C-Media CM65xx USB 1.0 Controller

Firmware Programmers' Guide

C-MEDIA Electronics Inc.

Revision History

Revision	Date	Description
1.00	2013/01/24	Draft
1.01	2013/05/20	Redefine the file structure of Framework.
2.01	2014/03/12	The new library CM65xxB-1.lib has been released, and three code templates have been released, include headset, speaker, and microphone.
2.02	2014/05/27	Updated structures, variables, macros, and functions of library. Updated customized functions for customized application. Added register tables of CM65xx, and codec CMA112.

Contents

1. Introduction

CM65xx is a family of R8051XC (1T)-based chip with on-chip memory for USB audio device. It complies with USB 1.1 and 2.0 specifications for full speed. Besides control endpoint 0, CM65xx provides one isochronous-in endpoints for ADC, one isochronous-out endpoints for DAC, one interrupt-in endpoints, two bulk-in endpoint, two bulk-out endpoint and one feedback-in endpoint.

The CM65xx firmware framework provides an easy-to-use software platform to simplify the development process of USB audio 1.0 peripherals with CM65xx family – CM65xx. This document acts as a programmer's reference manual for developers who need to design firmware based on CM65xx firmware framework.

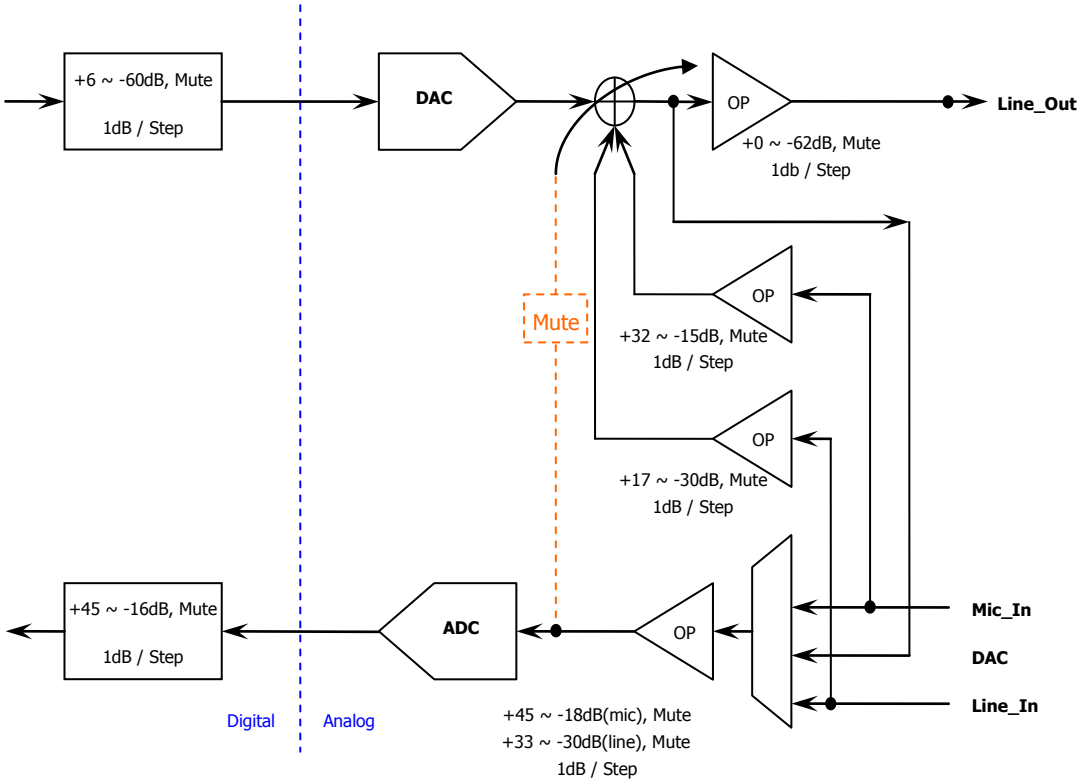
There are sample applications of USB audio device in the framework package. The sample applications can be run on hardware platform of C-Media. Users can modify the application to develop different USB devices.

2. Features

The features of CM65xx sample application are listed below.

- Supports USB 1.1 and 2.0 full-speed operation.
- Compliance
  - USB HID class 1.1.
  - USB audio class 1.0.
- User can design a user-defined USB device.
- User can easily control peripherals of CM65xx including GPIO, UART, …
- Programmable MCU clock Speed (3/6/12/24/48 MHz)(default = 12MHz)
- Supports buttons “play-mute”, “rec-mute”, “vol-up”, “vol-down”, “play/pause”, “stop”, “next”, and “previous”.
- Supports LEDs “config/play/record”, “play-mute”, “rec-mute”, “play-rec”, and “rec-clipping”.
- Output capability
  - Support 2-channel speaker output via internal DAC codec.
  - Support 2-channel SPDIF output.
- Input capability
  - Support 2-channel analog recording via internal ADC codec.
  - Support 2-channel SPDIF input
  - Support A/A paths for monitoring microphone and line inputs.
- Customized volume range
  - Analog
    - DAC gain: -62dB~0dB/step 1dB.
    - Mic ADC gain: 0dB~30dB/step 1dB.
    - Line ADC gain: -30dB~12dB/step 1dB.
    - Mic A/A gain: -15dB~22dB/step 1dB.
    - Line A/A gain: -30dB~12dB/step 1dB.
  - Digital (I2S)
    - DAC gain: -60dB~0dB/step 1dB.
    - ADC gain: -16dB~12dB/step 1dB.
- Three topologies are supported. They are headset, speaker only and microphone only. The PID and VID can be customized by the users no matter what the bonding option is.

3. Internal Codec Block Diagram



4. Packages

The directory “Framework” includes the CM65xx’s library which supports three topologies and sample applications of USB audio device. There are four sub-directories in the directory “Framework” that is shown as below.

Framework\
Tools\ - Hex2Rom.exe which is used to transfer hex files.
Doc\ - Documents
CM65xxB-1\ - CM65xx framework
inc\ - Header files
CM65XXB-1.LIB - The library of CM65xx framework
CM65xxB-1_xxx\ - Sample application source codes, xxx could be Headset, Speaker, or Microphone.
inc\ - Header files for customization
output\ - the output files after build
**.c, *.a51* - the source code for sample template
make.bat, build.bat - Batch files for building the project in the console mode

The source files, header files, and the library file of CM65xx firmware framework are listed in the following table:

Directory	Files	Description
Framework\ **CM65xxB-1**\	CM65xxB-1.LIB	The library file of framework.
Framework\ **CM65xxB-1**\ *inc*\	audio.h cm65xxlib.h cm65xx.h registers.h types.h usb.h	Header files of USB audio class 1.0 based on CM65xx ROM firmware.
Framework\ **CM65xxB-1_xxx**\ *inc*\	config.h	Header files for customization.
Framework\ **CM65xxB-1_xxx**\	main.c int.c usb.c io.c request.c	A sample application files of USB audio 1.0 device.

	audio.c dscr.a51	
Framework\ **CM65xxB-1_xxx**\	cm65xx.lin cm65xx_cmd.lin	Link script files, cm65xx.lin is for KeilC environment, and cm65xx_cmd.lin is for the console mode.
Framework\ **CM65xxB-1_xxx**\	CM65xx.Uv2 make.bat build.bat	CM65xx.Uv2 is the KeilC setting file, and batch files are for building in the console mode.
Framework\ **CM65xxB-1_xxx**\ *output*\		The output files after building.

5. Build and Update

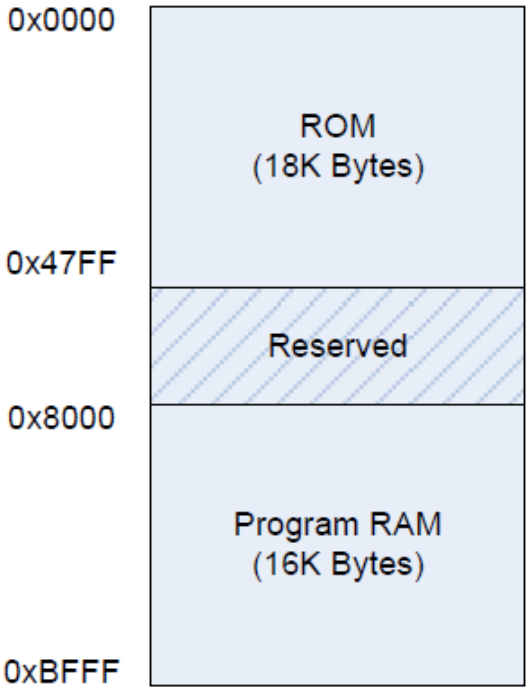
CM65xx firmware framework and sample applications are built by tool chain of Keil uVision3 V3.33. They have been built successfully by complier C51 V8.05a, assembler A51 V8.00b, and linker BL51 V6.02.

In the directory of sample application, there is a batch file “build.bat” for building the sample application. Users can build the firmware easily by going into directory “CM65xxB-1_xxx” and executing “build xxxx” in Windows’ console. Currently, one file “cm65xxfw.hex” will be created in the directory “CM65xxB-1_xxx”. Users can modify these batch files according to users’ requirement.

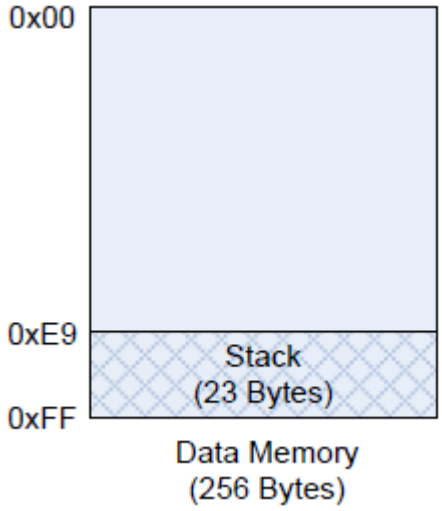
After building firmware, which the tool *Hex2Rom.exe* is also executed, two different bin files are made which are *Cm65xxCode.bin* and *CM65xxB-1_ROM.bin*; the *Cm65xxCode.bin* file is translated directly from *cm65xxfw.hex*, and *CM65xxB-1_ROM.bin* includes header settings so it can be written into EEPROM by using C-Media’s PC download tool.

6. Memory Usage

Program memory layout is shown below. The size of ROM memory is 18k bytes. One purpose of ROM is to be a boot loader that transmit the customization code, if it exists, from EEPROM to internal RAM sector (0x8000~0xBFFF); and the other purpose is to used as the default USB audio devices that include headset/speaker/microphone/docking/Lync_headset topologies.

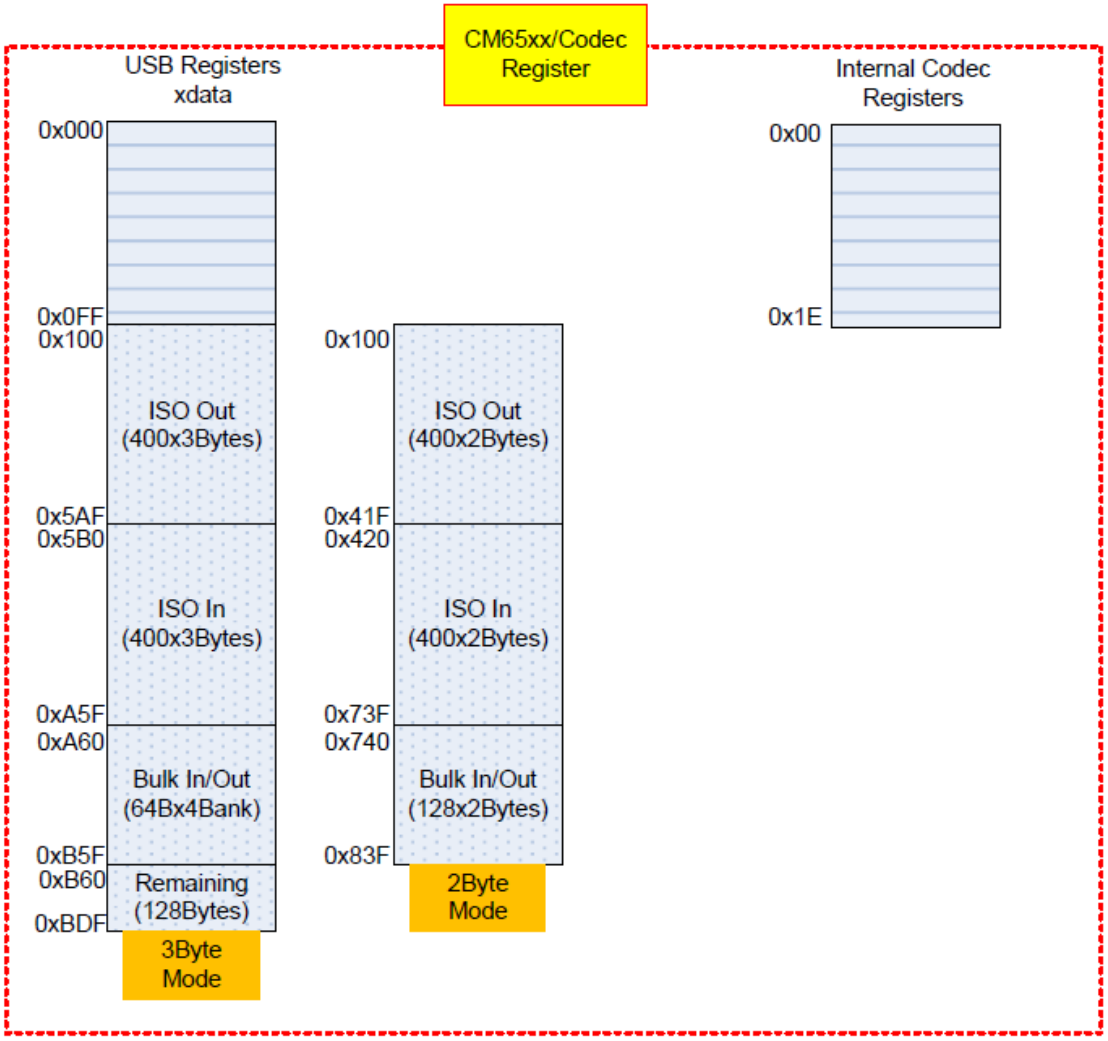


As general MCU 8051, the total data memory size is 256 bytes. Section DATA (0~0x7F) is accessible using direct addressing, the other section IDATA (0x80~0xFF) is accessible using indirect addressing. Currently, section 0xE9~0xFF is reserved for stack. This memory allocation can be changed by firmware developers.



The memory map of XDATA of CM65xx and internal codec registers are shown below.

Several memory areas can be used for customization if the internal data memory is not enough for development. For example, memory section 0xA60~0xBDF can be used for customized usage in 3Byte mode.



7. Interface of Standard CM65xx Library

In “Framework\CM65xxB-1\”, there is the library file and major header files in the folder “inc”; these files offer the customers the basic data structures and essential function calls to construct their own firmware.

7.1 Structures

- **Descriptor structure**

Device descriptor structure as follows:

```
typedef struct _USB_DEVICE_DSCR_STRUCT
{
    BYTE bLength;              // Descriptor length ( = sizeof(DEVICEDSCR) )
    BYTE bDescriptorType;      // Descriptor type (Device = 1)
    WORD bcdUSB;               // Specification Version (BCD)
    BYTE bDeviceClass;         // Device class
    BYTE bDeviceSubClass;      // Device sub-class
    BYTE bDeviceProtocol;      // Device sub-sub-class
    BYTE bMaxPacketSize0;      // Maximum packet size for endpoint zero
    WORD idVendor;             // Vendor ID
    WORD idProduct;            // Product ID
    WORD bcdDevice;            // Product version ID
    BYTE iManufacture;         // Manufacturer string index
    BYTE iProduct;             // Product string index
    BYTE iSerialNumber;        // Serial number string index
    BYTE bNumConfigurations;   // Number of configurations
} USB_DEVICE_DSCR_STRUCT;
```

- **HID descriptor structure**

```
typedef struct _USB_HID_DSCR_STRUCT
{
    BYTE bLength;              // Descriptor length ( = sizeof(DEVICEDSCR) )
    BYTE bDescriptorType;      // Descriptor type (Device = 1)
    WORD bcdHID;
    BYTE bCountryCode;
    BYTE bNumDescriptors;
    BYTE bDescriptorType1;
    WORD wDescriptorLength;
```

```
} USB_HID_DSCR_STRUCT;
```

- **USB request**

```
typedef struct _USB_CONTROL_COMMAND_STRUCT
{
    BYTE bmRequestType;
    BYTE bRequest;
    WORD wValue;
    WORD wIndex;
    WORD wLength;
} USB_CONTROL_COMMAND_STRUCT;
```

USB_CONTROL_COMMAND_STRUCT defines the structure of an USB request from host. It complies with USB specification.

- **Stages of USB control transfer**

```
typedef enum
{
    NONE_STAGE,
    SETUP_STAGE,
    DATA_IN_STAGE,
    DATA_OUT_STAGE,
    STATUS_STAGE,
    STALL_STAGE
} USB_CONTROL_STATE;
```

This stage enumeration defines several stages of a control transfer that firmware flow handles.

- **Audio class-specific enumerations**

These enumerations are dedicated for audio requests, features, and entities defined in USB audio class specification.

Classification of audio features and entities:

```
typedef enum
{
    AC_NODEF = 0,
    AC_IT,
    AC_OT,
    AC_FEATURE,
```

```
    AC_MIXER,
    AC_SELECTOR
} AC_NODE_TYPE;
```

Audio class-specific request codes:

```
typedef enum
{
    CMD_NODEF = 0,
    CMD_SET_CURRENT,
    CMD_SET_MIN,
    CMD_SET_MAX,
    CMD_SET_RES,
    CMD_GET_CURRENT = 0x81,
    CMD_GET_MIN,
    CMD_GET_MAX,
    CMD_GET_RES,
    CMD_GET_MEM = 0x85
} AC_CMD_ATTR;
```

Audio feature unit:

```
typedef enum
{
    FEATURE_DAC = 0,
    FEATURE_ADC_LINE,
    FEATURE_ADC_MIC,
    FEATURE_MIXER,
    FEATURE_MONITOR_LINE,
    FEATURE_MONITOR_MIC,
    FEATURE_SPDIF
} FEATURE_UNIT_NO;
```

FEATURE_DAC indicates the feature unit of DAC.
FEATURE_ADC_LINE indicates the feature unit of LINE-IN.
FEATURE_ADC_MIC indicates the feature unit of MIC-In.
FEATURE_MIXER indicates the feature unit of MIXER. (DAC path + AA path from MIC IN and LINE IN).
FEATURE_MONITOR_LINE indicates the feature unit of LINE-IN AA path.
FEATURE_MONITOR_MIC indicates the feature unit of MIC-IN AA path.
FEATURE_SPDIF indicates the feature unit of SPDIF-IN.

Sources of ADC selector unit:

There are four ADC input sources for the selector unit, defined as follows.

```
typedef enum
{
    SELECTOR_MIC = 1,
    SELECTOR_LINE_IN,
    SELECTOR_SPDIF_IN,
    SELECTOR_SMIX_IN
} SELECTOR_SELECT;
```

SELECTOR_MIC can be used to select MIC-IN source.

SELECTOR_LINE_IN can be used to select LIN-IN source.

SELECTOR_SPDIF_IN can be used to enable SPDIF-IN source.

SELECTOR_SMIX_IN can be used to select the stereo mixer input (DAC + AA path from MIC IN and LINE IN).

- Control tables for the USB device

Audio control structure:

Below functions are used to control the audio features like volume, mute, AGC, etc.

```
typedef struct _AUDIO_CONTROL_STRUCT
{
    void                (*featureVolume)();
    void                (*featureMute)();
    void                (*featureAgc)();
    void                (*setSelector)();
    void                (*recordMute)();
    BYTEcode            *pAcUnitTable;
    AS_CONTROL_STRUCT   asControl[2];
} AUDIO_CONTROL_STRUCT;
```

“featureVolume” function is used to control volume feature unit.

“featureMute” function is used to control mute feature unit.

“featureAgc” function is used to control AGC which can avoid the clipping from recording.

“setSelector” function is used to control the ADC input source.

“recordMute” function is used to mute the recording path.

“pAcUnitTable”is assigned to an audio control table.

“asControl[2]” is assigned to an audio stream interface.

Audio stream control structure:

```
typedef struct _AS_CONTROL_STRUCT
{
    BYTE attribute;             // bit7: invalid = 1, bit6: mono/stereo, bit5~4:
interface number, bit3~0: max alternate
    BYTE endpoint;              // endpoint number and direction
    void (*setFrequency)();
} AS_CONTROL_STRUCT;
```

Convertible device control structure:

```
typedef struct _CONVERTIBLE_CONTROL_TABLE_STRUCT
{
    DEVICE_CONTROL_STRUCT code   *pDeviceControl;
    AUDIO_CONTROL_STRUCT code    *pAudioControl;
    void (*initialize)();       // to configure GPIO settings and LED
    void (*reset)();
    void (*configure)();        // executed when got configured
    void (*event)();            // conduct extended events
    void (*gpi)();              // conduct GPI interrupt events
    void (*ir)();               // conduct IR events
} CONVERTIBLE_CONTROL_TABLE_STRUCT;
```

Device control structure:

```
typedef struct _DEVICE_CONTROL_STRUCT
{
    BYTE bAudioInterface;       // the quantity of audio interface
    BYTE bHidInterface;         // the number of HID interface
} DEVICE_CONTROL_STRUCT;
```

Enumeration of string descriptor:

```
typedef enum
{
    LANGID_S = 0,
    MANUFACTURE_S,
    PRODUCT_S,
    SERIAL_S,
    LYNC_S = 33,
    VERSION1_S = 50,
```

```
    VERSION2_S = 80
} STRING_DSCR;
```

IR module enumeration:

```
typedef enum
{
    NEC = 0,
    RC5,
    RC6
} IR_MODULE;
```

7.2 Public Variables

Necessary variables are defined in the library and will be used in the application firmware, these are listed as below:

Variable	Description
BOOL g_bmUsbPuReset	This bit indicates whether the pull-up resistance of D+ is connected.
BYTE idata g_bHidI2CSInfo[2]	This is used for saving the information comes from I2C slave interface.
BYTE xdata I2CM_DEV_ADDR; BYTE xdata I2CM_MAP_ADDR[2]; BYTE xdata I2CM_DATA_BUF[16]; BYTE xdata I2CM_DATA_LEN; BYTE xdata I2CM_16BIT_MODE;	These are defined in the xdata memory and offered an access interface for the users to control I2C master communication.
WORD g_wIrTemp	This is used to represent the latest IR code data.
WORD code BaudRateTbl[6][5]	This is used to set UART baud rate at different MCU clocks.

7.3 Macros

Below macros are used for USB status check and set.

Macros	Description
UsbEnableUsb()	Enable USB engine
UsbResetAllEp ()	Reset the FIFO pointer of all endpoints
UsbResetEpCtrl ()	Reset the FIFO pointer of the control endpoint
UsbResetEpInt ()	Reset the FIFO pointer of the interrupt endpoint
UsbResetEpBlk4 ()	Reset the FIFO pointer of the bulk endpoint 4

UsbResetEpBlk5 ()	Reset the FIFO pointer of the bulk endpoint 5
UsbResetEpBlk7 ()	Reset the FIFO pointer of the bulk endpoint 7
UsbResetEpBlk8 ()	Reset the FIFO pointer of the bulk endpoint 8
UsbResetEpIsoInFb ()	Reset the FIFO pointer of the feedback endpoint
UsbSelectEp()	Select a specific endpoint to control or check the status
UsbStartEp()	Activate the selected endpoint
UsbStartEp()	Deactivate the selected endpoint
UsbEventResume()	Check the USB resume event
UsbEventRst()	Check the end of USB reset event
UsbEventSuspend()	Check the USB suspend event
UsbEventEpCtrl()	Check the endpoint 0(control) event
UsbEventEpInt()	Check the endpoint 3(interrupt) event
UsbEventEpBlk4()	Check the endpoint 4(bulk) event
UsbEventEpBlk5()	Check the endpoint 5(bulk) event
UsbEventEpBlk7()	Check the endpoint 7(bulk) event
UsbEventEpBlk8()	Check the endpoint 8(bulk) event
UsbEventEpIsoInFb()	Check the feedback endpoint event
UsbClrEventEpBlk4()	Clear the endpoint 4(bulk) event
UsbClrEventEpBlk5()	Clear the endpoint 5(bulk) event
UsbClrEventEpBlk7()	Clear the endpoint 7(bulk) event
UsbClrEventEpBlk8()	Clear the endpoint 8(bulk) event
UsbClrEventEpIsoInFb()	Clear the feedback endpoint event
UsbClrEventEpCtrl()	Clear the endpoint 0(control) event
UsbClrEventEpInt()	Clear the flag of the endpoint 3 event
UsbClrEventResume()	Clear the flag of USB resume event
UsbClrEventRst()	Clear the flag of the end of USB reset event
UsbClrEventSuspend()	Clear the flag of USB suspend event
UsbSetAddrEn()	Activate the USB device address
UsbSetAddress(addr)	Set addr as an effective USB device address
UsbSetConfig()	Indicate the device is configured
UsbClrConfig()	Clear the flag that indicates the device is configured
UsbGetConfig()	Report the status of the configuration flag
UsbSetRemoteWakeup()	Set the remote-wake-up enable bit
UsbClrRemoteWakeup()	Clear the remote-wake-up enable bit
UsbGetRemoteWakeup()	Report the status of the remote-wake-up enable bit
UsbSetDir()	Set the endpoint direction bit
UsbSetStall()	Send a STALL to the host

UsbSetTxReady()	Set TX Packet Ready Control Bit
UsbSendStallComplete()	Check if the host has answered the STALL
UsbRcvSetupPkt()	Check if SETUP data package is received
UsbRcvOutData()	Check if the OUT data package is received
UsbSendDataComplete()	Check if the transmission of IN data is completed
UsbClrDir()	Clear the endpoint direction bit
UsbClrStall()	Clear the Stall Handshake Request bit
UsbClrRxSetup()	Clear the flag that indicates the SETUP data package is received
UsbClrRxOutB0()	Clear the flag that indicates the OUT data is received
UsbClrTxCmpl()	Clear the flag that indicates the IN data is transmitted

7.4 Functions

Below functions are used to offer fundamental controls, include the initialization of MCU, anti-pop procedure, codec settings, volume adjustment, audio path selection, sampling rate and resolution control, GPIO initialization, and peripheral IO control.

Functions	Description
General functions	
void POR_Init()	This initialization is executed in the power on reset handler.
void HubInSuspend()	This function prevents surplus current from being consumed when connected to a hub that is in suspend mode.
void AntiPopProcedure()	This function controls analog modules to follow the anti-pop flow.
void AdcCodecReset(BOOL *ext_i2s*, BOOL *slv*, BOOL *mclk_out*, BOOL *dsp_mode*)	Configure ADC digital interface. *ext_i2s* -> 1: external I2S. 0: internal I2S to analog module. *slv* -> 1: slave mode. 0: master mode. *mclk_out* -> 1: output mclk. 0: do not output mclk. *dsp_mode* ->

	1: analog in -> I2S out -> DSP -> I2S in. 0: analog in.
void DacCodecReset(BOOL *ext_i2s*, BOOL *slv*, BOOL *mclk_out*, BOOL *dsp_mode*)	Configure DAC digital interface. *ext_i2s* ->: 1: external I2S. 0: internal I2S to analog module. *slv* ->: 1: slave mode. 0: master mode. *mclk_out* ->: 1: output mclk. 0: do not output mclk. *dsp_mode* ->: 1: I2S out -> DSP -> I2S in -> analog out. 0: analog in.
void C2MCopy()	Copy information from code memory to data memory.
void M2MCopy()	Copy information from data memory to data memory.
void MemZero()	Clear a specific data memory.
Audio functions	
void SPDIFOutMuteSwitch(BOOL mute)	Mute SPDIF out stream.
void SPDIFInMuteSwitch(BOOL mute)	Mute SPDIF in stream.
void PlaybackMuteControl(BOOL *analog*, BOOL *mute*)	Mute or unmute playback. *analog*: 1: analog module. 0: external I2S interface.
void RecordMuteControl(BOOL *analog*, BOOL *mute*)	*analog*: 1: analog module. 0: external I2S interface.
void MicInAaMuteControl(BOOL mute)	Mute or unmute the side tone from microphone.
void LineInAaMuteControl(BOOL mute)	Mute or unmute the side tone from line-in.
void PlaybackMinVolHandle(BOOL	Mute playback when the volume reaches the minimum level.

analog, BYTE ch)	
void PlaybackVolControl(BOOL *analog*, BYTE *ch*, BYTE *vol*)	Control playback volume. *analog* -> 1: external I2S. 0: analog module. *ch* -> 1: left channel. 0: right channel. *vol* -> in dB.
void RecordVolControl(BOOL *analog*, BOOL *mic*, BYTE *ch*, BYTE *vol*)	Control recording volume. *analog* -> 1: external I2S. 0: analog module. *ch* -> 1: left channel. 0: right channel. *vol* -> in dB.
void AApathVolControl(BOOL *mic*, BYTE *ch*, BYTE *vol*)	Control the volume of the side tone. *mic* -> 1: the side tone from microphone. 0: the side tone from line-in.
void RecordfromSPDIF(BOOL en)	Switch the recording source to SPDIF in when SPDIF in is enabled.
void SelectRecPath(BYTE *source*)	Select the recording path. *source* -> 1: microphone. 2: line-in. 4: stereo mixer in.
void SetPlaySampleRate(BYTE *sr*)	Control playback sampling rate. *sr* -> 0: 8000Hz. 1: 11025Hz. 2: 16000Hz. 3: 22050Hz. 4: 32000Hz. 5: 44100Hz. 6: 48000Hz. 7: 88200Hz.

	8: 96000Hz.
void SetPlayBitResolution(BOOL *bit16*)	Select the bit resolution of playback. *bit16* -> 1: 16 bits. 0: 24 bits.
void SetRecordSampleRate(BYTE *sr*)	Control recording sampling rate. *sr* -> 0: 8000Hz. 1: 11025Hz. 2: 16000Hz. 3: 22050Hz. 4: 32000Hz. 5: 44100Hz. 6: 48000Hz. 7: 88200Hz. 8: 96000Hz.
void SetRecordBitResolution(BOOL *bit16*)	Select the bit resolution of recording. *bit16* -> 1: 16 bits. 0: 24 bits.
void RecordMonoEnable(BOOL mono)	Enable recording mono path.
GPIO functions	
void GpioPuPdSetting(WORD *dir_mask*, BOOL *r_type*)	This function is used to decide GPIO internal circuit is connected to pull-up or pull-down resistor. *dir_mask* -> GPI mask. *r_type* -> 1: pull-down. 0: pull-up.
void GPIO_Init(WORD *dir_mask*, WORD *int_en*, BOOL *remote_wakeup_en*)	Initialize the GPIO settings. *dir_mask* -> GPI mask. *int_en* -> enable interrupt. *remote_wakeup_en* -> enable remote wake up.
Peripheral functions	
void UartBaudRateInit()	Initialize the baud rate counter.
void RefreshWdt()	Enable and refresh watch dog timer.
void LED_Config(BYTE led1, BYTE	Configure the specific GPIO to be used as LED

led2, BYTE led3)	1/2/3.
void LED1_Switch(BOOL open)	Control LED 1 switch.
void LED2_Switch(BOOL open)	Control LED 2 switch.
void LED3_Switch(BOOL open)	Control LED 3 switch.
BYTE InternSpiReadByte(BYTE addr)	Read registers of internal codec via SPI interface.
void InternSpiWriteByte(BYTE addr, BYTE value)	Write registers of internal codec via SPI interface.
void I2cMasterRead()	Read data via I2C interface in master mode.
void I2cMasterWrite()	Write data via I2C interface in master mode.
void PDSW_Control(BOOL en)	This handler is used to power on/down peripheral devices.
void ModIrReset(BYTE *type*)	IR decoder mode reset. Parameter 0 "type" can be: 0: NEC. 1: RC5. 2: RC6.
BOOL HandleI2cSlave()	When being in I2C slave mode, this function can be used to conduct I2C commands from other devices.
BOOL HandleSpiSlave()	When being in SPI slave mode, this function can be used to conduct SPI commands from other devices.

8. Interface of Customized Application

In “Framework\CM65xxB-1_xxx\”, there are 7 basic files put in this folder:

main.c -> includes power-on-reset function, main loop control for run time, etc.

int.c -> external interrupt service routines.

io.c -> peripheral interface control functions.

usb.c -> USB stage control functions.

request.c -> functions that handle USB requests.

audio.c -> functions that handle USB audio class-specific requests.

dscr.a51 -> descriptors of device, configuration, strings.

Customers can modify these files to develop their own product firmware.

8.1 Entry-point Function

The entry-point function will be executed immediately and the customization will start after ROM code finishes downloading the firmware from EEPROM to internal program memory. This function must be implemented.

Functions	Description
void main()	The entry-point function. The ROM program will jump to this function for the customized FW.

8.2 Interrupt Handler Functions

The 8051 and its derivatives provide a number of hardware interrupts that may be used for counting, timing, detecting external events, and sending and receiving data using the serial interface. The standard interrupts found on an 8051 are listed in the following table:

Interrupt number	Name	Description	Address
0	?INT0_ISR	External int 0	8000h
1	?TIMER0_ISR	Timer/counter 0	8003h
2	?INT1_ISR	External int 1	8006h
3	?TIMER1_ISR	Timer/counter 1	8009h
4	?UART_ISR	Serial port	800ch

Once the hardware interrupt event happens, the corresponding function will be called and handle the related tasks.

Functions	Description
void INT0_ISR ()	Customized external INT0 interrupt handler
void TIMER0_ISR ()	Customized Timer0 interrupt handler
void TIMER1_ISR ()	Customized Timer1 interrupt handler
void INT1_ISR ()	Customized external INT1 interrupt handler
void UART_ISR ()	Customized Uart interrupt handler

8.3 Customized Functions

Some functions need to be implemented in customized program.

Functions	Description
main.c	
void PowerOnReset()	The function for power-on initialization: 1. System clock configuration 2. Interrupt configuration 3. GPIO configuration 4. Global variable initialization
Void Timer1Config()	Reload the counter of Timer1
void InputReportDataReady()	Pack the HID report ID 0 Data (fixed size: 16bytes) The HID report data format is defined. Please refer to the appendix B.
void HandlePeriHidReport(BYTE *peri_type*)	Call this function will send the HID report to the host; the parameter *peri_type* notices the host which HID event is triggered.
void PeriClkReset()	Peripheral clock reset and gating (optional) 1. IR clock gated 2. SPDIF IN clock gated 3. SPDIF OUT clock gated Playback or Record logic clock gated
void OriginInitialize()	GPIO/LED initialization, Timer1 and system clocks configuration, and peripheral IO interrupts setting.

void OriginReset()	This is executed when receiving USB reset.
void OriginConfig()	Power on the peripheral components.
void OriginEvent()	TBD.
void OriginGpi()	Handle GPI interrupt events.
void OriginIr()	Report the IR event by sending a HID report with a relative bit set.
void variablesInit()	Initialize variables.
void AnaCodecPowerReset()	Enable codec modules: 1. DAC. 2. ADC. 3. AA-path.
void usbEpReset()	Enable USB endpoints, include: 1. control endpoint. 2. iso out endpoint. 3. iso in endpoint. 4. interrupt endpoint.
void CodecReset()	Configure codec, I2S interface, and recording path.
void volumeReset()	Set volume ranges, current volumes, and mute states.
void HandleUsbReset()	The USB reset handler This function handles the initialization of global variables, USB EP reset and the codec reset, etc.
void HandleUsbSuspend()	The USB suspend handler This function handlers the configuration of peripheral and codec, and the anti-pop noise flow is processed. Finally, the USB device will go into a low-power mode.
void HandleUsbResume()	The USB resume handler This function handles the anti-pop noise control and the codec reset.
usb.c	
void HandleUsbCtrlTransfer()	This function handles the USB control transfer.

void SubmitUsbIntTransfer()	Submit HID input report to the host.
void HandleUsbIntTransfer()	Execute this to end USB interrupt transfer.
request.c	
void acquireDeviceDscrData()	Update the data buffer which be sent to report the device descriptor.
void acquireConfigDscrData()	Update the data buffer which be sent to report the configuration descriptor.
void acquireStringDscrData()	Update the data buffer which be sent to report the string descriptor.
void acquireHidReportDscrData()	Update the data buffer which be sent to report the HID report descriptor.
void acquireHidDscrData()	Update the data buffer which be sent to report the HID descriptor.
BOOL tackleGetDescriptor(BOOL dataStage)	This function handles reporting descriptor information.
BOOL tackleSetAddress()	Executed when receiving Set_Address request from the host.
void epIntClearHalt()	Executed when the host attempts to clear the halt state of the interrupt endpoint.
BOOL tackleSetConfiguration()	Executed when receiving Set_Configuration request from the host.
BOOL tackleSetInterface()	Executed when receiving Set_Interface request from the host.
BOOL tackleGetConfiguration()	This function reports the configuration status of the device.
BOOL tackleGetInterface()	This function reports the alternate setting of the specific interface.
BOOL tackleSetFeature()	Executed when receiving Set_Feature request from the host, now only supports Remote_Wake_Up for the device level and Halt for the endpoint level.
BOOL tackleClearFeature()	Executed when receiving Clear_Feature request from the host, now only supports Remote_Wake_Up for the device level and Halt for the endpoint level.

BOOL tackleGetStatus()	Executed when receiving Get_Status from the host, the recipient includes devices, interfaces, and endpoints.
BOOL tackleHidGetReport()	This function handles HID_GET_REPORT.
BOOL tackleHidSetReport()	This function handles HID_SET_REPORT.
BOOL tackleHidSetReportData()	This function handles HID_SET_REPORT.
BOOL tackleClassCommand(BOOL dataStage)	This function handles the class request command.
BOOL tackleVendorCommand(BOOL dataStage)	This function handles the vendor request command.
BOOL TackleControlRequest(BOOL dataStage)	This function handles the USB control request.
audio.c	
void freq2CtrlByte()	This function transforms the 3-byte sampling rate into 1 byte according to the mapping table.
void StartSpdifOut()	Enable SPDIF out channel.
void OriginSetFrequency1()	This is used to change the sampling rate of the playback channel.
void OriginSetFrequency2()	This is used to change the sampling rate of the recording channel.
void OriginFeatureMute()	Mute feature unit control.
void OriginFeatureVolume()	Volume feature unit control.
void OriginFeatureAgc()	Enable/Disable REC AGC and clipping LED.
void OriginSetSelector()	ADC input source selection: 1. MIC In 2. Line In 3. SPDIF In 4. Stereo Mixer In
void OriginRecordMute()	Mute control for the recording channel.

BOOL tackleSelector(BOOL dataStage)	This function handles Get/Set commands of the selector unit.
BOOL tackleFeatureVolume(BOOL dataStage)	This function handles Get/Set volume commands of the feature unit.
BOOL tackleFeatureMute(BOOL dataStage)	This function handles Get/Set mute status commands of the feature unit.
BOOL TackleAudioControl(BOOL dataStage)	This is used to handle the control of audio units or entities.
BOOL TackleAudioStream(BOOL dataStage)	This function handles Get_Sampling_Rate/Set_Sampling_ Rate commands.
BOOL TackleSetAudioInterface()	This function is used to start or stop the playback and recording channels.
int.c	
static BYTE intSpiRead(BYTE reg)	This is used to execute read action of the internal SPI interface.
static void intSpiWrite(BYTE reg, BYTE temp)	This is used to execute write action of the internal SPI interface.
void HandleInt0Isr()	The handler of INT0_ISR().
void HandleTimer0Isr()	The handler of TIMER0_ISR().
void HandleInt1Isr()	The handler of INT1_ISR().
void HandleTimer1Isr()	The handler of TIMER1_ISR().
void HandleUartIsr()	The handler of UART_ISR().
void INT0_ISR()	The interrupt routine service of INT 0.
void TIMER0_ISR()	The interrupt routine service of TIMER 0.
void INT1_ISR()	The interrupt routine service of INT 1.
void TIMER1_ISR()	The interrupt routine service of TIMER 1.
void UART_ISR()	The interrupt routine service of UART.
io.c	

void PeriIntInit()	Enable interrupts of peripheral IOs.

9. Configuration definition

In “Framework\CM65xxB-1_xxx\inc\config.h”, there are macros for configuring the device. They are listed below.

Macros	Description
VENDOR_ID	Define vendor ID of the USB device.
PRODUCT_ID	Define product ID of the USB device.
VERSION_ID	Define version ID of the USB device.

10. Link file

The link file “cm65xx_cmd.lin” is a command file that may contain an *inputlist*, *outputfile*, and *directives*. The LX51 Linker/Locator uses this file to output the absolute object module.
Below definitions with red mark is for customization. The customized FW is located in the memory area.

```
.\output\audio.c.obj,.\output\dscr.a51.obj,.\output\int.c.obj,
.\output\io.c.obj,.\output\main.c.obj,.\output\request.c.obj,.\output\usb.c.obj,
..\CM65xxB-1\CM65xxB-1.LIB to .\output\cm65xxfw
CLASSES
(
    CODE(C:0x8000-C:0xBFFF),
    CONST(C:0x8000-C:0xBFFF),
    XDATA_EXTMEM(X:0x0A60-X:0x0BDF)
)

&

SEGMENTS
(
    ?STACK(D:0xE9),
    ?PR?INT0_ISR?INT(C:0x8000),
    ?PR?TIMER0_ISR?INT(C:0x8003),
    ?PR?INT1_ISR?INT(C:0x8006),
    ?PR?TIMER1_ISR?INT(C:0x8009),
    ?PR?UART_ISR?INT(C:0x800C),
    ?PR?MAIN?MAIN(C:0x8012)
)
```

11. Programming notes

1. Stack memory usage
   - Key factors
     i. Stack size initialization can be defined by customers. Customers should be aware of this to avoid memory overlap.
     ii. Function call level: More function call level affects the stack memory usage.
   - Stack overflow check
     i. Review the compiler log and map file.
     ii. Use KeilC simulator to check max stack pointer (sp_max) in specific functions.
     iii. Fill known characters in the bottom of stack, and run real-time check if the context is polluted.
     iv. Put the stack pointer in a global variable inside ISRs to check if stack overflow happens.
     v. Print out the stack pointer information in some critical functions using HID or Uart.
2. Interrupt handler usage
   - Do **NOT** enable the event of endpoint interrupt. Please use the polling mechanism to check the endpoint status instead.
   - There are five hardware interrupts supported. They are open for customization. However, customers can define their own tasks inside the interrupt handler functions. Suggest not to change the “INT0” and “INT1” handlers.
3. Standalone mode
   - If no USB audio design is required, customers can use CM65xx as a standalone device without USB functions.
4. Anti-pop noise
   - The standard anti-pop noise control function is included in CM65xx standard library.
5. USB reset impact
   - Most internal registers are reset once USB reset event happens. The codec registers are not affected. Remember to set some critical registers after USB reset.
   - USB reset will impact the GPIO temporary state. Please disable GPIO debounce function if the GPI state check is required to during USB reset event.

6. C51 lib code
    - The C51 library “C51S.LIB” is required for ANSI C function calls. Therefore, if some ANSI C functions are included, the code size of the object codes of C51 library will increase.
7. Extra xdata memory usage
    - In addition to the internal 256-byte data memory, extra xdata memory area can be used. Please refer to the memory mapping. For example, the memory for bulk transfer which start address is 0xA60 can be used if no bulk transfer is required.
8. Extended GPIO usage
    - GPIO 16~23 is controlled by MCU Port1. GPIO 24~31 is controlled by MCU Port2.
    - The IO pin needs to be pulled high if high-level drive is required.

12. Design an USB Audio Device

With CM65xx firmware platform, it is easy to build up a new USB audio device. Firmware engineers have to perform only several main tasks:

(1) Building up all USB descriptors for full-speed. The descriptor file “dscr.a51” in “Framework\CM65xxB-1_xxx\” can be as a reference.

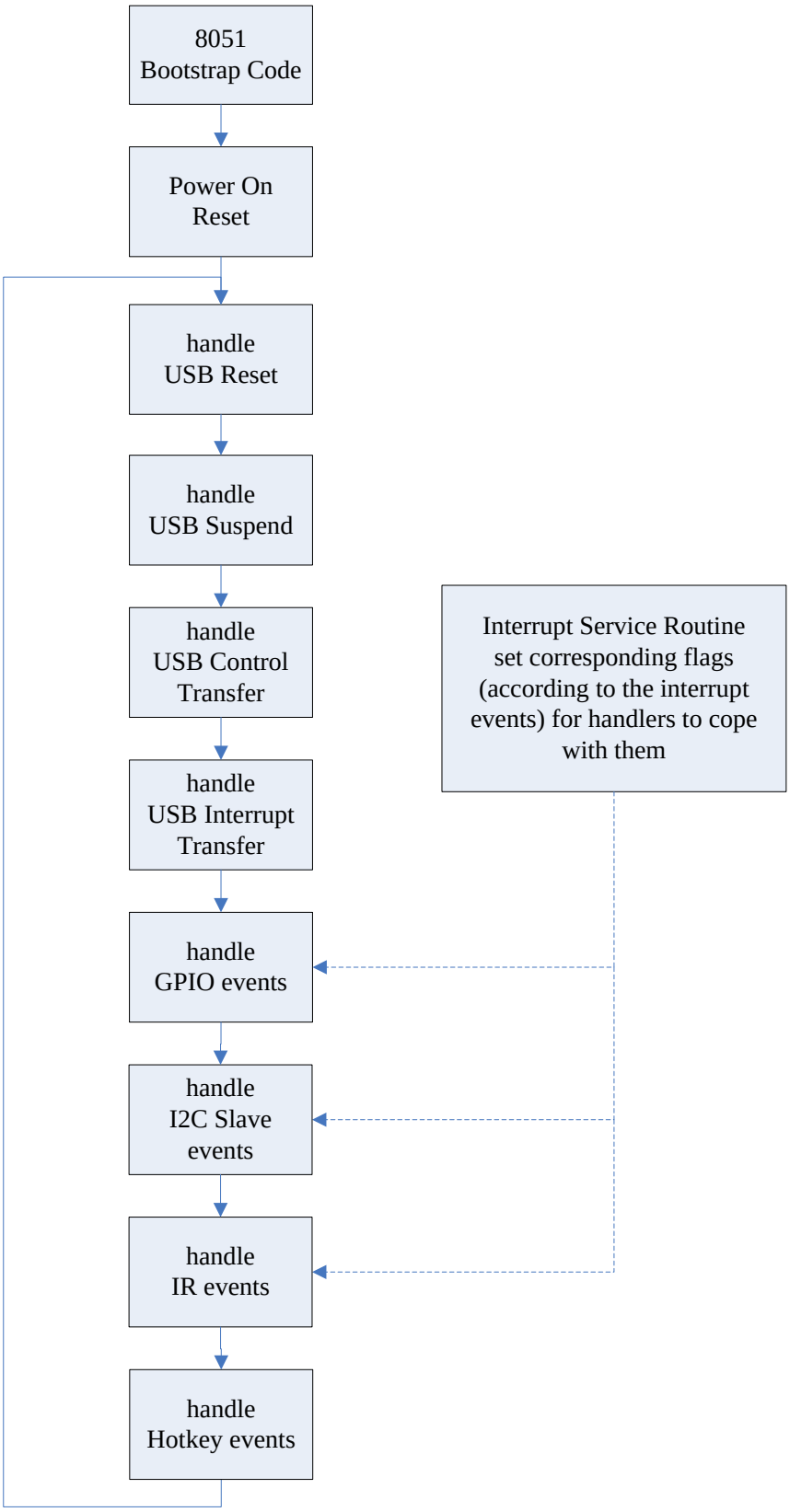
(2) Creating an instance of USB_DEVICE structure corresponding to the USB descriptors. "main.c” is an entry file of how USB_DEVICE is created.

(3) Implementing all customized functions that CM65xx framework needs. Firmware engineers can modify source files in sample application.

Then designer can get hex files for loading and testing by executing the batch file of building firmware.

In “Framework\CM65xxB-1_xxx\”, the files “audio.c” and “request.c” implement protocols of USB audio class 1.0 and internal codec control. If the topology or the configuration of the audio device is changed, these files are necessary to be modified to meet the USB descriptors. If CM65xx cooperates with different I2S codec, firmware engineers have to modify “audio.c” for the new codec.

13. The Example of Customized Firmware Flow



14. Register Table

➢ **CM65xx Register Table**

Address	Register Description
0x00	USB Global Control Register
0x01	USB Address Register
0x02	USB Global Interrupt Register
0x03	USB Global Interrupt Enable Register
0x04	USB Endpoint Number
0x05	USB Endpoint X Control Register
0x06	USB Endpoint X Status and Control Register
0x07	USB Endpoint FIFO Reset Register
0x08	USB Endpoint Interrupt Register
0x09	USB Endpoint Interrupt Enable Register
0x0a	USB Endpoint X FIFO Data Register
0x0b	USB Endpoint X Byte Count Register
0x0c	USB Frame Number Low Register
0x0d	USB Frame Number High Register
0x0e	COEDC Play_rate + Record_rate
0x0f	Resolution 24-bit Enable of play+ rec
0x10	Peripheral 1 Interrupt Register
0x11	Peripheral 1 Interrupt Enable Register
0x12	Peripheral 2 Interrupt Register
0x13	Peripheral 2 Interrupt Enable Register
0x14	ISO OUT FIFO Read Pointer Low Register
0x15	ISO OUT FIFO Read Pointer High Register
0x16	ISO OUT FIFO Write Pointer Low Register
0x17	ISO OUT FIFO Write Pointer High Register
0x18	ISO OUT FIFO Surplus Number Low Register
0x19	ISO OUT FIFO Surplus Number High Register
0x1a~0x1f	
0x20	Transfer FSM monitor
0x21	Transaction FSM monitor
0x22	MCU access FIFO mapping setting
0x23~0x27	
0x28~0x29	Playback write pointer enable and index
0x2a~0x2b	Playback read pointer enable and index

0x2c~0x2d	Record write pointer enable and index
0x2e~0x2f	Record read pointer enable and index
0x30~0x33	I2C slave data register
0x34~0x35	I2C slave Status register
0x36	I2C slave memory address pointer (MAP)
0x37	I2C slave status register
0x38	
0x39~0x3b	Ext SPI data register
0x3c	Ext SPI control reg0
0x3d	Ext SPI control reg1
0x3e	Ext SPI interrupt
0x3f	
0x40	EQ Band1's Gain Register
0x41	EQ Band2's Gain Register
0x42	EQ Band3's Gain Register
0x43	EQ Band4's Gain Register
0x44	EQ Band5's Gain Register
0x45	EQ Disable & EQ_TEST_MODE Register
0x46	Access which band of EQ filter coefficient
0x47	Filter coefficient port
0x48	EQ sample period
0x49	SoundSwitch setting from software
0x4a	EQ mode gain setting
0x4b~0x4d	Reserved for EQ OP gain
0x4e~0x4f	
0x50	PLL TESTMODE offset_p
0x51	PLL TESTMODE offset_r
0x52	PLL TESTMODE offset_rp
0x53	UART baud rate select
0x54	PLL TESTMODE mode_rp_48m setting
0x55	Software scale setting
0x56	PLL adaptive function enable
0x57	Ext I2S format + slave mode setting
0x58	PLL powerdown and dacfilter disable
0x59	External I2S signal pin and DSP process setting
0x5a	Record source + test setting
0x5b	Playback left channel volume control setting

0x5c	Playback right channel volume control setting
0x5d	Record left channel volume control setting
0x5e	Record right channel volume control setting
0x5f	VR scale read + AGC control path select
0x60	SPDIF Function Control Register(1)
0x61	SPDIF Function Control Register(2)
0x62	SPDIF/IN interrupt status
0x63	SPDIF/OUT channel status (1)
0x64	SPDIF/OUT channel status (2)
0x65	SPDIF/OUT channel status (3)
0x66	SPDIF/IN channel status (1)
0x67	SPDIF/IN channel status (2)
0x68	SPDIF/IN channel status (3)
0x69	SPDIF/IN channel status (4)
0x6a	UART RX Data Register
0x6b	UART TX Data Register
0x6c	Digital microphone setting
0x6d	OFFSET_P Monitor
0x6e	OFFSET_R Monitor
0x70	EQ-ADC Band1's Gain Register
0x71	EQ-ADC Band2's Gain Register
0x72	EQ-ADC Band3's Gain Register
0x73	EQ-ADC Band4's Gain Register
0x74	EQ-ADC Band5's Gain Register
0x75	EQ-ADC Disable & EQ_TEST_MODE Register
0x76	EQ-ADC Access which band of EQ filter coefficient
0x77	EQ-ADC Filter coefficient port
0x78 ~0x7f	
0x80	(Master mode I^2C) Slave Device Address and Read/Write Control Register
0x81	I^2C Memory Address Pointer (MAP) of Slave Device
0x82	I^2C Memory Address Pointer (MAP2) of Slave Device
0x83 ~0x92	I^2C Data Register(master mode)
0x93	I^2C Control and Status Register 0
0x94	I^2C Control and Status Register 1
0x95	I^2C Download Control and Status

0x96	I^2C Clock Period change
0x97	
0x98~0x99	Internal SPI Data Register
0x9a	Internal SPI Control Register
0x9b~0x9f	
0xa0 ~ 0xa2	IR Logical One Make and Space
0xa3 ~ 0xa5	IR Logical Zero Make and Space
0xa6 ~ 0xa8	IR Header Code Make and Space
0xa9 ~ 0xab	IR Repeat Code Make and Space
0xac	IR_CTRL_0
0xad	IR_CTRL_1
0xae	IR_CTRL_2
0xaf ~ 0xb2	IR Receiver Data Buffer
0xb3	IR RC6 CTRL
0xb4	USB phy power+debug
0xb5	Clock Switch for MCU
0xb6	Module clock ctrl setting1
0xb7	Module clock ctrl setting2
0xb8	Module Reset ctrl setting1
0xb9	IO DC test and PDSW
0xba	CMA112_CODEC analog test and monitor pop noise signal
0xbb	GPIO IO strength setting
0xbc	PDSW/I2C/SPI/SPDIF IO strength setting
0xbd	I2S IO strength setting
0xbe	PLL charge pump current ctrl
0xbf	Testmode value and THD mode
0xc0 ~ 0xc1	GPO Data Register
0xc2 ~ 0xc3	GPI Data Register
0xc4 ~ 0xc5	GPIO Direction Control Register
0xc6 ~ 0xc7	GPIO Interrupt Enable Mask Register
0xc8 ~ 0xc9	GPIO De-bouncing Register
0xca	General Control
0xcb	Buzzer Freq1
0xcc	Buzzer Duty1
0xcd	Buzzer Freq2
0xce	Buzzer Duty2
0xcf	Buzzer On Time

0xd0	Buzzer Cycle Time
0xd1	LED1 Freq
0xd2	LED1 Duty
0xd3	LED1 On Time
0xd4	LED1 Cycle Time
0xd5	LED2 On Time
0xd6	LED2 Cycle Time
0xd7	LED3 On Time
0xd8	LED3 Cycle Time
0xd9	Bounding Option
0xda	GPO_[1:0] Switch Source
0xdb	GPO_[3:2] Switch Source
0xdc	GPO_[5:4] Switch Source
0xdd	GPO_[7:6] Switch Source
0xde	GPO_[9:8] Switch Source
0xdf	GPO_[11:10] Switch Source
0xe0	GPO_[13:12] Switch Source
0xe1	GPO_[15:14] Switch Source
0xe2~0xe3	GPI Remote Choose
0xe4	GPIO pull-up/down Control Register0
0xe5	GPIO pull-up/down Control Register1
0xe6	GPIO pull-up/down Control Register2
0xe7	GPIO pull-up/down Control Register3
0xe8	Share MCU port0
0xe8~0xef	
0xf0	AGC Global Control Register
0xf1	AGC ATTACK TIME CONTROL Register
0xf2	AGC RELEASE TIME CONTROL Register
0xf3	AGC HOLD TIME CONTROL Register
0xf4	AGC Threshold Control Register
0xf5	AGC FIXED GAIN CONTROL Register
0xf6	AGC Global Control Register
0xf7	AGC FIXED GAIN CONTROL
0xf8	AGC Simulate CONTROL
0xf9	AGC GAIN MAX Limit
0xfa	AGC GAIN MIN Limit
0xfb	Clipping LED Timing Control I

0xfc	Clipping LED Timing Control II
0xfd	
0xfe	HID interrupt in start address MSB
0xff	HID interrupt in start address

(POR) : reset by power on reset

USBCON – USB Global Control Register

Address: 00h

Bits	R/W	Bit Mnemonic	Description	default
7	R/W	USBE	**USB Enable Bit** 1 : enable. 0 : disable.	1'b0 (POR)
6	R/W	SUSPCLK	**Suspend USB Clock Bit** 1 : Disable 48 MHz clock input to USB Controller. 0 : Enable 48 MHz clock input to USB Controller.	1'b0
5	R/W	SDRMWUP	**Send Remote Wake-up Bit** 1 : enable. 0 : disable. Must set up RMWUPE of USBCON and equal 1 first. Clear up after 1T write.	1'b0
4			**Reserved**	1'b0
3	R	UPRSM	**Upstream Resume Bit (read only)** Set by hardware when SDRMWUP has been set and if RMWUPE is enabled. Clear up after read.	1'b0
2	R/W	RMWUPE	**Remote Wake-up Enable Bit** 1 : enable. 0 : disable. Note: Do not set this bit if the host has not set the DEVICE_REMOTE_WAKEUP feature for the device.	1'b0
1	R/W	CONFG	**Configuration Bit** When SET_CONFIGURATION Value is set up here.	1'b0
0	R/W	FADDEN	**Function Address Enable Bit** When SET_ADDRESS the Firmware must set 1.	1'b0

USBADDR – USB Address Register

Address: 01h

Bits	R/W	Bit Mnemonic	Description	default
7	R/W	FEN	**Function Enable Bit** 1 : enable. 0 : disable.	1'b0
6-0	R/W	UADD6:0	**USB Address Bits** It shall be written with the value set by a SET_ADDRESS request received by the device firmware.	7'b0

USBINT – USB Global Interrupt Register

Address: 02h

Bits	R/W	Bit Mnemonic	Description	default
7-6			**Reserved**	2'b0
5	R/W	WUPCPU	**Wake Up CPU Interrupt Flag** 1 : interrupt. 0 : none. Set by hardware when EWUPCPU of USBIEN has been set.	1'b0 (POR)
4	R/W	EORINT	**End of Reset Interrupt Flag** 1 : interrupt. 0 : none. Set by hardware when EEORINT of USBIEN has been set.	1'b0 (POR)
3	R/W	SOFINT	**Start of Frame Interrupt Flag** 1 : interrupt. 0 : none. Set by hardware when ESOFINT of USBIEN has been set.	1'b0
2-1			**Reserved**	2'b0
0	R/W	SPINT	**Suspend Interrupt Flag**	1'b0 (POR)

			1 : interrupt. 0 : none. Set by hardware when ESPINT of USBIEN has been set.	

USBIEN – USB Global Interrupt Enable Register

Address: 03h

Bits	R/W	Bit Mnemonic	Description	default
7-6	R/W		**Reserved**	1'b0
5	R/W	EWUPCPU	**Wake up CPU Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0
4	R/W	EEOFIN	**End Of Reset Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0 (POR)
3	R/W	ESOFINT	**Start Of Frame Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0
2-1			**Reserved**	2'b0
0	R/W	ESPINT	**Suspend Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0

UEPNUM – USB Endpoint Number

Address: 04h

Bits	R/W	Bit Mnemonic	Description	default
7-4	R/W	EPCF7:4	**USB Endpoint Number configuration** Can setting Endpoint number 0~15 Default setting: EP0(Control) : only = 0 EP1(ISO OUT) : number 1 EP2(ISO IN) : number 2 EP3(Interrupt IN) : number 7 (for compatibility)	4'h0

			EP4(Bulk OUT) : number 4 EP5(Bulk IN) : number 5 EP6(Feedback IN) : number 6 EP7(Bulk OUT) : number 8 EP8(Bulk IN) : number 9	
3-0	R/W	EPNUM3:0	**Endpoint Number Bits** Set this field with the number of the endpoint which shall be accessed when reading or writing to registers UEPSTAX, UEPDATX, UBYCTLX or UEPCONX.	4'h0

UEPCONX –USB Endpoint X Control Register (X = EPNUM set in UEPNUM)

Address: 05h

Bits	**R/W**	**Bit Mnemonic**	**Description**	**default**
7	R/W	EPEN	**Endpoint Enable Bit** 1 : enable. 0 : disable.	1'b0
6	R/W	EPx_DIR	**Endpoint Direction Bit** 1 : Set to configure IN direction. 0 : Set to configure OUT direction. This bit has no effect for Control endpoints. Only valid for EP4, EP5, EP7, EP8 Default EP4 = OUT EP5 = IN EP7 = OUT EP8 = IN	1'bX
5	R/W	EPx_ZLEN	BULK Endpoint zero length packet For EP4, EP5, EP7, EP8 as bulk in endpoint 1'b1 : send zero-length IN packet 1'b0 : IN packet length depended on wt_point Clear to 1'b0 after this zero-length IN packet is transmitted	1'b0
4	R/W		**Reserved**	1'b0
3	R			1'b0
2	R/W		**Reserved**	1'b0

1-0	R/W			2'b0

UEPSTAX – USB Endpoint X Status and Control Register (X = EPNUM set in UEPNUM)

Address: 06h

Bits	R/W	Bit Mnemonic	Description	default
7	R/W	DIR	**Control Endpoint Direction Bit** 1: data stage. 0: Clear otherwise. **NOTE:** When Endpoint Number=2, this bit can turn on valid record bit (VALID_REC).	1'b0
6	R/W	PLAY_REC	**Alternate Setting Bit (AUDIO Device)** 1 : enable. 0 : disable When Endpoint Number = 1 expresses it is PLAY. When Endpoint Number = 2 expresses it is RECORD.	1'b0
5	R/W	STALLRQ	**Stall Handshake Request Bit** 1 : enable. 0 : disable. Set to send a STALL answer to the host for the next handshake. Clear otherwise	1'b0
4	R/W	TXRDY	**TX Packet Ready Control Bit** 1 : enable. 0 : disable. Cleared by hardware when ACK packet was received.	1'b0
3	R/W	STLCRC	**Stall Sent Interrupt Flag/CRC Error Interrupt Flag** 1 : interrupt. 0 : none. Set by hardware after a STALL handshake has been sent as requested by STALLRQ. Cleared by hardware when a SETUP packet was received.	1'b0
2	R/W	RXSETUP	**Received SETUP Interrupt Flag**	1'b0

			1 : interrupt. 0 : none. Clear by software after reading the SETUP data from the endpoint FIFO.	
1	R/W	RXOUT	**Received OUT Data Interrupt Flag** 1 : interrupt. 0 : none. Clear by software after reading the OUT data from the endpoint FIFO.	1'b0
0	R/W	TXCMP	**Transmitted IN Data Complete Interrupt Flag** 1 : interrupt. 0 : none. Clear by software before setting again TXRDY.	1'b0

UEPRST –USB Endpoint FIFO Reset Register

Address: 07h

Bits	**R/W**	**Bit Mnemonic**	**Description**	**default**
7			**Reserved**	1'b0
6	R/W	EP6RST	**Endpoint 6 FIFO Reset** 1 : reset. 0 : none. Clear up after write.	1'b0
5	R/W	EP5RST	**Endpoint 5 FIFO Reset** 1 : reset. 0 : none. Clear up after write.	1'b0
4	R/W	EP4RST	**Endpoint4 FIFO Reset** 1 : reset. 0 : none. Clear up after write.	1'b0
3	R/W	EP3RST	**Endpoint 3 FIFO Reset** 1 : reset. 0 : none.	1'b0

			Clear up after write.	
2	R/W	EP8RST	**Endpoint 8 FIFO Reset** 1 : reset. 0 : none. Clear up after write.	1'b0
1	R/W	EP7RST	**Endpoint 7 FIFO Reset** 1 : reset. 0 : none. Clear up after write.	1'b0
0	R/W	EP0RST	**Endpoint 0 FIFO Reset** 1 : reset. 0 : none. Clear up after write.	1'b0

UEPINT – USB Endpoint Interrupt Register

Address: 08h

Bits	**R/W**	**Bit Mnemonic**	**Description**	**default**
7			**Reserved**	1'b0
6	R	EP6INT	**Endpoint 6 Interrupt Flag** 1 : interrupt. 0 : none.	1'b0
5	R	EP5INT	**Endpoint 5 Interrupt Flag** 1 : interrupt. 0 : none.	1'b0
4	R	EP4INT	**Endpoint 4 Interrupt Flag** 1 : interrupt. 0 : none.	1'b0
3	R	EP3INT	**Endpoint 3 Interrupt Flag** 1 : interrupt. 0 : none.	1'b0
2	R	EP8INT	**Endpoint 8 Interrupt Flag** 1 : interrupt.	1'b0

			0 : none.	
1	R	EP7INT	**Endpoint 7 Interrupt Flag** 1 : interrupt. 0 : none.	1'b0
0	R	EP03INT	**Endpoint 0 Interrupt Flag** 1 : interrupt. 0 : none.	1'b0

UEPIEN – USB Endpoint Interrupt Enable Register

Address: 09h

Bits	**R/W**	**Bit Mnemonic**	**Description**	**default**
7			**Reserved**	1'b0
6	R/W	EP6INTE	**Endpoint 6 Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0
5	R/W	EP5INTE	**Endpoint 5 Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0
4	R/W	EP5INTE	**Endpoint 4 Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0
3	R/W	EP3INTE	**Endpoint 3 Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0
2	R/W	EP8INTE	**Endpoint 8 Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0
1	R/W	EP7INTE	**Endpoint 7 Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0
0	R/W	EP03INTE	**Endpoint 0 Interrupt Enable Bit** 1 : enable.	1'b0

			0 : disable.	

UEPDATX – USB Endpoint X FIFO Data Register (X = EPNUM set in UEPNUM)

Address: 0ah

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	FDAT7:0	**Endpoint X FIFO Data**	7'b0

UBYCTX – USB Endpoint X Byte Count Register (X = EPNUM set in UEPNUM)

Address: 0bh

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	BYCT7:0	**Byte Count** Byte count of a received data packet.	7'b0

UFNUML– USB Frame Number Low Register(HW not ready)

Address: 0ch

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	FNUM7:0	**Frame Number** Lower 8 Bits of the 11-bit Frame Number. Not ready	7'b0

UFNUMH – USB Frame Number High Register (HW not ready)

Address: 0dh

Bits	R/W	Bit Mnemonic	Description	default
7-6			**Reserved**	2'b0
5	R/W	CRCOK	**Frame Number CRC OK Bit** 1 : ok. 0 : none. Not ready	1'b0
4	R/W	CRCERR	**Frame Number CRC Error Bit** 1 : error. 0 : none. Not ready	1'b0
3	R/W		**Reserved**	1'b0

2-0	R/W	FNUM10:8	**Frame Number** Upper 3 Bits of the 11-bit Frame Number. Not ready	3'b0

COEDC Play_rate + Record_rate

Address: 0eh

Bits	**R/W**	**Bit Mnemonic**	**Description**	**default**
7-4	R/W	R_FREQ	**Record Frequency Rate** 4'b1000 : 96 KHz. 4'b0111 : 88.2 KHz. 4'b0110 : 48 KHz. (default) 4'b0101 : 44.1 KHz. 4'b0100 : 32 KHz. 4'b0011 : 22.05 KHz. 4'b0010 : 16 KHz. 4'b0001 : 11.025 KHz. 4'b0000 : 8 KHz.	4'b0110
3-0	R/W	P_FREQ	**PLAY Frequency Rate** 4'b1000 : 96 KHz. 4'b0111 : 88.2 KHz. 4'b0110 : 48 KHz.(default) 4'b0101 : 44.1 KHz. 4'b0100 : 32 KHz. 4'b0011 : 22.05 KHz. 4'b0010 : 16 KHz. 4'b0001 : 11.025 KHz. 4'b0000 : 8 KHz.	4'b0110

Resolution 24-bit Enable of play+ rec

Address: 0fh

Bits	**R/W**	**Bit Mnemonic**	**Description**	**default**
7-2			**Reserved**	6'b0

1	R/W	RBR24E	**Record Bit Resolution 24-bits Enable** 1 : enable. (24-bits) 0 : disable. (16-bits)	1'b0
0	R/W	PBR24E	**Play Bit Resolution 24-bits Enable** 1 : enable. (24-bits) 0 : disable. (16-bits)	1'b0

PER1INT – Peripheral 1 Interrupt Register

Address: 10h

Bits	R/W	Bit Mnemonic	Description	default
7	R/W	TESTMODE_INT	**TEST MODE Interrupt Flag** 1 : interrupt. 0 : none. Clear up after write.	1'b0
6	R/W	VR_INT	**VR Interrupt Flag** 1 : interrupt. 0 : none. Clear up after write.	1'b0
5	R/W	GPI_INT	**GPI Interrupt Flag** 1 : interrupt. 0 : none. Clear up after write.	1'b0 (POR)
4	R/W	SPIS_INT	**SPI Slave Interrupt Flag** 1 : interrupt. 0 : none. Clear up after write.	1'b0 (POR)
3	R/W	SPI_INT	**SPI Interrupt Flag** 1 : interrupt. 0 : none. Clear up after write.	1'b0 (POR)
2	R/W	I2CS_INT	**I2C Slave Interrupt Flag** 1 : interrupt. 0 : none.	1'b0 (POR)

			Clear up after write.	
1	R/W	I2CM_INT	**I2C Master Interrupt Flag** 1 : interrupt. 0 : none. Clear up after write.	1'b0 (POR)
0	R/W	IR_INT	**IR Interrupt Flag** 1 : interrupt. 0 : none. Clear up after write.	1'b0

PER1INT – Peripheral 1 Interrupt Enable Register

Address: 11h

Bits	R/W	Bit Mnemonic	Description	default
7	R/W	TESTMODE_INTE	**TestMode Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0
6	R/W	VR_INTE	**VR Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0
5	R/W	GPI_INTE	**GPI Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0 (POR)
4	R/W	SPIS_INTE	**SPI Slave Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0 (POR)
3	R/W	SPI_INTE	**SPI Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0 (POR)
2	R/W	I2CS_INE	**I2C Slave Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0 (POR)
1	R/W	I2CM_INTE	**I2C Master Interrupt Enable Bit** 1 : enable.	1'b0 (POR)

			0 : disable.	
0	R/W	IR_INTE	**IR Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0

PER2INT – Peripheral 2 Interrupt Register

Address: 12h

Bits	R/W	Bit Mnemonic	Description	default
7			**Reserved**	1'b0
6	RO	SE1_INT	**SE1 Interrupt Flag** 1 : interrupt. 0 : none.	1'b0 (POR)
5	R/W	UART_W_INT	**UART TX Interrupt Flag** 1 : interrupt. 0 : none. Clear up after write.	1'b0
4	R/W	BAUD_INT	**BAUD rate Interrupt Flag** 1 : interrupt. 0 : none. Clear up after write.	1'b0
3	R/W	SPI2_INT	**Internal SPI Interrupt Flag** 1 : interrupt. 0 : none. Clear up after write.	1'b0
2	R/W	RATE_INT	**SPDIF Rate Interrupt Flag** 1 : interrupt. 0 : none. Clear up after write.	1'b0
1	R/W	LOCK_INT	**SPDIF Lock Interrupt Flag** 1 : interrupt. 0 : none. Clear up after write.	1'b0

0	R/W	SENSE_INT	**SPDIF Sense Interrupt Flag** 1 : interrupt. 0 : none. Clear up after write.	1'b0

PER1INT – Peripheral 2 Interrupt Enable Register

Address: 13h

Bits	R/W	Bit Mnemonic	Description	default
7	R/W	SE1_Clear	**SE1 interrupt Clear Bit** Clear up REG-12[6] after write.	1'b0 (POR)
6	R/W	SE1_INTE	**SE1 Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0 (POR)
5	R/W	UART_W_INTE	**UART TX Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0
4	R/W	BAUD_INTE	**BAUD rate Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0
3	R/W	SPI2_INTE	**Internal SPI Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0
2	R/W	RATE_INE	**SPDIF Rate Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0
1	R/W	LOCK_INTE	**SPDIF Lock Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0
0	R/W	SENSE_INTE	**SPDIF Sense Interrupt Enable Bit** 1 : enable. 0 : disable.	1'b0

ISOFRPL – ISO OUT FIFO Read Pointer Low Register

Address: 14h

Bits	R/W	Bit Mnemonic	Description	default
7-0	R	ISOFRP7:0	**ISO OUT FIFO Read Pointer** Lower 8 Bits of the 10-bits Read Pointer.	8'h0

ISOFRPH – ISO OUT FIFO Read Pointer High Register

Address: 15h

Bits	R/W	Bit Mnemonic	Description	default
7-4			**Reserved**	4'b0
3	RO	VALID_REC	ISO in valid	1'b0
2	RO	VALID_PLAY	ISO out valid	1'b0
1-0	R	ISOFRP9:8	**ISO OUT FIFO Read Pointer** Upper 2 Bits of the 10-bits Read Pointer. p.s. SOFRP9 is indicate ping-pong status, FIFO read pointer over 10'd400 the SOFRP9 will set invert oneself.	2'h0

ISOFWPL – ISO OUT FIFO Write Pointer Low Register

Address: 16h

Bits	R/W	Bit Mnemonic	Description	default
7-0	R	ISOFWP7:0	**ISO OUT FIFO Write Pointer** Lower 8 Bits of the 10-bits Write Pointer.	8'h0

ISOFWPH – ISO OUT FIFO Write Pointer High Register

Address: 17h

Bits	R/W	Bit Mnemonic	Description	default
7-2			**Reserved**	6'b0
1-0	R	ISOFWP9:8	**ISO OUT FIFO Write Pointer** Upper 2 Bits of the 10-bits Write Pointer. p.s. SOFWP9 is indicate ping-pong status, FIFO Write pointer over 10'd400 the SOFWP9 will set invert oneself.	2'h0

ISOFSNL – ISO OUT FIFO Surplus Number Low Register

Address: 18h

Bits	R/W	Bit Mnemonic	Description	default
7-0	R	ISOFSN7:0	**ISO OUT FIFO Surplus Number** Lower 8 Bits of the 10-bits Surplus Number.	8'h0

ISOFSNH –ISO OUT FIFO Surplus Number High Register

Address: 19h

Bits	R/W	Bit Mnemonic	Description	default
7-2			**Reserved**	6'b0
1-0	R	ISOFSN9:8	**ISO OUT FIFO Surplus Number** Upper 2 Bits of the 10-bits Surplus Number.	2'h0

Debug monitor

Address: 20h

Bits	R/W	Bit Mnemonic	Description	default
7-3			**Reserved**	5'b0
2-0	R	TFFSM	**Debug monitor** Transfer FSM monitor.	3'h0

Address: 21h

Bits	R/W	Bit Mnemonic	Description	default
7-6			**Reserved**	3'b0
5-0	R	TAFSM	**Debug monitor** Transaction FSM monitor.	5'h0

MCU access FIFO mapping setting

Address: 22h

Bits	R/W	Bit Mnemonic	Description	default
7-1			**Reserved**	5'b0
0	R/W	**FIFO_24bit_selected**	**MCU access fifo 3/2 byte setting**	1'b1

			1 : 24 bit mapping 0 : 16 bit mapping	(POR)

Playback Write pointer enable and index

Address: 28h~29h

Bits	R/W	Bit Mnemonic	Description	default
15	R/W	REG_PLAY_WE	Enable register control playback write pointer.	0x0
14:10	RO		Reserved	0x00
9:0	R/W	REG_PLAY_WPT	Playback write pointer register.	0x00

Playback Read pointer enable and index

Address: 2ah~2bh

Bits	R/W	Bit Mnemonic	Description	default
15	R/W	REG_PLAY_RE	Enable register control playback read pointer.	0x0
14:10	RO		Reserved	0x00
9:0	R/W	REG_PLAY_RPT	Playback read pointer register.	0x00

Record Write pointer enable and index

Address: 2ch~2dh

Bits	R/W	Bit Mnemonic	Description	default
15	R/W	REG_REC_WE	Enable register control record write pointer.	0x0
14:11	RO		Reserved	0x00
10:0	R/W	REG_REC_WPT	Record write pointer register.	0x00

Record Read pointer enable and index

Address: 2eh~2fh

Bits	R/W	Bit Mnemonic	Description	default
15	R/W	REG_REC_RE	Enable register control record read pointer.	0x0
14:11	RO		Reserved	0x00
10:0	R/W	REG_REC_RPT	Record read pointer register.	0x00

I^2C Slave Register

~~Address: 30h~37h~~

~~Bits~~	~~R/W~~	~~Bit Mnemonic~~	~~Description~~	~~default~~

~~7-0~~	~~R/W~~	~~I2CS_DATA0~~	~~If external MCU wants to do the communication with internal MCU , it can draw on the 8 byte.~~	~~0x00~~

I2CS_DATA: 2-Wire serial bus data register

Address: 30~33h

Bits	R/W	Bit Mnemonic	Description	Default
31:0	R/W	MCU_data0~F	The data received from or transmitted to master device. This register can not be written when 2-wire slave serial bus status is busy.	0000h (POR)

I2CS_STATUS: 2-Wire serial bus status register

Address: 34~35h

Bits	R/W	Bit Mnemonic	Description	Default
15			Reserved	
14:12	R		Reserved	0h
11	R/W	Thld_int_mask	Threshold interrupt mask: 1: mask ; 0: non-mask ; default :0	0b (POR)
10	R	Write_data_ready	Interrupt happened, auto-cleared after read	0b (POR)
9	R/W	I2c_s_reset	0: 2-wire serial bus in normal operation (default) 1: 2-wire serial bus in reset state	0b (POR)
8	R/W	Dri_tran_st	**initiated transaction status** 1: The last initiated transaction failed, write 1 to clear.	0b (POR)
7	R/W	Rd_tran_st	**Read transaction status** 1: a new read transaction received, write 1 to clear.	0b (POR)
6	R/W	Wr_tran_st	**Write transaction status** 1: a new write transaction received, write 1 to clear.	0b (POR)
5:1	R	Data_len	The data length of the last write transaction received, 00000: 1 byte (MAP only) 00001: 2 byte (MAP + 1 byte data) 00010: 3 byte (MAP + 2 byte data) 00011: 4 byte (MAP + 3 byte data) 00100: 5 byte (MAP + 4 byte data) … 01111: 16 byte (MAP + 15 byte	0b (POR)

			data) . 10000: 17 byte (MAP + 16 byte data) Others: Reserved	
0	R	busy	The 2-wire serial bus status, 0: idle, 1: busy	0b (POR)

Note: When I2C issue interrupt to MCU, MCU need to read the data numbers that threshold data count specified. And wait another interrupt until the total data transfer completed.

I2CS_MAP: Memory address pointer (MAP)

Address: 36h

Bits	R/W	Bit Mnemonic	Description	Default
7:0	R/W	MCU_MAP	The memory address of the read or write transactions from MCU. Address 0 is reserved for initiated transaction.	00h (POR)

I2CS_STATUS: 2-Wire serial bus status register

Address: 37h

Bits	R/W	Bit Mnemonic	Description	Default
7	R/W	Sync_en	**Synchronization enable** 1: enable (the synchronization selection bit will decide the method adopted). 0: disable (MCU and ARC should guarantee no data lost themselves).	1b (POR)
6	R/W	Int_polarity	The polarity control of pin INT_OUT (initiated transaction interrupt), 0: high active, 1: low active	0b (POR)
5:4	R/W	Slave_addr	Slave Device Address 00: select 0001000 (10h) as slave address 01: select 0001001 (12h) as slave address 10: select 0001010 (14h) as slave address 11: select 0001011 (16h) as slave address	01b (POR)
3	R/W	Sync_sel	**Synchronization method selection** 1: Data synchronization. When this bit is one, if the current transaction has not been serviced by ARC, the clock line of the 2-wire serial bus will be pulled low. Under this situation, the MCU can not start a new transaction or continue the current read transaction until the clock line goes back to high. 0: Ready pin synchronization. If the MCU can not support open drain 2-wire serial bus, this bit should be set to zero. Under this situation, the MCU can	1b (POR)

			not start a new transaction or continue the current read transaction until the pin XSLAVE_RDY goes high to signal that the driver has serviced the current transaction. Driver should use "driver acknowledge" to signal the processing of the current transaction is completed.	
2	R/W	Int_mask	**Interrupt Mask** 0: interrupt will happen at a read/write transaction received or a driver initiated transaction failed 1: interrupt will not happen	0b (POR)
1	R/W	Dri_init_tran	**Driver initiated transaction** Write 1 to start Driver initiated transaction. This bit is cleared automatically, after ARC initiated transaction starts. The ARC initiated transaction should be issued only when the 2-wire slave serial bus is idle. Otherwise, it will be ignored. The ARC initiated transaction will cause pin INT_OUT to send out an interrupt for MCU. After MCU responded with a Write-MAP-Address-0- Only transaction and a subsequent read transaction, interrupt INT_OUT will be de-asserted. However, if the MCU does not act as what is expected (a write MAP-Address-0-Only transaction and a subsequent read transaction), the interrupt INT_OUT will be still de-asserted, but the ARC initiated transaction status is used to signal a fail status to ARC. In this case, the driver should consider to repeat the failed Driver initiated transaction again.	0b (POR)
0	R/W	ack	Driver Acknowledge means driver has processed the current transaction. Write 1 to acknowledge. This bit will be cleared automatically.	0b (POR)

SPI I/F Registers Descriptions for EXT

SPI Data Register

Address: 39h~3bh

Bits	R/W	Bit Mnemonic	Description	default
23-0	R/W	Data0	The data (which include address, r/w, and data bits) written to or read from the codec.	0x0000 0000

			The bits in this register should be interpreted according to the individual codec. The content of this register, after a write operation completes, has no meaning. The content of this register, after a read operation completes, should reference the document of individual codec to see how many bits in this register is valid. For example, if codec is Analog Device AD1837, then SPI_Data_Reg[9:0] will be valid data	(POR)

SPI Control Register 0

Address: 3ch

Bits	R/W	Bit Mnemonic	Description	default
7	R/W	slv_mst	SPI master/slave mode 0 : master mode 1: slave mode	1'b1 (POR)
6	R/W	long_mode	SPI slave address length 0: 1-byte address 1: 2-byte address	1'b1 (POR)
5			Reserved	1'b0(POR)
4	R/W	ra8815_rw	RA8815 3-wire mode R/W 0 : Write command 1: Read command	1'b0 (POR)
3	R/W	si_mode	Serial interface mode 0 : normal SPI mode 1: Serial interface mode	1'b0 (POR)
2	R/W	si_mode_rs	Serial interface RS/A0 output 0: RS/A0==0 for 8th bit 1: RS/A0==1 for 8th bit	1'b0 (POR)
1-0	R/W	data_len	The data length of read/write, 00: Reserved 01: 1 bytes 10: 2 bytes 11: 3 bytes	1'b0 (POR)

*Note:

1. Bit [4]: RA8815 3-wire mode is designated for RA8815 LCD controller.
2. Bit [3]: Serial interface mode is designated for other LCD controllers such as ST7565, NT7606, etc. It is write-only which the data cannot read out from LCD controller.
3. Bit [2]: In Serial interface mode, A0 is for ST7565; RS is for NT7606.

4. Bit [1:0]: Both RA8815 and Serial interface mode are two-byte in total length.

SPI Control Register 1

Address: 3dh

Bits	R/W	Bit Mnemonic	Description	default
7	R/W	spi_start	Trigger SPI read/write command 0->1: trigger SPI read/write command. 1->0: SPI interface had completed current task. 0 : SPI interface is idle and ready for work. 1 : SPI interface is running.	1'b0 (POR)
6	R/W	spi_lh_edge	SPI CEN control 0: codec latch control data at SPI clock low (default) 1: codec latch control data at SPI clock high	1'b1 (POR)
5			Reserved	
4-3	R/W	frq_sel	SPI clock period 00: 330 ns 01: 980 ns 10: 1300 ns 11: 1600 ns (1.6 micro-second)	2'b0 (POR)
2	R/W	first_leading_bit	First data bit of 2-bit leading mode	1'b0(POR)
1	R/W	second_leading_bit	Second data bit of 2-bit leading mode	1'b0(POR)
0	R/W	leading_bit_mode	RA8815 2-bit leading mode 0: No leading bits 1: 2-bit leading for each transaction	1'b0 (POR)

SPI Interrupt

Address: 3eh

Bits	R/W	Bit Mnemonic	Description	default
7	R/W	CPOL	Clock Polarity	1'b1(POR)
6	R/W	CPHA	Clock Phase	1'b1(POR)
5-4	R/W		Reserved	2'b0(POR)
3	RO	slv_hid	SPI slave flag to HID interrupt 0: access to internal register 1: flag to HID interrupt	1'b0 (POR)
2	RO	slv_rw	SPI slave read/write flag 0: read 1: write	1'b0 (POR)

1	R/W	slv_int_en	SPI slave interrupt 0: no interrupt 1: interrupt (Default) Ext MCU can program this bit to make slave mode interrupt	1'b1 (POR)
0	R/W	mst_int_en	SPI master interrupt enable 0: disable 1: enable (Default) Control HW to make master mode interrupt	1'b1 (POR)

*Note:

1. Bit [1]: When SPI interface is slave mode, SPI interrupt happened when bit [1] ==1, which is written by external MPU via SPI. Interrupt (HID) would be cleaned once address 0x10 was written.
2. Bit [0]: When SPI interface is master mode, SPI interrupt happened when bit [0] ==1 and every SPI master command completed. Interrupt (HID) would be cleaned once address 0x10 was written.

EQ Register Description for Playback

EQ Maximum Range

Bits	R/W	**Bit Mnemonic**	Description	Default
7-0	R/W		EQ Maximum range (defined by EEPROM now) MAX attributes can range from +31.75 dB (0x7F) down to 0.00 dB (0x00) in steps of 0.25 dB (0x01). 0x30 means +12 dB for MAX. MIN attributes is minus MAX. 0x30 means -12 dB for MIN.	

Gain Value (decimal)	Meaning
0	-12 dB
1	-11.5 dB
2	-11 dB
3	-10.5 dB
4	-10 dB
5	-9.5 dB
6	-9 dB
7	-8.5 dB

8	-8 dB
9	-7.5 dB
10	-7 dB
11	-6.5 dB
12	-6 dB
13	-5.5 dB
14	-5 dB
15	-4.5 dB
16	-4 dB
17	-3.5 dB
18	-3 dB
19	-2.5 dB
20	-2 dB
21	-1.5 dB
22	-1 dB
23	-0.5 dB
24	0 dB
25	+0.5 dB
26	+1 dB
27	+1.5 dB
28	+2 dB
29	+2.5 dB
30	+3 dB
31	+3.5 dB
32	+4 dB
33	+4.5 dB
34	+5 dB
35	+5.5 dB
36	+6 dB
37	+6.5 dB
38	+7 dB
39	+7.5 dB
40	+8 dB
41	+8.5 dB
42	+9 dB
43	+9.5 dB
44	+10 dB

45	+10.5 dB
46	+11 dB
47	+11.5 dB
48	+12 dB

EQ Band1's Gain Register

Address: 0x40

Bits	R/W	**Bit Mnemonic**	Description	Default
7-6	R/W		Reserved	0x0
5-0	R/W	Band1_ctrl_Mx	Band1's Gain : -3.5dB(default)	6'h11

EQ Band2's Gain Register

Address: 0x41

Bits	R/W	**Bit Mnemonic**	Description	Default
7-6	R/W		Reserved	0x0
5-0	R/W	Band2_ctrl_Mx	Band2's Gain : -3.5dB(default)	6'h11

EQ Band3's Gain Register

Address: 0x42

Bits	R/W	**Bit Mnemonic**	Description	Default
7-6	R/W		Reserved	0x0
5-0	R/W	Band3_ctrl_Mx	Band3's Gain : -3.5dB(default)	6'h11

EQ Band4's Gain Register

Address: 0x43

Bits	R/W	**Bit Mnemonic**	Description	Default
7-6	R/W		Reserved	0x0
5-0	R/W	Band4_ctrl_Mx	Band4's Gain : -3.5dB(default)	6'h11

EQ Band5's Gain Register

Address: 0x44

Bits	R/W	**Bit Mnemonic**	Description	Default
7-6	R/W		Reserved	0x0
5-0	R/W	Band5_ctrl_Mx	Band5's Gain : -3.5dB(default)	6'h11

EQ Disable & EQ_TEST_MODE Register

Address: 0x45

Bits	R/W	**Bit Mnemonic**	Description	Default
7			Reserved	

5-0	R/W	OP_gain_pre	OP gain	6'd24
1			Reserved	
0			Reserved	

【NOTE】EQ Disable only can be modified after USBRST and before sending audio data to RM5013

Registers from 0x47-0 ~ 0x47-b cannot be modified when playing audio data.

1. Write 0x46 to choose band number;
2. Write 0x47 to put 12 bytes coefficient

Access which band of EQ filter coefficient

Address: 0x46

Bits	R/W	**Bit Mnemonic**	Description	Default
7	R/W	Eq_test_mode	EQ_TEST_MODE	1'b0
6	R/W	Eq_off	EQ Disable	1'b1
5-4	R/W		Reserved	0x0
3	R/W	Load_org	Load coefficient of the selection of sound switch setting.	0x0
2-0	R/W	bandnum	Band number	3'b000

Filter coefficient port

Address: 0x47

Bits	R/W	**Bit Mnemonic**	Description	Default
7-0	R/W	Coeff_X	Band'x coefficient value	8'b0

EQ BandX Coefficient Registers Descriptions

EQ BandX's Coefficient A21 Register1

Address: 0x47-0

Bits	R/W	Description	Default
7-0	R/W	BandX's Coefficient A21 Register1(MSB)	TBD

EQ BandX's Coefficient A21 Register2

Address: 0x47-1

Bits	R/W	Description	Default
7-0	R/W	BandX's Coefficient A21 Register2	TBD

EQ BandX's Coefficient A21 Register3

Address: 0x47-2

Bits	R/W	Description	Default

7-0	R/W	BandX's Coefficient A21 Register3	TBD

EQ BandX's Coefficient A21 Register4

Address: 0x47-3

Bits	R/W	Description	Default
7-0	R/W	BandX's Coefficient A21 Register4(LSB)	TBD

EQ BandX's Coefficient A31 Register1

Address: 0x47-4

Bits	R/W	Description	Default
7-0	R/W	BandX's Coefficient A31 Register1(MSB)	TBD

EQ BandX's Coefficient A31 Register2

Address: 0x47-5

Bits	R/W	Description	Default
7-0	R/W	BandX's Coefficient A31 Register2	TBD

EQ BandX's Coefficient A31 Register3

Address: 0x47-6

Bits	R/W	Description	Default
7-0	R/W	BandX's Coefficient A31 Register3	TBD

EQ BandX's Coefficient A31 Register4

Address: 0x47-7

Bits	R/W	Description	Default
7-0	R/W	BandX's Coefficient A31 Register4(LSB)	TBD

EQ BandX's Coefficient S Register1

Address: 0x47-8

Bits	R/W	Description	Default
7-0	R/W	BandX's Coefficient S Register1(MSB)	TBD

EQ BandX's Coefficient S Register2

Address: 0x47-9

Bits	R/W	Description	Default
7-0	R/W	BandX's Coefficient S Register2	TBD

EQ BandX's Coefficient S Register3

Address: 0x47-a

Bits	R/W	Description	Default
7-0	R/W	BandX's Coefficient S Register3	TBD

EQ BandX's Coefficient S Register4

Address: 0x47-b

Bits	R/W	Description	Default
7-0	R/W	BandX's Coefficient S Register4(LSB)	TBD

EQ sample period

Address: 0x48

Bits	R/W	**Bit Mnemonic**	Description	Default
7-0	R/W	Sample_period	Gain tracking step, unit is one sample time	8'h0a

SoundSwitch setting from software

Address: 0x49

Bits	R/W	**Bit Mnemonic**	Description	Default
7-3	R/W		Reserved	0x00
2	R/W		SoundSwitch_sel. 1→means SoundSwitch value comes from software.	0x1
1-0	R/W		SoundSwitch[1:0] from software	0x0

EQ mode gain setting

Address: 0x4a

Bits	R/W	**Bit Mnemonic**	Description	Default
7-2	R/W		Reserved	0x0
1-0	R/W	Mode_sel	For access band gain in each mode 2'b00 : default 2'b01 : gaming 2'b10 : communication 2'b11 : movie **Please set this value first, then access 0x40 ~0x44 gain setting**	2'b0

PLL TESTMODE offset_p

Address: 50h

Bits	**R/W**	**Bit Mnemonic**	**Description**	**Default**
7-0	R/W	TESTOFT_P[8:1]	programming offset to debug PLL_P	8'h20

PLL TESTMODE offset_r

Address: 51h

Bits	R/W	Bit Mnemonic	Description	Default
7-0	R/W	TESTOFT_R[8:1]	programming offset to debug PLL_R	8'hBD

PLL TESTMODE offset_rp

Address: 52h

Bits	R/W	Bit Mnemonic	Description	default
7-4			Reserved	
3-0			Reserved	
7-4	R/W		Reserved	
3-2	R/W	TESTOFT_R[10:9]	programming offset to debug PLL_R	2'b10
1-0	R/W	TESTOFT_P[10:9]	programming offset to debug PLL_P	2'b01

UART baud rate select

Address: 53h

Bits	R/W	Bit Mnemonic	Description	default
2-0	R/W	Pllcodecreg03	UART baud rate select (Write this register will trigger BAUD_INT) 101: 57600Hz 100: 38400Hz 011: 19200Hz 010: 9600Hz 001: 4800Hz 000: 2400Hz	3'b101 (POR)

PLL TESTMODE mode_rp_48m setting

Address: 54h

Bits	R/W	Bit Mnemonic	Description	Default
7	R/W	OFFSET_REG_EN	Offset and mode can be controlled by register 1: The offset will be set from register setting.	1'b0
6			Reserved	
5	R/W	TMODE_P	TMODE_P	1'b0
4	R/W	TMODE_R	TMODE_R	1'b1

3-0			Reserved	

PLL Software scale setting

Address: 55h

Bits	R/W	Bit Mnemonic	Description	default
7-6			Reserved.	2'b0
5	R/W	SW_SCALE	Software scale enable 1:enable, 0:disable	1'b0
4:0	R/W	REG_SCALE	Scale value for PLL adjustment	5'h8

PLL adaptive function enable

Address: 56h

Bits	R/W	Bit Mnemonic	Description	default
7	R/W	PD_PLAY	Power-Down playback PLL.	1'b0
6	R/W	PD_REC	Power-Down record PLL.	1'b0
5:4	R/W	SW_USB	Control bits for USB switch: 2'b00: USB20 Off, USB11 Off. 2'b01: USB20 Off, USB11 On. 2'b10: USB20 On, USB11 Off. 2'b11: USB20 On, USB11 On.	2'b0 (POR)
3	R/W	HBSCALDN	Difference of Fifo read/write pointer judge value	1'b0
2	R/W	DISPLLAD	disable PLL HW adaptive adjustment; default = 1'b0	1'b0
1	R/W	PLLBINEN_R	PLL HW adaptive adjustment by binary tree enable when record;default = 1'b0 (disabled)	1'b0
0	R/W	PLLBINEN	PLL HW adaptive adjustment by binary tree enable when playback;default = 1'b0 (disabled)	1'b0

Ext I2S format + slave mode setting

Address: 57h

Bits	R/W	Bit Mnemonic	Description	Default
7	R/W	PLAY_SLV	Slave mode playback, use external BCLK and LRCK	1'b0
6	R/W	LJTFD_SLV_P	DAC slave mode Left-justified 1'b1 : left-justified	1'b0
5	R/W	LJTFD_P	DAC left justified format; 1'b0 : I2S format	1'b0
4	R/W	REC_SLV	Slave mode record, use external BCLK and LRCK	1'b0
3	R/W	LJTFD_SLV	ADC slave mode Left-justified 1'b1 : left-justified	1'b0

2	R/W	LJTFD	ADC left justified format; 1'b0 : I2S format	1'b0
1	R/W	FSX512P	Playback MCLK-to-LRCK ratio 1'b1 : I2S Mclk with 512fs; 1'b0 : with 256fs	1'b0
1	R/W	FSX512R	Record MCLK-to-LRCK ratio 1'b1 : I2S Mclk with 512fs; 1'b0 : with 256fs	1'b0

PLL powerdown and dacfilter disable

Address: 58h

Bits	R/W	Bit Mnemonic	Description	default
7:6	R/W	Codec_selp	Playback Codec Select ~~00: Select External HDA~~ 01: Select External I2S 10: Select Internal I2S 11: Select Internal I2S with DSP	2'b10
5:4	R/W	Codec_selr	Record Codec Select ~~00: Select External HDA~~ 01: Select External I2S 10: Select Internal I2S 11: Select Internal I2S with DSP	2'b10
3	R/W	PD_USB_PLL	Power down PLL used for USB	1'b0
2	R/W	PD_CODEC_PLL	Power down PLL used for codec	1'b0
1	R/W	PDBIAS	Power down PLL bias	1'b0
0	R/W	dac_disable	If dac filter is off but adc filter is on, set this bit to 1 let register be programmed by adc mclk.	1'b0

External I2S signal pin and DSP process setting

Address: 59h

Bits	R/W	Bit Mnemonic	Description	Default
7	R/W	ADCMK_EXT	Using Ext I2S ADC MCLK 1'b1: use ; 1'b0: doesn't	1'b0
6	R/W	ADC_MKEN	Ext I2S ADC master clock output enable 1'b1: enable; 1'b0:disable Include pin: ADC_MCLK	1'b0
5	R/W	ADC_BLKEN	Ext I2S ADC clock output enable, 1'b1:enable; 1'b0:disable Include pins: ADC_BCLK+ADC_LRCK	1'b0

4	R/W	ADC_DSPEN	Ext I2S ADC data output enable for DSP process, 1'b1 : enable; 1'b0:disable Include pin: ADC_DOUT	1'b0
3	R/W	DACMK_EXT	Using Ext I2S DAC MCLK 1'b1: use ; 1'b0: doesn't	1'b0
2	R/W	DAC_MKEN	Ext I2S DAC master clock output enable 1'b1: enable; 1'b0:disable Include pin: DAC_MCLK	1'b0
1	R/W	DAC_BLKEN	Ext I2S DAC clock output enable, 1'b1: enable ; 1'b0: disable Include pins : DAC_BCLK+DAC_LRCK	1'b0
0	R/W	DAC_DOEN	Ext I2S DAC data output enable, 1'b1: enable ; 1'b0: disable Include pin : DAC_DOUT	1'b0

Record source + test setting

Address: 5ah

Bits	**R/W**	**Bit Mnemonic**	**Description**	**default**
7	R/W	STAND_ALONE	Enable this bit if treated RM5023 as a stand-alone device.	1'b0
6	R/W	MONOCH	ADC Mono channel	1'b0
5	R/W	SPEEDUP	Speed up the clock for simulation	1'b0
4	R/W	ENBLTWAV	Playback HW building sine wave	1'b0
3	R/W	ADCMON_EN	ADC→ SPDIFO monitor ch enable; 1'b1 : enabl	1'b0
2	R/W	SFTMUTEPS	Soft_mute_bypass; 1'b1= by pass	**1'b1**
1	R/W	MUTE_FC	Mute SPDIF In	1'b0
0	R/W	DIGREC	Set this bit to 1 when record source is from spdif	1'b0

Playback left channel volume control setting

Address: 5bh

Bits	**R/W**	**Bit Mnemonic**	**Description**	**Default**
7	R/W	da_mute_l	Mute left channel for playback	1'b0
6:0	R/W	da_gain_l	-62~+0dB, 1dB/Step	7'b0111100

Playback right channel volume control setting

Address: 5ch

Bits	R/W	Bit Mnemonic	Description	Default
7	R/W	da_mute_r	Mute right channel for playback	1'b0
6:0	R/W	da_gain_r	-62~+0dB, 1dB/Step	7'b0111100

Record left channel volume control setting

Address: 5dh

Bits	R/W	Bit Mnemonic	Description	Default
7			Reserved	1'b0
6	R/W	ad_mute_l	Mute left channel for record	1'b0
5:0	R/W	ad_gain_l	-16~+12dB, 1dB/Step	6'b010000

Record right channel volume control setting

Address: 5eh

Bits	R/W	Bit Mnemonic	Description	Default
7			Reserved	1'b0
6	R/W	ad_mute_r	Mute right channel for record	1'b0
5:0	R/W	ad_gain_r	-16~+12dB, 1dB/Step	6'b010000

VR scale read and AGC control path

Address: 5fh

Bits	R/W	Bit Mnemonic	Description	Default
7	R/W	AGC_CTRL_ANA	AGC volume control path 1: AGC control analog gain 0: AGC control digital gain	1'b1
6	R/W	GPIO_CLK24M	Output 24MHz clock to GPIO_10 pin, providing CM7000 working clock, This bit is invalid if GPIO_10 used to be AGC clipping LED output. 1'b1=output 24MHz enable	1'b0
5:0	RO	VR_OUT	VR volume response from analog	1'b0

SPDIF Function Control Register(1)

Address: 60h

Bits	R/W	Bit Mnemonic	Description	default
7	R/W	MUTE_SPDIFO	Mute SPDIF-Out	1'b0
6	R/W	LOCK_PAR	LOCK_INTR trigger level 0: low to high, 1: high to low	1'b0
5	R/W	SENSE_PAR	SENSE_INTR trigger level 0: low to high; 1: high to low	1'b0
4	R/W	IG_SPDV	SPDIF/IN valid bit detect enabled.	1'b0

3	R/W	RATE_MASK	1: RATE_INT is un-masked	1'b0
2	R/W	LOCK_MASK	1: LOCK_INT is un-masked	1'b0
1	R/W	SENSE_MASK	1: SENSE_INT is un-masked	1'b0
0	R/W	SPDFLOOP	internal SPDIF/IN loopback to SPDIF/OUT	1'b0

SPDIF Function Control Register(2)

Address: 61h

Bits	R/W	Bit Mnemonic	Description	default
7	R/W	SPDI_COPY	Set to 1 if you want to enable copyright check function.	1'b0
6:5	R/W	SPDI_SMT	SPDIF input hysteresis 2'b00: 0 2'b01: 82mV(default) 2'b10: 153mV 2'b11: 282mV	2'b01
4	RO	rate_4844or96	Data sample rate (1: 96K 0: 44.1K or 48K). This bit is used in some device without correct sample frequency information (bit19:16)	1'b0
3	RO	VALID_IN	SPDIF-In Validity.	1'b0
2	R/W	SPDI_EN	enable SPDIF-In	1'b0
1	R/W	VALID	SPDIF-Out Validity.	1'b1
0	R/W	SPDO_EN	enable SPDIF-Out	1'b1

SPDIF IN Interrupt Status

Address: 62h

Bits	R/W	Bit Mnemonic	Description	default
7:6	R/W	SRC_NUM	01: one source number 00: Do not take into account	2'b0
5			Reserved	1'b0
4	RO	RATE_INT_	SPDIN smp rate change interrupt. Write 1 to clear	1'b0
3	RO	LOCK_INT_	Interrupt of SPDIN data is locked or not. Write 1 to clear	1'b0
2	RO	SENSE_INT_	Interrupt of SPDIN data is sensed or not. Write 1 to clear.	1'b0
1	RO	LOCK_STUS	indicate SPDIN data is locked or not	1'b0
0	RO	SENSE_STUS	indicate there is transition in SPDIN	1'b0

SPDIF Out Status Register(1)

Address: 63h

Bits	R/W	Bit Mnemonic	Description	default
7:6	R/W	LEN	11: indicates word length is 24 bits 10: indicates word length is 16 bits	2'b00

			00: Not indicated	
5:4	R/W	CHN_NUM	10: right channel for stereo channel format 01: left channel for stereo channel format 00: Do not take into account	2'b00
3	R/W	PRE	1: indicates filter pre-emphasis is 50/15us 0: indicates without pre-emphasis	1'b0
2	R/W	COPY	1: indicates no copyright is asserted 0: indicates copyright is asserted	1'b1
1	R/W	AUDIO	1: indicates data in Non-PCM Audio 0: indicates data is Linear PCM samples	1'b0
0	R/W	PRO	1: indicates Professional use 0: consumer use	1'b0

SPDIF Out Status Register(2)

Address: 64h

Bits	R/W	Bit Mnemonic	Description	default
7:0	R/W	CC[7:0]	Category Code; Programmed according to IEC standards	7'b0

SPDIF Out Status Register(3)

Address: 65h

Bits	R/W	Bit Mnemonic	Description	default
7:6	R/W	CGMS	Copy Generation Management System 00: copying is permitted without restriction	2'b00
5:4	R/W	ACC	Clock accuracy. 00: Level II	2'b00
3:0	R/W	OSF	Original sampling frequency. 0000: Not indicated	4'b0000

SPDIF In Status Register(1)

Address: 66h

Bits	R/W	Bit Mnemonic	Description	default
7-6	RO		Copy Generation Management System 00: copying is permitted without restriction	2'b0
5-4	RO		Clock accuracy. 00: Level II	2'b0
3	RO		1: indicates filter pre-emphasis is 50/15us 0: indicates without pre-emphasis	1'b0
2	RO		1: indicates no copyright is asserted 0: indicates copyright is asserted	1'b0
1	RO		1: indicates data in Non-PCM Audio 0: indicates data is Linear PCM samples	1'b0
0	RO		1: indicates Professional use 0: consumer use	1'b0

SPDIF In Status Register(2)

Address: 67h

Bits	R/W	Bit Mnemonic	Description	Default
7	RO		Programmed according to IEC standards	1'b0
6-0	RO		Category Code; Programmed according to IEC standards	7'b0

SPDIF In Status Register(3)

Address: 68h

Bits	R/W	Bit Mnemonic	Description	Default
7-4	RO		Original Sampling Frequency	4'b0000
3-0	RO		Sample Frequency 0000: 44.1KHz 0010: 48KHz 1000: 88.2KHz 1010: 96KHz 0001: Not indicated	4'b0000

SPDIF In Status Register(4)

Address: 69h

Bits	R/W	Bit Mnemonic	Description	Default
7-4	RO		Channel number	4'b0
3-0	RO		Source number	4'b0

UART RX Data Register

Address: 6ah

Bits	R/W	Bit Mnemonic	Description	Default
7-0	R/W	Pllcodecreg1a	UART RX Data Register	8'b0

UART TX Data Register

Address: 6bh

Bits	R/W	Bit Mnemonic	Description	Default
7-0	R/W	Pllcodecreg1b	UART TX Data Register	8'b0

Digital microphone setting

Address: 6ch

Bits	R/W	Bit Mnemonic	Description	Default

7	R/W	MASK_EQ_FS	Do not disable EQ function even if operated other than 44.1/48KHz. 1: EQ can support all sampling rate 0: EQ just support 44.1/48KHz	1'b0
6-3	R/W		Reserved	4'b0
2	R/W	DMICDIV2	Dig mic clock is 6.144M/2(48KHz) ; 5.6448M/2(44.1KHz)	1'b0
1	R/W	DMICDIV4	Dig mic clock is 6.144M/4(48KHz) ; 5.6448M/4(44.1KHz)	1'b0
0	R/W	DIGIMIC_EN	Digital microphone enable; 1'b1: enable; 1'b0 : disable	1'b0

PLL 98M offset monitor

Address: 6dh

Bits	R/W	Bit Mnemonic	Description	default
7-0	R	AOFF_P[7:0]	98.304MHz offset monitor	8'h80 (POR)

PLL 90M offset monitor

Address: 6eh

Bits	R/W	Bit Mnemonic	Description	default
7-0	R	AOFF_R[7:0]	90.3168MHz offset monitor	8'h10 (POR)

EQ Register Description for Record

The contents of Register 70~77h are same as Register 40~47h

(master mode)I^{2}C Slave Device Address and Read/Write Control Register (Power Reset)

Address: 0x80

Bits	R/W	Bit Mnemonic	Description	Default
7-1	R/W	SA_reg	The target slave device address.	0xA8(POR)
0	R/W	SA_reg	1: read, 0: write	1'b0(POR)

I^{2}C Memory Address Pointer (MAP) of Slave Device (Power Reset)

Address: 0x81

Bits	R/W	Bit Mnemonic	Description	Default
7-0	R/W	MAP_reg	The register low byte address of salve device to be read or written.	8'b0(POR)

I²C Memory Address Pointer (MAP2) of Slave Device (Power Reset)

Address: 0x82

Bits	R/W	Bit Mnemonic	Description	Default
7-0	R/W	MAP2_reg	The register high byte address of salve device to be read or written.	8'b0(POR)

I²C Data Register (Power Reset)

Address: 0x83 ~ 0x92

Bits	R/W	Bit Mnemonic	Description	Default
7-0	R/W	data0~ data15	The data read from or written to the slave device.	8'b0(POR)

I²C Control and Status Register 0 (Power Reset)

Address: 0x93

Bits	R/W	Bit Mnemonic	Description	Default
7-0	R/W	i2c_ctrl_reg1	Data length of read/write command 8'h1: 1 byte, minimum length 8'h2: 2 bytes … 8'h7: 7 bytes 8'h10: 16 bytes, maximum length Others: Reserved	0x14 (POR)

I²C Control and Status Register 1 (Power Reset)

Address: 0x94

Bits	R/W	Bit Mnemonic	Description	Default
7	R/W	i2c_start	Trigger I2C read/write command 0->1: trigger I2C read/write command. 1->0: I2C interface had completed current task. 1 : I2C interface is idle and ready for work.	1'b0 (POR)

			1 : I2C interface is running.	
6	R/W	i2c_reset	Reset I2C interface 0 : Not reset I2C interface 1 : Reset I2C interface	1'b0 (POR)
5	R/W	map_len	MAP length 0 : 8-bit MAP 1 : 16-bit MAP	1'b0 (POR)
4	R/W	clk_sync	Clock Synchronization 0: off 1: on, when slave pull-down SCLK, master would pause	1'b1 (POR)
3	R/W	fast_std	I2C speed mode 0 : Standard mode, 100kHz 1 : Fast mode, 400kHz	1'b0 (POR)
2	R/W	map_only	MAP only write command 0 : Write command. 1 : MAP only write command.	1'b0 (POR)
1	R/W	auto_rd	Auto read command 0 : Read command. 1 : Auto read command.	1'b1 (POR)
0	R	i2c_ctrl_reg2	Slave NACK error occur 2 : No error 3 : Slave NACK error occur	1'b0 (POR)

*Note: Write-MAP-Only: An operation which only writes the register MAP the salve device

I²C Download Control and Status (Power Reset)

Address: 0x95

Bits	/W	Bit Mnemonic	Description	Default
7	R/W	I2c_mas_sel	I2C master/slave select	1'b1 (POR)
6-4	R/W		Reserved	3'b000
3	RO	CHKSUM_ERR	Check sum Error 1. If in LD_PHASE, the check sum value was calculated by I2C load data. 2. If in CHK_PHASE, the check sum value was calculated by SRAM read content.	1'b0
2	RO	CHK_FINISH	CHECK phase done 1: finish download data CHECK	1'b0
1	R/W	CHK_PHASE	MCU select CHECK phase to read SRAM data for check-sum check. 1: enable (after disable LD_PHASE) 0: set 0 after complete	1'b0 (POR)
0	R/W	LD_PHASE	MCU select LOAD phase to access SRAM from download. 1: enable 0: set 0 after complete	1'b0 (POR)

I²C Clock Period Setting (Power Reset)

Address: 0x96

Bits	/W	Bit Mnemonic	Description	Default
7	W	CHG_ENABLE	MCU can program I2C clock; 1'b1: enable	1'b0 (POR)
6	R/W		Reserved	1'b0
5-0	W	CHG_FREQ	Set I2C-master clock period. The clock period=83.3*5*(CHG_FREQ+1) Ex: CHG_FREQ = 6'd48 I2C Clock Period=83.3*5*(48+1)=20408ns HW limitation CHG_FREQ >= 6'h3	6'h0 (POR)

SPI I/F Registers Descriptions for Internal Control

SPI Data Register

Address: 98h~99h

Bits	R/W	Bit Mnemonic	Description	default
15-0	R/W	Data0/1	The data (which include address, r/w, and data bits) written to or read from the codec. The bits in this register should be interpreted according to the individual codec. The content of this register, after a write operation completes, has no meaning. The content of this register, after a read operation completes, should reference the document of individual codec to see how many bits in this register is valid.	0x0000 (POR)

SPI Control Register

Address: 9Ah

Bits	R/W	Bit Mnemonic	Description	default
7	R/W	spi_start	Trigger SPI read/write command 0->1: trigger SPI read/write command. 1->0: SPI interface had completed current task. 4 : SPI interface is idle and ready for	1'b0 (POR)

			work. 1 : SPI interface is running.	
6:3	R/W		Reserved	1'b0(POR)
2:1	R/W	frq_sel	SPI clock period 00: 330 ns 01: 980 ns 10: 1300 ns 11: 1600 ns (1.6 micro-second)	2'b0 (POR)
0	R/W	mst_int_en	SPI master interrupt enable 0: disable (default) 1: enable ~~(default)~~ Control HW to make master mode interrupt	1'b0 (POR)

*Note:

1. Bit [0]: When SPI interface is master mode, SPI interrupt happened when bit [0] ==1 and every SPI master command completed. Interrupt (HID) would be cleaned once address 0x10 was written.

IR Logical One Make and Space (USB Reset)

Address Offset: A0~A2h

Bits	R/W	Bit Mnemonic	Description	Default
23:16	R/W	Logical_One_Control_3	Bit[23]:Data Level 1T 1: data level is high to low (RC5/RC6) 0: data level is low to high (NEC) Bit[19:18]:Logical One 2T counter[9:8] Bit[17:16]:Logical One 1T counter[9:8] (RC5=0x80, RC6=0x80, NEC=0x00)	8'h00
15:8	R/W	Logical_One_Control_2	Logical One 2T counter[7:0] (RC5=0x16, RC6=0x0B, NEC=0x2A)	8'h2A
7:0	R/W	Logical_One_Control_1	Logical One 1T counter[7:0] (RC5=0x16, RC6=0x0B, NEC=0x0E) Default Condition: RC5 Device rising edge means logic-one 1T: 22*40u = 880usec 2T: 22*40u = 880usec	8'h0E

IR Logical Zero Make and Space (USB Reset)

Address Offset: A3~A5h

Bits	R/W	Bit Mnemonic	Description	Default

23:16	R/W	Logical_Zero_Control_3	Bit[19:18]:Logical Zero 2T counter[9:8] Bit[17:16]:Logical Zero 1T counter[9:8] (RC5=0x00, RC6=0x00, NEC=0x00)	8'h0
15:8	R/W	Logical_Zero_Control_2	Logical Zero 2T counter[7:0] (RC5=0x16, RC6=0x0B, NEC=0x0E)	8'h0E
7:0	R/W	Logical_Zero_Control_1	Logical Zero 1T counter[7:0] (RC5=0x16, RC6=0x0B, NEC=0x0E)	8'h0E

IR Header Code Make and Space (USB Reset)

Address Offset: A6~A8h

Bits	R/W	Bit Mnemonic	Description	Default
23:16	R/W	Header_Control_3	Bit[23]:header Level 1T 1: data level is high to low (RC5) 0: data level is low to high (NEC/RC6) Bit[21:20]: header period Set to 1 for RC6/NEC Set to 2 for RC5 Bit[19:18]:Header 2T counter[9:8] Bit[17:16]:Header 1T counter[9:8] (RC5=0x90, RC6=0x10, NEC=0x10)	8'h10
15:8	R/W	Header_Control_2	Header 2T counter[7:0] (RC5=0x16, RC6=0x16, NEC=0x70)	8'h70
7:0	R/W	Header_Control_1	Header 1T counter[7:0] (RC5=0x16, RC6=0x42, NEC=0xE1)	8'hE1

IR Repeat Code Make and Space (USB Reset)

Address Offset: A9~Abh

Bits	R/W	Bit Mnemonic	Description	Default
23:16	R/W	Repeat_Control_3	Bit[19:18]:Repeat control 2T counter[9:8] Bit[17:16]:Repeat control 1T counter[9:8] (RC5=--, RC6=--, NEC=0x00)	8'h0
15:8	R/W	Repeat_Control_2	Repeat control 2T counter[7:0] (RC5=--, RC6=--, NEC=0x38)	8'h38
7:0	R/W	Repeat_Control_1	Repeat control 1T counter[7:0]	8'h00

			(RC5=--, RC6=--, NEC=0x00)	

IR_CTRL_0 (USB Reset)

Address Offset: Ach

Bits	R/W	Bit Mnemonic	Description	Default
7	R/W	Repeat_mask	1: Mask interrupt if repeat. 0: interrupt to MCU even if repeat.	1'b0
6	R/W	IR_Release	Release bit This bit is a write clear reg.	1'b0
5:0	R/W	Divider_N	Clock divider (Reference clock is 1MHz) Default working clock period is 40usec.	8'h28

IR_CTRL_1 (USB Reset)

Address Offset: Adh

Bits	R/W	Bit Mnemonic	Description	Default
7	R/W	DATA_Length[7]	1: rising phase detect means logic 1. (ex:RC5/RC6) 0: rising phase detect means logic 0. This bit is valid only if bit6 be set to 1.	1'b0
6	R/W	DATA_Length[6]	1: Bi-Phase coding (ex: RC5/RC6) 0: Pulse Distance coding (ex: NEC)	1'b0
5:0	R/W	DATA_Length[5:0]	Receiver Data Length (bits)	6'h20

IR_CTRL_2 (USB Reset)

Address Offset: Aeh

Bits	R/W	Bit Mnemonic	Description	Default
7:0	R/W	Tolerance	Tolerance range for determine data format is one or zero. [7:4]: data phase tolerance [3:0]: header phase tolerance	8'h55

IR Receiver Data Buffer

Address Offset: Afh~B2H

Bits	R/W	Bit Mnemonic	Description	Default

31:0	R	IR_Code	Receiver Data Buffer	32'h0

IR_RC6_CTRL (USB Reset)

Address Offset: B3h

Bits	R/W	Bit Mnemonic	Description	Default
7:6	R/W	Reserved		
5	RO	NEC_Rep	NEC Repeat status	1'b0
4	RO	RC6_Tgl	RC6 Toggle bit status	1'b0
3	R/W	RC6	RC6 Select	1'b0
2:0	RO	RC6_Mode	RC6 Mode bits	3'h0

USB phy power+debug

Address: B4h

Bits	R/W	Bit Mnemonic	Description	Default
7	R/W	IDNPDEN	When this pin is set to high, the internal DN pull-down resistor R3 (19.525 kΩ, ± 27%) is enabled.	1'b0 (POR)
6	R/W	IDPPDEN	When this pin is set to high, the internal DP pull-down resistor, R2 (19.525 kΩ ± 27%), is enabled.	1'b0 (POR)
5	R/W	IPUSW2EB	When this pin is set to low, the internal pull-up resistor R4 or R6 (1.02 kΩ, ± 48%) is disabled	1'b0 (POR)
4	R/W	EXT_USBPHY	Using external USB phy; 1'b1 : switch ext Philips usb phy 1'b0: using int 0.11um usb phy	1'b0 (POR)
3	R/W	SP	1'b1: operates at full speed; 1'b0: operates at low speed	1'b1 (POR)
2	R/W	USBPHY_MONT	Monitor Usb 1.1 phy all signal by GPIO25-32 pins 1'b1: monitor ; 1'b0 : disable	1'b0 (POR)
1	R/W	IPUSW1EB	When this pin is set to low, the internal pull-up resistor R1 or R5 (1.24 kΩ, ± 27%) is enabled Set "0" the internal pull-up R (1.24Kohm) is enabled ; MCU control this bit to pull-up R; default =1'b1 no pull-up R	1'b1 (POR)
0	R/W	EB(suspend)	RX enable(low active), RCV will drive to "0" when it goes high and enters a suspend mode; default = 1'b0 not suspend	1'b0 (POR)

Clock Switch for MCU

Address: B5h

Bits	R/W	Bit Mnemonic	Description	Default
7-5	R/W		Reserved	3'b0
4	R/W	CLKBYPASS	PLL stable clock filter disable 1'b1 : bypass CLK48M clock filter 1'b0: enable CLK48M clock filter Default : 1'b1	1'b1 (POR)
3	R/W	EXTBYPASS	Sub module reset extend 1'b1: bypass; only 1 T usbclk 1'b0 : extend > 1ms Default : 1'b1	1'b1 (POR)
2-0	R/W	MCUCLKSW	3'b100: 48Mhz 3'b011: 24Mhz 3'b010: 12Mhz 3'b001: 6Mhz 3'b000: 3Mhz Default : 6MHz	3'b010 (POR)

Module clock ctrl setting1

Address: B6h

Bits	R/W	Bit Mnemonic	Description	Default
7	R/W	SRAMCLK_G	Program SRAM CLK gated ctrl 1'b1 : gated	1'b1 (POR)
6	R/W	FIFOCLK_G	FIFO CLK gated ctrl 1'b1 : gated	1'b0 (POR)
5	R/W	RECCLK_G	Rec logic gated ctrl 1'b1 : gated	1'b0 (POR)
4	R/W	PLAYCLK_G	Play logic gated ctrl 1'b1 : gated	1'b0 (POR)
3	R/W	SPDOCLK_G	SPDIFOUT clock gated ctrl 1'b1 : gated	1'b0 (POR)
2	R/W	SPDICLK_G	SPDIFIN clock gated ctrl 1'b1 : gated	1'b0 (POR)

1	R/W	IRCLK_G	IR clock gated ctrl 1'b1 : gated	1'b0 (POR)
0	R/W	USBCLK_PRE_G	Peripheral USBCLK gated ctrl 1'b1 : gated	1'b0 (POR)

Module clock ctrl setting2

Address: B7h

Bits	R/W	Bit Mnemonic	Description	Default
7-4	R/W		Dummy bits	4'b0000
3	R/W	I2CMCLK_G	I2C-master clock gated ctrl 1'b1 : gated	1'b0 (POR)
2	R/W	I2CSCLK_G	I2C-slave clock gated ctrl 1'b1 : gated	1'b0 (POR)
1	R/W	SSPICLK_G	Internal SPI clock gated ctrl 1'b1 : gated	1'b0 (POR)
0	R/W	MSPICLK_G	External SPI clock gated ctrl 1'b1 : gated	1'b0 (POR)

Module Reset ctrl setting1

Address: B8h

Bits	R/W	Bit Mnemonic	Description	Default
7	R/W	FIFO_GRSTN	FIFO ctrl module reset ctrl 1'b1 : normal; 1'b0 : reset; 1T active (usbclk domain)	1'b1
6	R/W	IR_GRSTN	IR module reset ctrl 1'b1 : normal; 1'b0 : reset; 1T active (usbclk domain)	1'b1
5	R/W	REC_GRSTN	REC module reset ctrl 1'b1 : normal; 1'b0 : reset; 1T active (usbclk domain)	1'b1
4	R/W	PLAY_GRSTN	Play module reset ctrl 1'b1 : normal; 1'b0 : reset; 1T active (usbclk domain)	1'b1
3	R/W	I2C_GRSTN	I2C-master module reset ctrl 1'b1 : normal; 1'b0 : reset; 1T active (usbclk domain)	1'b1

2	R/W	SSPI_GRSTN	Internal SPI module reset ctrl 1'b1 : normal; 1'b0 : reset; 1T active (usbclk domain)	1'b1
1	R/W	MSPI_GRSTN	External SPI module reset ctrl 1'b1 : normal; 1'b0 : reset; 1T active (usbclk domain)	1'b1
0	R/W	SPDIF_GRSTN	SPDIF module reset ctrl 1'b1 : normal; 1'b0 : reset; 1T active (usbclk domain)	1'b1

IO DC test and PDSW

Address: B9h

Bits	R/W	Bit Mnemonic	Description	Default
7	R/W		Reserved	1'b0(POR)
6	R/W	IO_CELL_VAL	IO cell value	1'b0(POR)
5	R/W	SW_IOCELL_DC	Switch IO cell control 1'b1: for DC character test 1'b0: use as normal function	1'b0 (POR)
4	R/W	PIN_PULLUP	1'b1 :all bi-direction PIN pull-up	1'b0(POR)
3	R/W	PIN_PULLDOWN	1'b1: all bi-direction PIN pull-down	1'b0(POR)
2	R/W	PIN_ALL_OUT	Under SW_IOCELL_DC=1'b1 1'b1: all bi-direction PIN as output 1'b0: all bi-direction PIN as input	1'b0(POR)
1	R/W	PDSW_P	Power down Peripheral component 1'b1: power down All GPIO[31:0] are changed as input mode	1'b1 (POR)
0	R/W	GAINPASS	Mass product gain test pass 1'b1 : pass ; 1'b0: fail	1'b0 (POR)

CMA112_CODEC analog test and monitor pop noise signal

Address: Bah

Bits	R/W	Bit Mnemonic	Description	Default
7-3	R/W		Reserved	4'b0
3	R/W	MONT_POP	Monitor pop noise ctrl signal	1'b0(POR)
2	R/W	ADC_ASICOUT	ADC output D_LY, D_RY 1'b1 : output int D_LY/D_RY 1'b0: disable	1'b0 (POR)

1	R/W	ADC_FPGAIN	ADC filter input by FPGA 1'b1 : D_LY, D_RY from FPGA 1'b0: disable	1'b0 (POR)
0	R/W	DAC_DWA_TEST	DAC DWA test enable 1'b1 : enable ; 1'b0 : disable	1'b0 (POR)

GPIO IO Strength Setting

Address: BBh

Bits	R/W	Bit Mnemonic	Description	Default
7:6	R/W	GPIO_DRIVE3	GPIO25~32 IO strength setting	2'b01 (POR)
5:4	R/W	GPIO_DRIVE2	GPIO17~24 IO strength setting	2'b01 (POR)
3:2	R/W	GPIO_DRIVE1	GPIO9~16 IO strength setting	2'b01 (POR)
1:0	R/W	GPIO_DRIVE0	GPIO1~8 IO strength setting	2'b01 (POR)

Note : 2'b00=4mA, 2'b01=8mA, 2'b10=12mA, 2'b11=16mA

GPIO IO Strength Setting

Address: BCh

Bits	R/W	Bit Mnemonic	Description	Default
7:6	R/W	I2C_DRIVE	I2C IO strength setting 2'b00=2mA, 2'b01=4mA, 2'b10=6mA, 2'b11=8mA	2'b11 (POR)
5:4	R/W	SPI_DRIVE	SPI IO strength setting 2'b00=2mA, 2'b01=4mA,2'b10=6mA, 2'b11=8mA	2'b11 (POR)
3:2	R/W	SPDIF_DRIVE	SPDIFO IO strength setting 2'b00=2mA, 2'b01=4mA,2'b10=6mA, 2'b11=8mA	2'b11 (POR)
1:0	R/W	PDSW_DRIVE	PDSW IO strength setting 2'b00=2mA, 2'b01=4mA,2'b10=6mA, 2'b11=8mA	2'b11 (POR)

I2S IO Strength Setting

Address: BDh

Bits	R/W	Bit Mnemonic	Description	Default
7:5	R/W		Reserved	4'b0
3:2	R/W	I2SDA_DRIVE	I2S-DAC IO strength setting	2'b11 (POR)

			2'b00=2mA, 2'b01=4mA,2'b10=6mA, 2'b11=8mA	
1:0	R/W	I2SAD_DRIVE	I2S-ADC IO strength setting 2'b00=2mA, 2'b01=4mA,2'b10=6mA, 2'b11=8mA	2'b11 (POR)

PLL charge pump current ctrl

Address: BEh

Bits	R/W	Bit Mnemonic	Description	Default
7:4	R/W		Reserved	4'b0(POR)
3:0	R/W	B_PLL	I2S-ADC IO strength setting	4'b0(POR)

TESTMODE value and THD mode

Address Offset: 0xBFh

Bits	R/W	Bit Mnemonic	Description	Default
7	R/W		Reserved	1'b0
6~4	R	CODECmode[2:0] Same as {GPIO_4,GPIO_3,GPIO_2}	Under Tmode = 3'b001; See analog test	3'b0
3	R	TESTMODE	TESTMODE; 1'b1: at testmode	1'b0
2-0	R	TMODE[2:0]	Enter which testmode	3'b0

GPO Data Register (Power Reset)

Address Offset: C0-C1h

Bits	R/W	Bit Mnemonic	Description	Default
15:0	R/W	GPO_0_reg GPO_1_reg	GPO data register which represents	16'h0 (POR)

GPI Data Register

Address Offset: C2-C3h

Bits	R/W	Bit Mnemonic	Description	Default
15:0	R		GPI data register which represents	16'h0 (POR)

GPIO Direction Control Register (Power Reset)

Address Offset: C4-C5h

Bits	R/W	Bit Mnemonic	Description	Default
15:0	R/W	GPOE_0 GPOE_1	GPIO output enable register which represents for pin XGPIO[15:0] 1: the corresponding pins are used as output 0: the corresponding pins are used as input	16'h0 (POR)

GPIO Interrupt Enable Mask Register (Power Reset)

Address Offset: C6-C7h

Bits	R/W	Bit Mnemonic	Description	Default
15:0	R/W	GPI_EN	GPIO_E, GPIO interrupt enable mask which represents for pins, XGPIO[15:0] 1: enable, 0: disable	16'h0 (POR)

GPIO De-bouncing Register (Power Reset)

Address Offset: C8-C9h

Default Value: 0000h (MSB -> LSB)

Bits	R/W	Bit Mnemonic	Description	Default
15:0	R/W	GPI_Deb	Enable the clock scale of mini-second (32 ms) for de-bouncing, default 1 1: enable, 0: disable	16'h0 (POR)

General Control (Power Reset)

Address Offset: CAh

Bits	R/W	Bit Mnemonic	Description	default
7	R/W		Reserved	1'b0(POR)
6	R/W	General_Ctrl[6]	LED3 Control Selector (0: HW, 1: SW)	1'b0(POR)
5	R/W	General_Ctrl[5]	LED3 SW Output Enable	1'b0(POR)
4	R/W	General_Ctrl[4]	LED2 Control Selector (0: HW, 1: SW)	1'b0(POR)
3	R/W	General_Ctrl[3]	LED2 SW Output Enable	1'b0(POR)
2	R/W	General_Ctrl[2]	LED1 Control Selector (0: HW, 1: SW)	1'b0(POR)
1	R/W	General_Ctrl[1]	LED1 SW Output Enable	1'b0(POR)
0	R/W	General_Ctrl[0]	Buzzer Output Enable	1'b0(POR)

Buzzer Freq1 (Power Reset)

Address Offset: CBh

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	Buzzer_Freq_1	Buzzer Output Frequency Counter 1 (Base On 23.44 KHz Clock)	8'h8 (POR)

Buzzer Duty1 (Power Reset)

Address Offset: 0xCCh

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	Buzzer_Duty_1	Buzzer Output Duty Cycle Counter 1 (Base On 23.44 KHz Clock)	8'h4 (POR)

Buzzer Freq2 (Power Reset)

Address Offset: 0xCDh

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	Buzzer_Freq_2	Buzzer Output Frequency Counter 2 (Base On 23.44 KHz Clock)	8'h4C (POR)

Buzzer Duty2 (Power Reset)

Address Offset: 0xCEh

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	Buzzer_Duty_2	Buzzer Output Duty Cycle Counter 1 (Base On 23.44 KHz Clock)	8'h26 (POR)

Buzzer On Time (Power Reset)

Address Offset: 0xCFh

Bits	R/W	Bit Mnemonic	Description	Default
7-0	R/W	Buzzer_ON_Time	Buzzer Output ON Time	8'h50 (POR)

Buzzer Cycle Time (Power Reset)

Address Offset: 0XD0h

Bits	R/W	Bit Mnemonic	Description	Default
7-0	R/W	Buzzer_Cycle_Time	Buzzer Output ON Time	8'hC0 (POR)

LED Freq (Power Reset)

Address Offset: 0XD1h

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	LED_Freq	LED1 Output Frequency	8'h17 (POR)

LED Duty (Power Reset)

Address Offset: 0Xd2h

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	LED_Duty	LED1 Output Duty Cycle	8'h18 (POR)

LED1 On Time (Power Reset)

Address Offset: 0xD3h

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	LED1_ON_Time	LED1 Output ON Time	8'h0c (POR)

LED1 Cycle Time (Power Reset)

Address Offset: 0xD4h

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	LED1_Cycle_Time	LED1 Output Cycle Time Time unit is ~2.2msec 22msec * 24 = 528msec LED will ON-OFF 2 times per sec	8'h18 (POR)

LED2 On Time (Power Reset)

Address Offset: 0xD5h

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	LED2_ON_Time	LED2 Output ON Time	8'h31 (POR)

LED2 Cycle Time (Power Reset)

Address Offset: 0xD6h

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	LED2_Cycle_Time	LED2 Output Cycle Time	8'h30 (POR)

LED3 On Time (Power Reset)

Address Offset: 0xD7h

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	LED3_ON_Time	LED3 Output ON Time	8'h18 (POR)

LED3 Cycle Time (Power Reset)

Address Offset: 0xD8h

Bits	R/W	Bit Mnemonic	Description	default
7-0	R/W	LED3_Cycle_Time	LED3 Output Cycle Time LED will ON-OFF 1 times per sec	8'h30 (POR)

Bounding Option

Address Offset: 0xD9h

Bits	R/W	Bit Mnemonic	Description	Default
7	R/W	URSTN_MASK	USB Reset Mask: If this bit set to 1, we will ignore USB reset from host.	2'b0 (POR)
6	R/W	SOF_SEL	SOF select: 0: SOF is coming from host command 1: SOF is coming from IR clock counter	2'b0 (POR)
5	R	SEL6	Pin SEL6	1'b0
4	R	SEL5	Pin SEL5	1'b0
3	R	SEL4	Pin SEL4	1'b0
2	R	SEL3	Pin SEL3	1'b0
1	R	SEL2	Pin SEL2	1'b0
0	R	SEL1	Pin SEL1	1'b0

GPO Switch Source (Power Reset)

Address Offset: 0xDAh

Bits	R/W	Bit Mnemonic	Description	default
7-4	R/W	GPO_SW0_H	4'h0:GPO[0]=gpo[0] 4'h1:GPO[0]=LED1 4'h2:GPO[0]=LED2 4'h3:GPO[0]=LED3 4'h4:GPO[0]=Buzzer Others: GPO[0]=1'b0	4'h0 (POR)

3-0	R/W	GPO_SW0_L	4'h0:GPO[1]=gpo[1] 4'h1:GPO[1]=LED1 4'h2:GPO[1]=LED2 4'h3:GPO[1]=LED3 4'h4:GPO[1]=Buzzer Others: GPO[1]=1'b0	4'h0 (POR)

Address Offset: 0xDBh

Bits	R/W	Bit Mnemonic	Description	default
7-4	R/W	GPO_SW1_H	4'h0:GPO[2]=gpo[2] 4'h1:GPO[2]=LED1 4'h2:GPO[2]=LED2 4'h3:GPO[2]=LED3 4'h4:GPO[2]=Buzzer Others: GPO[2]=1'b0	4'h0 (POR)
3-0	R/W	GPO_SW1_L	4'h0:GPO[3]=gpo[3] 4'h1:GPO[3]=LED1 4'h2:GPO[3]=LED2 4'h3:GPO[3]=LED3 4'h4:GPO[3]=Buzzer Others: GPO[3]=1'b0	4'h0 (POR)

Address Offset: 0xDCh

Bits	R/W	Bit Mnemonic	Description	default
7-4	R/W	GPO_SW2_H	4'h0:GPO[4]=gpo[4] 4'h1:GPO[4]=LED1 4'h2:GPO[4]=LED2 4'h3:GPO[4]=LED3 4'h4:GPO[4]=Buzzer Others: GPO[4]=1'b0	4'h0 (POR)
3-0	R/W	GPO_SW2_L	4'h0:GPO[5]=gpo[5] 4'h1:GPO[5]=LED1 4'h2:GPO[5]=LED2 4'h3:GPO[5]=LED3 4'h4:GPO[5]=Buzzer Others: GPO[5]=1'b0	4'h0 (POR)

Address Offset: 0xDDh

Bits	R/W	Bit Mnemonic	Description	default
7-4	R/W	GPO_SW3_H	4'h0:GPO[6]=gpo[6] 4'h1:GPO[6]=LED1 4'h2:GPO[6]=LED2 4'h3:GPO[6]=LED3 4'h4:GPO[6]=Buzzer Others: GPO[6]=1'b0	4'h0 (POR)

3-0	R/W	GPO_SW3_L	4'h0:GPO[7]=gpo[7] 4'h1:GPO[7]=LED1 4'h2:GPO[7]=LED2 4'h3:GPO[7]=LED3 4'h4:GPO[7]=Buzzer Others: GPO[7]=1'b0	4'h0 (POR)

Address Offset: 0xDEh

Bits	R/W	Bit Mnemonic	Description	default
7-4	R/W	GPO_SW4_H	4'h0:GPO[8]=gpo[8] 4'h1:GPO[8]=LED1 4'h2:GPO[8]=LED2 4'h3:GPO[8]=LED3 4'h4:GPO[8]=Buzzer Others: GPO[8]=1'b0	4'h0 (POR)
3-0	R/W	GPO_SW4_L	4'h0:GPO[9]=gpo[9] 4'h1:GPO[9]=LED1 4'h2:GPO[9]=LED2 4'h3:GPO[9]=LED3 4'h4:GPO[9]=Buzzer Others: GPO[9]=1'b0	4'h0 (POR)

Address Offset: 0xDFh

Bits	R/W	Bit Mnemonic	Description	default
7-4	R/W	GPO_SW5_H	4'h0:GPO[10]=gpo[10] 4'h1:GPO[10]=LED1 4'h2:GPO[10]=LED2 4'h3:GPO[10]=LED3 4'h4:GPO[10]=Buzzer Others: GPO[10]=1'b0	4'h0 (POR)
3-0	R/W	GPO_SW5_L	4'h0:GPO[11]=gpo[11] 4'h1:GPO[11]=LED1 4'h2:GPO[11]=LED2 4'h3:GPO[11]=LED3 4'h4:GPO[11]=Buzzer Others: GPO[11]=1'b0	4'h0 (POR)

Address Offset: 0XE0h

Bits	R/W	Bit Mnemonic	Description	default
7-4	R/W	GPO_SW6_H	4'h0:GPO[12]=gpo[12] 4'h1:GPO[12]=LED1 4'h2:GPO[12]=LED2 4'h3:GPO[12]=LED3 4'h4:GPO[12]=Buzzer Others: GPO[12]=1'b0	4'h0 (POR)
3-0	R/W	GPO_SW6_L	4'h0:GPO[13]=gpo[13] 4'h1:GPO[13]=LED1	4'h0 (POR)

			4'h2:GPO[13]=LED2 4'h3:GPO[13]=LED3 4'h4:GPO[13]=Buzzer Others: GPO[13]=1'b0	

Address Offset: 0XE1h

Bits	R/W	Bit Mnemonic	Description	default
7-4	R/W	GPO_SW7_H	4'h0:GPO[14]=gpo[14] 4'h1:GPO[14]=LED1 4'h2:GPO[14]=LED2 4'h3:GPO[14]=LED3 4'h4:GPO[14]=Buzzer Others: GPO[14]=1'b0	4'h0 (POR)
3-0	R/W	GPO_SW7_L	4'h0:GPO[15]=gpo[15] 4'h1:GPO[15]=LED1 4'h2:GPO[15]=LED2 4'h3:GPO[15]=LED3 4'h4:GPO[15]=Buzzer Others: GPO[15]=1'b0	4'h0 (POR)

GPI Remote Choose

Address Offset: 0xE2~E3h

Bits	R/W	Bit Mnemonic	Description	default
15:0	R/W	GPI_RWL GPI_RWH	D0==1'b1:GPI[0] remote wake up enable D1==1'b1:GPI[1] remote wake up enable D2==1'b1:GPI[2] remote wake up enable D3==1'b1:GPI[3] remote wake up enable D4==1'b1:GPI[4] remote wake up enable D5==1'b1:GPI[5] remote wake up enable D6==1'b1:GPI[6] remote wake up enable D7==1'b1:GPI[7] remote wake up enable D8==1'b1:GPI[8] remote wake up enable D9==1'b1:GPI[9] remote wake up enable D10==1'b1:GPI[10] remote wake up enable D11==1'b1:GPI[11]	16'h0 (POR)

			remote wake up enable D12==1'b1:GPI[12] remote wake up enable D13==1'b1:GPI[13] remote wake up enable D14==1'b1:GPI[14] remote wake up enable D15==1'b1:GPI[15] remote wake up enable	

GPIO pull-up/down Control Register0

Address Offset: 0xE4

Bits	R/W	Bit Mnemonic	Description	default
7	R/W	GPIO_PD0[7]	GPIO_1 ~ GPIO_7 pad control 1'b1:75k pull down; 1'b0 : 75k pull up	1'b1 (POR)
6	R/W	GPIO_PD0[6]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
5	R/W	GPIO_PD0[5]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
4	R/W	GPIO_PD0[4]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
3	R/W	GPIO_PD0[3]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
2	R/W	GPIO_PD0[2]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
1	R/W	GPIO_PD0[1]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
0	R/W	GPIO_PD0[0]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)

GPIO pull-up/down Control Register1

Address Offset: 0xE5

Bits	R/W	Bit Mnemonic	Description	default
7	R/W	GPIO_PD1[7]	GPIO_8~GPIO_16 pad control 1'b1:75k pull down; 1'b0 : 75k pull up	1'b1 (POR)
6	R/W	GPIO_PD1[6]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
5	R/W	GPIO_PD1[5]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
4	R/W	GPIO_PD1[4]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
3	R/W	GPIO_PD1[3]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
2	R/W	GPIO_PD1[2]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
1	R/W	GPIO_PD1[1]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
0	R/W	GPIO_PD1[0]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)

GPIO pull-up/down Control Register2

Address Offset: 0xE6

Bits	R/W	Bit Mnemonic	Description	default
7	R/W	GPIO_PD2[7]	GPIO17~GPIO24 pad control 1'b1:75k pull down; 1'b0 : 75k pull up	1'b1 (POR)
6	R/W	GPIO_PD2[6]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
5	R/W	GPIO_PD2[5]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
4	R/W	GPIO_PD2[4]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
3	R/W	GPIO_PD2[3]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
2	R/W	GPIO_PD2[2]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
1	R/W	GPIO_PD2[1]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
0	R/W	GPIO_PD2[0]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)

GPIO pull-up/down Control Register3

Address Offset: 0xE7

Bits	R/W	Bit Mnemonic	Description	default
7	R/W	GPIO_PD3[7]	GPIO25~GPIO32 pad control 1'b1:75k pull down; 1'b0 : 75k pull up	1'b1 (POR)
6	R/W	GPIO_PD3[6]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
5	R/W	GPIO_PD3[5]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
4	R/W	GPIO_PD3[4]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
3	R/W	GPIO_PD3[3]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
2	R/W	GPIO_PD3[2]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
1	R/W	GPIO_PD3[1]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)
0	R/W	GPIO_PD3[0]	1'b1:75k pull down; 1'b0 : 75k pull up	1'b1(POR)

Share MCU port0

Address Offset: 0xE8

Bits	R/W	Bit Mnemonic	Description	Default
7-2	R/W		reserved	6'b0
1	R/W	SH_UART	Shared with {GPIO_17,GPIO_18} = {MCU_RXD, MCU_TRX}	1'b0 (POR)
0	R/W	SH_GPIO_P[0]	GPIO_8~GPIO_16 pins shared for Mcu port0	1'b0 (POR)

AGC Global Control Register

Address: 0xf0

Bits	R/W	Description	Default
7	R/W	AGC_PRE_EN: Enable AGC function even if Host didn't send ISO-In packet.	0x0
6	R/W	Gain increase enable, else Gain decrease Gain increase means that the real gain value increased	0x0

		follow by the numeric of gain increased.	
5	R/W	AGC Left channel ~~close~~ enable	0x0
4	R/W	AGC Right channel ~~close~~ enable	0x0
3	R/W	Reserved	0x0
2-0	R/W	Each increase GAIN Step	0x1

AGC ATTACK TIME CONTROL

Address: 0xf1

Bits	R/W	Description	Default
7-6	R/W	Reserved	0x00
5-0	R/W	AGC Attack time, time increases by 43us with every step, 6'h03 default 0.129ms	0x03

AGC RELEASE TIME CONTROL

Address: 0xf2

Bits	R/W	Description	Default
7-6	R/W	Reserved	0x00
5-0	R/W	AGC Release time, time increases by 22ms with every step, 6'h03 default 66ms	0x03

AGC HOLD TIME CONTROL

Address: 0xf3

Bits	R/W	Description	Default
7-6	R/W	Reserved	0x00
5-0	R/W	AGC Hold time, time increases by 22ms with every step, 6'h0C default 264ms	0x0c

AGC Threshold Control Register

Address: 0xf4~f6

Bits	R/W	Description	Default
7-0	R/W	Max Threshold of PCM unit. It will clip the output value equal as the threshold if estimated value exceeded. {PCM 7FFFFF * -3dB(0.708)} =~ 24'h5A9FBD {PCM 7FFFFF * -1dB(0.981)} =~ 24'h721483	0x721483

AGC FIXED GAIN CONTROL

Address: 0xf7

Bits	R/W	Description	Default
7-6	R/W	Pre Max Threshold add dB unit	0x1

		default add 1dB (0~3dB)	
5-0	R/W	Set the fixed gain of amplifier: two's compliment dB unit The maximum range of AGC adjustment. By default, if AGC enable, the fixed gain will be added in original gain value. F0[6]=1 means default increase fixed gain when AGC enable. F0[6]=1 means default decrease fixed gain.	0x9

AGC Simulate CONTROL

Address: 0xf8

Bits	R/W	Description	Default
7	R/W	TTEST enable for simulation 1: enable 0: disable	0x0
6-5	R/W	Reserved	0x0
4	R/W	Volume overflow output enable 1: enable 0: disable Don't care AGC function.	0x0
3-0	R/W	Reserved	0x0

AGC GAIN MAX Limit

Address: 0xf9

Bits	R/W	Description	Default
7-6	R/W	Reserved	0x0
5-0	R/W	ADC gain maximum limit form MCU Setting Default: ~~20dB~~ 30dB (0x30 invert = 0x0F = 30dB for Analog)	~~0x24~~ 0x30

AGC GAIN MIN Limit

Address: 0xfa

Bits	R/W	Description	Default
7-6	R/W	Reserved	0x0
5-0	R/W	ADC gain minimum limit form MCU Setting Default: ~~-6dB~~ 0dB (0x12 invert = 0x2D = 0dB for Analog)	~~0x0a~~ 0x12

Clipping LED Timing Control I

Address: 0xfb

Bits	R/W	Description	Default

7-4	R/W	LED turn off time by time unit reference 0xf8[5:4] default 4 * 64ms(time unit) = 256 ms	0x4
3-0	R/W	LED turn on time by time unit reference 0xf8[5:4] default 2 * 64ms(time unit) = 128 ms	0x2

Clipping LED Timing Control II

Address: 0xfc

Bits	R/W	Description	Default
7	R/W	Breathing Light function enable 1:enable ; 0:disable	0x0
6	R/W	Reserved	0x0
5-4	R/W	1. If Breathing Light disable, it means LED trun time unit 11:256, 10:128, 01:64, 00:32 ms 2. If enable, it means breathing light unit when over. 11:2.8sec, 10:1.4sec, 01:0.7sec, 00:0.35sec	0x0
3-0	R/W	LED total work time by 0.5s with every step Once the clipping LED be triggered, it will keep On-Off during total work time.	0x4

Start Address of Input Register Data

Address: 0xfe

Bits	R/W	Description	Default
7-0	R/W	Starting Address of HID Interrupt In and HID Get Report , MSB part	0x00 (POR)

Address: 0xff

Bits	R/W	Description	Default
7-0	R/W	Starting Address of HID Interrupt In and HID Get Report	0x00 (POR)

➢ Internal Codec CMA112 Register Table

I^2S Decoder Configuration Registers

Address Offset: 00h

Default Value: 92h (MSB -> LSB) (POR)

Bit	Attribute	Description
7:6	R/W	I2S PCM Resolution 00: 16-bit

		01: 20-bit 10: 24-bit (default) 11: 32-bit (only valid when BCLK/LRCK = 128 or 256)
5:4	R/W	Reserved
3	R/W	**Data Format** 0 : I2S Mode (default) 1 : Left Justified Mode
2:0	R/W	Reserved

Address Offset: 01h

Default Value: 80h (MSB -> LSB) (POR)

Bit	Attribute	Description
7:5	R/W	Reserved
4	R/W	Reserved
3	R/W	Reserved
2:1	R/W	BCLK/LRCK Ratio, how many BCLK cycles in one LRCK cycle DACR[1:0] 00 : 64 (default) 01 : 128 10 : 256 (only valid when MCLK/LRCK = 256 or 512) 11 : Reserved
0	R/W	Reserved

I²S Encoder Configuration Registers

Address Offset: 02h

Default Value: 92h (MSB -> LSB) (POR)

Bit	Attribute	Description
7:6	R/W	I2S PCM Resolution 00: 16-bit 01: 20-bit 10: 24-bit (default) 11: 32-bit (only valid when BCLK/LRCK = 128 or 256)
5:4	R/W	Reserved
3	R/W	**Data Format** 0 : I2S Mode (default) 1 : Left Justified Mode ~~(default)~~
2:0	R/W	Reserved

Address Offset: 03h

Default Value: 80h (MSB -> LSB) (POR)

Bit	Attribute	Description
7:5	R/W	Reserved
4	R/W	
3	R/W	

2:1	R/W	BCLK/LRCK Ratio, how many BCLK cycles in one LRCK cycle DACR[1:0] 00 : 64 (default) 01 : 128 10 : 256 (only valid when MCLK/LRCK = 256 or 512) 11 : Reserved
0	R/W	Reserved

Digital Codec Control Registers

Address Offset: 04h

Default Value: 40h (MSB -> LSB) (POR)

Bit	Attribute	Description
7	R/W	Reserved
6	R/W	**AD_HPF_EN,** 1: enable adc high pass filter, 0: disable adc high pass filter , default:1
5.	R/W	**DA_HPF_EN,** 1: enable dac high pass filter, 0: disable dac high pass filter , default:0
4	R/W	**DA_192K_EN**: set sampling rate of DAC path is 192K. Default: 0
3	R/W	**DA_96K_EN**: set sampling rate of DAC path is 96K. Default: 0
2	R/W	**AD_192K_EN:** set sampling rate of ADC path is 192K. Default: 0
1	R/W	**AD_96K_EN**: set sampling rate of ADC path is 96K. Default: 0
0	R/W	**TEST_MODE**: enable DA-AD internal loop. Default: 0

Address Offset: 05h

Default Value: 00h (MSB -> LSB) (POR)

Bit	Attribute	Description
7	R/W	**zc_en**: zero cross. Default: 0
6:2	R/W	Reserved
1	R/W	**mute_p_r**: mute Right channel in digital part. Default: 0
0	R/W	**mute_p_l:** mute Left channel in digital part. Default: 0

Analog Codec Control Registers

Address Offset: 06h

Default Value: 00h (MSB -> LSB) (POR)

Bit	Attribute	Description
7	RO	**D_OVR2D75**: VR status when XVOLADJ input voltage over 2.75v
6	R/W	**SVR_EN**: Default: 0
5:0	R/W	**SVRVOL**: Analog VR volume control(-1dB/step) 000000:-0.02dB 111110:-73dB 111111:mute Default: <000000> -0.02 dB

Address Offset: 07h

Default Value: 0Ch (MSB -> LSB) (POR)

Bit	Attribute	Description
7:4	R/W	Reserved
3	R/W	**VMICAM**: Mute Micro_in Left/Right path At mixer. Default:1
2	R/W	**VLNIAM**: Mute Line_in Left/Right path At mixer. Default:1
1		
0		

Address Offset: 08h

Default Value: 1Ch (MSB -> LSB) (POR)

Bit	Attribute	Description
7:6	R/W	Reserved
5:0	R/W	**VMICL<5:0>**: Micro_in volume control(-1dB/step) <010000>:32dB <111110>:-14dB <111111>:Mute Default**:** <011100> 20dB

Address Offset: 09h

Default Value: 1Ch (MSB -> LSB) (POR)

Bit	Attribute	Description
7:6	R/W	Reserved
5:0	R/W	**VMICR<5:0>**: Micro_in volume control(-1dB/step) <010000>:32dB <111110>:-14dB <111111>:Mute Default**:** <011100> 20dB

Address Offset: 0Ah

Default Value: 21h (MSB -> LSB) (POR)

Bit	Attribute	Description
7:6	R/W	Reserved
5:0	R/W	**VLNIL<5:0>**: Line_in volume control(-1dB/step) <010101>:12dB <111110>:-29dB <111111>:Mute Default**:** <100001> 0dB

Address Offset: 0Bh

Default Value: 21h (MSB -> LSB) (POR)

Bit	Attribute	Description

7:6	R/W	Reserved
5:0	R/W	**VLNIR<5:0>**: Line_in volume control(-1dB/step) <010101>:12dB <111110>:-29dB <111111>:Mute Default**:** <100001> 0dB

Address Offset: 0Ch

Default Value: 00h (MSB -> LSB) (POR)

Bit	Attribute	Description
7:1	R/W	Reserved
0	R/W	**VDAM**: Mute DAC analog Left/Right path At mixer. Default: 0

Address Offset: 0Dh

Default Value: 00h (MSB -> LSB) (POR)

Bit	Attribute	
7:6	R/W	Reserved
5:0	R/W	**VML<5:0>**: Mast volume control(-1dB/step) 000000: 0 dB 111110: -62dB 111111:mute Default**:** <000000>0 dB

Address Offset: 0Eh

Default Value: 00h (MSB -> LSB) (POR)

Bit	Attribute	Description
7:6	R/W	Reserved
5:0	R/W	**VMR<5:0>**: Mast volume control(-1dB/step) 000000: 0 dB 111110: -62dB 111111:mute Default**:** <000000>0 dB

Address Offset: 0Fh

Default Value: 3Fh (MSB -> LSB) (POR)

Bit	Attribute	Description
7	R/W	Reserved
6	R/W	**VADM**: Mute Recording. Default: 0
5	R/W	**EN_AA**: Enable power at Micro_in and Line_in monitor A_A path. Default: 1
4	R/W	**EN_DA**: Enable Power at Playback path. Default: 1
3	R/W	**VMASTM**: MUTE playback volume. Default:1
2	R/W	**EN_AD**: Enable total ADC path. Default: 1
1	R/W	**EN_VBG**: Enable Power at Bandgap and Current reference block. Default: 1

0	R/W	**EN_VAG**: Enable OP common refence source. Default: 1

Address Offset: 10h

Default Value: 02h (MSB -> LSB) (POR)

Bit	Attribute	Description
7:6	R/W	**ADSEL<1:0>**: To select ADC left/Right channel input path 00: Line_in 01:A_A path and DAC Mixer path 10:Micro_in 11: A_A Path mixer Default:00
5	R/W	**ENL_ACREF**: To enable (low) the reference point from board in the recording gain stage.
4:2		Reserved
1	R/W	**EN_XLOCOM**: Enable the power of Driver XLOCOM Default: 1
0	R/W	**MONO_EN**: Enable the MONO IN/OUT in AA path Default: 0

Address Offset: 11h

Default Value: 21h (MSB -> LSB) (POR)

Bit	Attribute	Description
7:6	R/W	Reserved
5:0	R/W	**VADL<5:0>**: Recording volume control (-1 dB/step) If ADSEL = 00 or 01 Default: <100001> 0 dB If ADSEL = 10 or 11 Default: <011001> 20 dB

Address Offset: 12h

Default Value: 21h (MSB -> LSB) (POR)

Bit	Attribute	Description
7:6	R/W	Reserved
5:0	R/W	**VADR<5:0>**: Recording volume control (-1 dB/step) If ADSEL = 00 or 01 Default: <100001> 0 dB If ADSEL = 10 or 11 Default: <011001> 20 dB

Address Offset: 13h

Default Value: 1Dh (MSB -> LSB) (POR)

Bit	Attribute	Description
7:5	R/W	Reserved
4	R/W	**CLAMP**: To enable adc input signal amplitude limiter. Default:1

3	R/W	**DAIN_SEL**: To select DAC analog filter from DACMOD. Default:1
2	R/W	**QA**: To enable Line driver at saving power mode. Default:1
1	R/W	**TMOD**: To enable ADC recording path from XMICL/R and XLNIL/R PAD. Default: 0
0	R/W	**VREF_SEL**: To select internal reference source from Bandgap. Default:1

Address Offset: 14h

Default Value: 10h (MSB -> LSB) (POR)

Bit	Attribute	Description
7:6	R/W	Reserved
5:4	R/W	**SEL_TVH** : Select DC voltage to XMICL/R and XLNIL/R path at test mode When SEL_TVH<1:0>=1x, the DC voltage is 2.8V at test mode. When SEL_TVH<1:0>=01, the DC voltage is 1.71875V at test mode. When SEL_TVH<1:0>=00, the DC voltage is 0.7V at test mode.
3	R/W	Reserved
2	R/W	**TST_AD_IN**: Enable to test Recording Volume step. Default: 0
1	R/W	**EN_VTST**: Enable DC voltage input to XMICL/R and XLNIL/R path at test mode. Default: 0
0		

IO pad Control Registers

Address Offset: 15h

Default Value: 02h (MSB -> LSB) (POR)

Bit	Attribute	Description
7:3	R/W	Reserved
2:0	R/W	**IO_DRIVE**, IO pad driving capability, default : 3'b010

Debug Mode Control Registers (for digital codec)

Digital dafilter

Address Offset: 16h

Default Value: 00h (MSB -> LSB) (POR)

Bit	Attribute	Description
7	R/W	Reserved
6	R/W	Reserved
5	R/W	1: Monitor left channel DAC output, default: 0
4	R/W	1: Monitor right channel DAC output, default: 0
3	R/W	1: Monitor left channel sigma delta modulator input, default: 0
2	R/W	1: Monitor right channel sigma delta modulator input, default: 0
1	R/W	1: Monitor left channel DAC input, default: 0
0	R/W	1: Monitor right channel DAC input, default: 0

Digital adfilter

Address Offset: 17h

Default Value: 00h (MSB -> LSB) (POR)

Bit	Attribute	Description
7	R/W	Reserved
6	R/W	Reserved
5	R/W	Reserved
4	R/W	Reserved
3	R/W	Reserved
2	R/W	1: Monitor left channel ADC output, default: 0
1	R/W	1: Monitor right channel ADC output, default: 0
0	R/W	1: Monitor left & right channel ADC input, default: 0

Resolve glitch from dac_mclk & adc_mclk

Address Offset: 18h

Default Value: 00h (MSB -> LSB) (POR)

Bit	Attribute	Description
7:2	R/W	Reserved
1	R/W	**DAC_GLITCH_RESET**: pull reset of 16 cycles period for DAC codec. Default: 0.
0	R/W	**ADC_GLITCH_RESET**: pull reset of 16 cycles period for ADC codec. Default: 0.

Digital Codec Gain Selection

Address Offset: 19h

Default Value: 00h (MSB -> LSB) (POR)

Bit	Attribute	Description
7	R/W	Reserved
6:4	R/W	Coefficient of ADC_HPF for lower band edge. $1-z^{-1} / 1-z^{-1}+coefficient*z^{-1}$ 3'b001 :2^-9, -1.85dB 3'b010 :2^-11, -0.156dB 3'b100 :2^-12, -0.05dB Others :2^-10, -0.54dB
3:2	R/W	**DAC_GAIN_SEL**: select digital dac gain degree. 00 : 1.92 01 : 1.92*0.99 10 : 1.92*0.98 11 : 1.92*0.97 Default: 00.
1:0	R/W	**ADC_GAIN_SEL**: select digital adc gain degree. 00 : 1.40625 01 : 1.40625*0.99 10 : 1.40625*0.98 11 : 1.40625*0.97 Default: 00.

Part 2 of Analog Codec Control Registers

Address Offset: 1Ah

Default Value: 0Fh (MSB -> LSB) (POR)

Bit	Attribute	Description
7:5	R/W	Reserved
4:0		

Control the Bias Current

Address Offset: 1Bh

Default Value: 00h (MSB -> LSB) (POR)

Bit	Attribute	Description
7:4	R/W	Reserved
3:0	R/W	**IREF_TEST<3:0>:** IREF_TEST<1:0>:to control the bias current for opamp in adc,dac. IREF_TEST<3:2>:to control the bias current for vag buffer. Default=0000

Reverse Clock Phase to Analog Part

Address Offset: 1Ch

Default Value: 00h (MSB -> LSB) (POR)

Bit	Attribute	Description
7:2	R/W	Reserved
1	R/W	1: Reverse CLKDA phase to analog part. Default : 0
0	R/W	1: Reverse CLKAD phase to analog part. Default : 0

MBIST

Address Offset: **1E**h

Default Value: 00h (MSB -> LSB) (POR)

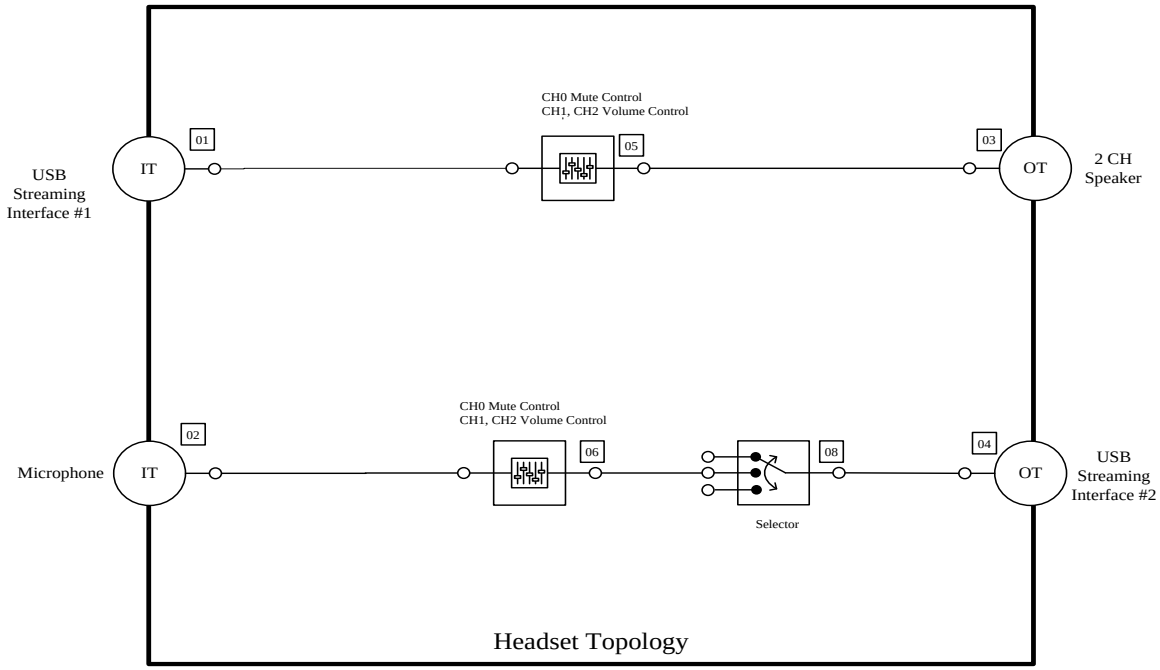
Bit	Attribute	Description
7:4	R/W	Reserved
3	RO	**BIST_Finish:** Selected memory BIST done without any error 0:uncomplete 1:completed successfully
2	RO	**BIST_Fail:** Selected memory BIST Fail, defect occurred in memory 0:n/a 1:BIST fail
1	R/W	**ADMODE:** ADC or DAC memory selected for BIST mode 0: DAC memory 1:ADC memory default:0
0	R/W	**BISTMODE:** Memory BIST mode enable, default: 0

Appendix

This chapter describes specifications of the USB device implemented by sample applications.

APPENDIX A. Headset Configuration

- **Audio Topology**



Headset Topology

- **USB Interfaces List**

Interface #	Interface Description	Endpoint
Interface 0	Audio Control Interface of Audio Device	0x00 (Control)
Interface 1	Audio Stream Interface for playback	0x01 (Iso. Out)
Interface 2	Audio Stream Interface for recording	0x82 (Iso. In)
Interface 3	HID Interface	0x87 (Interrupt In)

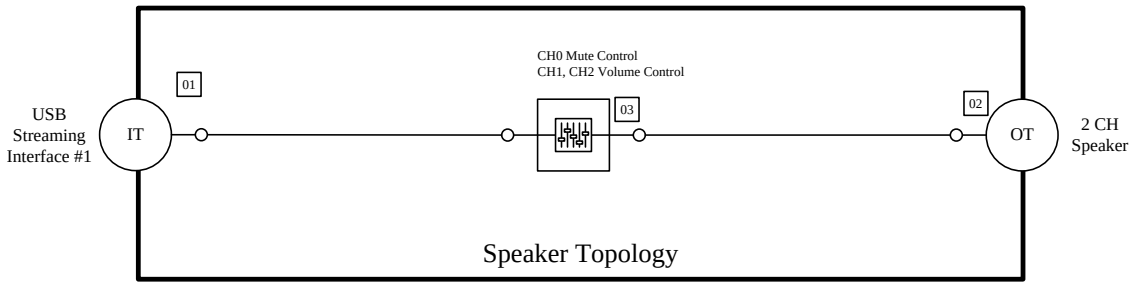
- **Audio Stream Interfaces' Alternate Setting List for Full-Speed**

Interface #	Alternate Setting	Data Format	Sampling Rate (Hz)

Interface 1	Alt 1	2CH, 16Bits PCM	8000, 11025, 16000, 22050, 32000, 44100, 48000
Interface 2	Alt 1	2CH, 16Bits PCM	8000, 11025, 16000, 22050, 32000, 44100, 48000

APPENDIX B. Speaker Configuration

- **Audio Topology**



Speaker Topology

- **USB Interfaces List**

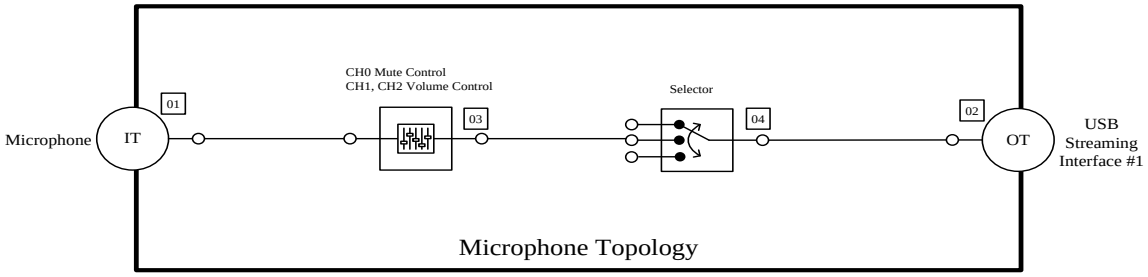
Interface #	Interface Description	Endpoint
Interface 0	Audio Control Interface of Audio Device	0x00 (Control)
Interface 1	Audio Stream Interface for playback	0x01 (Iso. Out)
Interface 2	HID Interface	0x87 (Interrupt In)

- **Audio Stream Interfaces' Alternate Setting List for Full-Speed**

Interface #	Alternate Setting	Data Format	Sampling Rate (Hz)
Interface 1	Alt 1	2CH, 16Bits PCM	8000 11025 16000 22050 32000 44100 48000

APPENDIX C. Microphone Configuration

- **Audio Topology**



Microphone Topology

- **USB Interfaces List**

Interface #	Interface Description	Endpoint
Interface 0	Audio Control Interface of Audio Device	0x00 (Control)
Interface 1	Audio Stream Interface for recording	0x81 (Iso. In)
Interface 2	HID Interface	0x87 (Interrupt In)

- **Audio Stream Interfaces' Alternate Setting List for Full-Speed**

Interface #	Alternate Setting	Data Format	Sampling Rate (Hz)
Interface 1	Alt 1	2CH, 16Bits PCM	8000 11025 16000 22050 32000 44100 48000

APPENDIX D. HID Interface

This section describes the details of HID interface.

- **HID Report Descriptor**

```
db    05H, 0CH                  ;;Usage Page(Consumer)
db    09H, 01H                  ;;Usage(Consumer Control)
db    0A1H, 01H                 ;;Collection(Application)
```

```
    db    85H, 01H                ;;Report ID(1)
    db    15H, 00H                ;;Logical Min.(0)
    db    25H, 01H                ;;Logical Max.(1)
    db    09H, 0E9H               ;;Usage(Vol. Increment)
    db    09H, 0EAH               ;;Usage(Vol. Decrement)
    db    75H, 01H                ;;Report Size(1)
    db    95H, 02H                ;;Report Count(2)
    db    81H, 42H                ;;Input(Data, Variable, Absolute,
Null state)
    db    09H, 0E2H               ;;Usage(Mute)
    db    95H, 01H                ;;Report Count(1)
    db    81H, 06H                ;;Input(Data, Variable, Relative)
    db    06H, 01H, 0FFH          ;;Usage Page(Vendor Defined)
    db    09H, 01H                ;;Usage(Vendor1 ??)
    db    95H, 09H                ;;Report Count (9)
    db    81H, 06H                ;;Input(Data, Variable, Relative)
    db    05H, 0CH                ;;Usage Page(Consumer)
    db    09H, 0CDH               ;;Usage(Play/Pause)
    db    09H, 0B7H               ;;Usage(Stop)
    db    09H, 0B5H               ;;Usage(Scan Next Track)
    db    09H, 0B6H               ;;Usage(Scan Previous Track)
    db    95H, 04H                ;;Report Count(4)
    db    81H, 06H                ;;Input(Data, Variable, Relative)
    db    06H, 07H, 0FFH          ;;Usage Page(Vendor Defined)
    db    09H, 01H                ;;Usage(Vendor1 ??)
    db    75H, 08H                ;;Report Size(8)
    db    95H, 0DH                ;;Report Count(13)
    db    81H, 06H                ;;Input(Data, Variable, Relative)
    db    09H, 00H                ;;Usage(Undefined)
    db    95H, 0FH                ;;Report Count(15)
    db    91H, 02H                ;;Output(Data, Variable, Absolute)
    db    0C0H                    ;;End Collection
```

- **HID Input Report**

The 16-bytes input report is defined as the following table. Host will be notified by an input report via interrupt pipe. Host can also get input report with class request “Get Report” via control pipe.

Host can read registers of CM65xx by HID input report’s byte 6~byte 15. Start address

and length of registers that host reads can be set by sending output report.

	Description	**Size**
Byte 0	Report ID (Always 1)	1
Byte 1~Byte 2	For defined HID event, and each event occupies one bit (this depends on HID report descriptor)	2
Byte 3	start address of returned data (H-start_addr)	1
Byte 4	start address of returned data (L-start_addr)	1
Byte 5	Interrupt source. Bit 7: Reserved Bit 6: UART_INT Bit 5: GPI_INT Bit 4: SPIS_INT Bit 3: SPIM_INT Bit 2: I2CS_INT Bit 1: I2CM_INT Bit 0: IR_INT	1
Byte 6~Byte15	Register content	10

- **HID Output Report**

HID output report is designed for writing registers to CM65xx. It is also used for setting start address and length of registers sent to host in input report.

	Description	Size
Byte 0	Report ID (Always 1)	5
Byte 1	1. 0x00: Set register read in input report 2. start address of returned data (H-start_addr)	1
Byte 2	1. 0xFE: Set register read in input report 2. start address of returned data (L-start_addr)	1
Byte 3	Effective write data length (<= 12) Effective read data length (<= 10)	1
Byte 4	1. If Byte1 is 0x00 and Byte2 is 0xFE, this byte is the value set to “Register Address(H)” in input report. 2. If Byte2 is not 0xFE, this byte is data written to register.	1
Byte 5	1. If Byte1 is 0x00 and Byte2 is 0xFE, this byte is	1

	the value set to “Register Address(L)” in input report. 2. If Byte2 is not 0xFE, this byte is data written to register.	
Byte 6~Byte15	Byte 7~Byte15 are data written to register.	10

APPENDIX E. Vendor Commands

The vendor requests implemented in the demo application are listed in the following table.

bmRequestType	bRequest	wValue	wIndex	wLength	Data
01000000b	Write Register 01h	Address offset	0	Byte count	Content of register
11000000b	Read Register 02h	Address offset	0	Byte count	Content of register