Synaptics®

ADVANCING
THE HUMAN
INTERFACE

# CX21986 Hi Res USB Type-C Compliant Crystal-less Audio CODEC Datasheet

*505-000826-01 Rev A*

***Synaptics Confidential Disclosed Under NDA***

# Revision History

| Document No. | Release Date | Change Description |
|---|---|---|
| 505-000826-01 | 02/01/18 | **Update:**<br>• Convert document to Synaptics template<br>• USB voltage values<br>• Table 104, “DC Supply Voltages,” on page 58<br>• Table 110, “Hardware Signal Definitions,” on page 64 |
| 053-21986R00 | 10/27/17 | Initial release |

# Table of Contents

# List of Figures

# List of Tables

# Introduction

The Synaptics® CX21986 is a highly-integrated, low-cost, crystal-free, USB audio CODEC solution that is optimized for USB Type-C headset applications.

The device features 24-bit stereo playback and record paths for music and voice communication applications. Sampling rates of up to 48 kHz are supported on the record path and up to 192 kHz at 16-bit or 96 kHz at 24-bit on the playback path. The CX21986 includes a digital five-band parametric Equalizer (EQ) for playback, a two-band EQ for recording, an adjustable microphone boost in 3 dB steps, and a stereo headphone driver.

The device uses a built-in, four-conductor headset jack that supports headphone/headset auto-detection and auto switching between CTIA and OMTP-style headsets without any external components.

The stereo playback path uses a capless headphone driver that eliminates the external BOM cost and PCB space of AC coupling capacitors, and supports 8, 16, 24, 32, 44.1, 48, 96, and 192 kHz sampling rates.

The CX21986 is compliant with industry specifications, including USB 2.0 Full Speed (FS), Windows HCK 2.1, as well as enterprise requirements for Skype for Business. The reference design requires a minimal PCB area and is ideal for USB headset and docking station designs.

# Feature Summary

- 1.3 mA USB suspend current
- Hi-Res playback on USB, supporting a 96 Kbps sample rate with a 24-bit sample size
- Additional support for 192 Ksps USB playback sample rate with a 16- bit sample size
- Stereo record path for single-ended microphone input for 8, 16, 24, 32, 44.1, and 48 kHz sampling rates
- In-line four-button support on the microphone path that meets the *Android Wired Audio Headset Specification*
- DRC on the playback path
- Microphone boost settings: 0 dB–36 dB in 3 dB steps.
- High performance stereo audio ADC and DAC:
  - ADC: 97 dB
  - DAC: 103 dB
- Stereo capless headphone driver support for 16/32/10 KΩ loads. 87 dB THD driving to 32/10 KΩ loads
- Crystal free on-chip clock generation saves BOM cost
- Proprietary Synaptics Hum Noise Cancellation Technology for external speakers
- Digital Signal Processing (DSP):
  - DRC on playback path
  - Five-band digital equalizer for playback/recording
  - Microphone Automatic Gain Control (AGC)
- Analog pass through and side tone with volume control for line and MIC inputs.
- Supports wide range input supply from 3.0V to 5.5V
- Power management:
  - Internal Charge Pump (CP) for headphone and digital core logic.

  - 1.8V analog supply generator
  - 3.3V supply for digital I/Os and external devices to support 90 mA drive capability.
- Digital MIC support
- Integrated 8Kbyte OTP memory
- Wake from either button push or headset plug-in
- 43-pin, 3.020 mm x 3.308 mm WLCSP package.

# Functional Description

The CX21986 minimizes the external BOM cost by removing the need for an external crystal, and also integrates a capless headphone driver that produces a full-range frequency response and eliminates AC coupling capacitors. The device integrates headset support with auto-detect and auto-switch between CTIA and OMTP-style headsets to eliminate all external BOM. The CX21986 can be powered by either 3.3V or 5V. The integrated LDO regulators provide clean micbias sources that ensure maximum performance. All internal analog circuitry uses differential signaling to guard against noise coupling from the GSM and TDMA radio frequencies. Current limiting circuitry that allows compliance with the USB inrush current requirement is also integrated. The CX21986 includes support for a PDM DMIC. A UART control interface is also available. A total of 11 GPIO pins are provided for peripheral connectivity. The CX21986 also includes an eight-button programmable keypad.

# System Block Diagram

Figure 1 shows the CX21986 system block diagram.

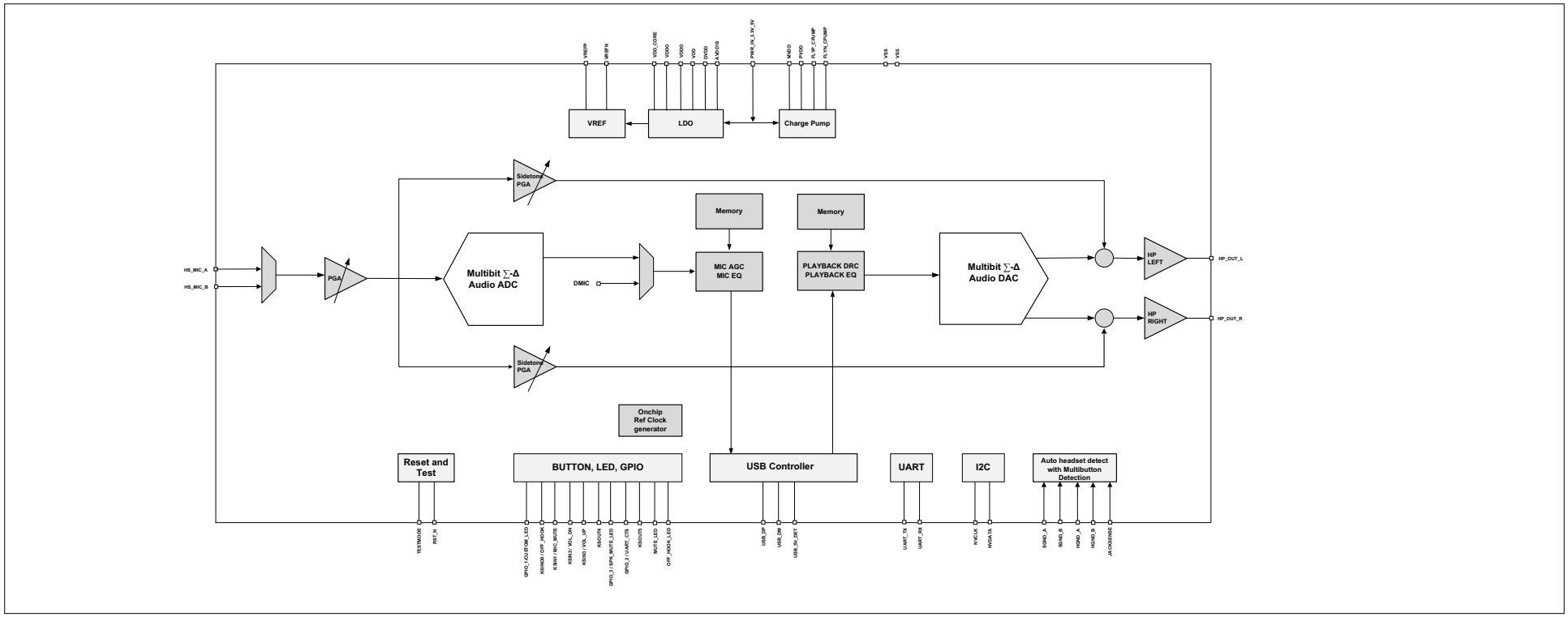



**Figure 1: CX21986 System Block Diagram**

# CX21986 CODEC Features

- USB Type-C compliant stereo audio CODEC
- USB 2.0 full-speed compliant
- Crystal-free solution
- Available in a small WLCSP 3.020 mm x 3.308 mm package
- 1.3 mA USB suspend current
- Stereo record path supports a single-ended microphone input that supports 8kHz, 16kHz, 24kHz, 32kHz, 44.1kHz, and 48kHz sampling rates
- Stereo playback path with a capless headphone driver that eliminates the external BOM cost and PCB space of AC coupling capacitors, and supports 8kHz, 16kHz, 24kHz, 32kHz, 44.1kHz, 48kHz, 96kHz, and 192kHz sampling rates
- Built-in, four-conductor headset jack supports headphone/headset auto-detection, as well as auto switching between CTIA and OMTP-style headsets without any external components
- In-line four-button support on the microphone path that meets the *Android Wired Audio Headset Specification*
- DRC on the playback path
- Five-band digital EQ for playback and recording
- Microphone boost settings of 0 dB to 36 dB, in 3 dB steps.
- Internal 3.3V LDO can supply 90 mA for external circuitry.

# Functional Applications

- USB Type-C Headset
- USB Type-C Docking Station
- USB Gaming Headset

# Device Description

In this section, details of the following items are discussed:

- CODEC
- Each audio interface
- Each control interface
- Each processing block
- Buttons
- LEDs
- GPIOs
- Detection circuits

The available processing in the CX21986 consists of the following:

- Two-band record EQ
- Five-band playback EQ
- AGC on the record path
- DRC on the playback path

The CX21986 has 11 GPIOs that can be used for custom button and LED behavior or other customer defined functions. The CX21986 integrates circuitry that can be used to detect:

- When an external jack has been plugged in
- When the output load has been plugged into the headphone port
- The type of headset used (standard vs. OMTP)

The CX21986 also supports multi-button detection on the microphone signal, as defined by the *Android Wired Audio Headset Specification*.

# USB Digital Audio

The CX21986 is a USB 2.0, full-speed compliant device. Volume settings, terminal types, and some additional customization are available, as detailed in the following sections.

## USB Configuration Registers

The host options in this section are meant to be compiled time options and are set with a firmware patch residing in an external EEPROM. The options listed in this section should not be changed during run time. These RAM register locations are meant for debugging and informational purposes only.

*Table 1: Host Options 0x004D*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:5 | Host [7:4] | Reserved. | 0x00 |
| 4 | Host [4] | Enables mono/stereo HP detection. | 0x0 |
| 3 | Host [3] | 1 = Use the SENSE pin to detect four buttons. The buttons are mapped to same as KSOUT5 (see 0x65–0x68), and KSOUT5 can be freed to another function<br>• 0 = Sense pins are used to detect normal jack sense | 0x0 |
| 2 | Host [2] | 1 = Use HS microphone<br>• 0 = Use standard stereo microphones | 0x0 |
| 1 | Host [1] | 1 = USB mono microphone<br>• 0 = USB stereo microphones | 0x0 |
| 0 | Host[0] | Keypad scanning disable:<br>• 0 = Scan full keypad<br>• 1 = Scan only KSOUT5 and KSOUT4 | 0x0 |

*Table 2: **Host1 Location 0x004F***

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7 | Host1 [7] | HID has telephony:<br>• 1 = HID descriptor has a telephony page<br>• 0 = HID descriptor does not have a telephone page | 0 |
| 6 | Host1 [6] | USB remote wake-up:<br>• 0 = Device is not capable of a remote wake-up<br>• 1 = Device is capable of a remote wake-up | 1 |
| 5 | Host1 [5] | USB-powered:<br>• 1 = USB bus-powered—USB enumerates the requesting power base on USBPower[7:0]<br>• 0 = USB self-powered—USB enumerates the requesting 0ma from the bus | 1 |
| 4 | Host1 [4] | USB line-in—if Host1[4]=1. When this bit is set, the USB enumerates with a line input terminal, volume feature unit, and mixer that allows the line input to be routed to the speaker. | 0 |
| 3 | Host1 [3] | USB TX only—if Host1[3]=1. When this bit is set, the USB enumerates with only the TX path exposed, which makes the USB look like a speaker only. | 0 |
| 2 | Host1 [2] | USB RX only—if Host1[2]=1. When this bit is set, the USB enumerates with only the RX path exposed, which makes the USB look like a microphone only. | 0 |
| 1 | Host1 [1] | USB local volume—if Host1[0] = 0:<br>• 1 = Button volume control is done locally and no HID information is transmitted<br>• 0 = Button volume information is set to the USB using the HID if enumerated | 0 |
| 0 | Host1 [0] | Use the volume infinity knob:<br>• 1 = Use the Volume knob<br>• 0 = Use the Volume buttons | 0 |

## USB Volume Defaults

The following defaults control how the USB volume control blocks enumerate. The blocks control the minimum, maximum, and default volume settings exposed on the USB. All the following values are in Q8.8 format with a range for 5 dB to –74 dB.

*Table 3: Speaker Volume Minimum Low 0x0060*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | SpkVolumeMinL [7:0] | Speaker volume minimum—0x0500 to 0xB600 default 0xCE00 (–50dB). | 0x00 |

*Table 4: Speaker Volume Minimum High 0x0061*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | SpkVolumeMinH [7:0] | Speaker volume minimum—0x0500 to 0xB600 default 0xCE00 (–50dB). | 0xCE |

*Table 5: Speaker Volume Maximum Low 0x0062*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | SpkVolumeMaxL [7:0] | Speaker volume maximum—0x0500 to 0xB600 default 0x0500 (0dB). | 0x00 |

*Table 6: Speaker Volume Maximum High 0x0063*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | SpkVolumeMaxH[7:0] | Speaker volume maximum—0x0500 to 0xB600 default 0x0000 (0dB). | 0x00 |

*Table 7: Speaker Volume Resolution Low 0x0064*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | SpkVolumeResL [7:0] | Speaker volume resolution—0x0000 to 0x7FFF default 0x0100 (1dB). | 0x00 |

*Table 8: Speaker Volume Resolution High 0x0065*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | SpkVolumeResH [7:0] | Speaker volume resolution—0x0000 to 0x7FFF default 0x0100 (1dB). | 0x01 |

*Table 9: Speaker Volume Default Low 0x0066*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | SpkVolumeDefL [7:0] | Speaker volume default—0x0500 to 0xB600 default 0xFB00 (0dB). | 0x00 |

*Table 10: Speaker Volume Default High 0x0067*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | SpkVolumeDefH [7:0] | Speaker volume default—0x0500 to 0xB600 default 0xFB00 (0dB). | 0x00 |

*Table 11: Microphone Volume Minimum Low 0x0070*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | MicVolumeMinL [7:0] | Microphone volume minimum—0x0500 to 0xB600 default 0xE200 (–30dB). | 0x00 |

*Table 12: Microphone Volume Minimum High 0x0071*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | MicVolumeMinH[7:0] | Microphone volume minimum—0x0500 to 0xB600 default 0xE200 (–30dB). | 0xE2 |

*Table 13: Microphone Volume Maximum Low 0x0072*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | MicVolumeMaxL[7:0] | Microphone volume maximum—0x0500 to 0xB600 default 0x0500 (5dB). | 0x00 |

*Table 14: Microphone Volume Minimum High 0x0073*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | MicVolumeMaxH[7:0] | Microphone volume maximum—0x0500 to 0xB600 default 0x0000 (0dB). | 0x00 |

*Table 15: Microphone Volume Resolution Low 0x0074*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | MicVolumeResL[7:0] | Microphone volume resolution—0x0000 to 0x7FFF default 0x0100 (1dB). | 0x00 |

*Table 16: Microphone Volume Resolution High 0x0075*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | MicVolumeResH[7:0] | Microphone volume resolution—0x0000 to 0x7FFF default 0x0100 (1dB). | 0x01 |

*Table 17: Microphone Volume Default Low 0x0076*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | MicVolumeDefL [7:0] | Microphone volume default—0x0500 to 0xB600 default 0x0000 (0dB). | 0x00 |

*Table 18: Microphone Volume Default High 0x0077*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | MicVolumeDefH [7:0] | Microphone volume default—0x0500 to 0xB600 default 0x0000 (0dB). | 0x00 |

*Table 19: Sidetone Volume Minimum Low 0x0078*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | SideToneMinL [7:0] | Sidetone volume minimum—0x0000 to 0xC700 default 0xC700 (–57dB). | 0x00 |

*Table 20: Sidetone Volume Minimum High 0x0079*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | SideToneMinH[7:0] | Sidetone volume minimum—0x0000 to 0xC700 default 0xC700 (–57dB). | 0xC7 |

*Table 21: Sidetone Volume Maximum Low 0x007A*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | SideToneMaxL[7:0] | Sidetone volume maximum—0x0000 to 0xC700 default 0x0000 (0dB). | 0x00 |

*Table 22: Sidetone Volume Minimum High 0x007B*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | SideToneMaxH[7:0] | Sidetone volume maximum—0x0000 to 0xC700 default 0x0000 (0dB). | 0x00 |

*Table 23: Sidetone Volume Resolution Low 0x007C*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | SideToneResL[7:0] | Sidetone volume resolution—0x0000 to 0x7FFF default 0x0180 (1.5dB). | 0x80 |

*Table 24: Sidetone Volume Resolution High 0x007D*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | SideToneResH[7:0] | Sidetone volume resolution—0x0000 to 0x7FFF default 0x0080 (1.5dB). | 0x01 |

*Table 25: Sidetone Volume Default Low 0x007E*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | SideToneDefL [7:0] | Sidetone volume default—0x0000 to 0xC700 default 0xE200 (–30dB). | 0x00 |

*Table 26: Sidetone Volume Default High 0x007F*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | SideToneDefH [7:0] | Sidetone volume default—0x0000 to 0xC700 default 0xE200 (–30dB). | 0xE2 |

## USB Bus Power Requirement Register

The USB bus power requirement is enumerated to the host when the device is bus powered Host1 [5]=1. The power is specified in mA/2 per the USB specification.

*Table 27: USB Bus Power 0x0088*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | USBPower [7:0] | USB bus power requirement is specified in mA/2—0x01 to 0xFF default 0x32 (100mA). | 0x32 |

## USB Terminal Types Registers

The CX21986 allows for customization of the terminal types of the playback and record endpoints, which can be defined as headset, headphone, communications speaker, etc. The options in this section allow the designer to modify the USB terminal type. For a complete list of terminal types, refer to the *USB Device Class Definition for Terminal Types Release 1.0 Guide*.

When defining headset terminal types, it may be necessary to associate the microphone and speaker terminal IDs:

- Microphone terminal ID = ID 1
- Speaker terminal ID = ID 6
- Chat speaker = ID 8

*Table 28: Speaker Terminal Type Low 0x0089*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | SpkTypeL [7:0] | Speaker terminal type—default 0x301 speaker. Common types:<br>• 0x301 = Speaker<br>• 0x302= Headphone<br>• 0x306= Communication speaker<br>• 0x401 = Handset<br>• 0x402= Headset | 0x01 |

*Table 29: Speaker Terminal Type High 0x008A*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | SpkTypeH [7:0] | Speaker terminal type—default 0x301 speaker. Common types:<br>• 0x301 = Speaker<br>• 0x302= Headphone<br>• 0x306= Communication speaker<br>• 0x401 = Handset<br>• 0x402= Headset | 0x03 |

*Table 30: Speaker Association 0x008B*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | SpkTypeH [7:0] | Speaker association:<br>• 0x00 = None<br>• 0x01 = Microphone | 0x00 |

*Table 31: Reserved 0x008C–008E*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| Reserved | - | Reserved. | - |

*Table 32: Microphone Terminal Type Low 0x008F*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | MicTypeL [7:0] | Microphone terminal type—default 0x201 microphone. Common types:<br>• 0x201 = Microphone<br>• 0x206 = Processing microphone array<br>• 0x401 = Handset<br>• 0x402 = Headset | 0x01 |

*Table 33: Microphone Terminal Type High 0x0090*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | MicTypeH [7:0] | Microphone terminal type—default 0x201 microphone. Common types:<br>• 0x201 = Microphone<br>• 0x206 = Processing microphone array<br>• 0x401 = Handset<br>• 0x402 = Headset | 0x02 |

*Table 34: Microphone Association 0x0091*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | MicTypeH [7:0] | Microphone association:<br>• 0x00 = None<br>• 0x06 = Speaker | 0x00 |

### *Volume Balance Control Registers*

There are options to choose if the volume control is a single control applied to both left and right, or is an individual left right control for balance control.

*Table 35: Volume Balance Control 0x0092*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:3 | Reserved | Reserved | 0x0 |
| 2 | MicMasterVol[3] | Microphone balance control:<br>• 0 = Balance Control<br>• 1 = Master Volume | 0x1 |
| 1 | Reserved | Reserved | 0x0 |
| 0 | SpkMasterVol [0] | Speaker balance control:<br>• 0 = Balance Control<br>• 1 = Master Volume | 0x0 |

# Analog Inputs

## Analog Microphone Input to the ADC

The audio input signal path is designed to support single-ended microphone and line input signals. The following diagram provides suggested configurations for each case.

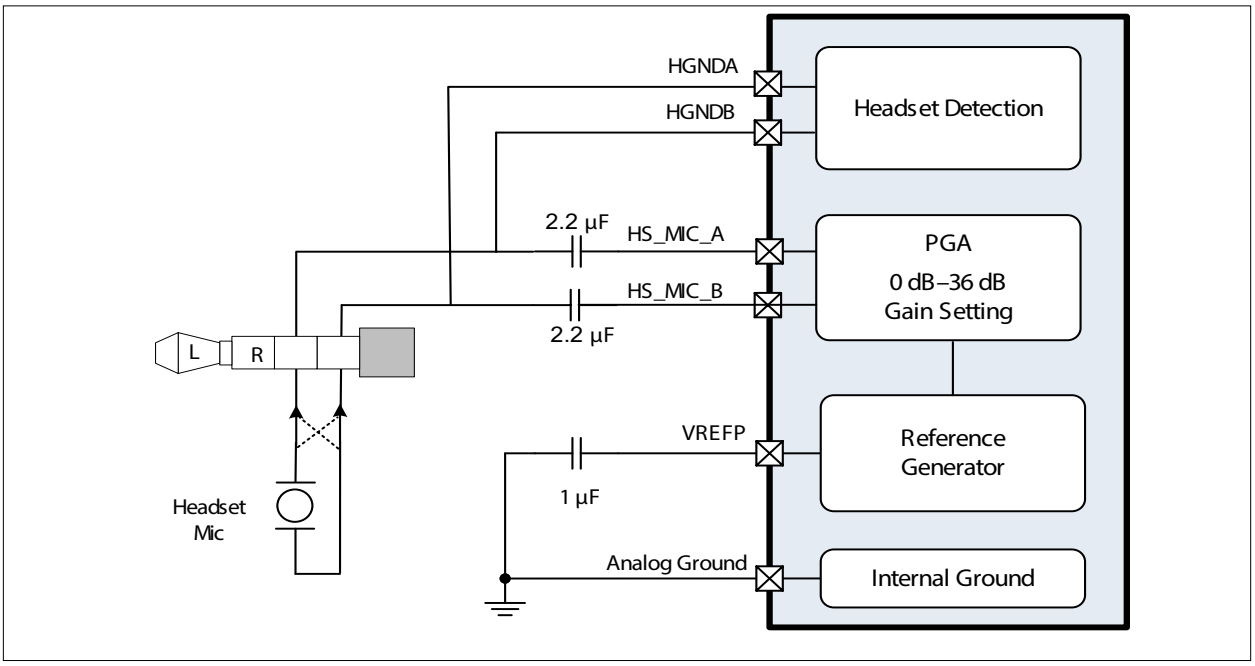



**Figure 2: Headset Micbiasing Configuration**

The Microphone Bias (micbias) circuit provides a bias voltage to a microphone connected to one of the microphones, or both depending on the selection. See Table 111 on page 66 for mic bias options.

### *Gain and Mute Registers for the Analog Microphone and Line Inputs*

This section provides the gain and mute registers for the analog microphone and line inputs.

***Table 36: Microphone Gain Left 0x100E***

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | MICL [7:0] | This gain is in series with the USB line-in volume control. The microphone gain is in 1db steps:<br>• 0x05 = 5dB<br>• 0x04 = 4dB<br>• …<br>• 0xB6 = –74dB | 0x00 |

*Table 37: Microphone Gain Right 0x100F*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | MICR [7:0] | This gain is in series with the USB line-in volume control. The microphone gain is in 1db steps:<br>• 0x05 = 5dB<br>• 0x04 = 4dB<br>• …<br>• 0xB6 = –74dB | 0x00 |

*Table 38: Volume Mutes 0x1012*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:4 | Reserved | Reserved | 0 |
| 3 | Mute [3] | Left microphone mute:<br>• 0 = Unmute<br>• 1 = Mute | 0 |
| 2 | Mute [2] | Right microphone mute:<br>• 0 = Unmute<br>• 1 = Mute | 0 |

# Output Signal Path

## Stereo Capless Headphone Driver

The CX21986 output signal path provides a stereo capless headphone driver that can drive loads as low as 16Ω.. The nominal load is 32Ω and includes the ability to operate safely in open circuits with capacitive loads of up to 400 pF. A line-output mode is also provided where the headphone amplifier can conserve power by driving light loads such as a 10 kΩ line input load.

The digital DAC headphone stereo path driver is a capless design that uses an output signal that is output referenced to ground. There are no AC coupling capacitors required for connecting to headphone loads.

The CX21986 output signal path consists of a 24-bit sigma delta DAC for high performance applications. The output signal path of the CX21986 is also designed for very low pop operation of less than –65 dB (A-Weighted).

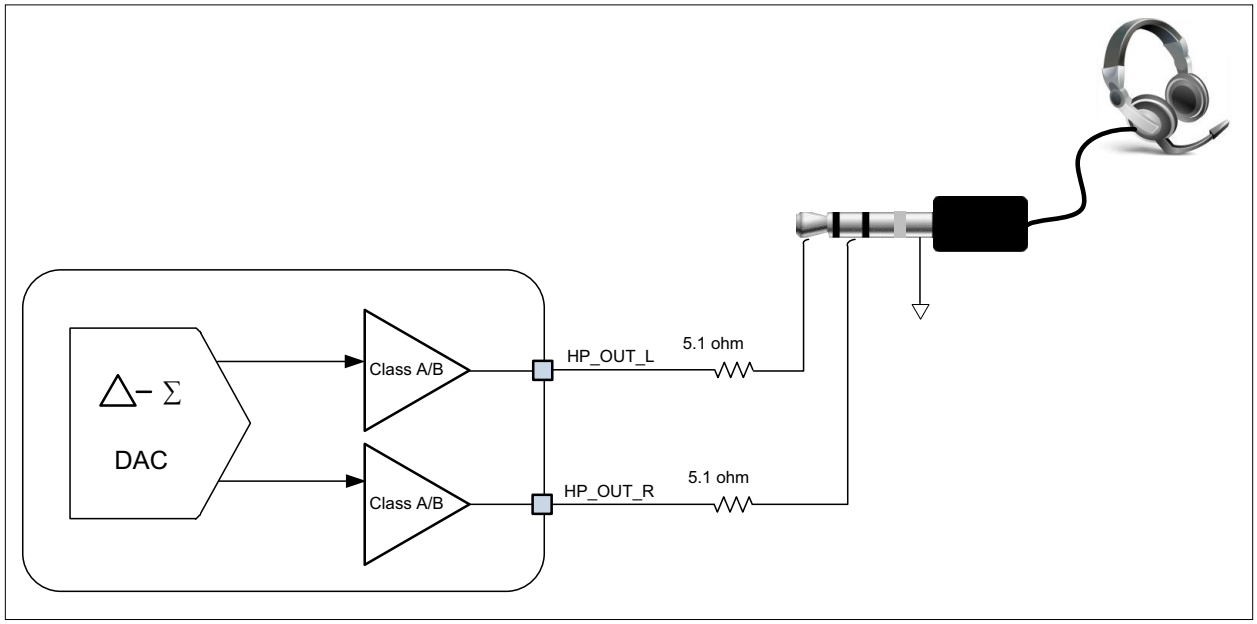



**Figure 3: Output Signal Path Diagram**

## Output Signal Path Volume Control Registers

*Table 39: DAC1 Left 0x100A*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | DACL [7:0] | This gain is in series with the USB volume control. The DAC gain is in 1db steps:<br>• 0x05 = 5dB<br>• 0x04 = 4dB<br>• …<br>• 0xB6 = –74db | 0x05 |

*Table 40: DAC2 Right 0x100B*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7:0 | DACR [7:0] | This gain is in series with the USB volume control. The DAC gain is in 1db steps:<br>• 0x05 = 5dB<br>• 0x04 = 4dB<br>• …<br>• 0xB6 = –74db | 0x05 |

*Table 41: Volume Mutes 0x1012*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 1 | Mute [1] | Left speaker mute:<br>• 0 = Unmute<br>• 1 = Mute | 0 |
| 0 | Mute [0] | Right speaker mute:<br>• 0 = Unmute<br>• 1 = Mute | 0 |

# Digital Core

## ADC DSP

A Digital Signal Processing (DSP) system, with an analog audio input signal in the range 0 kHz to –20 kHz, uses downsampling techniques to convert the analog single bit stream at a high rate to a digital code at a low rate.

A multistage decimator converts the output of a 3rd order delta-sigma from a single-bit stream at 12.288 MHz into 17 bit stream at (44.1 kHz/48 kHz/96 kHz/8 kHz/16 kHz/24 kHz/22.05 kHz/32 kHz).

The ADC datapath consists of:

- 3 Half Bands – Decimation
- Polyphase SRC
- Programmable gain
- Sync filter

## DAC DSP

A DSP system, with digital audio input signal in the range of 0 kHz to –20 kHz, uses oversampling techniques and a 3rd order delta-sigma modulator to convert the 16/24 bit digital signal at sampling frequencies 44.1 kHz/48 kHz/96 kHz/8 kHz/16 kHz/24 kHz/22.05 kHz/32 kHz into a multi-bit stream at a rate of 12.288 MHz.

The DAC datapath consists of:

- Equalizer and crossover
- Dynamic range compression logic
- 3 Half Bands – Interpolation
- Polyphase SRC
- Programmable gain
- Sigma delta modulation and scrambler

## Digital Signal Processing Registers

Register 0x118C shows the processing options available in the CX21986.

*Table 42: DSP Processing Enable Register 0x118C*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 7 | DSPEn1[7] | Reserved. | 0 |
| 6 | DSPEn1 [6] | Reserved. | 0 |
| 5 | DSPEn1 [5] | Side tone enable. | 1 |
| 4 | DSPEn1 [4] | Microphone AGC enable. The AGC only works if the sample rate is 8kHz or 16kHz. | 0 |
| 3 | DSPEn1 [3] | Playback DRC enable. | 0 |
| 2 | DSPEn1 [2] | Record path two-band EQ enable. | 0 |
| 1 | DSPEn1 [1] | Playback path five-band EQ enable. | 0 |

*Table 42: DSP Processing Enable Register 0x118C*

| Bit | Register Field Name | Register Description | Reset |
|---|---|---|---|
| 0 | DSPEn1 [0] | NEWC. The act/process changes to the host API. This bit clears when the update is done. When this bit is set, the host should read this bit to verify that it has cleared before modifying any other API registers. | 0 |

## Input EQ

The record path EQ consists of two bands, and there is a profile for each supported record frequency. The EQ RXEQEna [2] = 1.

The implementation consists of two second-order IIRs in series in each channel. Each second-order IIR has the three $b_0$, $b_1$, and $b_2$ forward coefficients that operate on the input, and the two $a_1$ and $a_2$ feedback coefficients that operate on the output and a scaling factor G:

$H(z) = (G \cdot b_0 + G \cdot b_1 \cdot z^{-1} + G \cdot b_2 \cdot z^{-2}) / (1 - G \cdot a_1 \cdot z^{-1} - G \cdot a_2 \cdot z^{-2})$.

$y_n = G \cdot (a_1 \cdot y_{n-1} + a_2 \cdot y_{n-2} + b_0 \cdot x_n + b_1 \cdot x_{n-1} + b_2 \cdot x_{n-2})$

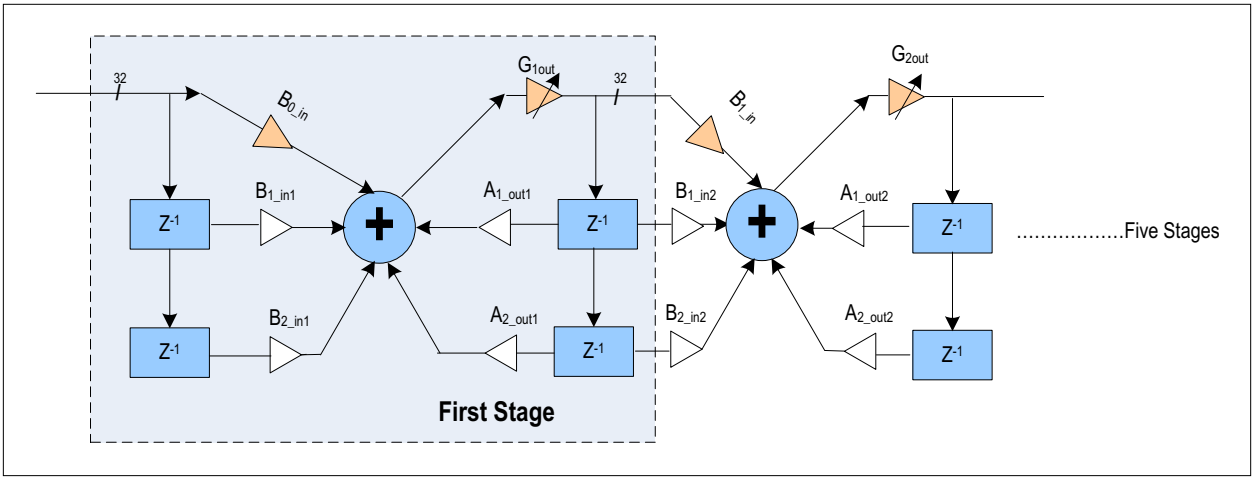



**Figure 4: EQ Block Diagram**

***Record Path Equalizer (EQ) 8K***

- EQ Band1 8K coefficient from address 0x112D–0x1137
- EQ Band2 8K coefficient from address 0x1138–0x1142

***Record Path EQ 16K***

- EQ Band1 16K coefficient from address 0x1143–0x114D
- EQ Band2 16K coefficient from address 0x114E–0x1158

***Record Path EQ 32K***

- EQ Band1 32K coefficient from address 0x1159–0x1163
- EQ Band2 32K coefficient from address 0x1164–0x116E

***Record Path EQ 44.1K***

- EQ Band1 44.1K coefficient from address 0x116F–0x1179
- EQ Band2 44.1K coefficient from address 0x117A–0x1184

***Record Path EQ 48K***

- EQ Band1 48K coefficient from address 0x1185–0x118F
- EQ Band2 48K coefficient from address 0x1190–0x119A

## Microphone AGC and Activity Detector

The microphone AGC function can be broken down in to three sections:

- Activity detector
- Gain control
- Dynamic Range Compression (DRC)

Because the microphone AGC is intended for voice conversation and both channels have similar energy characteristics, the gain is only calculated on one channel and applied to both channels. The microphone AGC block supports both 16kHz and 8kHz samples (16-bit). The AGC curve is shown in this section.

### *Microphone AGC API Registers*

*Table 43: AD_DC_REJECT_POLE—0x11A7*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | AD_DC_REJECT_POLE | 8'hC0 | R/W | POLE location for the datIn DC rejection IIR filter.<br>Z = AD_DC_REJECT_POLE/2^15<br>The AD_DC_REJECT_POLE is 16 bits:<br>• Bit 15 = Fixed at 0 for a positive only value<br>• Bits 14–7 = Programmed through this register<br>• Bits 6–0 = Fixed at 0 |

*Table 44: AD_BETA_VOICE SHFT—0x11A8*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:4 | Reserved | 4'h0 | R | Reserved. |
| 3:0 | AD_BETA_VOICE_SHFT | 4'h2 | R/W | PowerIn fast averaging IIR filter beta coefficient.<br>PowerIn = PowerIn * (1–Beta/2^15) + FrameEnergy *(Beta/2^15)<br>The range is 0–15 for a beta range of 2^(15–0) to 2^(15–15). The default beta is 2^(15–2) or 2^13. |

*Table 45: AD_BETA_NOISE_SHFT—0x11A9*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:4 | Reserved | 4'h0 | R | Reserved. |
| 3:0 | AD_BETA_NOISE_SHFT | 4'hB | R/W | PowerNoiseAvg slow averaging IIR filter beta coefficient.<br>PowerNoise = PowerNoise * (1–Beta/2^15) + FrameEnergy *(Beta/2^15)<br>The range is 0–15 for a beta range of 2^(15–0) to 2^(15–15). The default beta is 2^(15–11) or 2^4. |

*Table 46: AD_TEST_GAIN—0x11AA*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:4 | Reserved | 4'h0 | R | Reserved. |
| 3:0 | AD_TEST_GAIN | 4'h0 | R/W | Test only register. Left shift of the datIn input signal to the AGC module by the number of specified bits. |

*Table 47: AD_RELEASE_DELAY MSB—0x11AB*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | AD_RELEASE_DELAY | 8'h01 | R/W | In units of ms and used to program the period of time passed before allowing the noise power to increase. |

*Table 48: AD_RELEASE_DELAY LSB—0x11AC*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | AD_RELEASE_DELAY | 8'h2C | R/W | In units of ms and used to program the period of time passed before allowing the noise power to increase. |

*Table 49: AD_POWERIN_INITVAL MSB—0x11AD*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | AD_POWERIN_INITVAL | 8'h0A | R/W | PowerIn—Voice power register is 32 bits. This register programs the reset value for bits 23 to 30. Bit 31 is fixed at 0. |

*Table 50: AD_POWERIN_INITVAL LSB—0x11AE*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | AD_POWERIN_INITVAL | 8'h3D | R/W | PowerIn—Voice power register is 32 bits. This register programs the reset value for bits 15 to 22. Bits 0 to 14 are fixed at 0. |

*Table 51: AD_MIN_SNR MSB—0x11AF*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | AD_MIN_SNR | 8'hE9 | R/W | AD_MIN_SNR[15:8]: Minimum SNR in dB. |

*Table 52: AD_MIN_SNR LSB—0x11B0*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | AD_MIN_SNR | 8'h00 | R/W | AD_MIN_SNR[7:0]: Minimum SNR in dB. |

*Table 53: AGC_VOLUME_RAMP_STEP—0x11B1*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | AD_POWERNOISE_MULT | 8'h8 | R/W | AGC volume ramp step size in dB. The default is 8 = round(0.03 * 2^8). |

*Table 54: AD_POWERNOISE_INITVAL MSB—0x11B2*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | AD_POWERNOISE_INITVAL | 8'hE9 | R/W | Power noise—Noise power register is 16 bits. This register programs the reset value for bits 8 to 15. |

*Table 55: AD_POWERNOISE_INITVAL LSB—0x11B3*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | AD_POWERNOISE_INITVAL | 8’h00 | R/W | Power noise—Noise power register is 16 bits. This register programs the reset value for bits 0 to 7. |

*Table 56: AD_MIN_THRESH MSB—0x11B4*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | AD_MIN_THRESH | 8'hC1 | R/W | AD_MIN_THRESH [15:8] offset for the decision threshold to prevent false detection of voice when the signal is low. This value is 16-bit unsigned. |

*Table 57: AD_MIN_THRESH LSB—0x11B5*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | AD_MIN_THRESH | 8'h00 | R/W | AD_MIN_THRESH [7:0] offset for the decision threshold to prevent false detection of voice when the signal is low. This value is 16-bit unsigned. |

*Table 58: AD_ON2OFF_DELAY_BLOCKS—0x11B6*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:5 | Reserved | 3’h0 | R | Reserved. |
| 4:0 | AD_ON2OFF_DELAY_BLOCKS | 5’h14 | R/W | On to off delay in number of blocks. |

*Table 59: AGC_UPDATE_BKS—0x11B7*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:5 | Reserved | 3’h0 | R | Reserved. |
| 4:0 | AGC_UPDATE_BKS | 5’ha | R/W | Number of blocks of sample used to calculate a new gain. |

*Table 60: AGC_ENERGY_THRESH_LO MSB—0x11B8*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | AGC_ENERGY_THRESH_LO | 8'hE8 | R/W | Threshold for a decision if the gain needs to be increased. Only the upper 8 bits of the etmp value is compared. The default value is –24*2^8. |

*Table 61: AGC_ENERGY_THRESH_HI MSB—0x11B9*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | AGC_ENERGY_THRESH_HI | 8’hF7 | R/W | Threshold for a decision if the gain needs to be decreased. Only the upper 8 bits of the etmp value is compared. The default value is –9 * 2^8. |

*Table 62: AGC_STABILITY_BLOCKS—0x11BA*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7 | Reserved | 1'h0 | R | Reserved. |
| 6:0 | AGC_STABILITY_BLOCKS | 7'd60 | R/W | Value compared to the stability counter to ensure that the gain does not oscillate (increase/decrease) too rapidly. |

*Table 63: AGC_STEP_UP—0x11BB*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | AGC_STEP_UP | 8'd18 | R/W | The amount that the gain is to be increased by. |

*Table 64: AGC_STEP_DOW MSB—0x11BC*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | AGC_STEP_DOWN | 8'hEE | R/W | The amount that the gain is to be decreased by. |

*Table 65: AGC_STAB_CNT_POS_INC—0x11BD*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:4 | Reserved | 4'h0 | R | Reserved. |
| 3:0 | AGC_STAB_CNT_POS_INC | 4'h1 | R/W | The amount that the stability counter is to be increased by. The range is 0 to 15. |

*Table 66: AGC_STAB_CNT_NEG_INC—0x11BE*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 3:0 | AGC_STAB_CNT_NEG_INC | 4'hC | R/W | The amount that the stability counter is to be decreased by. The range is 0 to –16. |

*Table 67: DRC_RELEASE_DELAY MSB 0x11BF*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | DRC_RELEASE_DELAY | 8'h40 | R/W | In units of ms and used to program the period of time passed before MaxAbsIn can be decreased to a lower value if the input level is not increasing. The hysteresis for adjusting the gain upward. |

*Table 68: DRC_GAIN_STEP_SLOW 0x11C0*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | DRC_GAIN_STEP_SLOW | 8'h10 | R/W | Coefficient for the MaxAbsIn IIR filters. The DRC_GAIN_STEP_SLOW coefficient is 10-bit unsigned. This register programs bits 2 to 9, while bits 0 to 1 are fixed at 0. |

*Table 69: DRC_GAIN_STEP_FAST 0x11C1*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | DRC_GAIN_STEP_FAST | 8'h40 | R/W | Coefficient for the target gain filters. The DRC_GAIN_STEP_FAST coefficient is 15-bit unsigned. This register programs bits 7 to 14, while bits 0 to 6 are fixed at 0. |

*Table 70: DRC_MAX_LIN_OUT 0x11C2*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | DRC_MAX_LIN_OUT | 8'hE0 | R/W | Max linear output of the DRC. The DRC_MAX_LIN_OUT is 16-bit signed. This register programs bits 4 to 11. Bits 12 to 15 are fixed at 4'hF, and bits 0 to 3 are fixed at 4'h0. |

*Table 71: DRC_GAIN_SHIFT 0x11C3*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 2:0 | DRC_GAIN_SHIFT | 3'h5 | R/W | Gain is in Q6.10/Q6.26 format, 5-bit scale down from Q1.15/Q1.31. |

*Table 72: DRC_MAXABS_INITVAL 0x11C4*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | DRC_MAXABS_INITVAL | 8'h3A | R/W | • Initial value for MaxAbsIn is: MaxAbsIn = DRC_MAXABS_INITVAL * 2^24;<br>• Initial value for MaxAbsPre is: MaxAbsPre = DRC_MAXABS_INITVAL * 2^8;<br>The range is –32 to 31. |

*Table 73: AGC_GAIN_INITVAL 0x11C5*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | AGC_GAIN_INITVAL | 8'h00 | R/W | Initial value for the AGC gainctrl gain output to micdrc. |

*Table 74: DRC_OUTPUT_LIMIT 0x11C6*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | DRC_OUTPUT_LIMIT | 8'h00 | R/W | DRC output limit. This register programs the 8 MSB bit of the 16-bit signed value. |

*Table 75: AGC_CLIPPING_THRESH 0x11C7*

| Bits | Name | Default | R/W | Description |
|---|---|---|---|---|
| 7:0 | AGC_CLIPPING_THRESH | 8'hF9 | R/W | AGC clipping threshold. The default value is –7 * 2^8. |

## Output EQ

The implementation consists of five second-order IIRs in series in each channel. Each second-order IIR has the three $b_0$, $b_1$ and $b_2$ forward coefficients operating on the input, and the two $a_1$ and $a_2$ feedback coefficients operating on the output and a scaling factor G:

$H(z) = (G \cdot b_0 + G \cdot b_1 \cdot z^{-1} + G \cdot b_2 \cdot z^{-2}) / (1 - G \cdot a_1 \cdot z^{-1} - G \cdot a_2 \cdot z^{-2})$.

$y_n = G \cdot (a_1 \cdot y_{n-1} + a_2 \cdot y_{n-2} + b_0 \cdot x_n + b_1 \cdot x_{n-1} + b_2 \cdot x_{n-2})$

Because the coefficients are programmable, there is full flexibility of the filter type, band-frequency, Q factor, and gain. The only limitation is that the band's frequency usually cannot be below 60Hz (at a 48kHz sampling rate), which results from the fixed point resolution of 16 bits per coefficient.

Because the IIR implements a graphic EQ and the values are changed on a user command, the update is slow.

The graphic EQ tuning of each band's parameters (filter-type, frequency, Q factor, and gain) is done through a software tool, which internally calculates the fixed point coefficients and scaling that correspond with each set of parameters, and sends the coefficients/scaling to the device.

For more details, see the EQ block diagram in Figure 4 on page 29.

### *Playback EQ 8K API Registers*

This section lists the playback EQ API registers.

***Note:*** Each set of registers applies to a specific sample rate.

*Table 76: EQ Band1 8KCoefficient B0 Low 0x1018*

| Bits | Name | Description | Reset |
|---|---|---|---|
| 7:0 | TXBand18KB0L | CODEC IIR B0 feed forward coefficient low byte. | 0x00 |

*Table 77: EQ Band1 8K Coefficient B0 High 0x1019*

| Bits | Name | Description | Reset |
|---|---|---|---|
| 7:0 | TXBand18KB0H | CODEC IIR B0 feed forward coefficient high byte. | 0x40 |

*Table 78: EQ Band1 8K Coefficient B1 Low 0x101A*

| Bits | Name | Description | Reset |
|---|---|---|---|
| 7:0 | TXBand18KB1L | CODEC IIR B1 feed forward coefficient low byte. | 0x00 |

*Table 79: EQ Band1 8K Coefficient B1 High 0x101B*

| Bits | Name | Description | Reset |
|---|---|---|---|
| 7:0 | TXBand18KB1H | CODEC IIR B1 feed forward coefficient high byte. | 0x00 |

*Table 80: EQ Band1 8K Coefficient B2 Low 0x101C*

| Bits | Name | Description | Reset |
|---|---|---|---|
| 7:0 | TXBand18KB2L | CODEC IIR B2 feed forward coefficient low byte. | 0x00 |

*Table 81: EQ Band1 8K Coefficient B2 High 0x101D*

| Bits | Name | Description | Reset |
|---|---|---|---|
| 7:0 | TXBand18KB2H | CODEC IIR B2 feed forward coefficient high byte. | 0x00 |

*Table 82: EQ Band1 8K Coefficient A0 Low 0x101E*

| Bits | Name | Description | Reset |
|---|---|---|---|
| 7:0 | TXBand18KA0L | CODEC IIR A0 feed forward coefficient low byte. | 0x00 |

*Table 83: EQ Band1 8K Coefficient A0 High 0x101F*

| Bits | Name | Description | Reset |
|---|---|---|---|
| 7:0 | TXBand18KA0H | CODEC IIR A0 feed forward coefficient high byte. | 0x00 |

*Table 84: EQ Band1 8K Coefficient A1 Low 0x1020*

| Bits | Name | Description | Reset |
|---|---|---|---|
| 7:0 | TXBand18KA1L | CODEC IIR A1 feed forward coefficient low byte. | 0x00 |

*Table 85: EQ Band1 8K Coefficient A1 High 0x1021*

| Bits | Name | Description | Reset |
|---|---|---|---|
| 7:0 | TXBand18KA1H | CODEC IIR A1 feed forward coefficient high byte. | 0x00 |

*Table 86: EQ Band1 8K Gain 0x1022*

| Bits | Name | Description | Reset |
|---|---|---|---|
| 7:0 | TXBand18KGain | CODEC IIR coefficient gain:<br>• 0 = –12dB, range [–0.25, 0.25)<br>• 1 = –6dB, range [–0.5, 0.5)<br>• 2 = 0dB, range [–1,1)<br>• 3 = 6dB, range [–2,2)<br>• 4 = 12dB, range [–4,4)<br>• 5 = 18dB, range [–8,8) | 0x00 |

***EQ Band2 8K Coefficient from Address 0x1023–0x102D***

- B0 coefficient 0x4000
- B1 coefficient 0x0000
- B2 coefficient 0x0000
- A0 coefficient 0x0000
- A1 coefficient 0x0000
- Gain 0x03

***EQ Band3 8K Coefficient from Address 0x102E–0x1038***

- B0 coefficient 0x4000
- B1 coefficient 0x0000
- B2 coefficient 0x0000
- A0 coefficient 0x0000
- A1 coefficient 0x 0000
- Gain 0x03

***EQ Band4 8K Coefficient from Address 0x1039–0x1043***

- B0 coefficient 0x4000
- B1 coefficient 0x 0000
- B2 coefficient 0x0000
- A0 coefficient 0x0000
- A1 coefficient 0x 0000
- Gain 0x03

***EQ Band5 8K Coefficient from Address 0x1044–0x104E***

- B0 coefficient 0x4000
- B1 coefficient 0x0000
- B2 coefficient 0x0000
- A0 coefficient 0x0000
- A1 coefficient 0x0000
- Gain 0x03

***EQ 16K***

- EQ Band1 16K coefficient from address 0x104F–0x1059
- EQ Band2 16K coefficient from address 0x105A–0x1064
- EQ Band3 16K coefficient from address 0x1065–0x106F
- EQ Band4 16K coefficient from address 0x1070–0x107A
- EQ Band5 16K coefficient from address 0x107B–0x1085

***EQ 32K***

- EQ Band1 32K coefficient from address 0x1086–0x1090
- EQ Band2 32K coefficient from address 0x1091–0x109B
- EQ Band3 32K coefficient from address 0x109C–0x10A6
- EQ Band4 32K coefficient from address 0x10A7– 0x10B1
- EQ Band5 32K coefficient from address 0x10B2–0x10BC

***EQ 44.1K***

- EQ Band1 44.1K coefficient from address 0x10BD–0x10C7
- EQ Band2 44.1K coefficient from address 0x10C8–0x10D2
- EQ Band3 44.1K coefficient from address 0x10D3–0x10DD
- EQ Band4 44.1K coefficient from address 0x10DE– 0x10E8
- EQ Band5 44.1K coefficient from address 0x10E9–0x10F3

***EQ 48K***

- EQ Band1 48K coefficient from address 0x10F4–0x10FE
- EQ Band2 48K coefficient from address 0x10FF–0x1109
- EQ Band3 48K coefficient from address 0x110A–0x1014
- EQ Band4 48K coefficient from address 0x1115– 0x111F
- EQ Band5 48K coefficient from address 0x1120–0x112C

***EQ Gains***

The EQ gains are the gains applied at the end of the EQ stage whenever the EQ is enabled. The dB gain can be calculated using the following formula:

20 * Log10(X / 0x1000) = dB

Min 0x0000

Max 0x7FFF

Default 0x1000 (0dB)

*Table 87: **EQ Gain Low 0x112B***

| Bits | Name | Description | Reset |
|---|---|---|---|
| 7:0 | EQ_GAINL | Gain of the final output stage of the EQ chain. | 0x00 |

*Table 88: **EQ Gain High 0x112C***

| Bits | Name | Description | Reset |
|---|---|---|---|
| 7:0 | EQ_GAINH | Gain of final output stage of the EQ chain. | 0x10 |

### Playback Path Dynamic Range Compression (DRC)

The following figure shows the DRC curve.

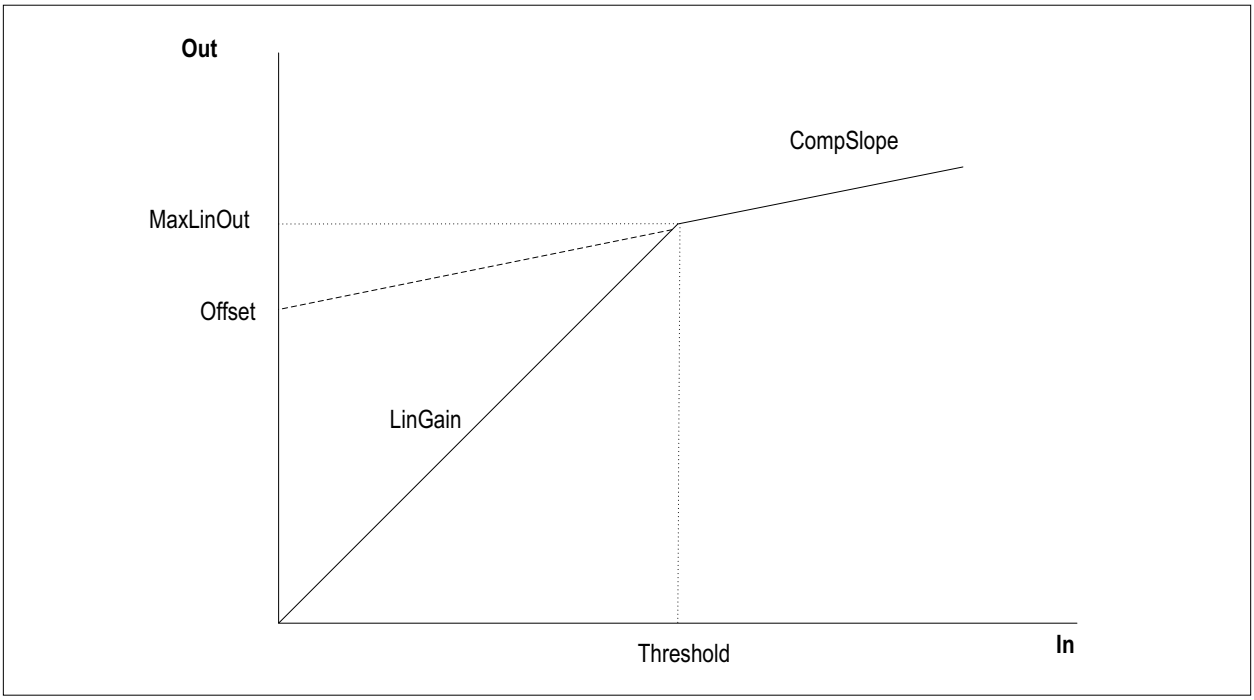



**Figure 5: DRC Curve**

The DRC allows boosting the signal level while preventing clipping distortion. The DRC behaves as a signal-dependent amplifier, with the amplifier's gain determined according to a gain function. This function has a programmable threshold, or "knee." The function provides a linear gain below the threshold, also known as a "boost," and above it there is compression. The compression slope is set to 0, which is referred to as "infinite compression."

The DRC has fast response in case of an "attack", which is defined as any event of a peak in the input signal becoming greater than the latest estimation of the input magnitude. If the signal goes into the high region (above threshold), fast reduction of the gain is needed to prevent audible distortion due to clipping. The rate of response to attack events is controlled by the attack time parameter.

The DRC has a slower response, known as "release", when all peaks in the input signal stay below the latest estimation of the input magnitude during a certain interval. The slower response includes some delay followed by a smooth change of the gain and its purpose is to prevent amplitude modulation (or amplitude jitter). If the signal has been in the high region, the gain is slowly increased during the release. The release rate is controlled by the release time parameter.

The DRC integrates into the volume control as if the volume is applied at the input to the DRC. This integration allows the DRC to not reduce the gain in case of a low volume setting, as in such case there is no risk of clipping.

The following lists the programmable parameters for DRC:

- Attack rate = Allows setting one of eight attack times: 4ms, 8ms, 16ms, 32ms, 64ms, 128ms, 256ms, and 512ms.
- Release rate threshold = In the range –30 to 30 dB. If the input peaks are below this threshold, then the slow release is selected—otherwise the fast release is selected.
- Slow release rate = Allows setting 1 of 16 release times (the same supported values as for the attack rate) plus 1024ms, 2048ms, 4096ms, 8192ms, 16384ms, 32768ms, 65536ms, and 131072ms.
- Slow release delay = Allows setting 1 of 16 delay values before the release starts. All powers of 2 from 1ms to 32768ms (1, 2, 4, 8 … 16384, 32768).
- Fast release rate = Allows setting 1 of 16 release times, same supported values as for slow release rate.
- Fast release delay = Allows setting one of 16 release times (the same supported values as for the slow release delay).
- Boost = In the range 0dB to 30 dB.
- Threshold = In the range –16dB to 0dB, specified for the output peak level. The threshold for the input peak level equals to the threshold for the output peaks minus the boost.
- Max gain = Allows limiting the maximal gain to one of eight values 0dB, 6dB, 12dB, 18dB, 24dB, 30dB, 36dB, and 42dB.

*Table 89: **SPKR DRC CTRL1 0x119B***

| Bits | Name | Default | Description |
|---|---|---|---|
| 7:4 | SPKR_DRC_RELEASE_UPDATE_STEP | 0xA | Used to program the gain step required to release the gain slowly when the signal is in the linear region. |
| 3:1 | SPKR_DRC_ATTACK_UPDATE_STEP | 0x2 | Used to program the gain step required to compress the incoming signal in the compression region. |
| 0 | Reserved | 0 | Reserved. |

*Table 90: **SPKR DRC CTRL2 0x119C***

| Bits | Name | Default | Description |
|---|---|---|---|
| 7:4 | Reserved | 0 | Reserved. |
| 3:0 | SPKR_DRC_RELEASE_DELAY | 0xA | In units of ms and used to program the period of time passed before MaxAbsIn can be decreased to a lower value if the input level is not increasing. The hysteresis for adjusting the gain upward. Internally, the 4 bits are used as RELEASE_DELAY = 2^ SPKR_DRC_RELEASE_DELAY_LOG2. |

*Table 91: **SPKR DRC CTRL3 0x119D***

| Bits | Name | Default | Description |
|---|---|---|---|
| 7:4 | SPKR_DRC_FAST_RELEASE_DELAY | 0x6 | SPKR_DRC_FAST_RELEASE_DELAY_LOG2—Used to program the release delay value. This is used to update the release delay value to make a quick transition in the gain value. Internally to SPKR DRC block the FAST_RELEASE_DELAY = 2^SPKR_DRC_ FAST_RELEASE_DELAY_LOG2. |

*Table 91: SPKR DRC CTRL3 0x119D*

| Bits | Name | Default | Description |
|---|---|---|---|
| 3:0 | SPKR_DRC_FAST_UPDATE_STEP | 0x5 | Used to program the step value. This will be used to quickly update the MAXABS IN value. |

*Table 92: SPKR DRC CTRL4 0x119E*

| Bits | Name | Default | Description |
|---|---|---|---|
| 7:6 | Reserved | 0 | Reserved. |
| 5:0 | SPKR_DRC_BOOST | 0x0 | The gain value initially set for DRC. 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 |

*Table 93: SPKR DRC CTRL5 0x119F*

| Bits | Name | Default | Description |
|---|---|---|---|
| 7:6 | Reserved | 0 | Reserved. |
| 5:0 | SPKR_DRC_ MAX_LIN_OUT | 0x0 | Used to set threshold and limit values. 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 |

*Table 94: SPKR DRC CTRL6 0x11A0*

| Bits | Name | Default | Description |
|---|---|---|---|
| 7:6 | Reserved | 0 | Reserved. |
| 5:0 | SPKR_DRC_BALANCE_RAMP_STEP | 0x4 | Determines the balance RAMP STEP size. 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 |

*Table 95: SPKR DRC CTRL7 0x11A1*

| Bits | Name | Default | Description |
|---|---|---|---|
| 7:6 | Reserved | 0 | Reserved. |
| 5:3 | SPKR_DRC_VOLUME_RAMP_STEP_SHIFT_SELECT | 0x4 | Selects the step and shift size for volume ramping. Internal calculations are:<br>• VOLUME_RAMP_STEP = 2*(2^ SPKR_DRC_VOLUME_RAMP_STEP_ SHIFT_SELECT);<br>• VOLUME_RAMP_MAX_SHIFT = 10 - SPKR_DRC_VOLUME_RAMP_STx4EP_ SHIFT_SELECT; |
| 2:0 | SPKR_DRC_GAIN_SHIFT | 0x4 | Gain shift value. |

*Table 96: SPKR DRC CTRL8 0x11A2*

| Bits | Name | Default | Description |
|---|---|---|---|
| 7:6 | Reserved | 0 | Reserved. |
| 5:0 | SPKR_DRCRELEASE_RATE_TH | 0 | Threshold for deciding whether to use slow release (with RELEASE_UPDATE_STEP) or fast release (with TRANSIENT_UPDATE_STEP). 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 |

*Table 97: DRC Gain Control 0x11A3*

| Bits | Name | Default | Description |
|---|---|---|---|
| 7:1 | R | - | Reserved. |
| 0 | INT_DRC_GAIN_CTRL | 1'b0 | Enables the interpolator gain and DRC gain that come from the register. The gain for:<br>• interpolator comes from 0xF15 and 0xF16<br>• DRC comes from 0xF31 and 0xF32 |

*Table 98: DAC 1 DRC Volume/Gain 0x11A4*

| Bits | Type | Default | Name | Description |
|---|---|---|---|---|
| 7 | R | 0 | R | Reserved. |

*Table 98: DAC 1 DRC Volume/Gain 0x11A4*

| Bits | Type | Default | Name | Description |
|---|---|---|---|---|
| 6:0 | RW | 4A | DAC 1 DRC Volume | Programmable gain stage in digital:<br>• 0x4F = 5dB<br>• 0x4E = 4dB<br>• 0x4D = 3dB<br>• 0x4C = 2dB<br>• …<br>• 0x3 = –71dB<br>• 0x2 = –72dB<br>• 0x1 = –73dB<br>• 0x0 = –74dB |

*Table 99: DAC 2 DRC Volume/Gain 0x11A5*

| Bits | Type | Default | Name | Description |
|---|---|---|---|---|
| 7 | R | 0 | R | Reserved. |
| 6:0 | RW | 4A | DAC 2 DRC Volume | Programmable gain stage in digital:<br>• 0x4F = 5dB<br>• 0x4E = 4dB<br>• 0x4D = 3dB<br>• 0x4C = 2dB<br>• …<br>• 0x3 = –71dB<br>• 0x2 = –72dB<br>• 0x1 = –73dB<br>• 0x0 = –74dB |

*Table 100: SPKR DRC Release Control 0x11A6*

| Bits | Type | Default | Name | Description |
|---|---|---|---|---|
| 7:0 | RW | 0 | SPKR DRC RELEASE CONTROL | Programmable speaker DRC release control. |

### One-Time Programmable Memory

The CX21986 CODEC can be configured and integrated within a design either by using an external EEPROM or the integrated OTP saving PCB area and cost.

The OTP is intended to store basic chip configurations and settings to optimize performance. A typical application of the OTP memory is to store product-specific customization of the standard firmware image.

The CX21986 device provides 8K Byte of One-Time-Programmable (OTP) memory. This memory is accessible by the on-chip CPU, and may be used for code or data storage.

While any location within the OTP memory may be written only once, the 8KB memory is sufficiently large enough to allow for 3 or 4 revisions of a typically sized data/code image. To program the OTP, a voltage of 6.0V must be provided to the CX21986 input power pin.

For use and configuration of OTP memory, please consult the CX21988 OTP Programming Procedure Application Note.

# External Accessory and Button Detection

## Multi-Standard Headset Support

Most headsets use four-terminal 3.5mm plugs to support stereo audio playback and mono microphone. Tip and Ring1 are always the left and right playback signals, but two standards exist for the location of the ground and microphone signals.

One type of headset has the ground signal on Ring2, and the microphone signal on the sleeve, which is referred to as the CTIA standard. The other type is referred to as OMTP, and has the ground signal on the sleeve and the microphone signal on Ring2.

Either of the four-terminal jack configurations are also backward-compatible with the normal three-terminal headphone plug. In this case, the sleeve of the headphone plug is twice as large and connects to both the Ring2 and sleeve terminals of the jack.

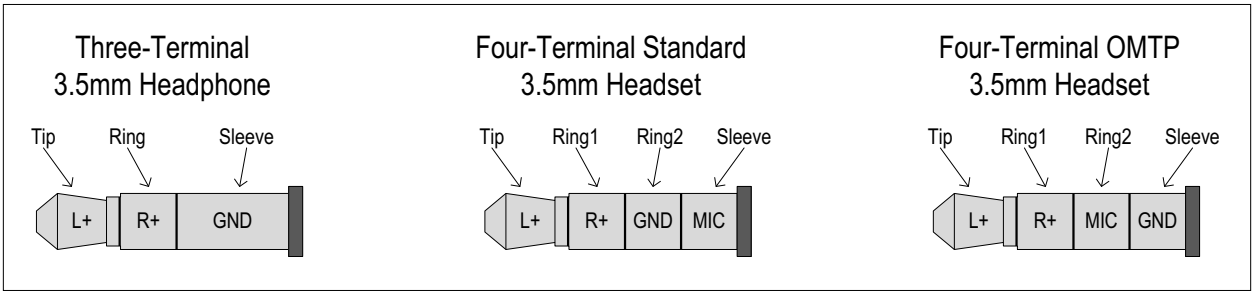



**Figure 6: Multi-Standard Headset Support**

***Insertion Scenarios***

Six main scenarios exist during insertion into a headset-capable jack—three plug types (four-terminal CTIA headset, four-terminal OMTP headsets, and three-terminal headphones), into two jack types (CTIA four-terminal and OMTP four-terminal).

For each of the main scenarios, two sub-scenarios exist—an intermediate state during insertion where the plug is 3/4-inserted into the jack (i.e., the terminals are shifted by one). This intermediate state, due to the design of typical four-terminal jacks, could potentially last several seconds.

1. The CTIA headset is plugged into the CTIA jack.
2. The OMTP headset is plugged into the OMTP jack.
3. The CTIA headset is plugged into the OMTP jack.
4. The OMTP headset is plugged into the CTIA jack.
5. The normal headphone (three-terminal plug) is plugged into the CTIA jack.
6. The normal headphone (three-terminal plug) is plugged into the OMTP jack.

The detection circuitry is triggered as a plug gets inserted into the headset jack, which is achieved by the presence signal from the jack sense state machine. In all scenarios, the jack sense terminal can trigger jack sense when the plug is only 3/4 inserted. Because the plug can be held in this position, it is possible that jack sense can be triggered several seconds before the plug is fully inserted.

## Jack Detect

The jack sensing circuit is used to detect the state of an off-chip resistor divider, whose value depends on which jacks are connected. This circuit is structured as a 4-bit successive approximation ADC, using an internal resistor-ladder DAC to mimic the external ladder, and a comparator that compares the external voltage versus the DAC voltage.

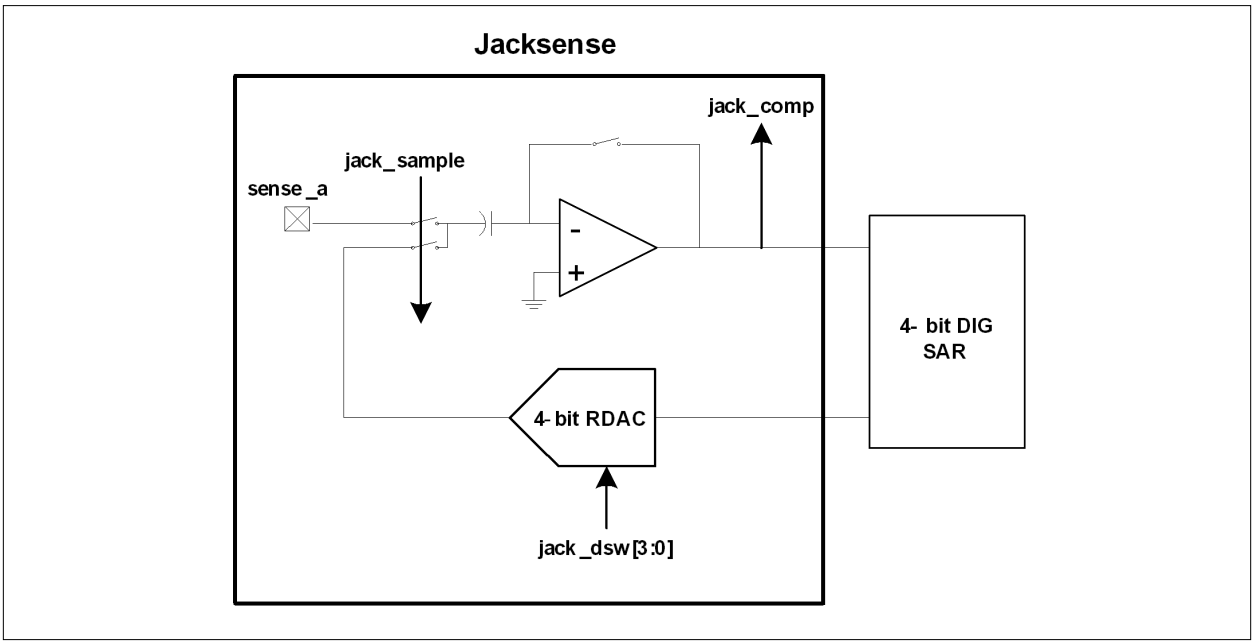



Figure 7: Jack Sensing

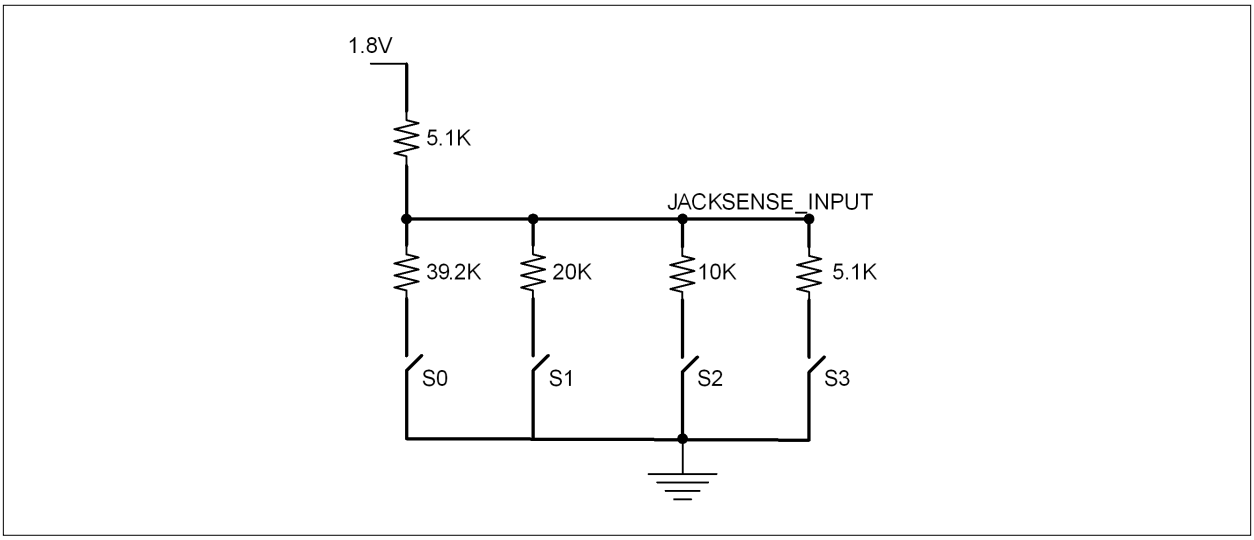



Figure 8: External Circuit

## Microphone Button Detect (Android Wired Audio Headset Specification)

The CX21986 includes support for a four-button interface using the microphone signal as defined in the *Android Wired Audio Headset Specification* (http://source.android.com/accessories/headset-spec.html)

The microphone itself is modeled as a 1kΩ or greater resistor. This is a reasonable model for common ECM microphones used in PCs and headsets. The CODEC is expected to provide a micbias voltage between 2.75V and 2.96V for compatibility with common microphones.

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

*Table 101:* ***Microphone Button Detect***

| 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 |

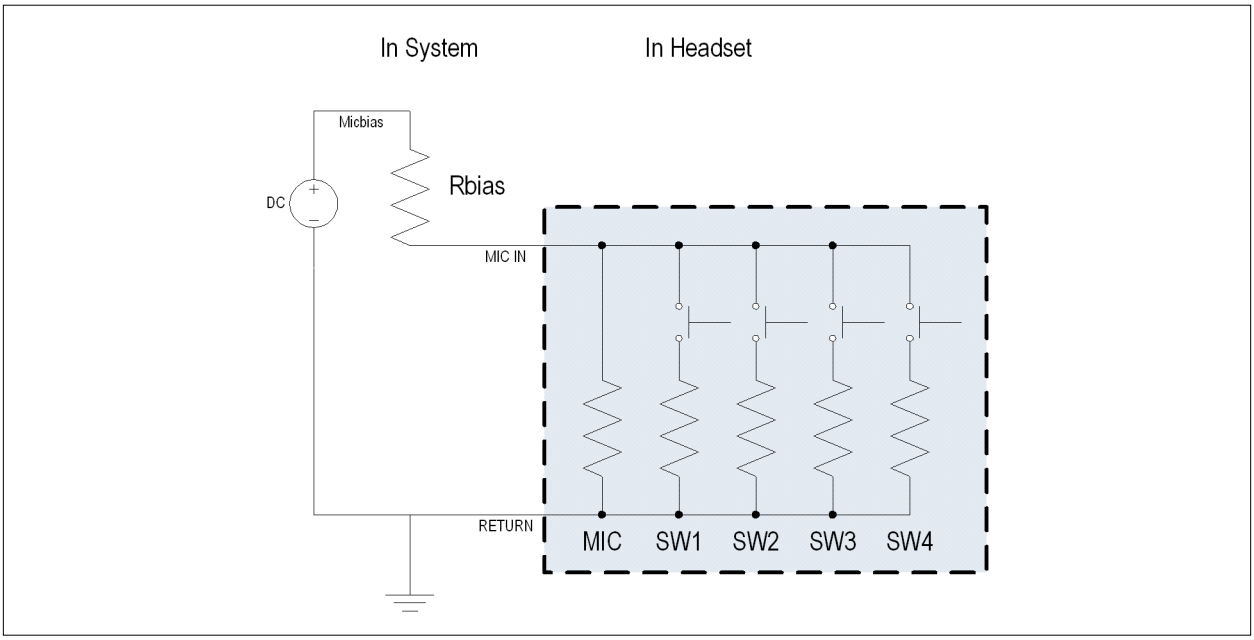



**Figure 9: Microphone Button Detect**

Button detection has an independent enable and several other controls. The initial detection of a button press requires a minimum load to avoid unintended events due to loud noises or microphones with large quiescent currents. The minimum load required to start a button detection cycle may be set to (values are approximate):

- 1kΩ
- 1.7kΩ
- 200Ω
- 500Ω

Because the button detection process measures the voltage at the microphone input in the presence of audio and noise, the ADC output is filtered before it is evaluated. The result is compared to a set of thresholds to determine which button (if any) was pressed. This result is further tested for stability before any action is taken.

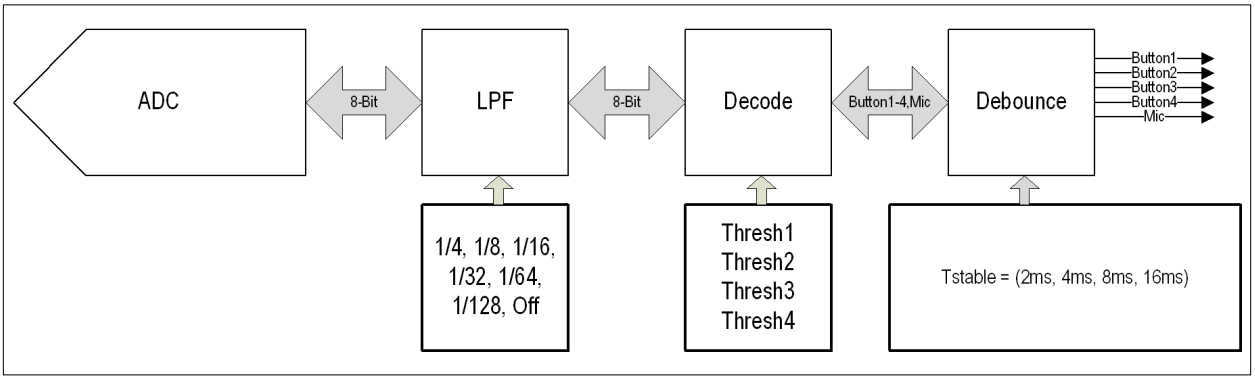



**Figure 10: ADC Clocked at 768 kHz**

The ADC is clocked at 768kHz for an effective sample rate of 48 kHz. This ADC is usually powered down until button detection is enabled, a headset is detected, and a button press is suspected. This is determined by monitoring the micbias detect signal when the headset detection state machine has determined that a headset has been inserted. This signal is filtered with the same settings as used during the headset detection process.

The raw ADC data is passed through a simple programmable single pole LPF. This filter has a minimum of four settings to adjust the response as needed. The filtered ADC output is then compared to four programmable 8-bit thresholds that determine which button is pressed:

- Button 1 threshold = An ADC value below this value is decoded as a B1 press
- Button 2 threshold = An ADC value below this value and greater than or equal to the button 1 threshold is decoded as a B2 press
- Button 3 threshold = An ADC value below this value and greater than or equal to the button 2 threshold is decoded as a B3 press
- Button 4 threshold = An ADC value below this value and greater than or equal to the button 3 threshold is decoded as a B4 press
- An ADC value equal to or above the button 4 threshold is decoded as MIC only—no buttons were pressed

The output from the button decoder is further filtered to reject the switch contact bounce and other issues. If the output from the decoder remains the same for the Tstable time (programmable with at least four values), then the value is considered valid.

In addition, a programmable long press counter (adjustable) is used to distinguish between a short and long press. The suggested timer values are 0.25s, 0.5s, 0.75s, 1s, 1.5s, 2s, 3s, and 4s.

The button status is available in a dedicated register with at least the following information:

- LP = Long press is detected
- MIC = Impedance was above the valid range for the buttons
- B4 = Button 4 was detected
- B3 = Button 3 was detected
- B2 = Button 2 was detected
- B1 = Button 1 was detected
- UP = A complete button press/release cycle has occurred
- DN = A button press has been detected, but the button may not have been released

# Control Signals

The CX21986 supports I2C, UART, and USB HID interfaces for serial communications with the CX21986 registers for command and control.

## I2C Configuration

Configure the following CX21986 pins for I2C operations (read or write):

- I2C_SPI_SCL = I2C clock
- I2C_SDA = I2C bidirectional serial data

### *I2C Write Operation*

Complete the following steps for a host writing to the slave CX21986 device.

1. Send a start sequence.
2. Send the I2C address of the slave with the R/W bit low.
3. Send the internal register address you want to write to, and then send the high byte followed by the low byte.
4. Send the data byte.
5. Optional: Send any further data bytes—burst operation.
6. Send the stop sequence.

**Write Single Location**

| S | Device Address | W (0) | A | ADDR byte 1 | A | ADDR byte 0 | A | DATA byte 0 | A | P |
|---|---|---|---|---|---|---|---|---|---|---|

**Write Auto increment**

| S | Device Address | W (0) | A | start ADDR byte 1 | A | start ADDR byte 0 | A | DATA [n] | A | DATA [n+1] | A | DATA [n+2] | A | DATA [n+3] | A | P |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

### *I2C Read Operation*

Complete the following steps for a host reading from the slave CX21986 device.

1. Send a start sequence.
2. Send the I2C address with the R/W bit low.
3. Send the internal register address you want to read, and then send the high byte followed by the low byte.
4. Send a start sequence again (repeated start).
5. Send the I2C address with the R/W bit high.
6. Read the data byte.
7. Optional: Send any further data bytes—Burst operation.

8. Send the stop sequence.

**Read Single Location**

| S | Device Address | W (0) | A | ADDR byte 1 | A | ADDR byte 0 | A | Sr | Device Address | R (1) | A | DATA byte 0 | N | P |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

**Read Auto increment**

| S | Device Address | W (0) | A | start ADDR byte 1 | A | start ADDR byte 0 | A | Sr | Device Address | R (1) | A | DATA [n] | A | DATA [n+1] | A | DATA [n+2] | A | DATA [n+3] | N | P |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

**Read Continuing Auto increment**

| S | Device Address | R | A | DATA [n+4] byte 1 | A | DATA [n+5] byte 0 | A | DATA [n+6] byte 1 | A | DATA [n+7] byte 0 | N | P |
|---|---|---|---|---|---|---|---|---|---|---|---|---|

## UART

The UART is a serial communication device that communicates with the CX21986 registers. The pins are UART_RX, UART_TX, and the GPIO0, which also serve as the CTS. The UART operates at 115,200bps, no parity bit, 8 data bits, and 1 stop bit. For a file transfer, the CTS should always be used to prevent a UART buffer overflow.

## Button Matrix

Each button in the keypad matrix returns a key value 0x0–0xF. This key value is matched to an action table, and the corresponding action is executed when the key is pressed. Any key function can be mapped to any key by changing the value in the BUTTON_DEFAULTS table, located from 0x50–0x5F. The following table keypad value matrix shows the KSOUT and KSIN lines.

*Table 102: **Button Matrix***

| | KSOUT4 | KSOUT5 |
|---|---|---|
| KSIN3 | 1 | 5 |
| KSIN2 | 2 | 6 |
| KSIN1 | 3 | 7 |
| KSIN0 | 4 | 8 |

## Read Key Value

Multi-press key combinations are supported on KSOUT4 and KSOUT5, and return a key with the value 0x9–0xE. All other multi-press key combinations are ignored and return 0x00 or no button.

*Table 103: Read Key Value*

| | KSOUT4 | KSOUT5 |
|---|---|---|
| KSIN3-KSIN2 | 9 | C |
| KSIN2-KSIN1 | A | D |
| KSIN1-KSIN0 | B | E |

When using the device in HS microphone mode (Host [2]=1), Google-style multi-button detection using the micbias can be used. The Google style micbias buttons map to keys 5, 6, 7, and 8.

***Note:*** The multi-key press is not supported with the micbias buttons.

The default of the table is as follows:

```
BUTTON_DEFAULTS: ;button location assignment variables. map buttons to specific
functions.
.DB    $00       ;50 NoButton              ;KPS5 --------
.DB    $01       ;51 Buttn1                ;KPS4 -----  |
.DB    $02       ;52 Buttn2                ; |  |
.DB    $03       ;53 Buttn3              ; ----------
.DB    $04       ;54 SPKMuteButtn        ;KPR3- 01 | 05 |  ;Multi Keypress
.DB    $05       ;55 VOLUPButtn          ; ----------      ;1-2 = 9
.DB    $06       ;56 VOLDNButtn          ;KPR2- 02 | 06 |  ;2-3 = A
.DB    $07       ;57 MicMuteButtn        ; ----------      ;3-4 = B
.DB    $08       ;58 HookButtn           ;KPR1- 03 | 07 |  ;5-6 = C
.DB    $09       ;59 Multi button 1-2    ; ----------      ;6-7 = D
.DB    $0A       ;5A Multi button 2-3        ;KPR0- 04 | 08 |  ;7-8 = E
.DB    $0B       ;5B Multi button 3-4        ; ----------
.DB    $0C       ;5C Multi button 5-6
.DB    $0D       ;5D Multi button 6-7
.DB    $0E       ;5E Multi button 7-8
.DB    $0F       ;5F N/A
```

The above table maps a telephone keypad that looks like the following:

| | KSOUT4 | KSOUT5 |
|---|---|---|
| KSIN3 | Button1 | VOLUP |
| KSIN2 | Button2 | VOLDN |
| KSIN1 | Button3 | MicMute |
| KSIN0 | SPKMute | Hook |

# Power Management

The CX21986 incorporates an internal charge pump (CP) and LDOs to generate all the power supplies required by the internal circuitry as well for external supporting functions.

## Charge Pump

The internal CP generates the positive (PVDD) and (MVDD) supplies required by the headphone driver. The CP takes the USB supply (3V to 5.5V) as input and generates PVDD (1.8V) and MVDD (–1.7V) voltages. To minimize power consumption, the charge pump can also power the core LDO which is used for supplying digital core functions.

The CP is designed with short circuit protection as well as in-rush current limit functions to comply with USB in-rush current specifications.

## Reference Voltages

The CX21986 incorporates built-in circuits for providing accurate reference voltages to all LDOs, charge pumps, ADCs and DACs.

## LDOs

The CX21986 consists of three LDOs:

- 3.3V LDO for digital I/O
- 1.8V for analog circuitry
- 1.25V for digital core functions

The LDOs are designed with short circuit protection as well as in-rush current limit functions to comply with USB in-rush current specifications.

## Wake-Up Sources

The CX21986 can be taken out of low power suspend mode by three sources:

- USB
- Button Push
- Jack Plug-In

# On-Chip Reference Clock Generation

The Internal Reference Clock (IRC) generator provides a stable 48 MHz clock using a 1 kHz source derived from USB Start of Frame (SOF) packets as a reference. The IRC output is provided as the reference clock to the PLL frequency synthesizer to generate the required audio master clock.

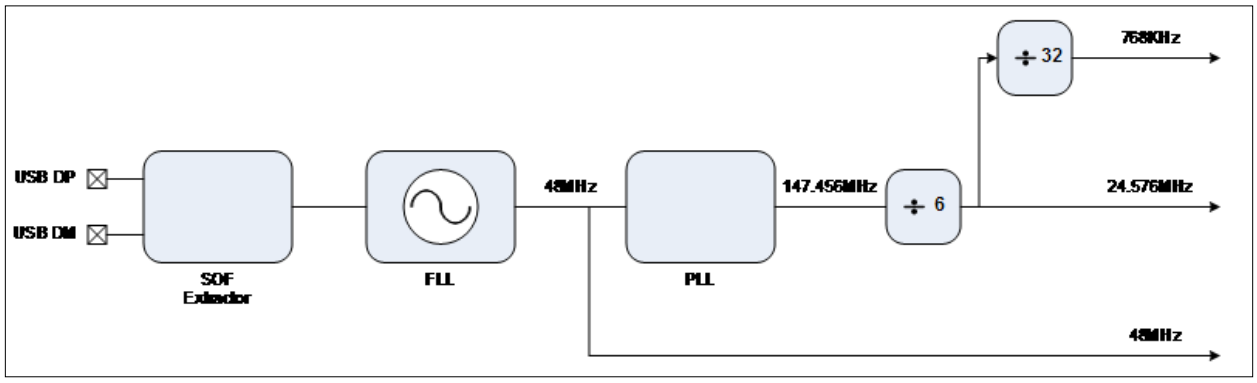



**Figure 11: On-Chip Reference Clock Generation**

# Synaptics Hum Noise Cancellation Technology

When a docking station is connected to powered speakers, and is in a sleep or power-down mode, hum noise occurs. The hum noise at the headphone output is caused by improper grounding.

The CX21986 hum noise cancellation IP incorporates internal circuitry to eliminate hum noise when the device is unpowered.

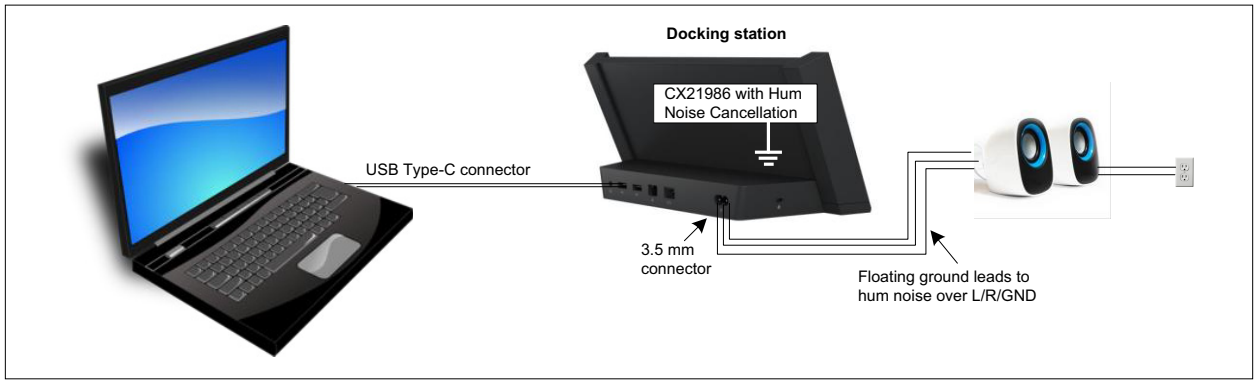



**Figure 12: Synaptics Hum Noise Cancellation Technology**

# Application Diagrams

## USB Type-C Docking Station and Audio Adapter

The CX21986 provides a high-quality, low-cost audio solution for USB docking stations and audio adapters. An audio adapter is simply a single dongle that plugs into a USB Type-C port and allows the user to plug in their legacy 3.5 mm headset. The CX21986 allows for the use of analog or digital microphones. Single or dual microphones are also supported. The capless stereo output port supports both headphones or external powered speakers. Further, the device can auto-switch between CTIA and OMTP-style headphones and detect whether or not a microphone is available. Multiple GPIO pins are available for button and LED support. Input power of 3.0V to 5.5V is required, and all other supplies are generated internally. The CX21986 also allows for powering external 3.3V circuitry, with the ability to deliver up to 90mA of current.

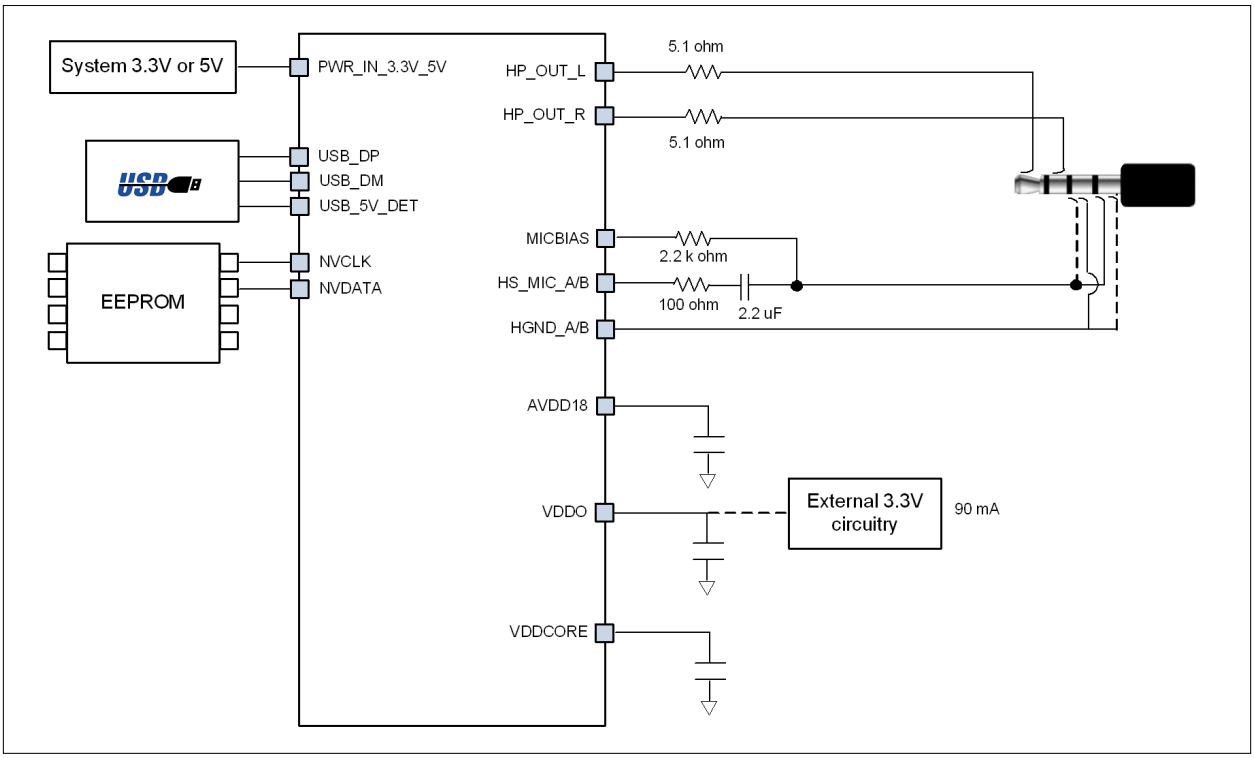



**Figure 13: USB Type-C Audio Adapter Docking Station and Audio Adapter Block Diagram**

# USB Type-C Headset with Android Button Support

The CX21986 provides a high-quality, low-cost solution for wired USB headsets used with smart phones or PCs. The CX21986 allows for the use of analog or digital microphones. Single or dual microphones are also supported. The capless stereo output port supports all headphones with an impedance of 16Ω or greater. Multiple GPIO pins are available for button and LED support. Input power of 3.0V to 5.5V is required, and all other supplies are generated internally. The CX21986 also allows for powering external 3.3V circuitry, with the ability to deliver up to 90mA of current.

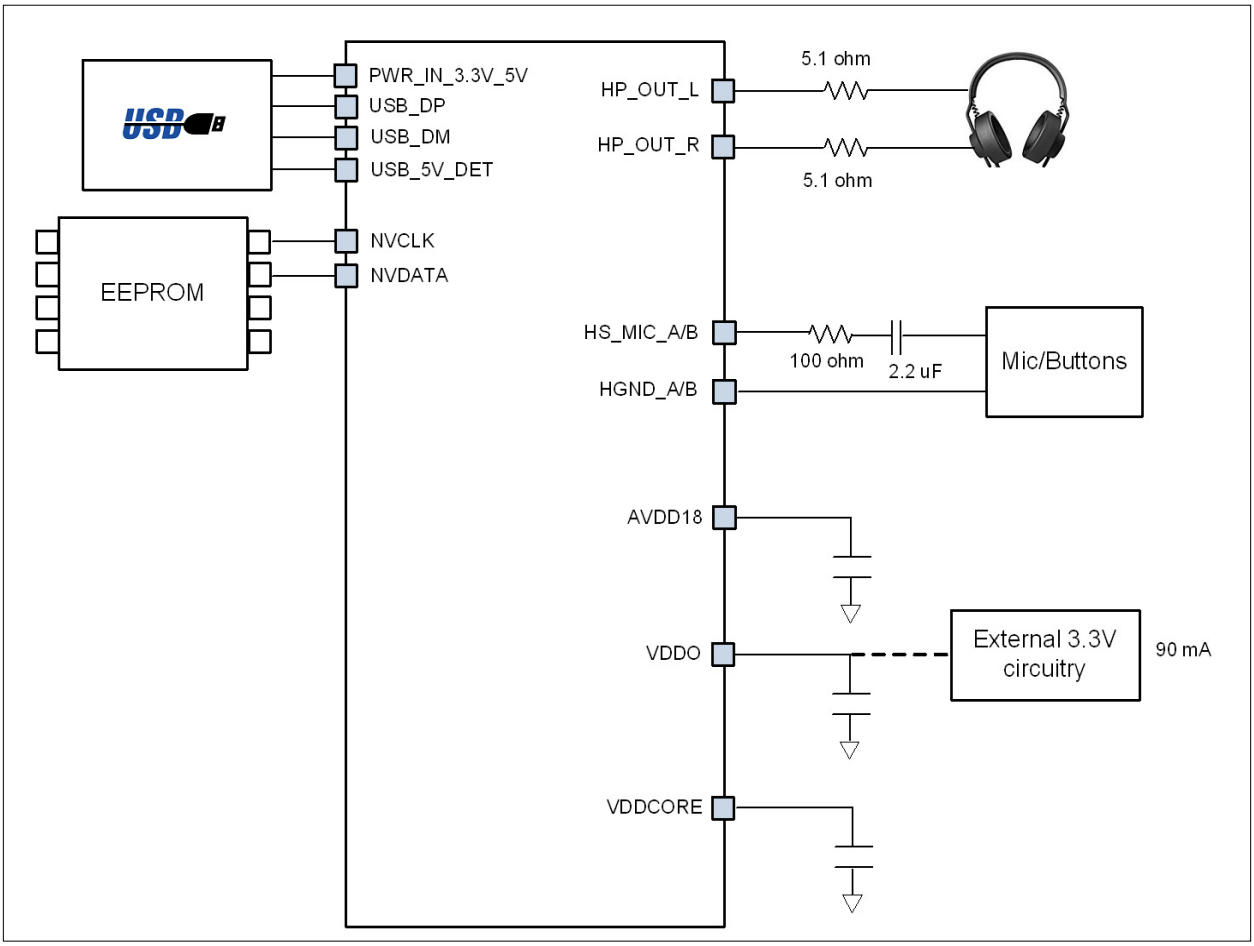



**Figure 14: USB Type-C Headset with Android Button Support Block Diagram**

# Electrical Characteristics

*Table 104: DC Supply Voltages*

| Parameter | Symbol | Minimum | Typical | Maximum | Units | Notes |
|---|---|---|---|---|---|---|
| Analog/Digital Power Supply | V5ANA/DIG | 3.0 | 5 | 5.5 | V | – |
| Analog Power | AVDD | 1.7 | 1.8 | 2.0 | V | – |
| Charge Pump Power | PVDD | 1.76 | 1.9 | 2.0 | V | – |
| | MVDD | –1.47 | –1.61 | –1.75 | | MVDD is shut off in idle mode. |
| Digital 3.3V I/O Supply | VDDO | 3.0 | 3.3 | 3.6 | V | This power rail can also source external circuitry with a maximum current of 90mA. |
| Digital Core Supply | VDD, VDD_CORE | 1.24 | 1.3 | 1.4 | V | - |
| VREF | VREF | 1.08 | 1.2 | 1.32 | V | VREF is shut off in idle mode. |

*Table 105: DC Characteristics—TTL Compatible (GPIOs and Keypad Interface)*

| Parameter | Symbol | Minimum | Typical | Maximum | Units | Notes |
|---|---|---|---|---|---|---|
| Input Voltage Low | $V_{IL}$ | –0.5 | - | 0.8 | V | - |
| Input Voltage High | $V_{IH}$ | 2 | - | Vdd+0.5 | V | - |
| Output Voltage Low | $V_{OL}$ | 0 | - | 0.4 | V | - |
| Output Voltage High | $V_{OH}$ | 2.4 | - | Vdd | V | - |
| GPIO Output Sink Current at 0.4V Maximum | - | - | - | 12 | mA | - |
| GPIO Output Source Current at 2.97V Minimum | - | - | - | 12 | mA | - |
| GPIO Rise/fall Time | - | - | - | 4 | ns | 25% to 75%. |

***Note:*** Test conditions unless otherwise stated:

- VDDO=3.3 + 0.3 VDC
- TA = 0°C to 70°C
- External load = 50 pF

*Table 106: DAC Performance (32-Ohm Load)*

| Parameters | Minimum | Typical | Maximum | Units | Comments |
|---|---|---|---|---|---|
| Dynamic Range 48 kHz, 24-bit | – | -103 | – | dBFS | A-weighted, 20 Hz to 20 kHz. -60 dBFS |
| Dynamic Range 96 kHz, 24-bit | – | -103 | – | dBFS | A-weighted, 20 Hz to 20 kHz. -60 dBFS |
| Dynamic Range 192 kHz, 16-bit | – | -97 | – | dBFS | A-weighted, 20 Hz to 20 kHz. -60 dBFS |
| THD+N 48 kHz, 24-bit | – | -87 | – | dBFS | 20 Hz to 20 kHz into 32 ohm. -3 dBFS |
| THD+N 96 kHz, 24-bit | – | -87 | – | dBFS | 20 Hz to 20 kHz into 32 ohm. -3 dBFS |
| THD+N 192 kHz, 16-bit | – | -87 | – | dBFS | 20 Hz to 20 kHz into 32 ohm. -3 dBFS |
| Crosstalk | – | -81 | – | dB | HGND series resistance = 0 ohms |
| Crosstalk | – | -76 | – | dB | HGND series resistance = 1 ohm |

*Table 107: DAC Performance (10k-Ohm Load)*

| Parameters | Min. | Typ. | Max. | Units | Comments |
|---|---|---|---|---|---|
| Dynamic Range 48kHz, 24-bit | – | -102 | – | dBFS | A-weighted, 20 Hz to 20 kHz. -60 dBFS |
| Dynamic Range 96kHz, 24-bit | – | -102 | – | dBFS | A-weighted, 20 Hz to 20 kHz. -60 dBFS |
| Dynamic Range 192kHz, 16-bit | – | -97 | – | dBFS | A-weighted, 20 Hz to 20 kHz. -60 dBFS |
| THD+N 48 kHz, 24-bit | – | -88 | – | dBFS | 20 Hz to 20 kHz into 10 kohm. -3 dBFS |
| THD+N 96 kHz, 24-bit | – | -88 | – | dBFS | 20 Hz to 20 kHz into 10 kohm. -3 dBFS |
| THD+N 192 kHz, 16-bit | – | -88 | – | dBFS | 20 Hz to 20 kHz into 10 kohm. -3 dBFS |
| Crosstalk | – | -81 | – | dB | – |

*Table 108: ADC Performance (Headset Mic)*

| Parameters | Minimum | Typical | Maximum | Units | Comments |
|---|---|---|---|---|---|
| Dynamic Range<br>Mic boost = 0 dB | – | -97 | – | dBFS | A-weighted, 20 Hz to 20 kHz,<br>-60 dBFS. 48 kHz, 24-bit |
| Dynamic Range<br>Mic boost = 12 dB | – | -96 | – | dBFS | A-weighted, 20 Hz to 20 kHz,<br>-60 dBFS. 48 kHz, 24-bit |
| Dynamic Range<br>Mic boost = 24 dB | – | -88 | – | dBFS | A-weighted, 20 Hz to 20 kHz,<br>-60 dBFS. 48 kHz, 24-bit |
| Dynamic Range<br>Mic boost = 36 dB | – | -77 | – | dBFS | A-weighted, 20 Hz to 20 kHz,<br>-60 dBFS. 48 kHz, 24-bit |
| Dynamic Range<br>Mic boost = 0 dB | – | -91 | – | dBFS | A-weighted, 20Hz to 20kHz,<br>-60 dBFS. 48 kHz, 16-bit |
| Dynamic Range<br>Mic boost = 12 dB | – | -91 | – | dBFS | A-weighted, 20 Hz to 20 kHz,<br>-60 dBFS. 48 kHz, 16-bit |
| Dynamic Range<br>Mic boost = 24 dB | – | -87 | – | dBFS | A-weighted, 20 Hz to 20 kHz,<br>-60 dBFS. 48 kHz, 16-bit |
| Dynamic Range<br>Mic boost = 36 dB | – | -77 | – | dBFS | A-weighted, 20 Hz to 20 kHz,<br>-60 dBFS. 48 kHz, 16-bit |
| THD+N<br>Mic boost = 0 dB | – | -86 | – | dBFS | 20 Hz to 20 kHz, -3 dBFS.<br>48 kHz, 16-bit |
| THD+N<br>Mic boost = 12 dB | – | -86 | – | dBFS | 20 Hz to 20 kHz, -3 dBFS.<br>48 kHz, 16-bit |
| THD+N<br>Mic boost = 24 dB | – | -83 | – | dBFS | 20 Hz to 20 kHz, -3 dBFS.<br>48 kHz, 16-bit |
| THD+N<br>Mic boost = 36 dB | – | -74 | – | dBFS | 20 Hz to 20 kHz, -3 dBFS.<br>48 kHz, 16-bit |

# Pin Descriptions

## 43-Pin WLCSP Hardware Interface Signals

The following figure shows the CX21986 43-pin WLCSP device signals by major interface.

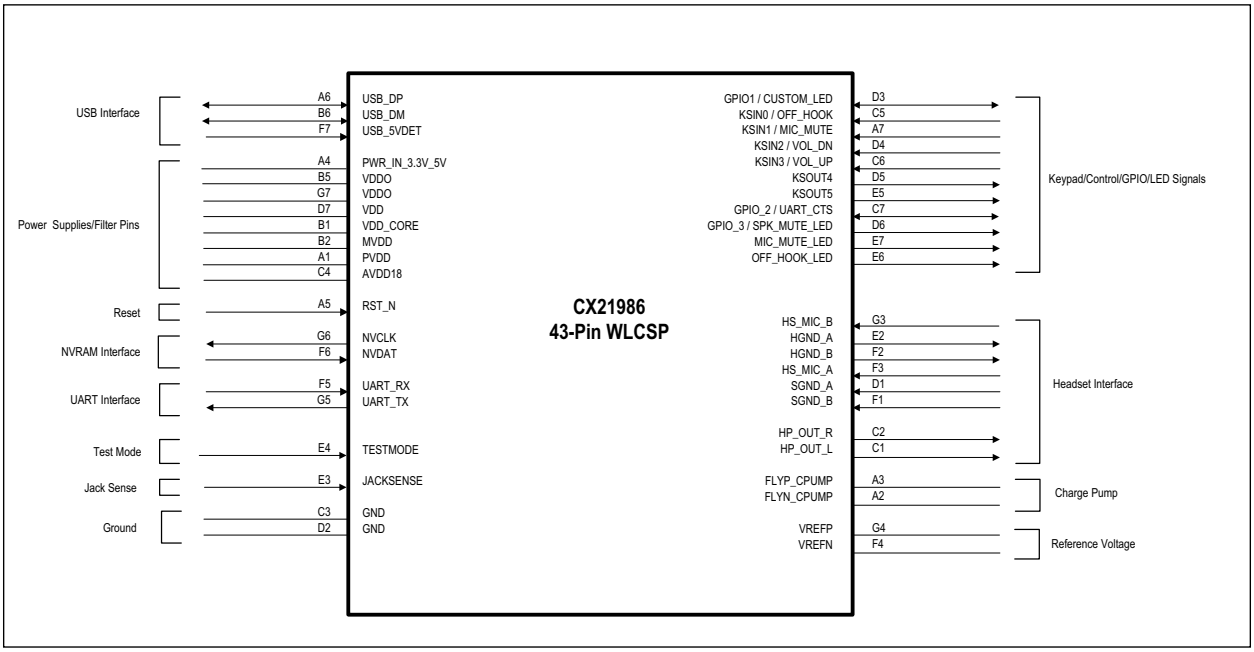



**Figure 15: CX21986 43-Pin WLCSP Hardware Interface Signals**

# 43-Pin WLCSP Pin Signals

Figure 16 and Table 109 show the CX21986 pinout.

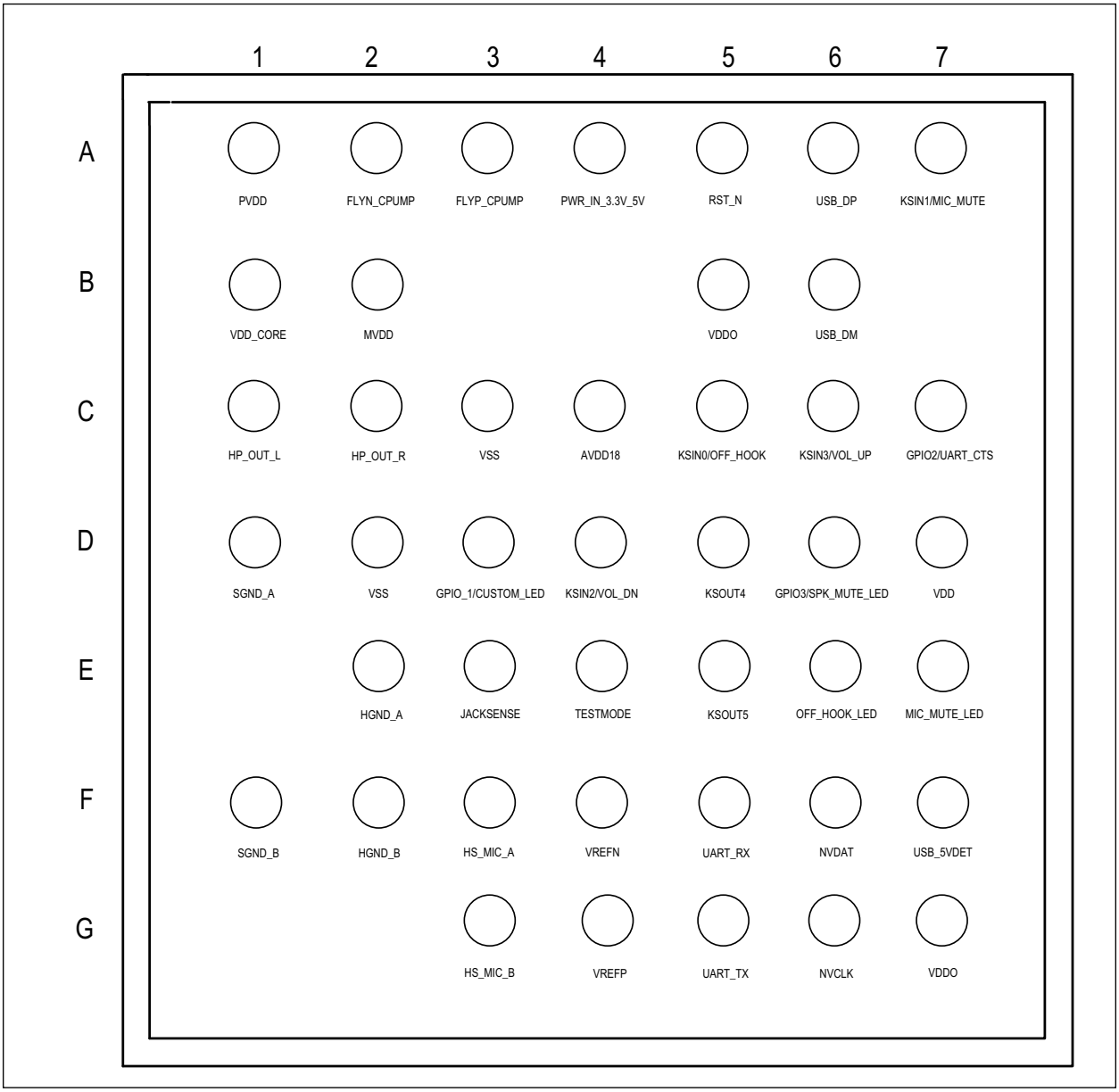



**Figure 16: CX21986 43-Pin WLCSP Pin Signals**

# Pin Assignments

The following table lists the CX21986 pin assignments.

*Table 109: Pin Assignments*

| Pin | Pin Name |
|---|---|
| A1 | PVDD |
| A2 | FLYN_CPUMP |
| A3 | FLYP_CPUMP |
| A4 | PWR_IN_3.3V_5V |
| A5 | RST_N |
| A6 | USB_DP |
| A7 | KSIN1/MIC_MUTE |
| B1 | VDD_CORE |
| B2 | MVDD |
| B3 | (NO BALL) |
| B4 | (NO BALL) |
| B5 | VDDO |
| B6 | USB_DM |
| B7 | (NO BALL) |
| C1 | HP_OUT_L |
| C2 | HP_OUT_R |
| C3 | VSS |
| C4 | AVDD18 |
| C5 | KSIN0/OFF_HOOK |
| C6 | KSIN3/VOL_UP |
| C7 | GPIO2/UART_CTS |
| D1 | SGND_A |
| D2 | VSS |
| D3 | GPIO1/CUSTOM_LED |
| D4 | KSIN2/VOL_DN |
| D5 | KSOUT4 |
| D6 | GPIO3/SPK_MUTE_LED |
| D7 | VDD |
| E1 | (NO BALL) |
| E2 | HGND_A |
| E3 | JACKSENSE |
| E4 | TESTMODE |
| E5 | KSOUT5 |
| E6 | OFF_HOOK_LED |
| E7 | MIC_MUTE_LED |
| F1 | SGND_B |
| F2 | HGND_B |
| F3 | HS_MIC_A |
| F4 | VREFN |
| F5 | UART_RX |
| F6 | NVDAT |
| F7 | USB_5VDET |
| G1 | (NO BALL) |
| G2 | (NO BALL) |
| G3 | HS_MIC_B |
| G4 | VREFP |
| G5 | UART_TX |
| G6 | NVCLK |
| G7 | VDDO |

# Hardware Signal Definitions

*Table 110: Hardware Signal Definitions*

| Label | Pin | I/O Type | Signal Name/Description |
|---|---|---|---|
| **USB Signals** | | | |
| USB_DP | A6 | Digital In | **USB Data Plus.** |
| USB_DM | B6 | Digital In | **USB Data Minus.** |
| **Control Signals** | | | |
| RST_N | A5 | Digital In | **Reset.** Active low input asserted to initialize registers, sequencers, and signals to a consistent reset state. |
| MIC_MUTE_LED | E7 | Digital Out | **MIC_MUTE_LED.** Turns the external LED on when the microphone mute button is pushed. Connect to an anode of the LED. |
| TEST MODE | E4 | Digital In | **TEST.** Connect to ground through a 0Ω resistor. Include the test point on this pin for one-time-programmable, non-volatile memory programming. |
| USB_5V_DET | F7 | Digital In | **5V_USB_DETECT.** Connect this pin to the USB 5V through the external resistor divider. |
| OFF_HOOK_LED | E6 | Digital Out | **Turns on the External LED upon the Phone Going Off-hook.** Connect to the anode of the LED. |
| GPIO_1/CUSTOM_LED | D3 | Digital I/O | **GPIO or Custom LED Driver.** Connect to the anode of the LED. |
| GPIO3/SPK_MUTE_LED | D6 | Digital I/O | **GPIO_3/SPK_MUTE_LED.** Turns the external LED on when the Playback Mute button is pushed. Connect to an anode of the LED. This pin can also be programmed as GPIO. |
| **Keypad Signals** | | | |
| KSIN[0:3] | C5, A7, D4, C6 | Digital In | **Keypad Scan Inputs.** Assignment for the following standard four buttons:<br>• KSIN0 = Off-hook<br>• KSIN1 = Microphone Mute<br>• KSIN2 = Volume Down<br>• KSIN3 = Volume Up |
| KSOUT4, KSOUT5 | D5, E5 | Digital Out | **Keypad Scan Output.** Outputs a strobe signal as part of the process to detect if the Volume Up, Volume Down, Microphone Mute, or Playback Mute buttons have been pushed. |
| **NVRAM Signals** | | | |
| NVCLK | G6 | Digital Out | **NVRAM Clock.** Connects to the clock pin of the external EEPROM. |
| NVDATA | F6 | Digital I/O | **NVRAM Data.** Connects to the data pin of the external EEPROM. |
| **UART Interface** | | | |
| UART_TX | G5 | Digital Out | **UART Serial Transmission Data.** |
| UART_RX | F5 | Digital In | **UART Serial Receiving Data.** |
| GPIO_2/UART_CTS | C7 | Digital Out | **GPIO or UART Clear-to-Send (CTS).** |
| **Jack Sense** | | | |
| JACKSENSE | E3 | Analog In | **Jack Sense.** Connect to the external resistor network to sense up to four jacks. |

| Label | Pin | I/O Type | Signal Name/Description |
|---|---|---|---|
| **Charge Pump (CP)** | | | |
| PVDD | A1 | Power | **CP Positive Power Supply.** Internally generated positive CP power. Connect to 10µF and 0.1µF external filtering. |
| FLYP_CPUMP | A3 | Power | **CP Positive Transfer Charge.** Connect to CPUMP_FLYN through a 2.2µF capacitor. |
| FLYN_CPUMP | A2 | Power | **CP Negative Transfer Charge.** Connect to CPUMP_FLYP through a 2.2µF capacitor. |
| MVDD | B2 | Power | **CP Negative Power Supply.** Internally generated negative CP power. Connect to 2.2µF and 0.1µF external filtering. |
| **Reference Voltage** | | | |
| VREFP | G4 | Power | **VREFP—Internally Generated Supply.** Connect to 0.1µF and 1.0µF to VREFN. |
| VREFN | F4 | Power | **Connect to Analog Ground (AGND).** |
| **Microphone and Headphone Interfaces** | | | |
| HS_MIC_B | G3 | Analog Input | **Headset Microphone Input.** Mono microphone input for OMTP-style headsets. |
| HS_MIC_A | F3 | Analog Input | **Headset Microphone Input.** Mono microphone input for CTIA-style headsets. |
| HGND_B | F2 | GND/Micbias | **Headset Microphone Ground.** Microphone:<br>• Ground terminal for OMTP-style headsets<br>• Bias for CTIA-style headsets<br>Note: 2.75V and 2.96 micbias is applicable. |
| HGND_A | E2 | GND/Micbias | **Headset Microphone Ground.** Microphone:<br>• Ground terminal for CTIA-style headsets<br>• Bias for OMTP-style headsets<br>Note: 2.75V and 2.96 micbias is applicable. |
| HP_OUT_L | C1 | Analog Out | **Left Headphone.** |
| HP_OUT_R | C2 | Analog Out | **Right Headphone.** |
| SGND_A | D1 | Analog Input | **Headset Sense Pin.** |
| SGND_B | F1 | Analog Input | **Headset Sense Pin.** |
| **Power Supplies** | | | |
| VDDO | B5, G7 | Digital I/O Power | **3.3V I/O Supply.** Connect to 2.2µF and 2 x 0.1µF to GND. |
| VDD_CORE | B1 | Digital Power Output | **Digital Supply for Core Logic.** Connect 0.1µF and 2.2µF to GND. |
| VDD | D7 | Digital Power Input | **Digital Supply.** Connect 0.1µF to GND. |
| AVDD18 | C4 | Analog power | **1.8V Analog Supply.** Connect 2.2µF and 0.1µF to AGND. |
| PWR_IN_3.3V_5V | A4 | Analog/Digital Power | **3.3V or 5V Analog/Digital Input Power Supply.** Connect 0.1µF and 4.7µF to GND. |
| **Ground Signal** | | | |
| VSS | D2, C3 | Power | **Ground.** |

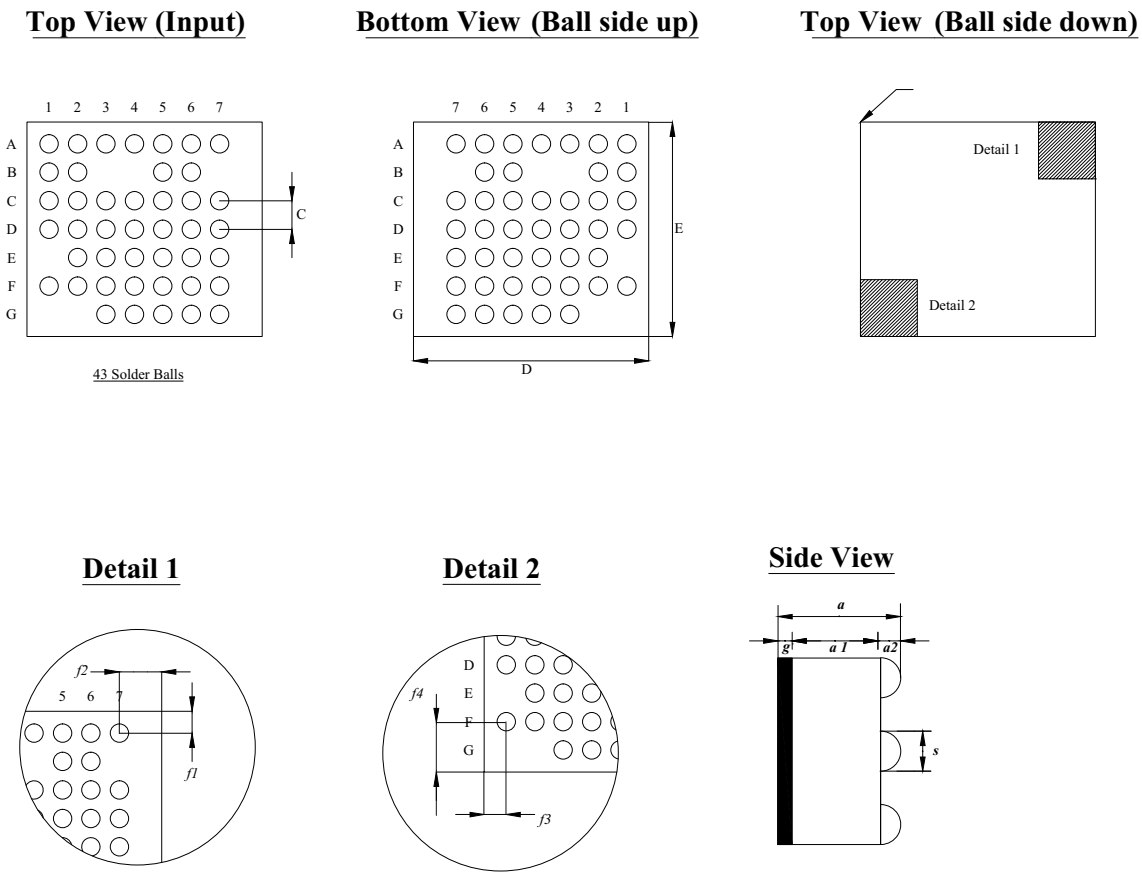
# Power Consumption

*Table 111:* ***Power Consumption***

| Parameters | Min. | Typ. | Max. | Units | Comments |
|---|---|---|---|---|---|
| USB Suspend w/o remote wakeup | – | 0.88 | – | mA | – |
| USB Suspend with remote wakeup and headphone plugged in | – | 0.89 | – | mA | – |
| USB Suspend with remote wakeup and headset plugged in | – | 1.29 | – | mA | – |
| Idle | – | 12.14 | – | mA | Powered on, playback, and record are not active. |
| Headphone Active, Playing Silence | – | 17.23 | – | mA | Headphone load = 32Ω |
| Headphone Active, Output Level = (0.1mW), playing out 16-bit, 48 kHz sample rate, 997 Hz tone. | – | 18.75 | – | mA | Headphone load = 32Ω |
| Headphone Active, Output Level = (0.1mW), playing out 24-bit, 96 kHz sample rate, 997 Hz tone. | – | 19.51 | – | mA | Headphone load = 32Ω |
| Headphone Active, Output Level = (0.1mW), playing out 16-bit, 192 kHz sample rate, 997 Hz tone. | – | 20.76 | – | mA | Headphone load = 32Ω |
| Headphone Active, Output Level = (0.5mW), playing out 16-bit, 48 kHz sample rate, 997 Hz tone. | – | 20.72 | – | mA | Headphone load = 32Ω |
| Headphone Active, Output Level = (0.5mW), playing out 24-bit, 96 kHz sample rate, 997 Hz tone. | – | 21.48 | – | mA | Headphone load = 32Ω |
| Headphone Active, Output Level = (0.5mW), playing out 16-bit, 192 kHz sample rate, 997 Hz tone. | – | 22.75 | – | mA | Headphone load = 32Ω |
| Headphone Active, Output Level = (1mW), playing out 16-bit, 48kHz sample rate, 997 Hz tone. | – | 22.27 | – | mA | Headphone load = 32Ω |
| Headphone Active, Output Level = (1mW), playing out 24-bit, 96 kHz sample rate, 997 Hz tone. | – | 22.99 | – | mA | Headphone load = 32Ω |
| Headphone Active, Output Level = (1mW), playing out 16-bit, 192 kHz sample rate, 997 Hz tone. | – | 24.27 | – | mA | Headphone load = 32Ω |
| Record Path Active | – | 16.45 | – | mA | Microphone input = -36 dBV, Input Boost = 36 dB |
| Record Path Active<br>Headphone Active, Output Level = (1mW), Playing out 16-bit, 48 kHz sample rate, 997 Hz tone. | – | 26.74 | – | mA | Microphone input = -36 dBV. Input boost = 36 dB<br>Headphone load = 32Ω. |
| Record Path Active<br>Headphone Active, Output Level = (1mW), playing out 24-bit, 96 kHz sample rate, 997 Hz tone. | – | 27.34 | – | mA | Microphone input = -36 dBV. Input boost = 36 dB<br>Headphone load = 32Ω. |

***Note:*** Input power = 5.0V

# Package Dimensions and Thermal Specifications




Unit : mm

| | DIMENSION | |
|---|---|---|
| Symbols | Nominal | Tolerance |
| a | 0.600 | |
| a 1 | 0.367 | ± 0.015 |
| a 2 | 0.208 | ± 0.015 |
| C | 0.400 | |
| D | 3.308 | ± 0.025 |
| E | 3.020 | ± 0.025 |
| f 1 | 0.310 | |
| f 2 | 0.598 | |
| f 3 | 0.310 | |
| f 4 | 0.310 | |
| g | 0.025 | ± 0.003 |
| s | 0.260 | ± 20% |

**Remarks**

. Unless otherwise specified dimension are in millimeters

| Decimal | Angular |
|---|---|
| X.X±0.1 | ±1° |
| X.XX±0.05 | |
| X.XXX±0.030 | |

. Coplanarity Spec (after Bumping Process)

1) Coplanarity Criteria : Max. 0.030 mm
2) Sample Size : All bumps/ die, 5 dies/ wafer, 5 wafers/ lot
3) Method : Peak to peak
(The tallest bump height - The shortest bump height, within die)

Figure 17: Package Dimensions 43-Pin WLCSP

The following table defines the thermal specifications.

***Table 112: Thermal Specifications***

| Parameter | Symbol | Minimum | Typical | Maximum | Units |
|---|---|---|---|---|---|
| Operating Temperature Range | $T_A$ | –25 | – | 85 | °C |
| Operating Junction Temperature | $T_J$ | –25 | – | 125 | °C |
| Thermal Resistance[1, 2, 3] (four layer) | $\theta_{JA}$ | – | 53 | – | °C/W |

1.Measurements per JEDEC EIA/JESD 51. The $\theta_{JA}$ of application boards with more than four layers stay the same or improve if the PCB construction is similar to the JEDEC EIA/JESD 51 defined four-layer PCB (2S2P plus vias).
2.For a given power dissipation, die temperature can be calculated as follows: $T_J = T_A$ + (power dissipated *$\theta_{JA}$).
3.Four layer board specification: PCB Dimension 101.5mm×114.5mm and 1.6mm thick FR-4, Top and Bottom Layer Thickness 0.070 (2 oz.), Inner plane Thickness 0.035 (1 oz.).

# Ordering Information

The following table shows the ordering information.

*Table 113: Ordering Information*

| Device Part Number | Part Number | Features | Package |
|---|---|---|---|
| DSAC-L986-10CH | CX21986-10Z | USB interface | 43-pin WLCSP |

The devices in this publication are lead-free (Pb-Free) and China RoHS compliant . Contact the local Synaptics sales office for advanced software options.

## Copyright



## Trademarks



## Contact Us

Visit our website at www.synaptics.com to locate the Synaptics office nearest you.