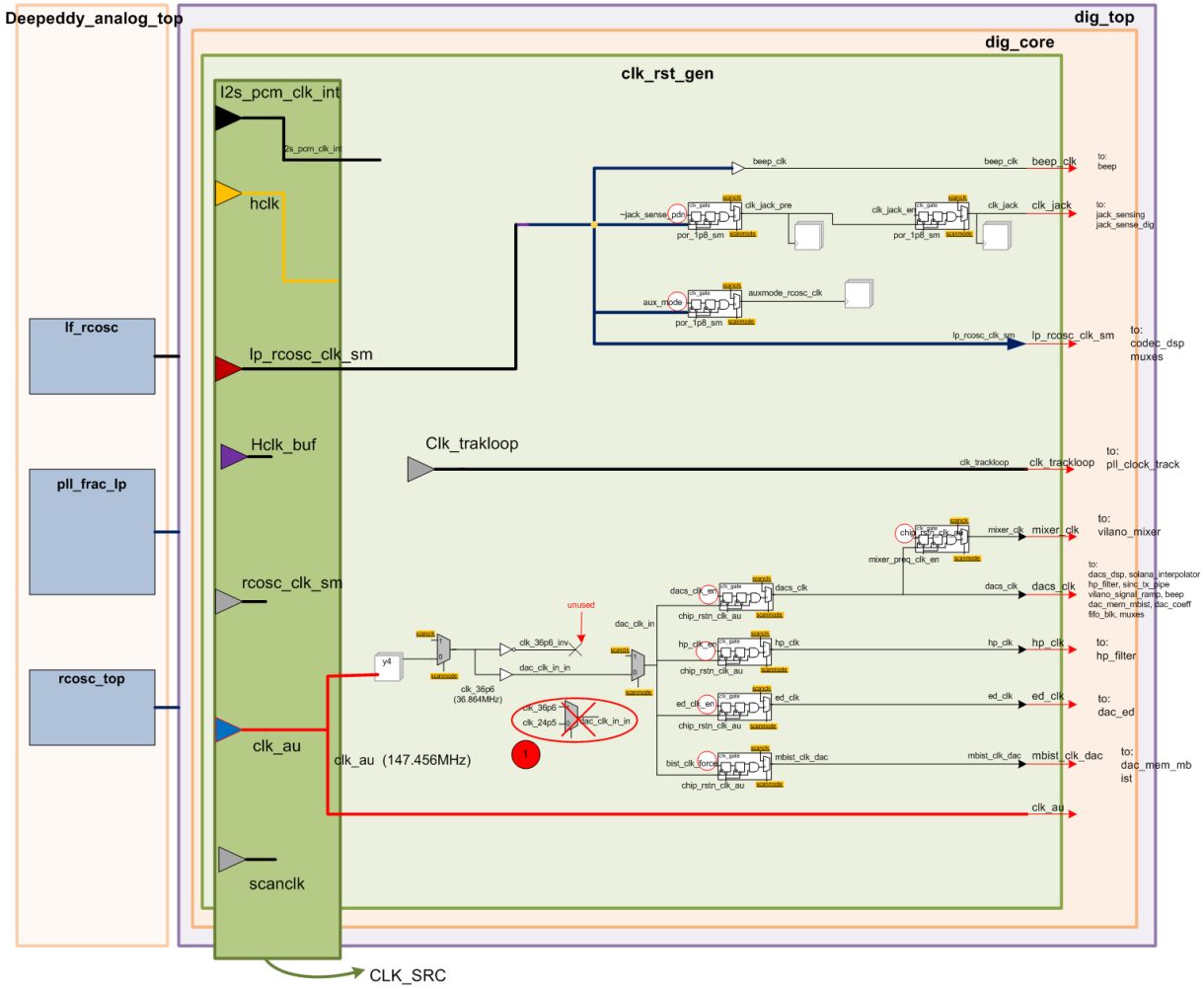



**Figure 6-3 Host Clocks, (Register, FIFO, Azalia, I2S)**

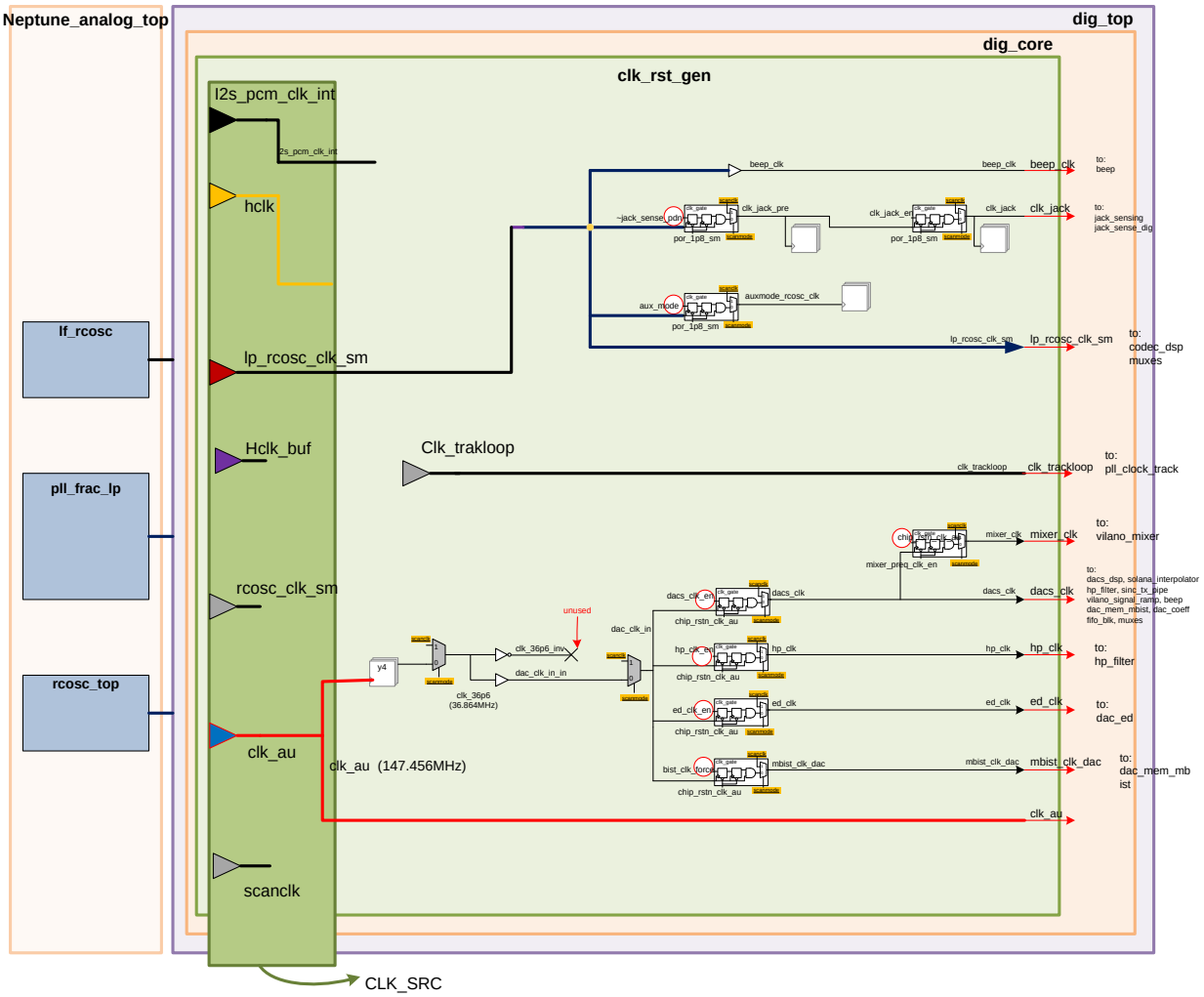



**Figure 6-4 Clocks Generated from clk_36p6, lp_rcosc_clk**

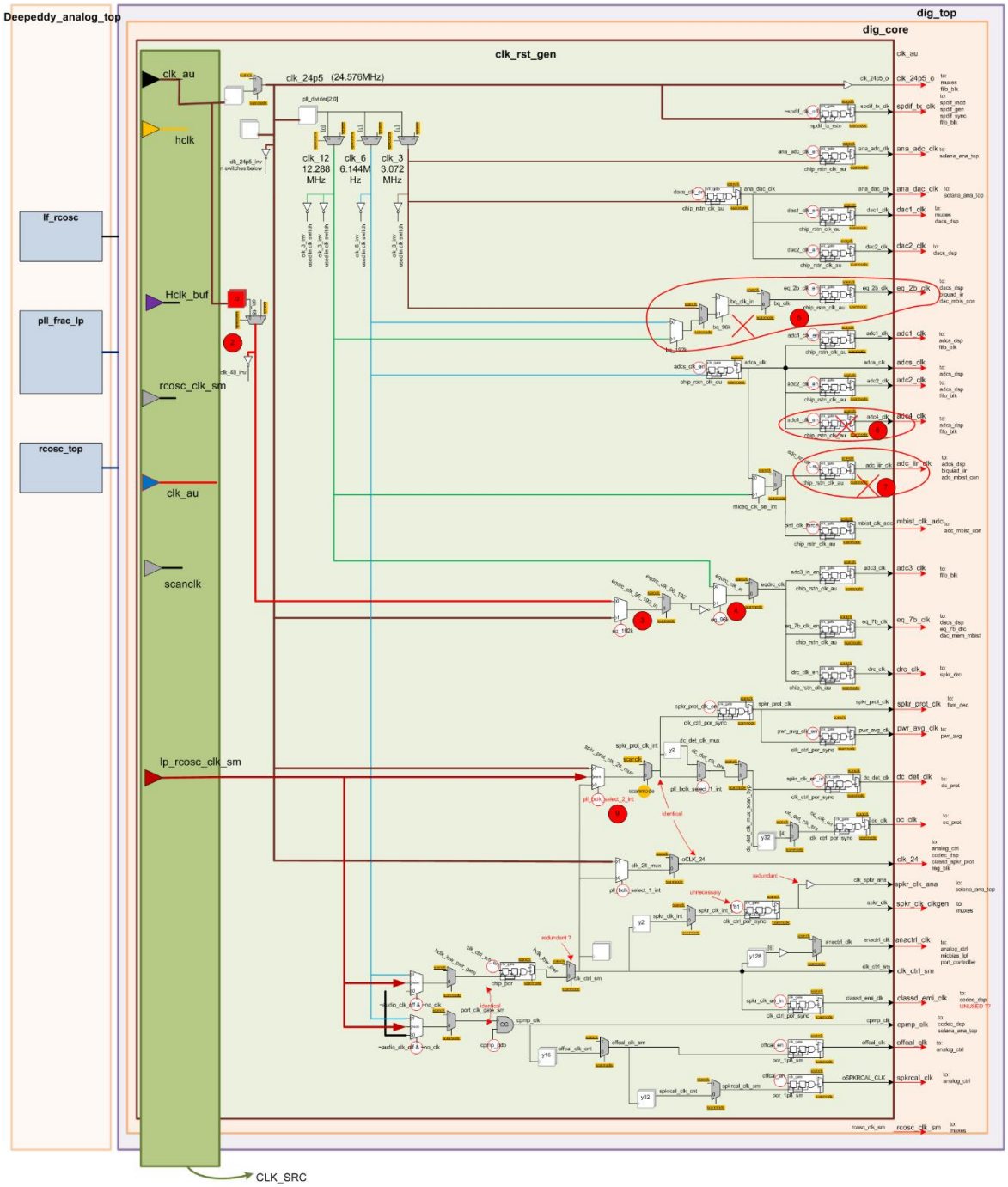
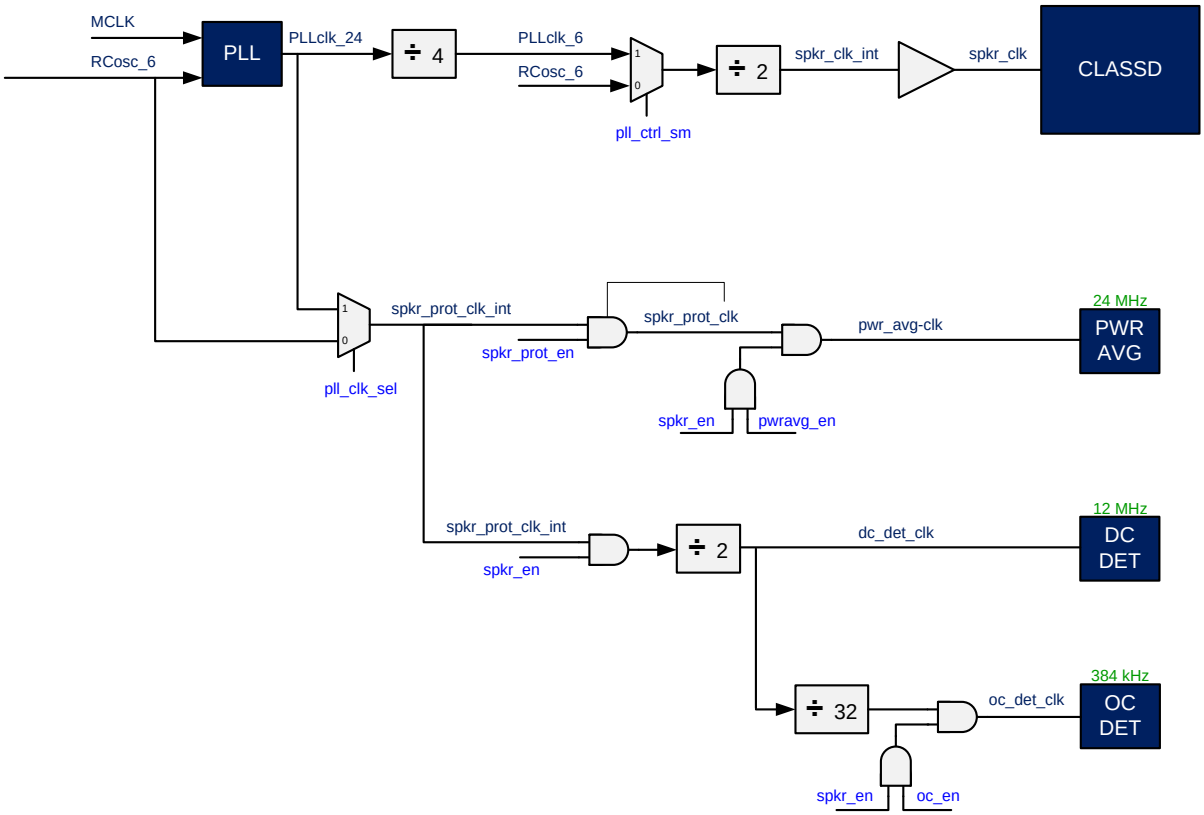


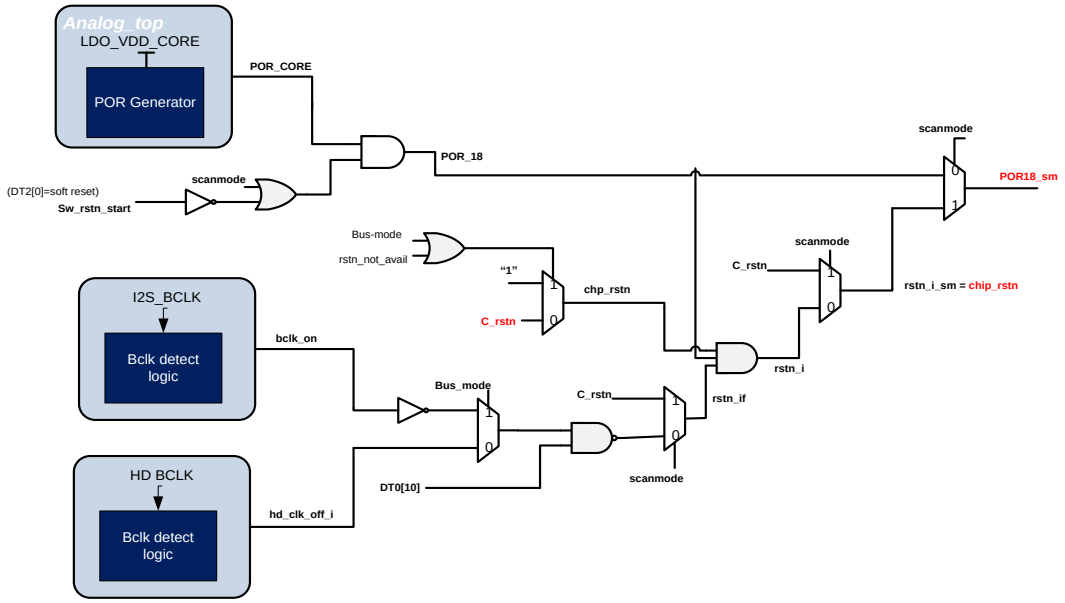

**Figure 6-5 Clocks Generated from clk_au (clk_24p5, clk_3 clk_6,clk_12).**

## 6.3.6 Speaker Protection Clocks Block diagram




## 6.4 Resets

Below shows main reset logic for Oceanside.



### 6.4.1 Soft-POR operation

ATE require the system-level reset between each ATE test for a stable start point of each test. But Oceanside lack of pins for the hardware reset control from pin. So, chip implement a soft-reset based on register.

This software reset will reset Oceanside digital to POR state. It will be 7 cycles of lp_rcosc_clk(200khz, free-running clock) long & will reset all digital logic except soft POR logic itself.

The Soft-POR bit is a protected bit. To use it, we must unlock it first by program the codec_test24/25 correctly with secret key:

Codec_Test24

Get Response - HD_A[27:0] = 0x1CE8000 Set - HD_A[27:11] = 0x1C68 I2C=7320

| Bit | Default | Description |
|---|---|---|
| 11:0 | 000 | Secret register |

Codec_Test25

Get Response - HD_A[27:0] = 0x1CE9000 Set - HD_A[27:11] = 0x1C69 I2C=7324

| Bit | Default | Description |
|---|---|---|
| 11:0 | 000 | Secret register |

After unlock, write digital_test2[0] to 1 will do the soft POR.

After soft POR, host can re-access the codec by I2C or re-start the enumeration with HDA.

Digital_Test2

Get Response - HD_A [27:0] = 0x1CC2000 Set – HD_A[27:11] = 0x1C42 I2C=7208

| Bit | Default | Description |
|---|---|---|
| 11:10 | 00000 | Send UM every frame<br>0: Send UM every other frame (default)<br>1: Send UM every frame |
| 10 | 0 | Reserved |
| 9 | 0 | 1-Bypass SPDIF clock off logic |
| 7:5 | 101 | HP limit value delect – range it from 3dB to 10dB<br>0x5 : 8dB (default) |
| 4 | 1<br>Protected | Override temperature shutdown signal<br>0 = Disable over temp shutdown<br>1 = put Class D in D3 , and disable the output |
| 3 | 0<br>Protected | Override under voltage shutdown signal<br>0: put Class D in D3 , and disable the output<br>1: Do not change power or output states. |
| 2 | 0<br>Protected | Override ClassD 5V POR signal.<br>0: Allow normal 5V POR signal to drive Class D POR<br>1: Force POR out of reset |
| 1 | 0<br>Protected | Override ClassD short circuit signal.<br>0: Allow normal short circuit detection<br>1: Force short circuit detection off |
| 0 | 0<br>Protected | Soft POR reset<br>1: enable soft POR reset |

### 6.4.2 SnM Reset (Known as D3live reset)

See Chapter Below:

SnM (Music Request Mode)

### 6.4.3 Link Reset

To minimize link housekeeping sequences, the HDA Link has been designed with a single initialization sequence that always follows a reset sequence.

A link reset is signaled on the Link by assertion of the **RST#** signal with pin, or monitoring the SYNC pin for missing Frame Syncs when there's no **RST#** pin.

Link reset results in all Link interface logic in both codec and controller, including registers, being initialized to their default state.

In Oceanside, only the azalia interface enters reset when link reset is asserted, the codec functional logic/registers are not reset.

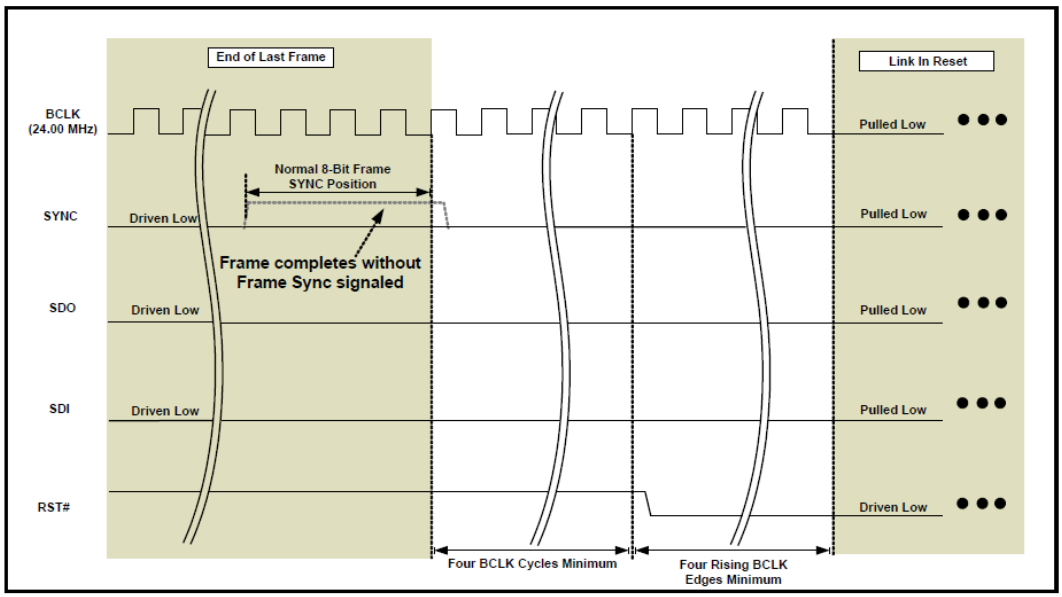



**Figure 28. Link Reset Entry Sequence**

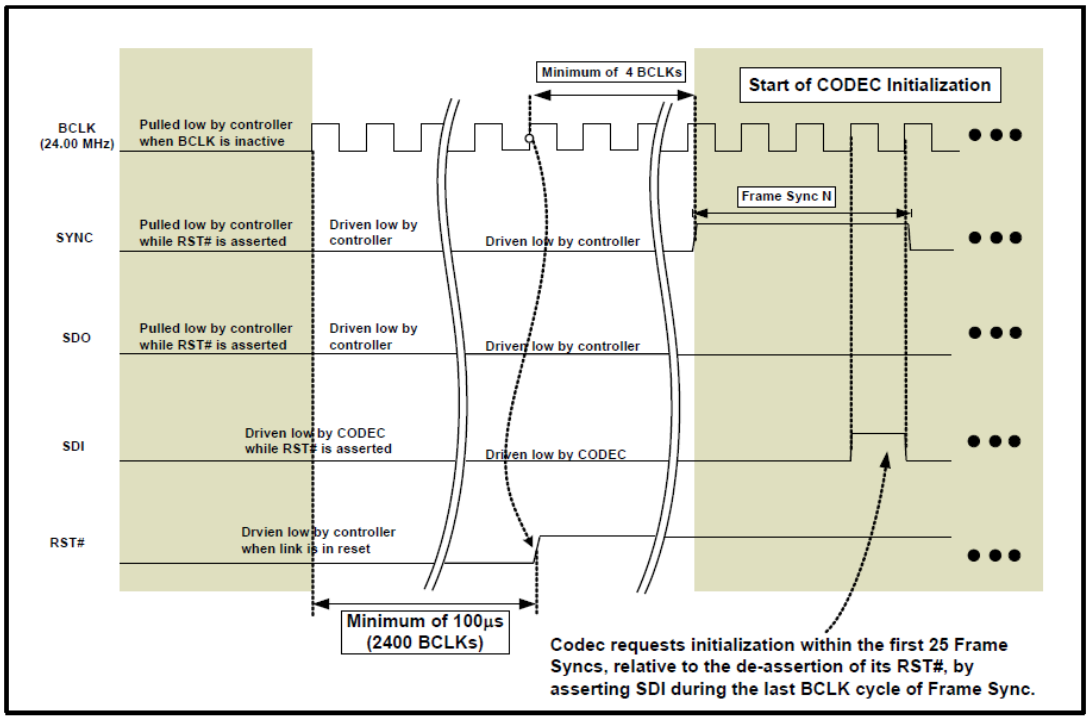



**Figure 29. Link Reset Exit Sequence**

Oceanside default without the **RST#** pin, if the bclk on, but the sync lost, it need at least 2 frames (2 x 21us) to enter the link reset.
If the bclk gone, detection logic work on lp rc osc clock(200Khz), need at least 5 cycles(25us) to know the bclk is gone to enter the link reset.
(1). There's SnM mode depend on link reset in Oceanside. Codec only can enter the SnM mode when link reset assert. For the SnM mode in I2S mode, Oceanside also treat the I2S bclk gone as the link reset & share the same logic as HDA mode that can make the SnM mode can work in I2S mode.
(2). Beep also use this link reset signal: when codec in link reset, the beep path be forced to analog path.

# 7 Dual Mode Host Interface

## 7.1 Overview

The host interface can operate in two distinct modes; HD Audio mode, and I2S mode. The default interface mode is selected by a bond option. The mode may be changed by command using the HD Audio interface or the I2C interface. It is expected that the bus mode will be changed only when the bus is idle (no active audio streams). Clocks may or may not be active when the request is made.

### 7.1.1 HD Audio mode

In HD Audio mode, the host interface will be compliant with the HD Audio specification 1.0a and ECR HDA048-A. The bus interface is comprised of BCLK, SYNC, SDI, SDO, and an optional RESET# pin. The I2C host interface is available but command and control is expected on the HD Audio interface. Although not expected, in the case that both the I2C interface and the HD Audio interface access the CODEC at the same time, both interfaces shall meet their respective timing requirements.

### 7.1.2 I2S Mode

In I2S mode, the host interface will use the I2S or TDM protocol to transfer audio data to and from the host. A slave I2C compatible interface is used for control of the CODEC.
The host interface is comprised of MCLK, BCLK, LRCLK, SDI, SDO, SDA, and SCL pins. I2S, Left Justified, Right Justified, and Intel TDM formats are supported. Master and Slave modes are supported. The default is 48KHz I2S format with 16-bit data.

MCLK is the master, high speed clock provided by the host. A range of about 5.5 to 50 Mhz is desired. It is usually, but not always, an integer multiple of the sample rate. See the PLL section for more detail.

BCLK is the bit, or shift, clock used to for data recovery. This is typically 64x the sample rate and is configurable. BCLK is an input in slave mode and an output in master mode.

LRCLK is the frame clock and is typically at the sample rate of the stream. TDM modes may allow multiple samples per channel per frame. LRCLK is an input in slave mode and an output in master mode. Both short and long sync profiles are supported. The start of frame may be programmed to be on the rising or falling edge. In addition, the sync, at a minimum, is programmable for a 0 or 1 BCLK delay to support I2S and TDM modes.

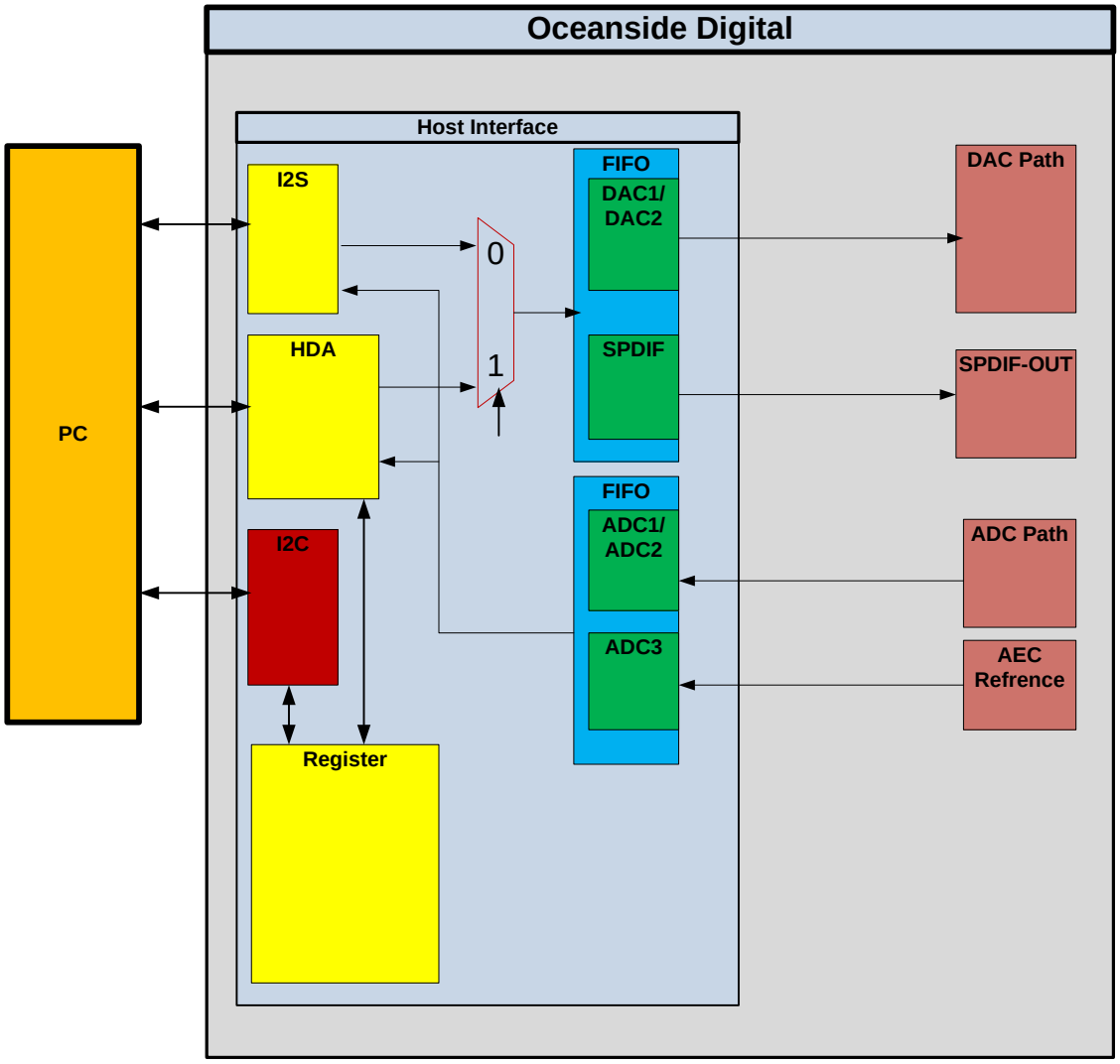
**Intel requirements.**
For some Intel platforms, the Master clock will be 25, 24, or 19.2MHz. The CODEC will be in slave mode. BCLK will be 2.40MHz. There will be 50 clocks per frame and the frame rate will be 48KHz. The LRCLK and BCLK will be an exact multiple of 48KHz when MCLK is 25MHz. Intel Haswell had about an 8ppm error on the sample rate. (At 25MHz MCLK, we believe that the competition uses the 2.4MHz BCLK for their PLL. For 19.2MHz, 24MHz, and 25MHz the PLL is set to 1.2MHz for the PFD and the VCO is 98.4MHz)

### 7.1.3 Functional Blocks

The host Interface Block consists of the Following Functional Blocks:

- Azalia Host interface (HD Audio Interface)
- Register Bank
- Audio in FIFO Block.
- I2C Host Slave (Register Control).
- I2S_PCM  Host Slave (Audio Data)

Block Diagram for Host Interface




## 7.2 Host Control Interface

The Cx8400/Cx8450 provides both an HDA interface and an I2C host control interface. Both can be used to access registers within the device. Both interfaces may be active simultaneously, however accessing the same register through both interfaces simultaneously will produce unpredictable results.
It is important to note that the requirements of the HDA interface specification impose significant constraints on the organization of registers within the device. These constraints include, for example, grouping of registers within HDA "nodes". These nodes may or may not correspond directly with device functional modules. This organization is reflected in both the HDA and I2C views of the internal register set.

### 7.2.1 Control Access – HDA Interface

The HDA interface is used to convey both host control interactions and audio data, to and from the CX8400/CX8450 device. On the other hand, the I2C host control interface is limited to control interactions.
HDA supports multiple attached codec devices, each individually addressable.

Unlike an I2C interface, the HDA interface provides separate data signals for each direction of data flow (from the host to the codec, and from the codec to the host). HDA also provides separate return data signals (from the codec to the host) for each attached codec. (I2C, on the other hand, shares a single data signal for all attached devices, and for data flow in both directions.)
Also unlike an I2C interface, the HDA interface implements data framing, with frames occurring at a constant rate of 48KHz. Within each frame, the HDA interface allocates a time slot for one command, sent from the host, which can address one, or all, of the attached codecs. Also within each frame, but in the reverse direction (from codec to host), the HDA interface allocates a time slot for one response.

A codec receiving a command will issue one response for each command it receives, with the response being transmitted in the subsequent data frame. Generally, every command requires a specific response, even if the command is used to write to registers within the Oceanside device.

#### 7.2.1.1 Commands (From Host to Codec)

The HDA command time slot (one per frame time, from the host to a codec) is 40-bits in length, and includes bit fields for address information as well as for an action specifier field, known as a Verb. The command slot is organized as follows:

| Reserved (0's) 8b | Codec Addr 4b | Node ID 8b | Verb 20b |
|---|---|---|---|
| 39 32 | 31 28 | 27 20 | 19 0 |

The HDA command address fields specify both a 4-bit codec address (to select one of potentially many codecs attached to an HDA interface) and an 8-bit Node Identifier (to select one node within the addressed codec).

#### 7.2.1.2 Verbs

As mentioned above, Verbs are "action specifiers". The HDA specification defines Verbs in a somewhat abstract sense, in terms of the action(s) each Verb causes.
In a practical sense, however, Verbs can be thought of as writing or reading values to/from registers at specific addresses. Whether writing or reading a given address causes any specific action or activity to occur can be thought of as the definition of the register at that address.

The encoding of the Verb field within the Command slot contains a Verb ID and a Payload. The syntax of these sub-fields can take on one of two forms, as follows:

| Verb ID<br>4b | Payload<br>16b |
|---|---|

OR

| Verb ID<br>12b | Payload<br>8b |
|---|---|
| 19 | 0 |

Note that a value of either 0x7 or 0xF in the first 4 bits of the Verb indicate that a 12-bit Verb ID syntax is used. Any other value represents a 4-bit Verb ID.
Verbs are generally divided between “Get” verbs and “Set” verbs.
“Get” verbs cause a response (returned in the next frame) which contains information from the codec, without otherwise affecting the codec’s operation. A “Get” verb can generally be thought of as reading the value of a register (or registers) within the codec, and returning the value in the Response field of the next frame.
A “Set” verb, on the other hand, can generally be thought of as writing a value to a specific location.

It should be noted that “Get” verbs are generally distinguished from “Set” verbs by the value of the most significant bit of the Verb ID. A Verb ID MSb of 1 indicates a “Get” (Read) verb, while a Verb ID MSb of 0 indicates a “Set” (Write) verb.
As noted previously, the address fields within a command are sufficient to identify a specific codec, and aa specific node within that codec. However, the address fields do not identify a specific register location within a node. The contents of each Verb provide this additional level of addressing.
For many Verbs, the value of the Verb ID is sufficient to uniquely identify the target of a given command access. In other cases, however, the contents of the Payload must also be considered. For example, the “Get Parameter” Verb uses a 12-bit Verb ID, and also uses the 8 bits of the Verb Payload as a Parameter ID. As another example, the “Get Connection List Entry” Verb also utilizes an 8-bit payload value, in this case to identify a specific connection list entry.
As a result of all of the above, all of the following must be taken into account in order to identify which register is accessed by a given Verb:

- Node ID
  - 8 bit field.
  - Only 7 bits used for “short form” NIDs.
  - Only 6 bits used w/in the CXxxxxx

- Verb ID
  - Up to 12 bits in length
  - MSb signifies read/write (leaving 11 bits)

- Verb Payload
  - 8 bit field (in the case of 12 bit VID)

The HDA interface therefore uses up to a total of 6+11+8 = 25 bits of address to uniquely identify registers, depending on which Verb is being used.

#### 7.2.1.3 Responses (From Codec to Host)

The HDA Response time slot (one per frame, from the codec to the host) is 36 bits in length. A Response occurs in the frame immediately following the frame which contained the initiating command. The Response time slot is organized as follows:

| V | U | 0 0 | Response Payload 32b |
|---|---|---|---|
| 35 | 34 | 33 32 | 31 0 |

Bit 35 (V) = “Valid”

Bit 34 (U) = “Unsolicited”

Noting that the Response payload is 32 bits (4 bytes) in length, each HDA Verb can be treated as effectively accesses a 4-byte register. (It can also be noted that in many cases, only a subset of the available 32 bits contain useful information.)

Since a given frame may not require a Response, the “Valid” bit field is used to identify valid Responses from a codec to the host.

In most cases, Responses are produced in response to a command in the previous frame. However, in some cases, a codec may need to send ‘unsolicited’ information to the host. The “Unsolicited” bit field is used to indicate that a Response is the result of an internal event within the codec. Unsolicited Responses provide an interrupt-like capability between a codec and the host.

### 7.2.2 Control Access – I2C Interface

The CX8400/8450 provides both an HDA interface and an I2C host control interface. Both can be used to access registers within the device.
The I2C interface utilizes a 7-bit slave address, to address multiple slave devices attached to the same interface. This is similar to the 4-bit codec address implemented by the HDA interface. The slave address is static, however, rather than assigned dynamically by the host (as is the case for HDA).
The I2C interface within the CX8400/8450 utilizes a 16-bit “sub-address” to access individual registers within the device. Per convention, the I2C sub-address is transmitted immediately after the I2C slave address, during write transfers only. Data to be written to the addressed register(s) then follow immediately after the sub-address.
Also by convention, read transfers utilize the sub-address established by the most recent write transfer. As a result, a read transfer is typically preceeded by a write transfer, which transmits only a sub-address value, with no data values. The read transfer then follows immediately after the write transfer.

**For complete explanation I2C Interface, refer to chapter I2C Host interface below.**

### 7.2.3 Register Addressing – HDA Compliant Verbs

This sub-section describes a mapping of HDA compliant Verbs into I2C register addresses.

As described above, the HDA interface uses 25 bits of address to uniquely identify registers within the device (not counting the Codec ID). Also as described, the I2C interface provides an address field of only 16 bits (not counting the slave address). It should also be kept in mind that HDA "addresses" are each capable of accessing a 32-bit (4 byte) quantity, in contrast to the byte oriented nature of I2C transfers. As a result, there is no simple mapping between HDA addresses and I2C addresses.
It can be noted, however, that, taking into account all of the currently defined Parameter ID values, and also taking into account the relatively short connection lists in the CX8400/8450 device, a total of only 58 unique "Get" Verbs are required to address registers within the CX8400/8450. Combined with a 6-bit Node ID, and adding 2 bits for to enable byte addressing, only 14 bits of address are actually required. Similarly, a total of only 46 "Set" (or "Execute") Verbs are required by HDA to address all registers within the CX8400/8450.
Taking the above into account, each HDA verb is translated into a corresponding I2C address, as described here.

| NID[5:0] | Derived[7:0] | Byte Offset |
|---|---|---|
| 15 10 | 9 2 | 1 0 |

- Bits [15:10] of the I2C address contain the lower 6 bits of the HDA Node ID value.
- Bits [1:0] of the I2C address are used to identify a byte within the 32-bit value addressable by the HDA Verb.
- Bits [9:2] of the I2C address contain an 8-bit value derived from the HDA Verb ID value and, where appropriate, the Verb Payload value. This derivation is as follows:

  - "Execute Function Reset" is encoded as 0xFF
  - "Get Parameter" is encoded as (0x80 + Parameter Index)
    - For example, Revision ID, with Parameter Index = 2, is encoded as 0x82
  - "Get Connection List Entry" is encoded as (0x40 + Index>>2).
    - Note that only the 'short form' list is supported, in which each list entry is a byte quantity.
    - Note that the "Index" value is a byte offset. The 2 least significant bits of the "Index" are therefore placed into the byte offset field of the derived address.
    - For example
      - With an index of 0x2, it is encoded as 0x40, with a byte offset field of 0x2
      - With an index of 0x5, it is encoded as 0x41, with a byte offset field of 0x1

  - "Get/Set Amp Gain/Mute" is encoded as (0x60 + AmpSpecifier)
    - AmpSpecifier consists of 2 1-bit values for "Output/Input" and "Left/Right, as well as a 3- bit "Index"
      - The bits within AmpSpecifier are ordered as {O/I, L/R, Index[2:0]}

- Unlike the Amplifier Gain/Mute Set verb, which can be applied to multiple amplifiers simultaneously, the I2C access can be applied to just a single amplifier at a time.

- Remaining verbs with 4-bit VID values are encoded as : 0xF0 + VID[2:0]
  - Note that VID[3] is also Verb[19], which indicates read vs. write.
  - For example, "Get Coefficient Index" with VID[2:0]=0xD, is encoded as 0xFD
- All of the remaining defined Verbs use a 12-bit VID format, but the least significant 6 bits are sufficient to distinguish between them. These are encoded using simply the value of the least significant 6 bits of the VID.
  - For example, "Get Beep Generation", with a VID of 0xF0A, is encoded as 0x0A.
  - However, it should be noted that the SetDigital Converter Control, Set Implementation Identification and Set Configuration Default verbs define a unique SET verb for writing to each byte of a 32-bit quantity. The individual bytes of these 32-bit values are mapped at consecutive byte addresses within the I2C address space.

For lists of the I2C addresses for all defined HDA verbs**: Refer to link below**
**Oceanside HDA Register Spec**

## 7.2.4 Register Addressing – Vendor Defined Verbs

This sub-section describes a mapping of "Vendor Defined" Verbs into I2C register addresses.
For Complete Address Mapping Refer to Doc in Share-Point.

### 7.2.4.1 Node 1B

Node 1B is used to control audio processing features, such as EQ and DRC. Node 1B implements one HDA compliant verb (Get Parameter - Audio Widget Capabilities), along with a number of Vendor Defined Verbs.

The (Get Parameter, Audio Widget Capabilities) Verb is mapped as described in the previous section. Specifically, with a Node ID of 1B, the address is 0x6E24. Since this verb returns a 32-bit value, they are mapped to the 4 consecutive byte addresses 0x6E24, 0x6E25, 0x6E26 and 0x6E27.

All of the Vendor Defined Verbs in node 1B utilize the following bit fields:

| Verb ID<br>4b | 0 0 | FRegister Select<br>6b | Payload<br>8b |
|---|---|---|---|
| 19 16 | 15 14 | 13 8 | 7 0 |

Consistent with HDA specified verbs, the most significant bit of the Verb ID field is used to select "Set" (MSb = 0) vs. "Get" (MSb = 1) verbs. As a result, only the least significant 3 bits of the Verb ID are used as a part of the I2C address. Also, each of the "Get" verbs returns only an 8-bit field, corresponding to the 8 bits written to by the related "Set" verb. As a result, consecutive register select values correspond to consecutive I2C byte addresses.
The I2C address for each register is therefore constructed as :

| Low 6bits of Node ID<br>0 1 1 0 1 1 | Verb ID<br>3b | 0 | Register Select<br>6b |
|---|---|---|---|
| 15 10 | 9 7 | 6 | 5 0 |

### 7.2.4.2 Node 1C

Node 1C is used to control various proprietary and test related features. Node 1C implements one HDA compliant verb (Get Parameter - Audio Widget Capabilities), along with a number of Vendor Defined Verbs.
The (Get Parameter, Audio Widget Capabilities) Verb is mapped as described in the previous section. Specifically, with a Node ID of 1C, the address is 0x7224. Since this verb returns a 32-bit value, they are mapped to the 4 consecutive byte addresses 0x7224, 0x7225, 0x7226 and 0x7227.
All of the Vendor Defined Verbs in node 1C utilize the following bit fields:

| Verb ID<br>4b | | Reg Select<br>6b | | Paylod<br>12b | |
|---|---|---|---|---|---|
| 19 | 16 | 15 | 10 | 9 | 0 |

Consistent with HDA specified verbs, the most significant bit of the Verb ID field is used to select "Set" (MSb = 0) vs. "Get" (MSb = 1) verbs. As a result, only the least significant 3 bits of the Verb ID are used

as a part of the I2C address. Also, each of the "Get" verbs returns only an 12-bit field, corresponding to the 12 bits written to by the related "Set" verb. In the I2C address space, these 12 bits are mapped to two consecutive bytes, with bits [7:0] in the first byte, and bits [11:8] in the lower nibble of the second byte. The I2C address for each register is therefore constructed as shown here for the first byte (bits 7:0 of the 12-bit payload) :

| Low 6 bits of Node ID<br>0 1 1 1 0 0 | | Verb ID<br>3b | | 0 | Reg Select<br>4b | | 0 | 0 |
|---|---|---|---|---|---|---|---|---|
| 15 | 10 | 9 | 7 | 6 | 5 | 2 | 1 | 0 |

And as shown here for the second byte (containing bits 11:8 of the 12-bit payload) :

| Low 6 bits of Node ID<br>0 1 1 1 0 0 | | Verb ID<br>3b | | 0 | Reg Select<br>4b | | 0 | 1 |
|---|---|---|---|---|---|---|---|---|
| 15 | 10 | 9 | 7 | 6 | 5 | 2 | 1 | 0 |

### I2C Host interface

The I2C host interface is a slave interface comprised of the HSCL (Host I2C SCL) and HSDA (Host I2C SDA) pins. These are dedicated pins that interface with the host SOC or other controller.
The interface is compatible with I2C and SMBus and supports at least 100KHz and 400KHz data rates.
The default address may be re-programmed by using a register/verb access that takes effect in the next bus cycle. The default address is 7h33 (0110011R/nW) but may be changed to 7h37 (0110111R/nW) through a bond option.

5) The I2C slave interface is always active
   a) This allows a guaranteed means of switching bus modes.
   b) It may be difficult to manage disabling access where we won't create a condition where we get stuck in one mode or another.
6) All commands are available to the I2C and HD Audio interfaces.
7) Typically, one control interface is active while the other is idle or off.
   a) In I2S mode, only the I2C interface can be active considering that we have not defined a means of transferring control data across the TDM/I2S interface.
   b) In HD Audio mode both can be active.

8) If a request comes from both the I2C and HD Audio interfaces at the same time:
   a) Both interfaces will meet their timing requirements. Valid data must be read.
   b) No attempt will be taken to prevent one interface from overwriting data from the other. Accesses to a register are on a first come first served basis. In the case of an absolute tie, it is expected that the I2C access will be held off until the HD Audio access is complete due to the more restrictive HD Audio timing requirements.

# 8 I2S_PCM Host Interface

## 8.1 General description

The device supports a configurable Digital Audio data Interface. Four different audio data formats are supported. They are:

- $I^2S$ with patch for Intel mode
- Left Justified
- Right Justified
- PCM mode

The digital audio interface uses these pins: (both DAC and ADC share same Bit Clock and same Frame Clock).

1. Host Interface Signals:
   - tx_bclk/rx_bclk: ADC/DAC bit clock, for synchronization
   - tx_lrck_in/rx_lrck_in: ADC/DAC Left/Right data alignment clock
   - txd1_data: ADC data output line
   - rxd1_in: DAC data input line
   - Tx_lrck_out/Rx_lrck_out: Lrck clock for Master mode.
   - Txd1_oen ADC output Enable
   - Tx_data_on_negedge_bclk/rx_data_on_negedge_bclk (No Connection, not used), Instead uses digital_bios_msb[0]
   - I2s_pcm_tx_master/i2s_pcm_rx_master: Master Mode

2. DMA Interface Signals
   - Tx[1:10]_data[23:0] (Input)
   - Tx[1:10]_dreq (output)
   - Tx[1:10]_data_valid (input)
   - Rx1[1:6]_data[23:0] (output)
   - Rx[1:6]_data_en (output)

3. Control Interface signals

   - I2s_pcm_reg[1:7] [31:0]

Digital Audio Data interface in transmit direction can handle 10 channels. All channels can be transmitted on single data line (txd1_data) in PCM format. Both the data lines by default will work as per tx_bclk bit clock and tx_lrck data alignment clock.

Digital Audio Data interface in receive direction can handle up to 6 channels. All channels can be received on single data line (rxd1_data).

All 4 audio data formats share BCLK and LRCK. In BCLK and LRCK sharing mode, the bit clocks and Left/Right data alignment clocks of ADCs and DACs are shared. The bit clock and LRCK will be configured as outputs when Oceanside operates as a master and as inputs when it is a slave.

Time Division Multiplexing (TDM) is available in all four data format modes.

PCM operation is supported using the PCM mode.

The audio data interface support sample rates from 8K to 192 KHz. The supported sample widths are 8, 16 and 24 bit.

Time Division Multiplexing (TDM) allows multiple devices to transfer data simultaneously on the same data line. Oceanside ADCs and DACs support TDM in master and slave modes for all data formats and word lengths. TDM is enabled by using the register fields TX_DATA_ONE_LINE (always set to "1"), TX_TRI_N and RX_DATA_ONE_LINE (always set to "1") (see register I2S_REG1, I2S_REG2).

Note: Oceanside is a 24-bit device. If user operates Oceanside in 32-bit mode, then 8 LSBs will be ignored on receiving side and not driven on transmitting side. It is therefore recommended to add a pull-down resistor, if necessary, to tx_data line and rx_data line in TDM mode.

## 8.2 Programmer's Reference

When in master mode the tx/rx bit clocks are generated by internal PLL. The Bit clock generation registers are also defined in Clock-Gen register space (see I2SPCM_CLK_CTRL register). All other registers required to configure Digital Audio Interface are defined in the sections below. To set Oceanside audio interface to master mode set TX/RX_MASTER field HIGH.

## 8.3 Theory of Operation

The data bits on serial data line are identified by slots. A frame is split into several chunks of slots. Each slot is 8 bits wide. In any mode of operation data of each transmit or receive channel is transmitted or received in slots. A channel is identified by its first slot number. The first slot starts from ZERO. For example, a 48KHz frame with a bit clock of 3.072MHz will have 64 bit clocks per frame. So, we have 8 slots in a frame (named slot-0, slot-1, …, slot-7). If channel-1 and channel-2 are each 24-bits wide they occupy 3 slots each, like {slot-0, slot-1, slot2} and {slot-4, slot-5, slot-6} respectively. Since there is only one data line, channels cannot overlap their slot numbers.

The definition of different modes and the register settings required to configure a specific mode are described in the below section.

Note: Each mode can be operated in Normal mode (non-TDM) or TDM mode. When no specific mode is mentioned, Normal mode takes priority.

In Oceanside, there is only one configuration where ADC's and DAC's should be always running at same rate. So only 4-wire (Rx and TX clocks are same) mode is supported.

## 8.4 I2S Mode

$I^2S$ bus has three types of wires. The serial bit clock is BCLK, Left/Right data alignment Clock (LRCK or word strobe - WS) and serial data. The serial data and word strobe are always driven on negative edge of BCLK. MSB bit of data is transmitted first.

Oceanside supports up to 10 ADC channels and 6 DAC channels. Accordingly, there are 1 $I^2S$ ADC serial output data line and 1 $I^2S$ DAC serial input data lines. The audio interface can operate in normal (non-TDM) and TDM modes. In TDM mode all 10 ADC channels and all 6 channels of DAC of data come out on i2s_tx_data_1 and i2s_rx_data_1 respectively. $I^2S$ is configured to share BCLK and LRCK signals for ADC and DAC. In Oceanside ADC's and DAC's always Run at same rate.

$I^2S$ protocol uses LRCK to define whether the data is being transmitted for the left channel or for right channel. LRCK is low for the left channel and high for the right channel. LRCK need not be symmetrical. There is a delay of one clock bit from the time the LRCK signal changes state to the first data bit(MSB) on the data line. The data is written MSB first and is valid on the rising edge of the bit clock. If the data sample is wider than programmed sample width, the later incoming extra bits are ignored.

When the number of bits in the sample word matches the number of clocks per channel the last bit of each word arrives after the LRCK signal changes state.

To configure the audio interface to $I^2S$ Normal mode, select TX/RX_WS_POL field HIGH, TX/RX_DATA_ONE_LINE field (High) , TX/RX_DSTART_DLY field HIGH and TX/RX_DATA_NEG_BCLK field HIGH. TX/RX_WS_WID, TX/RX_FRM_LEN are used to describe the LRCK and Frame length relation. TX/RX_SLOT_n corresponds to first slot number of transmit/receive channel-n.

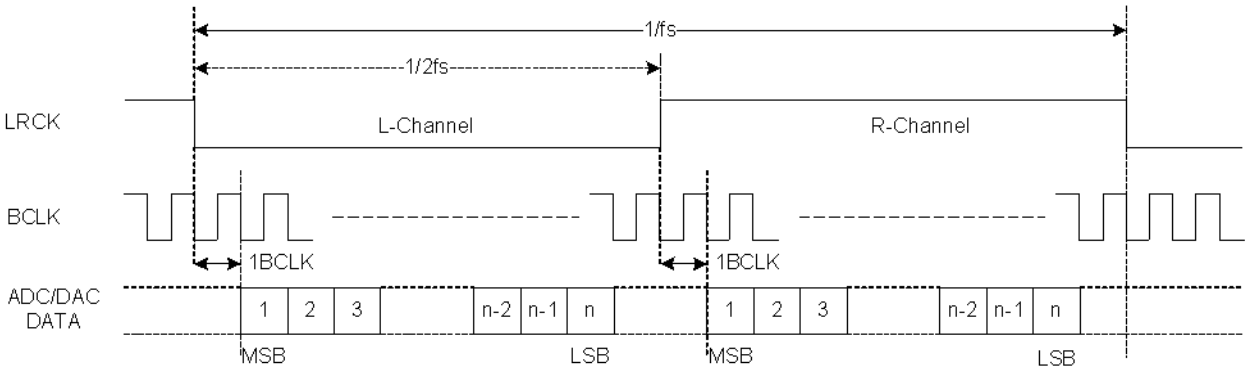



**Figure 8.4-1 I2S Justified Audio Interface (assuming n-bit word length)**

Configuring the audio interface to $I^2S$ TDM mode is like Normal mode with an exception to set TX/RX_DATA_ONE_LINE field HIGH (TX_TRI_N field LOW). The field TX_TRI_N will configure whether a tri-state on transmit data is needed when there is no active data on data line. The TX/RX_SLOT_n should be programmed according the respective slot positions.

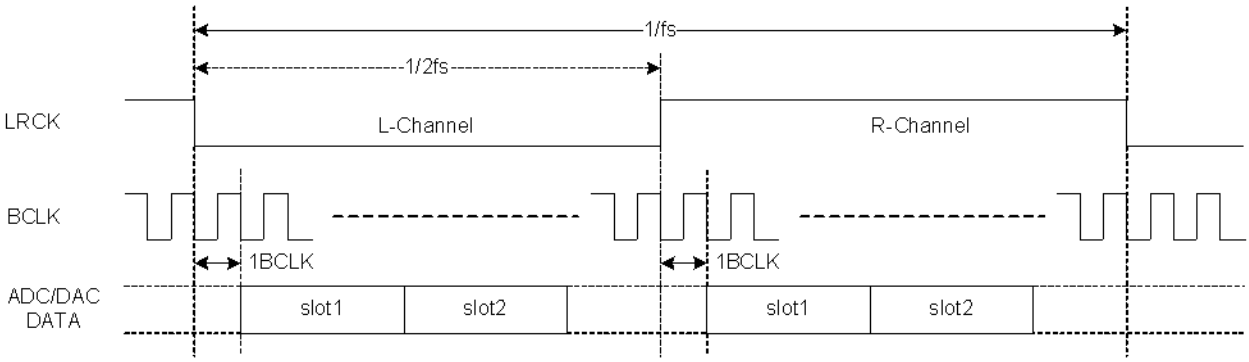



**Figure 8.4-2 I2S Justified Audio Interface in TDM mode**

## 8.4.1 Left Justified Mode

Left Justified Mode is a special case of $I^2S$ mode. LRCK is high for the left channel and low for the right channel. The MSB appears on the data line following the LRCK transition on the same BCLK. The data is written MSB first and is valid on the rising edge of bit clock. Once the programmed sample width is taken any remaining bits are ignored. If the LRCK toggles before the full word length is read the remaining bits are zeroed.

To program the audio interface to Left Justified Normal mode, select TX/RX_WS_POL field LOW, TX/RX_DATA_ONE_LINE field LOW, TX/RX_DSTART_DLY field LOW and TX/RX_DATA_NEG_BCLK field HIGH. TX/RX_WS_WID, TX/RX_FRM_LEN are used to describe the LRCK and Frame length relation. TX/RX_SLOT_n corresponds to first slot number of transmit/receive channel-n.

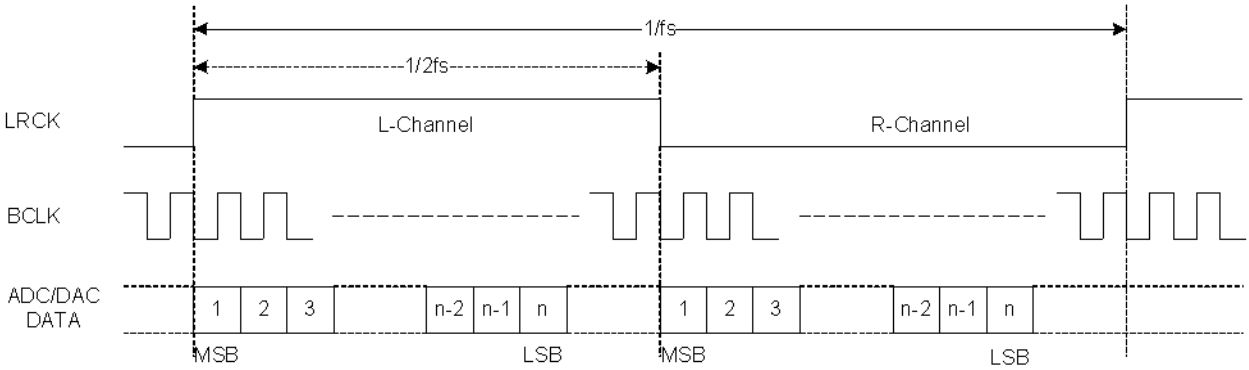



**Figure 8.4-3 Left Justified Audio Interface (assuming n-bit word length)**

Configuring the audio interface to Left Justified TDM mode is like Left Justified Normal mode with an exception to set TX/RX_DATA_ONE_LINE field HIGH. The field TX_TRI_N will configure whether a tristate on transmit data is needed when there is no active data on data line. The TX/RX_SLOT_n should be programmed according the respective slot positions.

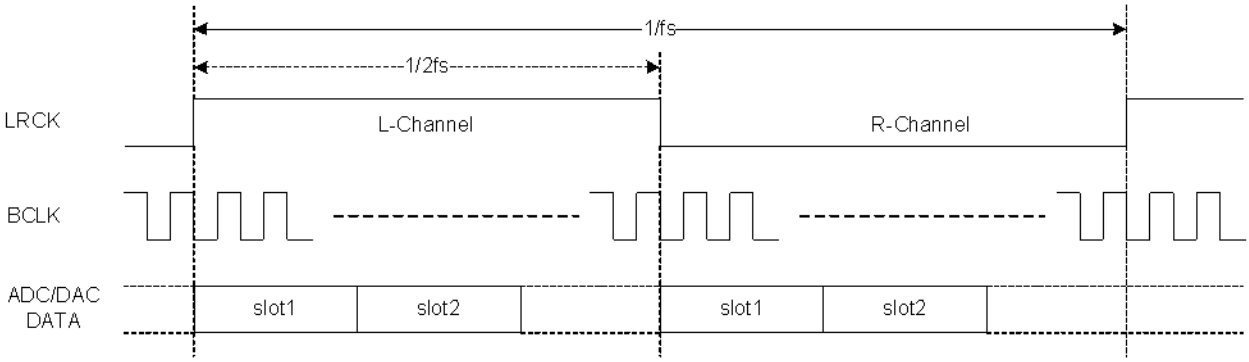



**Figure 8.4-4 Left Justified Audio Interface in TDM mode**

## 8.4.2 Right Justified Mode

Right Justified Mode is a special case of $I^2S$ mode. LRCK is high for the left channel and low for the right channel like Left Justified Mode. In Right Justified mode the LSB of data is always clocked by the last bit clock before the LRCK transitions. Data is captured in a 24bit shift register until LRCK toggles. Once LRCK toggles the last 24, 16 or 8 bits are transferred to the channel indicated by the previous state of LRCK. The data is written MSB first and is valid on the rising edge of bit clock. All leading bits are ignored. The number of bit clock delays for the first valid data bit from the toggle of LRCK is programmable though I2S_TX/RX_BCNT_DLY field. Although the bit count delay field is programmable, the right justified mode data makes more sense when the valid data bits are shifted to an extent where the LSB of data is clocked by the last bit clock before LRCK toggles.

To program the audio interface to Right Justified Normal mode, select TX/RX_WS_POL field LOW, TX/RX_DATA_ONE_LINE field LOW, TX/RX_DSTART_DLY field LOW and TX/RX_DATA_NEG_BCLK field HIGH. TX/RX_WS_WID, TX/RX_FRM_LEN are used to describe the LRCK and Frame length relation. TX/RX_SLOT_n corresponds to first slot number of transmit/receive channel-n. The TX/RX_SLOT_n should be programmed according the respective slot positions.

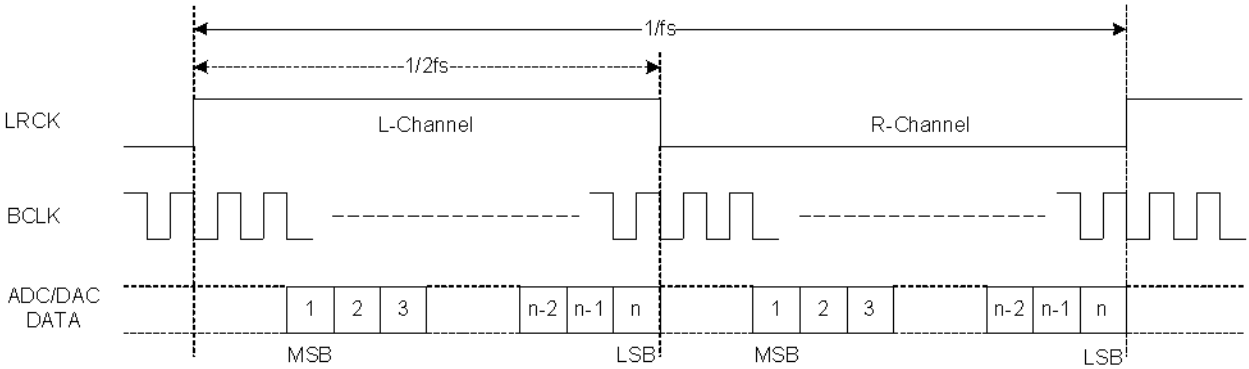



**Figure 8.4-5 Right Justified Audio Interface (assuming n-bit word length)**

Configuring the audio interface to Right Justified TDM mode is like Right Justified Normal mode with an exception to set TX/RX_DATA_ONE_LINE field HIGH. The field TX_TRI_N will configure whether a tri-state on transmit data is needed when there is no active data on data line. The TX/RX_SLOT_n should be programmed according the respective slot positions.

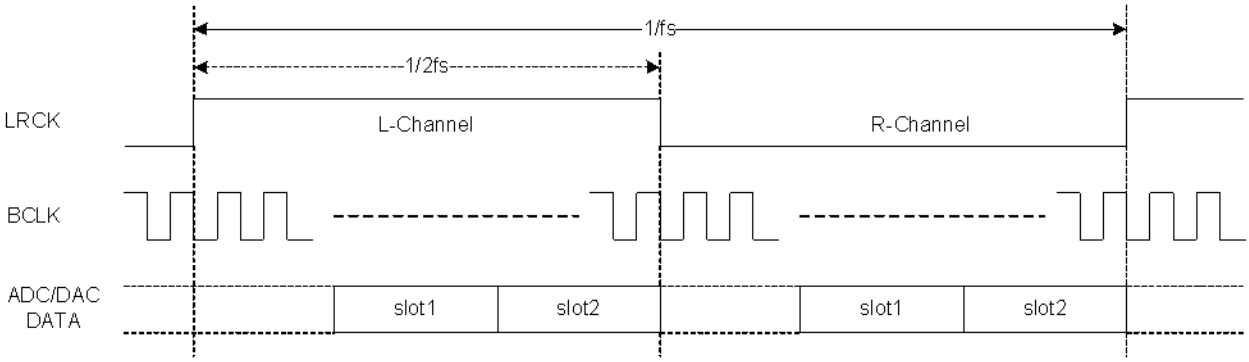



**Figure 8.4-6 Right Justified Audio Interface in TDM mode**

## 8.4.3 Intel Mode

Intel Mode is a special case of $I^2S$ mode. It's 25/25 cycles per frame with 24bit data width. Oceanside only support slot width 8, the frame length must be multiply of 8. To support Intel mode, Oceanside use a patch to halt the internal counter at specify position for specify cycles, that make the configuration still be slot based on multiply of 8 bits. To make most flexibility, the position & pause cycles are programmable.

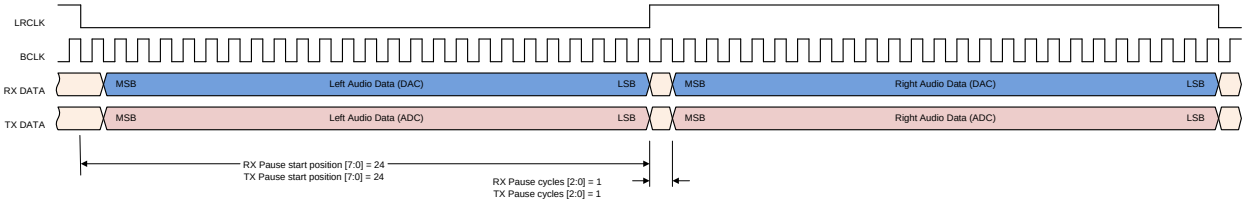



**Figure 8.4-7 Intel mode with Philips mode**

To program the audio interface to Intel node, select TX/RX_WS_WID, TX/RX_FRM_LEN to 24bit data & 48bit frame length. TX/RX_SLOT_0 corresponds to first slot number 0, TX/RX_SLOT_1 corresponds to first slot number 3. (3 x 8 = 24bit). Set TX/RX_PAUSE_START to 'd24 & TX/RX_PAUSE_CYCLE to 'd1.

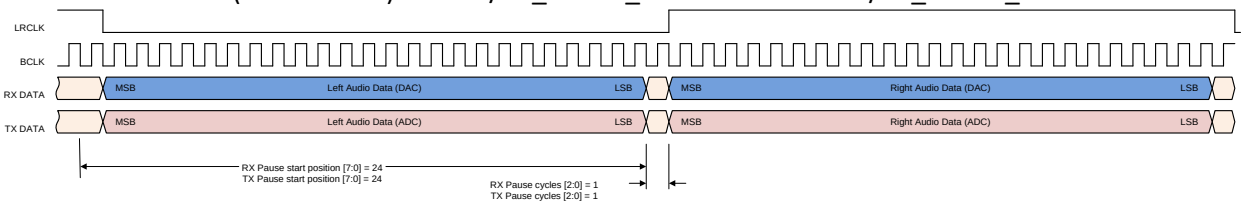



**Figure 8.4-8 Intel mode with Sony mode**

## 8.4.4 PCM Mode

PCM mode has three types of wires: The serial clock, Frame sync pulse (or LRCK), serial data line. Serial data and frame sync pulse is driven on positive edge of serial clock. In a normal PCM mode all data channels are transmitted/received on single data line and data line is tri-stated when there is no active data on line. In special cases of PCM mode channels may be split onto different data lines as per need.

Oceanside supports two modes of PCM operations. They are:

1. Short Frame Sync mode
2. Long Frame Sync mode

PCM is supported in PCM interface mode.
PCM mode can also be configured to drive/receive data and frame sync on negative edge of serial clock.

### 8.4.4.1 PCM Short Frame Sync Mode

In PCM Short Frame Sync mode the falling edge of LRCK indicates the start of the PCM word. LRCK is one clock long. Data is driven out on the rising edge of BCLK after the LRCK pulse.
To configure Oceanside PCM to Short Frame Sync normal mode select TX/RX_WS_POL field LOW, TX/RX_DSTART_DLY field HIGH and TX/RX_DATA_NEG_BCLK field LOW. TX/RX_WS_WID field is set to 7'h00. Select TX/RX_DATA_ONE_LINE field HIGH. TX/RX_FRM_LEN is used to describe the Frame length. TX/RX_SLOT_n corresponds to first slot number of transmit/receive channel-n.

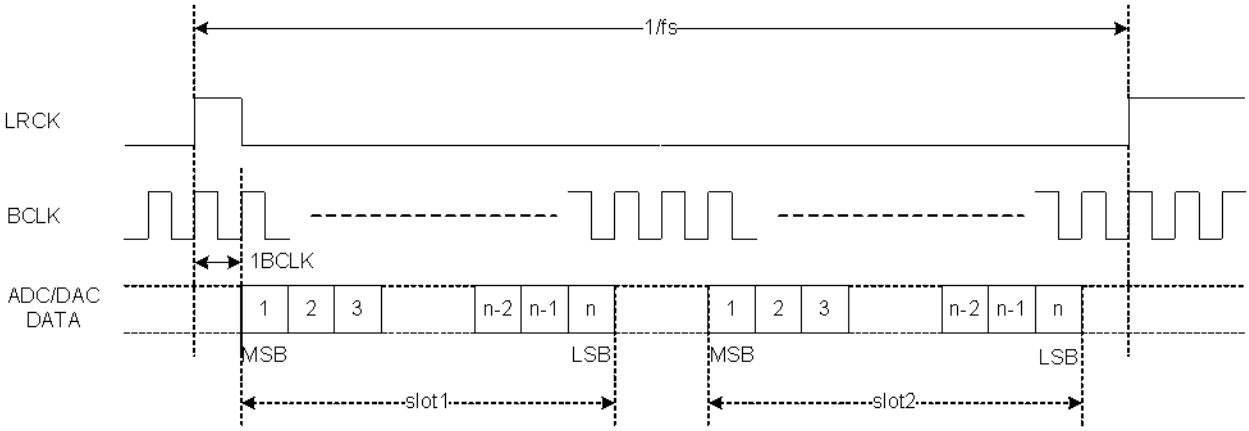



**Figure 8.4-9 PCM Short Frame Sync mode**

If data is time multiplexed with other channels from different CODECs in TDM mode TX_TRI_N is set LOW.

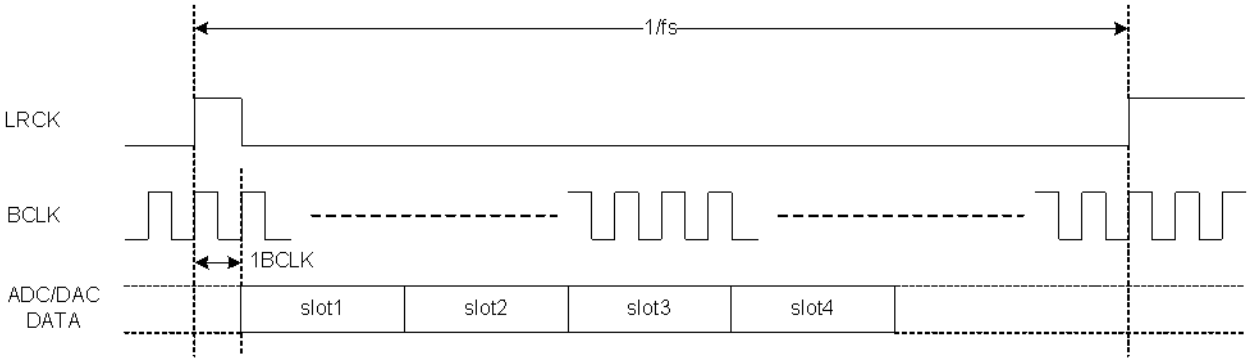



**Figure 8.4-10 PCM Short Frame Sync in TDM mode**

### 8.4.4.2 PCM Long Sync Frame Mode

In PCM Long Frame Sync mode the rising edge of LRCK indicates the start of the PCM word. LRCK is at least one Bclock long. Data is driven out on the rising edge of BCLK synchronizing with the rising edge of LRCK.
To configure Oceanside PCM to (long) Frame Sync normal mode select TX/RX_WS_POL field LOW, TX/RX_DSTART_DLY field LOW and TX/RX_DATA_NEG_BCLK field LOW. The TX/RX_WS_WID field is set

according to the number of bit clocks LRCK is HIGH. Select TX/RX_DATA_ONE_LINE field HIGH (Should be always high). TX/RX_FRM_LEN is used to describe the Frame length. TX/RX_SLOT_n corresponds to first slot number of transmit/receive channel-n.

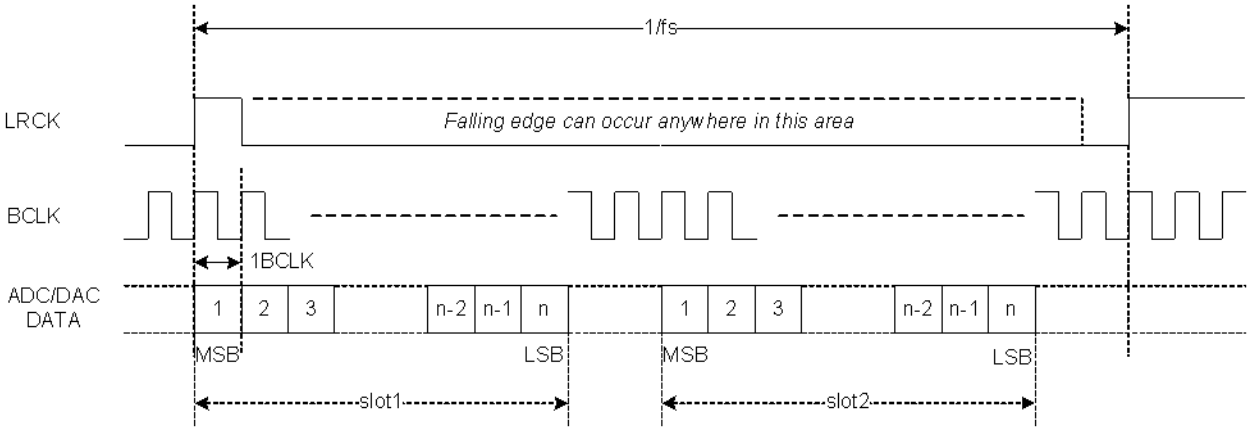



**Figure 8.4-11 PCM Long Frame Sync mode**

If data is time multiplexed with other channels from different CODECs in TDM mode TX_TRI_N is set LOW.

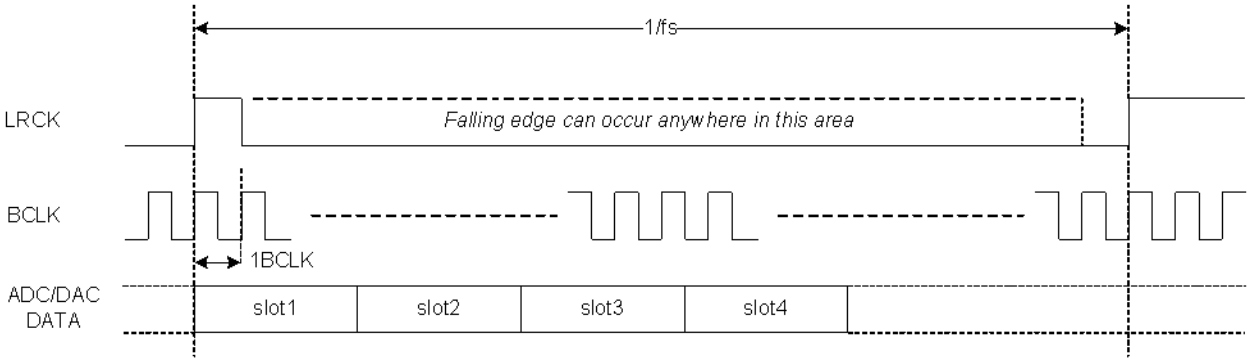



**Figure 8.4-12 PCM Long Frame Sync in TDM mode**

PCM can also be configured to work on negative edge of BCLK by setting TX/RX_DATA_NEG_BCLK field HIGH (use digital_bios_msb[0]) .

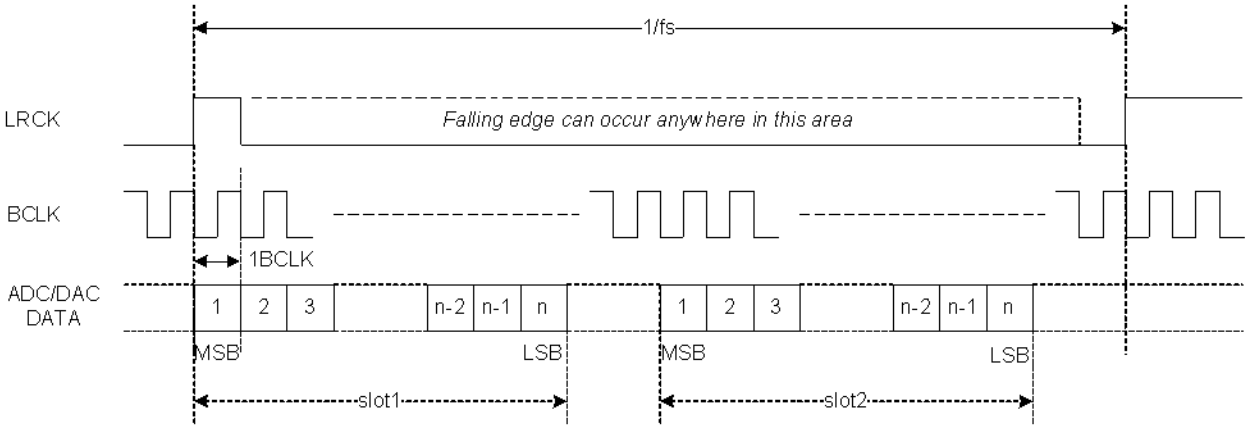



**Figure 8.4-13 PCM mode running on negative edge of BCLK**

## 8.5 Module Usage

In I2S mode, the host interface will use the I2S or TDM protocol to transfer audio data to and from the host. A slave I2C compatible interface is used for control of the CODEC. The host interface is comprised of MCLK, BCLK, LRCLK, SDI, SDO, SDA, and SCL pins. I2S, Left Justified, Right Justified, and Intel TDM formats are supported. Master and Slave modes are supported. The default is 48KHz I2S format with 16-bit data.

MCLK is the master, high speed clock provided by the host. A range of about 5.5 to 50 MHz is desired. It is usually, but not always, an integer multiple of the sample rate. See the PLL section for more detail.

BCLK is the bit, or shift, clock used to for data recovery. This is typically 64x the sample rate and is configurable. BCLK is an input in slave mode and an output in master mode.

LRCLK is the frame clock and is typically at the sample rate of the stream. TDM modes may allow multiple samples per channel per frame. LRCLK is an input in slave mode and an output in master mode. Both short and long sync profiles are supported. The start of frame may be programmed to be on the rising or falling edge. In addition, the sync, at a minimum, is programmable for a 0 or 1 BCLK delay to support I2S and TDM modes.

Intel requirements.

For some Intel platforms, the Master clock will be 25, 24, or 19.2MHz. The CODEC will be in slave mode. BCLK will be 2.40MHz. There will be 50 clocks per frame and the frame rate will be 48KHz. The LRCLK and BCLK will be an exact multiple of 48KHz when MCLK is 25MHz. Intel Haswell had about an 8ppm error on the sample rate. (At 25MHz MCLK, we believe that the competition uses the 2.4MHz BCLK for their PLL. For 19.2MHz, 24MHz, and 25MHz the PLL is set to 1.2MHz for the PFD and the VCO is 98.4MHz).

Refer to diagram below for System level integration.

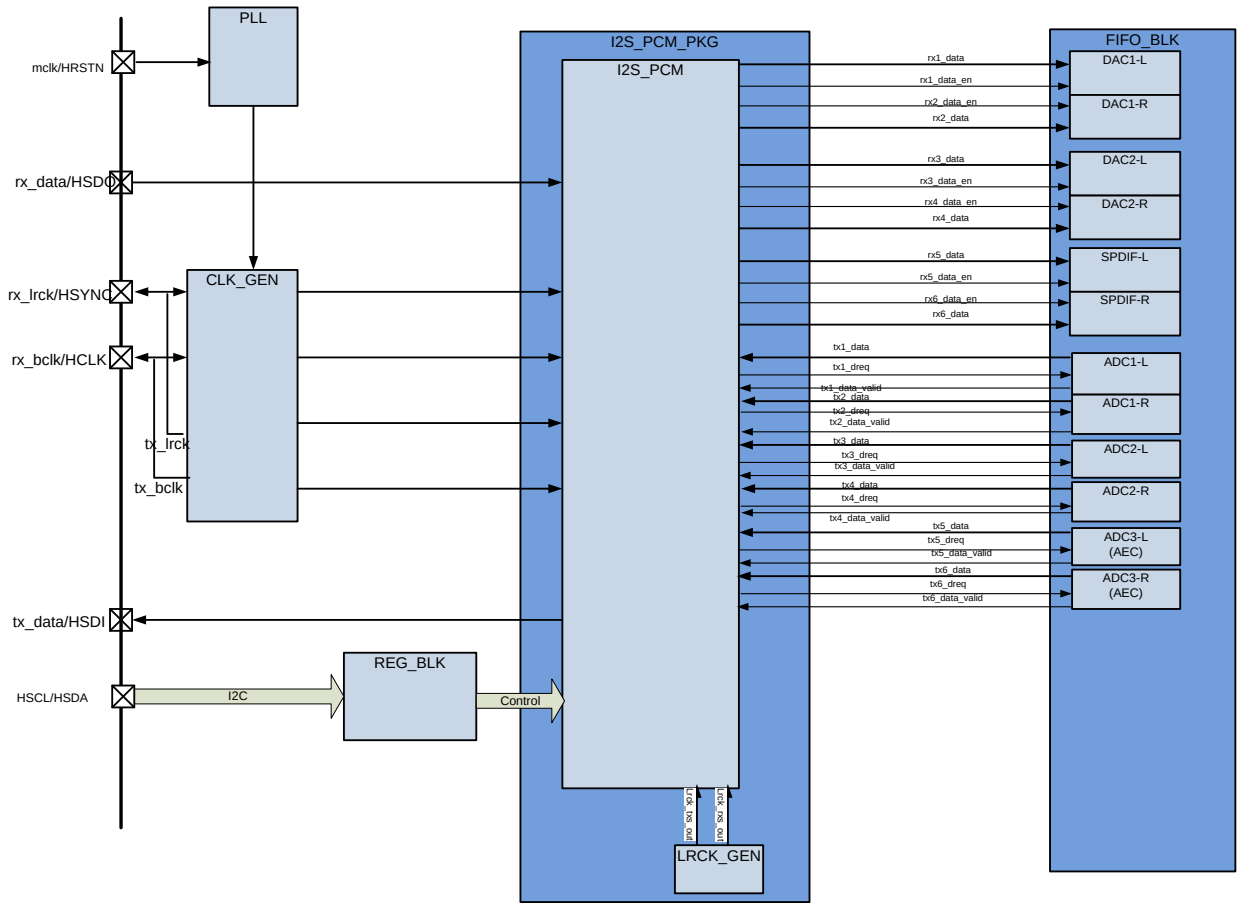



**Figure 8.5-1 I2S Module Integration Diagram**

## 8.5.1 I2s interface Design Update

1. Programmable bit to fall back to Deep Eddy mode in case, things need to be cared is that the timing also need to be fall back, new SDC has been created and guarantee it.
2. We only need to improve the mode we listed in 1.2 section to make both RX and TX work on full cycle, which will be enough for high speed TX and RX.
3. RTL change is straight forward, just added control bit to make RX can run on same clock as TX, and added control bit to bypass the negedge regsiters in TX to make it run on full cycle also, and some other logic to provide some flexibilities without affecting any old logic.
4. HW cost can be ignored, just some combination logic added.

### 8.5.2 Half cycle timing constraint

Master mode:

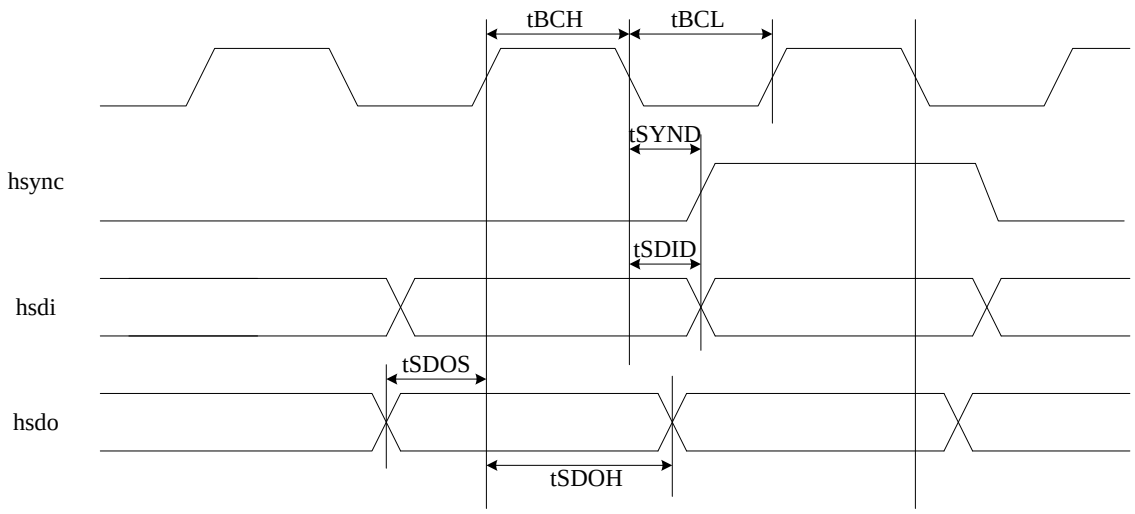



**Figure 8.5-2 Master Mode Half Cycle Timing**

| Parameter | Symbol | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| BCLK high pulse width | $t_{BCH}$ (1) | 20.3 | - | - | ns |
| BCLK low pulse width | $t_{BCL}$ (1) | 20.3 | - | - | ns |
| Hsync output to BCLK delay | $t_{SYND}$ (2) | - | - | 12.3 | ns |
| Hsdi output to BCLK delay | $t_{SDID}$ (2) | - | - | 12.3 | ns |
| Hsdo input setup time | $t_{SDOS}$ (3) | 8 | - | - | ns |
| Hsdo input hold time | $t_{SDOH}$ (4) | - | - | - | ns |

Note:

(1) BCLK high pulse width and low pulse width is from frequency 24.596Mhz
(2) Because output data on negedge and capture on rise edge so no need to constraint min delay.
(3) 8 ns setup is from 0.2 * cycle.
(4) Because hsdo is from negedge of clock so no hold timing requirement

Slave mod

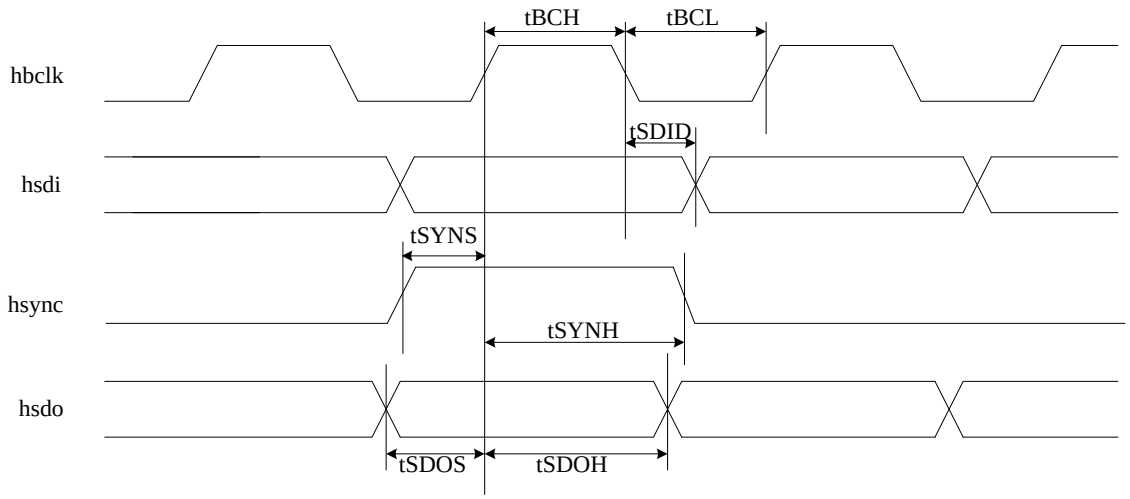



**Figure 8.5-3 Slave Mode Half Cycle Timing**

| Parameter | Symbol | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| BCLK high pulse width | tBCH (1) | 20 | - | - | ns |
| BCLK low pulse width | tBCL (1) | 20 | - | - | ns |
| Hsdi output to BCLK delay | tSDID (2) | - | - | 12 | ns |
| Hsync input setup time | tSYNS (3) | 8 | - | - | ns |
| Hsync input hold time | tSYNH (4) | - | - | - | ns |
| Hsdo input setup time | tSDOS (3) | 8 | - | - | ns |
| Hsdo input hold time | tSDOH (4) | - | - | - | ns |

Note:

(1) BCLK high pulse width and low pulse width is from frequency 25Mhz

(2) Because Solana TX can't meet half cycle timing so we may have -1~-3ns slack, I think it's OK because we will close timing on full-cycle design for higher frequency.

(3) 8 ns setup is from 0.2 * Cycle.

(4) Because hsync and hsdo is from negedge of clock so no hold timing requirement.

## 8.5.3 Half cycle timing numbers from timing report

**I2S Timing, ½ Cycle Slave Mode**

| | Min | Typ | Max | Condition |
|---|---|---|---|---|
| ½ Clock period | 20 | | | Max clock speed @ 24.576Mhz clock |
| Input setup time | 8 | | | Hsdo input setup |
| Input Hold time | - | - | - | Since ½ clock timing hold is not an issue. |
| Output Data valid | - | | 12 | Hsdi output to Bclk delay |

DeepEddy has -1.7 Violation (Solana had -1ns violation)

**I2S Timing, ½ Cycle Master Mode**

| | Min | Typ | Max | Condition |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| ½ Clock period | 20 | | | Max clock speed @ 24.576Mhz clock |
| Input setup time | 8 | | | Hsdo input setup |
| Input Hold Time | - | - | - | Since ½ clock timing hold not an issue |
| Output Data valid | - | - | 12 | Hsdi output to Bclk |

No Violation.

## 8.6 Register Spec

Detail Register Spec refer to link below.

I2SPCM Registers

# 9 I2S_PCM Peripheral Interfaces

## 9.1 General description

The peripheral I2S bus transfers audio samples between a host device and the peripheral devices (typically a Falcon or another smart amplifier, but may also be a complete CODEC). There are two peripheral I2S instances to simultaneously support two peripheral devices.
Up to six audio streams (three stereo pairs) are supported in the ADC direction (peripheral device to host) for each peripheral I2S instance.
Up to two audio streams (one stereo pair) are supported in the DAC direction (host to peripheral device) for each peripheral I2S instance.
The CNXT I2S IP V1.0 is used for the two peripheral I2S instances. The detailed features of CNXT I2S IP are listed as follow:

- Supports Philips/Sony/Right Justified/Intel/PCM/TDM mode.
- Supports master and slave mode.
- Supports sharing of the SCK signal between transmitter and receiver.
- Supports sharing of the WS signal between transmitter and receiver.
- Supports programmable assertion duration and frame length for WS signal in master mode.
- Supports programmable polarity of first frame for WS signal in master mode.
- Supports programmable sample size.
- Supports minimum slot of 1 cycle.
- Supports either SCK edge for SD signal.
- Supports mute of data.
- Supports programmable 'left then right' or 'right then left' order.
- Supports programmable bit order, MSB or LSB.
- Supports programmable polarity for WS signal.
- Supports any assignment for transmitter/receiver channels.
- Supports 48KHz / 96KHz / 192KHz sample rate.

The detailed specification of the peripheral I2S interface block may be found in the CNXT I2S IP V1.0 document.

## 9.2 Macro Definition

the Macro definition for Oceanside is listed as follows.

| Name | Value | Description |
|---|---|---|
| **WDT_WS_RX** | 8 | Sets bit width of WS related registers for receiver. For example, if the maximum frame length of WS is 512, this macro should be set to 9 ($2^9$ = 512, which is just equal or greater than 512). |
| **WDT_SA_RX** | 5 | Set bit width of sample size related registers for receiver. For example, if the maximum sample size is 24 bits, this macro should be set to 5 ($2^5$ = 32, which is just equal or greater than 24). |
| **WDT_DATA_RX** | 24 | Sets bit width of parallel data for receiver. |
| **WDT_SEL_RX** | 1 | Sets bit width of SD multiplexer for receiver. For example, if the amount of RX SD signal is 3, this macro should be set to 2 ($2^2$ = 4, which is just equal or greater than 3). |
| **NUM_CHNL_RX** | 6 | Sets number of channels for receiver. |
| **NUM_LINE_RX** | 1 | Sets number of serial lines for receiver. |
| **WDT_WS_TX** | 8 | Sets bit width of WS related registers for transmitter. For example, if the maximum frame length of WS is 512, this macro should be set to 9 ($2^9$ = 512, which is just equal or greater than 512). |
| **WDT_SA_TX** | 5 | Set bit width of sample size related registers for transmitter. For example, if the maximum sample size is 24 bits, this macro should be set to 5 ($2^5$ = 32, which is just equal or greater than 24). |
| **WDT_DATA_TX** | 24 | Sets bit width of parallel data for transmitter. |
| **NUM_CHNL_TX** | 2 | Sets number of channels for transmitter. |
| **NUM_LINE_TX** | 1 | Sets number of serial lines for transmitter. |

## 9.3 Registers

The peripheral I2S register definition for Oceanside is listed as follows.

**Name: PERIPHERAL_I2S_A_REG1**
**HDA Address: 0x01be_0000**
**I2C Address: 0x6f00**
**Default: 0x3f**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| 7:0 | rxtx_clk_frames | RW | RO | 8'h3f | Sets the frame length of WS signal for RX/TX in terms of bit clock cycles.<br>8'd0: 1 clock<br>8'd1: 2 clocks<br>etc…<br>This field is effective only in master mode.<br>This field is effective only after 'rxtx_lrck_en' is activated. |

**Name: PERIPHERAL_I2S_A_REG2**
**HDA Address: 0x01be_0100**
**I2C Address: 0x6f01**
**Default: 0x1f**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | rxtx_sync_lng | RW | RO | 8'h1f | Sets the high phase width of WS signal for RX/TX in terms of bit clock cycles.<br>8'd0: 1 clock<br>8'd1: 2 clocks<br>etc…<br>This field is effective only in master mode.<br>This field is effective only after 'rxtx_lrck_en' is activated. |

**Name: PERIPHERAL_I2S_A_REG3**
**HDA Address: 0x01be_0200**
**I2C Address: 0x6f02**
**Default: 0x32**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 5 | rxtx_on_negedge | RW | RO | 1'h1 | Sets the clock edge for RX/TX data, and generating WS signal.<br>1'b0: running on positive edge of BCLK<br>1'b1: running on negative edge of BCLK<br>This field is applied to reversing BCLK in clock generation block. |
| 4 | i2s_master | RW | RO | 1'h1 | Sets if running in master or slave mode for all the RX/TX channels.<br>1'b0: running as slave<br>1'b1: running as master<br>This field is applied to selection of BCLK and WS signal in clock generation block. |
| 1 | rxtx_lrck_en | RW | RO | 1'h1 | Enables to generate WS signal for RX/TX in master mode.<br>Active high. |

| | | | | | |
|---|---|---|---|---|---|
| 0 | rxtx_lrck_low_first | RW | RO | 1’h0 | This field can determine if a high-level pulse or low level pulse is generated firstly for RX/TX. This field provided a flexibility for the first frame.<br>1'b0: a high-level pulse come out firstly<br>1'b1: a low-level pulse come out firstly<br>This field is effective only after 'rxtx_lrck_en' is activated. |

**Name: PERIPHERAL_I2S_A_REG4**
**HDA Address: 0x01be_0300**
**I2C Address: 0x6f03**
**Default: 0x44**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 6 | rx_lrck_pol | RW | RO | 1’h1 | The polarity of WS signal for receiver can be reversed or not based on different mode.<br>1'b0: keep WS polarity<br>1'b1: reverse WS polarity |
| 4 | rx_order | RW | RO | 1’h0 | Sets which channel is received firstly for the first frame.<br>1'b0: 'left then right' order<br>1'b1: 'right then left' order<br>This field is effective only after 'rxN_en' is activated. (N=1, 2, …, depending on which channel it is.) |
| 3 | rx_dshift_sel | RW | RO | 1’h0 | Sets if little-endian or big-endian is active.<br>1'b0: MSB first on RX<br>1'b1: LSB first on RX<br>This field is effective only after 'rxN_en' is activated. (N=1, 2, …, depending on which channel it is.) |
| 2 | rx_dstart_dly | RW | RO | 1’h1 | Sets 1-bit delay for RX channels.<br>1'b0: no delay<br>1'b1: delays one clock cycle<br>This field is only for standard I2S (Philips) mode.<br>This field is effective only after 'rxN_en' is activated. (N=1, 2, …, depending on which channel it is.) |
| 0 | rx_mode | RW | RO | 1’h0 | I2S mode and PCM mode have different mechanism with WS signal. The I2S mode is based on the edge detect of WS signal, while the PCM mode is based on a slot number.<br>1'b0: I2S mode<br>1'b1: PCM mode<br>This field is effective only after 'rxN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |

**Name: PERIPHERAL_I2S_A_REG5**
**HDA Address: 0x01be_0400**
**I2C Address: 0x6f04**
**Default: 0x17**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 4:0 | rx_sa_wdt | RW | RO | 5’h17 | Sets the sample data width of transmitter for all the RX channels.<br>5'd0: 1-bit data<br>5'd1: 2-bit data<br>etc. …<br>This field is effective only after 'rxN_en' is activated. (N=1, 2, …, depending on which channel it is.) |

**Name: PERIPHERAL_I2S_A_REG7**
**HDA Address: 0x01be_0600**
**I2C Address: 0x6f06**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | rx1_slot | RW | RO | 8’h0 | When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel1.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel1.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc. …<br>This field is effective only after 'rx1_en' is activated. |

**Name: PERIPHERAL_I2S_A_REG8**
**HDA Address: 0x01be_0700**
**I2C Address: 0x6f07**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | rx2_slot | RW | RO | 8’h0 | When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel2.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel2.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc. …<br>This field is effective only after 'rx2_en' is activated. |

**Name: PERIPHERAL_I2S_A_REG9**
**HDA Address: 0x01be_0800**
**I2C Address: 0x6f08**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| 7:0 | rx3_slot | RW | RO | 8’h0 | When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel3.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel3.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc. …<br>This field is effective only after 'rx3_en' is activated. |
| **Name: PERIPHERAL_I2S_A_REG10**<br>**HDA Address: 0x01be_0900**<br>**I2C Address: 0x6f09**<br>**Default: 0x0** | | | | | |
| **bits** | **name** | **s/w** | **h/w** | **default** | **description** |
| 7:0 | rx4_slot | RW | RO | 8’h0 | When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel4.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel4.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc. …<br>This field is effective only after 'rx4_en' is activated. |
| **Name: PERIPHERAL_I2S_A_REG11**<br>**HDA Address: 0x01be_0a00**<br>**I2C Address: 0x6f0a**<br>**Default: 0x0** | | | | | |
| **bits** | **name** | **s/w** | **h/w** | **default** | **description** |
| 7:0 | rx5_slot | RW | RO | 8’h0 | When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel5.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel5.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc. …<br>This field is effective only after 'rx5_en' is activated. |
| **Name: PERIPHERAL_I2S_A_REG12**<br>**HDA Address: 0x01be_0b00**<br>**I2C Address: 0x6f0b**<br>**Default: 0x0** | | | | | |
| **bits** | **name** | **s/w** | **h/w** | **default** | **description** |

| | | | | | |
|---|---|---|---|---|---|
| 7:0 | rx6_slot | RW | RO | 8’h0 | When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel6.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel6.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc. …<br>This field is effective only after 'rx6_en' is activated. |

**Name: PERIPHERAL_I2S_A_REG13**
**HDA Address: 0x01be_0c00**
**I2C Address: 0x6f0c**
**Default: 0x3**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 6 | rx_mute | RW | RO | 1’h0 | Mute the I2S or PCM Receiver (all zeros will be send). |
| 5 | rx6_en | RW | RO | 1’h0 | Enables to receive data for RX channel6.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel6 must be steady before rx6_en is active. And rx6_en must be inactive before reconfiguration. |
| 4 | rx5_en | RW | RO | 1’h0 | Enables to receive data for RX channel5.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel5 must be steady before rx5_en is active. And rx5_en must be inactive before reconfiguration. |
| 3 | rx4_en | RW | RO | 1’h0 | Enables to receive data for RX channel4.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel4 must be steady before rx4_en is active. And rx4_en must be inactive before reconfiguration. |
| 2 | rx3_en | RW | RO | 1’h0 | Enables to receive data for RX channel3.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel3 must be steady before rx3_en is active. And rx3_en must be inactive before reconfiguration. |
| 1 | rx2_en | RW | RO | 1’h1 | Enables to receive data for RX channel2.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel2 must be steady before rx2_en is active. And rx2_en must be inactive before reconfiguration. |

| | | | | | |
|---|---|---|---|---|---|
| 0 | rx1_en | RW | RO | 1'h1 | Enables to receive data for RX channel1.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel1 must be steady before rx1_en is active. And rx1_en must be inactive before reconfiguration. |
| **Name: PERIPHERAL_I2S_A_REG14**<br>**HDA Address: 0x01be_0d00**<br>**I2C Address: 0x6f0d**<br>**Default: 0x44** | | | | | |
| **bits** | **name** | **s/w** | **h/w** | **default** | **description** |
| 6 | tx_lrck_pol | RW | RO | 1'h1 | The polarity of WS signal for transmitter can be reversed or not based on different mode.<br>1'b0: keep WS polarity<br>1'b1: reverse WS polarity |
| 4 | tx_order | RW | RO | 1'h0 | Sets which channel is transmitted firstly for the first frame.<br>1'b0: 'left then right' order<br>1'b1: 'right then left' order<br>This field is effective only after 'txN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |
| 3 | tx_dshift_sel | RW | RO | 1'h0 | Sets if little-endian or big-endian is active.<br>1'b0: MSB first on TX<br>1'b1: LSB first on TX<br>This field is effective only after 'txN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |
| 2 | tx_dstart_dly | RW | RO | 1'h1 | Sets 1-bit delay for TX channels.<br>1'b0: no delay<br>1'b1: delays one clock cycle<br>This field is only for standard I2S (Philips) mode.<br>This field is effective only after 'txN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |
| 0 | tx_mode | RW | RO | 1'h0 | I2S mode and PCM mode have different mechanism with WS signal. The I2S mode is based on the edge detect of WS signal, while the PCM mode is based on a slot number.<br>1'b0: I2S mode<br>1'b1: PCM mode<br>This field is effective only after 'txN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |
| **Name: PERIPHERAL_I2S_A_REG15**<br>**HDA Address: 0x01be_0e00**<br>**I2C Address: 0x6f0e**<br>**Default: 0x17** | | | | | |
| **bits** | **name** | **s/w** | **h/w** | **default** | **description** |

| | | | | | |
|---|---|---|---|---|---|
| 5 | tx_out_en | RW | RO | 1’h0 | The data output signal will be tri-stated only under TDM mode.<br>1'b0: the out data is not tri-stated when not driven<br>1'b1: the out data is tri-stated when not driven |
| 4:0 | tx_sa_wdt | RW | RO | 5’h17 | Sets the sample data width of transmitter for TX channel1.<br>5'd0: 1-bit data<br>5'd1: 2-bit data<br>etc. …<br>This field is effective only after 'tx1_en' is activated. |

**Name: PERIPHERAL_I2S_A_REG17**
**HDA Address: 0x01be_1000**
**I2C Address: 0x6f10**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | tx1_slot | RW | RO | 8’h0 | When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for TX channel1.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for TX channel1.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc. …<br>This field is effective only after 'tx1_en' is activated. |

**Name: PERIPHERAL_I2S_A_REG18**
**HDA Address: 0x01be_1100**
**I2C Address: 0x6f11**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | tx2_slot | RW | RO | 8’h0 | When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for TX channel2.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for TX channel2.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc. …<br>This field is effective only after 'tx2_en' is activated. |

**Name: PERIPHERAL_I2S_A_REG19**
**HDA Address: 0x01be_1200**
**I2C Address: 0x6f12**
**Default: 0x3**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| 2 | tx_mute | RW | RO | 1'h0 | Sets to mute all the TX channels (all zeros will be send). |
| 1 | tx2_en | RW | RO | 1'h1 | Enables to transmit data for TX channel2.<br>Active high.<br>[Notice] All of TX configuration fields related with TX channel2 must be steady before tx2_en is active. And tx2_en must be inactive before reconfiguration. |
| 0 | tx1_en | RW | RO | 1'h1 | Enables to transmit data for TX channel1.<br>Active high.<br>[Notice] All of TX configuration fields related with TX channel1 must be steady before tx1_en is active. And tx1_en must be inactive before reconfiguration. |

**Name: PERIPHERAL_I2S_A_REG20**
**HDA Address: 0x01be_1300**
**I2C Address: 0x6f13**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:6 | DRVCTL_dout | RW | RO | 2'h0 | Establishes the output drive strength for data out of peripheral I2S #1. Encoded as:<br>00: 1.8mA<br>01: 3.6mA<br>10: 5.4mA<br>11: 7.2mA |
| 5:4 | DRVCTL_lrclk | RW | RO | 2'h0 | Establishes the output drive strength for LRCK of peripheral I2S #1. Encoded as:<br>00: 1.8mA<br>01: 3.6mA<br>10: 5.4mA<br>11: 7.2mA |
| 3:2 | DRVCTL_bclk | RW | RO | 2'h0 | Establishes the output drive strength for bit clock of peripheral I2S #1. Encoded as:<br>00: 1.8mA<br>01: 3.6mA<br>10: 5.4mA<br>11: 7.2mA |
| 1:0 | DRVCTL_mclk | RW | RO | 2'h0 | Establishes the output drive strength for master clock of peripheral I2S #1. Encoded as:<br>00: 1.8mA<br>01: 3.6mA<br>10: 5.4mA<br>11: 7.2mA |

**Name: PERIPHERAL_I2S_B_REG1**
**HDA Address: 0x01be_1400**
**I2C Address: 0x6f14**
**Default: 0x3f**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | rxtx_clk_frames | RW | RO | 8’h3f | Sets the frame length of WS signal for RX/TX in terms of bit clock cycles.<br>8'd0: 1 clock<br>8'd1: 2 clocks<br>etc. …<br>This field is effective only in master mode.<br>This field is effective only after 'rxtx_lrck_en' is activated. |

**Name: PERIPHERAL_I2S_B_REG2**
**HDA Address: 0x01be_1500**
**I2C Address: 0x6f15**
**Default: 0x1f**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | rxtx_sync_lng | RW | RO | 8’h1f | Sets the high phase width of WS signal for RX/TX in terms of bit clock cycles.<br>8'd0: 1 clock<br>8'd1: 2 clocks<br>etc. …<br>This field is effective only in master mode.<br>This field is effective only after 'rxtx_lrck_en' is activated. |

**Name: PERIPHERAL_I2S_B_REG3**
**HDA Address: 0x01be_1600**
**I2C Address: 0x6f16**
**Default: 0x32**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 5 | rxtx_on_negedge | RW | RO | 1’h1 | Sets the clock edge for RX/TX data, and generating WS signal.<br>1'b0: running on positive edge of BCLK<br>1'b1: running on negative edge of BCLK<br>This field is applied to reversing BCLK in clock generation block. |
| 4 | i2s_master | RW | RO | 1’h1 | Sets if running in master or slave mode for all the RX/TX channels.<br>1'b0: running as slave<br>1'b1: running as master<br>This field is applied to selection of BCLK and WS signal in clock generation block. |
| 1 | rxtx_lrck_en | RW | RO | 1’h1 | Enables to generate WS signal for RX/TX in master mode.<br>Active high. |

| | | | | | |
|---|---|---|---|---|---|
| 0 | rxtx_lrck_low_first | RW | RO | 1’h0 | This field can determine if a high-level pulse or low level pulse is generated firstly for RX/TX. This field provided a flexibility for the first frame.<br>1'b0: a high-level pulse come out firstly<br>1'b1: a low-level pulse come out firstly<br>This field is effective only after 'rxtx_lrck_en' is activated. |

**Name: PERIPHERAL_I2S_B_REG4**
**HDA Address: 0x01be_1700**
**I2C Address: 0x6f17**
**Default: 0x44**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 6 | rx_lrck_pol | RW | RO | 1’h1 | The polarity of WS signal for receiver can be reversed or not based on different mode.<br>1'b0: keep WS polarity<br>1'b1: reverse WS polarity |
| 4 | rx_order | RW | RO | 1’h0 | Sets which channel is received firstly for the first frame.<br>1'b0: 'left then right' order<br>1'b1: 'right then left' order<br>This field is effective only after 'rxN_en' is activated. (N=1, 2, …, depending on which channel it is.) |
| 3 | rx_dshift_sel | RW | RO | 1’h0 | Sets if little-endian or big-endian is active.<br>1'b0: MSB first on RX<br>1'b1: LSB first on RX<br>This field is effective only after 'rxN_en' is activated. (N=1, 2, …, depending on which channel it is.) |
| 2 | rx_dstart_dly | RW | RO | 1’h1 | Sets 1-bit delay for RX channels.<br>1'b0: no delay<br>1'b1: delays one clock cycle<br>This field is only for standard I2S (Philips) mode.<br>This field is effective only after 'rxN_en' is activated. (N=1, 2, …, depending on which channel it is.) |
| 0 | rx_mode | RW | RO | 1’h0 | I2S mode and PCM mode have different mechanism with WS signal. The I2S mode is based on the edge detect of WS signal, while the PCM mode is based on a slot number.<br>1'b0: I2S mode<br>1'b1: PCM mode<br>This field is effective only after 'rxN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |

**Name: PERIPHERAL_I2S_B_REG5**
**HDA Address: 0x01be_1800**
**I2C Address: 0x6f18**
**Default: 0x17**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 4:0 | rx_sa_wdt | RW | RO | 5’h17 | Sets the sample data width of transmitter for all the RX channels.<br>5'd0: 1-bit data<br>5'd1: 2-bit data<br>etc. …<br>This field is effective only after 'rxN_en' is activated. (N=1, 2, …, depending on which channel it is.) |

**Name: PERIPHERAL_I2S_B_REG7**
**HDA Address: 0x01be_1a00**
**I2C Address: 0x6f1a**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | rx1_slot | RW | RO | 8’h0 | When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel1.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel1.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc. …<br>This field is effective only after 'rx1_en' is activated. |

**Name: PERIPHERAL_I2S_B_REG8**
**HDA Address: 0x01be_1b00**
**I2C Address: 0x6f1b**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | rx2_slot | RW | RO | 8’h0 | When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel2.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel2.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc. …<br>This field is effective only after 'rx2_en' is activated. |

**Name: PERIPHERAL_I2S_B_REG9**
**HDA Address: 0x01be_1c00**
**I2C Address: 0x6f1c**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| 7:0 | rx3_slot | RW | RO | 8’h0 | When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel3.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel3.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc. …<br>This field is effective only after 'rx3_en' is activated. |
| **Name: PERIPHERAL_I2S_B_REG10**<br>**HDA Address: 0x01be_1d00**<br>**I2C Address: 0x6f1d**<br>**Default: 0x0** | | | | | |
| **bits** | **name** | **s/w** | **h/w** | **default** | **description** |
| 7:0 | rx4_slot | RW | RO | 8’h0 | When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel4.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel4.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc. …<br>This field is effective only after 'rx4_en' is activated. |
| **Name: PERIPHERAL_I2S_B_REG11**<br>**HDA Address: 0x01be_1e00**<br>**I2C Address: 0x6f1e**<br>**Default: 0x0** | | | | | |
| **bits** | **name** | **s/w** | **h/w** | **default** | **description** |
| 7:0 | rx5_slot | RW | RO | 8’h0 | When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel5.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel5.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc. …<br>This field is effective only after 'rx5_en' is activated. |
| **Name: PERIPHERAL_I2S_B_REG12**<br>**HDA Address: 0x01be_1f00**<br>**I2C Address: 0x6f1f**<br>**Default: 0x0** | | | | | |
| **bits** | **name** | **s/w** | **h/w** | **default** | **description** |

| | | | | | |
|---|---|---|---|---|---|
| 7:0 | rx6_slot | RW | RO | 8’h0 | When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel6.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel6.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc. …<br>This field is effective only after 'rx6_en' is activated. |

**Name: PERIPHERAL_I2S_B_REG13**
**HDA Address: 0x01be_2000**
**I2C Address: 0x6f20**
**Default: 0x3**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 6 | rx_mute | RW | RO | 1’h0 | Mute the I2S or PCM Receiver (all zeros will be send). |
| 5 | rx6_en | RW | RO | 1’h0 | Enables to receive data for RX channel6.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel6 must be steady before rx6_en is active. And rx6_en must be inactive before reconfiguration. |
| 4 | rx5_en | RW | RO | 1’h0 | Enables to receive data for RX channel5.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel5 must be steady before rx5_en is active. And rx5_en must be inactive before reconfiguration. |
| 3 | rx4_en | RW | RO | 1’h0 | Enables to receive data for RX channel4.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel4 must be steady before rx4_en is active. And rx4_en must be inactive before reconfiguration. |
| 2 | rx3_en | RW | RO | 1’h0 | Enables to receive data for RX channel3.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel3 must be steady before rx3_en is active. And rx3_en must be inactive before reconfiguration. |
| 1 | rx2_en | RW | RO | 1’h1 | Enables to receive data for RX channel2.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel2 must be steady before rx2_en is active. And rx2_en must be inactive before reconfiguration. |

| | | | | | |
|---|---|---|---|---|---|
| 0 | rx1_en | RW | RO | 1'h1 | Enables to receive data for RX channel1.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel1 must be steady before rx1_en is active. And rx1_en must be inactive before reconfiguration. |
| **Name: PERIPHERAL_I2S_B_REG14**<br>**HDA Address: 0x01be_2100**<br>**I2C Address: 0x6f21**<br>**Default: 0x44** | | | | | |
| **bits** | **name** | **s/w** | **h/w** | **default** | **description** |
| 6 | tx_lrck_pol | RW | RO | 1'h1 | The polarity of WS signal for transmitter can be reversed or not based on different mode.<br>1'b0: keep WS polarity<br>1'b1: reverse WS polarity |
| 4 | tx_order | RW | RO | 1'h0 | Sets which channel is transmitted firstly for the first frame.<br>1'b0: 'left then right' order<br>1'b1: 'right then left' order<br>This field is effective only after 'txN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |
| 3 | tx_dshift_sel | RW | RO | 1'h0 | Sets if little-endian or big-endian is active.<br>1'b0: MSB first on TX<br>1'b1: LSB first on TX<br>This field is effective only after 'txN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |
| 2 | tx_dstart_dly | RW | RO | 1'h1 | Sets 1-bit delay for TX channels.<br>1'b0: no delay<br>1'b1: delays one clock cycle<br>This field is only for standard I2S (Philips) mode.<br>This field is effective only after 'txN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |
| 0 | tx_mode | RW | RO | 1'h0 | I2S mode and PCM mode have different mechanism with WS signal. The I2S mode is based on the edge detect of WS signal, while the PCM mode is based on a slot number.<br>1'b0: I2S mode<br>1'b1: PCM mode<br>This field is effective only after 'txN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |
| **Name: PERIPHERAL_I2S_B_REG15**<br>**HDA Address: 0x01be_2200**<br>**I2C Address: 0x6f22**<br>**Default: 0x17** | | | | | |
| **bits** | **name** | **s/w** | **h/w** | **default** | **description** |

| | | | | | |
|---|---|---|---|---|---|
| 5 | tx_out_en | RW | RO | 1’h0 | The data output signal will be tri-stated only under TDM mode.<br>1'b0: the out data is not tri-stated when not driven<br>1'b1: the out data is tri-stated when not driven |
| 4:0 | tx_sa_wdt | RW | RO | 5’h17 | Sets the sample data width of transmitter for TX channel1.<br>5'd0: 1-bit data<br>5'd1: 2-bit data<br>etc. …<br>This field is effective only after 'tx1_en' is activated. |

**Name: PERIPHERAL_I2S_B_REG17**
**HDA Address: 0x01be_2400**
**I2C Address: 0x6f24**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | tx1_slot | RW | RO | 8’h0 | When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for TX channel1.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for TX channel1.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc. …<br>This field is effective only after 'tx1_en' is activated. |

**Name: PERIPHERAL_I2S_B_REG18**
**HDA Address: 0x01be_2500**
**I2C Address: 0x6f25**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | tx2_slot | RW | RO | 8’h0 | When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for TX channel2.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for TX channel2.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc. …<br>This field is effective only after 'tx2_en' is activated. |

**Name: PERIPHERAL_I2S_B_REG19**
**HDA Address: 0x01be_2600**
**I2C Address: 0x6f26**
**Default: 0x3**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| 2 | tx_mute | RW | RO | 1'h0 | Sets to mute all the TX channels (all zeros will be send). |
| 1 | tx2_en | RW | RO | 1'h1 | Enables to transmit data for TX channel2.<br>Active high.<br>[Notice] All of TX configuration fields related with TX channel2 must be steady before tx2_en is active. And tx2_en must be inactive before reconfiguration. |
| 0 | tx1_en | RW | RO | 1'h1 | Enables to transmit data for TX channel1.<br>Active high.<br>[Notice] All of TX configuration fields related with TX channel1 must be steady before tx1_en is active. And tx1_en must be inactive before reconfiguration. |

**Name: PERIPHERAL_I2S_B_REG20**
**HDA Address: 0x01be_2700**
**I2C Address: 0x6f27**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:6 | DRVCTL_dout | RW | RO | 2'h0 | Establishes the output drive strength for data out of peripheral I2S #2. Encoded as:<br>00: 1.8mA<br>01: 3.6mA<br>10: 5.4mA<br>11: 7.2mA |
| 5:4 | DRVCTL_lrclk | RW | RO | 2'h0 | Establishes the output drive strength for LRCK of peripheral I2S #2. Encoded as:<br>00: 1.8mA<br>01: 3.6mA<br>10: 5.4mA<br>11: 7.2mA |
| 3:2 | DRVCTL_bclk | RW | RO | 2'h0 | Establishes the output drive strength for bit clock of peripheral I2S #2. Encoded as:<br>00: 1.8mA<br>01: 3.6mA<br>10: 5.4mA<br>11: 7.2mA |
| 1:0 | DRVCTL_mclk | RW | RO | 2'h0 | Establishes the output drive strength for master clock of peripheral I2S #2. Encoded as:<br>00: 1.8mA<br>01: 3.6mA<br>10: 5.4mA<br>11: 7.2mA |

**Name: PERIPHERAL_I2S_A_REG21**
**HDA Address: 0x01be_2800**
**I2C Address: 0x6f28**
**Default: 0xaf**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7 | mst_clk_div_en_a | RW | RO | 1'h1 | Enables the clock divider for bit clock of peripheral I2S_A. |
| 6:0 | mst_clk_div_a | RW | RO | 7'h2f | Sets the division ratio for bit clock of peripheral I2S_A.<br>7'd0: div by 1<br>7'd1: div by 2<br>7'd2: div by 3<br>Etc. … |

**Name: PERIPHERAL_I2S_A_REG22**
**HDA Address: 0x01be_2900**
**I2C Address: 0x6f29**
**Default: 0x8b**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7 | mclk_div_en_a | RW | RO | 1'h1 | Enables the clock divider for master clock of peripheral I2S_A. |
| 6:0 | mclk_div_a | RW | RO | 7'hb | Sets the division ratio for master clock of peripheral I2S_A.<br>7'd0: div by 1<br>7'd1: div by 2<br>7'd2: div by 3<br>Etc. … |

**Name: PERIPHERAL_I2S_A_REG23**
**HDA Address: 0x01be_2a00**
**I2C Address: 0x6f2a**
**Default: 0x55**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:6 | DCCTL_i2s1_dout | RW | RO | 2'h1 | Enables built-in pull-up or pull-down functionality for data out of peripheral I2S_A. Encoded as:<br>00: pull-up and pull-down disabled<br>01: weak pull-down enabled, pull-up disabled<br>10: pull-down disabled, weak pull-down value 1 enabled<br>11: pull-down disabled, weak pull-down value 2 enabled. |
| 5:4 | DCCTL_i2s1_lrclk | RW | RO | 2'h1 | Enables built-in pull-up or pull-down functionality for LRCK of peripheral I2S_A. Encoded as:<br>00: pull-up and pull-down disabled<br>01: weak pull-down enabled, pull-up disabled<br>10: pull-down disabled, weak pull-down value 1 enabled<br>11: pull-down disabled, weak pull-down value 2 enabled. |
| 3:2 | DCCTL_i2s1_bclk | RW | RO | 2'h1 | Enables built-in pull-up or pull-down functionality for bit clock of peripheral I2S_A. Encoded as:<br>00: pull-up and pull-down disabled<br>01: weak pull-down enabled, pull-up disabled |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | 10: pull-down disabled, weak pull-down value 1 enabled<br>11: pull-down disabled, weak pull-down value 2 enabled. |
| 1:0 | DCCTL_i2s1_mclk | RW | RO | 2'h1 | Enables built-in pull-up or pull-down functionality for master clock of peripheral I2S_A. Encoded as:<br>00: pull-up and pull-down disabled<br>01: weak pull-down enabled, pull-up disabled<br>10: pull-down disabled, weak pull-down value 1 enabled<br>11: pull-down disabled, weak pull-down value 2 enabled. |

**Name: PERIPHERAL_I2S_A_REG24**
**HDA Address: 0x01be_2b00**
**I2C Address: 0x6f2b**
**Default: 0x1**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 1:0 | DCCTL_i2s1_din | RW | RO | 2'h1 | Enables built-in pull-up or pull-down functionality for data in of peripheral I2S_A. Encoded as:<br>00: pull-up and pull-down disabled<br>01: weak pull-down enabled, pull-up disabled<br>10: pull-down disabled, weak pull-down value 1 enabled<br>11: pull-down disabled, weak pull-down value 2 enabled. |

**Name: PERIPHERAL_I2S_A_REG25**
**HDA Address: 0x01be_2c00**
**I2C Address: 0x6f2c**
**Default: 0x7**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 3 | OEN_I2S1_mclk | RW | RO | 1'h0 | Output enable signal for master clock of peripheral I2S_A. |
| 2 | mclk_en_a | RW | RO | 1'h1 | Master clock enable signal for peripheral I2S_A. |
| 1 | mst_clk_en_a | RW | RO | 1'h1 | Bit clock enable signal in master mode for peripheral I2S_A. |
| 0 | bclk_en_a | RW | RO | 1'h1 | Bit clock enable signal for peripheral I2S_A. |

**Name: PERIPHERAL_I2S_B_REG21**
**HDA Address: 0x01be_2d00**
**I2C Address: 0x6f2d**
**Default: 0xaf**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7 | mst_clk_div_en_b | RW | RO | 1'h1 | Enables the clock divider for bit clock of peripheral I2S_B. |
| 6:0 | mst_clk_div_b | RW | RO | 7'h2f | Sets the division ratio for bit clock of peripheral I2S_B.<br>7'd0: div by 1 |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | 7'd1: div by 2<br>7'd2: div by 3<br>Etc. … |
| **Name: PERIPHERAL_I2S_B_REG22**<br>**HDA Address: 0x01be_2e00**<br>**I2C Address: 0x6f2e**<br>**Default: 0x8b** | | | | | |
| **bits** | **name** | **s/w** | **h/w** | **default** | **description** |
| 7 | mclk_div_en_b | RW | RO | 1'h1 | Enables the clock divider for master clock of peripheral I2S_B. |
| 6:0 | mclk_div_b | RW | RO | 7'hb | Sets the division ratio for master clock of peripheral I2S_B.<br>7'd0: div by 1<br>7'd1: div by 2<br>7'd2: div by 3<br>Etc. … |
| **Name: PERIPHERAL_I2S_B_REG23**<br>**HDA Address: 0x01be_2f00**<br>**I2C Address: 0x6f2f**<br>**Default: 0x55** | | | | | |
| **bits** | **name** | **s/w** | **h/w** | **default** | **description** |
| 7:6 | DCCTL_i2s2_dout | RW | RO | 2'h1 | Enables built-in pull-up or pull-down functionality for data out of peripheral I2S_B. Encoded as:<br>00: pull-up and pull-down disabled<br>01: weak pull-down enabled, pull-up disabled<br>10: pull-down disabled, weak pull-down value 1 enabled<br>11: pull-down disabled, weak pull-down value 2 enabled. |
| 5:4 | DCCTL_i2s2_lrclk | RW | RO | 2'h1 | Enables built-in pull-up or pull-down functionality for LRCK of peripheral I2S_B. Encoded as:<br>00: pull-up and pull-down disabled<br>01: weak pull-down enabled, pull-up disabled<br>10: pull-down disabled, weak pull-down value 1 enabled<br>11: pull-down disabled, weak pull-down value 2 enabled. |
| 3:2 | DCCTL_i2s2_bclk | RW | RO | 2'h1 | Enables built-in pull-up or pull-down functionality for bit clock of peripheral I2S_B. Encoded as:<br>00: pull-up and pull-down disabled<br>01: weak pull-down enabled, pull-up disabled<br>10: pull-down disabled, weak pull-down value 1 enabled<br>11: pull-down disabled, weak pull-down value 2 enabled. |
| 1:0 | DCCTL_i2s2_mclk | RW | RO | 2'h1 | Enables built-in pull-up or pull-down functionality for master clock of peripheral I2S_B. Encoded as:<br>00: pull-up and pull-down disabled |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | 01: weak pull-down enabled, pull-up disabled<br>10: pull-down disabled, weak pull-down value 1 enabled<br>11: pull-down disabled, weak pull-down value 2 enabled. |
| **Name: PERIPHERAL_I2S_B_REG24**<br>**HDA Address: 0x01be_3000**<br>**I2C Address: 0x6f30**<br>**Default: 0x1** | | | | | |
| **bits** | **name** | **s/w** | **h/w** | **default** | **description** |
| 1:0 | DCCTL_i2s2_din | RW | RO | 2'h1 | Enables built-in pull-up or pull-down functionality for data in of peripheral I2S_B. Encoded as:<br>00: pull-up and pull-down disabled<br>01: weak pull-down enabled, pull-up disabled<br>10: pull-down disabled, weak pull-down value 1 enabled<br>11: pull-down disabled, weak pull-down value 2 enabled. |
| **Name: PERIPHERAL_I2S_B_REG25**<br>**HDA Address: 0x01be_3100**<br>**I2C Address: 0x6f31**<br>**Default: 0x7** | | | | | |
| **bits** | **name** | **s/w** | **h/w** | **default** | **description** |
| 3 | OEN_I2S2_mclk | RW | RO | 1'h0 | Output enable signal for master clock of peripheral I2S_B. |
| 2 | mclk_en_b | RW | RO | 1'h1 | Master clock enable signal for peripheral I2S_B. |
| 1 | mst_clk_en_b | RW | RO | 1'h1 | Bit clock enable signal in master mode for peripheral I2S_B. |
| 0 | bclk_en_b | RW | RO | 1'h1 | Bit clock enable signal for peripheral I2S_B. |
| **Name: PERIPHERAL_I2S_A_REG26**<br>**HDA Address: 0x01be_3200**<br>**I2C Address: 0x6f32**<br>**Default: 0x0** | | | | | |
| **bits** | **name** | **s/w** | **h/w** | **default** | **description** |
| 1 | pi2s_lb_mode_a | RW | RO | 1'h0 | Enables loop-back from I2S_A data out to I2S_A data in. |
| 0 | hda_sw_mode_a | RW | RO | 1'h0 | Enables software mode under HDA operation for I2S_A. This bit is used for the following fields:<br>'rx_sa_wdt_a'<br>'tx_sa_wdt_a'<br>'pi2s_a_mst_clk_div'<br>'i2s1_mclk_div' |
| **Name: PERIPHERAL_I2S_B_REG26**<br>**HDA Address: 0x01be_3300**<br>**I2C Address: 0x6f33**<br>**Default: 0x0** | | | | | |
| **bits** | **name** | **s/w** | **h/w** | **default** | **description** |

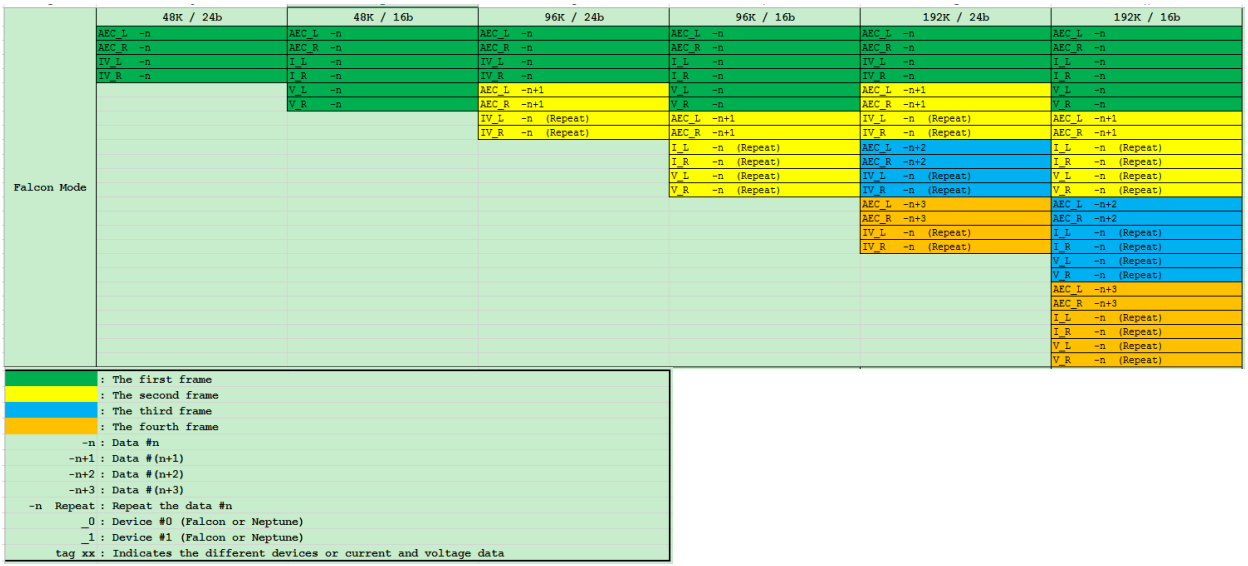
| | | | | | |
|---|---|---|---|---|---|
| 1 | pi2s_lb_mode_b | RW | RO | 1’h0 | Enables loop-back from I2S_B data out to I2S_B data in. |
| 0 | hda_sw_mode_b | RW | RO | 1’h0 | Enables software mode under HDA operation for I2S_B. This bit is used for the following fields:<br>‘rx_sa_wdt_b’<br>‘tx_sa_wdt_b’<br>‘pi2s_b_mst_clk_div’<br>‘i2s2_mclk_div’ |

## 9.4 IV Stream Packing

Supporting smart amplifiers connected to the peripheral interface requires the ability to pass current and voltage (I and V) measurement information from the amplifier to the host. Third party amplifiers may use the exposed input path for either AEC reference data or I/V sample data. Conexant amplifiers, however, have the option to use a special path that is not exposed to the OS for transporting IV data. This path provides the ability to translate between host and peripheral sample rates without loss of data and the ability to utilize fewer channels at higher sample rates (The IV sample rate is fixed at 48KHz in Neptune and Falcon). The exposed record path may be used for AEC data or other purposes.

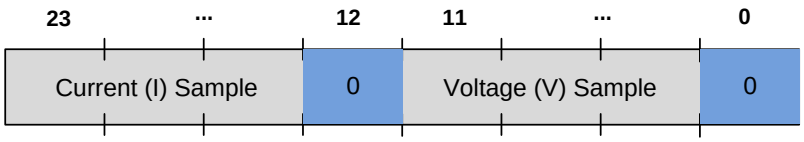
### 9.4.1 Supporting Falcon Format

Oceanside supports to parsing the Falcon format which is defined as follows.

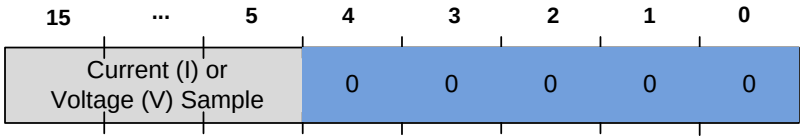



When the Falcon device is configured for 48Ksps operation, the SA2 I/V samples are produced at the same rate as the DAC and AEC audio samples. Therefore, the I/V samples are transported through the peripheral I2S interface at the same rate as the DAC and AEC audio samples.
When the Falcon device is configured for 96Ksps operation, the SA2 I/V samples are produced at half the rate of the DAC audio samples. In this case, each I/V sample is repeated a second time on the peripheral I2S interface.
When the Falcon device is configured for 192Ksps operation, the SA2 I/V samples are produced at a quarter the rate of the DAC audio samples. In this case, each I/V sample is repeated three additional times on the peripheral I2S interface.
The valid data for 24-bit or 16-bit mode is shown as follows.

**Falcon Format for 24 bits data**




**Falcon Format for 16 bits data**




## 9.4.2 Supporting Neptune Format

Oceanside supports parsing the Neptune format which is defined as follows.

| | 48K / 24b | 48K / 16b | 96K / 24b | 96K / 16b | 192K / 24b | 192K / 16b |
|---|---|---|---|---|---|---|
| Neptune Mode | AEC_L -n | AEC_L -n | AEC_L -n | AEC_L -n | AEC_L -n | AEC_L -n |
| | AEC_R -n | AEC_R -n | AEC_R -n | AEC_R -n | AEC_R -n | AEC_R -n |
| | IV_L -n | I_L -n | IV_L -n tag 1 | I_L -n tag 1 | IV_L -n tag 1 | I_L -n tag 1 |
| | IV_R -n | V_L -n | 0 | V_L -n tag 1 | 0 | 0 |
| | | I_R -n | AEC_L -n+1 | AEC_L -n+1 | AEC_L -n+1 | AEC_L -n+1 |
| | | V_R -n | AEC_R -n+1 | AEC_R -n+1 | AEC_R -n+1 | AEC_R -n+1 |
| | | | IV_R -n | I_R -n | IV_R -n | V_L -n |
| | | | 0 | V_R -n | 0 | 0 |
| | | | | | AEC_L -n+2 | AEC_L -n+2 |
| | | | | | AEC_R -n+2 | AEC_R -n+2 |
| | | | | | IV_L -n (Repeat) tag 1 | I_R -n |
| | | | | | 0 | 0 |
| | | | | | AEC_L -n+3 | AEC_L -n+3 |
| | | | | | AEC_R -n+3 | AEC_R -n+3 |
| | | | | | IV_R -n (Repeat) | V_R -n |
| | | | | | 0 | 0 |

: The first frame
: The second frame
: The third frame
: The fourth frame
-n : Data #n
-n+1 : Data #(n+1)
-n+2 : Data #(n+2)
-n+3 : Data #(n+3)
-n Repeat : Repeat the data #n
_0 : Device #0 (Falcon or Neptune)
_1 : Device #1 (Falcon or Neptune)
tag xx : Indicates the different devices or current and voltage data

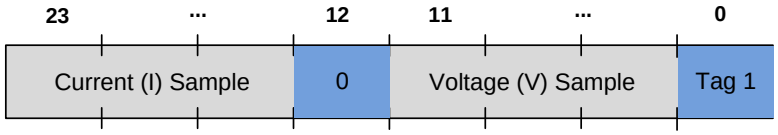
When the PCM interface is operating with a 48KHz frame rate, the left and right I/V samples are carried once per frame. In the case of 24-bit samples, this results in two sample slots per frame. In the case of 16-bit samples, this results in four sample slots per frame.

When the PCM interface is operating with a 96KHz frame rate, the left channel I/V samples and the right channel I/V samples are carried in alternate frames. In the case of 24-bit samples, this will result in one sample slot per frame. A second slot per frame may optionally be transmitted with all 0's, if the PCM receiving host is capable of stereo operation only. In the case of 16-bit samples, this results in two sample slots per frame. To allow software to differentiate left channel I/V samples from right channel I/V samples, the left channel samples are marked by setting the least significant bit of the sample slot to 1.

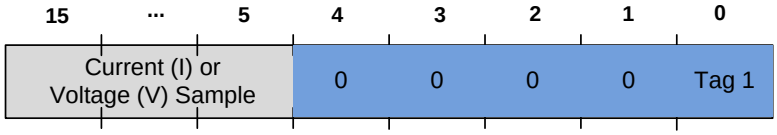
When the PCM interface is operating with a 192KHz frame rate, with 24-bit samples, the left channel I/V samples and the right channel I/V samples are carried in alternate frames, as in the 96KHz case (including marking of the left channel data). However, each set of left and right I and V samples is repeated in the subsequent two frames, which would otherwise not be needed to carry I/V samples. No distinctive marking is made between the original and repeated samples, and software may safely discard every other set from the data stream. As with the 96KHz case, an option exists for transmitting a second sample slot in each frame, containing 0's, if the receiving host is capable of stereo operation only. When the PCM interface is operating with a 192KHz frame rate, with 16-bit samples, the left channel I sample is carried in a first frame. The left channel V sample is carried in a second frame. The right channel I

sample is carried in a third frame, and the right channel V sample is carried in a fourth frame. This pattern then repeats with every four frames. To allow software to locate the beginning of the pattern, the left channel I sample is marked by setting the least significant bit of the sample slot to 1.
The valid data for 24-bit or 16-bit mode is shown as follow.

**Neptune Format for 24 bits data**




**Neptune Format for 16 bits data**




## 9.4.3 Re-packing the IV Stream

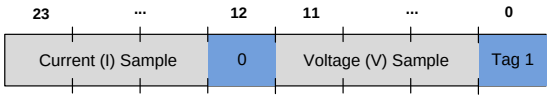
To support two devices (Falcon or Neptune) simultaneously, the format of IV stream re-packing is defined as follows.

| | 48K / 24b | 48K / 16b | 96K / 24b | 96K / 16b | 192K / 24b | 192K / 16b |
|---|---|---|---|---|---|---|
| Host Mode | AEC_L_0 -n | AEC_L_0 -n | AEC_L_0 -n | AEC_L_0 -n | AEC_L_0 -n | AEC_L_0 -n |
| | AEC_R_0 -n | AEC_R_0 -n | AEC_R_0 -n | AEC_R_0 -n | AEC_R_0 -n | AEC_R_0 -n |
| | AEC_L_1 -n | AEC_L_1 -n | AEC_L_1 -n | AEC_L_1 -n | AEC_L_1 -n | AEC_L_1 -n |
| | AEC_R_1 -n | AEC_R_1 -n | AEC_R_1 -n | AEC_R_1 -n | AEC_R_1 -n | AEC_R_1 -n |
| | IV_L_0 -n | I_L_0 -n | IV_L_0 -n | I_L_0 -n tag 00 | IV_L_0 -n | I_L_0 -n tag 00 |
| | IV_R_0 -n | I_R_0 -n | IV_R_0 -n | I_R_0 -n tag 00 | IV_R_0 -n | I_R_0 -n tag 00 |
| | IV_L_1 -n | V_L_0 -n | AEC_L_0 -n+1 | I_L_1 -n tag 01 | AEC_L_0 -n+1 | AEC_L_0 -n+1 |
| | IV_R_1 -n | V_R_0 -n | AEC_R_0 -n+1 | I_R_1 -n tag 01 | AEC_R_0 -n+1 | AEC_R_0 -n+1 |
| | | I_L_1 -n | AEC_L_1 -n+1 | AEC_L_0 -n+1 | AEC_L_1 -n+1 | AEC_L_1 -n+1 |
| | | I_R_1 -n | AEC_R_1 -n+1 | AEC_R_0 -n+1 | AEC_R_1 -n+1 | AEC_R_1 -n+1 |
| | | V_L_1 -n | IV_L_1 -n tag 1 | AEC_L_1 -n+1 | IV_L_1 -n tag 1 | I_L_1 -n tag 01 |
| | | V_R_1 -n | IV_R_1 -n tag 1 | AEC_R_1 -n+1 | IV_R_1 -n tag 1 | I_R_1 -n tag 01 |
| | | | | V_L_0 -n tag 10 | AEC_L_0 -n+2 | AEC_L_0 -n+2 |
| | | | | V_R_0 -n tag 10 | AEC_R_0 -n+2 | AEC_R_0 -n+2 |
| | | | | V_L_1 -n tag 11 | AEC_L_1 -n+2 | AEC_L_1 -n+2 |
| | | | | V_R_1 -n tag 11 | AEC_R_1 -n+2 | AEC_R_1 -n+2 |
| | | | | | IV_L_0 -n (Repeat) | V_L_0 -n tag 10 |
| | | | | | IV_R_0 -n (Repeat) | V_R_0 -n tag 10 |
| | | | | | AEC_L_0 -n+3 | AEC_L_0 -n+3 |
| | | | | | AEC_R_0 -n+3 | AEC_R_0 -n+3 |
| | | | | | AEC_L_1 -n+3 | AEC_L_1 -n+3 |
| | | | | | AEC_R_1 -n+3 | AEC_R_1 -n+3 |
| | | | | | IV_L_1 -n (Repeat) tag 1 | V_L_1 -n tag 11 |
| | | | | | IV_R_1 -n (Repeat) tag 1 | V_R_1 -n tag 11 |

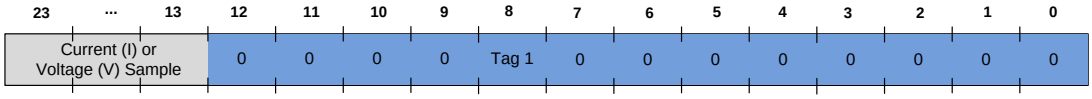
When the Oceanside is configured for 48Ksps operation, the I/V samples are produced at the same rate as the AEC audio samples. Therefore, the I/V samples are transported through the HDA or host I2S interface at the same rate as the AEC audio samples. For 24-bits mode, 4 channels are needed totally for two devices. For 16-bits mode, 8 channels are needed totally for two devices.
When the Oceanside is configured for 96Ksps operation, the I/V samples are produced at half the rate of the AEC audio samples. In this case, the I/V data of device #1 and device #2 are carried in alternate frames. To allow software to differentiate the devices and current or voltage samples, for 24-bits mode the device #2 data are marked by setting the least significant bit of the sample slot to 1, and for 16-bits mode 2-bits tag are used. For 24-bits mode, 2 channels are needed totally for two devices. For 16-bits mode, 4 channels are needed totally for two devices.
When the Oceanside is configured for 192Ksps operation, the I/V samples are produced at a quarter the rate of the DAC audio samples. In this case, for 24-bits mode the I/V data of device #1 and device #2 are carried in alternate frames, and for 16-bits mode the current samples, voltage samples, device #1 samples and device #2 samples are all carried in alternate frames. To allow software to differentiate the devices and current or voltage samples, for 24-bits mode the device #2 data are marked by setting the least significant bit of the sample slot to 1, and for 16-bits mode 2-bits tag are used. 2 channels are needed in both 24-bits mode and 16-bits mode.

The valid data for 24-bit or 16-bit mode is shown as follow.

**Oceanside Format for 24 bits data**



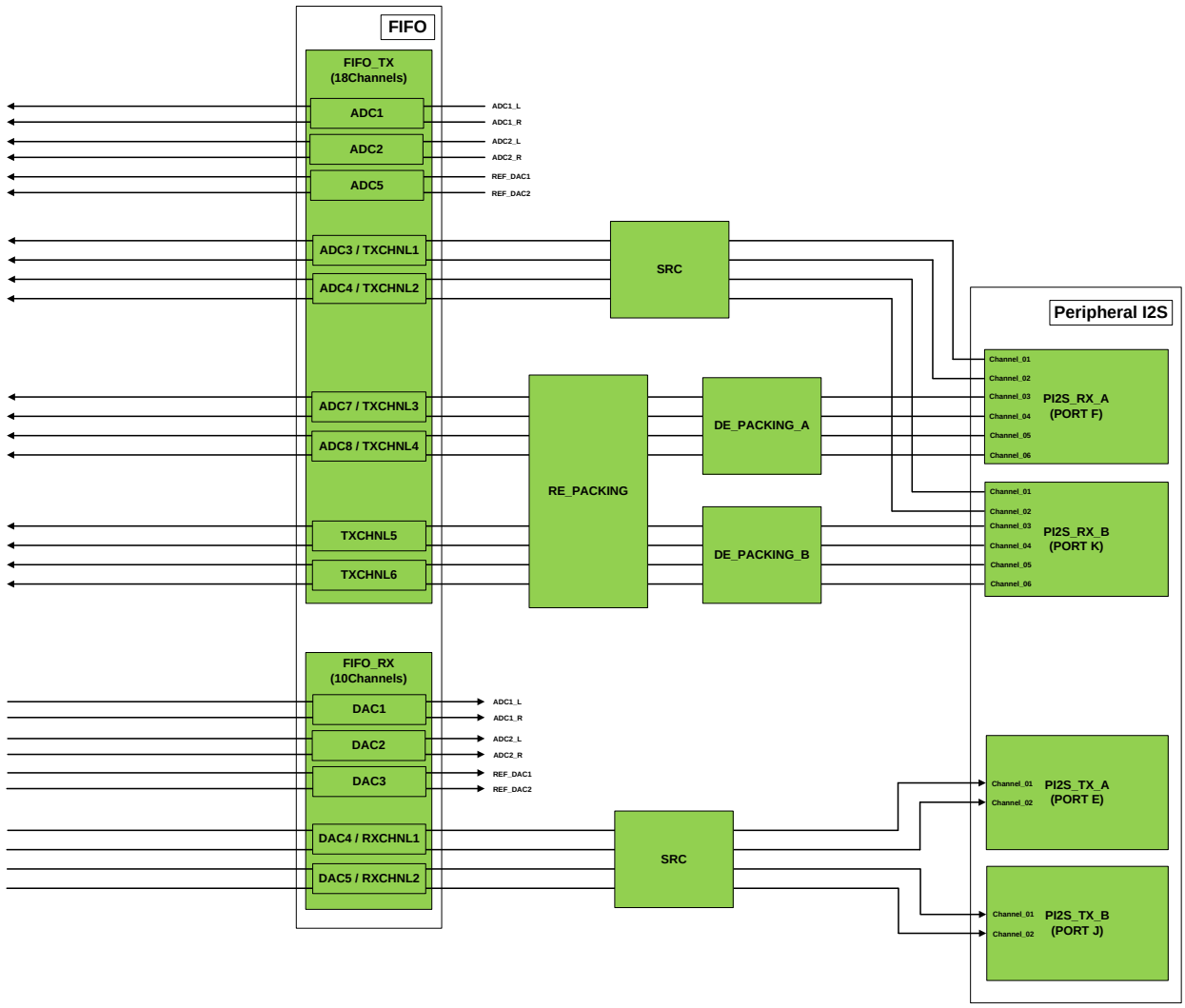

**Oceanside Format for 16 bits data**




**Oceanside Format for 16 bits data under 192 KHz**

## 9.4.4 Channel Assignment

The IV stream packing is assigned to channel #3 ~ channel #6 of peripheral I2S block.
The IV stream packing is assigned to TX channel #3 ~ TX channel #6 of Host FIFO.




## 9.5 IV Packing Register Description

The related registers definition of IV stream packing is listed as follow.

**Name: IV_PACKING _REG2**
**HDA Address: 0x01be_3900**
**I2C Address: 0x6f39**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:4 | dp_mode_1 | RW | RO | 4'h0 | Sets the IV de-packing mode for device #2.<br>4'b0000: Falcon mode, 48KHz, 24 bit<br>4'b0001: Falcon mode, 48KHz, 16 bit<br>4'b0010: Falcon mode, 96KHz, 24 bit<br>4'b0011: Falcon mode, 96KHz, 16 bit<br>4'b0100: Falcon mode, 192KHz, 24 bit<br>4'b0101: Falcon mode, 192KHz, 16 bit<br>4'b1000: Neptune mode, 48KHz, 24 bit<br>4'b1001: Neptune mode, 48KHz, 16 bit<br>4'b1010: Neptune mode, 96KHz, 24 bit<br>4'b1011: Neptune mode, 96KHz, 16 bit |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | 4’b1100: Neptune mode, 192KHz, 24 bit<br>4’b1101: Neptune mode, 192KHz, 16 bit<br>Others: reserved. |
| 3:0 | dp_mode_0 | RW | RO | 4’h0 | Sets the IV de-packing mode for device #1.<br>4’b0000: Falcon mode, 48KHz, 24 bit<br>4’b0001: Falcon mode, 48KHz, 16 bit<br>4’b0010: Falcon mode, 96KHz, 24 bit<br>4’b0011: Falcon mode, 96KHz, 16 bit<br>4’b0100: Falcon mode, 192KHz, 24 bit<br>4’b0101: Falcon mode, 192KHz, 16 bit<br>4’b1000: Neptune mode, 48KHz, 24 bit<br>4’b1001: Neptune mode, 48KHz, 16 bit<br>4’b1010: Neptune mode, 96KHz, 24 bit<br>4’b1011: Neptune mode, 96KHz, 16 bit<br>4’b1100: Neptune mode, 192KHz, 24 bit<br>4’b1101: Neptune mode, 192KHz, 16 bit<br>Others: reserved. |

**Name: IV_PACKING _REG4**
**HDA Address: 0x01be_3b00**
**I2C Address: 0x6f3b**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 6:4 | rp_mode | RW | RO | 3’h0 | Sets the IV re-packing mode for host.<br>3’b000: 48KHz, 24 bit<br>3’b001: 48KHz, 16 bit (I2S mode only)<br>3’b010: 96KHz, 24 bit<br>3’b011: 96KHz, 16 bit (I2S mode only)<br>3’b100: 192KHz, 24 bit<br>3’b101: 192KHz, 16 bit (I2S mode only)<br>Others: reserved. |
| 0 | wfifo_clk_en | RW | RO | 1’h0 | Clock enable signal for wfifo_clk. |

**Name: IV_PACKING _REG5**
**HDA Address: 0x01be_3c00**
**I2C Address: 0x6f3c**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 5 | switch_i_v_1 | RW | RO | 1’h0 | Switches the current and voltage channel for device #2. |
| 4 | switch_l_r_1 | RW | RO | 1’h0 | Switches the left and right channel for device #2. |
| 3 | switch_i_v_0 | RW | RO | 1’h0 | Switches the current and voltage channel for device #1. |
| 2 | switch_l_r_0 | RW | RO | 1’h0 | Switches the left and right channel for device #1. |
| 1 | dp_en_1 | RW | RO | 1’h0 | Enables the IV de-packing block for device #2. All the configuration related to device #2 above must be stable before asserting this bit. |
| 0 | dp_en_0 | RW | RO | 1’h0 | Enables the IV de-packing block for device #1. All the configuration related to device #1 above must be stable before asserting this bit. |

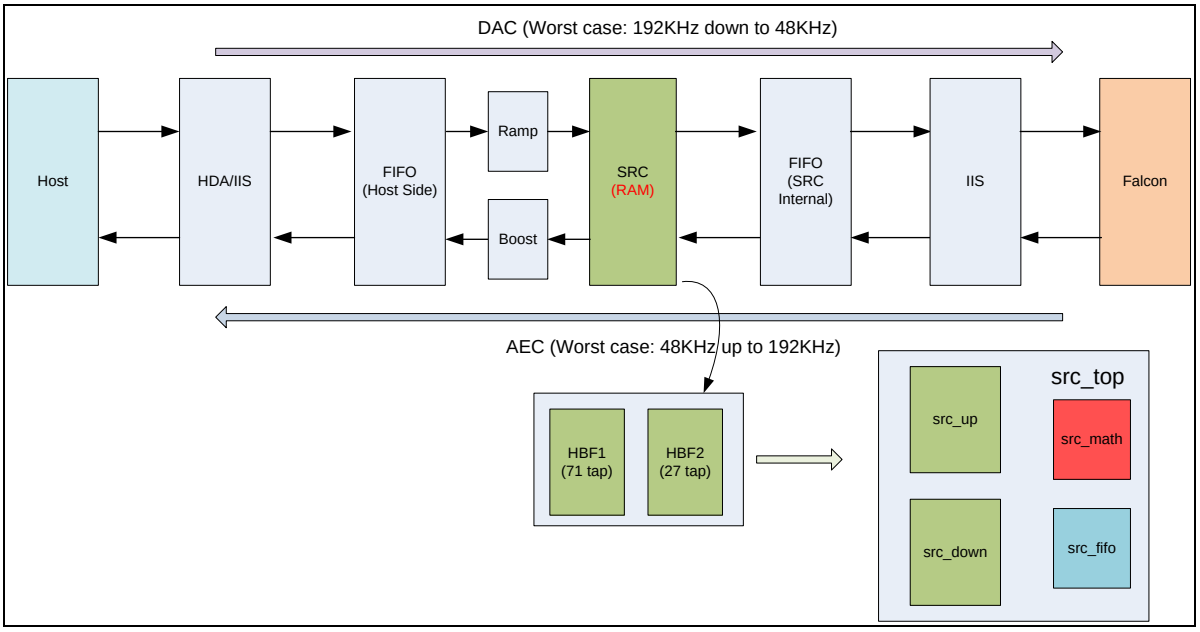
## 9.6 Sample rate conversion

The host and peripheral interfaces may run at different rates. In HD Audio mode, the host interface may configure each stream independently. However, in I2S mode, all streams to and from the host run at the same rate. Each of the two peripheral interfaces may run at a different rate. Since the input and output ports on each interface share a clock, they must run at the same rate. To support independent rates on each of the ports, sample rate conversion is required.

### 9.6.1 Features

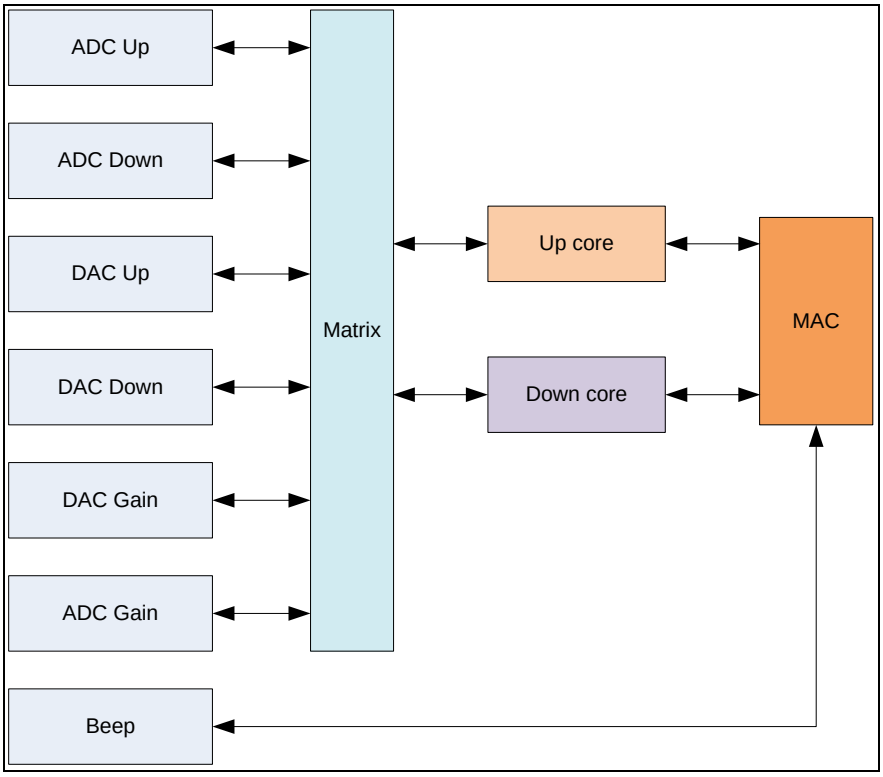
1) HBF
   a) Pipeline control
   b) Read/write host interface FIFO control
   c) Write/read output FIFO control
   d) Write/read internal buffer RAM
   e) Coefficient read control
   f) Performance:
   g) Tap: HB1--71; HB2--27
   h) ADC
      i) 19-bit input, 19-bit output, internal multiplier result is 39-bit
      ii) Internal adder is 20-bit
   i) DAC
      i) 19-bit input, 19-bit output, internal multiplier result is 39-bit
      ii) Internal adder is 20-bit
2) Gain (For system, ADC gain is [-74dB,6dB] by step 1dB; DAC is [-74dB,6dB] by step 0.25dB)
   a) Mute
   b) Gain adjust: Sample*Gain_Coefficient. It's after the HBF.
   c) Gain ramping

### 9.6.2 Path

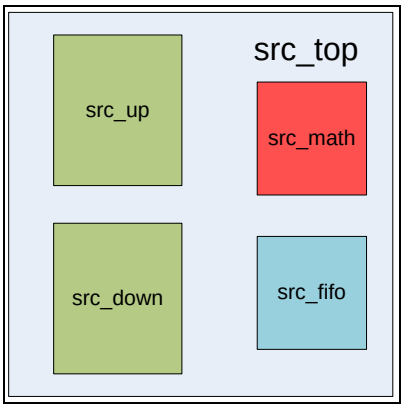



Note: We can't share the SRC internal FIFO with host side FIFO: because SRC calculation is based on L first and then R second, but the peripheral i2s and host side write/read FIFO is based on L&R at the same time!

### 9.6.3 Application in Oceanside



### 9.6.4 Module Interface




#### 9.6.4.1 src_top

| Name | Bits | Direction | Description |
|---|---|---|---|
| *Clock & Reset* | | | |
| i_clk_src | 1 | I | SRC block clock. Default, it's 24MHz. |
| i_clk_i2s_per | 1 | I | I2S read/write FIFO interface clock. |
| i_rstn | 1 | I | SRC block reset input. It may be asynchronous released. |
| *Read FIFO Interface (from host interface)* | | | |
| o_dacfifo_rd_l | 1 | O | FIFO read control, used to read the sample data from FIFO. |
| o_dacfifo_rd_r | 1 | O | FIFO read control, used to read the sample data from FIFO. |
| i_dacfifo_rvld_l | 1 | I | FIFO read data valid flag. It may be from the FIFO not empty flag, which indicate read operation can be launched when it's high. |
| i_dacfifo_rvld_r | 1 | I | FIFO read data valid flag. It may be from the FIFO not empty flag, which indicate read operation can be launched when it's high. |
| i_dacfifo_rdata_l | p_SRCDW | I | Sample data read from FIFO. |
| i_dacfifo_rdata_r | p_SRCDW | I | Sample data read from FIFO. |
| *Read FIFO Interface (by peripheral i2s interface)* | | | |

| | | | |
|---|---|---|---|
| i_dacfifo_rd_l | 1 | I | FIFO read control, used to read the sample data from FIFO. |
| i_dacfifo_rd_r | 1 | I | FIFO read control, used to read the sample data from FIFO. |
| o_dacfifo_rvld_l | 1 | O | FIFO read data valid flag. It may be from the FIFO not empty flag, which indicate read operation can be launched when it's high. |
| o_dacfifo_rvld_r | 1 | O | FIFO read data valid flag. It may be from the FIFO not empty flag, which indicate read operation can be launched when it's high. |
| o_dacfifo_rdata_l | p_SRCDW | O | Sample data read from FIFO. |
| o_dacfifo_rdata_r | p_SRCDW | O | Sample data read from FIFO. |
| *Write FIFO Interface (to host interface)* | | | |
| o_adcfifo_wr_l | 1 | O | FIFO write control, used to write the SRC result data to FIFO. |
| o_adcfifo_wr_r | 1 | O | FIFO write control, used to write the SRC result data to FIFO. |
| o_adcfifo_wdata_l | p_SRCDW | O | SRC result data which write to FIFO. |
| o_adcfifo_wdata_r | p_SRCDW | O | SRC result data which write to FIFO. |
| *Write FIFO Interface (write internal FIFO by peripheral I2S)* | | | |
| i_adcfifo_wr_l | 1 | I | FIFO write control, used to write the SRC result data to FIFO. |
| i_adcfifo_wr_r | 1 | I | FIFO write control, used to write the SRC result data to FIFO. |
| i_adcfifo_wdata_l | p_SRCDW | I | SRC result data which write to FIFO. |
| i_adcfifo_wdata_r | p_SRCDW | I | SRC result data which write to FIFO. |
| *DAC RAM Interface* | | | |
| o_dacram_0_addr | p_SRAMAW | O | SRC calculation buffer RAM address. |
| o_dacram_0_wen | 1 | O | Buffer RAM write enable:<br>1'b0: write<br>1'b1: read |
| o_dacram_0_wdata | p_SRCDW | O | Buffer RAM write data |
| o_dacram_0_cen | 1 | O | Buffer RAM chip enable:<br>1'b0: valid<br>1'b1: disable |
| i_dacram_0_rdata | p_SRCDW | I | Buffer RAM read data out. |
| o_dacram_1_addr | p_SRAMAW | O | SRC calculation buffer RAM address. |
| o_dacram_1_we | 1 | O | Buffer RAM write enable:<br>1'b0: write<br>1'b1: read |
| o_dacram_1_wdata | p_SRCDW | O | Buffer RAM write data |

| | | | |
|---|---|---|---|
| o_dacram_1_cen | 1 | O | Buffer RAM chip enable:<br>1'b0: valid<br>1'b1: disable |
| i_dacram_1_rdata | p_SRCDW | I | Buffer RAM read data out. |
| *ADC RAM Interface* | | | |
| o_adcram_0_addr | p_SRAMAW | O | SRC calculation buffer RAM address. |
| o_adcram_0_wen | 1 | O | Buffer RAM write enable:<br>1'b0: write<br>1'b1: read |
| o_adcram_0_wdata | p_SRCDW | O | Buffer RAM write data |
| o_adcram_0_cen | 1 | O | Buffer RAM chip enable:<br>1'b0: valid<br>1'b1: disable |
| i_adcram_0_rdata | p_SRCDW | I | Buffer RAM read data out. |
| o_adcram_1_addr | p_SRAMAW | O | SRC calculation buffer RAM address. |
| o_adcram_1_we | 1 | O | Buffer RAM write enable:<br>1'b0: write<br>1'b1: read |
| o_adcram_1_wdata | p_SRCDW | O | Buffer RAM write data |
| o_adcram_1_cen | 1 | O | Buffer RAM chip enable:<br>1'b0: valid<br>1'b1: disable |
| i_adcram_1_rdata | p_SRCDW | I | Buffer RAM read data out. |
| *ROM Interface* | | | |
| o_rom_addr | p_ROMAW | O | Coefficient ROM address, including gain coefficient and HBF coefficient. |
| i_rom_data | p_ROMDW | I | Coefficient value which read from ROM. |
| *Control Interface* | | | |
| o_adc_rstn | 1 | O | ADC reset output. When change the sample rate, it's valid. To host side FIFO to clear the write side logic. |
| adc_streamID_zero | 1 | I | From host HD-A side. Indicate the valid stream operation.<br>'b1: stream not valid, no stream transfer<br>'b0: valid stream start to transfer |
| i_src_en | 1 | I | src block enable control signal. High indicate enable valid. |
| i_beep_en | 1 | I | The beep enable of DAC path. When enable, beep will add to DAC stream. High active. |
| i_beep_dac | p_SRCDW | I | The beep data of DAC path. It's from digital beep. |
| i_mute_dac_l | 1 | I | DAC stream L channel mute control. High active. |
| i_mute_dac_r | 1 | I | DAC stream R channel mute control. High active. |

| | | | |
|---|---|---|---|
| i_mute_adc_l | 1 | I | ADC stream L channel mute control. High active. |
| i_mute_adc_r | 1 | I | ADC stream R channel mute control. High active. |
| i_adcsource_rate | 2 | I | The source audio stream sample rate:<br>2'b00: 48KHz<br>2'b01: 96KHz<br>2'b10: Reserved<br>2'b11: 192KHz |
| i_adcsink_rate | 2 | 1 | The source audio stream sample rate:<br>2'b00: 48KHz<br>2'b01: 96KHz<br>2'b10: Reserved<br>2'b11: 192KHz |
| i_adcgain_l | p_AGDW | I | Gain setting for left channel which is from relational volume control register. It's the gain value address. |
| i_adcgain_r | p_AGDW | I | Gain setting for right channel which is from relational volume control register. It's the gain value address. |
| i_adcgain_en | 1 | I | Gain control enable signal.<br>1'b1: enable the gain setting<br>1'b0: disable the gain setting, bypass |
| i_adcstereo | 1 | I | Source channel flag:<br>1'b1: stereo<br>1'b0: mono |
| i_dacsource_rate | 2 | I | The source audio stream sample rate:<br>2'b00: 48KHz<br>2'b01: 96KHz<br>2'b10: Reserved<br>2'b11: 192KHz |
| i_dacsink_rate | 2 | 1 | The source audio stream sample rate:<br>2'b00: 48KHz<br>2'b01: 96KHz<br>2'b10: Reserved<br>2'b11: 192KHz |
| i_dacgain_l | p_AGDW | I | Gain setting for left channel which is from relational volume control register. It's the gain value address. |
| i_dacgain_r | p_AGDW | I | Gain setting for right channel which is from relational volume control register. It's the gain value address. |

| | | | |
|---|---|---|---|
| i_dacgain_en | 1 | I | Gain control enable signal.<br>1'b1: enable the gain setting<br>1'b0: disable the gain setting, bypass |
| i_dacstereo | 1 | I | Source channel flag:<br>1'b1: stereo<br>1'b0: mono |

### 9.6.4.2 src_up/down core

| Name | Bits | Direction | Description |
|---|---|---|---|
| *Clock & Reset* | | | |
| i_clk_src | 1 | I | SRC block clock. Default, it's 24MHz. |
| i_rstn | 1 | I | SRC block reset input. It may be asynchronous released. |
| *Read FIFO Interface* | | | |
| o_fifo_rd_l | 1 | O | FIFO read control, used to read the sample data from FIFO. |
| o_fifo_rd_r | 1 | O | FIFO read control, used to read the sample data from FIFO. |
| i_fifo_rvld_l | 1 | I | FIFO read data valid flag. It may be from the FIFO not empty flag, which indicate read operation can be launched when it's high. |
| i_fifo_rvld_r | 1 | I | FIFO read data valid flag. It may be from the FIFO not empty flag, which indicate read operation can be launched when it's high. |
| i_fifo_rdata_l | p_SRCDW | I | Sample data read from FIFO. |
| i_fifo_rdata_r | p_SRCDW | I | Sample data read from FIFO. |
| *Write FIFO Interface* | | | |
| o_fifo_wr_l | 1 | O | FIFO write control, used to write the SRC result data to FIFO. |
| o_fifo_wr_r | 1 | O | FIFO write control, used to write the SRC result data to FIFO. |
| o_fifo_wdata_l | p_SRCDW | O | SRC result data which write to FIFO. |
| o_fifo_wdata_r | p_SRCDW | O | SRC result data which write to FIFO. |
| *RAM Interface* | | | |
| o_ram_0_addr | p_SRAMAW | O | SRC calculation buffer RAM address. |
| o_ram_0_wen | 1 | O | Buffer RAM write enable:<br>1'b0: write<br>1'b1: read |
| o_ram_0_wdata | p_SRCDW | O | Buffer RAM write data |

| | | | |
|---|---|---|---|
| o_ram_0_cen | 1 | O | Buffer RAM chip enable:<br>1'b0: valid<br>1'b1: disable |
| i_ram_0_rdata | p_SRCDW | I | Buffer RAM read data out. |
| o_ram_1_addr | p_SRAMAW | O | SRC calculation buffer RAM address. |
| o_ram_1_we | 1 | O | Buffer RAM write enable:<br>1'b0: write<br>1'b1: read |
| o_ram_1_wdata | p_SRCDW | O | Buffer RAM write data |
| o_ram_1_cen | 1 | O | Buffer RAM chip enable:<br>1'b0: valid<br>1'b1: disable |
| i_ram_1_rdata | p_SRCDW | I | Buffer RAM read data out. |
| *ROM Interface* | | | |
| o_rom_addr | p_ROMAW | O | Coefficient ROM address, including gain coefficient and HBF coefficient. |
| i_rom_data | p_ROMDW | I | Coefficient value which read from ROM. |
| *Control Interface* | | | |
| i_src_en | 1 | I | src block enable control signal. High indicate enable valid. |
| i_source_rate | 2 | I | The source audio stream sample rate:<br>2'b00: 48KHz<br>2'b01: 96KHz<br>2'b10: Reserved<br>2'b11: 192KHz |
| i_sink_rate | 2 | 1 | The source audio stream sample rate:<br>2'b00: 48KHz<br>2'b01: 96KHz<br>2'b10: Reserved<br>2'b11: 192KHz |
| i_gain_l | p_AGDW | I | Gain setting for left channel which is from relational volume control register. It’s the gain value address. |
| i_gain_r | p_AGDW | I | Gain setting for right channel which is from relational volume control register. It’s the gain value address. |
| i_gain_en | 1 | I | Gain control enable signal.<br>1'b1: enable the gain setting<br>1'b0: disable the gain setting, bypass |

| | | | |
|---|---|---|---|
| i_stereo | 1 | I | Source channel flag:<br>1'b1: stereo<br>1'b0: mono |
| o_cal_done | 1 | O | Calculation done flag. |
| *Arithmetic Interface* | | | |
| o_mult_a | p_SRCDW | O | Output from the control logic. Used for multiplier. It's the sample data. |
| o_mult_b | p_ROMDW | O | Output from the control logic. Used for multiplier. It's the gain/hbf coefficient. |
| o_acc_pre | p_MACDW | O | The previous adder result, output to arithmetic to complete the next accumulate operation. |
| i_mac_out | p_MACDW | I | From arithmetic calculation result, used for next operation. |

| **Comments:** | | |
|---|---|---|
| p_SRCDW | SRC audio stream sample data width, following signals are the same width:<br>1. Source and Sink data FIFO<br>2. Buffer RAM data<br>3. Multiplier input data | 19 |
| p_SRAMAW | The buffer RAM address width | 7 |
| p_ROMAW | The coefficient ROM address width | 10 |
| p_ROMDW | The coefficient data width | 19 |
| p_AGDW | The gain setting control register width | 10 |
| p_MACDW | The MAC calculation data width, it should be:<br>p_SRCDW+p_ROMDW+'d5<br>The 'd5 is extend bits: the maximum non-zero coefficient is 18, so the maximum adder number is 18. So, extend 5-bits for the carry-in. | 19+20+5 |

### 9.6.5 Memory

**Decimator**

| | Bank0 | Address | Bank1 | Address | |
|---|---|---|---|---|---|
| X | odd_r | base_bank0+0x67 | odd_r | base_bank1+0x67 | 18w |
| | odd_l | base_bank0+0x58 | odd_l | base_bank1+0x58 | 18w |
| | even_r | base_bank0+0x40 | even_r | base_bank1+0x40 | 36w |
| | even_l | base_bank0+0x22 | even_l | base_bank1+0x22 | 36w |
| Y | odd_r | base_bank0+0x18 | odd_r | base_bank1+0x18 | 7w |
| | odd_l | base_bank0+0x14 | odd_l | base_bank1+0x14 | 7w |
| | even_r | base_bank0+0x7 | even_r | base_bank1+0x7 | 14w |
| | even_l | base_bank0 | even_l | base_bank1 | 14w |

X: 71 tap, 18 non-zero coefficient. Even: 36; Odd: 18
Y: 27 tap, 7 non-zero coefficient. Even: 14; Odd: 7
total: 76/per bank

**Interpolator**

| | Bank0 | | Bank1 | | |
|---|---|---|---|---|---|
| X | sampl_r | base_bank0+0x35 | sampl_r | base_bank0+0x35 | 37w |
| | sampl_l | base_bank0+0x16 | sampl_l | base_bank0+0x16 | 37w |
| Y | sample_r | base_bank0+0x8 | sample_r | base_bank0+0x8 | 15w |
| | sample_l | base_bank0 | sample_l | base_bank0 | 15w |

X: 71 tap, 18 non-zero coefficient
Y: 27 tap, 7 non-zero coefficient
total: 54/per bank

not including the center coefficient

| | Bank0 | | Bank1 | |
|---|---|---|---|---|
| | sampleN | | sampleN | |
| | …. | | …. | |
| 4 | sampl14 | 2nd | sampl16 | 1st |
| 3 | sampl12 | 4th | sampl14 | 3rd |
| 2 | sampl8 | | sampl10 | |
| 1 | sampl4 | 3rd | sampl6 | 4th |
| 0 | sampl0 | 1st | sampl2 | 2nd |

For decimator, if even write twice, the odd need write wrap. So, process it specially for odd
For interpolator, every odd sample is used and only the center-coefficient sample is write-read through. So, the storage is not the same as decimator.

**Update of interpolator**

| Total | | | |
|---|---|---|---|
| Y | sampl_r | base_bank0+0x86 | 15w |
| | sampl_l | base_bank0+0x72 | 15w |
| X | sample_r | base_bank0+0x36 | 37w |
| | sample_l | base_bank0 | 37w |

| Two banks | Bank0 | Address | Bank1 | Address |
|---|---|---|---|---|
| Y | sampl_r | base_bank0+0x43 | sampl_r | base_bank0+0x43 |
| | sampl_l | base_bank0+0x36 | sampl_l | base_bank0+0x36 |
| X | sample_r | base_bank0+0x18 | sample_r | base_bank0+0x18 |

| | | | |
|---|---|---|---|
| sample_l | base_bank0 | sample_l | base_bank0 |

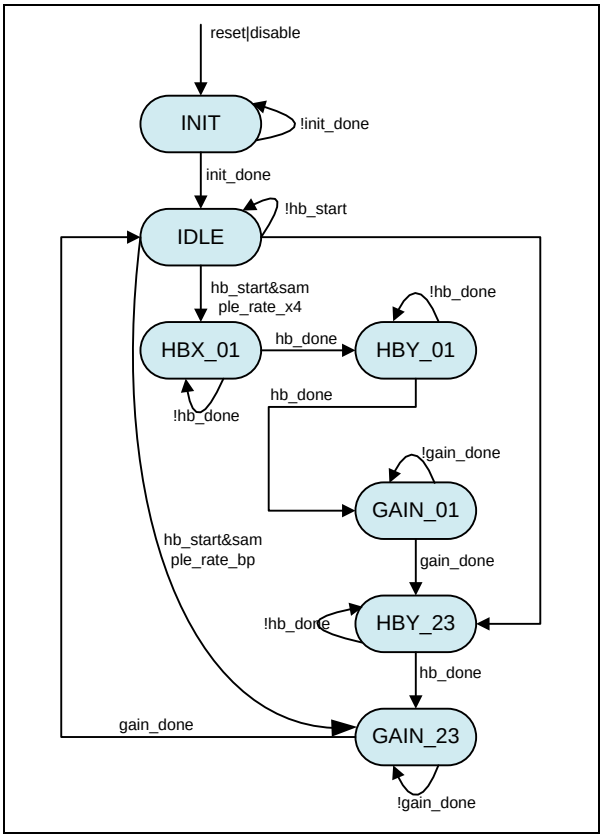
1) Bank0: read address is interval between pointer + counter and pointer - counter
2) Bank1: read address is interval between pointer - counter and pointer + counter
3) Newer sample:
   a) Min: write_index - process_counter (Initial example, top of the location)
   b) Max: Min + max_sample (decimator is Ntap/2; interpolator is max non-zero coefficient)
4) Older sample:
   a) Max: write_index + process_counter + 1 (Initial example, bottom of the location which is wrap to bottom)
   b) Min: max - max_sample (same as newer sample)

Final: all memory blocks are 80x20. 80 is to support double down; 20 is for optimize the memory area. Area of 20 is smaller than 19.

| | | number | result |
|---|---|---|---|
| RF1P_80X19 | 27674.941 | 8 | 221399.53 |
| coefficient | 44963 | 2 | 89926 |
| control | 200000 | 2 | 400000 |
| | | total | 711325.53 |

### 9.6.6 State Machines

**UP:**

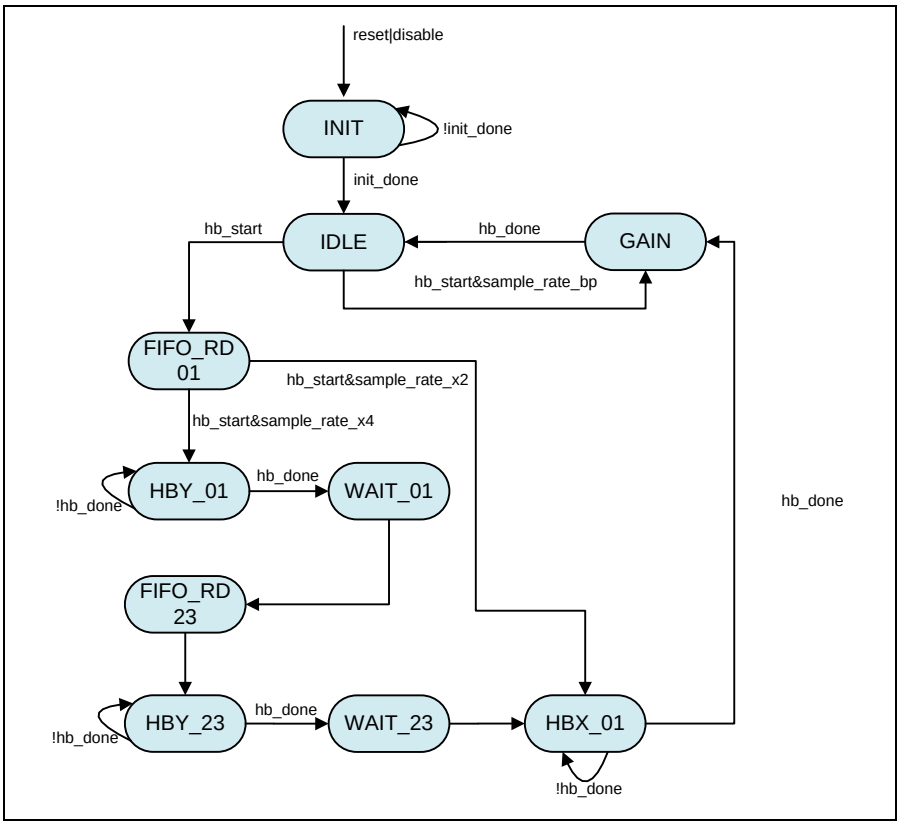



| State | State Description | Next_State | Stimulus | Description of Stimulus | Cycle |
|---|---|---|---|---|---|
| INIT | In this state, initial the buffer RAM to zero. When this block is reset or disabled, enter to this state and clear the buffer RAM. | IDLE | init_done | when the hb_counter==ram_address (more margin). For interpolator, it's up to 'd54. Add 6 margin cycles, to 'd60. | 80 |
| IDLE | Idle state for all other states. When there is no operation, enter to this state. | HBX_01 | hb_start & sample_rate_x4 | hb_start and sr_space should be synchronized to src_clk. The hb_start is determined by:<br>1. FIFO read valid<br>2. stereo | x |
| | | HBY_23 | hb_start & sample_rate_x2 | Same as above | |
| | | GAIN_23 | hb_start & sample_rate_bp | Only for gain adjust. When sample ration is bypass, but the gain is enable, this transfer valid | |
| HBX_01 | When get the first sample, start to calculate the interpolation samples. In this state, one input sample can be interpolated to two samples. | HBY_01 | hb_done | hb_counter==HBX_01_cycle (38/19). For interpolator, it's (ceil(tap_hbx/4)+1)*2. For one channel is 1/4, but for stereo, it should be 1/2 | 18 |
| HBY_01 | In this state, calculate the first pair of output sample. The input sample is from HBX_01 state. | HBX_WB | hb_done | hb_counter==HBY_01_cycle (16/8). For interpolator, it's (ceil(tap_hbx/4)+1)*2. For one channel is 1/4, but for stereo, it should be 1/2 | 7 |
| GAIN_01 | In this state, calculate the first pair of sample gain. The data is from HBY_01. | HBY_23 | gain_done | hb_counter=='d1. Which indicates two samples gain adjust cycles | 2 |

| | | | | | |
|---|---|---|---|---|---|
| HBY_23 | In this state, calculate the second pair of output sample. The input sample is from HBX_WB state. If the sample ration is x2, this state should calculate the 1st pair of output sample. | GAIN_23 | hb_done | hb_counter==HBY_01_cycle (16/8). For interpolator, it's (ceil(tap_hbx/4)+1)*2. For one channel is 1/4, but for stereo, it should be 1/2 | 7 |
| GAIN_23 | In this state, calculate the second pair of sample gain. The data is from HBY_23. If the sample ration is x2, this state should calculate the 1st pair of output sample. | IDLE | gain_done | hb_counter=='d1. Which indicates two samples gain adjust cycles | 2 |

Up: x4 Maximum cycles: 18+2*7+2*2=36
x2 cycles:7+2 = 9
Gain only: 1 cycle

**DOWN:**




| State | State Description | Next_State | Stimulus | Description of Stimulus | Cycles |
|---|---|---|---|---|---|
| INIT | In this state, initial the buffer RAM to zero. When this block is reset or disabled, enter to this state and clear the buffer RAM. | IDLE | init_done | when the hb_counter == ram_address (more margin). For decimator, it's up to 'd76. Add 6 margin cycles, to 'd80. | 80 |
| IDLE | Idle state for all other states. When there is no operation, enter to this state. | GAIN | hb_start & sample_rate_bp | HB start and sample ration is bypass | x |

| | | | | | |
|---|---|---|---|---|---|
| | | FIFO_RD01 | hb_start | The hb_start is determined by:<br>1. FIFO read valid<br>2. stereo | |
| FIFO_RD01 | Read 1st sample from FIFO for next sample calculation. Because not use the write-through of memory, independent state used for FIFO read | HBY_01 | hb_start & sample_rate_x4 | HB start and sample ration is x4 | 1 |
| | | HBX_01 | hb_start & sample_rate_x2 | HB start and sample ration is x2 | |
| HBY_01 | In this state, calculate the first pair input samples and get the first decimation output sample. The input is from gain output. | WAIT_01 | hb_done | hb_counter == HBY_01_cycle (16/8). For interpolator, it's (tap_hby+1)*2/4. For one channel is 1/4, but for stereo, it should be 1/2 | 8 |
| WAIT_01 | Two cycles:<br>1st: write the result of HBY_01 to RAM<br>2nd: read the 2nd sample from FIFO | FIFO_RD23 | wait_done | 2 wait cycles | 2 |
| FIFO_RD23 | Read 3rd samples from FIFO for next calculation. | HBY_23 | 1 cycle | 1 wait cycle | 1 |
| HBY_23 | In this state, calculate the second pair input samples and get the second decimation output sample. The input is from gain output. | WAIT_23 | hb_done | Same as above hb_done. | 8 |
| WAIT_23 | Two cycles:<br>1st: write the result of HBY_23 to RAM<br>2nd: read the 4th sample from FIFO | HBX_01 | wait_done | 2 wait cycles | 2 |
| HBX_01 | In this state, calculate the final decimation output sample. The input is from HBY result. | GAIN | hb_done | hb_counter==HBX_01_cycle (38/19). For interpolator, it's (ceil(tap_hbx/4)+1)*2. For one channel is 1/4, but for stereo, it should be 1/2 | 19 |
| GAIN | Calculate the sample gain, for every input sample, gain calculation is 1st step if gain enable. | IDLE | 1 cycle | 1 wait cycle | 1 |

Down: x4 Maximum cycles: 19+2*8+2*2+1*2+1=42
x2 cycles: 19+1=20
Gain only: 1 cycle

### 9.6.7 Registers

**src1_ctrl**

Get Response HD_A[27:0]=0x1ba2c00 Set HD_A[27:8] = 0x1b22c I2C=6D2C

| Bit | Default | Description |
|---|---|---|
| **7** | 0 | src1 enable control. High active.<br>1: enable<br>0: disable |
| **6** | 0 | Reserved |

| | | |
|---|---|---|
| **5:4** | 0 | Port F sample rate which indicate the adc3 in peripheral iis1 side sample rate.<br>2'b00: 48KHz<br>2'b01: 96KHz<br>2'b10: Reserved |
| **3:2** | 0 | Reserved |
| **1:0** | 0 | Port E sample rate which indicate the dac4 in peripheral iis1 side sample rate.<br>2'b00: 48KHz<br>2'b01: 96KHz<br>2'b10: Reserved<br>2'b11: 192KHz |

**src2_ctrl**

Get Response HD_A[27:0]=0x1ba2d00 Set HD_A[27:8] = 0x1b22d I2C=6D2D

| Bit | Default | Description |
|---|---|---|
| **7** | 0 | src2 enable control. High active.<br>1: enable<br>0: disable |
| **6** | 0 | Reserved |
| **5:4** | 0 | Port K sample rate which indicate the adc3 in peripheral iis1 side sample rate.<br>2'b00: 48KHz<br>2'b01: 96KHz<br>2'b10: Reserved |
| **3:2** | 0 | Reserved |
| **1:0** | 0 | Port J sample rate which indicate the dac4 in peripheral iis1 side sample rate.<br>2'b00: 48KHz<br>2'b01: 96KHz<br>2'b10: Reserved<br>2'b11: 192KHz |

**codec_Test13**

Get Response - HD_A [27:0] =0x1CAD000 Set - HD_A[27:11] = 0x1C2D I2C=7134

| Bit | Default | Description |
|---|---|---|
| **11:9** | 0 | Reserved |
| **8** | 0 | Mute disable on signal ramp & FIFO of DAC5<br>0: Mute work on signal ramp & FIFO<br>1: Mute not work on signal ramp & FIFO |

| | | |
|---|---|---|
| **7** | 0 | Signal ramping step size of DAC5<br>0 = 16<br>1 = 8 |
| **6** | 0 | 0 = Enable Signal ramping of DAC5<br>1 = Disable Signal ramping of DAC5 |
| **5:3** | 0 | Reserved |
| **2** | 0 | Mute disable on signal ramp & FIFO of DAC4<br>0: Mute work on signal ramp & FIFO<br>1: Mute not work on signal ramp & FIFO |
| **1** | 0 | Signal ramping step size of DAC4<br>0 = 16<br>1 = 8 |
| **0** | 0 | 0 = Enable Signal ramping of DAC4<br>1 = Disable Signal ramping of DAC4 |

**codec_Test17**

Get Response - HD_A [27:0] =0x1CE1000 Set - HD_A[27:11] = 0x1C61 I2C=7304

| Bit | Default | Description |
|---|---|---|
| **11:9** | 0 | ADC4 gain ramp rate: |
| | | 3'd0: Fs |
| | | 3'd1: Fs/2 |
| | | 3'd2: Fs/4 |
| | | 3'd3: Fs/8 |
| | | 3'd4: Fs/16 |
| | | 3'd5: Fs/32 |
| | | 3'd6: Fs/48 |
| | | 3'd7: Fs/96 |
| **8:6** | 0 | ADC3 gain ramp rate: |
| | | 3'd0: Fs |
| | | 3'd1: Fs/2 |
| | | 3'd2: Fs/4 |
| | | 3'd3: Fs/8 |
| | | 3'd4: Fs/16 |
| | | 3'd5: Fs/32 |
| | | 3'd6: Fs/48 |
| | | 3'd7: Fs/96 |
| **5:3** | 0 | DAC5 gain ramp rate: |
| | | 3'd0: Fs |
| | | 3'd1: Fs/2n |
| | | 3'd2: Fs/4 |
| | | 3'd3: Fs/8 |
| | | 3'd4: Fs/16 |
| | | 3'd5: Fs/32 |
| | | 3'd6: Fs/48 |

| | | |
|---|---|---|
| | | 3'd7: Fs/96 |
| **2:0** | 0 | DAC4 gain ramp rate: |
| | | 3'd0: Fs |
| | | 3'd1: Fs/2 |
| | | 3'd2: Fs/4 |
| | | 3'd3: Fs/8 |
| | | 3'd4: Fs/16 |
| | | 3'd5: Fs/32 |
| | | 3'd6: Fs/48 |
| | | 3'd7: Fs/96 |

### 9.6.8 FIFO RAM Access

**Interpolator**

| State | Read from FIFO | Write to FIFO | Read from RAM | Write to RAM | Comments |
|---|---|---|---|---|---|
| INIT | | | | √ | zero data for initial |
| IDLE | | | | | |
| HBX_01 | √ | | √ | √ | 1st cycle: read FIFO and write to ram; and read another bank ram<br>2nd cycle: use the write-through to get the 1st read ram data, for another ram bank, it's read at 1st cycle<br>other cycles: read interleave from two bank ram |
| HBY_01 | | | √ | √ | 1st cycle: write HBX_01 result to HBY ram space; and read another bank ram<br>2nd cycle: use the write-through to get the 1st read ram data, for another ram bank, it's read at 1st cycle<br>other cycles: read interleave from two bank ram |
| GAIN_01 | | √ | | | write result to FIFO. Once time, write two samples. Write after two cycles of gain state. |
| HBY_23 | √ | | √ | √ | if x2 ration, read FIFO and write ram operation are the same as HBX_01. if x4 ration, it's same as HBY_01 |
| GAIN_23 | | √ | | | write result to FIFO |

**Decimator**

| State | Read from FIFO | Write to FIFO | Read from RAM | Write to RAM | Comments |
|---|---|---|---|---|---|
| INIT | | | | √ | zero data initialization |
| IDLE | | | | | |
| FIFO_RD01 | √ | | | √ | Read sample data and write it to RAM. Only one cycle, use read ahead FIFO data, read and write to RAM right now |
| HBY_01 | | | √ | | Read sample data and calculation one cycle by one cycle |
| WAIT_01 | √ | | | √ | two cycles:<br>1st: write HBY_01 result to ram<br>2nd: read sample data and write to RAM. Like FIFO_RD state operation |
| FIFO_RD23 | √ | | | √ | Same as FIFO_RD01 |
| HBY_23 | | | √ | | Read sample data and calculation one cycle by one cycle |
| WAIT_23 | √ | | | √ | Same as WAIT_01 |
| HBX_01 | | | √ | | Read sample data and calculation one cycle by one cycle |
| GAIN | √ | √ | | | write result to FIFO after two cycles of gain state; read sample data at the gain state |

### 9.6.9 Cycle Operation

| Cycle | State | Operation |
|---|---|---|
| **1st** | HBX_01 | 1.Read FIFO and write to RAM<br>2.Read the other bank RAM<br>3. Read HBX coefficient |
| **2nd** | HBX_01 | 1.Read two banks RAM<br>2.Read HBX coefficient<br>3.MAC: sum of 1st two banks and coefficient. Accumulation is 0 |
| **3rd** | HBX_01 | 1.Read two banks RAM<br>2.Read HBX coefficient<br>3.MAC: sum of 2nd two banks and coefficient and the accumulation is the 1st MAC result<br>4.Register the 1st MAC result |
| **4th** | HBX_01 | Same as last operation |
| **5th** | HBX_01 | Same as last operation |
| **6th** | HBX_01 | Same as last operation |

| | | |
|---|---|---|
| **7th** | HBX_01 | Same as last operation |
| **8th** | HBX_01 | Same as last operation |
| **9th** | HBX_01 | Same as last operation |
| **10th** | HBX_01 | Same as last operation |
| **11th** | HBX_01 | Same as last operation |
| **12th** | HBX_01 | Same as last operation |
| **13th** | HBX_01 | Same as last operation |
| **14th** | HBX_01 | Same as last operation |
| **15th** | HBX_01 | Same as last operation |
| **16th** | HBX_01 | Same as last operation |
| **17th** | HBX_01 | Same as last operation |
| **18th** | HBX_01 | 1.Read the 1 value coefficient sample from corresponding bank of RAM<br>2.Read HBX coefficient<br>3.MAC: sum of 2nd two banks and coefficient and the accumulation is the 1st MAC result<br>4.Register the previous MAC result |
| **19th** | HBY_01 | 1.Register the last operation MAC result<br>2.Saturate the result to get the interpolator sample<br>3.Write the last cycle read 1 value coefficient sample to HBY space<br>4.Read the other bank ram to get the other sum element<br>5.Read HBY coefficient |
| **20th** | HBY_01 | 1.Store the 18th saturation result to register<br>2.Read two banks ram<br>3.Read HBY coefficient<br>4.MAC: sum of 19th two banks ram data and coefficient and the accumulation 0 |
| **21st** | HBY_01 | 1.Read two banks RAM<br>2.Read HBY coefficient<br>3.MAC: sum of 20th two banks and coefficient and the accumulation is the 20th MAC result<br>4.Register the 20th MAC result |
| **22nd** | HBY_01 | Same as last operation |
| **23rd** | HBY_01 | Same as last operation |
| **24th** | HBY_01 | Same as last operation |
| **25th** | HBY_01 | Same as last operation |
| **26th** | GAIN_01 | 1.MAC: sum of 24th two banks and coefficient and the accumulation is the 24th MAC result<br>2.Register the previous MAC result<br>3.Read the final HBY process sample (1 coefficient sample, 1st sample result of HBY_01) |
| **27th** | GAIN_01 | 1.Write the HBX 2nd sample to HBY space<br>2.Register the HBY 2nd sample<br>3.Gain mul the HBY 1st sample |
| **28th** | HBY_23 | 1.Gain mul the HBY 2nd sample<br>2.Register the 1st Gain mul result<br>3.Write FIFO: HBY 1st out<br>4.Read RAM: one is the 27th write through; the other one is from read another bank<br>5.Register the 2nd gain mul result<br>6.Read HBY coefficient |
| **29th** | HBY_23 | 1.Write FIFO: HBY 2nd out<br>2.Read two banks RAM |

| | | |
|---|---|---|
| | | 3.Read HBY coefficient<br>4.MAC: sum of 2nd two banks and coefficient and the accumulation is 0 |
| **30th** | HBY_23 | 1.Read two banks RAM<br>2.Read HBY coefficient<br>3.MAC: sum of 2nd two banks and coefficient and the accumulation is the 29th MAC result<br>4.Register the 29th MAC result |
| **31st** | HBY_23 | Same as last operation |
| **32nd** | HBY_23 | Same as last operation |
| **33rd** | HBY_23 | Same as last operation |
| **34th** | HBY_23 | Same as last operation |
| **35th** | GAIN_23 | 1.MAC: sum of 34th two banks and coefficient and the accumulation is the 33rd MAC result<br>2.Register the previous MAC result<br>3.Read the final HBY process sample (1 coefficient sample, 1st sample of HBY_23 result) |
| **36th** | GAIN_23 | 1. 1st sample of HBY_23 gain multiply<br>2. Register the 2nd sample of HBY_23 result |
| **37th** | | 1.Write 1st sample gain result to FIFO<br>2.2nd sample gain multiply |
| **38th** | | 1.Write the 2nd sample gain result to FIFO |

### 9.6.10 Other Details

1) Calculate cycles:
   a) DAC: Assume it's from 192KHz to 48KHz. One output sample need 4 input samples, 2 HB2 operation and 1 HB1 operation can get one output sample. The total cycles: 2*8+19+4 (Gain) = 35+4 cycles. So, the minimal clock frequency is: 48*39 = 1,872 KHz (per channel)
   b) ADC: Assume input is 48KHz, output is 192KHz. One 48KHz input sample can generate 4 192KHz samples. There are one HB1 operation and two HB2 operation. So, the total is 2*8+19+4 (Gain) = 39 cycles. The minimal frequency is: 48*39=1,872 KHz (per channel)
   c) For ADC+DAC, total 4 channel, the minimal calculation frequency is: 1872*4=7488 KHz
2) Gain ramp: if gain address not equal current, update it. But, increase and decrease is by 1 step, ramp to target gain.
3) ADC and DAC gain step and coefficient is different, so storage and address need process independently
4) Data Formats:
   a) Audio stream: from/to FIFO is 24-bit; when be used to calculate, [24:6] is truncated. Use S1.18 to indicate its format. MSB is the sign-bit
   b) Gain: S1.18 ??
      i) The maximum attitude is 6dB, so 20logx=6 =>x=1.99526. So, the integer need 1 bit. 19-bit gain coefficient can be formatted to S1.18 (-1 to 1.99999)
   c) HBX:&HBY: S1.18??
      i) The coefficient is normalized to 1. So, the range should be [-1,1]. So the integer is 1. 19-bit coefficient can be formatted to S1.18
5) Calculation flow (general flow):
   a) Even samples add: 19-bit+19-bit=20-bit (S1.18+S1.18=S2.18)

b) Multiplier: 20-bit*19-bit=39-bit (S2.18*S1.18=S3.36)
c) Accumulation for one sample: expand_39b(mul_reult)+expand_39b(pre_acc)=mac_out_40b (S3.36+S3.36=S4.36)
d) Saturate to 19-bit:
    i) HBX&HBY: [36:18]
    ii) Gain: [36:18]

# 10 I2C Host Slave Interface

## 10.1 General Description

The I2C Slave interface in Oceanside is used by an external device, such as a hosting CPU, to access registers within the Oceanside HDA register block. The interface supports I2C clock speeds up to 400 KHz.

## 10.2 Theory of Operation

### 10.2.1 I2C Bus Protocol

The following describes the I2C bus protocol, as supported by the Oceanside device.
The Oceanside device always acts as an I2C bus slave.
I2C utilizes just 2 signals, SCLK (serial clock) and SDA (serial data). The SCLK signal is always driven by the bus master, while the SDA signal is bidirectional.
Transfers are always initiated by the bus master with a 'START' sequence. Transfers are terminated by the bus master with a 'STOP' sequence. A transfer may be optionally appended to another with a 'RE-START' sequence, avoiding the need to transmit a STOP followed by a START.
All I2C transfers are byte oriented. Each byte, whether transmitted by the master or the Oceanside device, is transmitted as 8 consecutive bits, and is followed by a 1-bit ACK indication, which is transmitted by the receiver of the transmitted byte. (The bus direction is always reversed for ACK transmission.) Each byte is transmitted most-significant-bit first. Consecutive data bytes within a transfer are ordered from lower address to higher address.
The first byte of every transfer is transmitted by the bus master and contains a 7-bit device ID and a 1-bit indication of direction (read vs. write).
In the case of a write transfer, the following 2 bytes are also transmitted by the bus master, and contain the MSB and LSB (respectively) of a sub-address value. The sub-address value is to be used by the Oceanside device to access the appropriate location within its internal address space. The 2-byte sub-address value may be optionally followed by a series of 1 or more data bytes, transmitted by the bus master to the bus slave. The Oceanside device utilizes the given sub-address value to determine where within its address space the data bytes are written. The Oceanside device supports auto-incrementing of the sub-address value between written data bytes, as well as non-incrementing operation.
In the case of a read transfer, no sub-address value is transmitted by the bus master. By convention, a write transfer is utilized to establish a sub-address value prior to the initiation of a read transfer. Therefore, immediately following the device ID of a read transfer, the Oceanside device will begin transmitting a series of 1 or more data bytes. It will continue to do so until a STOP sequence is detected.
The following figures illustrate the I2C protocol, as supported by the Oceanside device. Elements indicated in **black** are transmitted by the bus master. Elements shown in **red** are transmitted by the Oceanside device.

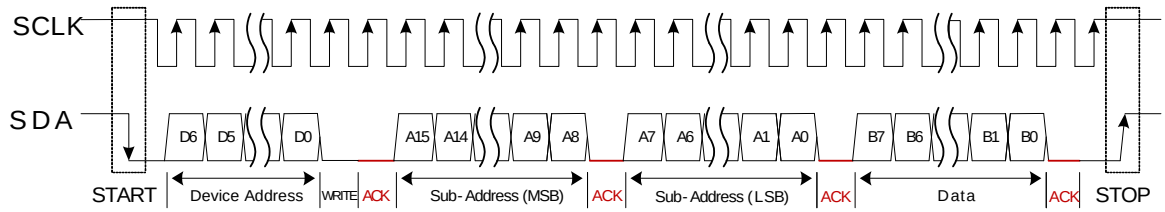



**Figure 10.2-1 I2C Write Transfer**

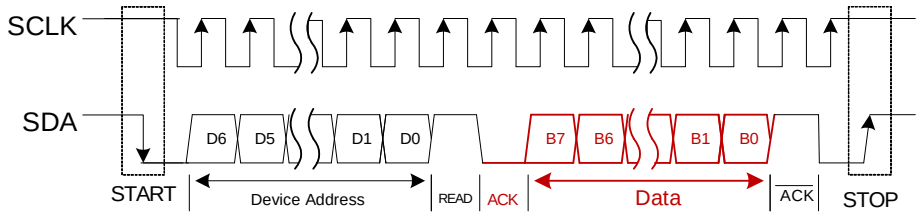



**Figure 10.2-2 I2C Read Transfer**

### 10.2.2 I2C Clock Speed

The I2C interface supports clock speeds up to 400 KHz.

### 10.2.3 I2C Device ID

The default I2C Device ID is selected at power-on reset by a pullup/high value or pulldown/low value on the Alternate Device ID Select pin. The selectable default IDs are as follows:

| Device ID Select Level | Selected ID |
|---|---|
| Pull- up (high) | 7'h37 (alternate address) |
| Pull- down (low) | 7'h33 (default) |

The default I2C Device ID may be read through the I2C_SLAVEID register in the I2C Slave Interface module. This register is also writeable, allowing the default ID to be overridden.

| Interface Access | I2C SlaveID Address |
|---|---|
| HDA | 20'h1B414 |
| I2C | 6E14 |

## 10.3 SoC Integration and Micro Architecture

### 10.3.1 Module Usage

The I2C host interface is a slave interface used to access device registers. The interface comprised of the HSCL (Host I2C SCL) and HSDA (Host I2C SDA) pins. These are dedicated pins that interface with the host SOC or other controller.

1) The I2C slave interface is always active
    a. This allows a guaranteed means of switching bus modes.
    b. It may be difficult to manage disabling access where we won't create a condition where we get stuck in one mode or another.
2) All commands are available to the I2C and HD Audio interfaces.
3) Typically, one control interface is active while the other is idle or off.

a. In I2S mode, only the I2C interface can be active.
b. In HD Audio mode both can be active.

4) If a request comes from both the I2C and HD Audio interfaces at the same time:
   a. Both interfaces will meet their timing requirements. Valid data must be read.
   b. No attempt will be taken to prevent one interface from overwriting data from the other. Accesses to a register are on a first come first served basis. In the case of an absolute tie, it is expected that the I2C access will be held off until the HD Audio access is complete due to the more restrictive HD Audio timing requirements.

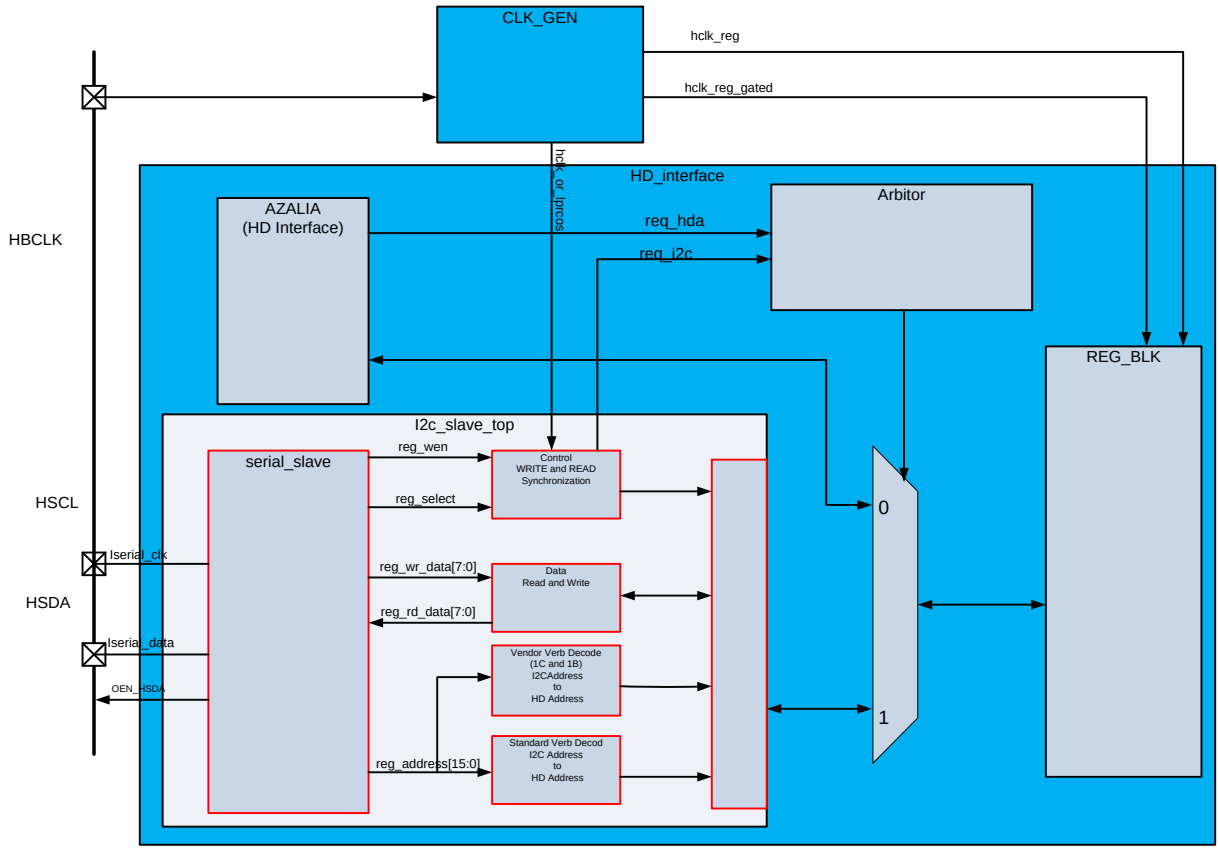



**Figure 10.3-1 I2C Module Integration Block Diagram**

## 10.3.2 I/O Specifications

### 10.3.2.1 I2C Register Interface

Table 12-1 shows I2C module register interface To Oceanside register block.

| Signal Name | Direction | Default | Function |
|---|---|---|---|
| Reg_address[15:0] | output | 0 | I2C register address |
| Reg_select | output | 0 | I2C chip select :<br>1: register select enabled<br>0: register select disabled. |
| Reg_wen | output | 0 | I2C Register write/read enable:<br>1: register read<br>0: register write |
| Reg_wr_data[7:0] | output | 0 | I2C Register write Data |
| Reg_rd_data[7:0] | input | 0 | I2C register Read Data |

**Table 10.3-1 I2C Register Interface Signals**

### 10.3.2.2 Functional description

The I2C register interface module, decodes the I2C address/Data into an HD (NodeID, Verb, Param) type word.

The interface is implemented as a standard Read/Write control and enable. Data written to module registers is latched on the rising edge of reg_select and synchronized to register clock (16-bit Data is held internally until the second -byte is written). Data read from module registers is supplied synchronously when link is down (BCLK is off), and asynchronously when HD clock is available. In I2S mode, the read operation is always synchronous with SCLK.

Note: it is important that when in HD MODE, the Bclk has to be minimum of 6MHz (BCLK). The Read operation is not guaranteed, If BCLK less than 6MHz while in HDA mode (Read is a real time operation).

Note: Writing 16-bit Data (1C Vender register space) requires two consecutive byte writes because the 1st byte will be held in a holding buffer until the 2nd byte is written.

### 10.3.2.3 I2C PAD (I/O) Slave Interface

| Signal Name | Direction | Default | Function |
|---|---|---|---|
| SCLK | I | 0 | I2C Clock input |
| SDA_i | I | 0 | I2C Data input |
| SDA_o | O | 0 | I2C Data output |
| SDA_oe | O | 0 | I2C Data output enable |

**Table 10.3-2 I2C PAD Slave Interface Signal List**

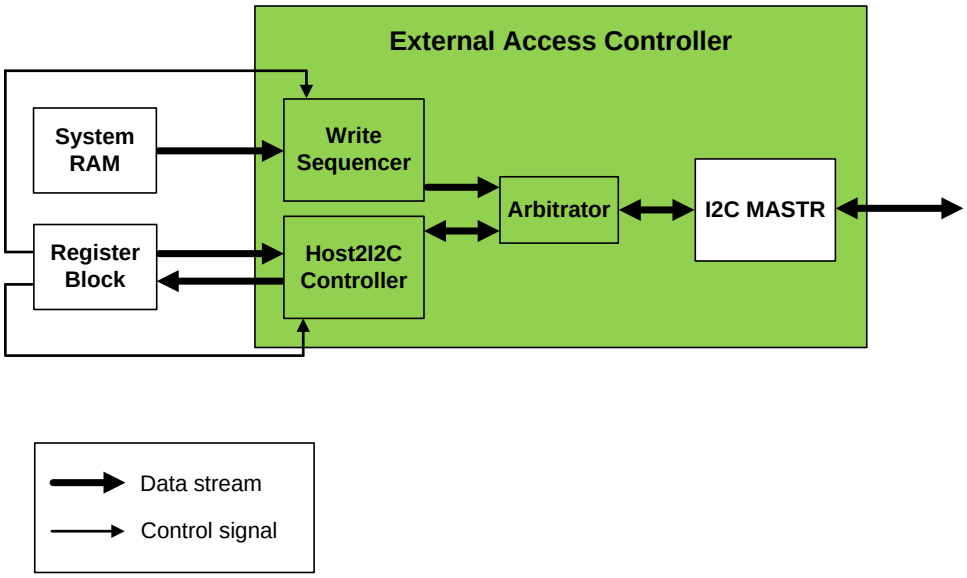
# 11 I2C Peripheral Master Interface

## 11.1 General Description

An I2C master interface is needed to control external peripherals. This interface operates at least up to 400 KHz though faster is better. The controller may be accessed from either the HD Audio or host I2C interface. It is desired to have efficient methods for bridging from HD Audio or I2C to I2C so that the bus bandwidth is maximized. System start-up delay is critical and external smart amplifiers may require considerable configuration.

To reduce the burden on the host to program and reprogram external peripherals (smart amps) during power state changes, a write sequencer may be used to store one or more configuration sequences that write a series of commands to the external device in response to a single command from the host (or autonomously). The expectation is that the host will program the sequencer initially which will require a delay but after initial programming the host need only to start the sequencer and may proceed with other resume activities while the CODEC configures the external device.

The external access controller (EAC) block integrates the host2i2c controller, write sequencer and I2C master core. The block diagram is showed as following.



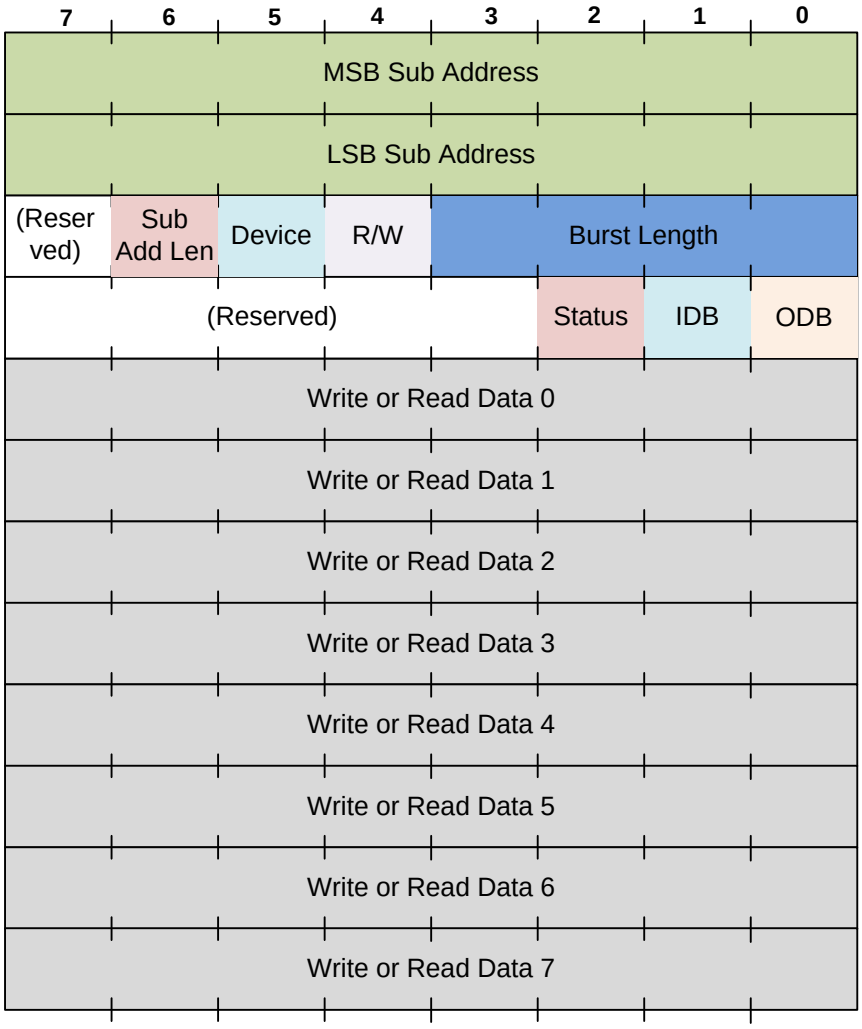

## 11.2 Theory of Operation

The host2i2c controller receives command from host (by HDA or I2C slave), and operates the I2C master core to write or read the peripheral device. The command and data locate in register block.

There are two sets of registers for host2i2c controller, so it can work by Ping-Pong mode.

The detailed frame format is showed as following.

## Host2I2C controller Frame Format
## (Located in Register Block)




**Name: I2C_MASTER_REG1**
**HDA Address: 0x01bd_2000**
**I2C Address: 0xXX20**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7 | i2c_clk_en | RW | RO | 0 | Enable I2C clock. Active high. |
| 1:0 | i2c_speed | RW | RO | 1 | Sets the speed mode for I2C transfer.<br>2'b00: 100KHz<br>2'b01: 400KHz<br>2'b10: 1MHz<br>2'b11: Reserved. |

**Name: I2C_MASTER_REG2**
**HDA Address: 0x01bd_2100**
**I2C Address: 0xXX21**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 6:0 | dev_addr0 | RW | RO | 0 | Sets the address for device #1. |

**Name: I2C_MASTER_REG3**
**HDA Address: 0x01bd_2200**
**I2C Address: 0xXX22**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 6:0 | dev_addr1 | RW | RO | 0 | Sets the address for device #2. |

**Name: HOST2I2C_CTRL0_REG1**
**HDA Address: 0x01bd_2600**
**I2C Address: 0xXX26**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | msb_sub_addr | RW | RO | 0 | Sets the sub-address (the most significant byte) for host2i2c controller. |

**Name: HOST2I2C_CTRL0_REG2**
**HDA Address: 0x01bd_2700**
**I2C Address: 0xXX27**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | lsb_sub_addr | RW | RO | 0 | Sets the sub-address (the least significant byte) for host2i2c controller. |

**Name: HOST2I2C_CTRL0_REG3**
**HDA Address: 0x01bd_2800**
**I2C Address: 0xXX28**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 6 | sub_addr_len | RW | RO | 0 | Sets the length of sub-address.<br>1'b0: one byte sub-address<br>1'b1: two bytes sub-address. |
| 5 | device | RW | RO | 0 | Sets the device ID for I2C transfer.<br>1'b0: access device 1<br>1'b1: access device 2. |
| 4 | rd_wr | RW | RO | 0 | Sets the transfer type.<br>1'b0: write operation<br>1'b1: read operation. |

| | | | | | |
|---|---|---|---|---|---|
| 3:0 | burst_length | RW | RO | 0 | Sets the burst length for I2C transfer.<br>4'd0: 1 byte<br>4'd1: 2 byte<br>…<br>4'd7: 8 bytes<br>Others: Reserved, treated as 8 bytes for both. |
| **Name: HOST2I2C_CTRL0_REG4**<br>**HDA Address: 0x01bd_2900**<br>**I2C Address: 0xXX29**<br>**Default: 0x0** | | | | | |
| **bits** | **name** | **s/w** | **h/w** | **default** | **description** |
| 2 | status | RO | RW | 0 | Status information about current I2C transfer.<br>1'b0: successful transfer<br>1'b1: failed transfer. |
| 1 | idb | R/W1C | RW | 0 | Inbound door bell, which is set by host2i2c controller and cleaned by host when writing 1. Acts as a level signal. |
| 0 | odb | RW | RW | 0 | Outbound door bell, which is set by host and cleaned when setting idb field by host2i2c controller. Acts as a level signal. |
| **Name: HOST2I2C_CTRL0_REG5**<br>**HDA Address: 0x01bd_2a00**<br>**I2C Address: 0xXX2a**<br>**Default: 0x0** | | | | | |
| **bits** | **name** | **s/w** | **h/w** | **default** | **description** |
| 7:0 | wr_rd_dat0 | RW | RW | 0 | Sets the write data #1 under write operation by host, or the read data #1 under read operation by host2i2c controller. |
| **Name: HOST2I2C_CTRL0_REG6**<br>**HDA Address: 0x01bd_2b00**<br>**I2C Address: 0xXX2b**<br>**Default: 0x0** | | | | | |
| **bits** | **name** | **s/w** | **h/w** | **default** | **description** |
| 7:0 | wr_rd_dat1 | RW | RW | 0 | Sets the write data #2 under write operation by host, or the read data #2 under read operation by host2i2c controller. |
| **Name: HOST2I2C_CTRL0_REG7**<br>**HDA Address: 0x01bd_2c00**<br>**I2C Address: 0xXX2c**<br>**Default: 0x0** | | | | | |
| **bits** | **name** | **s/w** | **h/w** | **default** | **description** |
| 7:0 | wr_rd_dat2 | RW | RW | 0 | Sets the write data #3 under write operation by host, or the read data #3 under read operation by host2i2c controller. |

**Name: HOST2I2C_CTRL0_REG8**
**HDA Address: 0x01bd_2d00**
**I2C Address: 0xXX2d**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | wr_rd_dat3 | RW | RW | 0 | Sets the write data #4 under write operation by host, or the read data #4 under read operation by host2i2c controller. |

**Name: HOST2I2C_CTRL0_REG9**
**HDA Address: 0x01bd_2e00**
**I2C Address: 0xXX2e**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | wr_rd_dat4 | RW | RW | 0 | Sets the write data #5 under write operation by host, or the read data #5 under read operation by host2i2c controller. |

**Name: HOST2I2C_CTRL0_REG10**
**HDA Address: 0x01bd_2f00**
**I2C Address: 0xXX2f**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | wr_rd_dat5 | RW | RW | 0 | Sets the write data #6 under write operation by host, or the read data #6 under read operation by host2i2c controller. |

**Name: HOST2I2C_CTRL0_REG11**
**HDA Address: 0x01bd_3000**
**I2C Address: 0xXX30**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | wr_rd_dat6 | RW | RW | 0 | Sets the write data #7 under write operation by host, or the read data #7 under read operation by host2i2c controller. |

**Name: HOST2I2C_CTRL0_REG12**
**HDA Address: 0x01bd_3100**
**I2C Address: 0xXX31**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | wr_rd_dat7 | RW | RW | 0 | Sets the write data #8 under write operation by host, or the read data #8 under read operation by host2i2c controller. |

**Name: HOST2I2C_CTRL1_REG1**
**HDA Address: 0x01bd_3200**
**I2C Address: 0xXX32**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | msb_sub_addr | RW | RO | 0 | Sets the sub-address (the most significant byte) for host2i2c controller. |

**Name: HOST2I2C_CTRL1_REG2**
**HDA Address: 0x01bd_3300**
**I2C Address: 0xXX33**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | lsb_sub_addr | RW | RO | 0 | Sets the sub-address (the least significant byte) for host2i2c controller. |

**Name: HOST2I2C_CTRL1_REG3**
**HDA Address: 0x01bd_3400**
**I2C Address: 0xXX34**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 6 | sub_addr_len | RW | RO | 0 | Sets the length of sub-address.<br>1'b0: one byte sub-address<br>1'b1: two bytes sub-address. |
| 5 | device | RW | RO | 0 | Sets the device ID for I2C transfer.<br>1'b0: access device 1<br>1'b1: access device 2. |
| 4 | rd_wr | RW | RO | 0 | Sets the transfer type.<br>1'b0: write operation<br>1'b1: read operation. |
| 3:0 | burst_length | RW | RO | 0 | Sets the burst length for I2C transfer.<br>4'd0: 1 byte<br>4'd1: 2 byte<br>…<br>4'd7: 8 bytes<br>Others: Reserved, treated as 8 bytes for both. |

**Name: HOST2I2C_CTRL1_REG4**
**HDA Address: 0x01bd_3500**
**I2C Address: 0xXX35**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 2 | status | RO | RW | 0 | Status information about current I2C transfer.<br>1'b0: successful transfer<br>1'b1: failed transfer. |
| 1 | idb | R/W1C | RW | 0 | Inbound door bell, which is set by host2i2c controller and cleaned by host when writing 1. Acts as a level signal. |
| 0 | odb | RW | RW | 0 | Outbound door bell, which is set by host and cleaned when setting idb field by host2i2c controller. Acts as a level signal. |

**Name: HOST2I2C_CTRL1_REG5**
**HDA Address: 0x01bd_3600**
**I2C Address: 0xXX36**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | wr_rd_dat0 | RW | RW | 0 | Sets the write data #1 under write operation by host, or the read data #1 under read operation by host2i2c controller. |

**Name: HOST2I2C_CTRL1_REG6**
**HDA Address: 0x01bd_3700**
**I2C Address: 0xXX37**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | wr_rd_dat1 | RW | RW | 0 | Sets the write data #2 under write operation by host, or the read data #2 under read operation by host2i2c controller. |

**Name: HOST2I2C_CTRL1_REG7**
**HDA Address: 0x01bd_3800**
**I2C Address: 0xXX38**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | wr_rd_dat2 | RW | RW | 0 | Sets the write data #3 under write operation by host, or the read data #3 under read operation by host2i2c controller. |

**Name: HOST2I2C_CTRL1_REG8**
**HDA Address: 0x01bd_3900**
**I2C Address: 0xXX39**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | wr_rd_dat3 | RW | RW | 0 | Sets the write data #4 under write operation by host, or the read data #4 under read operation by host2i2c controller. |

**Name: HOST2I2C_CTRL1_REG9**
**HDA Address: 0x01bd_3a00**
**I2C Address: 0xXX3a**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | wr_rd_dat4 | RW | RW | 0 | Sets the write data #5 under write operation by host, or the read data #5 under read operation by host2i2c controller. |

**Name: HOST2I2C_CTRL1_REG10**
**HDA Address: 0x01bd_3b00**
**I2C Address: 0xXX3b**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | wr_rd_dat5 | RW | RW | 0 | Sets the write data #6 under write operation by host, or the read data #6 under read operation by host2i2c controller. |
| **Name: HOST2I2C_CTRL1_REG11**<br>**HDA Address: 0x01bd_3c00**<br>**I2C Address: 0xXX3c**<br>**Default: 0x0** | | | | | |
| **bits** | **name** | **s/w** | **h/w** | **default** | **description** |
| 7:0 | wr_rd_dat6 | RW | RW | 0 | Sets the write data #7 under write operation by host, or the read data #7 under read operation by host2i2c controller. |
| **Name: HOST2I2C_CTRL1_REG12**<br>**HDA Address: 0x01bd_3d00**<br>**I2C Address: 0xXX3d**<br>**Default: 0x0** | | | | | |
| **bits** | **name** | **s/w** | **h/w** | **default** | **description** |
| 7:0 | wr_rd_dat7 | RW | RW | 0 | Sets the write data #8 under write operation by host, or the read data #8 under read operation by host2i2c controller. |

## 11.3 Operating Flow

A burst writing operation works as following.

Step #1: Enable I2C clock, and sets the speed mode for I2C transfer (100K, 400K or 1M);

Step #2: Host sets sub-address, burst length, R/W flag, device id and writing data based on the burst length;

Step #3: Host sets ODB to start the writing transfer;

Step #4: Host checks the IDB until it is set to be 1; checks the status to decide how to operate next;

Step #5: Finish this transfer.

A burst reading operation works as following.

Step #1: Enable I2C clock, and sets the speed mode for I2C transfer (100K, 400K or 1M);

Step #2: Host sets sub-address, burst length, R/W flag and device id;

Step #3: Host sets ODB to start the writing transfer;

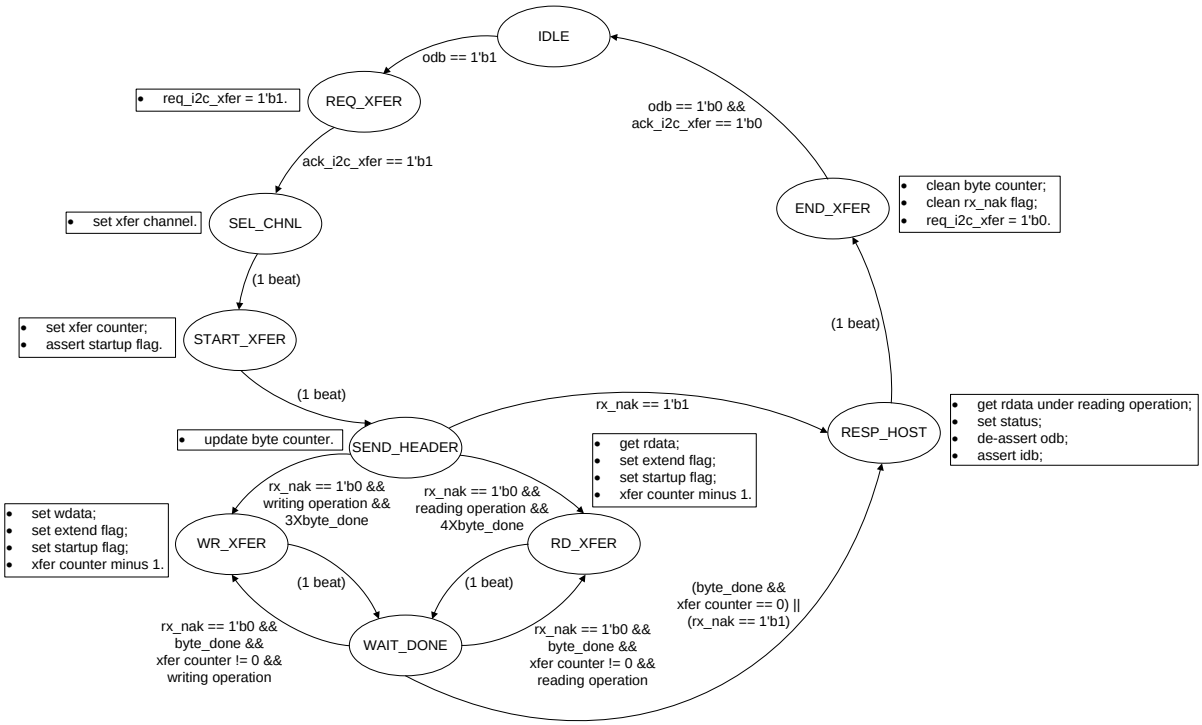
Step #4: Host checks the IDB until it is set to be 1; checks the status to decide how to operate next;

Step #5: Host gets the reading data when the status is successful transfer;

Step #6: Finish this transfer.

## 11.4 FSM

The detailed design plan is showed as following.




Conexant Systems Inc
Proprietary and Confidential
ALL INFORMATION IN THIS DOCUMENT IS PRELIMINARY AND SUBJECT TO CHANGE WITHOUT NOTICE

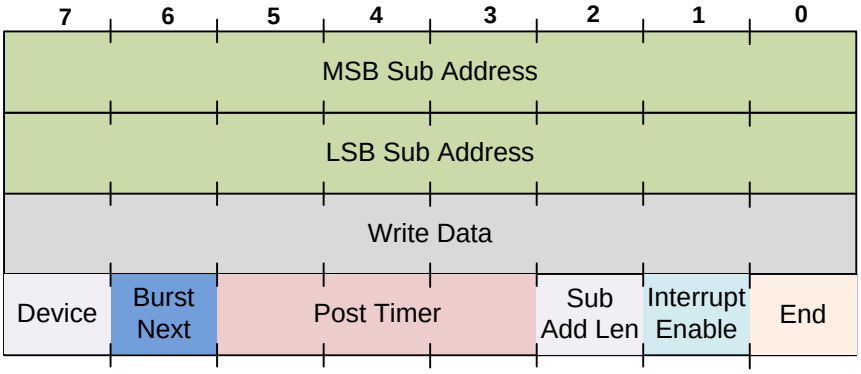
# 12 I2C Write Sequencer

To reduce the burden on the host to program and reprogram external peripherals (smart amps) during power state changes, a write sequencer may be used to store one or more configuration sequences that write a series of commands to the external device in response to a single command from the host (or autonomously). The expectation is that the host will program the sequencer initially which will require a delay but after initial programming the host need only to start the sequencer and may proceed with other resume activities while the CODEC configures the external device.

## 12.1 Operating Principle

The write sequencer receives commands from the sequencer RAM block (NID 28), and operates the I2C master core to write or read the peripheral device. Commands and data are in the sequencer RAM, and some control signals are from the register block.
The detailed frame format is shown as follows (See NID28 Widget for read/write verbs and I2C addresses):

**Write Sequencer Frame Format**
**(Located in RAM)**




| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 31 | device | RW | RO | 0 | Sets the device ID for I2C transfer.<br>1'b0: access device 1<br>1'b1: access device 2. |
| 30 | burst_nxt | RW | RO | 0 | Indicates the next command is a burst transfer with this one. |
| 29:27 | post_timer | RW | RO | 0 | Sets the wait time when finishing this transfer. |
| 26 | sub_addr_len | RW | RO | 0 | Sets the length of sub-address.<br>1'b0: one-byte sub-address<br>1'b1: two-byte sub-address. |
| 25 | int_en | RW | RO | 0 | Indicates there is an interruption for host when finishing this transfer. |
| 24 | end | RW | RO | 0 | Indicates the last byte transfer for write sequencer. |

| | | | | | |
|---|---|---|---|---|---|
| 23:16 | wr_dat | RW | RO | 0 | Sets the write data for write sequencer. |
| 15:8 | lsb_sub_addr | RW | RO | 0 | Sets the sub-address (the least significant byte) for write sequencer. |
| 7:0 | msb_sub_addr | RW | RO | 0 | Sets the sub-address (the most significant byte) for write sequencer. |

**Name: WRITE_SEQUENCER_REG1**
**HDA Address: 0x01bd_2300**
**I2C Address: 0xXX23**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:2 | timer_period[13:8] | RW | RO | 0 | Sets the timeout unit value for write sequencer. The basic unit of this timer is 68ns (14.7456MHz). |
| 1 | i2c_int | R/W1C | RW | 0 | Sets the interrupt state. Set by write sequencer and cleaned by host when writing 1. |
| 0 | start_wr | RW | RO | 0 | Startup signal for write sequencer. Set and cleaned by host. Acts as a level signal. |

**Name: WRITE_SEQUENCER_REG2**
**HDA Address: 0x01bd_2400**
**I2C Address: 0xXX24**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | ini_add | RW | RO | 0 | Sets the first address of RAM access for write sequencer. |

**Name: WRITE_SEQUENCER_REG3**
**HDA Address: 0x01bd_2500**
**I2C Address: 0xXX25**
**Default: 0x0**

| bits | name | s/w | h/w | default | description |
|---|---|---|---|---|---|
| 7:0 | timer_period[7:0] | RW | RO | 0 | Sets the timeout unit value for write sequencer. The basic unit of this timer is 68ns (14.7456MHz). |

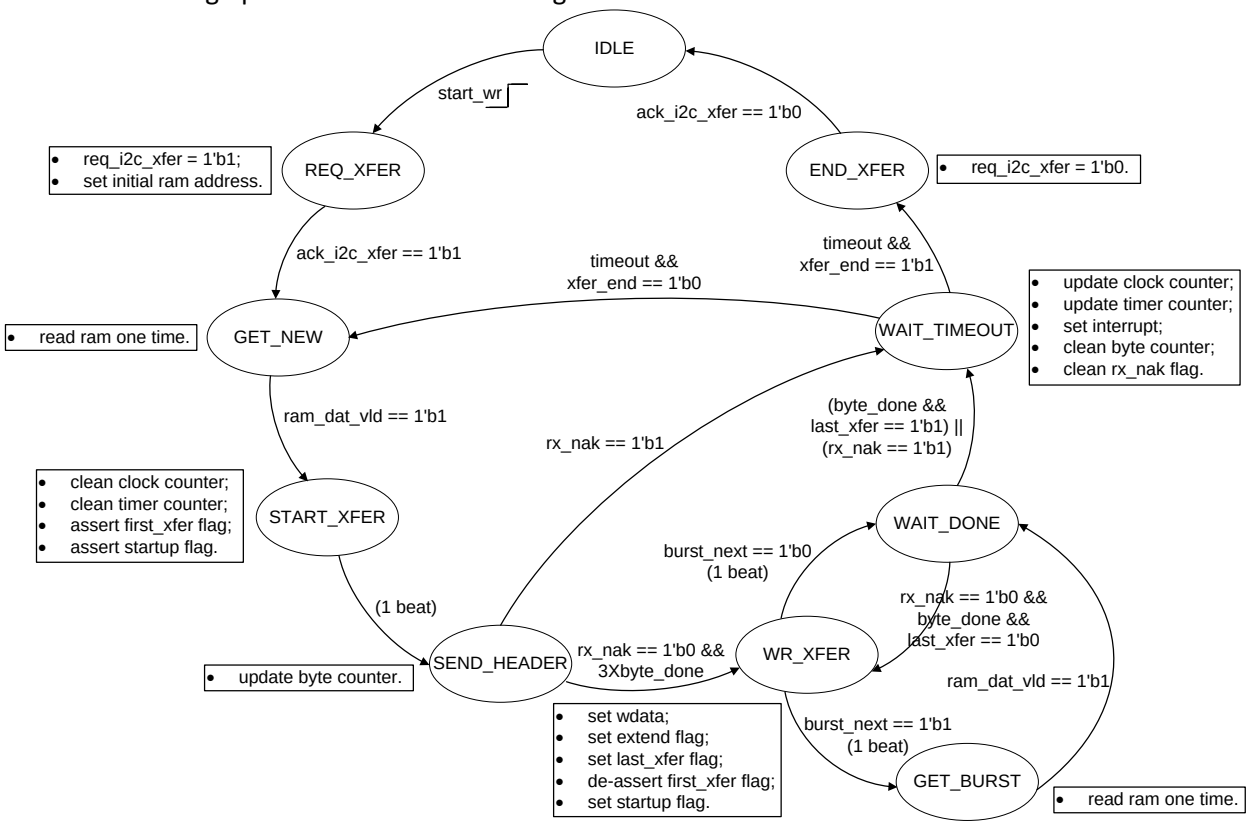
## 12.2 Operating Flow

The operation works as following.

Step #1: Enable I2C clock, and sets the speed mode for I2C transfer (100K, 400K or 1M);
Step #2: Host writes all the command into the sequencer RAM;
Step #3: Host sets the start_wr field to start the write sequencer;
Step #4: Write sequencer operates the I2C master core to access the peripheral device;
Step #5: Write sequencer interrupts host when finishing all the transfer.

## 12.3 FSM

The detailed design plan is showed as following.



# 13 DAC Envelope Detector

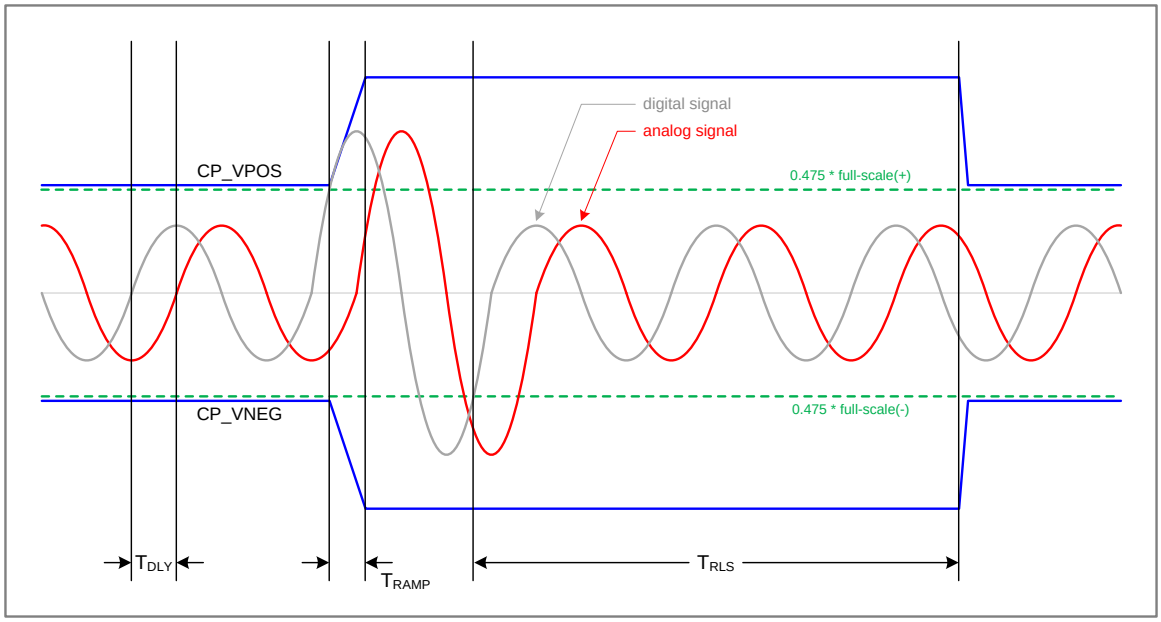
Oceanside device uses the envelope detector that was implemented for the Fuji device with following changes in the Design.

1- Buffer delay changed to 25us.
2- Buffer Delay Memory was changed from (104x19 for Solana-B2) to 22x20 for Oceanside.
3- Bit width changed to 20-bit with bit-0 not used.
4- Add 2.5db compensation. (Because analog AGC has 2.5db compensation but not feedback to ED, so ED need shift the threshold for this 2.5db compensation).
5- Oceanside does not use the bias level adjust feature.
6- Threshold Table Changes: Oceanside

| CPV_THRESH[1:0] | FUJI threshold value | Solana threshold value |
|---|---|---|
| 00 | 0.500 * full-scale (=0.707 Vpeak) | 0.350 * full-scale (=0.495 Vpeak) |
| 01 | 0.475 * full-scale (=0.672 Vpeak) | 0.475 * full-scale (=0.672 Vpeak) |
| 10 | 0.450 * full-scale (=0.636 Vpeak) | 0.450 * full-scale (=0.636 Vpeak) |
| 11 | 0.400 * full-scale (=0.565 Vpeak) | **0.400 * full-scale (=0.565 Vpeak) default** |

## 13.1 DAC Envelope Detector Architecture

The DAC envelope detector (ED) monitors the envelope of the incoming signal and may be used to adjust both the charge-pump voltage level and the biasing level (Oceanside does not use this feature, adjusting bias level) for the analog circuitry in the Class-H headphone amplifier appropriately. The ED performs comparisons on signals for both channels that determine when the thresholds for switching the charge-pump voltage level and the biasing level have been crossed. To allow time for the charge-pump to ramp when an incoming signal exceeds the threshold for requiring higher charge-pump voltage, the ED also contains a buffer that delays all incoming signals long enough that the ramping will complete before the signal is output from the ED. Reduction of the charge-pump voltage or the biasing level requires that the incoming signals remain below the associated threshold until a release timer expires.




Envelope Detector Control of Charge-Pump Voltage

Operation of the envelope detector for charge-pump voltage control is illustrated in the figure above. When the signal level is within the threshold limits (in figure, 0.475 * ±full-scale), the charge-pump voltage rails are set to half of the input supply. When the signal amplitude increases above the threshold, the supplies are immediately switched to full level of the input supply. To avoid clipping, the charge-pump ramp time must be less than the buffer delay within the ED.

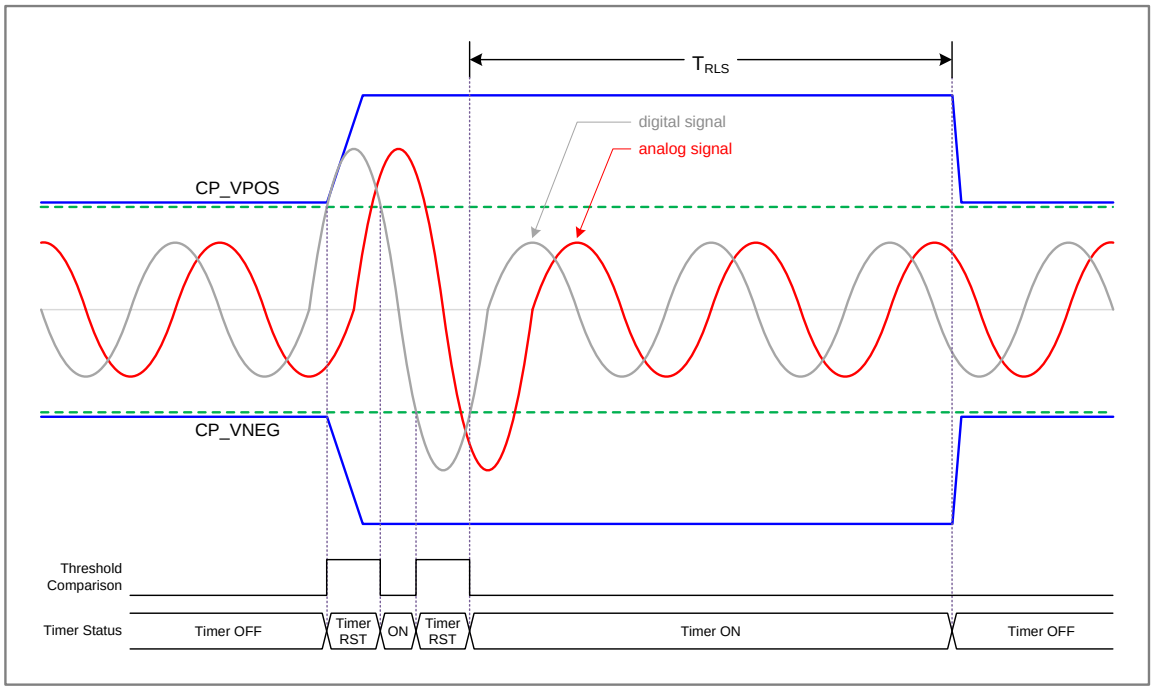
Once the signal falls back within the threshold range and remains there long enough for the release timer to expire, the charge-pump voltages are set back to half of the input supply. The release time is long enough to provide reasonable assessment that the signal level has dropped and to ensure that the charge-pump doesn't chatter between the 1.0x and 0.5x modes.

Control of the charge-pump voltage level and the Class-H bias level by the ED are independent and use separate thresholds and release timers. One of four threshold levels may be selected for controlling the charge-pump level, while a fixed threshold is used for controlling the bias level. The signal threshold comparisons made by the ED account for both digital PGA gain and analog gain implemented in class-AB stage of class-H headphone outputs. Use of the ED to control these levels is optional and is enabled by setting the DAC_ENV_ENABLE bit in the DAC Envelope Detect register.

Parameters of the ED are defined as follows. The buffer delay, $T_{DLY}$, is 26 µs for 48k rate family and 28 µs for 44.1k rate family. This delay is present when the ED is enabled, regardless of the current signal levels. The maximum charge-pump ramp time, $T_{RAMP}$, in Fuji is less than 85 µs. The release time, $T_{RLS}$, is configurable to 16, 32 or 64 ms, with this setting shared for both voltage and bias control. The table below shows the four selectable thresholds for charge-pump voltage control and the one fixed threshold for bias control.

| Threshold Name | CPV_THRESH[1:0] | Threshold Value |
|---|---|---|
| Charge-Pump Voltage Threshold | 00 | 0.500 * ±full-scale |
| | 01 | 0.475 * ±full-scale |
| | 10 | 0.450 * ±full-scale |
| | 11 | 0.400 * ±full-scale |
| Class-H Bias Threshold | n/a | 0.125 * ±full-scale |

The behavior of the release timers is illustrated in the diagram below for the charge-pump voltage control. The release timer is off when in the half voltage state. The release timer is reset when the signal exceeds the threshold range and begins counting when the signal returns within the threshold range. If the signal exceeds the threshold range before the timer expires, the release timer is immediately reset again and restarts counting only when the signal falls back within the threshold range. When the release timer expires, the charge-pump returns to the half voltage state.




Release Timing for Charge-Pump Voltage Control

### 13.1.1 Class-H Amplifier Design Notes

The charge-pump for Class-H amplifiers in Fuji supports two voltage levels:

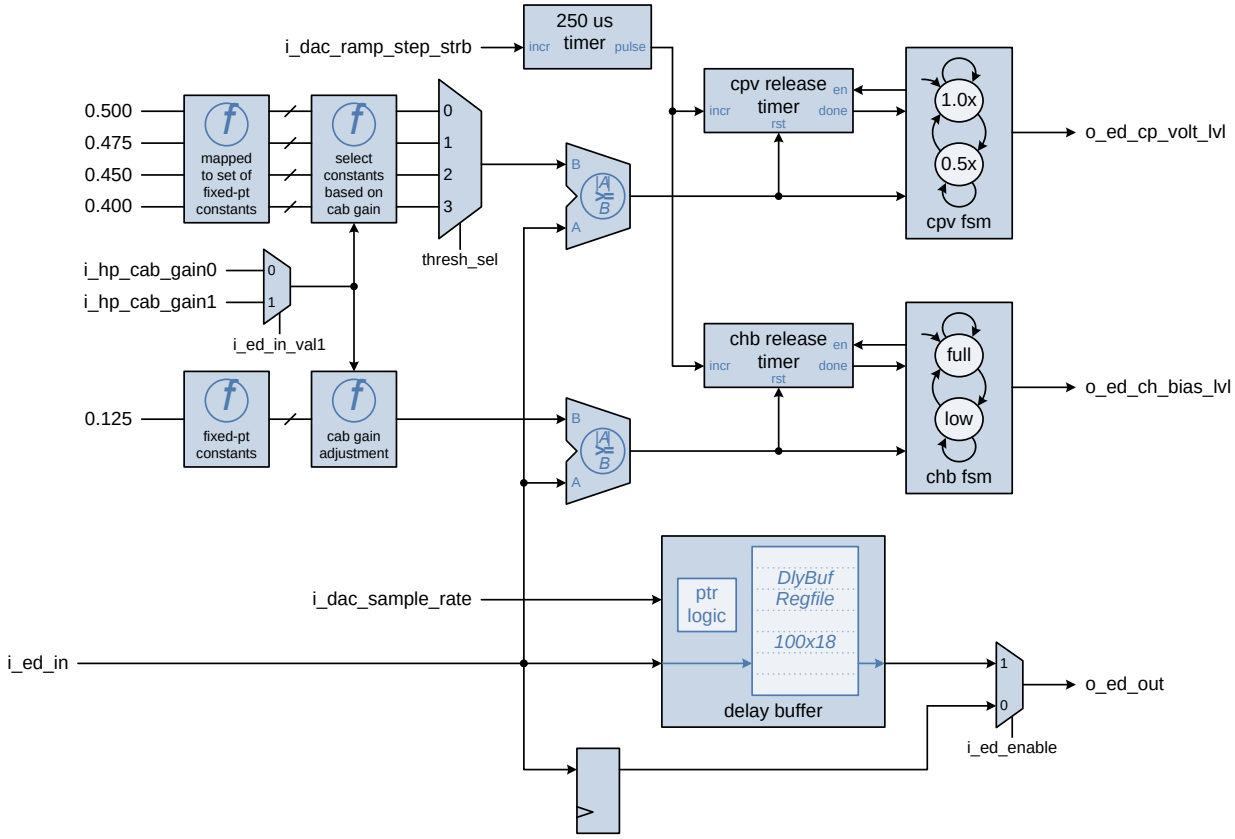
(1) 0.5x: CPVOUTP=CPVDD/2 when signal is below full-scale/2
(2) 1.0x: CPVOUTP=CPVDD when signal exceeds full-scale/2

Ramping from half voltage to full voltage takes less than 85µs.

The Class-H circuitry requires two biasing levels: low and full. The low-level biasing option was needed in Ellis to meet power goals for low-level signals (ex. playing back zeros). This potentially may not be necessary in Fuji, depending on analog design changes, but currently needs to supported. Signals below full-scale/8 may go into low-level biasing mode. The ramping time for biasing changes is almost instantaneous.

## 13.2 DAC Envelope Detector Implementation

The block diagram below illustrates the basic data path of the envelope detector block.




The design includes separate FSMs, threshold comparisons, and release timers for charge-pump voltage control and class-H bias control. A 250 us timer is shared. A delay buffer delays all incoming data when the ED is enabled. The FSMs for charge-pump voltage control (cpv fsm) and class-H bias control (chb fsm) that are both two-state FSMs.

The CPV FSM has states for full (1.0x) and half (0.5x) voltage levels. The CPV FSM transitions from 0.5x to 1.0x state when the CPV threshold range is exceeded. The CPV FSM transitions from 1.0x to 0.5x when the release timer expires. The initial state is 1.0x.

The CHB FSM has states for full and low bias levels. The CHB FSM transitions from LOW to FULL when the CHB threshold range is exceeded. The CHB FSM transitions from FULL to LOW when the release timer expires. The initial state is FULL.

The two threshold comparisons compensate for both digital PGA and analog class-AB gain settings. Since digital PGA gain is applied in either DRC or in half-band filter block, locating the envelope detector after these blocks means that digital gain is automatically accounted for in the threshold comparisons. To adjust for analog class-AB gain, each possible threshold setting is mapped to a set of different threshold values, with the appropriate value selected based on the current class-AB gain setting. The internal set of values is reduced using fact that +6dB shift in gain requires dividing threshold value approximately in half. Note that if the analog class-AB gain is set to -8 dB or lower, no signal can exceed the minimum charge-pump threshold setting of 0.40, and if the analog gain is set to -18 dB or lower, no

signal can exceed the bias control threshold of 0.125. The internal threshold values use narrower bit width than input data to reduce hardware cost with negligible impact on threshold accuracy (more conservative by less than -1% shift).

The number of samples buffered in the delay buffer depends on the sample rate. The following table shows the sample delay for each rate.

| Sample Rate (48k family) | Sample Rate (44.1k family) | ED Input SR (48k family) | ED Input SR (44.1k family) | Sample Delay |
|---|---|---|---|---|
| 48k | 44.1k | 384k | 352.8k | 10 |
| 96k | 88.2k | 384k | 352.8k | 10 |
| 192k | 176.4k | 384k | 352.8k | 10 |

These sample delays produce 125 us delay for sample rates from 48k rate family and 136 us delay for rates from 44.1k rate family. Note that timing of design is driven off the signals that incoming data is valid, which leads to write-before-read behavior. For this reason, the design is simplest if the memory used includes at least one more word than the maximum sample delay. Given the restricted set of memories from Ellis that were available, this worked out nicely with 100x18 register file. Note that the release times are much longer than the buffer delay so all samples in the buffer will be below the associated threshold when a release timer expires.

Besides removing the need to adjust thresholds for digital PGA gain, locating the envelope detector after the half-band filter is crucial to reduce both the maximum error and the lag in threshold comparisons. The up-sampling performed in the half-band filter block means that the signal output from this block is sampled at 8 times the Nyquist sample rate requirement for the maximum frequency content of the source signal (i.e. min 16 samples/T(fmax)). Performing the envelope detection on an up sampled version of the source signal is essential for avoiding unacceptably large error between the peak observed value of the sampled signal and maximum value of the reconstructed continuous time signal. The maximum error between peak sampled value and true peak value would occur if two contiguous samples were equally spaced before/after $\pi$ /2 phase point in an incoming sine wave source signal. Considering a 24kHz tone at 48kHz source sample rate and 384kHz sample rate out of half-band filter, there would be 16 samples per cycle and worst error would occur if samples were located ±1/2 * 2$\pi$/16 around $\pi$/2:

$$\sin\left(\frac{\pi}{2}-\frac{2\pi}{32}\right)=\sin\left(\frac{\pi}{2}+\frac{2\pi}{32}\right)=0.9808 \quad (error \sim 2\%)$$

Remembering that passband for 48 kHz sample rate is defined as flat to 20 kHz and considering a 19.2kHz tone at this rate, there would be 20 samples per period and the error reduces to:

$$\sin\left(\frac{\pi}{2}-\frac{2\pi}{40}\right)=\sin\left(\frac{\pi}{2}+\frac{2\pi}{40}\right)=0.9877 \quad (error \sim 1.2\%)$$

There is also a potential lag in detecting that a signal has crossed the threshold that is equal to the sampling period of the observed signal. The minimum sample rate for the output of the half-band filter is 64 kHz (8*8kHz source sample rate). This equates to a maximum lag of 15.625 us, which can be accommodated by the margin in the buffer delay.

## 13.3 SoC Integration

Diagram Below shows ED integration in Oceanside device. An Input Mux selects which interpolator Data will the input to data buffer and an output mux will select which Sinc will receive data from Envelope detector.

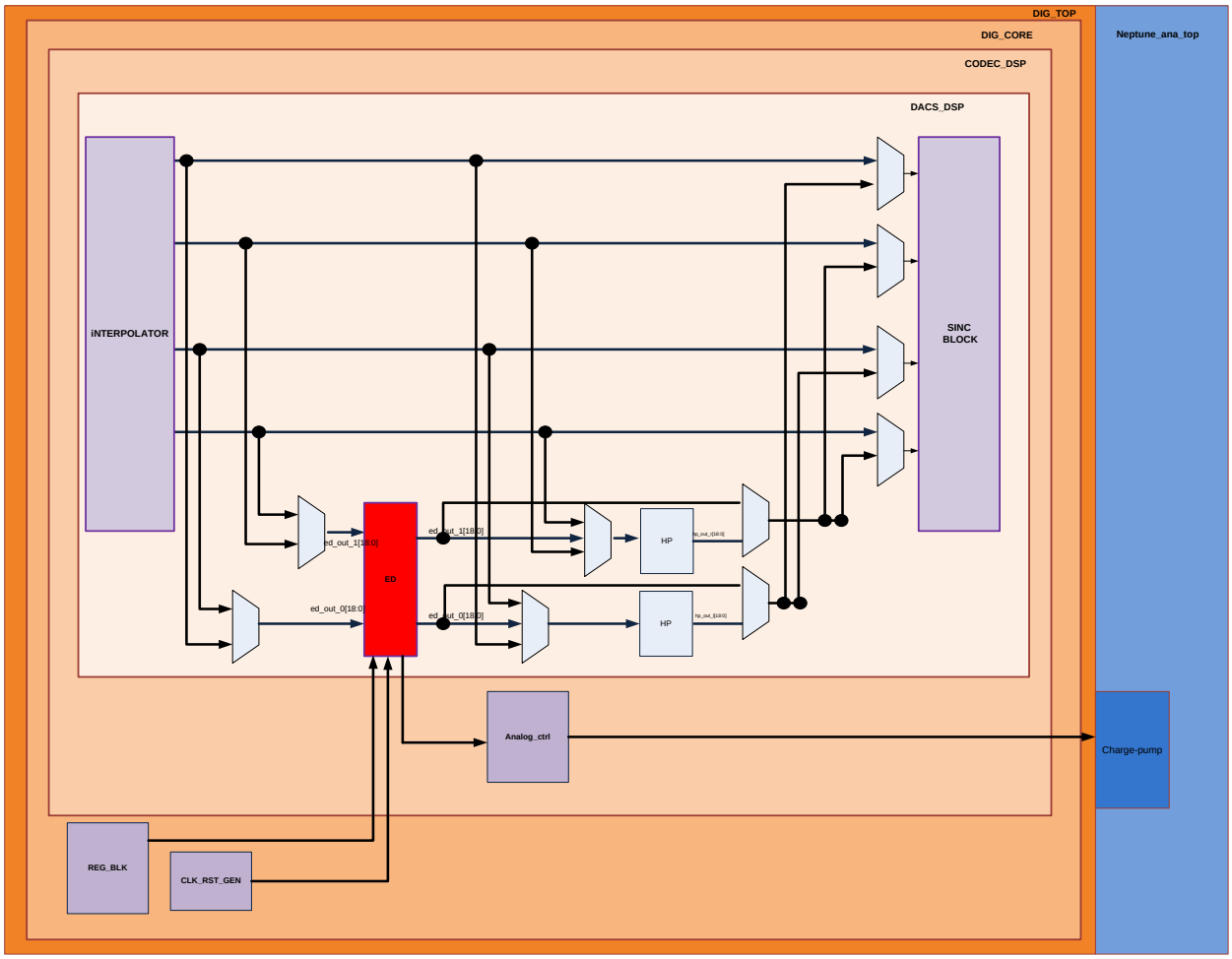



**Figure 13.3-1 ED SoC integration Diagram**

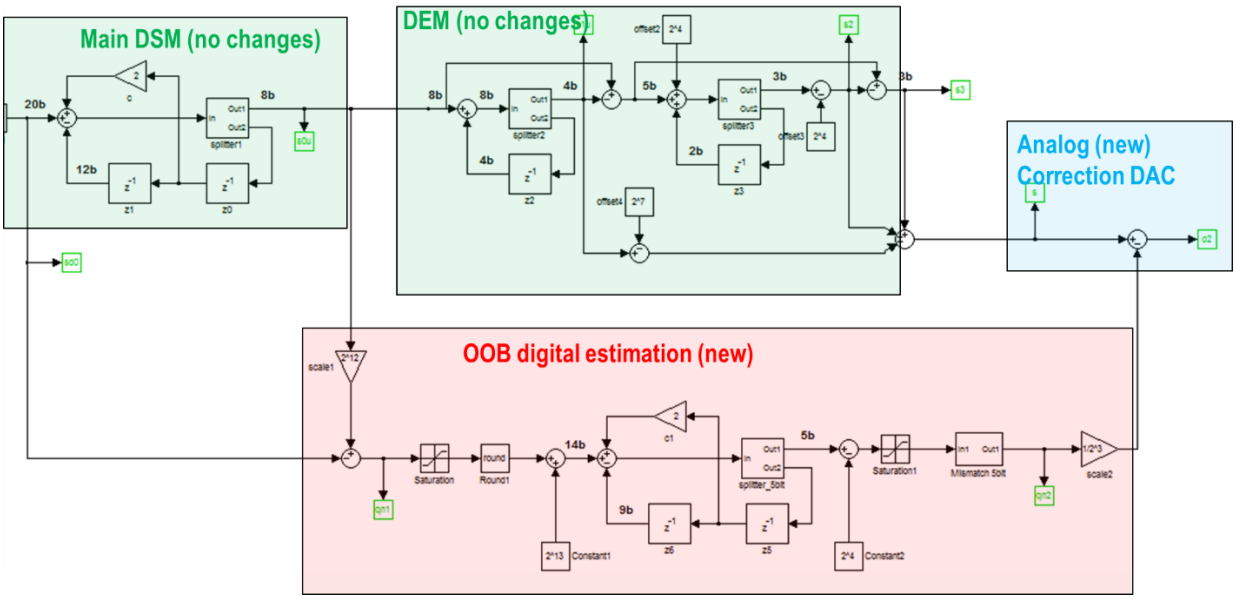
# 14OOB Digital Design

This Chapter Describes the OOB Digital Design.
Reference Document in Neptune Share Point:

- Neptune – OOB.
- Neptune OOB NC DAC Implementation Issues.
- Neptune OOB Noise Improvement.

## 14.1 General Description

This design purpose is using a small auxiliary DAC, dedicated to cancelling the shaped quantization noise (out of band (OOB) noise):

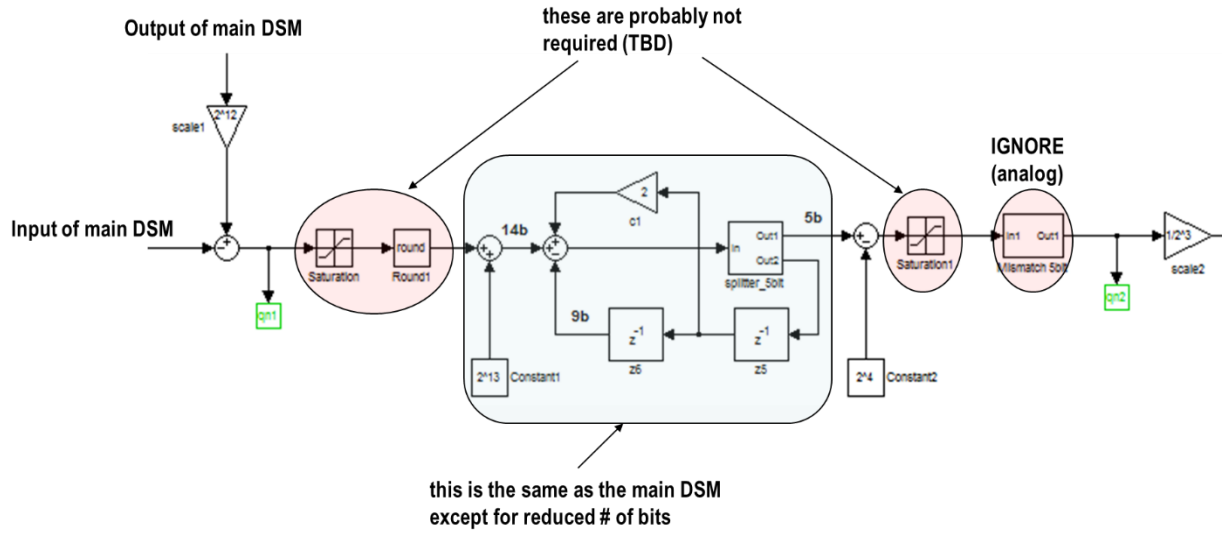



More background, check the analog spec.

## 14.2 Programmer's Reference

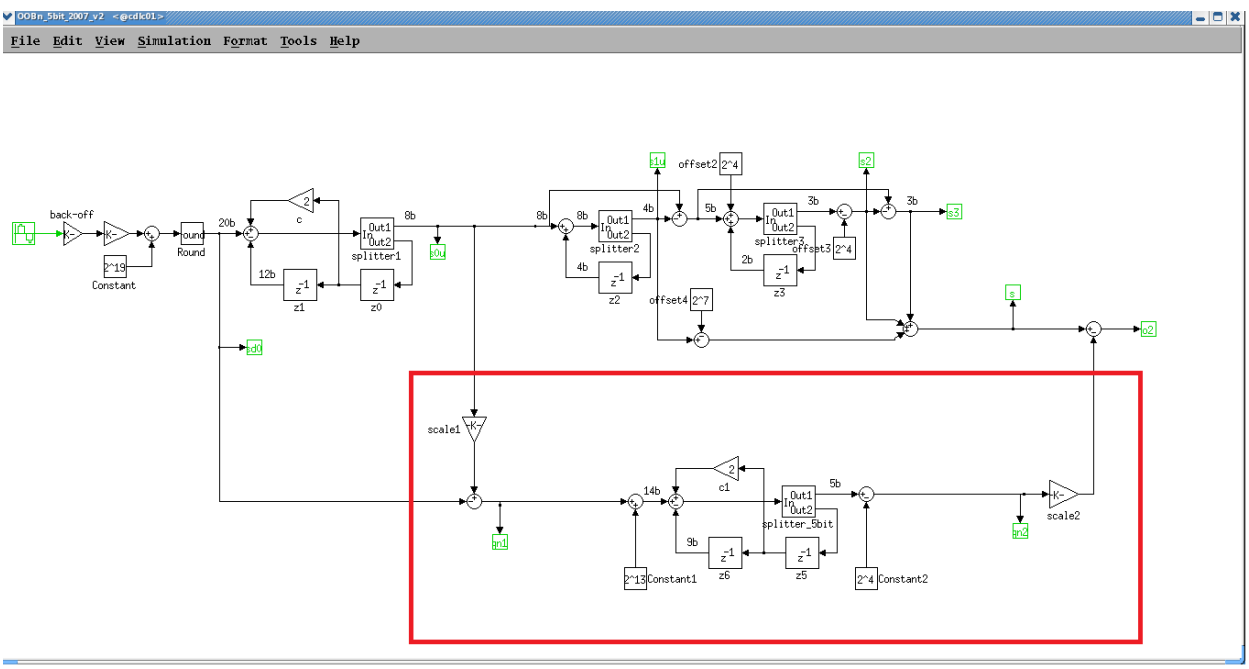
### 14.2.1 Theory of Operation

When enable the OOB, do the reduced bits DSM.




The matlab simlink code:

OOBn_5bit_2007_v2.mdl




define DSMINW 20
`define DSMOW 8
`define OOBFB  9
`define OOBW  14
`define DSMFB 12
`define OOBOW 5

Conexant Systems Inc
Proprietary and Confidential
ALL INFORMATION IN THIS DOCUMENT IS PRELIMINARY AND SUBJECT TO CHANGE WITHOUT NOTICE

```
input          rst_n,
input          clk,
input          dsm_en,
input          oob_disable,
input  [`DSMINW-1:0] dsm_in,
input  [`DSMOW-1:0]  dsm_out,

output reg [`OOBOW-1:0]  oob_out
```

And there's final map for D/A interface with additional: -16 map tp 11111

Use sign-and-magnitude format at D/A interface

| Decimal | D/A interface csdac1_l_oob[4:0] |
|---|---|
| 15 | 01111 |
| 14 | 01110 |
| 13 | 01101 |
| 12 | 01100 |
| 11 | 01011 |
| 10 | 01010 |
| 9 | 01001 |
| 8 | 01000 |
| 7 | 00111 |
| 6 | 00110 |
| 5 | 00101 |
| 4 | 00100 |
| 3 | 00011 |
| 2 | 00010 |
| 1 | 00001 |
| 0 | 00000 |
| -1 | 10001 |
| -2 | 10010 |
| -3 | 10011 |
| -4 | 10100 |
| -5 | 10101 |
| -6 | 10110 |
| -7 | 10111 |
| -8 | 11000 |
| -9 | 11001 |
| -10 | 11010 |
| -11 | 11011 |
| -12 | 11100 |
| -13 | 11101 |
| -14 | 11110 |
| -15 | 11111 |

## 14.3 Register Spec

### 14.3.1 CODEC_TEST11

***Get Codec Test11 Response - HD_A [27:0] =0x1CAB000***
***Set Codec Test11 - HD_A[27:11] = 0x1C2B***

| Bit | Default | Description |
|---|---|---|
| 11 | 00 | Disable the auxmode loopback data path<br>0= In SnM mode aux mode loop back path through codec from ADC engine to DAC engine is available<br>1= In SnM mode aux mode loop back data path through codec from ADC engine to DAC is not available.<br>If need to path the data through the FIFO |

| | | |
|---|---|---|
| 10 | 0 | Volume/Boost place: When DRC in Manual Mode set this bit 1 for Volume Ramp up/down<br>1: put on new volume controller<br>0: not on new volume controller |
| 9 | 0 | Enable aux mode loop back path from ADC to DAC even if SnM is not enabled.<br>0= ADC to DAC loop back through codec is enabled in non SnM mode<br>1= ADC to DAC loop back through codec is enabled in non SnM mode |
| 8 | 1 | Disable fix for DSM fifo write/read at same address issue.<br>0 = enable the fix<br>1 = disable the fix |
| 7 | 0 | Selects clk_au from bclk/rcosc(depending on AFG low power mode) for generating spkr clk and other logic in analog control block<br>0 – bclk/rcosc<br>1 – PLL/PLL_DIV_2 (selected using ctest12[11]) |
| 6 | 0 | Select PLL_DIV_2 or BCLK/RCOSC (depending on AFG low power mode) for DC detection and power averaging<br>0 = bclk/rcosc<br>1 = pll_div_2 |
| 5 | 0 | OOB disable control for DAC2<br>0: enable OOB<br>1: disable OOB |
| 4 | 0 | OOB disable control for DAC1<br>0: enable OOB<br>1: disable OOB |
| 3 | 0 | 1-bypass enable for clock detection logic with a programmable value from bit 2 of this register<br>0-clock detection h/w logic in place |
| 2 | 0 | 1-clock not available indication to analog I/F when bypass bit [3] is set<br>0-Clock available indication to analog I/F when bypass bit [3] is set. |
| 1:0 | 00 | Class-D Gain boost<br>00 = +3db<br>01 = +4db<br>10 = +5db<br>11 = +6db |

# 15 Multi-Button/TR impedance Design

This Chapter describes the multi-button Inline support and headphone Tip/Ring impedance detection digital design in Deep Eddy device.
Reference Documents in Deep Eddy Share Point:
Deep Eddy – MB_TR.
Deep Eddy - Spec Analog Controller.
Neptune – Multi Button testing (Analog Folder)

## 15.1 Overview

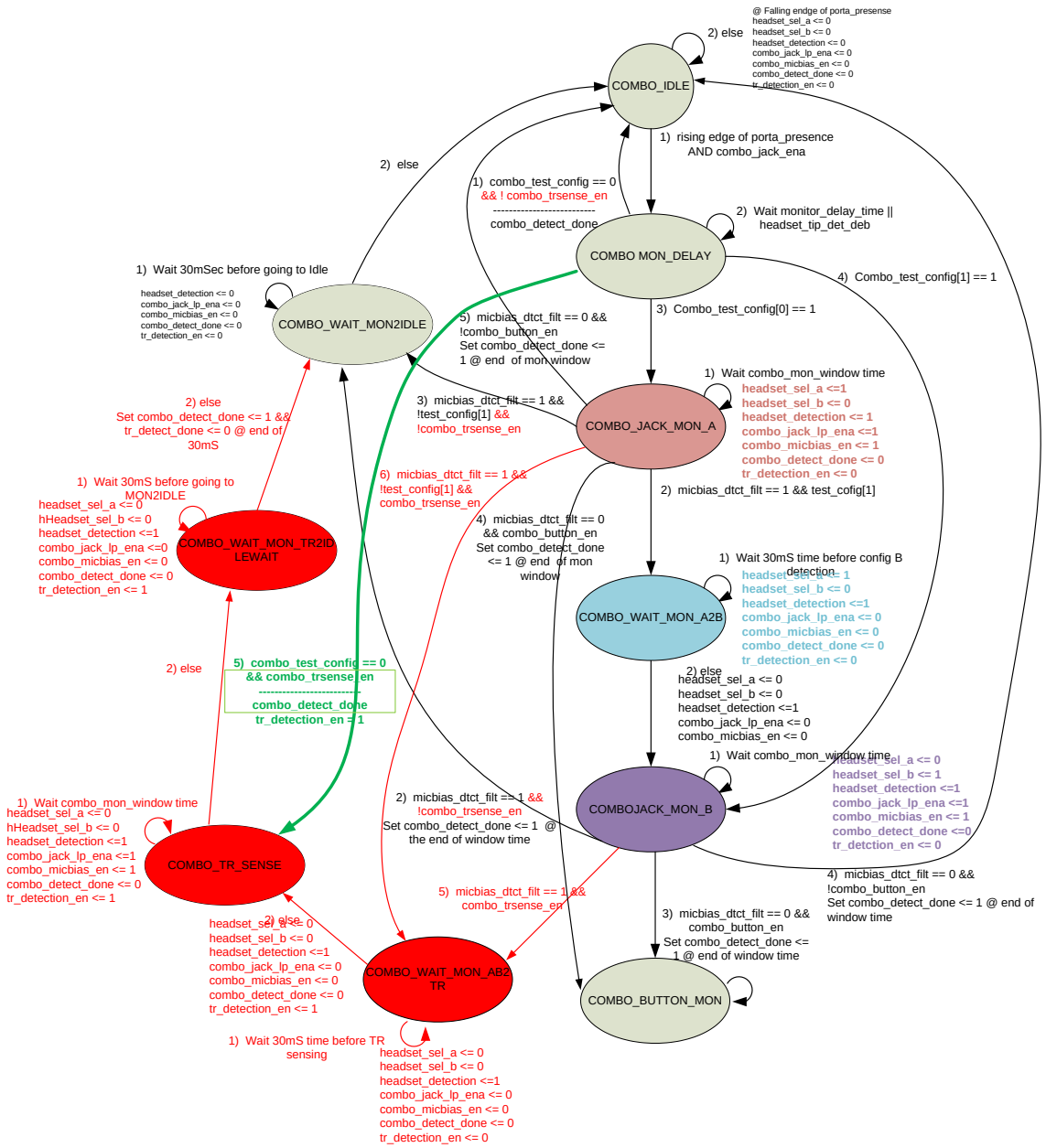
Communications headsets commonly provide control buttons integrated into the headset. The most common implementation is a single button that shorts the microphone input to ground. This button is typically used for call answer, call end, mute, and unmute depending on the duration of the button press. Many headsets provide two additional buttons for volume up/down and other functions. These buttons connect a resistance across the microphone to change the voltage measured on the mic pin. These resistances may be linear devices such as resistors or nonlinear devices such as diodes.

The Cx807x/Cx812x will implement support for single button detection or multi-button detection. The single-button detection circuit will detect short and long switch closures and generate unsolicited responses (including wake events) for button press and release events.

Multiple button support is targeted for 4 buttons as described by Google. The multi-button detection scheme should also support 3 buttons as an option. The detection circuit will detect short and long switch closures and generate unsolicited responses (including wake events) for button press and release events.

## 15.2 FSM for TR Sense and Multi-Button Detection

**Figure 15.2-1 FSM for Headset Detection**



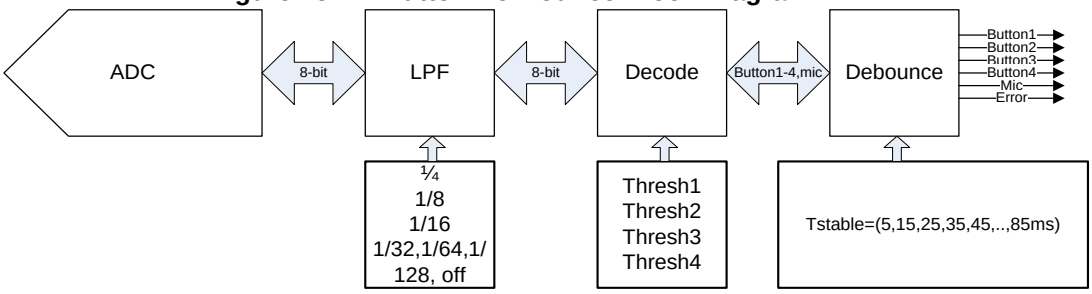

* From any state, upon falling edge of porta_presence, the FSM will reset to IDLE state. All controls will be initialized to reset value.

This FSM support the detection of multi-button & impedance sense of headphone: For TR Sensing, add a state on headset detection state machine when headphone detected. In this state, digital send tr_sense_enable to analog, then analog send back the result signal to indicate the impedance less than 1K ohm or not. Actually, there're 2 additional states before & after the COMBO_TR_SENSE state that make sure the wait time (30ms) before/after enable the tr_detection_en & combo_micbias_en(enable the micbias & tr detect in AFE side).

For Multi-Button detection, in COMBO_BUTTON_MON status. (when headset detected & button enable), once micbias_dtct become one (means button pushed), SAR ADC start working to output 8bit value. This 8bit value go to a Low-pass filter to filter out the signals below the audio frequency (normally 20Hz, depend on the LPF coefficient setting). Then depend on the threshold setting for 4 buttons, decoder logic output which button be pressed then after debounce logic for the stable button pressed event.

Conexant Systems Inc
Proprietary and Confidential
ALL INFORMATION IN THIS DOCUMENT IS PRELIMINARY AND SUBJECT TO CHANGE WITHOUT NOTICE

**Figure 15.2-2 Button De-Bounce Block Diagram**




For more background, check the analog spec for Multi-button & TR sensing.

## 15.3 Programmer's Reference

### 15.3.1 Theory of Operation

When jack insert, headset detection triggered, if it's headphone, state machine will go to the COMBO_TR_SENSE state. In this state, digital side enable the micbias & set tr_detection_en to AFE. AFE will do the TR detection, & send back the compare result (if < 1K ohm). At the end of this state, digital side will capture this result to register & can be readable by host. Then the state machine will be back to IDLE finally.

When headset detected with button, state machine will go to the COMBO_BUTTON_MON state. In this state, once micbias_dtct assert (after debounce), it means there's button pressed.

This signal will enable the SAR ADC & multi-button detection process:

(1). Multi-button processes run pipeline when micbias_dtct_filt(filtered micbias_dtct) = 1
& Jack state machine in COMBO_BUTTON_MON state & mb_enable = 1(register bit)

(2). When any button detected (B1-4, MIC), DN bit set & UM/Interrupt triggered.

(3). Any B1-4, MIC, DN, UP change from 0 to 1, latch to register block, host can write 1 to clear (W1C)

(4). Status =EVENT_COUNT, LP, MIC, B4, B3, B2, B1, UP, DN

(5). LP (Long Press) counter start from micbias_dtct_filt = 1. When reach the set time (configured by register), LP bit assert.

(6). When micbias_dtct_filt(after debounce) de-assert, UP bit set.

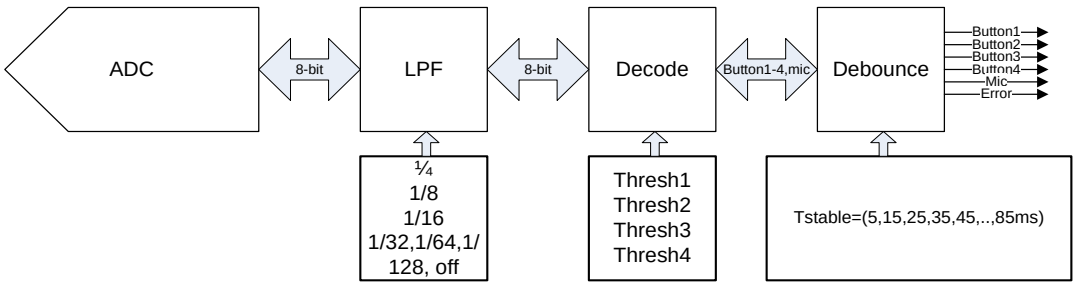
(7). There's event counter for record button multi-press before host response for the UM/Interrupt to read the status. Any posedge of DN (means a new press detected), event counter + 1, host W1C for any status bit can reset this event counter.

Basically, when host received the UM/Interrupt, it can read the status to know the button situation & then write 1 to clear it for next detection.

| Bit | Default | Description |
|---|---|---|
| 11 | RO | HP_RES_LS_1K |
| 10:8 | W1C | MB event count. (count on DN posedge).<br>When W1 below bits also clear the count value). |
| 7 | W1C | MB_LP |
| 6 | W1C | MB_MIC |
| 5 | W1C | MB_B4 |
| 4 | W1C | MB_B3 |
| 3 | W1C | MB_B2 |

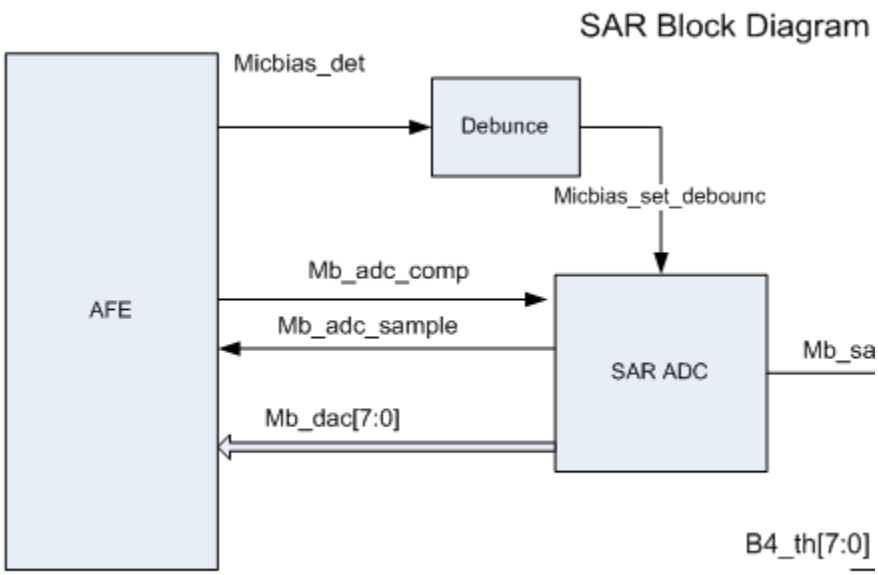
| | | |
|---|---|---|
| 2 | W1C | MB_B1 |
| 1 | W1C | MB_UP |
| 0 | W1C | MB_DN |

### 15.3.2 Data Flow Diagram

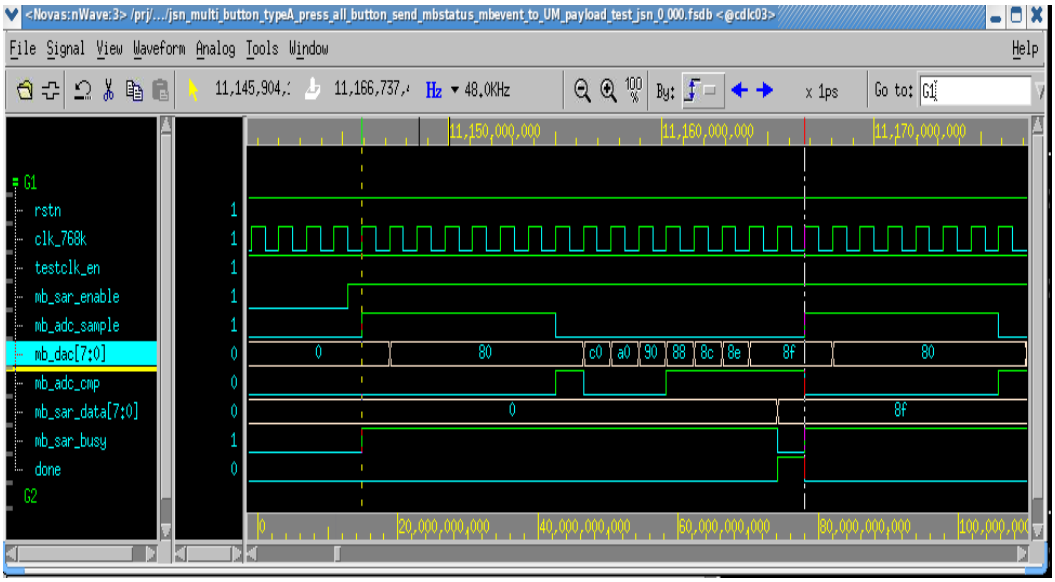


### 15.3.3 Module Usage

**Figure 15.3-1 SAR ADC Block Diagram**



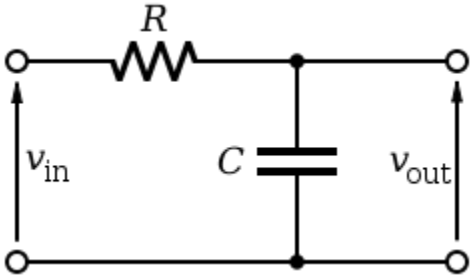

1. 768khz clock, when micbias_dtct_filt = 1 start.
2. Sample & output every 48khz.
3. cycles sample, 8 cycles conv, 1 cycles done signal, total 16 cycles




```
output mb_adc_sample, // to AFE
input    mb_adc_cmp,   // from AFE
output [7:0] mb_adc,    // to AFE
```

LPF (Low Pass Filter) for SAR output:

It's simple RC filter:




we may need the LPF to filter-out the freq > 20Hz.

For RC-filter:

Y[0] = X[0];

Y[n] = Y[n-1] + a * (X[n] –Y[n-1]);

Fc = 1 / (2 * Pi * RC): Fc is the cutoff freq.

RC = Dt * (1-a) / a

Dt = 1 / Fs.  : Fs is the sample rate.

If a = 1/N, n= 16, 32, 64, 128, 256, 512

Then, Fc = Fs / (2 * Pi * (N -1)).

When N = 128, Fs = 48K, the Fc ~= 60Hz.

When N = 256/512,  Fc ~= 30/15Hz.

| Bit | Default | Description |
|---|---|---|
| 11:9 | 0x4 | RC LPF coeff : a = dt/(dt+RC)<br>0: Bypass<br>1: a = 1/16<br>2: a = 1/32<br>3: a = 1/64<br>4: a = 1/128<br>5: a = 1/256<br>6,7 a = 1/512 |
| 8 | 0x0 | Reserved. |
| 7:0 | 0x2A | B4_TH<br>B4 Threshold |

The decoding for the button:

```
assign b1_cmp = (rc_filt_data < mb_b1_th);
assign b4_cmp = (rc_filt_data < mb_b4_th) & (rc_filt_data >= mb_b1_th);
assign b2_cmp = (rc_filt_data < mb_b2_th) & (rc_filt_data >= mb_b4_th);
assign b3_cmp = (rc_filt_data < mb_b3_th) & (rc_filt_data >= mb_b2_th);
assign mic_cmp = (rc_filt_data >= mb_b3_th);
```

The b1_th, b2_th, b3_th, b4_th come from registers.

Conexant Systems Inc
Proprietary and Confidential
ALL INFORMATION IN THIS DOCUMENT IS PRELIMINARY AND SUBJECT TO CHANGE WITHOUT NOTICE

Basically, we assume b1_th < b4_th < b2_th < b3_th. The mapping is:

```
0-------------b1_th--------------b4_th--------------b2_th---------------b3_th------------------max
|<----B1------>|
              |<-------B4-----→|
                             |<-------B2------→|
                                            |←-------B3----→|
                                                         |<--------MIC-----→|
```

If set the b1_th = b4_th, we can get the 3 buttons (B1, B2, B3, MIC) detection.
If set b1_th = b4_th = b2_th, we can get the 2 buttons (B1, B3, MIC) detection.
For single button detection, we can just disable the multi-button detection that will enable the single button detection.

| Bit | Default | Description |
|---|---|---|
| 11 | 0x1 | MB enable<br>0: single button mode<br>1: enable multi-button mode |
| 10 | 0x1 | Combo TR sense enable<br>1: enable<br>0: disable |
| 9 | 0x0 | MB status/event count enable in UM payload<br>1: enable in UM payload<br>0: disable (all 0s in payload). |
| 8 | 0x0 | MB LP send UM/INT<br>0: not send UM/INT for LP<br>1: send UM/INT for LP. |
| 7:0 | 0x14 | B1_TH<br>B1 Threshold |

## 15.4 Testability

Below list of signals that are brought out on test mux or via test registers.

- wire [6:0] group41_data = {1'b0,combo_trsense_en,combo_micbias_ena,micbiasd_dtct,headset_detection,tr_sense_compout,tr_detect_en};
- wire [6:0] group4_data = digital_test9[9] ? group41_data : group40_data;
- wire [6:0] group91_data = {mb_adc_sample, mb_adc_cmp, mb_dac[7:3]};
- wire [6:0] group9_data = digital_test9[9] ? group91_data : group90_data;
- assign tr_detect_en = tr_sw_dbg[1] ? tr_sw_dbg[0] : tr_detect_en_mux;
- wire [1:0] tr_sw_dbg = codec_test14[5:4];

## 15.5 Register Spec

Register summary and link to register Details is listed below.
Multi-Button/TR Registers

**Table 15.5-1 Register List (Node 1C)**

| Register Name | VERB[11:8] | Verb[7:0](addr) | Default |
|---|---|---|---|
| CODEC_TEST33[7:0] | [4/C] | 99 | 'hc14 |
| CODEC_TEST34[7:0] | [4/C] | 9a | 'h242 |
| CODEC_TEST35[7:0] | [4/C] | 9b | 'h67f |
| CODEC_TEST36[7:0] | [4/C] | 9c | 'h82a |
| CODEC_TEST37[7:0] | [4/C] | 9d | 0 |

Conexant Systems Inc
Proprietary and Confidential
ALL INFORMATION IN THIS DOCUMENT IS PRELIMINARY AND SUBJECT TO CHANGE WITHOUT NOTICE

# 16 EAPD and Speaker Mute

EAPD and Speaker Mute are two functions that enable the control of internal and external amplifiers. Speaker Mute can provide a way for an external controller to mute the internal amplifier while EAPD is a signal originally intended to control external amplifiers attached to an HD Audio CODEC. These two signals, however, may be configured in several different modes to provide flexibility in system implementation.

## 16.1 EAPD Pin

The EAPD pin is a dedicated, bi-directional control pin. As described in the HD Audio specification, EAPD (External Amplifier Power Down) is a logical construct that allows the control of an external amplifier. The EAPD value is reflected on the EAPD pin; a 1 causes the external amplifier to power up (equivalent to D0), and a 0 causes it to power down (equivalent to D3.) When the EAPD value = 1, the EAPD pin must be placed in a state appropriate to the current power state of the associated Pin Widget even though the EAPD value (in the register) may remain 1. The default state of this pin is 0 (driving low.) The pin defaults to an open-drain configuration (an external pull-up is recommended.)

Per the HD Audio specification and HDA015-B, multiple ports may control EAPD. The EAPD pin assumes the highest power state of all the EAPD bits in all of the pin complexes. The default value of EAPD is 1 (powered on), but the FG power state will override and the pin will be low. A port will request External Amp Power Up when its power state is active (FG and pin widget power state is D1 or D0) or (Analog PC_Beep is enabled and port is enabled as an output) and the port's EAPD bit is set to 1. The state of the EAPD pin (unless configured as an input or held low by an external circuit when configured as an open drain output) will be the logical OR of the external amp power up requests from all ports.
The digital_bios_test4[3] can get the value of EAPD pin.

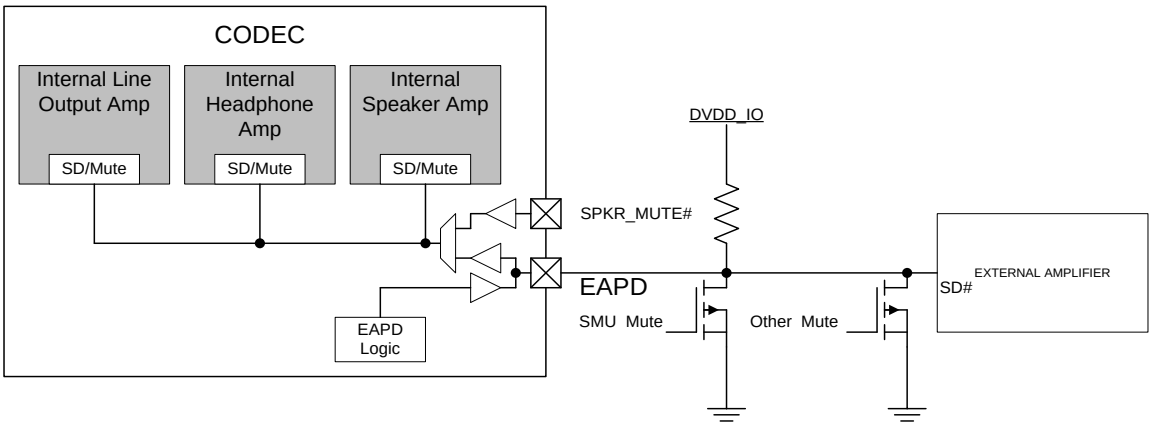
Digital Bios test4 LSB register
Get Response - HD_A [27:0] =0x1BB0800 Set - HD_A[27:8] = 0x1B308 I2C=6D88

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | 1-Force portA mapping to the same DAC where class D is mapped if class D is enabled.<br>0-Doesn't force portA mapping |
| 6 | 0 | Enable for additional low power mode on azalia interface<br>0-no low power mode on azalia interface<br>1-enable low power mode on azalia |
| 5:4 | | Reserved |
| 3 | RO | The value of EAPD pin. |
| 2 | 0 | 1 –enable s/w to disable the new clock gating added for RC OSC in Vilano from palm.<br>0-RCOSC clock is gated from rcosc top based on d3_lp signal |
| 1 | 0 | 1 –Clock disabled for rcosc/bclk for rcosc calibration block when bit 2 of this register is set<br>0-Clock enabled for rcosc/bclk rcosc calibration block when bit 2 of this register is set |
| 0 | 0 | 1 - Enable clock gating mux output from rcosc and bclk in D3 hot and D3 cold irrespective of reset is available or not.<br>0 – Disable clock gating |

| | | |
|---|---|---|
| | | 1 - D3 bcllk active, still achive low power mode then enable this bit to do more clock gating |

By default, the EAPD pin also functions as the shutdown input for the internal BTL amplifier. In this mode, a low value at the pin (either due to internal EAPD being 0, or to an external entity forcing the pin low) will cause the internal BTL amplifier to mute or enter a low power state depending on the amplifier configuration. (See below)
Vendor specific verbs are available to configure this pin. These verbs retain their values across link and single function group resets but are set to their default values by a power on reset:

| EAPD MODE1 | EAPD MODE0 | EAPD Pin Function | Description |
|---|---|---|---|
| 0 | 0 | Open Drain I/O | Value at pin is wired-AND of EAPD bit and external signal. (default) |
| 0 | 1 | CMOS Output | Pin assumes value of EAPD bit in pin widget. |
| 1 | 0 | CMOS Input | External signal controls internal amps. EAPD bit in pin widget is ignored. |
| 1 | 1 | CMOS Input | External signal controls internal amps. EAPD bit in pin widget is ignored. |




Digital_bios_test2[3:2] control the EAPD mode:
2'b01: Open drain mode(default), bit[1:0] =2'b10(weak pull-up, default)
2'b11: Output mod, bit[1:0] may set to 2'b00(no piull-up)

Set Digital Bios Test2 LSB register
Get Response - HD_A [27:0] =0x1Bb0400 Set - HD_A[27:11] = 0x1B304 I2C=6D84

| Bit | Default | Description |
|---|---|---|
| 7 | | Clock Enable for I2S bclock<br>1-Enable the clock<br>0-Disable the clock |
| 6 | 0 | In I2C/I2S bus mode this bit selects if bclk or mclk sources the PLL<br>1-Bclk<br>0-Mclk |
| 5:4 | 'b10 | EAPD pad output drive strength<br>00 – IOL/IOH=+/-1.3mA<br>01 – IOL/IOH=+/-2.0mA<br>10 – IOL/IOH=+/-2.7mA<br>11 – IOL/IOH=+/-3.3mA |
| 3 | 'b0 | 1-Push-pull mode: output enabled on EAPD<br>0-Open drain mode: output enable on EAPD controlled by signal self. |
| 2 | 'b1 | 1-Non-input only mode.<br>0-Input-only mode, external signal controls internal amps. EAPD bit in pin widget is ignored. |
| 1:0 | 'b10 | Pull down control for eapd pin.<br>00 – No pull up, No pull down<br>01 – Weak Pull down -> 28/65Kohm<br>10- Weak pull up value 1 -> 28-65Kohm<br>11 – Bus Keeper resistance -> 28 - 65kohm |

EAPD control register can configure if the PORTA/B/E/M/G control the EAPD pin.
If none of Ports control the EAPD pin, then EAPD can used as input only.

EAPD Control LSB register
Get Response - HD_A[27:0] = 0x1BB0200 Set - HD_A [27:8] = 0x1B302 I2C=6D82

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | 1 - Port B controls EAPD on GPIO5 if MSB bit[3]=1<br>0 – Port B doesn't control EAPD on GPIO5 |
| 6 | 0 | 1 - Port A controls EAPD on GPIO5 if MSB bit[3]=1<br>0 – Port A doesn't control EAPD on GPIO5 |
| 5 | 0 | 0 – No Polarity inversion of EAPD control<br>1 – Polarity Inversion on EAPD pin |
| 4 | 1 | 1 - Port G controls EAPD<br>0 – Port G doesn't control EAPD |
| 3 | 1 | 1 - Port M controls EAPD<br>0 – Port M doesn't control EAPD |
| 2 | 1 | 1 - Port E controls EAPD<br>0 – Port E doesn't control EAPD |
| 1 | 1 | 1 - Port B controls EAPD<br>0 – Port B doesn't control EAPD |
| 0 | 1 | 1 - Port A controls EAPD<br>0 – Port A doesn't control EAPD |

EAPD Control MSB register
Get Response - HD_A[27:0] = 0x1BB0300 Set - HD_A [27:8] = 0x1B303 I2C=6D83

| Bit | Default | Description |
|---|---|---|
| 7:4 | 0 | Reserved Read as 0 |
| 3 | 0 | 0 – No EAPD on GPIO5<br>1 – EAPD on GPIO5 |
| 2 | 0 | 1 - Port G controls EAPD on GPIO5 if bit[3]=1<br>0 – Port G doesn't control EAPD on GPIO5 |
| 1 | 0 | 1 - Port M controls EAPD on GPIO5 if bit[3]=1<br>0 – Port M doesn't control EAPD on GPIO5 |
| 0 | 0 | 1 - Port E controls EAPD on GPIO5 if bit[3]=1<br>0 – Port E doesn't control EAPD on GPIO5 |

## 16.2 Speaker Mute Pin (SPKR_MUTE#)

The SPKR_MUTE# pin may be used as an input to disable internal amplifiers or as a general purpose I/O pin. When configured as a GPIO, the logical state of SPKR_MUTE# is inactive. Any amplifier referencing the SPKR_MUTE# pin will behave as if the SPKR_MUTE# input were in a logical state that allows playback.
By default, the SPKR_MUTE# pin is configured as the Speaker Mute input but, to avoid problems when the system is designed to use the pin as a GPIO, no amplifiers are controlled by this pin. (See implementation details for how to use SPKR_MUTE# with EAPD to work around this limitation.)

## 16.3 Amplifier Control

The internal amplifiers, including the speaker amplifier, may be controlled by their port power state and enable. As described by the HD Audio specification and Microsoft Windows Logo Program documents, the output enable in the pin widget may act as a mute but is not intended to control the power state of the port. The power state is controlled by dedicated verbs. However, sometimes it is desirable to control the mute or power state of an internal amplifier by an external means. An example is Music Mode (D3 Live). In this mode, the host interface is inactive but simple audio playback from an external source is desired. Another example is analog PC beep. Analog PC Beep may be enabled before the source of PC

Beep has been configured. To avoid pops or odd sounds, the host may use EAPD or SPKR_MUTE# to mute outputs.

Each output amplifier may be controlled by their port controls, SPKR_MUTE#, or EAPD. This includes Port A (headphone amplifier), Port G (Speaker amplifier), Port E (Line Output), and Port M (mono Line Output). When responding to an external shutdown signal (EAPD or SPKR_MUTE#) the amplifier may be configured to mute or to turn off with either an active high or low signal. This is useful for situations when, for example, one desires to turn off the speaker amp (internal or external) when headphones on and turn off the headphones when the speakers are on.

| Amp Control (replicated for each amp) | Description |
|---|---|
| SD SEL | Selects how the port amplifier is controlled:<br>0 = Amp controlled by EAPD or SPKR_MUTE# pin (default)<br>1 = Amp controlled by pin widget control and power state |
| SD INV | Polarity when amp controlled by EAPD/SPKR_MUTE#<br>0 = Amp is disabled when input low (default)<br>1 = Amp is disabled when input is high |

EAPD/SPKR Mute Control bit [0] can control the Mute use EAPD or SPKR_MUTE#.
Each port's polarity bit also control the mute polarity.

BIOS EAPD/SPKR Mute Control LSB register
Get - HD_A [27:0] = 0x1BB1200 Set - HD_A[27:8] = 0x1B312 I2C=6D92

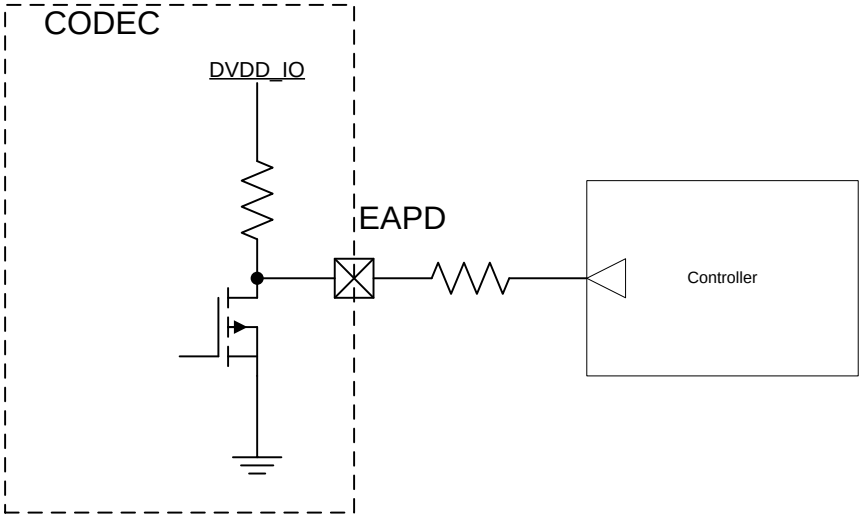
| Bit | Default | Description |
|---|---|---|
| 7 | 0 | PortG pad mute bypass<br>0: not bypass<br>1: bypass pad control |
| 6 | 0 | PortG pad mute polarity<br>0: don't change polarity (0 = mute)<br>1: change polarity (1 = mute) |
| 5 | 0 | PortM pad mute bypass<br>0: not bypass<br>1: bypass pad control |
| 4 | 0 | PortM pad mute polarity<br>0: don't change polarity (0 = mute)<br>1: change polarity (1 = mute) |
| 3:1 | 0 | Reserve |
| 0 | 0 | Pad select for mute:<br>0: EAPD<br>1; SPKR_MUTE |

BIOS EAPD/SPKR Mute Control MSB register
Get - HD_A [27:0] = 0x1BB1300 Set - HD_A[27:8] = 0x1B313 I2C=6D93

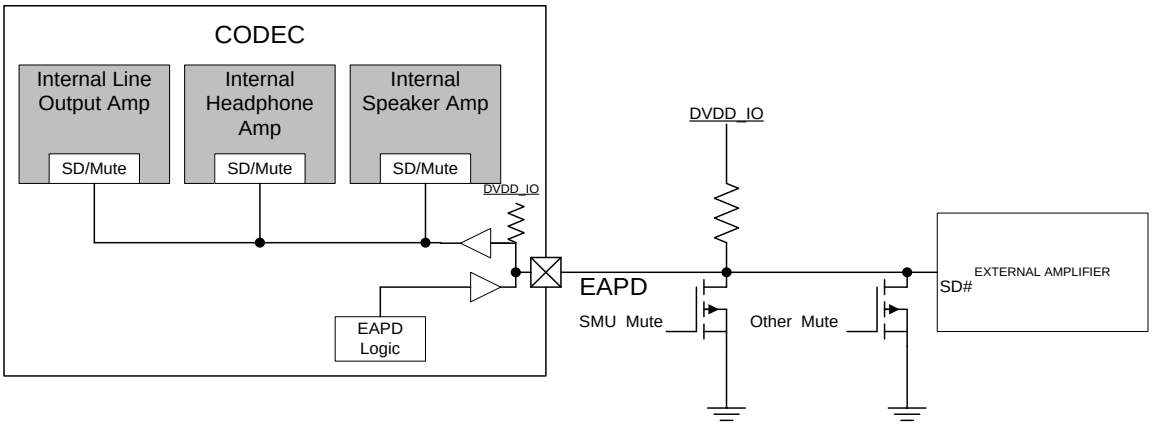
| Bit | Default | Description |
|---|---|---|
| 7:4 | 0 | Reserved |
| 3 | 0 | PortA or PortB pad mute bypass<br>0: not bypass<br>1: bypass pad control |
| 2 | 0 | PortA or PortB pad mute polarity<br>0: don't change polarity (0 = mute)<br>1: change polarity (1 = mute) |
| 1 | 0 | PortE pad mute bypass<br>0: not bypass<br>1: bypass pad control |
| 0 | 0 | PortE pad mute polarity<br>0: don't change polarity (0 = mute)<br>1: change polarity (1 = mute) |

## System Implementation

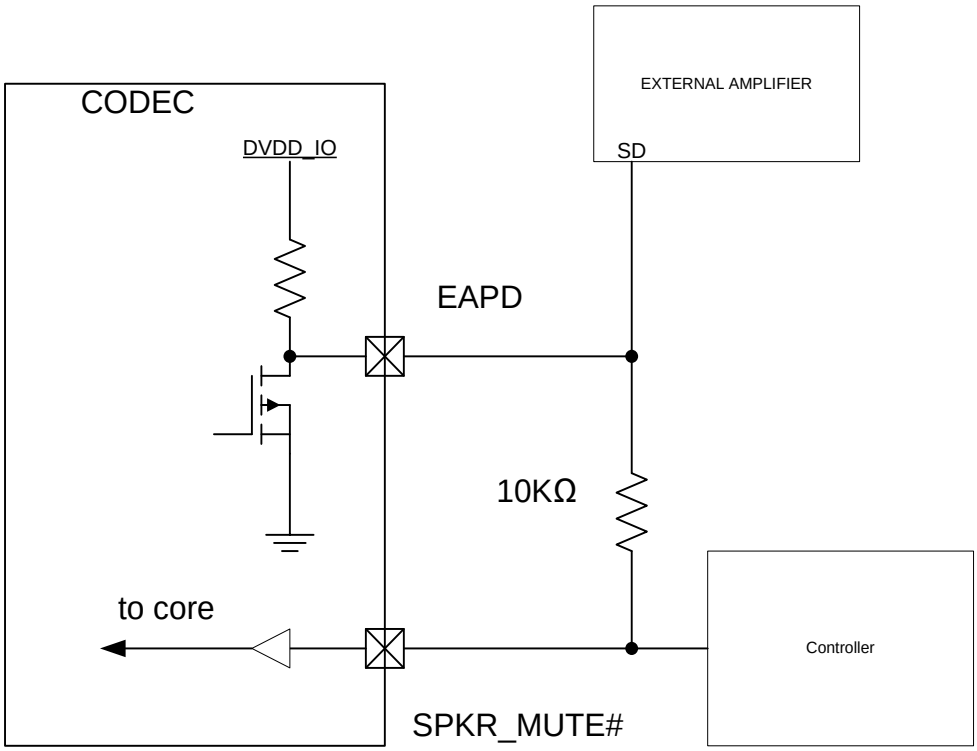
Neither the EAPD pin nor the SPKR_MUTE# pin is needed for proper operation of the device. SPKR_MUTE# may be programmed for other functions and EAPD may be left unconnected.
When using the EAPD pin, it is important to note that it is configured as an open drain output by default. If configuring the system for EAPD as a CMOS input, use a resistor to prevent damage due to bus contention that might arise before the EAPD pin can be configured as an input.




Multiple controllers may be connected in a wired AND configuration when the EAPD pin is in open drain or input mode. The state of the pin will be low unless all sources release the bus.

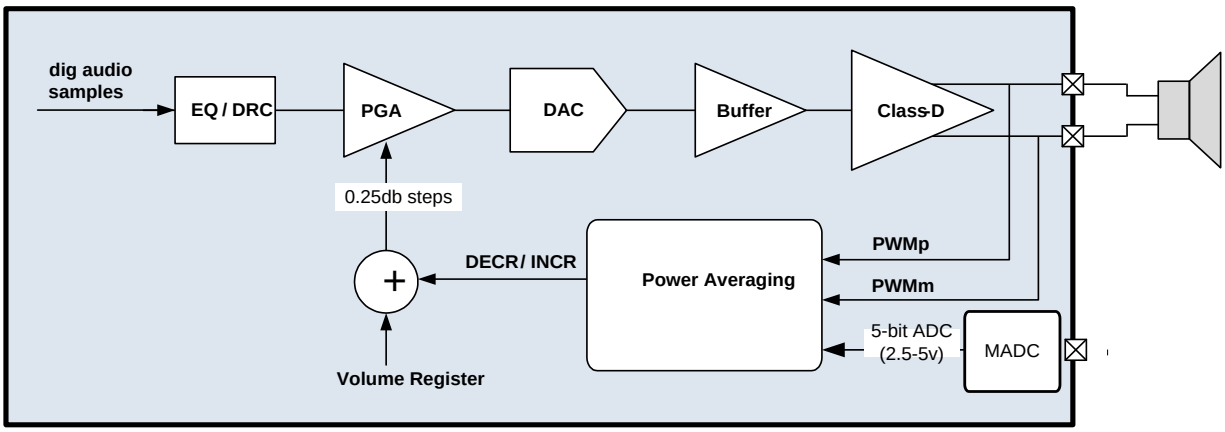



When implementing a system where SPKR_MUTE acts like a dedicated input and EAPD is a dedicated output (when replacing the CX20952 for example) the power-on default conditions of bi-directional EAPD and the SPKR_MUTE pin being ignored by the internal amplifiers present a challenge. To support this use case, the SPKR_MUTE and EAPD pins may be connect together using a resistor as shown. After the application of power, the EAPD pin will control the external and internal amplifiers and take precedence over the external SPKR_MUTE controller. After configuration, the EAPD pin controls the external amplifier and SPKR_MUTE may control the internal amplifier.


CODEC
DVDD_IO
EXTERNAL AMPLIFIER
SD
EAPD
10KΩ
to core
Controller
SPKR_MUTE#

# 17 Power Averaging (PA-1.1) Speaker Protection

## 17.1 PA-2 System Diagram




## 17.2 Overview

The PA function estimates the power driven into speakers which are attached to the Class-D amplifier and, if necessary, forces a reduction in the output signal amplitude to protect the speakers from damage.

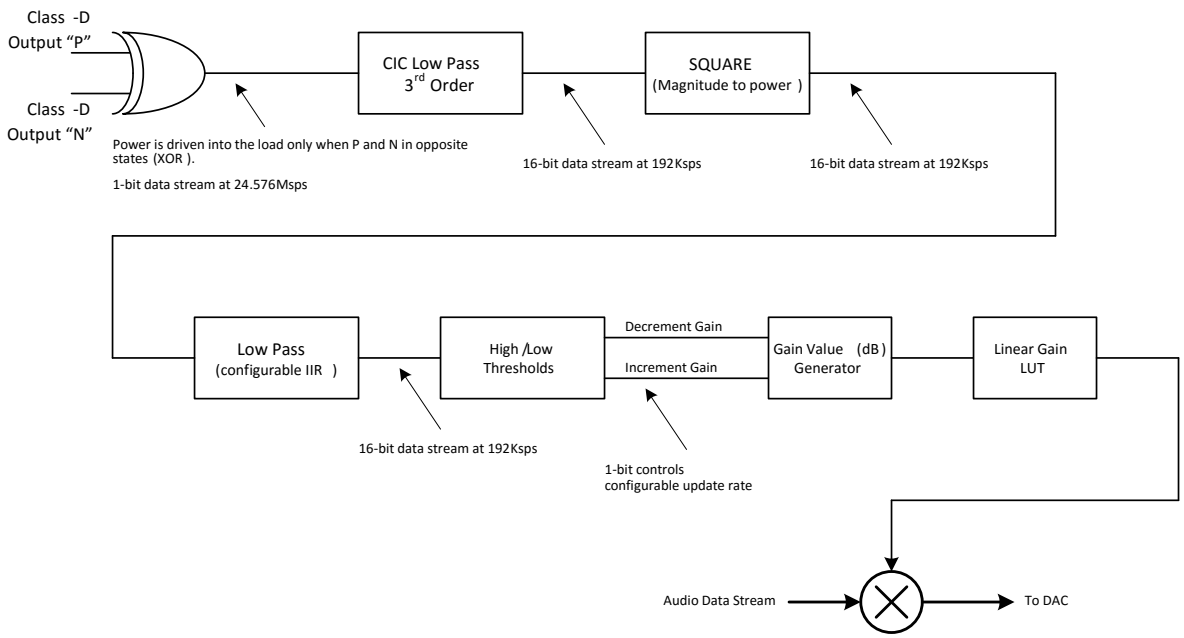
Ideally, the power into the attached load can be described by the equation:

$$P_{AVG} = (V_{RMS})^2/R_{LOAD}$$

This equation forms the basis of how an estimate is made of the power driven into the attached speakers. The power estimate is generated by continuously monitoring the drive state of the Class-D amplifier, and applying the filtering steps described below.

The estimated power is compared with programmed thresholds. The result of those comparisons is used to modify the output signal level before it reaches the Class-D amplifier. Modification of the output signal level takes the form of either decrementing or incrementing a gain value, which is applied to the audio signal within the interpolation datapath that is associated with the DAC which drives the Class-D amplifier. Note that the gain used to modify the signal level can take on values of 0dB or less, but never a value greater than 0dB. (In other words, the PA function can cause the signal to be attenuated, but not amplified.)

The following figure illustrates these basic steps, as performed by the PA function in the existing (Neptune) design (for a single channel):




These steps are carried out separately for the left and right Class-D amplifier channels, and then reconciled to maintain left/right balance.

Detailed descriptions of these steps are given in section 2 (below).

These steps are modified in the updated (Deep Eddy) design, as described in section 3 (below).

## 17.3 Deep Eddy (updated) Design

### 17.3.1 3rd Order CIC Filter

In the updated design, the CIC filter is modified to accept a 5-bit input value, rather than a 1-bit input value. In addition to "on" and "off", this 5-bit value will also represent the voltage level being applied by the Class-D amplifier to the attached speakers. The 5-bit value will come from a monitor ADC, which continuously measure the Class-D power supply voltage.

As with the previous design, the input value will represent only non-negative numbers. It will be encoded as follows:

| 5-bit Code | Decimal Code Value | Class-D Voltage | 5-bit Code | Decimal Code Value | Class-D Voltage |
|---|---|---|---|---|---|
| 00000 | 0 | 2.5 | 10000 | 16 | 4.1 |
| 00001 | 1 | 2.6 | 10001 | 17 | 4.2 |
| 00010 | 2 | 2.7 | 10010 | 18 | 4.3 |
| 00011 | 3 | 2.8 | 10011 | 19 | 4.4 |
| 00100 | 4 | 2.9 | 10100 | 20 | 4.5 |

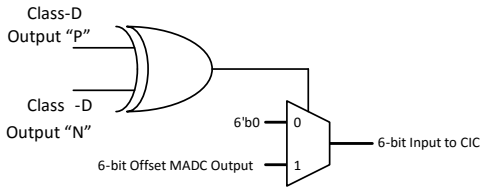
| | | | | | |
|---|---|---|---|---|---|
| 00101 | 5 | 3.0 | 10101 | 21 | 4.6 |
| 00110 | 6 | 3.1 | 10110 | 22 | 4.7 |
| 00111 | 7 | 3.2 | 10111 | 23 | 4.8 |
| 01000 | 8 | 3.3 | 11000 | 24 | 4.9 |
| 01001 | 9 | 3.4 | 11001 | 25 | 5.0 |
| 01010 | 10 | 3.5 | 11010 | 26 | 5.1 |
| 01011 | 11 | 3.6 | 11011 | 27 | 5.2 |
| 01100 | 12 | 3.7 | 11100 | 28 | 5.3 |
| 01101 | 13 | 3.8 | 11101 | 29 | 5.4 |
| 01110 | 14 | 3.9 | 11110 | 30 | 5.5 |
| 01111 | 15 | 4.0 | 11111 | 31 | 5.6 |

The 5-bit value coming from the battery monitor ADC will be launched on the falling edge of the 3.072MHz ADC clock.

Before the 5-bit ADC value is used, the quantity 'd25 (0x32) is added to it, forming a 6-bit unsigned value. This offset causes the 6-bit codes to be directly proportional to the represented voltage, as shown in this table:

| 6-bit Code | Decimal Code Value | Class-D Voltage | 6-bit Code | Decimal Code Value | Class-D Voltage |
|---|---|---|---|---|---|
| 011001 | 25 | 2.5 | 101001 | 41 | 4.1 |
| 011010 | 26 | 2.6 | 101010 | 42 | 4.2 |
| 011011 | 27 | 2.7 | 101011 | 43 | 4.3 |
| 011100 | 28 | 2.8 | 101100 | 44 | 4.4 |
| 011101 | 29 | 2.9 | 101101 | 45 | 4.5 |
| 011110 | 30 | 3.0 | 101110 | 46 | 4.6 |
| 011111 | 31 | 3.1 | 101111 | 47 | 4.7 |
| 100000 | 32 | 3.2 | 110000 | 48 | 4.8 |
| 100001 | 33 | 3.3 | 110001 | 49 | 4.9 |
| 100010 | 34 | 3.4 | 110010 | 50 | 5.0 |
| 100011 | 35 | 3.5 | 110011 | 51 | 5.1 |
| 100100 | 36 | 3.6 | 110100 | 52 | 5.2 |
| 100101 | 37 | 3.7 | 110101 | 53 | 5.3 |
| 100110 | 38 | 3.8 | 110110 | 54 | 5.4 |
| 100111 | 39 | 3.9 | 110111 | 55 | 5.5 |
| 101000 | 40 | 4.0 | 111000 | 56 | 5.6 |

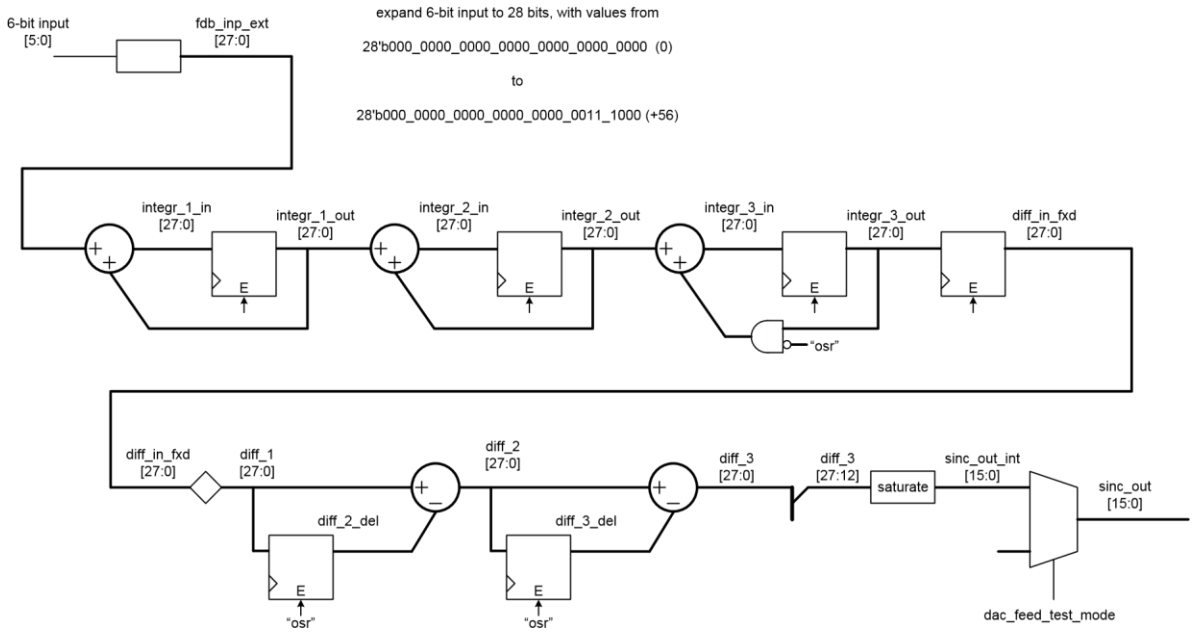
In the previous design, the XOR output was used directly, to represent the values 0 and 1. In the updated design, the XOR output will be combined with the 6-bit code (from the monitor ADC), as follows, to provide a 6-bit input to the CIC filter:




Note that the 6-bit value is also expanded, though zero-extension, into a 28-bit value at the input to the CIC filter.

The updated CIC filter uses a down-sample ratio of 128. The gain through the filter is therefore 2,097,152. Since the input may now take on values from 0 to +56, the output of the CIC filter may take on values from 0 to +117,440,512. Intermediate calculations within the CIC may take on negative values, however. Using a 2's complement notation, a minimum of 28 bits is required to correctly represent CIC filter computations.

The following figure illustrates the CIC filter implementation in the updated design:




As shown in the above figure, the final output is truncated to a 16-bit signed value, losing 12 LSb's. This effectively divides the output by 2^12, such that the overall gain through the CIC filter becomes 2,097,152 / 4,096 = 512. The final output can therefore take on values from 0 to +28,672.

Independent CIC filters are implemented for the left and right channels.

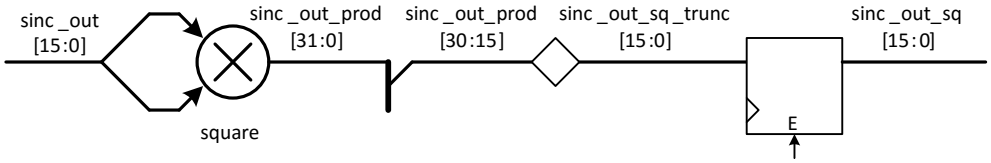
#### 17.3.1.1 Summary of Changes from Neptune to Oeanside/Deep Eddy

The following represent changes in the CIC filter portion of the PA function for Deep Eddy vs. Neptune:

- A 5-bit unsigned input value is received from a monitor ADC.
- The 5-bit input value, which may vary over time, replaces the '+1' input value coming from the input XOR gate.
- The constant 'd25 is added to the 5-bit value to produce a 6-bit unsigned value.
- The 6-bit input value is expanded (zero-extended) to the full width of the datapath
- The datapath width is increased from 23 bits to 28 bits
- The down-sample ratio is fixed at 128. (In Neptune, it was configurable to either 125 or 128, and the configuration was set to 125.)
- The output bit selection and saturation logic are adjusted to accommodate the 27-bit datapath width

### 17.3.2 Voltage to Power Conversion (squaring)

The output of the CIC filter represents the instantaneous voltage applied to the load by the Class-D amplifier. The power applied to the load is proportional to the square of the voltage, and the CIC filter output is therefore squared at this step, as shown in the following figure:




Note that the output from the squaring operating is truncated to a 16-bit, signed value, losing 1 MSb and
15 LSb's. This effectively divides the output by 2^15. The output can therefore take on values from 0 to +25,088.

Independent squaring functions are implemented for the left and right channels.

### 17.3.3 Low Pass (Averaging) Filter

A simple, single-pole, low-pass filter is applied to the output of the squaring function. The output of the low pass filter represents the average (or mean) of the squared voltage applied to the load.

Noting again the underlying equation for average power:

$$P_{AVG} = (V_{RMS})^2/R_{LOAD}$$

It can be seen that the output of the low pass filter is directly proportional to the average power in the attached load.

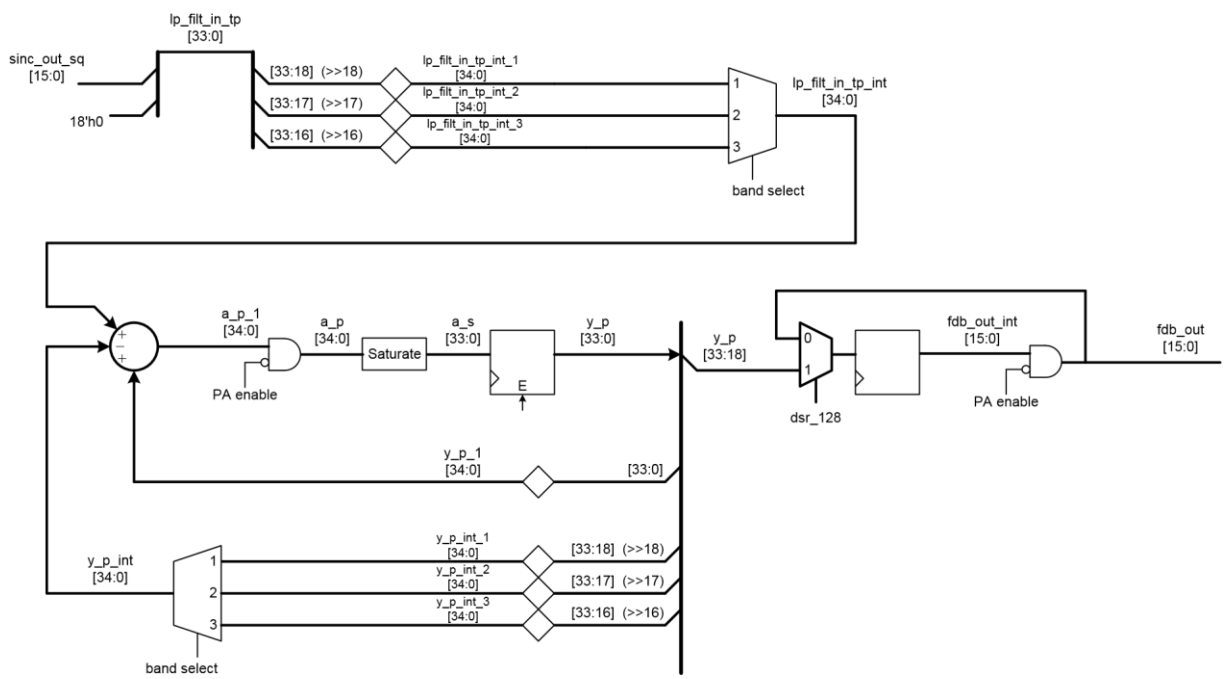
The single-pole filter is described by the following transfer function:

$$\frac{p}{1-(1-p)z^{-1}}$$

The pole location specifier ("$p$") for the low pass filter is configurable to have one of the three values. The selected pole location determines the filter bandwidth, and is accomplished through the Codec_Test26[3:2] register bit field, as follows:

| Register Setting | Value of "$p$" | Settling Times | | | |
|---|---|---|---|---|---|
| | | 63.2% | | 95% | |
| | | Sample Periods | Seconds | Sample Periods | Seconds |
| 00 | $1/2^{16}$ | 65536 | 0.341333 | 196608 | 1.024 |
| 01 | $1/2^{15}$ | 32768 | 0.170666 | 98304 | 0.512 |
| 10 | $1/2^{14}$ | 16384 | 0.085333 | 49152 | 0.256 |
| 11 | $1/2^{12}$ | 4096 | 0.021333 | 12288 | 0.064 |

The detailed implementation of the averaging filter is shown in the following diagram:




The averaging filter is designed to have unity DC gain. Therefore, with a constant (DC) input, the 16-bit output can take on values from 0 to +25,088.

### 17.3.3.1 Summary of Changes from Neptune to Deep Eddy

The following represent changes in the averaging filter portion of the PA function for Deep Eddy vs. Neptune:

- The third filter configuration is changed to use the pole specifier $1/2^{16}$ instead of $1/(2^{17} + 2^{18})$. This allows two additions to be removed from the hardware implementation.

Independent averaging filters are implemented for the left and right channels.

### 17.3.4 Threshold Comparison

As described above, the output of the averaging filter is proportional to the average power in the load driven by the Class-D amplifier.

This average power estimate is compared against a high and a low threshold value, to determine whether the signal to the Class-D amplifier should be adjusted, and if so, whether that adjustment should cause more or less attenuation. If the power estimate exceeds the high threshold, greater attenuation is applied. If the power estimate is below the low threshold, less attenuation is applied. (As noted previously the PA function can apply only attenuation, and cannot apply amplification.)

The high and low thresholds are individually configurable. They are selected through the register fields ????? and ?????, respectively.

The following table lists all of the available threshold values. Note that the percentage value listed for each threshold is relative to the maximum power estimate that can be produced with a 5V power supply. Noting that the input code representing 5V is 0x32 ('d50), the maximum power estimate for 5V is 'd20000 (0x4E20). The percentages are therefore relative to 20000.

| Selection | Threshold | % | Selection | Threshold | % | Selection | Threshold | % | Selection | Threshold | % |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 12800 | 64 | 16 | 9600 | 48 | 32 | 6400 | 32 | 48 | 3200 | 16 |
| 1 | 12600 | 63 | 17 | 9400 | 47 | 33 | 6200 | 31 | 49 | 3000 | 15 |
| 2 | 12400 | 62 | 18 | 9200 | 46 | 34 | 6000 | 30 | 50 | 2800 | 14 |
| 3 | 12200 | 61 | 19 | 9000 | 45 | 35 | 5800 | 29 | 51 | 2600 | 13 |
| 4 | 12000 | 60 | 20 | 8800 | 44 | 36 | 5600 | 28 | 52 | 2400 | 12 |
| 5 | 11800 | 59 | 21 | 8600 | 43 | 37 | 5400 | 27 | 53 | 2200 | 11 |
| 6 | 11600 | 58 | 22 | 8400 | 42 | 38 | 5200 | 26 | 54 | 2000 | 10 |
| 7 | 11400 | 57 | 23 | 8200 | 41 | 39 | 5000 | 25 | 55 | 1800 | 9 |
| 8 | 11200 | 56 | 24 | 8000 | 40 | 40 | 4800 | 24 | 56 | 1600 | 8 |
| 9 | 11000 | 55 | 25 | 7800 | 39 | 41 | 4600 | 23 | 57 | 1400 | 7 |
| 10 | 10800 | 54 | 26 | 7600 | 38 | 42 | 4400 | 22 | 58 | 1200 | 6 |
| 11 | 10600 | 53 | 27 | 7400 | 37 | 43 | 4200 | 21 | 59 | 1000 | 5 |
| 12 | 10400 | 52 | 28 | 7200 | 36 | 44 | 4000 | 20 | 60 | 800 | 4 |
| 13 | 10200 | 51 | 29 | 7000 | 35 | 45 | 3800 | 19 | 61 | 600 | 3 |
| 14 | 10000 | 50 | 30 | 6800 | 34 | 46 | 3600 | 18 | 62 | 400 | 2 |
| 15 | 9800 | 49 | 31 | 6600 | 33 | 47 | 3400 | 17 | 63 | 200 | 1 |

By default, the thresholds are used as shown above. A "Shift Select" (Codec_Test23[5:4]) register field may be optionally used to reduce the selected threshold by a factor of 4, 8 or 16.

The rate at which the threshold comparisons are updated is established through the register ????? as follows:

| Update Rate Select | Samples Between Updates | Time Between Updates (seconds) |
|---|---|---|
| 0001 | 786,432 | 4.096 |
| 0010 | 393,216 | 2.048 |
| 0011 | 196,608 | 1.024 |
| 0100 | 98,304 | 0.512 |
| 0101 | 49,152 | 0.256 |
| 0110 | 24,576 | 0.128 |
| 0111 | 12,288 | 0.064 |
| 1000 | 6,144 | 0.032 |
| 1001 | 3,072 | 0.016 |
| 1010 | 1,536 | 0.008 |

Selection values not listed should not be used, and may cause unpredictable results.

Independent threshold comparison functions are implemented for the left and right channels.

#### 17.3.4.1 Summary of Changes from Neptune to Deep Eddy

The following represent changes in the threshold comparison portion of the PA function for Deep Eddy vs.
Neptune:

- The number of available threshold values has been increased from 16 to 64
- The registers used to select the high and low threshold have been changed (to accommodate 2 additional bits each)
- The thresholds are no longer a percentage of the maximum possible output value. Instead, they are a percentage of the maximum possible value produced for the case that the Class-D power supply is at 5V.
- The threshold comparison window selection is now independent of the filter bandwidth selection.

### 17.3.5 Attenuation Control

The output of the Threshold Comparison function (above) is two signals, one which indicates that the average power is above the high threshold, and the other which indicates that the average power is below the low threshold.

These two signals are used to modify an attenuation control value that is applied to the audio stream, prior to it reaching the class-D amplifier.

The attenuation control value ("spkr_gain_decr[5:0]") is represented as a 6-bit, unsigned quantity, which indicates the amount of attenuation to be applied. A value of 0x0 represents 0dB. Each increment of the value then represents 0.25dB of added attenuation, to a maximum of 15.75dB of attenuation.

The attenuation value is updated, according to the two threshold comparison outputs, at the end of each filter window period. The update occurs as follows:

| High Threshold Compare | Low Threshold Compare | Attenuation Control Value Update |
|---|---|---|
| 0 | 0 | No Change |
| 0 | 1 | Subtract 1 from Attenuation Control Value |
| 1 | 0 | Add 1 to Attenuation Control Value |
| 1 | 1 | No Change |

The update logic also prevents overflow or underflow of the control value from occurring.

Independent attenuation control functions are implemented for the left and right channels. However, to maintain left/right balance, the larger of the two attenuation values is applied to both the left and right channels.

#### 17.3.5.1 Summary of Changes from Neptune to Deep Eddy

The following represent changes in the Attenuation Control portion of the PA function for Deep Eddy vs. Neptune:

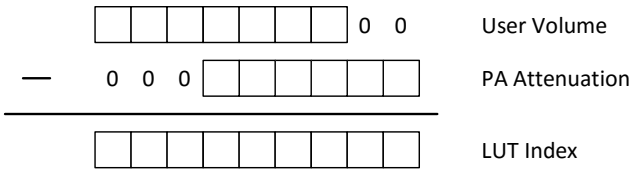
- The attenuation step size is reduced from 1dB to 0.25dB
- The control value is increased from 4 bits to 6 bits

#### 17.3.5.2 Application of Attenuation

The attenuation called for by the PA function is applied by modifying the user volume setting for the data stream which is driving the class-D amplifier. The volume control, and the application of the PA attenuation, is accomplished within the interpolator module which has been assigned to drive the class-D amplifier.

User volume control is established through a 7-bit, user-writable register. Each increment of the value in this register represents a 1dB step in volume. The 7-bit register supports 128 volume steps, of which 81 (0 through 80) are used. The 81 steps span the range from -74dB to +6dB (inclusive) of user volume. The value contained in the user volume control register is used as an index into a look-up table, which contains linear coefficients, which are applied to the data stream as it passes through the DAC interpolator module.

To modify the user volume setting, the 6-bit PA attenuation control value is subtracted from the 7-bit user volume value. However, because the PA volume step size is 0.25dB, vs. the 1dB user volume steps, the PA attenuation control is shifted 2 bits to the right, relative to the 7-bit user volume control value. The following figure illustrates the alignment:



The resulting LUT index is 9 bit in length, and may take on values from 0 to 320. The result of the subtraction is limited to remain within the range of 0 to 320.

The following table lists all of the linear volume coefficients within the lookup table:

| Selection | Coefficient | dB | Selection | Coefficient | dB | Selection | Coefficient | dB | Selection | Coefficient | dB |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 13 | -74.00 | 81 | 135 | -53.75 | 162 | 1385 | -33.50 | 243 | 14255 | -13.25 |
| 1 | 13 | -73.75 | 82 | 139 | -53.50 | 163 | 1426 | -33.25 | 244 | 14672 | -13.00 |
| 2 | 14 | -73.50 | 83 | 143 | -53.25 | 164 | 1467 | -33.00 | 245 | 15100 | -12.75 |
| 3 | 14 | -73.25 | 84 | 147 | -53.00 | 165 | 1510 | -32.75 | 246 | 15541 | -12.50 |
| 4 | 15 | -73.00 | 85 | 151 | -52.75 | 166 | 1554 | -32.50 | 247 | 15995 | -12.25 |
| 5 | 15 | -72.75 | 86 | 155 | -52.50 | 167 | 1599 | -32.25 | 248 | 16462 | -12.00 |
| 6 | 16 | -72.50 | 87 | 160 | -52.25 | 168 | 1646 | -32.00 | 249 | 16943 | -11.75 |
| 7 | 16 | -72.25 | 88 | 165 | -52.00 | 169 | 1694 | -31.75 | 250 | 17437 | -11.50 |
| 8 | 16 | -72.00 | 89 | 169 | -51.75 | 170 | 1744 | -31.50 | 251 | 17947 | -11.25 |
| 9 | 17 | -71.75 | 90 | 174 | -51.50 | 171 | 1795 | -31.25 | 252 | 18471 | -11.00 |
| 10 | 17 | -71.50 | 91 | 179 | -51.25 | 172 | 1847 | -31.00 | 253 | 19010 | -10.75 |
| 11 | 18 | -71.25 | 92 | 185 | -51.00 | 173 | 1901 | -30.75 | 254 | 19565 | -10.50 |
| 12 | 18 | -71.00 | 93 | 190 | -50.75 | 174 | 1957 | -30.50 | 255 | 20136 | -10.25 |
| 13 | 19 | -70.75 | 94 | 196 | -50.50 | 175 | 2014 | -30.25 | 256 | 20724 | -10.00 |
| 14 | 20 | -70.50 | 95 | 201 | -50.25 | 176 | 2072 | -30.00 | 257 | 21329 | -9.75 |
| 15 | 20 | -70.25 | 96 | 207 | -50.00 | 177 | 2133 | -29.75 | 258 | 21952 | -9.50 |
| 16 | 21 | -70.00 | 97 | 213 | -49.75 | 178 | 2195 | -29.50 | 259 | 22593 | -9.25 |
| 17 | 21 | -69.75 | 98 | 220 | -49.50 | 179 | 2259 | -29.25 | 260 | 23253 | -9.00 |
| 18 | 22 | -69.50 | 99 | 226 | -49.25 | 180 | 2325 | -29.00 | 261 | 23932 | -8.75 |
| 19 | 23 | -69.25 | 100 | 233 | -49.00 | 181 | 2393 | -28.75 | 262 | 24631 | -8.50 |
| 20 | 23 | -69.00 | 101 | 239 | -48.75 | 182 | 2463 | -28.50 | 263 | 25350 | -8.25 |
| 21 | 24 | -68.75 | 102 | 246 | -48.50 | 183 | 2535 | -28.25 | 264 | 26090 | -8.00 |
| 22 | 25 | -68.50 | 103 | 254 | -48.25 | 184 | 2609 | -28.00 | 265 | 26852 | -7.75 |
| 23 | 25 | -68.25 | 104 | 261 | -48.00 | 185 | 2685 | -27.75 | 266 | 27636 | -7.50 |
| 24 | 26 | -68.00 | 105 | 269 | -47.75 | 186 | 2764 | -27.50 | 267 | 28443 | -7.25 |
| 25 | 27 | -67.75 | 106 | 276 | -47.50 | 187 | 2844 | -27.25 | 268 | 29274 | -7.00 |
| 26 | 28 | -67.50 | 107 | 284 | -47.25 | 188 | 2927 | -27.00 | 269 | 30129 | -6.75 |
| 27 | 28 | -67.25 | 108 | 293 | -47.00 | 189 | 3013 | -26.75 | 270 | 31008 | -6.50 |
| 28 | 29 | -67.00 | 109 | 301 | -46.75 | 190 | 3101 | -26.50 | 271 | 31914 | -6.25 |
| 29 | 30 | -66.75 | 110 | 310 | -46.50 | 191 | 3191 | -26.25 | 272 | 32846 | -6.00 |
| 30 | 31 | -66.50 | 111 | 319 | -46.25 | 192 | 3285 | -26.00 | 273 | 33805 | -5.75 |
| 31 | 32 | -66.25 | 112 | 328 | -46.00 | 193 | 3380 | -25.75 | 274 | 34792 | -5.50 |
| 32 | 33 | -66.00 | 113 | 338 | -45.75 | 194 | 3479 | -25.50 | 275 | 35808 | -5.25 |
| 33 | 34 | -65.75 | 114 | 348 | -45.50 | 195 | 3581 | -25.25 | 276 | 36854 | -5.00 |
| 34 | 35 | -65.50 | 115 | 358 | -45.25 | 196 | 3685 | -25.00 | 277 | 37930 | -4.75 |
| 35 | 36 | -65.25 | 116 | 369 | -45.00 | 197 | 3793 | -24.75 | 278 | 39037 | -4.50 |
| Selection | Coefficient | dB | Selection | Coefficient | dB | Selection | Coefficient | dB | Selection | Coefficient | dB |
| 36 | 37 | -65.00 | 117 | 379 | -44.75 | 198 | 3904 | -24.50 | 279 | 40177 | -4.25 |
| 37 | 38 | -64.75 | 118 | 390 | -44.50 | 199 | 4018 | -24.25 | 280 | 41350 | -4.00 |
| 38 | 39 | -64.50 | 119 | 402 | -44.25 | 200 | 4135 | -24.00 | 281 | 42558 | -3.75 |
| 39 | 40 | -64.25 | 120 | 414 | -44.00 | 201 | 4256 | -23.75 | 282 | 43801 | -3.50 |
| 40 | 41 | -64.00 | 121 | 426 | -43.75 | 202 | 4380 | -23.50 | 283 | 45080 | -3.25 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 41 | 43 | -63.75 | 122 | 438 | -43.50 | 203 | 4508 | -23.25 | 284 | 46396 | -3.00 |
| 42 | 44 | -63.50 | 123 | 451 | -43.25 | 204 | 4640 | -23.00 | 285 | 47751 | -2.75 |
| 43 | 45 | -63.25 | 124 | 464 | -43.00 | 205 | 4775 | -22.75 | 286 | 49145 | -2.50 |
| 44 | 46 | -63.00 | 125 | 478 | -42.75 | 206 | 4915 | -22.50 | 287 | 50580 | -2.25 |
| 45 | 48 | -62.75 | 126 | 491 | -42.50 | 207 | 5058 | -22.25 | 288 | 52057 | -2.00 |
| 46 | 49 | -62.50 | 127 | 506 | -42.25 | 208 | 5206 | -22.00 | 289 | 53577 | -1.75 |
| 47 | 51 | -62.25 | 128 | 521 | -42.00 | 209 | 5358 | -21.75 | 290 | 55142 | -1.50 |
| 48 | 52 | -62.00 | 129 | 536 | -41.75 | 210 | 5514 | -21.50 | 291 | 56752 | -1.25 |
| 49 | 54 | -61.75 | 130 | 551 | -41.50 | 211 | 5675 | -21.25 | 292 | 58409 | -1.00 |
| 50 | 55 | -61.50 | 131 | 568 | -41.25 | 212 | 5841 | -21.00 | 293 | 60115 | -0.75 |
| 51 | 57 | -61.25 | 132 | 584 | -41.00 | 213 | 6011 | -20.75 | 294 | 61870 | -0.50 |
| 52 | 58 | -61.00 | 133 | 601 | -40.75 | 214 | 6187 | -20.50 | 295 | 63677 | -0.25 |
| 53 | 60 | -60.75 | 134 | 619 | -40.50 | 215 | 6368 | -20.25 | 296 | 65536 | 0.00 |
| 54 | 62 | -60.50 | 135 | 637 | -40.25 | 216 | 6554 | -20.00 | 297 | 67450 | 0.25 |
| 55 | 64 | -60.25 | 136 | 655 | -40.00 | 217 | 6745 | -19.75 | 298 | 69419 | 0.50 |
| 56 | 66 | -60.00 | 137 | 674 | -39.75 | 218 | 6942 | -19.50 | 299 | 71446 | 0.75 |
| 57 | 67 | -59.75 | 138 | 694 | -39.50 | 219 | 7145 | -19.25 | 300 | 73533 | 1.00 |
| 58 | 69 | -59.50 | 139 | 714 | -39.25 | 220 | 7353 | -19.00 | 301 | 75680 | 1.25 |
| 59 | 71 | -59.25 | 140 | 735 | -39.00 | 221 | 7568 | -18.75 | 302 | 77890 | 1.50 |
| 60 | 74 | -59.00 | 141 | 757 | -38.75 | 222 | 7789 | -18.50 | 303 | 80164 | 1.75 |
| 61 | 76 | -58.75 | 142 | 779 | -38.50 | 223 | 8016 | -18.25 | 304 | 82505 | 2.00 |
| 62 | 78 | -58.50 | 143 | 802 | -38.25 | 224 | 8250 | -18.00 | 305 | 84914 | 2.25 |
| 63 | 80 | -58.25 | 144 | 825 | -38.00 | 225 | 8491 | -17.75 | 306 | 87394 | 2.50 |
| 64 | 83 | -58.00 | 145 | 849 | -37.75 | 226 | 8739 | -17.50 | 307 | 89946 | 2.75 |
| 65 | 85 | -57.75 | 146 | 874 | -37.50 | 227 | 8995 | -17.25 | 308 | 92572 | 3.00 |
| 66 | 87 | -57.50 | 147 | 899 | -37.25 | 228 | 9257 | -17.00 | 309 | 95275 | 3.25 |
| 67 | 90 | -57.25 | 148 | 926 | -37.00 | 229 | 9528 | -16.75 | 310 | 98057 | 3.50 |
| 68 | 93 | -57.00 | 149 | 953 | -36.75 | 230 | 9806 | -16.50 | 311 | 100921 | 3.75 |
| 69 | 95 | -56.75 | 150 | 981 | -36.50 | 231 | 10092 | -16.25 | 312 | 103868 | 4.00 |
| 70 | 98 | -56.50 | 151 | 1009 | -36.25 | 232 | 10387 | -16.00 | 313 | 106901 | 4.25 |
| 71 | 101 | -56.25 | 152 | 1039 | -36.00 | 233 | 10690 | -15.75 | 314 | 110022 | 4.50 |
| 72 | 104 | -56.00 | 153 | 1069 | -35.75 | 234 | 11002 | -15.50 | 315 | 113235 | 4.75 |
| 73 | 107 | -55.75 | 154 | 1100 | -35.50 | 235 | 11323 | -15.25 | 316 | 116541 | 5.00 |
| 74 | 110 | -55.50 | 155 | 1132 | -35.25 | 236 | 11654 | -15.00 | 317 | 119944 | 5.25 |
| 75 | 113 | -55.25 | 156 | 1165 | -35.00 | 237 | 11994 | -14.75 | 318 | 123447 | 5.50 |
| 76 | 117 | -55.00 | 157 | 1199 | -34.75 | 238 | 12345 | -14.50 | 319 | 127052 | 5.75 |
| 77 | 120 | -54.75 | 158 | 1234 | -34.50 | 239 | 12705 | -14.25 | 320 | 130762 | 6.00 |
| 78 | 123 | -54.50 | 159 | 1271 | -34.25 | 240 | 13076 | -14.00 | | | |
| 79 | 127 | -54.25 | 160 | 1308 | -34.00 | 241 | 13458 | -13.75 | | | |
| 80 | 131 | -54.00 | 161 | 1346 | -33.75 | 242 | 13851 | -13.50 | | | |

### 17.3.5.3 Summary of Changes from Neptune to Deep Eddy

The following represent changes in the Application of Attenuation portion of the PA function for Deep Eddy vs. Neptune:

- The attenuation control value is increased from 4 to 6 bits
- The volume LUT is increased from 81 entries to 321
- The volume LUT steps are decreased from 1dB to 0.25dB

- However, the user volume continues to have a step size of 1dB

The PA attenuation control value is shifted right by 2 bits prior to being subtracted from the user volume

# 18 Monitor ADC

## 18.1 Overview

Oceanside actively monitors the power supply to the class-D amplifier to compensate for direct connection to the system battery. As the battery voltage drops from the fully charged value to the discharged value, the battery monitor will adjust the Smart Amplifier Power Averaging circuit to accurately predict the power to the load (the circuit does not measure the voltage across the load directly. Rather, it uses the modulation depth and the measured PVDD voltage to determine the approximate power delivered to the load.)
The battery monitor circuit will measure the PVDD voltage and adjust the signal so that the power delivered to the load is within +/-50mW of the target value (100mW granularity).

## 18.2 Block I/O Signals

**Figure 18.2-1 MADC I/O Signals**

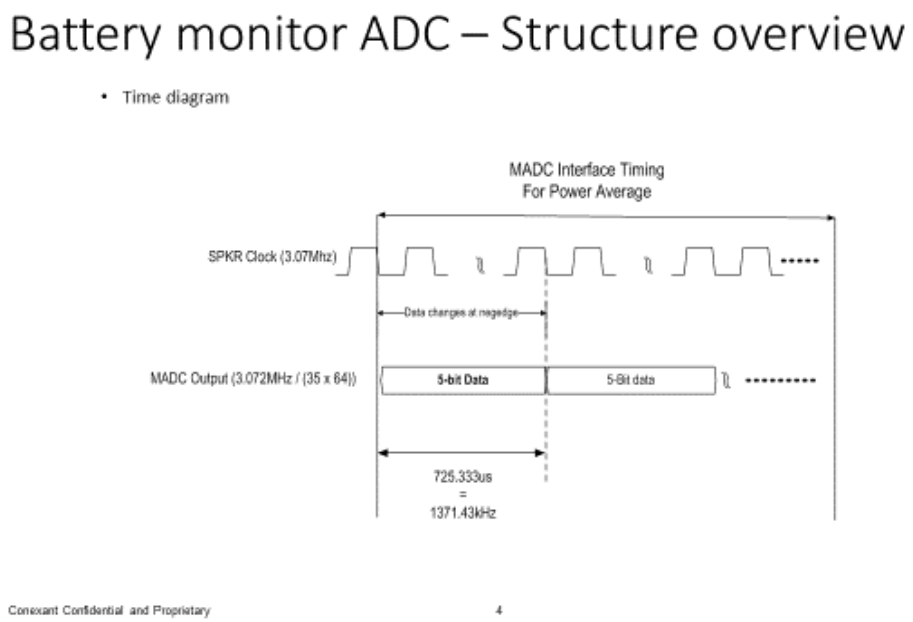
| Port name | Type | Level [V] | Description |
|---|---|---|---|
| Avdd | Analog bias | 1.8 | Power supply |
| Avss | Analog ground | 0 | Clean ground |
| Avsub | Analog ground | 0 | Substrate connection |
| Madc_en | Digital input | 0/1.8 | Enable and pdb |
| Madc_ibn | Analog bias | 0/1.8 | Analog bias current (1uA) |
| Spkr_clk | Digital input | 0/1.8 | Class-D input clock (3.072MHz) |
| Vin | Analog input | 2.5 – 5.6 | Battery output voltage |
| Vref_bg | Analog input | 1.3875 | DAC voltage reference (5.55/4 = 1.3875) |
| Vmid | Analog input | 0.9 | Analog ground (Avdd/2 = 0.9V) |
| Madc_code<4:0> | Digital output | 0/1.8 | ADC output (5 bit refreshed @48kHz) |

## 18.3 Electrical Spec and Area

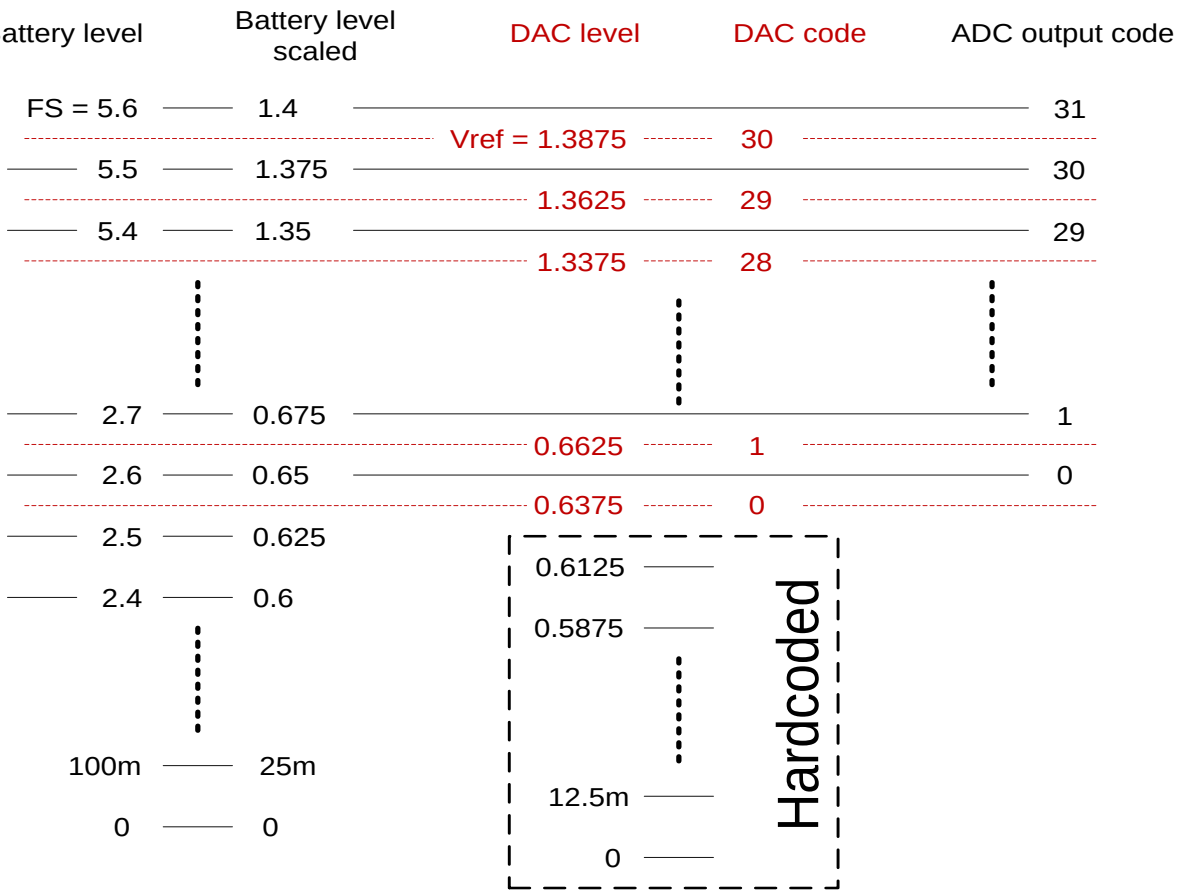
| Parameter | Test conditions/Note | MIN | TYP | MAX | Unit |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| Supply voltage | | 1.6 | 1.8 | 2.0 | V |
| Junction temperature range | | -40 | 27 | 125 | C |
| Bias current | Coming from bandgap | | 1u | | A |
| Total current consumption | | | 35u | | A |
| Area | Constrain on aspect ratio | | 20u x 650u = 0.013 | | mm2 |

## 18.4 Analog to Digital Interface Timing




## 18.5 Battery Monitor ADC Structure Overview




## 18.6 Testability

Refer to DFT Chapter. Link Below

PA MADC Monitor Mode

# 19 Port C Analog Input Operation

## 19.1 Port C Analog Input Operation

By default, Port C is a stereo digital microphone interface. If desired, the port may be configured as an analog input port and used for analog microphone or line level inputs.
The port configuration, however, is not intended to be dynamic. The system firmware or driver should configure this port at power-on and it should remain in that mode during normal operation.
When in digital input mode, the port may connect to one or two digital microphones. The port is visible to both the ADC1 and ADC2 converter widgets. Microphone bias is not supported. Due to the nature of this configurable port, the microphone data input is a nominal 10K ohms.
When in analog input mode, the port may be used for both analog microphone or line inputs. The port is only supported by the ADC1 converter widget connection list and is not available to ADC2. Microphone bias is only available when port A and Port B are in merge mode. Otherwise, an external supply is required to power the analog microphone. (See next section.)

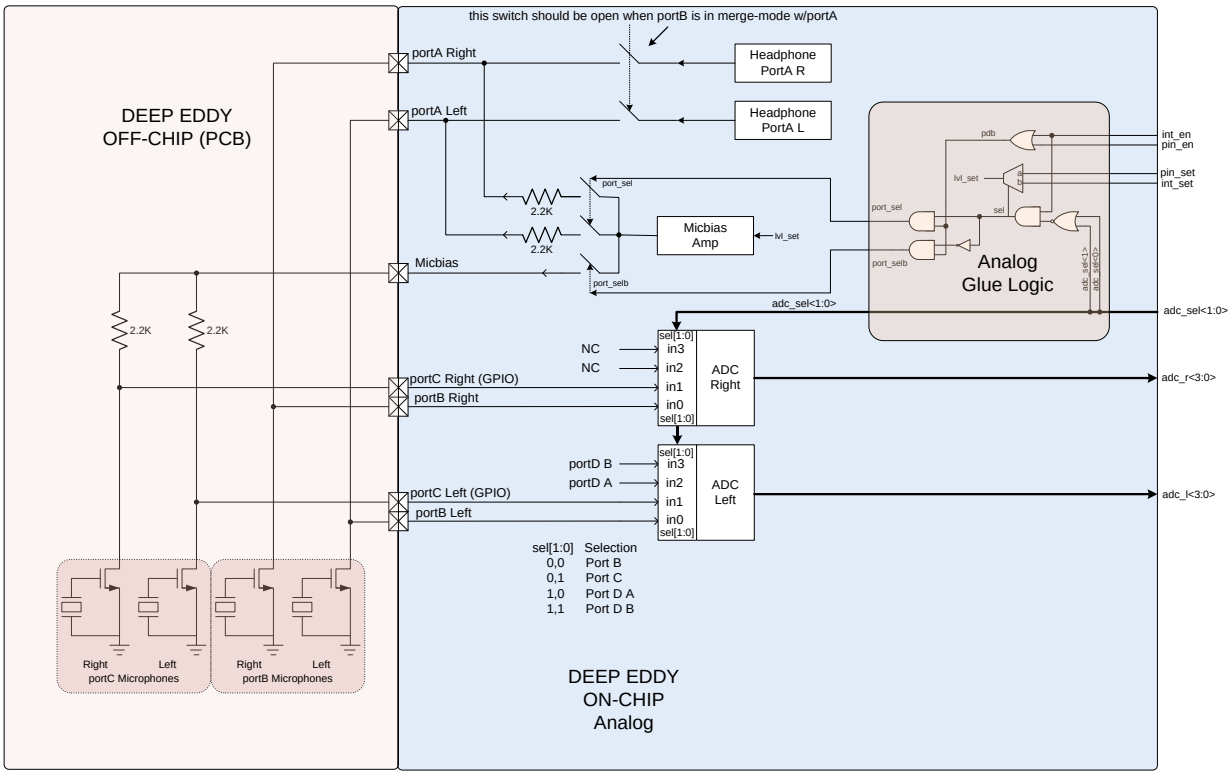
## 19.2 Microphone Bias Pin

The MICBIAS output pin services both Port B and Port C
The default condition for Port B is an independent input port with microphone capability while the default configuration for Port C is a digital microphone input. Port B may also be tied to port A using the "merge mode" and port C may be configured as an analog input. Since there is only one external microphone bias pin, this can create a conflict.
By default, the external microphone bias pin shall provide the bias for Port B (Port C being a digital microphone input). If Port B is configured for merge mode, the bias for port B will be supplied internally on resistors connected to the headphone Left and Right outputs. Port C may then use the micbias pin if it is configured as an analog input. If port B is not configured in merge mode, and port C is in analog input mode, the mic bias pin shall be controlled by the port B controls.

| Port B state | Port C state | Port B bias | Port C bias | MICBIAS PIN Controlled by |
|---|---|---|---|---|
| Independent | Digital mic | Pin | NA | PortB |
| Independent | Analog mic | Pin | Not supported (external supply needed) | PortB |
| Merged with PortA | Digital mic | Internal/Pin* | NA | Hi-Z/PortB* |
| Merged with PortA | Analog mic | Internal/Pin* | Pin/External* | PortC/PortB* |

Note: * Test bit provided to force use of external pin for port B in merge mode.

## 19.3 Micbias Circuit Diagram




Conexant Systems Inc
Proprietary and Confidential
ALL INFORMATION IN THIS DOCUMENT IS PRELIMINARY AND SUBJECT TO CHANGE WITHOUT NOTICE

# 20 EQ and DRC

## 20.1 EQ Coefficient Read Back

New coefficient read back logic was added in Neptune device. This logic was implemented for coefficient read of ADC EQ, DAC 7-Band EQ, DAC 2-Band EQ and PID.
Key design features of this logic:

- Reuse of coefficient logic and write state machine.
- Reuse of write registers for EQ
- When EQ in lock mode or force read mode, the Coeff latch enable signal work as read signal.

### 20.1.1 Register Spec

EQ Enable and Bypass MSB register
Get - HD_A[27:0] = 0x1BA0100 Set - HD_A[27:8] = 0x1B201 I2C=6D01

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Force coeff read mode (always read 0)<br>0: no force<br>1: force to coeff read mode |
| 6 | 1 | Clear ADC BQ memory when disable/enable<br>1: Clear memory enable<br>0: Clear memory disable |
| 5 | 1 | Clear DAC BQ memory when disable/enable<br>1: Clear memory enable<br>0: Clear memory disable |
| 4 | 1 | Clear 7B EQ memory when disable/enable<br>1: Clear memory enable<br>0: Clear memory disable |
| 3 | 0 | Biquad ADC channel1 bypass bit.<br>0: Biquad ADC channel 1 is bypass mode.<br>1: Biquad ADC channel 1 is normal mode |
| 2 | 0 | Biquad ADC channel 0 enable bit.<br>0: Biquad ADC channel 0 is disabled.<br>1: Biquad ADC channel 0 is enabled. |
| 1 | 1 | Biquad mono DAC bypass bit.<br>1: Biquad mono DAC is bypass mode.<br>0: Biquad mono DAC is normal mode |
| 0 | 0 | Biquad mono DAC enable bit.<br>0: Biquad mono DAC is disabled.<br>1: Biquad mono DAC is enabled. |

EQ Band Control register
Get - HD_A[27:0] = 0x1BA0D00 Set - HD_A[27:12] = 0x1B20D I2C=6D0D

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Read only coeff_done |
| 6 | 0 | Coeff latch enable to start the write/read process. Once complete, back to 0.<br>In write mode, when set to 1, the current values of B0,B1,B2,A1,A2,G, P1, P2 are loaded into EQ Memory based on bits [5:3] (channel number) and IIR band location based on [2:0].<br><br>In read mode, when set to 1, coeff will be read out from EQ memory based on bits [5:3] (channel number) and IIR band location based on [2:0]. Host can read B0,B1,B2,A1,A2,G, P1, P2 after process complete(within one frame). |
| 5:3 | 0 | EQ Channel selection bits for loading B0, B1, B2,A1,A2,G and P1,P2<br>0 – Ch0 coeff's are loaded into 7B EQ memory |

| | | |
|---|---|---|
| | | 1 – Ch1 coeff's are loaded into 7B EQ memory<br>2- Mono DAC subwoofer coeff's are loaded into Biquad memory (2 Band only)<br>3-MIC Ch0 coeff's are loaded in IIR memory<br>4-MIC Ch1 coeff's are loaded in IIR memory<br>5- PID channel 0<br>6- PID channel 1 |
| 2:0 | 0 | EQ IIR Band selection bits for loading B0,B1,B2,A1,A2 and P1, P2.<br>0 – Band 0 is loaded with programmed B0, B1, …P1, P2 values.<br>1 – Band 1 is loaded with programmed B0, B1, …P1, P2 values.<br>2 – Band 2 is loaded with programmed B0, B1, …P1, P2 values.<br>3 – Band 3 is loaded with programmed B0, B1, …P1, P2 values.<br>4 – Band 4 is loaded with programmed B0, B1, …P1, P2 values.<br>5 – Band 5 is loaded with programmed B0,B1, …P1, P2 values<br>6 – Band 6 is loaded with programmed B0,B1, …P1, P2 values<br>Subwoofer and Mic Biquads are<br>Program bit 0 and 1 of 1b20000 to 1 and then start loading values into BiQuads.<br>When EQ channel selection is PID channel 0 or PID channel 1, B0,B1,B2,A1,A2 and P1, P2 representation are showing below.<br>0 – SA 3Biqurd IIR Band0 is loaded with B0,B1,B2,A1,A2,G<br>P1 is loaded with LT_H coefficient(limit temperature MSB used in error-term),P2 is loaded with LT_L coefficient(limit temperature LSB used in error-term)<br>1 – SA 3Biqurd IIR Band1 is loaded with B0,B1,B2,A1,A2,G<br>P1 is loaded with I0 coefficient<br>P2 is loaded with I1 coefficient<br>2 – SA 3Biqurd IIR Band2 is loaded with B0,B1,B2,A1,A2,G<br>P1 is loaded with I2coefficient<br>P2 is loaded with C coefficient<br>3 – B0 is loaded with ADJ coefficient value used for adjustment coefficient of power calculation<br>B1 is loaded with KI1 coefficient<br>B1 is loaded with KI2 coefficient<br>A1 is loaded with KD coefficient<br>A2 is loaded with KP coefficient<br>G is loaded with {IV_abs,IV mode[1:0]},the same both in PID channel 0 and PID channel 1<br>IV_abs=1'b0, means no absolute option<br>IV_asb=1'b1, means absolute option<br>IV_mode=2'b00, means I*V option<br>IV_mode=2'b01, means I*I option<br>IV_mode=2'b10, means V*V option<br>IV_mode=2'b11, means DAC*DAC option<br>P1 is loaded with PIDG0 coefficient<br>P2 is loaded with PIDG1 coefficient |

## 20.2 Critical Configuration Locking

Maintaining correct configuration settings is important for reliable SA-2 operation. A number of features are provided for the purpose of avoiding corruption of configuration settings.

### 20.2.1 2.7.1 Register Locking

The predecessor device (Solana) includes a feature by which registers related to EQ and DRC configuration may be locked. While locked, these registers may be read, but not written. The lock mechanism is controlled within the ***EQ Enable and Bypass LSB*** register. A two-step sequence is written to bits 7:4 of this register to enable the lock (lock the protected registers). A separate two-step sequence is written to bits 7:4 to disable the lock (unlock the protected registers). The Neptune device expands the lock mechanism to include additional registers. The complete list of registers that are subject to the lock are as follows:

- EQ Enable and Bypass LSB (except bits 7:4, which are used to enable/disable the lock)
- EQ Enable and Bypass MSB
- EQ B0 coefficient LSB
- EQ B0 coefficient MSB
- EQ B1 coefficient LSB
- EQ B1 coefficient MSB
- EQ B2 coefficient LSB
- EQ B2 coefficient MSB
- EQ A1 coefficient LSB
- EQ A1 coefficient MSB
- EQ A2 coefficient LSB
- EQ A2 coefficient MSB
- EQ G coefficient
- EQ Band Control LSB
- One-Way Lock
- SPKR DRC Enable Step Control
- SPKR DRC Fast Release Delay and Update
- SPKR DRC Boost
- SPKR DRC Balance Ramp Step
- SPKR DRC Volume Step
- SPKR DRC Release Rate
- SPKR DRC Release Delay/Programmable Slope
- SPKR DRC Max Lineout
- SPKR DRC Test
- SPKR DRC Gain Read LSB
- SPKR DRC Gain Read MSB

# 21 Record Mixer

## 21.1 Overview

This section describes the record mixer in Oceanside. There are two types of record mixers in Oceanside. The Pre-Mixer and the Post-Mixer. In both Mixer modes the following functions apply:

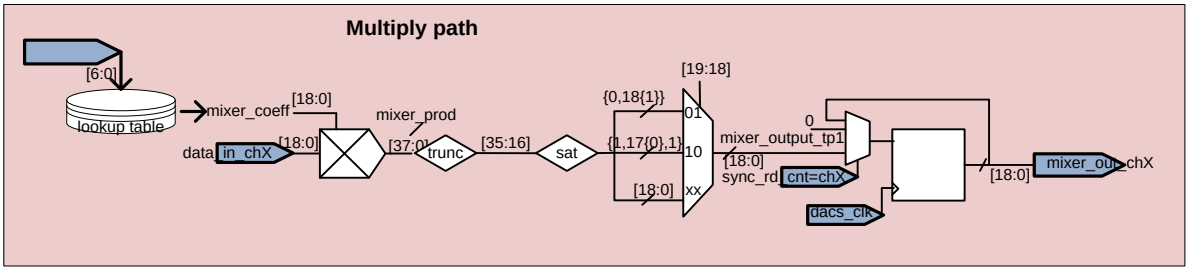
1. The 2 left playback channels (DAC1 and DAC2) will be added and drive to the left channel of record path (ADC Path).
2. The 2 right playback channels (DAC1 and DAC2) will be added and drive to the right channel of record path (ADC path).

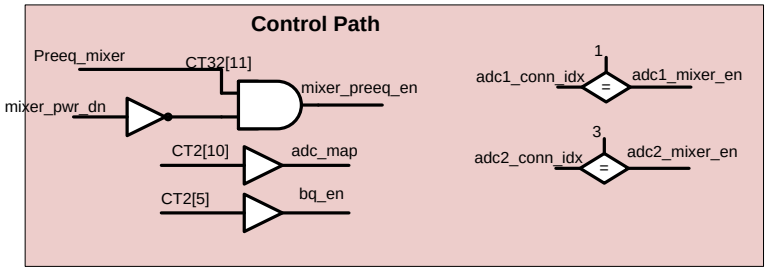
## 21.2 Micro Architecture

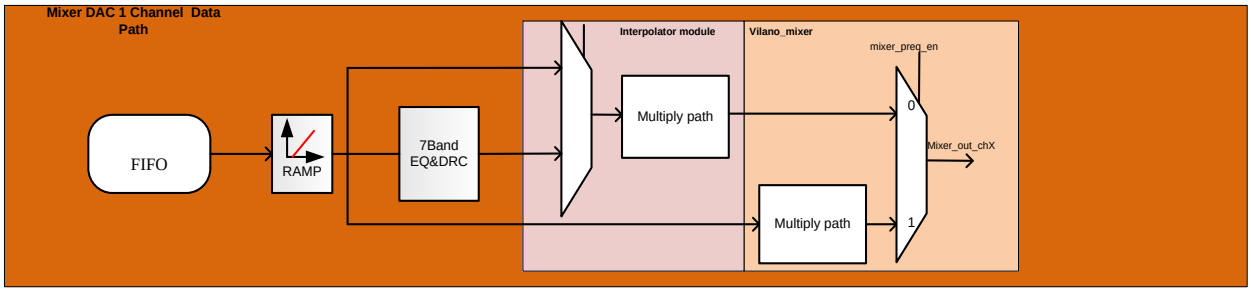
### 21.2.1 Multiply Path and Control

The Diagram Below Shows the Multiplier Path and control logic for Both Type mixers.
The Post-Mixer Multiply path is part of Oceanside Interpolator Block.
The Pre-Mixer, Multiply Path is included in the Oceanside Vilano_mixer module.

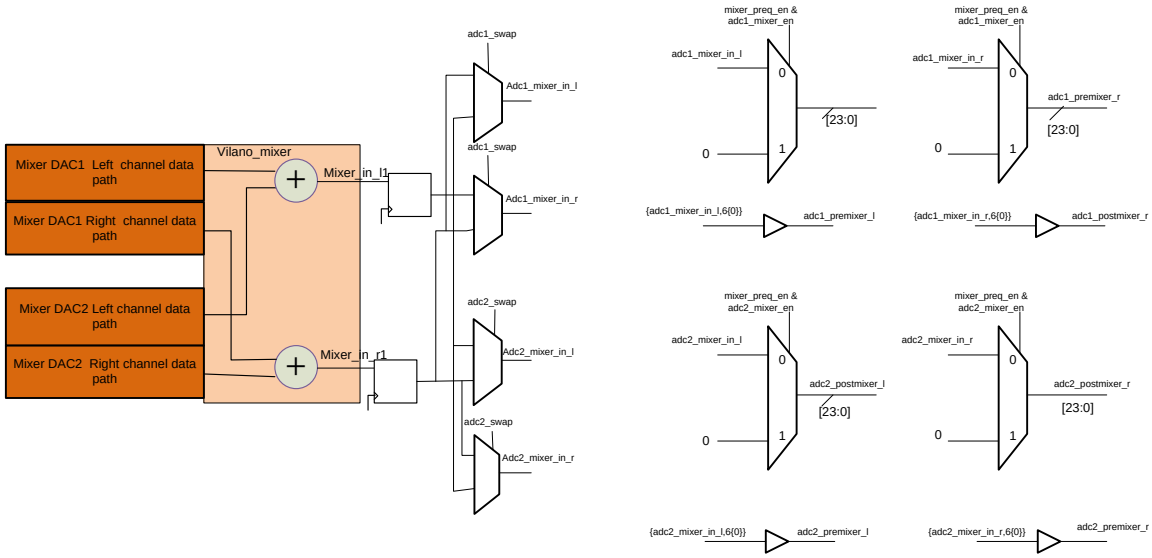
The Mixer Function for Both Pre and Post is inside the Vilano_mixer Module.

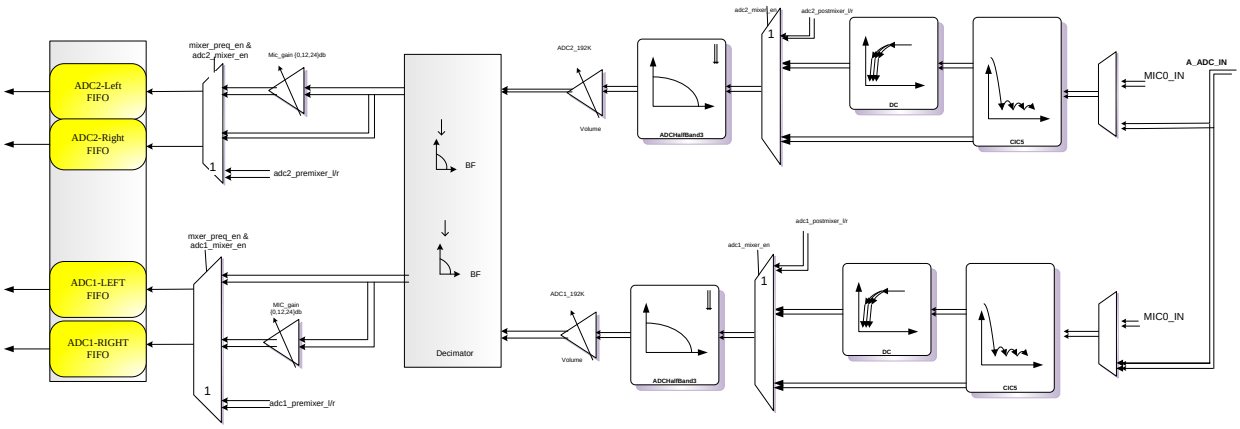



## 21.3 Pre-Mixer and Post Mixer Data Path Diagram




## 21.4 Node 15: Mixer Widget

Table21-1 Node 15 responses

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Get index 0 amp gain | 0xB00<br>0xB20 | 0x00 | 0x000000aa<br>0x000000aa | 0x0000004A<br>(P,W) | a = Right gain<br>a = Left gain |
| Get index 1 amp gain | 0xB00<br>0xB20 | 0x01 | 0x000000aa<br>0x000000aa | 0x0000004A<br>(P,W) | a = Right gain<br>a = Left gain |
| Set index 0 amp gain | 0x350<br>0x360<br>0x370 | 0xaa | 0x00000000 | | a = Right gain<br>a = Left gain<br>a = Left & Right gain |
| Set index 1 amp gain | 0x351<br>0x361<br>0x371 | 0xaa | 0x00000000 | | a = Right gain<br>a = Left gain<br>a = Left & Right gain |

| | | | | | |
|---|---|---|---|---|---|
| Audio widget Mixer | 0xF00 | 0x09 | 0x00F0050B | | Vendor defined with input amplifier |
| Input amp capabilities | 0xF00 | 0x0D | 0x80034A4A | | Mute, 1db step, 74 steps, step 74 is 0db |
| Connection length | 0xF00 | 0x0E | 0x00000002 | | Connected to 2 |
| Supported Power states | 0xF00 | 0x0F | 0x8000000F | | EPSS, D0, D1, D2 and D3 |
| Get connection list | 0xF02 | 0x00 | 0x00001110 | | Connected to DAC1, DAC2 |
| Get power state | 0xF05 | 0x00 | 0x00000abc | 0x00000433 (P,W) | a = Settings Reset<br>b = Actual state<br>c = Requested state<br>Settings Reset cleared by this verb or any write to this node. |
| Set power state | 0x705 | 0x0a | 0x00000000 | | a = Requested state |

## 21.5 Mixer Gain

-74db to +6db dB in 1 dB steps, plus mute control.

## 21.6 Register Spec

### 21.6.1 CODEC_TEST32

Get Codec Test32 Response - HD_A [27:0] =0x1C98000
Set Codec Test30 - HD_A[27:11] = 0x1C18000

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | 1-Enable PRE EQ Mixing |
| 10 | 0 | PRE EQ Mixing Base Select<br>0: Base on DAC1 signals<br>1: Base on DAC2 signals |
| 9 | 1 | Interpolator mute/unmute disable.<br>1: Mute/unmute gain disable.<br>0: Mute/Un-mute enable |
| 8 | 0 | Bypass ED delay buffer<br>1: Bypass ED delay buffer<br>0: No Bypass |
| 7 | 1 | Override value of cpmp_ctrl_dmode |
| 6 | 0 | Override ED output to AFE<br>0: Override<br>1: No override |
| 5 | 0 | PortM on D3 live<br>0: Disable<br>1: Enable |
| 4 | 0 | PortE on D3 live<br>0: Disable<br>1: Enable |
| 3 | 0 | Override ED output enable value<br>0: Disable ED output<br>1: Enable ED output |

| | | |
|---|---|---|
| 2 | 0 | Override for ED output enable<br>0: No override.<br>1: Enable ED output en override. |
| 1 | 0 | Override ED port selection value<br>0: DAC1<br>1: DAC2 |
| 0 | 0 | Override for ED port selection<br>0: No override<br>1: Enable ED port selection override |

# 22 SnM (Music Request Mode)

## 22.1 SnM Overview

Oceanside provides Mic in data looped back to class-D power amp while device is in sleep mode.
It uses the port B to feed in Mic data and use analog and digital stereo ADC path and loop back to Stereo DAC to the Headphone (If headphone plugged in) or Class D port (default). Below is the diagram of the SnM H/W implementation.
SnM mode uses the internal RC Oscillator for input to PLL (since in sleep mode clock is not available).

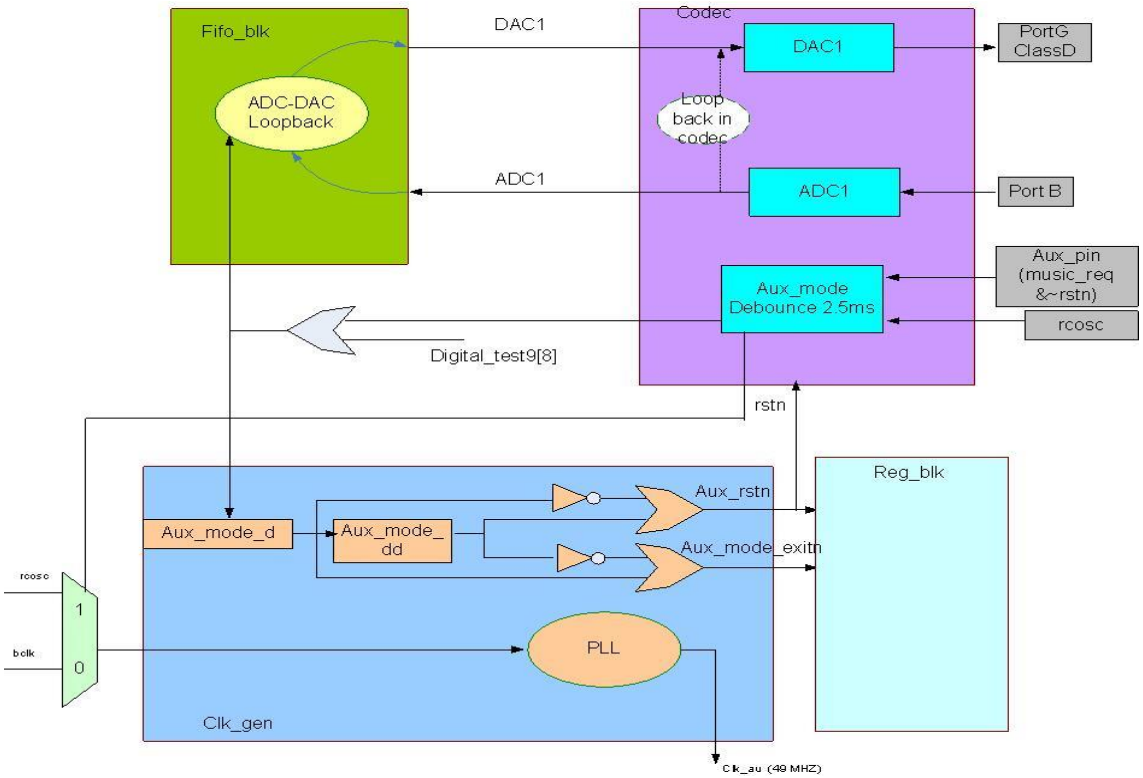



**Figure 22.1-1 SnM Functional Block Diagram**

Low power music mode is entered when HD Audio reset is low and MusicReq is high. If HD Audio reset goes high or MusicReq goes low, then the device will exit low power music mode. If we need a BIOS control bit to enable this mode, then we need to get agreement. We will also need to deal with power supply sequencing to save power. Upon Entering music request an aux_rstn (Known as D3 Live reset) pulse is generated that will set the default below register values.
Note: this table is written from the read perspective – what the register contents would be if you read the register.

| Node | Verb | Music mode | Description |
|---|---|---|---|
| 0x01 | 0xF05 | 0x00000400 | Power up codec |
| 0x10 | 0xA | 0x00000031 | 48k / 24-bit (POR default) |
| 0x10 | 0xB80<br>0xBA0 | 0x0000004A | Gain 0db (POR default)<br>Consider an option to keep the user gain setting |

| | | | |
|---|---|---|---|
| 0x10 | 0xF05 | 0x00000400 | Power up DAC |
| 0x13 | 0xF05 | 0x00000400 | Power up ADC |
| 0x13 | 0xA | 0x00000031 | 48k / 24-bit (POR default) |
| 0x13 | 0xB0000<br>0xB2000 | 0x0000004A | Index 0 gain 0db (POR default)<br>Consider an option to keep the user gain setting |
| 0x13 | 0xF01 | 0x00000000 | Index 0 (input from node 18, POR default) |
| 0x18 | 0xF05 | 0x00000400 | Power up input port |
| 0x18 | 0xB0000<br>0xB2000 | 0x00000000 | 0db boost (POR default)<br>Consider an option to keep the user gain setting |
| 0x18 | 0xF07 | 0x00000020 | Input enable |
| 0x17 | 0xF01 | 0x00000000 | Index 0 (input from node 10, POR default) |
| 0x17 | 0xF05 | 0x00000400 | Power up the port |
| 0x17 | 0xF07 | 0x00000040 | Output enable (POR default) |

Other Port could be used in Music request mode: See tables below for setup.

| PortM | | | |
|---|---|---|---|
| Music_req & ~rstn | Codec_test32[5] | Portm_con_indx | Portm_out_en |
| 1 | 1 | 1 | 1 |

| PortE | | | |
|---|---|---|---|
| Music_req & ~rstn | Codec_test32[4] | Porte_con_indx | Porte_out_en |
| 1 | 1 | 0 | 1 |

| PortA | | | | |
|---|---|---|---|---|
| Music_req & ~rstn | Digital_bios_test4[11] OR porta_presence | Porta_con_idx | Hp_hn_out_en[1:0] | Analog_test12 |
| 1 | 1 | 0 | 11 | 0x006 (-6db) |

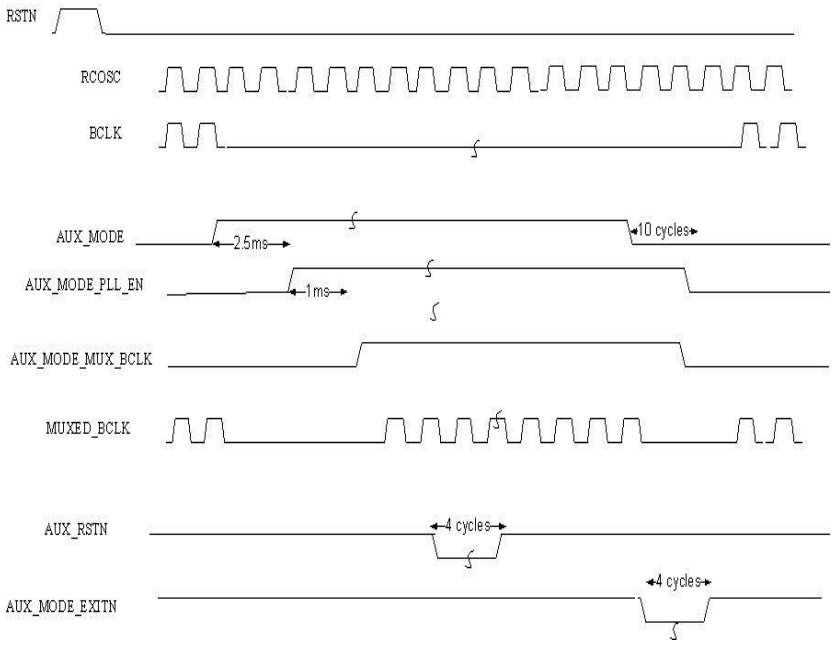
| PortG (condition when portA present) | | |
|---|---|---|
| Music_req & ~rstn | Digital_bios_test4[11] OR porta_presence | Portg_pwr_st |
| 1 | 1 | D3 |

## 22.2 Entering SnM Mode

SnM is enabled when GPIO0 is pulled high and the following condition exists:

1- gpio_en[0] = 0;
2- Digital_test1[10:9] =1;
3- Link reset = 0;

Diagram below shows the Timing for entering SnM and exiting SnM.


RSTN
RCOSC
BCLK
AUX_MODE
2.5ms
10 cycles
AUX_MODE_PLL_EN
1ms
AUX_MODE_MUX_BCLK
MUXED_BCLK
AUX_RSTN
4 cycles
AUX_MODE_EXITN
4 cycles

# 23HD Audio Widgets and Verbs

Red text indicates registers/widgets that are removed in Oceanside but were in Deep Eddy.
Green Highlighting shows registers/bits that are added or changed in Oceanside-A0
Yellow highlighting is used to point out items of interest or importance.

## 23.1 Widgets/verbs (registers) Removed

## 23.2 Widgets/verbs (registers) Additions or changes

Port H Widget (DMIC 2) added
Port E Widget (I2S) added
Port F Widget (I2S) added
Port J Widget (I2S) added
Port K Widget (I2S) added
DAC3 converter widget added
DAC3 converter widget added
ADC4 converter (I2S) widget added
ADC5 converter (I2S) widget added
I2S mapping registers to handle peripheral to host slot mapping

## 23.3 HD Audio verb to I2C Address Mapping

Table Below Shows the Mapping of an HDA Verb (Address) to I2C Address.

**Table 23-1: Verb to register address mapping**

| Verb | Verb Code (Parameter ID) | I2C Derived Address Field | I2C Address[9:0] |
|---|---|---|---|
| Get Parameter | F00 | 0x80 + Parameter Index | |
| Vendor ID | F00 (00) | 0x80 | 0x200 |
| Revision ID | F00 (02) | 0x82 | 0x208 |
| Subordinate Node Count | F00 (04) | 0x84 | 0x210 |
| Function Group Type | F00 (05) | 0x85 | 0x214 |
| Audio Function Group Capabilities | F00 (08) | 0x88 | 0x220 |
| Audio Widget Capabilities | F00 (09) | 0x89 | 0x224 |
| Supported PCM Size, Rates | F00 (0A) | 0x8A | 0x228 |
| Supported Stream Formats | F00 (0B) | 0x8B | 0x22C |
| Pin Capabilities | F00 (0C) | 0x8C | 0x230 |
| Amplifier Capabilities (input amp) | F00 (0D) | 0x8D | 0x234 |
| Amplifier Capabilities (output amp) | F00 (12) | 0x92 | 0x238 |
| Connection List Length | F00 (0E) | 0x8E | 0x23C |
| Supported Power States | F00 (0F) | 0x8F | 0x240 |
| Processing Capabilities | F00 (10) | 0x90 | 0x244 |
| GPIO Count | F00 (11) | 0x91 | 0x248 |
| Volume Knob Capabilities | F00 (13) | 0x93 | 0x24C |
| HD MI LPCM CAD (obsolete) | F00 (20) | 0xA0 | 0x280 |
| Get Connection Select Control | F01 | 0x01 | 0x4 |
| Set Connection Select Control | 701 | 0x01 | 0x4 |
| Get Connection List Entry | F02 | 0x40 + (Offset Index >> 2) | |
| Offset Index = 0 | F02 | 0x40 | 0x100,0x101,0x102,0x103 |
| Offset Index = 4 | F02 | 0x41 | 0x104,0x105,0x106,0x107 |

| Verb | Verb Code (Parameter ID) | I2C Derived Address Field | I2C Address[9:0] |
|---|---|---|---|
| Offset Index = 8 | F02 | 0x42 | 0x108,0x109,0x10A,0x10B |
| Offset Index = C | F02 | 0x43 | 0x10C,0x10D,0x10E,0x10F |
| Get Processing State | F03 | 0x03 | 0xC |
| Set Processing State | 703 | 0x03 | 0xC |
| Get Coefficient Index | D | 0xF5 | 0x3D4 |
| Set Coefficient Index | 5 | 0xF5 | 0x3D4 |
| Get Processing Coefficient | C | 0xF4 | 0x3D0 |
| Set Processing Coefficient | 4 | 0xF4 | 0x3D0 |
| Get Amplifier Gain/Mute | B | 0x60 + { O/I, L/R, Index[2:0] } | 0x180 + {O/I,L/R,Index[2:0],0,0} |
| Set Amplifier Gain/Mute | 3 | 0x60 + { O/I, L/R, Index[2:0] } | 0x180 + {O/I,L/R,Index[2:0],0,0} |
| Get Converter Format | A | 0xF2 | 0x3C8 |
| Set Converter Format | 2 | 0xF2 | 0x3C8 |
| Get Digital Converter Control | F0D | 0x0D | 0x34,0x35,0x36,0x37 |
| Set Digital Converter Control 1 | 70D | 0x0D | 0x34 |
| Set Digital Converter Control 2 | 70E | 0x0D | 0x35 |
| Set Digital Converter Control 3 | 73E | 0x0D | 0x36 |
| Set Digital Converter Control 4 | 73F | 0x0D | 0x37 |
| | | | |
| Get Power State | F05 | 0x05 | 0x14,0x15 |
| Set Power State | 705 | 0x05 | 0x14 |
| Get Converter Stream, Channel | F06 | 0x06 | 0x18 |
| Set Covnerter Stream, Channel | 706 | 0x06 | 0x18 |
| Get Input Converter SDI Select | F04 | 0x04 | 0x10 |
| Set Input Converter SDI Select | 704 | 0x04 | 0x10 |
| Get Pin Widget Control | F07 | 0x07 | 0x1C |
| Set Pin Widget Control | 707 | 0x07 | 0x1C |
| Get Unsilicited Response | F08 | 0x08 | 0x20 |
| Set Unsolicited Response | 708 | 0x08 | 0x20 |
| Get Pin Sense (analog pin) | F09 | 0x09 | 0x24,0x25,0x26,0x27 |
| Execute Pin Sense (analog pin) | 709 | 0x09 | 0x24 |
| Get Pin Sense (digital pin) | F09 | 0x09 | 0x24,0x25,0x26,0x27 |
| Execute Pin Sense (digital pin) | 709 | 0x09 | 0x24 |
| Get EAPD/BTL Enable | F0C | 0x0C | 0x30 |
| Set EAPD/BTL Enable | 70C | 0x0C | 0x30 |
| Get GPI Data | F10 | 0x10 | 0x40 |
| Set GPI Data | 710 | 0x10 | 0x40 |
| Get GPI Wake Enable Mask | F11 | 0x11 | 0x44 |
| Set GPI Wake Enable Mask | 711 | 0x11 | 0x44 |
| Get GPI Unsolicited Enable | F12 | 0x12 | 0x48 |
| Set GPI Unsolicited Enable | 712 | 0x12 | 0x48 |
| Get GPI Sticky Mask | F13 | 0x13 | 0x4C |
| Set GPI Sticky Mask | 713 | 0x13 | 0x4C |
| Get GPO Data | F14 | 0x14 | 0x50 |
| Set GPO Data | 714 | 0x14 | 0x50 |
| Get GPIO Data | F15 | 0x15 | 0x54 |
| Set GPIO Data | 715 | 0x15 | 0x54 |
| Get GPIO Enable Mask | F16 | 0x16 | 0x58 |
| Set GPIO Enable Mask | 716 | 0x16 | 0x58 |
| Get GPIO Direction | F17 | 0x17 | 0x5C |
| Set GPIO Direction | 717 | 0x17 | 0x5C |
| Get GPIO Wake Enable | F18 | 0x18 | 0x60 |
| Set GPIO Wake Enable | 718 | 0x18 | 0x60 |
| Get GPIO Unsolicited Enable Mask | F19 | 0x19 | 0x64 |
| Set GPIO Unsolicited Enable Mask | 719 | 0x19 | 0x64 |
| Get GPIO Sticky Mask | F1A | 0x1A | 0x68 |
| Set GPIO Sticky Mask | 71A | 0x1A | 0x68 |
| Get Beep Generation | F0A | 0x0A | 0x28 |
| Set Beep Generation | 70A | 0x0A | 0x28 |
| Get Volume Knob | F0F | 0x0F | 0x3C |
| Set Volume Knob | 70F | 0x0F | 0x3C |

| Verb | Verb Code (Parameter ID) | I2C Derived Address Field | I2C Address[9:0] |
|---|---|---|---|
| Get Implementation Identification | F20 | 0x20 | 0x80,0x81,0x82,0x83 |
| Set Implementation Identification 1 | 720 | 0x20 | 0x80 |
| Set Implementation Identification 2 | 721 | 0x20 | 0x81 |
| Set Implementation Identification 3 | 722 | 0x20 | 0x82 |
| Set Implementation Identification 4 | 723 | 0x20 | 0x83 |
| Get Configuration Default | F1C | 0x1C | 0x70,0x71,0x72,0x73 |
| Set Configuration Default 1 | 71C | 0x1C | 0x70 |
| Set Configuration Default 2 | 71D | 0x1C | 0x71 |
| Set Configuration Default 3 | 71E | 0x1C | 0x72 |
| Set Configuration Default 4 | 71F | 0x1C | 0x73 |
| Get Stripe Control | F24 | 0x24 | 0x90,0x91,0x92,0x93 |
| Set Stripe Control | 724 | 0x24 | 0x90 |
| Execute Function Reset | 7FF | 0xFF | 0x3FC |
| Get ELD Data | F2F | 0x2F | 0xBC |
| Get Converter Channel Count | F2D | 0x2D | 0xB4 |
| Set Converter Channel Count | 72D | 0x2D | 0xB4 |
| Get DIP Size | F2E | 0x2E | 0xB8 |
| Get DIP Index | F30 | 0x30 | 0xC0 |
| Set DIP Index | 730 | 0x30 | 0xC0 |
| Get DIP Data | F31 | 0x31 | 0xC4 |
| Set DIP Data | 731 | 0x31 | 0xC4 |
| Get DIP XmitCtrl | F32 | 0x32 | 0xC8 |
| Set DIP XmitCtrl | 732 | 0x32 | 0xC8 |
| Get Protection Control | F33 | 0x33 | 0xCC,0xCD |
| Set Protection Control | 733 | 0x33 | 0xCC |
| Get ASP Channel Mapping | F34 | 0x34 | 0xD0 |
| Set ASP Channel Mapping | 734 | 0x34 | 0xD0 |

## 23.4 General notes on verbs in this document

A verb is defined in the HD Audio specification as consisting of a codec ID, a node ID, a verb and a parameter. In the following tables and descriptions, the codec ID and the Node ID will not be listed as part of the verb. The codec ID is assigned during device discovery and therefore isn't constant or known. The Node ID will be given in the title of each of the subsections. The entire verb can be constructed by pre-pending the Codec ID and the Node ID to the information given in each section.

Register default values may have up to five letters included with the default value. These letters indicate which of the five different possible resets will force the register to this default value. The five possibilities are "P" power on reset, "R" HD Audio bus reset, "V" single verb reset, "W" double verb reset and "D" D-state changes. Only the letters in the list will force the register to its default value.

EPSS D-state to power state mapping scheme is used.

| Actual Power state | Device Behavior |
|---|---|
| 0 | Full power |
| 1 | Low Power 1 – 1mS recovery |
| 2 | Low Power 1 – 1mS recovery |
| 3 | Low Power 2 – 10mS recovery<br>+75mS to full fidelity |
| 4 | Low Power 3 – 200mS recovery |

When an unsupported bit pattern is written to a field there is two valid responses. 1) Retain the existing valid value in the field. 2) Set the field to all zeroes. For example, invalid writes to microphone bias or PGA gain settings must either retain the last valid value or revert to all zeroes.

## 23.5 Low Power Music Mode

Low power music mode is entered when HD Audio reset is low and D3 LIVE is high. When a reset pin is not available (I2S/I2C mode or mHDA mode without nRST) then the loss of SYNC (LRCLK) will substitute for bus reset as mentioned in ECR HDA048-A. If HD Audio reset goes high or D3 LIVE goes low, then the device will exit low power music mode. If we need a BIOS control bit to enable this mode, then we need to get agreement from the customer. We will also need to deal with power supply sequencing to save power. When low power music mode is active the following controls should be set.

Note: this table is written from the read perspective – what the register contents would be if you read the register.

| Node | Verb | Music mode | Description |
|---|---|---|---|
| 0x01 | 0xF05 | 0x00000400 | Power up codec |
| 0x10 | 0xA | 0x00000031 | 48k / 24-bit (POR default) |
| 0x10 | 0xB80<br>0xBA0 | 0x0000004A | Gain 0db (POR default)<br>Consider an option to keep the user gain setting |
| 0x10 | 0xF05 | 0x00000400 | Power up DAC |
| 0x13 | 0xF05 | 0x00000400 | Power up ADC |
| 0x13 | 0xA | 0x00000031 | 48k / 24-bit (POR default) |
| 0x13 | 0xB0000<br>0xB2000 | 0x0000004A | Index 0 gain 0db (POR default)<br>Consider an option to keep the user gain setting |
| 0x13 | 0xF01 | 0x00000000 | Index 0 (input from node 18, POR default) |
| 0x18 | 0xF05 | 0x00000400 | Power up input port |
| 0x18 | 0xB0000<br>0xB2000 | 0x00000000 | 0db boost (POR default)<br>Consider an option to keep the user gain setting |
| 0x18 | 0xF07 | 0x00000020 | Input enable |
| 0x17 | 0xF01 | 0x00000000 | Index 0 (input from node 10, POR default) |
| 0x17 | 0xF05 | 0x00000400 | Power up the Class D |
| 0x17 | 0xF07 | 0x00000040 | Output enable (POR default) |
| 0x16 | 0xF01 | 0x00000000 | Index 0 (input from node 10, POR default) |
| 0x16 | 0xF05 | 0x00000400 | Power up Port A (based on test bit setting) |
| 0x16 | 0xF07 | 0x000000C0 | Output enable headphone out (POR default) |

Line input port turn on pop due to external coupling cap charging to Vref should be suppressed. SpkMute will be controlled which will help a lot if the charging time is kept to 33ms. We need to also consider what the Class-D high pass will do with the ADC ramp up. The gain settings are all set to 0db to allow a full-scale input signal to create a full-scale output signal. If there are losses or gains in the transfer between the ADC path and the DAC path these gains must be compensated resulting in an overall gain of 0dbFS.

## 23.6 Node 0: Root Node

The following table defines a root note that has one Audio Function Group (AFG). This device is compliant with and follows the guidelines given in version 1.0a of the HD Audio specification and the Windows Logo Program Device Requirements for Windows 8.1.

**Table 23-2: Node 0 Responses**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Vendor ID | 0xF00 | 0x00 | 0x14F1xxxx | | See Table |
| Revision ID | 0xF00 | 0x02 | 0x00100001 | | "A1" (Y0 ID was A0) |
| Subordinate node count | 0xF00 | 0x04 | 0x00010001 | | Audio AFG |
| Bit Clock capabilities | 0xF00 | 0x16 | 0x00000007 | | 6,12 and 24Mhz supported |
| Interface capabilities | 0xF00 | 0x17 | 0x00000001 | | Default<br>Reset is not required |
| Get bond option | 0xF1C-0xF1F | 0x00 | 0xaabbccdd | 0x00000000 (P) | |
| Get Current BCLK frequency | 0xF37 | 0x00 | 0x0000000a | 0x00000004 (P) | 24Mhz |

### 23.6.1 Package and device options

Please notice that all devices have different PNP-IDs. The table below assumes a weak pull up on the bond pads to conserve power for the most used version. The bond pads control the device IDs and restrict hardware behavior as appropriate to the feature set / package of the specific device. Only supported options for the HP focused parts are listed. Other possible options are ignored.

**Table 23-3: Device Options and IDs**

| Device Name | Bond Pad values BOP{5:1} | STD part# | PNP_ID | PKG |
|---|---|---|---|---|
| Redbud Isle | 10001 | CX8400 | 0x14F120D0 | 60QFN |
| Starke | 10011 | CX8450 | 0x14F12102 | 60QFN |

Bond options decoded as follows:

| Bond Option | Net Name | 0 | 1 | Notes |
|---|---|---|---|---|
| BOP5 | OPT5 | No Speaker Amp | AudioSmart Amp | Polarity so Deep Eddy has no downbond (pad unconnected) |
| BOP4 | OPT4 | I2S bridge support | No bridge | Polarity so Deep Eddy has no downbond (pad unconnected) |
| BOP3 | I2C_DEV_ID0 | 0110011R/nW | 0110111R/nW | I2C Address D2 follows bond pad |
| BOP2 | HBUS_MODE | HD Audio | I2S/I2C | |
| BOP1 | OPT1 | Base DEVID (50pin Deep Eddy) | Alternate DEVID (42pin Deep Eddy) | Alternate Device ID (and package indicator for Deep Eddy / Hornsby) |

The responses for node 0 are decoded as follows:

**Table 23-4: Vendor ID Response**

| Bit | Description |
|---|---|
| 31:16 | Vendor ID = 14F1h. Conexant's PCI SIG vendor ID |
| 15:0 | Device ID = xxxx. See table above |

**Table 23-5: Revision ID Response**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Read as zero |
| 23:20 | MajRev = 1. Major revision of the HD Audio specification supported (1.0a) |
| 19:16 | MinRev = 0. Minor revision of the HD Audio specification supported (1.0a) |
| 15:8 | Revision ID = 0. Revision number of this device (first revision) |
| 7:0 | Stepping ID = 1. Optional step number of this device (mask changes) |

**Table 23-6: Subordinate Node Count Response**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Read as zero |
| 23:16 | Starting node number = 01h |
| 15:8 | Reserved. Read as zero |
| 7:0 | Number of subordinate nodes = 01h |

**Table 23-7: Get Supported BCLK**

| Bit | Description |
|---|---|
| 31:3 | Future |
| 2 | 24Mhz = 1; BCLK of 24Mhz is supported (500 clocks per frame) |
| 1 | 12Mhz = 1; BCLK of 12Mhz is supported (250 clocks per frame) |
| 0 | 6Mhz = 1; BCLK of 6Mhz is supported (125 clocks per frame) |

**Table 23-8: Get Supported Interface options**

| Bit | Description |
|---|---|
| 31:1 | Future |
| 0 | Reset = 1; Reset is not required (HDA048 reset compliant) |

**Table 23-9: Get Default Config Response**

| Bit | Description |
|---|---|
| 31:24 | Future |
| 23:16 | Future |
| 15:8 | Future |
| 7:0 | Value stored in Config byte by bond options |

**Table 23-10: Get Current BCLK**

| Bit | Description |
|---|---|
| 31:3 | Future |
| 2 | BCLK is currently 24Mhz (500 clocks per frame) |
| 1 | BCLK is currently 12Mhz (250 clocks per frame) |
| 0 | BCLK is currently 6Mhz (125 clocks per frame) |

Conexant Systems Inc
Proprietary and Confidential
ALL INFORMATION IN THIS DOCUMENT IS PRELIMINARY AND SUBJECT TO CHANGE WITHOUT NOTICE

## 23.7 Node 1: AFG Widget

The following table describes an audio function group made up of 19 nodes. It also indicates the default supported sample rates and bit widths for the entire device. Nodes that have differing capabilities will also respond to verb queries. The configuration default register in the AFG is not a standard feature. This verb has been brought into the AFG node to facilitate BIOS to audio function driver communication. This communication channel can be used to indicate platform specific features. An example would be whether the built-in speakers require equalization or not. This allows the release of a generic driver that can, through BIOS communication, gain an understanding of any special features that are required by any given hardware platform.

Note: 5th GPIO is input mode only used to generate UM for center button on headset.

**Table 23-11: Node 1 Responses**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Subordinate node count | 0xF00 | 0x04 | 0x00100019 | | Starting node = 10<br>Node count = 25(CX8400/50) |
| Function group | 0xF00 | 0x05 | 0x00000101 | | Audio function group<br>Unsolicited capable |
| Audio F.G. capabilities | 0xF00 | 0x08 | 0x00010F0F | | Sample delay in and out is 16. PC beep gen. |
| PCM size and rate | 0xF00 | 0x0A | 0x000A0540 | | 16 & 24 bit<br>48, 96, & 192kHz |
| PCM format | 0xF00 | 0x0B | 0x00000001 | | PCM only |
| Supported power states | 0xF00 | 0x0F | 0xE000001F | | EPSS, Clock stop, D0, D1, D2, D3 and D4 |
| GPIO Count | 0xF00 | 0x11 | 0xC0000008 | | GPIO, Unsolicited Message & wake (CX8400/50) |
| Get power state | 0xF05 | 0x00 | 0x00000abc | 0x00000633 (P, W) | a = Settings Reset<br>b = Actual state<br>c = Requested state<br>Settings Reset cleared by this verb or any write to this node. |
| Set power state | 0x705 | 0x0a | 0x00000000 | | a = Requested state |
| Get unsolicited | 0xF08 | 0x00 | 0x000000aa | 0x00000000 (P, W) | aa = Unsolicited enable and tag |
| Set unsolicited | 0x708 | 0xaa | 0x00000000 | | aa = Unsolicited enable and tag |
| Get GPIO Data | 0xF15 | 0x00 | 0x000000aa | 0x00000000 (P, W) | aa = GPIO data |
| Set GPIO Data | 0x715 | 0xaa | 0x00000000 | | aa = GPIO data |
| Get GPIO Enable | 0xF16 | 0x00 | 0x000000aa | 0x00000000 (P, W) | aa = GPIO enable |
| Set GPIO Enable | 0x716 | 0xaa | 0x00000000 | | aa = GPIO enable |

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Get GPIO direction | 0xF17 | 0x00 | 0x000000aa | 0x00000000 (P, W) | aa = GPIO direction |
| Set GPIO direction | 0x717 | 0xaa | 0x00000000 | | aa = GPIO direction |
| Get GPIO Wake | 0xF18 | 0x00 | 0x000000aa | 0x00000000 (P, W) | aa = GPIO wake |
| Set GPIO Wake | 0x718 | 0xaa | 0x00000000 | | aa = GPIO wake |
| Get GPIO UM enable | 0xF19 | 0x00 | 0x000000aa | 0x00000000 (P, W) | aa = Unsolicited Message enable |
| Set GPIO UM enable | 0x719 | 0xaa | 0x00000000 | | aa = Unsolicited Message enable |
| Get GPIO Sticky mask | 0xF1A | 0x00 | 0x000000aa | 0x00000000 (P, W) | aa = Sticky mask |
| Set GPIO Sticky mask | 0x71A | 0xaa | 0x00000000 | | aa = Sticky mask |
| Get default Config | 0xF1C-0xF1F | 0x00 | 0xaabbccdd | 0x00000000 (P) | aa = Config4<br>bb = Config3<br>cc = Config2<br>dd = Config1 |
| Set default Config 1 | 0x71C | 0xaa | 0x00000000 | | aa = Config1 |
| Set default Config 2 | 0x71D | 0xaa | 0x00000000 | | aa = Config2 |
| Set default Config 3 | 0x71E | 0xaa | 0x00000000 | | aa = Config3 |
| Set default Config 4 | 0x71F | 0xaa | 0x00000000 | | aa = Config4 |
| Get Subsystem ID | 0xF20-0xF23 | 0x00 | 0xaaaabbcc | 0x14F10101 (P) | aaaa = Subsystem ID<br>bb = SKU ID<br>cc = Assembly ID |
| Set Subsystem ID 1 | 0x720 | 0xaa | 0x00000000 | | aa = Assembly ID |
| Set Subsystem ID 2 | 0x721 | 0xaa | 0x00000000 | | aa = SKU ID |
| Set Subsystem ID 3 | 0x722 | 0xaa | 0x00000000 | | aa = Subsystem ID low byte |
| Set Subsystem ID 4 | 0x723 | 0xaa | 0x00000000 | | aa = Subsystem ID high byte |
| Soft reset | 0x7FF | 0x00 | 0x00000000 | | |

The responses for node 1 are decoded as follows:

**Table 23-12: Subordinate Node Count Response**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Read as zero |
| 23:16 | Starting node number = 0x10 |

| | |
|---|---|
| 15:8 | Reserved. Read as zero |
| 7:0 | Number of subordinate nodes = 0x19 |

**Table 23-13: Function Group Response**

| Bit | Description |
|---|---|
| 31:9 | Reserved. Read as zero |
| 8 | UnSol Capable = 1. The GPIOs associated with the AFG can generate unsolicited messages |
| 7:0 | Function Group Type = 0x01. This is an Audio Function Group |

**Table 23-14: Audio Function Group Capabilities Response**

| Bit | Description |
|---|---|
| 31:17 | Reserved. Read as zero |
| 16 | Beep Gen = 1. The optional beep generator is present |
| 15:12 | Reserved. Read as zero |
| 11:8 | Input Delay = 0xF. Sample delay due to modulator |
| 7:4 | Reserved. Read as zero |
| 3:0 | Output Delay = 0xF. Sample delay due to modulator |

**Table 23-15: Sample Size, Rate Caps Response**

| Bit | Description |
|---|---|
| 31:21 | Reserved. Read as zero |
| 20 | B32 = 0. 32-bit audio format is not supported |
| 19 | B24 = 1. 24-bit audio format is supported |
| 18 | B20 = 0. 20-bit audio format is supported |
| 17 | B16 = 1. 16-bit audio format is supported |
| 16 | B8 = 0. 8-bit audio format is not supported |
| 15:12 | Reserved. Read as zero |
| 11 | R12 = 0. 384 kHz audio format is not supported |
| 10 | R11 = 1. 192 kHz audio format is supported |
| 9 | R10 = 0. 176.4 kHz audio format is not supported |
| 8 | R9 = 1. 96 kHz audio format is supported |
| 7 | R8 = 0. 88.2 kHz audio format is not supported |
| 6 | R7 = 1. 48 kHz audio format is supported |
| 5 | R6 = 0. 44.1 kHz audio format is not supported |
| 4 | R5 = 0. 32 kHz audio format is not supported |
| 3 | R4 = 0. 22.05 kHz audio format is not supported |
| 2 | R3 = 0. 16 kHz audio format is not supported |
| 1 | R2 = 0. 11.025 kHz audio format is not supported |
| 0 | R1 = 0. 8 kHz audio format is not supported |

**Table 23-16: Stream Formats Response**

| Bit | Description |
|---|---|
| 31:3 | Reserved. Read as zero |
| 2 | AC3 = 0. AC3 format is not supported |
| 1 | Float32 = 0. 32-bit float format is not supported |
| 0 | PCM = 1. PCM audio format is supported |

**Table 23-17: Supported Power States Response**

| Bit | Description |
|---|---|
| 31 | EPSS = 1. EPSS is supported |
| 30 | Clock stop = 1. Clock stop indication is supported |
| 29 | S3D3coldSup = 1. D3Cold is supported for S3 states |
| 28:5 | Reserved. Read as zero |
| 4 | D4Sup = 1. D4 is supported |
| 3 | D3Sup = 1. D3 is supported |
| 2 | D2Sup = 1. D2 is supported |
| 1 | D1Sup = 1. D1 is supported |
| 0 | D0Sup = 1. D0 is supported |

**Table 23-18: GPI/O Count Response**

| Bit | Description |
|---|---|
| 31 | GPIWake = 1. GPI or GPIO wake is supported |
| 30 | GPIUnsol = 1. GPI or GPIO unsolicited response is supported |
| 29:24 | Reserved. Read as zero |
| 23:16 | NumGPIs = 0. No GPIs are supported |
| 15:8 | NumGPOs = 0. No GPOs are supported |
| 7:0 | NumGPIOs = 8. 6 GPI/Os, 1 virtual for headset button, 1 virtual for Write Seq. |

**Table 23-19: Get Power State Response**

| Bit | Description |
|---|---|
| 31:11 | Reserved. Read as zero |
| 10 | PS_SettingsReset. Codec reset has occurred. |
| 9 | PS-ClockStopOk. Is set when the actual power state is D3-Hot and analog has indicated that clock stop is okay. |
| 8:7 | Reserved. Read as zero |
| 6:4 | PS-Act. Actual power state. |
| 3 | Reserved. Read as zero |
| 2:0 | PS-Set. Requested power state |

**Table 23-20: Set Power State Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:4 | Reserved. Set to zero |
| 3 | Reserved. Set to zero |
| 2:0 | PS-Set. 000 = D0, 001 = D1, 010 = D2, 011 = D3 hot, 100 = D3 cold |

**Table 23-21: Get/Set Unsolicited**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | Unsolicited enable, 0 = disable, 1 = enable |
| 6 | Not implemented. Read as zero |
| 5:0 | Unsolicited tag value |

**Table 23-22: Get GPIO Data Response**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7:0 | Input value or output value depending on direction |

**Table 23-23: Set GPIO Data Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:0 | Output value. If a bit is configured as sticky, writing a 1 will clear that bit. |

**Table 23-24: Get/Set GPIO Enable**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7:0 | GPIO enable, 0 = Disable GPIO, 1 = Enable GPIO |

**Table 23-25: Get/Set GPIO Direction Response**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7:0 | GPIO direction, 0 = input, 1 = Output |

**Table 23-26: Get/Set GPIO Unsolicited Message Enable Response**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7:0 | GPIO Unsolicited Message enable, 0 = disable UM, 1 = enable UM |

**Table 23-27: Get/Set GPIO Sticky Enable Response**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7:0 | GPIO Sticky enable, 0 = level sensitive, 1 = edge sensitive |

**Table 23-28: Get Default Config Response**

| Bit | Description |
|---|---|
| 31:24 | Value stored in Config byte 4 |
| 23:16 | Value stored in Config byte 3 |
| 15:8 | Value stored in Config byte 2 |
| 7:0 | Value stored in Config byte 1 |

**Table 23-29: Set Default Config Data Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:0 | Value to store in Config byte 1, 2, 3, or 4 |

**Table 23-30: Get Subsystem ID Response**

| Bit | Description |
|---|---|
| 31:16 | Value stored in Subsystem ID byte 3 and 4 |
| 15:8 | Value stored in SKU ID byte 2 |
| 7:0 | Value stored in Assembly ID byte 1 |

**Table 23-31: Set Subsystem ID Data Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:0 | Value to store in Subsystem ID byte 3 or 4, SKU ID byte 2, or Assembly ID byte 1 |

The high byte, high word of configuration data has special control bits which are described below. These bits allow the BIOS to secure (mute) any mix of input ports. Once secured and locked only power on reset can re-enable the recording function.

**Table 23-32: Secure microphone and headphone limiting**

| Conf Read | Byte 4 Write | Description |
|---|---|---|
| 24 | 0 | 0 = Normal, 1 = EU limited headphone levels |
| 25 | 1 | 0 = Headphone unlocked, 1 = Headphone locked (power change required) |
| 26 | 2 | 0 = Mic unlocked (power change not required), 1 = Mic locked (power change required) |
| 27 | 3 | 0 = Normal, 1 = Port B analog microphone muted |
| 28 | 4 | 0 = Normal, 1 = Port C digital microphone muted |
| 29 | 5 | 0 = Normal, 1 = Port D analog headset microphone muted |
| 30 | 6 | 0 = Normal, 1 = Port F/K I2S inputs muted |
| 31 | 7 | 0 = Normal, 1 = Port H digital microphone muted |

## 23.8 Nodes 10 and 11: DAC 1 and 2 Widgets

The following table describes a stereo DAC that supports 16 and 24 bit widths and 48 kilohertz sample rates.

**Table 23-33: Node 10 and 11 responses**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Get converter format | 0xA | 0x0000 | 0x0000aaaa | 0x00000031 (P,W) | aaaa = converter format |
| Set converter format | 0x2 | 0xaaaa | 0x00000000 | | aaaa = converter format |
| Get amp gain | 0xB80<br>0xBA0 | 0x00<br>0x00 | 0x000000aa | 0x0000004A (P,W) | aa = Right gain<br>aa = Left Gain |
| Set amp gain | 0x390<br>0x3A0<br>0x3B0 | 0xaa | 0x00000000 | | aa = Right gain<br>aa = Left gain<br>aa = Right & Left gain |
| Audio widget DAC | 0xF00 | 0x09 | 0x00000C1D | | DAC – analog |
| PCM size and rate | 0xF00 | 0x0A | 0x000A0540 | | 16 & 24 bit<br>48, 96, and 192KHz |
| PCM Format | 0xF00 | 0x0B | 0x00000001 | | PCM only |
| Supported Power states | 0xF00 | 0x0F | 0x8000000F | | EPSS, D0, D1, D2, and D3 |
| Get output amp capabilities | 0xF00 | 0x12 | 0x80034A4A | | Mute, 1db step,<br>Step 74 is 0db<br>74 of 80 steps exposed |
| Get power state | 0xF05 | 0x00 | 0x00000abc | 0x00000433 (P,W) | a = Settings Reset<br>b = Actual state<br>c = Requested state<br>Settings Reset cleared by this verb or any write to this node. |
| Set power state | 0x705 | 0x0a | 0x00000000 | | a = Requested state |
| Get converter stream/chnl | 0xF06 | 0x00 | 0x000000ab | 0x00000000 (P,R,V,W,D) | a = Stream<br>b = Channel position |
| Set converter stream/chnl | 0x706 | 0xab | 0x00000000 | | a = Stream<br>b = Channel position |
| Get EAPD | 0xF0C | 0x00 | 0x0000000a | 0x00000000 (P,W) | a = left/right swap |
| Set EAPD | 0x70C | 0x0a | 0x00000000 | | a = left/right swap |

The responses are decoded as follows:

**Table 23-34: Get/Set Converter Format**

| Bit | Description |
|---|---|

| | |
|---|---|
| 31:16 | Reserved. Read as zero |
| 15 | Non-PCM = 0. Only PCM is supported |
| 14 | Sample Base Rate = 0 or 1, 48 or 44.1 kHz |
| 13:11 | Sample Base Rate Multiplier = 000, 001 or 011.<br>000b = times 1 (44.1 or 48 kHz)<br>001b = times 2 (96kHz)<br>011b = times 4 (192kHz) |
| 10:8 | Sample Base Rate Divisor = 000. Divide by 1<br>000b = divide 1 (48 or 44.1 kHz) |
| 7 | Reserved. Read as zero |
| 6:4 | Bits per Sample = 001 or 011.<br>001 = 16 bits<br>011 = 24 bits |
| 3:0 | Number of Channels = 0 or 1.<br>0 = 1 channel (mono)<br>1 = 2 channels (stereo) |

**Table 23-35: Get/Set Amplifier Gain**

| Bit | Description |
|---|---|
| 7 | Amplifier Mute = 0 or 1 |
| 6:0 | Amplifier Gain index. |

**Table 23-36: Audio Widget Capabilities Response**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Read as zero |
| 23:20 | Type = 0h. This is a DAC widget |
| 19:16 | Delay = 0h. Delay is described in the function group |
| 15:12 | Reserved. Read as zero |
| 11 | L-R Swap = 1. Left/right swap is supported. |
| 10 | Power Cntrl = 1. Power state control is available |
| 9 | Digital = 0. This widget is not digital |
| 8 | Conn List = 0. No connection list is available |
| 7 | Unsol Capable = 0. Jack sensing is not available |
| 6 | Proc Widget = 0. No processing capabilities are present |
| 5 | Stripe = 0. Striping is not supported |
| 4 | Format Override = 1. Format information is not available |
| 3 | Amp Parameter Override = 1. Amp parameters are available |
| 2 | Out Amp Present = 1. Output amp is present |
| 1 | In Amp Present = 0. No input amp is present |
| 0 | Stereo = 1. This widget supports stereo |

**Table 23-37: Sample Size, Rate Caps Response**

| Bit | Description |
|---|---|
| 31:21 | Reserved. Read as zero |
| 20 | B32 = 0. 32-bit audio format is not supported |
| 19 | B24 = 1. 24-bit audio format is supported |
| 18 | B20 = 0. 20-bit audio format is not supported |

| | |
|---|---|
| 17 | B16 = 1. 16-bit audio format is supported |
| 16 | B8 = 0. 8-bit audio format is not supported |
| 15:12 | Reserved. Read as zero |
| 11 | R12 = 0. 384 kHz audio format is not supported |
| 10 | R11 = 1. 192 kHz audio format is not supported |
| 9 | R10 = 0. 176.4 kHz audio format is not supported |
| 8 | R9 = 1. 96 kHz audio format is not supported |
| 7 | R8 = 0. 88.2 kHz audio format is not supported |
| 6 | R7 = 1. 48 kHz audio format is supported |
| 5 | R6 = 0. 44.1 kHz audio format is not supported |
| 4 | R5 = 0. 32 kHz audio format is not supported |
| 3 | R4 = 0. 22.05 kHz audio format is not supported |
| 2 | R3 = 0. 16 kHz audio format is not supported |
| 1 | R2 = 0. 11.025 kHz audio format is not supported |
| 0 | R1 = 0. 8.0 kHz audio format is not supported |

**Table 23-38: Stream Formats Response**

| Bit | Description |
|---|---|
| 31:3 | Reserved. Read as zero |
| 2 | AC3 = 0. AC3 format is not supported |
| 1 | Float32 = 0. 32-bit float format is not supported |
| 0 | PCM = 1. PCM audio format is supported |

**Table 23-39: Supported Power States Response**

| Bit | Description |
|---|---|
| 31 | EPSS = 1. EPSS is supported |
| 30:5 | Reserved. Read as zero |
| 4 | D4Sup = 0. D4 is not supported |
| 3 | D3Sup = 1. D3 is supported |
| 2 | D2Sup = 1. D2 is supported |
| 1 | D1Sup = 1. D1 is supported |
| 0 | D0Sup = 1. D0 is supported |

**Table 23-40: Output Amp Capabilities Response**

| Bit | Description |
|---|---|
| 31 | Mute Capable = 1. Mute is supported |
| 30:23 | Reserved. Read as zero |
| 22:16 | Step Size = 03h. Step size is 1dB |
| 15 | Reserved. Read as zero |
| 14:8 | NumSteps = 4Ah. There are 80 steps but only 74 are exposed |
| 7 | Reserved. Read as zero |
| 6:0 | Offset = 4Ah. Step 74 is 0dB |

**Table 23-41: Get Power State Response**

| Bit | Description |
|---|---|
| 31:11 | Reserved. Read as zero |
| 10 | PS_SettingsReset. Codec reset has occurred. |

| | |
|---|---|
| 9:7 | Reserved. Read as zero |
| 6:4 | PS-Act. Actual power state. |
| 3 | Reserved. Read as zero |
| 2:0 | PS-Set. Requested power state |

**Table 23-42: Set Power State Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:4 | Reserved. Set to zero |
| 3 | Reserved. Set to zero |
| 2:0 | PS-Set. 000 = D0, 001 = D1, 010 = D2, 011 = D3 hot, 100 = D3 cold |

**Table 23-43: Get/Set Channel and Stream ID**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7:4 | Stream ID, 0 = disable stream 1 –> F = enable stream |
| 3:0 | Channel ID |

**Table 23-44: Get EAPD/BTL Response**

| Bit | Description |
|---|---|
| 31:3 | Reserved. Read as zero |
| 2 | Left/Right swap = 0 or 1. |
| 1 | EAPD = 0. |
| 0 | BTL = 0. |

## 23.9 Node 12: PC Beep Generator Widget

The following table describes a beep generator. PC beep is mixed in with all enabled output ports including I2S while in D0. When the PC beep input pin is connected but inactive, not toggling, none of the system noise will be injected on the output pins. If the PC beep input is left unconnected there will not be any impact on the performance of the output ports.

Note that because this is a mono widget. Only the left channel volume request is valid. Any request, read or write, having to do with only the right channel will be ignored and return 0x00000000. If both left and right are present in the request only the value from the left side is used.

The beep gain level range (-4 to -32db) is for the speaker port. The default of setting of -20db for beep gain plus the headphone default of -24 results in -20db on the speaker, I2S and -44db on the headphone.

**Table 23-45: PC Beep generator responses**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Get amp gain | 0xBA0 | 0x00 | 0x0000000a | 0x00000003 (P,W) | aa = Left Gain |
| Set amp gain | 0x3A0<br>0x3B0 | 0xaa | 0x00000000 | | aa = Left gain<br>aa = Left gain |
| Audio widget PC Beep | 0xF00 | 0x09 | 0x0070000C | | PC beep generator with output amp |
| Get output amp capabilities | 0xF00 | 0x12 | 0x000F0707 | | 4db step, 8 steps, Step 8 is -4db |
| Get beep generation control | 0xF0A | 0x00 | 0x000000aa | 0x00000000 (P,W) | aa = divider |
| Set beep generation control | 0x70A | 0xaa | 0x00000000 | | aa = divider |

The responses are decoded as follows.

**Table 23-46: Get/Set Amplifier Gain**

| Bit | Description |
|---|---|
| 7 | Amplifier Mute = 0. (mute not supported) |
| 6:3 | Read as 0 |
| 2:0 | Amplifier Gain index. |

**Table 23-47: Audio Widget Capabilities Response**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Read as zero |
| 23:20 | Type = 7h. This is a BEEP GENERATOR widget |
| 19:16 | Delay = 0h. Delay does not apply to a BEEP GENERATOR |
| 15:12 | Reserved. Read as zero |
| 11 | L-R Swap = 0. Does not apply to a BEEP GENERATOR |
| 10 | Power Cntrl = 0. Does not apply to a BEEP GENERATOR |

| | |
|---|---|
| 9 | Digital = 0. Does not apply to a BEEP GENERATOR |
| 8 | Conn List = 0. Does not apply to a BEEP GENERATOR |
| 7 | Unsol Capable = 0. Does not apply to a BEEP GENERATOR |
| 6 | Proc Widget = 0. Does not apply to a BEEP GENERATOR |
| 5 | Stripe = 0. Does not apply to a BEEP GENERATOR |
| 4 | Format Override = 0. Does not apply to a BEEP GENERATOR |
| 3 | Amp Parameter Override = 1. Amp parameters are available |
| 2 | Out Amp Present = 1. An output amp is present |
| 1 | In Amp Present = 0. No input amp is present |
| 0 | Stereo = 0. Does not apply to a BEEP GENERATOR |

**Table 23-48: Output Amp Capabilities Response**

| Bit | Description |
|---|---|
| 31 | Mute Capable = 0. Mute is not supported |
| 30:23 | Reserved. Read as zero |
| 22:16 | Step Size = Fh. Step size is 4dB |
| 15 | Reserved. Read as zero |
| 14:8 | NumSteps = 7h. 8 steps |
| 7 | Reserved. Read as zero |
| 6:0 | Offset = 7h. Step 8 is -4dB |

**Table 23-49: Get/Set Beep Generator Control**

| Bit | Description |
|---|---|
| 7:0 | Divisor.<br>Beep frequency = 48kHz / (4 * Divisor)<br>00h = disable beep generator, 01h -> FFh enables. |

## 23.10 Node 13: ADC 1 Widget

The following table describes a stereo ADC that supports 16 and 24 bit widths and 48, 96, or 192 kilohertz sample rates. The ADC has a gain stage and a stereo one of 7 input selector. Any source which is mono will copy the contents of the left channel to the right channel at any convenient location prior to sending the data on the HD Audio bus.

**Table 23-50: Node 13 Responses**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Get converter format | 0xA | 0x0000 | 0x0000aaaa | 0x00000031 (P,W) | aaaa = converter format |
| Set converter format | 0x2 | 0xaaaa | 0x00000000 | | aaaa = converter format |
| Get index 0 amp gain | 0xB00<br>0xB20 | 0x00 | 0x000000aa<br>0x000000aa | 0x0000004A (P,W) | aa = Right gain<br>aa = Left gain |
| Get index 1 amp gain | 0xB00<br>0xB20 | 0x01 | 0x000000aa<br>0x000000aa | 0x0000004A (P,W) | aa = Right gain<br>aa = Left gain |
| Get index 2 amp gain | 0xB00<br>0xB20 | 0x02 | 0x000000aa<br>0x000000aa | 0x0000004A (P,W) | aa = Right gain<br>aa = Left gain |
| Get index 3 amp gain | 0xB00<br>0xB20 | 0x03 | 0x000000aa<br>0x000000aa | 0x0000004A (P,W) | aa = Right gain<br>aa = Left gain |
| Get index 4 amp gain | 0xB00<br>0xB20 | 0x04 | 0x000000aa<br>0x000000aa | 0x0000004A (P,W) | aa = Right gain<br>aa = Left gain |
| Set index 0 amp gain | 0x350<br>0x360<br>0x370 | 0xaa | 0x00000000 | | aa = Right gain<br>aa = Left gain<br>aa = Left & Right gain |
| Set index 1 amp gain | 0x351<br>0x361<br>0x371 | 0xaa | 0x00000000 | | aa = Right gain<br>aa = Left gain<br>aa = Left & Right gain |
| Set index 2 amp gain | 0x352<br>0x362<br>0x372 | 0xaa | 0x00000000 | | aa = Right gain<br>aa = Left gain<br>aa = Left & Right gain |
| Set index 3 amp gain | 0x353<br>0x363<br>0x373 | 0xaa | 0x00000000 | | aa = Right gain<br>aa = Left gain<br>aa = Left & Right gain |
| Set index 4 amp gain | 0x354<br>0x364<br>0x374 | 0xaa | 0x00000000 | | aa = Right gain<br>aa = Left gain<br>aa = Left & Right gain |
| Audio widget ADC | 0xF00 | 0x09 | 0x00100D1B | | ADC - analog |
| PCM size and rate | 0xF00 | 0x0A | 0x000A0540 | | 16&24 bit / 48, 96 & 192 kHz |
| PCM format | 0xF00 | 0x0B | 0x00000001 | | PCM only |
| Input amp capabilities | 0xF00 | 0x0D | 0x8003504A | | Mute, 1db step, 80 steps, step 74 is 0db |
| Connection length | 0xF00 | 0x0E | 0x00000005 | | |
| Supported | 0xF00 | 0x0F | 0x8000000F | | EPSS, D0, D1, D2 and D3 |

| | | | | | |
|---|---|---|---|---|---|
| Power states | | | | | |
| Get connection select | 0xF01 | 0x00 | 0x0000000a | 0x00000000 (P,W) | a = Connection Index |
| Set connection select | 0x701 | 0x0a | 0x00000000 | | a = Connection Index |
| Get connection list | 0xF02 | 0x00 | 0x151A1918 | | 18 = Port B input<br>19 = Headset microphone<br>1A = Digital mic1 input<br>15 = DAC Mix |
| Get connection list | 0xF02 | 0x04 | 0x0000001F | | 1F = Port H input |
| Get power state | 0xF05 | 0x00 | 0x00000abc | 0x00000433 (P,W) | a = Settings Reset<br>b = Actual state<br>c = Requested state<br>Settings Reset cleared by this verb or any write to this node. |
| Set power state | 0x705 | 0x0a | 0x00000000 | | a = Requested state |
| Get converter stream/chnl | 0xF06 | 0x00 | 0x000000ab | 0x00000000 (P,R,V,W,D) | a = Stream<br>b = Channel position |
| Set converter stream/chnl | 0x706 | 0xab | 0x00000000 | | a = Stream<br>b = Channel position |
| Get EAPD | 0xF0C | 0x00 | 0x0000000a | 0x00000000 (P,W) | a = left/right swap |
| Set EAPD | 0x70C | 0x0a | 0x00000000 | | a = left/right swap |

The responses are decoded as follows:

**Table 23-51: Get/Set Converter Format**

| Bit | Description |
|---|---|
| 31:16 | Reserved. Read as zero |
| 15 | Non-PCM = 0. Only PCM is supported |
| 14 | Sample Base Rate = 0 or 1, 48 or 44.1 kHz |
| 13:11 | Sample Base Rate Multiplier = 000, 001, or 011.<br>000b = times 1 (44.1 or 48 kHz)<br>001b = times 2 (96kHz)<br>011b = times 4 (192kHz) |
| 10:8 | Sample Base Rate Divisor = 000. Divide by 1<br>000b = divide 1 (48 or 44.1 kHz) |
| 7 | Reserved. Read as zero |
| 6:4 | Bits per Sample = 001 or 011.<br>001 = 16 bits<br>011 = 24 bits |
| 3:0 | Number of Channels = 0 or 1.<br>0 = 1 channel (mono)<br>1 = 2 channels (stereo) |

**Table 23-52: Get/Set Amplifier Gain**

| Bit | Description |
|---|---|
| 7 | Amplifier Mute = 0 or 1 |
| 6:0 | Amplifier Gain index. 0 – 0x50 is valid, 0x51-0x7F = 0x00 |

**Table 23-53: Audio Widget Capabilities Response**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Read as zero |
| 23:20 | Type = 1h. This is a ADC widget |
| 19:16 | Delay = 0h. Delay is described in the function group |
| 15:12 | Reserved. Read as zero |
| 11 | L-R Swap = 1. Left/right swap is supported. |
| 10 | Power Cntrl = 1. Power state control is available |
| 9 | Digital = 0. This widget is not digital |
| 8 | Conn List = 1. A connection list is available |
| 7 | Unsol Capable = 0. Jack sensing is not available |
| 6 | Proc Widget = 0. No processing capabilities are present |
| 5 | Stripe = 0. Striping is not supported |
| 4 | Format Override = 1. Format information is available |
| 3 | Amp Parameter Override = 1. Amp parameters are available |
| 2 | Out Amp Present = 0. No output amp is present |
| 1 | In Amp Present = 1. Input amp is present |
| 0 | Stereo = 1. This widget supports stereo |

**Table 23-54: Sample Size, Rate Caps Response**

| Bit | Description |
|---|---|
| 31:21 | Reserved. Read as zero |
| 20 | B32 = 0. 32-bit audio format is not supported |
| 19 | B24 = 1. 24-bit audio format is supported |
| 18 | B20 = 0. 20-bit audio format is not supported |
| 17 | B16 = 1. 16-bit audio format is supported |
| 16 | B8 = 0. 8-bit audio format is not supported |
| 15:12 | Reserved. Read as zero |
| 11 | R12 = 0. 384 kHz audio format is not supported |
| 10 | R11 = 1. 192 kHz audio format is supported |
| 9 | R10 = 0. 176.4 kHz audio format is not supported |
| 8 | R9 = 1. 96 kHz audio format is not supported |
| 7 | R8 = 0. 88.2 kHz audio format is not supported |
| 6 | R7 = 1. 48 kHz audio format is supported |
| 5 | R6 = 0. 44.1 kHz audio format is not supported |
| 4 | R5 = 0. 32 kHz audio format is not supported |
| 3 | R4 = 0. 22.05 kHz audio format is not supported |
| 2 | R3 = 0. 16 kHz audio format is not supported |
| 1 | R2 = 0. 11.025 kHz audio format is not supported |
| 0 | R1 = 0. 8.0 kHz audio format is not supported |

**Table 23-55: Stream Formats Response**

| Bit | Description |
|---|---|
| 31:3 | Reserved. Read as zero |
| 2 | AC3 = 0. AC3 format is not supported |
| 1 | Float32 = 0. 32-bit float format is not supported |
| 0 | PCM = 1. PCM audio format is supported |

**Table 23-56: Input Amp Capabilities Response**

| Bit | Description |
|---|---|
| 31 | Mute Capable = 1. Mute is supported |
| 30:23 | Reserved. Read as zero |
| 22:16 | Step Size = 03h. Step size is 1dB |
| 1 | Reserved. Read as zero |
| 14:8 | NumSteps = 50h. 80 steps |
| 7 | Reserved. Read as zero |
| 6:0 | Offset = 4Ah. Step 74 is 0dB |

**Table 23-57: Connection List Length Response**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | Long Form = 0. List is in short form |
| 6:0 | Connection List Length = 0x5 |

**Table 23-58: Supported Power States Response**

| Bit | Description |
|---|---|
| 31 | EPSS = 1. EPSS is supported |
| 30:5 | Reserved. Read as zero |
| 4 | D4Sup = 0. D4 is not supported |
| 3 | D3Sup = 1. D3 is supported |
| 2 | D2Sup = 1. D2 is supported |
| 1 | D1Sup = 1. D1 is supported |
| 0 | D0Sup = 1. D0 is supported |

**Table 23-59: Get/Set Connection Select Response**

| Bit | Description |
|---|---|
| 7:0 | 0b000 = 18, 0b001 = 19, 0b010 = 1A, 0b011 = 15, 0b100 = 1F |

**Table 23-60: Get Connection List Response index 0**

| Bit | Description |
|---|---|
| 31:0 | 0x151A1918 |

**Table 23-61: Get Connection List Response index 1**

| Bit | Description |
|---|---|
| 31:0 | 0x0000001F |

**Table 23-62: Get Power State Format**

| Bit | Description |
|---|---|

| | |
|---|---|
| 31:11 | Reserved. Read as zero |
| 10 | PS_SettingsReset. Codec reset has occurred. |
| 9:7 | Reserved. Read as zero |
| 6:4 | PS-Act. Actual power state. |
| 3 | Reserved. Read as zero |
| 2:0 | PS-Set. Requested power state |

**Table 23-63: Set Power State Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:4 | Reserved. Set to zero |
| 3 | Reserved. Set to zero |
| 2:0 | PS-Set. 000 = D0, 001 = D1, 010 = D2, 011 = D3 hot, 100 = D3 cold |

**Table 23-64: Get/Set Channel and Stream ID**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7:4 | Stream ID, 0 = disable stream 1 –> F = enable stream |
| 3:0 | Channel ID |

**Table 23-65: Get EAPD/BTL Response**

| Bit | Description |
|---|---|
| 31:3 | Reserved. Read as zero |
| 2 | Left/Right swap = 0 or 1. |
| 1 | EAPD = 0. |
| 0 | BTL = 0. |

## 23.11 Node 14: ADC 2 Widget

The following table describes a stereo ADC that supports 16 and 24 bit widths and 48, 96, or 192 kilohertz sample rates. The ADC has a gain stage and a stereo one of 3 input selector. Any source which is mono will copy the contents of the left channel to the right channel at any convenient location prior to sending the data on the HD Audio bus.

**Table 23-66: Node 14 Responses**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Get converter format | 0xA | 0x0000 | 0x0000aaaa | 0x00000031 (P,W) | aaaa = converter format |
| Set converter format | 0x2 | 0xaaaa | 0x00000000 | | aaaa = converter format |
| Get index 0 amp gain | 0xB00<br>0xB20 | 0x00 | 0x000000aa<br>0x000000aa | 0x0000004A (P,W) | aa = Right gain<br>aa = Left gain |
| Get index 1 amp gain | 0xB00<br>0xB20 | 0x01 | 0x000000aa<br>0x000000aa | 0x0000004A (P,W) | aa = Right gain<br>aa = Left gain |
| Get index 2 amp gain | 0xB00<br>0xB20 | 0x02 | 0x000000aa<br>0x000000aa | 0x0000004A (P,W) | aa = Right gain<br>aa = Left gain |
| Set index 0 amp gain | 0x350<br>0x360<br>0x370 | 0xaa | 0x00000000 | | aa = Right gain<br>aa = Left gain<br>aa = Left & Right gain |
| Set index 1 amp gain | 0x351<br>0x361<br>0x371 | 0xaa | 0x00000000 | | aa = Right gain<br>aa = Left gain<br>aa = Left & Right gain |
| Set index 2 amp gain | 0x352<br>0x362<br>0x372 | 0xaa | 0x00000000 | | aa = Right gain<br>aa = Left gain<br>aa = Left & Right gain |
| Audio widget ADC | 0xF00 | 0x09 | 0x00100D1B | | ADC - analog |
| PCM size and rate | 0xF00 | 0x0A | 0x000A0540 | | 16&24 bit / 48, 96, & 192 kHz |
| PCM format | 0xF00 | 0x0B | 0x00000001 | | PCM only |
| Input amp capabilities | 0xF00 | 0x0D | 0x8003504A | | Mute, 1db step, 80 steps, Step 74 is 0db |
| Connection length | 0xF00 | 0x0E | 0x00000003<br>0x00000002 | | Port C digital mode<br>Port C analog mode |
| Supported Power states | 0xF00 | 0x0F | 0x8000000F | | EPSS, D0, D1, D2 and D3 |
| Get connection select | 0xF01 | 0x00 | 0x0000000a | 0x00000000 (P,W) | a = Connection Index |
| Set connection select | 0x701 | 0x0a | 0x00000000 | | a = Connection Index |
| Get connection list | 0xF02 | 0x00 | 0x00151F1A<br>0x0000151F | | Port C digital mode<br>Port C analog mode |
| Get power state | 0xF05 | 0x00 | 0x00000abc | 0x00000433 (P,W) | a = Settings Reset<br>b = Actual state |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | c = Requested state<br>Settings Reset cleared by this verb or any write to this node. |
| Set power state | 0x705 | 0x0a | 0x00000000 | | a = Requested state |
| Get converter stream/chnl | 0xF06 | 0x00 | 0x000000ab | 0x00000000<br>(P,R,V,W,D) | a = Stream<br>b = Channel position |
| Set converter stream/chnl | 0x706 | 0xab | 0x00000000 | | a = Stream<br>b = Channel position |
| Get EAPD | 0xF0C | 0x00 | 0x0000000a | 0x00000000<br>(P,W) | a = left/right swap |
| Set EAPD | 0x70C | 0x0a | 0x00000000 | | a = left/right swap |

The responses are decoded as follows:

**Table 23-67: Get/Set Converter Format**

| Bit | Description |
|---|---|
| 31:16 | Reserved. Read as zero |
| 15 | Non-PCM = 0. Only PCM is supported |
| 14 | Sample Base Rate = 0 or 1, 48 or 44.1 kHz |
| 13:11 | Sample Base Rate Multiplier = 000, 001 or 011.<br>000b = times 1 (44.1 or 48 kHz)<br>001b = times 2 (96kHz)<br>011b = times 4 (192kHz) |
| 10:8 | Sample Base Rate Divisor = 000. Divide by 1<br>000b = divide 1 (48 or 44.1 kHz) |
| 7 | Reserved. Read as zero |
| 6:4 | Bits per Sample = 001 or 011.<br>001 = 16 bits<br>011 = 24 bits |
| 3:0 | Number of Channels = 0 or 1.<br>0 = 1 channel (mono)<br>1 = 2 channels (stereo) |

**Table 23-68: Get/Set Amplifier Gain**

| Bit | Description |
|---|---|
| 7 | Amplifier Mute = 0 or 1 |
| 6:0 | Amplifier Gain index. 0 – 0x50 is valid, 0x51-0x7F = 0x00 |

**Table 23-69: Audio Widget Capabilities Response**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Read as zero |
| 23:20 | Type = 1h. This is a ADC widget |
| 19:16 | Delay = 0h. Delay is described in the function group |
| 15:12 | Reserved. Read as zero |
| 11 | L-R Swap = 1. Left/right swap is supported. |

| | |
|---|---|
| 10 | Power Cntrl = 1. Power state control is available |
| 9 | Digital = 0. This widget is not digital |
| 8 | Conn List = 1. A connection list is available |
| 7 | Unsol Capable = 0. Jack sensing is not available |
| 6 | Proc Widget = 0. No processing capabilities are present |
| 5 | Stripe = 0. Striping is not supported |
| 4 | Format Override = 1. Format information is available |
| 3 | Amp Parameter Override = 1. Amp parameters are available |
| 2 | Out Amp Present = 0. No output amp is present |
| 1 | In Amp Present = 1. Input amp is present |
| 0 | Stereo = 1. This widget supports stereo |

**Table 23-70: Sample Size, Rate Caps Response**

| Bit | Description |
|---|---|
| 31:21 | Reserved. Read as zero |
| 20 | B32 = 0. 32-bit audio format is not supported |
| 19 | B24 = 1. 24-bit audio format is supported |
| 18 | B20 = 0. 20-bit audio format is not supported |
| 17 | B16 = 1. 16-bit audio format is supported |
| 16 | B8 = 0. 8-bit audio format is not supported |
| 15:12 | Reserved. Read as zero |
| 11 | R12 = 0. 384 kHz audio format is not supported |
| 10 | R11 = 1. 192 kHz audio format is supported |
| 9 | R10 = 0. 176.4 kHz audio format is not supported |
| 8 | R9 = 1. 96 kHz audio format is not supported |
| 7 | R8 = 0. 88.2 kHz audio format is not supported |
| 6 | R7 = 1. 48 kHz audio format is supported |
| 5 | R6 = 0. 44.1 kHz audio format is not supported |
| 4 | R5 = 0. 32 kHz audio format is not supported |
| 3 | R4 = 0. 22.05 kHz audio format is not supported |
| 2 | R3 = 0. 16 kHz audio format is not supported |
| 1 | R2 = 0. 11.025 kHz audio format is not supported |
| 0 | R1 = 0. 8.0 kHz audio format is not supported |

**Table 23-71: Stream Formats Response**

| Bit | Description |
|---|---|
| 31:3 | Reserved. Read as zero |
| 2 | AC3 = 0. AC3 format is not supported |
| 1 | Float32 = 0. 32-bit float format is not supported |
| 0 | PCM = 1. PCM audio format is supported |

**Table 23-72: Input Amp Capabilities Response**

| Bit | Description |
|---|---|
| 31 | Mute Capable = 1. Mute is supported |
| 30:23 | Reserved. Read as zero |
| 22:16 | Step Size = 03h. Step size is 1dB |

| | |
|---|---|
| 15 | Reserved. Read as zero |
| 14:8 | NumSteps = 50h. 80 steps |
| 7 | Reserved. Read as zero |
| 6:0 | Offset = 4Ah. Step 74 is 0dB |

**Table 23-73: Connection List Length Response**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | Long Form = 0. List is in short form |
| 6:0 | Connection List Length = 0x3 (digital mic mode), 0x2 (analog mic mode). |

**Table 23-74: Supported Power States Response**

| Bit | Description |
|---|---|
| 31 | EPSS = 1. EPSS is supported |
| 30:5 | Reserved. Read as zero |
| 4 | D4Sup = 0. D4 is not supported |
| 3 | D3Sup = 1. D3 is supported |
| 2 | D2Sup = 1. D2 is supported |
| 1 | D1Sup = 1. D1 is supported |
| 0 | D0Sup = 1. D0 is supported |

**Table 23-75: Get/Set Connection Select Response**

| Bit | Description |
|---|---|
| 7:0 | 0b000 = 1A, 0b001 = 1F, 0b010 = 15 (digital mic mode)<br>0b000 = 1F, 0b001 = 15 (analog mic mode) |

**Table 23-76: Get Connection List Response index 0**

| Bit | Description |
|---|---|
| 31:0 | 0x00151F1A Port C in digital mode<br>0x0000151F Port C in analog mode |

**Table 23-77: Get Power State Format**

| Bit | Description |
|---|---|
| 31:11 | Reserved. Read as zero |
| 10 | PS_SettingsReset. Codec reset has occurred. |
| 9:7 | Reserved. Read as zero |
| 6:4 | PS-Act. Actual power state. |
| 3 | Reserved. Read as zero |
| 2:0 | PS-Set. Requested power state |

**Table 23-78: Set Power State Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:4 | Reserved. Set to zero |
| 3 | Reserved. Set to zero |
| 2:0 | PS-Set. 000 = D0, 001 = D1, 010 = D2, 011 = D3 hot, 100 = D3 cold |

**Table 23-79: Get/Set Channel and Stream ID**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7:4 | Stream ID, 0 = disable stream 1 –> F = enable stream |
| 3:0 | Channel ID |

**Table 23-80: Get EAPD/BTL Response**

| Bit | Description |
|---|---|
| 31:3 | Reserved. Read as zero |
| 2 | Left/Right swap = 0 or 1. |
| 1 | EAPD = 0. |
| 0 | BTL = 0. |

## 23.12 Node 15: Mixer Widget

**Table 23-81: Node 15 responses**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Get index 0 amp gain | 0xB00<br>0xB20 | 0x00 | 0x000000aa<br>0x000000aa | 0x0000004A<br>(P,W) | a = Right gain<br>a = Left gain |
| Get index 1 amp gain | 0xB00<br>0xB20 | 0x01 | 0x000000aa<br>0x000000aa | 0x0000004A<br>(P,W) | a = Right gain<br>a = Left gain |
| Set index 0 amp gain | 0x350<br>0x360<br>0x370 | 0xaa | 0x00000000 | | a = Right gain<br>a = Left gain<br>a = Left & Right gain |
| Set index 1 amp gain | 0x351<br>0x361<br>0x371 | 0xaa | 0x00000000 | | a = Right gain<br>a = Left gain<br>a = Left & Right gain |
| Audio widget Mixer | 0xF00 | 0x09 | 0x00F0050B | | Vendor defined with input amplifier |
| Input amp capabilities | 0xF00 | 0x0D | 0x80034A4A | | Mute, 1db step, 74 steps, step 74 is 0db |
| Connection length | 0xF00 | 0x0E | 0x00000002 | | Connected to 2 |
| Supported Power states | 0xF00 | 0x0F | 0x8000000F | | EPSS, D0, D1, D2 and D3 |
| Get connection list | 0xF02 | 0x00 | 0x00001110 | | Connected to DAC1, DAC2 |
| Get power state | 0xF05 | 0x00 | 0x00000abc | 0x00000433<br>(P,W) | a = Settings Reset<br>b = Actual state<br>c = Requested state<br>Settings Reset cleared by this verb or any write to this node. |
| Set power state | 0x705 | 0x0a | 0x00000000 | | a = Requested state |

The responses are decoded as follows.

**Table 23-82: Get/Set Amplifier Gain**

| Bit | Description |
|---|---|
| 7 | Amplifier Mute = 1. Mute control is present |
| 6:0 | Amplifier Gain index. 0x00 – 0x4A is valid, 0x4B-0x7F = 0x00 |

**Table 23-83: Audio Widget Capabilities Response**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Read as zero |
| 23:20 | Type = Fh. This is a vendor defined MIXER widget |
| 19:16 | Delay = 0h. Delay is described in the function group |
| 15:12 | Reserved. Read as zero |
| 11 | L-R Swap = 0. Left/right swap is not supported. |

| | |
|---|---|
| 10 | Power Cntrl = 1. Power state control is available. |
| 9 | Digital = 0. This widget is not digital |
| 8 | Conn List = 1. A connection list is available |
| 7 | Unsol Capable = 0. Jack sensing is not available |
| 6 | Proc Widget = 0. No processing capabilities are present |
| 5 | Stripe = 0. Striping is not supported |
| 4 | Format Override = 0. No format information is available |
| 3 | Amp Parameter Override = 1. Amp parameters are available |
| 2 | Out Amp Present = 0. No output amp is present |
| 1 | In Amp Present = 1. Input amp is present |
| 0 | Stereo = 1. This widget supports stereo |

**Table 23-84: Input Amp Capabilities Response**

| Bit | Description |
|---|---|
| 31 | Mute Capable = 1. Mute is supported |
| 30:23 | Reserved. Read as zero |
| 22:16 | Step Size = 03h. Step size is 1dB |
| 15 | Reserved. Read as zero |
| 14:8 | NumSteps = 4Ah. 74 steps |
| 7 | Reserved. Read as zero |
| 6:0 | Offset = 4Ah. Step 74 is 0dB |

**Table 23-85: Connection List Length Response**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | Long Form = 0. List is in short form |
| 6:0 | Connection List Length = 0x02 There are two connections |

**Table 23-86: Supported Power States Response**

| Bit | Description |
|---|---|
| 31 | EPSS = 1. EPSS is supported |
| 30:5 | Reserved. Read as zero |
| 4 | D4Sup = 0. D4 is not supported |
| 3 | D3Sup = 1. D3 is supported |
| 2 | D2Sup = 1. D2 is supported |
| 1 | D1Sup = 1. D1 is supported |
| 0 | D0Sup = 1. D0 is supported |

**Table 23-87: Get Connection List Response**

| Bit | Description |
|---|---|
| 31:0 | 0x00001110 Connected to two widgets |

**Table 23-88: Get Power State Response**

| Bit | Description |
|---|---|
| 31:11 | Reserved. Read as zero |
| 10 | PS_SettingsReset. Codec reset has occurred. |
| 9:7 | Reserved. Read as zero |

| | |
|---|---|
| 6:4 | PS-Act. Actual power state. |
| 3 | Reserved. Read as zero |
| 2:0 | PS-Set. Requested power state |

**Table 23-89: Set Power State Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:4 | Reserved. Set to zero |
| 3 | Reserved. Set to zero |
| 2:0 | PS-Set. 000 = D0, 001 = D1, 010 = D2, 011 = D3 hot, 100 = D3 cold |

## 23.13 Node 16: PORT A / Vendor Widget

The following table describes a PIN that has selectable headphone or line drive and supports jack sensing.

**Table 23-90: Node 16 responses**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Audi widget Pin | 0xF00 | 0x09 | 0x00400581<br>0x00F00000 | | PIN – analog<br>Vendor Widget in merge mode |
| Get pin capabilities | 0xF00 | 0x0C | 0x0001001C | | Output, HP, jack sense, EAPD |
| Connection length | 0xF00 | 0x0E | 0x00000002 | | Connected to 2 |
| Supported Power states | 0xF00 | 0x0F | 0x8000000F | | EPSS, D0, D1, D2 and D3 |
| Get Connection | 0xF01 | 0x00 | 0x0000000a | 0x00000000 (P,W) | DAC 1 selected |
| Set Connection | 0x701 | 0x0a | 0x00000000 | | 0 = DAC 1<br>1 = DAC 2 |
| Get connection list | 0xF02 | 0x00 | 0x00001110 | | DAC 1, 2 |
| Get power state | 0xF05 | 0x00 | 0x00000abc | 0x00000433 (P,W) | a = Settings Reset<br>b = Actual state<br>c = Requested state<br>Settings Reset cleared by this verb or any write to this node. |
| Set power state | 0x705 | 0x0a | 0x00000000 | | a = Requested state |
| Get pin control | 0xF07 | 0x00 | 0x000000a0 | 0x000000C0 (P,W) | a = Headphone and output enable |
| Set pin control | 0x707 | 0xa0 | 0x00000000 | | a = Headphone and output enable |
| Get unsolicited response | 0xF08 | 0x00 | 0x000000aa | 0x00000000 (P,W) | aa = Unsolicited enable and tag |
| Set unsolicited response | 0x708 | 0xaa | 0x00000000 | | aa = Unsolicited enable and tag |
| Get pin sense | 0xF09 | 0x00 | 0xa0000000 | | a = Presence detect<br>8 = present, 0 = missing |
| Get EAPD | 0xF0C | 0x00 | 0x0000000a | 0x00000002 (P,W) | a = EAPD |
| Set EAPD | 0x70C | 0x0a | 0x00000000 | | a = EAPD |
| Get default config | 0xF1C-0xF1F | 0x00 | 0xaabbccdd | 0x042140F0 (P) | aa = Config4<br>bb = Config3<br>cc = Config2<br>dd = Config1 |

| | | | | | |
|---|---|---|---|---|---|
| Set default config 1 | 0x71C | 0xaa | 0x00000000 | | aa = Config1 |
| Set default config 2 | 0x71D | 0xaa | 0x00000000 | | aa = Config2 |
| Set default config 3 | 0x71E | 0xaa | 0x00000000 | | aa = Config3 |
| Set default config 4 | 0x71F | 0xaa | 0x00000000 | | aa = Config4 |

The responses are decoded as follows.

**Table 23-91: Audio Widget Capabilities Response**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Read as zero |
| 23:20 | Type = 4h. This is a PIN widget |
| 19:16 | Delay = 0h. Delay is described in the function group |
| 15:12 | Reserved. Read as zero |
| 11 | L-R Swap = 0. Left/right swap is not supported. |
| 10 | Power Cntrl = 1. Power state control is available. |
| 9 | Digital = 0. This widget is not digital |
| 8 | Conn List = 1. A connection list is available |
| 7 | Unsol Capable = 1. Jack sensing is available |
| 6 | Proc Widget = 0. No processing capabilities are present |
| 5 | Stripe = 0. Striping is not supported |
| 4 | Format Override = 0. No format information is available |
| 3 | Amp Parameter Override = 0. No amp parameters are available |
| 2 | Out Amp Present = 0. No output amp is present |
| 1 | In Amp Present = 0. No input amp is present |
| 0 | Stereo = 1. This widget supports stereo |

**Table 23-92: Pin Capabilities Response**

| Bit | Description |
|---|---|
| 31:17 | Reserved. Read as zero |
| 16 | EAPD capable = 1. EAPD control is supported |
| 15:14 | Reserved. Read as zero |
| 13 | 100% = 0. Vref is not supported by this PIN |
| 12 | 80% = 0. Vref is not supported by this PIN |
| 11 | Reserved. Read as zero |
| 10 | Ground = 0. Vref is not supported by this PIN |
| 9 | 50% = 0. Vref is not supported by this PIN |
| 8 | Hi-Z = 0. Vref is not supported by this PIN |
| 7 | Reserved. Read as zero |
| 6 | Balanced I/O Pins = 0. Balanced I/O is not supported |
| 5 | Input Capable = 0. Input is not supported |
| 4 | Output Capable = 1. Output is supported |
| 3 | Headphone Drive Capable = 1. Headphone drive is supported |
| 2 | Presence Detect Capable = 1. Jack sensing is supported |

| | |
|---|---|
| 1 | Trigger Req'd = 0. No trigger is required for impedance sensing |
| 0 | Impedance Sense Capable = 0. Impedance sensing is not supported |

**Table 23-93: Connection List Length Response**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | Long Form = 0. List is in short form |
| 6:0 | Connection List Length = 02h. There are two connections |

**Table 23-94: Supported Power States Response**

| Bit | Description |
|---|---|
| 31 | EPSS = 1. EPSS is supported |
| 30:5 | Reserved. Read as zero |
| 4 | D4Sup = 0. D4 is not supported |
| 3 | D3Sup = 1. D3 is supported |
| 2 | D2Sup = 1. D2 is supported |
| 1 | D1Sup = 1. D1 is supported |
| 0 | D0Sup = 1. D0 is supported |

**Table 23-95: Get/Set Pin Connection**

| Bit | Description |
|---|---|
| 7:0 | 0x00, 0x01 Connection select, 0x00 = DAC1, 0x01 = DAC2 |

**Table 23-96: Get Connection List Response**

| Bit | Description |
|---|---|
| 31:0 | 0x00001110 DAC1, 2 |

**Table 23-97: Get Power State Response**

| Bit | Description |
|---|---|
| 31:11 | Reserved. Read as zero |
| 10 | PS_SettingsReset. Codec reset has occurred. |
| 9:7 | Reserved. Read as zero |
| 6:4 | PS-Act. Actual power state. |
| 3 | Reserved. Read as zero |
| 2:0 | PS-Set. Requested power state |

**Table 23-98: Set Power State Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:4 | Reserved. Set to zero |
| 3 | Reserved. Set to zero |
| 2:0 | PS-Set. 000 = D0, 001 = D1, 010 = D2, 011 = D3 hot, 100 = D3 cold |

**Table 23-99: Get/Set Pin Widget Control**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | H-Phn Enable = 0 or 1 |

| | |
|---|---|
| 6 | Out Enable = 0 or 1. |
| 5 | In Enable = 0 |
| 4:3 | Reserved. Read as zero |
| 2:0 | VrefEn = 0. Vref not supported |

**Table 23-100: Get/Set Unsolicited**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | Unsolicited enable, 0 = disable, 1 = enable |
| 6 | Not implemented. Read as zero |
| 5:0 | Unsolicited tag value |

**Table 23-101: Get Pin Sense Response**

| Bit | Description |
|---|---|
| 31 | Presence Detect = 0 or 1.<br>0 = nothing inserted in the jack<br>1 = something inserted in the jack |
| 30:0 | Impedance sense value = 0h. |

**Table 23-102: Get Default Config Response**

| Bit | Description |
|---|---|
| 31:24 | Value stored in Config byte 4 |
| 23:16 | Value stored in Config byte 3 |
| 15:8 | Value stored in Config byte 2 |
| 7:0 | Value stored in Config byte 1 |

**Table 23-103: Set Default Config Data Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:0 | Value to store in Config byte 1, 2, 3, or 4 |

## 23.14 Node 17: PORT G BTL Pin Widget

The following pin accepts a stereo signal and drives stereo speakers.

**Table 23-104: Node 17 Responses**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Audio widget Pin | 0xF00 | 0x09 | 0x00400501 | | PIN – analog (opt5=1)<br>Vendor defined(opt5=0) |
| Get pin capabilities | 0xF00 | 0x0C | 0x00010010 | | Output, EAPD |
| Connection length | 0xF00 | 0x0E | 0x00000002 | | Connected to 2 |
| Supported Power states | 0xF00 | 0x0F | 0x8000000F | | EPSS D0, D1, D2 and D3 |
| Get Connection | 0xF01 | 0x00 | 0x0000000a | 0x00000000 (P,W) | DAC 1 selected |
| Set Connection | 0x701 | 0x0a | 0x00000000 | | 0 = DAC 1<br>1 = DAC 2 |
| Get connection list | 0xF02 | 0x00 | 0x00001110 | | Connected to DAC 1, 2 |
| Get power state | 0xF05 | 0x00 | 0x00000abc | 0x00000433 (P,W) | a = Settings Reset<br>b = Actual state<br>c = Requested state<br>Settings Reset cleared by this verb or any write to this node. |
| Set power state | 0x705 | 0x0a | 0x00000000 | | a = Requested state |
| Get pin control | 0xF07 | 0x00 | 0x000000a0 | 0x00000040 (P,W) | a = Output Enable |
| Set pin control | 0x707 | 0xa0 | 0x00000000 | | a = Output Enable |
| Get EAPD | 0xF0C | 0x00 | 0x0000000a | 0x00000002 (P,W) | a = EAPD |
| Set EAPD | 0x70C | 0x0a | 0x00000000 | | a = EAPD |
| Get default config | 0xF1C-0xF1F | 0x00 | 0xaabbccdd | 0x901700F0 (P) | aa = Config4<br>bb = Config3<br>cc = Config2<br>dd = Config1 |
| Set default config 1 | 0x71C | 0xaa | 0x00000000 | | aa = Config1 |
| Set default config 2 | 0x71D | 0xaa | 0x00000000 | | aa = Config2 |
| Set default config 3 | 0x71E | 0xaa | 0x00000000 | | aa = Config3 |
| Set default config 4 | 0x71F | 0xaa | 0x00000000 | | aa = Config4 |

The responses are decoded as follows.

**Table 23-105: Audio Widget Capabilities Response**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Read as zero |
| 23:20 | Type = 4h. This is a PIN widget (or vendor defined if opt5=0) |
| 19:16 | Delay = 0h. Delay is described in the function group |
| 15:12 | Reserved. Read as zero |
| 11 | L-R Swap = 0. Left/right swap is not supported. |
| 10 | Power Cntrl = 1. Power state control is available |
| 9 | Digital = 0. This widget is not digital |
| 8 | Conn List = 1. A connection list is available |
| 7 | Unsol Capable = 0. Jack sensing is not available |
| 6 | Proc Widget = 0. No processing capabilities are present |
| 5 | Stripe = 0. Striping is not supported |
| 4 | Format Override = 0. No format information is available |
| 3 | Amp Parameter Override = 0. No amp parameters are available |
| 2 | Out Amp Present = 0. No output amp is present |
| 1 | In Amp Present = 0. No input amp is present |
| 0 | Stereo = 1. This widget supports Stereo |

**Table 23-106: Pin Capabilities Response**

| Bit | Description |
|---|---|
| 31:17 | Reserved. Read as zero |
| 16 | EAPD capable = 1. EAPD control is supported |
| 15:14 | Reserved. Read as zero |
| 13 | 100% = 0. Vref is not supported by this PIN |
| 12 | 80% = 0. Vref is not supported by this PIN |
| 11 | Reserved. Read as zero |
| 10 | Ground = 0. Vref is not supported by this PIN |
| 9 | 50% = 0. Vref is not supported by this PIN |
| 8 | Hi-Z = 0. Vref is not supported by this PIN |
| 7 | Reserved. Read as zero |
| 6 | Balanced I/O Pins = 0. Balanced I/O is not supported |
| 5 | Input Capable = 0. Input is not supported |
| 4 | Output Capable = 1. Output is supported |
| 3 | Headphone Drive Capable = 0. Headphone drive is not supported |
| 2 | Presence Detect Capable = 0. Jack sensing is not supported |
| 1 | Trigger Req'd = 0. No trigger is required for impedance sensing |
| 0 | Impedance Sense Capable = 0. Impedance sensing is not supported |

**Table 23-107: Connection List Length Response**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | Long Form = 0. List is in short form |
| 6:0 | Connection List Length = 02h. There are 2 connections |

**Table 23-108: Supported Power States Response**

| Bit | Description |
|---|---|
| 31 | EPSS = 1. EPSS is supported |
| 30:5 | Reserved. Read as zero |
| 4 | D4Sup = 0. D4 is not supported |
| 3 | D3Sup = 1. D3 is supported |
| 2 | D2Sup = 1. D2 is supported |
| 1 | D1Sup = 1. D1 is supported |
| 0 | D0Sup = 1. D0 is supported |

**Table 23-109: Get/Set Pin Connection**

| Bit | Description |
|---|---|
| 7:0 | 0x00, 0x01 Connection select, 0x00 = 10, 0x01 = 11 |

**Table 23-110: Get Connection List Response**

| Bit | Description |
|---|---|
| 31:0 | 0x00001110 Connection list, DAC 1, DAC 2 |

**Table 23-111: Get Power State Response**

| Bit | Description |
|---|---|
| 31:11 | Reserved. Read as zero |
| 10 | PS_SettingsReset. Codec reset has occurred. |
| 9:7 | Reserved. Read as zero |
| 6:4 | PS-Act. Actual power state. |
| 3 | Reserved. Read as zero |
| 2:0 | PS-Set. Requested power state |

**Table 23-112: Set Power State Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:4 | Reserved. Set to zero |
| 3 | Reserved. Set to zero |
| 2:0 | PS-Set. 000 = D0, 001 = D1, 010 = D2, 011 = D3 hot, 100 = D3 cold |

**Table 23-113: Get/Set Pin Widget Control**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | H-Phn Enable = 0. |
| 6 | Out Enable = 0 or 1. |
| 5 | In Enable = 0. |
| 4:3 | Reserved. Read as zero |
| 2:0 | VrefEn = 0. Vref not supported |

**Table 23-114: Get Default Config Response**

| Bit | Description |
|---|---|
| 31:24 | Value stored in Config byte 4 |
| 23:16 | Value stored in Config byte 3 |
| 15:8 | Value stored in Config byte 2 |

| | |
|---|---|
| 7:0 | Value stored in Config byte 1 |

**Table 23-115: Set Default Config Data Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:0 | Value to store in Config byte 1, 2, 3, or 4 based on verb address |

## 23.15 Node 18: PORT B Widget

The following table describes a stereo PIN that can be configured to be a line input or a microphone input. There is a microphone boost control and microphone bias.

Mic bias targets; 80% must be 2.6V or above, 50% must be 2.1V or below. This is to facilitate the Apple headset switching modes between tone and resistive button press indications.

**Table 23-116: Node 24 response**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Get amp gain | 0xB00<br>0xB20 | 0x00 | 0x000000aa<br>0x000000aa | 0x00000000<br>(P,W) | aa = Right gain<br>aa = Left gain |
| Set amp gain | 0x350<br>0x360<br>0x370 | 0xaa | 0x00000000 | | aa = Right gain<br>aa = Left gain<br>aa = Left & Right gain |
| Audio widget Pin | 0xF00 | 0x09 | 0x0040048B<br>0x0040058B | Default<br>Merge Mode | Pin – analog<br>Add connection list |
| Get pin capabilities | 0xF00 | 0x0C | 0x00001324<br>0x0001133C | | Vref, In, jack sense |
| Input amp capabilities | 0xF00 | 0x0D | 0x002F0300 | | 12db step, 4 steps, Step 0 is 0db |
| Connection length | 0xF00 | 0x0E | 0x00000002 | | Connected to 2 |
| Supported Power states | 0xF00 | 0x0F | 0x8000000F | | EPSS, D0, D1, D2 and D3 |
| Get Connection | 0xF01 | 0x00 | 0x0000000a | 0x00000000<br>(P,W) | DAC 1 selected |
| Set Connection | 0x701 | 0x0a | 0x00000000 | | 0 = DAC 1<br>1 = DAC 2 |
| Get Connection list | 0xF02 | 0x00 | 0x00001110 | | DAC 1, 2 |
| Get power state | 0xF05 | 0x00 | 0x00000abc | 0x00000433<br>(P,W) | a = Settings Reset<br>b = Actual state<br>c = Requested state<br>Settings Reset cleared by this verb or any write to this node. |
| Set power state | 0x705 | 0x0a | 0x00000000 | | a = Requested state |
| Get pin control | 0xF07 | 0x00 | 0x000000aa | 0x00000000<br>(P,W) | aa = Vref, input enable, output, HP |
| Set pin control | 0x707 | 0xaa | 0x00000000 | | aa = Vref, input enable, output, HP |
| Get unsolicited response | 0xF08 | 0x00 | 0xaa | 0x00000000<br>(P,W) | aa = Unsolicited enable and tag |
| Set unsolicited response | 0x708 | 0xaa | 0x00000000 | | aa = Unsolicited enable and tag |
| Get pin sense | 0xF09 | 0x00 | 0xa0000000 | | a = Presence detect |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | 8 = present, 0 = missing |
| Get EAPD | 0xF0C | 0x00 | 0x0000000a | 0x00000002 (P,W) | a = EAPD |
| Set EAPD | 0x70C | 0x0a | 0x00000000 | | a = EAPD |
| Get default config | 0xF1C-0xF1F | 0x00 | 0xaabbccdd | 0x0481302E (P) | aa = Config4<br>bb = Config3<br>cc = Config2<br>dd = Config1 |
| Set default config 1 | 0x71C | 0xaa | 0x00000000 | | aa = Config1 |
| Set default config 2 | 0x71D | 0xaa | 0x00000000 | | aa = Config2 |
| Set default config 3 | 0x71E | 0xaa | 0x00000000 | | aa = Config3 |
| Set default config 4 | 0x71F | 0xaa | 0x00000000 | | aa = Config4 |

The responses are decoded as follows.

**Table 23-117: Get/Set Amplifier Gain**

| Bit | Description |
|---|---|
| 7 | Amplifier Mute = 0 |
| 6:3 | Read as 0 |
| 2:0 | Amplifier Gain index. |

**Table 23-118: Audio Widget Capabilities Response**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Read as zero |
| 23:20 | Type = 4h. This is a PIN widget |
| 19:16 | Delay = 0h. Delay is described in the function group |
| 15:12 | Reserved. Read as zero |
| 11 | L-R Swap = 0. Left/right swap is not supported. |
| 10 | Power Cntrl = 1. Power state control is available. |
| 9 | Digital = 0. This widget is not digital |
| 8 | Conn List = 0 or 1. Connection list is available in merge mode |
| 7 | Unsol Capable = 1. Jack sensing is available |
| 6 | Proc Widget = 0. No processing capabilities are present |
| 5 | Stripe = 0. Striping is not supported |
| 4 | Format Override = 0. No format information is available |
| 3 | Amp Parameter Override = 1. Amp parameters are available |
| 2 | Out Amp Present = 0. No output amp is present |
| 1 | In Amp Present = 1. Input amp is present |
| 0 | Stereo = 1. This widget supports Stereo |

**Table 23-119: Pin Capabilities Response**

| Bit | Description |
|---|---|
| 31:17 | Reserved. Read as zero |
| 16 | EAPD capable = 0 or 1. EAPD control is supported in merge mode |

| | |
|---|---|
| 15:14 | Reserved. Read as zero |
| 13 | 100% = 0. Vref is not supported by this PIN |
| 12 | 80% = 1. Vref is supported by this PIN |
| 11 | Reserved. Read as zero |
| 10 | Ground = 0. Vref is not supported by this PIN |
| 9 | 50% = 1. Vref is supported by this PIN |
| 8 | Hi-Z = 1. Vref is supported by this PIN |
| 7 | Reserved. Read as zero |
| 6 | Balanced I/O Pins = 0. Balanced I/O is not supported |
| 5 | Input Capable = 1. Input is supported |
| 4 | Output Capable = 0 or 1. Output is supported in merge mode |
| 3 | Headphone Drive Capable = 0 or 1. Headphone drive is supported in merge mode |
| 2 | Presence Detect Capable = 1. Jack sensing is supported |
| 1 | Trigger Req'd = 0. No trigger is required for impedance sensing |
| 0 | Impedance Sense Capable = 0. Impedance sensing is not supported |

**Table 23-120: Input Amp Capabilities Response**

| Bit | Description |
|---|---|
| 31 | Mute Capable = 0. Mute is not supported |
| 30:23 | Reserved. Read as zero |
| 22:16 | Step Size = 2Fh. Step size is 12dB |
| 15 | Reserved. Read as zero |
| 14:8 | NumSteps = 3h. 4 steps |
| 7 | Reserved. Read as zero |
| 6:0 | Offset = 0h. Step 0 is 0dB |

**Table 23-121: Connection List Length Response**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | Long Form = 0. List is in short form |
| 6:0 | Connection List Length = 02h. There are two connections |

**Table 23-122: Supported Power States Response**

| Bit | Description |
|---|---|
| 31 | EPSS = 1. EPSS is supported |
| 30:5 | Reserved. Read as zero |
| 4 | D4Sup = 0. D4 is not supported |
| 3 | D3Sup = 1. D3 is supported |
| 2 | D2Sup = 1. D2 is supported |
| 1 | D1Sup = 1. D1 is supported |
| 0 | D0Sup = 1. D0 is supported |

**Table 23-123: Get/Set Pin Connection**

| Bit | Description |
|---|---|
| 7:0 | 0x00, 0x01 Connection select, 0x00 = 10, 0x01 = 11 |

**Table 23-124: Get Connection List Response index 0**

| Bit | Description |
|---|---|
| 31:0 | 0x00001110 |

**Table 23-125: Get Power State Response**

| Bit | Description |
|---|---|
| 31:11 | Reserved. Read as zero |
| 10 | PS_SettingsReset. Codec reset has occurred. |
| 9:7 | Reserved. Read as zero |
| 6:4 | PS-Act. Actual power state. |
| 3 | Reserved. Read as zero |
| 2:0 | PS-Set. Requested power state |

**Table 23-126: Set Power State Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:4 | Reserved. Set to zero |
| 3 | Reserved. Set to zero |
| 2:0 | PS-Set. 000 = D0, 001 = D1, 010 = D2, 011 = D3 hot, 100 = D3 cold |

**Table 23-127: Get/Set Pin Widget Control**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | H-Phn Enable = 0 or 1. |
| 6 | Out Enable = 0 or 1. |
| 5 | In Enable = 0 or 1. |
| 4:3 | Reserved. Read as zero |
| 2:0 | VrefEn = 0 (Disable), 1 (50%), or 4 (80%). |

**Table 23-128: Get/Set Unsolicited**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | Unsolicited enable, 0 = disable, 1 = enable |
| 6 | Not implemented. Read as zero |
| 5:0 | Unsolicited tag value |

**Table 23-129: Get Pin Sense Response**

| Bit | Description |
|---|---|
| 31 | Presence Detect = 0 or 1.<br>0 = nothing inserted in the jack<br>1 = something inserted in the jack |
| 30:0 | Impedance sense value = 0h. |

**Table 23-130: Get Default Config Response**

| Bit | Description |
|---|---|
| 31:24 | Value stored in Config byte 4 |
| 23:16 | Value stored in Config byte 3 |

| | |
|---|---|
| 15:8 | Value stored in Config byte 2 |
| 7:0 | Value stored in Config byte 1 |

**Table 23-131: Set Default Config Data Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:0 | Value to store in Config byte 1, 2, 3, or 4 based on verb address |

## 23.16 Node 19: PORT D Widget

The following table describes a stereo PIN that is a microphone input. There is a microphone boost control and microphone bias.

Mic bias targets; 80% must be 2.6V or above, 50% must be 2.1V or below. This is to facilitate the Apple headset switching modes between tone and resistive button press indications.

**Table 23-132: Node 19 response**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Get amp gain | 0xB00<br>0xB20 | 0x00 | 0x000000aa<br>0x000000aa | 0x00000000<br>(P,W) | aa = Right gain<br>aa = Left gain |
| Set amp gain | 0x350<br>0x360<br>0x370 | 0xaa | 0x00000000 | | aa = Right gain<br>aa = Left gain<br>aa = Left & Right gain |
| Audio widget Pin | 0xF00 | 0x09 | 0x0040048B<br>0x0040040B | Headset enabled | Pin – analog |
| Get pin capabilities | 0xF00 | 0x0C | 0x00001324<br>0x00001320 | Headset enabled | Vref, In, jack sense<br>Vref, In |
| Input amp capabilities | 0xF00 | 0x0D | 0x002F0300 | | 12db step, 4 steps, Step 0 is 0db |
| Supported Power states | 0xF00 | 0x0F | 0x8000000F | | EPSS, D0, D1, D2 and D3 |
| Get power state | 0xF05 | 0x00 | 0x00000abc | 0x00000433<br>(P,W) | a = Settings Reset<br>b = Actual state<br>c = Requested state<br>Settings Reset cleared by this verb or any write to this node. |
| Set power state | 0x705 | 0x0a | 0x00000000 | | a = Requested state |
| Get pin control | 0xF07 | 0x00 | 0x000000aa | 0x00000000<br>(P,W) | aa = Vref, input enable |
| Set pin control | 0x707 | 0xaa | 0x00000000 | | aa = Vref, input enable |
| Get unsolicited response | 0xF08 | 0x00 | 0xaa | 0x00000000<br>(P,W) | aa = Unsolicited enable and tag |
| Set unsolicited response | 0x708 | 0xaa | 0x00000000 | | aa = Unsolicited enable and tag |
| Get pin sense | 0xF09 | 0x00 | 0xa0000000 | | a = Presence detect<br>8 = present, 0 = missing |
| Get default config | 0xF1C-<br>0xF1F | 0x00 | 0xaabbccdd | 0x04A19020<br>(P) | aa = Config4<br>bb = Config3<br>cc = Config2<br>dd = Config1 |
| Set default config 1 | 0x71C | 0xaa | 0x00000000 | | aa = Config1 |
| Set default config 2 | 0x71D | 0xaa | 0x00000000 | | aa = Config2 |

| | | | | | |
|---|---|---|---|---|---|
| Set default config 3 | 0x71E | 0xaa | 0x00000000 | | aa = Config3 |
| Set default config 4 | 0x71F | 0xaa | 0x00000000 | | aa = Config4 |

The responses are decoded as follows.

**Table 23-133: Get/Set Amplifier Gain**

| Bit | Description |
|---|---|
| 7 | Amplifier Mute = 0 |
| 6:3 | Read as 0 |
| 2:0 | Amplifier Gain index. |

**Table 23-134: Audio Widget Capabilities Response**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Read as zero |
| 23:20 | Type = 4h. This is a PIN widget |
| 19:16 | Delay = 0h. Delay is described in the function group |
| 15:12 | Reserved. Read as zero |
| 11 | L-R Swap = 0. Left/right swap is not supported. |
| 10 | Power Cntrl = 1. Power state control is available. |
| 9 | Digital = 0. This widget is not digital |
| 8 | Conn List = 0. Connection list is not available |
| 7 | Unsol Capable = 1. Jack sensing is available |
| 6 | Proc Widget = 0. No processing capabilities are present |
| 5 | Stripe = 0. Striping is not supported |
| 4 | Format Override = 0. No format information is available |
| 3 | Amp Parameter Override = 1. Amp parameters are available |
| 2 | Out Amp Present = 0. No output amp is present |
| 1 | In Amp Present = 1. Input amp is present |
| 0 | Stereo = 1. This widget supports Stereo |

**Table 23-135: Pin Capabilities Response**

| Bit | Description |
|---|---|
| 31:17 | Reserved. Read as zero |
| 16 | EAPD capable = 0. EAPD control is not supported |
| 15:14 | Reserved. Read as zero |
| 13 | 100% = 0. Vref is not supported by this PIN |
| 12 | 80% = 1. Vref is supported by this PIN |
| 11 | Reserved. Read as zero |
| 10 | Ground = 0. Vref is not supported by this PIN |
| 9 | 50% = 1. Vref is supported by this PIN |
| 8 | Hi-Z = 1. Vref is supported by this PIN |
| 7 | Reserved. Read as zero |
| 6 | Balanced I/O Pins = 0. Balanced I/O is not supported |
| 5 | Input Capable = 1. Input is supported |

| | |
|---|---|
| 4 | Output Capable = 0. Output is not supported |
| 3 | Headphone Drive Capable = 0. Headphone drive is not supported |
| 2 | Presence Detect Capable = 1 or 0. Jack sensing is optionally supported |
| 1 | Trigger Req'd = 0. No trigger is required for impedance sensing |
| 0 | Impedance Sense Capable = 0. Impedance sensing is not supported |

**Table 23-136: Input Amp Capabilities Response**

| Bit | Description |
|---|---|
| 31 | Mute Capable = 0. Mute is not supported |
| 30:23 | Reserved. Read as zero |
| 22:16 | Step Size = 2Fh. Step size is 12dB |
| 15 | Reserved. Read as zero |
| 14:8 | NumSteps = 3h. 4 steps |
| 7 | Reserved. Read as zero |
| 6:0 | Offset = 0h. Step 0 is 0dB |

**Table 23-137: Supported Power States Response**

| Bit | Description |
|---|---|
| 31 | EPSS = 1. EPSS is supported |
| 30:5 | Reserved. Read as zero |
| 4 | D4Sup = 0. D4 is not supported |
| 3 | D3Sup = 1. D3 is supported |
| 2 | D2Sup = 1. D2 is supported |
| 1 | D1Sup = 1. D1 is supported |
| 0 | D0Sup = 1. D0 is supported |

**Table 23-138: Get Power State Response**

| Bit | Description |
|---|---|
| 31:11 | Reserved. Read as zero |
| 10 | PS_SettingsReset. Codec reset has occurred. |
| 9:7 | Reserved. Read as zero |
| 6:4 | PS-Act. Actual power state. |
| 3 | Reserved. Read as zero |
| 2:0 | PS-Set. Requested power state |

**Table 23-139: Set Power State Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:4 | Reserved. Set to zero |
| 3 | Reserved. Set to zero |
| 2:0 | PS-Set. 000 = D0, 001 = D1, 010 = D2, 011 = D3 hot, 100 = D3 cold |

**Table 23-140: Get/Set Pin Widget Control**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | H-Phn Enable = 0. |
| 6 | Out Enable = 0. |

| | |
|---|---|
| 5 | In Enable = 0 or 1. |
| 4:3 | Reserved. Read as zero |
| 2:0 | VrefEn = 0 (Disable), 1 (50%0, or 4 (80%). |

**Table 23-141: Get/Set Unsolicited**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | Unsolicited enable, 0 = disable, 1 = enable |
| 6 | Not implemented. Read as zero |
| 5:0 | Unsolicited tag value |

**Table 23-142: Get Pin Sense Response**

| Bit | Description |
|---|---|
| 31 | Presence Detect = 0 or 1.<br>0 = nothing inserted in the jack<br>1 = something inserted in the jack |
| 30:0 | Impedance sense value = 0h. |

**Table 23-143: Get Default Config Response**

| Bit | Description |
|---|---|
| 31:24 | Value stored in Config byte 4 |
| 23:16 | Value stored in Config byte 3 |
| 15:8 | Value stored in Config byte 2 |
| 7:0 | Value stored in Config byte 1 |

**Table 23-144: Set Default Config Data Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:0 | Value to store in Config byte 1, 2, 3, or 4 based on verb address |

## 23.17 Node 1A: PORT C Widget

Port C is a stereo digital microphone input by default that is reported as a stereo analog pin complex. A register option is available to configure this widget as an analog microphone/line input. The connection list for ADC2 changes depending on the mode but, otherwise, the capabilities for port C are similar when in digital or analog mode with the exception of Vref. The mic bias pin (Vref), as described in a separate section, is shared with port B and is not responsive to port C controls under most circumstances.

The following table describes the analog stereo PIN that is a digital microphone input.

**Table 23-145: Node 1A Responses**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Get amp gain | 0xB00<br>0xB20 | 0x00 | 0x000000aa<br>0x000000aa | 0x00000000<br>(P,W) | aa = Right gain<br>aa = Left gain |
| Set amp gain | 0x350<br>0x360<br>0x370 | 0xaa | 0x00000000 | | aa = Right gain<br>aa = Left gain<br>aa = Left & Right gain |
| Audio widget Pin | 0xF00 | 0x09 | 0x0040048B | | PIN - analog |
| Get pin capabilities | 0xF00 | 0x0C | 0x00001324 | | Input, Vref, Presence |
| Input amp capabilities | 0xF00 | 0x0D | 0x002F0300 | | 12db step, 4 steps, Step 0 is 0db |
| Supported Power states | 0xF00 | 0x0F | 0x8000000F | | EPSS D0, D1, D2 and D3 |
| Get power state | 0xF05 | 0x00 | 0x00000abc | 0x00000433<br>(P,W) | a = Settings Reset<br>b = Actual state<br>c = Requested state<br>Settings Reset cleared by this verb or any write to this node. |
| Set power state | 0x705 | 0x0a | 0x00000000 | | a = Requested state |
| Get pin control | 0xF07 | 0x00 | 0x000000aa | 0x00000000<br>(P,W) | aa = Vref, Input Enable |
| Set pin control | 0x707 | 0xaa | 0x00000000 | | aa = Vref, Input Enable |
| Get unsolicited response | 0xF08 | 0x00 | 0xaa | 0x00000000<br>(P,W) | aa = Unsolicited enable and tag |
| Set unsolicited response | 0x708 | 0xaa | 0x00000000 | | aa = Unsolicited enable and tag |
| Get pin sense | 0xF09 | 0x00 | 0xa0000000 | | a = Presence detect<br>8 = present, 0 = missing |
| Get default config | 0xF1C-<br>0xF1F | 0x00 | 0xaabbccdd | 0x90A701F0<br>(P) | aa = Config4<br>bb = Config3<br>cc = Config2<br>dd = Config1 |
| Set default config 1 | 0x71C | 0xaa | 0x00000000 | | aa = Config1 |

| | | | | | |
|---|---|---|---|---|---|
| Set default config 2 | 0x71D | 0xaa | 0x00000000 | | aa = Config2 |
| Set default config 3 | 0x71E | 0xaa | 0x00000000 | | aa = Config3 |
| Set default config 4 | 0x71F | 0xaa | 0x00000000 | | aa = Config4 |

The responses are decoded as follows.

**Table 23-146: Get/Set Amplifier Gain**

| Bit | Description |
|---|---|
| 7 | Amplifier Mute = 0 |
| 6:3 | Read as 0 |
| 2:0 | Amplifier Gain index. |

**Table 23-147: Audio Widget Capabilities Response**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Read as zero |
| 23:20 | Type = 4h. This is a PIN widget |
| 19:16 | Delay = 0h. Delay is described in the function group |
| 15:12 | Reserved. Read as zero |
| 11 | L-R Swap = 0. Left/right swap is not supported. |
| 10 | Power Cntrl = 1. Power state control is available. |
| 9 | Digital = 0. This widget is not digital |
| 8 | Conn List = 0. A connection list is not available |
| 7 | Unsol Capable = 1. Jack sensing is available |
| 6 | Proc Widget = 0. No processing capabilities are present |
| 5 | Stripe = 0. Striping is not supported |
| 4 | Format Override = 0. No format information is available |
| 3 | Amp Parameter Override = 1. Amp parameters are available |
| 2 | Out Amp Present = 0. No output amp is present |
| 1 | In Amp Present = 1. Input amp is present |
| 0 | Stereo = 1. This widget supports stereo |

**Table 23-148: Pin Capabilities Response**

| Bit | Description |
|---|---|
| 31:17 | Reserved. Read as zero |
| 16 | EAPD capable = 0. EAPD control is not supported |
| 15:14 | Reserved. Read as zero |
| 13 | 100% = 0. Vref is not supported by this PIN |
| 12 | 80% = 1. Vref is optionally supported by this PIN |
| 11 | Reserved. Read as zero |
| 10 | Ground = 0. Vref is not supported by this PIN |
| 9 | 50% = 1. Vref is supported by this PIN |
| 8 | Hi-Z = 1. Vref is optionally supported by this PIN |
| 7 | Reserved. Read as zero |
| 6 | Balanced I/O Pins = 0. Balanced I/O is not supported |

| | |
|---|---|
| 5 | Input Capable = 1. Input is supported |
| 4 | Output Capable = 0. Output is not supported |
| 3 | Headphone Drive Capable = 0. Headphone drive is not supported |
| 2 | Presence Detect Capable = 1. Jack sensing is supported |
| 1 | Trigger Req'd = 0. No trigger is required for impedance sensing |
| 0 | Impedance Sense Capable = 0. Impedance sensing is not supported |

**Table 23-149: Supported Power States Response**

| Bit | Description |
|---|---|
| 31 | EPSS = 1. EPSS is supported |
| 30:5 | Reserved. Read as zero |
| 4 | D4Sup = 0. D4 is not supported |
| 3 | D3Sup = 1. D3 is supported |
| 2 | D2Sup = 1. D2 is supported |
| 1 | D1Sup = 1. D1 is supported |
| 0 | D0Sup = 1. D0 is supported |

**Table 23-150: Get Power State Response**

| Bit | Description |
|---|---|
| 31:11 | Reserved. Read as zero |
| 10 | PS_SettingsReset. Codec reset has occurred. |
| 9:7 | Reserved. Read as zero |
| 6:4 | PS-Act. Actual power state. |
| 3 | Reserved. Read as zero |
| 2:0 | PS-Set. Requested power state |

**Table 23-151: Set Power State Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:4 | Reserved. Set to zero |
| 3 | Reserved. Set to zero |
| 2:0 | PS-Set. 000 = D0, 001 = D1, 010 = D2, 011 = D3 hot, 100 = D3 cold |

**Table 23-152: Get/Set Pin Widget Control**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | H-Phn Enable = 0. |
| 6 | Out Enable = 0. |
| 5 | In Enable = 0 or 1. |
| 4:3 | Reserved. Read as zero |
| 2:0 | VrefEn = 0 (Disable) or 4 (80%). |

**Table 23-153: Get Default Config Response**

| Bit | Description |
|---|---|
| 31:24 | Value stored in Config byte 4 |
| 23:16 | Value stored in Config byte 3 |
| 15:8 | Value stored in Config byte 2 |

| | |
|---|---|
| 7:0 | Value stored in Config byte 1 |

**Table 23-154: Set Default Config Data Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:0 | Value to store in Config byte 1, 2, 3, or 4 based on verb address |

## 23.18 Node 1D, PORT E Widget

Port E is a digital audio interface output port that supports I2S, and TDM (PCM) modes. It is described as an analog pin for software compatibility and is typically connected to an external CODEC for docking applications or an internal speaker amplifier.

The following table describes a PIN that is a stereo line output and supports jack sensing.
This port is available on the CX8400 and CX8450 parts.

**Table 23-155: Node 1D responses**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Audio widget Pin | 0xF00 | 0x09 | 0x00400581 | | PIN – analog CX84xx |
| Get pin capabilities | 0xF00 | 0x0C | 0x0001001C | | Output, HP, jack sense, EAPD |
| Connection length | 0xF00 | 0x0E | 0x00000001 | | Connected to DAC 4 |
| Supported Power states | 0xF00 | 0x0F | 0x8000000F | | EPSS, D0, D1, D2 and D3 |
| Get Connection | 0xF01 | 0x00 | 0x0000000a | 0x00000000 (P,W) | DAC 4 selected |
| Set Connection | 0x701 | 0x0a | 0x00000000 | | 0 = DAC 4 |
| Get connection list | 0xF02 | 0x00 | 0x00000022 | | DAC 4 |
| Get power state | 0xF05 | 0x00 | 0x00000abc | 0x00000433 (P,W) | a = Settings Reset<br>b = Actual state<br>c = Requested state<br>Settings Reset cleared by this verb or any write to this node. |
| Set power state | 0x705 | 0x0a | 0x00000000 | | a = Requested state |
| Get pin control | 0xF07 | 0x00 | 0x000000a0 | 0x00000040 (P,W) | a = Vref, input enable and output enable |
| Set pin control | 0x707 | 0xa0 | 0x00000000 | | a = Vref, input enable and output enable |
| Get unsolicited response | 0xF08 | 0x00 | 0x000000aa | 0x00000000 (P,W) | aa = Unsolicited enable and tag |
| Set unsolicited response | 0x708 | 0xaa | 0x00000000 | | aa = Unsolicited enable and tag |
| Get pin sense | 0xF09 | 0x00 | 0xa0000000 | | a = Presence detect<br>8 = present, 0 = missing |
| Get EAPD | 0xF0C | 0x00 | 0x0000000a | 0x00000002 (P,W) | a = EAPD |
| Set EAPD | 0x70C | 0x0a | 0x00000000 | | a = EAPD |
| Get default config | 0xF1C-0xF1F | 0x00 | 0xaabbccdd | 0x240140F0 (P) | aa = Config4<br>bb = Config3<br>cc = Config2<br>dd = Config1 |

| | | | | | |
|---|---|---|---|---|---|
| Set default config 1 | 0x71C | 0xaa | 0x00000000 | | aa = Config1 |
| Set default config 2 | 0x71D | 0xaa | 0x00000000 | | aa = Config2 |
| Set default config 3 | 0x71E | 0xaa | 0x00000000 | | aa = Config3 |
| Set default config 4 | 0x71F | 0xaa | 0x00000000 | | aa = Config4 |

The responses are decoded as follows.

**Table 23-156: Audio Widget Capabilities Response**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Read as zero |
| 23:20 | Type = 4h or Fh. This is a pin widget or vendor defined |
| 19:16 | Delay = 0h. Delay is described in the function group |
| 15:12 | Reserved. Read as zero |
| 11 | L-R Swap = 0. Left/right swap is not supported. |
| 10 | Power Cntrl = 1. Power state control is available when a pin complex. |
| 9 | Digital = 0. This widget is not digital |
| 8 | Conn List = 1. A connection list is available |
| 7 | Unsol Capable = 1. Jack sensing is available |
| 6 | Proc Widget = 0. No processing capabilities are present |
| 5 | Stripe = 0. Striping is not supported |
| 4 | Format Override = 0. No format information is available |
| 3 | Amp Parameter Override = 0. No amp parameters are available |
| 2 | Out Amp Present = 0. No output amp is present |
| 1 | In Amp Present = 0. No input amp is present |
| 0 | Stereo = 1. This widget supports stereo |

**Table 23-157: Pin Capabilities Response**

| Bit | Description |
|---|---|
| 31:17 | Reserved. Read as zero |
| 16 | EAPD capable = 1. EAPD control is supported |
| 15:14 | Reserved. Read as zero |
| 13 | 100% = 0. Vref is not supported by this PIN |
| 12 | 80% = 0. Vref is not supported by this PIN |
| 11 | Reserved. Read as zero |
| 10 | Ground = 0. Vref is not supported by this PIN |
| 9 | 50% = 0. Vref is not supported by this PIN |
| 8 | Hi-Z = 0. Vref is not supported by this PIN |
| 7 | Reserved. Read as zero |
| 6 | Balanced I/O Pins = 0. Balanced I/O is not supported |
| 5 | Input Capable = 0. Input is supported |
| 4 | Output Capable = 1. Output is supported |

| | |
|---|---|
| 3 | Headphone Drive Capable =1. Headphone drive is supported (for docking). |
| 2 | Presence Detect Capable = 1. Jack sensing is supported |
| 1 | Trigger Req'd = 0. No trigger is required for impedance sensing |
| 0 | Impedance Sense Capable = 0. Impedance sensing is not supported |

**Table 23-158: Connection List Length Response**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | Long Form = 0. List is in short form |
| 6:0 | Connection List Length = 01h. There is one connection |

**Table 23-159: Supported Power States Response**

| Bit | Description |
|---|---|
| 31 | EPSS = 1. EPSS is supported |
| 30:5 | Reserved. Read as zero |
| 4 | D4Sup = 0. D4 is not supported |
| 3 | D3Sup = 1. D3 is supported |
| 2 | D2Sup = 1. D2 is supported |
| 1 | D1Sup = 1. D1 is supported |
| 0 | D0Sup = 1. D0 is supported |

**Table 23-160: Get/Set Pin Connection**

| Bit | Description |
|---|---|
| 7:0 | 0x00 Connection select, 0x00 = DAC4 |

**Table 23-161: Get Connection List Response**

| Bit | Description |
|---|---|
| 31:0 | 0x00000022 DAC4 |

**Table 23-162: Get Power State Response**

| Bit | Description |
|---|---|
| 31:11 | Reserved. Read as zero |
| 10 | PS_SettingsReset. Codec reset has occurred. |
| 9:7 | Reserved. Read as zero |
| 6:4 | PS-Act. Actual power state. |
| 3 | Reserved. Read as zero |
| 2:0 | PS-Set. Requested power state |

**Table 23-163: Set Power State Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:4 | Reserved. Set to zero |
| 3 | Reserved. Set to zero |
| 2:0 | PS-Set. 000 = D0, 001 = D1, 010 = D2, 011 = D3 hot, 100 = D3 cold |

**Table 23-164: Get/Set Pin Widget Control**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | H-Phn Enable = 0 or 1. |
| 6 | Out Enable = 0 or 1. |
| 5 | In Enable = 0 |
| 4:3 | Reserved. Read as zero |
| 2:0 | VrefEn = 0 (Disabled), 1 (50%), or 4 (80%) |

**Table 23-165: Get/Set Unsolicited**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | Unsolicited enable, 0 = disable, 1 = enable |
| 6 | Not implemented. Read as zero |
| 5:0 | Unsolicited tag value |

**Table 23-166: Get Pin Sense Response**

| Bit | Description |
|---|---|
| 31 | Presence Detect = 0 or 1.<br>0 = nothing inserted in the jack<br>1 = something inserted in the jack |
| 30:0 | Impedance sense value = 0h. |

**Table 23-167: Get Default Config Response**

| Bit | Description |
|---|---|
| 31:24 | Value stored in Config byte 4 |
| 23:16 | Value stored in Config byte 3 |
| 15:8 | Value stored in Config byte 2 |
| 7:0 | Value stored in Config byte 1 |

**Table 23-168: Set Default Config Data Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:0 | Value to store in Config byte 1, 2, 3, or 4 |

## 23.19 Node 1E, PORT F Widget

Port F is a digital audio interface input port that supports I2S, and TDM (PCM) modes. It is described as an analog pin for software compatibility and is typically connected to an external CODEC for docking applications or an internal speaker amplifier for AEC.

The following table describes a PIN that is a stereo line or mic input and supports jack sensing.
This port is available on the CX8400 and CX8450 parts.

**Table 23-169: Node 1E response**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Get amp gain | 0xB00<br>0xB20 | 0x00 | 0x000000aa<br>0x000000aa | 0x00000000<br>(P,W) | aa = Right gain<br>aa = Left gain |
| Set amp gain | 0x350<br>0x360<br>0x370 | 0xaa | 0x00000000 | | aa = Right gain<br>aa = Left gain<br>aa = Left & Right gain |
| Audio widget Pin | 0xF00 | 0x09 | 0x0040048B | | Pin – analog CX84xx |
| Get pin capabilities | 0xF00 | 0x0C | 0x00000024 | | Input, jack sense |
| Input amp capabilities | 0xF00 | 0x0D | 0x002F0300 | | 12db step, 4 steps, step 0 is 0db |
| Supported Power states | 0xF00 | 0x0F | 0x8000000F | | EPSS, D0, D1, D2 and D3 |
| Get power state | 0xF05 | 0x00 | 0x00000abc | 0x00000433<br>(P,W) | a = Settings Reset<br>b = Actual state<br>c = Requested state<br>Settings Reset cleared by this verb or any write to this node. |
| Set power state | 0x705 | 0x0a | 0x00000000 | | a = Requested state |
| Get pin control | 0xF07 | 0x00 | 0x000000aa | 0x00000000<br>(P,W) | aa = input enable |
| Set pin control | 0x707 | 0xaa | 0x00000000 | | aa = input enable |
| Get unsolicited response | 0xF08 | 0x00 | 0xaa | 0x00000000<br>(P,W) | aa = Unsolicited enable and tag |
| Set unsolicited response | 0x708 | 0xaa | 0x00000000 | | aa = Unsolicited enable and tag |
| Get pin sense | 0xF09 | 0x00 | 0xa0000000 | | a = Presence detect<br>8 = present, 0 = missing |
| Get default config | 0xF1C-<br>0xF1F | 0x00 | 0xaabbccdd | 0x248130F0<br>(P) | aa = Config4<br>bb = Config3<br>cc = Config2<br>dd = Config1 |
| Set default config 1 | 0x71C | 0xaa | 0x00000000 | | aa = Config1 |

| | | | | | |
|---|---|---|---|---|---|
| Set default config 2 | 0x71D | 0xaa | 0x00000000 | | aa = Config2 |
| Set default config 3 | 0x71E | 0xaa | 0x00000000 | | aa = Config3 |
| Set default config 4 | 0x71F | 0xaa | 0x00000000 | | aa = Config4 |

The responses are decoded as follows.

**Table 23-170: Get/Set Amplifier Gain**

| Bit | Description |
|---|---|
| 7 | Amplifier Mute = 0 |
| 6:3 | Read as 0 |
| 2:0 | Amplifier Gain index. 0=0db, 1=12db, 2=24db, 3=36db |

**Table 23-171: Audio Widget Capabilities Response**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Read as zero |
| 23:20 | Type = 4h or Fh. This is a PIN widget or a vendor defined widget |
| 19:16 | Delay = 0h. Delay is described in the function group |
| 15:12 | Reserved. Read as zero |
| 11 | L-R Swap = 0. Left/right swap is not supported. |
| 10 | Power Cntrl = 1. Power state control is available. |
| 9 | Digital = 0. This widget is not digital |
| 8 | Conn List = 0. Connection list is available |
| 7 | Unsol Capable = 1. Jack sensing is available |
| 6 | Proc Widget = 0. No processing capabilities are present |
| 5 | Stripe = 0. Striping is not supported |
| 4 | Format Override = 0. No format information is available |
| 3 | Amp Parameter Override = 1. Amp parameters are available |
| 2 | Out Amp Present = 0. No output amp is present |
| 1 | In Amp Present = 1. Input amp is present |
| 0 | Stereo = 1. This widget supports Stereo |

**Table 23-172: Pin Capabilities Response**

| Bit | Description |
|---|---|
| 31:17 | Reserved. Read as zero |
| 16 | EAPD capable = 0. EAPD control is not supported |
| 15:14 | Reserved. Read as zero |
| 13 | 100% = 0. Vref is not supported by this PIN |
| 12 | 80% = 0. Vref is not supported by this PIN |
| 11 | Reserved. Read as zero |
| 10 | Ground = 0. Vref is not supported by this PIN |
| 9 | 50% = 0. Vref is not supported by this PIN |
| 8 | Hi-Z = 0. Vref is not supported by this PIN |
| 7 | Reserved. Read as zero |
| 6 | Balanced I/O Pins = 0. Balanced I/O is not supported |

| | |
|---|---|
| 5 | Input Capable = 1. Input is supported |
| 4 | Output Capable = 0. Output is not supported |
| 3 | Headphone Drive Capable = 0. Headphone drive is not supported |
| 2 | Presence Detect Capable = 1. Jack sensing is supported |
| 1 | Trigger Req'd = 0. No trigger is required for impedance sensing |
| 0 | Impedance Sense Capable = 0. Impedance sensing is not supported |

**Table 23-173: Input Amp Capabilities Response**

| Bit | Description |
|---|---|
| 31 | Mute Capable = 0. Mute is not supported |
| 30:23 | Reserved. Read as zero |
| 22:16 | Step Size = 2Fh. Step size is 12dB |
| 15 | Reserved. Read as zero |
| 14:8 | NumSteps = 3h. 4 steps |
| 7 | Reserved. Read as zero |
| 6:0 | Offset = 0h. Step 0 is 0dB |

**Table 23-174: Supported Power States Response**

| Bit | Description |
|---|---|
| 31 | EPSS = 1. EPSS is supported |
| 30:5 | Reserved. Read as zero |
| 4 | D4Sup = 0. D4 is not supported |
| 3 | D3Sup = 1. D3 is supported |
| 2 | D2Sup = 1. D2 is supported |
| 1 | D1Sup = 1. D1 is supported |
| 0 | D0Sup = 1. D0 is supported |

**Table 23-175: Get Power State Response**

| Bit | Description |
|---|---|
| 31:11 | Reserved. Read as zero |
| 10 | PS_SettingsReset. Codec reset has occurred. |
| 9:7 | Reserved. Read as zero |
| 6:4 | PS-Act. Actual power state. |
| 3 | Reserved. Read as zero |
| 2:0 | PS-Set. Requested power state |

**Table 23-176: Set Power State Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:4 | Reserved. Set to zero |
| 3:2 | Reserved. Set to zero |
| 1:0 | PS-Set. 000 = D0, 001 = D1, 010 = D2, 011 = D3 hot, 100 = D3 cold |

**Table 23-177: Get/Set Pin Widget Control**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | H-Phn Enable = 0. |

| | |
|---|---|
| 6 | Out Enable = 0. |
| 5 | In Enable = 0 or 1. |
| 4:3 | Reserved. Read as zero |
| 2:0 | VrefEn = 0. |

**Table 23-178: Get/Set Unsolicited**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | Unsolicited enable, 0 = disable, 1 = enable |
| 6 | Not implemented. Read as zero |
| 5:0 | Unsolicited tag value |

**Table 23-179: Get Pin Sense Response**

| Bit | Description |
|---|---|
| 31 | Presence Detect = 0 or 1.<br>0 = nothing inserted in the jack<br>1 = something inserted in the jack |
| 30:0 | Impedance sense value = 0h. |

**Table 23-180: Get Default Config Response**

| Bit | Description |
|---|---|
| 31:24 | Value stored in Config byte 4 |
| 23:16 | Value stored in Config byte 3 |
| 15:8 | Value stored in Config byte 2 |
| 7:0 | Value stored in Config byte 1 |

**Table 23-181: Set Default Config Data Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:0 | Value to store in Config byte 1, 2, 3, or 4 based on verb address |

## 23.20 Node 1F: PORT H Widget

Port H is a stereo digital microphone.

The following table describes the analog stereo PIN that is a digital microphone input.

**Table 23-182: Node 1F Responses**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Get amp gain | 0xB00<br>0xB20 | 0x00 | 0x000000aa<br>0x000000aa | 0x00000000<br>(P,W) | aa = Right gain<br>aa = Left gain |
| Set amp gain | 0x350<br>0x360<br>0x370 | 0xaa | 0x00000000 | | aa = Right gain<br>aa = Left gain<br>aa = Left & Right gain |
| Audio widget Pin | 0xF00 | 0x09 | 0x0040040B | | PIN - analog CX8400/CX8450 |
| Get pin capabilities | 0xF00 | 0x0C | 0x00000020 | | Input only |
| Input amp capabilities | 0xF00 | 0x0D | 0x002F0300 | | 12db step, 4 steps, Step 0 is 0db |
| Supported Power states | 0xF00 | 0x0F | 0x8000000F | | EPSS D0, D1, D2 and D3 |
| Get power state | 0xF05 | 0x00 | 0x00000abc | 0x00000433<br>(P,W) | a = Settings Reset<br>b = Actual state<br>c = Requested state<br>Settings Reset cleared by this verb or any write to this node. |
| Set power state | 0x705 | 0x0a | 0x00000000 | | a = Requested state |
| Get pin control | 0xF07 | 0x00 | 0x000000aa | 0x00000000<br>(P,W) | aa = Input Enable |
| Set pin control | 0x707 | 0xaa | 0x00000000 | | aa = Input Enable |
| Get default config | 0xF1C-<br>0xF1F | 0x00 | 0xaabbccdd | 0x90A700F0<br>(P) | aa = Config4<br>bb = Config3<br>cc = Config2<br>dd = Config1 |
| Set default config 1 | 0x71C | 0xaa | 0x00000000 | | aa = Config1 |
| Set default config 2 | 0x71D | 0xaa | 0x00000000 | | aa = Config2 |
| Set default config 3 | 0x71E | 0xaa | 0x00000000 | | aa = Config3 |
| Set default config 4 | 0x71F | 0xaa | 0x00000000 | | aa = Config4 |

The responses are decoded as follows.

**Table 23-183: Get/Set Amplifier Gain**

| Bit | Description |
|---|---|
| 7 | Amplifier Mute = 0 |
| 6:3 | Read as 0 |
| 2:0 | Amplifier Gain index. |

**Table 23-184: Audio Widget Capabilities Response**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Read as zero |
| 23:20 | Type = 4h. This is a PIN widget |
| 19:16 | Delay = 0h. Delay is described in the function group |
| 15:12 | Reserved. Read as zero |
| 11 | L-R Swap = 0. Left/right swap is not supported. |
| 10 | Power Cntrl = 1. Power state control is available. |
| 9 | Digital = 0. This widget is not digital |
| 8 | Conn List = 0. A connection list is not available |
| 7 | Unsol Capable = 0. Jack sensing is not available |
| 6 | Proc Widget = 0. No processing capabilities are present |
| 5 | Stripe = 0. Striping is not supported |
| 4 | Format Override = 0. No format information is available |
| 3 | Amp Parameter Override = 1. Amp parameters are available |
| 2 | Out Amp Present = 0. No output amp is present |
| 1 | In Amp Present = 1. Input amp is present |
| 0 | Stereo = 1. This widget supports stereo |

**Table 23-185: Pin Capabilities Response**

| Bit | Description |
|---|---|
| 31:17 | Reserved. Read as zero |
| 16 | EAPD capable = 0. EAPD control is not supported |
| 15:14 | Reserved. Read as zero |
| 13 | 100% = 0. Vref is not supported by this PIN |
| 12 | 80% = 0. Vref is optionally supported by this PIN |
| 11 | Reserved. Read as zero |
| 10 | Ground = 0. Vref is not supported by this PIN |
| 9 | 50% = 0. Vref is not supported by this PIN |
| 8 | Hi-Z = 0. Vref is optionally supported by this PIN |
| 7 | Reserved. Read as zero |
| 6 | Balanced I/O Pins = 0. Balanced I/O is not supported |
| 5 | Input Capable = 1. Input is supported |
| 4 | Output Capable = 0. Output is not supported |
| 3 | Headphone Drive Capable = 0. Headphone drive is not supported |
| 2 | Presence Detect Capable = 0. Jack sensing is not supported |
| 1 | Trigger Req'd = 0. No trigger is required for impedance sensing |
| 0 | Impedance Sense Capable = 0. Impedance sensing is not supported |

**Table 23-186: Supported Power States Response**

| Bit | Description |
|---|---|

| | |
|---|---|
| 31 | EPSS = 1. EPSS is supported |
| 30:5 | Reserved. Read as zero |
| 4 | D4Sup = 0. D4 is not supported |
| 3 | D3Sup = 1. D3 is supported |
| 2 | D2Sup = 1. D2 is supported |
| 1 | D1Sup = 1. D1 is supported |
| 0 | D0Sup = 1. D0 is supported |

**Table 23-187: Get Power State Response**

| Bit | Description |
|---|---|
| 31:11 | Reserved. Read as zero |
| 10 | PS_SettingsReset. Codec reset has occurred. |
| 9:7 | Reserved. Read as zero |
| 6:4 | PS-Act. Actual power state. |
| 3 | Reserved. Read as zero |
| 2:0 | PS-Set. Requested power state |

**Table 23-188: Set Power State Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:4 | Reserved. Set to zero |
| 3 | Reserved. Set to zero |
| 2:0 | PS-Set. 000 = D0, 001 = D1, 010 = D2, 011 = D3 hot, 100 = D3 cold |

**Table 23-189: Get/Set Pin Widget Control**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | H-Phn Enable = 0. |
| 6 | Out Enable = 0. |
| 5 | In Enable = 0 or 1. |
| 4:3 | Reserved. Read as zero |
| 2:0 | VrefEn = 0 (Disable) or 4 (80%). |

**Table 23-190: Get Default Config Response**

| Bit | Description |
|---|---|
| 31:24 | Value stored in Config byte 4 |
| 23:16 | Value stored in Config byte 3 |
| 15:8 | Value stored in Config byte 2 |
| 7:0 | Value stored in Config byte 1 |

**Table 23-191: Set Default Config Data Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:0 | Value to store in Config byte 1, 2, 3, or 4 based on verb address |

## 23.21 Node 1B: Vendor Specific Widget

The following table describes the BIOS, EQ and DRC registers available in this device.

There is a 7-channel EQ that is in the Class-D, stereo line out path to protect and equalize the performance of the small speakers found in notebooks. The 7-channel EQ is implemented with 7 bi-quad filters that are programmable. A 7-channel EQ is one possible use for these 7 bi-quads. There is also a single order high pass filter with a programmable -3db point. The purpose of this high pass is to prevent audio content with a significant DC offset from heating and possibly burning out the speakers. Audio content with a large DC component can easily be found in home made movies and from internet sites such as YouTube. The high pass filter supports the frequency range of 30Hz to 1890Hz in 64 steps, 30Hz per step.

**Please refer to the Vendor Defined Registers section for the list of implemented registers.**

**Table 23-192: Node 1B responses**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Get EQ Configuration | 0xA | 0xaa00 | 0x000000bb | 0x00000000 (P) | aa= 6-bit register number (right aligned)<br>bb = 8-bit value |
| Set EQ Configuration | 0x2 | 0xaabb | 0x00000000 | | aa= 6-bit register number<br>bb = 8-bit value |
| Get BIOS test | 0xB | 0xa000 | 0x000000bb | 0x00000000 (P) | aa= 6-bit register number<br>bb = 8-bit value |
| Set BIOS test | 0x3 | 0xaabb | 0x00000000 | | aa= 6-bit register number<br>bb = 8-bit value |
| | 0xC | 0xaa00 | 0x000000bb | 0x00000000 (P) | aa = 6-bit register number<br>bb = 8-bit value |
| | 0x4 | 0xaabb | 0x00000000 | | aa= 6-bit register number<br>bb = 8-bit value |
| Audio widget Vendor | F00 | 0x09 | 0x00F00000 | | |

### 23.21.1 EQ Registers

HD Audio widget 1B verbs A00/200 to A0E/20E
I2C register address 0x6D00 to 0x6D0E

### 23.21.2 DRC Registers

HD Audio widget 1B verbs A10/210 to A1A/21A
I2C register address 0x6D10 to 0x6D1A

### 23.21.3 EQ Lock Registers

HD Audio widget 1B verbs A0F, 22E to 22F
I2C register address 0x6D0F, 0x6D2E, and 0X6D2F

### 23.21.4 BIOS Registers

HD Audio widget 1B verbs B00/300 to B13/313, D0C/50C to D0F/50F, D1C/51C to D1E/51E
I2C register address 0x6D80 to 0x6D93, 6E8C to 6E8F, 0X6E9C to 0x6E9E

### 23.21.5 MBIST Registers

HD Audio widget 1B verbs B1F/31F to B21/321
I2C register address 0x6D9F to 0x6DA1

### 23.21.6 I2S Registers

HD Audio widget 1B verbs C00/400 to C1D/41D, C20/420 to C2F/42F, E30/630 to E33/633
I2C register address 0x6E00 to 0x6E1D, 0x6E20 to 0x6E2F, 0x6F30 to 0x6F33

## 23.22 Node 1C: Vendor Specific Widget

Please refer to the Vendor1C register section in the Vendor Defined Registers chapter.

**Table 23-193: Node 1C responses**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| | 0xA | 0xa000 | 0x00000bbb | 0x00000000 (P) | a= register number bbb = 12-bit value |
| | 0x2 | 0xabbb | 0x00000000 | | a= register number bbb = 12-bit value |
| | 0xB | 0xa000 | 0x00000bbb | 0x00000000 (P) | a= register number bbb = 12-bit value |
| | 0x3 | 0xabbb | 0x00000000 | | a= register number bbb = 12-bit value |
| | 0xC | 0xa000 | 0x00000bbb | 0x00000000 (P) | a= register number bbb = 12-bit value |
| | 0x4 | 0xabbb | 0x00000000 | | a= register number bbb = 12-bit value |
| Audio widget Vendor | F00 | 0x09 | 0x00F00000 | | |

### 23.22.1 Codec Test Registers

### 23.22.2 Analog Test registers

### 23.22.3 Digital Test registers

## 23.23 Node 20: DAC 3 – SPDIF OUTPUT Widget

The following table describes an S/PDIF DAC that supports 16 and 24 bit widths and 48 kilohertz sample rate.

**Table 23-194: Node 20 response**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Get converter format | 0xA | 0x0000 | 0x0000aaaa | 0x00000031 (P,W) | aaaa = Converter format |
| Set converter format | 0x2 | 0xaaaa | 0x00000000 | | aaaa = Converter format |
| Audio widget S/PDIF | 0xF00 | 0x09 | 0x00000611 | | DAC – digital, stereo |
| PCM size and rate | 0xF00 | 0x0A | 0x000E0540 | | 16, 20, & 24 bit, 48k,96k, 192k sample rate |
| PCM format | 0xF00 | 0x0B | 0x00000005 | | PCM & AC3 |
| Supported Power states | 0xF00 | 0x0F | 0x8000000F | | EPSS, D0, D1, D2 and D3 |
| Get power state | 0xF05 | 0x00 | 0x00000abc | 0x00000433 (P,W) | a = Settings Reset<br>b = Actual state<br>c = Requested state<br>Settings Reset cleared by this verb or any write to this node. |
| Set power state | 0x705 | 0x0a | 0x00000000 | | a = Requested state |
| Get converter stream/chnl | 0xF06 | 0x00 | 0x000000ab | 0x00000000 (P,R,V,W,D) | a = Stream, b = Channel |
| Set converter stream/chnl | 0x706 | 0xab | 0x00000000 | | a = Stream, b = Channel |
| Get digital converter control | 0xF0D<br>0xF0E | 0x00 | 0xaabbccdd | 0x00000000 (P,W) | aa = Reserved<br>bb = Coding Mode<br>cc =Category code<br>dd = Header information |
| Set digital control 1 | 0x70D | 0xaa | 0x00000000 | | aa = Header information |
| Set digital control 2 | 0x70E | 0xaa | 0x00000000 | | aa = Category code |
| Set digital control 3 | 0x73E | 0xaa | 0x00000000 | | aa = IEC coding type and Keep Alive Enable |
| Set digital control 4 | 0x73F | 0x00 | 0x00000000 | | Reserved read as 0 |

The responses are decoded as follows:

**Table 23-195: Get/Set Converter Format**

| Bit | Description |
|---|---|
| 31:16 | Reserved. Read as zero |
| 15 | Non-PCM = 0 or 1. PCM and AC3 is supported |

| | |
|---|---|
| 14 | Sample Base Rate = 0 or 1, 48 or 44.1 kHz |
| 13:11 | Sample Base Rate Multiplier = 000, 001 or 011.<br>000b = times 1 (44.1 or 48 kHz)<br>001b = times 2 (88.2 or 96 kHz)<br>011b = times 4 (176.4 or 192 kHz) |
| 10:8 | Sample Base Rate Divisor = 000. Divide by 1 |
| 7 | Reserved. Read as zero |
| 6:4 | Bits per Sample = 001, 010 or 011.<br>001 = 16 bits<br>010 = 20 bits<br>011 = 24 bits |
| 3:0 | Number of Channels = 0 or 1.<br>0 = 1 channel (mono)<br>1 = 2 channels (stereo) |

**Table 23-196: Audio Widget Capabilities Response**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Read as zero |
| 23:20 | Type = 0h. This is a DAC widget |
| 19:16 | Delay = 0h. Delay is described in the function group |
| 15:12 | Reserved. Read as zero |
| 11 | L-R Swap = 0. Left/right swap is not supported. |
| 10 | Power Cntrl = 1. Power state control is available |
| 9 | Digital = 1. This widget is digital |
| 8 | Conn List = 0. No connection list is available |
| 7 | Unsol Capable = 0. Jack sensing is not available |
| 6 | Proc Widget = 0. No processing capabilities are present |
| 5 | Stripe = 0. Striping is not supported |
| 4 | Format Override = 1. Format information is available |
| 3 | Amp Parameter Override = 0. No amp parameters are available |
| 2 | Out Amp Present = 0. No output amp is present |
| 1 | In Amp Present = 0. No input amp is present |
| 0 | Stereo = 1. This widget supports stereo |

**Table 23-197: Sample Size, Rate Caps Response**

| Bit | Description |
|---|---|
| 31:21 | Reserved. Read as zero |
| 20 | B32 = 0. 32-bit audio format is not supported |
| 19 | B24 = 1. 24-bit audio format is supported |
| 18 | B20 = 1. 20-bit audio format is supported |
| 17 | B16 = 1. 16-bit audio format is supported |
| 16 | B8 = 0. 8-bit audio format is not supported |
| 15:12 | Reserved. Read as zero |
| 11 | R12 = 0. 384 kHz audio format is not supported |
| 10 | R11 = 1. 192 kHz audio format is supported |
| 9 | R10 = 0. 176.4 kHz audio format is not supported |
| 8 | R9 = 1. 96 kHz audio format is supported |

| | |
|---|---|
| 7 | R8 = 0. 88.2 kHz audio format is not supported |
| 6 | R7 = 1. 48 kHz audio format is supported |
| 5 | R6 = 0. 44.1 kHz audio format is not supported |
| 4 | R5 = 0. 32 kHz audio format is not supported |
| 3 | R4 = 0. 22.05 kHz audio format is not supported |
| 2 | R3 = 0. 16 kHz audio format is not supported |
| 1 | R2 = 0. 11.025 kHz audio format is not supported |
| 0 | R1 = 0. 8.0 kHz audio format is not supported |

**Table 23-198: Stream Formats Response**

| Bit | Description |
|---|---|
| 31:3 | Reserved. Read as zero |
| 2 | AC3 = 1. AC3 format is supported |
| 1 | Float32 = 0. 32-bit float format is not supported |
| 0 | PCM = 1. PCM audio format is supported |

**Table 23-199: Supported Power States Response**

| Bit | Description |
|---|---|
| 31 | EPSS = 1. EPSS is supported |
| 30:5 | Reserved. Read as zero |
| 4 | D4Sup = 0. D4 is not supported |
| 3 | D3Sup = 1. D3 is supported |
| 2 | D2Sup = 1. D2 is supported |
| 1 | D1Sup = 1. D1 is supported |
| 0 | D0Sup = 1. D0 is supported |

**Table 23-200: Get Power State Response**

| Bit | Description |
|---|---|
| 31:11 | Reserved. Read as zero |
| 10 | PS_SettingsReset. Codec reset has occurred. |
| 9:7 | Reserved. Read as zero |
| 6:4 | PS-Act. Actual power state. |
| 3 | Reserved. Read as zero |
| 2:0 | PS-Set. Requested power state |

**Table 23-201: Set Power State Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:4 | Reserved. Set to zero |
| 3 | Reserved. Set to zero |
| 2:0 | PS-Set. 000 = D0, 001 = D1, 010 = D2, 011 = D3 hot, 100 = D3 cold |

**Table 23-202: Get/Set Channel and Stream ID**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7:4 | Stream ID, 0 = disable stream 1 –> F = enable stream |
| 3:0 | Channel ID |

**Table 23-203: Get Digital Converter Info format**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Set to zero |
| 23 | Keep Alive Enable |
| 22:20 | Reserved. Set to zero |
| 19:16 | IEC Coding Type [3:0] |
| 15 | Reserved. Set to zero |
| 14:8 | CC[6:0]. |
| 7 | L |
| 6 | Pro |
| 5 | /Audio |
| 4 | COPY |
| 3 | PRE |
| 2 | VCFG |
| 1 | V |
| 0 | DigEn.<br>0 = Digital data doesn't pass through the node<br>1 = Digital data does pass through the node |

**Table 23-204: Set Digital Converter 1 Info format**

| Bit | Description |
|---|---|
| 7 | L |
| 6 | Pro |
| 5 | /Audio |
| 4 | COPY |
| 3 | PRE |
| 2 | VCFG |
| 1 | V |
| 0 | DigEn.<br>0 = Digital data doesn't pass through the node<br>1 = Digital data does pass through the node |

**Table 23-205: Set Digital Converter 2 Info format**

| Bit | Description |
|---|---|
| 7 | Reserved. Set to zero |
| 6:0 | CC[6:0]. |

**Table 23-206: Set Digital Converter 3 Info format**

| Bit | Description |
|---|---|
| 7 | Keep Alive Enable |
| 6:4 | Reserved. Set to zero |
| 3:0 | IEC Coding Type [3:0] |

**Table 23-207: Set Digital Converter 4 Info format**

| Bit | Description |
|---|---|
| 7:0 | Reserved. Set to zero |

## 23.24 Node 21: PORT I SPDIF OUTPUT Widget

This table describes a digital output PIN.

When the SPDIF DAC is connected setting the Pin output enable to 0 (disable output) will act as if the SPDIF Vcfg bit was set to 0 and the V bit set to 1. This "mutes" the transmitter by marking all the data as invalid. The SPDIF clock, header and data will still be sent. Setting this bit to 1 allows the transmitter to behave as specified by the V and Vcfg bits.

**Table 23-208: Node 21 responses**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Audio widget Pin | 0xF00 | 0x09 | 0x00400701 | | PIN – digital |
| Get pin capabilities | 0xF00 | 0x0C | 0x00000010 | | Output |
| Connection length | 0xF00 | 0x0E | 0x00000001 | | |
| Supported Power states | 0xF00 | 0x0F | 0x8000000F | | EPSS D0, D1, D2 and D3 |
| Get connection list | 0xF02 | 0x00 | 0x00000020 | | Connects to S/PDIF DAC (DAC3) |
| Get power state | 0xF05 | 0x00 | 0x00000abc | 0x00000433 (P,W) | a = Settings Reset<br>b = Actual state<br>c = Requested state<br>Settings Reset cleared by this verb or any write to this node. |
| Set power state | 0x705 | 0x0a | 0x00000000 | | a = Requested state |
| Get pin control | 0xF07 | 0x00 | 0x000000a0 | 0x00000000 (P,W) | a = Output Enable |
| Set pin control | 0x707 | 0xa0 | 0x00000000 | | a = Output Enable |
| Get default config | 0xF1C-0xF1F | 0x00 | 0xaabbccdd | 0x044510F0 (P) | aa = Config4<br>bb = Config3<br>cc = Config2<br>dd = Config1 |
| Set default config 1 | 0x71C | 0xaa | 0x00000000 | | aa = Config1 |
| Set default config 2 | 0x71D | 0xaa | 0x00000000 | | aa = Config2 |
| Set default config 3 | 0x71E | 0xaa | 0x00000000 | | aa = Config3 |
| Set default config 4 | 0x71F | 0xaa | 0x00000000 | | aa = Config4 |

The responses are decoded as follows.

**Table 23-209: Audio Widget Capabilities Response**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Read as zero |
| 23:20 | Type = 4h. This is a PIN widget |
| 19:16 | Delay = 0h. Delay is described in the function group |
| 15:12 | Reserved. Read as zero |
| 11 | L-R Swap = 0. Left/right swap is not supported. |
| 10 | Power Cntrl = 1. Power state control is available. |
| 9 | Digital = 1. This widget is digital |
| 8 | Conn List = 1. A connection list is available |
| 7 | Unsol Capable = 0. Jack sensing is not available |
| 6 | Proc Widget = 0. No processing capabilities are present |
| 5 | Stripe = 0. Striping is not supported |
| 4 | Format Override = 0. No format information is available |
| 3 | Amp Parameter Override = 0. No amp parameters are available |
| 2 | Out Amp Present = 0. No output amp is present |
| 1 | In Amp Present = 0. No input amp is present |
| 0 | Stereo = 1. This widget supports stereo |

**Table 23-210: Pin Capabilities Response**

| Bit | Description |
|---|---|
| 31:17 | Reserved. Read as zero |
| 16 | EAPD capable = 0. EAPD control is not supported |
| 15:14 | Reserved. Read as zero |
| 13 | 100% = 0. Vref is not supported by this PIN |
| 12 | 80% = 0. Vref is not supported by this PIN |
| 11 | Reserved. Read as zero |
| 10 | Ground = 0. Vref is not supported by this PIN |
| 9 | 50% = 0. Vref is not supported by this PIN |
| 8 | Hi-Z = 0. Vref is not supported by this PIN |
| 7 | HDMI = 0. HDMI is not supported |
| 6 | Balanced I/O Pins = 0. Balanced I/O is not supported |
| 5 | Input Capable = 0. Input is not supported |
| 4 | Output Capable = 1. Output is supported |
| 3 | Headphone Drive Capable = 0. Headphone drive is not supported |
| 2 | Presence Detect Capable = 0. Jack sensing is not supported |
| 1 | Trigger Req'd = 0. No trigger is required for impedance sensing |
| 0 | Impedance Sense Capable = 0. Impedance sensing is not supported |

**Table 23-211: Connection List Length Response**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | Long Form = 0. List is in short form |
| 6:0 | Connection List Length = 01h. There is one connection |

**Table 23-212: Supported Power States Response**

| Bit | Description |
|---|---|
| 31 | EPSS = 1. EPSS is supported |

| | |
|---|---|
| 30:5 | Reserved. Read as zero |
| 4 | D4Sup = 0. D4 is not supported |
| 3 | D3Sup = 1. D3 is supported |
| 2 | D2Sup = 1. D2 is supported |
| 1 | D1Sup = 1. D1 is supported |
| 0 | D0Sup = 1. D0 is supported |

**Table 23-213: Get Connection List Response**

| Bit | Description |
|---|---|
| 31:0 | 0x00000020 Connection list |

**Table 23-214: Get Power State Response**

| Bit | Description |
|---|---|
| 31:11 | Reserved. Read as zero |
| 10 | PS_SettingsReset. Codec reset has occurred. |
| 9:7 | Reserved. Read as zero |
| 6:4 | PS-Act. Actual power state. |
| 3 | Reserved. Read as zero |
| 2:0 | PS-Set. Requested power state |

**Table 23-215: Set Power State Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:4 | Reserved. Set to zero |
| 3 | Reserved. Set to zero |
| 2:0 | PS-Set. 000 = D0, 001 = D1, 010 = D2, 011 = D3 hot, 100 = D3 cold |

**Table 23-216: Get/Set Pin Widget Control**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | H-Phn Enable = 0. |
| 6 | Out Enable = 0 or 1. |
| 5 | In Enable = 0. |
| 4:3 | Reserved. Read as zero |
| 2:0 | VrefEn = 0. Vref not supported |

**Table 23-217: Get Default Config Response**

| Bit | Description |
|---|---|
| 31:24 | Value stored in Config byte 4 |
| 23:16 | Value stored in Config byte 3 |
| 15:8 | Value stored in Config byte 2 |
| 7:0 | Value stored in Config byte 1 |

**Table 23-218: Set Default Config Data Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:0 | Value to store in Config byte 1, 2, 3, or 4 based on verb address |

Conexant Systems Inc
Proprietary and Confidential
ALL INFORMATION IN THIS DOCUMENT IS PRELIMINARY AND SUBJECT TO CHANGE WITHOUT NOTICE

## 23.25 Nodes 22 and 23: DAC 4 and 5 Widgets

The following table describes a stereo DAC that supports 16 and 24 bit widths and 48/96/192 kilohertz sample rates. These DACs connect to digital audio interface ports and allow for setting the format on the host interface.

**Table 23-219: Node 10 and 11 responses**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Get converter format | 0xA | 0x0000 | 0x0000aaaa | 0x00000031 (P,W) | aaaa = converter format |
| Set converter format | 0x2 | 0xaaaa | 0x00000000 | | aaaa = converter format |
| Get amp gain | 0xB80<br>0xBA0 | 0x00<br>0x00 | 0x000000aa | 0x0000004A (P,W) | aa = Right gain<br>aa = Left Gain |
| Set amp gain | 0x390<br>0x3A0<br>0x3B0 | 0xaa | 0x00000000 | | aa = Right gain<br>aa = Left gain<br>aa = Right & Left gain |
| Audio widget DAC | 0xF00 | 0x09 | 0x00000C1D | | DAC – analog |
| PCM size and rate | 0xF00 | 0x0A | 0x000A0540 | | 16 & 24 bit<br>48, 96, and 192KHz |
| PCM Format | 0xF00 | 0x0B | 0x00000001 | | PCM only |
| Supported Power states | 0xF00 | 0x0F | 0x8000000F | | EPSS, D0, D1, D2, and D3 |
| Get output amp capabilities | 0xF00 | 0x12 | 0x80034A4A | | Mute, 1db step,<br>Step 74 is 0db<br>74 of 80 steps exposed |
| Get power state | 0xF05 | 0x00 | 0x00000abc | 0x00000433 (P,W) | a = Settings Reset<br>b = Actual state<br>c = Requested state<br>Settings Reset cleared by this verb or any write to this node. |
| Set power state | 0x705 | 0x0a | 0x00000000 | | a = Requested state |
| Get converter stream/chnl | 0xF06 | 0x00 | 0x000000ab | 0x00000000 (P,R,V,W,D) | a = Stream<br>b = Channel position |
| Set converter stream/chnl | 0x706 | 0xab | 0x00000000 | | a = Stream<br>b = Channel position |
| Get EAPD | 0xF0C | 0x00 | 0x0000000a | 0x00000000 (P,W) | a = left/right swap |
| Set EAPD | 0x70C | 0x0a | 0x00000000 | | a = left/right swap |

The responses are decoded as follows:

**Table 23-220: Get/Set Converter Format**

| Bit | Description |
|---|---|
| 31:16 | Reserved. Read as zero |
| 15 | Non-PCM = 0. Only PCM is supported |
| 14 | Sample Base Rate = 0 or 1, 48 or 44.1 kHz |
| 13:11 | Sample Base Rate Multiplier = 000, 001 or 011.<br>000b = times 1 (44.1 or 48 kHz)<br>001b = times 2 (96kHz)<br>011b = times 4 (192kHz) |
| 10:8 | Sample Base Rate Divisor = 000. Divide by 1<br>000b = divide 1 (48 or 44.1 kHz) |
| 7 | Reserved. Read as zero |
| 6:4 | Bits per Sample = 001 or 011.<br>001 = 16 bits<br>011 = 24 bits |
| 3:0 | Number of Channels = 0 or 1.<br>0 = 1 channel (mono)<br>1 = 2 channels (stereo) |

**Table 23-221: Get/Set Amplifier Gain**

| Bit | Description |
|---|---|
| 7 | Amplifier Mute = 0 or 1 |
| 6:0 | Amplifier Gain index. |

**Table 23-222: Audio Widget Capabilities Response**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Read as zero |
| 23:20 | Type = 0h. This is a DAC widget |
| 19:16 | Delay = 0h. Delay is described in the function group |
| 15:12 | Reserved. Read as zero |
| 11 | L-R Swap = 1. Left/right swap is supported. |
| 10 | Power Cntrl = 1. Power state control is available |
| 9 | Digital = 0. This widget is not digital |
| 8 | Conn List = 0. No connection list is available |
| 7 | Unsol Capable = 0. Jack sensing is not available |
| 6 | Proc Widget = 0. No processing capabilities are present |
| 5 | Stripe = 0. Striping is not supported |
| 4 | Format Override = 1. Format information is not available |
| 3 | Amp Parameter Override = 1. Amp parameters are available |
| 2 | Out Amp Present = 1. Output amp is present |
| 1 | In Amp Present = 0. No input amp is present |
| 0 | Stereo = 1. This widget supports stereo |

**Table 23-223: Sample Size, Rate Caps Response**

| Bit | Description |
|---|---|
| 31:21 | Reserved. Read as zero |
| 20 | B32 = 0. 32-bit audio format is not supported |
| 19 | B24 = 1. 24-bit audio format is supported |

| | |
|---|---|
| 18 | B20 = 0. 20-bit audio format is not supported |
| 17 | B16 = 1. 16-bit audio format is supported |
| 16 | B8 = 0. 8-bit audio format is not supported |
| 15:12 | Reserved. Read as zero |
| 11 | R12 = 0. 384 kHz audio format is not supported |
| 10 | R11 = 1. 192 kHz audio format is not supported |
| 9 | R10 = 0. 176.4 kHz audio format is not supported |
| 8 | R9 = 1. 96 kHz audio format is not supported |
| 7 | R8 = 0. 88.2 kHz audio format is not supported |
| 6 | R7 = 1. 48 kHz audio format is supported |
| 5 | R6 = 0. 44.1 kHz audio format is not supported |
| 4 | R5 = 0. 32 kHz audio format is not supported |
| 3 | R4 = 0. 22.05 kHz audio format is not supported |
| 2 | R3 = 0. 16 kHz audio format is not supported |
| 1 | R2 = 0. 11.025 kHz audio format is not supported |
| 0 | R1 = 0. 8.0 kHz audio format is not supported |

**Table 23-224: Stream Formats Response**

| Bit | Description |
|---|---|
| 31:3 | Reserved. Read as zero |
| 2 | AC3 = 0. AC3 format is not supported |
| 1 | Float32 = 0. 32-bit float format is not supported |
| 0 | PCM = 1. PCM audio format is supported |

**Table 23-225: Supported Power States Response**

| Bit | Description |
|---|---|
| 31 | EPSS = 1. EPSS is supported |
| 30:5 | Reserved. Read as zero |
| 4 | D4Sup = 0. D4 is not supported |
| 3 | D3Sup = 1. D3 is supported |
| 2 | D2Sup = 1. D2 is supported |
| 1 | D1Sup = 1. D1 is supported |
| 0 | D0Sup = 1. D0 is supported |

**Table 23-226: Output Amp Capabilities Response**

| Bit | Description |
|---|---|
| 31 | Mute Capable = 1. Mute is supported |
| 30:23 | Reserved. Read as zero |
| 22:16 | Step Size = 03h. Step size is 1dB |
| 15 | Reserved. Read as zero |
| 14:8 | NumSteps = 4Ah. There are 80 steps but only 74 are exposed |
| 7 | Reserved. Read as zero |
| 6:0 | Offset = 4Ah. Step 74 is 0dB |

**Table 23-227: Get Power State Response**

| Bit | Description |
|---|---|
| 31:11 | Reserved. Read as zero |

| | |
|---|---|
| 10 | PS_SettingsReset. Codec reset has occurred. |
| 9:7 | Reserved. Read as zero |
| 6:4 | PS-Act. Actual power state. |
| 3 | Reserved. Read as zero |
| 2:0 | PS-Set. Requested power state |

**Table 23-228: Set Power State Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:4 | Reserved. Set to zero |
| 3 | Reserved. Set to zero |
| 2:0 | PS-Set. 000 = D0, 001 = D1, 010 = D2, 011 = D3 hot, 100 = D3 cold |

**Table 23-229: Get/Set Channel and Stream ID**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7:4 | Stream ID, 0 = disable stream 1 –> F = enable stream |
| 3:0 | Channel ID |

**Table 23-230: Get EAPD/BTL Response**

| Bit | Description |
|---|---|
| 31:3 | Reserved. Read as zero |
| 2 | Left/Right swap = 0 or 1. |
| 1 | EAPD = 0. |
| 0 | BTL = 0. |

## 23.26 Nodes 24 and 25: ADC 3 and 4 Widgets

The following table describes a stereo ADC that supports 16 and 24 bit widths and 48, 96, or 192 kilohertz sample rates. The ADC has a gain stage. The ADC connects to a digital audio interface port and configures the format on the host interface. These widgets are not available in the CX807x and CX812x parts and are not present in the widget list.

**Table 23-231: Node 14 Responses**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Get converter format | 0xA | 0x0000 | 0x0000aaaa | 0x00000031 (P,W) | aaaa = converter format |
| Set converter format | 0x2 | 0xaaaa | 0x00000000 | | aaaa = converter format |
| Get index 0 amp gain | 0xB00<br>0xB20 | 0x00 | 0x000000aa<br>0x000000aa | 0x0000004A (P,W) | aa = Right gain<br>aa = Left gain |
| Set index 0 amp gain | 0x350<br>0x360<br>0x370 | 0xaa | 0x00000000 | | aa = Right gain<br>aa = Left gain<br>aa = Left & Right gain |
| Audio widget ADC | 0xF00 | 0x09 | 0x00100D1B | | ADC - analog |
| PCM size and rate | 0xF00 | 0x0A | 0x000A0540 | | 16&24 bit / 48, 96, & 192 kHz |
| PCM format | 0xF00 | 0x0B | 0x00000001 | | PCM only |
| Input amp capabilities | 0xF00 | 0x0D | 0x8003504A | | Mute, 1db step, 80 steps, Step 74 is 0db |
| Connection length | 0xF00 | 0x0E | 0x00000001 | | |
| Supported Power states | 0xF00 | 0x0F | 0x8000000F | | EPSS, D0, D1, D2 and D3 |
| Get connection select | 0xF01 | 0x00 | 0x0000000a | 0x00000000 (P,W) | a = Connection Index |
| Set connection select | 0x701 | 0x0a | 0x00000000 | | a = Connection Index |
| Get connection list | 0xF02 | 0x00 | 0x0000001E<br>0x00000027 | | Port F for ADC 3<br>Port K for ADC 4 |
| Get power state | 0xF05 | 0x00 | 0x00000abc | 0x00000433 (P,W) | a = Settings Reset<br>b = Actual state<br>c = Requested state<br>Settings Reset cleared by this verb or any write to this node. |
| Set power state | 0x705 | 0x0a | 0x00000000 | | a = Requested state |
| Get converter stream/chnl | 0xF06 | 0x00 | 0x000000ab | 0x00000000 (P,R,V,W,D) | a = Stream<br>b = Channel position |

| | | | | | |
|---|---|---|---|---|---|
| Set converter stream/chnl | 0x706 | 0xab | 0x00000000 | | a = Stream<br>b = Channel position |
| Get EAPD | 0xF0C | 0x00 | 0x0000000a | 0x00000000 (P,W) | a = left/right swap |
| Set EAPD | 0x70C | 0x0a | 0x00000000 | | a = left/right swap |

The responses are decoded as follows:

**Table 23-232: Get/Set Converter Format**

| Bit | Description |
|---|---|
| 31:16 | Reserved. Read as zero |
| 15 | Non-PCM = 0. Only PCM is supported |
| 14 | Sample Base Rate = 0 or 1, 48 or 44.1 kHz |
| 13:11 | Sample Base Rate Multiplier = 000, 001 or 011.<br>000b = times 1 (44.1 or 48 kHz)<br>001b = times 2 (96kHz)<br>011b = times 4 (192kHz) |
| 10:8 | Sample Base Rate Divisor = 000. Divide by 1<br>000b = divide 1 (48 or 44.1 kHz) |
| 7 | Reserved. Read as zero |
| 6:4 | Bits per Sample = 001 or 011.<br>001 = 16 bits<br>011 = 24 bits |
| 3:0 | Number of Channels = 0 or 1.<br>0 = 1 channel (mono)<br>1 = 2 channels (stereo) |

**Table 23-233: Get/Set Amplifier Gain**

| Bit | Description |
|---|---|
| 7 | Amplifier Mute = 0 or 1 |
| 6:0 | Amplifier Gain index. 0 – 0x50 is valid, 0x51-0x7F = 0x00 |

**Table – 62: Audio Widget Capabilities Response**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Read as zero |
| 23:20 | Type = 1h. This is a ADC widget |
| 19:16 | Delay = 0h. Delay is described in the function group |
| 15:12 | Reserved. Read as zero |
| 11 | L-R Swap = 1. Left/right swap is supported. |
| 10 | Power Cntrl = 1. Power state control is available |
| 9 | Digital = 0. This widget is not digital |
| 8 | Conn List = 1. A connection list is available |
| 7 | Unsol Capable = 0. Jack sensing is not available |
| 6 | Proc Widget = 0. No processing capabilities are present |
| 5 | Stripe = 0. Striping is not supported |
| 4 | Format Override = 1. Format information is available |
| 3 | Amp Parameter Override = 1. Amp parameters are available |

| | |
|---|---|
| 2 | Out Amp Present = 0. No output amp is present |
| 1 | In Amp Present = 1. Input amp is present |
| 0 | Stereo = 1. This widget supports stereo |

**Table 23-234: Sample Size, Rate Caps Response**

| Bit | Description |
|---|---|
| 31:21 | Reserved. Read as zero |
| 20 | B32 = 0. 32-bit audio format is not supported |
| 19 | B24 = 1. 24-bit audio format is supported |
| 18 | B20 = 0. 20-bit audio format is not supported |
| 17 | B16 = 1. 16-bit audio format is supported |
| 16 | B8 = 0. 8-bit audio format is not supported |
| 15:12 | Reserved. Read as zero |
| 11 | R12 = 0. 384 kHz audio format is not supported |
| 10 | R11 = 1. 192 kHz audio format is supported |
| 9 | R10 = 0. 176.4 kHz audio format is not supported |
| 8 | R9 = 1. 96 kHz audio format is not supported |
| 7 | R8 = 0. 88.2 kHz audio format is not supported |
| 6 | R7 = 1. 48 kHz audio format is supported |
| 5 | R6 = 0. 44.1 kHz audio format is not supported |
| 4 | R5 = 0. 32 kHz audio format is not supported |
| 3 | R4 = 0. 22.05 kHz audio format is not supported |
| 2 | R3 = 0. 16 kHz audio format is not supported |
| 1 | R2 = 0. 11.025 kHz audio format is not supported |
| 0 | R1 = 0. 8.0 kHz audio format is not supported |

**Table 23-235: Stream Formats Response**

| Bit | Description |
|---|---|
| 31:3 | Reserved. Read as zero |
| 2 | AC3 = 0. AC3 format is not supported |
| 1 | Float32 = 0. 32-bit float format is not supported |
| 0 | PCM = 1. PCM audio format is supported |

**Table 23-236: Input Amp Capabilities Response**

| Bit | Description |
|---|---|
| 31 | Mute Capable = 1. Mute is supported |
| 30:23 | Reserved. Read as zero |
| 22:16 | Step Size = 03h. Step size is 1dB |
| 15 | Reserved. Read as zero |
| 14:8 | NumSteps = 50h. 80 steps |
| 7 | Reserved. Read as zero |
| 6:0 | Offset = 4Ah. Step 74 is 0dB |

**Table 23-237: Connection List Length Response**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | Long Form = 0. List is in short form |

| | |
|---|---|
| 6:0 | Connection List Length = 0x1 |

**Table 23-238: Supported Power States Response**

| Bit | Description |
|---|---|
| 31 | EPSS = 1. EPSS is supported |
| 30:5 | Reserved. Read as zero |
| 4 | D4Sup = 0. D4 is not supported |
| 3 | D3Sup = 1. D3 is supported |
| 2 | D2Sup = 1. D2 is supported |
| 1 | D1Sup = 1. D1 is supported |
| 0 | D0Sup = 1. D0 is supported |

**Table 23-239: Get/Set Connection Select Response**

| Bit | Description |
|---|---|
| 7:0 | 0b000 = 1E (NID24 – ADC3 connects to port F)<br>0b000 = 27 (NID25 – ADC4 connects to port K) |

**Table 23-240: Get Connection List Response index 0**

| Bit | Description |
|---|---|
| 31:0 | 0x0000001E Port F for ADC 3<br>0x00000027 Port K for ADC 4 |

**Table 23-241: Get Power State Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 10 | PS_SettingsReset. Codec reset has occurred. |
| 9:7 | Reserved. Read as zero |
| 6:4 | PS-Act. Actual power state. |
| 3 | Reserved. Read as zero |
| 2:0 | PS-Set. Requested power state |

**Table 23-242: Set Power State Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:4 | Reserved. Set to zero |
| 3 | Reserved. Set to zero |
| 2:0 | PS-Set. 000 = D0, 001 = D1, 010 = D2, 011 = D3 hot, 100 = D3 cold |

**Table 23-243: Get/Set Channel and Stream ID**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7:4 | Stream ID, 0 = disable stream 1 –> F = enable stream |
| 3:0 | Channel ID |

**Table 23-244: Get EAPD/BTL Response**

| Bit | Description |
|---|---|
| 31:3 | Reserved. Read as zero |

| | |
|---|---|
| 2 | Left/Right swap = 0 or 1. |
| 1 | EAPD = 0. |
| 0 | BTL = 0. |

## 23.27 Node 26, PORT J Widget

Port J is a digital audio interface output port that supports I2S, and TDM (PCM) modes. It is described as an analog pin for software compatibility and is typically connected to an internal speaker amplifier.

The following table describes a PIN that is a stereo line output.
This port is available on the CX84xx parts.

**Table 23-245: Node 1D responses**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Audio widget Pin | 0xF00 | 0x09 | 0x00400501 | | PIN – analog CX84xx |
| Get pin capabilities | 0xF00 | 0x0C | 0x00010010 | | Output, EAPD |
| Connection length | 0xF00 | 0x0E | 0x00000001 | | Connected to DAC 5 |
| Supported Power states | 0xF00 | 0x0F | 0x8000000F | | EPSS, D0, D1, D2 and D3 |
| Get Connection | 0xF01 | 0x00 | 0x0000000a | 0x00000000 (P,W) | DAC 5 selected |
| Set Connection | 0x701 | 0x0a | 0x00000000 | | 0 = DAC 5 |
| Get connection list | 0xF02 | 0x00 | 0x00000023 | | DAC 5 |
| Get power state | 0xF05 | 0x00 | 0x00000abc | 0x00000433 (P,W) | a = Settings Reset<br>b = Actual state<br>c = Requested state<br>Settings Reset cleared by this verb or any write to this node. |
| Set power state | 0x705 | 0x0a | 0x00000000 | | a = Requested state |
| Get pin control | 0xF07 | 0x00 | 0x000000a0 | 0x00000040 (P,W) | a = Vref, input enable and output enable |
| Set pin control | 0x707 | 0xa0 | 0x00000000 | | a = Vref, input enable and output enable |
| Get unsolicited response | 0xF08 | 0x00 | 0x000000aa | 0x00000000 (P,W) | aa = Unsolicited enable and tag |
| Set unsolicited response | 0x708 | 0xaa | 0x00000000 | | aa = Unsolicited enable and tag |
| Get pin sense | 0xF09 | 0x00 | 0xa0000000 | | a = Presence detect<br>8 = present, 0 = missing |
| Get EAPD | 0xF0C | 0x00 | 0x0000000a | 0x00000002 (P,W) | a = EAPD |
| Set EAPD | 0x70C | 0x0a | 0x00000000 | | a = EAPD |
| Get default config | 0xF1C-0xF1F | 0x00 | 0xaabbccdd | 0x040410F0 (P) | aa = Config4<br>bb = Config3<br>cc = Config2 |

| | | | | | |
|---|---|---|---|---|---|
| | | | | | dd = Config1 |
| Set default config 1 | 0x71C | 0xaa | 0x00000000 | | aa = Config1 |
| Set default config 2 | 0x71D | 0xaa | 0x00000000 | | aa = Config2 |
| Set default config 3 | 0x71E | 0xaa | 0x00000000 | | aa = Config3 |
| Set default config 4 | 0x71F | 0xaa | 0x00000000 | | aa = Config4 |

The responses are decoded as follows.

**Table 23-246: Audio Widget Capabilities Response**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Read as zero |
| 23:20 | Type = 4h. This is a pin widget |
| 19:16 | Delay = 0h. Delay is described in the function group |
| 15:12 | Reserved. Read as zero |
| 11 | L-R Swap = 0. Left/right swap is not supported. |
| 10 | Power Cntrl = 1. Power state control is available when a pin complex. |
| 9 | Digital = 0. This widget is not digital |
| 8 | Conn List = 1. A connection list is available |
| 7 | Unsol Capable = 0. Jack sensing is not available |
| 6 | Proc Widget = 0. No processing capabilities are present |
| 5 | Stripe = 0. Striping is not supported |
| 4 | Format Override = 0. No format information is available |
| 3 | Amp Parameter Override = 0. No amp parameters are available |
| 2 | Out Amp Present = 0. No output amp is present |
| 1 | In Amp Present = 0. No input amp is present |
| 0 | Stereo = 1. This widget supports stereo |

**Table 23-247: Pin Capabilities Response**

| Bit | Description |
|---|---|
| 31:17 | Reserved. Read as zero |
| 16 | EAPD capable = 1. EAPD control is supported |
| 15:14 | Reserved. Read as zero |
| 13 | 100% = 0. Vref is not supported by this PIN |
| 12 | 80% = 0. Vref is not supported by this PIN |
| 11 | Reserved. Read as zero |
| 10 | Ground = 0. Vref is not supported by this PIN |
| 9 | 50% = 0. Vref is not supported by this PIN |
| 8 | Hi-Z = 0. Vref is not supported by this PIN |
| 7 | Reserved. Read as zero |
| 6 | Balanced I/O Pins = 0. Balanced I/O is not supported |
| 5 | Input Capable = 0. Input is supported |
| 4 | Output Capable = 1. Output is supported |

| | |
|---|---|
| 3 | Headphone Drive Capable = 0 Headphone drive is not supported. |
| 2 | Presence Detect Capable = 0. Jack sensing is not supported |
| 1 | Trigger Req'd = 0. No trigger is required for impedance sensing |
| 0 | Impedance Sense Capable = 0. Impedance sensing is not supported |

**Table 23-248: Connection List Length Response**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | Long Form = 0. List is in short form |
| 6:0 | Connection List Length = 01h. There is one connection |

**Table 23-249: Supported Power States Response**

| Bit | Description |
|---|---|
| 31 | EPSS = 1. EPSS is supported |
| 30:5 | Reserved. Read as zero |
| 4 | D4Sup = 0. D4 is not supported |
| 3 | D3Sup = 1. D3 is supported |
| 2 | D2Sup = 1. D2 is supported |
| 1 | D1Sup = 1. D1 is supported |
| 0 | D0Sup = 1. D0 is supported |

**Table 23-250: Get/Set Pin Connection**

| Bit | Description |
|---|---|
| 7:0 | 0x00 Connection select, 0x00 = DAC5 |

**Table 23-251: Get Connection List Response**

| Bit | Description |
|---|---|
| 31:0 | 0x00000023 DAC5 |

**Table 23-252: Get Power State Response**

| Bit | Description |
|---|---|
| 31:11 | Reserved. Read as zero |
| 10 | PS_SettingsReset. Codec reset has occurred. |
| 9:7 | Reserved. Read as zero |
| 6:4 | PS-Act. Actual power state. |
| 3 | Reserved. Read as zero |
| 2:0 | PS-Set. Requested power state |

**Table 23-253: Set Power State Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:4 | Reserved. Set to zero |
| 3 | Reserved. Set to zero |
| 2:0 | PS-Set. 000 = D0, 001 = D1, 010 = D2, 011 = D3 hot, 100 = D3 cold |

**Table 23-254: Get/Set Pin Widget Control**

| Bit | Description |
|---|---|

| | |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | H-Phn Enable = 0. |
| 6 | Out Enable = 0 or 1. |
| 5 | In Enable = 0. |
| 4:3 | Reserved. Read as zero |
| 2:0 | VrefEn = 0 (Disabled), 1 (50%), or 4 (80%) |

**Table 23-255: Get Default Config Response**

| Bit | Description |
|---|---|
| 31:24 | Value stored in Config byte 4 |
| 23:16 | Value stored in Config byte 3 |
| 15:8 | Value stored in Config byte 2 |
| 7:0 | Value stored in Config byte 1 |

**Table 23-256: Set Default Config Data Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:0 | Value to store in Config byte 1, 2, 3, or 4 |

## 23.28 Node 27, PORT K Widget

Port K is a digital audio interface input port that supports I2S, and TDM (PCM) modes. It is described as an analog pin for software compatibility and is typically connected to an internal speaker amplifier for AEC.

The following table describes a PIN that is a stereo line or mic input and supports jack sensing.
This port is available on the CX84xx parts.

**Table 23-257: Node 1E response**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Get amp gain | 0xB00<br>0xB20 | 0x00 | 0x000000aa<br>0x000000aa | 0x00000000<br>(P,W) | aa = Right gain<br>aa = Left gain |
| Set amp gain | 0x350<br>0x360<br>0x370 | 0xaa | 0x00000000 | | aa = Right gain<br>aa = Left gain<br>aa = Left & Right gain |
| Audio widget Pin | 0xF00 | 0x09 | 0x0040040B | | Pin – analog CX84xx |
| Get pin capabilities | 0xF00 | 0x0C | 0x00000020 | | Input |
| Input amp capabilities | 0xF00 | 0x0D | 0x002F0300 | | 12db step, 4 steps, step 0 is 0db |
| Supported Power states | 0xF00 | 0x0F | 0x8000000F | | EPSS, D0, D1, D2 and D3 |
| Get power state | 0xF05 | 0x00 | 0x00000abc | 0x00000433<br>(P,W) | a = Settings Reset<br>b = Actual state<br>c = Requested state<br>Settings Reset cleared by this verb or any write to this node. |
| Set power state | 0x705 | 0x0a | 0x00000000 | | a = Requested state |
| Get pin control | 0xF07 | 0x00 | 0x000000aa | 0x00000000<br>(P,W) | aa = input enable |
| Set pin control | 0x707 | 0xaa | 0x00000000 | | aa = input enable |
| Get unsolicited response | 0xF08 | 0x00 | 0xaa | 0x00000000<br>(P,W) | aa = Unsolicited enable and tag |
| Set unsolicited response | 0x708 | 0xaa | 0x00000000 | | aa = Unsolicited enable and tag |
| Get pin sense | 0xF09 | 0x00 | 0xa0000000 | | a = Presence detect<br>8 = present, 0 = missing |
| Get default config | 0xF1C-<br>0xF1F | 0x00 | 0xaabbccdd | 0x0484E0F0<br>(P) | aa = Config4<br>bb = Config3<br>cc = Config2<br>dd = Config1 |
| Set default config 1 | 0x71C | 0xaa | 0x00000000 | | aa = Config1 |

| | | | | | |
|---|---|---|---|---|---|
| Set default config 2 | 0x71D | 0xaa | 0x00000000 | | aa = Config2 |
| Set default config 3 | 0x71E | 0xaa | 0x00000000 | | aa = Config3 |
| Set default config 4 | 0x71F | 0xaa | 0x00000000 | | aa = Config4 |

The responses are decoded as follows.

**Table 23-258: Get/Set Amplifier Gain**

| Bit | Description |
|---|---|
| 7 | Amplifier Mute = 0 |
| 6:3 | Read as 0 |
| 2:0 | Amplifier Gain index. 0=0db, 1=12db, 2=24db, 3=36db |

**Table 23-259: Audio Widget Capabilities Response**

| Bit | Description |
|---|---|
| 31:24 | Reserved. Read as zero |
| 23:20 | Type = 4h. This is a PIN widget |
| 19:16 | Delay = 0h. Delay is described in the function group |
| 15:12 | Reserved. Read as zero |
| 11 | L-R Swap = 0. Left/right swap is not supported. |
| 10 | Power Cntrl = 1. Power state control is available. |
| 9 | Digital = 0. This widget is not digital |
| 8 | Conn List = 0. Connection list is available |
| 7 | Unsol Capable = 0. Jack sensing is not available |
| 6 | Proc Widget = 0. No processing capabilities are present |
| 5 | Stripe = 0. Striping is not supported |
| 4 | Format Override = 0. No format information is available |
| 3 | Amp Parameter Override = 1. Amp parameters are available |
| 2 | Out Amp Present = 0. No output amp is present |
| 1 | In Amp Present = 1. Input amp is present |
| 0 | Stereo = 1. This widget supports Stereo |

**Table 23-260: Pin Capabilities Response**

| Bit | Description |
|---|---|
| 31:17 | Reserved. Read as zero |
| 16 | EAPD capable = 0. EAPD control is not supported |
| 15:14 | Reserved. Read as zero |
| 13 | 100% = 0. Vref is not supported by this PIN |
| 12 | 80% = 0. Vref is not supported by this PIN |
| 11 | Reserved. Read as zero |
| 10 | Ground = 0. Vref is not supported by this PIN |
| 9 | 50% = 0. Vref is not supported by this PIN |
| 8 | Hi-Z = 0. Vref is not supported by this PIN |
| 7 | Reserved. Read as zero |
| 6 | Balanced I/O Pins = 0. Balanced I/O is not supported |

| | |
|---|---|
| 5 | Input Capable = 1. Input is supported |
| 4 | Output Capable = 0. Output is not supported |
| 3 | Headphone Drive Capable = 0. Headphone drive is not supported |
| 2 | Presence Detect Capable = 0. Jack sensing is supported |
| 1 | Trigger Req'd = 0. No trigger is required for impedance sensing |
| 0 | Impedance Sense Capable = 0. Impedance sensing is not supported |

**Table 23-261: Input Amp Capabilities Response**

| Bit | Description |
|---|---|
| 31 | Mute Capable = 0. Mute is not supported |
| 30:23 | Reserved. Read as zero |
| 22:16 | Step Size = 2Fh. Step size is 12dB |
| 15 | Reserved. Read as zero |
| 14:8 | NumSteps = 3h. 4 steps |
| 7 | Reserved. Read as zero |
| 6:0 | Offset = 0h. Step 0 is 0dB |

**Table 23-262: Supported Power States Response**

| Bit | Description |
|---|---|
| 31 | EPSS = 1. EPSS is supported |
| 30:5 | Reserved. Read as zero |
| 4 | D4Sup = 0. D4 is not supported |
| 3 | D3Sup = 1. D3 is supported |
| 2 | D2Sup = 1. D2 is supported |
| 1 | D1Sup = 1. D1 is supported |
| 0 | D0Sup = 1. D0 is supported |

**Table 23-263: Get Power State Response**

| Bit | Description |
|---|---|
| 31:11 | Reserved. Read as zero |
| 10 | PS_SettingsReset. Codec reset has occurred. |
| 9:7 | Reserved. Read as zero |
| 6:4 | PS-Act. Actual power state. |
| 3 | Reserved. Read as zero |
| 2:0 | PS-Set. Requested power state |

**Table 23-264: Set Power State Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:4 | Reserved. Set to zero |
| 3:2 | Reserved. Set to zero |
| 1:0 | PS-Set. 000 = D0, 001 = D1, 010 = D2, 011 = D3 hot, 100 = D3 cold |

**Table 23-265: Get/Set Pin Widget Control**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Read as zero |
| 7 | H-Phn Enable = 0. |

| | |
|---|---|
| 6 | Out Enable = 0. |
| 5 | In Enable = 0 or 1. |
| 4:3 | Reserved. Read as zero |
| 2:0 | VrefEn = 0. |

**Table 23-266: Get Default Config Response**

| Bit | Description |
|---|---|
| 31:24 | Value stored in Config byte 4 |
| 23:16 | Value stored in Config byte 3 |
| 15:8 | Value stored in Config byte 2 |
| 7:0 | Value stored in Config byte 1 |

**Table 23-267: Set Default Config Data Format**

| Bit | Description |
|---|---|
| 31:8 | Reserved. Set to zero |
| 7:0 | Value to store in Config byte 1, 2, 3, or 4 based on verb address |

## 23.29 Node 28, Vendor Specific Widget

This widget is used to access the I2C Master Write Sequencer memory block
Please refer to the I2C Master section for more information.
RAM contents are written 1 byte at a time. RAM contents are read 4 bytes at a time with every byte address in a 32-bit word returning the same 32-bit value. For example, if RAM addresses 00 = 0x11, 01 =0x22, 02 = 0x33, and 03 = 0x44; reading RAM using verbs A0000, A0100, A0200, and A0300 all return the same 32-bit word 0x44332211.

The RAM block is addressed through the I2C slave interface at addresses 0xA000-0xA1FF

**Table 23-268: Node 28 responses**

| Description | Verb | Parameter | Response | Default | Comments |
|---|---|---|---|---|---|
| Get RAM 000-0FF | 0xA | 0xrr00 | 0xaabbccdd | 0x00000000 (P) | rr = RAM offset<br>aa = byte offset 3<br>bb = byte offset 2<br>cc = byte offset 1<br>dd = byte offset 0 |
| Set RAM 000-0FF | 0x2 | 0xaabb | 0x00000000 | | aa = register number<br>bb = 8-bit value |
| Get RAM 100-1FF | 0xB | 0xrr00 | 0xaabbccdd | 0x00000000 (P) | rr = RAM offset<br>aa = byte offset 3<br>bb = byte offset 2<br>cc = byte offset 1<br>dd = byte offset 0 |
| Set RAM 100-1FF | 0x3 | 0xaabb | 0x00000000 | | aa = register number<br>bb = 8-bit value |
| Audio widget Vendor | F00 | 0x09 | 0x00F00000 | | |

## 23.30 Audio Unsolicited Messages:

The following table describes all possible bit coding and sources of unsolicited messages for the audio function group. The “Tag” comes from the unsolicited message tag stored in the node that generated the event. When the bus is inactive, a wake event will be generated and the unsolicited response will be sent after the bus becomes active per the Intel HD Audio 1.0a specification.

When the host interface is configured for I2S/I2C, it is not possible to send a message across the I2C slave interface without a request by the host. Therefore, a GPIO pin may be configured as an interrupt output. The behavior is similar to a wake event in HD Audio mode where an event causes the INT pin to change state (programmable active high or active low). The host will have to query the CODEC using the I2C interface to determine the source of the interrupt and clear the interrupt.

| Node ID | Description |
|---|---|
| Node 1 | GPIO state change |
| Node 16 | Port A jack change (Sense Pin) |
| Node 18 | Port B jack change (Sense Pin) |
| Node 1A | Port C jack change (Sense Pin) |
| Node 19 | Port D jack change (Headset logic) |
| Node 1D | Port E jack change (Sense Pin - GPIO) |
| Node 1E | Port F jack change (Sense Pin - GPIO) |
| | Headset button events |

# 24Vender Defined Registers

## 24.1 Oceanside (A0) Register Change List from Deep Eddy (A1)

**Red** Shows all registers removed from Deep Eddy and are in Neptune B0
**Purple** Shows Registers changed in Deep Eddy from Neptune B0.
**Green** shows registers/bits that are changed in Deepeddy-A1 and were in DeepEddy-A0
**Blue** shows registers/bits that are changed in Oceanside from Deep Eddy A1

## 24.2 Vendor 1B Registers

### 24.2.1 SPKR DRC and EQ registers

#### 24.2.1.1 EQ Enable and Bypass LSB register

Get Response - HD_A[27:0] = 0x1BA0000 Set - HD_A[27:8] = 0x1B200 I2C = 6D00

| Bit | Default | Description |
|---|---|---|
| 7:4 | 0000 | Lock sequence<br>To lock EQ/DRC write a 0xA followed by 0x4<br>To unlock EQ/DRC write a 0x2 followed by 0xC<br>When the lock bit is set the enable/disable bits of EQ/DRC blocks are read only. The writing of EQ/DRC registers is blocked. |
| 3 | 1 | EQ ch1 bypass bit.<br>1: EQ ch1 is bypass mode.<br>0: EQ ch1 is normal mode |
| 2 | 1 | EQ ch0 bypass bit.<br>1: EQ ch0 is bypass mode.<br>0: EQ ch0 is normal mode |
| 1 | 0 | EQ ch1 enable bit.<br>0: EQ ch1 is reset, EQ ch1 clock is disabled.<br>1: EQ ch1 is enabled when class-D is enabled or when EQ is forced on. |
| 0 | 0 | EQ ch0 enable bit.<br>0: EQ ch0 is reset, EQ ch0 clock is disabled.<br>1: EQ ch0 is enabled when class-D is enabled or when EQ is forced on.<br><br>Set bit 0 and 1 to 1 and then start loading values into BiQuads. |

#### 24.2.1.2 EQ Enable and Bypass MSB register

Get Response - HD_A[27:0] = 0x1BA0100 Set - HD_A[27:8] = 0x1B201 I2C = 6D01

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Force coefficient read mode<br>0: no force<br>1: force to coefficient read mode |
| 6 | 1 | Reserved |
| 5 | 1 | Reserved |
| 4 | 1 | Clear 7B EQ memory when disable/enable<br>1: Clear memory enable<br>0: Clear memory disable |
| 3 | 0 | Reserved |

| | | |
|---|---|---|
| 2 | 0 | Reserved |
| 1 | 1 | Reserved |
| 0 | 0 | Reserved |

### 24.2.1.3 EQ B0 coefficient LSB register

Get Response - HD_A[27:0] = 0x1BA0200 Set - HD_A[27:8] = 0x1B202 I2C = 6D02

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | EQ B0 coefficient LSB [7:0] is loaded. |

### 24.2.1.4 EQ B0 coefficient MSB register

Get Response - HD_A[27:0] = 0x1BA0300 Set - HD_A[27:8] = 0x1B203 I2C = 6D03

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | EQ B0 coefficient MSB [15:8] is loaded. |

### 24.2.1.5 EQ B1 coefficient LSB register

Get Response - HD_A[27:0] = 0x1BA0400 Set EQ - HD_A[27:8] = 0x1B204 I2C = 6D04

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | EQ B1 coefficient LSB [7:0] is loaded. |

### 24.2.1.6 EQ B1 coefficient MSB register

Get Response - HD_A[27:0] = 0x1BA0500 Set - HD_A[27:8] = 0x1B205 I2C = 6D05

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | EQ B1 coefficient MSB [15:8] is loaded. |

### 24.2.1.7 EQ B2 coefficient LSB register

Get Response - HD_A[27:0] = 0x1BA0600 Set - HD_A[27:8] = 0x1B206 I2C = 6D06

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | EQ B2 coefficient LSB [7:0] is loaded. |

### 24.2.1.8 EQ B2 coefficient MSB register

Get Response - HD_A[27:0] = 0x1BA0700 Set - HD_A[27:8] = 0x1B207 I2C = 6D07

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | EQ B2 coefficient MSB [15:8] is loaded. |

### 24.2.1.9 EQ A1 coefficient LSB register

Get Response - HD_A[27:0] = 0x1BA0800 Set - HD_A[27:8] = 0x1B208 I2C = 6D08

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | EQ A1coeff 's LSB [7:0] is loaded. |

### 24.2.1.10 EQ A1 coefficient MSB register

Get Response - HD_A[27:0] = 0x1BA0900 Set - HD_A[27:8] = 0x1B209 I2C = 6D09

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | EQ A1 coefficient MSB [15:8] is loaded. |

### 24.2.1.11 EQ A2 coefficient LSB register

Get Response - HD_A[27:0] = 0x1BA0A00 Set - HD_A[27:8] = 0x1B20A I2C = 6D0A

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | EQ A2 coefficient LSB [7:0] is loaded. |

### 24.2.1.12 EQ A2 coefficient MSB register

Get Response - HD_A[27:0] = 0x1BA0B00 Set - HD_A[27:8] = 0x1B20B I2C = 6D0B

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | EQ A2 coefficient MSB [15:8] is loaded. |

### 24.2.1.13 EQ G coefficient MSB register

Get Response - HD_A[27:0] = 0x1BA0C00 Set - HD_A[27:8] = 0x1B20C I2C = 6D0C

| Bit | Default | Description |
|---|---|---|
| 7:3 | 00000 | Reserved – Read as 0 |
| 2:0 | 0 | EQ bi quad stage gain G is loaded. |

### 24.2.1.14 EQ Band Control register

Get Response - HD_A[27:0] = 0x1BA0D00 Set - HD_A[27:12] = 0x1B20D I2C = 6D0D

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Read only coeff_done |
| 6 | 0 | Coefficient latch enable used to start the write/read process. Automatically returned to 0 once the operation is complete.<br>In write mode, when set to 1, the current values of B0, B1, B2, A1, A2, G, P1, P2 are loaded into EQ Memory based on bits [5:3] (channel number) and IIR band location based on [2:0].<br><br>In read mode, when set to 1, coefficients will be read out from EQ memory based on bits [5:3] (channel number) and IIR band location based on [2:0]. Host can read B0, B1, B2, A1, A2, G, P1, P2 after process complete (within one frame). |
| 5:3 | 0 | EQ Channel selection bits for loading B0, B1, B2, A1, A2, G and P1, P2<br>0 – Ch0 coeff's are loaded into 7B EQ memory<br>1 – Ch1 coeff's are loaded into 7B EQ memory<br>2 to 7 - Reserved |
| 2:0 | 0 | EQ IIR Band selection bits for loading B0, B1, B2, A1, A2 and P1, P2.<br>0 – Band 0 is loaded with programmed B0, B1, …P1, P2 values.<br>1 – Band 1 is loaded with programmed B0, B1, …P1, P2 values.<br>2 – Band 2 is loaded with programmed B0, B1, …P1, P2 values.<br>3 – Band 3 is loaded with programmed B0, B1, …P1, P2 values.<br>4 – Band 4 is loaded with programmed B0, B1, …P1, P2 values.<br>5 – Band 5 is loaded with programmed B0, B1, …P1, P2 values<br>6 – Band 6 is loaded with programmed B0, B1, …P1, P2 values |

### 24.2.1.15 EQ Bank Control register

Get Response - HD_A[27:0] = 0x1BA0E00 Set - HD_A[27:12] = 0x1B20E I2C = 6D0E

| Bit | Default | Description |
|---|---|---|
| 7:2 | 0 | Reserved |
| 1:0 | 0 | 7B EQ bank selection when read/write coeff.<br>00: default for 48K<br>01: default for 96K<br>11: default for 192K<br>10: reserved |

### 24.2.1.16 Unlock Count register

Get Unlock Count - HD_A[27:0] = 0x1BA0F00 I2C = 6D0F

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Unlock count register:<br>Count the unlock. Each unlock, plus 1. Read only. |

### 24.2.1.17 SPKR DRC Enable Step register

Get Response - HD_A[27:0] = 0x1BA1000 Set - HD_A[27:8] = 0x1B210 I2C = 6D10

| Bit | Default | Description |
|---|---|---|
| 7:4 | 0xA | SPKR_DRC_RELEASE_UPDATE_STEP, used to program the Gain step required to release the gain slowly once the signal is in linear region |
| 3:1 | 0x2 | SPKR_DRC_ATTACK_UPDATE_STEP, used to program the Gain step required to compress the incoming signal in compression region. |
| 0 | 0 | SPKR_DRC_EN, this bit will enable spkr drc block. |

### 24.2.1.18 SPKR DRC FAST RELEASE DELAY & UPDATE Enable Step register

Get Response - HD_A[27:0] = 0x1bA1100 Set - HD_A[27:8] = 0x1b211 I2C = 6D11

| Bit | Default | Description |
|---|---|---|
| 7:4 | 0x6 | SPKR_DRC_FAST_RELEASE_DELAY_LOG2, used to program the release delay value. This will be used to update the release delay value to make quick transition in the gain value. Internally to SPKR DRC block the FAST_RELEASE_DELAY = 2^SPKR_DRC_FAST_RELEASE_DELAY_LOG2. |
| 3:0 | 0x5 | SPKR_DRC_FAST_UPDATE_SHIFT, used to program the step value. This will be used to quickly update the MAXABS IN value. |

### 24.2.1.19 SPKR DRC Boost/Max Lineout register

Get Response - HD_A[27:0] = 0x1bA12000 Set - HD_A[27:8] = 0x1b212 I2C = 6D12

| Bit | Default | Description |
|---|---|---|
| 7:6 | 0 | Reserved |
| 5:0 | 0 | SPKR_DRC_BOOST This is the Gain value initially set for DRC.<br>The 6-bits are split internally to SPKR DRC block as<br>DRC_BOOST_MANT[3:0] = SPKR_DRC_BOOST[3:0]<br>DRC_BOOST_EXP[1:0] = SPKR_DRC_BOOST[5:4]<br>The Final BOOST is as follows<br>BOOST[15:10+EXP] = 0;<br>BOOST[9+EXP:6+EXP] = MANT<br>BOOST[5+EXP:0] = 0 |

### 24.2.1.20 SPKR DRC BALANCE RAMP STEP register

Get Response - HD_A[27:0] = 0x1bA1300 Set - HD_A[27:8] = 0x1b213 I2C = 6D13

| Bit | Default | Description |
|---|---|---|

| | | |
|---|---|---|
| 7 | 0 | Reserved |
| 6:0 | 0x4 | SPKR_DRC_BALANCE_RAMP_STEP, this value determines Balance RAMP STEP size.<br>The 6-bits are split internally to SPKR DRC block as<br>DRC_BLNCE_RAMP_MANT[3:0] = SPKR_DRC_BALANCE_RAMP_STEP[3:0]<br>DRC_BALANCE_RAMP_EXP[2:0] = SPKR_DRC_BALANCE_RAMP_STEP[6:4]<br>The Final BALANCE RAMP STEP is as follows<br>BALANCE_RAMP_STEP[15:7+EXP] = 0;<br>BALANCE_RAMP_STEP[6+EXP:3+EXP] = MANT<br>BALANCE_RAMP_STEP[2+EXP:0] = 0 |

### 24.2.1.21 SPKR DRC VOLUME RAMP STEP & GAIN STEP register

Get Response - HD_A[27:0] = 0x1bA1400 Set - HD_A[27:12] = 0x1b214 I2C = 6D14

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Reserved |
| 6:4 | 0x2 | SPKR_DRC_VOLUME_RAMP_STEP_SHIFT_SELECT, Selects the Step and Shift size for Volume Ramping.<br>Internally calculations are<br>VOLUME_RAMP_STEP = 2*(2^ SPKR_DRC_VOLUME_RAMP_STEP_SHIFT_SELECT);<br>VOLUME_RAMP_MAX_SHIFT = 10 -<br>SPKR_DRC_VOLUME_RAMP_STx4EP_SHIFT_SELECT; |
| 3 | 0 | 0: Disable SPKR_DRC_GAIN_SHIFT<br>1: Enable SPKR_DRC_GAIN_SHIFT |
| 2:0 | 0x4 | SPKR_DRC_GAIN_SHIFT, Gain shift value.<br>0: +24db<br>1: +18db<br>2: +12db<br>3: +6db<br>4: 0db<br>5: -6db<br>6: -12db<br>7: -18db |

### 24.2.1.22 SPKR DRC RELEASE RATE register

Get Response - HD_A[27:0] =0x1bA1500 Set - HD_A[27:8] = 0x1b215 I2C = 6D15

| Bit | Default | Description |
|---|---|---|
| 7:6 | 0 | Reserved |
| 5:0 | 0 | SPKR_DRCRELEASE_RATE_TH, this is the threshold for deciding whether to use slow release (with “RELEASE_UPDATE_STEP”) or fast release (with “TRANSIENT_UPDATE_STEP”)<br>The 6-bits are split internally to SPKR DRC block as<br>DRC_RELEASE_RATE_TH_MANT[3:0] = SPKR_DRC_RELEASE_RATE_TH[3:0]<br>DRC_MAX_RELEASE_RATE_TH_EXP[1:0] = SPKR_DRC_RELEASE_RATE_TH[5:4]<br>The Final RELEASE_RATE_TH is as follows<br>RELEASE_RATE_TH[15:10+EXP] = sign extend MSB of MANT;<br>RELEASE_RATE_TH[9+EXP:6+EXP] = MANT<br>RELEASE_RATE_TH[5+EXP:0] = 0 |

### 24.2.1.23 SPKR DRC Release Delay/programmable slope register

Get Response - HD_A[27:0] = 0x1bA1600 Set - HD_A[27:8] = 0x1b216 I2C = 6D16

| Bit | Default | Description |
|---|---|---|
| 7 | 0x0 | Reserved |

| | | |
|---|---|---|
| 6:4 | 0x7 | COMP_SLP_HI_REGION, with values 0-6 the compression is 2^register_value (or: the slope is 1/2^register_value, i.e. 1 to 1/64); With value 7 the compression is infinite (same as in Palm, i.e. slope 0) |
| 3:0 | 0xB | SPKR_DRC_RELEASE_DELAY_LOG2, In unit of 1 mSec. Used to program the period of time passed before MaxAbsIn can be decreased to a lower value if the input level is not increasing. Hysteresis for adjusting the gain upward. Internally the 4-bits are used as RELEASE_DELAY = 2^ SPKR_DRC_RELEASE_DELAY_LOG2. |

### 24.2.1.24 SPKR DRC Max Lineout register

Get Response - HD_A[27:0] = 0x1BA1700 Set - HD_A[27:8] = 0x1B217 I2C = 6D17

| Bit | Default | Description |
|---|---|---|
| 7:6 | | Reserved |
| 5:0 | 0 | SPKR_DRC_ MAX_LIN_OUT, this used to set Threshold and Limit values.<br>The 6-bits are split internally to SPKR DRC block as<br>DRC_MAX_LIN_OUT_MANT[3:0] = SPKR_DRC_MAX_LIN_OUT[3:0]<br>DRC_MAX_LIN_OUT_EXP[1:0] = SPKR_DRC_MAX_LIN_OUT[5:4]<br>The Final MAX_LIN_OUT is as follows<br>MAX_LIN_OUT[15:10+EXP] = sign extend MSB of MANT;<br>MAX_LIN_OUT[9+EXP:6+EXP] = MANT<br>MAX_LIN_OUT[5+EXP:0] = 0 |

### 24.2.1.25 SPKR DRC Test register

Get Response - HD_A[27:0] = 0x1BA1800 Set - HD_A[27:8] = 0x1B218 I2C = 6D18

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Reserved |
| 6 | 0 | Selection of DAC1 and DAC2 manually by overriding the Automatic selection of DAC to which Class-D was assigned. |
| 5 | 0 | Selection of EQ on to desired DAC.<br>0 – DAC0 will have EQ and SPKR DRC.<br>1 –DAC1 will have EQ and SPKR DRC. |
| 4 | 0 | Manual Mode enabled and DAC selection when Class-D is not enabled or above bit-6 is not set. |
| 3 | 0 | Reserved |
| 2 | 0 | This bit enables the Ch2 of EQ Block when below bit 0 is enabled.<br>0 – disables EQ when below bit 0 is se t to 1.<br>1 – enables EQ when below bit 0 is set to 0. |
| 1 | 0 | This bit enables the Ch1 of EQ Block when below bit 0 is enabled.<br>0 – disables EQ when below bit 0 is se t to 1.<br>1 – enables EQ when below bit 0 is set to 0. |
| 0 | 0 | This bit disables the EQ block working on SPKR enabled DAC.<br>0 – EQ block is enabled on the DAC for which SPKR block is assigned and eq_enable bits in test register 0x1B20 are set.<br>1 -- EQ block is disabled to work automatically on the DAC for which SPKR is assigned. The EQ will work when above bits 1 and 2 are set and along with eq_enable bitsin test register 1B20. |

### 24.2.1.26 SPKR DRC Gain Read LSB register

Get Response - HD_A[27:0] = 0x1BA1900 I2C = 6D19

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Status of DRC Gain value applied (read only) |

### 24.2.1.27 SPKR DRC Gain Read MSB register

Get Response - HD_A[27:0] = 0x1BA1A00 I2C = 6D1A

| Bit | Default | Description |
|---|---|---|

| | | |
|---|---|---|
| 7:0 | 0 | Status of DRC Gain value applied (read only) |

## 24.2.2 EQ Lock registers

### 24.2.2.1 One-Way Lock1 register

Set One-Way Lock1 - HD_A[27:12] = 0x1B22E I2C = 6D2E

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | One-Way lock1:<br>Passcode "'hA3"<br>Once enable one-way lock, if locked, only POR can unlock. |

### 24.2.2.2 One-Way Lock2 register

Set One-Way Lock2 - HD_A[27:12] = 0x1B22F I2C = 6D2F

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | One-Way lock2:<br>Passcode "'h69"<br>Once enable one-way lock, if locked, only POR can unlock. |

## 24.2.3 Digital Bios Test Registers

### 24.2.3.1 Digital Bios Test0 LSB register

Get Response - HD_A [27:0] = 0x1BB0000 Set - HD_A[27:8] = 0x1B300 I2C = 6D80

| Bit | Default | Description |
|---|---|---|
| 7 | 1 | 0 – Normal mode<br>1 – Enable gating of Azalia clock to state machine when we are in SnM |
| 6 | 0 | 0 = spdif clk enabled in HD Audio Bus mode if audio link not down???<br>1 = spdif clk disabled in HD Audio Bus mode |
| 5 | 0 | 0 – No Polarity inversion of EAPD control on GPIO5 pin<br>1 – Polarity Inversion on EAPD control on GPIO5 pin |
| 4 | 0 | 1 = when bclk gone, switch lp_rcosc_clk to fifo clock. & switch start_frame of codec to 48k signal generated by clk_au.<br>0 = when bclk gone, fifo clock & start_frame gone. |
| 3:2 | 2'b10 | GPIO5 DRVCTL |
| 1:0 | 2'b10 | GPIO4 DRVCTL |

### 24.2.3.2 Digital Bios Test0 MSB register

Get Response - HD_A [27:0] = 0x1BB0100 Set - HD_A[27:8] = 0x1B301 I2C = 6D81

| Bit | Default | Description |
|---|---|---|
| 7:6 | 0 | Reserved |
| 5 | 0 | 1 – spkr_prot_clk on 24Mhz<br>0 - spkr_prot_clk on 6M |
| 4 | 1 | 1: Low_power_mode_en is function of low_pwr_mode signal only |
| 3 | 0 | 1 – Bypass automatic clock gating of clk_ctrl_sm(ie mux of rcosc and bclk (based on D3)) for jack sensing and wake on jack feature, unsolicited message HP charge pump clocking and input analog control logic(described below for the 9th bit)). If set to 1 then value on bit 10 determines the clock gating |

| | | |
|---|---|---|
| | | 0 – Automatic clock gating of clk_ctrl_sm(ie mux of rcosc and bclk (based on D3)) for jack sensing and wake on jack feature, unsolicited message HP charge pump clocking and input analog control logic) . Also, Refer clocking diagram |
| 2 | 0 | 1 – When Bypass bit (bit 11=1), then setting this bit to 1 enables clock to jack sensing and wake on jack feature, unsolicited message HP charge pump clocking and input analog control logic (described below for the 9th bit)<br>0 - When Bypass bit (bit 11=1), then setting this to 0 disables clock to jack sensing and wake on jack feature, unsolicited message HP charge pump clocking and input analog control logic If bit 9 of this register is zero then this bit doesn't have any function Also Refer clocking diagram |
| 1 | 0 | 1 – Bypass automatic clock gating of clk_24(i.e. mux of rcosc and bclk (based on D3) or PLL) for analog control logic (port controller, class D clocking, class D/spkr protection logic). If set to 1 then value on bit 8 determines the clock gating<br>0 – Automatic clock gating of clk_24(i.e. mux of rcosc and bclk based on D3 or PLL) for analog control logic, port controller, class D clocking, class D/spkr protection logic). Also, Refer clocking diagram |
| 0 | 0 | 1 – When Bypass bit (bit 9=1), then setting this bit to 1 enables clock to analog digital interface circuit<br>0 - When Bypass bit (bit 9=1), then setting this to 0 disables clock to analog digital interface circuit<br>If bit 9 of this register is zero, then this bit doesn't have any function. Also, Refer clocking diagram |

### 24.2.3.3 EAPD Control LSB register

Get Response - HD_A[27:0] = 0x1BB0200 Set - HD_A [27:8] = 0x1B302 I2C = 6D82

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | 1 - Port B controls EAPD on GPIO5 if MSB bit[3]=1<br>0 – Port B doesn't control EAPD on GPIO5 |
| 6 | 0 | 1 - Port A controls EAPD on GPIO5 if MSB bit[3]=1<br>0 – Port A doesn't control EAPD on GPIO5 |
| 5 | 0 | 0 – No Polarity inversion of EAPD control<br>1 – Polarity Inversion on EAPD pin |
| 4 | 1 | 1 - Port G controls EAPD<br>0 – Port G doesn't control EAPD |
| 3 | 1 | 1 - Port J controls EAPD<br>0 – Port J doesn't control EAPD |
| 2 | 1 | 1 - Port E controls EAPD<br>0 – Port E doesn't control EAPD |
| 1 | 1 | 1 - Port B controls EAPD<br>0 – Port B doesn't control EAPD |
| 0 | 1 | 1 - Port A controls EAPD<br>0 – Port A doesn't control EAPD |

### 24.2.3.4 EAPD Control MSB register

Get Response - HD_A[27:0] = 0x1BB0300 Set - HD_A [27:8] = 0x1B303 I2C = 6D83

| Bit | Default | Description |
|---|---|---|
| 7:4 | 0 | Reserved Read as 0 |
| 3 | 0 | 0 – No EAPD on GPIO5<br>1 – EAPD on GPIO5 |
| 2 | 0 | 1 - Port G controls EAPD on GPIO5 if bit[3]=1<br>0 – Port G doesn't control EAPD on GPIO5 |
| 1 | 0 | 1 - Port J controls EAPD on GPIO5 if bit[3]=1<br>0 – Port J doesn't control EAPD on GPIO5 |
| 0 | 0 | 1 - Port E controls EAPD on GPIO5 if bit[3]=1<br>0 – Port E doesn't control EAPD on GPIO5 |

### 24.2.3.5 Digital Bios Test2 LSB register

Get Response - HD_A [27:0] = 0x1BB0400 Set - HD_A[27:11] = 0x1B304 I2C = 6D84

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Clock Enable for I2S BCLK (Host)<br>1-Enable the clock<br>0-Disable the clock |
| 6 | 0 | In I2C/I2S bus mode this bit selects if bclk or mclk sources the PLL<br>0-BCLK<br>0-MCLK |
| 5:4 | ‘b10 | EAPD pad output drive strength<br>00 – IOL/IOH=+/-1.3mA<br>01 – IOL/IOH=+/-2.0mA<br>10 – IOL/IOH=+/-2.7mA<br>11 – IOL/IOH=+/-3.3mA |
| 3 | ‘b0 | 1-Push-pull mode: output enabled on EAPD<br>0-Open drain mode: output enable on EAPD controlled by signal self. |
| 2 | ‘b1 | 1-Non-input only mode.<br>0-Input-only mode, external signal controls internal amps. EAPD bit in pin widget is ignored. |
| 1:0 | ‘b10 | Pull down control for eapd pin.<br>00 – No pull up, no pull down<br>01 – Weak Pull down -> 28/65Kohm<br>10- Weak pull up value 1 -> 28-65Kohm<br>11 – Bus Keeper resistance -> 28 - 65kohm |

### 24.2.3.6 Digital Bios Test2 MSB register

Get Response - HD_A [27:0] = 0x1BB0500 Set - HD_A[27:11] = 0x1B305 I2C = 6D85

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Port E jack sense enable (pi2s_out_jsense_en) |
| 6 | 0 | Port F jack sense enable (pi2s_in_jsense_en) |
| 5 | 0 | Port E/F polarity (pi2s_jsense_pri)<br>0 = input active low (GPIO = 0 sets port presense =1)<br>1 = input active high (GPIO = 1 sets port presence = 1)<br>Note: Only used for GPIO selection,does not change SAR polarity |
| 4 | 0 | Port E/F GPIO mapping<br>0 = port E uses GPIO4 / port F uses GPIO 5<br>1 = port E uses GPIO5 / port F uses GPIO 4 |
| 3 | 0 | 1-Audio PLL bypass mode using spkr_mute_gpio1 |
| 2 | 0 | 1-PLL select for class Speaker shield blocks if not in low power mode<br>0-Blcock/MCLK source the class D spkr shield block |
| 1 | 0 | 1-PLL out divided clock(6.144Mhz) as RCOSC calibration reference clock<br>0-PLL in clock (expect 6Mhz) as RCOSC calibration reference clock |
| 0 | 0 | Polarity control for host I2S BCLK<br>1-Invert the polarity of the clock<br>0-No inversion |

### 24.2.3.7 Digital Bios Test3 LSB register

Get Response - HD_A [27:0] = 0x1BB0600 Set - HD_A[27:11] = 0x1B306 I2C = 6D86

| Bit | Default | Description |
|---|---|---|
| 7:6 | 10 | Spkrmuteb_gpio1 pad output drive strength<br>00 – IOL/IOH=+/-1.3mA<br>01 – IOL/IOH=+/-2.0mA<br>10 – IOL/IOH=+/-2.7mA<br>11 – IOL/IOH=+/-3.3mA |
| 5:4 | 01 | Pull down control for Spkrmuteb_gpio1 pin.<br>00 – No pull up, no pull down<br>01 – Weak Pull down -> 28/65Kohm<br>10- Weak pull up value 1 -> 28-65Kohm |

| | | |
|---|---|---|
| | | 11 – Bus Keeper resistance -> 28 - 65kohm |
| 3:2 | 10 | music_req_gpio0 pad output drive strength<br>00 – IOL/IOH=+/-1.3mA<br>01 – IOL/IOH=+/-2.0mA<br>10 – IOL/IOH=+/-2.7mA<br>11 – IOL/IOH=+/-3.3mA |
| 1:0 | 01 | Pull down control for music_req_gpio0 pin.<br>00 – No pull up, no pull down<br>01 – Weak Pull down -> 28/65Kohm<br>10- Weak pull up value 1 -> 28-65Kohm<br>11 – Bus Keeper resistance -> 28 - 65kohm |

### 24.2.3.8 Digital Bios Test3 MSB register

Get Response - HD_A [27:0] = 0x1BB0700 Set - HD_A[27:11] = 0x1B307 I2C = 6D87

| Bit | Default | Description |
|---|---|---|
| 7:4 | 0 | Reserved |
| 3 | 0 | Read only register providing the status rcosc calibration completed and ready to use |
| 2 | | 1- Digital bios test4 [10:8] port A force functions are disabled if a port A jack sense event is detected.<br>0-digital bios test4 [10:8] functions as normal. |
| 1 | 0 | 0- Bus mode bond option decides the bilingual<br>1- Control register overrides busmode bond option |
| 0 | | When bit 1 is = 1 bond option bypassed then<br>0-Bus mode=0 no I2S mode<br>1-Bus mode =1 force to I2S mode |

### 24.2.3.9 Digital Bios Test4 LSB register

Get Response - HD_A [27:0] = 0x1BB0800 Set - HD_A[27:8] = 0x1B308 I2C = 6D88

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | 1-Force Port A mapping to the same DAC where class D is mapped if class D is enabled.<br>0-Doesn't force port A mapping |
| 6 | 0 | Enable for additional low power mode on Azalia interface<br>0-no low power mode on Azalia interface<br>1-enable low power mode on Azalia |
| 5:4 | 00 | Reserved |
| 3 | RO | The value of EAPD pin. |
| 2 | 0 | 1 –enable s/w to disable the new clock gating added for RC OSC in Vilano from palm.<br>0-RCOSC clock is gated from rcosc top based on d3_lp signal |
| 1 | 0 | 1 –Clock disabled for rcosc/bclk for rcosc calibration block when bit 2 of this register is set<br>0-Clock enabled for rcosc/bclk rcosc calibration block when bit 2 of this register is set |
| 0 | 0 | 1 - Enable clock gating mux output from rcosc and bclk in D3 hot and D3 cold irrespective of reset is available or not.<br>0 – Disable clock gating<br>1 - D3 bcllk active, still achieve low power mode then enable this bit to do more clock gating |

### 24.2.3.10 Digital Bios Test4 MSB register

Get Response - HD_A [27:0] = 0x1BB0900 Set - HD_A[27:8] = 0x1B309 I2C = 6D89

| Bit | Default | Description |
|---|---|---|
| 7:6 | 0 | Reserved |
| 5 | 0 | Select SAR data bit 3 (SENSE A 5.1K ohm pull-down) for port F port presence<br>0: GPIO 4 or 5 selected (determined in Digital Bios Test2[4])<br>1: SAR data bit 3 selected |

| | | |
|---|---|---|
| 4 | 0 | Select SAR data bit 3 (SENSE A 5.1K ohm pull-down) for port E port presence<br>0: GPIO 4 or 5 selected (determined in Digital Bios Test2[4])<br>1: SAR data bit 3 selected |
| 3 | 0 | 1-Enable HP port during SnM<br>0-Disable HP port during SnM |
| 2 | 0 | 1-Force Port A output enable<br>0-Doesn't enable port A output enable |
| 1 | 0 | 1-Force Port A power state to D0<br>0-Doesn't force port A power state |
| 0 | 0 | 1-Force Port A output enable and port power state to D0 based on class port enable if bit 10 and 11 is set<br>0-Doesn't force port A enable based on class D enable |

### 24.2.3.11 RC OSC Control LSB register

Get Response - HD_A[27:0] = 0x1BB0A00 Set - HD_A [27:11] = 0x1B30A I2C = 6D8A

| Bit | Default | Description |
|---|---|---|
| 7 | | Reserved |
| 6 | 0 | RC OSC start calibration (R/W)<br>Before enable the RC OSC calibration make sure the 0x1B308 bit [2:1] are set to 0x3, to make sure clocks are available for calibration logic. |
| 5 | 0 | Enable forcing of calibration value with bits on 4:0 |
| 4:0 | 00000 | Forced Calibration Value (R/W) when bit 5 is set |

### 24.2.3.12 RC OSC Control MSB register

Get Response - HD_A[27:0] = 0x1BB0B00 Set - HD_A [27:11] = 0x1B30B I2C = 6D8B

| Bit | Default | Description |
|---|---|---|
| 7:6 | reserved | |
| 5 | 0 | Read only register providing the status rcosc calibration completed and ready to use |
| 4:0 | 0x10 | RC OSC calibration output, calibrated value |

### 24.2.3.13 SPKR Reference Stream ID register

Get Response - HD_A [27:0] = 0x1BB0C00 Set - HD_A[27:8] = 0x1B30C I2C = 6D8C

| Bit | Default | Description |
|---|---|---|
| 7:4 | 0 | Stream ID<br>0 = disable stream<br>1 –> F = enable stream |
| 3:0 | 0 | Channel ID |

### 24.2.3.14 SPKR Reference Enable register

Get Response - HD_A [27:0] = 0x1BB0D00 Set - HD_A[27:8] = 0x1B30D I2C = 6D8D

| Bit | Default | Description |
|---|---|---|
| 7:3 | 0 | Reserved |
| 2 | 0 | ADC5_on_ADC1 (ADC5 AEC reference is routed thru ADC1) |
| 1 | 0 | 1-Mute the SPKR reference stream |
| 0 | 0 | 1-Enable SPKR reference stream<br>0-Power down SPKR reference stream |

### 24.2.3.15 BIOS DAC_ENV_DETECT register

Get Response - HD_A [27:0] = 0x1BB0F00 Set - HD_A[27:8] = 0x1B30F I2C = 6D8F

| Bit | Default | Description |
|---|---|---|

| | | |
|---|---|---|
| 7:5 | 3'h0 | Reserved |
| 4:3 | 2'b11 | DAC envelope charge-pump voltage control threshold:<br>00: 0.350 * full scale<br>01: 0.475 * full scale<br>10: 0.450 * full scale<br>11: 0.400 * full scale |
| 2:1 | 2'b01 | DAC envelope release time:<br>00: 16ms<br>01: 32ms<br>10: 64ms<br>11: reserved |
| 0 | 1'b0 | 1 = DAC envelope detect is enabled<br>0 = DAC envelope detect is disabled |

### 24.2.3.16 BIOS MIC_MUTE LSB register

Get Response - HD_A [27:0] = 0x1BB1000 Set - HD_A[27:8] = 0x1B310 I2C = 6D90

| Bit | Default | Description |
|---|---|---|
| 7:4 | 0 | Reserved |
| 4 | 0 | 0 = Mic_Mute input and port selections unlocked (power change not required)<br>1 = Mic_Mute input and port selection locked (power change required) |
| 3:1 | 000 | GPIO capable pin selected as MIC_MUTE input<br>000 = No GPIO selected<br>001 = GPIO 0 / MUSIC_REQ<br>010 = GPIO 1 / SPDIF<br>011 = GPIO 2 / DMIC_CLK (Port C)<br>100 = GPIO 3 / DMIC_DAT (Port C)<br>101 = GPIO 4 / DMIC_CLK (Port H)<br>110 = GPIO 5 / DMIC_DAT (Port H)<br>111 = Reserved |
| 0 | 0 | MIC_MUTE polarity<br>0 = MIC_MUTE active high<br>1 = MIC_MUTE active low |

### 24.2.3.17 BIOS MIC_MUTE MSB register

Get Response - HD_A [27:0] = 0x1BB1100 Set - HD_A[27:8] = 0x1B311 I2C = 6D91

| Bit | Default | Description |
|---|---|---|
| 7:6 | 0 | Reserved |
| 5 | 0 | 0 = Normal, 1 = Port H digital microphone muted |
| 4 | 0 | 0 = Normal, 1 = Port F muted (ADC3) |
| 3 | 0 | 0 = Normal, 1 = Port K muted (ADC4) |
| 2 | 0 | 0 = Normal, 1 = Port D analog headset microphone muted |
| 1 | 0 | 0 = Normal, 1 = Port C digital microphone muted |
| 0 | 0 | 0 = Normal, 1 = Port B analog microphone muted |

### 24.2.3.18 BIOS EAPD/SPKR Mute Control LSB register

Get Response - HD_A [27:0] = 0x1BB1200 Set - HD_A[27:8] = 0x1B312 I2C = 6D92

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | PortG pad mute bypass<br>0: not bypass<br>1: bypass pad control |
| 6 | 0 | PortG pad mute polarity<br>0: don't change polarity (0 = mute)<br>1: change polarity (1 = mute) |

| | | |
|---|---|---|
| 5 | 0 | PortJ pad mute bypass<br>0: not bypass<br>1: bypass pad control |
| 4 | 0 | PortJ pad mute polarity<br>0: don't change polarity (0 = mute)<br>1: change polarity (1 = mute) |
| 3:1 | 0 | Reserved |
| 0 | 0 | Pad select for mute:<br>0: EAPD<br>1; SPKR_MUTE |

### 24.2.3.19 BIOS EAPD/SPKR Mute Control MSB register

Get Response - HD_A [27:0] = 0x1BB1300 Set - HD_A[27:8] = 0x1B313 I2C = 6D93

| Bit | Default | Description |
|---|---|---|
| 7:4 | 0 | Reserved |
| 3 | 0 | PortA or PortB pad mute bypass<br>0: not bypass<br>1: bypass pad control |
| 2 | 0 | PortA or PortB pad mute polarity<br>0: don't change polarity (0 = mute)<br>1: change polarity (1 = mute) |
| 1 | 0 | PortE pad mute bypass<br>0: not bypass<br>1: bypass pad control |
| 0 | 0 | PortE pad mute polarity<br>0: don't change polarity (0 = mute)<br>1: change polarity (1 = mute) |

### 24.2.3.20 EXT SPKRA IV / Reference Enable register

Get Response - HD_A [27:0] = 0x1BB1A00 Set - HD_A[27:8] = 0x1B31A I2C = 6D9A

| Bit | Default | Description |
|---|---|---|
| 7:5 | 000 | Reserved |
| 4 | 0 | EXT SPKRA(ADC7) sample width:<br>0 = 24-bit<br>1 = 16-bit |
| 3:2 | 00 | EXT SPKRA (ADC7) reference stream sample rate:<br>00 = 48KHz<br>01 = 96KHz<br>10 = 192KHz<br>11 = 44.1KHz |
| 1 | 0 | 1-Mute the EXT SPKRA IV sample / AEC reference stream (ADC7) |
| 0 | 0 | 1-Enable EXT SPKRA IV sample / AEC reference stream (ADC7)<br>0-Power down EXT SPKRA IV sample / AEC reference stream (ADC7) |

### 24.2.3.21 EXT SPKRA IV / Reference Stream ID register

Get Response - HD_A [27:0] = 0x1BB1B00 Set - HD_A[27:8] = 0x1B31B I2C = 6D9B

| Bit | Default | Description |
|---|---|---|
| 7:4 | 0 | ADC7 Stream ID<br>0 = disable stream<br>1 –> F = enable stream |
| 3:0 | 0 | ADC7 Channel ID |

### 24.2.3.22 EXT SPKRB IV / Reference Enable register

Get Response - HD_A [27:0] = 0x1BB1C00 Set - HD_A[27:8] = 0x1B31C I2C = 6D9C

| Bit | Default | Description |
|---|---|---|
| 7:5 | 000 | Reserved |
| 4 | 0 | EXT SPKRB (ADC8) sample width:<br>0 = 24-bit<br>1 = 16-bit |
| 3:2 | 00 | EXT SPKRB (ADC8) reference stream sample rate:<br>00 = 48KHz<br>01 = 96KHz<br>10 = 192KHz<br>11 = 44.1KHz |
| 1 | 0 | 1-Mute the EXT SPKRB IV sample / AEC reference stream (ADC8) |
| 0 | 0 | 1-Enable EXT SPKRB IV sample / AEC reference stream (ADC8)<br>0-Power down EXT SPKRB IV sample / AEC reference stream (ADC8) |

### 24.2.3.23 EXT SPKRB IV / Reference Stream ID register

Get Response - HD_A [27:0] = 0x1BB1D00 Set - HD_A[27:8] = 0x1B31D I2C = 6D9D

| Bit | Default | Description |
|---|---|---|
| 7:4 | 0 | ADC8 Stream ID<br>0 = disable stream<br>1 –> F = enable stream |
| 3:0 | 0 | ADC8 Channel ID |

### 24.2.3.24 Digital Bios Test22 register

Get Response - HD_A [27:0] = 0x1BD0C00 Set - HD_A[27:8] = 0x1B50C I2C = 6E8C

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Reserved |
| 6:0 | ‘h44 | New Volume control RAMP_STEP, this value determines RAMP STEP value.<br>The 6-bits are split internally as<br>RAMP_MANT[3:0] = RAMP_STEP[3:0]<br>RAMP_EXP[2:0] = RAMP_STEP[6:4]<br>The Final RAMP STEP is as follows<br>RAMP_STEP_VALUE[15:7+EXP] = 0;<br>RAMP_STEP_VALUE[6+EXP:3+EXP] = MANT<br>RAMP_STEP_VALUE[2+EXP:0] = 0<br>= ‘h44: mait = 4, exp =4, value = 4 * (2 ^(3+4)) = 512 |

### 24.2.3.25 Digital Bios Test23 register

Get Response - HD_A [27:0] = 0x1BD0D00 Set - HD_A[27:8] = 0x1B50D I2C = 6E8D

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | RO<br>Multi-Button SAR DATA Read |

### 24.2.3.26 Digital Bios Test25 register

Get Response - HD_A [27:0] = 0x1BD0F00 Set - HD_A[27:8] = 0x1B50F I2C = 6E8F

| Bit | Default | Description |
|---|---|---|
| 7:6 | 0 | Reserved |
| 5:0 | ‘h4 | Combo head set LP filter band select for T/R detection.<br>0x4 = 5ms (default)<br>01 – sets filter Time constant to 82ms |

| | | |
|---|---|---|
| | | 02 – sets filter Time constant to 21ms<br>04 – sets filter Time constant to 5ms<br>08 – sets filter Time constant to 1.3ms<br>63 - sets filter Time constant to 35us |

### 24.2.3.27 Digital Bios Test26 register

Get Response - HD_A [27:0] =0x1BD1C00 Set - HD_A[27:8] = 0x1B51C I2C = 6E9C

| Bit | Default | Description |
|---|---|---|
| 7:6 | 0 | Reserved |
| 3:2 | 'h2 | Combo_mon_window for T/R SENSE 2<br>Monitor window before detection result.<br>00 = 50 mSec<br>01 = 100 mSec<br>10 = 125 mSec<br>11 = 150 mSec |
| 1:0 | 'h2 | Combo_mon_window for T/R SENSE 1<br>Monitor window before detection result.<br>00 = 50 mSec<br>01 = 100 mSec<br>10 = 125 mSec<br>11 = 150 mSec |

### 24.2.3.28 Digital Bios Test27 register

Get Response - HD_A [27:0] = 0x1BD1D00 Set - HD_A[27:8] = 0x1B51D I2C = 6E9D

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Disable the DRC release stop fix<br>0: no fix disable<br>1: disable the fix |
| 6:0 | 'h3C | The DRC release stop threshold as two's complement in dB (lower 7bit, the value's bit 8 assume be 1)<br>Default 'h3C means {1'b1, 7'h3C} = -68 |

### 24.2.3.29 Digital Bios Test28 register

Get Response - HD_A [27:0] = 0x1BD1E00 Set - HD_A[27:8] = 0x1B51E I2C = 6E9E

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Override for por5_thrshsel |
| 6 | 0 | Override value for Por5 threshold selection<br>0 = 0.875v<br>1 = 1.52v |
| 5:4 | 0 | Reserved |
| 3 | 1'b1 | I2C_ID pad input enable |
| 2 | 1'b1 | HBUS_MODE pad input enable |
| 1 | 1'b1 | OTP1 pad input enable |
| 0 | 1'b1 | OPT0 pad input enable |

## 24.2.4 I2C Master Registers

### 24.2.4.1 I2C Master REG1 register

Get Response - HD_A [27:0] = 0x1BD2000 Set - HD_A[27:8] = 0x1B520 I2C = 6EA0

| Bit | Default | Description |
|---|---|---|

| | | |
|---|---|---|
| 7 | 0 | I2c_clk_en – Enable I2C clock<br>0 = disabled<br>1 = enabled |
| 6:2 | 0 | Reserved |
| 1:0 | 01 | I2C transfer rate mode<br>00: 100KHz<br>01: 400KHz<br>10: 1MHz<br>11: Reserved |

### 24.2.4.2 I2C Master REG2 register

Get Response - HD_A [27:0] = 0x1BD2100 Set - HD_A[27:8] = 0x1B521 I2C = 6EA1

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Reserved |
| 6:0 | 0 | Dev_addr0 - Sets the address for device #1 |

### 24.2.4.3 I2C Master REG3 register

Get Response - HD_A [27:0] = 0x1BD2200 Set - HD_A[27:8] = 0x1B522 I2C = 6EA2

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Reserved |
| 6:0 | 0 | Dev_addr1 - Sets the address for device #2 |

### 24.2.4.4 Host2I2C CTRL0 REG1 register

Get Response - HD_A [27:0] = 0x1BD2600 Set - HD_A[27:8] = 0x1B526 I2C = 6EA6

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Msb_sub_address - Sets the sub-address (the most significant byte) for the host2I2C controller. |

### 24.2.4.5 Host2I2C CTRL0 REG2 register

Get Response - HD_A [27:0] = 0x1BD2700 Set - HD_A[27:8] = 0x1B527 I2C = 6EA7

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Lsb_sub_address - Sets the sub-address (the least significant byte) for the host2I2C controller. |

### 24.2.4.6 Host2I2C CTRL0 REG3 register

Get Response - HD_A [27:0] = 0x1BD2800 Set - HD_A[27:8] = 0x1B528 I2C = 6EA8

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Reserved |
| 6 | 0 | Sub_addr_len - Sets the length of the sub-address.<br>0: One-byte sub-address<br>1: Two-byte sub-address |
| 5 | 0 | Device – sets the device ID for the I2C transfer<br>0: Access device #1<br>1: Access device #2 |
| 4 | 0 | Rd_wr – Sets the transfer type.<br>0: Write operation<br>1: Read operation |
| 3:0 | 0 | Burst_length – Sets the length for the I2C transfer<br>0: 1 byte<br>1: 2 bytes<br>… |

| | | |
|---|---|---|
| | | 7: 8 bytes<br>Others: Reserved, treated as 8 bytes for both. |

### 24.2.4.7 Host2I2C CTRL0 REG4 register

Get Response - HD_A [27:0] = 0x1BD2900 Set - HD_A[27:8] = 0x1B529 I2C = 6EA9

| Bit | Default | Description |
|---|---|---|
| 7:3 | 0 | Reserved |
| 2 | 0 | Status – information about current transfer<br>0: Successful transfer<br>1: Failed transfer |
| 1 | 0 | idb – Inbound Doorbell, which is set by host2I2C controller and cleared by host when writing 1. Acts as a level signal. (R/W1C) |
| 0 | 0 | odb – Outbound Doorbell, which is set by host2I2C controller and cleared by host when writing 1 to idb by host2I2C controller. Acts as a level signal. |

### 24.2.4.8 Host2I2C CTRL0 REG5 register

Get Response - HD_A [27:0] = 0x1BD2A00 Set - HD_A[27:8] = 0x1B52A I2C = 6EAA

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Wr_rd_dat0 – Sets the write data #1 under write operation by host, or the read data #1 under read operation by the host2I2C controller. |

### 24.2.4.9 Host2I2C CTRL0 REG6 register

Get Response - HD_A [27:0] = 0x1BD2B00 Set - HD_A[27:8] = 0x1B52B I2C = 6EAB

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Wr_rd_dat1 – Sets the write data #2 under write operation by host, or the read data #2 under read operation by the host2I2C controller. |

### 24.2.4.10 Host2I2C CTRL0 REG7 register

Get Response - HD_A [27:0] = 0x1BD2C00 Set - HD_A[27:8] = 0x1B52C I2C = 6EAC

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Wr_rd_dat2 – Sets the write data #3 under write operation by host, or the read data #3 under read operation by the host2I2C controller. |

### 24.2.4.11 Host2I2C CTRL0 REG8 register

Get Response - HD_A [27:0] = 0x1BD2D00 Set - HD_A[27:8] = 0x1B52D I2C = 6EAD

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Wr_rd_dat3 – Sets the write data #4 under write operation by host, or the read data #4 under read operation by the host2I2C controller. |

### 24.2.4.12 Host2I2C CTRL0 REG9 register

Get Response - HD_A [27:0] = 0x1BD2E00 Set - HD_A[27:8] = 0x1B52E I2C = 6EAE

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Wr_rd_dat4 – Sets the write data #5 under write operation by host, or the read data #5 under read operation by the host2I2C controller. |

### 24.2.4.13 Host2I2C CTRL0 REG10 register

Get Response - HD_A [27:0] = 0x1BD2F00 Set - HD_A[27:8] = 0x1B52F I2C = 6EAF

| Bit | Default | Description |
|---|---|---|

| | | |
|---|---|---|
| 7:0 | 0 | Wr_rd_dat5 – Sets the write data #6 under write operation by host, or the read data #6 under read operation by the host2I2C controller. |

### 24.2.4.14 Host2I2C CTRL0 REG11 register

Get Response - HD_A [27:0] = 0x1BD3000 Set - HD_A[27:8] = 0x1B530 I2C = 6EB0

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Wr_rd_dat6 – Sets the write data #7 under write operation by host, or the read data #7 under read operation by the host2I2C controller. |

### 24.2.4.15 Host2I2C CTRL0 REG12 register

Get Response - HD_A [27:0] = 0x1BD3100 Set - HD_A[27:8] = 0x1B531 I2C = 6EB1

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Wr_rd_dat7 – Sets the write data #8 under write operation by host, or the read data #8 under read operation by the host2I2C controller. |

### 24.2.4.16 Host2I2C CTRL1 REG1 register

Get Response - HD_A [27:0] = 0x1BD3200 Set - HD_A[27:8] = 0x1B532 I2C = 6EB2

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Msb_sub_address - Sets the sub-address (the most significant byte) for the host2I2C controller. |

### 24.2.4.17 Host2I2C CTRL1 REG2 register

Get Response - HD_A [27:0] = 0x1BD3300 Set - HD_A[27:8] = 0x1B533 I2C = 6EB3

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Lsb_sub_address - Sets the sub-address (the least significant byte) for the host2I2C controller. |

### 24.2.4.18 Host2I2C CTRL1 REG3 register

Get Response - HD_A [27:0] = 0x1BD3400 Set - HD_A[27:8] = 0x1B534 I2C = 6EB4

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Reserved |
| 6 | 0 | Sub_addr_len - Sets the length of the sub-address.<br>0: One-byte sub-address<br>1: Two-byte sub-address |
| 5 | 0 | Device – sets the device ID for the I2C transfer<br>0: Access device #1<br>1: Access device #2 |
| 4 | 0 | Rd_wr – Sets the transfer type.<br>0: Write operation<br>1: Read operation |
| 3:0 | 0 | Burst_length – Sets the length for the I2C transfer<br>0: 1 byte<br>1: 2 bytes<br>…<br>7: 8 bytes<br>Others: Reserved, treated as 8 bytes for both. |

### 24.2.4.19 Host2I2C CTRL1 REG4 register

Get Response - HD_A [27:0] = 0x1BD3500 Set - HD_A[27:8] = 0x1B535 I2C = 6EB5

| Bit | Default | Description |
|---|---|---|
| 7:3 | 0 | Reserved |

| | | |
|---|---|---|
| 2 | 0 | Status – information about current transfer<br>0: Successful transfer<br>1: Failed transfer |
| 1 | 0 | idb – Inbound Doorbell, which is set by host2I2C controller and cleared by host when writing 1. Acts as a level signal. (R/W1C) |
| 0 | 0 | odb – Outbound Doorbell, which is set by host2I2C controller and cleared by host when writing 1 to idb by host2I2C controller. Acts as a level signal. |

### 24.2.4.20 Host2I2C CTRL1 REG5 register

Get Response - HD_A [27:0] = 0x1BD3600 Set - HD_A[27:8] = 0x1B536 I2C = 6EB6

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Wr_rd_dat0 – Sets the write data #1 under write operation by host, or the read data #1 under read operation by the host2I2C controller. |

### 24.2.4.21 Host2I2C CTRL1 REG6 register

Get Response - HD_A [27:0] = 0x1BD3700 Set - HD_A[27:8] = 0x1B537 I2C = 6EB7

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Wr_rd_dat1 – Sets the write data #2 under write operation by host, or the read data #2 under read operation by the host2I2C controller. |

### 24.2.4.22 Host2I2C CTRL1 REG7 register

Get Response - HD_A [27:0] = 0x1BD3800 Set - HD_A[27:8] = 0x1B538 I2C = 6EB8

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Wr_rd_dat2 – Sets the write data #3 under write operation by host, or the read data #3 under read operation by the host2I2C controller. |

### 24.2.4.23 Host2I2C CTRL1 REG8 register

Get Response - HD_A [27:0] = 0x1BD3900 Set - HD_A[27:8] = 0x1B539 I2C = 6EB9

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Wr_rd_dat3 – Sets the write data #4 under write operation by host, or the read data #4 under read operation by the host2I2C controller. |

### 24.2.4.24 Host2I2C CTRL1 REG9 register

Get Response - HD_A [27:0] = 0x1BD3A00 Set - HD_A[27:8] = 0x1B53A I2C = 6EBA

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Wr_rd_dat4 – Sets the write data #5 under write operation by host, or the read data #5 under read operation by the host2I2C controller. |

### 24.2.4.25 Host2I2C CTRL1 REG10 register

Get Response - HD_A [27:0] = 0x1BD3B00 Set - HD_A[27:8] = 0x1B53B I2C = 6EBB

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Wr_rd_dat5 – Sets the write data #6 under write operation by host, or the read data #6 under read operation by the host2I2C controller. |

### 24.2.4.26 Host2I2C CTRL1 REG11 register

Get Response - HD_A [27:0] = 0x1BD3C00 Set - HD_A[27:8] = 0x1B53C I2C = 6EBC

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Wr_rd_dat6 – Sets the write data #7 under write operation by host, or the read data #7 under read operation by the host2I2C controller. |

#### 24.2.4.27 Host2I2C CTRL1 REG12 register

Get Response - HD_A [27:0] = 0x1BD3D00 Set - HD_A[27:8] = 0x1B53D I2C = 6EBD

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Wr_rd_dat7 – Sets the write data #8 under write operation by host, or the read data #8 under read operation by the host2I2C controller. |

### 24.2.5 MBIST Registers

#### 24.2.5.1 DAC BIST RAM Status register

Get Response - HD_A [27:0] = 0x1BB1F00 Set - HD_A[27:8] = 0x1B31F I2C = 6D9F

| Bit | Default | Description |
|---|---|---|
| 7:0 | | DAC RAM BIST Status[7:0] |

#### 24.2.5.2 ADC BIST RAM Status register

Get Response - HD_A [27:0] = 0x1BB2000 Set - HD_A[27:8] = 0x1B320 I2C = 6DA0

| Bit | Default | Description |
|---|---|---|
| 7:5 | | reserved |
| 4:0 | | ADC RAM BIST Status[4:0] |

#### 24.2.5.3 ADC/DAC BIST RAM Start register

Get Response - HD_A [27:0] = 0x1BB2100 Set - HD_A[27:8] = 0x1B321 I2C=6DA1

| Bit | Default | Description |
|---|---|---|
| 7:6 | | reserved |
| 5:4 | | ADC RAM BIST Status[6:5] |
| 3:2 | | DAC RAM BIST Status[9:8] |
| 1 | 0 | BIST Clock Force |
| 0 | 0 | BIST RAM Start |

#### 24.2.5.4 SRC BIST RAM Status register

Get Response - HD_A [27:0] = 0x1BB2200 Set - HD_A[27:8] = 0x1B322 I2C=6DA2

| Bit | Default | Description |
|---|---|---|
| 7:6 | 0 | reserved |
| 5:0 | 0 | Peripheral I2S A SRC RAM BIST Status |

#### 24.2.5.5 SRC WS BIST RAM Status register

Get Response - HD_A [27:0] = 0x1BB2300 Set - HD_A[27:8] = 0x1B323 I2C=6DA3

| Bit | Default | Description |
|---|---|---|
| 7:6 | 0 | Write Sequencer RAM BIST Status |
| 5:0 | 0 | Peripheral I2S B SRC RAM BIST Status |

## 24.2.6 I2SPCM Registers

### 24.2.6.1 I2SPCM_CTRL_REG1A register

Get Response - HD_A [27:0] = 0x1BC0000 Set - HD_A[27:8] = 0x1B400 I2C=6E00

| Bit | Default | Description |
|---|---|---|
| 7:2 | 0 | RX_WS_WID[5:0] – Receive word strobe (sync/LRCK) high phase width, in terms of bit clock cycles, when in master mode.<br>0000000 – 1 Cycle<br>0000001 – 2 Cycle<br>…<br>1111111 – 128 Cycles |
| 1 | 0 | RX_WS_POL - DAC LRCK inversion control<br>0 = DAC LRCK is not inverted<br>1 = DAC LRCK is inverted<br>This bit is used in I2S modes only |
| 0 | 1 | RX_DATA_ONE_LINE<br>0 – Receive DAC0, DAC1 data on i2s_rx_data_1; DAC2, DAC3 data on i2s_rx_data_2; DAC4, DAC5 data on i2s_rx_data_3.<br>1 – Receive all 6 channels of DAC data on i2s_rx_data_1 |

### 24.2.6.2 I2SPCM_CTRL_REG1B register

Get Response - HD_A [27:0] = 0x1BC0100 Set - HD_A[27:8] = 0x1B401 I2C = 6E01

| Bit | Default | Description |
|---|---|---|
| 7:6 | 00 | RX_SA_SIZE - Receive sample size control<br>00 – 8-bit data<br>01 – 16-bit data<br>10 – 24-bit data<br>11 – Reserved |
| 5:1 | 00000 | RX_FRM_LEN - Receive frame length in terms of bit clock cycles, when in master mode. It is (N+1)*8<br>00000 – 8 Clocks per Frame<br>00001 – 16 Clocks per Frame<br>00010 – 24 Clocks per Frame<br>00011 – 32 Clocks per Frame<br>…<br>11111 – 256 Clocks per Frame |
| 0 | 0 | RX_WS_WID[6] – Receive word strobe (sync/LRCK) high phase width, in terms of bit clock cycles, when in master mode.<br>0000000 – 1 Cycle<br>0000001 – 2 Cycle<br>…<br>1111111 – 128 Cycles |

### 24.2.6.3 I2SPCM_CTRL_REG1C register

Get Response - HD_A [27:0] = 0x1BC0200 Set - HD_A[27:8] = 0x1B402 I2C = 6E02

| Bit | Default | Description |
|---|---|---|
| 7:2 | 000000 | TX_WS_WID[5:0] - Transmit word strobe (sync/LRCK) high phase width, in terms of bit clock cycles, when in master mode.<br>0000000 – 1 Cycle<br>0000001 – 2 Cycle<br>…<br>1111111 – 128 Cycles |
| 1 | 0 | TX_WS_POL - ADC LRCK inversion control<br>0 = ADC LRCK is not inverted<br>1 = ADC LRCK is inverted<br>This bit is used in I2S mode only |

| | | |
|---|---|---|
| 0 | 1 | TX_DATA_ONE_LINE<br>0 – Send ADC0, ADC1 data on i2s_tx_data_1 and ADC2, ADC3 data on i2s_tx_data_2.<br>1 – Send all 4 channels of ADC data on i2s_tx_data_1 |

### 24.2.6.4 I2SPCM_CTRL_REG1D register

Get Response - HD_A [27:0] = 0x1BC0300 Set - HD_A[27:8] = 0x1B403 I2C = 6E03

| Bit | Default | Description |
|---|---|---|
| 7:6 | 00 | TX_SA_SIZE - Transmit sample size control<br>00 – 8-bit data<br>01 – 16-bit data<br>10 – 24-bit data<br>11 – Reserved |
| 5:1 | 00000 | TX_FRM_LEN - Transmit frame length in terms of bit clock cycles, when in master mode. It is (N+1)*8<br>00000 – 8 Clocks per Frame<br>00001 – 16 Clocks per Frame<br>00010 – 24 Clocks per Frame<br>00011 – 32 Clocks per Frame<br>…<br>11111 – 256 Clocks per Frame |
| 0 | 0 | TX_WS_WID[6] - Transmit word strobe (sync/LRCK) high phase width, in terms of bit clock cycles, when in master mode.<br>0000000 – 1 Cycle<br>0000001 – 2 Cycle<br>…<br>1111111 – 128 Cycles |

### 24.2.6.5 I2SPCM_CTRL_REG2A register

Get Response - HD_A [27:0] = 0x1BC0400 Set - HD_A[27:8] = 0x1B404 I2C = 6E04

| Bit | Default | Description |
|---|---|---|
| 7:6 | 00 | TX_SLOT_1[1:0] - First Slot number for transmit channel-1. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 5 | 0 | Enable transmit on channel 6. |
| 4 | 0 | Enable transmit on channel 5. |
| 3 | 0 | Enable transmit on channel 4. |
| 2 | 0 | Enable transmit on channel 3. |
| 1 | 0 | Enable transmit on channel 2. |
| 0 | 0 | Enable transmit on channel 1. |

### 24.2.6.6 I2SPCM_CTRL_REG2B register

Get Response - HD_A [27:0] = 0x1BC0500 Set - HD_A[27:8] = 0x1B405 I2C = 6E05

| Bit | Default | Description |
|---|---|---|
| 7:3 | 00000 | TX_SLOT_2[4:0] - First Slot number for transmit channel-2. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 2:0 | 000 | TX_SLOT_1[4:2] - First Slot number for transmit channel-1. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |

### 24.2.6.7 I2SPCM_CTRL_REG2C register

Get Response - HD_A [27:0] = 0x1BC0600 Set - HD_A[27:8] = 0x1B406 I2C = 6E06

| Bit | Default | Description |
|---|---|---|
| 7:5 | 000 | TX_SLOT_4[2:0] - First Slot number for transmit channel-4. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 4:0 | 00000 | TX_SLOT_3[4:0] - First Slot number for transmit channel-3. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots, respectively. |

### 24.2.6.8 I2SPCM_CTRL_REG2D register

Get Response - HD_A [27:0] = 0x1BC0700 Set - HD_A[27:8] = 0x1B407 I2C = 6E07

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | TX_DSTART_DLY - Transmit data start delay control<br>0 = Data is not delayed<br>1 = Data is delayed by one bit clock<br>Used in all modes where data is to be shifted by one bit clock |
| 6 | 0 | TX_ENDIAN_SEL – Transmit data endianness select. Determines data shift direction.<br>0 = MSB first<br>1 = LSB first |
| 5 | 0 | TX_TRI_N - Enable the transmit output enable only for the slots that are actively supposed to drive.<br>0 – Tristate<br>1 – Don't tristate<br>This bit is used in all the TDM modes. |
| 4 | 0 | TX_MASTER<br>0 – Transmitter is slave<br>1 – Transmitter is master |
| 3 | 0 | TX_DATA_NEG_BCLK - Transmit data and LRCK is driven on negative edge of BCLK<br>0 – Data and LRCK driven on positive edge of BCLK<br>1 – Data and LRCK driven on negative edge of BCLK |
| 2 | 0 | Reserved |
| 1:0 | 00 | TX_SLOT4[4:3] |

### 24.2.6.9 I2SPCM_CTRL_REG3A register

Get Response - HD_A [27:0] = 0x1BC0800 Set - HD_A[27:8] = 0x1B408 I2C = 6E08

| Bit | Default | Description |
|---|---|---|
| 7:6 | 00 | RX_SLOT_1[1:0] - First Slot number for receive channel-1. Each slot is 8 bits. All the enabled receive channels should be configured with unique slot numbers.<br>Note: If receive sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 5 | 0 | Enable receive on channel 6. |
| 4 | 0 | Enable receive on channel 5. |
| 3 | 0 | Enable receive on channel 4. |
| 2 | 0 | Enable receive on channel 3. |
| 1 | 0 | Enable receive on channel 2. |
| 0 | 0 | Enable receive on channel 1. |

### 24.2.6.10 I2SPCM_CTRL_REG3B register

Get Response - HD_A [27:0] = 0x1BC0900 Set - HD_A[27:8] = 0x1B409 I2C = 6E09

| Bit | Default | Description |
|---|---|---|
| 7:3 | 00000 | RX_SLOT_2 - First Slot number for receive channel-2. Each slot is 8 bits. All the enabled receive channels should be configured with unique slot numbers.<br>Note: If receive sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 2:0 | 000 | RX_SLOT_1[4:2] - First Slot number for receive channel-1. Each slot is 8 bits. All the enabled receive channels should be configured with unique slot numbers.<br>Note: If receive sample size is 16 or 24, it uses two or three consecutive slots respectively. |

### 24.2.6.11 I2SPCM_CTRL_REG3C register

Get Response - HD_A [27:0] = 0x1BC0A00 Set - HD_A[27:8] = 0x1B40A I2C = 6E0A

| Bit | Default | Description |
|---|---|---|
| 7:5 | 000 | RX_SLOT_4[2:0] - First Slot number for receive channel-4. Each slot is 8 bits. All the enabled receive channels should be configured with unique slot numbers.<br>Note: If receive sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 4:0 | 00000 | RX_SLOT_3- First Slot number for receive channel-3. Each slot is 8 bits. All the enabled receive channels should be configured with unique slot numbers.<br>Note: If receive sample size is 16 or 24, it uses two or three consecutive slots respectively. |

### 24.2.6.12 I2SPCM_CTRL_REG3D register

Get Response - HD_A [27:0] = 0x1BC0B00 Set - HD_A[27:8] = 0x1B40B I2C = 6E0B

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | RX_DSTART_DLY - Receive data start delay control<br>0 = Data is not delayed<br>1 = Data is delayed by one bit clock<br>Used in all modes where data is to be shifted by one bit clock |
| 6 | 0 | RX_ENDIAN_SEL – Receive data endianness select. Determines data shift direction.<br>0 = MSB first<br>1 = LSB first |
| 5 | 0 | Reserved |
| 4 | 0 | RX_MASTER<br>0 – Receiver is slave<br>1 – Receiver is master |
| 3 | 0 | RX_DATA_NEG_BCLK - Receive data and LRCK is driven on negative edge of BCLK<br>0 – Data and LRCK driven on positive edge of BCLK<br>1 – Data and LRCK driven on negative edge of BCLK |
| 2 | 0 | Reserved |
| 1:0 | 00 | RX_SLOT_4[4:3] - First Slot number for receive channel-4. Each slot is 8 bits. All the enabled receive channels should be configured with unique slot numbers.<br>Note: If receive sample size is 16 or 24, it uses two or three consecutive slots respectively. |

### 24.2.6.13 I2SPCM_CTRL_REG4A register

Get Response- HD_A [27:0] = 0x1BC0C00 Set - HD_A[27:8] = 0x1B40C I2C = 6E0C

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Reserved |
| 6 | 0 | TX_LAST_ADC_CH_EN - Enable to use Last channel programmed in TX_LAST_ADC_CH field.<br>1 – Enable |

| | | |
|---|---|---|
| | | 0 – Disable (Use default)<br>Note: This bit should be enabled only if all channels are on same data line and transmit channels are not in ascending order, i.e. tx_slot1 < tx_slot2 < tx_slot3 < tx_slot4. |
| 5 | 0 | Reserved |
| 4 | 0 | Reserved |
| 3 | 0 | Reserved |
| 2 | 0 | Reserved |
| 1 | 0 | TX_MUTE - Mute the transmitter (all zeros will be sent) |
| 0 | 0 | RX_MUTE - Mute the receiver (data received will be zeroed out) |

### 24.2.6.14 I2SPCM_CTRL_REG4B register

Get Response - HD_A [27:0] = 0x1BC0D00 Set - HD_A[27:8] = 0x1B40D I2C = 6E0D

| Bit | Default | Description |
|---|---|---|
| 7:4 | 0000 | TX_LAST_ADC_CH[3:0]<br>Combined with TX_LAST_ADC_CH[4], indicates channel number which represents the last TX channel in a frame.<br>5'h00 – Channel-1 is the last channel in a frame (adc1)<br>5'h01 – Channel-2 is the last channel in a frame (adc1)<br>5'h02 – Channel-3 is the last channel in a frame (adc2)<br>5'h03 – Channel-4 is the last channel in a frame (adc2)<br>5'h04 – Channel-5 is the last channel in a frame (adc3)<br>5'h05 – Channel-6 is the last channel in a frame (adc3)<br>5'h06 – Channel-7 is the last channel in a frame<br>5'h07 – Channel-8 is the last channel in a frame<br>5'h08 – Channel-9 is the last channel in a frame<br>5'h09 – Channel-10 is the last channel in a frame<br>5'h0a – Channel-11 is the last channel in a frame<br>5'h0b – Channel-12 is the last channel in a frame<br>5'h0c – Channel-13 is the last channel in a frame<br>5'h0d – Channel-14 is the last channel in a frame<br>5'h0e – Channel-15 is the last channel in a frame<br>5'h0f – Channel-16 is the last channel in a frame<br>5'h10 – Channel-17 is the last channel in a frame<br>5'h11 – Channel-18 is the last channel in a frame<br>All other combinations are invalid. |
| 3:1 | 000 | RX_LAST_DAC_CH[2:0]<br>Combined with RX_LAST_DAC_CH[3], indicates channel number which represents the last RX channel in a frame.<br>4'h0 – Channel-1 is the last channel in a frame<br>4'h1 – Channel-2 is the last channel in a frame<br>4'h2 – Channel-3 is the last channel in a frame<br>4'h3 – Channel-4 is the last channel in a frame<br>4'h4 – Channel-5 is the last channel in a frame<br>4'h5 – Channel-6 is the last channel in a frame<br>4'h6 – Channel-7 is the last channel in a frame<br>4'h7 – Channel-8 is the last channel in a frame<br>4'h8 – Channel-9 is the last channel in a frame<br>4'h9 – Channel-10 is the last channel in a frame<br>All other combinations are invalid. |
| 0 | 0 | RX_LAST_DAC_CH_EN - Enable to use Last channel programmed in RX_LAST_DAC_CH field.<br>1 – Enable<br>0 – Disable (Use default)<br>Note: This bit should be enabled only if all channels are on same data line and receive channels are not in ascending order, i.e. rx_slot1 < rx_slot2 < rx_slot3 < rx_slot4 < rx_slot5 < rx_slot6. |

### 24.2.6.15 I2SPCM_CTRL_REG4C register

Get Response - HD_A [27:0] = 0x1BC0E00 Set - HD_A[27:8] = 0x1B40E I2C = 6E0E

| Bit | Default | Description |
|---|---|---|
| 7:5 | 000 | RX_SLOT_6[2:0] - First Slot number for receive channel-6. Each slot is 8 bits. All the enabled receive channels should be configured with unique slot numbers.<br>Note: If receive sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 4:0 | 00000 | RX_SLOT_5- First Slot number for receive channel-5. Each slot is 8 bits. All the enabled receive channels should be configured with unique slot numbers.<br>Note: If receive sample size is 16 or 24, it uses two or three consecutive slots respectively. |

### 24.2.6.16 I2SPCM_CTRL_REG4D register (bits 7:6 Changed in Oceanside)

Get Response - HD_A [27:0] = 0x1BC0F00 Set - HD_A[27:8] = 0x1B40F I2C = 6E0F

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | RX_LAST_DAC_CH[3]<br>Assembling with RX_LAST_DAC_CH[2:0], indicates channel number which represents the last RX channel in a frame.<br>4'h0 – Channel-1 is the last channel in a frame<br>4'h1 – Channel-2 is the last channel in a frame<br>4'h2 – Channel-3 is the last channel in a frame<br>4'h3 – Channel-4 is the last channel in a frame<br>4'h4 – Channel-5 is the last channel in a frame<br>4'h5 – Channel-6 is the last channel in a frame<br>4'h6 – Channel-7 is the last channel in a frame<br>4'h7 – Channel-8 is the last channel in a frame<br>4'h8 – Channel-9 is the last channel in a frame<br>4'h9 – Channel-10 is the last channel in a frame<br>All other combinations are invalid. |
| 6 | 0 | TX_LAST_ADC_CH[4]<br>Assembling with TX_LAST_ADC_CH[3:0], indicates channel number which represents the last TX channel in a frame.<br>5'h00 – Channel-1 is the last channel in a frame (adc1)<br>5'h01 – Channel-2 is the last channel in a frame (adc1)<br>5'h02 – Channel-3 is the last channel in a frame (adc2)<br>5'h03 – Channel-4 is the last channel in a frame (adc2)<br>5'h04 – Channel-5 is the last channel in a frame (adc3)<br>5'h05 – Channel-6 is the last channel in a frame (adc3)<br>5'h06 – Channel-7 is the last channel in a frame<br>5'h07 – Channel-8 is the last channel in a frame<br>5'h08 – Channel-9 is the last channel in a frame<br>5'h09 – Channel-10 is the last channel in a frame<br>5'h0a – Channel-11 is the last channel in a frame<br>5'h0b – Channel-12 is the last channel in a frame<br>5'h0c – Channel-13 is the last channel in a frame<br>5'h0d – Channel-14 is the last channel in a frame<br>5'h0e – Channel-15 is the last channel in a frame<br>5'h0f – Channel-16 is the last channel in a frame<br>5'h10 – Channel-17 is the last channel in a frame<br>5'h11 – Channel-18 is the last channel in a frame<br>All other combinations are invalid. |
| 5 | 0 | Lrck_clk_pol<br>0: Don't invert TX BCLK polarity (1/2 Clock Cycle timing)<br>1: Invert TX BCLK polarity (Full clock Cycle Timing). |
| 4 | 0 | Tx_dstart_dly_more<br>0: Don't Delay TX data (1/2 Clock Cycle Timing)<br>1: Delay TX data by 1 clock (Full clock cycle timing) |
| 3 | 0 | Tx_fs_bypass<br>0: Frame synch not bypassed (1/2 Clock cycle timing mode)<br>1: Frame synch bypassed for full cycle Timing mode. |
| 2 | 0 | Rx_keep_pol |

| | | |
|---|---|---|
| | | 0: Invert RX BCLK polarity (½ Clock cycle timing mode).<br>1: Keeps RX_BCLK polarity (Full clock cycle Timing mode) |
| 1:0 | 00 | RX_SLOT_6[4:3] - First Slot number for receive channel-6. Each slot is 8 bits. All the enabled receive channels should be configured with unique slot numbers.<br>Note: If receive sample size is 16 or 24, it uses two or three consecutive slots respectively. |

### 24.2.6.17 I2SPCM_CTRL_REG5A register

Get Response - HD_A [27:0] = 0x1BC1000 Set - HD_A[27:8] = 0x1B410 I2C = 6E10

| Bit | Default | Description |
|---|---|---|
| 7:5 | 000 | Reserved |
| 4:0 | 5'h0 | TX_SLOT5 - First Slot number for transmit channel-5. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots, respectively. |

### 24.2.6.18 I2SPCM_CTRL_REG5B register

Get Response - HD_A [27:0] = 0x1BC1100 Set - HD_A[27:8] = 0x1B411 I2C = 6E11

| Bit | Default | Description |
|---|---|---|
| 7:5 | 000 | Reserved |
| 4:0 | 00 | TX_SLOT6 – First slot number for transmit channel-6. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: if transmit sample size is 16 or 24, it uses two or three consecutive slots, respectively. |

### 24.2.6.19 I2SPCM_CTRL_REG5C register

Get Response- HD_A [27:0] = 0x1BC1200 Set - HD_A[27:8] = 0x1B412 I2C = 6E12

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Reserved |

### 24.2.6.20 I2SPCM_CTRL_REG5D register

Get Response - HD_A [27:0] = 0x1BC1300 Set - HD_A[27:8] = 0x1B413 I2C = 6E13

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | I2S_PCM_CLK_DIV_CHNG_EN:<br>Enable for divisor change of I2S/PCM clock in master mode.<br>The divisor value on I2S_PCM_CLK_DIV can get effect once we have a zero to one transition on this bit |
| 6:0 | 0x0B | I2S_PCM_CLK_DIV: Divisor value for I2C/PCM clock in master mode. Source clock can be PLL or Bypassed MCLK.<br>0 = divide by 1<br>1 = divide by 2<br>…<br>N = divide by N+1 |

### 24.2.6.21 I2C Slave ID register

Get Response - HD_A [27:0] = 0x1BC1400 Set - HD_A[27:8] = 0x1B414 I2C = 6E14

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Reserved |
| 6:0 | 0x33 | I2C Slave ID |

### 24.2.6.22 HDA2I2S / I2S2HD Transition register

Get Response - HD_A [27:0] = 0x1BC1500 Set - HD_A[27:8] = 0x1B415 I2C = 6E15

| Bit | Default | Description |
|---|---|---|
| 7:2 | 000000 | Reserved |
| 1 | 0 | = Set I2S to HD pending transition Bit |
| 0 | 0 | =1 set HD to I2S Pending Transition Bit |

### 24.2.6.23 I2C CLK CNT register

Get Response - HD_A [27:0] = 0x1BC1600 Set - HD_A[27:8] = 0x1B416 I2C = 6E16

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0x02 | This register set the # of clocks required when accessing Register Block<br>0x00: 1 Clock<br>0x01: 2 Clocks<br>…<br>0xff: 256 Clocks |

### 24.2.6.24 Unsolicited Message Interrupt Control register

Get Response - HD_A [27:0] = 0x1BC1700 Set - HD_A[27:8] = 0x1B417 I2C = 6E17

| Bit | Default | Description |
|---|---|---|
| 7:5 | 000 | Reserved |
| 4 | 0 | Interrupt Clear Bit<br>= Clear interrupt (Non-Sticky) |
| 3:1 | 000 | GPIO capable pin selected as Interrupt<br>000 No GPIO selected<br>001 = GPIO 0 / MUSIC_REQ<br>010 = GPIO 1 / SPDIF<br>011 = GPIO 2 / DMIC_CLK (Port C)<br>100 = GPIO 3 / DMIC_DAT (Port C)<br>101 = GPIO 4 / DMIC_CLK (Port H)<br>110 = GPIO 5 / DMIC_DAT (Port H)<br>111 = Reserved |
| 0 | 0 | Interrupt Polarity<br>0= Active high interrupt<br>1= Active low interrupt |

### 24.2.6.25 I2SPCM_CTRL_REG6A register

Get Response - HD_A [27:0] = 0x1BC1800 Set - HD_A[27:8] = 0x1B418 I2C = 6E18

| Bit | Default | Description |
|---|---|---|
| 7:5 | 0 | RX Pause cycles |
| 4:0 | 0 | Reserved |

### 24.2.6.26 I2SPCM_CTRL_REG6B register

Get Response - HD_A [27:0] = 0x1BC1900 Set - HD_A[27:8] = 0x1B419 I2C = 6E19

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | RX Pause start position |

### 24.2.6.27 I2SPCM_CTRL_REG6C register

Get Response - HD_A [27:0] = 0x1BC1A00 Set - HD_A[27:8] = 0x1B41A I2C = 6E1A

| Bit | Default | Description |
|---|---|---|
| 7:5 | 0 | TX Pause cycles |
| 4:0 | 0 | Reserved |

### 24.2.6.28 I2SPCM_CTRL_REG6D register

Get Response - HD_A [27:0] = 0x1BC1B Set - HD_A[27:8] = 0x1B41B I2C = 6E1B

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | TX Pause start position |

### 24.2.6.29 I2C Unsolicited Interrupt Status0 register

Get Response - HD_A [27:0] = 0x1BC1C00 Set - HD_A[27:8] = 0x1B41C I2C = 6E1C

| Bit | Default | Description |
|---|---|---|
| 7:5 | 000 | Reserved |
| 4 | 0 | Read Port F unsol interrupt status, writing “1” will clear status |
| 3 | 0 | Read Port E unsol interrupt status, writing “1” will clear status |
| 2 | 0 | Read Port D unsol interrupt status, writing “1” will clear status |
| 1 | 0 | Read Port B unsol interrupt status, writing “1” will clear status |
| 0 | 0 | Read Port A unsol interrupt status, writing “1” will clear status |

### 24.2.6.30 I2C Unsolicited Interrupt Status1 register

Get Response - HD_A [27:0] = 0x1BC1D00 Set - HD_A[27:8] = 0x1B41D I2C = 6E1D

| Bit | Default | Description |
|---|---|---|
| 7 | | Reserved |
| 6 | 0 | Read GPIO6 unsol interrupt status, writing “1” will clear status |
| 5 | 0 | Read GPIO5 unsol interrupt status, writing “1” will clear status |
| 4 | 0 | Read GPIO4 unsol interrupt status, writing “1” will clear status |
| 3 | 0 | Read GPIO3 unsol interrupt status, writing “1” will clear status |
| 2 | 0 | Read GPIO2 unsol interrupt status, writing “1” will clear status |
| 1 | 0 | Read GPIO1 interrupt status, writing “1” will clear status |
| 0 | 0 | Read GPIO0 interrupt status, writing “1” will clear status |

### 24.2.6.31 I2SPCM_CTRL_REG7A register (New)

Get Response - HD_A [27:0] = 0x1BC2000 Set - HD_A[27:8] = 0x1B420 I2C = 6E20

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Reserved |
| 6 | 0 | Enable transmit on channel 8. |
| 5 | 0 | Enable transmit on channel 7. |
| 4:0 | 00000 | TX_SLOT7 – First slot number for transmit channel-7. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: if transmit sample size is 16 or 24, it uses two or three consecutive slots, respectively. |

### 24.2.6.32 I2SPCM_CTRL_REG7B register (New)

Get Response - HD_A [27:0] = 0x1BC2100 Set - HD_A[27:8] = 0x1B421 I2C = 6E21

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Reserved |
| 6 | 0 | Enable transmit on channel 10. |

| | | |
|---|---|---|
| 5 | 0 | Enable transmit on channel 9. |
| 4:0 | 00000 | TX_SLOT8 – First slot number for transmit channel-8. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: if transmit sample size is 16 or 24, it uses two or three consecutive slots, respectively. |

### 24.2.6.33 I2SPCM_CTRL_REG7C register (New)

Get Response - HD_A [27:0] = 0x1BC2200 Set - HD_A[27:8] = 0x1B422 I2C = 6E22

| Bit | Default | Description |
|---|---|---|
| 7:5 | 000 | Reserved |
| 4:0 | 00000 | TX_SLOT9 – First slot number for transmit channel-9. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: if transmit sample size is 16 or 24, it uses two or three consecutive slots, respectively. |

### 24.2.6.34 I2SPCM_CTRL_REG7D register (New)

Get Response - HD_A [27:0] = 0x1BC2300 Set - HD_A[27:8] = 0x1B423 I2C = 6E23

| Bit | Default | Description |
|---|---|---|
| 7:5 | 000 | Reserved |
| 4:0 | 00000 | TX_SLOT10 – First slot number for transmit channel-10. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: if transmit sample size is 16 or 24, it uses two or three consecutive slots, respectively. |

### 24.2.6.35 I2SPCM_CTRL_REG8A register (New)

Get Response - HD_A [27:0] = 0x1BC2400 Set - HD_A[27:8] = 0x1B424 I2C = 6E24

| Bit | Default | Description |
|---|---|---|
| 7:5 | 000 | TX_SLOT12[2:0] – First slot number for transmit channel-12. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: if transmit sample size is 16 or 24, it uses two or three consecutive slots, respectively. |
| 4:0 | 00000 | TX_SLOT11 – First slot number for transmit channel-11. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: if transmit sample size is 16 or 24, it uses two or three consecutive slots, respectively. |

### 24.2.6.36 I2SPCM_CTRL_REG8B register (New)

Get Response - HD_A [27:0] = 0x1BC2500 Set - HD_A[27:8] = 0x1B425 I2C = 6E25

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | TX_SLOT14[0] – First slot number for transmit channel-14. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: if transmit sample size is 16 or 24, it uses two or three consecutive slots, respectively. |
| 6:2 | 00000 | TX_SLOT13 – First slot number for transmit channel-12. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: if transmit sample size is 16 or 24, it uses two or three consecutive slots, respectively. |
| 1:0 | 00 | TX_SLOT12[4:3] – First slot number for transmit channel-12. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: if transmit sample size is 16 or 24, it uses two or three consecutive slots, respectively. |

### 24.2.6.37 I2SPCM_CTRL_REG8C register (New)

Get Response - HD_A [27:0] = 0x1BC2600 Set - HD_A[27:8] = 0x1B426 I2C = 6E26

| Bit | Default | Description |
|---|---|---|
| 7:4 | 0000 | TX_SLOT15[3:0] – First slot number for transmit channel-15. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: if transmit sample size is 16 or 24, it uses two or three consecutive slots, respectively. |
| 3:0 | 0000 | TX_SLOT14[4:1] – First slot number for transmit channel-14. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: if transmit sample size is 16 or 24, it uses two or three consecutive slots, respectively. |

### 24.2.6.38 I2SPCM_CTRL_REG8D register (New)

Get Response - HD_A [27:0] = 0x1BC2700 Set - HD_A[27:8] = 0x1B427 I2C = 6E27

| Bit | Default | Description |
|---|---|---|
| 7:6 | 00 | TX_SLOT17[1:0] – First slot number for transmit channel-17. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: if transmit sample size is 16 or 24, it uses two or three consecutive slots, respectively. |
| 5:1 | 00000 | TX_SLOT16 – First slot number for transmit channel-16. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: if transmit sample size is 16 or 24, it uses two or three consecutive slots, respectively. |
| 0 | 0 | TX_SLOT15[4] – First slot number for transmit channel-15. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: if transmit sample size is 16 or 24, it uses two or three consecutive slots, respectively. |

### 24.2.6.39 I2SPCM_CTRL_REG9A register (New)

Get - HD_A [27:0] = 0x1BC2800 Set - HD_A[27:8] = 0x1B428 I2C = 6E28

| Bit | Default | Description |
|---|---|---|
| 7:3 | 00000 | TX_SLOT18 – First slot number for transmit channel-18. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: if transmit sample size is 16 or 24, it uses two or three consecutive slots, respectively. |
| 2:0 | 000 | TX_SLOT17[4:2] – First slot number for transmit channel-17. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: if transmit sample size is 16 or 24, it uses two or three consecutive slots, respectively. |

### 24.2.6.40 I2SPCM_CTRL_REG9B register (New)

Get Response - HD_A [27:0] = 0x1BC2900 Set - HD_A[27:8] = 0x1B429 I2C = 6E29

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Enable transmit on channel 18. |
| 6 | 0 | Enable transmit on channel 17. |
| 5 | 0 | Enable transmit on channel 16. |
| 4 | 0 | Enable transmit on channel 15. |
| 3 | 0 | Enable transmit on channel 14. |
| 2 | 0 | Enable transmit on channel 13. |
| 1 | 0 | Enable transmit on channel 12. |
| 0 | 0 | Enable transmit on channel 11. |

### 24.2.6.41 I2SPCM_CTRL_REG9C register (New)

Get Response - HD_A [27:0] = 0x1BC2A00 Set - HD_A[27:8] = 01B42A I2C = 6E2A

| Bit | Default | Description |
|---|---|---|
| 7:0 | 00000000 | Reserved |

### 24.2.6.42 I2SPCM_CTRL_REG9D register (New)

Get Response - HD_A [27:0] = 0x1BC2B00 Set - HD_A[27:8] = 0x1B42B I2C = 6E2B

| Bit | Default | Description |
|---|---|---|
| 7:0 | 000000000 | Reserved |

### 24.2.6.43 I2SPCM_CTRL_REG10A register (New)

Get Response - HD_A [27:0] = 0x1BC2C00 Set - HD_A[27:8] = 0x1B42C I2C = 6E2C

| Bit | Default | Description |
|---|---|---|
| 7:5 | 000 | RX_SLOT8[2:0] – First slot number for transmit channel-8. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: if transmit sample size is 16 or 24, it uses two or three consecutive slots, respectively. |
| 4:0 | 00000 | RX_SLOT_7 - First Slot number for receive channel-7. Each slot is 8 bits. All the enabled receive channels should be configured with unique slot numbers.<br>Note: If receive sample size is 16 or 24, it uses two or three consecutive slots respectively. |

### 24.2.6.44 I2SPCM_CTRL_REG10B register (New)

Get Response - HD_A [27:0] = 0x1BC2D00 Set - HD_A[27:8] = 0x1B42D I2C = 6E2D

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | RX_SLOT_10[0] - First Slot number for receive channel-10. Each slot is 8 bits. All the enabled receive channels should be configured with unique slot numbers.<br>Note: If receive sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 6:2 | 00000 | RX_SLOT_9 - First Slot number for receive channel-9. Each slot is 8 bits. All the enabled receive channels should be configured with unique slot numbers.<br>Note: If receive sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 1:0 | 00 | RX_SLOT_8[4:3] - First Slot number for receive channel-8. Each slot is 8 bits. All the enabled receive channels should be configured with unique slot numbers.<br>Note: If receive sample size is 16 or 24, it uses two or three consecutive slots respectively. |

### 24.2.6.45 I2SPCM_CTRL_REG10C register (New)

Get Response - HD_A [27:0] = 0x1BC2E00 Set - HD_A[27:8] = 0x1B42E I2C = 6E2E

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Enable receiver on channel 10. |
| 6 | 0 | Enable receiver on channel 9. |
| 5 | 0 | Enable receiver on channel 8. |
| 4 | 0 | Enable receiver on channel 7. |
| 3:0 | 0000 | RX_SLOT_10[4:1] - First Slot number for receive channel-10. Each slot is 8 bits. All the enabled receive channels should be configured with unique slot numbers.<br>Note: If receive sample size is 16 or 24, it uses two or three consecutive slots respectively. |

### 24.2.6.46 I2SPCM_CTRL_REG10D register (New)

Get Response - HD_A [27:0] = 0x1BC2F00 Set - HD_A[27:8] = 0x1B42F I2C = 6E2F

| Bit | Default | Description |
|---|---|---|
| 7:0 | 00000000 | Reserved |

### 24.2.6.47 HI2S_EXT_REG1 register (New)

Get Response - HD_A [27:0] = 0x1BC3000 Set - HD_A[27:8] = 0x1B430 I2C = 6E30

| Bit | Default | Description |
|---|---|---|
| 7:6 | 00 | TX_FRM_LEN[6:5] - Transmit frame length in terms of bit clock cycles, when in master mode. It is (N+1)*8<br>7'h0 – 8 Clocks per Frame<br>7'h1 – 16 Clocks per Frame<br>7'h2 – 24 Clocks per Frame<br>7'h3 – 32 Clocks per Frame<br>…<br>7'h7f – 1024 Clocks per Frame |
| 5:4 | 00 | TX_WS_WID[8:7] - Transmit word strobe (sync/LRCK) high phase width, in terms of bit clock cycles, when in master mode.<br>9'h0 – 1 Cycle<br>9'h1 – 2 Cycles<br>…<br>9'h1ff – 512 Cycles |
| 3:2 | 00 | RX_FRM_LEN[6:5] - Receive frame length in terms of bit clock cycles, when in master mode. It is (N+1)*8<br>7'h0 – 8 Clocks per Frame<br>7'h1 – 16 Clocks per Frame<br>7'h2 – 24 Clocks per Frame<br>7'h3 – 32 Clocks per Frame<br>…<br>7'h7f – 1024 Clocks per Frame |
| 1:0 | 00 | RX_WS_WID[8:7] – Receive word strobe (sync/LRCK) high phase width, in terms of bit clock cycles, when in master mode.<br>9'h0 – 1 Cycle<br>9'h1 – 2 Cycles<br>…<br>9'h1ff – 512 Cycles |

### 24.2.6.48 HI2S_EXT_REG2 register (New)

Get Response - HD_A [27:0] = 0x1BC3100 Set - HD_A[27:8] = 0x1B431 I2C = 6E31

| Bit | Default | Description |
|---|---|---|
| 7:6 | 00 | RX_SLOT_4[6:5] - First Slot number for transmit channel-4. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 5:4 | 00 | RX_SLOT_3[6:5] - First Slot number for transmit channel-3. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 3:2 | 00 | RX_SLOT_2[6:5] - First Slot number for transmit channel-2. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 1:0 | 00 | RX_SLOT_1[6:5] - First Slot number for transmit channel-1. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |

### 24.2.6.49 HI2S_EXT_REG3 register (New)

Get Response - HD_A [27:0] = 0x1BC3200 Set - HD_A[27:8] = 0x1B432 I2C = 6E32

| Bit | Default | Description |
|---|---|---|
| 7:6 | 00 | RX_SLOT_8[6:5] - First Slot number for transmit channel-8. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 5:4 | 00 | RX_SLOT_7[6:5] - First Slot number for transmit channel-7. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 3:2 | 00 | RX_SLOT_6[6:5] - First Slot number for transmit channel-6. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 1:0 | 00 | RX_SLOT_5[6:5] - First Slot number for transmit channel-5. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |

### 24.2.6.50 HI2S_EXT_REG4 register (New)

Get Response - HD_A [27:0] = 0x1BC3300 Set - HD_A[27:8] = 0x1B433 I2C = 6E33

| Bit | Default | Description |
|---|---|---|
| 7:6 | 00 | Reserved |
| 5:4 | 00 | RX Pause start position[9:8] |
| 3:2 | 00 | RX_SLOT_10[6:5] - First Slot number for transmit channel-10. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 1:0 | 00 | RX_SLOT_9[6:5] - First Slot number for transmit channel-9. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |

### 24.2.6.51 HI2S_EXT_REG5 register (New)

Get Response - HD_A [27:0] = 0x1BC3400 Set - HD_A[27:8] = 0x1B434 I2C = 6E34

| Bit | Default | Description |
|---|---|---|
| 7:6 | 00 | TX_SLOT_4[6:5] - First Slot number for transmit channel-4. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively |
| 5:4 | 00 | TX_SLOT_3[6:5] - First Slot number for transmit channel-3. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 3:2 | 00 | TX_SLOT_2[6:5] - First Slot number for transmit channel-2. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 1:0 | 00 | TX_SLOT_1[6:5] - First Slot number for transmit channel-1. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |

### 24.2.6.52 HI2S_EXT_REG6 register (New)

Get Response - HD_A [27:0] = 0x1BC3500 Set - HD_A[27:8] = 0x1B435 I2C = 6E35

| Bit | Default | Description |
|---|---|---|
| 7:6 | 00 | TX_SLOT_8[6:5] - First Slot number for transmit channel-8. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively |
| 5:4 | 00 | TX_SLOT_7[6:5] - First Slot number for transmit channel-7. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 3:2 | 00 | TX_SLOT_6[6:5] - First Slot number for transmit channel-6. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 1:0 | 00 | TX_SLOT_5[6:5] - First Slot number for transmit channel-5. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |

### 24.2.6.53 HI2S_EXT_REG7 register (New)

Get Response - HD_A [27:0] = 0x1BC3600 Set - HD_A[27:8] = 0x1B436 I2C = 6E36

| Bit | Default | Description |
|---|---|---|
| 7:6 | 00 | TX_SLOT_12[6:5] - First Slot number for transmit channel-12. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively |
| 5:4 | 00 | TX_SLOT_11[6:5] - First Slot number for transmit channel-11. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 3:2 | 00 | TX_SLOT_10[6:5] - First Slot number for transmit channel-10. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 1:0 | 00 | TX_SLOT_9[6:5] - First Slot number for transmit channel-9. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |

### 24.2.6.54 HI2S_EXT_REG8 register (New)

Get Response - HD_A [27:0] = 0x1BC3700 Set - HD_A[27:8] = 0x1B437 I2C = 6E37

| Bit | Default | Description |
|---|---|---|
| 7:6 | 00 | TX_SLOT_16[6:5] - First Slot number for transmit channel-16. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively |
| 5:4 | 00 | TX_SLOT_15[6:5] - First Slot number for transmit channel-15. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 3:2 | 00 | TX_SLOT_14[6:5] - First Slot number for transmit channel-14. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 1:0 | 00 | TX_SLOT_13[6:5] - First Slot number for transmit channel-13. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |

### 24.2.6.55 HI2S_EXT_REG9 register (New)

Get Response - HD_A [27:0] = 0x1BC3800 Set - HD_A[27:8] = 0x1B438 I2C = 6E38

| Bit | Default | Description |
|---|---|---|
| 7:6 | 00 | Reserved |
| 5:4 | 00 | TX Pause start position[9:8] |
| 3:2 | 00 | TX_SLOT_18[6:5] - First Slot number for transmit channel-18. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |
| 1:0 | 00 | TX_SLOT_17[6:5] - First Slot number for transmit channel-17. Each slot is 8 bits. All the enabled transmit channels should be configured with unique slot numbers.<br>Note: If transmit sample size is 16 or 24, it uses two or three consecutive slots respectively. |

## 24.2.7 Peripheral I2SPCM Registers

### 24.2.7.1 PERIPHERAL_I2S_A_REG1 register

Get Response - HD_A [27:0] = 0x1BE0000 Set - HD_A[27:8] = 0x1B600 I2C = 6F00

| Bit | Default | Description |
|---|---|---|
| 7:0 | 3F | rxtx_clk_frames - Sets the frame length of WS signal for RX/TX in terms of bit clock cycles.<br>8'd0: 1 clock<br>8'd1: 2 clocks<br>etc…<br>This field is effective only in master mode.<br>This field is effective only after 'rxtx_lrck_en' is activated. |

### 24.2.7.2 PERIPHERAL_I2S_A_REG2 register

Get Response - HD_A [27:0] = 0x1BE0100 Set - HD_A[27:8] = 0x1B601 I2C = 6F01

| Bit | Default | Description |
|---|---|---|
| 7:0 | 1F | rxtx_sync_lng - Sets the high phase width of WS signal for RX/TX in terms of bit clock cycles.<br>8'd0: 1 clock<br>8'd1: 2 clocks<br>etc…<br>This field is effective only in master mode.<br>This field is effective only after 'rxtx_lrck_en' is activated. |

### 24.2.7.3 PERIPHERAL_I2S_A_REG3 register

Get Response - HD_A [27:0] = 0x1BE0200 Set - HD_A[27:8] = 0x1B602 I2C = 6F02

| Bit | Default | Description |
|---|---|---|
| 7:6 | 0 | Reserved |
| 5 | 1 | **rxtx_on_negedge** - Sets the clock edge for RX/TX data, and generating WS signal.<br>1'b0: running on positive edge of BCLK<br>1'b1: running on negative edge of BCLK<br>This field is applied to reversing BCLK in clock generation block. |
| 4 | 1 | **i2s_master** - Sets if running in master or slave mode for all the RX/TX channels.<br>1'b0: running as slave<br>1'b1: running as master<br>This field is applied to selection of BCLK and WS signal in clock generation block. |
| 3:2 | 0 | Reserved |

| | | |
|---|---|---|
| 1 | 1 | **rxtx_lrck_en** - Enables to generate WS signal for RX/TX in master mode.<br>Active high. |
| 0 | 0 | **rxtx_lrck_low_first** - This field can determine if a high level pulse or low level pulse is generated firstly for RX/TX. This field provided a flexibility for the first frame.<br>1'b0: a high level pulse come out firstly<br>1'b1: a low level pulse come out firstly<br>This field is effective only after 'rxtx_lrck_en' is activated. |

### 24.2.7.4 PERIPHERAL_I2S_A_REG4 register

Get Response - HD_A [27:0] = 0x1BE0300 Set - HD_A[27:8] = 0x1B603 I2C = 6F03

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Reserved |
| 6 | 1 | **rx_lrck_pol** - The polarity of WS signal for receiver can be reversed or not based on different mode.<br>1'b0: keep WS polarity<br>1'b1: reverse WS polarity |
| 5 | 0 | **Reserved** |
| 4 | 0 | **rx_order**- Sets which channel is received firstly for the first frame.<br>1'b0: 'left then right' order<br>1'b1: 'right then left' order<br>This field is effective only after 'rxN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |
| 3 | 0 | **rx_dshift_sel** - Sets if little-endian or big-endian is active.<br>1'b0: MSB first on RX<br>1'b1: LSB first on RX<br>This field is effective only after 'rxN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |
| 2 | 1 | **rx_dstart_dly - Sets 1-bit delay for RX channels.**<br>**1'b0: no delay**<br>**1'b1: delays one clock cycle**<br>**This field is only for standard I2S (Philips) mode.**<br>**This field is effective only after 'rxN_en' is activated. (N=1, 2, ..., depending on which channel it is.)** |
| 1 | 0 | **Reserved** |
| 0 | 0 | **rx_mode** - I2S mode and PCM mode have different mechanism with WS signal. The I2S mode is based on the edge detect of WS signal, while the PCM mode is based on a slot number.<br>1'b0: I2S mode<br>1'b1: PCM mode<br>This field is effective only after 'rxN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |

### 24.2.7.5 PERIPHERAL_I2S_A_REG5 register

Get Response - HD_A [27:0] = 0x1BE0400 Set - HD_A[27:8] = 0x1B604 I2C = 6F04

| Bit | Default | Description |
|---|---|---|
| 7:5 | 0 | Reserved |
| 4:0 | 0x17 | **rx_sa_wdt** - Sets the sample data width of transmitter for all the RX channels.<br>5'd0: 1 bit data<br>5'd1: 2 bit data<br>etc...<br>This field is effective only after 'rxN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |

### 24.2.7.6 PERIPHERAL_I2S_A_REG6 register

Get Response - HD_A [27:0] = 0x1BE0500 Set - HD_A[27:8] = 0x1B605 I2C = 6F05

| Bit | Default | Description |
|---|---|---|

| | | |
|---|---|---|
| 7:0 | 0 | Reserved |

### 24.2.7.7 PERIPHERAL_I2S_A_REG7 register

Get Response - HD_A [27:0] = 0x1BE0600 Set - HD_A[27:8] = 0x1B606 I2C = 6F06

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | **rx1_slot** - When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel1.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel1.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc…<br>This field is effective only after 'rx1_en' is activated. |

### 24.2.7.8 PERIPHERAL_I2S_A_REG8 register

Get Response - HD_A [27:0] = 0x1BE0700 Set - HD_A[27:8] = 0x1B607 I2C = 6F07

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | **rx2_slot** - When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel2.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel2.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc…<br>This field is effective only after 'rx2_en' is activated. |

### 24.2.7.9 PERIPHERAL_I2S_A_REG9 register

Get Response - HD_A [27:0] = 0x1BE0800 Set - HD_A[27:8] = 0x1B608 I2C = 6F08

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | **rx3_slot** - When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel3.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel3.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc…<br>This field is effective only after 'rx3_en' is activated. |

### 24.2.7.10 PERIPHERAL_I2S_A_REG10 register

Get Response - HD_A [27:0] = 0x1BE0900 Set - HD_A[27:8] = 0x1B609 I2C = 6F09

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | **rx4_slot** - When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel4.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel4.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc…<br>This field is effective only after 'rx4_en' is activated. |

### 24.2.7.11 PERIPHERAL_I2S_A_REG11 register

Get Response - HD_A [27:0] = 0x1BE0A00 Set - HD_A[27:8] = 0x1B60A I2C = 6F0A

| Bit | Default | Description |
|---|---|---|

| | | |
|---|---|---|
| 7:0 | 0 | **rx5_slot** - When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel5.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel5.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc…<br>This field is effective only after 'rx5_en' is activated. |

### 24.2.7.12 PERIPHERAL_I2S_A_REG12 register

Get Response - HD_A [27:0] = 0x1BE0B00 Set - HD_A[27:8] = 0x1B60B I2C = 6F0B

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | **rx6_slot** - When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel6.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel6.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc…<br>This field is effective only after 'rx6_en' is activated. |

### 24.2.7.13 PERIPHERAL_I2S_A_REG13 register

Get Response - HD_A [27:0] = 0x1BE0C00 Set - HD_A[27:8] = 0x1B60C I2C = 6F0C

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | **Reserved** |
| 6 | 0 | **rx_mute** - Mute the I2S or PCM Receiver (all zeros will be send). |
| 5 | 0 | **rx6_en** - Enables to receive data for RX channel6.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel6 must be steady before rx6_en is active. And rx6_en must be inactive before reconfiguration. |
| 4 | 0 | **rx5_en** - Enables to receive data for RX channel5.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel5 must be steady before rx5_en is active. And rx5_en must be inactive before reconfiguration. |
| 3 | 0 | **rx4_en** - Enables to receive data for RX channel4.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel4 must be steady before rx4_en is active. And rx4_en must be inactive before reconfiguration. |
| 2 | 0 | **rx3_en** - Enables to receive data for RX channel3.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel3 must be steady before rx3_en is active. And rx3_en must be inactive before reconfiguration. |
| 1 | 0 | **rx2_en** - Enables to receive data for RX channel2.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel2 must be steady before rx2_en is active. And rx2_en must be inactive before reconfiguration. |
| 0 | 0 | **rx1_en** - Enables to receive data for RX channel1.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel1 must be steady before rx1_en is active. And rx1_en must be inactive before reconfiguration. |

### 24.2.7.14 PERIPHERAL_I2S_A_REG14 register

Get Response - HD_A [27:0] = 0x1BE0D00 Set - HD_A[27:8] = 0x1B60D I2C = 6F0D

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | **Reserved** |

| | | |
|---|---|---|
| 6 | 1 | **tx_lrck_pol** - The polarity of WS signal for transmitter can be reversed or not based on different mode.<br>1'b0: keep WS polarity<br>1'b1: reverse WS polarity |
| 5 | 0 | **Reserved** |
| 4 | 0 | **tx_order** - Sets which channel is transmitted firstly for the first frame.<br>1'b0: 'left then right' order<br>1'b1: 'right then left' order<br>This field is effective only after 'txN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |
| 3 | 0 | **tx_dshift_sel** - Sets if little-endian or big-endian is active.<br>1'b0: MSB first on TX<br>1'b1: LSB first on TX<br>This field is effective only after 'txN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |
| 2 | 1 | **tx_dstart_dly** - Sets 1-bit delay for TX channels.<br>1'b0: no delay<br>1'b1: delays one clock cycle<br>This field is only for standard I2S (Philips) mode.<br>This field is effective only after 'txN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |
| 1 | 0 | **Reserved** |
| 0 | 0 | **tx_mode** - I2S mode and PCM mode have different mechanism with WS signal. The I2S mode is based on the edge detect of WS signal, while the PCM mode is based on a slot number.<br>1'b0: I2S mode<br>1'b1: PCM mode<br>This field is effective only after 'txN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |

### 24.2.7.15 PERIPHERAL_I2S_A_REG15 register

Get Response - HD_A [27:0] = 0x1BE0E00 Set - HD_A[27:8] = 0x1B60E I2C = 6F0E

| Bit | Default | Description |
|---|---|---|
| 7:6 | 00 | **Reserved** |
| 5 | 0 | **tx_out_en** - The data output signal will be tri-stated only under TDM mode.<br>1'b0: the out data is not tri-stated when not driven<br>1'b1: the out data is tri-stated when not driven |
| 4:0 | 0x17 | **Tx_sa_wdt** - Sets the sample data width of transmitter for TX channel1.<br>5'd0: 1-bit data<br>5'd1: 2-bit data<br>etc. ...<br>This field is effective only after 'tx1_en' is activated. |

### 24.2.7.16 PERIPHERAL_I2S_A_REG16 register

Get Response - HD_A [27:0] = 0x1BE0F00 Set - HD_A[27:8] = 0x1B60F I2C = 6F0F

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0x00 | **Reserved** |

### 24.2.7.17 PERIPHERAL_I2S_A_REG17 register

Get Response - HD_A [27:0] = 0x1BE1000 Set - HD_A[27:8] = 0x1B610 I2C = 6F10

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | **tx1_slot** - When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for TX channel1.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for TX channel1. |

| | | |
|---|---|---|
| | | 8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc. …<br>This field is effective only after 'tx1_en' is activated. |

### 24.2.7.18 PERIPHERAL_I2S_A_REG18 register

Get Response - HD_A [27:0] = 0x1BE1100 Set - HD_A[27:8] = 0x1B611 I2C = 6F11

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0x00 | **Tx2_slot** - When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for TX channel2.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for TX channel2.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc. …<br>This field is effective only after 'tx2_en' is activated. |

### 24.2.7.19 PERIPHERAL_I2S_A_REG19 register

Get Response - HD_A [27:0] = 0x1BE1200 Set - HD_A[27:8] = 0x1B612 I2C = 6F12

| Bit | Default | Description |
|---|---|---|
| 7:3 | 00000 | **Reserved** |
| 2 | 0 | **tx_mute** - Sets to mute all the TX channels (all zeros will be send). |
| 1 | 1 | **tx2_en** - Enables to transmit data for TX channel2.<br>Active high.<br>[Notice] All of TX configuration fields related with TX channel2 must be steady before tx2_en is active. And tx2_en must be inactive before reconfiguration. |
| 0 | 1 | **tx1_en** - Enables to transmit data for TX channel1.<br>Active high.<br>[Notice] All of TX configuration fields related with TX channel1 must be steady before tx1_en is active. And tx1_en must be inactive before reconfiguration. |

### 24.2.7.20 PERIPHERAL_I2S_A_REG20 register

Get Response - HD_A [27:0] = 0x1BE1300 Set - HD_A[27:8] = 0x1B613 I2C = 6F13

| Bit | Default | Description |
|---|---|---|
| 7:6 | 00 | **DRVCTL_dout** - Establishes the output drive strength for data out of peripheral I2S #1.<br>Encoded as:<br>00: 1.8mA<br>01: 3.6mA<br>10: 5.4mA<br>11: 7.2mA |
| 5:4 | 00 | **DRVCTL_lrclk** - Establishes the output drive strength for LRCK of peripheral I2S #1.<br>Encoded as:<br>00: 1.8mA<br>01: 3.6mA<br>10: 5.4mA<br>11: 7.2mA |
| 3:2 | 00 | **DRVCTL_bclk** - Establishes the output drive strength for bit clock of peripheral I2S #1.<br>Encoded as:<br>00: 1.8mA<br>01: 3.6mA<br>10: 5.4mA<br>11: 7.2mA |
| 1:0 | 00 | **DRVCTL_mclk** - Establishes the output drive strength for master clock of peripheral I2S #1. Encoded as:<br>00: 1.8mA<br>01: 3.6mA |

| | | |
|---|---|---|
| | | 10: 5.4mA<br>11: 7.2mA |

### 24.2.7.21 PERIPHERAL_I2S_B_REG1 register

Get Response - HD_A [27:0] = 0x1BE1400 Set - HD_A[27:8] = 0x1B614 I2C = 6F14

| Bit | Default | Description |
|---|---|---|
| 7:0 | 3F | rxtx_clk_frames - Sets the frame length of WS signal for RX/TX in terms of bit clock cycles.<br>8'd0: 1 clock<br>8'd1: 2 clocks<br>etc…<br>This field is effective only in master mode.<br>This field is effective only after 'rxtx_lrck_en' is activated. |

### 24.2.7.22 PERIPHERAL_I2S_B_REG2 register

Get Response - HD_A [27:0] = 0x1BE1500 Set - HD_A[27:8] = 0x1B615 I2C = 6F15

| Bit | Default | Description |
|---|---|---|
| 7:0 | 1F | rxtx_sync_lng - Sets the high phase width of WS signal for RX/TX in terms of bit clock cycles.<br>8'd0: 1 clock<br>8'd1: 2 clocks<br>etc…<br>This field is effective only in master mode.<br>This field is effective only after 'rxtx_lrck_en' is activated. |

### 24.2.7.23 PERIPHERAL_I2S_B_REG3 register

Get Response - HD_A [27:0] = 0x1BE1600 Set - HD_A[27:8] = 0x1B616 I2C = 6F16

| Bit | Default | Description |
|---|---|---|
| 7:6 | 0 | Reserved |
| 5 | 1 | **rxtx_on_negedge** - Sets the clock edge for RX/TX data, and generating WS signal.<br>1'b0: running on positive edge of BCLK<br>1'b1: running on negative edge of BCLK<br>This field is applied to reversing BCLK in clock generation block. |
| 4 | 1 | **i2s_master** - Sets if running in master or slave mode for all the RX/TX channels.<br>1'b0: running as slave<br>1'b1: running as master<br>This field is applied to selection of BCLK and WS signal in clock generation block. |
| 3:2 | 0 | Reserved |
| 1 | 1 | **rxtx_lrck_en** - Enables to generate WS signal for RX/TX in master mode.<br>Active high. |
| 0 | 0 | **rxtx_lrck_low_first** - This field can determine if a high-level pulse or low level pulse is generated first for RX/TX. This field provided a flexibility for the first frame.<br>1'b0: a high-level pulse comes out first<br>1'b1: a low-level pulse comes out first<br>This field is effective only after 'rxtx_lrck_en' is activated. |

### 24.2.7.24 PERIPHERAL_I2S_B_REG4 register

Get Response - HD_A [27:0] = 0x1BE1700 Set - HD_A[27:8] = 0x1B617 I2C = 6F17

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | Reserved |
| 6 | 1 | **rx_lrck_pol** - The polarity of WS signal for receiver can be reversed or not based on different mode. |

| | | |
|---|---|---|
| | | 1'b0: keep WS polarity<br>1'b1: reverse WS polarity |
| 5 | 0 | **Reserved** |
| 4 | 0 | **rx_order**- Sets which channel is received firstly for the first frame.<br>1'b0: 'left then right' order<br>1'b1: 'right then left' order<br>This field is effective only after 'rxN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |
| 3 | 0 | **rx_dshift_sel** - Sets if little-endian or big-endian is active.<br>1'b0: MSB first on RX<br>1'b1: LSB first on RX<br>This field is effective only after 'rxN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |
| 2 | 1 | **rx_dstart_dly - Sets 1-bit delay for RX channels.**<br>**1'b0: no delay**<br>**1'b1: delays one clock cycle**<br>**This field is only for standard I2S (Philips) mode.**<br>**This field is effective only after 'rxN_en' is activated. (N=1, 2, ..., depending on which channel it is.)** |
| 1 | 0 | **Reserved** |
| 0 | 0 | **rx_mode** - I2S mode and PCM mode have different mechanism with WS signal. The I2S mode is based on the edge detect of WS signal, while the PCM mode is based on a slot number.<br>1'b0: I2S mode<br>1'b1: PCM mode<br>This field is effective only after 'rxN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |

### 24.2.7.25 PERIPHERAL_I2S_B_REG5 register

Get Response - HD_A [27:0] = 0x1BE1800 Set - HD_A[27:8] = 0x1B618 I2C = 6F18

| Bit | Default | Description |
|---|---|---|
| 7:5 | 0 | Reserved |
| 4:0 | 0x17 | **rx_sa_wdt** - Sets the sample data width of transmitter for all the RX channels.<br>5'd0: 1 bit data<br>5'd1: 2 bit data<br>etc...<br>This field is effective only after 'rxN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |

### 24.2.7.26 PERIPHERAL_I2S_B_REG6 register

Get Response - HD_A [27:0] = 0x1BE1900 Set - HD_A[27:8] = 0x1B619 I2C = 6F19

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | Reserved |

### 24.2.7.27 PERIPHERAL_I2S_B_REG7 register

Get Response - HD_A [27:0] = 0x1BE1A00 Set - HD_A[27:8] = 0x1B61A I2C = 6F1A

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | **rx1_slot** - When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel1.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel1.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc...<br>This field is effective only after 'rx1_en' is activated. |

### 24.2.7.28 PERIPHERAL_I2S_B_REG8 register

Get Response - HD_A [27:0] = 0x1BE1B00 Set - HD_A[27:8] = 0x1B61B I2C = 6F1B

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | **rx2_slot** - When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel2.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel2.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc…<br>This field is effective only after 'rx2_en' is activated. |

### 24.2.7.29 PERIPHERAL_I2S_B_REG9 register

Get Response - HD_A [27:0] = 0x1BE1C00 Set - HD_A[27:8] = 0x1B61C I2C = 6F1C

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | **rx3_slot** - When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel3.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel3.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc…<br>This field is effective only after 'rx3_en' is activated. |

### 24.2.7.30 PERIPHERAL_I2S_B_REG10 register

Get Response - HD_A [27:0] = 0x1BE1D00 Set - HD_A[27:8] = 0x1B61D I2C = 6F1D

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | **rx4_slot** - When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel4.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel4.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc…<br>This field is effective only after 'rx4_en' is activated. |

### 24.2.7.31 PERIPHERAL_I2S_B_REG11 register

Get Response - HD_A [27:0] = 0x1BE1E00 Set - HD_A[27:8] = 0x1B61E I2C = 6F1E

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | **rx5_slot** - When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel5.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel5.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc…<br>This field is effective only after 'rx5_en' is activated. |

### 24.2.7.32 PERIPHERAL_I2S_B_REG12 register

Get Response - HD_A [27:0] = 0x1BE1F00 Set - HD_A[27:8] = 0x1B61F I2C = 6F1F

| Bit | Default | Description |
|---|---|---|

| | | |
|---|---|---|
| 7:0 | 0 | **rx6_slot** - When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for RX channel6.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for RX channel6.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc…<br>This field is effective only after 'rx6_en' is activated. |

### 24.2.7.33 PERIPHERAL_I2S_B_REG13 register

Get Response - HD_A [27:0] = 0x1BE2000 Set - HD_A[27:8] = 0x1B620 I2C = 6F20

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | **Reserved** |
| 6 | 0 | **rx_mute** - Mute the I2S or PCM Receiver (all zeros will be send). |
| 5 | 0 | **rx6_en** - Enables to receive data for RX channel6.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel6 must be steady before rx6_en is active. And rx6_en must be inactive before reconfiguration. |
| 4 | 0 | **rx5_en** - Enables to receive data for RX channel5.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel5 must be steady before rx5_en is active. And rx5_en must be inactive before reconfiguration. |
| 3 | 0 | **rx4_en** - Enables to receive data for RX channel4.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel4 must be steady before rx4_en is active. And rx4_en must be inactive before reconfiguration. |
| 2 | 0 | **rx3_en** - Enables to receive data for RX channel3.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel3 must be steady before rx3_en is active. And rx3_en must be inactive before reconfiguration. |
| 1 | 0 | **rx2_en** - Enables to receive data for RX channel2.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel2 must be steady before rx2_en is active. And rx2_en must be inactive before reconfiguration. |
| 0 | 0 | **rx1_en** - Enables to receive data for RX channel1.<br>Active high.<br>[Notice] All of RX configuration fields related with RX channel1 must be steady before rx1_en is active. And rx1_en must be inactive before reconfiguration. |

### 24.2.7.34 PERIPHERAL_I2S_B_REG14 register

Get Response - HD_A [27:0] = 0x1BE2100 Set - HD_A[27:8] = 0x1B621 I2C = 6F21

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | **Reserved** |
| 6 | 1 | **tx_lrck_pol** - The polarity of WS signal for transmitter can be reversed or not based on different mode.<br>1'b0: keep WS polarity<br>1'b1: reverse WS polarity |
| 5 | 0 | **Reserved** |
| 4 | 0 | **tx_order** - Sets which channel is transmitted firstly for the first frame.<br>1'b0: 'left then right' order<br>1'b1: 'right then left' order<br>This field is effective only after 'txN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |
| 3 | 0 | **tx_dshift_sel** - Sets if little-endian or big-endian is active.<br>1'b0: MSB first on TX<br>1'b1: LSB first on TX<br>This field is effective only after 'txN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |

| | | |
|---|---|---|
| 2 | 1 | **tx_dstart_dly** - Sets 1-bit delay for TX channels.<br>1'b0: no delay<br>1'b1: delays one clock cycle<br>This field is only for standard I2S (Philips) mode.<br>This field is effective only after 'txN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |
| 1 | 0 | **Reserved** |
| 0 | 0 | **tx_mode** - I2S mode and PCM mode have different mechanism with WS signal. The I2S mode is based on the edge detect of WS signal, while the PCM mode is based on a slot number.<br>1'b0: I2S mode<br>1'b1: PCM mode<br>This field is effective only after 'txN_en' is activated. (N=1, 2, ..., depending on which channel it is.) |

### 24.2.7.35 PERIPHERAL_I2S_B_REG15 register

Get Response - HD_A [27:0] = 0x1BE2200 Set - HD_A[27:8] = 0x1B622 I2C = 6F22

| Bit | Default | Description |
|---|---|---|
| 7:6 | 00 | **Reserved** |
| 5 | 0 | **tx_out_en** - The data output signal will be tri-stated only under TDM mode.<br>1'b0: the out data is not tri-stated when not driven<br>1'b1: the out data is tri-stated when not driven |
| 4:0 | 0x17 | **Tx_sa_wdt** - Sets the sample data width of transmitter for TX channel1.<br>5'd0: 1-bit data<br>5'd1: 2-bit data<br>etc. …<br>This field is effective only after 'tx1_en' is activated. |

### 24.2.7.36 PERIPHERAL_I2S_B_REG16 register

Get Response - HD_A [27:0] = 0x1BE2300 Set - HD_A[27:8] = 0x1B623 I2C = 6F23

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0x00 | **Reserved** |

### 24.2.7.37 PERIPHERAL_I2S_B_REG17 register

Get Response - HD_A [27:0] = 0x1BE2400 Set - HD_A[27:8] = 0x1B624 I2C = 6F24

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0 | **tx1_slot** - When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for TX channel1.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for TX channel1.<br>8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc. …<br>This field is effective only after 'tx1_en' is activated. |

### 24.2.7.38 PERIPHERAL_I2S_B_REG18 register

Get Response - HD_A [27:0] = 0x1BE2500 Set - HD_A[27:8] = 0x1B625 I2C = 6F25

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0x00 | **Tx2_slot** - When running in I2S mode, sets the delay cycles relative to the edge (positive or negative) of WS signal for TX channel2.<br>When running in PCM mode, sets the delay cycles relative to the positive edge of WS signal for TX channel2. |

| | | |
|---|---|---|
| | | 8'd0: start from the 1st cycle of bclk<br>8'd1: start from the 2nd cycle of bclk<br>etc. ...<br>This field is effective only after 'tx2_en' is activated. |

### 24.2.7.39 PERIPHERAL_I2S_B_REG19 register

Get Response - HD_A [27:0] = 0x1BE2600 Set - HD_A[27:8] = 0x1B626 I2C = 6F26

| Bit | Default | Description |
|---|---|---|
| 7:3 | 00000 | **Reserved** |
| 2 | 0 | **tx_mute** - Sets to mute all the TX channels (all zeros will be send). |
| 1 | 1 | **tx2_en** - Enables to transmit data for TX channel2.<br>Active high.<br>[Notice] All of TX configuration fields related with TX channel2 must be steady before tx2_en is active. And tx2_en must be inactive before reconfiguration. |
| 0 | 1 | **tx1_en** - Enables to transmit data for TX channel1.<br>Active high.<br>[Notice] All of TX configuration fields related with TX channel1 must be steady before tx1_en is active. And tx1_en must be inactive before reconfiguration. |

### 24.2.7.40 PERIPHERAL_I2S_B_REG20 register

Get Response - HD_A [27:0] = 0x1BE2700 Set - HD_A[27:8] = 0x1B627 I2C = 6F27

| Bit | Default | Description |
|---|---|---|
| 7:6 | 00 | **DRVCTL_dout** - Establishes the output drive strength for data out of peripheral I2S #2.<br>Encoded as:<br>00: 1.8mA<br>01: 3.6mA<br>10: 5.4mA<br>11: 7.2mA |
| 5:4 | 00 | **DRVCTL_lrclk** - Establishes the output drive strength for LRCK of peripheral I2S #2.<br>Encoded as:<br>00: 1.8mA<br>01: 3.6mA<br>10: 5.4mA<br>11: 7.2mA |
| 3:2 | 00 | **DRVCTL_bclk** - Establishes the output drive strength for bit clock of peripheral I2S #2.<br>Encoded as:<br>00: 1.8mA<br>01: 3.6mA<br>10: 5.4mA<br>11: 7.2mA |
| 1:0 | 00 | **DRVCTL_mclk** - Establishes the output drive strength for master clock of peripheral I2S #2. Encoded as:<br>00: 1.8mA<br>01: 3.6mA<br>10: 5.4mA<br>11: 7.2mA |

### 24.2.7.41 PERIPHERAL_I2S_A_REG21 register

Get Response - HD_A [27:0] = 0x1BE2800 Set - HD_A[27:8] = 0x1B628 I2C = 6F28

| Bit | Default | Description |
|---|---|---|
| 7 | 1 | **mst_clk_div_en_a** - Enables the clock divider for bit clock of peripheral I2S_A. |
| 6:0 | 0x2F | **mst_clk_div_a** - Sets the division ratio for bit clock of peripheral I2S_A.<br>7'd0: div by 1<br>7'd1: div by 2<br>7'd2: div by 3 |

| | | |
|---|---|---|
| | | Etc. … |

### 24.2.7.42 PERIPHERAL_I2S_A_REG22 register

Get Response - HD_A [27:0] = 0x1BE2900 Set - HD_A[27:8] = 0x1B629 I2C = 6F29

| Bit | Default | Description |
|---|---|---|
| 7 | 1 | **mclk_div_en_a** - Enables the clock divider for master clock of peripheral I2S_A. |
| 6:0 | 0x2F | **mclk_div_a** - Sets the division ratio for master clock of peripheral I2S_A.<br>7'd0: div by 1<br>7'd1: div by 2<br>7'd2: div by 3<br>Etc. … |

### 24.2.7.43 PERIPHERAL_I2S_A_REG23 register

Get Response - HD_A [27:0] = 0x1BE2A00 Set - HD_A[27:8] = 0x1B62A I2C = 6F2A

| Bit | Default | Description |
|---|---|---|
| 7:6 | 01 | **DCCTL_i2s1_dout** - Enables built-in pull-up or pull-down functionality for data out of peripheral I2S_A. Encoded as:<br>00: pull-up and pull-down disabled<br>01: weak pull-down enabled, pull-up disabled<br>10: pull-down disabled, weak pull-down value 1 enabled<br>11: pull-down disabled, weak pull-down value 2 enabled |
| 5:4 | 01 | **DCCTL_i2s1_lrclk** - Enables built-in pull-up or pull-down functionality for LRCK of peripheral I2S_A. Encoded as:<br>00: pull-up and pull-down disabled<br>01: weak pull-down enabled, pull-up disabled<br>10: pull-down disabled, weak pull-down value 1 enabled<br>11: pull-down disabled, weak pull-down value 2 enabled. |
| 3:2 | 01 | **DCCTL_i2s1_bclk** - Enables built-in pull-up or pull-down functionality for bit clock of peripheral I2S_A. Encoded as:<br>00: pull-up and pull-down disabled<br>01: weak pull-down enabled, pull-up disabled<br>10: pull-down disabled, weak pull-down value 1 enabled<br>11: pull-down disabled, weak pull-down value 2 enabled. |
| 1:0 | 01 | **DCCTL_i2s1_mclk** - Enables built-in pull-up or pull-down functionality for master clock of peripheral I2S_A. Encoded as:<br>00: pull-up and pull-down disabled<br>01: weak pull-down enabled, pull-up disabled<br>10: pull-down disabled, weak pull-down value 1 enabled<br>11: pull-down disabled, weak pull-down value 2 enabled. |

### 24.2.7.44 PERIPHERAL_I2S_A_REG24 register

Get Response - HD_A [27:0] = 0x1BE2B00 Set - HD_A[27:8] = 0x1B62B I2C = 6F2B

| Bit | Default | Description |
|---|---|---|
| 7:2 | 0 | **Reserved** |
| 1:0 | 01 | **DCCTL_i2s1_din** - Enables built-in pull-up or pull-down functionality for data in of peripheral I2S_A. Encoded as:<br>00: pull-up and pull-down disabled<br>01: weak pull-down enabled, pull-up disabled<br>10: pull-down disabled, weak pull-down value 1 enabled<br>11: pull-down disabled, weak pull-down value 2 enabled |

### 24.2.7.45 PERIPHERAL_I2S_A_REG25 register

Get Response - HD_A [27:0] = 0x1BE2C00 Set - HD_A[27:8] = 0x1B62C I2C = 6F2C

| Bit | Default | Description |
|---|---|---|

| | | |
|---|---|---|
| 7:4 | 0 | **Reserved** |
| 3 | 0 | **OEN_I2S1_mclk** - Output enable signal for master clock of peripheral I2S_A. |
| 2 | 1 | **mclk_en_a** - Master clock enable signal for peripheral I2S_A. |
| 1 | 1 | **mst_clk_en_a** – Bit clock enable signal in master mode for peripheral I2S_A. |
| 0 | 1 | **bclk_en_a** – Bit clock enable signal for peripheral I2S_A. |

### 24.2.7.46 PERIPHERAL_I2S_B_REG21 register

Get Response - HD_A [27:0] = 0x1BE2D00 Set - HD_A[27:8] = 0x1B62D I2C = 6F2D

| Bit | Default | Description |
|---|---|---|
| 7 | 1 | **mst_clk_div_en_b** - Enables the clock divider for bit clock of peripheral I2S_B. |
| 6:0 | 0x2F | **mst_clk_div_b** - Sets the division ratio for bit clock of peripheral I2S_B.<br>7'd0: div by 1<br>7'd1: div by 2<br>7'd2: div by 3<br>Etc. … |

### 24.2.7.47 PERIPHERAL_I2S_B_REG22 register

Get Response - HD_A [27:0] = 0x1BE2E00 Set - HD_A[27:8] = 0x1B62E I2C = 6F2E

| Bit | Default | Description |
|---|---|---|
| 7 | 1 | **mclk_div_en_b** - Enables the clock divider for master clock of peripheral I2S_B. |
| 6:0 | 0x2F | **mclk_div_b** - Sets the division ratio for master clock of peripheral I2S_B.<br>7'd0: div by 1<br>7'd1: div by 2<br>7'd2: div by 3<br>Etc. … |

### 24.2.7.48 PERIPHERAL_I2S_B_REG23 register

Get Response - HD_A [27:0] = 0x1BE2F00 Set - HD_A[27:8] = 0x1B62F I2C = 6F2F

| Bit | Default | Description |
|---|---|---|
| 7:6 | 01 | **DCCTL_i2s2_dout** - Enables built-in pull-up or pull-down functionality for data out of peripheral I2S_B. Encoded as:<br>00: pull-up and pull-down disabled<br>01: weak pull-down enabled, pull-up disabled<br>10: pull-down disabled, weak pull-down value 1 enabled<br>11: pull-down disabled, weak pull-down value 2 enabled |
| 5:4 | 01 | **DCCTL_i2s2_lrclk** - Enables built-in pull-up or pull-down functionality for LRCK of peripheral I2S_B. Encoded as:<br>00: pull-up and pull-down disabled<br>01: weak pull-down enabled, pull-up disabled<br>10: pull-down disabled, weak pull-down value 1 enabled<br>11: pull-down disabled, weak pull-down value 2 enabled. |
| 3:2 | 01 | **DCCTL_i2s2_bclk** - Enables built-in pull-up or pull-down functionality for bit clock of peripheral I2S_B. Encoded as:<br>00: pull-up and pull-down disabled<br>01: weak pull-down enabled, pull-up disabled<br>10: pull-down disabled, weak pull-down value 1 enabled<br>11: pull-down disabled, weak pull-down value 2 enabled. |
| 1:0 | 01 | **DCCTL_i2s2_mclk** - Enables built-in pull-up or pull-down functionality for master clock of peripheral I2S_B. Encoded as:<br>00: pull-up and pull-down disabled<br>01: weak pull-down enabled, pull-up disabled<br>10: pull-down disabled, weak pull-down value 1 enabled<br>11: pull-down disabled, weak pull-down value 2 enabled. |

### 24.2.7.49 PERIPHERAL_I2S_B_REG24 register

Get Response - HD_A [27:0] = 0x1BE3000 Set - HD_A[27:8] = 0x1B630 I2C = 6F30

| Bit | Default | Description |
|---|---|---|
| 7:2 | 0 | **Reserved** |
| 1:0 | 01 | **DCCTL_i2s2_din** - Enables built-in pull-up or pull-down functionality for data in of peripheral I2S_B. Encoded as:<br>00: pull-up and pull-down disabled<br>01: weak pull-down enabled, pull-up disabled<br>10: pull-down disabled, weak pull-down value 1 enabled<br>11: pull-down disabled, weak pull-down value 2 enabled |

### 24.2.7.50 PERIPHERAL_I2S_B_REG25 register

Get Response - HD_A [27:0] = 0x1BE3100 Set - HD_A[27:8] = 0x1B631 I2C = 6F31

| Bit | Default | Description |
|---|---|---|
| 7:4 | 0 | **Reserved** |
| 3 | 0 | **OEN_I2S2_mclk** - Output enable signal for master clock of peripheral I2S_B. |
| 2 | 1 | **mclk_en_b** - Master clock enable signal for peripheral I2S_B. |
| 1 | 1 | **mst_clk_en_b** – Bit clock enable signal in master mode for peripheral I2S_B. |
| 0 | 1 | **bclk_en_b** – Bit clock enable signal for peripheral I2S_B. |

### 24.2.7.51 PERIPHERAL_I2S_A_REG26 register

Get Response - HD_A [27:0] = 0x1BE3200 Set - HD_A[27:8] = 0x1B632 I2C = 6F32

| Bit | Default | Description |
|---|---|---|
| 7:2 | 0 | **Reserved** |
| 1 | 0 | **pi2s_lb_mode_a** – Enables loop-back from I2S_A data out to I2S_A data in. |
| 0 | 0 | **hda_sw_mode_a** – Enables software mode under HDA operation for I2S_A. This bit is used for the following fields:<br>'rx_sa_wdt_a'<br>'tx_sa_wdt_a'<br>'pi2s_a_mst_clk_div'<br>'i2s1_mclk_div'. |

### 24.2.7.52 PERIPHERAL_I2S_B_REG26 register

Get Response - HD_A [27:0] = 0x1BE3300 Set - HD_A[27:8] = 0x1B633 I2C = 6F33

| Bit | Default | Description |
|---|---|---|
| 7:2 | 0 | **Reserved** |
| 1 | 0 | **pi2s_lb_mode_b** – Enables loop-back from I2S_B data out to I2S_B data in. |
| 0 | 0 | **hda_sw_mode_b** – Enables software mode under HDA operation for I2S_B. This bit is used for the following fields:<br>'rx_sa_wdt_b'<br>'tx_sa_wdt_b'<br>'pi2s_b_mst_clk_div'<br>'i2s2_mclk_div' |

### 24.2.7.53 PERIPHERAL_I2S_SPARE register

Get Response - HD_A [27:0] = 0x1BE3400 Set - HD_A[27:8] = 0x1B634 I2C = 6F34

| Bit | Default | Description |
|---|---|---|

| | | |
|---|---|---|
| 7:0 | 0 | **Reserved** |

### 24.2.7.54 PERIPHERAL_I2S_REG27 register

Get Response - HD_A [27:0] = 0x1BE3500 Set - HD_A[27:8] = 0x1B635 I2C = 6F35

| Bit | Default | Description |
|---|---|---|
| 7:4 | 0 | **Reserved** |
| 3 | 0 | **i2s_mclk_share_b** – Selects the MCLK from PI2S_A<br>This bit is just used for the PI2S_B block.<br>1'b0: use the MCLK from PI2S_B<br>1'b1: use the MCLK from PI2S_A |
| 2 | 0 | **i2s_mclk_share_a** – Selects the MCLK from PI2S_B<br>This bit is just used for the PI2S_A block.<br>1'b0: use the MCLK from PI2S_A<br>1'b1: use the MCLK from PI2S_B |
| 1 | 0 | **i2s_bclk_lrclk_share_b** – Selects the BCLK and LRCLK from PI2S_A<br>This bit is just used for the PI2S_B block.<br>1'b0: use the BCLK and LRCLK from PI2S_B<br>1'b1: use the BCLK and LRCLK from PI2S_A |
| 0 | 0 | **i2s_bclk_lrclk_share_a** – Selects the BCLK and LRCLK from PI2S_B<br>This bit is just used for the PI2S_A block.<br>1'b0: use the BCLK and LRCLK from PI2S_A<br>1'b1: use the BCLK and LRCLK from PI2S_B |

### 24.2.7.55 PERIPHERAL_I2S_REG28 register

Get Response - HD_A [27:0] = 0x1BE3600 Set - HD_A[27:8] = 0x1B636 I2C = 6F36

| Bit | Default | Description |
|---|---|---|
| 7:2 | 0 | **Reserved** |
| 1 | 0 | **Sync_adc34_en** – Enable to synchronize the ADC3 and ADC4<br>When sets, the one which is configured to power up in the end will start both ADCs.<br>When sets, either one which is configured to power down will stop both ADCs.<br>When not set, both ADCs will work independently. |
| 0 | 0 | **Sync_dac45_en** – Enable to synchronize the DAC4 and DAC5<br>When sets, the one which is configured to power up in the end will start both DACs.<br>When sets, either one which is configured to power down will stop both DACs.<br>When not set, both DACs will work independently. |

### 24.2.7.56 IV_PACKING_REG1 register

Get Response - HD_A [27:0] = 0x1BE3800 Set - HD_A[27:8] = 0x1B638 I2C = 6F38

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0x00 | **Reserved** |

### 24.2.7.57 IV_PACKING_REG2 register

Get Response - HD_A [27:0] = 0x1BE3900 Set - HD_A[27:8] = 0x1B639 I2C = 6F39

| Bit | Default | Description |
|---|---|---|
| 7:4 | 0x0 | **dp_mode_1** - Sets the IV de-packing mode for device #2.<br>4'b0000: Falcon mode, 48KHz, 24 bit<br>4'b0001: Falcon mode, 48KHz, 16 bit<br>4'b0010: Falcon mode, 96KHz, 24 bit |

| | | |
|---|---|---|
| | | 4'b0011: Falcon mode, 96KHz, 16 bit<br>4'b0100: Falcon mode, 192KHz, 24 bit<br>4'b0101: Falcon mode, 192KHz, 16 bit<br>4'b1000: Neptune mode, 48KHz, 24 bit<br>4'b1001: Neptune mode, 48KHz, 16 bit<br>4'b1010: Neptune mode, 96KHz, 24 bit<br>4'b1011: Neptune mode, 96KHz, 16 bit<br>4'b1100: Neptune mode, 192KHz, 24 bit<br>4'b1101: Neptune mode, 192KHz, 16 bit<br>Others: reserved. |
| 3:0 | 0x0 | **dp_mode_0** - Sets the IV de-packing mode for device #1.<br>4'b0000: Falcon mode, 48KHz, 24 bit<br>4'b0001: Falcon mode, 48KHz, 16 bit<br>4'b0010: Falcon mode, 96KHz, 24 bit<br>4'b0011: Falcon mode, 96KHz, 16 bit<br>4'b0100: Falcon mode, 192KHz, 24 bit<br>4'b0101: Falcon mode, 192KHz, 16 bit<br>4'b1000: Neptune mode, 48KHz, 24 bit<br>4'b1001: Neptune mode, 48KHz, 16 bit<br>4'b1010: Neptune mode, 96KHz, 24 bit<br>4'b1011: Neptune mode, 96KHz, 16 bit<br>4'b1100: Neptune mode, 192KHz, 24 bit<br>4'b1101: Neptune mode, 192KHz, 16 bit<br>Others: reserved. |

### 24.2.7.58 IV_PACKING_REG3 register

Get Response - HD_A [27:0] = 0x1BE3A00 Set - HD_A[27:8] = 0x1B63A I2C = 6F3A

| Bit | Default | Description |
|---|---|---|
| 7:0 | 0x00 | **Reserved** |

### 24.2.7.59 IV_PACKING_REG4 register

Get Response - HD_A [27:0] = 0x1BE3B00 Set - HD_A[27:8] = 0x1B63B I2C = 6F3B

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | **Reserved** |
| 6:4 | 000 | **rp_mode** - Sets the IV de-packing mode for host.<br>3'b000: 48KHz, 24 bit<br>3'b001: 48KHz, 16 bit (I2S mode only)<br>3'b010: 96KHz, 24 bit<br>3'b011: 96KHz, 16 bit (I2S mode only)<br>3'b100: 192KHz, 24 bit<br>3'b101: 192KHz, 16 bit (I2S mode oly)<br>Others: Reserved |
| 3:1 | 000 | **Reserved** |
| 0 | 0 | **Wfifo_clk_en** – Clock enable signal for wfifo_clk. |

### 24.2.7.60 IV_PACKING_REG5 register

Get Response - HD_A [27:0] = 0x1BE3C00 Set - HD_A[27:8] = 0x1B63C I2C = 6F3C

| Bit | Default | Description |
|---|---|---|
| 7:6 | 00 | **Reserved** |
| 5 | 0 | **switch_i_v_1** - Switches the current and the voltage for device #2. |
| 4 | 0 | **switch_l_r_1** - Switches the left and the right channel for device #2. |
| 3 | 0 | **switch_i_v_0** - Switches the current and the voltage for device #1. |
| 2 | 0 | **switch_l_r_0** - Switches the left and the right channel for device #1. |

| | | |
|---|---|---|
| 1 | 0 | **dp_en_1** – Enables the IV de-packing block for device #2. All the configuration related to device #2 above must be stable before asserting this bit. |
| 0 | 0 | **dp_en_0** – Enables the IV de-packing block for device #1. All the configuration related to device #1 above must be stable before asserting this bit. |

### 24.2.7.61 SRC1_CTRL register

Get Response - HD_A [27:0] = 0x1BA2C00 Set - HD_A[27:8] = 0x1B22C I2C = 6D2C

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | SRC1 enable control. Active high.<br>0: disabled<br>1: enabled |
| 6 | 0 | **Reserved** |
| 5:4 | 00 | Port F sample rate which indicate the ADC3 in peripheral I2S1 side sample rate.<br>2'b00: 48KHz<br>2'b01: 96KHz<br>2'b10: Reserved |
| 3:2 | 0 | **Reserved** |
| 1:0 | 0 | Port E sample rate which indicate the DAC4 in peripheral I2S1 side sample rate.<br>2'b00: 48KHz<br>2'b01: 96KHz<br>2'b10: Reserved<br>2'b11: 192KHz |

### 24.2.7.62 SRC2_CTRL register

Get Response - HD_A [27:0] = 0x1BA2D00 Set - HD_A[27:8] = 0x1B22D I2C = 6D2D

| Bit | Default | Description |
|---|---|---|
| 7 | 0 | SRC2 enable control. Active high.<br>0: disabled<br>1: enabled |
| 6 | 0 | **Reserved** |
| 5:4 | 00 | Port K sample rate which indicate the ADC4 in peripheral I2S2 side sample rate.<br>2'b00: 48KHz<br>2'b01: 96KHz<br>2'b10: Reserved |
| 3:2 | 0 | **Reserved** |
| 1:0 | 0 | Port J sample rate which indicate the DAC5 in peripheral I2S2 side sample rate.<br>2'b00: 48KHz<br>2'b01: 96KHz<br>2'b10: Reserved<br>2'b11: 192KHz |

## 24.3 Vendor 1C Registers

### 24.3.1 Codec Test Registers

#### 24.3.1.1 Codec_Test0 -> Calibration control

Get Response - HD_A[27:0] = 0x1CA0000 Set - HD_A[27:11] = 0x1C20 I2C=7100

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | Enable HP offset calibration for one specific path; port A left or right based on ctest3[11:10] |
| 10 | 0 | Bypass portX offcal en using test bits |
| 9 | 0 | PortA offcal en data when bypass bit is set |
| 8 | 0 | Reserved |
| 7 | 0 | Enable bypass of offcal value with test register value on port A left & right channels with value written on ctest 18 [6:0] |
| 6 | 0 | Enables calibration on audio power states from d3/d2 to d0/d1 |
| 5 | 0 | Data for cpmp_clk_en to initiate the offset cancellation state machine when bypass bit is set through bit 4 of ctest0 |
| 4 | 0 | Bypass charge pump clock enable for offset cancellation logic bypassing automatic start conversion<br>0: Normal mode<br>1: Bypass mode |
| 3 | 0 | Clears calibration enable latch from good cad to allow calibration to occur for the next good CAD |
| 2 | 0 | Reserved – Read as 0 |
| 1 | 0 | Resets the HP Offset calibration engine when set to 1 |
| 0 | 0 | Forces automatic HP offset calibration to be run using this bit.<br>1 – Enable calibration manually |

#### 24.3.1.2 Codec_Test1 -> Digital MIC Control

Get Response - HD_A[27:0] = 0x1CA1000 Set - HD_A[27:11] = 0x1C21 I2C=7104

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | 1-DCFILTER bypass on ADC2<br>0-No bypass |
| 10 | 0 | 1-DCFILTER bypass on ADC1<br>0-No bypass |
| 9 | 0 | 1-Test mode to internally invert the digmic1 clock for latching<br>0-Default latching |
| 8 | 0 | 1-Test mode to internally invert the digmic1 clock for latching<br>0-Default latching |
| 7:6 | 0 | Selects the DigMic2 clock frequency<br>00-3.072Mhz<br>01-1.536Mhz<br>10-768Khz<br>11- Reserved |
| 5:4 | 0 | Selects the DigMic1 clock frequency<br>00-3.072Mhz<br>01-1.536Mhz<br>10-768Khz<br>11- Reserved |
| 3 | 0 | L Shape DMIC feature.<br>1-Enable the swap of right channels on the dmic interfaces<br>0-No swap of right channels on dmic |
| 2 | 0 | 1-The Digmic2 clock to the external mic is inverted in relation with the internal mic data latching.<br>0-No clock inversion. |
| 1 | 0 | 1-The Digmic2 clock is same as Digmic1 clock<br>0- Digmic2 uses dedicated Digmic2 clock irrespective of digmic1 |

| | | |
|---|---|---|
| 0 | 0 | 1-The Digmic1 clock to the external mic is inverted in relation with the internal mic data latching.<br>0-No clock inversion. |

### 24.3.1.3 Codec_Test2 -> BI-QUAD EQ and ADC control

Get Response - HD_A[27:0] = 0x1CA2000 Set - HD_A[27:11] = 0x1C22 I2C=7108

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | 1-EQ memory chip enable test mode |
| 10 | 0 | Reserved |
| 9 | 0 | Reserved |
| 8 | 0 | Reserved |
| 7 | 0 | Swap the Left and Right channels of digital mic2 (Port H) |
| 6 | 0 | Swap the digital mic1 channel. Set this bit as digmic_clk_mod. |
| 5 | 0 | Reserved |
| 4 | 0 | 1-Enable test mode on filter for dmic 768Khz clock on ADC2<br>0-Default operation |
| 3 | 0 | 1-Enable test mode on filter for dmic 768Khz clock on ADC1<br>0-Default operation |
| 2 | 0 | ADC0 analog gain compensation is bypassed<br>0 – Gain compensation is applied<br>1 – No gain compensation |
| 1:0 | 0 | Used to control when the PGA gain comparison needs to be done for ADC0. The gain compare result is used for gain ramping section in the PGA<br>00 – Gain comparison at Fs<br>01 – Gain comparison at Fs/2<br>10 – Gain comparison at Fs/4<br>11 – Gain checking at 4*Fs |

### 24.3.1.4 Codec_Test3 -> Micbias, SPKR Beep/Mute Control

Get Response - HD_A[27:0] = 0x1CA3000 Set - HD_A[27:11] = 0x1C23 I2C=710C

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | Control micbias pin assignment to portB or portC:<br>0 = Port C<br>1 = Port B |
| 10 | 0 | Test data for analog port micbiasc_en during bypass mode (ctest3[6] = 1) |
| 9 | 0 | Test data for analog port micbiase_en during bypass mode (ctest3[6] =1) |
| 8 | 0 | Test data for analog port micbiasd_en during bypass mode (ctest3[6] =1) |
| 7 | 0 | Test data for analog port micbiasb_en during bypass mode (ctest3[6] =1) |
| 6 | 0 | Enable force bypass mode for micbias B, micbias C and micbiasD enables<br>0 = Normal mode<br>1 = Bypass using individual test bits |
| 5 | 0<br>Protected | Codec all done forced value<br>0 = force to 0 (see bit 4)<br>1 = force to 1 (see bit 4) |
| 4 | 0<br>Protected | Codec all done bypass<br>0 = Normal mode, state machine driven<br>1 = Bypass codec all done (force to bit 5 value) |
| 3 | 0<br>Protected | Enables mapped DAC on the port enable<br>0 = Disable<br>1 = Enable |
| 2 | 0<br>Protected | 0 = Spkr un-muted (set codec test15[7])<br>1 = Spkr muted (set codec test15[7]) |
| 1 | 0<br>Protected | 0 = Spkr beep disabled (set codec test15[7])<br>1 = Spkr beep enabled (set codec test15[7]) |
| 0 | 0<br>Protected | 0 = Spkr buffer powered down (set codec test15[7])<br>1 = Spkr buffer powered up (set codec test15[7]) |

### 24.3.1.5 Codec_Test4

Get Response - HD_A[27:0] = 0x1CA4000 Set - HD_A[27:11] = 0x1C24 I2C=7110

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | Reduce beep de-bounce by 1:100 ratio<br>0 = normal<br>1 = Reduced ratio |
| 10 | 1 | Disable DAC half band gain<br>0 = Normal +2.5db gain<br>1 = 0db gain |
| 9 | 0 | DAC3 swap feature in Interpolator.<br>0 – DAC3 left and right channels give out data as received.<br>1 - DAC3 left and right channels data are swapped. |
| 8 | 0 | Signal ramping step size<br>0 = 512<br>1 = 256 |
| 7 | 0 | DAC2 swap feature in Interpolator.<br>0 – DAC2 left and right channels give out data as received.<br>1 - DAC2 left and right channels data are swapped. |
| 6 | 0 | DAC1 swap feature in Interpolator.<br>0 – DAC1 left and right channels give out data as received.<br>1 - DAC1 left and right channels data are swapped. |
| 5 | 0<br>Protected | Disable High Pass filter following Class-D enabled DAC.<br>0 - When Class-D enabled follow Class-D mapped DAC.<br>1 - Follow Manual mode based on Codec_test9 [11] High pass will get assigned to based on Codec_test9 [9:8] or follow Class-D mapping. |
| 4 | 0<br>Protected | DAC High pass filter bypass enable<br>0 = High pass is active<br>1 = High pass is bypassed |
| 3 | 0 | 0 = Enable Signal ramping<br>1 = Disable Signal ramping |
| 2 | 1 | Selects which dac1 / dac2 data can be sent to subwoofer (dac3)<br>1-DAC1 left +DAC1 right data<br>0-DAC1 Left data if dacX_swap is zero<br>0-DAC1 right data if dacX_swap is one |
| 1 | 0 | 0 = PC-Beep pin enabled when reset is high<br>1 = PC-Beep pin disabled when reset is high |
| 0 | 0 | 0 = PC beep takes analog path<br>1 = PC beep takes digital path |

### 24.3.1.6 Codec_Test5

Get Response - HD_A [27:0] = 0x1CA5000 Set - HD_A[27:11] = 0x1C25 I2C=7114

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | IIR memory CEN programmable control. By default, (0) automatic control of CEN of memory from state machine. (1) Always enable memory whenever with biquad is enabled |
| 10 | 0 | DRC Gain bypass for channel right<br>0 = DRC gain is multiplied with eq block output.<br>1 = DRC gain is ignored and eq block 5th stage output is given out to DAC interpolator. |
| 9 | 0 | DRC Gain bypass for channel left<br>0 = DRC gain is multiplied with eq block output.<br>1 = DRC gain is ignored and eq block 5th stage output is given out to DAC interpolator. |
| 8 | 0 | ADC5 FIFO empty select (ECO for I2S IO Timing) |
| 7:4 | 3 | Delay Register to control SPKR mute while in SnM mode (DEDY-17)<br>0---bypass the ECO function, no delay<br>1---50ms<br>2---100ms<br>3---200ms<br>4---400ms<br>5---800ms<br>6---1000ms<br>7---1600ms<br>8---2000ms |
| 3 | 0 | Read Only - indicates the SPKR calibration error for right channel |

| | | |
|---|---|---|
| 2 | 0 | Read Only - indicates the SPKR calibration error for left channel |
| 1 | 1 | ADC2 FIFO empty Select (ECO for I2S IO Timing) |
| 0 | 0 | ADC1 FIFO empty Select (ECO for I2S IO Timing) |

### 24.3.1.7 Codec_Test6 -> Combo Jack, SCDAC2, Refs Control

Get Response - HD_A[27:0] = 0x1CA6000 Set - HD_A[27:11] = 0x1C26 I2C=7118

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | Bypass for combo jack filter enable from Jack sense state machine<br>0 = No bypass Combo Jack filter enable<br>1 = Bypass Combo Jack filter enable |
| 10 | 0 | Combo jack filter enable control when bypass mode ctest6[11] is set to 1 |
| 9 | 0 | Test data for analog port porta_hpmode (controllable when ctest6[8] = 1) |
| 8 | 0 | Test enable for bypassing analog port a hp_mode<br>1 – Bypass port A HP mode<br>0 – No bypass |
| 7 | 0 | 0 = Data scdac3 powered down (set codec_test15[6])<br>1 = Data scdac3 powered up |
| 6 | 0 | 0 = Data scdac2 unmuted (set codec_test15[1])<br>1 = Data scdac2 muted |
| 5 | 0 | 0 = Data scdac2 powered down (set codec_test15[1])<br>1 = Data scdac2 powered up |
| 4 | 0 | 0 = Data scdac1 not muted (set codec_test15[0])<br>1 = Data scdac1 muted |
| 3 | 0 | 0 = Data scdac1 powered down (set codec_test15[0])<br>1 = Data scdac1 powered up |
| 2 | 0 | Test bit to bypass spkrmute from disabling class D<br>0 – No operation<br>1 – Bypass spkrmute pad signal |
| 1 | 0 | 0 = Force value refs powered down (set codec_test6[0])<br>1 = Force value refs powered up |
| 0 | 0 | Force refs power state<br>0 = Refs power state is controlled by state machine<br>1 = Refs power state is controlled by bit 1 |

### 24.3.1.8 Codec_Test7 -> SPKR Mono Control

Get Response - HD_A[27:0] = 0x1CA7000 Set - HD_A[27:11] = 0x1C27 I2C=711C

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | 0 = Spkr mono L+R sum divide by 2 (-6db)<br>1 = Spkr mono L+R sum not divide by 2 |
| 10 | 0 | 0= Disabled Spkr(class D) mono mode<br>1 – Enable Spkr(Class D) mono mode |
| 9:0 | | Protected and reserved |

### 24.3.1.9 Codec_Test8 -> Port A, Beep, Charge-pump control

Get Response - HD_A [27:0] = 0x1CA8000 Set - HD_A[27:11] = 0x1C28 I2C=7120

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | Value for port A High-Z when bypass is set |
| 10 | 0 | 0 = Port A not muted (set codec test15[3])<br>1 = Port A muted |
| 9 | 0 | 0 = Port A beep disable (set codec test15[3])<br>1 = Port A beep enable |
| 8 | 0 | 0 = Port A bias off (set codec test15[3])<br>1 = Port A bias on |
| 7 | 0 | 0 = Port A powered down (set codec test15[3])<br>1 = Port A powered up |
| 6 | 0 | 0 = Beep enable \|\| vref_ok goes to port controller<br>1 = vref_ok goes to port controller |

| | | |
|---|---|---|
| 5 | 0<br>Protected | Force beep power down<br>0 = Normal<br>1 = Force beep power down |
| 4 | 0 | Forced value for beep gain MSB |
| 3 | 0 | Enable Forcing MSB of beep gain<br>0 = Normal mode<br>1 = Force mode |
| 2 | 0 | 0 = normal mono mode when spkr mono enable.<br>1 = dual mono mode when spkr mono enable |
| 1 | 0 | 0 = Charge pump powered down (set codec test8[0])<br>1 = Charge pump powered up (set codec test8[0]) |
| 0 | 0 | 0 = internal values control charge pump<br>1 = Test bits control charge pump |

### 24.3.1.10 Codec_Test9

Get Response - HD_A [27:0] = 0x1CA9000 Set - HD_A[27:11] = 0x1C29 I2C=7124

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | 0 = Automatic DAC selection and enable (default), Always HP Filter follows Class-D until bit-10 is 0.<br>1 = Manual DAC selection and enable only if Class-D is not enabled or Bit -10 is set to 1. |
| 10 | 0 | Sync_adc12_en<br>0 = ADCs work independently<br>1 = Last ADC powered up will start both ADCs and first one powered down will stop both ADCs |
| 9:8 | 00 | 00, 11 = High pass filter bypassed on all DACs<br>01 = High pass filter active on DAC 1, manual mode<br>10 = High pass filter active on DAC 2, manual mode |
| 7:6 | 00 | Reserved |
| 5:0 | 00100 | High Pass filter frequency selection N * 30<br>0x00 = 120Hz<br>0x01 = 30Hz<br>0x02 = 60Hz<br>…<br>0x3E = 1860Hz<br>0x3F = 1890Hz |

### 24.3.1.11 Codec_Test10

Get Response - HD_A [27:0] = 0x1CAA000 Set - HD_A[27:11] = 0x1C2A I2C=7128

| Bit | Default | Description |
|---|---|---|
| 11:10 | 00 | 00, 11= Class-D boost off on all DACs (default)<br>01 = Class-D boost active on DAC 1 (Manual mode)<br>10 = Class-D boost active on DAC 2 (Manual mode) |
| 9 | 0 | Disable DAC gain tracking for beep on DAC1<br>DAC PGA tracks beep gain |
| 8 | 0 | Disable DAC gain tracking for beep on DAC1<br>DAC PGA tracks beep gain |
| 7 | 0 | 0 = Class-D Boost Manual mode (default)<br>1 = Class-D Boost Tracking mode |
| 6 | 0 | ~~Set Mono Dac2 Enable~~<br>~~0: based on port mapping~~<br>~~1: based on portM connect index~~<br>Reserved (read/write capable) |
| 5 | 0 | ~~Set Mono Dac1 Enable~~<br>~~0: based on port mapping~~<br>~~1: based on portM connect index~~<br>Reserved (read/write capable) |
| 4 | 0 | 1-Enable power averaging based on the class D enable and DAC is not playing silence.<br>0-Enabling power averaging doesn't depend on class D enable or DAC have some non-zero value being playing |

| | | |
|---|---|---|
| 3 | 0 | Mute disable on signal ramp & FIFO<br>0: Mute work on signal ramp & FIFO<br>1: Mute not work on signal ramp & FIFO |
| 2:0 | 001 | hp_beep_gain to select required gain in steps of -4dB |

### 24.3.1.12 Codec_Test11

Get Response - HD_A [27:0] = 0x1CAB000 Set - HD_A[27:11] = 0x1C2B I2C=712C

| Bit | Default | Description |
|---|---|---|
| 11 | 00 | Disable the AUX mode loopback data path<br>0= In SnM mode aux mode loop back path through codec from ADC engine to DAC engine is available<br>1= In SnM mode aux mode loop back data path through codec from ADC engine to DAC is not available.<br>If need to path the data through the FIFO |
| 10 | 0 | Volume/Boost place:<br>1: put on new volume controller<br>0: not on new volume controller |
| 9 | 0 | Enable aux mode loop back path from ADC to DAC even if SnM is not enabled.<br>0= ADC to DAC loop back through codec is enabled in non SnM mode<br>1= ADC to DAC loop back through codec is enabled in non SnM mode |
| 8 | 1 | Disable fix for DSM fifo write/read at same address issue.<br>0 = enable the fix<br>1 = disable the fix |
| 7 | 0 | Selects clk_au from bclk/rcosc (depending on AFG low power mode) for generating spkr clk and other logic in analog control block<br>0 – bclk/rcosc<br>1 – PLL/PLL_DIV_2 (selected using ctest12[11]) |
| 6 | 0 | Select PLL_DIV_2 or BCLK/RCOSC (depending on AFG low power mode) for DC detection and power averaging<br>0 = bclk/rcosc<br>1 = pll_div_2 |
| 5 | 0 | OOB disable control for DAC2<br>0: enable OOB<br>1: disable OOB |
| 4 | 0 | OOB disable control for DAC1<br>0: enable OOB<br>1: disable OOB |
| 3 | 0 | 0-clock detection h/w logic in place<br>1-bypass enable for clock detection logic with a programmable value from bit 2 of this register |
| 2 | 0 | 0-Clock available indication to analog I/F when bypass bit [3] is set.<br>1-clock not available indication to analog I/F when bypass bit [3] is set |
| 1:0 | 00 | Class-D Gain boost<br>00 = +3db<br>01 = +4db<br>10 = +5db<br>11 = +6db |

### 24.3.1.13 Codec_Test12

Get Response - HD_A [27:0] = 0x1CAC000 Set - HD_A[27:11] = 0x1C2C I2C=7130

| Bit | Default | Description |
|---|---|---|
| 11:5 | (reserved) | |
| 4:0 | 0 | MADC data: from AFE for PA. The data is based on 48KHz |

### 24.3.1.14 Codec_Test13

Get Response - HD_A [27:0] = 0x1CAD000 Set - HD_A[27:11] = 0x1C2D I2C=7134

| Bit | Default | Description |
|---|---|---|
| 11:9 | 0 | Reserved |

| | | |
|---|---|---|
| 8 | 0 | Mute disable on signal ramp & FIFO of DAC5<br>0: Mute work on signal ramp & FIFO<br>1: Mute not work on signal ramp & FIFO |
| 7 | 0 | Signal ramping step size of DAC5<br>0 = 512<br>1 = 256 |
| 6 | 0 | 0 = Enable Signal ramping of DAC5<br>1 = Disable Signal ramping of DAC5 |
| 5:3 | 0 | Reserved |
| 2 | 0 | Mute disable on signal ramp & FIFO of DAC4<br>0: Mute work on signal ramp & FIFO<br>1: Mute not work on signal ramp & FIFO |
| 1 | 0 | Signal ramping step size of DAC4<br>0 = 512<br>1 = 256 |
| 0 | 0 | 0 = Enable Signal ramping of DAC4<br>1 = Disable Signal ramping of DAC4 |

### 24.3.1.15 Codec_Test14 (bit 5-4 changed in Neptune)

Get Response - HD_A [27:0] = 0x1CAE000 Set - HD_A[27:11] = 0x1C2E I2C=7138

| Bit | Default | Description |
|---|---|---|
| 11:7 | 0 | Reserved |
| 6 | 0 | TR switch force bit when bit 5 = 1. |
| 5 | 0 | TR enable force enable<br>1: tr_detect_en = bit 4 |
| 4 | 0 | TR enable force bit when bit 5 =1. |
| 3 | 0 | Used to bypass head set state machine in jack sense and allow s/w control the inputs to analog |
| 2 | 0 | If bit 3 is set, this bit will control the state of headset_detection input to analog |
| 1 | 0 | If bit 3 is set, this bit will control the state of headset_sel_b input to analog |
| 0 | 0 | If bit 3 is set, this bit will control the state of headset_sel_a input to analog |

### 24.3.1.16 Codec_Test15

Get Response - HD_A [27:0] = 0x1CAF000 Set - HD_A[27:11] = 0x1C2F I2C=713C

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | 0 = Data scdac3 unmuted (set codec_test15[6])<br>1 = Data scdac3 muted |
| 10 | 0 | Enable non-zero test mode for port controller as it waits for data to be non-zero from digital DAC<br>0 – Disable non-zero test mode<br>1 –Enable non-zero test mode |
| 9 | 0 | Enable D2 test mode for port controller<br>0 - Disable D2 test mode for port controller<br>1 - Enable D2 test mode for port controller |
| 8 | 0 | Enable mute test mode for port controller<br>0 - Disable mute test mode<br>1 - Enable mute test mode for port controller |
| 7 | 0<br>Protected | Speaker controller signals controlled by test bits |
| 6 | 0 | SCDAC3 pdb and mute bypass enable |
| 5 | 0 | Reserved |
| 4 | 0 | Reserved |
| 3 | 0 | PORTA controller signals bypass mode, controlled by test bits<br>bypass mode<br>Non-bypass mode |
| 2 | | Reserved |
| 1 | 0 | SCDAC2 pdb and mute bypass enable |
| 0 | 0 | SCDAC1 pdb and mute bypass enable |

### 24.3.1.17 Codec_Test16

Get Response - HD_A [27:0] = 0x1CE0000 Set - HD_A[27:11] = 0x1C60 I2C=7300

| Bit | Default | Description |
|---|---|---|
| 11:9 | 0 | sets the HP offset calibration path that need to be calibrated manually<br>001 – Port A left channel<br>010 – Port A right channel<br>011 – Port E left channel (port removed)<br>100 – Port E right channel (port removed)<br>101 – Port M channel (port removed)<br>When ctest0[11] is set |
| 8:6 | 0 | Read selection mux for HP calibration results on node verb = 1ccd.<br>0 - Read PORT A Left calibrated value<br>1 - Read PORT A Right calibrated value<br>2 - Read PORT E Left calibrated value (vestigial, not implemented in analog)<br>3 - Read PORT E Right calibrated value (vestigial, not implemented in analog)<br>4 - Read PORT M Right calibrated value (vestigial, not implemented in analog)<br>5 - Read PORT G Left calibrated value<br>6 - Read PORT G Right calibrated value |
| 5:0 | 0 | Reserved |

### 24.3.1.18 Codec_Test17

Get Response - HD_A [27:0] = 0x1CE1000 Set - HD_A[27:11] = 0x1C61 I2C=7304

| Bit | Default | Description |
|---|---|---|
| 11:9 | 000 | ADC4 gain ramp rate:<br>3'd0: Fs<br>3'd1: Fs/2<br>3'd2: Fs/4<br>3'd3: Fs/8<br>3'd4: Fs/16<br>3'd5: Fs/32<br>3'd6: Fs/48<br>3'd7: Fs/96 |
| 8:6 | 000 | ADC3 gain ramp rate:<br>3'd0: Fs<br>3'd1: Fs/2<br>3'd2: Fs/4<br>3'd3: Fs/8<br>3'd4: Fs/16<br>3'd5: Fs/32<br>3'd6: Fs/48<br>3'd7: Fs/96 |
| 5:3 | 000 | DAC5 gain ramp rate:<br>3'd0: Fs<br>3'd1: Fs/2<br>3'd2: Fs/4<br>3'd3: Fs/8<br>3'd4: Fs/16<br>3'd5: Fs/32<br>3'd6: Fs/48<br>3'd7: Fs/96 |
| 2:0 | 000 | DAC4 gain ramp rate:<br>3'd0: Fs<br>3'd1: Fs/2<br>3'd2: Fs/4<br>3'd3: Fs/8<br>3'd4: Fs/16<br>3'd5: Fs/32<br>3'd6: Fs/48<br>3'd7: Fs/96 |

### 24.3.1.19 Codec_Test18

Get Response - HD_A [27:0] = 0x1CE2000 Set - HD_A [27:11] = 0x1C62 I2C=7308

| Bit | Default | Description |
|---|---|---|
| 11:8 | 0 | Force calibration value based on selection values on these bits, used when state machine is bypassed.<br>1-selection for PORT A Left<br>2-selection for PORT A Right<br>3-selection for PORT E Left (vestigial, not implemented in analog)<br>4-selection for PORT E Right (vestigial, not implemented in analog)<br>5-selection for PORT M (vestigial, not implemented in analog)<br>6-selection for PORT G left<br>7-selection for PORT G Right |
| 7 | | Reserved |
| 6:0 | | Contains the 7-bit value that can be injected for the DAC port combination selected on [11:8] |

### 24.3.1.20 Codec_Test19

Get Response - HD_A [27:0] = 0x1CE3000 Set - HD_A[27:11] = 0x1C63 I2C=730C

| Bit | Default | Description |
|---|---|---|
| 11:9 | 01 | Threshold for shutdown decision for DC detection on speakers<br>000 = 5%<br>001 = 10%<br>010 = 15%<br>011 = 20%<br>100 = 25%<br>101 = 30%<br>110 = 35%<br>111 = 40% |
| 8:6 | 010 | Window selection options for DC detection. This is used for variability of trip times. As trip times decreases then above %DC need to vary to get the trip. Highlighted in red is to provide more programmability to allow faster trip. But requires to send higher DC value. New programmable threshold are test modes<br>000/001 = 4.1 sec<br>010 = 2.05 sec<br>011 = 1.35 sec<br>100 = 1 sec<br>101 = 680 ms<br>110 = 450 ms<br>111 = 340 ms |
| 5 | | Reserved |
| 4 | 0 | The gain decrement value selection between spkr protection or sum of spkr protection, OC and OT blocks. 0 – spkr protection only. 1 – Sum of all blocks. |
| 3 | | Reserved |
| 2 | 0 | Enable the OC detection for right channel of the class D protection<br>0=<br>1= |
| 1 | 1 | Enable the DC detection for right channel of the class D protection<br>0= Disable DC detection<br>1= Enable DC Detection |
| 0 | 1 | Enable the DC detection for left channel of the class D protection<br>0= Disable DC detection<br>1= Enable DC Detection |

### 24.3.1.21 Codec_Test20 -> Power Averaging, SPKR protection Control

Get Response - HD_A [27:0] = 0x1CE4000 Set - HD_A [27:11] = 0x1C64 I2C=7310

| Bit | Default | Description |
|---|---|---|
| 11:8 | 0011 | Gain decrement for Power averaging<br>0000 = No increment or decrement<br>0001 = -1db |

| | | |
|---|---|---|
| | | 0010 = -2db<br>0011 = -3db<br>0100 = -4db<br>0101 = -5db<br>0110 = -6db<br>0111 = -7db<br>…<br>1111 = -15db |
| 7 | 0 | 0-Enable class D errors/dc detection on spkr_pwr_dis analog pad<br>1-Disables the dc detection/or any error latching signal to be sent to on spkr_pwr_dis analog pad |
| 6 | 0 | 0 – Doesn't reset of latched Class D error/DC Detection status on spkr_pwr_dis analog pad for an enable on portg port output enable<br>1 – Enables reset of latched Class D errors/DC Detection status on spkr_pwr_dis analog pad based on class D port enable |
| 5 | 0 | 0 - Enable any of class D errors to causes the spkr_pwr_dis signal input to analog set to 1<br>1 - Enable DC detection to causes the spkr_pwr_dis signal input to analog set to 1 |
| 4 | 0 | Selection of left or right channel power averaging gain increment and decrement signals to be observed on gpio explained in dtest14 register<br>0 – left channel observed<br>1 – right channel observed |
| 3 | 0 | Over current protection Enabled for left channel speaker<br>0 = Disable protection<br>1 = Enable protection |
| 2 | 0 | reserved |
| 1 | 0 | 0 – Test mode disabled<br>1 - Test mode to feed High pass filter output from DAC chain to power averaging block bypassing reconstructed class D output signal |
| 0 | 0 | Spkr gain test bit, Spkr gain test bit. When set the DAC gain to spkr will be affected based on the input from the OC and OT protection scheme |

### 24.3.1.22 Codec_Test21 - > SPKR protection control.

Get Response - HD_A[27:0] = 0x1CE5000 Set - HD_A[27:11] = 0x1C65 I2C=7314

| Bit | Default | Description |
|---|---|---|
| 11:10 | 00 | Over current slow threshold<br>00 = 50%<br>01 = 25%<br>10 = 12.5%<br>11 = ~1% |
| 9:8 | 00 | Over current fast threshold<br>00 = 50%<br>01 = 67.5%<br>10 = 75%<br>11 = ~100% |
| 7:6 | 00 | Over current slow window size<br>00 = 8mS<br>01 = 32mS<br>10 =128mS<br>11 = 512mS |
| 5:4 | 00 | Over current fast window size<br>00 = 2mS<br>01 = 8mS<br>10 = 32mS<br>11 = 128mS |
| 3:1 | 011 | Gain decrement for OT and OC protection<br>000 = No increment or decrement<br>001 = -1db<br>010 = -2db<br>011 = -3db<br>100 = -4db<br>101 = -5db |

| | | |
|---|---|---|
| | | 110 = -6db<br>111 = -7db |
| 0 | 0 | If ctest20[7] bit is set == 1, then value on this register determines the data to be sent to analog spkr_pwr_dis pad |

### 24.3.1.23 Codec_Test22

Get Response - HD_A[27:0] = 0x1CE6000 Set - HD_A[27:11] = 0x1C66 I2C=7318

| Bit | Default | Description |
|---|---|---|
| 11:9 | 0 | Reserved |
| 8 | 0 | LSB bit for portf_c_gain. This bit adds to the LSB of Portf_X_gain from spec creating more steps 0-12dB, 1-6dB steps |
| 7 | 0 | LSB bit for portf_X_gain. This bit adds to the LSB of Portf_X_gain from spec creating more steps 0-12dB, 1-6dB steps |
| 6 | 0 | LSB bit for porte_X_gain. This bit adds to the LSB of Porte_X_gain from spec creating more steps 0-12dB, 1-6dB steps |
| 5 | 0 | LSB bit for portd_X_gain. This bit adds to the LSB of Portd_X_gain from spec creating more steps 0-12dB, 1-6dB steps |
| 4 | 0 | LSB bit for portb_X_gain. This bit adds to the LSB of Portb_X_gain from spec creating more steps 0-12dB, 1-6dB steps |
| 3:2 | 0 | Reserved |
| 1 | 0 | 1 – Polarity of spkr_pwr_dis pad enable is inverted.<br>0 – No change in the polarity |
| 0 | 0 | If ctest20[7] bit is set == 1, then value on this register determines the data to be sent to analog spkr_pwr_dis pad |

### 24.3.1.24 Codec_Test23

Get Response - HD_A[27:0] = 0x1CE7000 Set - HD_A[27:11] = 0x1C67 I2C=731C

| Bit | Default | Description |
|---|---|---|
| 11:9 | 000 | Reserved Read as 0 |
| 8 | 0 | 0 – SnM mode doesn't depend on Port B presence.<br>1 - When set enables SnM mode only based on Port B presence. |
| 7 | 0 | If set = 1, Hardware will auto set lp_en=1'b0 immediately when detected SnM pin pull high or EC set to SnM mode, after SnM debounce time. |
| 6 | 0 | reserved |
| 5:4 | 00 | Power Average threshold shift select (both low and high)<br>00: No shift<br>01: divide by 4<br>10: divide by 8<br>11: divide by 16 |
| 3:0 | 0000 | The rate at which the threshold comparisons are updated: (default is 0.032s)<br>0000: 0.032s<br>0001: 4.096s<br>0010: 2.048s<br>0011: 1.024s<br>0100: 0.512s<br>0101: 0.256s<br>0110: 0.128s<br>0111: 0.064s<br>1000: 0.032s<br>1001: 0.016s<br>1010: 0.008s |

### 24.3.1.25 Codec_Test24

Get Response - HD_A[27:0] = 0x1CE8000 Set - HD_A[27:11] = 0x1C68 I2C=7320

| Bit | Default | Description |
|---|---|---|
| 11:0 | 000 | Secret register |

### 24.3.1.26 Codec_Test25

Get Response - HD_A[27:0] = 0x1CE9000 Set - HD_A[27:11] = 0x1C69 I2C=7324

| Bit | Default | Description |
|---|---|---|
| 11:0 | 000 | Secret register |

### 24.3.1.27 Codec_Test26 –> Power Averaging Control

Get Response - HD_A[27:0] = 0x1CEA0000 Set - HD_A[27:11] = 0x1C6A I2C=7328

| Bit | Default | Description |
|---|---|---|
| 11:4 | 0 | reserved |
| 3:2 | 10 | Power Average LP filters band selection. See Table Below. |
| 1 | 0 | 0 – Disable power averaging block for class D right channel<br>1 - Enable power averaging block for class D right channel |
| 0 | 0 | 0 – Disable power averaging block for class D left channel<br>1 - Enable power averaging block for class D left channel |

| Register codec_test26[3:2] | Value of "$p$" | Settling Times | | | |
|---|---|---|---|---|---|
| | | 63.2% | | 95% | |
| | | Sample Periods | Seconds | Sample Periods | Seconds |
| 00 | $1/2^{16}$ | 65536 | 0.341333 | 196608 | 1.024 |
| 01 | $1/2^{15}$ | 32768 | 0.170666 | 98304 | 0.512 |
| 10 | $1/2^{14}$ | 16384 | 0.085333 | 49152 | 0.256 |
| 11 | $1/2^{12}$ | 4096 | 0.021333 | 12288 | 0.064 |

### 24.3.1.28 Codec_Test27

Get Response - HD_A[27:0] = 0x1CEB000 Set - HD_A[27:11] = 0x1C6B I2C=732C

| Bit | Default | Description |
|---|---|---|
| 11:6 | 0 | Threshold value low for the making the decision of increment the gain (threshold lo)<br>000000--70%<br>000001--69%<br>-<br>-<br>-<br>111110--6%<br>111111--7%<br>Step is %1 |
| 5:0 | 0 | Threshold value high for the making the decision of decrement the gain (threshold hi)<br>000000--70%<br>000001--69%<br>-<br>-<br>-<br>111110--6%<br>111111--7%<br>Step is 1% |

### 24.3.1.29 Codec_Test28

Get Response - HD_A [27:0] = 0x1CEC000 Set - HD_A[27:11] = 0x1C6C I2C=7330

| Bit | Default | Description |
|---|---|---|

| | | |
|---|---|---|
| 11:5 | 0 | Reserved |
| 4 | 0 | 0 = Port F not high-z (set [0])<br>1 = Port F high-z |
| 3 | 0 | 0 = Port F not muted (set [0])<br>1 = Port F muted |
| 2 | 0 | 0 = Port F bias off (set bit [0])<br>1 = Port F bias on |
| 1 | 0 | 0 = Port F powered down (set bit [0])<br>1 = Port F powered up |
| 0 | 0 | PORTF controller signals bypass mode, controlled by test bits<br>0-Non-bypass mode<br>1-Bypass mode |

### 24.3.1.30 Codec_Test29 (bit 4:0 changed in Neptune)

Get Response - HD_A [27:0] =0x1CED000 Set - HD_A[27:11] = 0x1C6D I2C=7334

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | Spkr prot local reset enable<br>0: no local reset<br>1: local reset enabled when spkr clk en =0, the Spkr Prot block will be in async reset |
| 10 | 0 | reserved |
| 9 | 0 | reserved |
| 8 | 0 | DAC high pass filter clipping<br>0 = -2.5db clipping enabled<br>1 = disabled |
| 7:6 | 2'b11 | 192Khz sample rate bank selection mapping in manual mode<br>00: use bank 00<br>01: use bank 01<br>11: use bank 11<br>10: bypass EQ/DRC |
| 5:4 | 2'b01 | 96Khz sample rate bank selection mapping in manual mode<br>00: use bank 00<br>01: use bank 01<br>11: use bank 11<br>10: bypass EQ/DRC |
| 3:2 | 2'b00 | 48Khz sample rate bank selection mapping in manual mode<br>00: use bank 00<br>01: use bank 01<br>11: use bank 11<br>10: bypass EQ/DRC |
| 1 | 0 | reserved |
| 0 | 0 | D4/D3.5 button wake/UM enable<br>0: disable<br>1: enable |

### 24.3.1.31 Codec_Test30 -> SPKR offset calibration control state machine

Get Response - HD_A [27:0] = 0x1CEE000 Set - HD_A[27:11] = 0x1C6E I2C=7338

| Bit | Default | Description |
|---|---|---|
| 11:8 | 0000 | Reserved |
| 7 | 0 | 1: Enable Speaker/PortG auto calibration |
| 6 | 0 | Force PortG calibration to be run when = 1 |
| 5 | 0 | Reset Offset Calibration Engine (Portg) when set to "1" |
| 4 | 0 | Enable bypass offcal value with test register (18) bits [6:0] |
| 3:2 | 00 | Set the offset calibration path that need to be set manually<br>1 = PORTG left channel<br>2 = PORTG right channel |
| 1 | 0 | Set to enable manual Calibration thru register bit 0<br>0 = Bit [0] not used for calibration enable<br>1 = Bit [0] is used for calibration enable |
| 0 | 0 | Enable SPKR calibration if bit 1 is set |

| | | |
|---|---|---|
| | | 0 = Don't enable<br>1 = Enable |

### 24.3.1.32 Codec_Test31 (Not Used)

Get Response - HD_A [27:0] = 0x1CEF000 Set - HD_A[27:11] = 0x1C6F I2C=733C

| Bit | Default | Description |
|---|---|---|
| | | |
| | | |

### 24.3.1.33 Codec_Test32

Get Response - HD_A [27:0] = 0x1C98000 Set - HD_A[27:11] = 0x1C18 I2C=70A0

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | 1-Enable PRE-EQ Mixing |
| 10 | 0 | PRE-EQ Mixing Base Select<br>0: Base on DAC1 signals<br>1: Base on DAC2 signals |
| 9 | 1 | Interpolator mute/unmute disable.<br>1: Mute/unmute gain disable.<br>0: Mute/Un-mute enable |
| 8 | 0 | Bypass ED delay buffer<br>1: Bypass ED delay buffer<br>0: No Bypass |
| 7 | 1 | Override value of cpmp_ctrl_dmode |
| 6 | 0 | Override ED output to AFE<br>0: Override<br>1: No override |
| 5 | 0 | ~~PortM on D3 live~~<br>~~0: Disable~~<br>~~1: Enable~~<br>Reserved (read/write capable) |
| 4 | 0 | PortE on D3 live<br>0: Disable<br>1: Enable |
| 3 | 0 | Override ED output enable value<br>0: Disable ED output<br>1: Enable ED output |
| 2 | 0 | Override for ED output enable<br>0: No override.<br>1: Enable ED output en override. |
| 1 | 0 | Override ED port selection value<br>0: DAC1<br>1: DAC2 |
| 0 | 0 | Override for ED port selection<br>0: No override<br>1: Enable ED port selection override |

## 24.3.2 Multi-Button/TR Registers

### 24.3.2.1 Codec_Test33 Multi-Button Control

Get Response - HD_A [27:0] = 0x1C99000 Set - HD_A[27:11] = 0x1C19 I2C=70A4

| Bit | Default | Description |
|---|---|---|
| 11 | 0x1 | MB enable<br>0: single button mode<br>1: enable multi-button mode |
| 10 | 0x0 | Combo TR sense enable |

| | | |
|---|---|---|
| | | 1: enable<br>0: disable |
| 9 | 0x0 | MB status/event count enable in UM payload<br>0: disable (all 0s in payload).<br>1: enable in UM payload |
| 8 | 0x0 | MB LP send UM/INT<br>0: not send UM/INT for LP<br>1: send UM/INT for LP |
| 7:0 | 0x14 | B1_TH<br>B1 Threshold |

### 24.3.2.2 Codec_Test34 Multi-Button Control

Get Response - HD_A [27:0] = 0x1C9A000 Set - HD_A[27:11] = 0x1C1A I2C=70A8

| Bit | Default | Description |
|---|---|---|
| 11:8 | 2 | MB debounce range<br>0: 5ms<br>1: 15ms<br>2: 25ms<br>3: 35ms<br>4: 45ms<br>5: 55ms<br>6: 65ms<br>7: 75ms<br>8: 85ms<br>Other: 0ms (no debounce). |
| 7:0 | 0x42 | B2_TH<br>B2 Threshold |

### 24.3.2.3 Codec_Test35 Multi-Button Control

Get Response - HD_A [27:0] = 0x1C9B000 Set - HD_A[27:11] = 0x1C1B I2C=70AC

| Bit | Default | Description |
|---|---|---|
| 11:9 | 3 | Long Press time selection<br>0: 0.25s<br>1: 0.5s<br>2: 0.75s<br>3: 1s<br>4: 1.5s<br>5: 2s<br>6: 3s<br>7: 4s |
| 8 | 0 | Reserved |
| 7:0 | 0x7F | B3_TH<br>B3 Threshold |

### 24.3.2.4 Codec_Test36 Multi-Button Control

Get Response - HD_A [27:0] = 0x1C9c000 Set - HD_A[27:11] = 0x1C1C I2C=70B0

| Bit | Default | Description |
|---|---|---|
| 11:9 | 0x4 | RC LPF coeff: a = dt/(dt+RC)<br>0: Bypass<br>1: a = 1/16<br>2: a = 1/32<br>3: a = 1/64<br>4: a = 1/128<br>5: a = 1/256<br>6,7: a = 1/512 |
| 8 | RO | HP_RES_GT_5K |
| 7:0 | 0x2A | B4_TH |

| | | |
|---|---|---|
| | | B4 Threshold |

### 24.3.2.5 Codec_Test37 MB + TR

Get Response - HD_A [27:0] = 0x1C9d000 Set - HD_A[27:11] = 0x1C1d I2C=70B4

| Bit | Default | Description |
|---|---|---|
| 11 | RO | HP_RES_LS_1K |
| 10:8 | W1C | MB event count. (count on DN pos edge)<br>When W1 below bits also clear the count value). |
| 7 | W1C | MB_LP |
| 6 | W1C | MB_MIC |
| 5 | W1C | MB_B4 |
| 4 | W1C | MB_B3 |
| 3 | W1C | MB_B2 |
| 2 | W1C | MB_B1 |
| 1 | W1C | MB_UP |
| 0 | W1C | MB_DN |

### 24.3.2.6 Codec_Test38

Get Response - HD_A [27:0] = 0x1C9e000 Set - HD_A[27:11] = 0x1C1e I2C=70B8

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | Headset_tip_detcct_deb as GPIO UM/INT<br>0: disable<br>1: enable |
| 10 | 0 | Micbias_dtct_filt as GPIO UM/INT<br>0: disable<br>1: enable |
| 9:7 | 000 | Headset_tip_detect_deb & micbias_dtct_filt GPIO UM/INT mapping<br>0: map as GPIO0 UM/INT<br>1: map as GPIO1 UM/INT<br>2: map as GPIO2 UM/INT<br>3: map as GPIO3 UM/INT<br>4: map as GPIO4 UM/INT<br>If both bit11 & bit10 set, the headset_tip_detect_deb & micbias_dtct_filt map to GPIO[n+1] & GPIO[n]. |
| 6 | 0 | 0: Porta presence after headset detection done in combo jack mode<br>1: Porta presence assert before the headset detection in combo jack mode |
| 5 | 0 | Headset detection restart<br>0 -> 1 trigger the headset detection restart |
| 4:0 | 11000 | Program de-bounce time for headset tip detection signals (Default 5'h18 = (deb_time + 1) 10ms = 250ms de-bounce) |

### 24.3.2.7 Codec_Test39 Headset Detection Manual Mode

Get Response - HD_A [27:0] = 0x1C9f000 Set - HD_A[27:11] = 0x1C1F I2C = 70BC

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | Headset detection manual mode enable<br>1: enable manual mode, signals mapping as below<br>0: disable manual mode |
| 10:0 | 0x0 | combo_jack_lp_ena = reg[0]<br>combo_micbias_ena = reg[1]<br>combo_portd_presence = reg[2]<br>combo_detect_done = reg[3]<br>headset_sel_a = reg[4]<br>headset_sel_b = reg[5] |

| | | |
|---|---|---|
| | | headset_detection = reg[6]<br>tr_detect_en = reg[7]<br>tr_detect_switch = reg[8] |

## 24.3.3 Digital test registers

### 24.3.3.1 Digital_Test0

Get Response - HD_A [27:0] = 0x1CC0000 Set HD_A[27:11] = 0x1C40 I2C=7200

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | 1-PLL bypassed using PLL reference input clock |
| 10 | 1 | 1-Enable for automatic detection of chip reset on sync<br>Used in scenarios where chip reset is not available |
| 9 | 0 | 1-Enable clock detect to switch the the PLL reference clock to rcosc |
| 8 | 0 | 1-Enable clock detect to switch the analog control clock to rcosc |
| 7 | 0 | 1-Observe any jack sense event on spkr_mute_gpio1 |
| 6 | 0 | 1-Observe any jack sense event on music_req_gpio0 |
| 5:4 | ~~01~~<br>00 | ~~Functional muxing on digmic_dat2_gpio5~~<br>~~01-GPIO5~~<br>~~10-SPDIF OUT~~<br>Reserved |
| 3:2 | 01 | Functional muxing on digmic_clk2_gpio4<br>01-GPIO4<br>10-SPDIF OUT |
| ~~3~~ | ~~0~~ | ~~Portc type selection: analog or digital~~<br>~~1: analog~~<br>~~0: digital~~ |
| ~~2~~ | ~~0~~ | ~~Indicates the test group data high 3-bits or low 3-bits:~~<br>~~0: test group data [3:0]~~<br>~~1: test group data [6:4]~~ |
| 1:0 | 0 | Functional muxing on spkrmutb_gpio1<br>00-Spkr mute<br>01-GPIO1<br>10-SPDIF OUT<br>11 – DMIC2 Data (Port H) |

### 24.3.3.2 Digital_Test1

Get Response - HD_A [27:0] = 0x1CC1000 Set - HD_A[27:11] = 0x1C41 I2C=7204

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | D3 live mode control(when codec in D3)<br>0: exit the D3 live<br>1: trigger the D3 live |
| 10:9 | 01 | Function applied on music_req_gpio0<br>0-GPIO function<br>1-Music request function (if GPIO is not enabled)<br>2-SPDIF Out function |
| 8 | 0 | Observe hbridge out left and right on the GPIOs<br>hbridge_outp_l – portc_l_gpio_b<br>hbridge_outn_l – portc_r_gpio_c<br>hbridge_outp_r – spkrmuteb (ctest6[2] = 1)<br>hbridge_outn_r – EAPD |
| 7:6 | 01 | Sets the pull up control for the dmic_dat1_gpio3.<br>00 – No pull up, no pull down<br>01 – Weak Pull down -> 28/65Kohm |

| | | |
|---|---|---|
| | | 10- Weak pull up value 1 -> 28-65Kohm<br>11 – Bus Keeper resistance -> 28 - 65kohm |
| 4 | 0 | How to control D3 live mode<br>0: D3 live mode control from register bit[11]<br>1: D3 live mode control from SnM pin |
| 5 | 0 | Merge portA and portB<br>0: portA and portB are separate ports (default).<br>1: merge portA and portB: PortA becomes a vendor widget, portB becomes an inout port) |
| 4 | 0 | Reserved |
| 3 | 0 | Set this bit to enable DAC support for 192 and 96K<br>0 – DAC support only 44.1 and 48k<br>1 – DAC support all sample rate from 44.1, 48, 96 and 192K |
| 2 | 0 | Indicates the test group data high 3-bits or low 3-bits:<br>0: test group data [3:0]<br>1: test group data [6:4] |
| 1:0 | 10 | DigMic_dat1_gpio3 pad output drive strength<br>00 – IOL/IOH=+/-1.3mA<br>01 – IOL/IOH=+/-2.0mA<br>10 – IOL/IOH=+/-2.7mA<br>11 – IOL/IOH=+/-3.3mA |

### 24.3.3.3 Digital_Test2

Get Response - HD_A [27:0] = 0x1CC2000 Set – HD_A[27:11] = 0x1C42 I2C=7208

| Bit | Default | Description |
|---|---|---|
| 11 | 00000 | Send UM every frame<br>0: Send UM every other frame (default)<br>1: Send UM every frame |
| 10 | 0 | ~~Reserved~~<br>PortC type selection: analog or digital<br>1: analog<br>0: digital |
| 9 | 0 | 1-Bypass SPDIF clock off logic |
| 8 | 0 | 1: Enable read back the PLL tracking result; 0: Read back the PLL tracking setting |
| 7:5 | 101 | HP limit value detect – range is from 3dB to 10dB<br>0x5: 8dB (default) |
| 4 | 1<br>Protected | Override temperature shutdown signal<br>0 = Disable over temp shutdown<br>1 = put Class D in D3, and disable the output |
| 3 | 0<br>Protected | Override under voltage shutdown signal<br>0: put Class D in D3, and disable the output<br>1: Do not change power or output states. |
| 2 | 0<br>Protected | Override Class-D 5V POR signal.<br>0: Allow normal 5V POR signal to drive Class D POR<br>1: Force POR out of reset |
| 1 | 0<br>Protected | Override Class-D short circuit signal.<br>0: Allow normal short circuit detection<br>1: Force short circuit detection off |
| 0 | 0<br>Protected | Soft POR reset<br>1: enable soft POR reset |

### 24.3.3.4 Digital_Test3

Get Response - HD_A[27:0] = 0x1CC3000 Set - HD_A[27:11] = 0x1C43 I2C=720C

| Bit | Default | Description |
|---|---|---|
| 11:8 | 0 | Reserved – Read as 0 |
| 7 | 0 | B0-DC_DET pin status created for Sony. The raw status of the pin reflects on this bit |
| 6 | 0 | Class D OT protect status – Write 1 to clear |

| | | |
|---|---|---|
| | | 0 = No OT engine event<br>1 = OT engine caused shutdown |
| 5 | 0 | Class D OC protect status – Write 1 to clear<br>0 = No OC engine event<br>1 = OC engine caused shutdown |
| 4 | 0 | Class D DC protect status – Write 1 to clear<br>0 = No DC detected<br>1 = DC detect caused shutdown |
| 3 | 0 | Class D short circuit status – Write 1 to clear<br>0 = No short circuit<br>1 = Short circuit detect caused shutdown |
| 2 | 0 | Class D temperature status – Write 1 to clear<br>0 = Temperature is below threshold<br>1 = Temperature above threshold caused shutdown |
| 1 | 0 | Class-D por5V status – Write 1 to clear.<br>0: Class-D supply is above 4V<br>1: Class-D supply dropped below 4V cause shutdown |
| 0 | 0 | Class-D error status (OR of the above bits except POR5).<br>0: Class-D error has not happened<br>1: Class-D error has happened |

### 24.3.3.5 Digital_Test4

Get Response - HD_A[27:0] = 0x1CC4000 Set - HD_A[27:11] = 0x1C44 I2C=7210

| Bit | Default | Description |
|---|---|---|
| 11:8 | 0x0 | GPIO5 output – Decoded as follows:<br>0000 normal gpio/eapd function (default)<br>0011 rcosc_clk ~6 MHz out<br>0100 36Mhz (audio PLL/4)<br>0101 24.576 Mhz (audio PLL/6)<br>0110 dac_adc_clk 3.07 MHz out<br>0111 lp_rcosc_clk ~200KHz<br>1110 SPKR Clock ~3.07MHz |
| 7 | 0<br>Protected | disable setting of audio_clk_stop_ok bit<br>0: enable<br>1: disable |
| 6 | 0<br>Protected | PLL bypass<br>0: disable PLL bypass<br>1: PLL bypassed:<br>Audio PLL output = EAPD<br>SPDIF PLL output = GPIO1 |
| 5 | 0 | Force RCOSC Calibration value.<br>0: use calibration logic (default)<br>1: force a calibration value) |
| 4 | 1 | SPKR Calibration Clock Divide<br>0: 384Khz/8 (final Clock = 12KHz, Internal Clock is divide by 4)<br>1: 284K/32 (Final clock = 3KHz, Internal Clock is divide by 4) |
| 3:2 | 10 | DigMic_clk1_gpio2 pad output drive strength<br>00 – IOL/IOH=+/-1.3mA<br>01 – IOL/IOH=+/-2.0mA<br>10 – IOL/IOH=+/-2.7mA<br>11 – IOL/IOH=+/-3.3mA |
| 1:0 | 01 | Sets the pull up control for the dmic_clk1_gpio2.<br>00 – No pull up, no pull down<br>01 – Weak Pull down -> 28/65Kohm<br>10- Weak pull up value 1 -> 28-65Kohm<br>11 – Bus Keeper resistance -> 28 - 65kohm |

### 24.3.3.6 Digital_Test5

Get Response - HD_A[27:0] = 0x1CC5000 Set - HD_A[27:11] = 0x1C45 I2C=7214

| Bit | Default | Description |
|---|---|---|

| | | |
|---|---|---|
| 11:8 | 0110 | Charge pump clk freq = 24Meg / (24 + 4*N)<br>0000 = 1000kHz<br>0001 = 857kHz<br>0010 = 750kHz<br>0011 = 667kHz<br>0100 = 600kHz<br>0101 = 545kHz<br>0110 = 500kHz (default)<br>0111 = 461kHz<br>1000 = 428kHz<br>1001 = 400kHz<br>1010 = 375kHz<br>1011 = 353kHz<br>1100 = 333kHz<br>1101 = 315kHz<br>1110 = 300kHz<br>1111 = 285kHz |
| 7:4 | 0000 | Jacksense test data, replacing jack_data[3:0] from the SAR in Jacksense test mode. |
| 3 | 0 | Jacksense test jack_sel, replacing jack_sel signal in Jacksense test mode. |
| 2 | 0 | Jacksense test busy, replacing jack_busy from the SAR in Jacksense test mode. |
| 1 | 0 | Jacksense test mode set.<br>0: normal mode<br>1: Jacksense test mode |
| 0 | 0 | Jacksense power down.<br>0: Power up<br>1: Power down |

### 24.3.3.7 Digital_Test6

Get Response - HD_A[27:0] = 0x1CC6000 Set - HD_A[27:11] = 0x1C46 I2C=7218

| Bit | Default | Description |
|---|---|---|
| 11 | 0<br>Protected | 0 = Audio PLL power follows state machine<br>1 = Audio PLL powered down |
| 10 | 0 | Observe short circuit OR output on spkrmuteb |
| 9 | 0 | Scrambler Analog Test – input mode<br>0: Scrambler inputs to analog macro are not forced in<br>1: Scrambler lower and upper nibble inputs to analog macro are forced in from<br>dmic_clk_gpio_d, dmic_dat_gpio_e, portc_l_gpio_b, portc_r_gpio_c<br>scdac_adc_clk is forced in at EAPD<br>nibble select is forced in at spkrmuteb |
| 8 | 0 | Observe dc_error on GPIO1. |
| 7 | 0 | |
| 6 | 0<br>Protected | Analog FIFO loopback (Analog through the codec to ADC FIFO connected to DAC FIFO then through the codec back to analog).<br>0: Disable loopback (default)<br>1: Loopback:<br>ADC1 FIFO to DAC1 FIFO<br>ADC2 FIFO to DAC2 FIFO<br>ADC3 FIFO to DAC3 FIFO |
| 5 | 0<br>Protected | HD Audio FIFO loopback (Bus to DAC FIFO connected to ADC FIFO back to the bus).<br>0: Do not loopback<br>1: Loopback:<br>DAC1 FIFO to ADC1 FIFO<br>DAC2 FIFO to ADC2 FIFO<br>DAC3 FIFO to ADC3 FIFO |
| 4:0 | 11000 | Program de-bounce time for following signals (Default 5'h18 = (deb_time + 1) * 10ms = 250ms de-bounce)<br>Jack, headset_tip_detect, |

### 24.3.3.8 Digital_Test7

Get Response - HD_A [27:0] = 0x1CC7000 Set - HD_A[27:11] = 0x1C47 I2C=721C

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | Force headset into gpio debug mode<br>0 – No operation<br>forces<br>headset_sel_a -> portc_l_gpio_b<br>headset_sel_a -> portc_r_gpio_c<br>headset_detection -> dmic_clk_gpio_d<br>micbiasd_dtct -> dmic_dat_gpio_e<br>headset_tip_detect -> jsense_gpio_a |
| 10 | 0 | Force ClassD PMW in<br>0: Do not force<br>1: Force classD PWM from pins portc_l_gpio_b and portc_r_gpio_c |
| 9 | 0<br>Protected | Force ClassD clock in<br>0: Do not force<br>1: Force classD clock in from pin spkrmuteb |
| 8:0 | 0x01F | ClassD clock divider value.<br>002 – 1FF: 12MHz -> 47kHz<br>0x03E = 387.1kHz |

### 24.3.3.9 Digital_Test8

Get Response - HD_A [27:0] = 0x1CC8000 Set – HD_A [27:11] = 0x1C48 I2C=7220

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | When set to 1 able to inject hbridge high meas low and high bypassing signals from analog<br>hbridge_meas_high_l – jsense_gpio_a<br>hbridge_meas_high_r – spkrmuteb<br>hbridge_meas_low_r – dmic_dat_gpio_e<br>hbridge_meas_low_l – dmic_clk_gpio_d |
| 10 | 0<br>Protected | Override setting of audio_clk_off<br>0: Do not override<br>1: Override. Audio_clk_off is reset to 0 |
| 9 | 0 | Aux mode pll set signal bypassed the de-bouncing for PLL and enabled the audio PLL immediately |
| 8 | 0 | Observe beep signals on GPIOs<br>beep_pin -> portc_l_gpio_b<br>beep_pin_debounce -> portc_r_gpio_c<br>beep_dig_en -> dmic_dat_gpio_e |
| 7 | 0 | Disable Temp error headphone shutdown |
| 6 | 0 | Inject hbridge_meas_high from external source. Bypass Analog signal for DC detection and power averaging<br>hbridge_outp_l – portc_l_gpio_b<br>hbridge_outn_l – portc_r_gpio_c<br>hbridge_outp_r – spkrmuteb (ctest6[2] = 1)<br>hbridge_outn_r – jsense_gpio_a |
| 5 | 0 | Inject temp_error from external source. Bypass Analog signal. |
| 4 | 0 | When set bypass port D data from analog and inject data from gpios to digital<br>portd_mono_data – portc_r_gpio_c<br>portd_data_rdy – portc_l_gpio_b |
| 3 | 0 | When set bypass portC data from analog and inject data from gpios to digital<br>portc_l_mono_data – portc_r_gpio_c<br>portc_r_data – portc_l_gpio_b<br>portc_data_rdy – dmic_clk_gpio_d<br>portc_mono_status – dmic_dat_gpio_e |
| 2 | 0 | When set bypass portb data from analog and inject data from gpios to digital<br>portb_l_data – portc_r_gpio_c<br>portb_r_data – portc_l_gpio_c<br>portb_data_rdy – dmic_clk_gpio_d |
| 1 | 0 | ADC –DAC analog loop back through FIFO without azalia being running |

| | | |
|---|---|---|
| 0 | 1 | Response in D3Cold<br>0: send response to HD bus after the first soft reset (default)<br>1: send response to HD bus after the second soft reset) |

### 24.3.3.10 Digital_Test9

Get Response - HD_A[27:0] = 0x1CC9000 Set - HD_A[27:11] = 0x1C49 I2C=7224

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | Speaker async clock<br>0 = normal mode<br>1 = async clock output on portc_l_gpio_b, clk2 output on _portc_r_gpio_c |
| 10 | 0 | Class D status output<br>spkr_error, por_5 & temp_error to portc_l_gpio_b, portc_r_gpio_c & dmic_dat_gpio_e respectively |
| 9 | 0 | Selects analog adc1 left /right observed on the gpios. Refer to analog group muxing in new Neptune analog muxing.<br>0= left<br>1= right<br>When [6:3] select group 4:<br>0: normal jack<br>1: multi-button |
| 8 | 0 | Test bit to enter SnM mode for verification<br>0 = Not in SnM mode<br>1 = SnM mode |
| 7 | 0 | When set to 1 forces auxmode to go through FIFO even though azalia is not present |
| 6:3 | 0 | Selects 1 out 10 analog group muxing on gpio[5:0] for monitor<br>Select 11 = inject data on gpio{3:0]<br>Refer to analog muxing group documents.<br>0000: No group selected<br>0001: group1 selected<br>0010: group2 selected<br>……<br>1010: group10 selected<br>1011: inject data to DAC1_L Modulator input (gpio3-0) for injection and gpio4 for DAC1_CLK monitor. |
| 2 | 0 | reserved |
| 1 | 0 | Select any of the DAC1 DSM output on gpios (refer to analog group muxing)<br>0-DAC1 left<br>1-DAC1 right |
| 0 | 0 | Jacks_data_mux for Jacksense SAR Output Data and I2S Jsense, Digital Test10[6,5,3:0]<br>0-{jack_data}<br>1-{portf_presence,porte_presence,portb_presence,porta_presence} |

**Table 24:1 - Analog Test Mux Groups**

| Group | Digital_ test9[9] | Digital_ test1[2] | GPIO3 | GPIO2 | GPIO1 | GPIO0 |
|---|---|---|---|---|---|---|
| 1 | NA | 1 | 1'b0 | temp_error | spkr_por_status | spkr_sc_status |
| | | 0 | spkr_test_out[3] | spkr_test_out[2] | spkr_test_out[1] | spkr_test_out[0] |
| 2 | NA | 1 | 1'b0 | combo_mic_bias_ena | scdac_adc_clk | lp_rcosc_clk |
| | | 0 | rcosc_clk | audio_pll_out | pll_clktstb | pll_clktsta |
| 3 | N | 1 | 1'b0 | 1'b0 | 1'b0 | scdac_adc_clk |
| | | 0 | adc_data_mux3 | adc_data_mux[2] | adc_data_mux[1] | adc_data_mux[0] |
| 4-0 | 0 | 1 | 1'b0 | 1'b0 | \|jack_sar_data | headset_tip_detect |
| | | 0 | micbiasd_dtct | headset_detection | headset_sel_b | headset_sel_a |
| 4-1 | 1 | 1 | 1'b0 | tr_sense_switch | combo_trsense_en | combo_micbias_ena |
| | | 0 | micbiasd_dtct | headset_detection | tr_sense_compout | tr_detect_en |
| 5 | NA | 1 | 1'b0 | 1'b0 | 1'b0 | nibble_sel_dly |
| | | 0 | dsm_data_mux3 | dsm_data_mux[2] | dsm_data_mux[1] | dsm_data_mux[0] |
| 6 | NA | 1 | 1'b0 | spkr_clk2x_mux | spkr_pdb | beep_ana_en |
| | | 0 | beep_anamux_en | beep_pin_en_deb | beep_pin_en | beep_pin |

| | | | | | | |
|---|---|---|---|---|---|---|
| 7-0 | 0 | 1 | 1'b0 | offcal_cmp_r | offcal_cmp_l | cpmp_avee_ok |
| | | 0 | hp_offcal_bsy | portm_offcal_en | porta_offcal_en | porte_offcal_en |
| 7-1 | 1 | 1 | 1'b0 | spkr_mute | spkr_pdb | cpmp_avee_ok |
| | | 0 | spkr_offcal_cmp_r | spkr_offcal_cmp_l | spkr_offcal_bsy | spkr_offcal_clk_en |
| 8-0 | 0 | 1 | 1'b0 | ed_cp_volt_lvl | ed_out_en | oc_shutdown |
| | | 0 | oc_prot_decr_l | oc_prot_incr_l | spkr_oc_l | spkr_clk_clkgen |
| 8-1 | 1 | 1 | 1'b0 | analog_danbo[6] | analog_danbo[5] | analog_danbo[4] |
| | | 0 | analog_danbo[3] | analog_danbo[2] | analog_danbo[1] | analog_danbo[0] |
| 9-0 | 0 | 1 | 1'b0 | audio_pll_div2 | spkr_sc_status | dc_shutdown_r |
| | | 0 | dc_shutdown_l | oc_shutdown | spkr_pwr_dis | spkr_clk_clkgen |
| 9-1 | 1 | 1 | 1'b0 | micbias_dtct_filt | mb_sar_enable | mb_enable |
| | | 0 | mb_adc_sample | mb_adc_cmp | mb_dac7 | mb_dac6 |
| 10-0 | 0 | 1 | 1'b0 | analog_pdbs[1] | analog_pdbs[0] | hbridge_short_dig |
| | | 0 | spkr_prot_decr | spkr_prot_incr | spkr_pwr_dis | spkr_clk_clkgen |
| 10-1 | 1 | 1 | 1'b0 | tr_detect_en | micbias_dtct_filt | mb_flag |
| | | 0 | 1'b0 | 1'b0 | 1'b0 | 1'b0 |
| 11 | NA | 0 | DAC1_L_DSM Inputs | | | |
| 12-0 | 0 | 1 | 1'b0 | 1'b0 | hd_clk_off | rstn_if |
| | | 0 | audio_pll_out | iPORTG_STATE[2] | iPORTG_STATE[1] | iPORTG_STATE[0] |
| 12-1 | 1 | 1 | 1'b0 | avr_inn_l | avr_inp_l | ot_prot_decr |
| | | 0 | ot_prot_incr | porta_offcal_cmp_r | porta_offcal_cmp_l | temp_error |
| 13-0 | 0 | 1 | 1'b0 | analog_pdbs[3] | analog_pdbs[2] | clk_ctrl_sm |
| | | 0 | audio_pll_out | iPORTA_STATE[2] | iPORTA_STATE[1] | iPORTA_STATE[0] |
| 13-1 | 1 | 1 | 1'b0 | spdif_mod | iI2C_INT_UNSOL_MSG | spkr_error |
| | | 0 | spkr_async_clk | Modulator_out_test[1] | spdif_pin_low_power3 & spdif_keep_alive_en | spdif_pin_pwr_dn & ~spdif_keep_alive_en |
| 14 | NA | 1 | 1'b0 | analog_pdbs[5] | analog_pdbs[4] | hclk_reg_gated |
| | | 0 | audio_pll_out | iPORTE_STATE[2] | iPORTE_STATE[1] | iPORTE_STATE[0] |
| 15 | NA | 1 | 1'b0 | analog_pdbs[6] | cpmp_avee_ok | ed_cpvolt_lvl |
| | | 0 | ed_out_en | iPORTM_STATE[2] | iPORTM_STATE[1] | iPORTM_STATE[0] |

### 24.3.3.11 Digital_Test10 -> JACK Sense

Get Response - HD_A[27:0] = 0x1CCA000 Set - HD_A[27:11] = 0x1C4A I2C = 7228

| Bit | Default | Description |
|---|---|---|
| 11:9 | 0 | Reserved |
| 8 | RO | T/R sense compout debounce signal. (read only) |
| 7 | RO | Combo headset detection done (read only) |
| 6 | RO | Headset tip detection debounce signal. (read-only) |
| 5 | RO | Micbias detect debounce signal (read only) |
| 4 | RO | Jacksense SAR Busy Output (Read only) |
| 3:0 | RO | Jacksense SAR Output Data (Read only)<br>When jacks_data_mux = 0<br>Jack_data[3:0]<br>When jacks_data_mux = 1<br>{ portf_presence, porte_presence, portb_presence, porta_presence } |

### 24.3.3.12 Digital_Test11

Get Response - HD_A[27:0] = 0x1CCB000 Set - HD_A[27:11] = 0x1C4B I2C = 722C

| Bit | Default | Description |
|---|---|---|
| 11:10 | 00 | It's a read only resgister<br>00 – Headset not present<br>10 – headset_a present |

| | | |
|---|---|---|
| | | 01 – headset_b present<br>11 – doesn't exist |
| 9 | 0 | Combo button status (read only) |
| 9:8 | 0 | Reserved |
| 7 | 0 | Ored of Class D short circuit filter outputs (read only) |
| 6 | 0x0 | Calibration complete status (read only)<br>0 = not complete<br>1 = complete |
| 5:0 | 0x0 | Class-D calibration value (read only) |

### 24.3.3.13 Digital_Test12

Get Response - HD_A[27:0] = 0x1CCC000 Set – HD_A[27:11] = 0x1C4C I2C = 7230

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | In FPGA, only, control fpga_micbias_dtct when headset_detection is low. This can be used to toggle combo_button after portd is detected. |
| 10 | 0 | In FPGA, only, select a pulse or a high signal for fpga_micbis_dtct for config B detect test<br>0: high signal (no portd for configB detect test-default)<br>1: pulse (portd detected for configB detect test) |
| 9 | 0 | PortA inversion to support normally closed jack<br>0: normally open, no inversion (default)<br>1: normally closed, invert the SAR result (ie 0 = detect, 1 = no detect) |
| 8 | 0 | In FPGA, only, select a pulse or a high signal for fpga_micbis_dtct for config A detect test<br>0: high signal (no portd for configA detect test-default)<br>1: pulse (portd detected for configA detect test) |
| 7 | 1 | Combo jack mode<br>0: normal mode (default)<br>1: combo jack mode |
| 6 | 0 | Combo button enable<br>0: disable (default)<br>1: enable |
| 5:0 | 0x4 | Combo head set LP filter band select<br>0x4 = 5ms (default)<br>01 – sets filter Time constant to 82ms<br>02 – sets filter Time constant to 21ms<br>04 – sets filter Time constant to 5ms<br>08 – sets filter Time constant to 1.3ms<br>63 - sets filter Time constant to 35us |

### 24.3.3.14 Digital_Test13

Get Response - HD_A[27:0] = 0x1CCD000 Set - HD_A[27:11] = 0x1C4D I2C = 7234

| Bit | Default | Description |
|---|---|---|
| 11:9 | 0x000 | Reserved – Read as 0 |
| 8 | 0 | When set, Port E is treated as the combo headset output port for the purposes of headset detection (CTIA vs. OMTP)<br>When clear, port A is treated as the combo headset output port. |
| 7 | 0 | Reserved – Read as 0 |
| 6:0 | | HP calibrated read value<br>If codec_test_regs0[8], if set to 0 read port left channel value<br>If codec_test_regs0[8], if set to 1 read port right channel value.<br>Default value will be based on the left channel value calibrated on first good cad. |

### 24.3.3.15 Digital_Test14 -> GPIO4, GPIO5, OT, OC control

Get Response - HD_A[27:0] = 0x1CCE000 Set - HD_A[27:11] = 0x1C4E I2C=7238

| Bit | Default | Description |
|---|---|---|
| 11:10 | 2'b01 | GPIO5 pullup/down control |

| | | |
|---|---|---|
| 9:8 | 2'b01 | GPIO4 pullup/down control |
| 7 | 0 | Monitor spkr protection gain decrement and gain increment on GPIOs<br>PWR Avg gain incr → jsense_gpio_a<br>PWR Avg gain decr → portc_l_gpio_b |
| 6 | 0 | Monitor OC protect gain increment, OC protect gain decrement, OC shut down on GPIOs<br>OC protect gain increment → portc_l_gpio_b<br>OC protect gain decrement → portc_r_gpio_b<br>OC shut down → dmic_dat_gpio_e |
| 5 | 0 | Monitor hbridge_meas_low_l on dmic_dat_gpio_e and<br>hbridge_meas_low_r on dmic_clk_gpio_d |
| 4 | 0 | Monitor hbridge_meas_high_l on dmic_dat_gpio_e and<br>hbridge_meas_high_r on dmic_clk_gpio_d |
| 3 | 0 | Monitor basic temperature protection output on dmic_dat_gpio_e |
| 2 | 0 | Monitor OT protect gain increment, OT protect gain decrement, OT shut down on GPIOs<br>OT protect gain increment → portc_l_gpio_b<br>OT protect gain decrement → portc_r_gpio_b<br>OT shut down → jsense_gpio_a |
| 1 | 0 | 1 = Monitor class D error sticky on jsense_gpio_a |
| 0 | 0 | 1= Monitor DC Detection left and right channel and power averaging increment decrement value on GPIOs<br>DC Shutdown left → portc_l_gpio_b<br>DC Shutdown right → portc_r_gpio_b<br>SPKR_PWR_DIS → dmic_clk_gpio_d |

### 24.3.3.16 Digital_Test15 -> Jack Sense (continued)

Get Response - HD_A[27:0] = 0x1CCF000 Set - HD_A[27:11] = 0x1C4F I2C = 723C

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | Portd2a_delay_time base:<br>0: 83.3us (4 Azalia(48khz)) frames based.<br>1: 80 ms based. |
| 10:7 | 1 | Portd2a_delay_time<br>When portd (combo) is detected, the unsolicited message for portd will be sent first. This register programs the time elapsed before porta unsolicited message will be sent. This portd and porta delay is maintained when unsolicited message is sent for jack removal.<br>LSB bit is 83.3 us (4 azalia frames) or 80ms depend on bit 11<br>Default is 1 for 83.3us (4 azalia frames)<br>The max delay is: 15x80ms = 1.2s when use 80ms base. |
| 6:5 | 11 | Combo_mon_window<br>Monitor window after micbiasd_en forced to high and micbias_dtct filter is enabled before sampling micbias_dtct filter output to determine combo detection result.<br>00 = 50 mSec<br>01 = 125 mSec<br>10 = 200 mSec<br>11 = 275 mSec |
| 4:2 | 100 | Monitor_delay_time<br>000 = no delay<br>001 = 0.25 s<br>010 = 0.5 s<br>011 = 1 s<br>100 = 2 s<br>101 = 3 s<br>110 = 4 s<br>111 = no time-out |
| 1:0 | 11 | Combo_test_config<br>00 = no combo detect for both config A and config B<br>01 = combo detect for config A only<br>10 = combo detect for config B only<br>combo detect for config A first then for config B |

### 24.3.3.17 Digital_Test16 -> PLL Track

Get Response - HD_A[27:0] = 0x1C90000 Set - HD_A[27:11] = 0x1C10 I2C = 7080

| Bit | Default | Description |
|---|---|---|
| 11:8 | 0 | Reserved |
| 7 | 0 | PLL Track Limit Accum Error |
| 6 | 0 | Reserved |
| 5:4 | 2 | PLL Track Control – Target Select |
| 3:1 | 0 | Reserved |
| 0 | 1 | PLL Track Control – Enable |

### 24.3.3.18 Digital_Test17 -> PLL Track

Get Response - HD_A[27:0] = 0x1C91000 Set - HD_A[27:11] = 0x1C11 I2C=7084

| Bit | Default | Description |
|---|---|---|
| 11:8 | 0 | Reserved |
| 7:0 | 0x18 | PLL Track Threshold 0 |

### 24.3.3.19 Digital_Test18 -> PLL Track

Get Response - HD_A[27:0] = 0x1C92000 Set - HD_A[27:11] = 0x1C12 I2C=7088

| Bit | Default | Description |
|---|---|---|
| 11:8 | 0 | Reserved |
| 7:0 | 0x24 | PLL Track Threshold 1 |

### 24.3.3.20 Digital_Test19 -> PLL Track

Get Response - HD_A[27:0] = 0x1C93000 Set - HD_A[27:11] = 0x1C13 I2C=708C

| Bit | Default | Description |
|---|---|---|
| 11:8 | 0 | Reserved |
| 7:0 | 0x01 | PLL Track Adjust 0 |

### 24.3.3.21 Digital_Test20 -> PLL Track

Get Response - HD_A[27:0] = 0x1C94000 Set - HD_A[27:11] = 0x1C14 I2C=7090

| Bit | Default | Description |
|---|---|---|
| 11:8 | 0 | Reserved |
| 7:0 | 0x02 | PLL Track Adjust 1 |

### 24.3.3.22 Digital_Test21

Get Response - HD_A[27:0] = 0x1C95000 Set - HD_A[27:11] = 0x1C15 I2C=7094

| Bit | Default | Description |
|---|---|---|
| 11:0 | 0 | Spkr gain L channel LSB[11:0] |

### 24.3.3.23 Digital_Test22

Get Response - HD_A[27:0] = 0x1C96000 Set - HD_A[27:11] = 0x1C16 I2C=7098

| Bit | Default | Description |
|---|---|---|
| 11:0 | 0 | Spkr gain R channel LSB[11:0] |

#### 24.3.3.24 Digital_Test23

Get Response - HD_A[27:0] = 0x1C97000 Set - HD_A[27:11] = 0x1C17 I2C=709C

| Bit | Default | Description |
|---|---|---|
| 11:8 | 0x10 | Spkr gain R channel MSB[17:12] |
| 5:0 | 0x10 | Spkr gain L channel MSB[17:12] |

### 24.3.4 Analog Codec Test registers

#### 24.3.4.1 Analog_Test0 → PWR_TOP Controls

Get Response - HD_A[27:0] = 0x1CB0000 Set - HD_A[27:11] = 0x1C30 I2C = 7180

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | Digital LDO low power mode<br>1 = place digital LDO into low power mode |
| 10 | 0 | Analog LDO low power mode<br>1 = place analog LDO into low power mode |
| 9 | 0 | Control current limit in digital LDO<br>1 = Disable current limit |
| 8 | 0 | Control current limit in analog LDO<br>1 = Disable current limit |
| 7:5 | 000 | Trimming for VREF bandgap voltage<br>000 = 1.2133 V (Nominal)<br>001 = Nominal + 33mV<br>010 = Nominal + 66mV<br>011 = Nominal + 99mV<br>100 = Nominal + 133mV<br>101 = Nominal – 28mV<br>110 = Nominal – 56mV<br>111 = Nominal – 84mV |
| 4:2 | 000 | Control VDDCORE Digital LDO output voltage<br>000 = 1.26 V<br>001 = 1.01 V<br>010 = 1.06 V<br>011 = 1.11 V<br>100 = 1.16 V<br>101 = 1.21 V<br>110 = 1.26 V<br>111 = 1.31 V |
| 1:0 | 00 | PCBEEP threshold control |

#### 24.3.4.2 Analog_Test1 → CSDAC controls

Get Response - HD_A[27:0] = 0x1CB1000 Set - HD_A[27:11] = 0x1C31 I2C = 7184

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | 1 = Enable monomode on portA |
| 10:9 | 00 | Control bias current for 6 CSDAC main opamps<br>00 = 1.50 uA<br>01 = 1.00 uA<br>10 = 2.50 uA<br>11 = 2.00 uA |
| 8 | 0 | ClassD offset calibration bias current control<br>= Doubles the calibration range |
| 7 | 0 | 1 = Disable OOB DAC1 used for Headphone driver |
| 6:4 | 000 | ClassD gain fine tuning controls<br>000 = 0dB (default)<br>001 = +0.48 dB<br>010 = +0.32 dB<br>011 = +0.16 dB |

| | | |
|---|---|---|
| | | 100 = 0 dB<br>101 = -0.17 dB<br>110 = -0.33 dB<br>111 = -0.5 dB |
| 3:0 | 0101 | Control CSDAC gain for Class D<br>Gain (dB) = -10 dB + decimal_equivalent_code<br>0000 = -10 dB<br>0001 = -9 dB<br>0010 = -8 dB<br>0011 = -7 dB<br>0100 = -6 dB<br>0101 = -5 dB<br>0110 = -4 dB<br>0111 = -3 dB<br>1000 = -2 dB<br>1001 = -1 dB<br>1010 = 0 dB<br>1011 = 1 dB<br>1100 = 2 dB<br>1101 = 3 dB<br>1110 = 4 dB<br>1111 = 5 dB |

### 24.3.4.3 Analog_Test2 → RX Controls

Get Response - HD_A[27:0] = 0x1CB2000 Set - HD_A[27:11] = 0x1C32 I2C = 7188

| Bit | Default | Description |
|---|---|---|
| 11:9 | 000 | Control bias current for AAF opamp<br>000 = 1.50 uA<br>001 = 1.75 uA<br>010 = 1.00 uA<br>011 = 1.25 uA<br>100 = 2.50 uA<br>101 = 2.75 uA<br>110 = 2.00 uA<br>111 = 2.25 uA |
| 6:4 | 000 | Control base bias current for analog top<br>000 = 1.1 uA<br>001 = 0.7 uA<br>010 = 0.8 uA<br>011 = 0.9 uA<br>100 = 1.0 uA<br>101 = 1.1 uA<br>110 = 1.2 uA<br>111 = 1.3 uA |
| 5:4 | 00 | Micbiasd output current comparator threshold<br>0 0 = ~66% (% of applied micbias voltage / 2.2K Ohms / 8)<br>0 1 = ~55%<br>1 0 = ~88%<br>1 1 = ~77% |
| 3 | 0 | 1 = enable switch to ground on PGA inputs when not selected |
| 2:0 | 000 | Control bias current for PGA opamp<br>000 = 2.50 uA<br>001 = 2.75 uA<br>010 = 2.00 uA<br>011 = 2.25 uA<br>100 = 1.50 uA<br>101 = 1.75 uA<br>110 = 1.00 uA<br>111 = 1.25 uA |

### 24.3.4.4 Analog_Test3 → Micbiasb + Micbiasd + PCBEEP + VREFP Controls

Get Response - HD_A[27:0] = 0x1CB3000 Set - HD_A[27:11] = 0x1C33 I2C = 718C

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | Micbiasd output grounding switch control<br>0 = switch open (not grounded)<br>1 = switch closed (grounded) |
| 10 | 0 | Sets lf_rcosc_clk frequency<br>1 = 500KHz<br>0 = 200KHz |
| 9 | 0 | Power down PCBEEP PAD<br>0 = PCBEEP PAD Enabled<br>1 = PCBEEP PAD disabled |
| 8 | 0 | 1 = Block PCBEEP signal |
| 7 | 0 | 1 = Disable scanmode signal |
| 6 | 0 | 1 = bypass avdd5V comparator control on lpmicbias voltage |
| 5 | 0 | 1 = bypass avdd5V comparator control on LDO gain |
| 4 | 0 | 0 = LDO output is set to 2.9V when comparator is bypassed by setting AT3[5]<br>1 = LDO output is set to 3.4V when comparator is bypassed by setting AT3[5] |
| 3 | 0 | Headset Micbias Boost LOW (this allows the headset micbias boost voltage to be 1.9V IF bit [1] is a "1" as well<br>0 = Normal Headset Micbias Boost = 2.9V<br>1 = Headset Micbias Boost = 1.9V |
| **2** | **0** | **Micbias_dtct comparator setting**<br>**0 = current comparator mode**<br>**1 = voltage comparator mode** |
| 1:0 | 00 | Micbias voltage setting<br>00 → 2.9V<br>01 → 2.78V<br>10 → 2.68V<br>11→ 2.55V |

### 24.3.4.5 Analog_Test4→ PLL CONTROLS

Get - HD_A[27:0] = 0x1CB4000 Set - HD_A[27:11] = 0x1C34 I2C = 7190

| Bit | Default | Description |
|---|---|---|
| 11:9 | 000 | Input Divide Ratio<br>000 = div by 1 (bypass)<br>001 = div by 2<br>010 = div by 3<br>011 = div by 4<br>100 = div by 5<br>101 = div by 6<br>110 = div by 7<br>111 = div by 8 |
| 8 | 0 | SPARE |
| 7:6 | 00 | Clock test point control bits now output at two ports from the analog going to the GPIOs<br>PORT#1<br>00 = normal operation, set to 0<br>01 = reference divided clock<br>10 = feedback clock<br>11 = vco clock<br>PORT#2<br>00 = normal operation, set to 0<br>01 = feedback clock<br>10 = vco clock<br>11 = reference divided clock |
| 5 | 0 | 1 = Disables yank boost block logic |
| 4 | 0 | 1 = Disable pll_clkin detection circuit |
| 3:2 | 00 | Phase Detector, 'Dn', reset delay, control bits |

| | | |
|---|---|---|
| 1:0 | 00 | Phase Detector, 'Up', reset delay, control bits |

### 24.3.4.6 Analog_Test5 → PLL CONTROLS

Get Response - HD_A[27:0] = 0x1CB5000 Set - HD_A[27:11] = 0x1C35 I2C = 7194

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | Spares |
| 10 | 0 | 1 – Bypasses r3 resistor in LPF. |
| 9 | 0 | 1 = r3 is controlled by rc3_TEST<1:0> control bits<br>0 = r3 is controlled automatically by TEST_bw<1:0> control bits |
| 8 | 0 | Chargepump Reference, fixed or tracking |
| 7:6 | 0 | Third parasitic pole control bit |
| 5:4 | 0 | Bandwidth control bits |
| 3:2 | 0 | Charge Pump Reference Current Scale |
| 1:0 | 0 | Charge Pump Reference Current Adjustment |

### 24.3.4.7 Analog_Test6 → PLL CONTROLS

Get Response - HD_A[27:0] = 0x1CB6000 Set - HD_A[27:11] = 0x1C36 I2C = 7198

| Bit | Default | Description |
|---|---|---|
| 11:0 | 3'h59A | Feedback Divide Ratio, Fractional part – LSB [11:0] |

### 24.3.4.8 Analog_Test7 → PLL CONTROLS

Get Response - HD_A[27:0] = 0x1CB7000 Set - HD_A[27:11] = 0x1C37 I2C = 719C

| Bit | Default | Description |
|---|---|---|
| 11:10 | 0 | Spares |
| 9 | 0 | 0 = DSM dither is off<br>1 = DSM dither is on |
| 8 | 0 | 0 = normal fractional operation<br>1 = integer mode enabled, DSM disabled |
| 7:0 | 8'hA7 | Feedback Divide Ratio, Fractional part – middle bits [19:12] |

### 24.3.4.9 Analog_Test8 → PLL CONTROLS

Get Response - HD_A[27:0] = 0x1CB8000 Set - HD_A[27:11] = 0x1C38 I2C = 71A0

| Bit | Default | Description |
|---|---|---|
| 11:7 | 0 | Spares |
| 6:0 | 7'd 23 | Feedback Divide Ratio<br>Decimal ratio = Binary code + 1 |

### 24.3.4.10 Analog_Test9 → HP controls

Get Response - HD_A[27:0] = 0x1CB9000 Set - HD_A[27:11] = 0x1C39 I2C = 71A4

| Bit | Default | Description |
|---|---|---|
| 11 | 1 | 1 = reverts the DFET enable logic to depend on the supply ramping |
| 10 | 0 | Headphone Current Limit, disable |
| 9 | 0 | Tip Detect, force high |
| 8 | 1 | Tip Detect, disable |
| 7:6 | 0 | Headphone Quiescent Control Current Settings<br>00 = 40uA Ireplica, 400uA Iquiescent<br>01 = 20uA Ireplica, 200uA Iquiescent |

| | | |
|---|---|---|
| | | 10 = 80uA Ireplica, 800uA Iquiescent<br>11 = 60uA Ireplica, 600uA Iquiescent |
| 5:4 | 0 | Headphone Current Limit Control Settings<br>00 = 100mA Current Limit<br>01 = 140mA Current Limit<br>10 = 60mA Current Limit<br>11 = 80mA Current Limit |
| 3 | 0 | Headset micbias voltage setting in bypass mode<br>0 = 2.9V when avdd5V comparator is bypassed<br>1 = 2.6V when avdd5V comparator is bypassed |
| 2 | 0 | 1= bypasses vdd5V comparator control on headset micbias voltage setting |
| 1:0 | 0 | Headphone Gain Settings<br>0db: -3dB: -9dB.<br>00 = 0dB<br>01 = -3dB<br>10 = -6dB<br>11 = -9dB |

### 24.3.4.11 Analog_Test10 → CPUMP Controls

Get Response - HD_A[27:0] = 0x1CBA000 Set - HD_A[27:11] = 0x1C3A I2C = 71A8

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | Disable skip regulation in D1<br>0: skip regulation ON<br>1: skip off (CP runs at fixed frequency) |
| 10 | 0 | Disable linear regulation in D1<br>0: linear regulation ON<br>1: linear regulation off (CP runs at max strength in D1 active mode) |
| 9 | 0 | Disable skip regulation in D2<br>0: skip regulation ON<br>1: skip off (CP runs at fixed frequency) |
| 8 | 0 | Disable linear regulation in D2<br>0: linear regulation ON<br>1: linear regulation off (CP runs at max strength in D2 active mode) |
| 7 | 0 | 1: set CP to Hi-Z mode, all switches open |
| 6:5 | 0 | Select CP switching frequency<br>00: 6.144Mhz<br>01: 3.072MHz<br>10: 1.536MHz<br>11: 768kHz |
| 4:2 | 0 | Control for digital output test mux MSBs test_digout<3:0><br>000: test_digout<3:0> = 0000<br>001: test_digout<3:0> = [q1,q0] of top SM; timer_done; first_skip_n<br>010: test_digout<3:0> = [q1,q0] of D2 SM; pump_p, pump_n<br>011: test_digout<3:0> = state of D1/D2 SM (00 = TN; 01 = TP; 10 = S2; 11 = S1); pump_on,d1mode<br>100: test_digout<3:0> = count_n<2:1>, count_p<2:1><br>101: test_digout<3:0> = clk_in, ctrl_dmode, vout_ok, pdni0<br>110: test_digout<3:0> = 0, por, vref_ok, pdb<br>111: test_digout<3:0> = HiZ |
| 1:0 | 0 | target voltage for negative rail in D1 (% of avdd18)<br>00: 90%<br>01: 87.5%<br>10: 92.5%<br>11: 97.5% |

### 24.3.4.12 Analog_Test11 → CPUMP+ MICASBISD controls

Get Response - HD_A[27:0] = 0x1CBB000 Set - HD_A[27:11] = 0x1C3B I2C = 71AC

| Bit | Default | Description |
|---|---|---|
| 11:9 | 000 | Depletion FET, Generate programable OFF voltage for DFET |

| | | |
|---|---|---|
| | | 000 = -1.2V<br>001 = -1.16V<br>010 = -1.11V<br>011 = -1.05V<br>100 = -1.05V<br>101 = -0.98V<br>110 = -0.9V<br>111 = -0.8V |
| 8 | 0 | 1= Disable CPUMP and enable DFET's |
| 7:4 | 0000 | Headset Micbias Resistor Value Adjustment Trim<br>0000 Selects the default setting (hardwired to 0101, 2.20 K Ohms)<br>0001 2.00 K Ohms<br>0010 2.05 K Ohms<br>0011 2.10 K Ohms<br>0100 2.15 K Ohms<br>0101 2.20 K Ohms (Default)<br>0110 2.25 K Ohms<br>0111 2.30 K Ohms<br>1000 2.35 K Ohms<br>1001 2.40 K Ohms (max value)<br>1010 2.40 K Ohms<br>1011 2.40 K Ohms<br>1100 2.40 K Ohms<br>1101 2.40 K Ohms<br>1110 2.40 K Ohms<br>0 K Ohms |
| 3:2 | 00 | target voltage for positive rail in D2 (% of avdd18/2)<br>00: 95%<br>01: 87.5%<br>10: 92.5%<br>11: 97.5% |
| 1:0 | 00 | target voltage for negative rail in D2 (% of avdd18/2)<br>00: 80%<br>01: 75%<br>10: 85%<br>11: 95% |

### 24.3.4.13 Analog_Test12 → Class-D CONTROLS

Get Response - HD_A[27:0] = 0x1CBC000 Set - HD_A[27:11] = 0x1C3C I2C = 71B0

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | Int1_fstg_bst<br>0 = First stg opamp of Integrator1 set by AT21[3:2]<br>1 = First stg opamp of Integrator1 set by AT21[3:2] + 50% of the Nominal value |
| 10:8 | 000 | Slew_rate<2:0>: power stage drivers slew-rate ctrl speed<br>000 = 52ns<br>001 = 89ns<br>010 = 72ns<br>011 = 60ns<br>100 = 52ns<br>101 = 46ns<br>110 = 40ns<br>111 = 37ns |
| 7 | 0 | Reduce N-thresh<br>0 – Short circuit NMOS threshold is the one specified in AT20<9:7><br>1 – Short circuit NMOS threshold is 4% smaller than the one specified in AT20<9:7> |
| 6 | 0 | Reduce P-thresh<br>0 – Short circuit PMOS threshold is the one specified in AT20<9:7><br>1 – Short circuit PMOS threshold is 4% smaller than the one specified in AT20<9:7> |
| 5 | 0 | SPARE |

| | | |
|---|---|---|
| 4:3 | 00 | Rasc<1:0>: saturation ctrl system ctrl bits<br>Rasc<0> = sat ctrl enable. 0 = sat ctrl off, 1 = sat ctrl on<br>Rasc<1> = 3rd int reset enable. 0 = 3rd reset off, 1 = 3rd int reset on |
| 2 | 0 | Pre-driver dead time amount<br>0 = 0.2n<br>1 = 2.9n |
| 1:0 | 00 | Slrc_count_max<1:0>: selfrecovery times after a short<br>00 = 16 times<br>01 = 8 times<br>10 = 4 times<br>11 = Infinite (Selfrecovery always ON) |

### 24.3.4.14 Analog_Test13→ Class-D CONTROLS

Get Response - HD_A[27:0] =0x1CBD000 Set - HD_A[27:11] = 0x1C3D I2C = 71B4

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | Fast_mode = Slew rate ctrl disable<br>0 = Slew rate ctrl enabled<br>1 = Slew rate ctrl disabled |
| 10 | 0 | Oc_sc_low_curr_mode: enable the sc and oc ctrl test mode reducing the ctrl thresholds<br>0 = Regular threshold levels<br>1 = Thresholds scale as reported<br>000 = Depending on the Nominal status<br>001 = 133mA<br>010 = 167mA<br>011 = 200mA<br>100 = 233mA<br>101 = 266mA<br>110 = 300mA<br>111 = 333mA |
| 9:7 | 000 | Sc_thr_ctrl<2:0>: short circuit ctrl thresholds<br>000 = 1.6A (Nominal)<br>001 = 0.8A<br>010 = 1.0A<br>011 = 1.2A<br>100 = 1.4A<br>101 = 1.6A<br>110 = 1.8A<br>111 = 2.0A |
| 6:4 | 0 | Oc_thr_ctrl<2:0>: over current ctrl thresholds<br>000 = 1.2A (Nominal)<br>001 = 0.8A<br>010 = 1.0A<br>011 = 1.2A<br>100 = 1.4A<br>101 = 1.6A<br>110 = 1.8A<br>111 = 2.0A |
| 3:1 | 0 | Test_mux<2:0>: test outputs selector<br>000 = hi-z, hi-z, hi-z, hi-z<br>001 = win_en, spkr_sc_err, win_count, sc_hit<br>010 = sc_m_l_sync, sc_m_r_sync, sc_p_l_sync, sc_p_r_sync<br>011 = sc_state_reg<2>, sc_state_reg<1>, sc_state_reg<0>, n/c<br>100 = ts_error_ana, ts_error_dig, ts_state_reg<1>, ts_state_reg<2><br>101 = pdb_bias, pdb_clk_dig_rstb, hi_z<br>110 = test_comp_dgltch_m, test_comp_dgltch_p, test_satn, test_satp<br>111 = test_hout_fall_m, test_hout_fall_p, test_hout_rise_m, test_hout_rise_p |
| 0 | 0 | Ctrl_delay: start up state machine programmability<br>0 = 4 separated start-up phases (pdb_bias, pdb_clk, dig_rstb, hi-z)<br>1 = 3 separated start-up phases (dig_rstb & hi-z rise together) |

### 24.3.4.15 Analog_Test14 → Class-D CONTROLS

Get Response - HD_A[27:0] = 0x1CBE000 Set - HD_A[27:11] = 0x1C3E I2C = 71B8

| Bit | Default | Description |
|---|---|---|
| 11:10 | 00 | Edge_det_rise_ctrl<1:0><br>Edge_det_rise_ctrl<1><br>0: use default cap connected to rise_in<br>1: switch in rise_in_sw and increase cap<br>Edge_det_rise_ctrl<0><br>0: use RES = 250kohms<br>1: use RES = 500kohms |
| 9:8 | 00 | Edge_det_fall_ctrl<1:0><br>Edge_det_fall_ctrl<1><br>0: use default cap connected to fall_in<br>1: switch in fall_in_sw and increase cap<br>Edge_det_fall_ctrl<0><br>0: use RES = 250kohms<br>1: use RES = 500kohms |
| 7:6 | 00 | Ib_car<1:0>: PWM carrier amplitude<br>00 = 15.5uA (Nominal)<br>01 = 12uA<br>10 = 25uA<br>11 = 20uA |
| 5:4 | 00 | Ib_comp<1:0>: PWM comparators bias current amplitude<br>00 = 1uA (Nominal)<br>01 = +25%<br>10 = -50%<br>11 = -25% |
| 3:2 | 00 | Ib_fstg<1:0>: first stage bias current amplitude<br>00 = 1uA (Nominal)<br>01 = +25%<br>10 = -50%<br>11 = -25% |
| 1:0 | 00 | Ib_sstg<1:0>: second stage bias current amplitude<br>00 = 1uA (Nominal)<br>01 = +25%<br>10 = -50%<br>11 = -25% |

### 24.3.4.16 Analog_Test15→ Class-D CONTROLS

Get Response - HD_A[27:0] = 0x1CBF000 Set - HD_A[27:11] = 0x1C3F I2C = 71BC

| Bit | Default | Description |
|---|---|---|
| 11:9 | 00 | Sc_thresh<2:0>: number of sc events to be counted in order to trigger the sc-ctrl<br>000 = 2^17<br>001 = 2^16<br>010 = 2^15<br>011 = 2^14<br>100 = 2^13<br>101 = 2^12<br>110 = 2^11<br>111 = 2^10 |
| 8:7 | 00 | Sc_count_win<1:0>: time window during which the sc-ctrl increments the sc events counter<br>00 = 2^19 (1.36s)<br>01 = 2^18 (681ms)<br>10 = 2^15 (86ms)<br>11 = 2^13 (22ms) |
| 6 | 0 | Self_recovery_dis: if not disabled, the self-recovery turns back on the class-D after a fixed amount of time (~5s) after the sc-ctrl has switched it off<br>0 = Self recovery on<br>1 = Self recovery off |
| 5 | 0 | Enable Battery Voltage Monitor ADC |

| | | |
|---|---|---|
| | | 0 = Disable MADC<br>1 = Enable MADC |
| 4 | 0 | Force_hi_z: manual hi-z override ctrl bit<br>0 = class-D output managed by the protection logics & start-up state machine<br>1 = class-D output in hi-z |
| 3:0 | 0000 | Ts_th<2:0>: Temp sensor threshold (HT: hysteresis high temp, LT: hysteresis low temp)<br>0000 = HT:130°C, LT:108°C<br>0001 = HT:140°C, LT:120°C<br>0010 = HT:120°C, LT:100°C<br>0011 = HT:108°C, LT:88°C<br>0100 = HT:100°C, LT:80°C<br>0101 = HT:90°C, LT:70°C<br>0110 = HT:80°C, LT:60°C<br>0111 = HT:70°C, LT:50°C<br>1000 = HT:60°C, LT:40°C<br>1001 = HT: 50°C, LT:30°C<br>1010 = HT:40°C, LT:20°C<br>1011 = HT:30°C, LT:10°C<br>1100 = HT:20°C, LT:0°C<br>1101 = HT:10°C, LT: -6°C<br>1110 = HT:5°C, LT: -6°C<br>1111 = HT:0°C, LT: -6°C |

### 24.3.4.17 Analog_Test16 → Class-D CONTROLS

Get Response - HD_A[27:0] = 0x1CD0000 Set - HD_A[27:11] = 0x1C50 I2C = 7280

| Bit | Default | Description |
|---|---|---|
| 11 | 0 | Edge_det_filt_byp<br>0 = Edge detector filter enabled<br>1 = Edge detector filter disabled |
| 10 | 0 | Dtc_logic_byp<br>0 = Dead time ctrl deglitch enabled<br>1 = Dead time ctrl deglitch disabled |
| 9 | 0 | Dtc_filt_en<br>0 = Dead time ctrl filter enabled<br>1 = Dead time ctrl filter disabled |
| 8 | 0 | Comp_dgltch_byp: loop filter comparator deglitcher bypass<br>0 = deglitcher on<br>1 = deglitcher bypassed |
| 7 | 0 | Edge_detect_dis: edge detector block disable<br>0 = edge detector enabled<br>1 = edge detector disabled |
| 6:5 | 00 | Ckr_ph<1:0>: Phase shift between L & R channel clks<br>00 = 45°<br>01 = 0°<br>10 = 135°<br>11 = 90° |
| 4 | 0 | Off_cal_sign: offset calibration SAR loop sign<br>0 = Nominal<br>1 = Inverted |
| 3 | 0 | Sw_ff_ctrl: Loop filter feed-forward path switches<br>0 = loop filter feed forward paths open during offset calibration<br>1 = loop filter feed forward paths always closed |
| **2:0** | **--** | **Spare** |

### 24.3.4.18 Analog_Test17 → Class-D CONTROLS Protected register

Get Response - HD_A[27:0] = 0x1CD1000 Set - HD_A[27:11] = 0x1C51 I2C = 7284

| Bit | Default | Description |
|---|---|---|
| 11 | 0<br>Protected | Class-D temperature sensor enable control (ts_disable):<br>0 = powered on (default)<br>1 = powered off and output forced low |
| 10 | 0<br>Protected | Class-D POR override (por_override):<br>0 = use normal internal POR circuitry<br>1 = bypass internal circuitry and force POR = 1 |
| 9 | 0<br>Protected | Class-D over current ignore (sc_ignore) (both channels)<br>0 = normal operation (default)<br>1 = ignore sc control outputs |
| 8 | 0<br>Protected | Class-D clock detection bypass (spkr_clk_detect_byp)<br>0 = enabled (default)<br>1 = bypass clock detection and force clk_good = 1 |
| 7 | 0<br>Protected | Class-D H-bridge auto dead-time control (autodtc_dis)<br>0 = enable auto break-before-make DTC<br>1 = disable auto circuitry and use static dead-time<br>In both cases, dead-time amount set by Ana_test19<2:0><br>To avoid current shoot-through is advisable to maintain autodtc_dis=1 only with fast_mode ON (Ana_test20<11>) |
| 6 | 0<br>Protected | Class-D loop bypass control (spkr_test_en):<br>0 = normal mode: loop output is fed to H-bridge<br>1 = bypass mode: ignore loop and feed external PWM signals into the H-bridge |
| 5 | 0<br>Protected | Sc_oc_dis: sc and oc sensing current mirror disable<br>0 = sc and oc sensing current mirror enabled<br>1 = sc and oc sensing current mirror disabled |
| 4 | 0 | Sc_global_dis: sc logic disable<br>0 = sc logic ON<br>1 = sc logic OFF |
| 3 | 0 | Short circuit control logic test mode enable::<br>0 – sc events threshold = value set by AT15<11:9>, sc_win = value set by AT15<8:7><br>1 – sc events threshold = 64, sc_win = 333us |
| 2:0 | -- | Spares |

# 25Design for Test

## 25.1 General Description

This chapter documents the overall use of DFT and its interface at chip-level I/O.

This document will only discuss aspects of the controller related to DFT - internal scan, memory built-in self-test (MBIST). Please note in the following paragraphs, DFT test modes indicates scan, MBIST are referred as functional test modes, if just generally saying test modes, then it means all test modes.

Scan and MBIST will be inserted by Cadence flow. Scan mode signal is generated by internal analog AFE (scanmode pin), MBIST is achieved by MDA (MBIST Direct Access) form and test on function mode by configure MBIST enable register through I2C port.

For Oceanside, the DFT logic is quite independent to function so it is inserted in netlist level, and its function will be guaranteed by ATPG tool of Cadence, or gate level simulation with the pattern generated by Cadence tool.

## 25.2 Scan

### 25.2.1 Oceanside scan pin mapping

**Table 25-1: Oceanside scan pins mapping**

| Signal | Oceanside pin | Direction | Description |
|---|---|---|---|
| TEST | oceanside_ana_top/scanmode | OUT | AFE output |
| SCAN_IN | spkrmuteb_gpio1 | IN | |
| SCAN_OUT | dmic_clk1_gpio2 | OUT | |
| SCAN_IN | dmic_dat1_gpio3 | IN | |
| SCAN_OUT | eapd | OUT | |
| SCAN_EN | hsync | IN | |
| SCAN_RESET | hrstn | IN | |
| SCAN_CLK | hbclk | IN | |
| COMPRESS_EN | music_req_gpio0 | IN | |

## 25.2.2 IO timing (TBD)

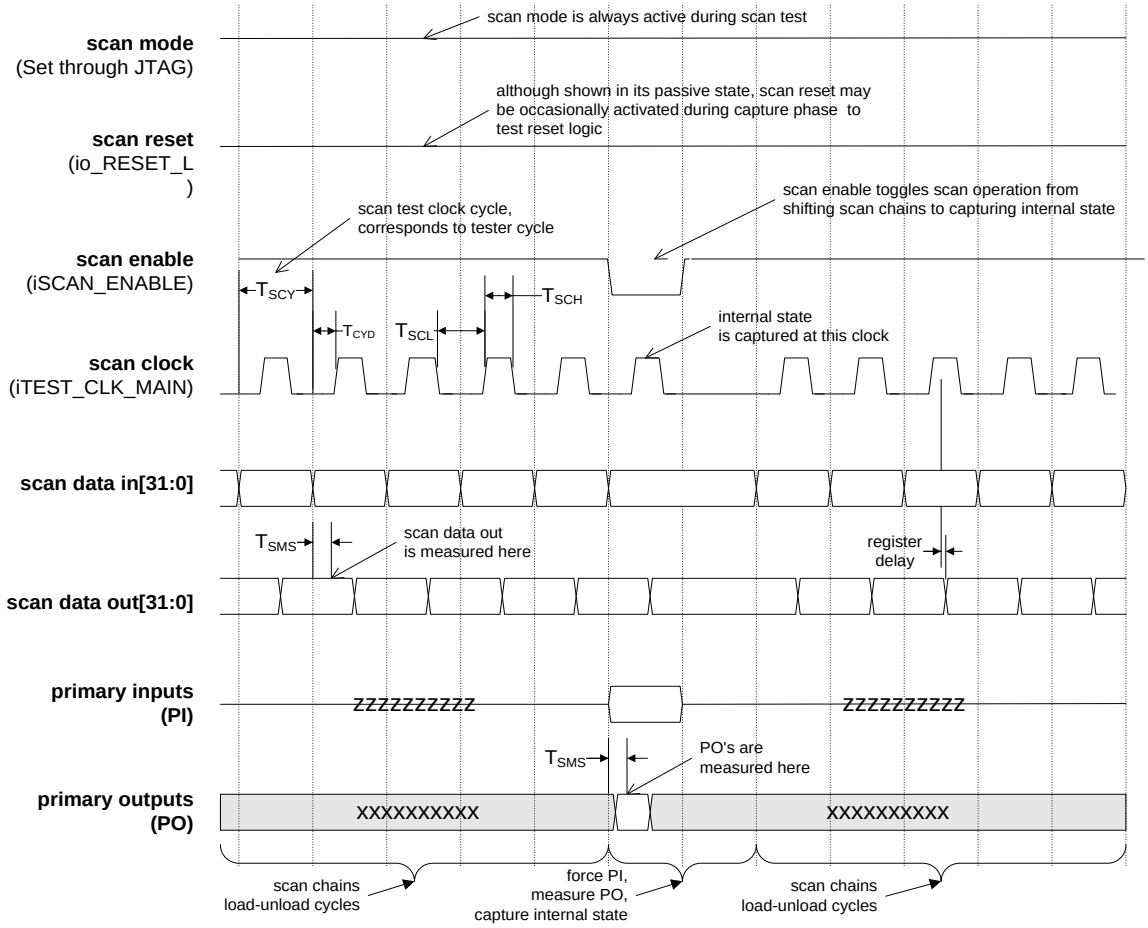



**Figure 25.2.1 IO timing for normal scan**

**Table 25-2: Normal Scan Mode Timing**

| Parameter | Symbol | Min | Max | Units |
|---|---|---|---|---|
| Scan Mode Cycle | $T_{SCY}$ | - | 100 | ns |
| Scan Mode Cycle Delay | $T_{CYD}$ | | 60 | ns |
| Scan Mode Cycle Low | $T_{SCL}$ | - | 70 | ns |
| Scan Mode Cycle High | $T_{SCH}$ | - | 30 | ns |
| Scan Mode Output Measure Delay | $T_{SMS}$ | | 50 | ns |

### 25.2.3 Compressed Scan

Oceanside has compression logic to increase the number of scan chains and reduce the ATE runtime. 2 scan channels are selected while the number of internal scan chains is increased to 20.

Pin music_req_gpio0 can be used to bypass compression logic. (1: compressed chain, 0: bypassed chain)

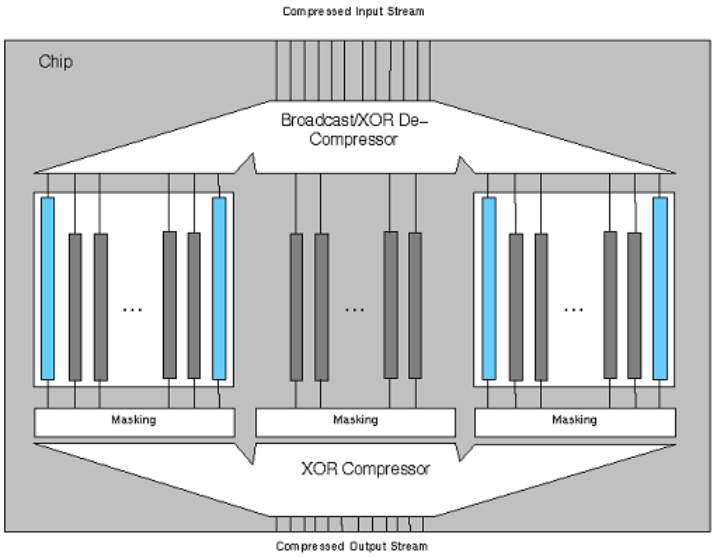



**Figure 25.2.2 Compress scan structure**

### 25.2.4 Scan coverage (TBD)

Scan coverage should be higher than 99%.

## 25.3 Memory BIST

### 25.3.1 MBIST overview

MBIST is controlled by MBIST enable register in design, March-LR algorithm is selected for all memory, and Cadence also has other test such as connection test is implemented by default. All memories have bypass logic used scanmode signal to improve scan coverage, so do not need memory ATPG models developed for each memory to improve coverage.

MBIST verification and pattern generation are handled by Cadence tool and not transparency to us, it starts MBIST with MBIST enable signal, and then register done/pass/fail signals to result register, if chips fail on production test, the log files can be loaded back to Cadence debug tool and it will report which memory has failure. This is quite same from what we did with Mentor's tool.

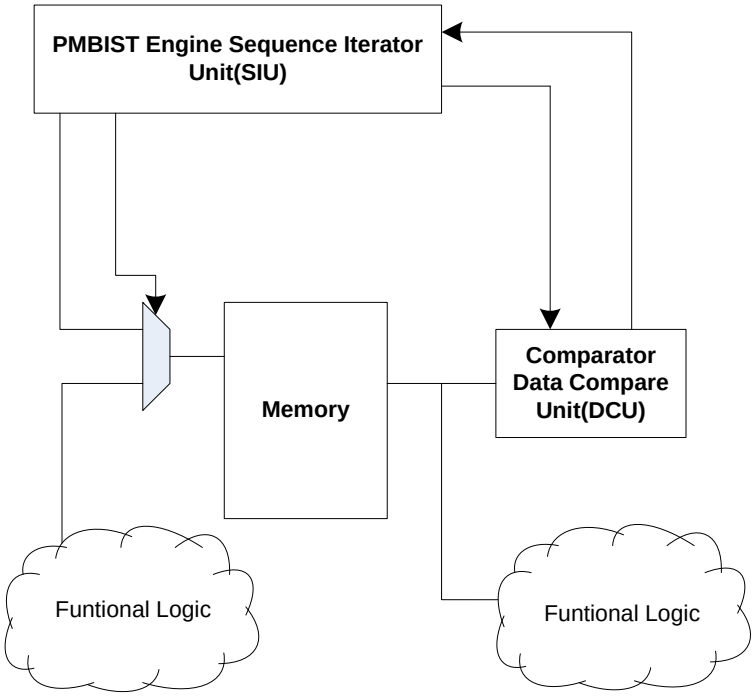



**Figure 25.3.1 Memory BIST Structure**

### 25.3.2 MBIST Groups

| Group | Memory | Clock (highest) | MBIST Location |
|---|---|---|---|
| Group 1 | dig_top/dig_core/codec_dsp/adcs_dsp/adc_mbist_con/mem0 | CLK_MBIST_ADC (6.144Mhz) | dig_top/dig_core/codec_dsp/adcs_dsp/adc_mbist_con |
| | dig_top/dig_core/codec_dsp/adcs_dsp/adc_mbist_con/mem1 | | |
| | dig_top/dig_core/codec_dsp/adcs_dsp/adc_mbist_con/mem2 | | |
| | dig_top/dig_core/codec_dsp/adcs_dsp/adc_mbist_con/mem3 | | |
| Group 2 | dig_top/dig_core/codec_dsp/dacs_dsp/dac_mbist_con/RF1P_22X20_instance_0 | CLK_MBIST_DAC (36.8Mhz) | dig_top/dig_core/codec_dsp/dacs_dsp/dac_mbist_con |
| | dig_top/dig_core/codec_dsp/dacs_dsp/dac_mbist_con/RF1P_88X32_instance_0 | | |
| | dig_top/dig_core/codec_dsp/dacs_dsp/dac_mbist_con/RF1P_88X32_instance_1 | | |
| | dig_top/dig_core/codec_dsp/dacs_dsp/dac_mbist_con/SRAM1P_264X19_instance_0 | | |
| Group 3 | dig_top/dig_core/hd_interface/u_ext_acc_ctrl/u_SRAM1P_128X32_mbist/u_SRAM1P_128X32 | CLK_MBIST_WSQ (30Mhz) | dig_top/dig_core/hd_interface/u_ext_acc_ctrl/u_SRAM1P_128X32_mbist |
| Group 4 | dig_top/dig_core/u_src0_mbist/u0_src_adc_0 | CLK_MBIST_SRC (24.576Mhz) | dig_top/dig_core/ |
| | dig_top/dig_core/u_src0_mbist/u0_src_adc_1 | | |
| | dig_top/dig_core/u_src0_mbist/u0_src_dac_0 | | |
| | dig_top/dig_core/u_src0_mbist/u0_src_dac_1 | | |
| | dig_top/dig_core/u_src1_mbist/u1_src_adc_0 | | |
| | dig_top/dig_core/u_src1_mbist/u1_src_adc_1 | | |

**Table 25-3 Oceanside mbist group**