4 Audio

4.1 Audio Codec

4.1.1 Overview

The Audio Codec is high-performance audio encoder and decoder module which supports DAC/ADC, dynamic range controller (DRC) and dynamic voltage controller (DVC) functions.

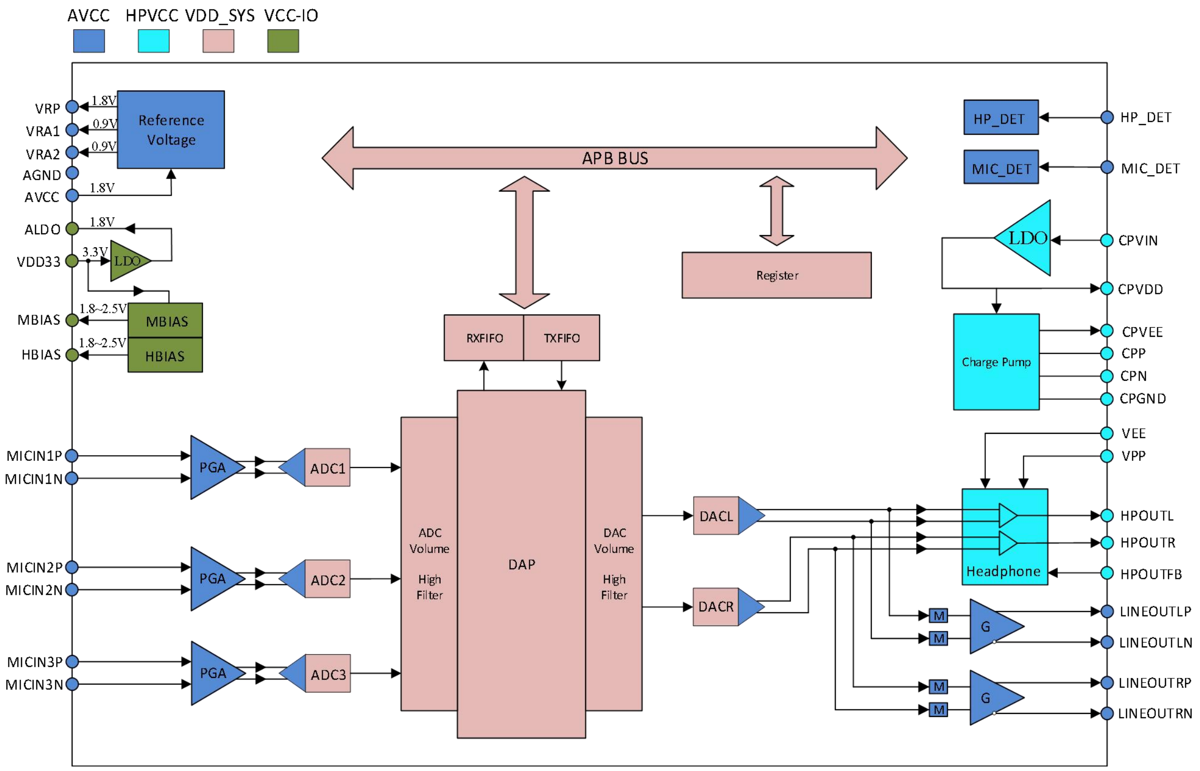
The Audio Codec has the following features:

- Two audio digital-to-analog converter (DAC) channels
  - 16-bit and 20-bit sample resolution
  - 8 kHz to 192 kHz DAC sample rate
  - 100 ± 2 dB SNR@A-weight, -85 ± 3 dB THD+N
- Three audio outputs
  - One stereo headphone output: HPOUTL/R
  - Two differential lineout outputs: LINEOUTLP/N and LINEOUTRP/N
- Three audio analog-to-digital converter (ADC) channels
  - 16-bit and 20-bit sample resolution
  - 8 kHz to 48 kHz ADC sample rate
  - 95 ± 3dB SNR@A-weight, -80 ± 3dB THD+N
- Three differential microphone inputs: MICIN1P/1N, MICIN2P/2N, and MICIN3P/3N (for echo reduction)
- Two low-noise analog microphone bias outputs: MBIAS and HBIAS
- Supports Dynamic Range Controller adjusting the DAC playback and ADC recording
- One 128x20-bits FIFO for DAC data transmit, one 128x20-bits FIFO for ADC data receive
- Programmable FIFO thresholds
- Supports interrupts and DMA
- Internal ALDO output for AVCC

4.1.2 Block Diagram

The following figure shows the block diagram of Audio Codec.

Figure 4-1 Audio Codec Block Diagram



4.1.3 Functional Description

4.1.3.1 External Signals

Table 4-1 Audio Codec External Signals

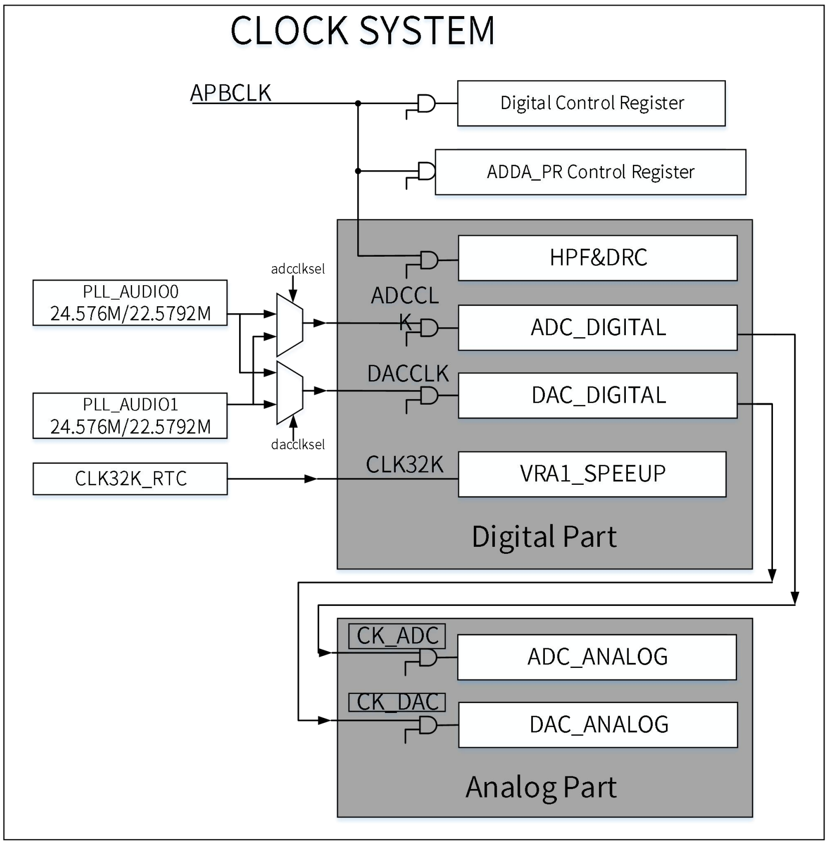
Signal Name	Description	Type
MICIN1N	Microphone Differential Negative Input 1	AI
MICIN1P	Microphone Differential Positive Input 1	AI
MICIN2N	Microphone Differential Negative Input 2	AI
MICIN2P	Microphone Differential Positive Input 2	AI
MICIN3N	Microphone Differential Negative Input 3	AI
MICIN3P	Microphone Differential Positive Input 3	AI
LINEOUTLN	Lineout Left Channel Negative Differential Output	AO
LINEOUTLP	Lineout Left Channel Positive Differential Output	AO
LINEOUTRN	Lineout Right Channel Negative Differential Output	AO
LINEOUTRP	Lineout Right Channel Positive Differential Output	AO
HPOUTL	Headphone Light Output	AO
HPOUTR	Headphone Right Output	AO
HPOUTFB	Pseudo Differential Headphone Ground Reference	AI

Signal Name	Description	Type
MIC-DET	Headphone MIC detect	AI
HP-DET	Headphone Jack detect	AI
MBIAS	First bias voltage output for main microphone	AO
HBIAS	Second bias voltage output for headset microphone	AO
CPVDD	Analog power for headphone charge pump	P
CPVEE	Charge pump negative voltage output	P
CPVIN	Analog power for LDO	P
AVCC	Power Supply for Analog Part	P
ALDO-OUT	Power Supply for AVCC	P
VDD33	Power Supply for 3.3V Analog Part	P
VEE	Negative Voltage to Headphone	P
VRA1	Internal Reference Voltage	AO
VRA2	Internal Reference Voltage	AO
VRP	Internal Reference Voltage	AO
AGND	Analog Ground	G

4.1.3.2 Clock Sources

The following figure describes the clock source of Audio Codec. For clock setting, configuration, and gating information, refer to section 2.11 Power Reset Clock Management (PRCM).

Figure 4-2 Audio Codec Clock Diagram



- Digital Part

The clock sources for the digital ADC and DAC are the PLL_AUDIO0 and PLL_AUDIO1. Configure the CLK_SRC_SEL bit (bit [26:24]) of AUDIO_CODEC_ADC_CLK_REG register to select clock sources for ADC. Configure CLK_SRC_SEL bit (bit [26:24]) of AUDIO_CODEC_DAC_CLK_REG register to select clock sources for DAC. The PK-PK jitter of PLL_AUDIO0 and PLL_AUDIO1 should be less than 200 ps.

The clock source for VRA1_SPEEDUP is CLK32K from RTC.

- Analog Part

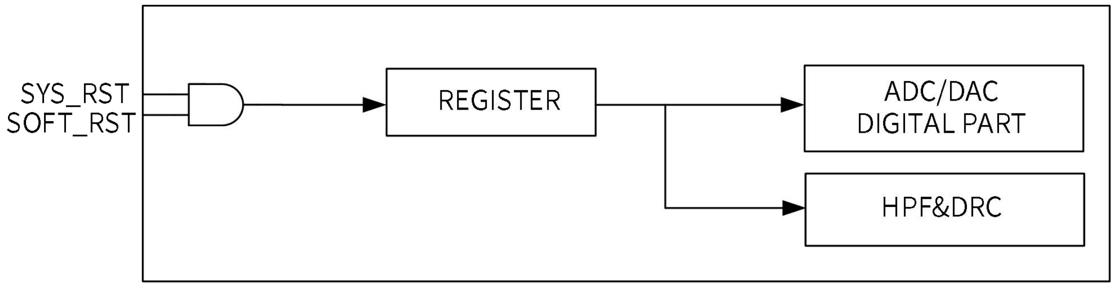
The clock source for the analog part is the CK_ADC and CK_DAC, both of which are divided from the digital part.

4.1.3.3 Reset System

Digital Part Reset System

The following figure shows the reset system of the audio codec digital part.

Figure 4-3 Audio Codec Digital Part Reset System

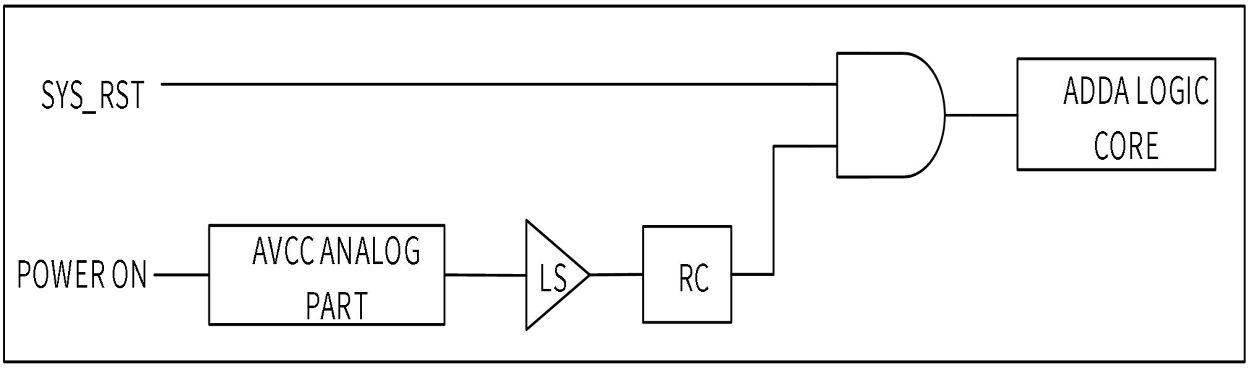


The MCU_SYS_RST comes from the VDD-SYS domain and is produced by the RTC domain which is controlled by MCU_PRCM. Each domain has the de-bounce to confirm the reset system is strong. For the codec register part, MIX can be reset by the MCU_SYS_RST when being powered on or the system soft is writing the reset control logic. The other parts can be reset by the soft configuration through writing the register.

Analog Part Reset System

The following figure shows the reset system of the audio codec analog part.

Figure 4-4 Audio Codec Analog Part Reset System

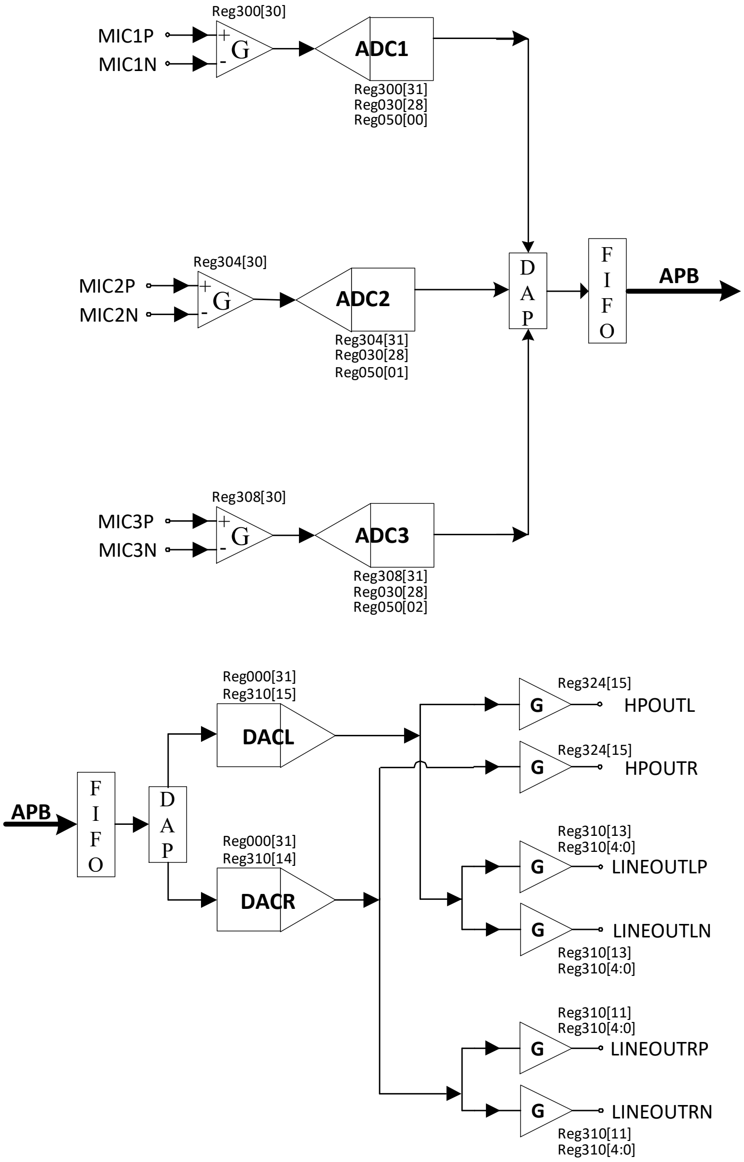


When AVCC is powered on, it sends the AVCC_POR signal. The AVCC_POR signal passes the level shift and RC filter part to the Audio Codec logic core.

4.1.3.4 Data Path Diagram

The following figure shows a data path of the Audio Codec.

Figure 4-5 Audio Codec Data Path Diagram



4.1.3.5 Three ADCs

The three ADCs are used for recording stereo sound and a reference signal. The sample rates of the three ADCs are independent of the DAC sample rate. The digital ADC part can be enabled or disabled by the bit[28] of the AC_ADC_FIFOC register.

4.1.3.6 Stereo DACs

The stereo DAC sample rate can be configured by setting the register. To save power, the analog DACL can be enabled or disabled by setting the bit [15] of the DAC_REG register, and the analog

DACR can be enabled or disabled by setting the bit [14] of the DAC_REGregister. The digital DAC part can be enabled or disabled by the bit [31] of the AC_DAC_DPC register.

4.1.3.7 Analog Audio Input Path

The Audio Codec supports 3 analog audio input paths:

- MICIN1P/N
- MICIN2P/N
- MICIN3P/N

The MICIN is a high impedance, low capacitance input suitable for connecting to various differential microphones of different dynamics and sensitivity. The gain for each pre-amplifier can be set independently. MBIAS provide reference voltage for electret condenser type(ECM) microphones.

4.1.3.8 Analog Audio Output Path

The Audio Codec has two types of analog output ports:

- LINEOUTLP
- LINEOUTLN
- LINEOUTRP
- LINEOUTRN
- HPOUTL
- HPOUTR

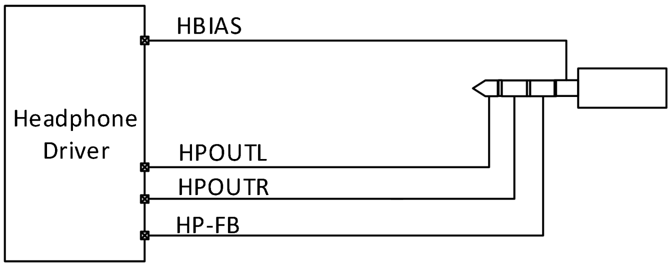
LINEOUTL/R

LINEOUTL/R provides one differential output to drive line signals to external audio equipment. The LINEOUTLP/N output source from DACL. The LINEOUTRP/N output source from DACR. The volume control is logarithmic with a 43.5 dB rang in 1.5 dB step from -43.5 dB to 0 dB. The LINEOUTL/R output buffer power up or down by bit [13] or bit [11] of DAC_REG (Offset: 0x0310).

Headphone Output

The headphone PA power up or down by bit [15] of HP_REG (Offset: 0x0324). HPOUTL/R can drive a 16R or 32R headphone load without DC capacitors by using Charge Pump to generate the negative rails. HP-FB is the ground loop noise rejection feedback. HBIAS provides reference voltage for electret condenser type (ECM) microphones.

Figure 4-6 Headphone Output Application



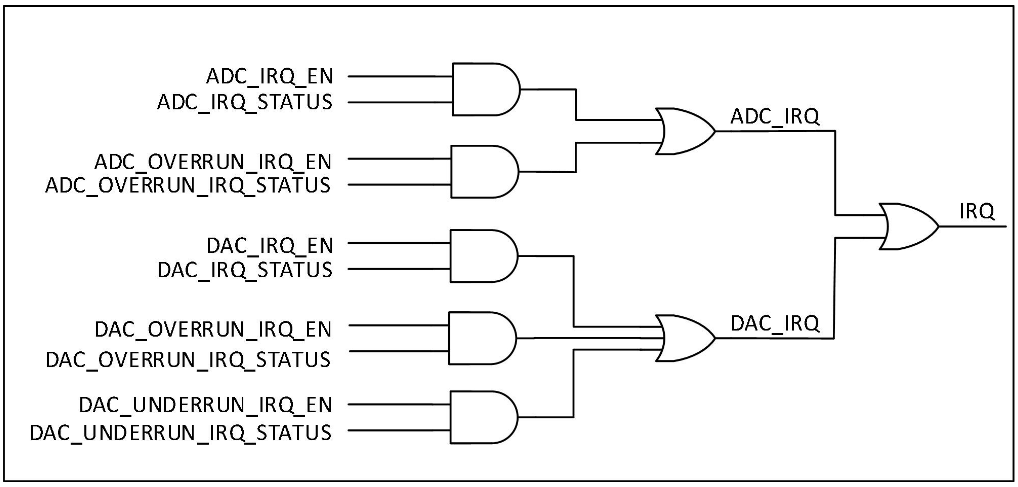
4.1.3.9 Microphone BIAS

The MBIAS output provides a low noise reference voltage suitable for biasing electrets type microphones and the associated external resistor biasing network.

4.1.3.10 Interrupt

The Audio Codec has two groups of interrupt. The following figure describes the Audio Codec interrupt system.

Figure 4-7 Audio Codec Interrupt System

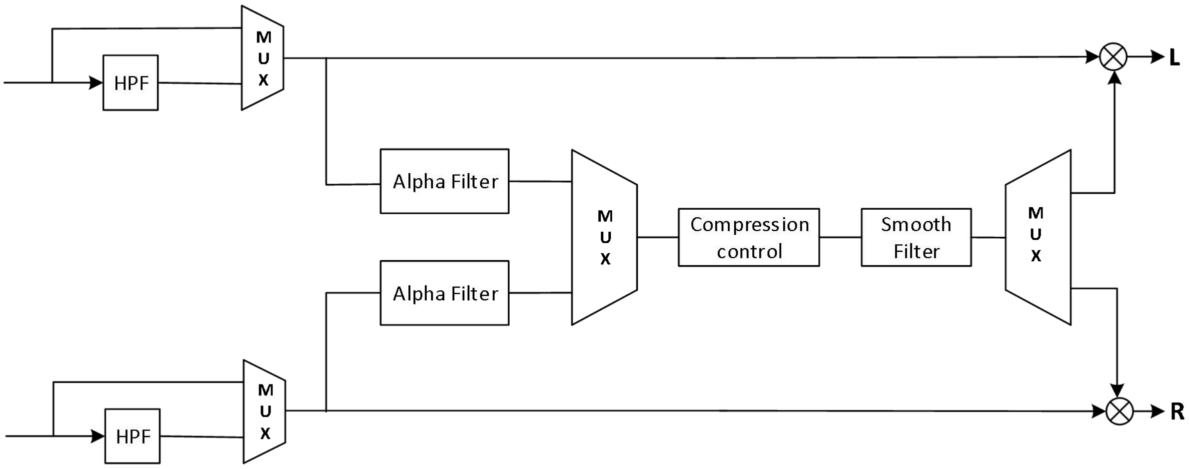


4.1.3.11 Digital Audio Processor (DAP)

The DAP module is used to remove the DC offset and automatically adjusts the volume to a flatten volume level. It mainly consists of two HPF and one DRC.

The following figure shows the DAP data flow.

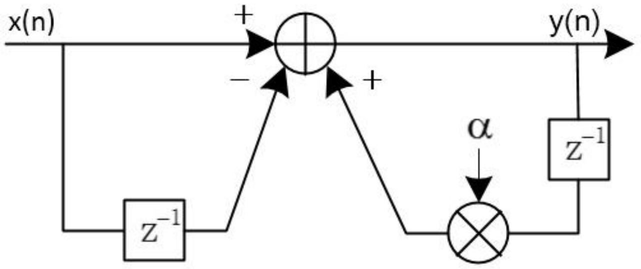
Figure 4-8 DAP Data Flow



HPF Function

The DAP has individual channel high pass filter (HPF, -3dB cutoff < 1Hz) that can be enabled and disabled. The filter cutoff frhequency is less than 1 Hz that can be removed DC offset from ADC recording. The HPF can also be bypassed.

Figure 4-9 HPF Logic Structure



HPF transfer function is $H(z) = \frac{1 - z^{-1}}{1 - \alpha z^{-1}}$,that is y(n)=αy(n-1)+x(n)-x(n-1).

For cut-off frequency Fpass: w = Fpass/Fs * 2 * pi. Generally, w is small. So, $\alpha < 0, |\alpha| < 1$.

DRC Function

The DRC scheme has three thresholds, three offsets, and four slopes (all programmable). There is one ganged DRC for the left and right channels. The following figure shows the diagram of DRC input/output.

Figure 4-10 DRC static Curve Parameters

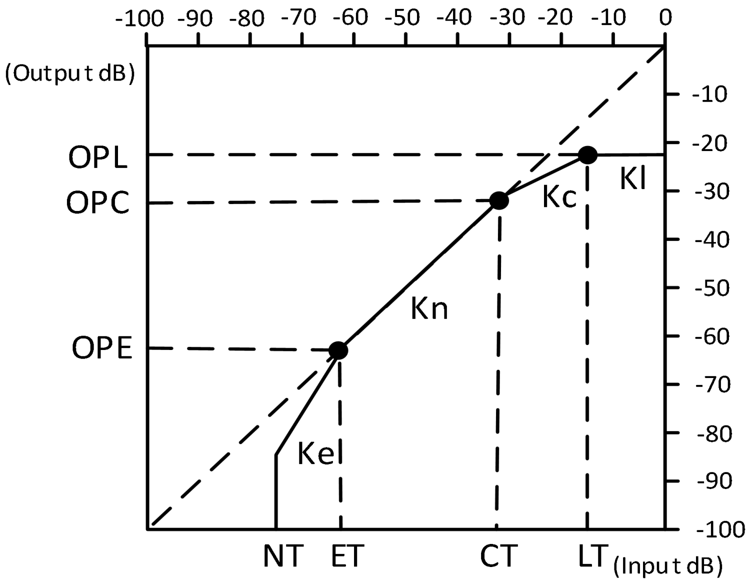
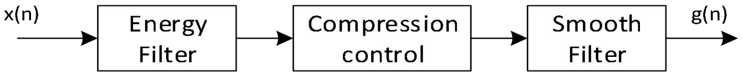


Figure 4-11 DRC Block Diagram



Professional-quality dynamic range compression automatically adjusts volume to flatten volume level.

One DRC for left/right and one DRC for subwoofer.

Each DRC has the adjustable threshold, offset, compression levels, programmable energy, attack, and decay time constants.

Transparent compression: Compressors can attack fast enough to avoid apparent clipping before engaging, and decay times can be set slow enough to avoid pumping.

Configure the DRC parameters according to the following guidelines:

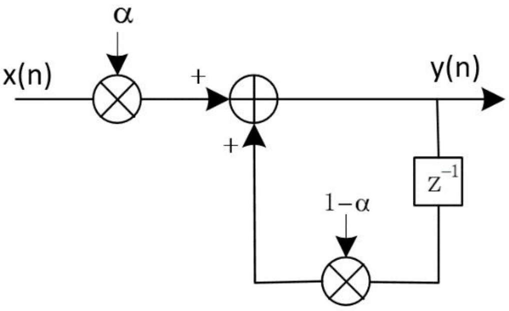
- Number format

The Number format is N.M which means there are N bits to the left of the decimal point including the sign bit and M bits to the right of the decimal point. For example, Numbers formatted 9.23 means that there are 9 bits at the left of the decimal point and 23 bits at the right decimal point.

- Energy Filter

The following figure shows the structure of the energy filter.

Figure 4-12 Energy Filter Structure



The Energy Filter is to estimate of the RMS value of the audio data stream into DRC, and has two parameters, which determine the time window over which RMS to be made. The parameter is computed by $\alpha = 1 - e^{-2.2Ts/ta}$.

- Compression Control

This element has six parameters (ET, CT, LT, Ke, Kn, Kc, Kl, OPL, OPC, OPE), which are all programmable, and the computation will be explained as follows.

Threshold Parameter Computation (T parameter)

The threshold is the value that determines the signal to be compressed or not. When the signal's RMS is larger than the threshold, the signal will be compressed. The value of threshold input to the coefficient register is computed by

$$Tin = -\frac{T_{dB}}{6.0206}$$

Where, T_{dB} must less than zero, the positive value is illegal.

For example, it is desired to set CT=-40dB, then the Tin require to set CT to -40dB is CTin = -(-40dB)/6.0206 = 6.644, CTin is entered as a 32-bit number in 8.24 format.

Therefore, CTin = 6.644 = 0000 0110.1010 0100 1101 0011 1100 0000 = 0x06A4 D3C0 in 8.24 format.

Slope Parameter Computation (K parameter)

The K is the slope within compression region. For example, a n:1 compression means that an output increase 1dB is for n dB RMS input. The k input to the coefficient ram is computed by

$$K = \frac{1}{n}.$$

Where, n is from 1 to 50, and must be integer.

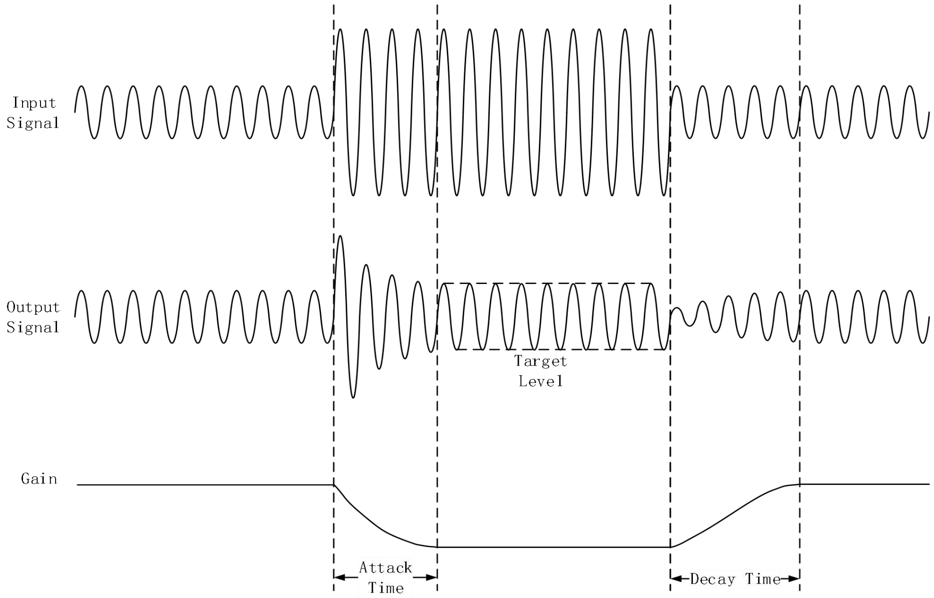
For example, it is desired to set 2:1, then the Kc require to set to 2:1 is Kc = 1/2 = 0.5, Kc is entered as a 32-bit number in 8.24 format.

Therefore, Kc = 0.5 = 0000 0000.1000 0000 0000 0000 0000 0000 = 0x0080 0000 in 8.24 format.

- Gain Smooth Filter

The Gain Smooth Filter is to smooth the gain and control the ratio of gain increase and decrease. The decay time and attack is shown in Figure 1-17. The structure of the Gain Smooth filter is also the Alpha filter, so the rise time computation is the same as the Energy filter which is $\alpha = 1 - e^{-2.2Ts/ta}$.

Figure 4-13 Gain Smooth Filter



4.1.4 Programming Guidelines

4.1.4.1 Playing

Step 1 Codec initialization: configure AUDIO_CODEC_BGR_REG and AUDIO_CODEC_DAC_CLK_REG to open the audio codec bus clock gating, release the bus reset, and open the PLL_AUDIO DAC clock gating; set up PLL_AUDIO0_CTRL_REG to configure PLL_AUDIO0 frequency and enable PLL_AUDIO0. For details of PLL_AUDIO0, refer to section 2.5 Clock Controller Unit (CCU).

Step 2 Set up the sample rate and data transfer format, then open the DAC.

Step 3 Configure the DMA and DMA request.

Step 4 Enable the DAC DRQ and DMA.

4.1.4.2 Recording

Step 1 Codec initialization: configure AUDIO_CODEC_BGR_REG and AUDIO_CODEC_ADC_CLK_REG to open the audio codec bus clock gating, release the bus reset, and open the PLL_AUDIO1 ADC clock gating; set up PLL_AUDIO1_CTRL_REG to configure PLL_AUDIO1 frequency and enable PLL_AUDIO1. For details, refer to section 2.11 Power Reset Clock Management (PRCM).

Step 2 Configure the sample rate and data transfer format, then open the ADC.

Step 3 Configure the DMA and DMA request.

Step 4 Enable the ADC DRQ and DMA.

4.1.5 Register List

Module Name	Base Address
AUDIO CODEC	0x07110000

Register Name	Offset	Description
AC_DAC_DPC	0x0000	DAC Digital Part Control Register
DAC_VOL_CTRL	0x0004	DAC Volume Control Register
AC_DAC_FIFOC	0x0010	DAC FIFO Control Register
AC_DAC_FIFOS	0x0014	DAC FIFO Status Register
AC_DAC_TXDATA	0x0020	DAC TX DATA Register
AC_DAC_CNT	0x0024	DAC TX FIFO Counter Register
AC_DAC_DG	0x0028	DAC Debug Register
AC_ADC_FIFOC	0x0030	ADC FIFO Control Register
ADC_VOL_CTRL1	0x0034	ADC Volume Control1 Register
AC_ADC_FIFOS	0x0038	ADC FIFO Status Register
AC_ADC_RXDATA	0x0040	ADC RX Data Register
AC_ADC_CNT	0x0044	ADC RX Counter Register
AC_ADC_DG	0x004C	ADC Debug Register
ADC_DIG_CTRL	0x0050	ADC Digtial Control Register
VRA1SPEEDUP_DOWN_CTRL	0x0054	VRA1Speedup Down Control Register
AC_DAC_DAP_CTRL	0x00F0	DAC DAP Control Register
AC_ADC_DAP_CTR	0x00F8	ADC DAP Control Register
AC_DAC_DRC_HHPFC	0x0100	DAC DRC High HPF Coef Register
AC_DAC_DRC_LHPFC	0x0104	DAC DRC Low HPF Coef Register
AC_DAC_DRC_CTRL	0x0108	DAC DRC Control Register
AC_DAC_DRC_LPFHAT	0x010C	DAC DRC Left Peak Filter High Attack Time Coef Register
AC_DAC_DRC_LPFLAT	0x0110	DAC DRC Left Peak Filter Low Attack Time Coef Register
AC_DAC_DRC_RPFHAT	0x0114	DAC DRC Right Peak Filter High Attack Time Coef Register
AC_DAC_DRC_RPFLAT	0x0118	DAC DRC Left Peak Filter Low Attack Time Coef

4.2 I2S/PCM

4.2.1 Overview

The I2S/PCM controller is designed to transfer streaming audio-data between the system memory and the codec chip. The controller supports standard I2S format, Left-justified mode format, Right-justified mode format, PCM mode format, and TDM mode format.

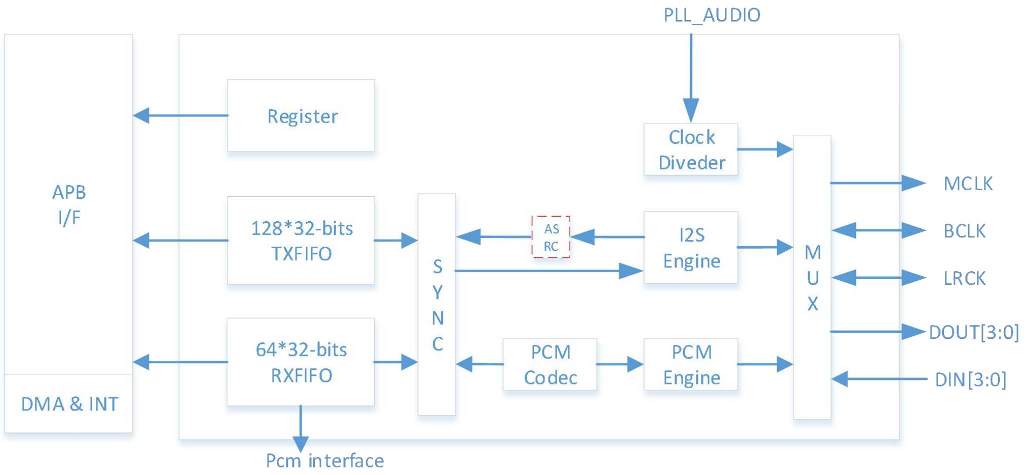
The I2S/PCM controller includes the following features:

- Four I2S/PCM external interfaces (I2S0, I2S1, I2S2, and I2S3) for connecting external power amplifier and MIC ADC
- Compliant with standard Philips Inter-IC sound (I2S) bus specification
  - Left-justified, Right-justified, PCM mode, and Time Division Multiplexing (TDM) format
  - Programmable PCM frame width: 1 BCLK width (short frame) and 2 BCLKs width (long frame)
- FIFOs for transmitting and receiving data
  - Programmable FIFO thresholds
  - 128 depth x 32-bit width TXFIFO and 64 depth x 32-bit width RXFIFO
- Supports multiple function clocks
  - Clock up to 24.576 MHz Data Output of I2S/PCM in Master mode (Only if the IO PAD and Peripheral I2S/PCM satisfy Timing Parameters)
  - Clock up to 12.288 MHz Data Input of I2S/PCM in Master mode
- Supports TX/RX DMA slave interface
- Supports multiple application scenarios
  - Up to 16 channels (fs = 48 kHz) which has adjustable width from 8-bit to 32-bit
  - Sample rate from 8 kHz to 384 kHz (sample rate * channel * slot width ≤ 24.576 MHz)
  - 8-bit u-law and 8-bit A-law companded sample
- Supports master/slave mode

4.2.2 Block Diagram

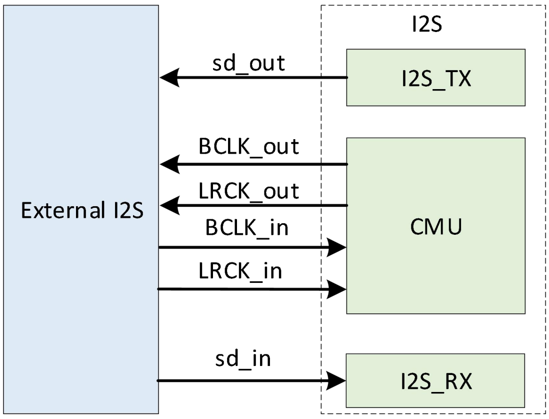
The following figure shows the functional block diagram of the I2S/PCM interface.

Figure 4-14 I2S/PCM Interface System Block Diagram



The following figure shows the typical application of the I2S/PCM interface.

Figure 4-15 Typical Application of I2S/PCM Interface



The I2S/PCM interface system integrates one I2S_TX and one I2S_RX.

- The I2S_TX is for playing music in I2S or PCM format.
- The I2S_RX is for receiving data in I2S or PCM format.
- When the I2S works in the master mode, the external I2S module provides BCLK_in and LRCK_in for the clock management unit (CMU), and the I2S_TX and I2S_RX work with the two external clocks.
- When the I2S works in the slave mode, the CMU provides clocks BCLK_out and LRCK_out for the external I2S module, and the I2S_TX and I2S_RX work with the internal clocks.

4.2.3 Functional Description

4.2.3.1 External Signals

The following table describes the external signals of the I2S/PCM interface.

LRCK and BCLK are bidirectional I/O. When the I2S/PCM interface works in the Master mode, LRCK and BCLK are output pins. When the I2S/PCM interface works in the Slave mode, LRCK and BCLK are input pins.

MCLK is an output pin for external devices. DOUT are the serial data output pins and DIN are the serial data input pins. For details about General Purpose I/O port, refer to section 8.5 GPIO.

Table 4-2 I2S/PCM External Signals

Signal Name	Description	Type
I2S0-DOUT[3:0]	I2S0/PCM0 Serial Data Output Channel [3:0]	O
I2S0-DIN[3:0]	I2S0/PCM0 Serial Data Input Channel [3:0]	I
I2S0-MCLK	I2S0 Master Clock	O
I2S0-LRCK	I2S0/PCM0 Sample Rate Clock/Sync	I/O
I2S0-BCLK	I2S0/PCM0 Bit Rate Clock	I/O
I2S1-DOUT[1:0]	I2S1/PCM1 Serial Data Output Channel [1:0]	O
I2S1-DIN[1:0]	I2S1/PCM1 Serial Data Input Channel [1:0]	I
I2S1-MCLK	I2S1 Master Clock	O
I2S1-LRCK	I2S1/PCM01Sample Rate Clock/Sync	I/O
I2S1-BCLK	I2S1/PCM1 Bit Rate Clock	I/O
I2S2-DOUT[3:0]	I2S2/PCM2 Serial Data Output Channel [3:0]	O
I2S2-DIN[3:0]	I2S2/PCM2 Serial Data Input Channel [3:0]	I
I2S2-MCLK	I2S2 Master Clock	O
I2S2-LRCK	I2S2/PCM2 Sample Rate Clock/Sync	I/O
I2S2-BCLK	I2S2/PCM2 Bit Rate Clock	I/O
I2S3-DOUT[3:0]	I2S3/PCM3 Serial Data Output Channel [3:0]	O
I2S3-DIN[3:0]	I2S3/PCM3 Serial Data Input Channel [3:0]	I
I2S3-MCLK	I2S3 Master Clock	O
I2S3-LRCK	I2S3/PCM3 Sample Rate Clock/Sync	I/O
I2S3-BCLK	I2S3/PCM3 Bit Rate Clock	I/O

4.2.3.2 Clock Sources

The following table describes the clock sources for I2S/PCM. For clock setting, configurations, and gating information, refer to section 2.11 Power Reset Clock Management (PRCM).

Table 4-3 I2S/PCM Clock Sources

Clock Source	Description	Module
PLL_AUODIO(4x)	By default, PLL_AUDIO(4X) is 98.2856 MHz.	CCU

Clock Source	Description	Module
PLL_AUDIO1 (DIV2)	By default, PLL_AUDIO1 is 3072 MHz, PLL_AUDIO1 (DIV2) is 1536 MHz, and PLL_AUDIO1 (DIV5) is 614.4 MHz (24.576 MHz*25).	PRCM
PLL_AUDIO1 (DIV5)		

4.2.3.3 Timing Diagram

The I2S/PCM supports standard I2S mode, Left-justified I2S mode, Right-justified I2S mode, PCM mode, and TDM mode. The software can select the modes by setting I2S/PCM_CTL. The following figures describe the waveforms for SYNC, BCLK, DOUT, and DIN in different modes.

Each sampling period contains an LRCK. The low level of LRCK is the left channel corresponding to the even slots, and the high level is the right channel corresponding to the odd slots. Each slot is the sampling point of a mono channel. The sampling period can support the transmission of 2/4/8/16 slots. The BCLK corresponds to the serial data bit.

Figure 4-16 I2S Standard Mode Timing

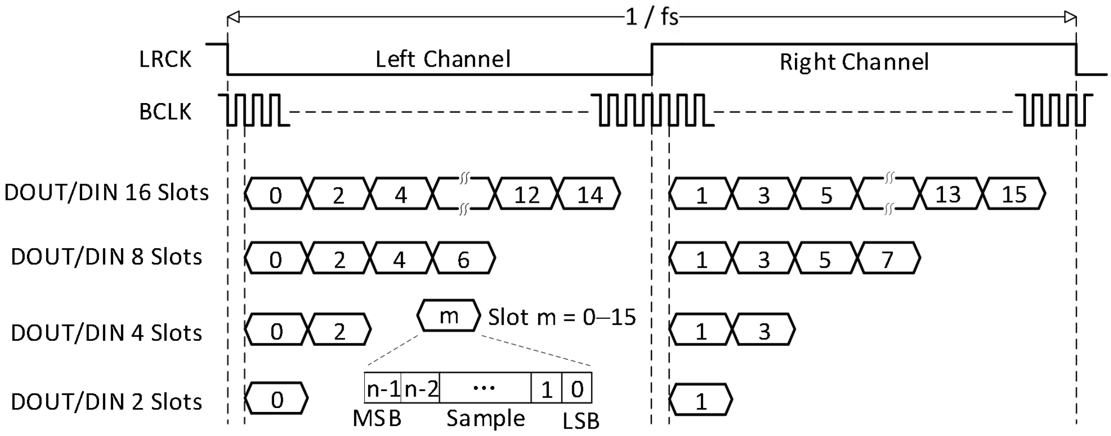


Figure 4-17 Left-Justified Mode Timing

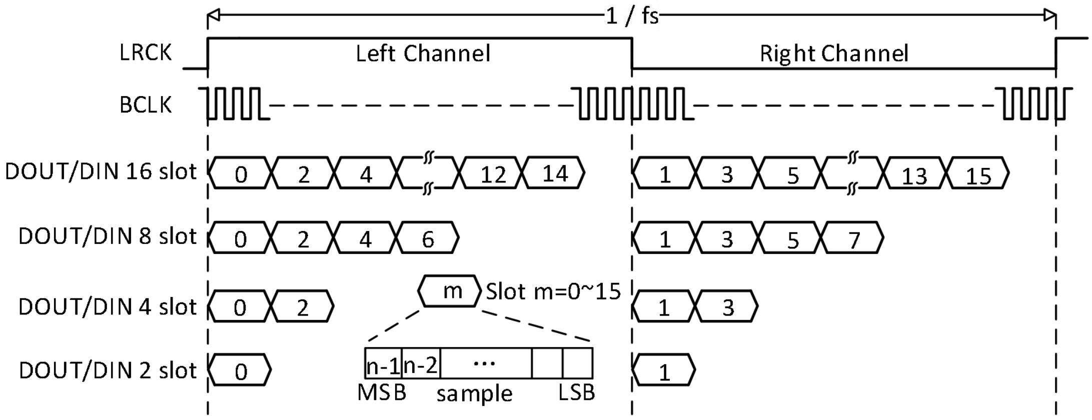


Figure 4-18 Right-Justified Mode Timing

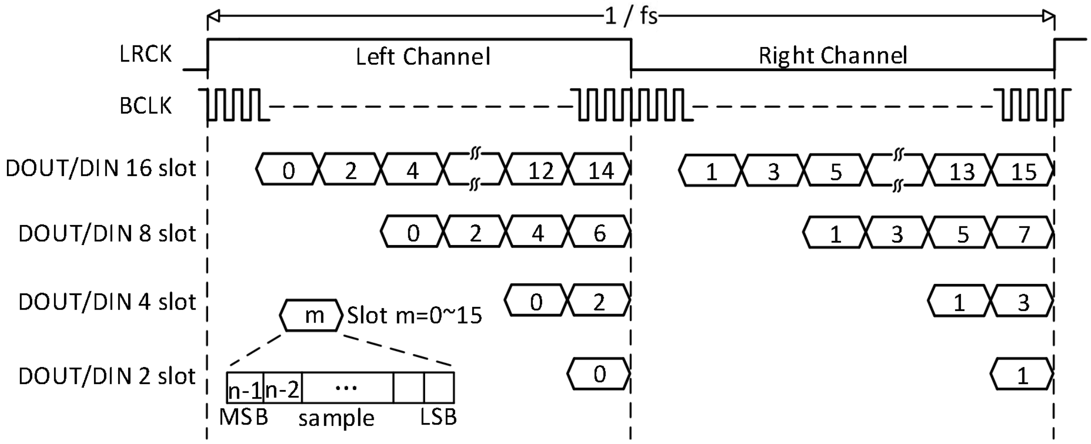


Figure 4-19 PCM Long Frame Mode Timing

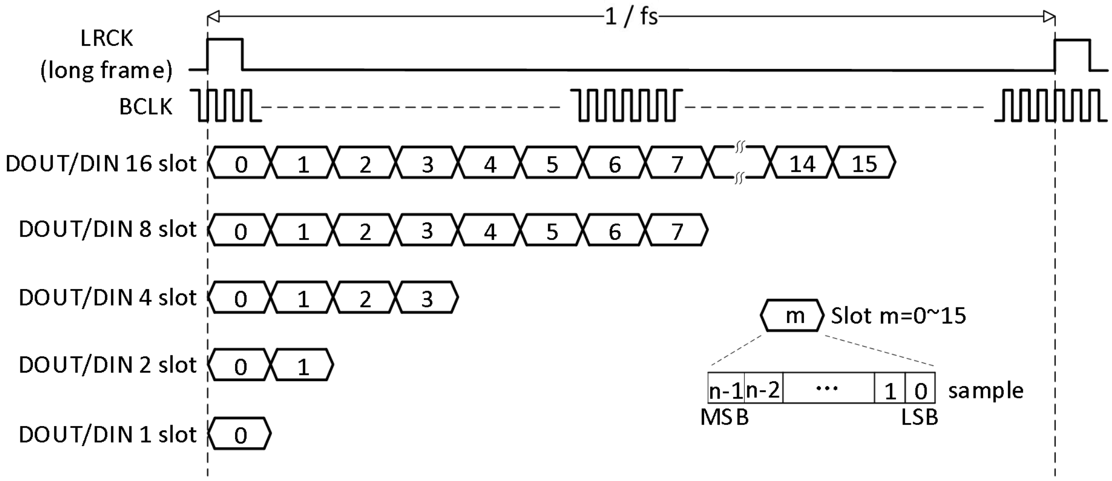
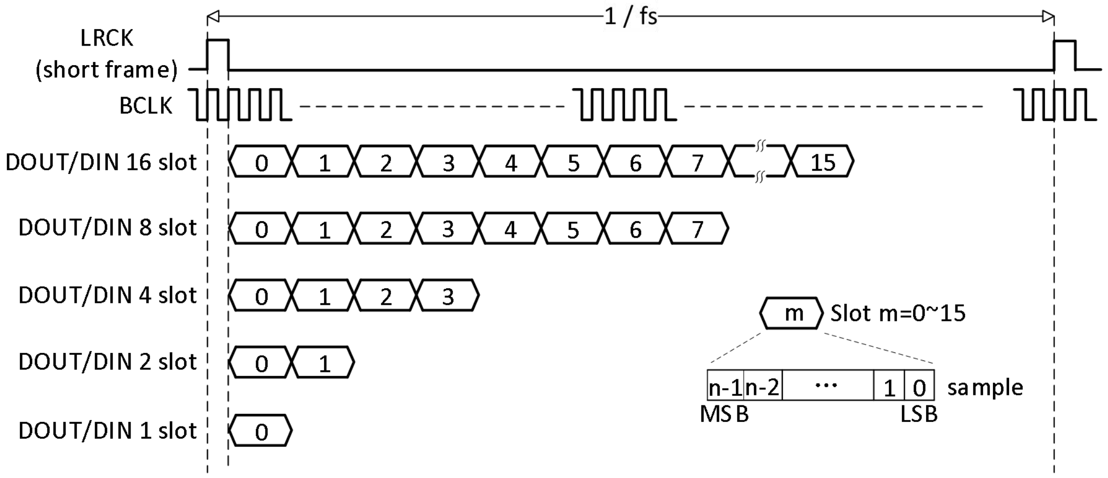


Figure 4-20 PCM Short Frame Mode Timing (one BCLK cycle)



4.2.3.4 ASRC

The ASRC module supports sampling rate conversion between the up-sampling and down-sampling. The ASRC also supports sampling rate conversion between dual-channel audio data, and the size of the sampling data is up to 24 bits.

The ASRC module has the following features:

- Typical THD + N: -130 dB (Range: -125 dB to -139 dB)
- Supports sampling rate conversion between the up-sampling and down-sampling to implement the sampling rate conversion for stereo data
  - The up-sampling ratio ranges from 1 to 7.5x
  - The down-sampling ratio ranges from 8 to 1x
- Supports sampling rate conversion between two identical frequencies
- Sampling rate for both the input and output range is from 8 kHz to 192 kHz and can be decimal
- Sampling rate can be configured manually or via adaptive generation
- The ASRC input is connected to I2S RX_FIFO_WDATA [31:8], and the input data is 24-bit MSB big-endian. For the input data that is less than 24 bits, use zeros to pad out the values at the low bits instead of high bits
- The ASRC needs some time to calculate the result. The output outsamplea/b will keep 0 during the calculation, and then change to the valid value when the result comes out

Calculating the ASRC Latency

Calculate the ASRC up-sampling and down-sampling latency according to the following formulas.

$$\text{Upsampling Latency} = \text{Phase Delay} + \text{FIFO Delay} = 32 + 16 = 48 \text{ Input Sample Periods}$$

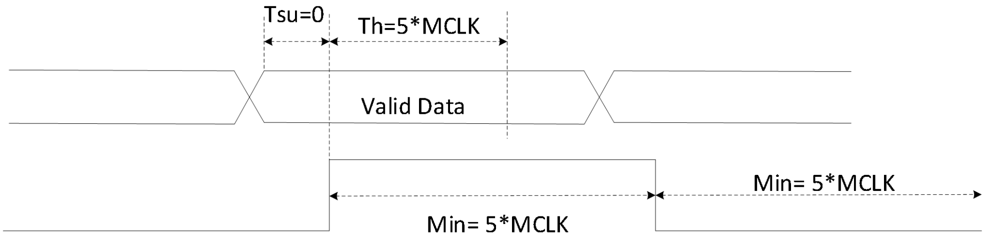
$$\text{Downsampli ng Latency} = \text{Phase Delay} + \text{FIFO Delay} = (32 * f_{sout}/f_{sin}) + 16 \text{ Input Sample Periods}$$

ASRC Timing

The MCLK samples the input clock CLKIN to generate pulse signals.

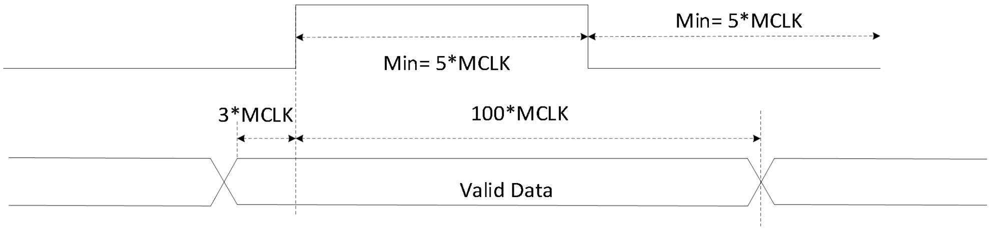
The following figure shows the timing requirements for the inputs.

Figure 4-21 Timing Requirements for Inputs



The following figure shows the timing requirements for the outputs.

Figure 4-22 Timing Requirements for Outputs



For the up-sampling, $F_{MCLK} = F_{sout} * 1350$

For the down-sampling, $F_{MCLK} = F_{sin} * 0.30 + F_{sout} * 295$

The following table provides the proper values of MCLK in MHz with different Fsin and Fsout in kHz.

Table 4-4 Proper MCLK Values with Different Fsin and Fsout

Fsout / Fsin	32	44.1	48	88.2	96	144	192
32	45	60	65	120	130	195	260
44.1	55	60	65	120	130	195	260
48	60	65	65	120	130	195	260
88.2	105	105	110	120	130	195	260
96	110	115	115	125	130	195	260
144	160	165	165	175	180	195	260
192	210	215	215	225	230	245	260

NOTE

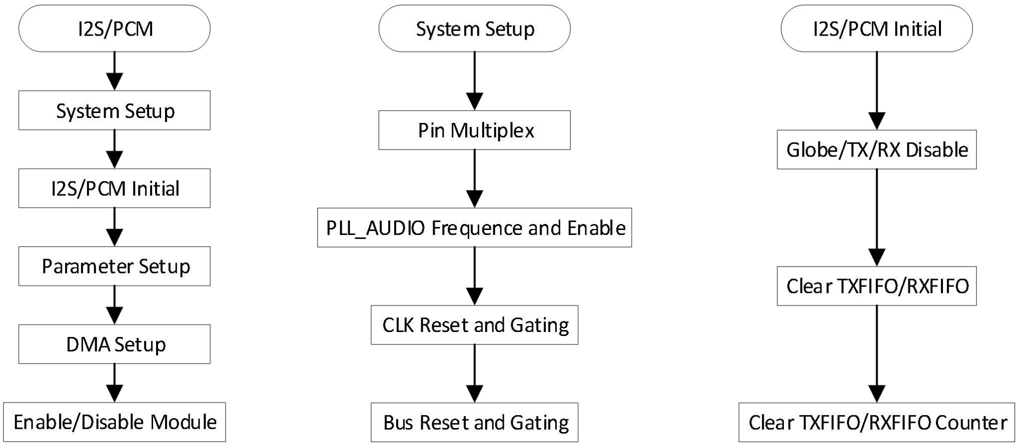
The units for Fsin and Fsout are kHz and MCLK is MHz.

4.2.3.5 Operation Modes

The software operation of the I2S/PCM is divided into five steps: system setup, I2S/PCM initialization, the channel setup, DMA setup, and Enable/Disable module.

The following figure shows the whole operation flow of I2S/PCM.

Figure 4-23 I2S/PCM Operation Flow



Step 1 System Setup and I2S/PCM Initialization

The clock source for the I2S/PCM should be followed. Firstly, disable the PLL_AUDIO through PLL_AUDIOx Control Register[PLL_ENABLE] in the CCU or PRCM. Secondly, set up the frequency of the PLL_AUDIO in the PLL_AUDIOx Control Register. After that, enable the I2S/PCM gating through the I2S/PCMx_CLK_REG when you checkout that the PLL_AUDIOx Control Register[LOCK] becomes to 1. At last, reset and enable the I2S/PCM bus gating by setting I2S_BGR_REG.

After the system setup, the register of I2S/PCM can be setup. Firstly, initialize the I2S/PCM. You should close the Globe Enable bit (I2S/PCM_CTL[0]), Transmitter Block Enable bit (I2S/PCM_CTL[2]), and Receiver Block Enable bit (I2S/PCM_CTL[1]) by writing 0. After that, clear the TX/RX FIFO by writing 0 to the bit[25:24] of I2S/PCM_FCTL. At last, you can clear the TX FIFO and RX FIFO counter by writing 0 to I2S/PCM_TXCNT and I2S/PCM_RXCNT.

Step 2 Parameter Setup and DMA Setup

First, you can set up the I2S/PCM of master and slave. The configuration can be referred to the protocol of I2S/PCM. Then, you can set up the translation mode, the sample resolution, the wide of the slot, the channel slot number, and the trigger level, and so on. The setup of the register can be found in the specification.

The I2S/PCM supports two methods to transfer the data. The most common way is DMA, the setup of DMA can be found in the section 2.6 DMA Controller (DMAC). In this module, you just enable the DRQ.

Step 3 Enable and Disable I2S/PCM

To enable the function, you can enable TX/RX by writing I2S/PCM_CTL[TXEN]/I2S/PCM_CTL[RXEN]. After that, enable I2S/PCM by writing 1 to I2S/PCM_CTL[Globe Enable]. Write 0 to the Globe Enable bit to disable I2S/PCM.

4.2.4 Programming Guidelines

4.2.4.1 Application Example of Processing ASRC Input and Output Data

The following example shows a typical application of ASRC: the input data is 24-bit valid, and the output data is a 32-bit data whose highest 24 bits are valid output and the lowest eight bits are padded out with zeros.

To implement the application, configure the sample resolution and slot width as 32 bits. Follow the steps below:

Step 1 For the input register: 0x04 [6:4] sample_res = 3`h7, 0x04 [2:0] slot_width = 3`h7.

The format of the input data: 32'hXXXXXXXX, where, bit [31] is the MSB and X is the valid data bit.

Step 2 For the output register: 0x04 [6:4] sample_res = 3`h7, 0x04 [2:0] slot_width = 3`h7

The format of the output data: 32'hXXXXXX00, where, bit [31] is the MSB, X is the valid data bit, and bit [7:0] are the padded zeros.

4.2.4.2 Converting the Sampling Rate with ASRC

Converting a 48 kHz sampling rate to 16 kHz is the most common scenario in actual applications. Follow the steps below to convert the sampling rate from 48 kHz to 16 kHz for the 32-bit data.

Step 1 Configure the PLL_AUDIO Register

a) Configure PLL_AUDIO_CTRL_REG[31:0] as 0x8814AB01. That is, PLL_AUDIO = 24*(171+1)/ (1+1)/ (1+0)/ (1+20) = 98.286 MHz. According to the relationship among the Fsin, Fsout, and MCLK, the MCLK should be greater than 60 MHz. In the simulation phase, the HOSC frequency is 25 MHz, so the output frequency of PLL_AUDIO should be 25*(171+1)/ (1+1)/ (1+0)/ (1+20) = 102.381 MHz. In the IC test phase, configure the frequency of PLL_AUDIO according to its actual output frequency.

b) It is suggested that you configure the ASRC MCLK as an equal-duty-cycle signal. You can specify an odd number for bit[21:16] (PLL_POST_DIV_P) of PLL_AUDIO_CTRL_REG to get an equal-duty-cycle output clock of PLL_AUDIO.

c) Configure bit[25:24] of I2S3_ASRC_CLK_REG as 0x00 to select the PLL_AUDIO(4X).

Step 2 Configure the I2S Registers

a) Configure bit[7:4] (BCLKDIV) of I2S/PCM_CLKD as 4`h9, that is, the frequency of BCLK will be 98.286 MHz/32 = 3.072 MHz.

b) Configure bit[17:8] (LRCK_PERIOD) of I2S/PCM_FMT0 as 10`h1F. That is, the LRCK_PERIOD width is configured as 32 BLCKs and can generate the ASRC CLKIN with a 48 kHz sampling rate. ($\frac{3.072\text{MHz}}{32*2} = 48\text{kHz}$)

c) Configure bit[6:4] (Sample Resolution bits) of I2S/PCM_FMT0 as 3`h7 to specify the sample resolution as 32-bit.

d) Configure bit[2:0] (Slot Width bits) of I2S/PCM_FMT0 as 3`h7 to specify the slot width as 32-bit.

Step 3 Configure the ASRC Registers

a) Configure bit[16] (clock gate) of MCLKCFG as 1`h1 to open the clock gating.

b) Configure bit[3:0] (division factor) of MCLKCFG as 1`h1 to specify the division factor as 1.

c) Configure bit[20] (clock gate) of FsoutCFG as 1`h1 to open the clock gating.

d) Configure bit[19:16] (clock select) of FsoutCFG as 4`h0 to select I2S0_ASRC_CLK as the clock source.

e) Configure bit[7:4] (the first division factor) of FsoutCFG as 16`h13 to configure the first division factor as 128.

f) Configure bit[3:0] (the second division factor) of FsoutCFG as 16`h10 to configure the second division factor as 48.

g) Configure the ASRC ratio.

To configure the ASRC ratio manually, configure bit[31] (Manual Configuration of ASRC Ratio Enable) of ASRCMANCFG as 1`h1 to enable the manual configuration of ASRC ratio. Configure bit[25:0] of ASRCMANCFG as 26`h155555 to specify the ratio value as 0x155555. The calculation formula for the ratio value: Dec2Hex (Fsout/Fsin) *222). In this example, Fsout/Fsin = 16 kHz/48 kHz =1/3, then the ratio is 0x155555.

To configure the ASRC ratio automatically, configure bit[31] (Manual Configuration of ASRC Ratio Enable) of ASRCMANCFG as 1`h0 to enable the automatic configuration of ASRC ratio. Then the system will automatically calculate the ratio value based on the MCLK, Fsout, and Fsin.

4.2.5 Register List

Module Name	Base Address	Comments
I2S|PCM0	0x07112000	Use for Speech Input.
I2S|PCM1	0x07113000	I2S|PCM1 register is the same with I2S|PCM0 .
I2S|PCM2	0x07114000	I2S|PCM2 register is the same with I2S|PCM0.
I2S|PCM3	0x07115000	I2S|PCM3 register is the same with I2S|PCM0.

Register Name	Offset	Description
I2S|PCM_CTL	0x0000	I2S|PCM Control Register
I2S|PCM_FMT0	0x0004	I2S|PCM Format Register 0
I2S|PCM_FMT1	0x0008	I2S|PCM Format Register 1

4.3 DMIC

4.3.1 Overview

The DMIC controller supports one 8-channel digital microphone interface and can output 128 fs or 64 fs (fs = ADC sample rate).

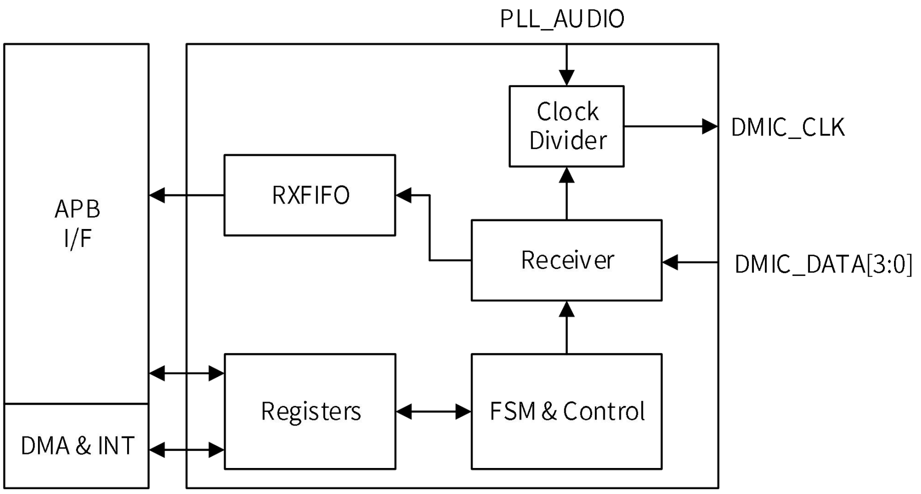
The DMIC controller includes the following features:

- Supports up to 8 channels
- Sample rate from 8 kHz to 48 kHz

4.3.2 Block Diagram

The following figure shows a block diagram of the DMIC.

Figure 4-24 DMIC Block Diagram



4.3.3 Functional Description

4.3.3.1 External Signals

The following table describes the external signals of DMIC.

Table 4-5 DMIC External Signals

Signal Name	Description	Type
DMIC-DATA[3:0]	Digital Microphone Data Input	I
DMIC-CLK	Digital Microphone Clock Output	O

4.3.3.2 Clock Sources

The following table describes the clock source for DMIC. For clock setting, configurations, and gating information, refer to section 2.11 Power Reset Clock Management (PRCM).

Table 4-6 DMIC Clock Sources

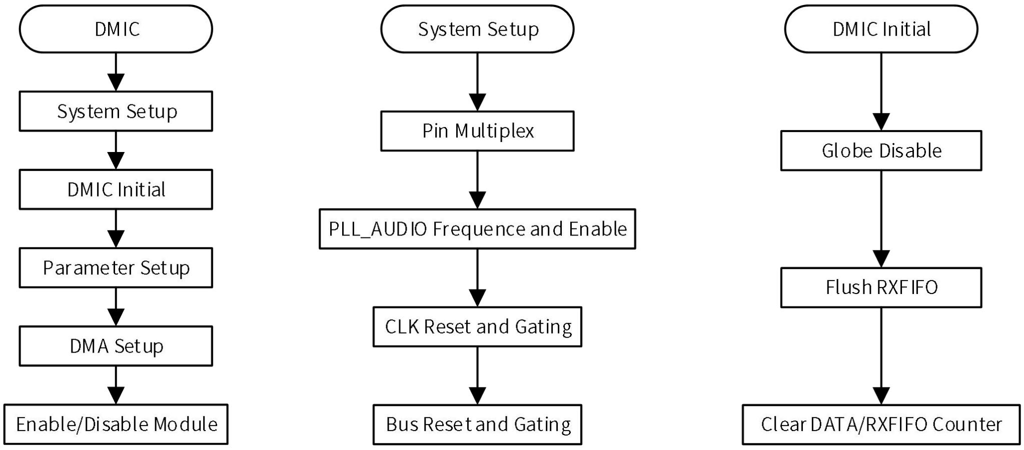
Clock Source	Description	Module
PLL_AUODIO(4x)	By default, PLL_AUDIO(4X) is 98.2856 MHz.	CCU
PLL_AUDIO1(DIV2)	By default, PLL_AUDIO1 is 3072 MHz, PLL_AUDIO1(DIV2) is 1536 MHz, and PLL_AUDIO1(DIV5) is 614.4 MHz (24.576 MHz*25).	PRCM
PLL_AUDIO1(DIV5)		

4.3.3.3 Operation Mode

The software operation of the DMIC is divided into five steps: system setup, DMIC initialization, channel setup, DMA setup, and Enable/Disable module.

The following figure shows the flow chart of the whole operation, the system setup, and the DMIC initialization.

Figure 4-25 DMIC Operation Mode



Step 1 System Setup and DMIC Initialization

The first step in the system setup is properly programming the GPIO because the DMIC port is a multiplex pin. For functions of the multiplex pins, refer to section 8.5 GPIO.

Perform the following steps for the clock source. Firstly, disable the PLL_AUDIO through PLL_AUDIOx Control Register[PLL_ENABLE]. Secondly, set up the frequency of the PLL_AUDIO in PLL_AUDIOx Control Register. Then enable PLL_AUDIO. After that, enable the DMIC gating through DMIC_CLK_REG when you checkout that the LOCK bit of PLL_AUDIOx Control Register becomes 1. At last, reset and enable the DMIC bus gating by DMIC_BGR_REG.

After the system setup, the register of DMIC can be setup. Firstly, initialize the DMIC. You should close the globe enable bit (DMIC_EN[8]), data channel enable bit (DMIC_EN[7:0]) by writing 0 to it. After that, flush the RXFIFO by writing 1 to DMIC_RXFIFO_CTR[31]. At last, you can clear the Data/RXFIFO counter by writing 1 to DMIC_RXFIFO_STA, DMIC_CNT.

Step 2 Parameter Setup and DMA Setup

You can set up the sample rate, the sample resolution, the over-sample rate, the channel number, the RXFIFO output mode, the RXFIFO trigger level, and so on. The setup of the register can be found in the specification.

The DMIC supports two methods to transfer the data. The most common way is DMA, the setup of DMA can be found in section 2.6 DMA Controller (DMAC). In this module, you just enable the DRQ.

Step 3 Enable and Disable DMIC

To enable the function, you can enable the data channel enable bit (DMIC_EN[7:0]) by writing 1 to it. After that, enable DMIC by writing 1 to the Globe Enable bit (DMIC_EN[8]). Write 0 to DMIC_EN[8] to disable DMIC

4.3.4 Register List

Module Name	Base Address
DMIC	0x0711 1000

Register Name	Offset	Description
DMIC_EN	0x0000	DMIC Enable Control Register
DMIC_SR	0x0004	DMIC Sample Rate Register
DMIC_CTR	0x0008	DMIC Control Register
DMIC_DATA	0x0010	DMIC DATA Register
DMIC_INTC	0x0014	DMIC Interrupt Control Register
DMIC_INTS	0x0018	DMIC Interrupt Status Register
DMIC_RXFIFO_CTR	0x001C	DMIC RXFIFO Control Register
DMIC_RXFIFO_STA	0x0020	DMIC RXFIFO Status Register
DMIC_CH_NUM	0x0024	DMIC Channel Numbers Register
DMIC_CH_MAP	0x0028	DMIC Channel Mapping Register
DMIC_CNT	0x002C	DMIC Counter Register
DATA0_DATA1_VOL_CTR	0x0030	DATA0 And DATA1 Volume Control Register
DATA2_DATA3_VOL_CTR	0x0034	DATA2 And DATA3 Volume Control Register
HPF_EN_CTR	0x0038	High Pass Filter Enable Control Register
HPF_COEF_REG	0x003C	High Pass Filter Coef Register
HPF_GAIN_REG	0x0040	High Pass Filter Gain Register
DMIC_REV	0x0050	DMIC Revision Register

4.4 One Wire Audio (OWA)

4.4.1 Overview

The One Wire Audio (OWA) provides a serial bus interface for audio data. This interface is widely used for consumer audio.

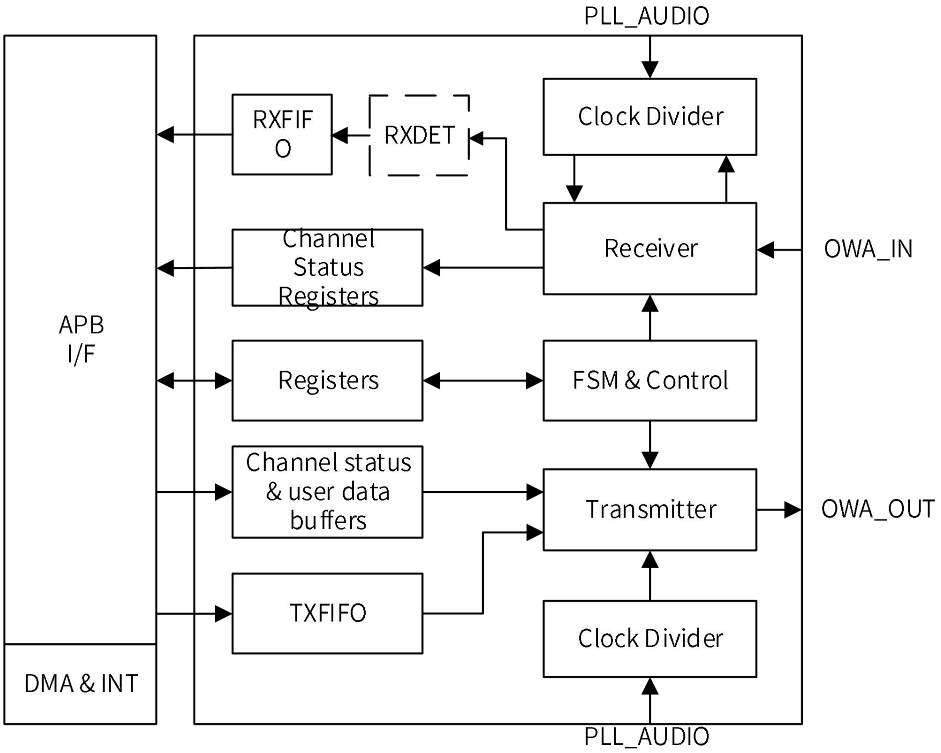
The OWA includes the following features:

- One OWA TX and One OWA RX
- Compliance with S/PDIF interface
- IEC-60958 and IEC-61937 transmitter and receiver functionality
- IEC-60958 supports data formats: 16 bits, 20 bits, and 24 bits
- TXFIFO and RXFIFO
  - One 128×24bits TXFIFO and one 64×24bits RXFIFO for audio data transfer
  - Programmable FIFO thresholds
- Supports TX/RX DMA slave interface
- Multiple function clock
  - Separate clock for OWA TX and OWA RX
  - The clock of TX function includes 24.576 MHz and 22.5792 MHz
  - The clock of RX function includes 24.576*8 MHz
- Supports hardware parity on TX/RX
  - Hardware parity generation on the transmitter
  - Hardware parity checking on the receiver
- Supports channel status capture on the receiver
- Supports channel sample rate capture on the receiver
- Supports insertion detection for the receiver
- Supports channel status insertion for the transmitter

4.4.2 Block Diagram

The following figure shows the OWA block diagram.

Figure 4-26 OWA Block Diagram



OWA contains the following sub-blocks:

Table 4-7 OWA Sub-blocks

Sub-block	Description
Registers	Analyze the configuration parameter, DMA requests, and IRQ feedbacks.
Receiver	Parses the frame header and receives the data.
Transmitter	Sends the data
FSM	Finite state machine
Clock Divider	Clock divider circuit

4.4.3 Functional Description

4.4.3.1 External Signals

The OWA is a Biphasic-Mark Encoding Digital Audio Transfer protocol. In this protocol, the CLK signal and data signal are transferred in the same line. The following table describes the external signals of OWA. OWA-OUT is the output pin for the output CLK and DATA, and OWA-IN is the input pin for the input CLK and DATA.

Table 4-8 OWA External Signals

Signal Name	Description	Type
OWA-IN	One Wire Audio Input	I
OWA-OUT	One Wire Audio Output	O

4.4.3.2 Clock Sources

The OWA has separate clock for OWA_TX and OWA_RX. The following tables describe the clock sources for OWA_TX and OWA_RX. For clock setting, configurations and gating information, refer to section 2.11 Power Reset Clock Management (PRCM).

Table 4-9 OWA_TX Clock Sources

Clock Sources	Description	Module
PLL_AUODIO(4x)	By default, PLL_AUDIO(4X) is 98.2856 MHz.	CCU
PLL_AUDIO1(DIV2)	By default, PLL_AUDIO1 is 3072 MHz, PLL_AUDIO1(DIV2) is 1536 MHz, and PLL_AUDIO1(DIV5) is 614.4 MHz (24.576 MHz*25).	PRCM
PLL_AUDIO1(DIV5)		

Table 4-10 OWA_RX Clock Sources

Clock Sources	Description	Module
PERI0_300M	By default, PERI0_300M is 300 MHz.	CCU
PLL_AUDIO1(DIV2)	By default, PLL_AUDIO1 is 3072 MHz, PLL_AUDIO1(DIV2) is 1536 MHz, and PLL_AUDIO1(DIV5) is 614.4 MHz (24.576 MHz*25).	PRCM
PLL_AUDIO1(DIV5)		

4.4.3.3 Biphase-Mark Code (BMC)

In the OWA format, the digital signal is coded using the biphase-mark code (BMC). The clock, frame, and data are embedded in only one signal—the data pin. In the BMC system, each data bit is encoded into two logical states (00, 01, 10, or 11) at the pin. The following figure and table show how data is encoded to the BMC format.

The frequency of the clock is twice the data bit rate, as shown in the following figure. Also, the clock is always programmed to 128 x fs, where fs is the sample rate. The device receiving in the OWA format can recover the clock and frame information from the BMC signal.

Figure 4-27 OWA Biphase-Mark Code

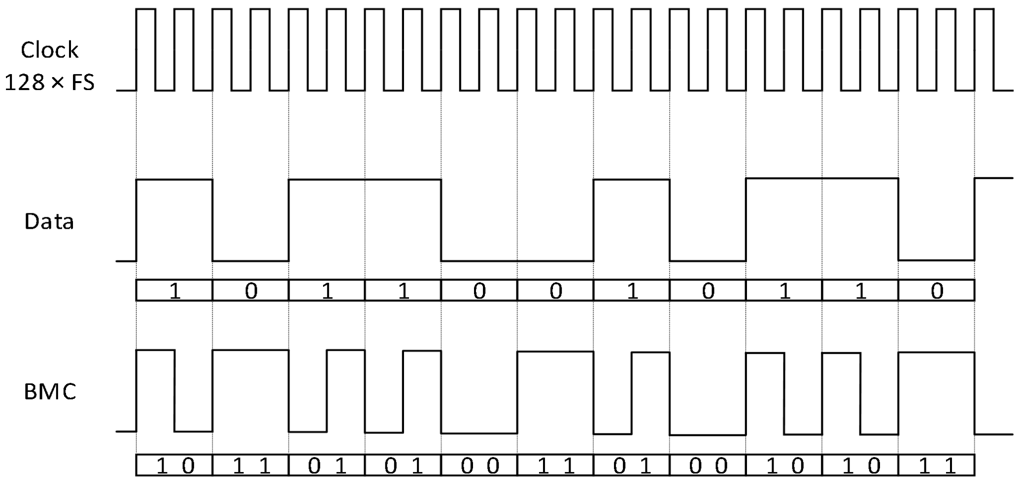


Table 4-11 Biphase-Mark Encoder

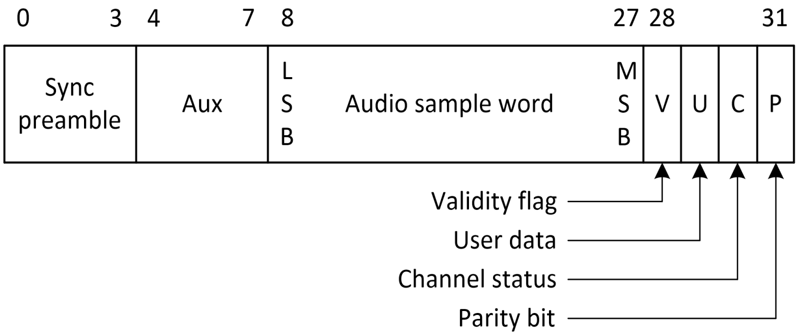
Data	Previous State	BMC
0	0	11
0	1	00
1	0	10
1	1	01

4.4.3.4 IEC60958 Transmit Format

The OWA supports digital audio data transfer and receive. It also supports full-duplex synchronous work mode. The software can set the work mode by the OWA Control Register.

Every audio sample transmitted in a sub-frame consists of 32-bit, numbered from 0 to 31. The following figure shows a sub-frame.

Figure 4-28 OWA Sub-Frame Format



Bits 0-3 carry one of the four permitted preambles to signify the type of audio sample in the current sub-frame. The preamble is not encoded in BMC format, and therefore the preamble code can contain more than two consecutive 0 or 1 logical states in a row.

Bits 4-27 carry the audio sample word in linear 2s-complement representation. The most-significant bit (MSB) is carried by bit 27. When a 24-bit coding range is used, the least-significant bit (LSB) is in bit 4. When a 20-bit coding range is used, bit [8:27] carry the audio sample word with the LSB in bit 8. Bit [4:7] may be used for other applications and are designated auxiliary sample bits.

If the source provides fewer bits than the interface allows (either 20 or 24), the unused LSBs are set to logical 0. For a nonlinear PCM audio application or a data application, the main data field may carry any other information.

Bit 28 carries the validity bit (V) associated with the main data field in the sub-frame.

Bit 29 carries the user data channel (U) associated with the main data field in the sub-frame.

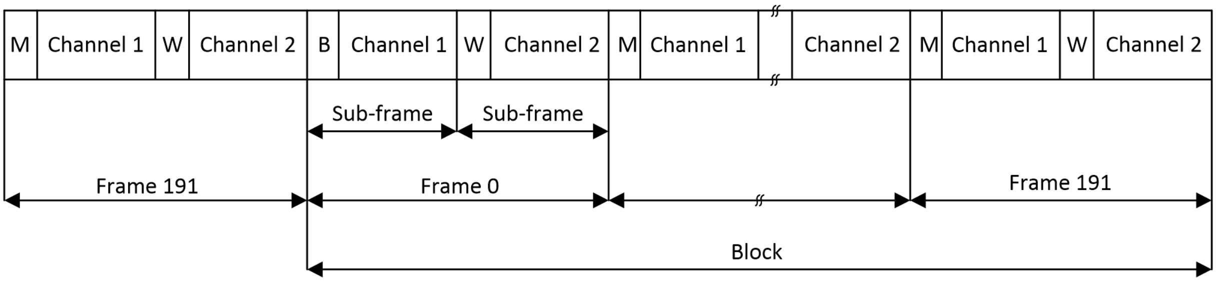
Bit 30 carries the channel status information (C) associated with the main data field in the sub-frame. The channel status indicates if the data in the sub-frame is a digital audio or some other type of data.

Bit 31 carries a parity bit (P) such that bit 4-31 carry an even number of 1s and an even number of 0s (even parity). As shown in the following table, the preambles (bit 0-3) are also defined with even parity.

Table 4-12 Preamble Codes

Preamble Code	Previous Logical State	Logical State	Description
B (or Z)	0	1110 1000	Start of a block and sub-frame 1
M (or X)	0	1110 0010	Sub-frame 1
W (or Y)	0	1110 0100	Sub-frame 2

Figure 4-29 OWA Frame/Block Format



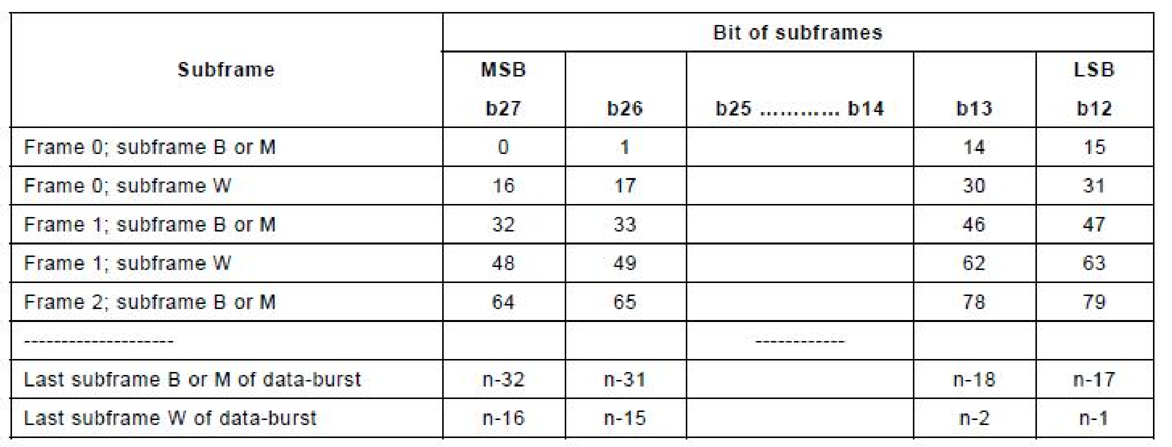
4.4.3.5 IEC61937 Transmit Format

IEC 61937 applies to the digital audio interface by using the IEC 60958 series for the conveying of non-linear PCM encoded audio bitstreams. The non-linear PCM encoded audio bitstream is transferred by using the basic 16-bit data area of the IEC 60958 subframes, i.e. in time-slots 12 to 27. Because the non-linear PCM encoded audio bitstream to be transported is at a lower data rate than that supported by the IEC 60958 interface, the audio bitstream is broken into a sequence of discrete data-bursts, and stuffing between the data-bursts is necessary.

IEC 60958 Data Burst

The method of placing the data into the IEC 60958 bitstream is to format the data to be transmitted into data-bursts and to send each data-burst in a continuous sequence of IEC 60958 frames.

Table 4-13 Bit Allocation of Data-Burst in IEC 60958 Subframes

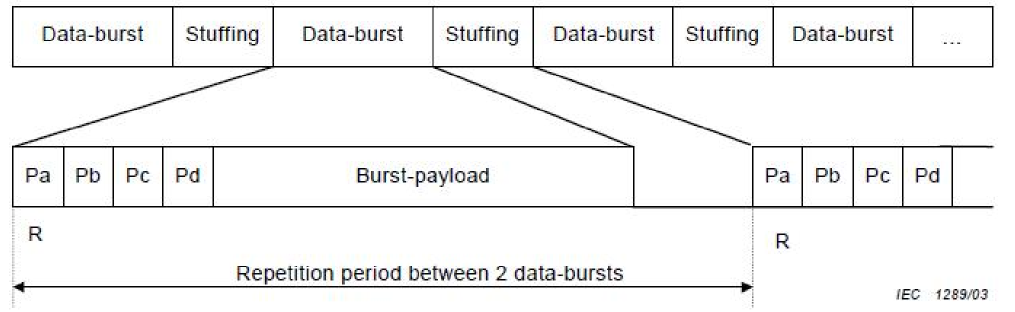


Subframe	Bit of subframes				
	MSB				LSB
	b27	b26	b25 ………. b14	b13	b12
Frame 0; subframe B or M	0	1		14	15
Frame 0; subframe W	16	17		30	31
Frame 1; subframe B or M	32	33		46	47
Frame 1; subframe W	48	49		62	63
Frame 2; subframe B or M	64	65		78	79
--------------------			------------		
Last subframe B or M of data-burst	n-32	n-31		n-18	n-17
Last subframe W of data-burst	n-16	n-15		n-2	n-1

Data Burst Format

Each data-burst contains a burst-preamble consisting of four 16-bit words (Pa, Pb, Pc and Pd) followed by the burst-payload which contains data of an encoded audio frame.

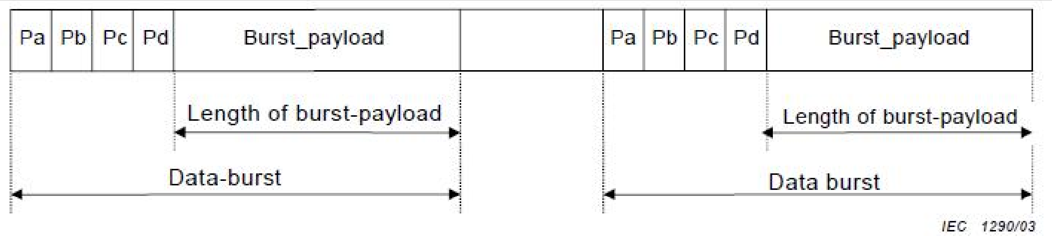
Figure 4-30 Data-Burst Format



- Burst-preamble

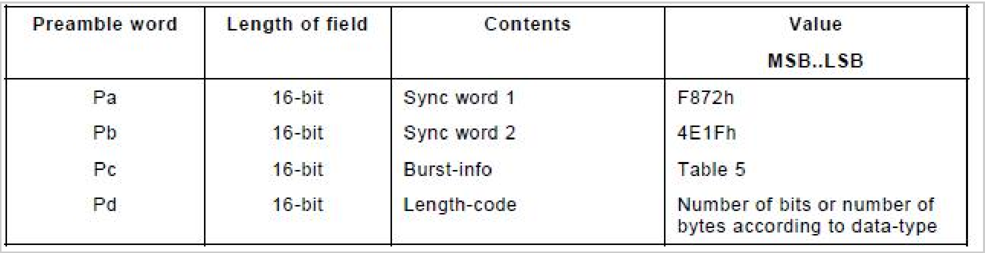
The burst-preamble consists of four mandatory fields. Pa and Pb represent a synchronization word. Pc gives information about the type of data, and some information/control for the receiver. Pd gives the length of the burst-payload, and is limited to 65535 bits in the case of Pd represent bits' length, or is limited to 65535 bytes in the case of Pd represent bytes' length.

Figure 4-31 Data-burst Preamble



The four preamble words are contained in two sequential IEC 60958 frames. The frame beginning the data-burst contains preamble word Pa in subframe 1, and Pb in subframe 2. The next frame contains Pc in subframe 1 and Pd in subframe 2. When placed into an IEC 60958 subframe, the MSB of a 16-bit burst-preamble word is placed into time-slot 27 and the LSB is placed into time-slot 12.

Figure 4-32 Data-burst Preamble words

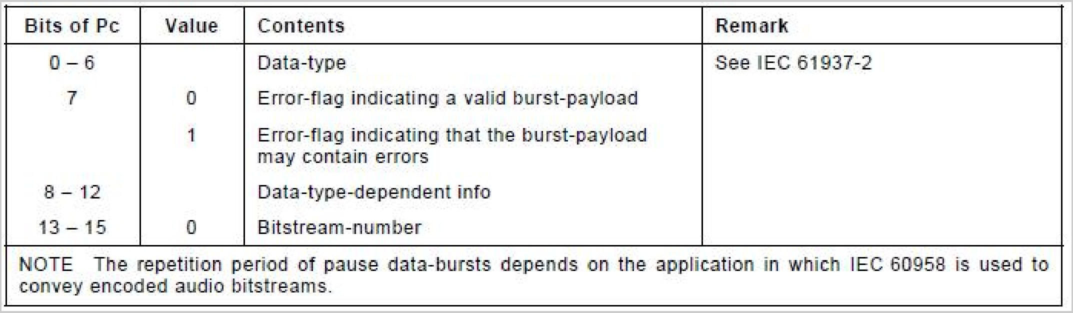


Preamble word	Length of field	Contents	Value MSB..LSB
Pa	16-bit	Sync word 1	F872h
Pb	16-bit	Sync word 2	4E1Fh
Pc	16-bit	Burst-info	Table 5
Pd	16-bit	Length-code	Number of bits or number of bytes according to data-type

- Burst-information

The 16-bit burst-information contains information about the data which will be found in the data-burst.

Figure 4-33 Fields of Burst-information



Bits of Pc	Value	Contents	Remark
0 – 6		Data-type	See IEC 61937-2
7	0	Error-flag indicating a valid burst-payload	
	1	Error-flag indicating that the burst-payload may contain errors	
8 – 12		Data-type-dependent info	
13 – 15	0	Bitstream-number	

NOTE The repetition period of pause data-bursts depends on the application in which IEC 60958 is used to convey encoded audio bitstreams.

The 7-bit data-type is defined in bits 0-6 of the burst-preamble Pc, the bit 6 is the MSB. This data-type field indicates the format of the burst-payload, which will be conveyed in the data-burst. Typical properties of a data-type are the reference point and repetition period of the burst, which is the number of sampling periods of the audio between the reference point of the current data-burst and the reference point of the next data-burst. The reference point is inherently defined for each data-type.

The error-flag bit is available to indicate if the contents of the data-burst contain data errors. If a data-burst is thought to be error-free, or if the data source does not know if the data contains errors, then the value of this bit is set to a ‘0’. If the data source does know that a particular data-burst contains some errors this bit may be set to a ’1’. The usage of this bit by receiver is optional.

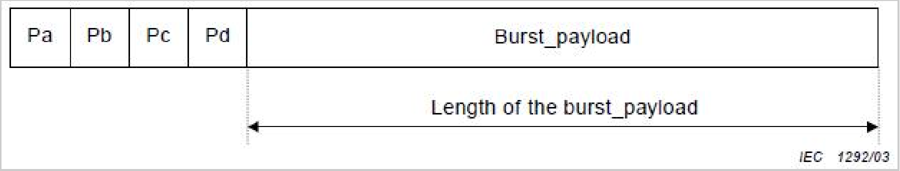
The meaning of the 5-bit data-type-dependent info depends on the value of the data-type.

The 3-bit bitstream-number indicates to which bitstream the data-burst belongs. Eight codes (0-7) are available so that up to eight independent bitstreams may be multiplexed in one bitstream in a time multiplex. Each independent bitstream shall use a unique bit-streamnumber.

- Length-code

The length-code indicates the number of bits or bytes according to data-type within the databurst, from 0 to 65535. The size of the Pa, Pb, Pc and Pd is not counted in the value of the length-code. In other words, the length-code indicates the number of bits of the burst-payload in bits, plus the conditional length of Pe and Pf, or the number of bytes of the burst-payload in bytes, plus the conditional length of Pe and Pf if exists.

Figure 4-34 Length of the Burst-Payload Specified by Pd



4.4.3.6 Audio Sample Ratio Detection

The sampling rate is calculated according to the data pulse back-stepping method. In the first phase lock of the CDR, find 1 Frame period, count by using the high-speed sampling clock, and read the counting value of the pulse, then the sampling rate can be calculated.

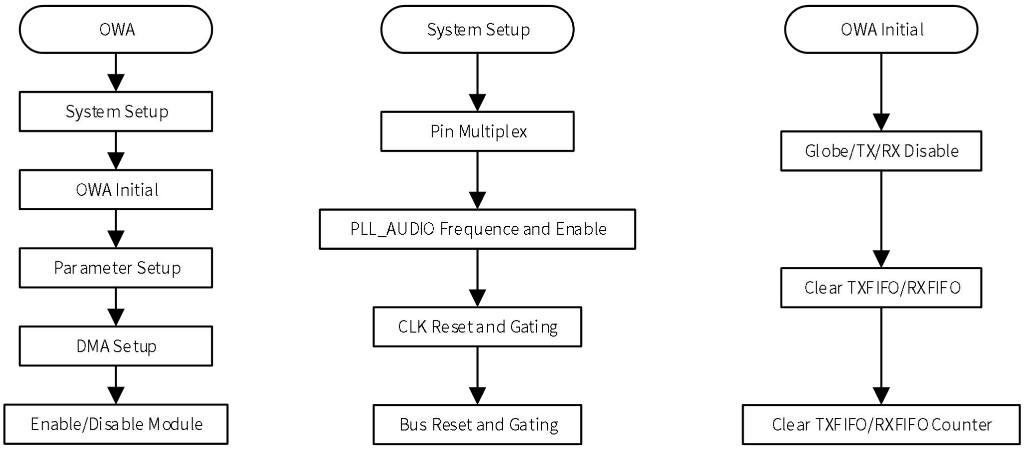
Table 4-14 The Corresponding Relation between Different System Clock and Sample Ratio

	Sample Clock Cycles		
TX Sample Rate(kHz)	196.608 MHz-SysClk	200 MHz-SysClk	300 MHz-SysClk
22.05	8916(±5)	9070(±5)	13605(±5)
24	8192(±5)	8333(±5)	12500(±5)
32	6144(±5)	6250(±5)	9375(±5)
44.1	4458(±5)	4535(±5)	6802(±5)
48	4096(±5)	4166(±5)	6250(±5)
96	2048(±5)	2083(±5)	3125(±5)
176.4	1114(±5)	1133(±5)	1700(±5)
192	1024(±5)	1041(±5)	1562(±5)

4.4.3.7 Operation Modes

The software operation of the OWA is divided into five steps: system setup, OWA initialization, channel setup, DMA setup and enable/disable module. The following sections describe these five steps.

Figure 4-35 OWA Operation Flow



Step 1 System Setup and OWA Initialization

The first step in the OWA initialization is properly programming the GPIO because the OWA port is a multiplex pin. You can find the function in section 8.5 GPIO.

The clock source for the OWA should be followed. Firstly, reset the audio PLL in PLL_AUDIOx Control Register. Secondly, set up the frequency of the Audio PLL in the PLL_AUDIOx Control Register. After that, enable the OWA gating. Lastly, enable the OWA bus gating.

After the system setup, the register of OWA can be set up. Firstly, reset the OWA by writing 1 to OWA_CTL[0] and clear the TX/RX FIFO by writing 1 to OWA_FCTL[17:16]. After that, enable the globe enable bit by writing 1 to OWA_CTL[1] and clear the interrupt and TX/RX counter by setting OWA_ISTA and OWA_TX_CNT/OWA_RX_CNT.

Step 2 Parameter Setup and DMA Setup

You can set up the audio type, clock divider ratio, the sample format, and the trigger level, and so on. The setup of the register can be found in the specification.

The OWA supports two methods to transfer the data. The most common way is DMA; the configuration of DMA can be found in section 3.9 “DMAC”. In this module, you just enable the DRQ in OWA_INT[7].

Step 3 Enable and Disable OWA

To enable the function, you can enable TX/RX by writing OWA_TX_CFIG[31]/OWA_RX_CFIG[0]. After that, enable OWA by writing 1 to OWA_CTL[1]. Writing 0 to OWA_CTL[1] to disable process.

4.4.4 Programming Guidelines

4.4.4.1 Configuring RX CDR

The RX_CDR_MANUAL bit (bit [5]) of OWA_RX_CFIG register decides whether manual mode or automatic mode is used for clock recovery.

- Automatic Mode (default mode)

Configure RX_CDR_MANUAL bit as 1'b0 to choose automatic mode.

- Manual Mode

The manual mode could be chosen by configuring RX_CDR_MANUA bit as 1'b1. In this mode, you can select RX CDR from preamble X (11100010) or preamble X (00011101) by configuring the RX_CDR_PREAMBLE bit (bit [2]) of OWA_RX_CFIG register. The operation steps are as follows:

Step 1 Configure the INSERT_INT_EN bit (bit [16]) of OWA_EXP_ISTA register to enable insert interrupt.

Step 2 Insert TX device and wait for a while.

Step 3 Keep reading the INSERT_INT bit (bit [0]) of OWA_EXP_ISTA register until INSERT_INT = 1.

Step 4 Keep reading the RX_LOCK_FLAG bit (bit [4]) of OWA_RX_CFIG register for a software-setting period.

- If the value of RX_LOCK_FLAG bit turns to 1, it is unnecessary to configure the RX_CDR_PREAMBLE bit (bit [2]) of OWA_RX_CFIG register.
- If the value of RX_LOCK_FLAG bit doesn't turn to 1, configure the RX_CDR_PREAMBLE bit (bit [2]) of OWA_RX_CFIG register to the opposite value.

4.4.5 Register List

Module Name	Base Address
OWA	0x0711 6000

-

Register Name	Offset	Description
OWA_GEN_CTL	0x0000	OWA General Control Register
OWA_TX_CFIG	0x0004	OWA TX Configuration Register
OWA_RX_CFIG	0x0008	OWA RX Configuration Register
OWA_ISTA	0x000C	OWA Interrupt Status Register
OWA_RXFIFO	0x0010	OWA RXFIFO Register
OWA_FCTL	0x0014	OWA FIFO Control Register
OWA_FSTA	0x0018	OWA FIFO Status Register
OWA_INT	0x001C	OWA Interrupt Control Register
OWA_TX_FIFO	0x0020	OWA TX FIFO Register
OWA_TX_CNT	0x0024	OWA TX Counter Register
OWA_RX_CNT	0x0028	OWA RX Counter Register
OWA_TX_CHSTA0	0x002C	OWA TX Channel Status Register0
OWA_TX_CHSTA1	0x0030	OWA TX Channel Status Register1
OWA_RXCHSTA0	0x0034	OWA RX Channel Status Register0
OWA_RXCHSTA1	0x0038	OWA RX Channel Status Register1