# CM6523 / CM6523B

USB Audio Sound Chip

## DESCRIPTION

CM6523 / CM6523B are a USB 2.0 audio chip builds in 8051 for flexible applications. With internal 2-channel ADC and DAC, S/PDIF in/out interface and USB 2.0 High speed switch makes it suits for different kinds of Docking applications, such as iDevice docking, Android Phone or Tablet/Slate docking device and Netbook/Notebook docking.

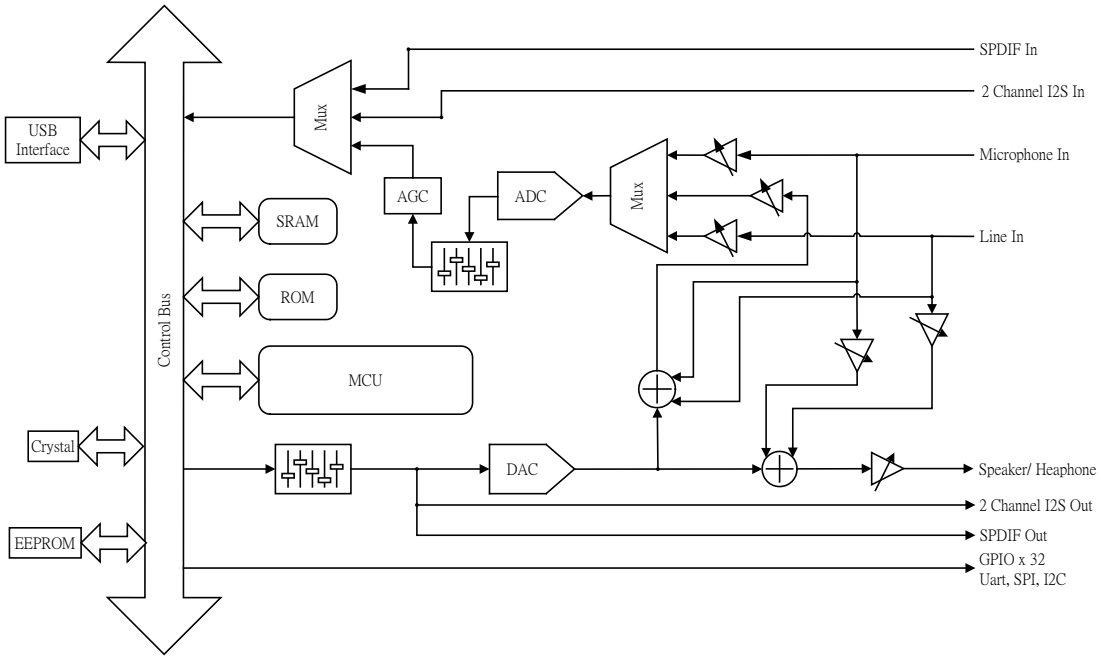
CM6523 / CM6523B is compatible with USB audio Class 1.0, thus it can plug & play without additional software installation on the major operation systems. The internal DAC/ADC and S/PDIF out interface support 96K/88.2K/48K/44.1KHz sampling rate and 16/24bit resolution.

CM6523 / CM6523B integrates the Equalizer on both playback and recording paths. With I2S in/out interface, it can connect external DAC to get higher audio quality or external DSP to process the audio data.

## FEATURES

- USB 2.0 full-speed compliant
- USB Audio Class 1.0 compliant
- USB Human Interface Device (HID) Class 1.1 compliant
- Build in USB 2.0 High speed switch
- 2 channel DAC for audio output interface
- 2 channel ADC for audio input interface
- Build in 96K/88.2K/48K/44.1KHz and 16/24bit S/PDIF transmitter
- Build in Equalizer on both playback and recording paths
- Build in AGC on recording path
- Support Digital Microphone interface
- Support control, interrupt, bulk, and isochronous data transfers
- Support Synchronous and Asynchronous audio data synchronization
- Embedded 1T 8051
- Master I2C control interface for external audio

## BLOCK DIAGRAM



# Release Note

| Revision | Date | Description |
|---|---|---|
| 0.9 | 2011/11/10 | First release of preliminary technical information |
| 0.91 | 2011/1/12 | Modify some pin description |
| 0.92 | 2012/02/10 | -Modify GPIO default status |
| 0.93 | 2012/04/03 | -modify PDSW as DO pin<br>-add information in chapter 6, 7 |
| 0.94 | 2012/04/20 | -modify some wordings |
| 0.95 | 2012/05/17 | -add more audio quality test into audio performance chapter |
| 0.96 | 2012/06/13 | -modify power pin description |
| 0.97 | 2012/06/21 | -modify GPIO pin number, subtract 1 |
| 0.98 | 2012/06/29 | -modify package part number to CM6523 |
| 1.00 | 2012/10/19 | -Formal Release |
| 1.01 | 2013/01/25 | -Remove S/PDIF 32K sample rate support |
| 1.02 | 2013/03/01 | -Add CM6523B support. |

## TABLE OF CONTENTS

# 1 Description and Overview

CM6523 / CM6523B is a USB 2.0 audio chip builds in 8051 for flexible applications. With internal 2-channel ADC and DAC, S/PDIF in/out interface and USB 2.0 High speed switch makes it suits for different kinds of Docking applications, such as iDevice docking, Android Phone or Tablet/Slate docking device and Netbook/Notebook docking.

CM6523 / CM6523B is compatible with USB audio Class 1.0, thus it can plug & play without additional software installation on the major operation systems. The internal DAC/ADC and S/PDIF out interface support 96K/88.2K/48K/44.1KHz sampling rate and 16/24bit resolution.

CM6523 / CM6523B integrates the Equalizer on both playback and recording paths. With I2S in/out interface, it can connect external DAC to get higher audio quality or external DSP to process the audio data.

# 2 Features

## USB Compliance

- USB Spec. Rev.2.0 full-speed mode compatible
- 3 USB upstream ports for connecting to PC and IPOD at the same time
- Latest USB Audio Device Class Definition Release 1.0 compatible
- USB Human Interface Device (HID) Class Definition Release 1.1 compliant
- Supports USB suspend/resume/reset functions
- Supports control, interrupt, bulk, and isochronous data transfers

## Audio Engine

- Playback Streams:
  - Default Sample Rates: 8K/11.025K/16K/22.05K/32K/44.1K/48K/88.2K/96K
  - Supported Bit Length: 16/24 bit
  - DMA supports 2-channel data to DAC/I2S output
  - DMA supports S/PDIF output
- Capture Streams:
  - Default Sample Rates: 8K/11.025K/16K/22.05K/32K/44.1K/48K/88.2K/96K
  - Supported Bit Length: 16/24 bit
  - DMA supports 2-channel data from ADC/I2S input
  - DMA supports S/PDIF input
- Digital mixing/routing engine to mix input streams to output streams

## Audio I/O

- 2 pairs I2S or Left-Justified serial audio output interface
- 2 pairs I2S or Left-Justified serial audio input interface

- 2 channel Mic in
- 2 channel digital Mic in
- 2 channel line in
- 2 channel line out
- Built-in 96K/88.2K/48K/44.1KHz and 16/24-bit S/PDIF receiver
- Built-in 96K/88.2K/48K/44.1KHz and 16/24-bit S/PDIF transmitter

## Integrated 8051 Micro-processor

- Embedded 8051 micro-processor to handle the comment/protocol transactions
- Connects to an external EEPROM memory for firmware codes
- HID interrupts can be implemented via firmware codes
- Provides maximum HW configuration flexibility with firmware code upgrade
- VID/PID/Product String can be customized via firmware code programming

## Control Interface

- Master I2C control interface for external audio devices or EEPROM access
- Master SPI control interface for external audio devices or EEPROM access
- Max. 32 GPIO pins can be configured via firmware programming
- GPIOs are configured as HID key and LED indicators and IR receiver

## General

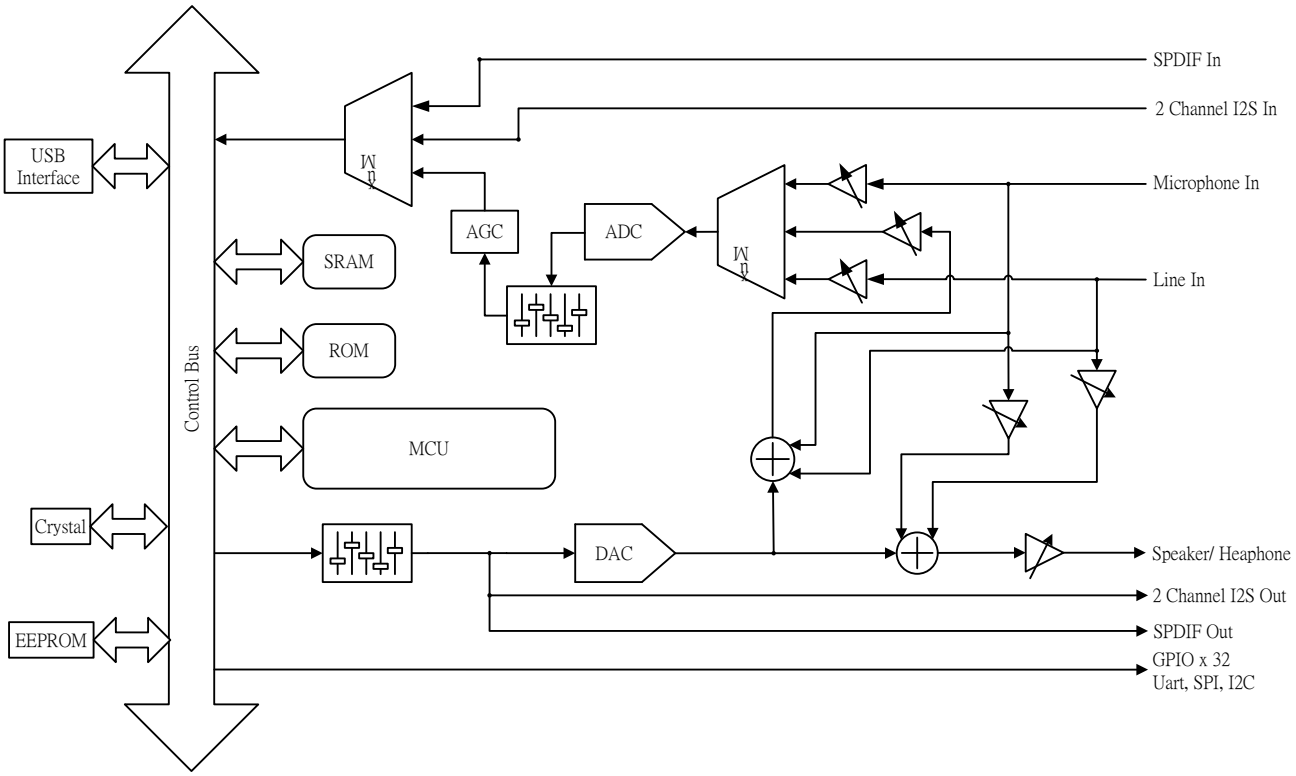
- HW pin for USB Audio Class 1.0 application mode configuration including Speaker/Headset/Docking/Mic
- HW pin for A-A path enable/disable
- HW pin for Self-power or Bus-power mode selection
- HW EQ for both playback and record path
- HW pin for EQ enable/disable selection
- HW pin for PID selection
- Only single 12MHz crystal input is required (embedded PLL function)
- Single 5V power supply (embedded 5V to 1.8V regulator for digital core, 5V to 3.3V regulator for digital IO, 5V to 3.5V regulator for analog codec)
- 3.3V digital I/O pads with 5V tolerance
- Industrial standard LQFP-128 package (14x14 mm)

## Optional Value-added Software Features:

# 3 Applications

- iDevice Docking
- Android Docking
- Notebook/Netbook Docking
- Audio Box
- USB DAC

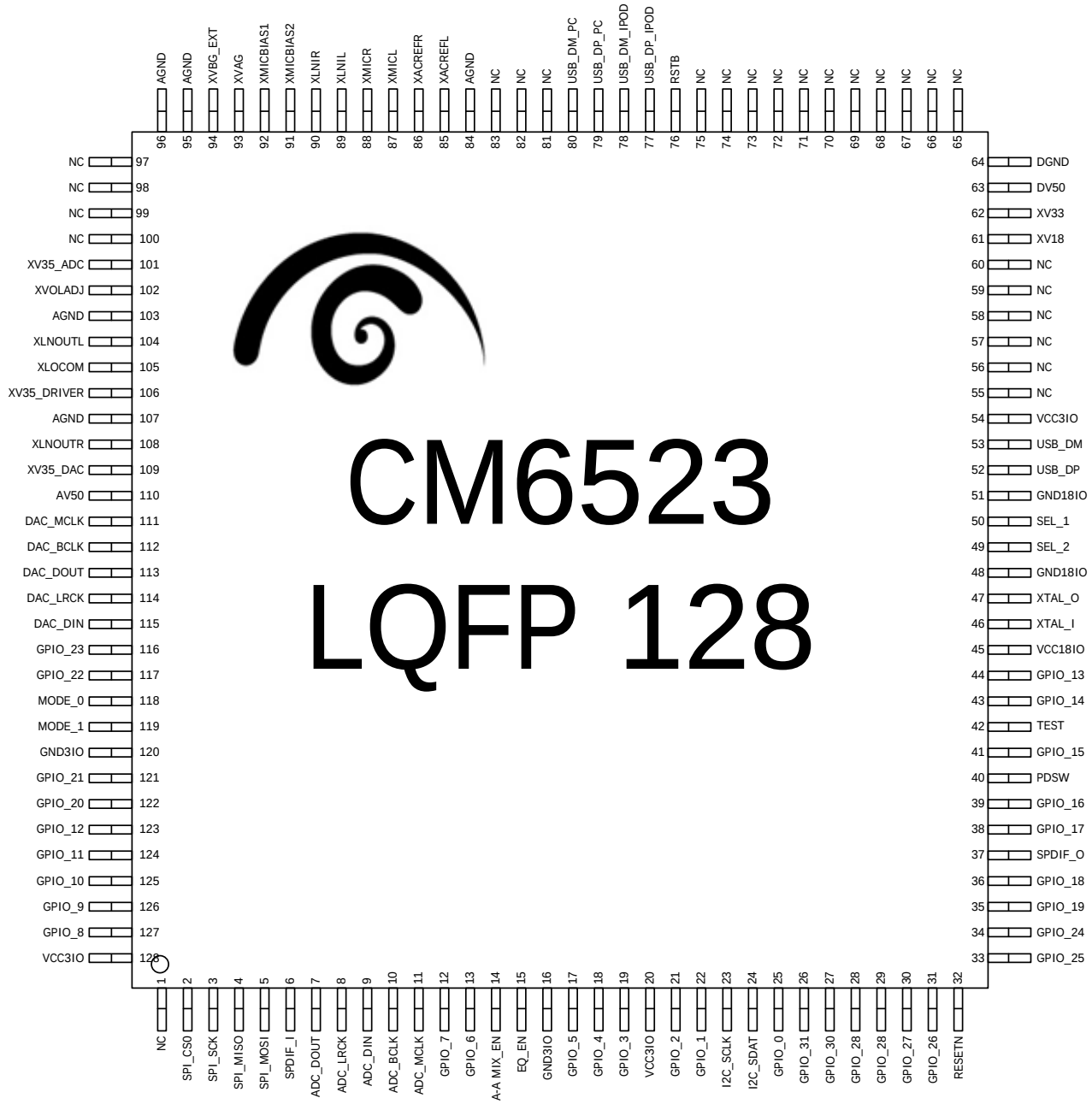
## 4 Block Diagram

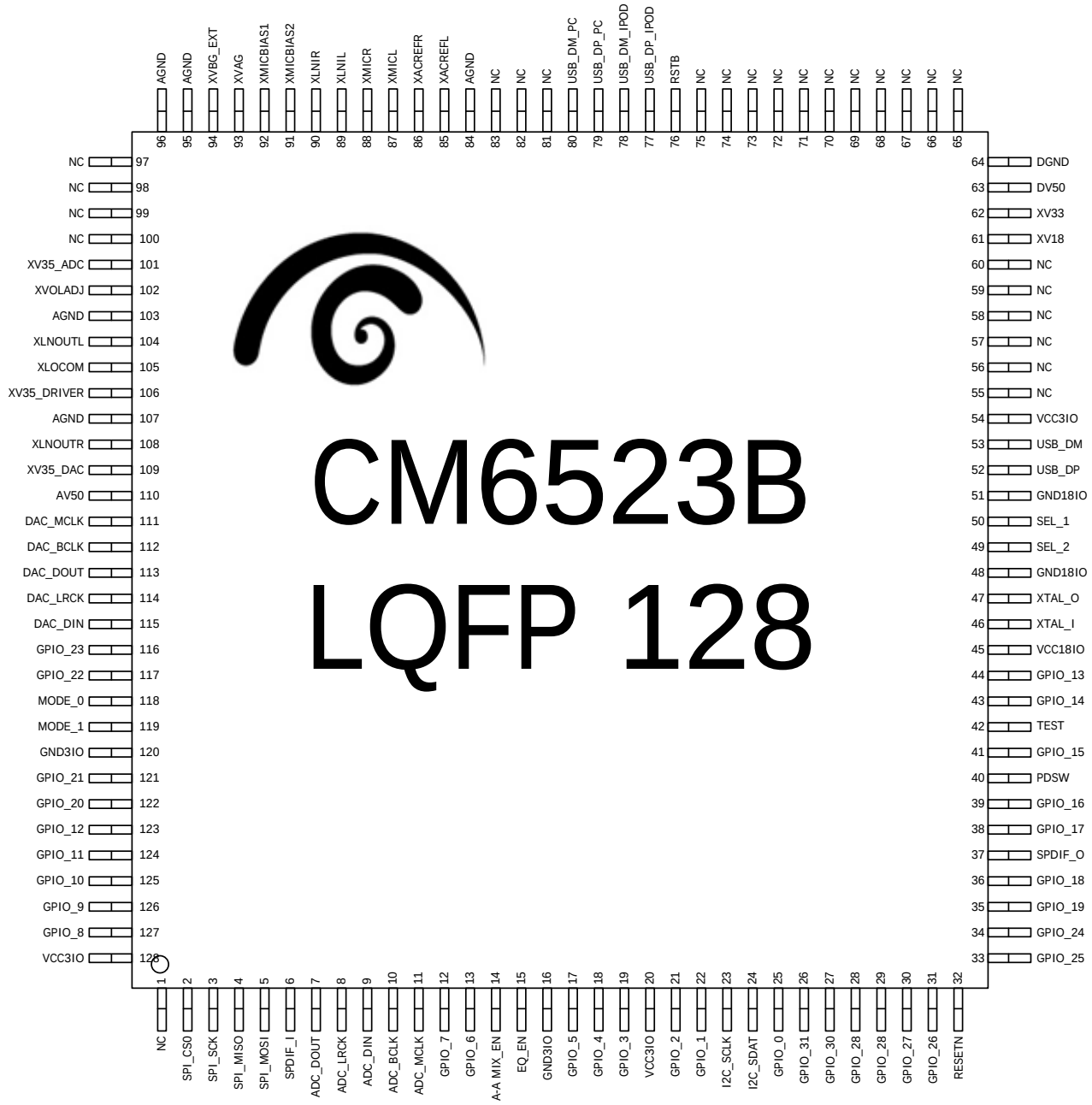



**CM6523 / CM6523B Functional Block Diagram**

# 5 Pin Assignment

## 5.1 Pin-Out Diagram





## 5.2 Pin Description

| Pin # | Symbol | I/O | Description |
|---|---|---|---|
| **Clock** | | | |
| 47 | XTAL_O | AO | 12MHz crystal oscillator output |
| 46 | XTAL_I | AI | 12MHz crystal oscillator input |
| **USB2.0 BUS Interface** | | | |
| 53 | USB_DM | AIO | USB 2.0 data negative (USB D- signal) |
| 52 | USB_DP | AIO | USB 2.0 data positive (USB D+ signal) |
| 78 | USB_DM_IPOD | AIO | USB 2.0 data negative (USB D- signal) |
| 77 | USB_DP_IPOD | AIO | USB 2.0 data positive (USB D+ signal) |
| 80 | USB_DM_PC | AIO | USB 2.0 data negative (USB D- signal) |
| 79 | USB_DP_PC | AIO | USB 2.0 data positive (USB D+ signal) |
| **Power/Ground** | | | |
| 63 | DV50 | PWR | 5V digital power for 5/3.3 regulator |
| 45 | VCC18IO | AO | 1.8V power for digital I/O |
| 54 | VCC3IO | PWR | 3.3V power for digital I/O |
| 62 | XV33 | AO | Regulator 3.3V output, drive capacity 150mA for USB and digital I/O |
| 48 | GND18IO | GND | Digital Ground |
| 128 | VCC3IO | PWR | 3.3V power for digital I/O |
| 20 | VCC3IO | PWR | 3.3V power for digital I/O |
| 51 | GND18 | GND | Digital Ground |
| 61 | XV18 | AO | Regulator 1.8V output, drive capacity 100mA for digital core |
| 120 | GND3IO | GND | Digital Ground |
| 16 | GND3IO | GND | Digital Ground |
| 64 | DGND | GND | Digital Ground |
| 110 | AV50 | PWR | 5V analog power for 5/3.5 regulator |
| 84 | AGND | GND | Analog Ground |
| 109 | XV35_DAC | AO | Regulator 3.5V output, drive capacity 100mA for analog and amplifier |
| 101 | XV35_ADC | AO | 3.5V power for ADC and Voltage and Current Reference |
| 95 | AGND | GND | Analog Ground |
| 96 | AGND | GND | Analog Ground |
| 103 | AGND | GND | Analog Ground |
| 106 | XV35_DRIVER | AO | 3.5V power for Driver |
| 107 | AGND | GND | Analog Ground |

| | | | |
|---|---|---|---|
| **Audio Interface** | | | |
| 85 | XACREFL | AO | common reference voltage for input signal |
| 86 | XACREFR | AO | common reference voltage for input signal |
| 87 | XMICL | AI | Mic in left channel |
| 88 | XMICR | AI | Mic in right channel |
| 89 | XLINL | AI | Line in left channel |
| 90 | XLINR | AI | Line in right channel |
| 91 | XMICBIAS2 | AO | Microphone bias |
| 92 | XMICBIAS1 | AO | Microphone bias |
| 93 | XVAG | AO | Voltage reference cap filter |
| 94 | XVBG_EXT | AI | External bandgap reference voltage input(level:1.24V) |
| 102 | XVOLADJ | AI | Analog control voltage input for playback volume control |
| 104 | XLNOUTL | AO | Line out left channel |
| 105 | XLOCOM | AO | Line out common reference for cap-less connection |
| 108 | XLNOUTR | AO | Line out right channel |
| **2-channel I2S DAC Interface** | | | |
| 111 | DAC_MCLK | DO | I2S master clock<br>Programmable 3.3V output buffer |
| 112 | DAC_BCLK | DIO | I2S bit clock<br>Programmable 3.3V bidirectional buffer, pull-down |
| 113 | DAC_DOUT | DO | I2S serial data output for channel 0, 1<br>Programmable 3.3V output buffer |
| 114 | DAC_LRCK | DIO | I2S left/right clock<br>Programmable 3.3V bidirectional buffer, pull-down |
| 115 | DAC_DIN | DI | Input from DSP to DAC for Playback |
| **2-channel I2S ADC interface** | | | |
| 7 | ADC_ DOUT | DO | Output from ADC to DSP for data processing |
| 8 | ADC_ LRCK | DIO | I2S left/right clock<br>Programmable 3.3V bidirectional buffer, pull-down |
| 9 | ADC_ DIN | DI | I2S serial data input for channel 0, 1<br>Programmable 3.3V input buffer, Schmitt trigger, pull-down |
| 10 | ADC_ BCLK | DIO | I2S bit clock<br>Programmable 3.3V bidirectional buffer, pull-down |
| 11 | ADC_ MCLK | DO | I2S master clock<br>Programmable 3.3V output buffer |
| **S/PDIF I/O** | | | |
| 6 | SPDIF_I | DI | S/PDIF receiver<br>Programmable 3.3V output buffer |
| 37 | SPDIF_O | DO | S/PDIF transmitter<br>Programmable 3.3V output buffer |
| **GPIO** | | | |

| | | | | |
|---|---|---|---|---|
| 25 | GPIO_0 | DIO | General purpose input/output (default Volume Up).<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-up | |
| 22 | GPIO_1 | DIO | General purpose input/output (default Volume Down).<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-up | |
| 21 | GPIO_2 | DIO | General purpose input/output (default Play Mute).<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-up | |
| 19 | GPIO_3 | DIO | General purpose input/output (default Rec Mute).<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-up | |
| 18 | GPIO_4 | DIO | General purpose input/output (default LED Live, 2K Hz).<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down | |
| 17 | GPIO_5 | DIO | General purpose input/output (default LED Play Mute).<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down | |
| 13 | GPIO_6 | DIO | General purpose input/output (default LED Rec Mute, 1K Hz).<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down | |
| 12 | GPIO_7 | DIO | General purpose input/output (default EQ Mode Select0).<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down | GPIO[8:7]=0,0: Normal<br>GPIO[8:7]=1,0: Communication<br>GPIO[8:7]=0,1: Gaming<br>GPIO[8:7]=1,1: Movie |
| 127 | GPIO_8 | DIO | General purpose input/output (default EQ Mode Selet1).<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down | |
| 126 | GPIO_9 | DIO | General purpose input/output (default Rec Clip Indicator).<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down | |
| 125 | GPIO_10 | DIO | General purpose input/output (default Wave Volume Up).<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down | |
| 124 | GPIO_11 | DIO | General purpose input/output (default Wave Volume Down).<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down | |
| 123 | GPIO_12 | DIO | General purpose input/output (default Play/Pause).<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-up | |
| 44 | GPIO_13 | DIO | General purpose input/output (default Stop).<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-up | |
| 43 | GPIO_14 | DIO | General purpose input/output (default Next).<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-up | |
| 41 | GPIO_15 | DIO | General purpose input/output (default Previous).<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-up | |
| 39 | GPIO_16 | DIO | General purpose input/output (default MCU_RXD).<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down | |
| 38 | GPIO_17 | DIO | General purpose input/output (default MCU_TRX).<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down | |
| 36 | GPIO_18 | DIO | General purpose input/output (default IR Module).<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down | |
| 35 | GPIO_19 | DIO | General purpose input/output<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down | |
| 122 | GPIO_20 | DIO | General purpose input/output<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down | |
| 121 | GPIO_21 | DIO | General purpose input/output<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down | |
| 117 | GPIO_22 | DIO | General purpose input/output<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down | |
| 116 | GPIO_23 | DIO | General purpose input/output<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down | |
| 34 | GPIO_24 | DIO | General purpose input/output<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down | |
| 33 | GPIO_25 | DIO | General purpose input/output<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down | |
| 31 | GPIO_26 | DIO | General purpose input/output<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down | |
| 30 | GPIO_27 | DIO | General purpose input/output<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down | |
| 29 | GPIO_28 | DIO | General purpose input/output<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down | |

| | | | |
|---|---|---|---|
| 28 | GPIO_29 | DIO | General purpose input/output<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down |
| 27 | GPIO_30 | DIO | General purpose input/output<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down |
| 26 | GPIO_31 | DIO | General purpose input/output<br>Programmable 3.3V/5V tolerance bidirectional buffer, pull-down |
| **2-Wire Master Serial Bus (I2C)** | | | |
| 24 | I2C_SDAT | DIO | 2-wire master serial data<br>Programmable 3.3V/5V tolerant bidirectional buffer, pull-down |
| 23 | I2C_SCLK | DIO | 2-wire master serial clock<br>Programmable 3.3V/5V tolerant bidirectional buffer, pull-down |
| **4-Wire SPI Serial Bus** | | | |
| 2 | SPI_CS0 | DO | Chip Select |
| 3 | SPI_SCK | DO | Serial Clock |
| 4 | SPI_MISO | DO | Serial Data Out |
| 5 | SPI_MOSI | DI | Serial Data In |
| **Miscellaneous** | | | |
| 76 | RSTB | | Power on reset |
| 32 | RESETN | | Reset pin |
| 40 | PDSW | DO | Power Down Switch<br>Normal: 0<br>Suspend: 1 |
| 42 | TEST | DI | For test |
| 14 | A-A MIX_EN | DI | 0: A-A path disable<br>1: A-A path enable |
| 15 | EQ_EN | DI | 0: Disable EQ/ 1: Enable EQ |
| 49 | SEL2 | DI | HW Select for different PID |
| 50 | SEL1 | DI | HW Select for different PID |
| 118 | MODE_0 | DI | MODE_1= 0, MODE_0= 0 for Headset<br>MODE_1= 0, MODE_0= 1 for Microphone<br>MODE_1= 1, MODE_0= 0 for Speaker<br>MODE_1= 1, MODE_0= 1 for Docking |
| 119 | MODE_1 | DI | |

# 6 USB Audio Topology

CM6523 / CM6523B supports 4 types of topology by default. They are Headset, Docking, Speaker, Mic. Different topology can be selected by pin Mode_0 and Mode_1. The combinations are as below.

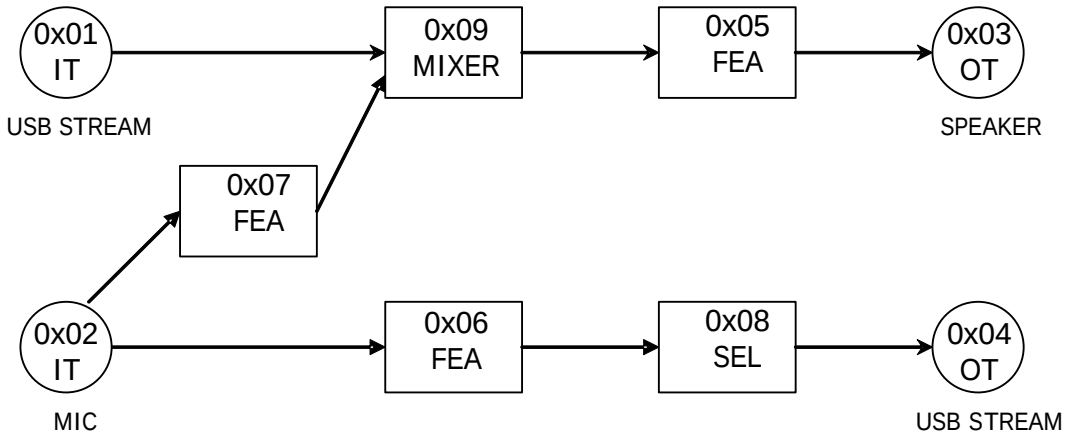
MODE_1= 0, MODE_0= 0 for Headset

MODE_1= 0, MODE_0= 1 for Microphone

MODE_1= 1, MODE_0= 0 for Speaker

MODE_1= 1, MODE_0= 1 for Docking

## 6.1 Headset Topology




Device Descriptor

| Offset | Field | Size | Value (Hex) | Description |
|---|---|---|---|---|
| 0 | bLength | 1 | 12 | Descriptor length |
| 1 | bDescriptorType | 1 | 01 | Device Descriptor |
| 2 | bcdUSB | 2 | 0110 | USB 1.1 compliant |
| 4 | bDeviceClass | 1 | 00 | Device class specified by interface |
| 5 | bDeviceSubClass | 1 | 00 | Device subclass specified by interface |
| 6 | bDeviceProtocol | 1 | 00 | Device protocol specified by interface |
| 7 | bMaxPacketSize0 | 1 | 10 | Endpoint zero packet size |
| 8 | idVendor | 2 | 0D8C | Vendor ID |
| 10 | idProduct | 2 | 0178~017F | Product ID |
| 12 | bcdDevice | 2 | 0000 | Device release number |
| 14 | iManufacturer | 1 | 01 | Index of string descriptor describing manufacturer |
| 15 | iProduct | 1 | 02 | Index of string descriptor describing product |
| 16 | iSerialNumber | 1 | 00 | Index of string descriptor describing serial number |
| 17 | bNumConfigurations | 1 | 01 | Number of configuration |

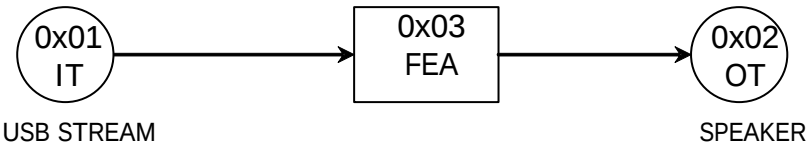
Configuration Descriptor

| Offset | Field | Size | Value (Hex) | Description |
|---|---|---|---|---|
| 0 | bLength | 1 | 09 | Descriptor length |
| 1 | bDescriptorType | 1 | 02 | Configuration Descriptor |
| 2 | wTotalLength | 2 | 011D | Total length of data returned for this configuration:<br>285 Bytes |
| 4 | bNumInterfaces | 1 | 04 | Number of interfaces supported by this Configuration:<br>00: Control<br>01: ISO-Out<br>02: ISO-In<br>03: INT-In (HID) |
| 5 | bConfigurationValue | 1 | 01 | Configuration value |
| 6 | iConfiguration | 1 | 00 | Index of string descriptor describing this configuration |
| 7 | bmAttributes | 1 | 80 | Attributes(Bus Powered) |
| 8 | bMaxPower | 1 | 32 | Maximum power consumption from bus = 100mA:<br>8'h32 (50x2 mA) (PWRSEL_2 = 1) |

Audio Control Interface 0 Descriptor 0

| Offset | Field | Size | Value (Hex) | Description |
|---|---|---|---|---|
| 0 | bLength | 1 | 09 | Descriptor length |
| 1 | bDescriptorType | 1 | 04 | Interface Descriptor |
| 2 | bInterfaceNumber | 1 | 00 | Interface number |
| 3 | bAlternateSetting | 1 | 00 | Alternate interface |
| 4 | bNumEndpoints | 1 | 00 | Number of endpoint used by this interface |
| 5 | bInterfaceClass | 1 | 01 | Audio Interface Class |
| 6 | bInterfaceSubClass | 1 | 01 | Subclass code: AUDIO_CONTROL |
| 7 | bInterfaceProtocol | 1 | 00 | Protocol code |
| 8 | iInterface | 1 | 00 | Index of string descriptor describing this interface |

## 6.2 Speaker Topology




Device Descriptor

| Offset | Field | Size | Value (Hex) | Description |
|---|---|---|---|---|
| 0 | bLength | 1 | 12 | Descriptor length |
| 1 | bDescriptorType | 1 | 01 | Device Descriptor |
| 2 | bcdUSB | 2 | 0110 | USB 1.1 compliant |
| 4 | bDeviceClass | 1 | 00 | Device class specified by interface |
| 5 | bDeviceSubClass | 1 | 00 | Device subclass specified by interface |
| 6 | bDeviceProtocol | 1 | 00 | Device protocol specified by interface |
| 7 | bMaxPacketSize0 | 1 | 10 | Endpoint zero packet size |
| 8 | idVendor | 2 | 0D8C | Vendor ID |

| | | | | |
|---|---|---|---|---|
| 10 | idProduct | 2 | 0180~018F | Product ID |
| 12 | bcdDevice | 2 | 0000 | Device release number |
| 14 | iManufacturer | 1 | 01 | Index of string descriptor describing manufacturer |
| 15 | iProduct | 1 | 02 | Index of string descriptor describing product |
| 16 | iSerialNumber | 1 | 00 | Index of string descriptor describing serial number |
| 17 | bNumConfigurations | 1 | 01 | Number of configuration |

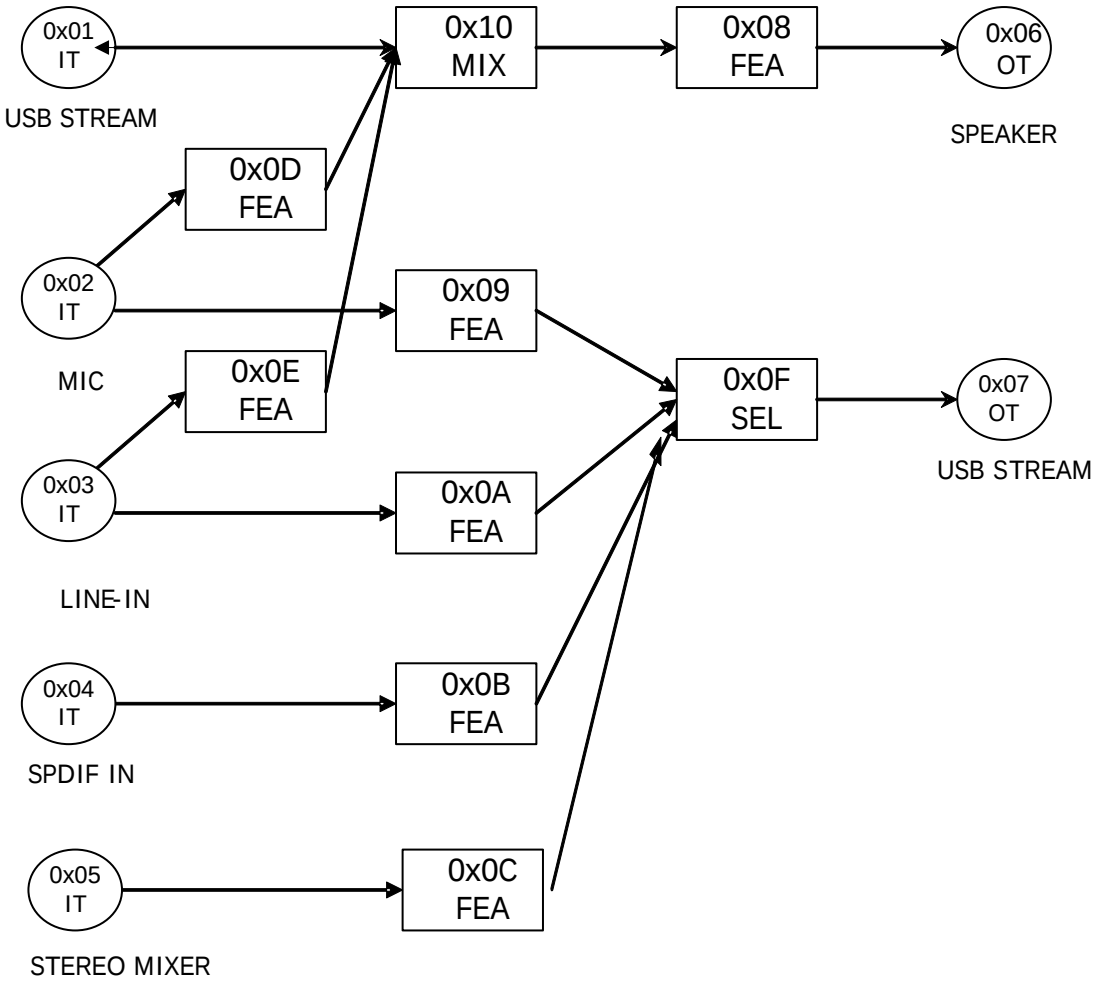
Configuration Descriptor

| Offset | Field | Size | Value (Hex) | Description |
|---|---|---|---|---|
| 0 | bLength | 1 | 09 | Descriptor length |
| 1 | bDescriptorType | 1 | 02 | Configuration Descriptor |
| 2 | wTotalLength | 2 | 0099 | Total length of data returned for this configuration:<br>153 Bytes |
| 4 | bNumInterfaces | 1 | 03 | Number of interfaces supported by this Configuration:<br>00: Control<br>01: ISO-Out<br>02: INT-In (HID) |
| 5 | bConfigurationValue | 1 | 01 | Configuration value |
| 6 | iConfiguration | 1 | 00 | Index of string descriptor describing this configuration |
| 7 | bmAttributes | 1 | 80 | Attributes(Bus Powered) |
| 8 | bMaxPower | 1 | 32 | Maximum power consumption from bus = 100mA:<br>8'h32 (50x2 mA) (PWRSEL_2 = 1) |

Audio Control Interface 0 Descriptor 0

| Offset | Field | Size | Value (Hex) | Description |
|---|---|---|---|---|
| 0 | bLength | 1 | 09 | Descriptor length |
| 1 | bDescriptorType | 1 | 04 | Interface Descriptor |
| 2 | bInterfaceNumber | 1 | 00 | Interface number |
| 3 | bAlternateSetting | 1 | 00 | Alternate interface |
| 4 | bNumEndpoints | 1 | 00 | Number of endpoint used by this interface |
| 5 | bInterfaceClass | 1 | 01 | Audio Interface Class |
| 6 | bInterfaceSubClass | 1 | 01 | Subclass code: AUDIO_CONTROL |
| 7 | bInterfaceProtocol | 1 | 00 | Protocol code |
| 8 | iInterface | 1 | 00 | Index of string descriptor describing this interface |

## 6.3 Docking Topology




Device Descriptor

| Offset | Field | Size | Value (Hex) | Description |
|---|---|---|---|---|
| 0 | bLength | 1 | 12 | Descriptor length |
| 1 | bDescriptorType | 1 | 01 | Device Descriptor |
| 2 | bcdUSB | 2 | 0110 | USB 1.1 compliant |
| 4 | bDeviceClass | 1 | 00 | Device class specified by interface |
| 5 | bDeviceSubClass | 1 | 00 | Device subclass specified by interface |
| 6 | bDeviceProtocol | 1 | 00 | Device protocol specified by interface |
| 7 | bMaxPacketSize0 | 1 | 10 | Endpoint zero packet size |
| 8 | idVendor | 2 | 0D8C | Vendor ID |
| 10 | idProduct | 2 | 01A8~01AF | Product ID |
| 12 | bcdDevice | 2 | 0000 | Device release number |
| 14 | iManufacturer | 1 | 01 | Index of string descriptor describing manufacturer |
| 15 | iProduct | 1 | 02 | Index of string descriptor describing product |
| 16 | iSerialNumber | 1 | 00 | Index of string descriptor describing serial number |

| | | | | |
|---|---|---|---|---|
| 17 | bNumConfigurations | 1 | 01 | Number of configuration |

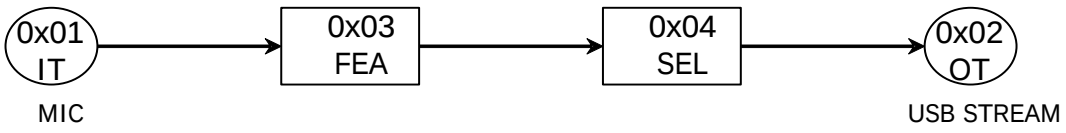
Configuration Descriptor

| Offset | Field | Size | Value (Hex) | Description |
|---|---|---|---|---|
| 0 | bLength | 1 | 09 | Descriptor length |
| 1 | bDescriptorType | 1 | 02 | Configuration Descriptor |
| 2 | wTotalLength | 2 | 016D | Total length of data returned for this configuration:<br>365 Bytes |
| 4 | bNumInterfaces | 1 | 04 | Number of interfaces supported by this Configuration:<br>00: Control<br>01: ISO-Out<br>02: ISO-In<br>03: HID |
| 5 | bConfigurationValue | 1 | 01 | Configuration value |
| 6 | iConfiguration | 1 | 00 | Index of string descriptor describing this configuration |
| 7 | bmAttributes | 1 | 80 | Attributes(Bus Powered) |
| 8 | bMaxPower | 1 | 32 | Maximum power consumption from bus = 100mA:<br>8'h32 (50x2 mA) (PWRSEL_2 = 1) |

Audio Control Interface 0 Descriptor 0

| Offset | Field | Size | Value (Hex) | Description |
|---|---|---|---|---|
| 0 | bLength | 1 | 09 | Descriptor length |
| 1 | bDescriptorType | 1 | 04 | Interface Descriptor |
| 2 | bInterfaceNumber | 1 | 00 | Interface number |
| 3 | bAlternateSetting | 1 | 00 | Alternate interface |
| 4 | bNumEndpoints | 1 | 00 | Number of endpoint used by this interface |
| 5 | bInterfaceClass | 1 | 01 | Audio Interface Class |
| 6 | bInterfaceSubClass | 1 | 01 | Subclass code: AUDIO_CONTROL |
| 7 | bInterfaceProtocol | 1 | 00 | Protocol code |
| 8 | iInterface | 1 | 00 | Index of string descriptor describing this interface |

## 6.4 Microphone (Stereo) Topology




Device Descriptor

| Offset | Field | Size | Value (Hex) | Description |
|---|---|---|---|---|
| 0 | bLength | 1 | 12 | Descriptor length |
| 1 | bDescriptorType | 1 | 01 | Device Descriptor |
| 2 | bcdUSB | 2 | 0110 | USB 1.1 compliant |
| 4 | bDeviceClass | 1 | 00 | Device class specified by interface |
| 5 | bDeviceSubClass | 1 | 00 | Device subclass specified by interface |
| 6 | bDeviceProtocol | 1 | 00 | Device protocol specified by interface |
| 7 | bMaxPacketSize0 | 1 | 10 | Endpoint zero packet size |
| 8 | idVendor | 2 | 0D8C | Vendor ID |
| 10 | idProduct | 2 | 0190~019F | Product ID |

| | | | | |
|---|---|---|---|---|
| 12 | bcdDevice | 2 | 0000 | Device release number |
| 14 | iManufacturer | 1 | 01 | Index of string descriptor describing manufacturer |
| 15 | iProduct | 1 | 02 | Index of string descriptor describing product |
| 16 | iSerialNumber | 1 | 00 | Index of string descriptor describing serial number |
| 17 | bNumConfigurations | 1 | 01 | Number of configuration |

Configuration Descriptor

| Offset | Field | Size | Value (Hex) | Description |
|---|---|---|---|---|
| 0 | bLength | 1 | 09 | Descriptor length |
| 1 | bDescriptorType | 1 | 02 | Configuration Descriptor |
| 2 | wTotalLength | 2 | 00A0 | Total length of data returned for this configuration:<br>160 Bytes |
| 4 | bNumInterfaces | 1 | 03 | Number of interfaces supported by this Configuration:<br>00: Control<br>01: ISO-In<br>03: HID |
| 5 | bConfigurationValue | 1 | 01 | Configuration value |
| 6 | iConfiguration | 1 | 00 | Index of string descriptor describing this configuration |
| 7 | bmAttributes | 1 | 80 | Attributes(Bus Powered) |
| 8 | bMaxPower | 1 | 32 | Maximum power consumption from bus = 100mA:<br>8'h32 (50x2 mA) (PWRSEL_2 = 1) |

Audio Control Interface 0 Descriptor 0

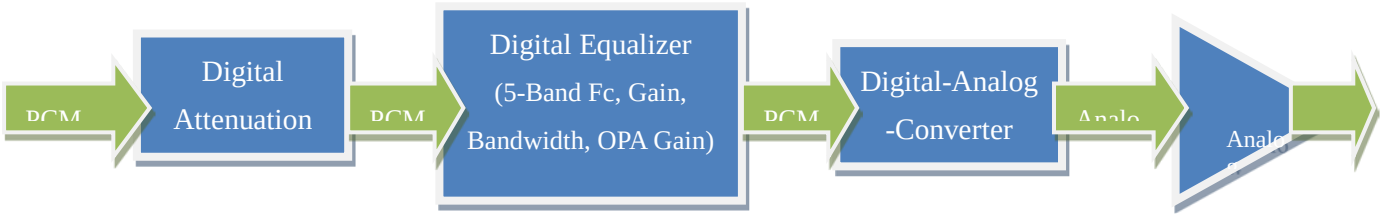
| Offset | Field | Size | Value (Hex) | Description |
|---|---|---|---|---|
| 0 | bLength | 1 | 09 | Descriptor length |
| 1 | bDescriptorType | 1 | 04 | Interface Descriptor |
| 2 | bInterfaceNumber | 1 | 00 | Interface number |
| 3 | bAlternateSetting | 1 | 00 | Alternate interface |
| 4 | bNumEndpoints | 1 | 00 | Number of endpoint used by this interface |
| 5 | bInterfaceClass | 1 | 01 | Audio Interface Class |
| 6 | bInterfaceSubClass | 1 | 01 | Subclass code: AUDIO_CONTROL |
| 7 | bInterfaceProtocol | 1 | 00 | Protocol code |
| 8 | iInterface | 1 | 00 | Index of string descriptor describing this interface |

# 7 Function Description

## 7.1 Playback Equalizer

### 7.1.1 5-band Equalizer

CM6523 / CM6523B has integrated 5-band hardware digital equalizer (EQ) engine inside the chips to fulfill various application usages. It provides up-to-4 preset modes on customer's product design for different user scenarios including default/music, movies, Gaming and communication modes. Customers could also change the gain parameters for each of the preset application EQ mode via EEPROM coding. In addition, the EQ engine could also be utilized for compensating and fine-tuning the headphone driver for Sound Pressure Level (SPL) performance to a specific preference. In this case, customers could fully customize all EQ coefficients (center frequency, gain values, and bandwidth) to one optimized frequency response curve and setting in terms of the headphone driver and housing's acoustics characteristics, also via EEPROM programming.




The EQ engine contains 5 frequency bands (Fc) of digital filters to conduct transfer functions of the frequency response over the audio band. It allows maximum +-12dB digital gain (Gain) for each band with 0.5dB adjustment per step. Each filter will have its bandwidth (BW) factor between 0 and 1.0.

Fc: Center Frequency, F1~F5, 20<Fc<20K (Hz)
Gain: Digital Frequency Gain, -12dB <= Gain <=+12dB, 0.5dB/step
BW: Filter Bandwidth Factor, 0<BW<1
OPA Gain: Analog Gain Compensation setting for each equalizer mode

The EQ engine already provides 4 preset modes/settings based on the same preset F1~F5 center frequencies and OPA gain:

F1 (Bass)= 100Hz
F2 = 350Hz
**F3 = 1KHz**
**F4 = 3.5KHz**
**F5 (Treble) = 13KHz**

With the 4 preset EQ modes, customers could use EEPROM parameters to change the gain values for each band of the center frequency and hence customize the 4-preset EQ curves based on the preset center frequencies and bandwidth. Alternatively, customers could also skip the 4 preset modes and create a customized EQ curve by changing the center frequencies, gain values and even the bandwidth factors in EEPROM parameters to make the headphone sound better or meet some frequency requirements. However, in this case, the product will always use one optimized EQ setting and could not allow users to dynamically change among different preset modes. Customer could also consider reporting Treble/Bass feature unit by EEPROM to Windows UAA driver to allow end-users to adjust Bass (F1) and Treble (F5) by themselves. Therefore there are three usage/application scenarios as the summary table below:

3 EQ Usage/Application Scenarios

| No | Scenario | Gain Value | Center Frequency / Bandwidth Factor | Number of Modes | User Control Type |
|---|---|---|---|---|---|
| **1** | **4 Switchable Presets** | Configurable | Fixed | 4 | Hardware |
| **2** | **Full-Customized EQ** | Configurable | Configurable | 1 | N.A. |
| **3** | **Treble/Bass Feature Unit** | Configurable | Configurable | 1 | Software |

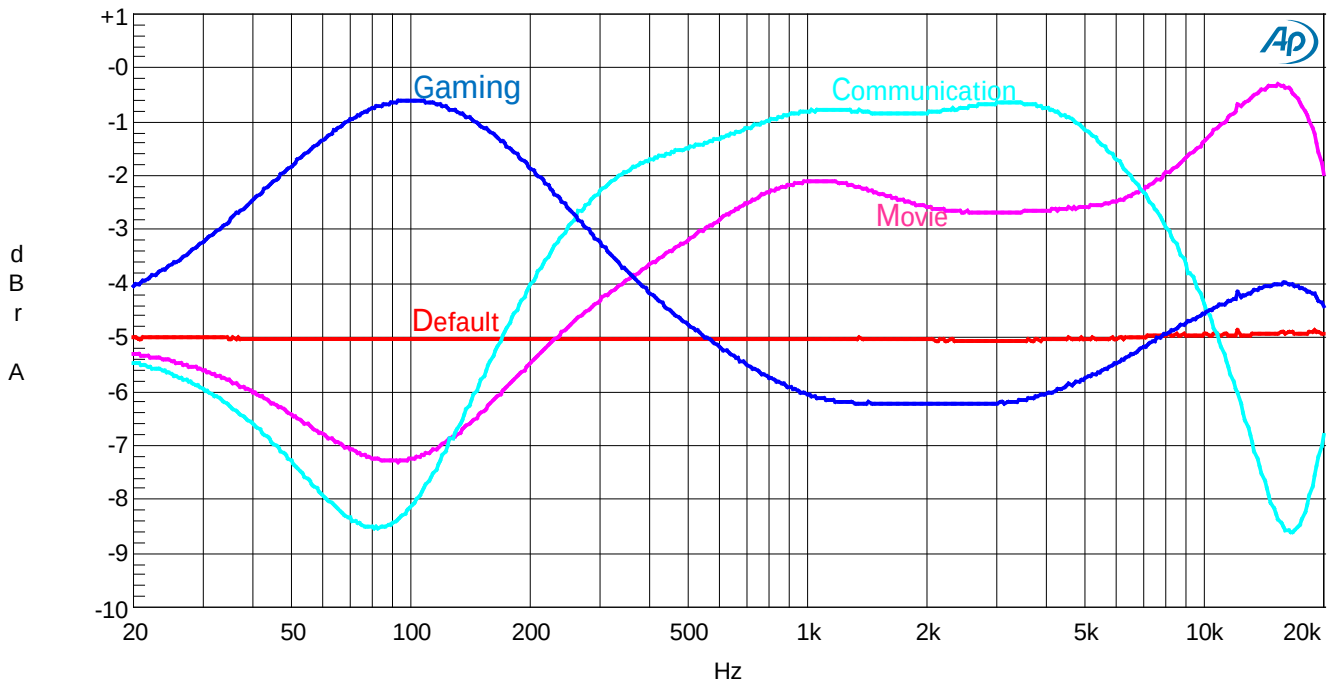
Note: Hardware user control type means end-users could select which EQ mode they're going to use by a hardware switch/button on the product; Software control means they could control the treble/bass gain values by GUI in Windows OS sound device advanced settings.

### 7.1.2 4 preset EQ mode

As mentioned above, EQ engine already provides 4 preset EQ modes for different user scenarios/applications. End users could use the hardware switch on the product (determined by 2 EQ configuration input pins) to dynamically change to different EQ modes. The following shows the frequency response of each mode.

| Mode | GPIO8 | GPIO7 | Color |
|---|---|---|---|
| Default | 0 | 0 | ---------- |
| Gaming | 0 | 1 | ---------- |
| Communication | 1 | 0 | ---------- |
| Movie | 1 | 1 | ---------- |

Audio Precision 04/20/11 15:35:35




| Sweep | Trace | Color | Line Style | Thick | Data | Axis | Comment |
|---|---|---|---|---|---|---|---|
| 1 | 1 | Red | Solid | 2 | Anlr.Ampl | Left | 00 |
| 2 | 1 | Magenta | Solid | 2 | Anlr.Ampl | Left | 11 |
| 3 | 1 | Cyan | Solid | 2 | Anlr.Ampl | Left | 10 |
| 4 | 1 | Blue | Solid | 2 | Anlr.Ampl | Left | |

DA-EQ-SPDIF_In_DA_Out.at27

## 7.2 Recording Equalizer

CM6523 / CM6523B also provide 5-band Equalizer for the input. It can be used to compensate the frequency response of Microphone unit. Customers could fully customize all EQ coefficients (center frequency, gain values, and bandwidth) through external EEPROM.

## 7.3 Recording AGC

Automatic Gain Control (AGC) is an automatically controlled method to adjust with intensity of signal; AGC is closes the return circuit; that is by the negative response system too.

AGC is by way of compressing volume, Will increase Gain first when AGC is started, Set up the upper and lower limits of the signal; compress the dynamic range of sound. Usually use the occasion of AGC, should be recording and producing and speaking sound, or volume is being changed under little environment. If the lasting low voice of volume, AGC will enlarge volume, volume is sustained loudly, AGC will reduce volume.

**FEATURES**

Programmable AGC Parameters

Selectable Gain from –12 dB to 45 dB in 1-dB Steps

Selectable Attack, Release and Hold Times

AGC Enable/Disable Function

Limiter Enable/Disable Function

Pre-Detect Limiter Level Function

Two-Channel AGC Independent

Under input source types, to set AGC gain max/min limit

| | | |
|---|---|---|
| I2S rec | +12~ -16DB | 0xf9= 0x1c (max)+fix gain(9db) = 0x25<br>0cfA= 0x00(min) |
| Digmic | +20 ~ -16DB | 0xf9= 0x24 (max)+fix gain(9db) =0x2d<br>0xfA= 0x00(min) |
| Analog mic | +30 ~ 0DB | 0xf9=0x0F+fix gain(9db) inv -> 0x39(max)<br>0xfA=0x2D inv -> 0x12(min) |

**AGC Variable Description**

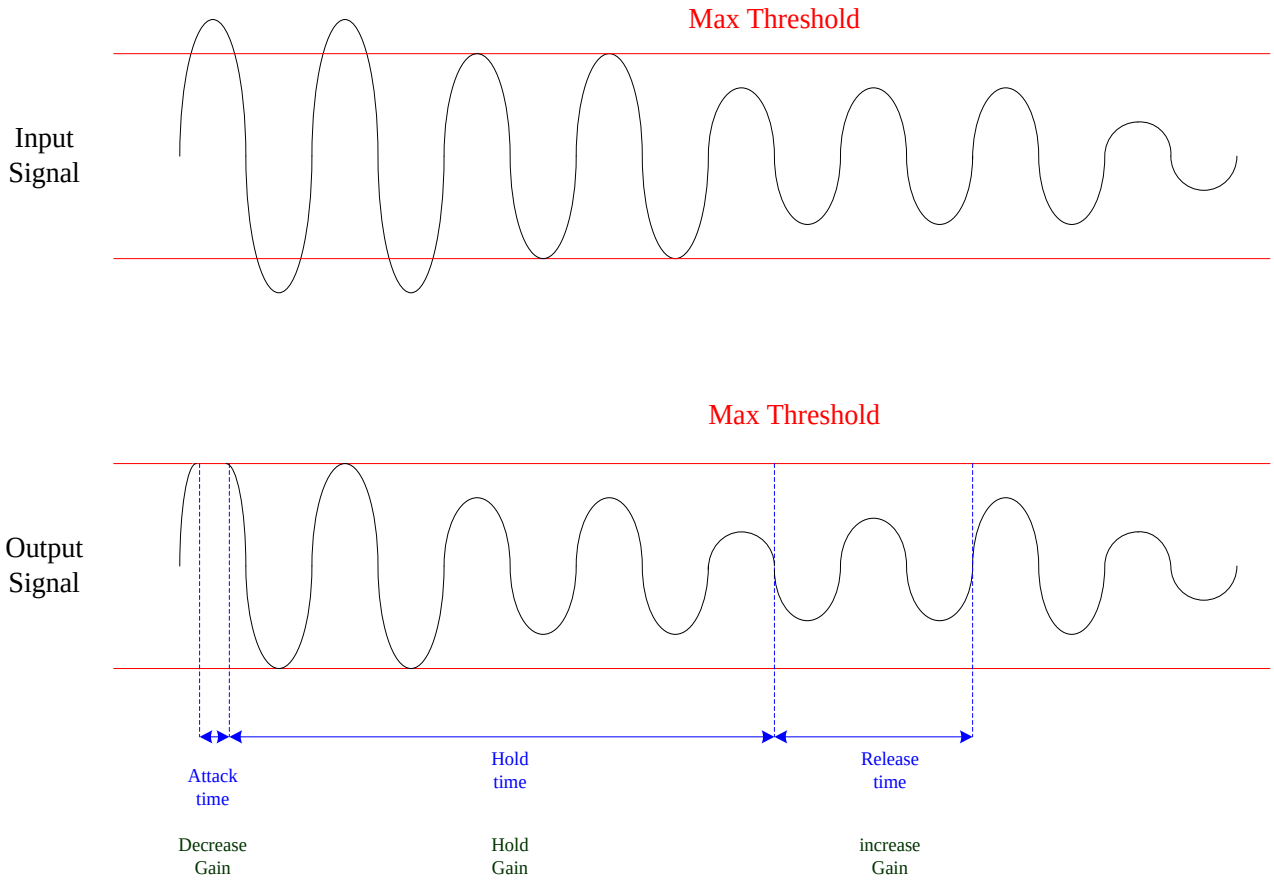
Fixed Gain: The normal gain of the device when the AGC is inactive.

Limiter Level: The value that sets the maximum allowed output amplitude.

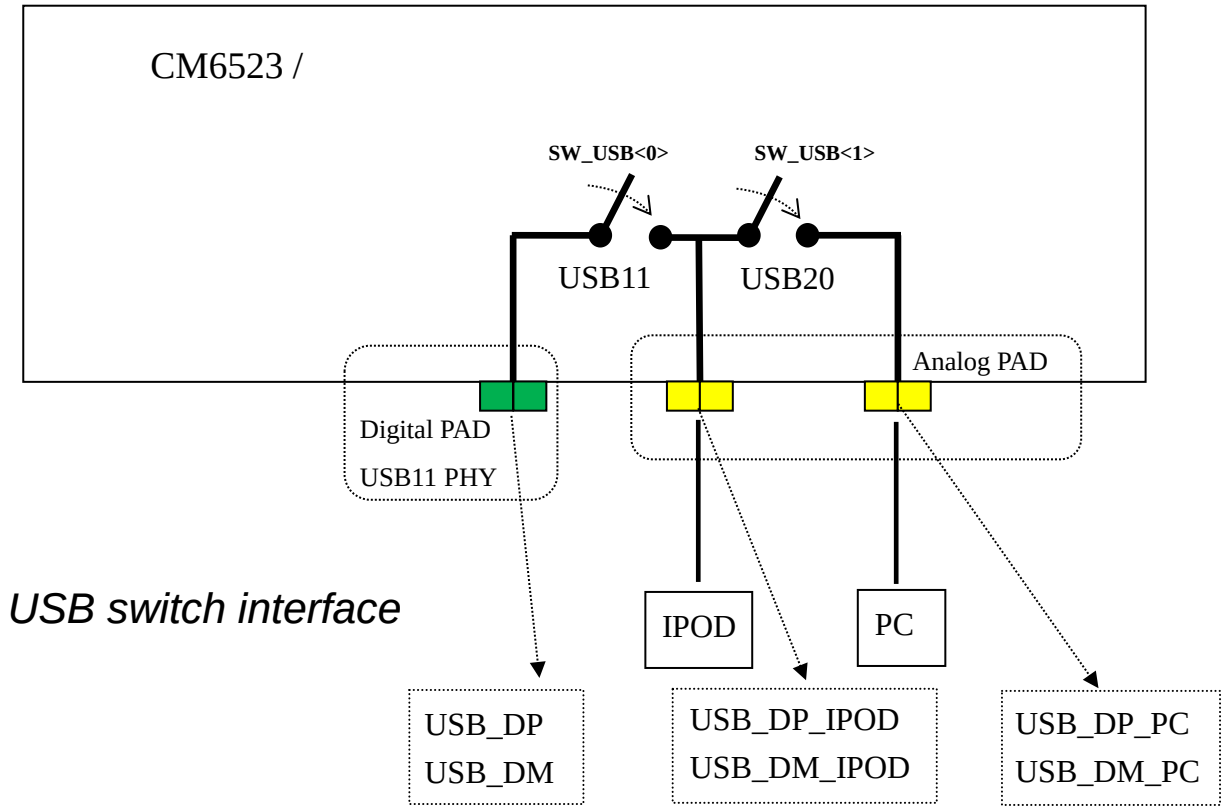
Attack Time: The minimum time between two gain decrements.

Release Time: The minimum time between two gain increments.

Hold Time: The time it takes for the very first gain increment after the input signal amplitude decreases.



## 7.4 USB 2.0 switch




| Signal NAME | BIT NUM | I/O | Function | Control Bits |
|---|---|---|---|---|
| SW_USB<1:0> | 2 | I | Control bits for USB switch:<br>USB20 Off, USB11 Off.<br>USB20 Off, USB11 On.<br>USB20 On, USB11 Off.<br>USB20 On, USB11 On. | <br>SW_USB<1,0>= (00)<br>SW_USB<1,0>= (01)<br>SW_USB<1,0>= (10)<br>SW_USB<1,0>= (11) |

## 7.5 HID Function

HID is **H**uman **I**nterface **D**evice, it's a type of computer device to interact with human for the input and output. The most common HID devices are the USB keyboard and Mouse. CM6523 / CM6523B also provide some basic HID buttons, such as Volume up, Volume down, Play back mute, etc. And also through set output report and get input report to communicate with external device.

### 7.5.1 HID interrupt in

Input Data Format:

| | |
|---|---|
| byte 0 | always 1 for org HID event report ID |
| byte1 | for defined HID event, and each event occupies one bit |
| byte2 | |
| byte3 | start address of returned data (H-start_addr) |
| byte4 | start address of returned data (L-start_addr) |
| byte5 | bit7 |
| | bit6:UART_INT |
| | bit5:GPI_INT |
| | bit4:SPIS_INT(slavemode int) |
| | bit3: SPIM_INT(mastermode int) |
| | bit2:I2CS_INT(slavemode int) |
| | bit1:I2CM_INT(mastermode int) |
| | bit0: IR_INT |
| byte6 | read data of [start_addr] |
| byte7 | read data of [start_addr+1] |
| byte8 | read data of [start_addr+2] |
| byte9 | read data of [start_addr+3] |
| byte10 | read data of [start_addr+4] |
| byte11 | read data of [start_addr+5] |
| byte12 | read data of [start_addr+6] |
| byte13 | read data of [start_addr+7] |
| byte14 | read data of [start_addr+8] |
| byte15 | read data of [start_addr+9] |

HID Get_Input_report

Command Format:

| bmRequestType | bRequest | wValue | wIndex | wLength | Data |
|---|---|---|---|---|---|
| 8'h A1 | 8'h 01 (Get_Report) | 16'h 01 01 (Rpt Type + Rpt ID) | 16'h 00 03 (Interface) | 16'h 00 10 (16 bytes) | Report |

*Note: The Start_Addr value in the input reported is put in the Internal Register Address 0xff.

Software must set the value of Start_Addr Register to make sure Get Input Report can read the proper data you want.

Input Data Format:

| | |
|---|---|
| byte 0 | always 1 for org HID event report ID |
| byte1 | for defined HID event, and each event occupies one bit |
| byte2 | |
| byte3 | start address of returned data (H-start_addr) |
| byte4 | start address of returned data (L-start_addr) |
| byte5 | bit7 |
| | bit6:UART_INT |
| | bit5:GPI_INT |
| | bit4:SPIS_INT(slavemode int) |
| | bit3: SPIM_INT(mastermode int) |
| | bit2:I2CS_INT(slavemode int) |
| | bit1:I2CM_INT(mastermode int) |
| | bit0: IR_INT |
| byte6 | read data of [start_addr] |
| byte7 | read data of [start_addr+1] |
| byte8 | read data of [start_addr+2] |
| byte9 | read data of [start_addr+3] |
| byte10 | read data of [start_addr+4] |
| byte11 | read data of [start_addr+5] |
| byte12 | read data of [start_addr+6] |
| byte13 | read data of [start_addr+7] |
| byte14 | read data of [start_addr+8] |
| byte15 | read data of [start_addr+9] |

### 7.5.2 HID Set_Output_report

Command Format:

| bmRequestType | bRequest | wValue | wIndex | wLength | Data |
|---|---|---|---|---|---|
| 8'h 21 | 8'h 09 (Set_Report) | 16'h 02 01 (Rpt Type + Rpt | 16'h 00 03 (Interface) | 16'h 00 10 (16 bytes) | Report |

| | | | | | |
|---|---|---|---|---|---|
| | | ID) | | | |

*Note: Byte5 is the beginning address of this write sequence.

Output Data Format:

| | |
|---|---|
| byte 0 | always 1 for org HID event report ID |
| byte1 | start address of write reg (H-start_addr) |
| byte2 | start address of write reg (L-start_addr) |
| byte3 | effective write/read data length (<=12) |
| byte4 | write data to [start_addr] |
| byte5 | write data to [start_addr+1] |
| byte6 | write data to [start_addr+2] |
| byte7 | write data to [start_addr+3] |
| byte8 | write data to [start_addr+4] |
| byte9 | write data to [start_addr+5] |
| byte10 | write data to [start_addr+6] |
| byte11 | write data to [start_addr+7] |
| byte12 | write data to [start_addr+8] |
| byte13 | write data to [start_addr+9] |
| byte14 | write data to [start_addr+10] |
| byte15 | write data to [start_addr+11] |

## 7.6 Vendor Command Definition

### 7.6.1 Vender Command Read

Command Format:

| bmRequestType | bRequest | wValue | wIndex | wLength | Data |
|---|---|---|---|---|---|
| 8'h C3 | 8'h 02 (Command 2) | 16'h -- -- (Start Address of input Data) | 16'h 00 00 | 16'h 00 – (<=64 bytes) | Data |

Input Data Format:

| | |
|---|---|
| **Byte 0** | Data of Reg[wValue] |
| **Byte 1** | Data of Reg[wValue + 1] |
| **Byte 2** | Data of Reg[wValue + 2] |
| **…** | … |
| **Byte 63** | Data of Reg[wValue + 63] |

### 7.6.2 Vender Command Write

Command Format:

| bmRequestType | bRequest | wValue | wIndex | wLength | Data |
|---|---|---|---|---|---|
| 8'h 43 | 8'h 01 (Command 1) | 16'h -- -- (Start Address of Output Data) | 16'h 00 00 | 16'h 00 – (<=64 bytes) | Data |

Output Data Format:

| | |
|---|---|
| **Byte 0** | Data of Reg[wValue] |
| **Byte 1** | Data of Reg[wValue + 1] |
| **Byte 2** | Data of Reg[wValue + 2] |
| **…** | … |
| **Byte 63** | Data of Reg[wValue + 63] |

## 7.7 I2S Control description

### 7.7.1 I2S Format description

**I2S Interface Setting**

$I^2S$ has three clock signals, MCLK, BCLK and LRCK, and at least one data line depending on the channels supported. One data line contains two channels. Therefore, there have four data lines for a 8-channel $I^2S$ DAC controller. The three $I^2S$ clock symbols are explained below.

MCLK = main clock.

BCLK = bit clock.

LRCK = left and right clock.

**Basic of I2S Bus**

Both master and slave modes of $I^2S$ are supported, namely $I^2S$ DAC, $I^2S$ ADC 1, $I^2S$ ADC 2, $I^2S$ ADC 3. Master mode means BCLK and LRCK are provided as shown in below left. On the contrary, slave mode means BCLK and

LRCK are provided by the I²S codecs as below right.

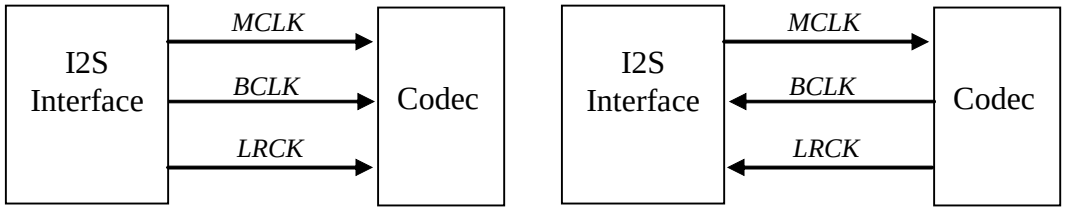



**Figure -1 I2S Master/Slave Block Diagram**

Below figure indicates the basic waveform of I²S. Note that BCLK is generated at the positive edges of MCLK with the ratios 1, 1/2, 1/4, or 1/8, and LRCK is generated at the negative edges of BCLK with the ratios 1/64, 1/128, 1/256. Data lines are transited at the negative edges of BCLK, and are sampled at the positive edges of BCLK by codecs in case of playback or recording.

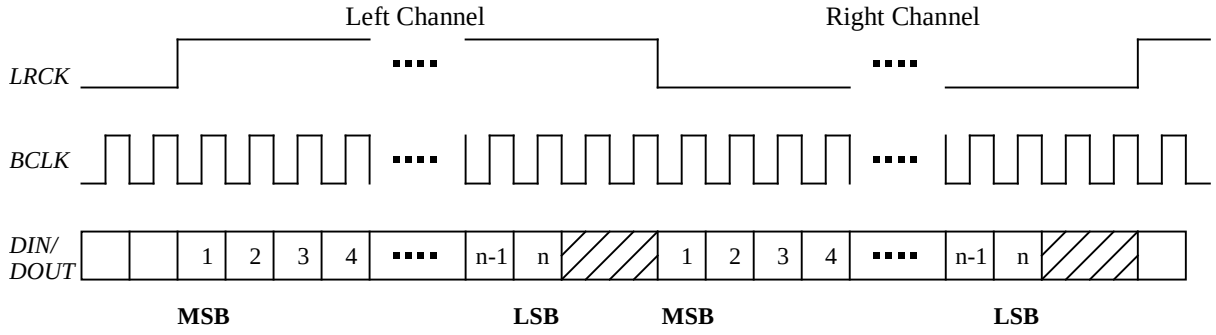



**Figure -2 I2S timing diagram**

For the I²S DAC controller, the audio data is transformed from the parallel format to the serial format before transmitted. Then, the bit data is shifted out one by one with the MSB first via DOUT signal. If the I²S DAC controller is set to 32 bits, at least 32 BCLK clocks must exist in both LRCK left and right channels. In the same manner, the audio data is transformed from the coming serial format to the parallel format for a I²S ADC controller.

### Left Justified Mode

In the left justified mode of the I²S DAC controller, the MSB data bit is clocked out at the negative edge of BCLK which is aligned to the transition of LRCK. In the left justified mode of I²S ADC controllers, the MSB data bit is clocked out by codecs and sampled at the first positive edge of BCLK which follows a LRCK transition. LRCK is high during left channel transmission and low during right channel transmission in the left justified mode.

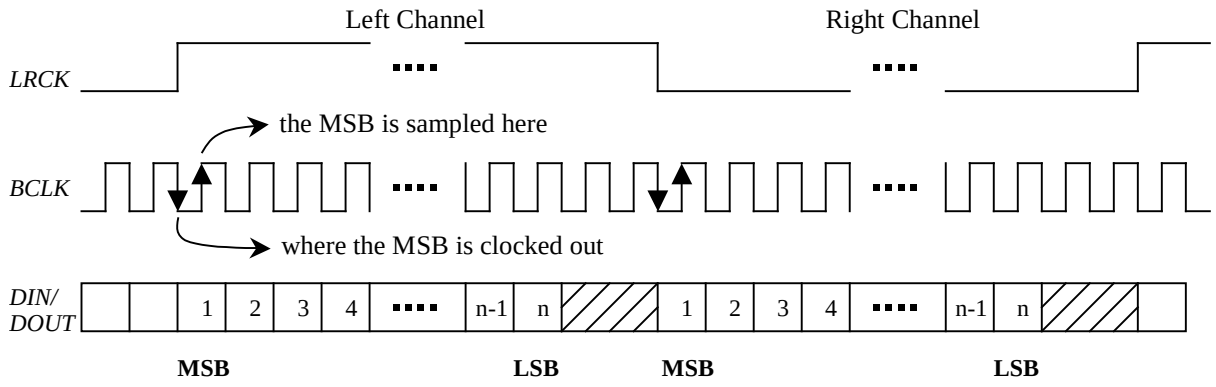



**Figure -3 Left Justified mode timing diagram of I2S**

### I2S Mode

In the I$^2$S mode of the I$^2$S DAC controller, the MSB data bit is clocked out by CMI8788 at the first negative edge of BCLK which follows a LRCK transition. In the I$^2$S mode of I$^2$S ADC controllers, the MSB data bit is clocked out by codecs and sampled at the second positive edge of BCLK which follows a LRCK transition. LRCK is low during left channel transmission and high during right channel transmission in the I$^2$S mode.

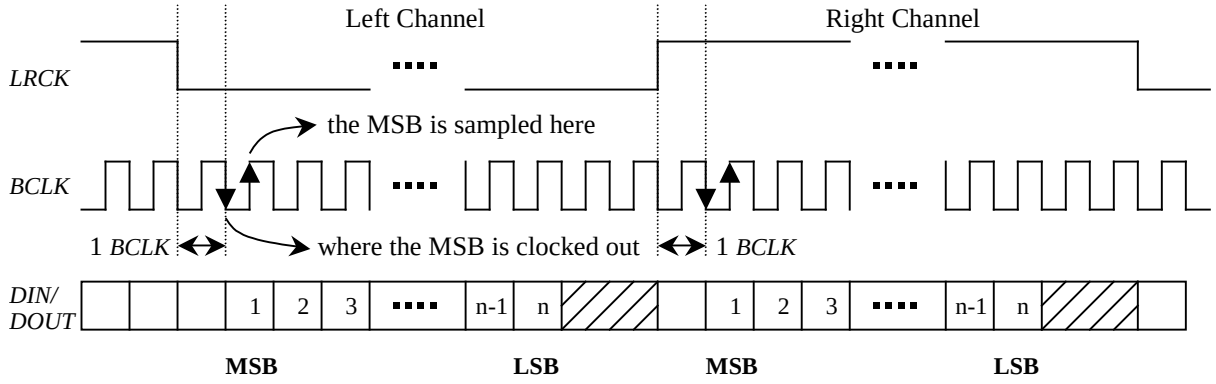
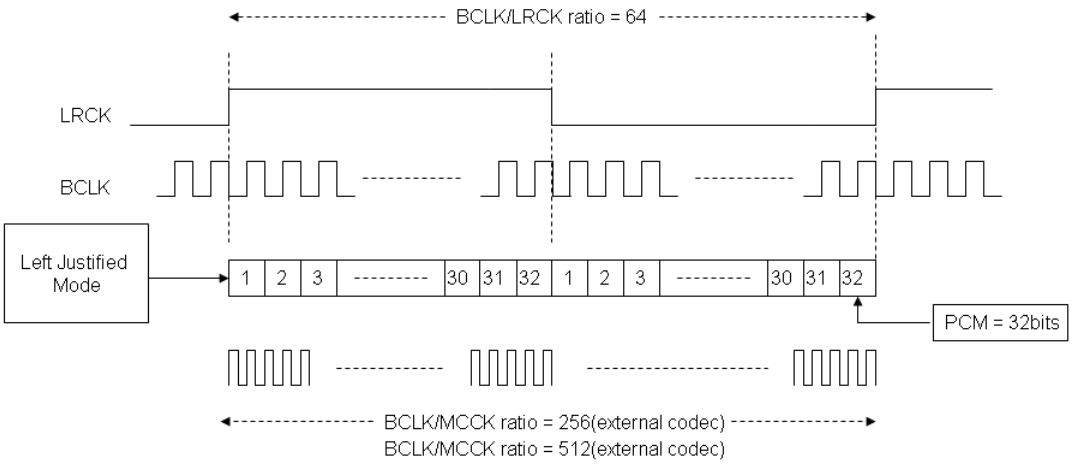


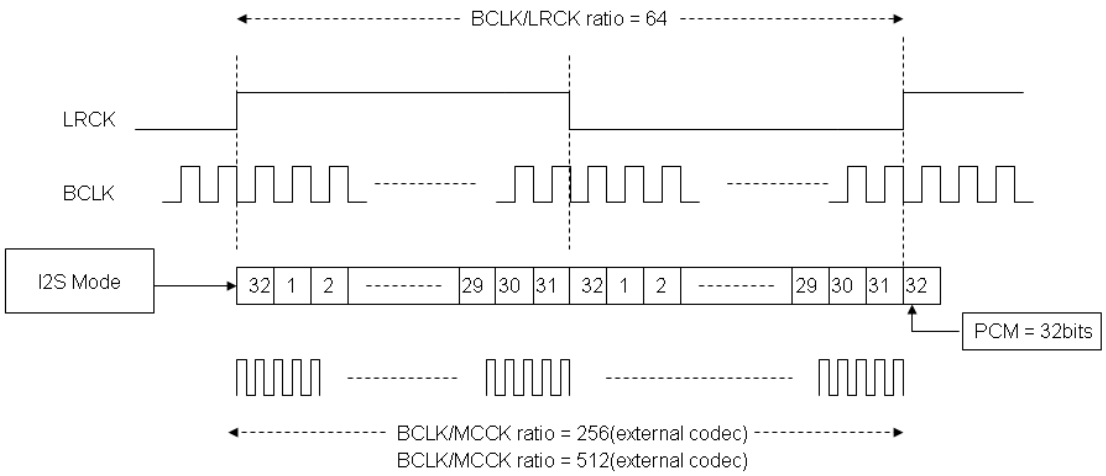

**Figure -4 I2S mode timing diagram of I2S**

### 7.7.2 I2S MCLK/BCLK/LRCK Ratio and Format for CM6523 / CM6523B

| Internal Codec | | | | | |
|---|---|---|---|---|---|
| | Sampling Freq. | Resolution | Format | BCLK/LRCK | MCLK/LRCK |
| Default | | 24 bits | Left Justified | 64 | 256 |
| Others | 8/11.025/16/ 22.5/32/44.1/48 /88.2/96 | 16/24 bits | Left Justified / I2S-Mode | 64 | 256 |

| External Codec | | | | | |
|---|---|---|---|---|---|
| | Sampling Freq. | Resolution | Format | BCLK/LRCK | MCLK/LRCK |
| Master Mode | 8/11.025/16/ 22.5/32/44.1/48 | 16/24 bits | Left Justified / I2S-Mode | 64 | 256/512 |
| | 88.2/96 | 16/24 bits | Left Justified / I2S-Mode | 64 | 256 |
| Slave Mode mclk from CM6523 / CM6523B | 8/11.025/16/ 22.5/32/44.1/48 | 16/24 bits | Left Justified / I2S-Mode | 64 | 256/512 |
| | 88.2/96 | 16/24 bits | Left Justified / I2S-Mode | 64 | 256 |
| Slave Mode mclk from external | 8/11.025/16/ 22.5/32/44.1/48 /88.2/96 | 16/24 bits | Left Justified / I2S-Mode | 64 | 128/256/512 |






## 7.8 SPDIF Control Description

### 7.8.1 SPDIF Frame Description

- Audio format : linear 16 bit default, up to 24 bit expandable
- Allowed sampling frequencies (Fs) of the audio:
  - 44.1kHz from CD
  - 48 kHz from DAT
  - 32 kHz from DSR
- One way communication: from a transmitter to a receiver.
- Control information:
  - V (validity) bit : indicates if audio sample is valid.
  - U (user) bit : user free coding i.e. running time song, track number.
  - C (channel status) bit : emphasis, sampling rate and copy permit.
  - P (parity) bit : error detection bit to check for good reception.
- Coding format: biphase mark except the headers (preambles), for sync purposes.
- Bandwidth occupation : 100kHz up to 6Mhz (no DC!)
- Signal bitrate is 2.8Mhz (Fs=44.1kHz), 2Mhz (Fs=32kHz) and 3.1Mhz (Fs=48kHz).

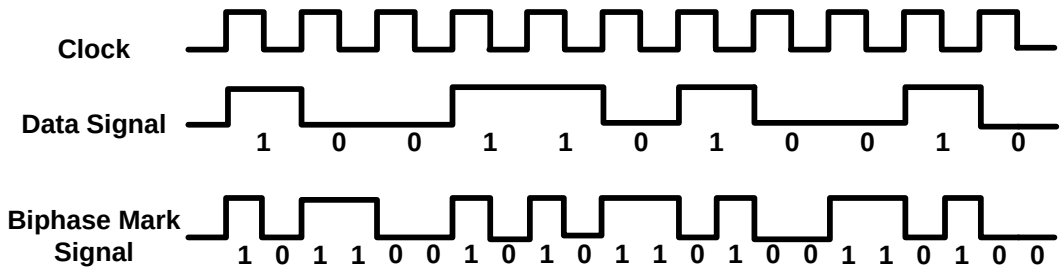



**Figure -17 Biphase Mark signal of SPDIF**

| Preamble | cell-order (last cell "0") | cell-order (last cell "1") |
|---|---|---|
| "B" | 11101000 | 00010111 |
| "M" | 11100010 | 00011101 |
| "W" | 11100100 | 00011011 |

Preamble B:

Marks a word containing data for channel A (left) at the start of the data-block.

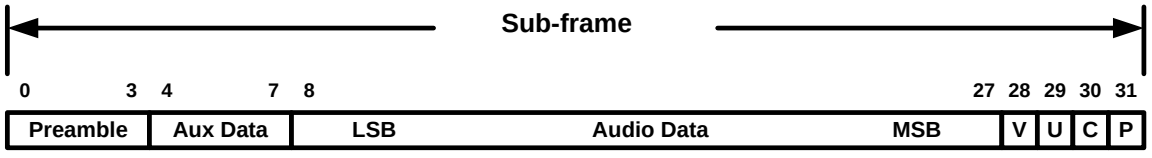
Preamble M:

Marks a word with data for channel A that isn't at the start of the data-block.

Preamble W:

Marks a word containing data for channel B. (right, for stereo). When using more than 2 channels, this could also be any other channel (except for A).

The number of subframes that are used depends on the number of channels that is transmitted. A CD-player uses Channels A and B (left/right) and so each frame contains two subframes. A block contains 192 frames and starts with a preamble "B":




V: Valid, U:User-Data, C:Channel-Status-Data, P:Parity-Bit

**Figure -5 SPDIF sub-frame description**

In each block, 384 bits of channel status and subcode info are transmitted. The Channel-status bits are equal for both subframes, so actually only 192 useful bits are transmitted:

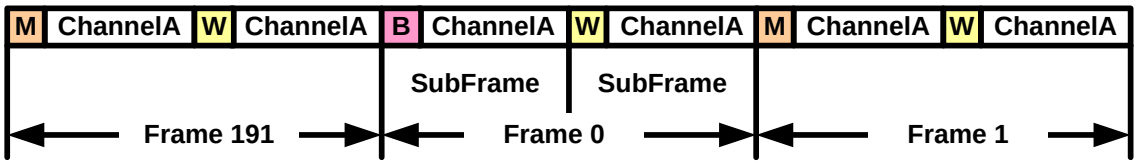



**Figure -6 Preamble Description of 192 SPDIF frame**

### 7.8.2 SPDIF Out Channel Status

| | bit0 | bit1 | bit2 | bit3 | bit4 | bit5 | bit6 | bit7 |
|---|---|---|---|---|---|---|---|---|
| **byte0** | **consumer /professional** | **audio/ non-audio** | **copyright** | **pre-emphasis** | | | **mode** | |
| default | 0(P) | 0(P) | 1(P) | 0(P) | 0(fixed) | 0(fixed) | 0(fixed) | 0(fixed) |
| **byte1** | **category code** | | | | | | | **L** |
| default | 0(P) | 0(P) | 0(P) | 0(P) | 0(P) | 0(P) | 0(P) | 0(P) |
| **byte2** | **source number** | | | | **channel number** | | | |
| default | 0(fixed) | 0(fixed) | 0(fixed) | 0(fixed) | 0(fixed) | 0(fixed) | 0(fixed) | 0(fixed) |
| **byte3** | **sampling frequency** | | | | **clock accuracy** | | **reserved** | |
| default | 0(P) | 0(P) | 0(P) | 0(P) | 0(fixed) | 0(fixed) | 0(fixed) | 0(fixed) |

NOTE

P : these bit can be programmed by USB HID or USB vendor command

### 7.8.3 SPDIF In Channel Status

| | bit0 | bit1 | bit2 | bit3 | bit4 | bit5 | bit6 | bit7 |
|---|---|---|---|---|---|---|---|---|
| **byte0** | **consumer /professional** | **audio/ non-audio** | **copyright** | **pre-emphasis** | | | **mode** | |
| default | 0( R) | 0( R) | 1( R) | 0( R) | 0( R) | 0( R) | 0( R) | 0( R) |
| **byte1** | **category code** | | | | | | | **L** |
| default | 0( R) | 0( R) | 0( R) | 0( R) | 0( R) | 0( R) | 0( R) | 0( R) |
| **byte2** | **source number** | | | | **channel number** | | | |
| default | 0( R) | 0( R) | 0( R) | 0( R) | 0( R) | 0( R) | 0( R) | 0( R) |

| byte3 | sampling frequency | | | | clock accuracy | | reserved | |
|---|---|---|---|---|---|---|---|---|
| default | 0( R) | 0( R) | 0( R) | 0( R) | 0( R) | 0( R) | 0(fixed) | 0(fixed) |

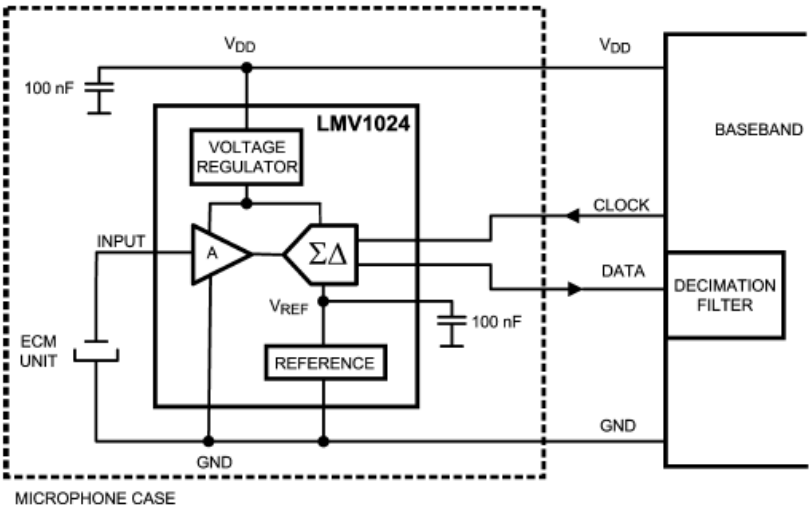
NOTE

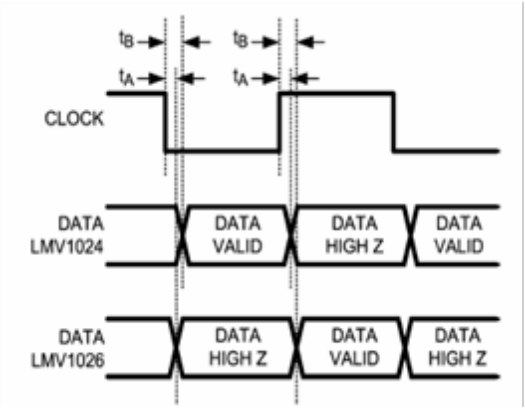
R : these bit can be read by USB HID or USB vendor command

## 7.9 Digital Mic

Digital_Mic Clock and Data Timing

**Typical Application**

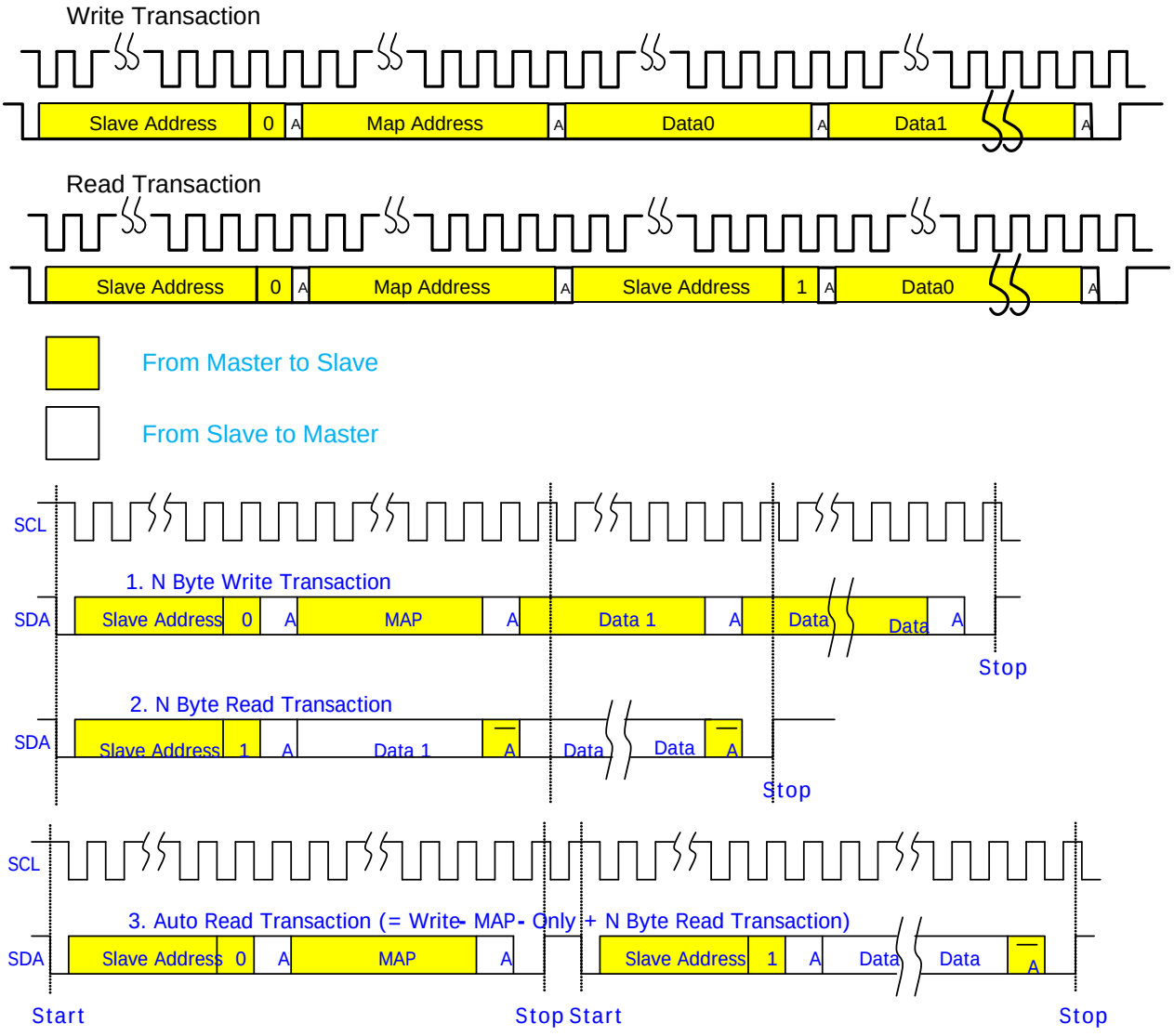


## 7.10 I2C Interface

### 7.10.1 I2C Master Mode

I2C protocol timing




from master to slave

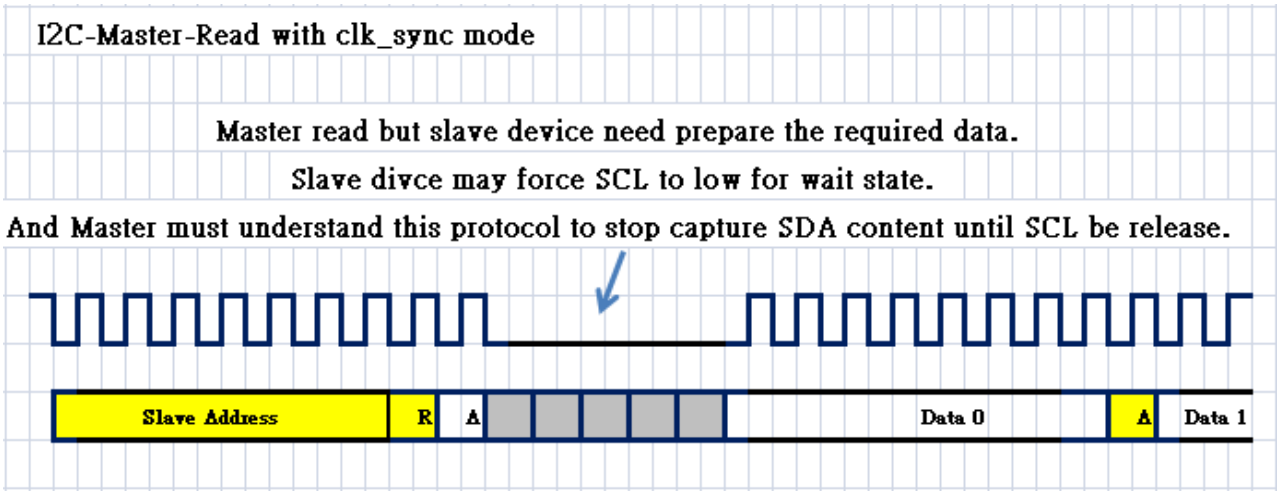
from slave to master

$A = acknowledge\ (SDA\ Low)$

$\overline{A} = not\ acknowledge\ (SDA\ High)$

MAP: Memory Address Pointer
(the target register address
in slave device )

### 7.10.2 I2C-Master Read with clk_sync mode




### 7.10.3 I2C Slave Mode

**Slave Mode Architecture**

**"7-bit slave address = 7'b0001000 to 7'b0001011"**

CM6523 / CM6523B can serves as a slave device with bit rate up to 400Kbps (fast mode). External MCU can write data to CM6523 / CM6523B or read data from CM6523 / CM6523B (No Size limitation in I2C Interface). Since host side and MCU can both access to all the internal registers.

CM6523 / CM6523B will transfer an interrupt to internal MCU until the INT bit of I2C control Register have been clean by internal MCU. The interrupt will be trigger when write transaction done or detect read-slave-address.

The main usage of 2-wire slave bus is to become the interface between the CM6523 / CM6523B and a external micro control unit (EMCU).

## 7.11 SPI Interface

### 7.11.1 SPI Master Mode

The SPI interface is used to transfer control data between the CM6523 / CM6523B and external codecs. It is not a standard interface. Every vendor has its implementation, and the implementation is somewhat different, but generally speaking, all of them comprise four signals, *spi_cen*, *spi_clock*, *spi_data_o*, *spi_data_i*. Their meanings are as follows.

- ***spi_cen***: the SPI chip enable signal that is used to inform a codec when it should latch the data.
- ***spi_clock***: the SPI clock signal.
- ***spi_data_o***: the SPI data output to codec.
- ***spi_data_i***: the SPI data input from codec.

**The SPI Design Goal and SPI Transactions**

Our goal is to design a robust SPI interface which can be suitable for all the existing codec. After analyzing the SPI of codec, we have written down the following difference among them.

1). An SPI interface which can read data from codec and write data to codec has 4 wires, but some codec only support input data. In other words, the data in the codec registers cannot be retrieved by audio processor. This kind of codec only needs 3 wires.
2). An SPI transaction length is 2 or 3 bytes depending on the codec.
3). Some codec latch control data at SPI clock high state, but some codec latch control data at SPI clock low state
4). SPI clock polarity and data capture phase can be selected by CPOL/CPHA register.
5). The upmost SPI clock frequencies are different for the codec.

For the difference 1 listed above, we have designed a 4-wire SPI interface, which is able to accommodate the 3-wire SPI interface as well. For difference 2 and 3, control bits in the SPI interface of the CM6523 / CM6523B is used to initiated a 2-byte or 3-byte data transfer, and maintain SPI clock high or low at codec latching data.

### 7.11.2 SPI transfer length 2B/3B

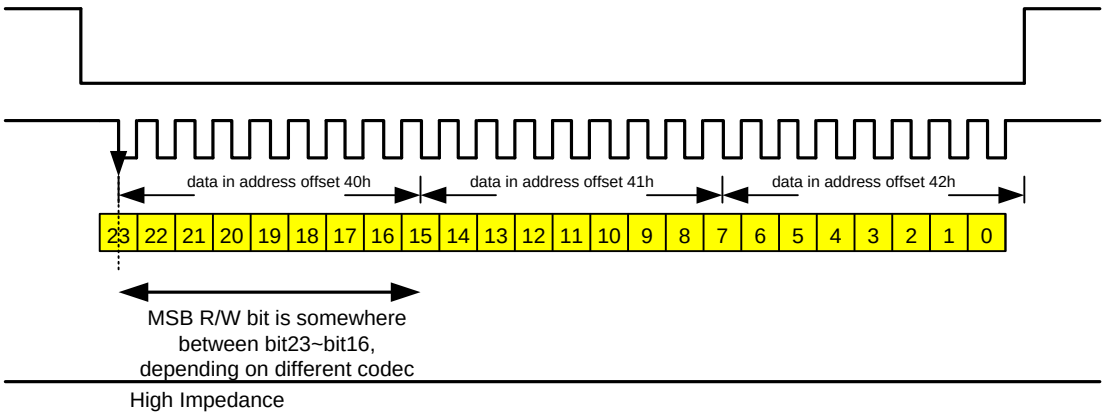



Fig. 28 An SPI 3-Byte Write transaction with codec latching data at spi_clk low state.

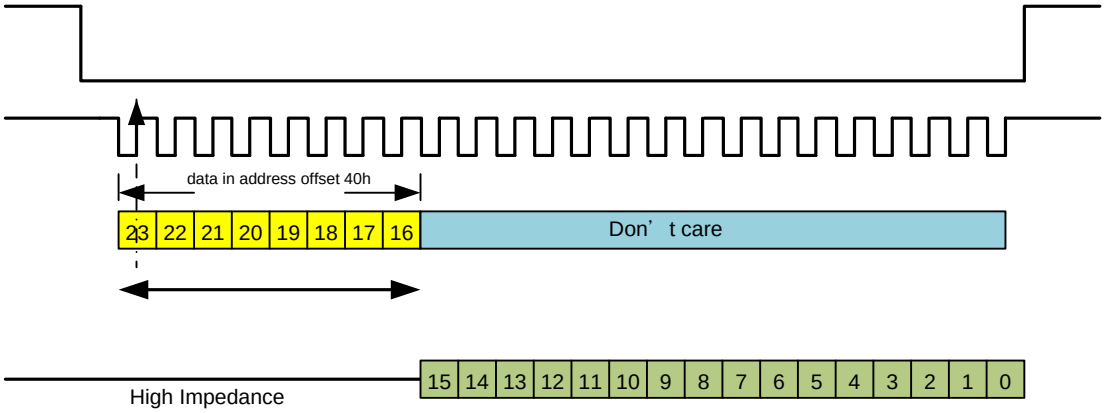



Fig. 29 An SPI 3-Byte Read transaction with codec latching data at spi_clk high state.

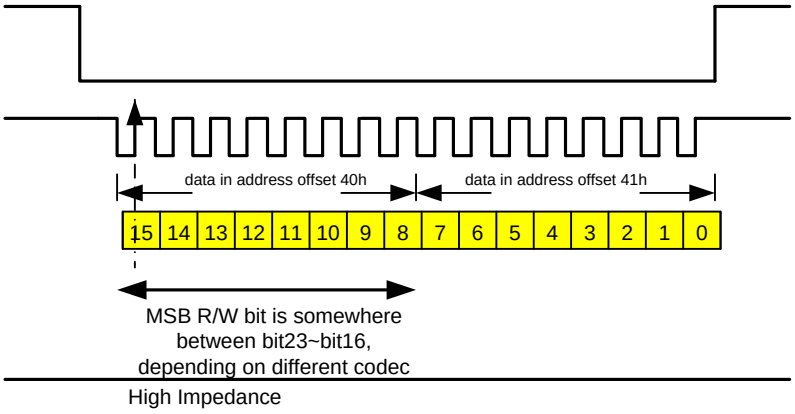



Fig. 30 An SPI 2-Byte Write transaction with codec latching data at spi_clk high state.

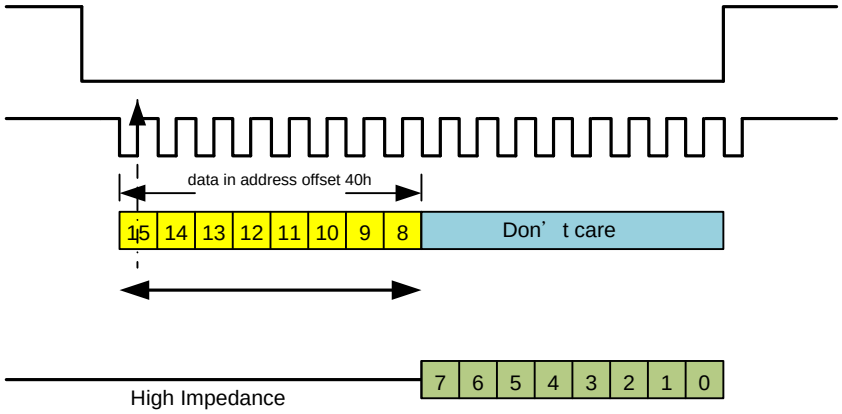
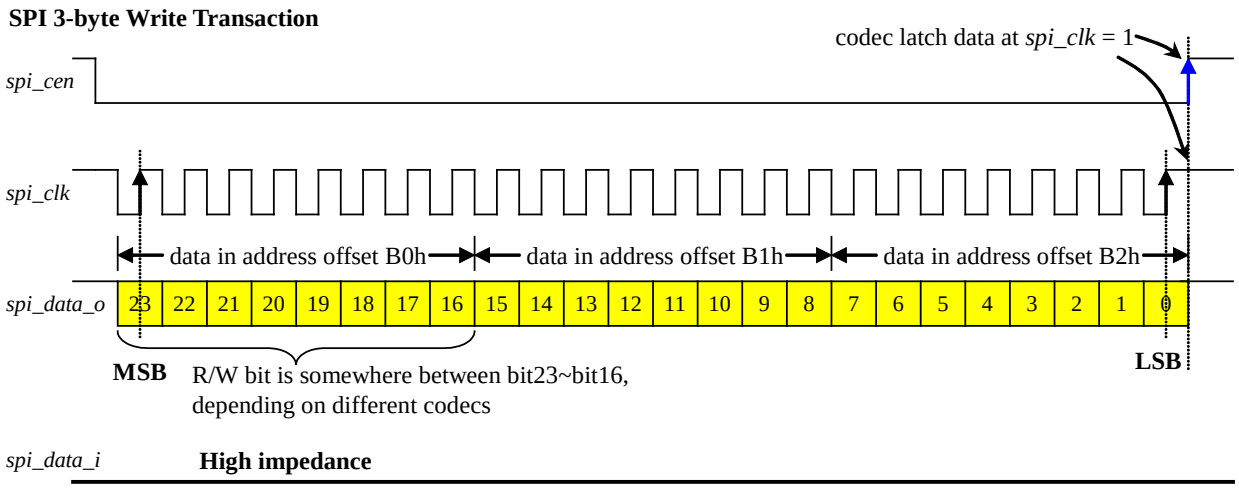
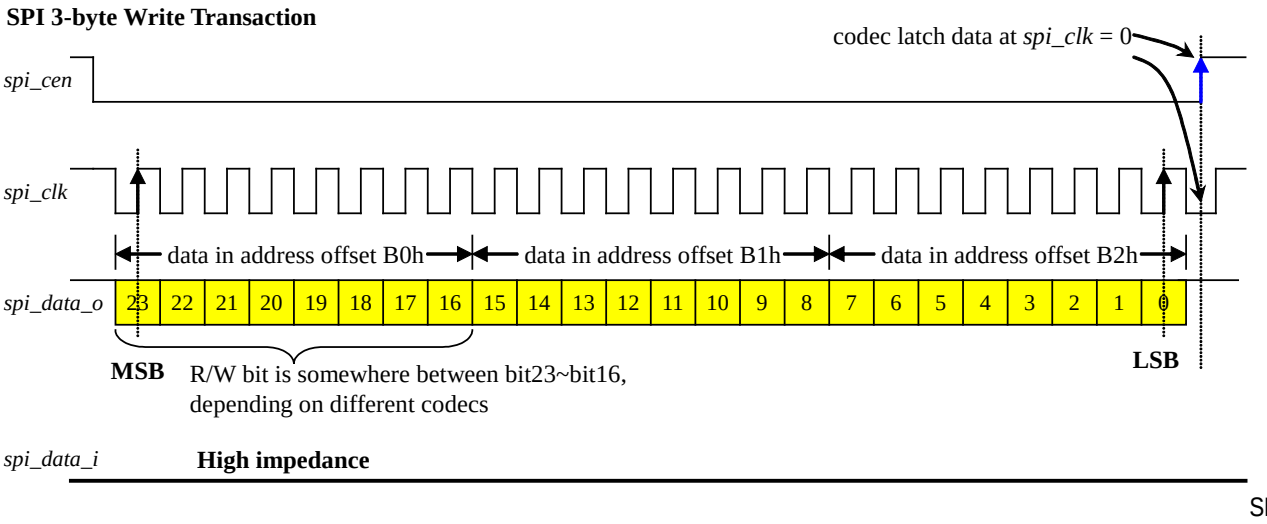



Fig. 31 An SPI 3-Byte Read transaction with codec latching data at spi_clk low state.

### 7.11.3 SPI latch data at high/low clock state

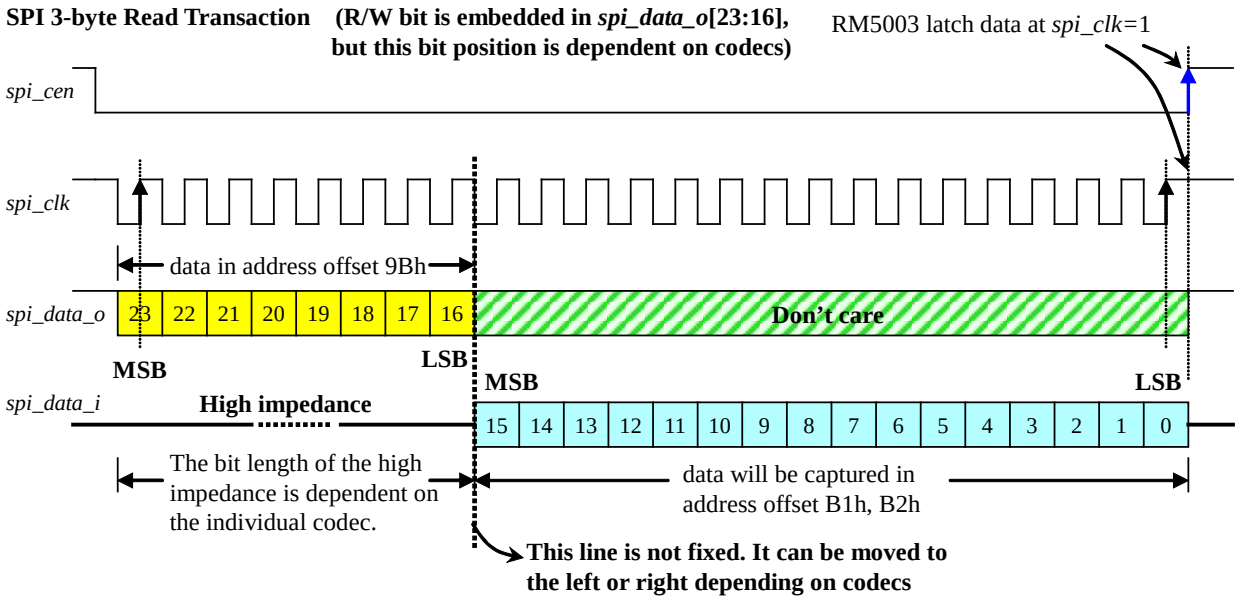



**SPI 3-byte Write Transaction with High state Latch**



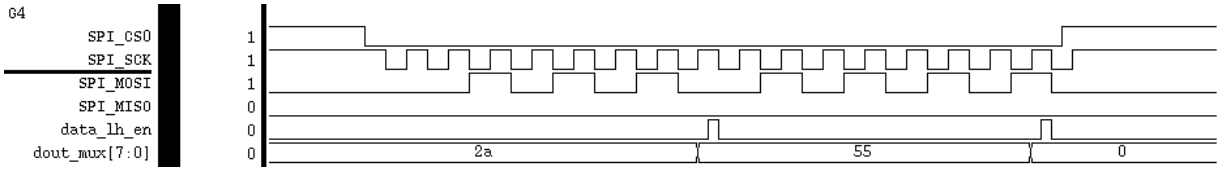

SPI

**3-byte Write Transaction with Low state Latch**




**SPI 3-byte Read Transaction with Low state Latch**

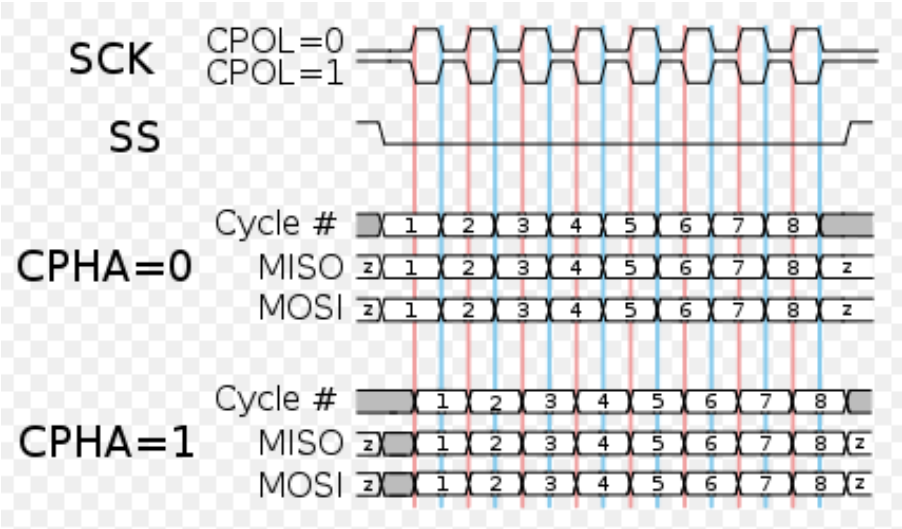
| | REG3E~3C | Description | Figure |
|---|---|---|---|
| 4 | C1_80_02 | CPOL=1/CPHA=1/MstIntEn/CodecLatch@SPICK_low/Len=2B | SPI-Fig.4 |




SPI-Fig.4

### 1) The SPI of CPOL/CPHA selection

A timing diagram showing clock polarity and phase




In addition to setting the clock frequency, the master must also configure the clock polarity and phase with respect to the data. Freescale's SPI Block Guide [1] names these two options as CPOL and CPHA respectively, and most vendors have adopted that convention.

The timing diagram is shown to the right. The timing is further described below and applies to both the master and the slave device.

- At CPOL=0 the base value of the clock is zero
  - For CPHA=0, data is captured on the clock's rising edge (low→high transition) and data is propagated on a falling edge (high→low clock transition).
  - For CPHA=1, data is captured on the clock's falling edge and data is propagated on a rising edge.
- At CPOL=1 the base value of the clock is one (inversion of CPOL=0)
  - For CPHA=0, data is captured on clock's falling edge and data is propagated on a rising edge.
  - For CPHA=1, data is captured on clock's rising edge and data is propagated on a falling edge.

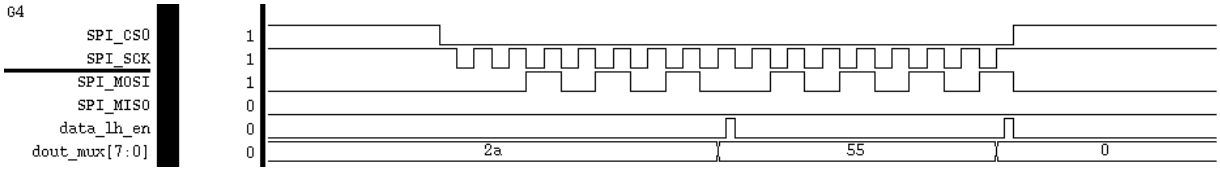
That is, CPHA=0 means sample on the leading (first) clock edge, while CPHA=1 means sample on the trailing (second) clock edge, regardless of whether that clock edge is rising or falling. Note that with CPHA=0, the data must be stable for a half cycle before the first clock cycle. For all CPOL and CPHA modes, the initial clock value must be stable before the chip select line goes active.

Also, note that "data *are* read" in this document more typically means "data *may be* read". The MOSI and MISO signals are usually stable (at their reception points) for the half cycle until the next clock transition. SPI master and slave devices may well sample data at different points in that half cycle.

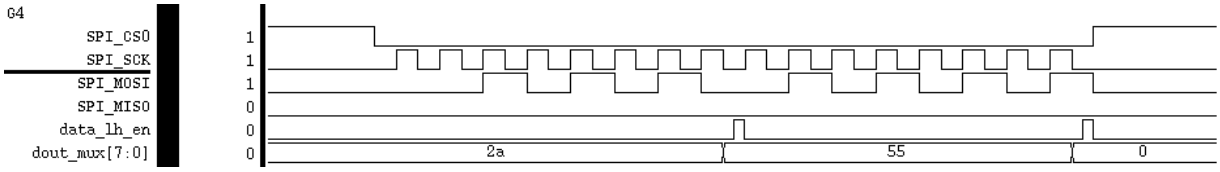
This adds more flexibility to the communication channel between the master and slave.

Some products use different naming conventions. For example, the TI MSP430 uses the name UCCKPL instead of CPOL, and its UCCKPH is the inverse of CPHA. When connecting two chips together, carefully examine the clock phase initialization values to be sure of using the right settings.

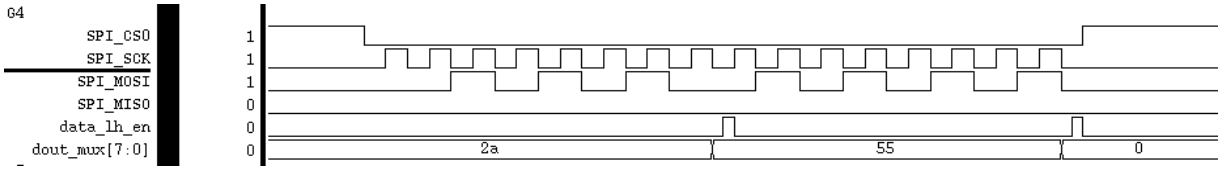
| | REG3E~3C | Description | Figure |
|---|---|---|---|
| 1 | C1_C0_02 | CPOL=1/CPHA=1/MstIntEn/Len=2B | SPI-Fig.1 |
| 2 | 41_C0_02 | CPOL=0/CPHA=1/MstIntEn/Len=2B | SPI-Fig.2 |
| 3 | 01_C0_02 | CPOL=0/CPHA=0/MstIntEn/Len=2B | SPI-Fig.3 |




SPI-Fig.1




SPI-Fig.2




SPI-Fig.3

### 7.11.4 SPI Slave Mode

In CM6523 / CM6523B, SPI-slave will trigger interrupt when receive 9th bit data in 3Byte mode. There are 6bits SPI-clocks reserved for MCU to prepare wanted read data. In this case, the SPI-clock operated at 800KHz and MCU have enough time to prepare read data. The procedure of MCU deal **SPI-slave-read** list below.

Step1: MCU get interrupt and clear.

Step2: MCU read REG3E[2] to determine it's a read or write command. And REG3E[3] defined as HID flag.

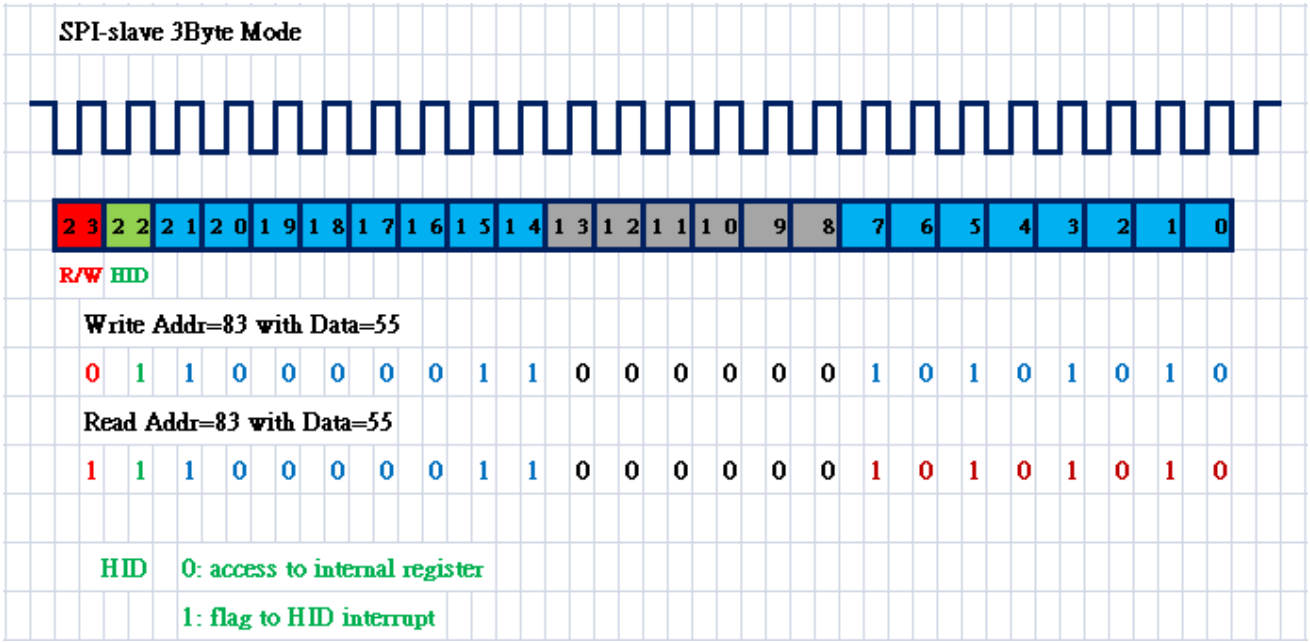
Step3: MCU read REG39 to get read address.

Step4: MCU prepare the content of read address.

Step5: MCU put the content into REG3B.

SPI-slave can read correctly only if all MCU operation under 6 SPI-clocks. We can NOT support too faster SPI negotiation.

**16b(2B-Addr-Phase) - 1b(R/W) - 1b(HID) - 8b(Addr) = 6b(Reserved)**




The procedure of MCU deal **SPI-slave-write** list below.

Step1: MCU get interrupt and clear.

Step2: MCU read REG3E[2] to determine it's a read or write command. And REG3E[3] defined as HID flag.

Step3: MCU read REG39 to get read address.

# 8 Electrical Characteristics

## 8.1 Absolute Maximum Ratings

**Test Conditions: DV50 = 5V, AV50 = 5V, DGND =0V, TA=+25°C**

| Parameter | Symbol | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Storge temperature | - | -25 | - | 150 | °C |
| Operating ambient temperature | - | 0 | 25 | 75 | °C |
| Digital supply voltage(DV50) | - | 4.5 | 5.0 | 5.5 | V |
| Analog Supply Voltage(AV50) | | 4.5 | 5.0 | 5.5 | V |
| I/O pin voltage | - | GND | - | $V_{DD}$ | V |
| ESD(Human Body Mode) | | | ±4000 | | V |
| ESD(Machine Mode) | | | ±200 | | V |

## 8.2 Recommended Operation Conditions

| Parameter | Symbol | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Analog Supply Voltage | - | | 5 | | V |
| Digital Supply Voltage | | | 5 | | |
| Operating Ambient Temperature | | | 25 | | °C |
| Crystal Clock | - | | 12.000 | | MHz |

## 8.3 Power Consumption

**Test Conditions: DV50=5V, AV50 = 5V, DGND =0V, TA=+25°C**

**Sample Rate=48Khz, 16Bits, Operation: HP-Out Playback+Mic-In Recording, EQ disable, Spdif out disable**

| Parameter | Symbol | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Total Power Consumption (Playback+Record) | - | - | 55 | - | mA |
| Standby Power Consumption | - | - | 50 | - | mA |
| Suspend Mode Power Consumption | - | - | 10 | - | uA |

## 8.4 DC Characteristics

**Test Conditions: DV50=5V, $V_{DD}$ = 3.3V, DGND =0V, TA=+25°C, $V_{DD}$ = 3.3V**

| Parameter | Symbol | Min | Typ | Max | Units |
|---|---|---|---|---|---|
| Input voltage range | Vin | $V_{DD}$-0.3 | $V_{DD}$ | $V_{DD}$+0.3 | V |
| Output voltage range | Vout | 0 | - | $V_{DD}$ | V |
| High level input voltage | Vih | 0.7$V_{DD}$ | - | - | V |
| Low level input voltage | Vil | - | - | 0.3$V_{DD}$ | V |
| High level output voitage | Voh | 2.4 | - | - | V |

| | | | | | |
|---|---|---|---|---|---|
| Low level output voltage | Vol | | - | 0.4 | V |
| Input leakage current | Iil | -10 | - | 10 | uA |
| Output leakage current | Iol | -10 | - | 10 | uA |
| Output buffer driver current | - | 2 | 8 | 16 | mA |
| SPDIF transmit output driver current | - | 2 | 8 | 16 | mA |

## 8.5 Audio Performance

### 8.5.1 DAC Audio Quality

TA=25℃, DV50=5V, AV50=5V

| Items | Test Conditions | Test Values | | | Unit |
|---|---|---|---|---|---|
| | | Min | Typ | Max | |
| Full scale output voltage | 10KΩ loading fs=48KHz | | 0.95 | | Vrms |
| | 32Ω loading fs=48KHz | | 0.82 | | Vrms |
| THD+N @ Vout=-3dB | 10KΩ loading fs=48KHz/16bits,A-Weighted | -78 | | -93 | dB |
| | 10KΩ loading fs=96KHz/24bits, A-Weighted | -79 | | -94 | dB |
| | 32Ω loading fs=48KHz/16bits,A-Weighted | -67 | | -92 | dB |
| | 32Ω loading fs=96KHz/24bits,A-Weighted | -67 | | -95 | dB |
| Dynamic range with signal present | 10KΩ loading fs=48KHz/16bits,A-Weighted | | 91 | | dB |
| | 10KΩ loading fs=96KHz/24bits, A-Weighted | | 94 | | dB |
| | 32Ω loading fs=48KHz/16bits,A-Weighted | | 92 | | dB |
| | 32Ω loading fs=96KHz/24bits,A-Weighted | | 94 | | dB |
| Noise level during system activity | 10KΩ loading fs=48KHz/16bits,A-Weighted | | 94 | | dB |
| | 10KΩ loading fs=96KHz/24bits, A-Weighted | | 96 | | dB |
| | 32Ω loading fs=48KHz/16bits,A-Weighted | | 96 | | dB |
| | 32Ω loading fs=96KHz/24bits,A-Weighted | | 94 | | dB |
| Inter channel phase delay | 100Hz ~ 20KHz | +0.02 | | +1.05 | deg |

| | | | | | |
|---|---|---|---|---|---|
| Sampling frequency accuracy | | 10KΩ loading<br>fs=48KHz/16bits,A-Weighted | -0.0043 | +0.0015 | % |
| Channel separation | | 10KΩ loading<br>fs=48KHz/16bits,A-Weighted | 98 | 119 | dB |
| | | 32Ω loading<br>fs=48KHz/16bits,A-Weighted | 67 | 78 | dB |
| Magnitude Response | Frequency Response | 10KΩ loading<br>fs=48KHz/16bits,A-Weighted | -0.085 | -0.937 | dB |
| | Passband Ripple | 10KΩ loading<br>fs=48KHz/16bits,A-Weighted | | 0.291 | dB |

### 8.5.2 ADC Audio Quality

TA=25℃, DV50=5V, AV50=5V, Input test signal is 997Hz sine wave, measure bandwidth is 20Hz to 20KHz

| Items | Test Conditions | Test Values | | | Unit |
|---|---|---|---|---|---|
| | | Min | Typ | Max | |
| Full scale output voltage | Microphone<br>fs=48KHz | | 1.11 | | Vrms |
| | Line in<br>fs=48KHz | | 1.08 | | |
| THD+N @ Vout=-3dB | Microphone<br>fs=48KHz/16bits,A-Weighted | -81 | | -89 | dB |
| | Microphone<br>fs=96KHz/24bits, A-Weighted | --82 | | -91 | dB |
| | Line in<br>fs=48KHz/16bits,A-Weighted | -82 | | -90 | dB |
| | Line in<br>fs=96KHz/24bits,A-Weighted | -82 | | -90 | dB |
| Dynamic range with signal present | Microphone<br>fs=48KHz/16bits,A-Weighted | | 90 | | dB |
| | Microphone<br>fs=96KHz/24bits, A-Weighted | | 91 | | dB |
| | Line in<br>fs=48KHz/16bits,A-Weighted | | 90 | | dB |
| | Line in<br>fs=96KHz/24bits,A-Weighted | | 90 | | dB |

| | | | | |
|---|---|---|---|---|
| Sampling frequency accuracy | Microphone fs=48KHz/16bits | +0.0001 | +0.009 | % |
| | Line in fs=48KHz/16bits | -0.0048 | -0.0034 | |
| Channel separation | Microphone fs=48KHz/16bits | 81 | 91 | dB |
| | Microphone fs=96KHz/24bits | 83 | 91 | dB |
| | Line in fs=48KHz/16bits | 86 | 89 | |
| | Line in fs=96KHz/24bits | 87 | 90 | |
| Frequency Response | Microphone fs=48KHz/16bits,A-Weighted | -0.433 | -0.484 | dB |
| | Line in fs=48KHz/16bits,A-Weighted | -0.313 | -0.695 | |
| Passband Ripple | Microphone fs=48KHz/16bits,A-Weighted | | 0.204 | dB |
| | Line in fs=48KHz/16bits,A-Weighted | | 0.159 | |

### 8.5.3 A-A path Audio Quality

**TA=25℃, DV50=5V, AV50=5V**

| Items | Test Conditions | Test Values | | | Unit |
|---|---|---|---|---|---|
| | | Min | Typ | Max | |
| Full scale output voltage | Microphone to Line out | | 1.09 | | Vrms |
| THD+N @ Vout=-3dB | Microphone to Line out fs=48KHz/16bits,A-Weighted | -80 | | -81 | dB |
| Dynamic range with signal present | Microphone to Line out fs=48KHz/16bits,A-Weighted | | 92 | | dB |
| Channel separation | Microphone to Line out fs=48KHz/16bits,A-Weighted | 74 | | 119 | dB |
| Frequency Response | Microphone to Line out fs=48KHz/16bits,A-Weighted | -0.194 | | +0.484 | dB |

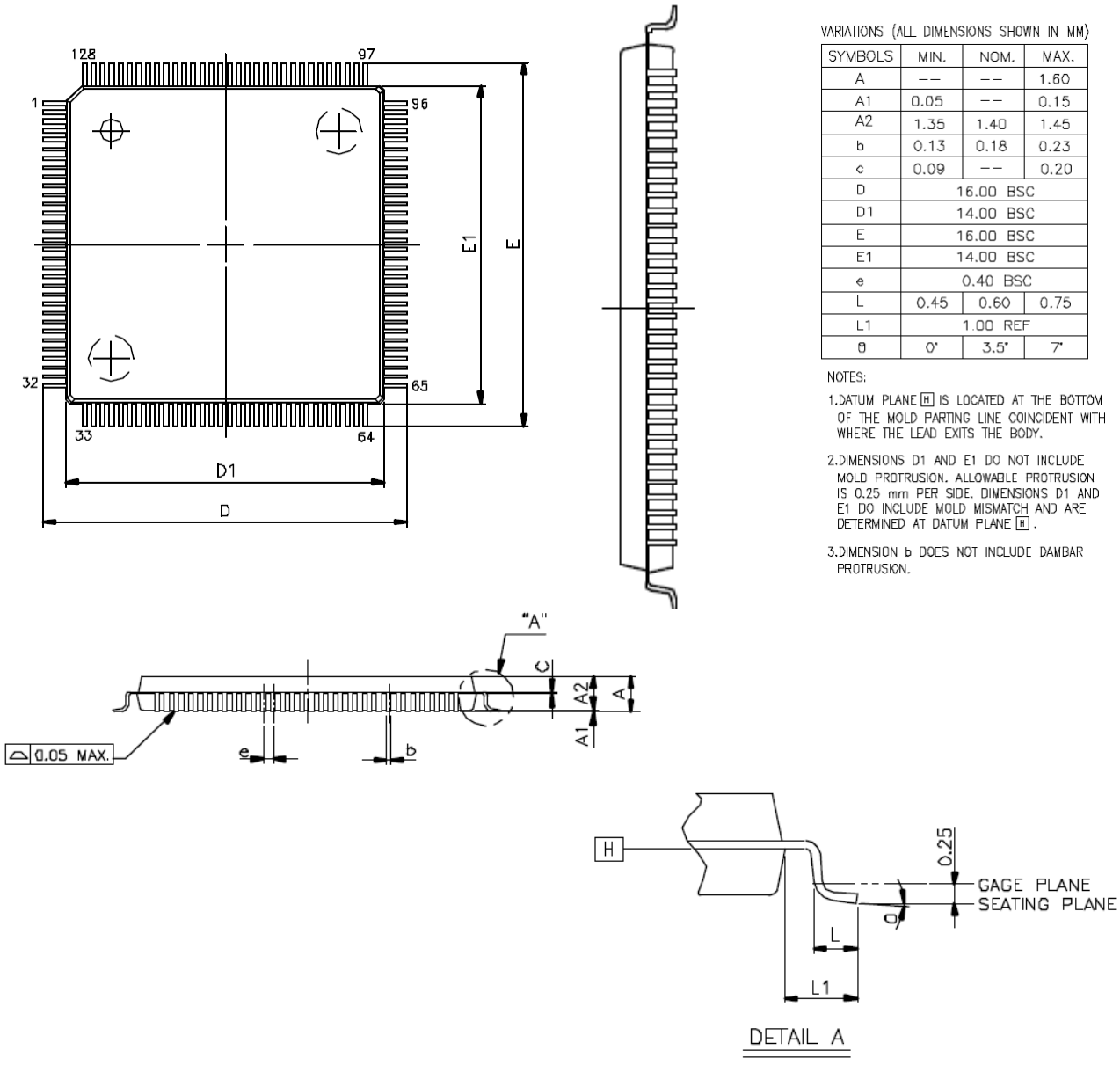
| | | | | | |
|---|---|---|---|---|---|
| Passband Ripple | Microphone<br>fs=48KHz/16bits,A-Weighted | | | 0.1 | dB |

## 9 Package Dimension

| Model Number | Package | Operating Ambient Temperature | Supply Range |
|---|---|---|---|
| CM6523 / CM6523B | 128-Pin LQFP 14mm×14mm×1.4mm (Plastic) | -15°C to +70°C | DVdd = 5V, AVdd = 5V |

**Outline Dimensions** ***Dimensions shown in inches and (mm)**

**128-Lead Thin Plastic Quad Flatpack (LQFP)**




VARIATIONS (ALL DIMENSIONS SHOWN IN MM)

| SYMBOLS | MIN. | NOM. | MAX. |
|---|---|---|---|
| A | -- | -- | 1.60 |
| A1 | 0.05 | -- | 0.15 |
| A2 | 1.35 | 1.40 | 1.45 |
| b | 0.13 | 0.18 | 0.23 |
| c | 0.09 | -- | 0.20 |
| D | 16.00 BSC | | |
| D1 | 14.00 BSC | | |
| E | 16.00 BSC | | |
| E1 | 14.00 BSC | | |
| e | 0.40 BSC | | |
| L | 0.45 | 0.60 | 0.75 |
| L1 | 1.00 REF | | |
| θ | 0° | 3.5° | 7° |

NOTES:

1. DATUM PLANE [H] IS LOCATED AT THE BOTTOM OF THE MOLD PARTING LINE COINCIDENT WITH WHERE THE LEAD EXITS THE BODY.
2. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 mm PER SIDE. DIMENSIONS D1 AND E1 DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE [H].
3. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.

**Package Dimension of CM6523 / CM6523B**

## —End of Specifications—

**C-MEDIA ELECTRONICS INC.**

6F., 100, Sec. 4, Civil Boulevard, Taipei, Taiwan 106 R.O.C.

TEL：+886-2-8773-1100

FAX：+886-2-8773-2211

E-MAIL：sales@cmedia.com.tw