8.1.3 Functional Description

8.1.3.1 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 9.7 "GPIO".

Table 8-1 I2S/PCM External Signals

Signal Name	Description	Type
I2S1-MCLK	I2S1 Master Clock	O
I2S1-LRCK	I2S1/PCM1 Sample Rate Clock/Sync	I/O
I2S1-BCLK	I2S1/PCM1 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
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
I2S2-DOUT[2:0]	I2S2/PCM2 Serial Data Output Channel [2:0]	O
I2S2-DIN[2:0]	I2S2/PCM2 Serial Data Input Channel [2:0]	I

8.1.3.2 Clock Sources

The following table describes the clock sources for I2S/PCM. For clock setting, configurations, and gating information, refer to section 3.3 "CCU".

Table 8-2 I2S/PCM Clock Sources

Clock Name	Description
PLL_AUDIO0(1X)	By default, PLL_AUDIO0(1X) is 24.5714 MHz, and PLL_AUDIO0(4X) is 98.2856 MHz.
PLL_AUDIO0(4X)	
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).
PLL_AUDIO1(DIV5)	

8.1.3.3 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 8-3 I2S Standard Mode Timing

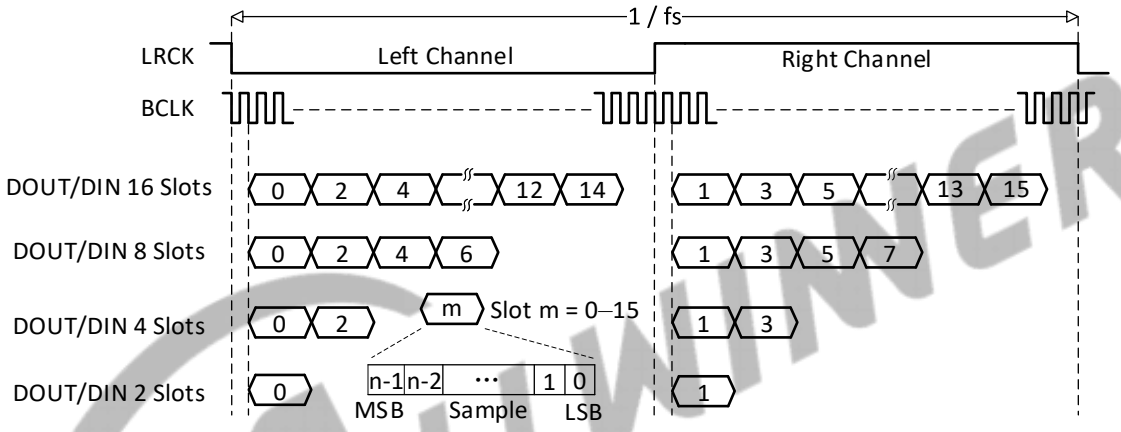


Figure 8-4 Left-Justified Mode Timing

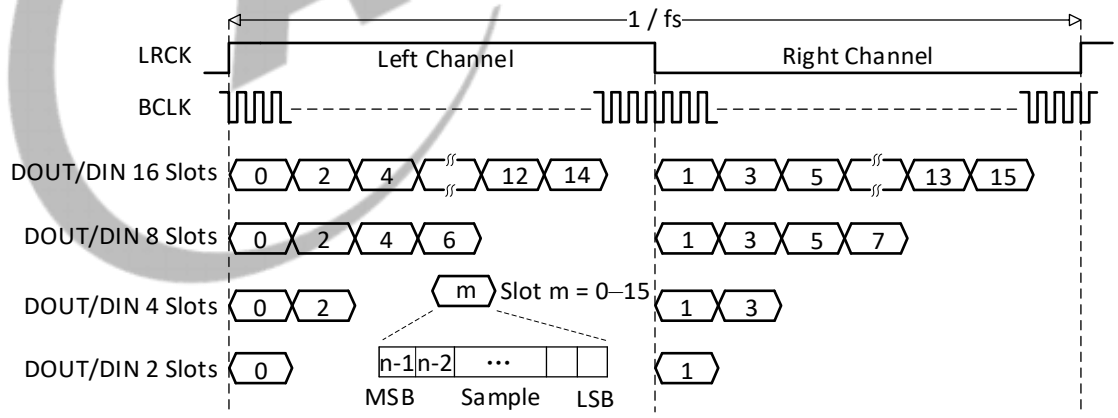


Figure 8-5 Right-Justified Mode Timing

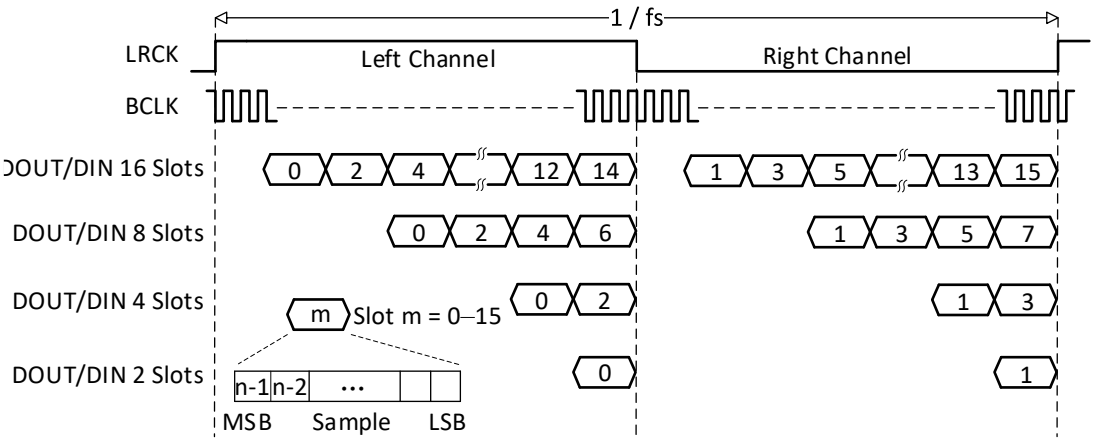


Figure 8-6 PCM Long Frame Mode Timing

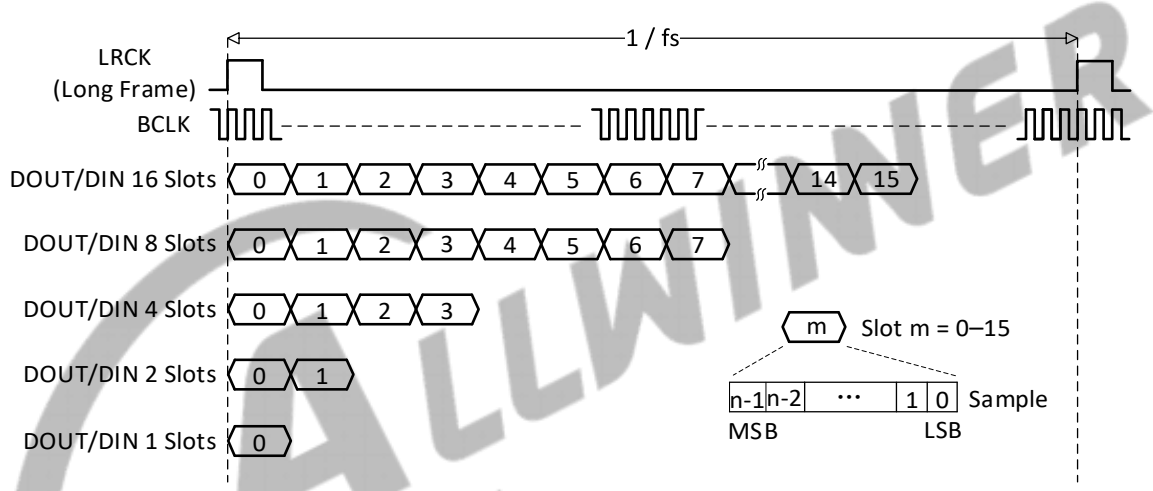
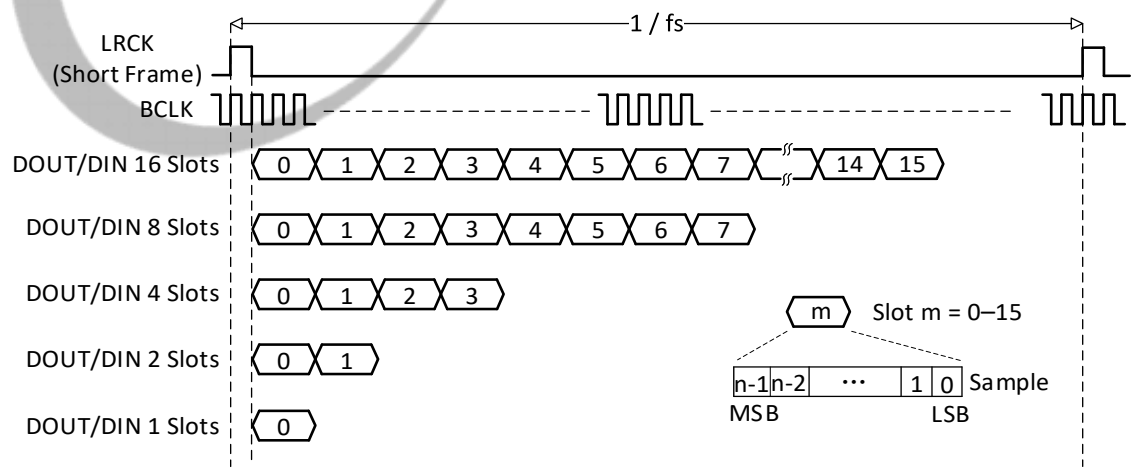


Figure 8-7 PCM Short Frame Mode Timing



8.1.3.4 DIN Slot Mapping

The 4-wire DIN has 64 slots, each wire DIN has 16 slots. However, only 16 slots are valid and act as the RX channels.

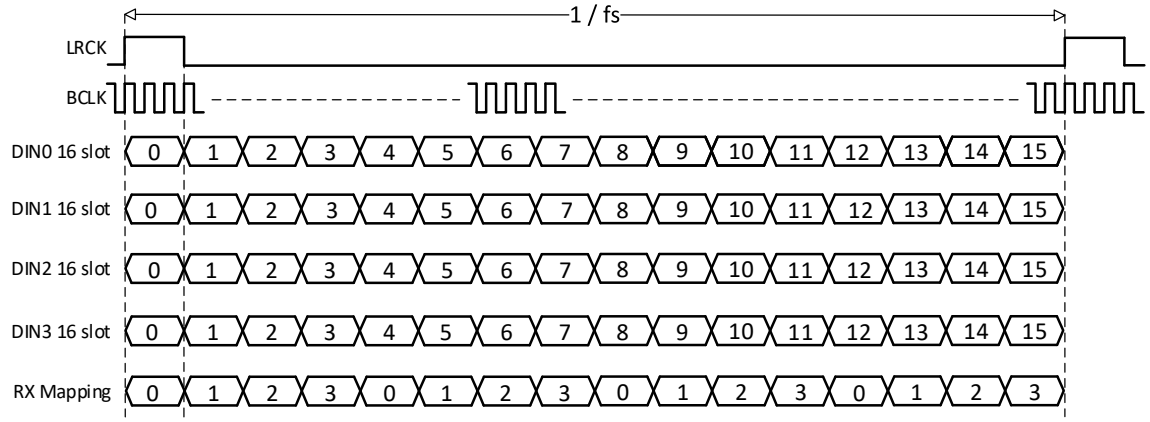
The following table shows the relationship between the slot id and encoder.

Table 8-3 DIN Slot ID and Encoder

DIN0 Slot ID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
DIN1 Slot ID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
DIN2 Slot ID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
DIN3 Slot ID	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

There are 16 channels mapping configuration, each wire selects four slots for RX. The following figure shows the 16-channel mapping configuration.

Figure 8-8 16-Channel Mapping Configuration



8.1.3.5 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.

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

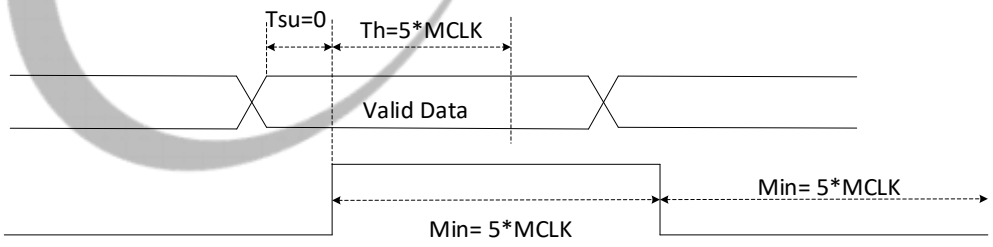
$$\text{Downsampling Latency} = \text{Phase Delay} + \text{FIFO Delay} = (32 * f_{sout}/f_{sin}) + 16 \text{ Input Sample Periods}$$

ASRC Timing

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

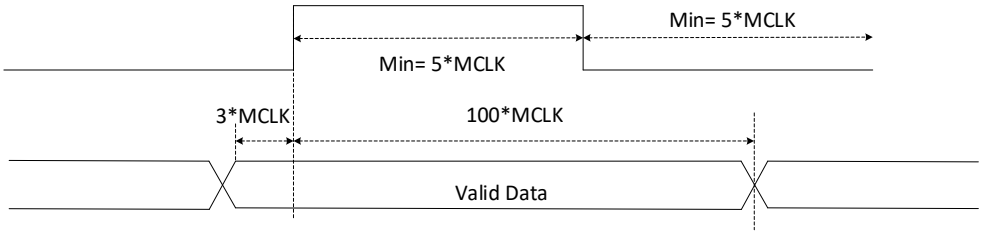
The following figure shows the timing requirements for the inputs.

Figure 8-9 Timing Requirements for Inputs



The following figure shows the timing requirements for the outputs.

Figure 8-10 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 8-4 Proper MCLK Values with Different Fsin and Fsout

Fsin \ Fsout	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

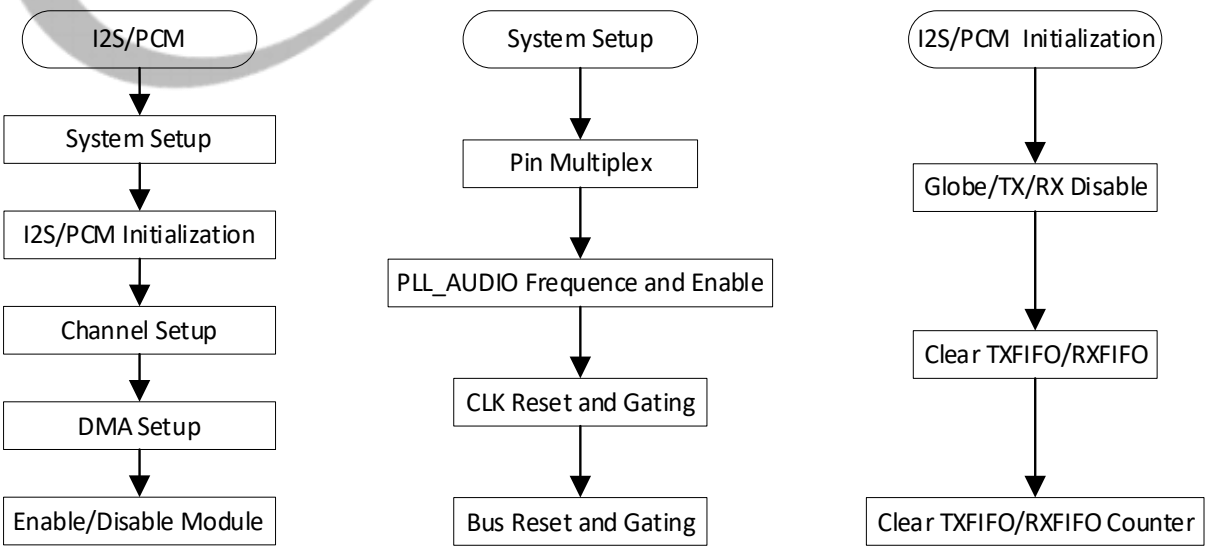
Note: The units for Fsin and Fsout are kHz and MCLK is MHz.

8.1.3.6 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 8-11 I2S/PCM Operation Flow



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. 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/PCM_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**.

2. **Channel 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 **"**DMA**"**. In this module, you just enable the DRQ.

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.

8.1.4 Programming Guidelines

8.1.4.1 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.

8.1.4.2 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.

1. Configure the PLL_AUDIO Register

a) Configure the **PLL_AUDIO0_CTRL_REG**[31:0] as 0x8814AB01. That is, PLL_AUDIO0 = 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_AUDIO0 should be 25*(171+1)/(1+1)/(1+0)/(1+20) = 102.381 MHz. In the IC test phase, configure the frequency of PLL_AUDIO0 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_AUDIO0_CTRL_REG** to get an equal-duty-cycle output clock of PLL_AUDIO0.

c) Configure bit[25:24] of **I2S/PCM2_ASRC_CLK_REG** as 0x00 to select the PLL_AUDIO0(4X).

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.

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 **F_{SOUT}CFG** as 1`h1 to open the clock gating.

d) Configure bit[19:16] (clock select) of **F_{SOUT}CFG** as 4`h0 to select I2S0_ASRC_CLK as the clock source.

e) Configure bit[7:4] (the first division factor) of **F_{SOUT}CFG** as 16`h13 to configure the first division factor as 128.

f) Configure bit[3:0] (the second division factor) of **F_{SOUT}CFG** 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)*2^{22}). 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.

8.1.5 Register List

Module Name	Base Address
I2S/PCM1	0x02033000
I2S/PCM2	0x02034000

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
I2S/PCM_ISTA	0x000C	I2S/PCM Interrupt Status Register
I2S/PCM_RXFIFO	0x0010	I2S/PCM RXFIFO Register
I2S/PCM_FCTL	0x0014	I2S/PCM FIFO Control Register
I2S/PCM_FSTA	0x0018	I2S/PCM FIFO Status Register
I2S/PCM_INT	0x001C	I2S/PCM DMA & Interrupt Control Register
I2S/PCM_TXFIFO	0x0020	I2S/PCM TXFIFO Register
I2S/PCM_CLKD	0x0024	I2S/PCM Clock Divide Register
I2S/PCM_TXCNT	0x0028	I2S/PCM TX Sample Counter Register
I2S/PCM_RXCNT	0x002C	I2S/PCM RX Sample Counter Register

Register Name	Offset	Description
I2S/PCM_CHCFG	0x0030	I2S/PCM Channel Configuration Register
I2S/PCM_TX0CHSEL	0x0034	I2S/PCM TX0 Channel Select Register
I2S/PCM_TX1CHSEL	0x0038	I2S/PCM TX1 Channel Select Register
I2S/PCM_TX2CHSEL	0x003C	I2S/PCM TX2 Channel Select Register
I2S/PCM_TX3CHSEL	0x0040	I2S/PCM TX3 Channel Select Register
I2S/PCM_TX0CHMAP0	0x0044	I2S/PCM TX0 Channel Mapping Register0
I2S/PCM_TX0CHMAP1	0x0048	I2S/PCM TX0 Channel Mapping Register1
I2S/PCM_TX1CHMAP0	0x004C	I2S/PCM TX1 Channel Mapping Register0
I2S/PCM_TX1CHMAP1	0x0050	I2S/PCM TX1 Channel Mapping Register1
I2S/PCM_TX2CHMAP0	0x0054	I2S/PCM TX2 Channel Mapping Register0
I2S/PCM_TX2CHMAP1	0x0058	I2S/PCM TX2 Channel Mapping Register1
I2S/PCM_TX3CHMAP0	0x005C	I2S/PCM TX3 Channel Mapping Register0
I2S/PCM_TX3CHMAP1	0x0060	I2S/PCM TX3 Channel Mapping Register1
I2S/PCM_RXCHSEL	0x0064	I2S/PCM RX Channel Select Register
I2S/PCM_RXCHMAP0	0x0068	I2S/PCM RX Channel Mapping Register0
I2S/PCM_RXCHMAP1	0x006C	I2S/PCM RX Channel Mapping Register1
I2S/PCM_RXCHMAP2	0x0070	I2S/PCM RX Channel Mapping Register2
I2S/PCM_RXCHMAP3	0x0074	I2S/PCM RX Channel Mapping Register3
MCLKCFG	0x0080	ASRC MCLK Configuration Register
FsoutCFG	0x0084	ASRC Out Sample Rate Configuration Register
FsinEXTCFG	0x0088	ASRC Input Sample Pulse Extend Configuration Register
ASRCCFG	0x008C	ASRC Enable Register
ASRCMANCFG	0x0090	ASRC Manual Ratio Configuration Register
ASRCRATIOSTAT	0x0094	ASRC Status Register
ASRCFIFOSTAT	0x0098	ASRC FIFO Level Status Register
ASRCMBISTCFG	0x009C	ASRC MBIST Test Configuration Register
ASRCMBISTSTAT	0x00A0	ASRC MBIST Test Status Register

8.1.6 Register Description

8.1.6.1 0x0000 I2S/PCM Control Register (Default Value: 0x0006_0000)

Offset: 0x0000			Register Name: I2S/PCM_CTL
Bit	Read/Write	Default/Hex	Description
31:22	/	/	/
21	R/W	0x0	RX_SYNC_EN_START The bit takes effect only when RX_SYNC_EN is set to 1. I2S1/I2S2/OWA RX Synchronize Enable Start. 0: Disabled 1: Enabled
20	R/W	0x0	RX_SYNC_EN I2S RX Synchronize Enable 0: Disabled 1: Enabled
19	/	/	/
18	R/W	0x1	BCLK_OUT Bit Clock Direction Select 0: Input 1: Output
17	R/W	0x1	LRCK_OUT LRCK Direction Select 0: Input 1: Output
16:12	/	/	/
11	R/W	0x0	DOUT3_EN Data3 Output Enable 0: Disabled, Hi-Z State 1: Enabled
10	R/W	0x0	DOUT2_EN Data2 Output Enable 0: Disabled, Hi-Z State 1: Enabled
9	R/W	0x0	DOUT1_EN Data1 Output Enable 0: Disabled, Hi-Z State 1: Enabled

Offset: 0x0000			Register Name: I2S/PCM_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
8	R/W	0x0	DOUT0_EN Data0 Output Enable 0: Disabled, Hi-Z State 1: Enabled
7	/	/	/
6	R/W	0x0	OUT_MUTE Data Output Mute Enable 0: Normal Transfer 1: Force DOUT to output 0
5:4	R/W	0x0	MODE_SEL Mode Selection 00: PCM Mode (offset 0: Long Frame; offset 1: Short Frame) 01: Left Mode (offset 0: LJ Mode; offset 1: I2S Mode) 10: Right-Justified Mode 11: Reserved
3	R/W	0x0	LOOP Loopback Test 0: Normal Mode 1: Loopback Test When set to '1', the bit indicates that the DOUT is connected to the DIN.
2	R/W	0x0	TXEN Transmitter Block Enable 0: Disabled 1: Enabled
1	R/W	0x0	RXEN Receiver Block Enable 0: Disabled 1: Enabled
0	R/W	0x0	GEN Globe Enable 0: Disabled 1: Enabled

8.1.6.2 0x0004 I2S/PCM Format Register 0 (Default Value: 0x0000_0033)

Offset: 0x0004			Register Name: I2S/PCM_FMT0
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30	R/W	0x0	LRCK_WIDTH LRCK Width (only applies to the PCM mode) 0: LRCK = 1 BCLK Width (Short Frame) 1: LRCK = 2 BCLK Width (Long Frame)
29:20	/	/	/
19	R/W	0x0	LRCK_POLARITY In I2S/Left-Justified/Right-Justified mode: 0: Left Channel when LRCK is low. 1: Left channel when LRCK is high. In PCM mode: 0: PCM LRCK asserted at the negative edge. 1: PCM LRCK asserted at the positive edge.
18	/	/	/
17:8	R/W	0x0	LRCK_PERIOD It is used to program the number of BCLKs per channel of the sample frame. This value is interpreted as follows. PCM mode: Number of BCLKs within (Left + Right) channel width. I2S/Left-Justified/Right-Justified mode: Number of BCLKs within each channel width (Left or Right). For example: N = 7: 8 BCLKs width … N = 1023: 1024 BCLKs width
7	R/W	0x0	BCLK_POLARITY 0: Normal mode, DOUT drives data at negative edge 1: Invert mode, DOUT drives data at positive edge

Offset: 0x0004			Register Name: I2S/PCM_FMT0
Bit	**Read/Write**	**Default/Hex**	**Description**
6:4	R/W	0x3	SR Sample Resolution 000: Reserved 001: 8-bit 010: 12-bit 011: 16-bit 100: 20-bit 101: 24-bit 110: 28-bit 111: 32-bit
3	R/W	0x0	EDGE_TRANSFER Edge Transfer 0: DOUT drives data and DIN sample data at the different BCLK edge 1: DOUT drives data and DIN sample data at the same BCLK edge BCLK_POLARITY = 0, EDGE_TRANSFER = 0, DIN sample data at positive edge; BCLK_POLARITY = 0, EDGE_TRANSFER = 1, DIN sample data at negative edge; BCLK_POLARITY = 1, EDGE_TRANSFER = 0, DIN sample data at negative edge; BCLK_POLARITY = 1, EDGE_TRANSFER = 1, DIN sample data at positive edge.
2:0	R/W	0x3	SW Slot Width Select 000: Reserved 001: 8-bit 010: 12-bit 011: 16-bit 100: 20-bit 101: 24-bit 110: 28-bit 111: 32-bit

8.1.6.3 0x0008 I2S/PCM Format Register 1 (Default Value: 0x0000_0030)

Offset: 0x0008			Register Name: I2S/PCM_FMT1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	RX MLS MSB/LSB First Select 0: MSB First 1: LSB First
6	R/W	0x0	TX MLS MSB/LSB First Select 0: MSB First 1: LSB First
5:4	R/W	0x3	SEXT Sign Extend in Slot [Sample Resolution < Slot Width] 00: Zeros or audio gain padding at LSB position 01: Sign extension at MSB position 10: Reserved 11: Transfer 0 after each sample in each Slot
3:2	R/W	0x0	RX_PDM PCM Data Mode 00: Linear PCM 01: Reserved 10: 8-bit u-law 11: 8-bit A-law
1:0	R/W	0x0	TX_PDM PCM Data Mode 00: Linear PCM 01: Reserved 10: 8-bit u-law 11: 8-bit A-law

8.1.6.4 0x000C I2S/PCM Interrupt Status Register (Default Value: 0x0000_0010)

Offset: 0x000C			Register Name: I2S/PCM_ISTA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:7	/	/	/

Offset: 0x000C			Register Name: I2S/PCM_ISTA
Bit	**Read/Write**	**Default/Hex**	**Description**
6	R/W1C	0x0	TXU_INT TXFIFO Underrun Pending Interrupt 0: No pending interrupt 1: TXFIFO underrun pending interrupt Write '1' to clear this interrupt.
5	R/W1C	0x0	TXO_INT TXFIFO Overrun Pending Interrupt 0: No pending interrupt 1: TXFIFO overrun pending interrupt Write '1' to clear this interrupt.
4	R	0x1	TXE_INT TXFIFO Empty Pending Interrupt 0: No pending IRQ 1: TXFIFO empty pending interrupt when data in TXFIFO are less than TX trigger level
3	/	/	/
2	R/W1C	0x0	RXU_INT RXFIFO Underrun Pending Interrupt 0: No pending interrupt 1: RXFIFO underrun pending interrupt Write '1' to clear this interrupt.
1	R/W1C	0x0	RXO_INT RXFIFO Overrun Pending Interrupt 0: No pending IRQ 1: RXFIFO overrun pending IRQ Write '1' to clear this interrupt.
0	R/W	0x0	RXA_INT RXFIFO Data Available Pending Interrupt 0: No pending IRQ 1: Data available pending IRQ when data in RXFIFO are more than RX trigger level

8.1.6.5 0x0010 I2S/PCM RXFIFO Register (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: I2S/PCM_RXFIFO
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	RX_DATA RX Sample The host can get one sample by reading this register. The left channel sample data is first and then the right channel sample.

8.1.6.6 0x0014 I2S/PCM FIFO Control Register (Default Value: 0x0004_00F0)

Offset: 0x0014			Register Name: I2S/PCM_FCTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	HUB_EN Audio Hub Enable The bit takes effect only when TXEN is set to 1. I2S1/I2S2/OWA TXFIFO Hub Enable. 0: Disabled 1: Enabled
30:26	/	/	/
25	R/WAC	0x0	FTX Write '1' to flush TXFIFO, self clear to '0'.
24	R/WAC	0x0	FRX Write '1' to flush RXFIFO, self clear to '0'.
23:19	/	/	/
18:12	R/W	0x40	TXTL TXFIFO Empty Trigger Level Interrupt and DMA request trigger level for TXFIFO normal condition. Trigger Level = TXTL
11:10	/	/	/
9:4	R/W	0xF	RXTL RXFIFO Trigger Level Interrupt and DMA request trigger level for RXFIFO normal condition. Trigger Level = RXTL + 1
3	/	/	/

Offset: 0x0014			Register Name: I2S/PCM_FCTL
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R/W	0x0	TXIM TXFIFO Input Mode (Mode 0, 1) 0: Valid data at the MSB of TXFIFO register 1: Valid data at the LSB of TXFIFO register Example for 20-bit transmitted audio sample: Mode 0: TXFIFO[31:0] = {APB_WDATA[31:12], 12'h0} Mode 1: TXFIFO[31:0] = {APB_WDATA[19:0], 12'h0}
1:0	R/W	0x0	RXOM RXFIFO Output Mode (Mode 0, 1, 2, 3) 00: Expanding '0' at LSB of RXFIFO register 01: Expanding received sample sign bit at MSB of RXFIFO register 10: Truncating received samples at high half-word of RXFIFO register and low half-word of RXFIFO register is filled by '0' 11: Truncating received samples at low half-word of RXFIFO register and high half-word of RXFIFO register is expanded by its sign bit Example for 20-bit received audio sample: Mode 0: APB_RDATA[31:0] = {RXFIFO[31:12], 12'h0} Mode 1: APB_RDATA[31:0] = {12{RXFIFO[31]}, RXFIFO[31:12]} Mode 2: APB_RDATA [31:0] = {RXFIFO[31:16], 16'h0} Mode 3: APB_RDATA[31:0] = {16{RXFIFO[31], RXFIFO[31:16]}

8.1.6.7 0x0018 I2S/PCM FIFO Status Register (Default Value: 0x1080_0080)

Offset: 0x0018			Register Name: I2S/PCM_FSTA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	/	/	/
28	R	0x1	TXE TXFIFO Empty 0: No room for new sample in TXFIFO 1: More than one room for new sample in TXFIFO (>= 1 Word)
27:24	/	/	/
23:16	R	0x80	TXE_CNT TXFIFO Empty Space Word Counter

Offset: 0x0018			Register Name: I2S/PCM_FSTA
Bit	**Read/Write**	**Default/Hex**	**Description**
15:9	/	/	/
8	R	0x0	RXA RXFIFO Available 0: No available data in RXFIFO 1: More than one sample in RXFIFO (>= 1 Word)
7	R	0x1	PLACE_HOLDER NO Meaning.
6:0	R	0x0	RXA_CNT RXFIFO available sample word counter

8.1.6.8 0x001C I2S/PCM DMA & Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: I2S/PCM_INT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	TX_DRQ TXFIFO Empty DRQ Enable 0: Disabled 1: Enabled
6	R/W	0x0	TXUI_EN TXFIFO Underrun Interrupt Enable 0: Disabled 1: Enabled
5	R/W	0x0	TXOI_EN TXFIFO Overrun Interrupt Enable 0: Disabled 1: Enabled When set to '1', an interrupt happens when writing new audio data if TXFIFO is full.
4	R/W	0x0	TXEI_EN TXFIFO Empty Interrupt Enable 0: Disabled 1: Enabled

Offset: 0x001C			Register Name: I2S/PCM_INT
Bit	**Read/Write**	**Default/Hex**	**Description**
3	R/W	0x0	RX_DRQ RXFIFO Data Available DRQ Enable 0: Disabled 1: Enabled When set to '1', RXFIFO DMA request line is asserted if data is available in RXFIFO.
2	R/W	0x0	RXUI_EN RXFIFO Underrun Interrupt Enable 0: Disabled 1: Enabled
1	R/W	0x0	RXOI_EN RXFIFO Overrun Interrupt Enable 0: Disabled 1: Enabled
0	R/W	0x0	RXAI_EN RXFIFO Data Available Interrupt Enable 0: Disabled 1: Enabled

8.1.6.9 0x0020 I2S/PCM TXFIFO Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: I2S/PCM_TXFIFO
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	W	0x0	TX_DATA TX Sample Transmitting left, right channel sample data should be written to this register one by one. The left channel sample data is first and then the right channel sample.

8.1.6.10 0x0024 I2S/PCM Clock Divide Register (Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: I2S/PCM_CLKD
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/

Offset: 0x0024			Register Name: I2S/PCM_CLKD
Bit	**Read/Write**	**Default/Hex**	**Description**
8	R/W	0x0	MCLKO_EN 0: Disable MCLK Output 1: Enable MCLK Output **Note: Whether in slave or master mode, when this bit is set to '1', MCLK should be output.**
7:4	R/W	0x0	BCLKDIV BCLK Divide Ratio from PLL_AUDIO 0000: Reserved 0001: Divide by 1 0010: Divide by 2 0011: Divide by 4 0100: Divide by 6 0101: Divide by 8 0110: Divide by 12 0111: Divide by 16 1000: Divide by 24 1001: Divide by 32 1010: Divide by 48 1011: Divide by 64 1100: Divide by 96 1101: Divide by 128 1110: Divide by 176 1111: Divide by 192

Offset: 0x0024			Register Name: I2S/PCM_CLKD
Bit	**Read/Write**	**Default/Hex**	**Description**
3:0	R/W	0x0	MCLKDIV MCLK Divide Ratio from PLL_AUDIO 0000: Reserved 0001: Divide by 1 0010: Divide by 2 0011: Divide by 4 0100: Divide by 6 0101: Divide by 8 0110: Divide by 12 0111: Divide by 16 1000: Divide by 24 1001: Divide by 32 1010: Divide by 48 1011: Divide by 64 1100: Divide by 96 1101: Divide by 128 1110: Divide by 176 1111: Divide by 192

8.1.6.11 0x0028 I2S/PCM TX Sample Counter Register (Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: I2S/PCM_TXCNT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	TX_CNT TX Sample Counter The audio sample number of sending into TXFIFO. When one sample is put into TXFIFO by DMA or by host IO, the TX sample counter register increases by one. The TX sample counter register can be set to any initial value at any time. After being updated by the initial value, the counter register should count on the base of this initial value.

8.1.6.12 0x002C I2S/PCM RX Sample Counter Register (Default Value: 0x0000_0000)

Offset: 0x002C			Register Name: I2S/PCM_RXCNT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	RX_CNT RX Sample Counter The audio sample number of writing into RXFIFO. When one sample is written by Digital Audio Engine, the RX sample counter register increases by one. The RX sample counter register can be set to any initial value at any time. After being updated by the initial value, the counter register should count on the base of this initial value.

8.1.6.13 0x0030 I2S/PCM Channel Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: I2S/PCM_CHCFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	/	/	/
9	R/W	0x0	TX_SLOT_HIZ 0: Normal mode for the last half-cycle of BCLK in the slot 1: Turn to Hi-Z state for the last half-cycle of BCLK in the slot
8	R/W	0x0	TX_STATE 0: Transfer level 0 in non-transferring slot 1: Turn to Hi-Z State (TDM) in non-transferring slot
7:4	R/W	0x0	RX_SLOT_NUM RX Channel/Slot number between CPU/DMA and RXFIFO 0000: 1 channel or slot ... 0111: 8 channels or slots 1000: 9 channels or slots ... 1111: 16 channels or slots

Offset: 0x0030			Register Name: I2S/PCM_CHCFG
Bit	**Read/Write**	**Default/Hex**	**Description**
3:0	R/W	0x0	TX_SLOT_NUM TX Channel/Slot number between CPU/DMA and TXFIFO 0000: 1 channel or slot ... 0111: 8 channels or slots 1000: 9 channels or slots ... 1111:16 channels or slots

8.1.6.14 0x0034 I2S/PCM TX0 Channel Select Register (Default Value: 0x0000_0000)

Offset: 0x0034			Register Name: I2S/PCM_TX0CHSEL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:22	/	/	/
21:20	R/W	0x0	TX0_OFFSET TX0 Offset Tune (TX0 Data offset to LRCK) 0: No offset n: Data is offset by n BCLKs to LRCK
19:16	R/W	0x0	TX0_CHSEL TX0 Channel (Slot) Number Select for Each Output 0000: 1 channel or slot ... 0111: 8 channels or slots 1000: 9 channels or slots ... 1111: 16 channels or slots
15:0	R/W	0x0	TX0_CHEN TX0 Channel (Slot) Enable The bit[15:0] refer to Slot [15:0]. When one or more slots are disabled, the affected slots are set to the disable state. 0: Disabled 1: Enabled

8.1.6.15 0x0038 I2S/PCM TX1 Channel Select Register (Default Value: 0x0000_0000)

Offset: 0x0038			Register Name: I2S/PCM_TX1CHSEL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:22	/	/	/
21:20	R/W	0x0	TX1_OFFSET TX1 Offset Tune (TX1 Data offset to LRCK) 0: No offset n: Data is offset by n BCLKs to LRCK
19:16	R/W	0x0	TX1_CHSEL TX1 Channel (Slot) Number Select for Each Output 0000: 1 channel or slot ... 0111: 8 channels or slots 1000: 9 channels or slots ... 1111: 16 channels or slots
15:0	R/W	0x0	TX1_CHEN TX1 Channel (Slot) Enable The bit[15:0] refers to Slot [15:0]. When one or more slots are disabled, the affected slots are set to the disable state. 0: Disabled 1: Enabled

8.1.6.16 0x003C I2S/PCM TX2 Channel Select Register (Default Value: 0x0000_0000)

Offset: 0x003C			Register Name: I2S/PCM_TX2CHSEL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:22	/	/	/
21:20	R/W	0x0	TX2_OFFSET TX2 Offset Tune (TX2 Data offset to LRCK) 0: No offset n: Data is offset by n BCLKs to LRCK

Offset: 0x003C			Register Name: I2S/PCM_TX2CHSEL
Bit	**Read/Write**	**Default/Hex**	**Description**
19:16	R/W	0x0	TX2_CHSEL TX2 Channel (Slot) Number Select for Each Output 0000: 1 channel or slot ... 0111: 8 channels or slots 1000: 9 channels or slots ... 1111: 16 channels or slots
15:0	R/W	0x0	TX2_CHEN TX2 Channel (Slot) Enable The bit[15:0] refers to Slot [15:0]. When one or more slots are disabled, the affected slots are set to the disable state. 0: Disabled 1: Enabled

8.1.6.17 0x0040 I2S/PCM TX3 Channel Select Register (Default Value: 0x0000_0000)

Offset: 0x0040			Register Name: I2S/PCM_TX3CHSEL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:22	/	/	/
21:20	R/W	0x0	TX3_OFFSET TX3 Offset Tune (TX3 Data offset to LRCK) 0: No offset n: Data is offset by n BCLKs to LRCK
19:16	R/W	0x0	TX3_CHSEL TX3 Channel (Slot) Number Select for Each Output 0000: 1 channel or slot ... 0111: 8 channels or slots 1000: 9 channels or slots ... 1111: 16 channels or slots

Offset: 0x0040			Register Name: I2S/PCM_TX3CHSEL
Bit	**Read/Write**	**Default/Hex**	**Description**
15:0	R/W	0x0	TX3_CHEN TX3 Channel (Slot) Enable The bit[15:0] refers to Slot [15:0]. When one or more slots are disabled, the affected slots are set to the disable state. 0: Disabled 1: Enabled

8.1.6.18 0x0044 I2S/PCM TX0 Channel Mapping0 Register 0 (Default Value: 0x0000_0000)

Offset: 0x0044			Register Name: I2S/PCM_TX0CHMAP0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	TX0_CH15_MAP TX0 Channel 15 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
27:24	R/W	0x0	TX0_CH14_MAP TX0 Channel 14 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
23:20	R/W	0x0	TX0_CH13_MAP TX0 Channel 13 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample

Offset: 0x0044			Register Name: I2S/PCM_TX0CHMAP0
Bit	**Read/Write**	**Default/Hex**	**Description**
19:16	R/W	0x0	TX0_CH12_MAP TX0 Channel 12 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
15:12	R/W	0x0	TX0_CH11_MAP TX0 Channel 11 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
11:8	R/W	0x0	TX0_CH10_MAP TX0 Channel 10 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
7:4	R/W	0x0	TX0_CH9_MAP TX0 Channel 9 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

Offset: 0x0044			Register Name: I2S/PCM_TX0CHMAP0
Bit	**Read/Write**	**Default/Hex**	**Description**
3:0	R/W	0x0	TX0_CH8_MAP TX0 Channel 8 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

8.1.6.19 0x0048 I2S/PCM TX0 Channel Mapping1 Register 1 (Default Value: 0x0000_0000)

Offset: 0x0048			Register Name: I2S/PCM_TX0CHMAP1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	TX0_CH7_MAP TX0 Channel 7 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
27:24	R/W	0x0	TX0_CH6_MAP TX0 Channel 6 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

Offset: 0x0048			Register Name: I2S/PCM_TX0CHMAP1
Bit	**Read/Write**	**Default/Hex**	**Description**
23:20	R/W	0x0	TX0_CH5_MAP TX0 Channel 5 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
19:16	R/W	0x0	TX0_CH4_MAP TX0 Channel 4 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
15:12	R/W	0x0	TX0_CH3_MAP TX0 Channel 3 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
11:8	R/W	0x0	TX0_CH2_MAP TX0 Channel 2 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample

Offset: 0x0048			Register Name: I2S/PCM_TX0CHMAP1
Bit	**Read/Write**	**Default/Hex**	**Description**
7:4	R/W	0x0	TX0_CH1_MAP TX0 Channel 1 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
3:0	R/W	0x0	TX0_CH0_MAP TX0 Channel 0 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

8.1.6.20 0x004C I2S/PCM TX1 Channel Mapping0 Register 0 (Default Value: 0x0000_0000)

Offset: 0x004C			Register Name: I2S/PCM_TX1CHMAP0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	TX1_CH15_MAP TX1 Channel 15 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

Offset: 0x004C			Register Name: I2S/PCM_TX1CHMAP0
Bit	**Read/Write**	**Default/Hex**	**Description**
27:24	R/W	0x0	TX1_CH14_MAP TX1 Channel 14 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
23:20	R/W	0x0	TX1_CH13_MAP TX1 Channel 13 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
19:16	R/W	0x0	TX1_CH12_MAP TX1 Channel 12 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
15:12	R/W	0x0	TX1_CH11_MAP TX1 Channel 11 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

Offset: 0x004C			Register Name: I2S/PCM_TX1CHMAP0
Bit	**Read/Write**	**Default/Hex**	**Description**
11:8	R/W	0x0	TX1_CH10_MAP TX1 Channel 10 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
7:4	R/W	0x0	TX1_CH9_MAP TX1 Channel 9 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
3:0	R/W	0x0	TX1_CH8_MAP TX1 Channel 8 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

8.1.6.21 0x0050 I2S/PCM TX1 Channel Mapping1 Register 1 (Default Value: 0x0000_0000)

Offset: 0x0050			Register Name: I2S/PCM_TX1CHMAP1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	TX1_CH7_MAP TX1 Channel 7 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
27:24	R/W	0x0	TX1_CH6_MAP TX1 Channel 6 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
23:20	R/W	0x0	TX1_CH5_MAP TX1 Channel 5 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
19:16	R/W	0x0	TX1_CH4_MAP TX1 Channel 4 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample

Offset: 0x0050			Register Name: I2S/PCM_TX1CHMAP1
Bit	**Read/Write**	**Default/Hex**	**Description**
15:12	R/W	0x0	TX1_CH3_MAP TX1 Channel 3 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
11:8	R/W	0x0	TX1_CH2_MAP TX1 Channel 2 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
7:4	R/W	0x0	TX1_CH1_MAP TX1 Channel 1 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
3:0	R/W	0x0	TX1_CH0_MAP TX1 Channel 0 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample

8.1.6.22 0x0054 I2S/PCM TX2 Channel Mapping0 Register 0 (Default Value: 0x0000_0000)

Offset: 0x0054			Register Name: I2S/PCM_TX2CHMAP0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	TX2_CH15_MAP TX2 Channel 15 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
27:24	R/W	0x0	TX2_CH14_MAP TX2 Channel 14 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
23:20	R/W	0x0	TX2_CH13_MAP TX2 Channel 13 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
19:16	R/W	0x0	TX2_CH12_MAP TX2 Channel 12 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample

Offset: 0x0054			Register Name: I2S/PCM_TX2CHMAP0
Bit	**Read/Write**	**Default/Hex**	**Description**
15:12	R/W	0x0	TX2_CH11_MAP TX2 Channel 11 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
11:8	R/W	0x0	TX2_CH10_MAP TX2 Channel 10 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
7:4	R/W	0x0	TX2_CH9_MAP TX2 Channel 9 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
3:0	R/W	0x0	TX2_CH8_MAP TX2 Channel 8 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample

8.1.6.23 0x0058 I2S/PCM TX2 Channel Mapping1 Register 1 (Default Value: 0x0000_0000)

Offset: 0x0058			Register Name: I2S/PCM_TX2CHMAP1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	TX2_CH7_MAP TX2 Channel 7 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
27:24	R/W	0x0	TX2_CH6_MAP TX2 Channel 6 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
23:20	R/W	0x0	TX2_CH5_MAP TX2 Channel 5 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
19:16	R/W	0x0	TX2_CH4_MAP TX2 Channel 4 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

Offset: 0x0058			Register Name: I2S/PCM_TX2CHMAP1
Bit	**Read/Write**	**Default/Hex**	**Description**
15:12	R/W	0x0	TX2_CH3_MAP TX2 Channel 3 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
11:8	R/W	0x0	TX2_CH2_MAP TX2 Channel 2 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
7:4	R/W	0x0	TX2_CH1_MAP TX2 Channel 1 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
3:0	R/W	0x0	TX2_CH0_MAP TX2 Channel 0 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

8.1.6.24 0x005C I2S/PCM TX3 Channel Mapping0 Register 0 (Default Value: 0x0000_0000)

Offset: 0x005C			Register Name: I2S/PCM_TX3CHMAP0
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x0	TX3_CH15_MAP TX3 Channel 15 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
27:24	R/W	0x0	TX3_CH14_MAP TX3 Channel 14 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
23:20	R/W	0x0	TX3_CH13_MAP TX3 Channel 13 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
19:16	R/W	0x0	TX3_CH12_MAP TX3 Channel 12 Mapping 0000: The first sample ... 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

Offset: 0x005C			Register Name: I2S/PCM_TX3CHMAP0
Bit	**Read/Write**	**Default/Hex**	**Description**
15:12	R/W	0x0	TX3_CH11_MAP TX3 Channel 11 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
11:8	R/W	0x0	TX3_CH10_MAP TX3 Channel 10 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
7:4	R/W	0x0	TX3_CH9_MAP TX3 Channel 9 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
3:0	R/W	0x0	TX3_CH8_MAP TX3 Channel 8 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample

8.1.6.25 0x0060 I2S/PCM TX3 Channel Mapping1 Register 1 (Default Value: 0x0000_0000)

Offset: 0x0060			Register Name: I2S/PCM_TX3CHMAP1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:28	R/W	0x0	TX3_CH7_MAP TX3 Channel 7 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
27:24	R/W	0x0	TX3_CH6_MAP TX3 Channel 6 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
23:20	R/W	0x0	TX3_CH5_MAP TX3 Channel 5 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
19:16	R/W	0x0	TX3_CH4_MAP TX3 Channel 4 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample

Offset: 0x0060			Register Name: I2S/PCM_TX3CHMAP1
Bit	**Read/Write**	**Default/Hex**	**Description**
15:12	R/W	0x0	TX3_CH3_MAP TX3 Channel 3 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
11:8	R/W	0x0	TX3_CH2_MAP TX3 Channel 2 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
7:4	R/W	0x0	TX3_CH1_MAP TX3 Channel 1 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
3:0	R/W	0x0	TX3_CH0_MAP TX3 Channel 0 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample

8.1.6.26 0x0064 I2S/PCM RX Channel Select Register (Default Value: 0x0000_0000)

Offset: 0x0064			Register Name: I2S/PCM_RXCHSEL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:22	/	/	/
21:20	R/W	0x0	RX_OFFSET RX Offset Tune (RX Data offset to LRCK) 0: No offset n: Data is offset by n BCLKs to LRCK
19:16	R/W	0x0	RX_CHSEL RX Channel (Slot) Number Select for Input 0000: 1 channel or slot ... 0111: 8 channels or slots 1000: 9 channels or slots ... 1111: 16 channels or slots
15:0	/	/	/

8.1.6.27 0x0068 I2S/PCM RX Channel Mapping Register0 (Default Value: 0x0000_0000)

Offset: 0x0068			Register Name: I2S/PCM_RXCHMAP0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29:28	R/W	0x0	RX_CH15_SELECT RX Channel 15 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3

Offset: 0x0068			Register Name: I2S/PCM_RXCHMAP0
Bit	**Read/Write**	**Default/Hex**	**Description**
27:24	R/W	0x0	RX_CH15_MAP RX Channel 15 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
23:22	/	/	/
21:20	R/W	0x0	RX_CH14_SELECT RX Channel 14 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
19:16	R/W	0x0	RX_CH14_MAP RX Channel 14 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
15:14	/	/	/
13:12	R/W	0x0	RX_CH13_SELECT RX Channel 13 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3

Offset: 0x0068			Register Name: I2S/PCM_RXCHMAP0
Bit	**Read/Write**	**Default/Hex**	**Description**
11:8	R/W	0x0	RX_CH13_MAP RX Channel 13 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample
7:6	/	/	/
5:4	R/W	0x0	RX_CH12_SELECT RX Channel 12 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
3:0	R/W	0x0	RX_CH12_MAP RX Channel 12 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample … 1111: The sixteenth sample

8.1.6.28 0x006C I2S/PCM RX Channel Mapping Register1 (Default Value: 0x0000_0000)

Offset: 0x006C			Register Name: I2S/PCM_RXCHMAP1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29:28	R/W	0x0	RX_CH11_SELECT RX Channel 11 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3

Offset: 0x006C			Register Name: I2S/PCM_RXCHMAP1
Bit	**Read/Write**	**Default/Hex**	**Description**
27:24	R/W	0x0	RX_CH11_MAP RX Channel 11 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
23:22	/	/	/
21:20	R/W	0x0	RX_CH10_SELECT RX Channel 10 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
19:16	R/W	0x0	RX_CH10_MAP RX Channel 10 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
15:14	/	/	/
13:12	R/W	0x0	RX_CH9_SELECT RX Channel 9 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3

Offset: 0x006C			Register Name: I2S/PCM_RXCHMAP1
Bit	**Read/Write**	**Default/Hex**	**Description**
11:8	R/W	0x0	RX_CH9_MAP RX Channel 9 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
7:6	/	/	/
5:4	R/W	0x0	RX_CH8_SELECT RX Channel 8 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
3:0	R/W	0x0	RX_CH8_MAP RX Channel 8 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

8.1.6.29 0x0070 I2S/PCM RX Channel Mapping Register2 (Default Value: 0x0000_0000)

Offset: 0x0070			Register Name: I2S/PCM_RXCHMAP2
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29:28	R/W	0x0	RX_CH7_SELECT RX Channel 7 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3

Offset: 0x0070			Register Name: I2S/PCM_RXCHMAP2
Bit	**Read/Write**	**Default/Hex**	**Description**
27:24	R/W	0x0	RX_CH7_MAP RX Channel 7 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
23:22	/	/	/
21:20	R/W	0x0	RX_CH6_SELECT RX Channel 6 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
19:16	R/W	0x0	RX_CH6_MAP RX Channel 6 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
15:14	/	/	/
13:12	R/W	0x0	RX_CH5_SELECT RX Channel 5 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3

Offset: 0x0070			Register Name: I2S/PCM_RXCHMAP2
Bit	**Read/Write**	**Default/Hex**	**Description**
11: 8	R/W	0x0	RX_CH5_MAP RX Channel 5 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
7:6	/	/	/
5:4	R/W	0x0	RX_CH4_SELECT RX Channel 4 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
3:0	R/W	0x0	RX_CH4_MAP RX Channel 4 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

8.1.6.30 0x0074 I2S/PCM RX Channel Mapping Register3 (Default Value: 0x0000_0000)

Offset: 0x0074			Register Name: I2S/PCM_RXCHMAP3
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29:28	R/W	0x0	RX_CH3_SELECT RX Channel 3 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3

Offset: 0x0074			Register Name: I2S/PCM_RXCHMAP3
Bit	**Read/Write**	**Default/Hex**	**Description**
27:24	R/W	0x0	RX_CH3_MAP RX Channel 3 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
23:22	/	/	/
21:20	R/W	0x0	RX_CH2_SELECT RX Channel 2 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
19:16	R/W	0x0	RX_CH2_MAP RX Channel 2 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
15:14	/	/	/
13:12	R/W	0x0	RX_CH1_SELECT RX Channel 1 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3

Offset: 0x0074			Register Name: I2S/PCM_RXCHMAP3
Bit	**Read/Write**	**Default/Hex**	**Description**
11:8	R/W	0x0	RX_CH1_MAP RX Channel 1 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample
7:6	/	/	/
5:4	R/W	0x0	RX_CH0_SELECT RX Channel 0 Select 00: SDI0 01: SDI1 10: SDI2 11: SDI3
3:0	R/W	0x0	RX_CH0_MAP RX Channel 0 Mapping 0000: The first sample … 0111: The eighth sample 1000: The ninth sample ... 1111: The sixteenth sample

8.1.6.31 0x0080 I2S/PCM ASRC MCLK Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0080			Register Name: MCLKCFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	R	0x0	/
16	R/W	0x0	ASRC_MCLK_GATE ASRC Clock Gate Enable Control 0: Gated 1: Not gated
15:4	/	/	/

Offset: 0x0080			Register Name: MCLKCFG
Bit	**Read/Write**	**Default/Hex**	**Description**
3:0	R/W	0x0	ASRC_MCLK_FREQ_DIV_COE Frequency Division Coefficient 4'd0 = Res (no output), 4'd1 = 1x, 4'd2 = 1/2x, 4'd3 = 1/4x, 4'd4 = 1/6x, 4'd5 = 1/8x, 4'd6 = 1/12x, 4'd7 = 1/16x, 4'd8 = 1/24x, 4'd9 = 1/32x, 4'd10 = 1/48, 4'd11 = 1/64x, 4'd12 = 1/96x, 4'd13 = 1/128x, 4'd14 = 1/176x, 4'd15 = 1/192x, others = Res

8.1.6.32 0x0084 I2S/PCM ASRC OUT Sample Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0084			Register Name: F_{SOUT}CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:21	R	0x0	/
20	R/W	0x0	FSOUT_GATE fsout Clock Gate Enable Control 0: Gated 1: Not gated

Offset: 0x0084			Register Name: F_{SOUT}CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
19:16	R/W	0x0	FSOUT_CLK_SRC_SEL fsout Clock Source Select 00: I2S0_ASRC_CLK 01: ACLK 10: ACLKM 11: BCLK Others: Reserved
15:8	/	/	/
7:4	R/W	0x0	FSOUT_CLK_FREQ_DIV_COE1 fsout Frequency Division Coefficient 1 The First Division Factor It has two levels of frequency division, the first level is bit[7:4], the second level is bit[3:0], and the frequency division factors are multiplied by the two division factors, the division relationship of the two divisions are the same. 4'd0 = Res (no output), 4'd1 = 1x, 4'd2 = 1/2x, 4'd3 = 1/4x, 4'd4 = 1/6x, 4'd5 = 1/8x, 4'd6 = 1/12x, 4'd7 = 1/16x, 4'd8 = 1/24x, 4'd9 = 1/32x, 4'd10 = 1/48, 4'd11 = 1/64x, 4'd12 = 1/96x, 4'd13 = 1/128x, 4'd14 = 1/176x, 4'd15 = 1/192x

Offset: 0x0084			Register Name: F_{SOUT}CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
3:0	R/W	0x0	FSOUT_CLK_FREQ_DIV_COE2 fsout Frequency Division Coefficient 2 The Second Division Factor 4'd0 = Res (no output), 4'd1 = 1x, 4'd2 = 1/2x, 4'd3 = 1/4x, 4'd4 = 1/6x, 4'd5 = 1/8x, 4'd6 = 1/12x, 4'd7 = 1/16x, 4'd8 = 1/24x, 4'd9 = 1/32x, 4'd10 = 1/48, 4'd11 = 1/64x, 4'd12 = 1/96x, 4'd13 = 1/128x, 4'd14 = 1/176x, 4'd15 = 1/192x

8.1.6.33 0x0088 I2S/PCM IN Sample Pulse Extend Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0088			Register Name: FsinEXTCFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	R	0x0	/
16	R/W	0x0	Extend Enable 0: Disabled 1: Enabled Enable the bit when using ASRC.
15:0	R/W	0x0	The Cycle Number of Pulse Extend The cycle is BCLK clock and is 1 at least.

8.1.6.34 0x008C I2S/PCM ASRC Enable Register (Default Value: 0x0000_0000)

Offset: 0x008C			Register Name: ASRCEN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	R	0x0	/
0	R/W	0x0	ASRC Function Enable 0: Disabled 1: Enabled

8.1.6.35 0x0090 I2S/PCM ASRC Manual Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0090			Register Name: ASRCMANCFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	ASRC_RATIO_MANUAL_EN Manual Configuration of ASRC Ratio Enable 0: Disabled 1: Enabled
30:26	R	0x0	/
25:0	R/W	0x0	ASRC_RATIO_VALUE_MANUAL_CFG ASRC Ratio Value Manual Configure The ratio value is an unsigned 26-bit number and uses 4.22 data format, which means there are 4 bits to the left of the decimal point and 22 bits to the right of the decimal point.

8.1.6.36 0x0094 I2S/PCM ASRC Ratio State Register (Default Value: 0x0040_0000)

Offset: 0x0094			Register Name: ASRCRATIOSTAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	R	0x0	/
29	R	0x0	ASRC_BUF_OVERFLOW_STA ASRC Receive Data Buffer Overflow State It can control the mute with lock. 0: No overflow 1: Overflow

Offset: 0x0094			Register Name: ASRCRATIOSTAT
Bit	**Read/Write**	**Default/Hex**	**Description**
28	R	0x0	ADAPT_COMPUT_LOCK Adaptive Ratio Computational Lock 0: Unlocked 1: Locked
27:26	R	0x0	/
25:0	R	0x400000	ADAPT_COMPUT_VALUE Adaptive Ratio Computational Value

8.1.6.37 0x0098 I2S/PCM ASRC FIFO State Register (Default Value: 0x0000_0000)

Offset: 0x0098			Register Name: ASRCFIFOSTAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8:0	R	0x0	ASRC_RX_FIFO_FULL_LEVAL ASRC RXFIFO Full Level The manually-configured FIFO fill level for the ratio value of the received data.

8.1.6.38 0x009C I2S/PCM MBIST Test Configuration Register (Default Value: 0x0000_0000)

Offset: 0x009C			Register Name: ASRCMBISTCFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8	R/W	0x0	ASRC_RAM_BIST_EN ASRC RAM BIST Enable Enable the RAM BIST.
7:1	/	/	/
0	R/W	0x0	ASRC_ROM_BIST_EN ASRC ROM BIST Enable Enable the ROM BIST.

8.1.6.39 0x00A0 I2S/PCM ASRC MBIST Test State Register (Default Value: 0x0000_0002)

Offset: 0x00A0			Register Name: ASRCMBISTSTA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:19	/	/	/
18	R	0x0	ROM_BIST_ERROR_XOR ROM BIST error xor
17	R	0x0	ROM_BIST_ERROR_SUM ROM BIST error sum
16	R	0x0	ROM_BUSY_STATUS ROM BUSY STATUS 1: ROM busy 0: ROM idle
15:8	/	/	/
7	R	0x0	RAM_BIST_ERR_STATUS RAM BIST error status 1: Error 0: No effect
6:4	R	0x0	RAM_BIST_ERROR_PATTERN. RAM BIST error pattern
3:2	R	0x0	RAM_BIST_ERROR_CYCLE RAM BIST error cycle
1	R	0x1	RAM_STOP_STATUS RAM stop status 1: Stop 0: Running
0	R	0x0	RAM_BUSY_STATUS RAM busy status 1: RAM busy 0: RAM idle

8.2 DMIC

8.2.1 Overview

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

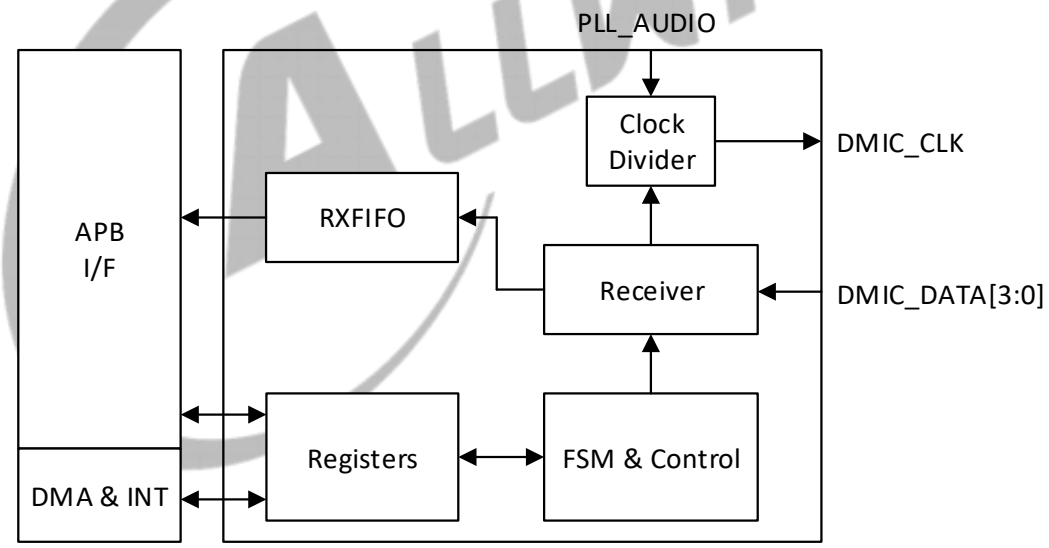
The DMIC controller includes the following features:

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

8.2.2 Block Diagram

The following figure shows a block diagram of the DMIC.

Figure 8-12 DMIC Block Diagram



8.2.3 Functional Description

8.2.3.1 External Signals

The following table describes the external signals of DMIC.

Table 8-5 DMIC External Signals

Signal	Description	Type
DMIC-CLK	Digital Microphone Clock Output	O
DMIC-DATA0	Digital Microphone Data Input	I
DMIC-DATA1	Digital Microphone Data Input	I
DMIC-DATA2	Digital Microphone Data Input	I
DMIC-DATA3	Digital Microphone Data Input	I

8.2.3.2 Clock Sources

The following table describes the clock source for DMIC. For clock setting, configurations, and gating information, refer to section 3.3 "CCU".

Table 8-6 DMIC Clock Sources

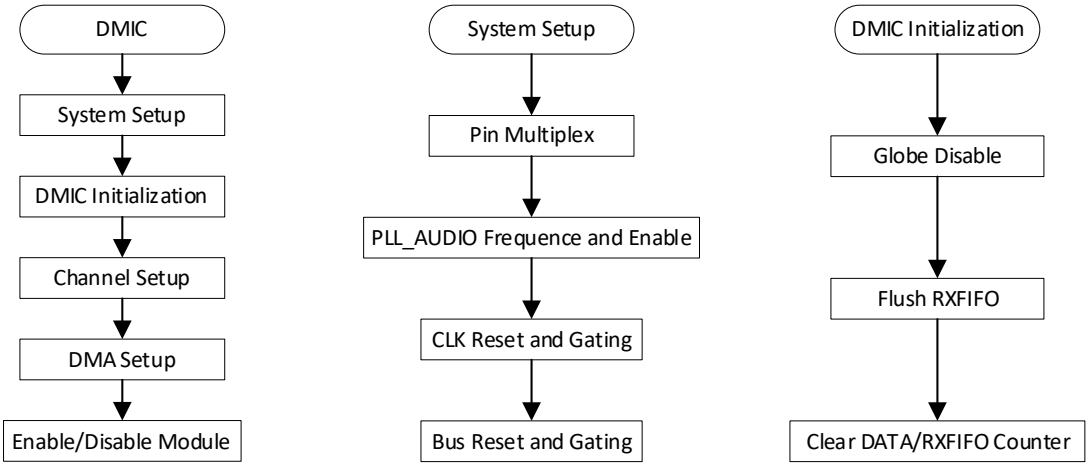
Clock Sources	Description
PLL_AUDIO0(1X)	By default, PLL_AUDIO0(1X) is 24.5714 MHz.
PLL_AUDIO1(DIV2)	By default, PLL_AUDIO1 is 3072 MHz, PLL_AUDIO1(DIV2) is 1536 MHz, and PLL_AUDIO1(DIV5) is 614.4 MHz (24.576 MHz*25).
PLL_AUDIO1(DIV5)	

8.2.3.3 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 8-13 DMIC Operation Mode



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 the pin multiplex specification.

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**.

2. Channel 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 3.9 “DMAC”. In this module, you just enable the DRQ.

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.

8.2.4 Register List

Module Name	Base Address
DMIC	0x02031000

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	MIC 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 Coefficient Register
HPF_GAIN_REG	0x0040	High Pass Filter Gain Register

8.2.5 Register Description

8.2.5.1 0x0000 DMIC Enable Control Register (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: DMIC_EN
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/

Offset: 0x0000			Register Name: DMIC_EN
Bit	**Read/Write**	**Default/Hex**	**Description**
29	R/W	0x0	RX_SYNC_EN_START Audio Subsys RX Synchronize Enable Start Includes Audio codec/I2S1/I2S2/DMIC/OWA RX. The bit takes effect only when RX_SYNC_EN is set to 1. 0: Disabled 1: Enabled
28	R/W	0x0	RX_SYNC_EN DMIC RX Synchronize Enable 0: Disabled 1: Enabled
27:9	/	/	/
8	R/W	0x0	GLOBE_EN DMIC Globe Enable 0: Disabled 1: Enabled
7	R/W	0x0	DATA3_CHR_EN DATA3 Right Channel Enable 0: Disabled 1: Enabled
6	R/W	0x0	DATA3_CHL_EN DATA3 Left Channel Enable 0: Disabled 1: Enabled
5	R/W	0x0	DATA2_CHR_EN DATA2 Right Channel Enable 0: Disabled 1: Enabled
4	R/W	0x0	DATA2_CHL_EN DATA2 Left Channel Enable 0: Disabled 1: Enabled
3	R/W	0x0	DATA1_CHR_EN DATA1 Right Channel Enable 0: Disabled 1: Enabled

Offset: 0x0000			Register Name: DMIC_EN
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R/W	0x0	DATA1_CHL_EN DATA1 Left Channel Enable 0: Disabled 1: Enabled
1	R/W	0x0	DATA0_CHR_EN DATA0 Right Channel Enable 0: Disabled 1: Enabled
0	R/W	0x0	DATA0_CHL_EN DATA0 Left Channel Enable 0: Disabled 1: Enabled

8.2.5.2 0x0004 DMIC Sample Rate Register (Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: DMIC_SR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2:0	R/W	0x0	DMIC_SR Sample Rate of DMIC 000: 48 kHz 010: 24 kHz 100: 12 kHz 110: Reserved 001: 32 kHz 011: 16 kHz 101: 8 kHz 111: Reserved 44.1 kHz/22.05 kHz/11.025 kHz can be supported by Audio PLL Configure Bit.

8.2.5.3 0x0008 DMIC Control Register (Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: DMIC_CTR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:9	R/W	0x0	DMICFDT DMIC RXFIFO Delay Time for Writing Data after GLOBE_EN 00: 5 ms 01: 10 ms 10: 20 ms 11: 30 ms
8	R/W	0x0	DMICDFEN DMIC RXFIFO Delay Function for Writing Data after GLOBE_EN 0: Disabled 1: Enabled
7	R/W	0x0	DATA3 Left Data and Right Data Swap Enable 0: Disabled 1: Enabled
6	R/W	0x0	DATA2 Left Data and Right Data Swap Enable 0: Disabled 1: Enabled
5	R/W	0x0	DATA1 Left Data and Right Data Swap Enable 0: Disabled 1: Enabled
4	R/W	0x0	DATA0 Left Data and Right Data Swap Enable 0: Disabled 1: Enabled
3:1	/	/	/
0	R/W	0x0	DMIC Oversample Rate 0: 128 (Supports 8 kHz to 24 kHz) 1: 64 (Supports 16 kHz to 48 kHz)

8.2.5.4 0x0010 DMIC DATA Register (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: DMIC_DATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	DMIC_DATA

8.2.5.5 0x0014 DMIC Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x0014			Register Name: DMIC_INTC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2	R/W	0x0	RXFIFO_DRQ_EN DMIC RXFIFO Data Available DRQ Enable 0: Disabled 1: Enabled
1	R/W	0x0	RXFIFO_OVERRUN_IRQ_EN DMIC RXFIFO Overrun IRQ Enable 0: Disabled 1: Enabled
0	R/W	0x0	DATA_IRQ_EN DMIC RXFIFO Data Available IRQ Enable 0: Disabled 1: Enabled

8.2.5.6 0x0018 DMIC Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: DMIC_INTS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
1	R/W1C	0x0	RXFIFO_OVERRUN_IRQ_PENDING DMIC RXFIFO Overrun Pending Interrupt 0: No pending IRQ 1: RXFIFO overrun pending IRQ Writing '1' to clear this interrupt or automatically clear if the interrupt condition fails.
0	R/ W1C	0x0	RXFIFO_DATA_IRQ_PENDING DMIC RXFIFO Data Available Pending Interrupt 0: No pending IRQ 1: Data available pending IRQ Writing '1' to clear this interrupt or automatically clear if the interrupt condition fails.

8.2.5.7 0x001C DMIC RXFIFO Control Register (Default Value: 0x0000_0040)

Offset: 0x001C			Register Name: DMIC_RXFIFO_CTR
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W1C	0x0	DMIC_RXFIFO_FLUSH DMIC RXFIFO Flush Writing '1' to flush RXFIFO, self clear to '0'
30:10	/	/	/
9	R/W	0x0	RXFIFO_MODE RXFIFO Output Mode (Mode 0, 1) 0: Expanding '0' at LSB of RXFIFO register 1: Expanding received sample sign bit at MSB of RXFIFO register For 24-bit received audio sample: Mode 0: RXDATA[31:0] = {RXFIFO_O[20:0], 11'h0} Mode 1: RXDATA[31:0] = {8{RXFIFO_O[20]}, RXFIFO_O[20:0], 3'h0} For 16-bit received audio sample: Mode 0: RXDATA[31:0] = {RXFIFO_O[20:5], 16'h0} Mode 1: RXDATA[31:0] = {16{RXFIFO_O[20]}, RXFIFO_O[20:5]}
8	R/W	0x0	Sample_Resolution 0: 16-bit 1: 24- bit
7:0	R/W	0x40	RXFIFO_TRG_LEVEL RXFIFO Trigger Level (TRLV[7:0]) Interrupt and DMA request trigger level for DMIC RXFIFO normal condition IRQ/DRQ Generated when WLEVEL > TRLV[7:0]) WLEVEL represents the number of valid samples in the DMIC RXFIFO

8.2.5.8 0x0020 DMIC RXFIFO Status Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: DMIC_RXFIFO_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8	R	0x1	Reserved

Offset: 0x0020			Register Name: DMIC_RXFIFO_STA
Bit	Read/Write	Default/Hex	Description
7:0	R/W	0x0	DMIC_DATA_CNT DMIC RXFIFO Available Sample Word Counter

8.2.5.9 0x0024 DMIC Channel Numbers Register (Default Value: 0x0000_0001)

Offset: 0x0024			Register Name: DMIC_CH_NUM
Bit	Read/Write	Default/Hex	Description
31:3	/	/	/
2:0	R/W	0x1	DMIC_CH_NUM DMIC enable channel numbers are (N + 1).

8.2.5.10 0x0028 DMIC Channel Mapping Register (Default Value: 0x7654_3210)

Offset: 0x0028			Register Name: DMIC_CH_MAP
Bit	Read/Write	Default/Hex	Description
31:28	R/W	0x7	DMIC_CH7_MAP DMIC Channel 7 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel

Offset: 0x0028			Register Name: DMIC_CH_MAP
Bit	**Read/Write**	**Default/Hex**	**Description**
27:24	R/W	0x6	DMIC_CH6_MAP DMIC Channel 6 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel
23:20	R/W	0x5	DMIC_CH5_MAP DMIC Channel 5 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel
19:16	R/W	0x4	DMIC_CH4_MAP DMIC Channel 4 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel

Offset: 0x0028			Register Name: DMIC_CH_MAP
Bit	**Read/Write**	**Default/Hex**	**Description**
15:12	R/W	0x3	DMIC_CH3_MAP DMIC Channel 3 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel
11:8	R/W	0x2	DMIC_CH2_MAP DMIC Channel 2 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel
7:4	R/W	0x1	DMIC_CH1_MAP DMIC Channel 1 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel

Offset: 0x0028			Register Name: DMIC_CH_MAP
Bit	**Read/Write**	**Default/Hex**	**Description**
3:0	R/W	0x0	DMIC_CH0_MAP DMIC Channel0 Mapping 0000: DATA0 Left Channel 0001: DATA0 Right Channel 0010: DATA1 Left Channel 0011: DATA1 Right Channel 0100: DATA2 Left Channel 0101: DATA2 Right Channel 0110: DATA3 Left Channel 0111: DATA3 Right Channel

8.2.5.11 0x002C DMIC Counter Register (Default Value: 0x0000_0000)

Offset: 0x002C			Register Name: DMIC_CNT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	DMIC_CNT RX Sample Counter The audio sample number of writing into RXFIFO. When one sample is read by Digital Audio Engine, the RX sample counter register increases by one. The RX sample counter register can be set to any initial valve at any time. After being updated by the initial value, the counter register should count on the base of this initial value. **Note: It is used for Audio/Video Synchronization.**

8.2.5.12 0x0030 DATA0 and DATA1 Volume Control Register (Default Value: 0xA0A0_A0A0)

Offset: 0x0030			Register Name: DATA0_DATA1_VOL_CTR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R/W	0xA0	DATA1L_VOL Data1 Left Channel Volume Control (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ... 0xFF: 71.25 dB
23:16	R/W	0xA0	DATA1R_VOL Data1 Right Channel Volume Control (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ... 0xFF: 71.25 dB
15:8	R/W	0xA0	DATA0L_VOL Data0 Left Channel Volume Control (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ... 0xFF: 71.25 dB

Offset: 0x0030			Register Name: DATA0_DATA1_VOL_CTR
Bit	**Read/Write**	**Default/Hex**	**Description**
7:0	R/W	0xA0	DATA0R_VOL Data0 Right Channel Volume Control (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ... 0xFF: 71.25 dB

8.2.5.13 0x0034 DATA2 and DATA3 Volume Control Register (Default Value: 0xA0A0_A0A0)

Offset: 0x0034			Register Name: DATA2_DATA3_VOL_CTR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R/W	0xA0	DATA3L_VOL Data3 Light Channel Volume Control (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ... 0xFF: 71.25 dB

Offset: 0x0034			Register Name: DATA2_DATA3_VOL_CTR
Bit	**Read/Write**	**Default/Hex**	**Description**
23:16	R/W	0xA0	DATA3R_VOL Data3 Right Channel Volume Control (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ... 0xFF: 71.25 dB
15:8	R/W	0xA0	DATA2L_VOL Data2 Light Channel Volume Control (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ... 0xFF: 71.25 dB
7:0	R/W	0xA0	DATA2R_VOL Data2 Right Channel Volume Control (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ... 0xFF: 71.25 dB

8.2.5.14 0x0038 High Pass Filter Enable Control Register (Default Value: 0x0000_0000)

Offset: 0x0038			Register Name: HPF_EN_CTR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	HPF_DATA3_CHR_EN High Pass Filter DATA3 Right Channel Enable 0: Disabled 1: Enabled
6	R/W	0x0	HPF_DATA3_CHL_EN High Pass Filter DATA3 Left Channel Enable 0: Disabled 1: Enabled
5	R/W	0x0	HPF_DATA2_CHR_EN High Pass Filter DATA2 Right Channel Enable 0: Disabled 1: Enabled
4	R/W	0x0	HPF_DATA2_CHL_EN High Pass Filter DATA2 Left Channel Enable 0: Disabled 1: Enabled
3	R/W	0x0	HPF_DATA1_CHR_EN High Pass Filter DATA1 Right Channel Enable 0: Disabled 1: Enabled
2	R/W	0x0	HPF_DATA1_CHL_EN High Pass Filter DATA1 Left Channel Enable 0: Disabled 1: Enabled
1	R/W	0x0	HPF_DATA0_CHR_EN High Pass Filter DATA0 Right Channel Enable 0: Disabled 1: Enabled
0	R/W	0x0	HPF_DATA0_CHL_EN High Pass Filter DATA0 Left Channel Enable 0: Disabled 1: Enabled

8.2.5.15 0x003C High Pass Filter Coefficient Register (Default Value: 0x00FF_AA45)

Offset: 0x003C			Register Name: HPF_COEF_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x00FFAA45	HPF_COE High Pass Filter Coefficient

8.2.5.16 0x0040 High Pass Filter Gain Register (Default Value: 0x00FF_D522)

Offset: 0x0040			Register Name: HPF_GAIN_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x00FFD522	HPF_GAIN High Pass Filter Gain

8.3 OWA

8.3.1 Overview

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

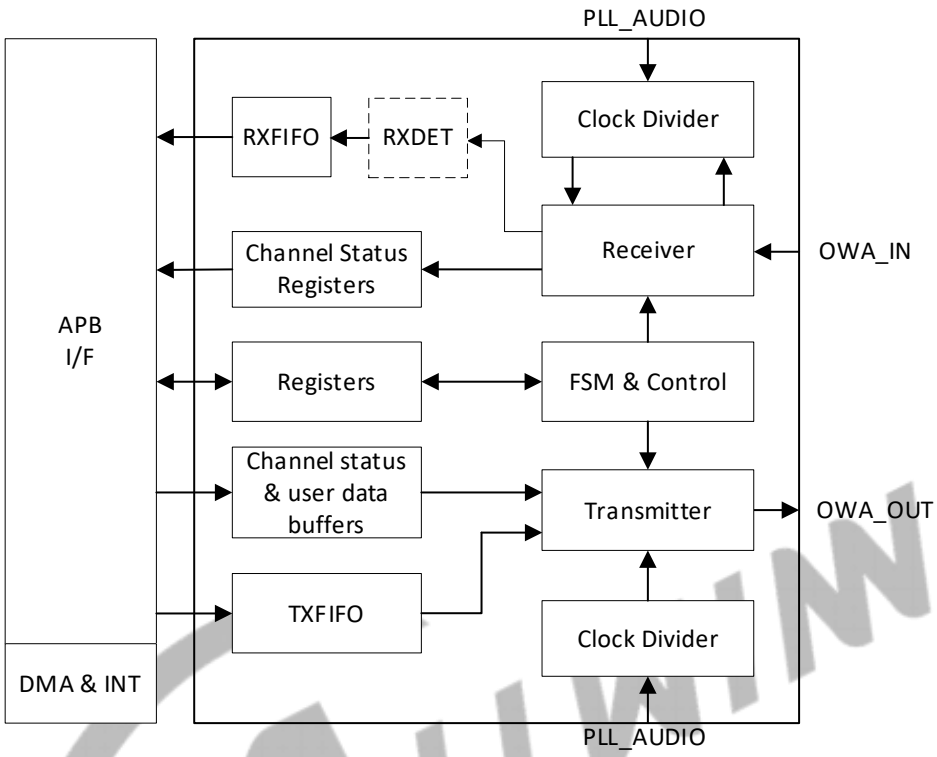
The OWA includes the following features:

- Compliance with S/PDIF Interface
- Compatible with standard IEC-60958 and IEC-61937
  - IEC-60958 supports 16-bit, 20-bit and 24-bit data formats
  - IEC-61937 uses the IEC-60958 series for the conveying of non-linear PCM bit streams, each sub-frame transmits 16-bit
- 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
- Supports multiple function clock
  - Separate clock for OWA TX and OWA RX
  - The clock of TX function includes a series of 24.576 MHz and 22.579 MHz frequency
  - The clock of RX function includes a series of 24.576*8 MHz frequency (RX function clock 24.576*8 MHz supports CDR of sample rate from 8 kHz to 192 kHz)
- Supports hardware parity on TX/RX
  - Hardware parity generation on the transmitter
  - Hardware parity checking on the receiver
- Supports channel status capture for the receiver
- Supports channel sample rate capture on the receiver
- Supports insertion detection for the receiver
- Supports channel status insertion for the transmitter
- Supports interrupts and DMA

8.3.2 Block Diagram

The following figure shows the OWA block diagram.

Figure 8-14 OWA Block Diagram



OWA contains the following sub-blocks:

Table 8-7 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

8.3.3 Functional Description

8.3.3.1 External Signals

The OWA is a Biphase-Mark Encoding Digital Audio Transfer protocol. In this protocol, the CLK signal and data signal are transferred in the same line. The following figure 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 8-8 OWA External Signals

Signal Name	Description	Type
OWA-OUT	OWA output	O
OWA-IN	OWA input	I

8.3.3.2 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 3.3 "CCU".

Table 8-9 OWA_TX Clock Sources

Clock Name	Description
PLL_AUDIO0(1X)	By default, PLL_AUDIO0(1X) is 24.5714 MHz, and PLL_AUDIO0(4X) is 98.2856 MHz.
PLL_AUDIO0(4X)	
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).
PLL_AUDIO1(DIV5)	

Table 8-10 OWA_RX Clock Sources

Clock Name	Description
PLL_PERI(1X)	The default frequency of PLL_PERI(1X) is 600 MHz.
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).
PLL_AUDIO1(DIV5)	

8.3.3.3 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 8-15 OWA Biphase-Mark Code

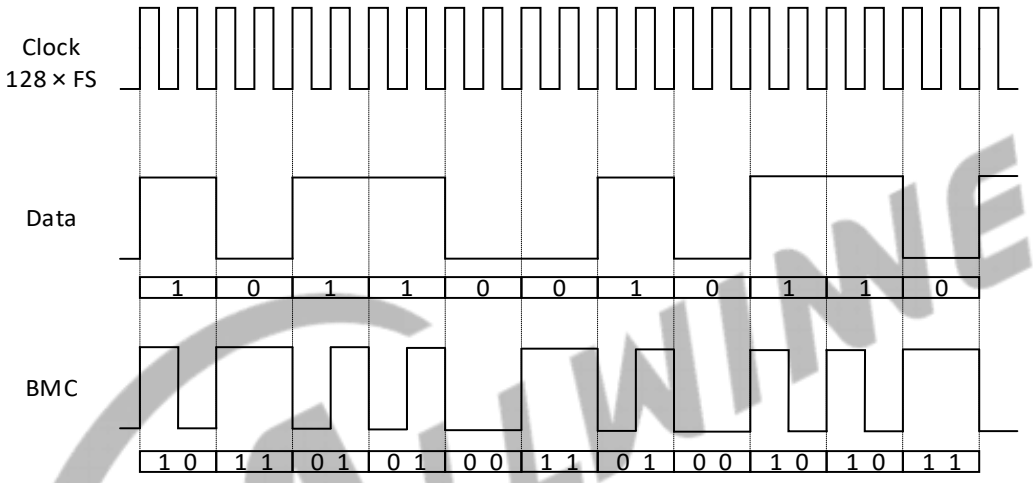


Table 8-11 Biphase-Mark Encoder

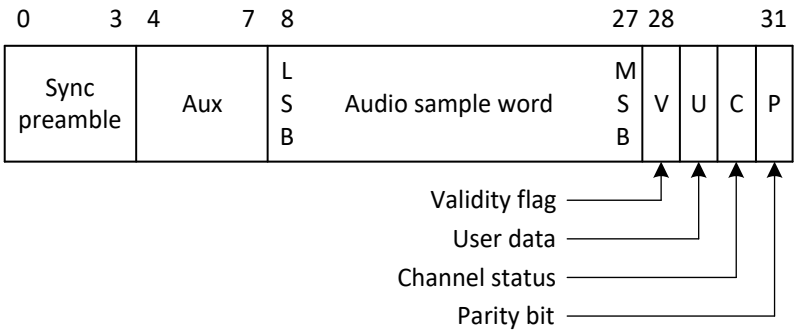
Data	Previous State	BMC
0	0	11
0	1	00
1	0	10
1	1	01

8.3.3.4 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 8-16 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.

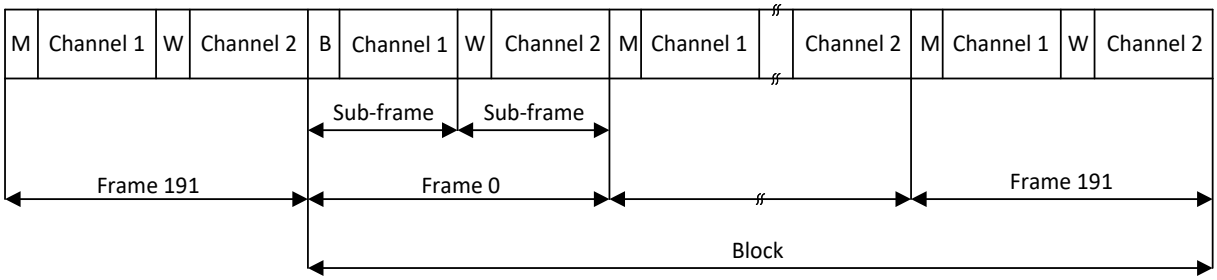
Bit 30 carries the channel status information (C) associated with the main data field in the sub-frame. The channel status indicates if the data in the sub-frame is a digital audio or some other type of data.

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 8-12 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 8-17 OWA Frame/Block Format



8.3.3.5 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.

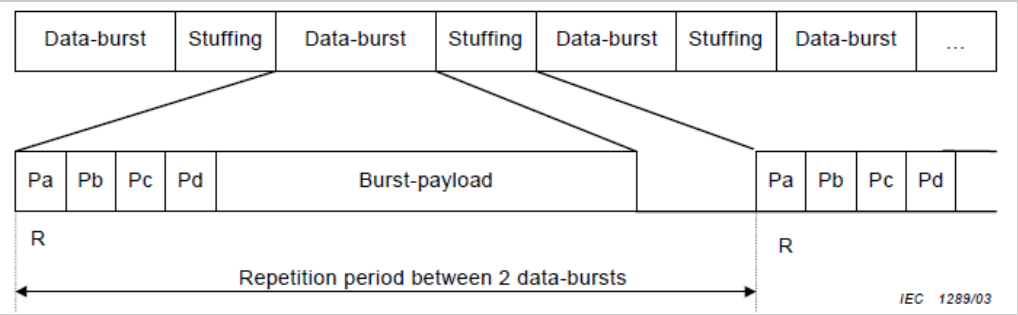
Table 8-13 Bit Allocation of Data-Burst in IEC 60958 Subframes

Subframe	Bit of subframes				
	MSB b27	b26	b25 ..... b14	b13	LSB 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.

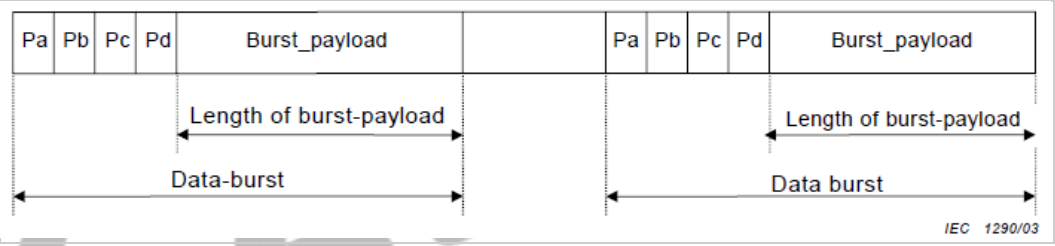
Figure 8-18 Data-Burst Format



(1) 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 8-19 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 8-20 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

(2) Burst-information

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

Figure 8-21 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.

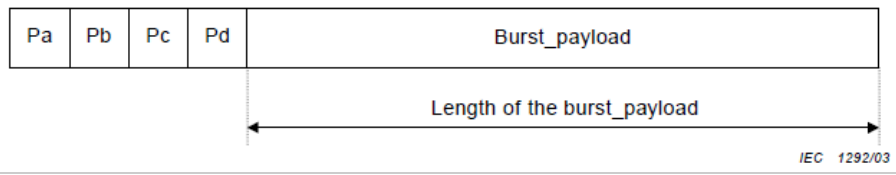
The meaning of the 5-bit data-type-dependent info depends on the value of the data-type.

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.

(3) 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 8-22 Length of the Burst-Payload Specified by Pd



8.3.3.6 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.

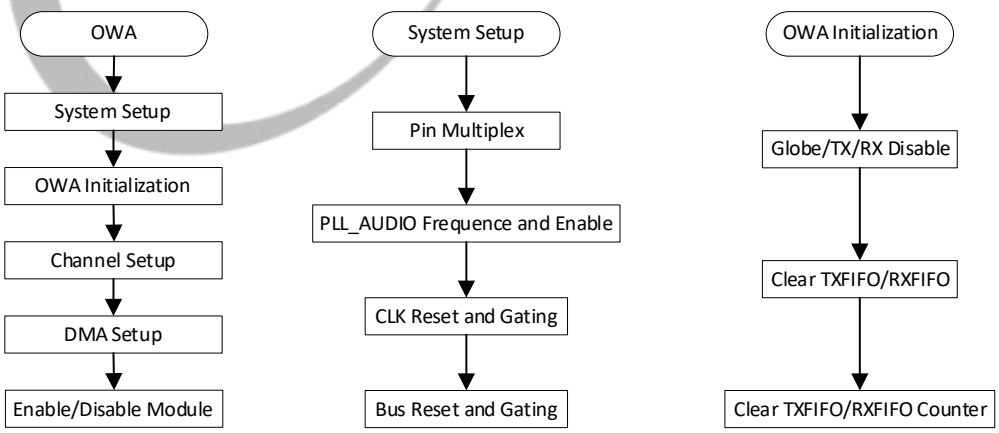
Table 8-14 The Corresponding Relation between Different System Clock and Sample Ratio

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

8.3.3.7 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 8-23 OWA Operation Flow



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 9.7 “”.

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**.

2. Channel 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].

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.

8.3.4 Register List

Module Name	Base Address
OWA	0x02036000

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

Register Name	Offset	Description
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
OWA_EXP_CTL	0x0040	OWA Expand Control Register
OWA_EXP_ISTA	0x0044	OWA Expand Interrupt Status Register
OWA_EXP_INFO_0	0x0048	OWA Expand Infomation Register0
OWA_EXP_INFO_1	0x004C	OWA Expand Infomation Register1
OWA_EXP_DBG_0	0x0050	OWA Expand Debug Register0
OWA_EXP_DBG_1	0x0054	OWA Expand Debug Register1

8.3.5 Register Description

8.3.5.1 0x0000 OWA General Control Register (Default Value: 0x0000_0080)

Offset: 0x0000			Register Name: OWA_CTL
Bit	Read/Write	Default/Hex	Description
31:13	/	/	/
12	R/WAC	0x0	RST_RX Reset RX 0: Normal 1: Reset Self clear to '0'.
11:8	/	/	/
7	R	0x1	Reserved
6:3	/	/	/
2	R/W	0x0	LOOP Loopback Test 0: Normal Mode 1: Loopback Test When the bit is set to '1', the DOUT and DIN need be connected.

Offset: 0x0000			Register Name: OWA_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
1	R/W	0x0	GEN Global Enable Disabling this bit overrides the operations of enabling and flushing all FIFOs by any other blocks or channels. 0: Disabled 1: Enabled
0	R/W	0x0	RST_TX Reset TX 0: Normal 1: Reset Self clear to 0.

8.3.5.2 0x0004 OWA TX Configure Register (Default Value: 0x0000_00F0)

Offset: 0x0004			Register Name: OWA_TX_CFIG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	TX_SINGLE_MODE Tx Single Channel Mode 0: Disabled 1: Enabled
30:18	/	/	/
17	R/W	0x0	ASS Audio Sample Select when TX FIFO Underrun 0: Sending 0 1: Sending the last audio **Note: This bit is only valid in PCM mode.**
16	R/W	0x0	TX_AUDIO TX Data Type 0: Linear PCM (Valid bit of both sub-frame set to 0) 1: Non-audio (Valid bit of both sub-frame set to 1)
15:9	/	/	/
8:4	R/W	0xF	TX_RATIO TX Clock Divide Ratio Clock divide ratio = TX_TATIO + 1 Fs = PLL_AUDIO/[(TX_TATIO + 1)*64*2]

Offset: 0x0004			Register Name: OWA_TX_CFIG
Bit	Read/Write	Default/Hex	Description
3:2	R/W	0x0	TX_SF TX Sample Format 00: 16 bits 01: 20 bits 10: 24 bits 11: Reserved
1	R/W	0x0	TX_CHM CHSTMODE 0: Channel status A and B set to 0 1: Channel status A and B generated from TX_CHSTA
0	R/W	0x0	TXEN TX Enable 0: Disabled 1: Enabled

8.3.5.3 0x0008 OWA RX Configure Register (Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: OWA_RX_CFIG
Bit	Read/Write	Default/Hex	Description
31:5	/	/	/
4	R	0x0	RX_LOCK_FLAG RX Lock Flag 0: Unlocked 1: Locked
3	R/W	0x0	RX_CHST_SRC RX Channel State Source Select 0: RX_CH_STA register holds status from Channel A 1: RX_CH_STA register holds status from Channel B
2	/	/	/
1	R/W	0x0	CHST_CP Channel Status Capture 0: Idle or Capture End 1: Capture Channel Status Start The field must be set to 1 at each operation (such as recording). When set to '1', the system starts to capture the channel status. When finished, the bit will automatically turn to '0'.

Offset: 0x0008			Register Name: OWA_RX_CFIG
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	RXEN 0: Disabled 1: Enabled

8.3.5.4 0x000C OWA Interrupt Status Register (Default Value: 0x0000_0010)

Offset: 0x000C			Register Name: OWA_ISTA
Bit	Read/Write	Default/Hex	Description
31:19	/	/	/
18	R/W1C	0x0	RX_LOCK_INT RX Lock Interrupt 0: No Pending IRQ 1: RX Lock Pending Interrupt (RX_LOCK_FLAG turns from 0 to 1) Write '1' to clear this interrupt.
17	R/W1C	0x0	RX_UNLOCK_INT RX Unlock Pending Interrupt 0: No Pending IRQ 1: RX Unlock Pending Interrupt (RX_LOCK_FLAG turns from 0 to 1) Write 1 to clear this interrupt.
16	R/W1C	0x0	RX_PARERRI_INT RX Parity Error Pending Interrupt 0: No Pending IRQ 1: RX Parity Error Pending Interrupt Write "1" to clear this interrupt.
15:7	/	/	/
6	R/W1C	0x0	TXU_INT TX FIFO Underrun Pending Interrupt 0: No Pending IRQ 1: FIFO Underrun Pending Interrupt Writing "1" to clear this interrupt.

Offset: 0x000C			Register Name: OWA_ISTA
Bit	**Read/Write**	**Default/Hex**	**Description**
5	R/W1C	0x0	TXO_INT TX FIFO Overrun Pending Interrupt 0: No Pending IRQ 1: FIFO Overrun Pending Interrupt Writing “1” to clear this interrupt.
4	R/W1C	0x1	TXE_INT TX FIFO Empty Pending Interrupt 0: No Pending IRQ 1: FIFO Empty Pending Interrupt Writing “1” to clear this interrupt or automatically clear if the interrupt condition fails.
3:2	/	/	/
1	R/W1C	0x0	RXO_INT RXFIFO Overrun Pending Interrupt 0: RXFIFO Overrun Pending Write ‘1’ to clear this interrupt.
0	R/W1C	0x0	RXA_INT RXFIFO Available Pending Interrupt 0: No Pending IRQ 1: Data Available Pending IRQ Write ‘1’ to clear this interrupt or automatically clear if the interrupt condition fails.

8.3.5.5 0x0010 OWA RXFIFO Register (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: OWA_RXFIFO
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	RX_DATA The host can get one sample by reading this register, A channel data is first, and then the B channel data.

8.3.5.6 0x0014 OWA FIFO Control Register (Default Value: 0x0004_0200)

Offset: 0x0014			Register Name: OWA_FCTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	HUB_EN Audio Hub Enable The bit takes effect only when the TXEN is set to 1. Audio codec/I2S1/I2S2/OWA TXFIFO Hub Enable. 0: Disabled 1: Enabled
30	R/W1C	0x0	FTX Write '1' to flush TXFIFO, self clear to '0'.
29	R/W1C	0x0	FRX Write '1' to flush RXFIFO, self clear to '0'.
28:22	/	/	/
21	R/W	0x0	RX_SYNC_EN_START The bit takes effect only when the RX_SYNC_EN is set to 1. Audio Codec/I2S1/I2S2/DMIC/OWA RX Synchronize Enable Start. 0: Disabled 1: Enabled
20	R/W	0x0	RX_SYNC_EN OWA RX Synchronize Enable 0: Disabled 1: Enabled
19:12	R/W	0x40	TXTL TX FIFO Empty Trigger Level Interrupt and DMA request trigger level for TX FIFO normal condition. Trigger Level = TXTL
11	/	/	/
10:4	R/W	0x20	RXTL RX FIFO Trigger Level Interrupt and DMA request trigger level for RX FIFO normal condition. Trigger Level = RXTL + 1
3	/	/	/

Offset: 0x0014			Register Name: OWA_FCTL
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R/W	0x0	TXIM TXFIFO Input Mode (Mode0, 1) 0: Valid data at the MSB of TXFIFO Register 1: Valid data at the LSB of TXFIFO Register Example for 20-bit transmitted audio sample: Mode 0: TXFIFO[23:0] = {APB_WDATA[31:12], 4'h0} Mode 1: TXFIFO[23:0] = {APB_WDATA[19:0], 4'h0}
1:0	R/W	0x0	RXOM RXFIFO Output Mode(Mode 0,1,2,3) 00: Expanding '0' at LSB of RXFIFO Register 01: Expanding received sample sign bit at MSB of RXFIFO Register 10: Truncating received samples at high half-word of RXFIFO Register and low half-word of RXFIFO Register is filled by '0' 11: Truncating received samples at low half-word of RXFIFO Register and high half-word of RXFIFO Register is expanded by its sigh bit Mode 0: APB_RDATA[31:0] = {RXFIFO[23:0], 8'h0} Mode 1: APB_RDATA[31:0] = {8'RXFIFO[23], RXFIFO[23:0]} Mode 2: APB_RDATA[31:0] = {RXFIFO[23:8], 16'h0} Mode 3: APB_RDATA[31:0] = {16'RXFIFO[23], RXFIFO[23:8]}

8.3.5.7 0x0018 OWA FIFO Status Register (Default Value: 0x8080_0000)

Offset: 0x0018			Register Name: OWA_FSTA
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R	0x1	TXE TXFIFO Empty (indicate the TXFIFO is not full) 0: No room for new sample in TXFIFO 1: More than one room for new sample in TXFIFO (>= 1 Word)
30:24	/	/	/
23:16	R	0x80	TXE_CNT TXFIFO Empty Space Word Counter
15	R	0x0	RXA RXFIFO Available 0: No available data in RXFIFO 1: More than one sample in RXFIFO (>= 1 Word)

Offset: 0x0018			Register Name: OWA_FSTA
Bit	Read/Write	Default/Hex	Description
14:7	/	/	/
6:0	R	0x0	RXA_CNT RXFIFO Available Sample Word Counter

8.3.5.8 0x001C OWA Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: OWA_INT
Bit	Read/Write	Default/Hex	Description
31:19	/	/	/
18	R/W	0x0	RX_LOCKI_EN RX LOCK Interrupt Enable 0: Disabled 1: Enabled
17	R/W	0x0	RX_UNLOCKI_EN RX UNLOCK Interrupt Enable 0: Disabled 1: Enabled
16	R/W	0x0	RX_PARERRI_EN RX PARITY ERORR Interrupt Enable 0: Disabled 1: Enabled
15:8	/	/	/
7	R/W	0x0	TX_DRQ TXFIFO Empty DRQ Enable 0: Disabled 1: Enabled
6	R/W	0x0	TXUI_EN TXFIFO Underrun Interrupt Enable 0: Disabled 1: Enabled
5	R/W	0x0	TXOI_EN TXFIFO Overrun Interrupt Enable 0: Disabled 1: Enabled

Offset: 0x001C			Register Name: OWA_INT
Bit	**Read/Write**	**Default/Hex**	**Description**
4	R/W	0x0	TXEI_EN TXFIFO Empty Interrupt Enable 0: Disabled 1: Enabled
3	/	/	/
2	R/W	0x0	RX_DRQ RXFIFO Data Available DRQ Enable When set to '1', RXFIFO DMA Request is asserted if data is available in RXFIFO. 0: Disabled 1: Enabled
1	R/W	0x0	RXOI_EN RXFIFO Overrun Interrupt Enable 0: Disabled 1: Enabled
0	R/W	0x0	RXAI_EN RXFIFO Data Available Interrupt Enable 0: Disabled 1: Enabled

8.3.5.9 0x0020 OWA TX FIFO Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: OWA_TXFIFO
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	W	0x0	TX_DATA Transmitting A, B channel data should be written this register one by one. A channel data is first, and then the B channel data.

8.3.5.10 0x0024 OWA TX Counter Register (Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: OWA_TX_CNT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	TX_CNT TX Sample Counter The audio sample number of sending into TXFIFO. When one sample is put into TXFIFO by DMA or by host IO, the TX sample counter register increases by one. The TX sample counter register can be set to any initial value at any time. After updated by the initial value, the counter register should count on the base of this initial value.

8.3.5.11 0x0028 OWA RX Counter Register (Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: OWA_RX_CNT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	RX_CNT RX Sample Counter The audio sample number of writing into RXFIFO. When one sample is written by Codec, the RX sample counter register increases by one. The RX counter register can be set to any initial value at any time. After being updated by the initial value, the counter register should count on the base of this value.

8.3.5.12 0x002C OWA TX Channel Status Register0 (Default Value: 0x0000_0000)

Offset: 0x002C			Register Name: OWA_TX_CHSTA0
Bit	**Read/Write**	**Default/Hex**	**Description**
31: 30	/	/	/
29:28	R/W	0x0	CA Clock Accuracy 00: Level 2 01: Level 1 10: Level 3 11: Not matched

Offset: 0x002C			Register Name: OWA_TX_CHSTA0
Bit	**Read/Write**	**Default/Hex**	**Description**
27:24	R/W	0x0	FREQ Sampling Frequency 0000: 44.1 kHz 0001: Not indicated 0010: 48 kHz 0011: 32 kHz 0100: 22.05 kHz 0101: Reserved 0110: 24 kHz 0111: Reserved 1000: Reserved 1001: 768 kHz 1010: 96 kHz 1011: Reserved 1100: 176.4 kHz 1101: Reserved 1110: 192 kHz 1111: Reserved
23:20	R/W	0x0	CN Channel Number
19:16	R/W	0x0	SN Source Number
15:8	R/W	0x0	CC Category Code Indicates the kind of equipment that generates the digital audio interface signal.
7:6	R/W	0x0	MODE Mode 00: Default Mode 01 to 11: Reserved

Offset: 0x002C			Register Name: OWA_TX_CHSTA0
Bit	**Read/Write**	**Default/Hex**	**Description**
5:3	R/W	0x0	EMP Emphasis Additional format information For bit 1 = “0”, Linear PCM audio mode: 000: 2 audio channels without pre-emphasis 001: 2 audio channels with 50 μs/15 μs pre-emphasis 010: Reserved (for 2 audio channels with pre-emphasis) 011: Reserved (for 2 audio channels with pre-emphasis) 100 to 111: Reserved For bit 1 = “1”, other than Linear PCM applications: 000: Default state 001 to 111: Reserved
2	R/W	0x0	CP Copyright 0: Copyright is asserted 1: No copyright is asserted
1	R/W	0x0	TYPE Audio Data Type 0: Linear PCM samples 1: Non-linear PCM audio
0	R/W	0x0	PRO Application Type 0: Consumer application 1: Professional application This bit must be fixed to “0”.

8.3.5.13 0x0030 OWA TX Channel Status Register1 (Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: OWA_TX_CHSTA1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	/	/	/
9:8	R/W	0x0	CGMS_A 00: Copying is permitted without restriction 01: One generation of copies may be made 10: Condition not be used 11: No copying is permitted

Offset: 0x0030			Register Name: OWA_TX_CHSTA1
Bit	**Read/Write**	**Default/Hex**	**Description**
7:4	R/W	0x0	ORIG_FREQ Original Sampling Frequency 0000: Not indicated 0001: 192 kHz 0010: 12 kHz 0011: 176.4 kHz 0100: Reserved 0101: 96 kHz 0110: 8 kHz 0111: 88.2 kHz 1000: 16 kHz 1001: 24 kHz 1010: 11.025 kHz 1011: 22.05 kHz 1100: 32 kHz 1101: 48 kHz 1110: Reserved 1111: 44.1 kHz
3:1	R/W	0x0	WL Sample Word Length For bit 0 = “0”: 000: Not indicated 001: 16 bits 010: 18 bits 100: 19 bits 101: 20 bits 110: 17 bits 111: Reserved For bit 0 = “1”: 000: Not indicated 001: 20 bits 010: 22 bits 100: 23 bits 101: 24 bits 110: 21 bits 111: Reserved

Offset: 0x0030			Register Name: OWA_TX_CHSTA1
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	MWL Max Word Length 0: Maximum audio sample word length is 20 bits 1: Maximum audio sample word length is 24 bits

8.3.5.14 0x0034 OWA RX Channel Status Register0 (Default Value: 0x0000_0000)

Offset: 0x0034			Register Name: OWA_RX_CHSTA0
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:28	R/W	0x0	CA Clock Accuracy 00: Level 2 01: Level 1 10: Level 3 11: Not Matched
27:24	R/W	0x0	FREQ Sampling Frequency 0000: 44.1 kHz 0001: Not Indicated 0010: 48 kHz 0011: 32 kHz 0100: 22.05 kHz 0101: Reserved 0110: 24 kHz 0111: Reserved 1000: Reserved 1001: 768 kHz 1010: 96 kHz 1011: Reserved 1100: 176.4 kHz 1101: Reserved 1110: 192 kHz 1111: Reserved
23:20	R/W	0x0	CN Channel Number

Offset: 0x0034			Register Name: OWA_RX_CHSTA0
Bit	**Read/Write**	**Default/Hex**	**Description**
19:16	R/W	0x0	SN Source Number
15:8	R/W	0x0	CC Category Code Indicates the Kind of Equipment that Generates the digital audio interface Signal.
7:6	R/W	0x0	MODE Mode 00: Default mode 01 to 11: Reserved
5:3	R/W	0x0	EMP Emphasis Additional Format Information For bit 1 = '0', Linear PCM Audio mode: 000: 2 Audio channels without pre-emphasis 001: 2 Audio channels with 50 μs/15 μs pre-emphasis 010: Reserved (For 2 Audio channels with pre-emphasis) 011: Reserved (For 2 Audio channels with pre-emphasis) 100 to 111: Reserved For bit 1 = '1', Other than Linear PCM applications: 000: Default state 001 to 111: Reserved
2	R/W	0x0	CP Copyright 0: Copyright is asserted 1: No Copyright is asserted
1	R/W	0x0	TYPE Audio Data Type 0: Linear PCM samples 1: Non-linear PCM audio
0	R/W	0x0	PRO Application Type 0: Consumer application 1: Professional application

8.3.5.15 0x0038 OWA RX Channel Status Register1 (Default Value: 0x0000_0000)

Offset: 0x0038			Register Name: OWA_RX_CHSTA1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:10	/	/	/
9:8	R/W	0x0	CGMS_A 00: Copying is permitted without restriction 01: One generation of copies may be made 10: Condition is not be used 11: No copying is permitted
7:4	R/W	0x0	ORIG_FREQ Original Sampling Frequency 0000: Not indicated 0001: 192 kHz 0010: 12 kHz 0011: 176.4 kHz 0100: Reserved 0101: 96 kHz 0110: 8 kHz 0111: 88.2 kHz 1000: 16 kHz 1001: 24 kHz 1010: 11.025 kHz 1011: 22.05 kHz 1100: 32 kHz 1101: 48 kHz 1110: Reserved 1111: 44.1 kHz

Offset: 0x0038			Register Name: OWA_RX_CHSTA1
Bit	**Read/Write**	**Default/Hex**	**Description**
3:1	R/W	0x0	WL Sample Word Length For bit 0 = '0': 000: Not indicated 001: 16 bits 010: 18 bits 100: 19 bits 101: 20 bits 110: 17 bits 111: Reserved For bit 0 = '1': 000: Not indicated 001: 20 bits 010: 22 bits 100: 23 bits 101: 24 bits 110: 21 bits 111: Reserved
0	R/W	0x0	MWL Max Word Length 0: Maximum Audio sample word length is 20 bits 1: Maximum Audio sample word length is 24 bits

8.3.5.16 0x0040 OWA Expand Control Register (Default Value: 0x0000_000F)

Offset: 0x0040			Register Name: OWA_EXP_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/
30	R/W	0x0	BURST_DATAOUT_SELECT Burst data output select 0: Burst preamble and payload 1: Burst payload

Offset: 0x0040			Register Name: OWA_EXP_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
29:16	R/W	0x0	REPEAT_PERIOD_OF_FR_NUM The number for the repetition period of the burst frame Configure this field according to RX data. A mismatch between the configuration data and the received data will result in an error interrupt.
15	R/W	0x0	UNIT_SELECT Unit Select Configure this field according to RX data type 0: In units of 16-bit 1: In units of 2-byte
14	R/W	0x0	OWA_RX_MODE_MAN OWA RX Proteocol Select 0: IEC60958 1: IEC61937
13	R/W	0x0	OWA_RX_MODE OWA RX Mode Select 0: Manual Ctrl. Configure by OWA_RX_MODE_MAN 1: Auto Ctrl. Configure by the channel status values resolved by hardware
12	R/W	0x0	AUDIO_DATA_BITORDER_EN Audio data bitorder enable 0: The audio data received by RX is stored directly into FIFO 1: The audio data received by RX is reversed high and low bits, then stored into FIFO
11	R/W	0x0	DATA_LENGTH_BITORDER_EN Data length bitorder enable 0: The received PD data is as the length of the valid audio data 1: The received PD data is reversed high and low bits, then as the length of the valid audio data
10	R/W	0x0	DATA_TYPE_BITORDER_EN Data type bitorder enable 0: The received PC data is as the data length of the valid audio 1: The received PC data is reversed high and low bits, then as the length of the valid audio data

Offset: 0x0040			Register Name: OWA_EXP_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
9	R/W	0x0	SYNCW_BITORDER_EN Syncw_bitorder_en 0: Pa/Pb is the sync code of audio data 1: Pa/Pb reversed high and low bits is the sync code of audio data
8	R/W	0x0	INSERT_DETECTION_ENABLE Insert detection enable 0: Disable 1: Enable
7:0	R/W	0x0F	INSERT_DETECTION_NUM Insert detection number Configure how many jumping edges are detected to generate an insertion interrupt

8.3.5.17 0x0044 OWA Expand Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0044			Register Name: OWA_EXP_ISTA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24	R/W	0x0	PD_CHANGE_INT_EN PD_LENGTH_CHANGE Interrupt Enable 0: Disable 1: Enable
23	R/W	0x0	PC_PAUSE_STOP_INT PC_PAUSE_BURSTS_STOP Interrupt Enable 0: Disable 1: Enable
22	R/W	0x0	PC_BITSTRM_CHANGE_INT_EN PC_BITSTREAM_CHANGE Interrupt Enable 0: Disable 1: Enable
21	R/W	0x0	PC_ERR_FLAG_INT PC_ERROR_FLAG Interrupt Enable 0: Disable 1: Enable

Offset: 0x0044			Register Name: OWA_EXP_ISTA
Bit	**Read/Write**	**Default/Hex**	**Description**
20	R/W	0x0	PC_DTYPE_CHANGE_INT_EN PC_DATATYPE_CHANGE Interrupt Enable 0: Disable 1: Enable
19	R/W	0x0	RPDB_ERR_INT_EN RPDB_ERROR Interrupt Enable 0: Disable 1: Enable
18	R/W	0x0	PCPD_CAP_INT_EN PCPD_CAP Interrupt Enable 0: Disable 1: Enable
17	R/W	0x0	PAPB_CAP_INT_EN PAPB_CAP Interrupt Enable 0: Disable 1: Enable
16	R/W	0x0	INSERT_INT_EN INSERT Interrupt Enable 0: Disable 1: Enable
15:9	/	/	/
8	R/W1C	0x0	PD_CHANGE_INT PD CHANGE INT 0: No Pending IRQ 1: PD Data length information is change. (except Pause/Null data burst type) Write '1' to clear this interrupt.
7	R/W1C	0x0	PC_PAUSE_STOP_INT Audio bitstream is interrupted. When stopped, the interface becomes idle. 0: No Pending IRQ 1: PC Pause burst Stop, frame sequence discontinued. Transmitters may optionally use the STOP value to indicate that the transmission of the current encoded Write '1' to clear this interrupt.

Offset: 0x0044			Register Name: OWA_EXP_ISTA
Bit	**Read/Write**	**Default/Hex**	**Description**
6	R/W1C	0x0	PC_BITSTRM_CHANGE_INT PC BITSTRM CHANGE INT 0: No Pending IRQ 1: PC Bitstream Number is change. Bitstream Number indicates which bitstream the data burst belongs. (except Pause/Null data bursts type) Write '1' to clear this interrupt.
5	R/W1C	0x0	PC_ERR_FLAG_INT PC ERR FLAG INT 0: No Pending IRQ 1: PC Error-flag is available to indicate if the contents of the data-burst contain data errors (except Pause/Null data bursts type) . The using of this bit by receivers is optional. Write '1' to clear this interrupt.
4	R/W1C	0x0	PC_DTYPE_CHANGE_INT PC DTYPE CHANGE INT 0: No Pending IRQ 1: PC Datatype (except Pause/Null data type) information is change Write '1' to clear this interrupt.
3	R/W1C	0x0	RPDB_ERR_INT RPDB ERR INT 0: No Pending IRQ 1: Hardware counts the repetition period of the burst frame is different from register configuration number Write '1' to clear this interrupt.
2	R/W1C	0x0	PCPD_CAP_INT PCPD CAP INT 0: No Pending IRQ 1: IEC61937 mode captures PC and PD Write '1' to clear this interrupt.
1	R/W1C	0x0	PAPB_CAP_INT PAPB CAP INT 0: No Pending IRQ 1: IEC61937 mode captures PA and PB Write '1' to clear this interrupt.

Offset: 0x0044			Register Name: OWA_EXP_ISTA
Bit	Read/Write	Default/Hex	Description
0	R/W1C	0x0	INSERT_INT INSERT INT 0: No Pending IRQ 1: OWA RX detects device insertion Write '1' to clear this interrupt.

8.3.5.18 0x0048 OWA Expand Information Register 0 (Default Value: 0x0000_0000)

Offset: 0x0048			Register Name: OWA_EXP_INFO_0
Bit	Read/Write	Default/Hex	Description
31:16	R	0x0	PC_DATA PC Data information
15:0	R	0x0	PD_DATA PD Data information

8.3.5.19 0x004C OWA Expand Information Register 1 (Default Value: 0x0000_0000)

Offset: 0x004C			Register Name: OWA_EXP_INFO_1
Bit	Read/Write	Default/Hex	Description
31:30	/	/	/
29:16	R	0x0	REPET_PERIOD_OF_FR_VALUE Repetition period of the burst frame value Check whether the repetition period of the burst frame calculated by hardware is consistent with the configuration value.
15:0	R	0x0	SR_VALUE Sample Rate Value Read this value after RX_LOCK.

8.3.5.20 0x0050 OWA Expand Debug Register 0 (Default Value: 0x0000_0000)

Offset: 0x0050			Register Name: OWA_EXP_DBG_0
Bit	**Read/Write**	**Default/Hex**	**Description**
31:19	/	/	/
18:16	R	0x0	IEC61937_DATA_CAP_FSM IEC61937 Data Captures State Machine 000: IDLE 001: SYNC_PA 010: SYNC_PB 011: DTYPE_PC 100: DLEN_PD 101: RX_ACTIVE
15:0	R	0x0	DATA_CAP_NUM Remains Data Counter Value See the value of the sampled valid data in real time.

8.3.5.21 0x0054 OWA Expand Debug Register 1 (Default Value: 0x0000_0000)

Offset: 0x0054			Register Name: OWA_EXP_DBG_1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/
29:16	R	0x0	REPET_PERIOD_OF_FR_CNT Repetition period of the burst frame counter See the value of repetition period counter in real time.
15:0	R	0x0	SR_CNT Sample Rate Counter See the value of audio sample ratio in real time.

8.4 Audio Codec

8.4.1 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:

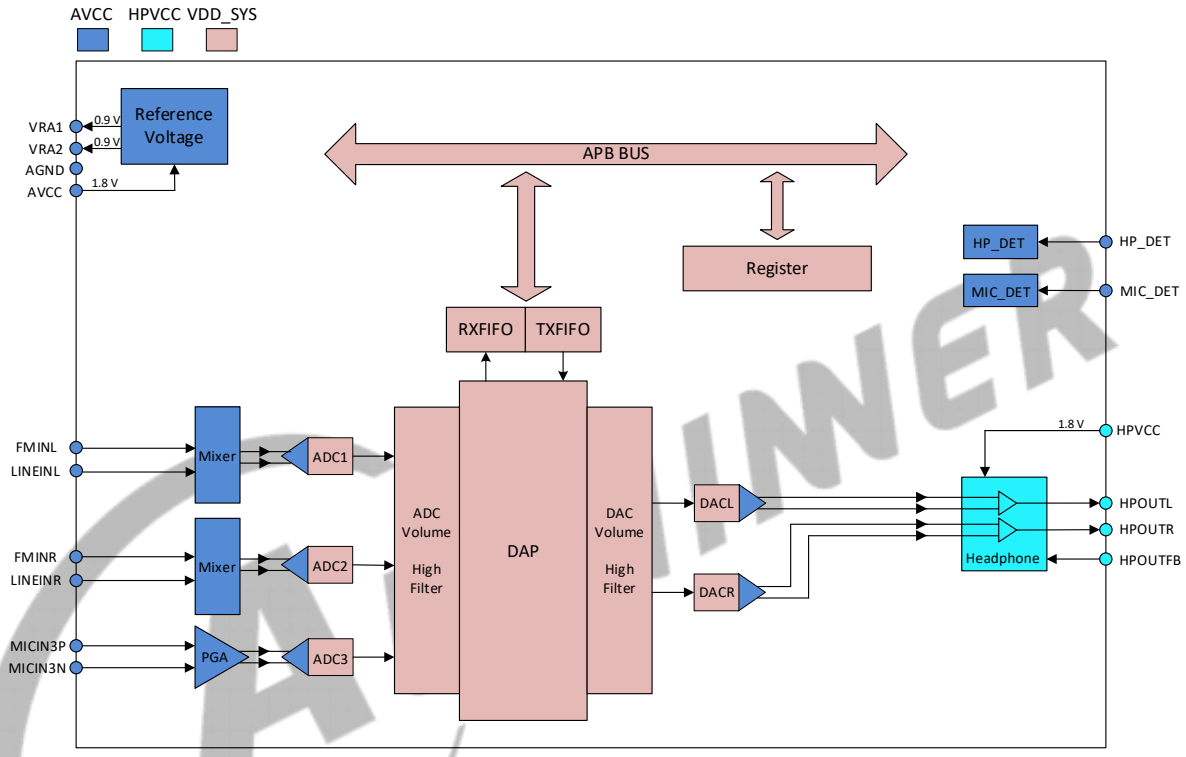
- Two audio digital-to-analog converter (DAC) channels
  - Supports the DAC sample rate from 8 kHz to 192 kHz
  - 100 ± 2 dB SNR@A-weight, -85 ± 3 dB THD + N
  - Supports 16-bit and 20-bit audio sample resolution
- One audio output
  - One stereo headphone output: HPOUTL/R
- There audio analog-to-digital converter (ADC) channels
  - Supports the ADC sample rate from 8 kHz to 48 kHz
  - 95 ± 3 dB SNR@A-weight, -80 ± 3 dB THD + N
  - Supports 16-bit and 20-bit audio sample resolution
- Three audio inputs
  - One differential microphone input: MICIN3P/3N, or one single-end microphone input: MICIN3P
  - One stereo LINEIN input: LINEINL/R
  - One stereo FMIN input: FMINL/R
- Stereo headphone driver
  - 95±3dB SNR@A-weight
  - Output Level 0.55 Vrms@10 kΩ/THD+N -77±3dB, 0.37Vrms@16 Ω/THD+N -40dB
- Supports Dynamic Range Controller (DRC) adjusting the ADC recording and DAC playback
- One 128x20-bit FIFO for DAC data transmit, one 128x20-bit FIFO for ADC data receive
- Programmable FIFO thresholds
- Supports interrupts and DMA
- Internal HPLDO output for HPVCC

- Internal ALDO output for AVCC

8.4.2 Block Diagram

The following figure shows the block diagram of Audio Codec.

Figure 8-24 Audio Codec Block Diagram



8.4.3 Functional Description

8.4.3.1 External Signals

Table 8-15 Audio Codec External Signals

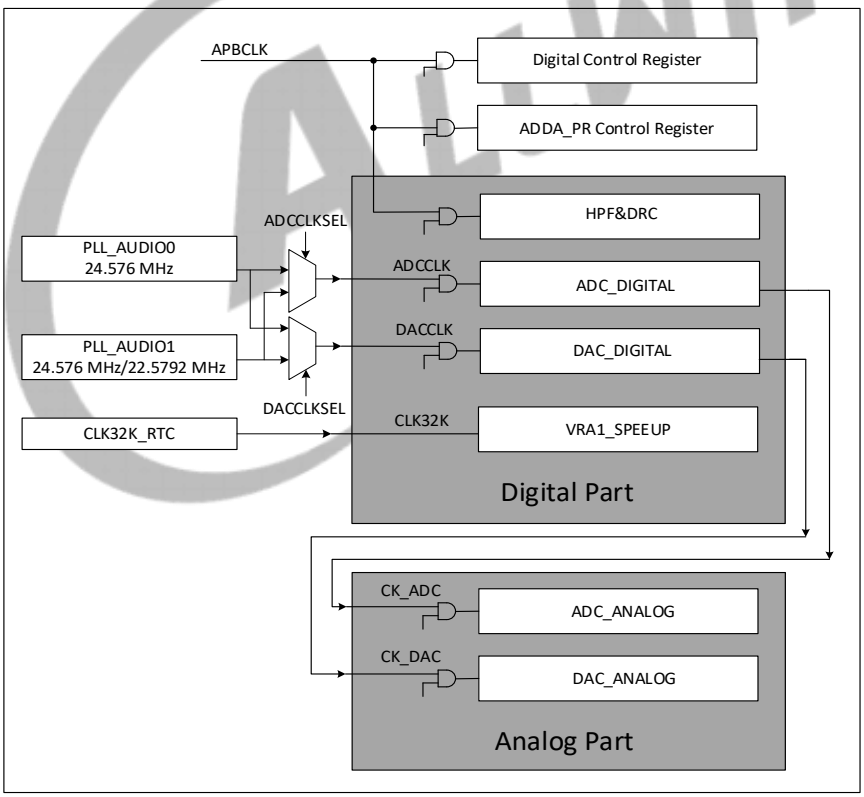
Signal	Type	Description
MICIN3P	AI	Positive Differential Input for MIC3
MICIN3N	AI	Negative Differential Input for MIC3
FMINL	AI	FMIN Left Input
FMINR	AI	FMIN Right Input
LINEINL	AI	LINEIN Left Single-End Input
LINEINR	AI	LINEIN Right Single-End Input

Signal	Type	Description
HPOUTL	AO	Headphone Light Output
HPOUTR	AO	Headphone Right Output
HPOUTFB	AI	Pseudo Differential Headphone Ground Reference
VRA1	AO	Internal Reference Voltage
VRA2	AO	Internal Reference Voltage
HPVCC	P	Headphone Power
AVCC	P	Analog Power
AGND	G	Analog Ground

8.4.3.2 Clock Sources

The following figure describes the clock source of Audio Codec. For clock setting, configuration, and gating information, refer to section 3.3 "CCU".

Figure 8-25 Audio Codec Clock Diagram



The clock source for the digital part is the PLL_AUDIO0 and PLL_AUDIO1. For the ADC clock, configure **AUDIO_CODEC_ADC_CLK_REG**[25:24] to select the clock source. For the DAC clock, configure

AUDIO_CODEC_DAC_CLK_REG[25:24] to select the clock source. The PK-PK jitter of PLL_AUDIO0 and PLL_AUDIO1 should be less than 200 ps.

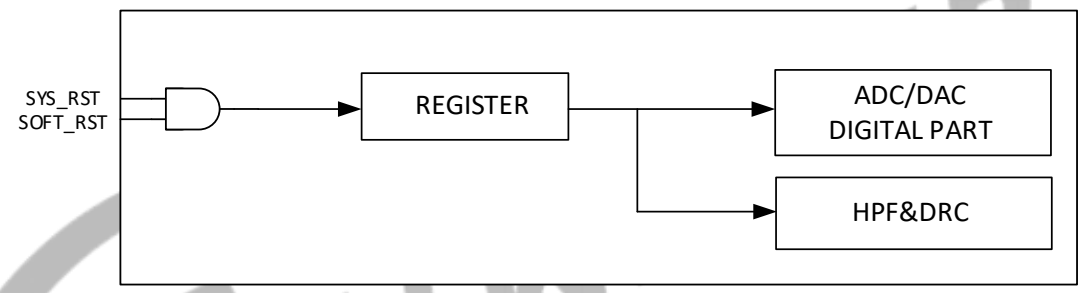
The clock source for the analog part is the CK_ADC and CK_DAC, both of which are divided from the digital part.

8.4.3.3 Reset System

Digital Part Reset System

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

Figure 8-26 Audio Codec Digital Part Reset System

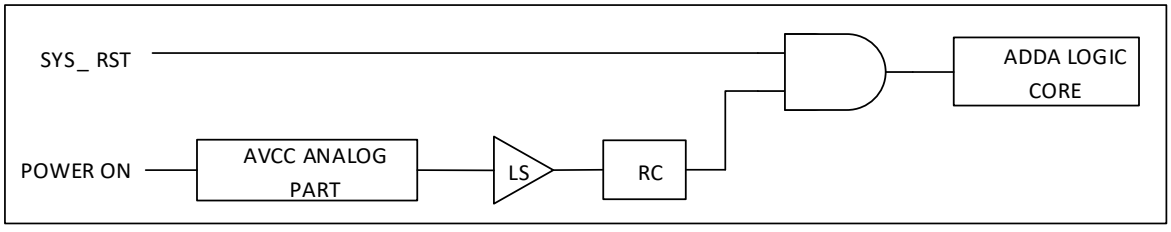


The SYS_RST comes from the VDD-SYS domain and is produced by the RTC domain. Each domain has the de-bounce to confirm the reset system is strong. For the codec register part, MIX can be reset by the SYS_RST when being powered on or the system soft is writing the reset control logic. The other parts can 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 8-27 Audio Codec Analog Part Reset System

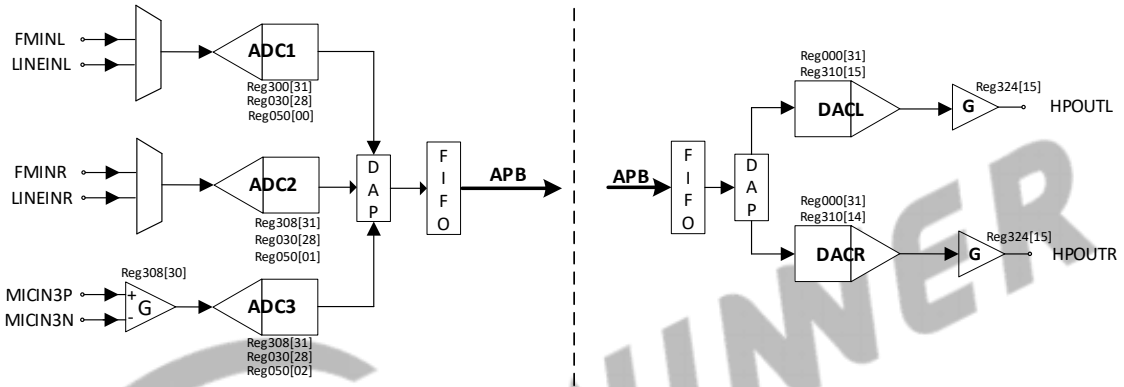


When AVCC is powered on, it sends the AVCC_POR signal. The AVCC_POR signal passes the level shift and RC filter part to the ADDA logic core.

8.4.3.4 Data Path Diagram

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

Figure 8-28 Audio Codec Data Path Diagram



8.4.3.5 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.

8.4.3.6 Stereo DAC

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_REG** register. The digital DAC part can be enabled or disabled by the bit[31] of the **AC_DAC_DPC** register.

8.4.3.7 Analog Audio Input Path

The Audio Codec supports 3 analog audio input paths:

- MICIN3P/N
- LINEINL/R
- FMINL/R

LINEINL, FMINL provide differential input that can be mixed into the ADC1 record mixer. LINEINR, FMINR provide differential input that can be mixed into the ADC2 record mixer. MICIN3P/N provides differential input. 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.

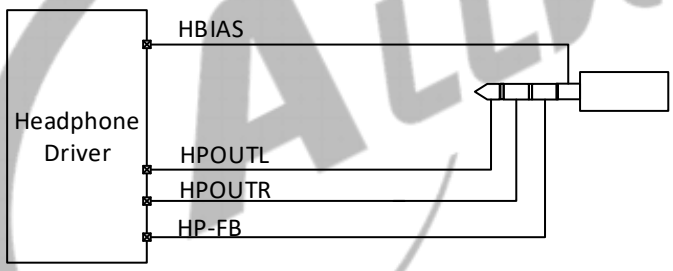
8.4.3.8 Analog Audio Output Path

The Audio Codec has 1 analog output paths:

- HPOUTL/R

The headphone PA is powered up or down by HP_REG[bit15] (HPPA_EN). 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.

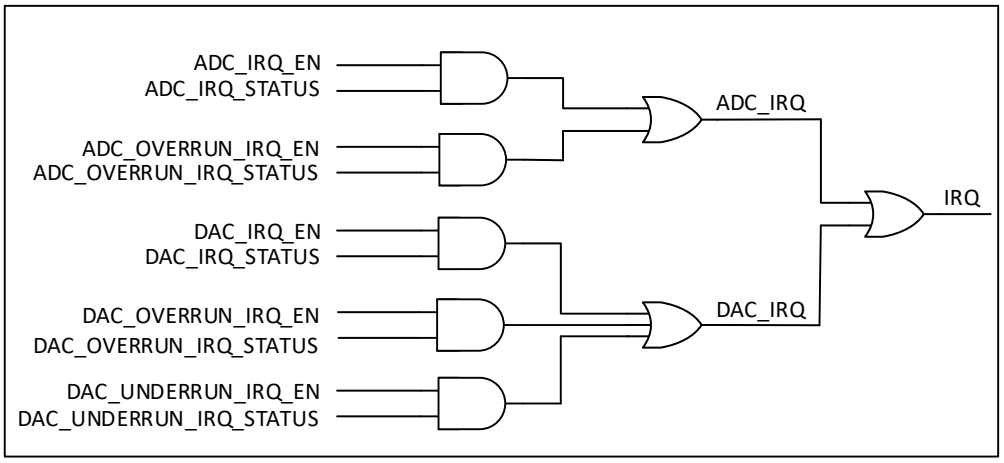
Figure 8-29 Headphone Output Application



8.4.3.9 Interrupts

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

Figure 8-30 Audio Codec Interrupt System

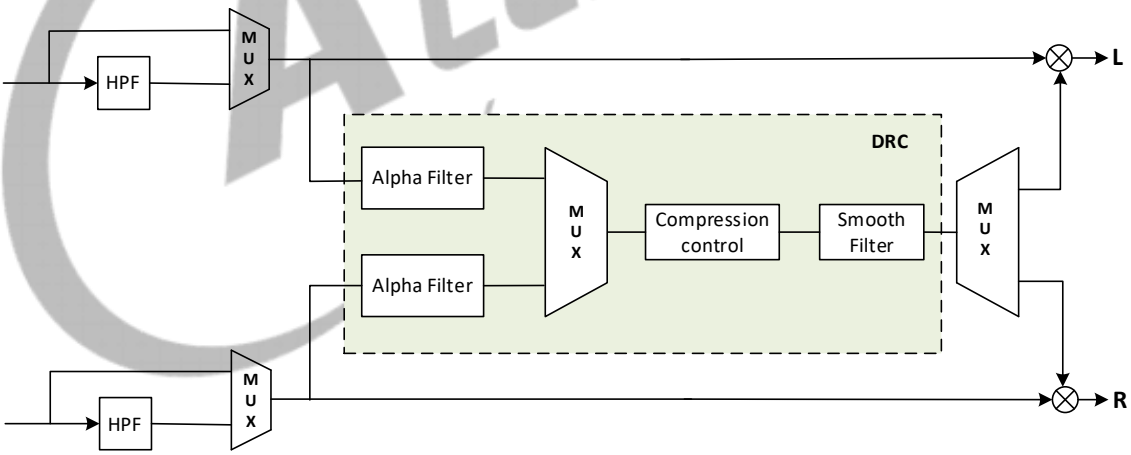


8.4.3.10 Digital Audio Processor (DAP)

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.

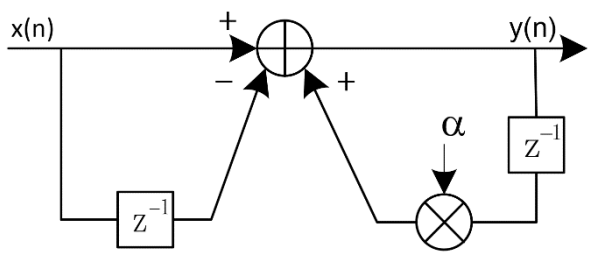
Figure 8-31 DAP Data Flow



HPF Function

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

Figure 8-32 HPF Logic Structure



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 8-33 DRC Block Diagram

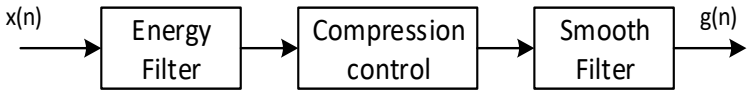
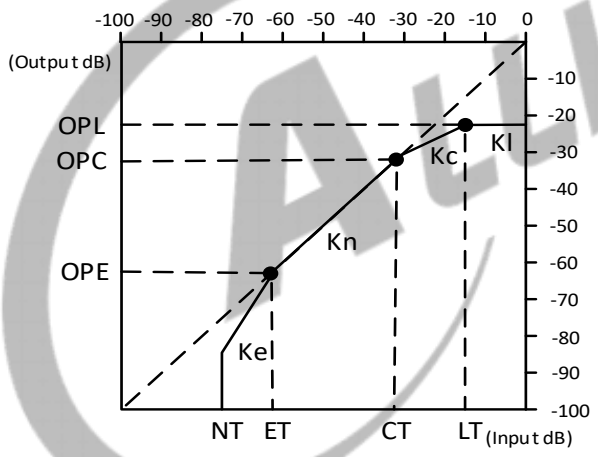


Figure 8-34 DRC Static Curve Parameters



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

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

Each DRC has an adjustable threshold, offset, and 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:

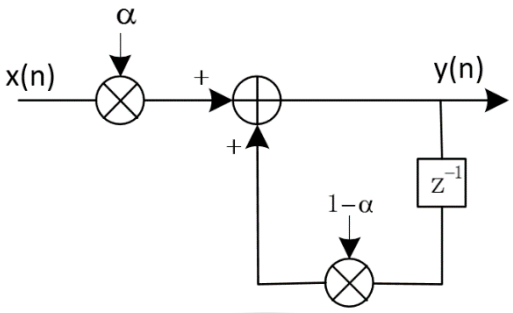
- **Number format**

  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 8-35 Energy Filter Structure



The Energy Filter is to estimate 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 $\alpha = 1 - e^{-2.2Ts/ta}$.

For the Compression Control, there are ten parameters (ET, CT, LT, Ke, Kn, Kc, Kl, OPL, OPC, and 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 RMS of the signal 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 = -40 dB, then the Tin require to set CT to -40 dB is CTin = - (-40 dB)/6.0206 = 6.644, CT_{in} is entered as a 32-bit number in 8.24 format.

  Therefore, CT_{in} = 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 the compression region. For example, an n: 1 compression means that an output increase of 1 dB 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 an integer.

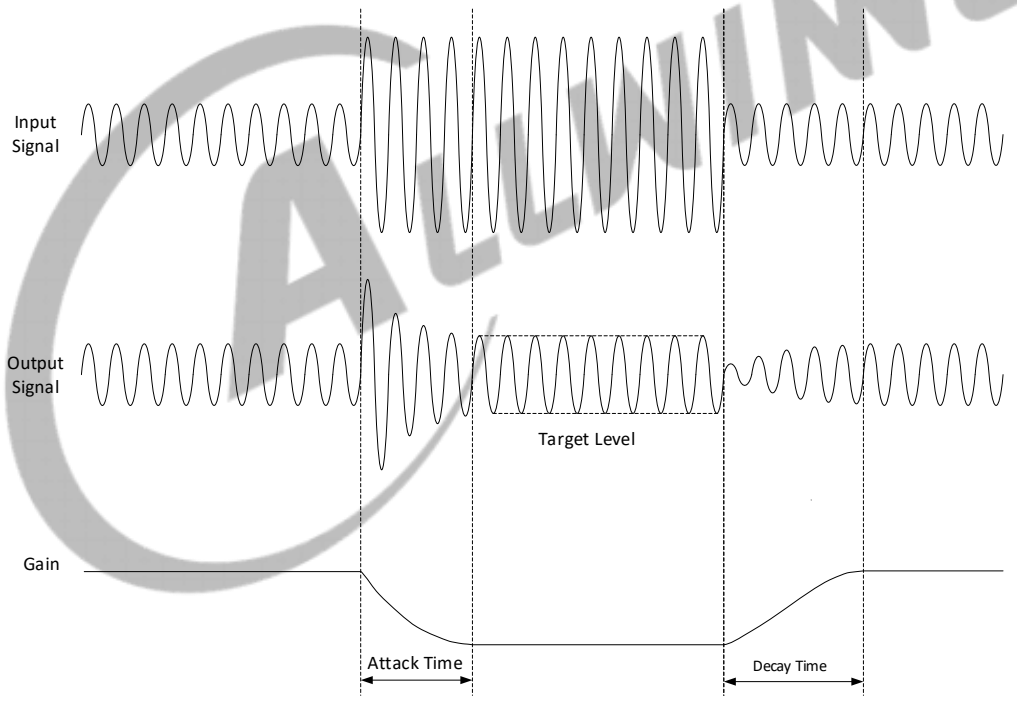
  For example, it is desired to set to 2:1, then the Kc requires to set to 2:1, is Kc = 1/2 = 0.5, Kc is entered as a 32-bit number in 8.24 format.

  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 are shown in Figure 8-36. 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 8-36 Gain Smooth Filter



8.4.4 Programming Guidelines

8.4.4.1 Record Process

In recording mode, the analog audio signals are recorded from the microphones at the specified sample rate, processed by the ADC, and then transferred to the DRAM via the DMA.

Step 1 Codec initialization: configure **AUDIO_CODEC_BGR_REG** to open the audio codec bus clock gating and de-assert bus reset; configure **AUDIO_CODEC_ADC_CLK_REG** and **PLL_AUDIO0_CTRL_REG** to configure PLL_Audio0 frequency and enable PLL_Audio0. For details, refer to section 3.3 "CCU".

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.

8.4.4.2 Playback Process

In playback mode, the audio data are transferred from the DRAM via DMA, processed by the DAC, and finally output via the analog interface.

Step 1 Codec initialization: configure **AUDIO_CODEC_BGR_REG** to open the audio codec bus clock gating and de-assert bus reset; configure **AUDIO_CODEC_DAC_CLK_REG** and **PLL_AUDIO1_CTRL_REG** to configure PLL_Audio1 frequency and enable PLL_Audio1. For details, refer to section 3.3 "CCU".

Step 2 Configure 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.

8.4.5 Register List

Module Name	Base Address
Audio Codec	0x02030000

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

Register Name	Offset	Description
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	VRA1 Speedup 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 Peak Filter Low Attack Time Coef Register
AC_DAC_DRC_LPFHRT	0x011C	DAC DRC Left Peak Filter High Release Time Coef Register
AC_DAC_DRC_LPFLRT	0x0120	DAC DRC Left Peak Filter Low Release Time Coef Register
AC_DAC_DRC_RPFHRT	0x0124	DAC DRC Right Peak filter High Release Time Coef Register
AC_DAC_DRC_RPFLRT	0x0128	DAC DRC Right Peak filter Low Release Time Coef Register
AC_DAC_DRC_LRMSHAT	0x012C	DAC DRC Left RMS Filter High Coef Register
AC_DAC_DRC_LRMSLAT	0x0130	DAC DRC Left RMS Filter Low Coef Register
AC_DAC_DRC_RRMSHAT	0x0134	DAC DRC Right RMS Filter High Coef Register
AC_DAC_DRC_RRMSLAT	0x0138	DAC DRC Right RMS Filter Low Coef Register
AC_DAC_DRC_HCT	0x013C	DAC DRC Compressor Threshold High Setting Register
AC_DAC_DRC_LCT	0x0140	DAC DRC Compressor Slope High Setting Register
AC_DAC_DRC_HKC	0x0144	DAC DRC Compressor Slope High Setting Register
AC_DAC_DRC_LKC	0x0148	DAC DRC Compressor Slope Low Setting Register
AC_DAC_DRC_HOPC	0x014C	DAC DRC Compressor High Output at Compressor Threshold Register

Register Name	Offset	Description
AC_DAC_DRC_LOPC	0x0150	DAC DRC Compressor Low Output at Compressor Threshold Register
AC_DAC_DRC_HLT	0x0154	DAC DRC Limiter Threshold High Setting Register
AC_DAC_DRC_LLT	0x0158	DAC DRC Limiter Threshold Low Setting Register
AC_DAC_DRC_HKl	0x015C	DAC DRC Limiter Slope High Setting Register
AC_DAC_DRC_LKl	0x0160	DAC DRC Limiter Slope Low Setting Register
AC_DAC_DRC_HOPL	0x0164	DAC DRC Limiter High Output at Limiter Threshold
AC_DAC_DRC_LOPL	0x0168	DAC DRC Limiter Low Output at Limiter Threshold
AC_DAC_DRC_HET	0x016C	DAC DRC Expander Threshold High Setting Register
AC_DAC_DRC_LET	0x0170	DAC DRC Expander Threshold Low Setting Register
AC_DAC_DRC_HKE	0x0174	DAC DRC Expander Slope High Setting Register
AC_DAC_DRC_LKE	0x0178	DAC DRC Expander Slope Low Setting Register
AC_DAC_DRC_HOPE	0x017C	DAC DRC Expander High Output at Expander Threshold
AC_DAC_DRC_LOPE	0x0180	DAC DRC Expander Low Output at Expander Threshold
AC_DAC_DRC_HKN	0x0184	DAC DRC Linear Slope High Setting Register
AC_DAC_DRC_LKN	0x0188	DAC DRC Linear Slope Low Setting Register
AC_DAC_DRC_SFHAT	0x018C	DAC DRC Smooth filter Gain High Attack Time Coef Register
AC_DAC_DRC_SFLAT	0x0190	DAC DRC Smooth filter Gain Low Attack Time Coef Register
AC_DAC_DRC_SFHRT	0x0194	DAC DRC Smooth filter Gain High Release Time Coef Register
AC_DAC_DRC_SFLRT	0x0198	DAC DRC Smooth filter Gain Low Release Time Coef Register
AC_DAC_DRC_MXGHS	0x019C	DAC DRC MAX Gain High Setting Register
AC_DAC_DRC_MXGLS	0x01A0	DAC DRC MAX Gain Low Setting Register
AC_DAC_DRC_MNGHS	0x01A4	DAC DRC MIN Gain High Setting Register
AC_DAC_DRC_MNGLS	0x01A8	DAC DRC MIN Gain Low Setting Register
AC_DAC_DRC_EPSHC	0x01AC	DAC DRC Expander Smooth Time High Coef Register
AC_DAC_DRC_EPSLC	0x01B0	DAC DRC Expander Smooth Time Low Coef Register
AC_DAC_DRC_HPFHGAIN	0x01B8	DAC DRC HPF Gain High Coef Register
AC_DAC_DRC_HPFLGAIN	0x01BC	DAC DRC HPF Gain Low Coef Register
AC_ADC_DRC_HHPFC	0x0200	ADC DRC High HPF Coef Register
AC_ADC_DRC_LHPFC	0x0204	ADC DRC Low HPF Coef Register
AC_ADC_DRC_CTRL	0x0208	ADC DRC Control Register

Register Name	Offset	Description
AC_ADC_DRC_LPFHAT	0x020C	ADC DRC Left Peak Filter High Attack Time Coef Register
AC_ADC_DRC_LPFLAT	0x0210	ADC DRC Left Peak Filter Low Attack Time Coef Register
AC_ADC_DRC_RPFHAT	0x0214	ADC DRC Right Peak Filter High Attack Time Coef Register
AC_ADC_DRC_RPFLAT	0x0218	ADC DRC Right Peak Filter Low Attack Time Coef Register
AC_ADC_DRC_LPFHRT	0x021C	ADC DRC Left Peak Filter High Release Time Coef Register
AC_ADC_DRC_LPFLRT	0x0220	ADC DRC Left Peak Filter Low Release Time Coef Register
AC_ADC_DRC_RPFHRT	0x0224	ADC DRC Right Peak Filter High Release Time Coef Register
AC_ADC_DRC_RPFLRT	0x0228	ADC DRC Right Peak Filter Low Release Time Coef Register
AC_ADC_DRC_LRMSHAT	0x022C	ADC DRC Left RMS Filter High Coef Register
AC_ADC_DRC_LRMSLAT	0x0230	ADC DRC Left RMS Filter Low Coef Register
AC_ADC_DRC_RRMSHAT	0x0234	ADC DRC Right RMS Filter High Coef Register
AC_ADC_DRC_RRMSLAT	0x0238	ADC DRC Right RMS Filter Low Coef Register
AC_ADC_DRC_HCT	0x023C	ADC DRC Compressor Threshold High Setting Register
AC_ADC_DRC_LCT	0x0240	ADC DRC Compressor Slope High Setting Register
AC_ADC_DRC_HKC	0x0244	ADC DRC Compressor Slope High Setting Register
AC_ADC_DRC_LKC	0x0248	ADC DRC Compressor Slope Low Setting Register
AC_ADC_DRC_HOPC	0x024C	ADC DRC Compressor High Output at Compressor Threshold Register
AC_ADC_DRC_LOPC	0x0250	ADC DRC Compressor Low Output at Compressor Threshold Register
AC_ADC_DRC_HLT	0x0254	ADC DRC Limiter Threshold High Setting Register
AC_ADC_DRC_LLT	0x0258	ADC DRC Limiter Threshold Low Setting Register
AC_ADC_DRC_HKl	0x025C	ADC DRC Limiter Slope High Setting Register
AC_ADC_DRC_LKl	0x0260	ADC DRC Limiter Slope Low Setting Register
AC_ADC_DRC_HOPL	0x0264	ADC DRC Limiter High Output at Limiter Threshold
AC_ADC_DRC_LOPL	0x0268	ADC DRC Limiter Low Output at Limiter Threshold
AC_ADC_DRC_HET	0x026C	ADC DRC Expander Threshold High Setting Register
AC_ADC_DRC_LET	0x0270	ADC DRC Expander Threshold Low Setting Register
AC_ADC_DRC_HKE	0x0274	ADC DRC Expander Slope High Setting Register

Register Name	Offset	Description
AC_ADC_DRC_LKE	0x0278	ADC DRC Expander Slope Low Setting Register
AC_ADC_DRC_HOPE	0x027C	ADC DRC Expander High Output at Expander Threshold
AC_ADC_DRC_LOPE	0x0280	ADC DRC Expander Low Output at Expander Threshold
AC_ADC_DRC_HKN	0x0284	ADC DRC Linear Slope High Setting Register
AC_ADC_DRC_LKN	0x0288	ADC DRC Linear Slope Low Setting Register
AC_ADC_DRC_SFHAT	0x028C	ADC DRC Smooth filter Gain High Attack Time Coef Register
AC_ADC_DRC_SFLAT	0x0290	ADC DRC Smooth filter Gain Low Attack Time Coef Register
AC_ADC_DRC_SFHRT	0x0294	ADC DRC Smooth filter Gain High Release Time Coef Register
AC_ADC_DRC_SFLRT	0x0298	ADC DRC Smooth filter Gain Low Release Time Coef Register
AC_ADC_DRC_MXGHS	0x029C	ADC DRC MAX Gain High Setting Register
AC_ADC_DRC_MXGLS	0x02A0	ADC DRC MAX Gain Low Setting Register
AC_ADC_DRC_MNGLS	0x02A4	ADC DRC MIN Gain High Setting Register
AC_ADC_DRC_MXGLS	0x02A8	ADC DRC MIN Gain Low Setting Register
AC_ADC_DRC_EPSHC	0x02AC	ADC DRC Expander Smooth Time High Coef Register
AC_ADC_DRC_EPSLC	0x02B0	ADC DRC Expander Smooth Time Low Coef Register
AC_ADC_DRC_HPFHGAIN	0x02B8	ADC DRC HPF Gain High Coef Register
AC_ADC_DRC_HPFLGAIN	0x02BC	ADC DRC HPF Gain Low Coef Register
Analog Domain Register		
ADC1_REG	0x0300	ADC1 Analog Control Register
ADC2_REG	0x0304	ADC2 Analog Control Register
ADC3_REG	0x0308	ADC3 Analog Control Register
DAC_REG	0x0310	DAC Analog Control Register
MICBIAS_REG	0x0318	MICBIAS Analog Control Register
RAMP_REG	0x031C	BIAS Analog Control Register
BIAS_REG	0x0320	BIAS Analog Control Register
ADC5_REG	0x0330	ADC5 Analog Control Register

8.4.6 Register Description

8.4.6.1 0x0000 DAC Digital Part Control Register (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: AC_DAC_DPC
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	EN_DA DAC Digital Part Enable 0: Disabled 1: Enabled
30:29	/	/	/
28:25	R/W	0x0	MODQU Internal DAC Quantization Levels Levels = [7*(21 + MODQU[3:0])]/128 Default levels = 7*21/128 = 1.15
24	R/W	0x0	DWA DWA Function Disable 0: Enabled 1: Disabled
23:19	/	/	/
18	R/W	0x0	HPF_EN High Pass Filter Enable 0: Disabled 1: Enabled
17:12	R/W	0x0	DVOL Digital volume control: DVC, ATT = DVC[5:0]*(-1.16 dB) 64 steps, -1.16 dB/step
11:1	/	/	/
0	R/W	0x0	HUB_EN Audio Hub Enable The bit takes effect only when the EN_DA is set to 1. System Domain: Audio Codec/I2S1/I2S2/OWA TXFIFO Hub Enable. 0: Disabled 1: Enabled

8.4.6.2 0x0004 DAC Volume Control Register (Default Value: 0x0000_A0A0)

Offset: 0x0004			Register Name: DAC_VOL_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	R/W	0x0	DAC_VOL_SEL DAC Volume Control Selection Enable 0: Disabled 1: Enabled
15:8	R/W	0xA0	DAC_VOL_L DAC left channel volume (-119.25 dB to 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ... 0x9F = -0.75 dB 0xA0 = 0 dB 0xA1 = 0.75 dB ... 0xFF = 71.25 dB
7:0	R/W	0xA0	DAC_VOL_R DAC right channel volume (-119.25 dB To 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ... 0x9F = -0.75 dB 0xA0 = 0 dB 0xA1 = 0.75 dB ... 0xFF = 71.25 dB

8.4.6.3 0x0010 DAC FIFO Control Register (Default Value: 0x0000_4000)

Offset: 0x0010			Register Name: AC_DAC_FIFOC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	R/W	0x0	DAC_FS Sample Rate of DAC 000: 48 kHz 010: 24 kHz 100: 12 kHz 110: 192 kHz 001: 32 kHz 011: 16 kHz 101: 8 kHz 111: 96 kHz 44.1 kHz/22.05 kHz/11.025 kHz can be supported by Audio PLL Configure Bit
28	R/W	0x0	FIR_VER FIR Version 0: 64-Tap FIR 1: 32-Tap FIR
27	/	/	/
26	R/W	0x0	SEND_LASAT Audio sample select when TX FIFO underrun 0: Sending zero 1: Sending the last audio sample
25:24	R/W	0x0	FIFO_MODE For 20-bit transmitted audio sample: 00/10: FIFO_I[19:0] = {TXDATA[31:12]} 01/11: FIFO_I[19:0] = {TXDATA[19:0]} For 16-bit transmitted audio sample: 00/10: FIFO_I[19:0] = {TXDATA[31:16], 4'b0} 01/11: FIFO_I[19:0] = {TXDATA[15:0], 4'b0}
23	/	/	/

Offset: 0x0010			Register Name: AC_DAC_FIFOC
Bit	**Read/Write**	**Default/Hex**	**Description**
22:21	R/W	0x0	DAC_DRQ_CLR_CNT When TX FIFO available room is less than or equal N, the DRQ request will be de-asserted. N is defined here: 00: IRQ/DRQ de-asserted when WLEVEL > TXTL 01: 4 10: 8 11: 16
20:15	/	/	/
14:8	R/W	0x40	TX_TRIG_LEVEL TX FIFO Empty Trigger Level (TXTL[12:0]) Interrupt and DMA request trigger level for TX FIFO normal condition. IRQ/DRQ generated when WLEVEL ≤ TXTL **Note: WLEVEL represents the number of valid samples in the TX FIFO. Only TXTL[6:0] valid when TXMODE = 0**
7	/	/	/
6	R/W	0x0	DAC_MONO_EN DAC Mono Enable 0: Stereo, 64 levels FIFO 1: Mono, 128 levels FIFO When enabled, L & R channel send the same data.
5	R/W	0x0	TX_SAMPLE_BITS Transmitting Audio Sample Resolution 0: 16 bits 1: 20 bits
4	R/W	0x0	DAC_DRQ_EN DAC FIFO Empty DRQ Enable 0: Disabled 1: Enabled
3	R/W	0x0	DAC_IRQ_EN DAC FIFO Empty IRQ Enable 0: Disabled 1: Enabled

Offset: 0x0010			Register Name: AC_DAC_FIFOC
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R/W	0x0	FIFO_UNDERRUN_IRQ_EN DAC FIFO Underrun IRQ Enable 0: Disabled 1: Enabled
1	R/W	0x0	FIFO_OVERRUN_IRQ_EN DAC FIFO Overrun IRQ Enable 0: Disabled 1: Enabled
0	R/WC	0x0	FIFO_FLUSH DAC FIFO Flush Write '1' to flush TX FIFO, self clear to '0'

8.4.6.4 0x0014 DAC FIFO Status Register (Default Value: 0x0080_8008)

Offset: 0x0014			Register Name: AC_DAC_FIFOS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23	R	0x1	TX_EMPTY TX FIFO Empty 0: No room for new sample in TX FIFO 1: More than one room for new sample in TX FIFO (>= 1 word)
22:8	R	0x80	TXE_CNT TX FIFO Empty Space Word Counter
7:4	/	/	/
3	R/W1C	0x1	TXE_INT TX FIFO Empty Pending Interrupt 0: No Pending IRQ 1: FIFO Empty Pending Interrupt Write '1' to clear this interrupt or automatically clear if the interrupt condition fails.
2	R/W1C	0x0	TXU_INT TX FIFO Underrun Pending Interrupt 0: No Pending Interrupt 1: FIFO Underrun Pending Interrupt Write '1' to clear this interrupt

Offset: 0x0014			Register Name: AC_DAC_FIFOS
Bit	**Read/Write**	**Default/Hex**	**Description**
1	R/W1C	0x0	TXO_INT TX FIFO Overrun Pending Interrupt 0: No Pending Interrupt 1: FIFO Overrun Pending Interrupt Write '1' to clear this interrupt
0	/	/	/

8.4.6.5 0x0020 DAC TX DATA Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: AC_DAC_TXDATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	W	0x0	TX_DATA Write the transmitting left and right channel sample data to this register one by one. Write the left channel sample data first and then the right channel sample.

8.4.6.6 0x0024 DAC TX Counter Register (Default Value: 0x0000_0000)

Offset: 0x0024			Register Name: AC_DAC_CNT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	TX_CNT TX Sample Counter The audio sample number of sending into TXFIFO. When one sample is put into TXFIFO by DMA or by host IO, the TX sample counter register increases by one. The TX sample counter register can be set to any initial valve at any time. After being updated by the initial value, the counter register should count from this initial value. **Note: It is used for Audio/Video Synchronization.**

8.4.6.7 0x0028 DAC Debug Register (Default Value: 0x0000_0000)

Offset: 0x0028			Register Name: AC_DAC_DG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:12	/	/	/
11	R/W	0x0	DAC_MODU_SELECT DAC Modulator Debug 0: DAC Modulator Normal Mode 1: DAC Modulator Debug Mode
10:9	R/W	0x0	DAC_PATTERN_SELECT DAC Pattern Select 00: Normal (Audio sample from TX FIFO) 01: -6 dB Sin wave 10: -60 dB Sin wave 11: Silent wave
8	R/W	0x0	CODEC_CLK_SELECT CODEC Clock Source Select 0: CODEC clock from PLL 1: CODEC clock from OSC (for Debug)
7	/	/	/
6	R/W	0x0	DA_SWP DAC Output Channel Swap Enable 0: Disabled 1: Enabled
5:3	/	/	/
2:0	R/W	0x0	ADDA_LOOP_MODE ADDA Loop Mode Select 000: Disabled 001: ADDA LOOP MODE DACL/DACR is connected to ADC1/ADC2 010: ADDA LOOP MODE DACL/DACR is connected to ADC3 Others: Reserved

8.4.6.8 0x0030 ADC FIFO Control Register (Default Value: 0x0000_0400)

Offset: 0x0030			Register Name: AC_ADC_FIFOC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:29	R/W	0x0	ADFS Sample Rate of ADC 000: 48 kHz 010: 24 kHz 100: 12 kHz 110: Reserved 001: 32 kHz 011: 16 kHz 101: 8 kHz 111: Reserved 44.1 kHz/22.05 kHz/11.025 kHz can be supported by Audio PLL Configure Bit.
28	R/W	0x0	EN_AD ADC Digital Part Enable 0: Disabled 1: Enabled
27:26	R/W	0x0	ADCFDT ADC FIFO delay time for writing data after EN_AD 00: 5 ms 01: 10 ms 10: 20 ms 11: 30 ms
25	R/W	0x0	ADCDFEN ADC FIFO delay function for writing data after EN_AD 0: Disabled 1: Enabled

Offset: 0x0030			Register Name: AC_ADC_FIFOC
Bit	**Read/Write**	**Default/Hex**	**Description**
24	R/W	0x0	RX_FIFO_MODE RX FIFO Output Mode (Mode 0, 1) 0: Expanding '0' at LSB of TX FIFO register 1: Expanding received sample sign bit at MSB of TX FIFO register For 20-bit received audio sample: Mode 0: RXDATA[31:0] = {FIFO_O[19:0], 12'h0} Mode 1: RXDATA[31:0] = {12{FIFO_O[19]}, FIFO_O[19:0]} For 16-bit received audio sample: Mode 0: RXDATA[31:0] = {FIFO_O[19:4], 16'h0} Mode 1: RXDATA[31:0] = {16{FIFO_O[19]}, FIFO_O[19:4]}
23:22	/	/	/
21	R/W	0x0	RX_SYNC_EN_START The bit takes effect only when RX_SYNC_EN is set to 1. System Domain: Audio codec/I2S1/I2S2/DMIC/OWA RX Synchronize Enable Start. 0: Disabled 1: Enabled
20	R/W	0x0	RX_SYNC_EN Audiocodec RX Synchronize Enable 0: Disabled 1: Enabled
19:17	/	/	/
16	R/W	0x0	RX_SAMPLE_BITS Receiving Audio Sample Resolution 0: 16 bits 1: 20 bits
15:12	/	/	/
11:4	R/W	0x40	RX_FIFO_TRG_LEVEL RX FIFO Trigger Level (RXTL[5:0]) Interrupt and DMA request trigger level for RX FIFO normal condition IRQ/DRQ generated when WLEVEL > RXTL[5:0] **Note: WLEVEL represents the number of valid samples in the RX FIFO.**

Offset: 0x0030			Register Name: AC_ADC_FIFOC
Bit	**Read/Write**	**Default/Hex**	**Description**
3	R/W	0x0	ADC_DRQ_EN ADC FIFO Data Available DRQ Enable 0: Disabled 1: Enabled
2	R/W	0x0	ADC_IRQ_EN ADC FIFO Data Available IRQ Enable 0: Disabled 1: Enabled
1	R/W	0x0	ADC_OVERRUN_IRQ_EN ADC FIFO Overrun IRQ Enable 0: Disabled 1: Enabled
0	R/WC	0x0	ADC_FIFO_FLUSH ADC FIFO Flush Write '1' to flush TX FIFO, self clear to '0'.

8.4.6.9 0x0034 ADC Volume Control1 Register (Default Value: 0xA0A0_A0A0)

Offset: 0x0034			Register Name: ADC_VOL_CTRL1
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	R/W	0xA0	Reserved
23:16	R/W	0xA0	ADC3_VOL ADC3 channel volume (-119.25 dB To 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ... 0xFF: 71.25 dB

Offset: 0x0034			Register Name: ADC_VOL_CTRL1
Bit	Read/Write	Default/Hex	Description
15:8	R/W	0xA0	ADC2_VOL ADC2 channel volume (-119.25 dB To 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ... 0xFF: 71.25 dB
7:0	R/W	0xA0	ADC1_VOL ADC1 channel volume (-119.25 dB To 71.25 dB, 0.75 dB/Step) 0x00: Mute 0x01: -119.25 dB ... 0x9F: -0.75 dB 0xA0: 0 dB 0xA1: 0.75 dB ... 0xFF: 71.25 dB

8.4.6.10 0x0038 ADC FIFO Status Register (Default Value: 0x0000_0001)

Offset: 0x0038			Register Name: AC_ADC_FIFOS
Bit	Read/Write	Default/Hex	Description
31:24	/	/	/
23	R	0x0	RXA RX FIFO Available 0: No available data in RX FIFO 1: More than one sample in RX FIFO (>= 1 word)
22:17	/	/	/
16:8	R	0x0	RXA_CNT RX FIFO Available Sample Word Counter

Offset: 0x0038			Register Name: AC_ADC_FIFOS
Bit	**Read/Write**	**Default/Hex**	**Description**
7:4	/	/	/
3	R/W1C	0x0	RXA_INT RX FIFO Data Available Pending Interrupt 0: No Pending IRQ 1: Data Available Pending IRQ Write '1' to clear this interrupt or automatic clear if the interrupt condition fails.
2	/	/	/
1	R/W1C	0x0	RXO_INT RX FIFO Overrun Pending Interrupt 0: No Pending IRQ 1: FIFO Overrun Pending IRQ Write '1' to clear this interrupt.
0	R	0x1	Reserved

8.4.6.11 0x0040 ADC RX DATA Register (Default Value: 0x0000_0000)

Offset: 0x0040			Register Name: AC_ADC_RXDATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	RX_DATA RX Sample The host can get one sample by reading this register. The left channel sample data comes first and then the right channel sample.

8.4.6.12 0x0044 ADC RX Counter Register (Default Value: 0x0000_0000)

Offset: 0x0044			Register Name: AC_ADC_CNT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	RX_CNT RX Sample Counter The audio sample number of writing into RXFIFO. When one sample is written by Digital Audio Engine, the RX sample counter register increases by one. The RX sample counter register can be set to any initial valve at any time. After being updated by the initial value, the counter register should count from this initial value. **Note: It is used for Audio/Video Synchronization.**

8.4.6.13 0x004C ADC Debug Register (Default Value: 0x0000_0000)

Offset: 0x004C			Register Name: AC_ADC_DG_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:26	/	/	/
25	R/W	0x0	AD_SWP2 ADC output channel swap enable (for digital filter) 0: Disabled 1: Enabled **Note: ADC3 and ADC4 swap data.**
24	R/W	0x0	AD_SWP1 ADC output channel swap enable (for digital filter) 0: Disabled 1: Enabled **Note: ADC1 and ADC2 swap data.**
23:0	/	/	/

8.4.6.14 0x0050 ADC Digtial Control Register (Default Value: 0x0000_0000)

Offset: 0x0050			Register Name: ADC_DIG_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/

Offset: 0x0050			Register Name: ADC_DIG_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
17	R/W	0x0	ADC3_VOL_EN ADC3 Volume Control Enable 0: Disabled 1: Enabled
16	R/W	0x0	ADC1_2_VOL_EN ADC1/2 Volume Control Enable 0: Disabled 1: Enabled
15:3	/	/	/
2:0	R/W	0x0	ADC_CHANNEL_EN Bit 3: ADC4 enabled Bit 2: ADC3 enabled Bit 1: ADC2 enabled Bit 0: ADC1 enabled

8.4.6.15 0x0054 VRA1 Speedup Down Control Register (Default Value: 0x0000_0000)

Offset: 0x0054			Register Name: VRA1SPEEDUP_DOWN_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:5	/	/	/
4	R	0x0	VRA1SPEEDUP_DOWN_STATE Only if VAR1SPEEDUP_DOWN_Further_CTRL (0x310[22]) is set 0, VAR1Speedup Down State is valid. 0: VAR1Speedup_Down does not work. 1: VAR1Speedup_Down works.
3:2	/	/	/
1	R/W	0x0	VRA1SPEEDUP_DOWN_CTRL VAR1Speedup Down Manual Control Enable 0: Disabled. VAR1Speedup Down converts to 1 after the bus rst releases 32 ms. 1: Enabled. VAR1Speedup Down converts to 1 immediately.

Offset: 0x0054			Register Name: VRA1SPEEDUP_DOWN_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	VRA1SPEEDUP_DOWN_RST_CTRL VAR1Speedup Down RST Manual Control Enable 0: Disabled. VAR1Speedup Down converts to 1 after the bus rst releases 32 ms. 1: Enabled. VAR1Speedup Down reset 0 immediately.

8.4.6.16 0x00F0 DAC DAP Control Register (Default Value: 0x0000_0000)

Offset: 0x00F0			Register Name: AC_DAC_DAP_CTR
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	DDAP_EN DAP for DRC enable 0: Bypassed 1: Enabled
30	/	/	/
29	R/W	0x0	DDAP_DRC_EN DRC enable control 0: Disabled 1: Enabled
28	R/W	0x0	DDAP_HPF_EN HPF enable control 0: Disabled 1: Enabled
27:0	/	/	/

8.4.6.17 0x00F8 ADC DAP Control Register (Default Value: 0x0000_0000)

Offset: 0x00F8			Register Name: AC_ADC_DAP_CTR
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	ADC_DAP0_EN (control the DAP of ADC1/2) DAP for ADC enable 0: Bypassed 1: Enabled
30	/	/	/

Offset: 0x00F8			Register Name: AC_ADC_DAP_CTR
Bit	Read/Write	Default/Hex	Description
29	R/W	0x0	ADC_DRC0_EN ADC DRC0 enable control 0: Disabled 1: Enabled
28	R/W	0x0	ADC_HPF0_EN ADC HPF0 enable control 0: Disabled 1: Enabled
27	R/W	0x0	ADC_DAP1_EN (control the DAP of ADC3) ADC DAP1 enable control
26	/	/	/
25	R/W	0x0	ADC_DRC1_EN ADC DRC1 enable control 0: Disabled 1: Enabled
24	R/W	0x0	ADC_HPF1_EN ADC HPF1 enable control 0: Disabled 1: Enabled
23:0	/	/	/

8.4.6.18 0x0100 DAC DRC High HPF Coef Register (Default Value: 0x0000_00FF)

Offset: 0x0100			Register Name: AC_DAC_DRC_HHPFC
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0xFF	HPF coefficient setting and the data is 3.24 format.

8.4.6.19 0x0104 DAC DRC Low HPF Coef Register (Default Value: 0x0000_FAC1)

Offset: 0x0104			Register Name: AC_DAC_DRC_LHPFC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x0104			Register Name: AC_DAC_DRC_LHPFC
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0xFAC1	HPF coefficient setting and the data is 3.24 format.

8.4.6.20 0x0108 DAC DRC Control Register (Default Value: 0x0000_0080)

Offset: 0x0108			Register Name: AC_DAC_DRC_CTRL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15	R	0x0	DRC delay buffer data output state when The DRC delay function is enabled and the DRC function is disabled. After disabling the DRC function and this bit goes to 0, write the DRC delay function bit to 0. 0: Not completed 1: Completed
14:10	/	/	/
13:8	R/W	0x0	Signal delay time setting 6'h00: (8x1) fs 6'h01: (8x2) fs 6'h02: (8x3) fs ---------- 6'h2e: (8*47) fs 6'h2f: (8*48) fs 6'h30 -- 6'h3f: (8*48) fs Delay time = 8*(n + 1) fs, n<6'h30; When the delay function is disabled, the signal delay time is unused.
7	R/W	0x1	DAC_DRC_DELAY_BUF_EN The delay buffer use or not when the DRC is disabled and the DRC buffer data output completely. 0: Do not use the buffer. 1: Use the buffer.
6	R/W	0x0	DAC_DRC_GAIN_MAX_LIMIT_EN DRC gain max limit enable 0: Disabled 1: Enabled

Offset: 0x0108			Register Name: AC_DAC_DRC_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
5	R/W	0x0	DAC_DRC_GAIN_MIN_LIMIT_EN DRC gain min limit enable When this function is enabled, it will overwrite the noise detect function. 0: Disabled 1: Enabled
4	R/W	0x0	DAC_DRC_DETECT_NOISE_EN Control the DRC to detect noise when ET is enabled. 0: Disabled 1: Enabled
3	R/W	0x0	DAC_DRC_SIGNAL_FUNC_SEL Signal function select 0: RMS filter 1: Peak filter When the signal function selects the Peak filter, the RMS parameter is unused. (AC_DRC_LRMSHAT, AC_DRC_LRMSLAT, AC_DRC_LRMSHAT, AC_DRC_LRMSLAT) When the signal function selects the RMS filter, the Peak filter parameter is unused. (AC_DRC_LPFHAT, AC_DRC_LPFLAT, AC_DRC_RPFHAT, AC_DRC_RPFLAT, AC_DRC_LPFHRT, AC_DRC_LPFLRT, AC_DRC_RPFHRT, and AC_DRC_RPFLRT)
2	R/W	0x0	DAC_DRC_DELAY_FUNC_EN Delay function enable 0: Disabled 1: Enabled When the bit is disabled, the signal delay time is unused.
1	R/W	0x0	DAC_DRC_LT_EN DRC LT enable 0: Disabled 1: Enabled When the bit is disabled, Kl and OPL parameter is unused.
0	R/W	0x0	DAC_DRC_ET_EN DRC ET enable 0: Disabled 1: Enabled When the bit is disabled, Ke and OPE parameter is unused.

8.4.6.21 0x010C DAC DRC Left Peak Filter High Attack Time Coef Register (Default Value: 0x0000_000B)

Offset: 0x010C			Register Name: AC_DAC_DRC_LPFHAT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x000B	DAC_DRC_LPFHAT The left peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/ta). The format is 3.24. (The default value is 1 ms)

8.4.6.22 0x0110 DAC DRC Left Peak Filter Low Attack Time Coef Register (Default Value: 0x0000_77BF)

Offset: 0x0110			Register Name: AC_DAC_DRC_LPFLAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x77BF	DAC_DRC_LPFLAT The left peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/ta). The format is 3.24. (The default value is 1 ms)

8.4.6.23 0x0114 DAC DRC Right Peak Filter High Attack Time Coef Register (Default Value: 0x0000_000B)

Offset: 0x0114			Register Name: AC_DAC_DRC_RPFHAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xB	DAC_DRC_RPFHAT The right peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/ta). The format is 3.24. (The default value is 1 ms)

8.4.6.24 0x0118 DAC DRC Left Peak Filter Low Attack Time Coef Register (Default Value: 0x0000_77BF)

Offset: 0x0118			Register Name: AC_DAC_DRC_RPFLAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x0118			Register Name: AC_DAC_DRC_RPFLAT
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x77BF	DAC_DRC_RPFLAT The left peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/ta). The format is 3.24. (The default value is 1 ms)

8.4.6.25 0x011C DAC DRC Left Peak Filter High Release Time Coef Register (Default Value: 0x0000_00FF)

Offset: 0x011C			Register Name: AC_DAC_DRC_LPFHRT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x00FF	DAC_DRC_LPFHRT The left peak filter release time parameter setting, which is determined by the equation that RT = exp (-2.2Ts/tr). The format is 3.24. (The default value is 100 ms)

8.4.6.26 0x0120 DAC DRC Left Peak Filter Low Release Time Coef Register (Default Value: 0x0000_E1F8)

Offset: 0x0120			Register Name: AC_DAC_DRC_LPFLRT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xE1F8	DAC_DRC_LPFLRT The left peak filter release time parameter setting, which is determined by the equation that RT = exp (-2.2Ts/tr). The format is 3.24. (The default value is 100 ms)

8.4.6.27 0x0124 DAC DRC Right Peak Filter High Release Time Coef Register (Default Value: 0x0000_00FF)

Offset: 0x0124			Register Name: AC_DAC_DRC_RPFHRT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x0124			Register Name: AC_DAC_DRC_RPFHRT
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0xFF	DAC_DRC_RPFHRT The right peak filter release time parameter setting, which is determined by the equation that RT = exp (-2.2Ts/tr). The format is 3.24. (The default value is 100 ms)

8.4.6.28 0x0128 DAC DRC Right Peak filter Low Release Time Coef Register(Default Value: 0x0000_E1F8)

Offset: 0x0128			Register Name: AC_DAC_DRC_RPFLRT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xE1F8	DAC_DRC_RPFLRT The right peak filter release time parameter setting, which is determined by the equation that AT = exp(-2.2Ts/tr). The format is 3.24. (The default value is 100 ms)

8.4.6.29 0x012C DAC DRC Left RMS Filter High Coef Register (Default Value: 0x0000_0001)

Offset: 0x012C			Register Name: AC_DAC_DRC_LRMSHAT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x0001	DAC_DRC_LRMSHAT The left RMS filter average time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/tav). The format is 3.24. (The default value is 10 ms)

8.4.6.30 0x0130 DAC DRC Left RMS Filter Low Coef Register (Default Value: 0x0000_2BAF)

Offset: 0x0130			Register Name: AC_DAC_DRC_LRMSLAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x0130			Register Name: AC_DAC_DRC_LRMSLAT
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x2BAF	DAC_DRC_LRMSLAT The left RMS filter average time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/tav). The format is 3.24. (The default value is 10 ms)

8.4.6.31 0x0134 DAC DRC Right RMS Filter High Coef Register (Default Value: 0x0000_0001)

Offset: 0x0134			Register Name: AC_DAC_DRC_RRMSHAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x2BAF	DAC_DRC_RRMSHAT The right RMS filter average time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/tav). The format is 3.24. (The default value is 10 ms)

8.4.6.32 0x0138 DAC DRC Right RMS Filter Low Coef Register (Default Value: 0x0000_2BAF)

Offset: 0x0138			Register Name: AC_DAC_DRC_RRMSLAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x2BAF	DAC_DRC_RRMSLAT The right RMS filter average time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/tav). The format is 3.24. (The default value is 10 ms)

8.4.6.33 0x013C DAC DRC Compressor Theshold High Setting Register (Default Value: 0x0000_06A4)

Offset: 0x013C			Register Name: AC_DAC_DRC_HCT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x013C			Register Name: AC_DAC_DRC_HCT
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x06A4	DAC_DRC_HCT The compressor threshold setting, which is set by the equation that CTin = -CT/6.0206. The format is 8.24. (The default value is -40 dB)

8.4.6.34 0x0140 DAC DRC Compressor Slope High Setting Register (Default Value: 0x0000_D3C0)

Offset: 0x0140			Register Name: AC_DAC_DRC_LCT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xD3C0	DAC_DRC_LCT The compressor threshold setting, which is set by the equation that CTin = -CT/6.0206. The format is 8.24. (The default value is -40 dB)

8.4.6.35 0x0144 DAC DRC Compressor Slope High Setting Register (Default Value: 0x0000_0080)

Offset: 0x0144			Register Name: AC_DAC_DRC_HKC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0080	DAC_DRC_HKC The slope of the compressor, which is determined by the equation that Kc = 1/R. R is the ratio of the compressor, which is always an integer. The format is 8.24. (The default value is 2:1)

8.4.6.36 0x0148 DAC DRC Compressor Slope Low Setting Register (Default Value: 0x0000_0000)

Offset: 0x0148			Register Name: AC_DAC_DRC_LKC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x0148			Register Name: AC_DAC_DRC_LKC
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x0000	DAC_DRC_LKC The slope of the compressor, which is determined by the equation that Kc = 1/R. R is the ratio of the compressor, which is always an integer. The format is 8.24. (The default value is 2:1)

8.4.6.37 0x014C DAC DRC Compressor High Output at Compressor Threshold Register (Default Value: 0x0000_F95B)

Offset: 0x014C			Register Name: AC_DAC_DRC_HOPC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xF95B	DAC_DRC_HOPC The output of the compressor, which is determined by the equation -OPC/6.0206. The format is 8.24 (The default value is -40 dB)

8.4.6.38 0x0150 DAC DRC Compressor Low Output at Compressor Threshold Register (Default Value: 0x0000_2C3F)

Offset: 0x0150			Register Name: AC_DAC_DRC_LOPC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x2C3F	DAC_DRC_LOPC The output of the compressor, which is determined by the equation OPC/6.0206. The format is 8.24. (The default value is -40 dB)

8.4.6.39 0x0154 DAC DRC Limiter Theshold High Setting Register (Default Value: 0x0000_01A9)

Offset: 0x0154			Register Name: AC_DAC_DRC_HLT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x01A9	DAC_DRC_HLT The limiter threshold setting, which is set by the equation that LTin = -LT/6.0206, The format is 8.24. (The default value is -10 dB)

8.4.6.40 0x0158 DAC DRC Limiter Theshold Low Setting Register (Default Value: 0x0000_34F0)

Offset: 0x0158			Register Name: AC_DAC_DRC_LLT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x34F0	DAC_DRC_LLT The limiter threshold setting, which is set by the equation that LTin = -LT/6.0206. The format is 8.24. (The default value is -10 dB)

8.4.6.41 0x015C DAC DRC Limiter Slope High Setting Register (Default Value: 0x0000_0005)

Offset: 0x015C			Register Name: AC_DAC_DRC_HKl
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x0005	DAC_DRC_HKl The slope of the limiter which is determined by the equation that Kl = 1/R. R is the ratio of the limiter, which is always an integer. The format is 8.24. (The default value is <50:1>)

8.4.6.42 0x0160 DAC DRC Limiter Slope Low Setting Register (Default Value: 0x0000_1EB8)

Offset: 0x0160			Register Name: AC_DAC_DRC_LKl
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x1EB8	DAC_DRC_LKl The slope of the limiter, which is determined by the equation that Kl = 1/R. R is the ratio of the limiter, which is always an integer. The format is 8.24. (The default value is <50:1>)

8.4.6.43 0x0164 DAC DRC Limiter High Output at Limiter Threshold Register (Default Value: 0x0000_FBD8)

Offset: 0x0164			Register Name: AC_DAC_DRC_HOPL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0xFBD8	DAC_DRC_HOPL The output of the limiter, which is determined by equation OPT/6.0206. The format is 8.24. (The default value is -25 dB)

8.4.6.44 0x0168 DAC DRC Limiter Low Output at Limiter Threshold Register (Default Value: 0x0000_FBA7)

Offset: 0x0168			Register Name: AC_DAC_DRC_LOPL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0xFBA7	DAC_DRC_LOPL The output of the limiter, which is determined by equation OPT/6.0206. The format is 8.24. (The default value is -25 dB)

8.4.6.45 0x016C DAC DRC Expander Theshold High Setting Register (Default Value: 0x0000_0BA0)

Offset: 0x016C			Register Name: AC_DAC_DRC_HET
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x0BA0	DAC_DRC_HET The expander threshold setting, which is set by the equation that ETin = -ET/6.0206. The format is 8.24. (The default value is -70 dB)

8.4.6.46 0x0170 DAC DRC Expander Theshold Low Setting Register (Default Value: 0x0000_7291)

Offset: 0x0170			Register Name: AC_DAC_DRC_LET
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x7291	DAC_DRC_LET The expander threshold setting, which is set by the equation that ETin = -ET/6.0206. The format is 8.24. (The default value is -70 dB)

8.4.6.47 0x0174 DAC DRC Expander Slope High Setting Register (Default Value: 0x0000_0500)

Offset: 0x0174			Register Name: AC_DAC_DRC_HKE
Bit	**Read/Write**	**Default/Hex**	**Description**
31:14	/	/	/
13:0	R/W	0x0500	DAC_DRC_HKE The slope of the expander, which is determined by the equation that Ke = 1/R. R is the ratio of the expander, which is always an integer and the ke must larger than 50. The format is 8.24. (The default value is <1:5>)

8.4.6.48 0x0178 DAC DRC Expander Slope Low Setting Register (Default Value: 0x0000_0000)

Offset: 0x0178			Register Name: AC_DAC_DRC_LKE
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x0000	DAC_DRC_LKE The slope of the expander, which is determined by the equation that Ke = 1/R. R is the ratio of the expander, which is always an integer and the ke must larger than 50. The format is 8.24. (The default value is <1:5>)

8.4.6.49 0x017C DAC DRC Expander High Output at Expander Threshold Register (Default Value: 0x0000_F45F)

Offset: 0x017C			Register Name: AC_DAC_DRC_HOPE
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0xF45F	DAC_DRC_HOPE The output of the expander, which is determined by equation OPE/6.0206. The format is 8.24. (The default value is -70 dB)

8.4.6.50 0x0180 DAC DRC Expander Low Output at Expander Threshold Register (Default Value: 0x0000_8D6E)

Offset: 0x0180			Register Name: AC_DAC_DRC_LOPE
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x8D6E	DAC_DRC_LOPE The output of the expander which is determined by equation OPE/6.0206. The format is 8.24. (The default value is -70 dB)

8.4.6.51 0x0184 DAC DRC Linear Slope High Setting Register (Default Value: 0x0000_0100)

Offset: 0x0184			Register Name: AC_DAC_DRC_HKN
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0100	DAC_DRC_HKN The slope of the linear, which is determined by the equation that Kn = 1/R. R is the ratio of the linear, which is always an integer. The format is 8.24. (The default value is <1:1>)

8.4.6.52 0x0188 DAC DRC Linear Slope Low Setting Register (Default Value: 0x0000_0000)

Offset: 0x0188			Register Name: AC_DAC_DRC_LKN
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0000	DAC_DRC_LKN The slope of the linear, which is determined by the equation that Kn = 1/R. R is the ratio of the linear, which is always an integer. The format is 8.24. (The default value is <1:1>)

8.4.6.53 0x018C DAC DRC Smooth Filter Gain High Attack Time Coef Register (Default Value: 0x0000_0002)

Offset: 0x018C			Register Name: AC_DAC_DRC_SFHAT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/

Offset: 0x018C			Register Name: AC_DAC_DRC_SFHAT
Bit	**Read/Write**	**Default/Hex**	**Description**
10:0	R/W	0x0002	DAC_DRC_SFHAT The smooth filter attack time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/tr). The format is 3.24. (The default value is 5 ms)

8.4.6.54 0x0190 DAC DRC Smooth Filter Gain Low Attack Time Coef Register (Default Value: 0x0000_5600)

Offset: 0x0190			Register Name: AC_DAC_DRC_SFLAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x5600	DAC_DRC_SFLAT The smooth filter attack time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/tr). The format is 3.24. (The default value is 5 ms)

8.4.6.55 0x0194 DAC DRC Smooth Filter Gain High Release Time Coef Register (Default Value: 0x0000_0000)

Offset: 0x0194			Register Name: AC_DAC_DRC_SFHRT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R/W	0x0000	DAC_DRC_SFHRT The gain smooth filter release time parameter setting, which is determined by the equation that RT = 1-exp (-2.2Ts/tr). The format is 3.24. (The default value is 200 ms)

8.4.6.56 0x0198 DAC DRC Smooth Filter Gain Low Release Time Coef Register (Default Value: 0x0000_0F04)

Offset: 0x0198			Register Name: AC_DAC_DRC_SFLRT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/

Offset: 0x0198			Register Name: AC_DAC_DRC_SFLRT
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x0F04	DAC_DRC_SFLRT The gain smooth filter release time parameter setting, which is determined by the equation that RT = 1-exp (-2.2Ts/tr). The format is 3.24. (The default value is 200 ms)

8.4.6.57 0x019C DAC DRC MAX Gain High Setting Register (Default Value: 0x0000_FE56)

Offset: 0x019C			Register Name: AC_DAC_DRC_MXGHS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xFE56	DAC_DRC_MXGHS The max gain setting, which is determined by equation MXG_{in}=MXG/6.0206. The format is 8.24 and must -20 dB < MXG < 30 dB (The default value is -10 dB)

8.4.6.58 0x01A0 DAC DRC MAX Gain Low Setting Register (Default Value: 0x0000_CB0F)

Offset: 0x01A0			Register Name: AC_DAC_DRC_MXGLS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xCB0F	DAC_DRC_MXGLS The max gain setting, which is determined by equation MXG_{in}=MXG/6.0206. The format is 8.24 and must -20 dB < MXG < 30 dB (The default value is -10 dB)

8.4.6.59 0x01A4 DAC DRC MIN Gain High Setting Register (Default Value: 0x0000_F95B)

Offset: 0x01A4			Register Name: AC_DAC_DRC_MNGHS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x01A4			Register Name: AC_DAC_DRC_MNGHS
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0xF95B	DAC_DRC_MNGHS The min gain setting, which is determined by equation MXG_{in}=MXG/6.0206. The format is 8.24 and must -60 dB ≤ MNG ≤ -40 dB (The default value is -40 dB)

8.4.6.60 0x01A8 DAC DRC MIN Gain Low Setting Register (Default Value: 0x0000_2C3F)

Offset: 0x01A8			Register Name: AC_DAC_DRC_MNGLS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x2C3F	DAC_DRC_MNGLS The min gain setting, which is determined by equation MXG_{in}=MNG/6.0206. The format is 8.24 and must -60 dB ≤ MNG ≤ -40 dB (The default value is -40 dB)

8.4.6.61 0x01AC DAC DRC Expander Smooth Time High Coef Register (Default Value: 0x0000_0000)

Offset: 0x01AC			Register Name: AC_DAC_DRC_EPSHC
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x0000	DAC_DRC_EPSHC The gain smooth filter release and attack time parameter setting in expander region, which are determined by the equation that RT = 1-exp (-2.2Ts/tr). The format is 3.24. (The default value is 30 ms)

8.4.6.62 0x01B0 DAC DRC Expander Smooth Time Low Coef Register (Default Value: 0x0000_640C)

Offset: 0x01B0			Register Name: AC_DAC_DRC_EPSLC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x01B0			Register Name: AC_DAC_DRC_EPSLC
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x640C	DAC_DRC_EPSLC The gain smooth filter release and attack time parameter setting in expander region, which are determined by the equation that RT = 1-exp (-2.2Ts/tr). The format is 3.24. (The default value is 30 ms)

8.4.6.63 0x01B8 DAC DRC HPF Gain High Coef Register (Default Value: 0x0000_0100)

Offset: 0x01B8			Register Name: AC_DAC_DRC_HPFHGAIN
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x100	DAC_DRC_HPFHGAIN The gain of HPF coefficient. The format is 3.24. (gain = 1)

8.4.6.64 0x01BC DAC DRC HPF Gain Low Coef Register (Default Value: 0x0000_0000)

Offset: 0x01BC			Register Name: AC_DAC_DRC_HPFLGAIN
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0000	DAC_DRC_HPFLGAIN The gain of HPF coefficient. The format is 3.24. (gain = 1)

8.4.6.65 0x0200 ADC DRC High HPF Coef Register (Default Value: 0x0000_00FF)

Offset: 0x0200			Register Name: AC_ADC_DRC_HHPFC
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0xFF	ADC_DRC_HHPFC HPF coefficient setting and the data is 3.24 format.

8.4.6.66 0x0204 ADC DRC Low HPF Coef Register (Default Value: 0x0000_FAC1)

Offset: 0x0204			Register Name: AC_ADC_DRC_LHPFC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xFAC1	HPF coefficient setting and the data is 3.24 format.

8.4.6.67 0x0208 ADC DRC Control Register (Default Value: 0x0000_0080)

Offset: 0x0208			Register Name: AC_ADC_DRC_CTRL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15	R	0x0	ADC_DRC_DELAY_BUF_OUTPUT_STATE DRC delay buffer data output state when The DRC delay function is enabled and the DRC function is disabled. After disabled DRC function and this bit goes to 0, the user should write the DRC delay function bit to 0. 0: Not completed 1: Completed
14:10	/	/	/
13:8	R/W	0x0	ADC_DRC_SIGNAL_DELAY_TIME_SET Signal delay time setting 6'h00: (8x1) fs 6'h01: (8x2) fs 6'h02: (8x3) fs ---------- 6'h2e: (8*47) fs 6'h2f: (8*48) fs 6'h30 -- 6'h3f: (8*48) fs Delay time = 8*(n + 1) fs, n < 6'h30; When the delay function is disabled, the signal delay time is unused.
7	R/W	0x1	ADC_DRC_DELAY_BUF_EN The delay buffer use or not when the DRC is disabled and the DRC buffer data output completely. 0: Do not use the buffer 1: Use the buffer

Offset: 0x0208			Register Name: AC_ADC_DRC_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
6	R/W	0x0	ADC_DRC_GAIN_MAX_LIMIT_EN DRC gain max limit enable 0: Disabled 1: Enabled
5	R/W	0x0	ADC_DRC_GAIN_MIN_LIMIT_EN DRC gain min limit enable When this fuction is enabled, it will overwrite the noise detect function. 0: Disabled 1: Enabled
4	R/W	0x0	ADC_DRC_DETECT_NOISE_EN Control the DRC to detect noise when ET is enabled 0: Disabled 1: Enabled
3	R/W	0x0	ADC_DRC_SIGNAL_FUNC_SEL Signal function select 0: RMS filter 1: Peak filter When the signal function selects the Peak filter, the RMS parameter is unused. (AC_DRC_LRMSHAT, AC_DRC_LRMSLAT, AC_DRC_LRMSHAT, and AC_DRC_LRMSLAT) When the signal function selects the RMS filter, the Peak filter parameter is unused. (AC_DRC_LPFHAT, AC_DRC_LPFLAT, AC_DRC_RPFHAT, AC_DRC_RPFLAT, AC_DRC_LPFHRT, AC_DRC_LPFLRT, AC_DRC_RPFHRT, and AC_DRC_RPFLRT)
2	R/W	0x0	ADC_DRC_DELAY_FUNC_EN Delay function enable 0: Disabled 1: Enabled When the bit is disabled, the signal delay time is unused.
1	R/W	0x0	ADC_DRC_LT_EN DRC LT enable 0: Disabled 1: Enabled When the bit is disabled, Kl and OPL parameter is unused.

Offset: 0x0208			Register Name: AC_ADC_DRC_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	ADC_DRC_ET_EN DRC ET enable 0: Disabled 1: Enabled When the bit is disabled, Ke and OPE parameter is unused.

8.4.6.68 0x020C ADC DRC Left Peak Filter High Attack Time Coef Register (Default Value: 0x0000_000B)

Offset: 0x020C			Register Name: AC_ADC_DRC_LPFHAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R/W	0x000B	ADC_DRC_LPFHAT The left peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/ta). The format is 3.24. (The default value is 1 ms)

8.4.6.69 0x0210 ADC DRC Left Peak Filter Low Attack Time Coef Register (Default Value: 0x0000_77BF)

Offset: 0x0210			Register Name: AC_ADC_DRC_LPFLAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x77BF	ADC_DRC_LPFLAT The left peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/ta). The format is 3.24. (The default value is 1 ms)

8.4.6.70 0x0214 ADC DRC Right Peak Filter High Attack Time Coef Register (Default Value: 0x0000_000B)

Offset: 0x0214			Register Name: AC_ADC_DRC_RPFHAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/

Offset: 0x0214			Register Name: AC_ADC_DRC_RPFHAT
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x000B	ADC_DRC_RPFHAT The right peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/ta). The format is 3.24. (The default value is 1 ms)

8.4.6.71 0x0218 ADC DRC Right Peak Filter Low Attack Time Coef Register (Default Value: 0x0000_77BF)

Offset: 0x0218			Register Name: AC_ADC_DRC_RPFLAT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x77BF	ADC_DRC_RPFLAT The right peak filter attack time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/ta). The format is 3.24. (The default value is 1 ms)

8.4.6.72 0x021C ADC DRC Left Peak Filter High Release Time Coef Register (Default Value: 0x0000_00FF)

Offset: 0x021C			Register Name: AC_ADC_DRC_LPFHRT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x00FF	ADC_DRC_LPFHRT The left peak filter release time parameter setting, which is determined by the equation that RT = exp (-2.2Ts/tr). The format is 3.24. (The default value is 100 ms)

8.4.6.73 0x0220 ADC DRC Left Peak Filter Low Release Time Coef Register (Default Value: 0x0000_E1F8)

Offset: 0x0220			Register Name: AC_ADC_DRC_LPFLRT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x0220			Register Name: AC_ADC_DRC_LPFLRT
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0xE1F8	ADC_DRC_LPFLRT The left peak filter release time parameter setting, which is determined by the equation that RT = exp (-2.2Ts/tr). The format is 3.24. (The default value is 100 ms)

8.4.6.74 0x0224 ADC DRC Right Peak Filter High Release Time Coef Register (Default Value: 0x0000_00FF)

Offset: 0x0224			Register Name: AC_ADC_DRC_RPFHRT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x00FF	ADC_DRC_RPFHRT The right peak filter release time parameter setting, which is determined by the equation that RT = exp (-2.2Ts/tr). The format is 3.24. (The default value is 100 ms)

8.4.6.75 0x0228 ADC DRC Right Peak Filter Low Release Time Coef Register (Default Value: 0x0000_E1F8)

Offset: 0x0228			Register Name: AC_ADC_DRC_RPFLRT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xE1F8	ADC_DRC_RPFLRT The right peak filter release time parameter setting, which is determined by the equation that RT = exp (-2.2Ts/tr). The format is 3.24. (The default value is 100 ms)

8.4.6.76 0x022C ADC DRC Left RMS Filter High Coef Register (Default Value: 0x0000_0001)

Offset: 0x022C			Register Name: AC_ADC_DRC_LRMSHAT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/

Offset: 0x022C			Register Name: AC_ADC_DRC_LRMSHAT
Bit	**Read/Write**	**Default/Hex**	**Description**
10:0	R/W	0x0001	ADC_DRC_LRMSHAT The left RMS filter average time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/tav). The format is 3.24. (The default value is 10 ms)

8.4.6.77 0x0230 ADC DRC Left RMS Filter Low Coef Register (Default Value: 0x0000_2BAF)

Offset: 0x0230			Register Name: AC_ADC_DRC_LRMSLAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x2BAF	ADC_DRC_LRMSLAT The left RMS filter average time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/tav). The format is 3.24. (The default value is 10 ms)

8.4.6.78 0x0234 ADC DRC Right RMS Filter High Coef Register (Default Value: 0x0000_0001)

Offset: 0x0234			Register Name: AC_ADC_DRC_RRMSHAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R/W	0x0001	ADC_DRC_RRMSHAT The right RMS filter average time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/tav). The format is 3.24. (The default value is 10 ms)

8.4.6.79 0x0238 ADC DRC Right RMS Filter Low Coef Register (Default Value: 0x0000_2BAF)

Offset: 0x0238			Register Name: AC_ADC_DRC_RRMSLAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/

Offset: 0x0238			Register Name: AC_ADC_DRC_RRMSLAT
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x2BAF	ADC_DRC_RRMSLAT The right RMS filter average time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/tav). The format is 3.24. (The default value is 10 ms)

8.4.6.80 0x023C ADC DRC Compressor Theshold High Setting Register (Default Value: 0x0000_06A4)

Offset: 0x023C			Register Name: AC_ADC_DRC_HCT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x06A4	ADC_DRC_HCT The compressor threshold setting, which is set by the equation that CTin = -CT/6.0206. The format is 8.24. (The default value is -40 dB)

8.4.6.81 0x0240 ADC DRC Compressor Slope High Setting Register (Default Value: 0x0000_D3C0)

Offset: 0x0240			Register Name: AC_ADC_DRC_LCT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xD3C0	ADC_DRC_LCT The compressor threshold setting, which is set by the equation that CTin = -CT/6.0206. The format is 8.24. (The default value is -40 dB)

8.4.6.82 0x0244 ADC DRC Compressor Slope High Setting Register (Default Value: 0x0000_0080)

Offset: 0x0244			Register Name: AC_ADC_DRC_HKC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x0244			Register Name: AC_ADC_DRC_HKC
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x0080	ADC_DRC_HKC The slope of the compressor which is determined by the equation that Kc = 1/R. R is the ratio of the compressor, which is always an integer. The format is 8.24. (The default value is <2:1>)

8.4.6.83 0x0248 ADC DRC Compressor Slope Low Setting Register (Default Value: 0x0000_0000)

Offset: 0x0248			Register Name: AC_ADC_DRC_LKC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0000	ADC_DRC_LKC The slope of the compressor, which is determined by the equation that Kc = 1/R. R is the ratio of the compressor, which is always an integer. The format is 8.24. (The default value is <2:1>)

8.4.6.84 0x024C ADC DRC Compressor High Output at Compressor Threshold Register (Default Value: 0x0000_F95B)

Offset: 0x024C			Register Name: AC_ADC_DRC_HOPC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xF95B	ADC_DRC_HOPC The output of the compressor, which is determined by the equation -OPC/6.0206. The format is 8.24. (The default value is -40 dB)

8.4.6.85 0x0250 ADC DRC Compressor Low Output at Compressor Threshold Register (Default Value: 0x0000_2C3F)

Offset: 0x0250			Register Name: AC_ADC_DRC_LOPC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x0250			Register Name: AC_ADC_DRC_LOPC
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x2C3F	ADC_DRC_LOPC The output of the compressor, which is determined by the equation OPC/6.0206. The format is 8.24. (The default value is -40 dB)

8.4.6.86 0x0254 ADC DRC Limiter Theshold High Setting Register (Default Value: 0x0000_01A9)

Offset: 0x0254			Register Name: AC_ADC_DRC_HLT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x01A9	ADC_DRC_HLT The limiter threshold setting, which is set by the equation that LTin = -LT/6.0206. The format is 8.24. (The default value is -10 dB)

8.4.6.87 0x0258 ADC DRC Limiter Theshold Low Setting Register (Default Value: 0x0000_34F0)

Offset: 0x0258			Register Name: AC_ADC_DRC_LLT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x34F0	ADC_DRC_LLT The limiter threshold setting, which is set by the equation that LTin = -LT/6.0206. The format is 8.24. (The default value is -10 dB)

8.4.6.88 0x025C ADC DRC Limiter Slope High Setting Register (Default Value: 0x0000_0005)

Offset: 0x025C			Register Name: AC_ADC_DRC_HKl
Bit	Read/Write	Default/Hex	Description
31:14	/	/	/
13:0	R/W	0x0005	ADC_DRC_HKl The slope of the limiter, which is determined by the equation that Kl = 1/R. R is the ratio of the limiter, which is always an integer. The format is 8.24. (The default value is <50:1>)

8.4.6.89 0x0260 ADC DRC Limiter Slope Low Setting Register (Default Value: 0x0000_1EB8)

Offset: 0x0260			Register Name: AC_ADC_DRC_LKl
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x1EB8	ADC_DRC_LKl The slope of the limiter, which is determined by the equation that Kl = 1/R. R is the ratio of the limiter, which is always an integer. The format is 8.24. (The default value is <50:1>)

8.4.6.90 0x0264 ADC DRC Limiter High Output at Limiter Threshold Register (Default Value: 0x0000_FBD8)

Offset: 0x0264			Register Name: AC_ADC_DRC_HOPL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xFBD8	ADC_DRC_HOPL The output of the limiter, which is determined by equation OPT/6.0206. The format is 8.24. (The default value is -25 dB)

8.4.6.91 0x0268 ADC DRC Limiter Low Output at Limiter Threshold Register (Default Value: 0x0000_FBA7)

Offset: 0x0268			Register Name: AC_ADC_DRC_LOPL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xFBA7	ADC_DRC_LOPL The output of the limiter which is determined by equation OPT/6.0206. The format is 8.24. (The default value is -25 dB)

8.4.6.92 0x026C ADC DRC Expander Theshold High Setting Register (Default Value: 0x0000_0BA0)

Offset: 0x026C			Register Name: AC_ADC_DRC_HET
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x026C			Register Name: AC_ADC_DRC_HET
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x0BA0	ADC_DRC_HET The expander threshold setting, which is set by the equation that ETin = -ET/6.0206, The format is 8.24. (The default value is -70 dB)

8.4.6.93 0x0270 ADC DRC Expander Theshold Low Setting Register (Default Value: 0x0000_7291)

Offset: 0x0270			Register Name: AC_ADC_DRC_LET
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x7291	ADC_DRC_LET The expander threshold setting, which is set by the equation that ETin = -ET/6.0206, The format is 8.24. (The default value is -70 dB)

8.4.6.94 0x0274 ADC DRC Expander Slope High Setting Register (Default Value: 0x0000_0500)

Offset: 0x0274			Register Name: AC_ADC_DRC_HKE
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0500	ADC_DRC_HKE The slope of the expander, which is determined by the equation that Ke = 1/R. R is the ratio of the expander, which is always an integer and the ke must larger than 50. The format is 8.24. (The default value is <1:5>)

8.4.6.95 0x0278 ADC DRC Expander Slope Low Setting Register (Default Value: 0x0000_0000)

Offset: 0x0278			Register Name: AC_ADC_DRC_LKE
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x0278			Register Name: AC_ADC_DRC_LKE
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0x0000	ADC_DRC_LKE The slope of the expander, which is determined by the equation that Ke = 1/R. R is the ratio of the expander, which is always an integer and the ke must be larger than 50. The format is 8.24. (The default value is <1:5>)

8.4.6.96 0x027C ADC DRC Expander High Output at Expander Threshold Register (Default Value: 0x0000_F45F)

Offset: 0x027C			Register Name: AC_ADC_DRC_HOPE
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xF45F	ADC_DRC_HOPE The output of the expander, which is determined by equation OPE/6.0206. The format is 8.24. (The default value is -70 dB)

8.4.6.97 0x0280 ADC DRC Expander Low Output at Expander Threshold Register (Default Value: 0x0000_8D6E)

Offset: 0x0280			Register Name: AC_ADC_DRC_LOPE
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x8D6E	ADC_DRC_LOPE The output of the expander, which is determined by equation OPE/6.0206. The format is 8.24. (The default value is -70 dB)

8.4.6.98 0x0284 ADC DRC Linear Slope High Setting Register (Default Value: 0x0000_0100)

Offset: 0x0284			Register Name: AC_ADC_DRC_HKN
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0100	ADC_DRC_HKN The slope of the linear, which is determined by the equation that Kn = 1/R. R is the ratio of the linear, which is always an integer. The format is 8.24. (The default value is <1:1>)

8.4.6.99 0x0288 ADC DRC Linear Slope Low Setting Register (Default Value: 0x0000_0000)

Offset: 0x0288			Register Name: AC_ADC_DRC_LKN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x0000	ADC_DRC_LKN The slope of the linear, which is determined by the equation that Kn = 1/R. R is the ratio of the linear, which is always an integer. The format is 8.24. (The default value is <1:1>)

8.4.6.100 0x028C ADC DRC Smooth Filter Gain High Attack Time Coef Register (Default Value: 0x0000_0002)

Offset: 0x028C			Register Name: AC_ADC_DRC_SFHAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:11	/	/	/
10:0	R/W	0x0002	ADC_DRC_SFHAT The smooth filter attack time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/tr). The format is 3.24. (The default value is 5 ms)

8.4.6.101 0x0290 ADC DRC Smooth Filter Gain Low Attack Time Coef Register (Default Value: 0x0000_5600)

Offset: 0x0290			Register Name: AC_ADC_DRC_SFLAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x5600	ADC_DRC_SFLAT The smooth filter attack time parameter setting, which is determined by the equation that AT = 1-exp (-2.2Ts/tr). The format is 3.24. (The default value is 5 ms)

8.4.6.102 0x0294 ADC DRC Smooth Filter Gain High Release Time Coef Register (Default Value: 0x0000_0000)

Offset: 0x0294			Register Name: AC_ADC_DRC_SFHRT
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x0000	ADC_DRC_SFHRT The gain smooth filter release time parameter setting, which is determined by the equation that RT = 1-exp (-2.2Ts/tr). The format is 3.24. (The default value is 200 ms)

8.4.6.103 0x0298 ADC DRC Smooth Filter Gain Low Release Time Coef Register (Default Value: 0x0000_0F04)

Offset: 0x0298			Register Name: AC_ADC_DRC_SFLRT
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0F04	ADC_DRC_SFLRT The gain smooth filter release time parameter setting, which is determined by the equation that RT = 1-exp (-2.2Ts/tr). The format is 3.24. (The default value is 200 ms)

8.4.6.104 0x029C ADC DRC MAX Gain High Setting Register (Default Value: 0x0000_FE56)

Offset: 0x029C			Register Name: AC_ADC_DRC_MXGHS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xFE56	ADC_DRC_MXGHS The max gain setting, which is determined by equation MXG_{in}=MXG/6.0206. The format is 8.24 and must -20 dB < MXG < 30 dB (The default value is -10 dB)

8.4.6.105 0x02A0 ADC DRC MAX Gain Low Setting Register (Default Value: 0x0000_CB0F)

Offset: 0x02A0			Register Name: AC_ADC_DRC_MXGLS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/

Offset: 0x02A0			Register Name: AC_ADC_DRC_MXGLS
Bit	Read/Write	Default/Hex	Description
15:0	R/W	0xCB0F	ADC_DRC_MXGLS The max gain setting, which is determined by equation MXG_{in}=MXG/6.0206. The format is 8.24 and must -20 dB < MXG < 30 dB (The default value is -10 dB)

8.4.6.106 0x02A4 ADC DRC MIN Gain High Setting Register (Default Value: 0x0000_F95B)

Offset: 0x02A4			Register Name: AC_ADC_DRC_MNGHS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0xF95B	ADC_DRC_MNGHS The min gain setting, which is determined by equation MXG_{in}=MXG/6.0206. The format is 8.24 and must -60 dB ≤ MNG ≤ -40 dB (The default value is -40 dB)

8.4.6.107 0x02A8 ADC DRC MIN Gain Low Setting Register (Default Value: 0x0000_2C3F)

Offset: 0x02A8			Register Name: AC_ADC_DRC_MNGLS
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x2C3F	ADC_DRC_MNGLS The min gain setting, which is determined by equation MXG_{in}=MNG/6.0206. The format is 8.24 and must -60 dB ≤ MNG ≤ -40 dB (The default value is -40 dB)

8.4.6.108 0x02AC ADC DAP Expander Smooth Time High Coef Register (Default Value: 0x0000_0000)

Offset: 0x02AC			Register Name: AC_ADC_DRC_EPSHC
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/

Offset: 0x02AC			Register Name: AC_ADC_DRC_EPSHC
Bit	Read/Write	Default/Hex	Description
10:0	R/W	0x0000	ADC_DRC_EPSHC The gain smooth filter release and attack time parameter setting in expander region, which is determined by the equation that RT = 1-exp (-2.2Ts/tr). The format is 3.24. (The default value is 30 ms)

8.4.6.109 0x02B0 ADC DRC Expander Smooth Time Low Coef Register (Default Value: 0x0000_640C)

Offset: 0x02B0			Register Name: AC_ADC_DRC_EPSLC
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x640C	ADC_DRC_EPSLC The gain smooth filter release and attack time parameter setting in expander region, which is determined by the equation that RT = 1-exp (-2.2Ts/tr). The format is 3.24. (The default value is 30 ms)

8.4.6.110 0x02B8 ADC DRC HPF Gain High Coef Register (Default Value: 0x0000_0100)

Offset: 0x02B8			Register Name: AC_ADC_DRC_HPFHGAIN
Bit	Read/Write	Default/Hex	Description
31:11	/	/	/
10:0	R/W	0x100	ADC_DRC_HPFHGAIN The gain of HPF coefficient setting, which format is 3.24. (gain = 1)

8.4.6.111 0x02BC ADC DRC HPF Gain Low Coef Register (Default Value: 0x0000_0000)

Offset: 0x02BC			Register Name: AC_ADC_DRC_HPFLGAIN
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0000	ADC_DRC_HPFLGAIN The gain of HPF coefficient setting, which format is 3.24. (gain = 1)

8.4.6.112 0x0300 ADC1 Analog Control Register (Default Value: 0x001C_C055)

Offset: 0x0300			Register Name: ADC1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	ADC1_EN ADC1 Channel Enable 0: Disabled 1: Enabled
30	R/W	0x0	Reserved
29	R/W	0x0	ADC1 Dither Control 0: New Dither Off 1: New Dither On
28	R/W	0x0	Reserved
27	R/W	0x0	FMINLEN FMINL Enable 0: Disable 1: Enable
26	R/W	0x0	FMINLG FMINL Gain Control 0: 0 dB 1: 6 dB
25:24	R/W	0x0	DSM_DITHER_LVL Dither Level Control (Dither level is positive ralated to the ctrl bits) 00: No Level 01: Min Level 10: Middle Level 11: Max Level
23	R/W	0x0	LINEINLEN LINEINL Enable 0: Disable 1: Enable
22	R/W	0x0	LINEINLG LINEINL Gain Control 0: 0 dB 1: 6 dB

Offset: 0x0300			Register Name: ADC1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
21:20	R/W	0x1	IOPBUFFER PGA Vcm Buffer OP Bias Current Select 00: 6 uA 01: 7 uA 10: 8 uA 11: 9 uA
19:18	R/W	0x3	ADC1_PGA_CTRL_RCM ADC1 PGA Common Mode Input Impedance Control for MICIN 00: 100 kΩ 01: 75 kΩ 10: 50 kΩ 11: 25 kΩ
17:16	R/W	0x0	ADC1_PGA_IN_VCM_CTRL ADC1 PGA Common-Mode Voltage Control 00: 900 mV 01: 800 mV 10: 750 mV 11: 700 mV
15:14	R/W	0x3	IOPADC ADC1-ADC3 Bias Current Select 00: 1 uA 01: 2 uA 10: 3 uA 11: 4 uA
13	/	/	/

Offset: 0x0300			Register Name: ADC1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
12:8	R/W	0x0	ADC1_PGA_GAIN_CTRL ADC1 PGA gain settings: 0x0: 0 dB 0x10: 21 dB 0x1: 6 dB 0x11: 22 dB 0x2: 6 dB 0x12: 23 dB 0x3: 6 dB 0x13: 24 dB 0x4: 9 dB 0x14: 25 dB 0x5: 10 dB 0x15: 26 dB 0x6: 11 dB 0x16: 27 dB 0x7: 12 dB 0x17: 28 dB 0x8: 13 dB 0x18: 29 dB 0x9: 14 dB 0x19: 30 dB 0xA: 15 dB 0x1A: 31 dB 0xB: 16 dB 0x1B: 32 dB 0xC: 17 dB 0x1C: 33 dB 0xD: 18 dB 0x1D: 34 dB 0xE: 19 dB 0x1E: 35 dB 0xF: 20 dB 0x1F: 36 dB
7:6	R/W	0x1	ADC1_IOPAAF ADC1 OP AAF Bias Current Select 00: 1.50*IOPADC 01: 1.75*IOPADC 10: 2.00*IOPADC 11: 2.25*IOPADC IOPADC is defined by ADC1_REG<15:14> from 1 uA to 4 uA. For example: ADC1_REG<15:14> = 11, IOPADC = 4 uA 00: 1.50*4 uA = 6 uA 01: 1.75*4 uA = 7 uA 10: 2.00*4 uA = 8 uA 11: 2.25*4 uA = 9 uA

Offset: 0x0300			Register Name: ADC1_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
5:4	R/W	0x1	ADC1_IOPSDM1 ADC1 OP SDM Bias Current Select 1 00: 1.50*IOPADC 01: 1.75*IOPADC 10: 2.00*IOPADC 11: 2.25*IOPADC IOPADC is defined by ADC1_REG<15:14> from 1 uA to 4 uA.
3:2	R/W	0x1	ADC1_IOPSDM2 ADC1 OP SDM Bias Current Select 2 00: 1.50*IOPADC 01: 1.75*IOPADC 10: 2.00*IOPADC 11: 2.25*IOPADC IOPADC is defined by ADC1_REG<15:14> from 1 uA to 4 uA.
1:0	R/W	0x1	ADC1_IOPMIC ADC1 OP MIC Bias Current Select 00: 1.50*IOPADC 01: 1.75*IOPADC 10: 2.00*IOPADC 11: 2.25*IOPADC IOPADC is defined by ADC1_REG<15:14> from 1 uA to 4 uA.

8.4.6.113 0x0304 ADC2 Analog Control Register (Default Value: 0x001C_0055)

Offset: 0x0304			Register Name: ADC2_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	ADC2_EN ADC2 Channel Enable 0: Disabled 1: Enabled
30	R/W	0x0	Reserved
29	R/W	0x0	ADC2 Dither Control 0: New Dither Off 1: New Dither On
28	R/W	0x0	Reserved

Offset: 0x0304			Register Name: ADC2_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
27	R/W	0x0	FMINREN FMINR Enable 0: Disable 1: Enable
26	R/W	0x0	FMINRG FMINR Gain Control 0: 0 dB 1: 6 dB
25:24	R/W	0x0	DSM_DITHER_LVL Dither Level Control (Dither level is positive ralated to the ctrl bits) 00: No Level 01: Min Level 10: Middle Level 11: Max Level
23	R/W	0x0	LINEINREN LINEINR Enable 0: Disable 1: Enable
22	R/W	0x0	LINEINRG LINEINR Gain Control 0: 0 dB 1: 6 dB
21:20	R/W	0x1	IOPBUFFER PGA Vcm Buffer OP Bias Current Select 00: 6 uA 01: 7 uA 10: 8 uA 11: 9 uA
19:18	R/W	0x3	ADC2_PGA_CTRL_RCM ADC2 PGA Common Mode Input Impedance Control for MICIN 00: 100 kΩ 01: 75 kΩ 10: 50 kΩ 11: 25 kΩ

Offset: 0x0304			Register Name: ADC2_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
17:16	R/W	0x0	ADC2_PGA_IN_VCM_CTRL ADC2 PGA Common-Mode Voltage Control 00: 900 mV 01: 800 mV 10: 750 mV 11: 700 mV
15:13	/	/	/
12:8	R/W	0x0	ADC2_PGA_GAIN_CTRL ADC2 PGA Gain Settings 0x0: 0 dB 0x10: 21 dB 0x1: 6 dB 0x11: 22 dB 0x2: 6 dB 0x12: 23 dB 0x3: 6 dB 0x13: 24 dB 0x4: 9 dB 0x14: 25 dB 0x5: 10 dB 0x15: 26 dB 0x6: 11 dB 0x16: 27 dB 0x7: 12 dB 0x17: 28 dB 0x8: 13 dB 0x18: 29 dB 0x9: 14 dB 0x19: 30 dB 0xA: 15 dB 0x1A: 31 dB 0xB: 16 dB 0x1B: 32 dB 0xC: 17 dB 0x1C: 33 dB 0xD: 18 dB 0x1D: 34 dB 0xE: 19 dB 0x1E: 35 dB 0xF: 20 dB 0x1F: 36 dB
7:6	R/W	0x1	ADC2_IOPAAF ADC2 OP AAF Bias Current Select 00: 1.50*IOPADC 01: 1.75*IOPADC 10: 2.00*IOPADC 11: 2.25*IOPADC IOPADC is defined by ADC1_REG<15:14> from 1 uA to 4 uA.

Offset: 0x0304			Register Name: ADC2_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
5:4	R/W	0x1	ADC2_IOPSDM1 ADC2 OP SDM Bias Current Select 1 00: 1.50*IOPADC 01: 1.75*IOPADC 10: 2.00*IOPADC 11: 2.25*IOPADC IOPADC is defined by ADC1_REG<15:14> from 1 uA to 4 uA.
3:2	R/W	0x1	ADC2_IOPSDM2 ADC2 OP SDM Bias Current Select 2 00: 1.50*IOPADC 01: 1.75*IOPADC 10: 2.00*IOPADC 11: 2.25*IOPADC IOPADC is defined by ADC1_REG<15:14> from 1 uA to 4 uA.
1:0	R/W	0x1	ADC2_IOPMIC ADC2 OP MIC Bias Current Select 00: 1.50*IOPADC 01: 1.75*IOPADC 10: 2.00*IOPADC 11: 2.25*IOPADC IOPADC is defined by ADC1_REG<15:14> from 1 uA to 4 uA.

8.4.6.114 0x0308 ADC3 Analog Control Register (Default Value: 0x001C_0055)

Offset: 0x0308			Register Name: ADC3_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	ADC3_EN ADC3 Channel Enable 0: Disabled 1: Enabled
30	R/W	0x0	MIC3_PGA_EN MIC3 PGA Enable 0: Disabled 1: Enabled

Offset: 0x0308			Register Name: ADC3_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
29	R/W	0x0	ADC3 Dither Control 0: New Dither Off 1: New Dither On
28	R/W	0x0	MIC3_SIN_EN MIC3 Single Input Enable 0: Disabled 1: Enabled
27:26	/	/	/
25:24	R/W	0x0	DSM_DITHER_LVL Dither Level Control (Dither level is positive ralated to the ctrl bits) 00: No Level 01: Min Level 10: Middle Level 11: Max Level
23:22	/	/	/
21:20	R/W	0x1	IOPBUFFER PGA Vcm Buffer OP Bias Current Select 00: 6 uA 01: 7 uA 10: 8 uA 11: 9 uA
19:18	R/W	0x3	ADC3_PGA_CTRL_RCM ADC3 PGA Common Mode Input Impedance Control for MICIN 00: 100 kΩ 01: 75 kΩ 10: 50 kΩ 11: 25 kΩ
17:16	R/W	0x0	ADC3_PGA_IN_VCM_CTRL ADC3 PGA Common-Mode Voltage Control 00: 900 mV 01: 800 mV 10: 750 mV 11: 700 mV
15:13	/	/	/

Offset: 0x0308			Register Name: ADC3_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
12:8	R/W	0x0	ADC3_PGA_GAIN_CTRL ADC3 PGA Gain Settings 0x0: 0 dB 0x10: 21 dB 0x1: 6 dB 0x11: 22 dB 0x2: 6 dB 0x12: 23 dB 0x3: 6 dB 0x13: 24 dB 0x4: 9 dB 0x14: 25 dB 0x5: 10 dB 0x15: 26 dB 0x6: 11 dB 0x16: 27 dB 0x7: 12 dB 0x17: 28 dB 0x8: 13 dB 0x18: 29 dB 0x9: 14 dB 0x19: 30 dB 0xA: 15 dB 0x1A: 31 dB 0xB: 16 dB 0x1B: 32 dB 0xC: 17 dB 0x1C: 33 dB 0xD: 18 dB 0x1D: 34 dB 0xE: 19 dB 0x1E: 35 dB 0xF: 20 dB 0x1F: 36 dB
7:6	R/W	0x1	ADC3_IOPAAF ADC3 OP AAF Bias Current Select 00: 1.50*IOPADC 01: 1.75*IOPADC 10: 2.00*IOPADC 11: 2.25*IOPADC IOPADC is defined by ADC1_REG<15:14> from 1 uA to 4 uA.
5:4	R/W	0x1	ADC3_IOPSDM1 ADC3 OP SDM Bias Current Select 1 00: 1.50*IOPADC 01: 1.75*IOPADC 10: 2.00*IOPADC 11: 2.25*IOPADC IOPADC is defined by ADC1_REG<15:14> from 1 uA to 4 uA.

Offset: 0x0308			Register Name: ADC3_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
3:2	R/W	0x1	ADC3_IOPSDM2 ADC3 OP SDM Bias Current Select 2 00: 1.50*IOPADC 01: 1.75*IOPADC 10: 2.00*IOPADC 11: 2.25*IOPADC IOPADC is defined by ADC1_REG<15:14> from 1 uA to 4 uA.
1:0	R/W	0x1	ADC3_IOPMIC ADC3 OP MIC Bias Current Select 00: 1.50*IOPADC 01: 1.75*IOPADC 10: 2.00*IOPADC 11: 2.25*IOPADC IOPADC is defined by ADC1_REG<15:14> from 1 uA to 4 uA.

8.4.6.115 0x0310 DAC Analog Control Register (Default Value: 0x0015_0000)

Offset: 0x0310			Register Name: DAC_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23	R/W	0x0	CURRENT_TEST_SELECT Internal Current Sink Test Enable (from MICIN3P pin) 0: Normal 1: For Debug
22	/	/	/
21:20	R/W	0x1	IOPVRS VRA2 Buffer OP and Headphone Feedback Buffer OP Bias Current Select 00: 6 uA 01: 7 uA 10: 8 uA 11: 9 uA
19:18	R/W	0x1	Reserved

Offset: 0x0310			Register Name: DAC_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
17:16	R/W	0x1	IOPDACS OPDACL/R Bias Current Select 00: 6 uA 01: 7 uA 10: 8 uA 11: 9 uA
15	R/W	0x0	DACL_EN DACL Enable 0: Disabled 1: Enabled
14	R/W	0x0	DACR_EN DACR Enable 0: Disabled 1: Enabled
13	R/W	0x0	Reserved
12	R/W	0x0	Reserved
11	R/W	0x0	Reserved
10	R/W	0x0	Reserved
9:7	/	/	/
6	R/W	0x0	Reserved
5	R/W	0x0	Reserved
4:0	R/W	0x0	Reserved

8.4.6.116 0x0318 MICBIAS Analog Control Register (Default Value: 0x4000_3030)

Offset: 0x0318			Register Name: MICBIAS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	/	/	/

Offset: 0x0318			Register Name: MICBIAS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
30:28	R/W	0x4	SELDETADCFS Select sample interval of the ADC sample 000: 2 ms … 100: 32 ms … 111: 256 ms
27:26	R/W	0x0	SELDETADCDB Select debounce time when jack removal 00: 128 ms 01: 256 ms 10: 512 ms 11: 1024 ms
25:24	R/W	0x0	SELDETADCBF Select the time to enable HBIAS before MICADC work 00: 2 ms 01: 4 ms 10: 8 ms 11: 16 ms
23	R/W	0x0	JACKDETEN Jack detect enable 0: Disable 1: Enable
22:21	R/W	0x0	SELDETADCDY Select the delay time to pull low the micdet when jack removal 00: 0.5 ms 01: 1 ms 10: 1.5 ms 11: 2 ms
20	R/W	0x0	MICADCEN Microphone detect ADC enable 0: Disabled 1: Enabled
19	R/W	0x0	POPFREE When this bit is 0, HBIAS MICADC is controlled by registor

Offset: 0x0318			Register Name: MICBIAS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
18	R/W	0x0	DET_MODE MIC Detect Mode 0: Jack in pull low 1: Jack in pull high
17	R/W	0x0	AUTOPLEN Enable the function to auto pull low MICDET when jack removal 0: Disabled 1: Enabled
16	R/W	0x0	MICDETPL When this bit is 1and AUTOPLEN is 0, the MICDET is pulled down to GND.
15	R/W	0x0	HMICBIASEN Headphone Microphone Bias Enable 0: Disabled 1: Enabled
14:13	R/W	0x1	HBIASSEL HMICBIAS Voltage Level Select 00: 1.88 V 01: 2.09 V 10: 2.33 V 11: 2.55 V
12	R/W	0x1	HMIC_BIAS_CHOPPER_EN HMIC BIAS Chopper Enable 0: Disabled 1: Enabled
11:10	R/W	0x0	HMIC_BIAS_CHOPPER_CLK_SEL HMIC BIAS Chopper Clock Select 00: 250 kHz 01: 500 kHz 10: 1 MHz 11: 2 MHz
9:8	/	/	/
7	R/W	0x0	MMICBIASEN Master Microphone Bias Enable 0: Disabled 1: Enabled

Offset: 0x0318			Register Name: MICBIAS_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
6:5	R/W	0x1	MBIASSEL MMICBIAS Voltage Level Select 00: 1.88 V 01: 2.09 V 10: 2.33 V 11: 2.50 V
4	R/W	0x1	MMIC_BIAS_CHOPPER_EN MMIC BIAS Chopper Enable 0: Disabled 1: Enabled
3:2	R/W	0x0	MMIC_BIAS_CHOPPER_CLK_SEL MMIC BIAS Chopper Clock Select 00: 250 kHz 01: 500 kHz 10: 1 MHz 11: 2 MHz
1:0	/	/	/

8.4.6.117 0x031C Ramp Control Register (Default Value: 0x0018_0000)

Offset: 0x031C			Register Name: RAMP_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	RAMP_RISE_INT_EN RAMP Rise Interrupt Enable 0: Enabled 1: Disabled
30	R/W1C	0x0	RAMP_RISE_INT RK Increase Upward Finish and Rampen Pull Down Instruction 0: No Pending IRQ 1: Ramp Rise Finish Pending Interrupt Write '1' to clear this interrupt.
29	R/W	0x0	RAMP_FALL_INT_EN RAMP Fall Int Enable 0: Enabled 1: Disabled

Offset: 0x031C			Register Name: RAMP_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
28	R/W1C	0x0	RAMP_FALL_INT RK Downward Decrease Finish and Rampen Pull Down Instruction 0: No Pending IRQ 1: Ramp Fall Finish Pending Interrupt Write '1' to clear this interrupt.
27:25	/	/	/
24	R/W	0x0	RAMP_SRST Ramp Soft Reset 0: Disabled 1: Enabled
23:21	/	/	/
20:16	R/W	0x18	RAMP_CLK_DIV_M Analog Ramp Clk Div Freq Value : M (from 0 to 31, Default: 24). Ana_Ramp_Clk= 24MHz/(M+1) Default Ramp Clk Freq: 24MHz/(24+1)=960 kHz
15	R/W	0x0	HP_PULL_OUT_EN Heanphone Pullout Enable 0: Disabled 1: Enabled

Offset: 0x031C			Register Name: RAMP_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
14:12	R/W	0x0	RAMP_HOLD_STEP RAMP HOLD STEP 000: 9600 001: 19200 010: 38400 011: 76800 100: 96000 101: 115200 110: 153600 111: 192000 Ramp Hold Time = Ramp Hold Step/Ramp Clk Freq When Ramp Clk Freq is equal to 960 kHz, the corresponding Ramp Hold time of each gear is: 000: 9600/960 kHz=10 ms 001: 19200/960 kHz=20 ms 010: 38400/960 kHz=40 ms 011: 76800/960 kHz=80 ms 100: 96000/960 kHz=100 ms 101: 115200/960 kHz=120 ms 110: 153600/960 kHz=160 ms 111: 192000/960 kHz=200 ms
11:10	/	/	/
9:8	R/W	0x0	GAP_STEP Gap Step 00: ramp step 01: ramp step*2 10: ramp step*3 11: ramp step*4
7	/	/	/

Offset: 0x031C			Register Name: RAMP_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
6:4	R/W	0x0	RAMP_STEP RK Frequency Gear, Control Ramp Rise/Fall Total Time 000: 20 001: 30 010: 40 011: 60 100: 80 101: 120 110: 160 111: 240 Ramp Rise/Fall Total Time =(Ramp Step/Ramp Clk Freq)*4096 When Default Ramp Clk Freq is equal to 960 kHz, the corresponding time of each gear is: 000: (20/960kHz)*4096=85.3 ms 001: (30/960kHz)*4096=128 ms 010: (40/960kHz)*4096=170.6 ms 011: (60/960kHz)*4096=256 ms 100: (80/960kHz)*4096=341.3 ms 101: (120/960kHz)*4096=512 ms 110: (160/960kHz)*4096=682.6 ms 111: (240/960kHz)*4096=1024 ms
3	R/W	0x0	RMD_EN Ramp Manual Down Enable 0: Disabled 1: Enabled
2	R/W	0x0	RMU_EN Ramp Manual Up Enable 0: Disabled 1: Enabled
1	R/W	0x0	RMC_EN Ramp Manual Control Enable 0: Disabled 1: Enabled

Offset: 0x031C			Register Name: RAMP_REG
Bit	Read/Write	Default/Hex	Description
0	R/W	0x0	RD_EN Ramp Digital Enable 0: Disabled 1: Enabled

8.4.6.118 0x0320 BIAS Analog Control Register (Default Value: 0x0000_0080)

The register is not controlled by the clock and reset of Audio Codec, only controlled by the clock and reset of system bus.

Offset: 0x0320			Register Name: BIAS_REG
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:0	R/W	0x80	BIASDATA Bias Current Register Setting Data

8.4.6.119 0x0328 HMIC Control Register (Default Value: 0x0000_0008)

Offset: 0x0328			Register Name: HMIC_CTRL
Bit	Read/Write	Default/Hex	Description
31:23	/	/	/
22:21	R/W	0x0	HMIC_SAMPLE_SELECT Down Sample Setting Select 00: Down by 1, 128 Hz 01: Down by 2, 64 Hz 10: Down by 4, 32 Hz 11: Down by 8, 16 Hz
20:16	R/W	0x0	MDATA_THRESHOLD MIC DET EN Threshold Value

Offset: 0x0328			Register Name: HMIC_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
15:14	R/W	0x0	HMIC_SF HMIC Smooth Filter setting 00: by pass 01: (x1+x2)/2 10: (x1+x2+x3+x4)/4 11: (x1+x2+x3+x4+ x5+x6+x7+x8)/8
13:10	R/W	0x0	HMIC_M Debounce when the MIC Key down or up. 0000 :1 samlpe data 0001 :2 samlpe data ... 1111 :16 samlpe data
9:6	R/W	0X0	HMIC_N Debounce when earphone plug in or pull out 125 ms to 2 s 0000:125 ms 0001:250 ms ... 1111:2 s
5:3	R/W	0x1	MDATA_THRESHOLD_DEBOUNCE MDATA Threshold Debounce 000: 0 001: 1 010: 2 011: 3 100: 4 101: 5 110: 6 111: 7
2	R/W	0x0	JACK_OUT_IRQ_EN MIC Detect Interrupt Set 0: Disabled 1: Enabled

Offset: 0x0328			Register Name: HMIC_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
1	R/W	0x0	JACK_IN_IRQ_EN MIC Detect Interrupt Set 0: Disabled 1: Enabled
0	R/W	0x0	MIC_DET_IRQ_EN MIC Detect Interrupt Set 0: Disabled 1: Enabled

8.4.6.120 0x032C HMIC Status Register (Default Value: 0x0000_6000)

Offset: 0x032C			Register Name: HMIC_STS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:15	/	/	/
14:13	R/W	0x3	MDATA_DISCARD After MIC DATA data is received, the first N-data will be discarded. N defined as follows: 00: None discarded 01: 1-data discarded 10: 2-data discarded 11: 4-data discarded
12:8	R	0x0	HMIC_DATA HMIC Average Data
7:5	/	/	/
4	R/W1C	0x0	JACK_DET_OIRQ Jack output detect pending interrupt 0: No Pending IRQ 1: Pending IRQ Writing 1 clear pending.
3	R/W1C	0x0	JACK_DET_IIRQ Jack input detect pending interrupt 0: No Pending IRQ 1: Pending IRQ Writing 1 clear pending.
2:1	/	/	/

Offset: 0x032C			Register Name: HMIC_STS
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W1C	0x0	MIC_DET_ST MIC detect pending interrupt 0: No pending IRQ 1: Pending IRQ Writing 1 clear pending.

8.4.6.121 0x0340 Headphone2 Analog Control Register (Default Value: 0x0640_4000)

Offset: 0x0340			Register Name: HP2_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x0	HPFB_BUF_EN Headphone Feedback Buffer OP Enable 0: Disable 1: Enable
30:28	R/W	0x0	HEADPHONE_GAIN HeadPhone Gain 000: 0 dB 001: -6 dB 010: -12 dB 011: -18 dB 100: -24 dB 101: -30 dB 110: -36 dB 111: -42 dB
27:26	R/W	0x1	HPFB_RES Headphone Feedback Big Resistor Control 00: 0.88 MΩ 01: 1.00 MΩ 10: 1.08 MΩ 11: 1.20 MΩ
25:24	R/W	0x2	OPDRV_CUR Headphone OP Output Stage Current Setting 00: Min 11: Max

Offset: 0x0340			Register Name: HP2_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
23:22	R/W	0x1	IOPHP Headphone L/R OP Bias Current Select 00: 6 uA 01: 7 uA 10: 8 uA 11: 9 uA
21	R/W	0x0	HP_DRVEN Headphone Driver Enable 0: Disable 1: Enable
20	R/W	0x0	HP_DRVOUTEN Headphone Driver Output Enable 0: Disabled 1: Enabled
19	R/W	0x0	RSWITCH RSwitch 0: HPOUT OUTPUT VCM of RAMP_DAC 1: VRA1
18	R/W	0x0	RAMPEN Ramp DAC Enable 0: Disabled 1: Enabled
17	R/W	0x0	HPFB_IN_EN Headphone Feedback PAD IN Switch Enable 0: Disabled 1: Enabled
16	R/W	0x0	RAMP_FINAL_CONTROL Headphone Ramp Final Step Control 0: Ramp Output Select Ramp 1: Ramp Output Select HPFB buffer Output
15	R/W	0x0	RAMP_OUT_EN Ramp Output Switch Enable 0: Disable 1: Enable

Offset: 0x0340			Register Name: HP2_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
14:13	R/W	0x2	RAMP_FINAL_STATE_RES Ramp Final State Resistor 00: 2.5k 01: 5.0k 10: 10k 11: 20k
9:8	R/W	0x0	HPFB_BUF_OUTPUT_CURRENT Headphone Feedback Buffer Output Current Select 00: 35I 01: 28I 10: 45I 11: 38I I=7 uA
7:0	/	/	/

8.4.6.122 0x0348 POWER Analog Control Register (Default Value: 0x8000_3325)

The register is not controlled by the clock and reset of Audio Codec, only controlled by the clock and reset of system bus.

Offset: 0x0348			Register Name: POWER_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/W	0x1	ALDO_EN ALDO Enable 0: Disabled 1: Enabled
30	R/W	0x0	HPLDO_EN HPLDO Enable 0: Disabled 1: Enabled
29	R/W	0x0	VAR1SPEEDUP_DOWN_Further_CTRL VAR1 Speedup Down Further Control In Adda Analog 0: The digital logic signal input by the digital-analog interface pin controls the var1_speedup_down function (that is, the var1 signal is rapidly pulled up/down) 1: Writing 1 can finish the var1_speedup_down function (ignore the control of the digital-analog interface pin)

Offset: 0x0348			Register Name: POWER_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
28:17	/	/	/
16	R	0x0	AVCCPOR Avccpor Monitor
15	/	/	/
14:12	R/W	0x3	ALDO_OUTPUT_VOLTAGE ALDO Output Voltage Control 000: 2.03 V 001: 1.95 V 010: 1.87 V 011: 1.80 V 100: 1.73 V 101: 1.67 V 110: 1.61 V 111: 1.56 V
11	/	/	/
10:8	R/W	0x3	HPLDO_OUTPUT_VOLTAGE HPLDO Output Voltage Control 000: 2.03 V 001: 1.95 V 010: 1.87 V 011: 1.80 V 100: 1.73 V 101: 1.67 V 110: 1.61 V 111: 1.56 V
7:0	R/W	0x25	BG_TRIM BG Output Voltage Trimming Only low 6-bit is used. The BG output voltage range is from 0.7 V to 1.208 V.

8.4.6.123 0x034C ADC Current Analog Control Register (Default Value: 0x0015_1515)

Offset: 0x034C			Register Name: ADC_CUR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:22	/	/	/

Offset: 0x034C			Register Name: ADC_CUR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
21:20	R/W	0x1	ADC3_IOPMIC2 ADC3 OP MIC2 Bias Current Select 00: 1.50*IOPADC 01: 1.75*IOPADC 10: 2.00*IOPADC 11: 2.25*IOPADC IOPADC is defined by ADC1_REG<15:14> from 1 uA to 4 uA
19:18	R/W	0x1	ADC3_OUTPUT_CURRENT ADC3 OP MIC1 Output Current Select 00: 15I 01: 20I 10: 35I 11: 40I I=7 uA
17:16	R/W	0x1	ADC3_OUTPUT_CURRENT ADC3 OP MIC2 Output Current Select 00: 15I 01: 20I 10: 35I 11: 40I I=7 uA
15:14	/	/	/
13:12	R/W	0x1	ADC2_IOPMIC2 ADC2 OP MIC2 Bias Current Select 00: 1.50*IOPADC 01: 1.75*IOPADC 10: 2.00*IOPADC 11: 2.25*IOPADC IOPADC is defined by ADC1_REG<15:14> from 1 uA to 4 uA.
11:10	R/W	0x1	ADC2_OUTPUT_CURRENT ADC2 OP MIC1 Output Current Select 00: 15I 01: 20I 10: 35I 11: 40I I=7 uA

Offset: 0x034C			Register Name: ADC_CUR_REG
Bit	**Read/Write**	**Default/Hex**	**Description**
9:8	R/W	0x1	ADC2_OUTPUT_CURRENT ADC2 OP MIC2 Output Current Select 00: 15I 01: 20I 10: 35I 11: 40I I=7 uA
7:6	/	/	/
5:4	R/W	0x1	ADC1_IOPMIC2 ADC1 OP MIC2 Bias Current Select 00: 1.50*IOPADC 01: 1.75*IOPADC 10: 2.00*IOPADC 11: 2.25*IOPADC IOPADC is defined by ADC1_REG<15:14> from 1 uA to 4 uA.
3:2	R/W	0x1	ADC1_OUTPUT_CURRENT ADC1 OP MIC1 Output Current Select 00: 15I 01: 20I 10: 35I 11: 40I I=7 uA
1:0	R/W	0x1	ADC1_OUTPUT_CURRENT ADC1 OP MIC2 Output Current Select 00: 15I 01: 20I 10: 35I 11: 40I I=7 uA

Contents

Figures

Tables

9 Interfaces

9.1 TWI

9.1.1 Overview

The Two Wire Interface (TWI) provides an interface between a CPU and any TWI-bus-compatible device that connects via the TWI bus. The TWI is designed to be compatible with the standard I2C bus protocol. The communication of the TWI is carried out by a byte-wise mode based on interrupt polled handshaking. Each device on the TWI bus is recognized by a unique address and can operate as either transmitter or receiver, a device connected to the TWI bus can be considered as master or slave when performing data transfers. Note that a master device is a device that initiates a data transfer on the bus and generates the clock signals to permit the transfer. During this transfer, any device addressed by this master is considered a slave.

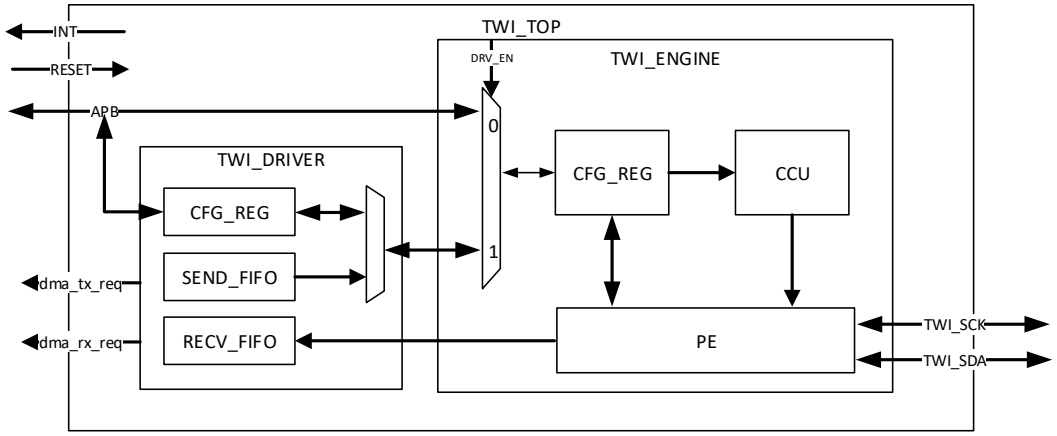
The TWI has the following features:

- Compliant with I2C bus standard
- Supports standard mode (up to 100 kbit/s) and fast mode (up to 400 kbit/s)
- Supports 7-bit and 10-bit device addressing modes
- Supports master mode or slave mode
- Master mode features:
  - Supports the bus arbitration in the case of multiple master devices
  - Supports clock synchronization and bit and byte waiting
  - Supports packet transmission and DMA
- Slave mode features:
  - Interrupt on address detection
- The TWI controller includes one TWI engine and one TWI driver. And the TWI driver supports packet transmission and DMA mode when TWI works in master mode

9.1.2 Block Diagram

Figure 9-1 shows the block diagram of TWI.

Figure 9-1 TWI Block Diagram



TWI contains the following sub-blocks:

Table 9-1 TWI Sub-blocks

Sub-block	Description
RESET	Module reset signal
INT	Module output interrupt signal
CFG_REG	Module configuration register in TWI
PE	Packet encoding/decoding
CCU	Module clock controller unit
SEND_FIFO	The register address bytes and the written data bytes are buffered in SEND_FIFO
RECV_FIFO	The read data bytes are buffered in RECV_FIFO

The controller includes TWI engine and TWI driver. Each time the TWI engine sends a START signal, a STOP signal, or a BYTE data, or a corresponding ACK, the TWI engine will generate an interrupt, and wait for the CPU to process and clear the interrupt before the next START, STOP, or BYTE, ACK transmission can be performed. Therefore, when a device communication is completed, many interrupts will be generated, and the CPU needs to wait for the previous interrupt before it can configure the next one. The TWI driver defines each communication with the device as a packet transmission. The CPU can directly configure the slave address, register address and data transmission for one or more package transmissions without waiting for interruption, then start the TWI driver, and the TWI driver can control the TWI engine to complete a pre-configured communication, and report an interrupt to the CPU after completion.

9.1.3 Functional Description

9.1.3.1 External Signals

The TWI controller has 4 TWI modules called TWI0, TWI1, TWI2, and TWI3. The following table describes the external signals of the TWI. The TWIn-SCK and TWIn-SDA are bidirectional I/O, when the TWI is configured as a master device, the TWIn-SCK is an output pin; when the TWI is configurable as a slave device, the TWIn-SCK is an input pin. When using TWI, the corresponding PADs are selected as TWI function via section 9.7 "GPIO".

Table 9-2 TWI External Signals

Signal	Description	Type
TWI0-SCK	TWI0 Clock Signal	I/O, OD
TWI0-SDA	TWI0 Serial Data	I/O, OD
TWI1-SCK	TWI1 Clock Signal	I/O, OD
TWI1-SDA	TWI1 Serial Data	I/O, OD
TWI2-SCK	TWI2 Clock Signal	I/O, OD
TWI2-SDA	TWI2 Serial Data	I/O, OD
TWI3-SCK	TWI3 Clock Signal	I/O, OD
TWI3-SDA	TWI3 Serial Data	I/O, OD

9.1.3.2 Clock Sources

Each TWI controller has an input clock source. The following table describes the clock sources for TWI. After selecting a proper clock, users must open the gating of TWI and release the corresponding reset bit.

For more details on the clock setting, configuration, and gating information, see section 3.3 "CCU".

Table 9-3 TWI Clock Sources

Clock Sources	Description
APB1 Bus	TWI clock source. Refer to CCU for details on APB1.

9.1.3.3 Write/Read Timing in Standard and Extended Addressing Mode

This section is the 7-bit/10-bit addressing mode of the entire TWI protocol to read and write device registers. It can be achieved by directly using the TWI engine or using the TWI driver to control the TWI engine.

Figure 9-2 describes the write timing in 7-bit standard addressing mode.

Figure 9-2 Write Timing in 7-bit Standard Addressing Mode

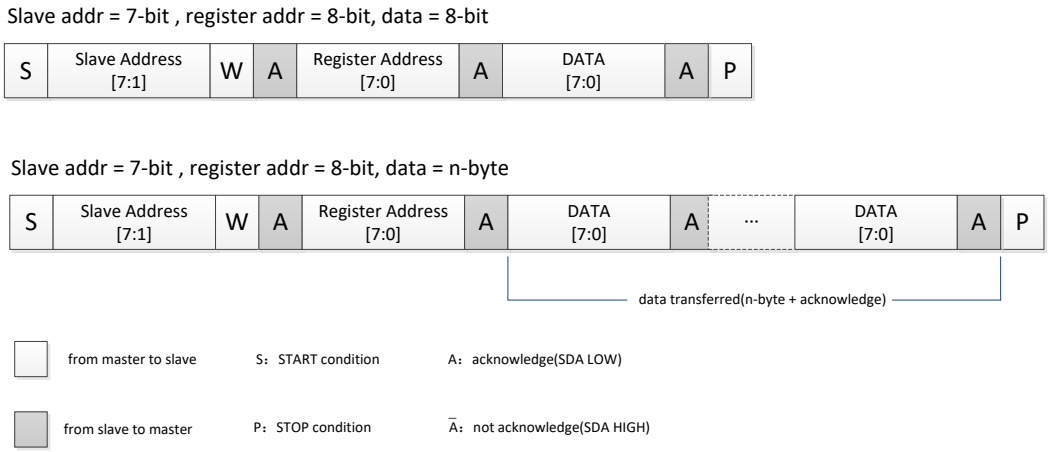


Figure 9-3 describes the read timing in 7-bit standard address mode.

Figure 9-3 Read Timing in 7-bit Standard Addressing Mode

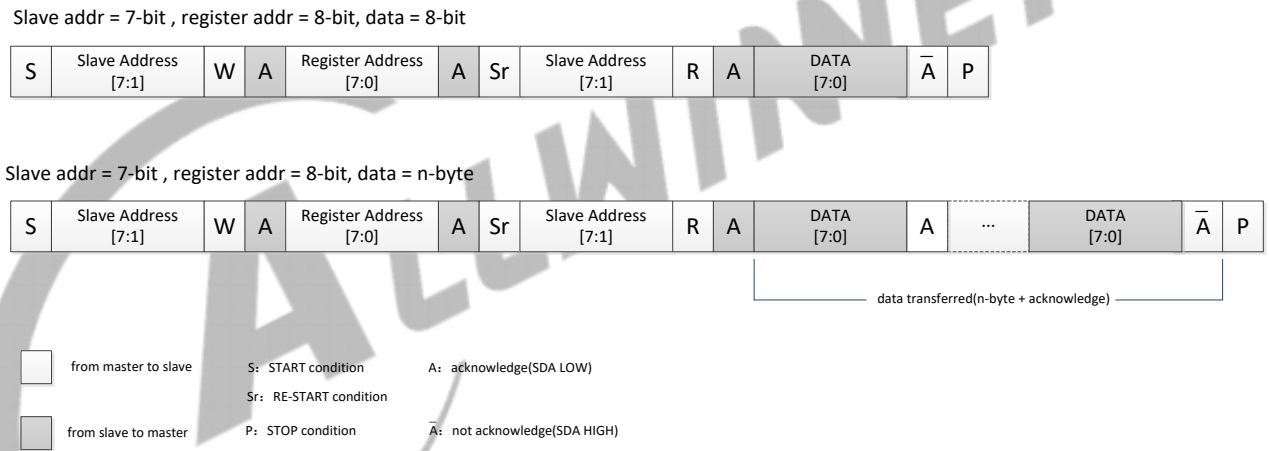


Figure 9-4 describes the write timing in 10-bit extended address mode.

Figure 9-4 Write Timing in 10-bit Extended Addressing Mode

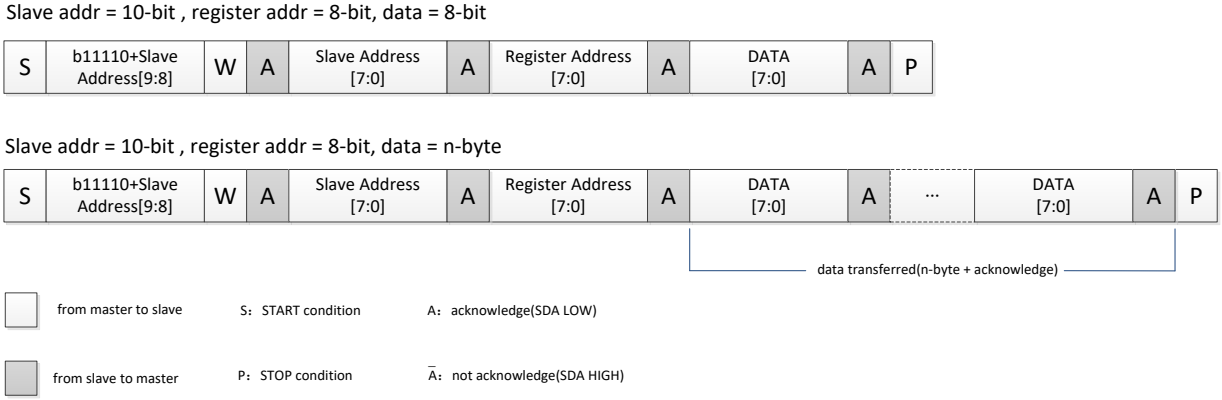
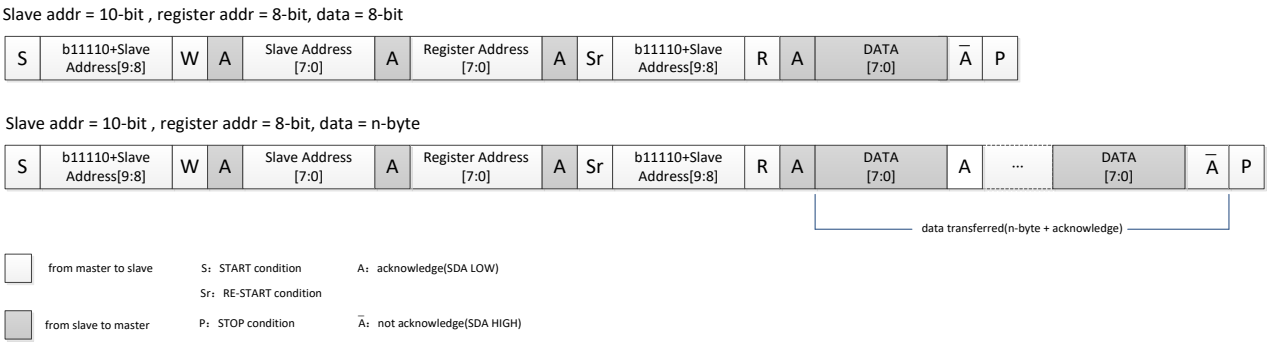


Figure 9-5 describes the read timing in 10-bit extended address mode.

Figure 9-5 Read Timing in 10-bit Extended Addressing Mode



9.1.3.4 Write/Read Packet Transmission of TWI Driver

The TWI driver is only supported for master mode. When the TWI works in master mode, the TWI driver drives the TWI engine for one or more packet transmission instead of the CPU host. Packet transmission is defined in the following figures. The register address bytes and the written data bytes are buffered in SEND_FIFO, the read data bytes are buffered in RECV_FIFO.

Figure 9-6 TWI Driver Write Packet Transmission

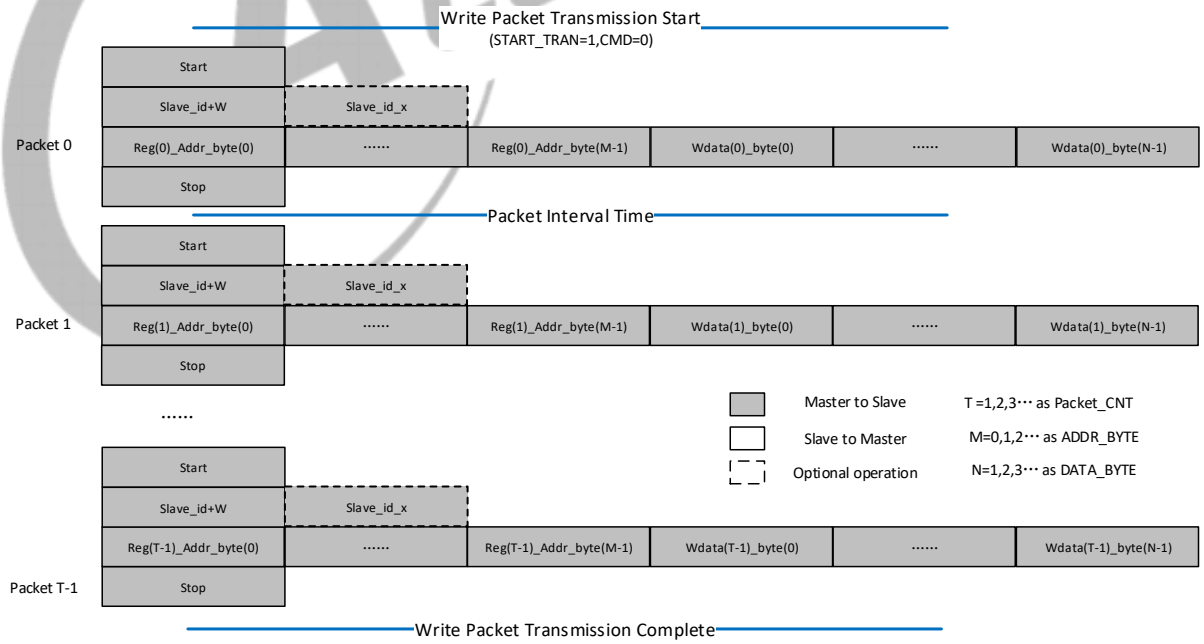
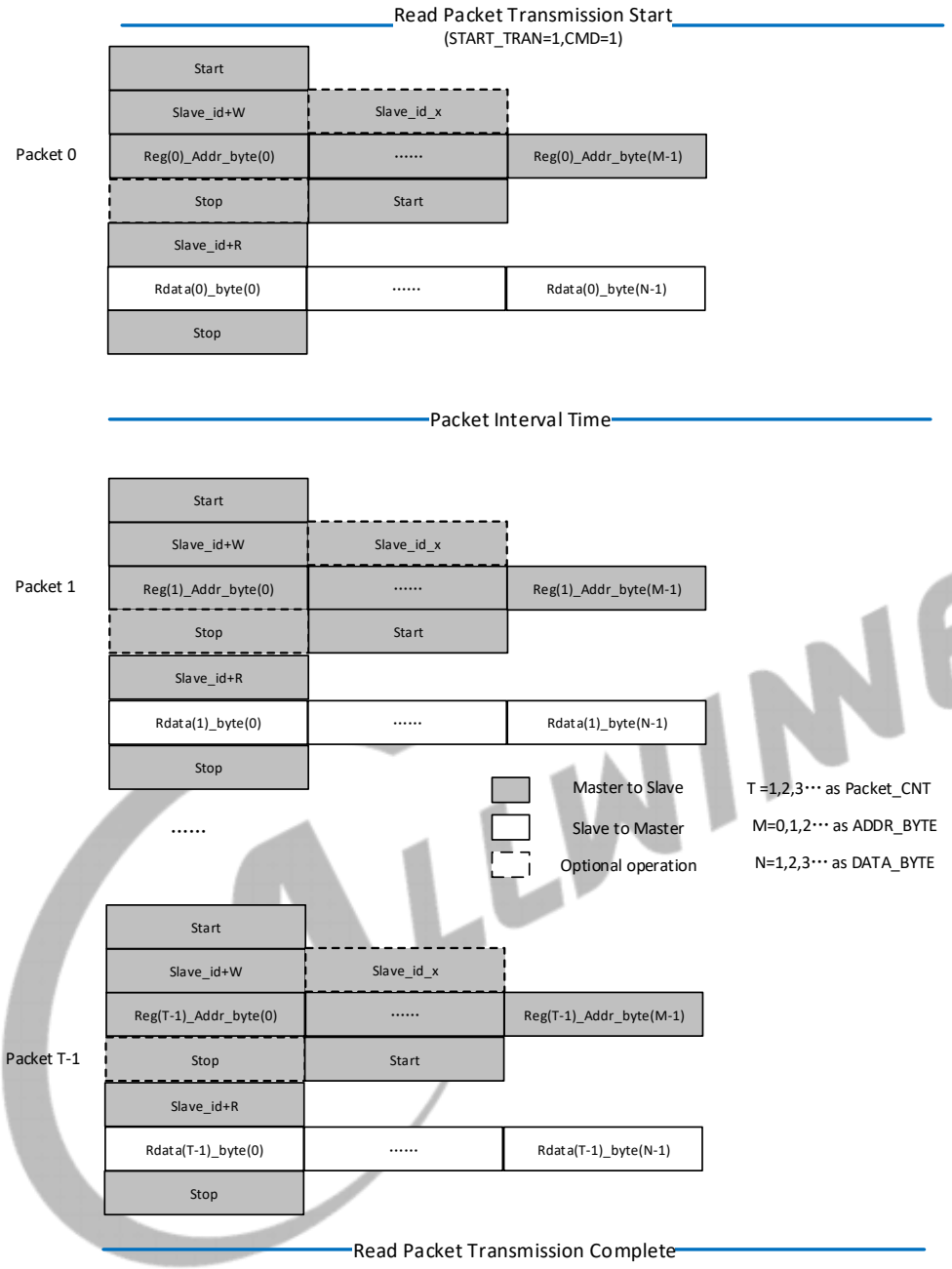


Figure 9-7 TWI Driver Read Packet Transmission



9.1.3.5 Master and Slave Mode of TWI Engine

In Master mode, the CPU host controls the TWI engine by writing command and data to its registers. The TWI engine transmits an interrupt to CPU when each time a byte transfer is done or a START/STOP command is detected. The CPU host can poll the status register if the interrupt mechanism is not disabled by the CPU host.

When the CPU host wants to start a bus transfer, it initiates a bus START to enter the master mode by setting **TWI_CNTR**[M_STA] to high. The TWI engine will assert the INT line and **TWI_CNTR**[INT_FLAG] to indicate a completion for the START command and each consequent byte transfer. At each interrupt, the CPU host needs

to check the current state by the TWI_STAT register. A transfer must conclude with the STOP command by setting TWI_CNTR[M_STP] to high.

In Slave mode, the TWI engine also constantly samples the bus and look for its own slave address during addressing cycles. Once a match is found, it is addressed, and the TWI engine interrupts the CPU host with the corresponding status. Upon request, the CPU host should read the status, read/write the TWI_DATA register, and set the TWI_CNTR register. After each byte transfer, a slave device always stops the operation of the remote master by holding the next low pulse on the SCL line until the CPU host responds to the status of the previous byte transfer or START command.

9.1.3.6 Generation of Repeated Start

After the data transfer, if the master still requires the bus, it can signal another Start followed by another slave address without signaling a Stop.

9.1.3.7 Programming State Diagram

Figure 9-8 shows the TWI programming state diagram. For the value between two states, see the TWI_STAT register in section 9.1.6.5.

M_SEND_S: master sends START signal;

M_SEND_ADDR: master sends slave address;

M_SEND_XADD: master sends slave extended address;

M_SEND_SR: master repeated start;

M_SEND_DATA: master sends data;

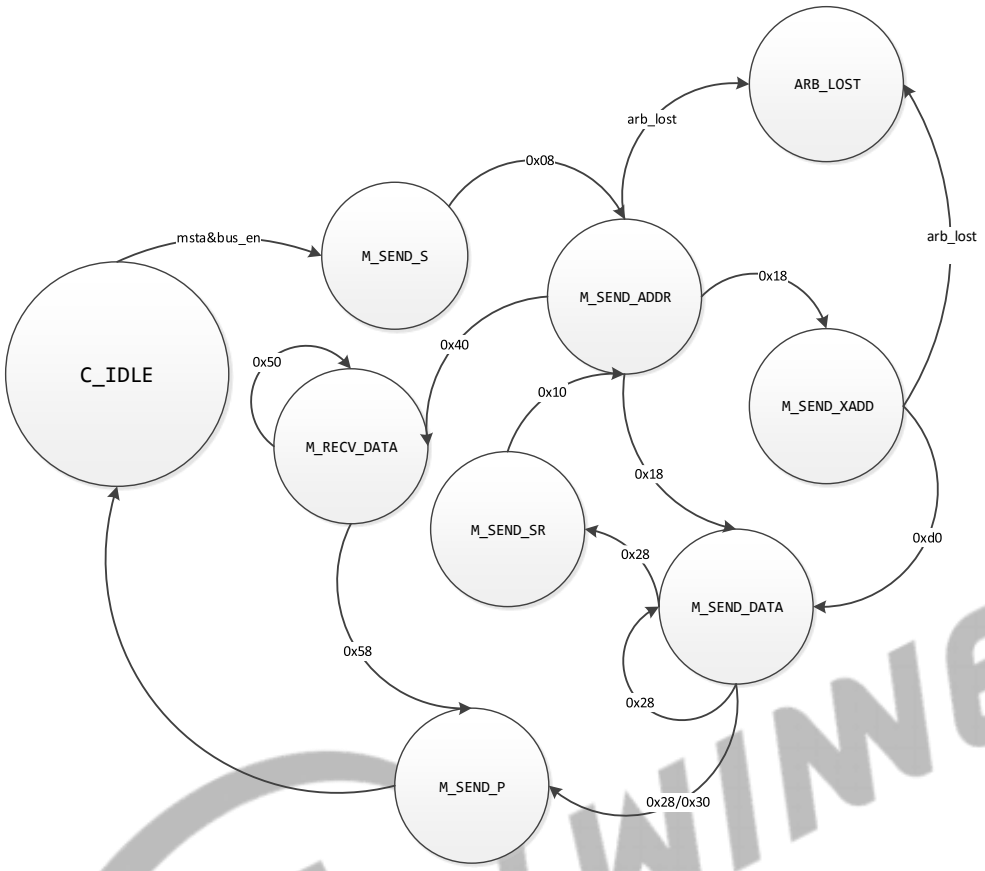
M_SEND_P: master sends STOP signal;

M_RECV_DATA: master receives data;

ARB_LOST: Arbitration lost;

C_IDLE: Idle.

Figure 9-8 TWI Programming State Diagram



9.1.4 Programming Guidelines

The TWI controller operates in an 8-bit data format. The data on the TWI_SDA line is always 8 bits long. At first, the TWI controller sends a start condition. When in the addressing formats of 7-bit, the TWI sends out an 8-bit message which includes 7 MSB slave address and 1 LSB read/write flag. The least significant of the salve address indicates the direction of transmission. When the TWI works in 10-bit slave address mode, the operation will be divided into two steps, for details on the operation, refer to register description in Section 9.1.6.1 and 9.1.6.2.

The following takes the TWI module in the CPUX domain as an example.

9.1.4.1 Initialization for TWI Engine

To initialize the TWI engine, perform the following steps:

Step 1 Configure corresponding GPIO multiplex function as TWI mode.

Step 2 For TWIn, set **TWI_BGR_REG**[TWIn_GATING] in CCU module to 0 to close TWIn clock.

Step 3 For TWIn, set **TWI_BGR_REG**[TWIn_RST] in CCU module to 0, then set to 1 to reset TWIn.

Step 4 For TWIn, set **TWI_BGR_REG**[TWIn_GATING] in CCU module to 1 to open TWIn clock.

Step 5 Configure **TWI_CCR**[CLK_M] and **TWI_CCR**[CLK_N] to get the needed rate (The clock source of TWI is from APB1).

Step 6 Configure **TWI_CNTR**[BUS_EN] and **TWI_CNTR**[A_ACK], when using interrupt mode, set **TWI_CNTR**[INT_EN] to 1, and register the system interrupt through GIC module. In slave mode, configure **TWI_ADDR** and **TWI_XADDR** registers to finish TWI initialization configuration.

9.1.4.2 Writing Data Operation for TWI Engine

To write data to the device, perform the following steps:

Step 1 Clear **TWI_EFR** register, and configure **TWI_CNTR**[M_STA] to 1 to transmit the START signal.

Step 2 After the START signal is transmitted, the first interrupt is triggered, then write device ID to **TWI_DATA** (For a 10-bit device ID, firstly write the first byte ID, secondly write the second byte ID in the next interrupt).

Step 3 The Interrupt is triggered again after device ID transmission completes, write device data address to be read to **TWI_DATA** (For a 16-bit address, firstly write the first-byte address, secondly write the second-byte address).

Step 4 Interrupt is triggered after data address transmission completes, write data to be transmitted to **TWI_DATA** (For consecutive write data operation, every byte transmission completion triggers interrupt, during interrupt write the next byte data to **TWI_DATA**).

Step 5 After transmission completes, write **TWI_CNTR**[M_STP] to 1 to transmit the STOP signal and end this write-operation.

9.1.4.3 Reading Data Operation for TWI Engine

To read data from the device, perform the following steps:

Step 1 Clear **TWI_EFR** register, and set **TWI_CNTR**[A_ACK] to 1, and configure **TWI_CNTR**[M_STA] to 1 to transmit the START signal.

Step 2 After the START signal is transmitted, the first interrupt is triggered, then write device ID to **TWI_DATA** (For a 10-bit device ID, firstly write the first-byte ID, secondly write the second-byte ID in the next interrupt).

Step 3 The Interrupt is triggered again after device ID transmission completes, write device data address to be read to **TWI_DATA** (For a 16-bit address, firstly write the first-byte address, secondly write the second-byte address).

Step 4 The Interrupt is triggered after data address transmission completes, write **TWI_CNTR**[M_STA] to 1 to transmit new START signal, and after interrupt triggers, write device ID to **TWI_DATA** to start read-operation.

Step 5 After device address transmission completes, each receive completion will trigger an interrupt, in turn, read **TWI_DATA** to get data, when receiving the previous interrupt of the last byte data, clear C:\Users\zengjing\Downloads_Hlk50046436 - _Hlk50051279[A_ACK] to stop acknowledge signal of the last byte.

Step 6 Write **TWI_CNTR**[M_STP] to 1 to transmit the STOP signal and end this read-operation.

9.1.4.4 Initialization for TWI Driver

To initialize the TWI driver, perform the following steps:

Step 1 Configure corresponding GPIO multiplex function as TWI mode.

Step 2 For TWIn, set **TWI_BGR_REG**[TWIn_GATING] in CCU module to 0 to close TWIn clock.

Step 3 For TWIn, set **TWI_BGR_REG**[TWIn_RST] in CCU module to 0, then set to 1 to reset TWIn.

Step 4 For TWIn, set **TWI_BGR_REG**[TWIn_GATING] in CCU module to 1 to open TWIn clock.

Step 5 Set **TWI_DRV_CTRL**[TWI_DRV_EN] to 1 to enable the TWI driver.

Step 6 Configure **TWI_DRV_BUS_CTRL**[CLK_M] and **TWI_DRV_BUS_CTRL**[CLK_N] to get the needed rate (The clock source of TWI is from APB1).

Step 7 Set **TWI_DRV_CTRL**[RESTART_MODE] to 0 and C:\Users\zengjing\Downloads_Hlk50051373 - _Hlk50051463[READ_TRAN_MODE] to 1, set **TWI_DRV_INT_CTRL**[TRAN_COM_INT_EN] to 1.

Step 8 When using DMA for data transmission, set **TWI_DRV_DMA_CFG**[DMA_RX_EN] and **TWI_DRV_DMA_CFG**[DMA_TX_EN] to 1, and configure **TWI_DRV_DMA_CFG**[RX_TRIG] and **TWI_DRV_DMA_CFG**[TX_TRIG] to set the thresholds of RXFIFO and TXFIFO.

9.1.4.5 Writing Packet Transmission for TWI Driver

To write package to the device, perform the following steps:

Step 1 Configure **TWI_DRV_SLV**[SLV_ID] to set the device ID, and configure **TWI_DRV_SLV**[CMD] to 0 to set the write operation.

Step 2 Configure **TWI_DRV_FMT**[ADDR_BYTE] according to the address width of the device register, and **TWI_DRV_FMT**[DATA_BYTE] according to the written data count in a packet.

Step 3 Configure **TWI_DRV_CFG**[PACKET_CNT] to set the written packet number.

Step 4 Configure DMA channel, including TWI TXFIFO, device register address, and the written data.

Step 5 Set C:\Users\zengjing\Downloads_Hlk50051463_-_Hlk50051932[START_TRAN] to 1 to start TWI Driver transmission.

Step 6 When TWI driver transmission completes, the interrupt is triggered, it indicates that the write packet transmission ends.

9.1.4.6 Reading Packet Transmission for TWI Driver

To read package from the device, perform the following steps:

Step 1 Configure **TWI_DRV_SLV**[SLV_ID] to set the device ID, and configure **TWI_DRV_SLV**[CMD] to 1 to set the read operation.

Step 2 Configure **TWI_DRV_FMT**[ADDR_BYTE] according to the address width of the device register, and **TWI_DRV_FMT**[DATA_BYTE] according to the read data count in a packet.

Step 3 Configure **TWI_DRV_CFG**[PACKET_CNT] to set the read packet number.

Step 4 Configure DMA channel, including TWI TXFIFO, TWI RXFIFO, device register address and the read data.

Step 5 Set C:\Users\zengjing\Downloads_Hlk50051463_-_Hlk50051932[START_TRAN] to 1 to start TWI Driver transmission.

Step 6 When TWI driver transmission completes, the interrupt is triggered, it indicates that the read packet transmission ends.

9.1.5 Register List

Module Name	Base Address
TWI0	0x02502000
TWI1	0x02502400
TWI2	0x02502800
TWI3	0x02502C00

Register Name	Offset	Description
TWI_ADDR	0x0000	TWI Slave Address Register
TWI_XADDR	0x0004	TWI Extended Slave Address Register
TWI_DATA	0x0008	TWI Data Byte Register
TWI_CNTR	0x000C	TWI Control Register

Register Name	Offset	Description
TWI_STAT	0x0010	TWI Status Register
TWI_CCR	0x0014	TWI Clock Control Register
TWI_SRST	0x0018	TWI Software Reset Register
TWI_EFR	0x001C	TWI Enhance Feature Register
TWI_LCR	0x0020	TWI Line Control Register
TWI_DRV_CTRL	0x0200	TWI_DRV Control Register
TWI_DRV_CFG	0x0204	TWI_DRV Transmission Configuration Register
TWI_DRV_SLV	0x0208	TWI_DRV Slave ID Register
TWI_DRV_FMT	0x020C	TWI_DRV Packet Format Register
TWI_DRV_BUS_CTRL	0x0210	TWI_DRV Bus Control Register
TWI_DRV_INT_CTRL	0x0214	TWI_DRV Interrupt Control Register
TWI_DRV_DMA_CFG	0x0218	TWI_DRV DMA Configure Register
TWI_DRV_FIFO_CON	0x021C	TWI_DRV FIFO Content Register
TWI_DRV_SEND_FIFO_ACC	0x0300	TWI_DRV Send Data FIFO Access Register
TWI_DRV_RECV_FIFO_ACC	0x0304	TWI_DRV Receive Data FIFO Access Register

9.1.6 Register Description

9.1.6.1 0x0000 TWI Slave Address Register (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: TWI_ADDR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:1	R/W	0x0	SLA Slave Address For 7-bit addressing, the bit[7:1] indicates: SLA6, SLA5, SLA4, SLA3, SLA2, SLA1, SLA0 For 10-bit addressing, the bit[7:1] indicates: 1, 1, 1, 1, 0, SLAX[9:8]
0	R/W	0x0	GCE General Call Address Enable 0: Disable 1: Enable

NOTE

For 7-bit addressing:

SLA6 – SLA0 is the 7-bit address of the TWI in slave mode. When the TWI receives this address after a START condition, it will generate an interrupt and enter slave mode. (SLA6 corresponds to the first bit received from the TWI bus.) If GCE is set to '1', the TWI will also recognize the general call address (00h).

For 10-bit addressing:

When the address received starts with b'11110, the TWI recognizes b'11110 as the first part of a 10-bit address and if the next two bits match ADDR[2:1] (For example, SLAX9 and SLAX8 for the extended address of the device), it sends an ACK. (The device does not generate an interrupt at this point.) If the next byte of the address matches the XADDR register (SLAX7 – SLAX0), the TWI generates an interrupt and goes into slave mode.

9.1.6.2 0x0004 TWI Extend Address Register (Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: TWI_XADDR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0x0	SLAX Extend Slave Address SLAX[7:0]

9.1.6.3 0x0008 TWI Data Register (Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: TWI_DATA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0x0	TWI_DATA Data byte transmitted or received

9.1.6.4 0x000C TWI Control Register (Default Value: 0x0000_0000)

Offset: 0x000C			Register Name: TWI_CNTR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/

Offset: 0x000C			Register Name: TWI_CNTR
Bit	**Read/Write**	**Default/Hex**	**Description**
7	R/W	0x0	INT_EN Interrupt Enable 0: The interrupt line always low 1: The interrupt line will go high when INT_FLAG is set.
6	R/W	0x0	BUS_EN TWI Bus Enable 0: The TWI bus SDA/SCL is ignored and the TWI controller will not respond to any address on the bus. 1: The TWI will respond to call to its slave address – and to the general call address if the GCE bit in the ADDR register is set. **Note: In master operation mode, this bit should be set to '1'.**
5	R/WAC	0x0	M_STA Master Mode Start When the M_STA is set to '1', the TWI controller enters master mode and will transmit a START condition on the bus when the bus is free. If the M_STA bit is set to '1' when the TWI controller is already in master mode and one or more bytes have been transmitted, then a repeated START condition will be sent. If the M_STA bit is set to '1' when the TWI is accessed in slave mode, the TWI will complete the data transfer in slave mode then enter master mode when the bus has been released. The M_STA bit is cleared automatically after a START condition has been sent. Writing a '0' to this bit has no effect.
4	R/W1C	0x0	M_STP Master Mode Stop If the M_STP is set to '1' in master mode, a STOP condition is transmitted on the TWI bus. If the M_STP bit is set to '1' in slave mode, the TWI will indicate if a STOP condition has been received, but no STOP condition will be transmitted on the TWI bus. If both M_STA and M_STP bits are set, the TWI will first transmit the STOP condition (if in master mode), then transmit the START condition. The M_STP bit is cleared automatically. Writing a '0' to this bit has no effect.

Offset: 0x000C			Register Name: TWI_CNTR
Bit	**Read/Write**	**Default/Hex**	**Description**
3	R/W1C	0x0	INT_FLAG Interrupt Flag The INT_FLAG is automatically set to '1' when any of the 28 (out of the possible 29) states is entered (see 'STAT Register' below). The state that does not set INT_FLAG is state F8h. If the INT_EN bit is set, the interrupt line goes high when INT_FLAG is set to '1'. If the TWI is operating in slave mode, the data transfer is suspended when INT_FLAG is set and the low period of the TWI bus clock line (SCL) is stretched until '1' is written to INT_FLAG. The TWI clock line is then released and the interrupt line goes low.
2	R/W	0x0	A_ACK Assert Acknowledge When A_ACK is set to '1', an Acknowledge (low level on SDA) will be sent during the acknowledge clock pulse on the TWI bus if: (1). Either the whole of a matching 7-bit slave address or the first or the second byte of a matching 10-bit slave address has been received. (2). The general call address has been received and the GCE bit in the ADDR register is set to '1'. (3). A data byte has been received in master or slave mode. When A_ACK is '0', a Not Acknowledge (high level on SDA) will be sent when a data byte is received in master or slave mode. If A_ACK is cleared to '0' in slave transmitter mode, the byte in the DATA register is assumed to be the 'last byte'. After this byte has been transmitted, the TWI will enter state C8h then return to the idle state (status code F8h) when INT_FLAG is cleared. The TWI will not respond as a slave unless A_ACK is set.
1	/	/	/
0	R/W	0x0	CLK_COUNT_MODE 0: scl clock high period count on oscl 1: scl clock high period count on iscl

9.1.6.5 0x0010 TWI Status Register (Default Value: 0x0000_00F8)

Offset: 0x0010			Register Name: TWI_STAT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/

Offset: 0x0010			Register Name: TWI_STAT
Bit	**Read/Write**	**Default/Hex**	**Description**
7:0	R	0xF8	STA Status Information Byte Code Status 0x00: Bus error 0x08: START condition transmitted 0x10: Repeated START condition transmitted 0x18: Address + Write bit transmitted, ACK received 0x20: Address + Write bit transmitted, ACK not received 0x28: Data byte transmitted in master mode, ACK received 0x30: Data byte transmitted in master mode, ACK not received 0x38: Arbitration lost in address or data byte 0x40: Address + Read bit transmitted, ACK received 0x48: Address + Read bit transmitted, ACK not received 0x50: Data byte received in master mode, ACK transmitted 0x58: Data byte received in master mode, not ACK transmitted 0x60: Slave address + Write bit received, ACK transmitted 0x68: Arbitration lost in the address as master, slave address + Write bit received, ACK transmitted 0x70: General Call address received, ACK transmitted 0x78: Arbitration lost in the address as master, General Call address received, ACK transmitted 0x80: Data byte received after slave address received, ACK transmitted 0x88: Data byte received after slave address received, not ACK transmitted 0x90: Data byte received after General Call received, ACK transmitted 0x98: Data byte received after General Call received, not ACK transmitted 0xA0: STOP or repeated START condition received in slave mode 0xA8: Slave address + Read bit received, ACK transmitted 0xB0: Arbitration lost in the address as master, slave address + Read bit received, ACK transmitted 0xB8: Data byte transmitted in slave mode, ACK received 0xC0: Data byte transmitted in slave mode, ACK not received 0xC8: The Last byte transmitted in slave mode, ACK received 0xD0: Second Address byte + Write bit transmitted, ACK received

Offset: 0x0010			Register Name: TWI_STAT
Bit	**Read/Write**	**Default/Hex**	**Description**
			0xD8: Second Address byte + Write bit transmitted, ACK not received 0xF8: No relevant status information, INT_FLAG=0 Others: Reserved

9.1.6.6 0x0014 TWI Clock Register (Default Value: 0x0000_0080)

Offset: 0x0014			Register Name: TWI_CCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x1	CLK_DUTY Setting duty cycle of clock as master 0: 50% 1: 40%
6:3	R/W	0x0	CLK_M The TWI SCL output frequency, in master mode, is F1/10: F1 = F0/(CLK_M + 1) Fscl = F1/10 = Fin/(2^CLK_N * (CLK_M + 1)*10) Specially, Fscl = F1/11 when CLK_M=0 and CLK_DUTY=40% due to the delay of SCL sample debounce.
2:0	R/W	0x0	CLK_N The TWI bus is sampled by the TWI at the frequency defined by F0: Fsamp = F0 = Fin/2^CLK_N The TWI SCL output frequency, in master mode, is F1/10: F1 = F0/(CLK_M + 1) Fscl = F1/10 = Fin/(2^CLK_N * (CLK_M + 1)*10) Specially, Fscl = F1/11 when CLK_M=0 and CLK_DUTY=40% due to the delay of SCL sample debounce. For Example: Fin = 24 MHz (APB clock input) For 400 kHz full speed 2-wire, CLK_N = 1, CLK_M = 2 F0 = 24 MHz/2^1 = 12 MHz, F1 = F0/(10*(2+1)) = 0.4 MHz For 100 kHz standard speed 2-wire, CLK_N = 1, CLK_M = 11 F0 = 24 MHz/2^1 = 12 MHz, F1 = F0/(10*(11+1)) = 0.1 MHz

9.1.6.7 0x0018 TWI Soft Reset Register (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: TWI_SRST
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
0	R/WAC	0x0	SOFT_RST Soft Reset Write '1' to this bit to reset the TWI and clear to '0' when completing Soft Reset operation.

9.1.6.8 0x001C TWI Enhance Feature Register (Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: TWI_EFR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
1:0	R/W	0x0	DBN Data Byte Number Follow Read Command Control 00: No data byte can be written after the read command 01: Only 1-byte data can be written after the read command 10: 2-bytes data can be written after the read command 11: 3-bytes data can be written after the read command

9.1.6.9 0x0020 TWI Line Control Register (Default Value: 0x0000_003A)

Offset: 0x0020			Register Name: TWI_LCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:6	/	/	/
5	R	0x1	SCL_STATE Current State of TWI_SCL 0: Low 1: High
4	R	0x1	SDA_STATE Current State of TWI_SDA 0: Low 1: High

Offset: 0x0020			Register Name: TWI_LCR
Bit	**Read/Write**	**Default/Hex**	**Description**
3	R/W	0x1	SCL_CTL TWI_SCL Line State Control Bit When the line control mode is enabled (bit[2] is set), this bit decides the output level of TWI_SCL. 0: Output low level 1: Output high level
2	R/W	0x0	SCL_CTL_EN TWI_SCL Line State Control Enable When this bit is set, the state of TWI_SCL is controlled by the value of bit[3]. 0: Disable TWI_SCL line control mode 1: Enable TWI_SCL line control mode
1	R/W	0x1	SDA_CTL TWI_SDA Line State Control Bit When the line control mode is enabled (bit[0] is set), this bit decides the output level of TWI_SDA. 0: Output low level 1: Output high level
0	R/W	0x0	SDA_CTL_EN TWI_SDA Line State Control Enable When this bit is set, the state of TWI_SDA is controlled by the value of bit[1]. 0: Disable TWI_SDA line control mode 1: Enable TWI_SDA line control mode

9.1.6.10 0x0200 TWI_DRV Control Register (Default Value: 0x00F8_1000)

Offset: 0x0200			Register Name: TWI_DRV_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/WAC	0x0	START_TRAN Start transmission 0: Transmission idle 1: Start transmission Automatically cleared to '0' when finished. If the slave is not responding for the expected status over the time defined by TIMEOUT, the current transmission will stop. All setting formats and data will be loaded from registers and FIFO when the transmission starts.
30	/	/	/
29	R/W	0x0	RESTART_MODE Restart mode 0: RESTART 1: STOP+START Define the TWI_DRV action after sending the register address.
28	R/W	0x0	READ_TRAN_MODE Read transition mode 0: Send slave_id+W 1: Not send slave_id+W Setting this bit to 1 if reading from a slave in which the register width is equal to 0.
27:24	R	0x0	TRAN_RESULT Transition result 000: OK 001: FAIL Other: Reserved

Offset: 0x0200			Register Name: TWI_DRV_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
23:16	R	0xf8	TWI_STA TWI status 0x00: bus error 0x08: START condition transmitted 0x10: Repeated START condition transmitted 0x18: Address + Write bit transmitted, ACK received 0x20: Address + Write bit transmitted, ACK not received 0x28: Data byte transmitted in master mode, ACK received 0x30: Data byte transmitted in master mode, ACK not received 0x38: Arbitration lost in address or data byte 0x40: Address + Read bit transmitted, ACK received 0x48: Address + Read bit transmitted, ACK not received 0x50: Data byte received in master mode, ACK received 0x58: Data byte received in master mode, ACK not received 0x01: Timeout when sending the 9th SCL clock Other: Reserved
15:8	R/W	0x10	TIMEOUT_N Timeout number When sending the 9th clock, assert fail signal when the slave device does not respond after N*F_{SCL} cycles. And the software must do a reset to the TWI_DRV module and send a stop condition to slave.
7:2	/	/	/
1	R/W	0x0	SOFT_RESET Software reset 0: Normal 1: Reset
0	R/W	0x0	TWI_DRV_EN TWI driver enable 0: Module disable 1: Module enable (only use in TWI Master Mode)

9.1.6.11 0x0204 TWI_DRV Transmission Configuration Register (Default Value: 0x0000_0001)

Offset: 0x0204			Register Name: TWI_DRV_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	R/W	0x0	PKT_INTERVAL Define the interval between each packet for PKT_INTERVAL F_{SCL} cycles.
15:0	R/W	0x1	PACKET_CNT The FIFO data is transmitted as PACKET_CNT packets in current format.

9.1.6.12 0x0208 TWI_DRV Slave ID Register (Default Value: 0x0000_0000)

Offset: 0x0208			Register Name: TWI_DRV_SLV
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:9	R/W	0x0	SLV_ID Slave device ID For 7-bit addressing, the bit[7:1] indicates: SLA6, SLA5, SLA4, SLA3, SLA2, SLA1, SLA0 For 10-bit addressing, the bit[7:1] indicates: 1, 1, 1, 1, 0, SLAX[9:8]
8	R/W	0x0	CMD R/W operation to slave device 0: Write 1: Read
7:0	R/W	0x0	SLV_ID_X SLAX[7:0] The low 8 bits for slave device ID with 10-bit addressing.

9.1.6.13 0x020C TWI_DRV Packet Format Register (Default Value: 0x0001_0001)

Offset: 0x020C			Register Name: TWI_DRV_FMT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/

Offset: 0x020C			Register Name: TWI_DRV_FMT
Bit	**Read/Write**	**Default/Hex**	**Description**
23:16	R/W	0x1	ADDR_BYTE How many bytes be sent as slave device reg address 0~255
15:0	R/W	0x1	DATA_BYTE How many bytes be sent/received as data 1~65535

9.1.6.14 0x0210 TWI_DRV Bus Control Register (Default Value: 0x0000_80C0)

Offset: 0x0210			Register Name: TWI_DRV_BUS_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:17	/	/	/
16	W	0x0	CLK_COUNT_MODE Clock count mode 0: scl clock high period count on oscl 1: scl clock high period count on iscl
15	R/W	0x1	CLK_DUTY Setting duty cycle of clock as Master 0: 50% 1: 40%
14:12	R/W	0x0	CLK_N TWI_DRV bus sampling clock F0=24MHz/2^CLK_N
11:8	R/W	0x0	CLK_M TWI_DRV output SCL frequency is F_{SCL}=F1/10=(F0/(CLK_M+1))/10 Specially, Foscl = F1/11 when CLK_M=0 and CLK_DUTY=40% due to the delay of SCL sample debounce.
7	R	0x1	SCL_STA SCL current status
6	R	0x1	SDA_STA SDA current status
5:4	/	/	/
3	R/W	0x0	SCL_MOV SCL manual output value

Offset: 0x0210			Register Name: TWI_DRV_BUS_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R/W	0x0	SDA_MOV SDA manual output value
1	R/W	0x0	SCL_MOE SCL manual output enable
0	R/W	0x0	SDA_MOE SDA manual output enable

9.1.6.15 0x0214 TWI_DRV Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x0214			Register Name: TWI_DRV_INT_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:18	/	/	/
19	R/W	0x0	RX_REQ_INT_EN If set, an interrupt is sent when RX_REQ_PD sets.
18	R/W	0x0	TX_REQ_INT_EN If set, an interrupt is sent when TX_REQ_PD sets.
17	R/W	0x0	TRAN_ERR_INT_EN If set, an interrupt is sent when TRAN_ERR_PD sets.
16	R/W	0x0	TRAN_COM_INT_EN If set, an interrupt is sent when TRAN_COM_PD sets.
15:4	/	/	/
3	R/W1C	0x0	RX_REQ_PD Set when the data byte number in RECV_FIFO reaches RX_TRIG.
2	R/W1C	0x0	TX_REQ_PD Set when there is no less than DMA_TX_TRIG empty byte number in SEND_FIFO.
1	R/W1C	0x0	TRAN_ERR_PD Packet transmission failure pending
0	R/W1C	0x0	TRAN_COM_PD Packet transmission completion pending

9.1.6.16 0x0218 TWI_DRV DMA Configure Register (Default Value: 0x0010_0010)

Offset: 0x0218			Register Name: TWI_DRV_DMA_CFG
Bit	**Read/Write**	**Default/Hex**	**Description**
31:25	/	/	/
24	R/W	0x0	DMA_RX_EN DMA RX Enable
23:22	/	/	/
21:16	R/W	0x10	RX_TRIG RX trigger When DMA_RX_EN is set, send DMA RX Req when the data byte number in RECV_FIFO reaches RX_TRIG, or the read transmission is completed, the data of RECV_FIFO does not reach RX_TRIG but as long as the RECV_FIFO is not empty.
15:9	/	/	/
8	R/W	0x0	DMA_TX_EN DMA TX Enable
7:6	/	/	/
5:0	R/W	0x10	TX_TRIG TX trigger When DMA_TX_EN is set, send DMA TX Req when the space of SEND_FIFO (FIFO Level – data volume) reaches TX_TRIG.

9.1.6.17 0x021C TWI_DRV FIFO Content Register (Default Value: 0x0000_0000)

Offset: 0x021C			Register Name: TWI_DRV_FIFO_CON
Bit	**Read/Write**	**Default/Hex**	**Description**
31:23	/	/	/
22	R/WAC	0x0	RECV_FIFO_CLEAR Set this bit to clear RECV_FIFO pointer, and this bit is cleared automatically.
21:16	R	0x0	RECV_FIFO_CONTENT The number of data in RECV_FIFO
15:7	/	/	/
6	R/WAC	0x0	SEND_FIFO_CLEAR Set this bit to clear SEND_FIFO pointer, and this bit is cleared automatically.

Offset: 0x021C			Register Name: TWI_DRV_FIFO_CON
Bit	**Read/Write**	**Default/Hex**	**Description**
5:0	R	0x0	SEND_FIFO_CONTENT The number of data in SEND_FIFO

9.1.6.18 0x0300 TWI_DRV Send Data FIFO Access Register (Default Value: 0x0000_0000)

Offset: 0x0300			Register Name: TWI_DRV_SEND_FIFO_ACC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	W	0x0	SEND_DATA_FIFO Address of a 32x8 SEND_FIFO, which stores reg address and data sending to the slave device.

9.1.6.19 0x0304 TWI_DRV Receive Data FIFO Access Register (Default Value: 0x0000_0000)

Offset: 0x0304			Register Name: TWI_DRV_RECV_FIFO_ACC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R	0x0	RECV_DATA_FIFO Address of a 32x8 RECV_FIFO, which stores data received from the slave device.

9.2 UART

9.2.1 Overview

The universal asynchronous receiver transmitter (UART) provides an asynchronous serial communication with external devices, modem (data carrier equipment, DCE). It performs serial-to-parallel conversion on the data received from peripherals and transmits the converted data to the internal bus. It also performs parallel-to-serial conversion on the data that is transmitted to peripherals.

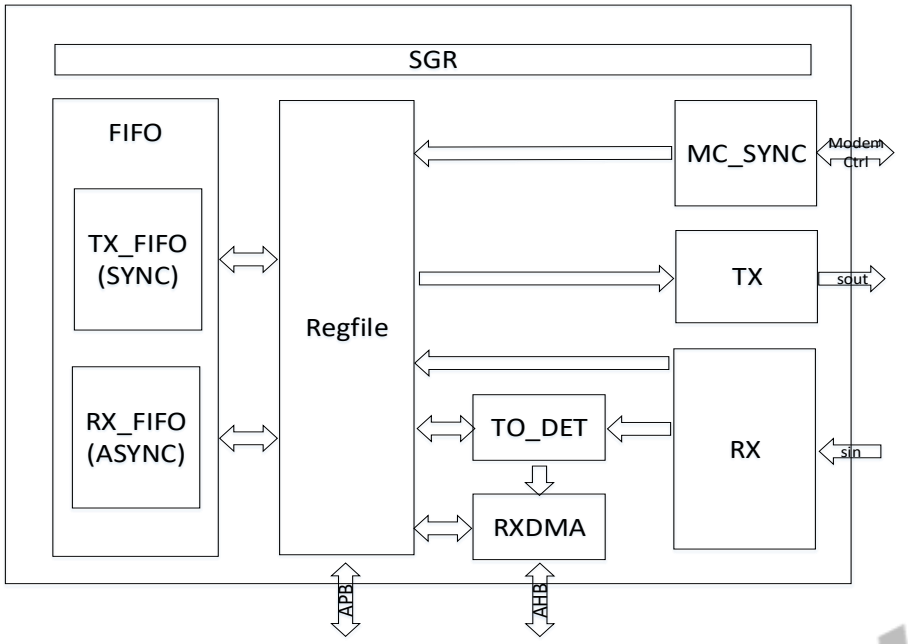
The UART has the following features:

- Compatible with industry-standard 16450/16550 UARTs
- Supports IrDA-compatible slow infrared (SIR) format
- Two separate FIFOs: one is RX FIFO, and the other is TX FIFO
  - Each of them is 64 bytes (For UART0)
  - Each of them is 256 bytes (For UART1, UART2, UART3, UART4, and UART5)
- The working reference clock is from the APB bus clock
  - Speed up to 4 Mbit/s with 64 MHz APB clock
  - Speed up to 1.5 Mbit/s with 24 MHz APB clock
- 5 to 8 data bits for RS-232 characters, or 9 bits RS-485 format
- 1, 1.5 or 2 stop bits
- Programmable parity (even, odd, or no parity)
- Supports TX/RX DMA slave controller interface
- Supports software/hardware flow control
- Supports RX DMA Master interface (Only for UART1)
- Supports auto-flow by using CTS & RTS (Only for UART1/2/3)

9.2.2 Block Diagram

Figure 9-9 shows a block diagram of the UART.

Figure 9-9 UART Block Diagram



9.2.3 Functional Description

9.2.3.1 External Signals

The following table describes the external signals of UART.

Table 9-4 UART External Signals

Signal	Description	Type
UART0-TX	UART0 Data Transmit	O
UART0-RX	UART0 Data Receive	I
UART1-TX	UART1 Data Transmit	O
UART1-RX	UART1 Data Receive	I
UART1-CTS	UART1 Data Clear to Send	I
UART1-RTS	UART1 Data Request to Send	O
UART2-TX	UART2 Data Transmit	O
UART2-RX	UART2 Data Receive	I
UART2-CTS	UART2 Data Clear to Send	I
UART2-RTS	UART2 Data Request to Send	O
UART3-TX	UART3 Data Transmit	O
UART3-RX	UART3 Data Receive	I

Signal	Description	Type
UART3-CTS	UART3 Data Clear to Send	I
UART3-RTS	UART3 Data Request to Send	O
UART4-TX	UART4 Data Transmit	O
UART4-RX	UART4 Data Receive	I
UART5-TX	UART5 Data Transmit	O
UART5-RX	UART5 Data Receive	I

9.2.3.2 Clock Sources

The following table describes the clock sources of UART.

Table 9-5 UART Clock Sources

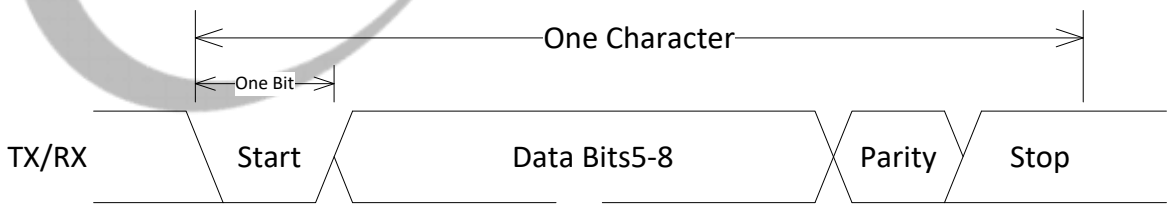
Clock Sources	Description
APB1 Bus	UART clock source. Refer to CCU for details on APB1.

9.2.3.3 Typical Applications and Timing Diagram

UART Serial Data Format

The following figure shows the UART serial data format. The start bit, data bit, parity bit, and stop bit can be configured.

Figure 9-10 UART Serial Data Format



Using UART for RTS/CTS Autoflow Control

Figure 9-11 shows the typical application diagram for RTS/CTS autoflow control. Figure 9-12 shows the data format of the RTS/CTS autoflow control.

Figure 9-11 Application Diagram for RTS/CTS Autoflow Control

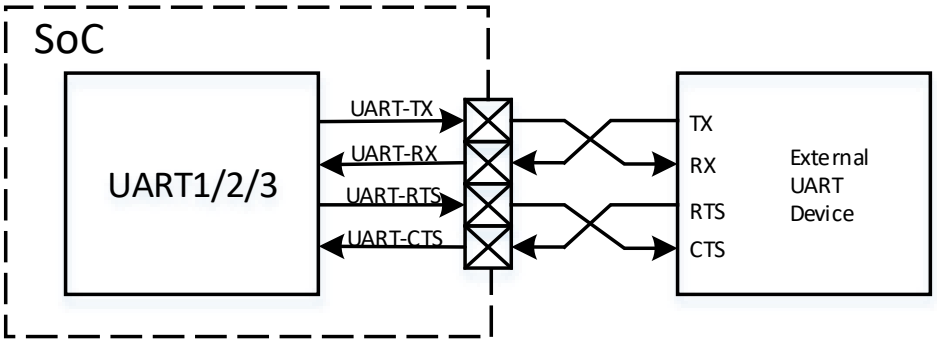
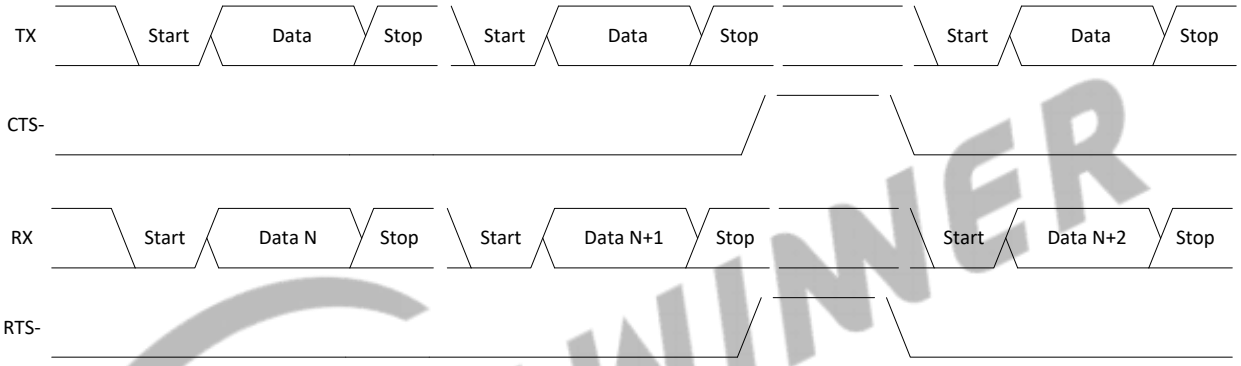


Figure 9-12 RTS/CTS Autoflow Control Data Format



Using UART for Serial IrDA

Figure 9-13 shows the application diagram for the IrDA transceiver. Figure 9-14 shows the data format of the serial IrDA.

Figure 9-13 Application Diagram for IrDA Transceiver

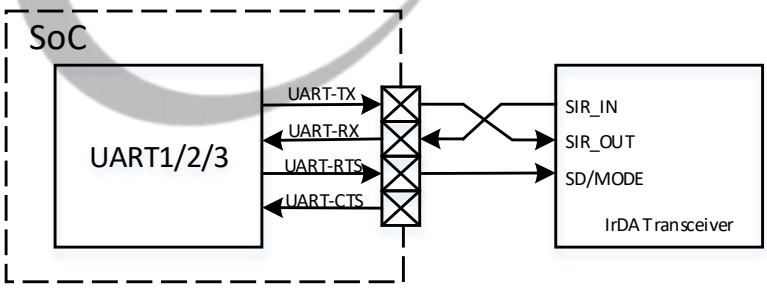
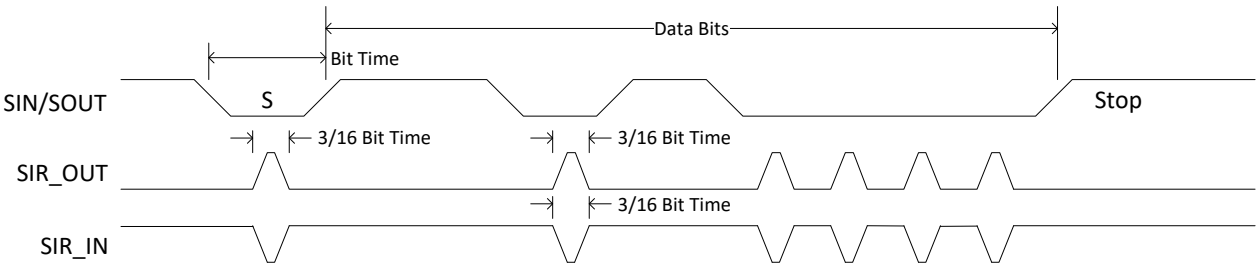


Figure 9-14 Serial IrDA Data Format



Using UART for RS-485

Figure 9-15 shows the application diagram for the RS-485 transceiver. Figure 9-16 shows the data format of the RS-485.

Figure 9-15 Application Diagram for RS-485 Transceiver

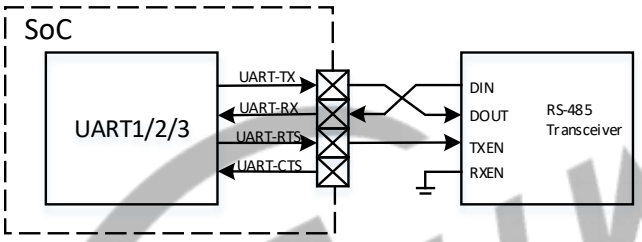
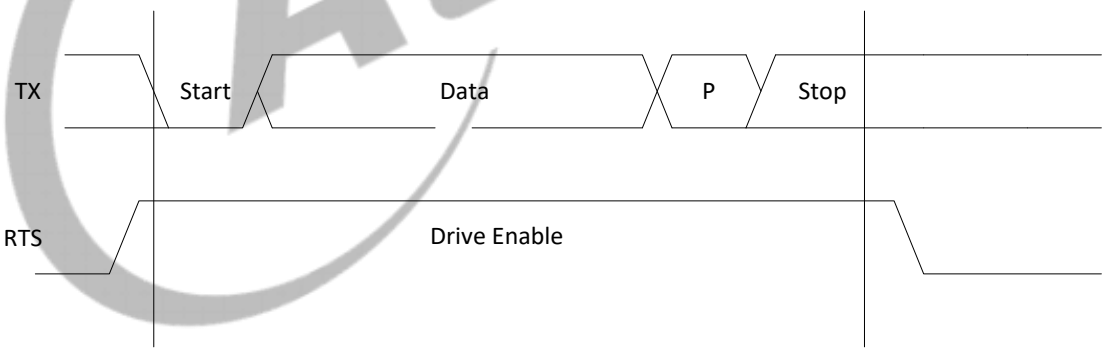


Figure 9-16 RS-485 Data Format



9.2.3.4 UART Operating Mode

Data Frame Format

The UART_LCR register can set the basic parameter of a data frame: data width (5 to 8 bits), stop bit number (1/1.5/2), parity type.

A frame transfer of the UART includes the start signal, data signal, parity bit, and stop signal. The LSB is transmitted first.

- Start signal (start bit): It is the start flag of a data frame. According to the UART protocol, the low level of the TXD signal indicates the start of a data frame. When the UART transmits data, the level needs to hold high.
- Data signal (data bit): The data bit width can be configured as 5-bit, 6-bit, 7-bit, and 8-bit through different applications.
- Parity bit: It is a 1-bit error correction signal. Parity bit includes odd parity, even parity. The UART can enable and disable the parity bit by setting the **UART_LCR** register.
- Stop Signal (stop bit): It is the stop bit of a data frame. The stop bit can be set to 1-bit, 1.5-bit, and 2-bit by the **UART_LCR** register. The high level of the TXD signal indicates the end of a data frame.

Baud and Error Rates

The baud rate is calculated as follows: Baud rate = SCLK/(16 * divisor).

The SCLK is usually APB1 and can be set in section 3.3 "CCU".

The divisor is frequency divider of UART. The frequency divider has 16-bit, the low 8-bit is in the **UART_DLL** register, the high 8-bit is in the **UART_DLH** register.

The relationship between the different UART mode and the error rate is as follows.

Figure 9-17 UART Mode Baud and Error Rates

Clock source	Divisor	Baud rate	Over sampling	Error (%)
24000000	5000	300	16	0
24000000	2500	600	16	0
24000000	1250	1200	16	0
24000000	625	2400	16	0
24000000	313	4800	16	-0.16
24000000	156	9600	16	0.16
24000000	78	19200	16	0.16
24000000	39	38400	16	0.16
24000000	26	57600	16	0.16
24000000	13	115200	16	0.16
48000000	13	230400	16	0.16
64000000	7	576000	16	-0.794
75000000	5	921600	16	1.725

Clock source	Divisor	Baud rate	Over sampling	Error (%)
48000000	3	1000000	16	0
24000000	1	1500000	16	0
48000000	1	3000000	16	0
64000000	1	4000000	16	0

Figure 9-18 IrDA Mode Baud and Error Rates

Clock source	Divisor	Baud rate	Encoding	Error (%)
24000000	5000	300	3/16	0
24000000	2500	600	3/16	0
24000000	1250	1200	3/16	0
24000000	625	2400	3/16	0
24000000	313	4800	3/16	-0.16
24000000	156	9600	3/16	0.16
24000000	78	19200	3/16	0.16
24000000	39	38400	3/16	0.16
24000000	26	57600	3/16	0.16
24000000	13	115200	3/16	0.16

Figure 9-19 RS485 Mode Baud and Error Rates

Clock source	Divisor	Baud rate	Encoding	Error (%)
24000000	5000	300	16	0
24000000	2500	600	16	0
24000000	1250	1200	16	0
24000000	625	2400	16	0
24000000	313	4800	16	-0.16
24000000	156	9600	16	0.16
24000000	78	19200	16	0.16
24000000	39	38400	16	0.16
24000000	26	57600	16	0.16
24000000	13	115200	16	0.16

DLAB Definition

The DLAB control bit (UART_LCR[7]) is the access control bit of the divisor Latch register.

If DLAB is 0, then the 0x00 offset address is the UART_RBR/UART_THR (RX/TX FIFO) register, and the 0x04 offset address is the UART_IER register.

If DLAB is 1, then the 0x00 offset address is the UART_DLL register, and the 0x04 offset address is the UART_DLH register.

When the UART initials, the divisor needs to be set. That is, writing 1 to DLAB can access the UART_DLL and UART_DLH register, after finished the configuration, writing 0 to DLAB can access the UART_RBR/UART_THR register.

CHCFG_AT_BUSY Definition

The function of the CHCFG_AT_BUSY (UART_HALT[1]) and CHANGE_UPDATE (UART_HALT[2]) are as follows.

CHCFG_AT_BUSY: Enable the bit, the software can also set the UART controller when UART is busy, such as the UART_LCR, UART_DLH, UART_DLL register.

CHANGE_UPDATE: If CHCFG_AT_BUSY is enabled, and CHANGE_UPDATE is written to 1, the configuration of the UART controller can be updated. After completed the update, the bit is cleared to 0 automatically.

Setting divisor performs the following steps:

Step 1 Write 1 to CHCFG_AT_BUSY to enable “configure at busy”.

Step 2 Write 1 to DLAB (UART_LCR[7]) and set the UART_DLH and UART_DLL registers.

Step 3 Write 1 to CHANGE_UPDATE to update the configuration. The bit is cleared to 0 automatically after completing the update.

UART Busy Flag

The UART_USR[0] is a busy flag of the UART controller.

When the TX transmits data, or the RX receives data, or the TX FIFO is not empty, or the RX FIFO is not empty, then the busy flag bit can be set to 1 by hardware, which indicates the UART controller is busy.

9.2.4 Programming Guidelines

The following takes the UART module in the CPUX domain as an example.

9.2.4.1 Initialization

Step 1 System Initialization

- Configure **APB1_CFG_REG** in the CCU module to set the APB1 bus clock (The clock is 24MHz by default).
- Set **UART_BGR_REG**[UARTx_GATING] to 1 to enable the module clock, and set **UART_BGR_REG**[UARTx_RST] to 1 to de-assert the module.

Step 2 UART Controller Initialization

- IO configuration: Configure GPIO multiplex as UART function, and set UART pins to internal pull-up mode (For detail, see the description in section 9.7 "GPIO").
- Baud-rate configuration:
  - Set UART baud-rate (refer to section 9.2.3.4);
  - Write **UART_FCR**[FIFOE] to 1 to enable TX/RX FIFO;
  - Write **UART_HALT**[HALT_TX] to 1 to disable TX transfer;
  - Set **UART_LCR**[DLAB] to 1, remain default configuration for other bits; set 0x00 offset address to the **UART_DLL** register, set 0x04 offset address to the **UART_DLH** register;
  - Write the high 8-bit of divisor to the **UART_DLH** register, and write the low 8-bit of divisor to the **UART_DLL** register;
  - Set **UART_LCR**[DLAB] to 0, remain default configuration for other bits; set 0x00 offset address to the **UART_RBR**/**UART_THR** register, set 0x04 offset address to the **UART_IER** register;
  - Set **UART_HALT**[HALT_TX] to 0 to enable TX transfer.

Step 3 Controller Parameter Configuration

- Set data width, stop bits, and even/odd parity type by writing the **UART_LCR** register.
- Reset, enable FIFO and set FIFO trigger condition by writing the **UART_FCR** register.
- Set the flow control parameter by writing the **UART_MCR** register.

Step 4 Interrupt Configuration

- Configure UART interrupt vector number to request UART interrupt (Refer to section 3.8 "GIC" module for interrupt vector number).
- In DMA mode, write **UART_IER** to 0 to disable interrupt; write **UART_HSK**[Handshake configuration] to 0xE5 to set DMA handshake mode; write **UART_FCR**[DMAM] to 1 to set DMA transmission/reception mode; set DMA parameter and request DMA interrupt according to DMA configuration process.

- In Interrupt mode, configure **UART_IER** to enable the corresponding interrupt according to requirements: such as transmit (TX) interrupt, receive (RX) interrupt, receive line status interrupt, RS48 interrupt, etc. (Here TX/RX interrupt is usually used).

9.2.4.2 Transferring/Receiving Data in Query Mode

Data transfer

Step 1 Write data to **UART_THR** to start data transfer.

Step 2 Check TX_FIFO status by reading **UART_USR**[TFNF]. If the bit is 1, data can continue to be written; if the bit is 0, wait for data transfer, and data cannot continue to write until FIFO is not full.

Data receive

Step 1 Check RX_FIFO status by reading **UART_USR**[RFNE].

Step 2 Read data from **UART_RBR** if RX_FIFO is not empty.

Step 3 If **UART_USR**[RFNE] is 0, data is received completely.

9.2.4.3 Transferring/Receiving Data in Interrupt Mode

Data transfer

Step 1 Set **UART_IER**[ETBEI] to 1 to enable the *UART transmission interrupt*.

Step 2 Write the data to be transmitted to **UART_THR**.

Step 3 When the data of TX_FIFO meets trigger condition (such as FIFO/2, FIFO/4), the UART transfer interrupt is generated.

Step 4 Check **UART_USR**[TFE] and determine whether TX_FIFO is empty. If **UART_USR**[TFE] is 1, it indicates that the data in TX_FIFO is transmitted completely.

Step 5 Clear **UART_IER**[ETBEI] to 0 to disable transfer interrupt.

Data receive

Step 1 Set **UART_IER**[ERBFI] to 1 to enable the *UART reception interrupt*.

Step 2 When the received data from RX_FIFO meets trigger condition (such as FIFO/2, FIFO/4), the UART receive interrupt is generated.

Step 3 Read data from **UART_RBR**.

Step 4 Check RX_FIFO status by reading **UART_USR**[RFNE] and determine whether to read data. If the bit is 1, continue to read data from **UART_RBR** until **UART_USR**[RFNE] is cleared to 0, which indicates data is received completely.

9.2.4.4 Transferring/Receiving Data in DMA Mode

Data transfer

Step 1 Configure the UART DMA interrupt according to the initialization process.

Step 2 Configure DMA data channel, including the transfer source address, the transfer destination address, the number of data to be transferred, and the transfer type, and so on. (For details, see section 3.9 "DMAC").

Step 3 Enable the DMA transfer function of the UART by setting the register of the DMA module.

Step 4 Determine whether UART data is transferred completely based on the DMA status. If all data is transferred completely, disable the DMA transfer function of the UART.

Data receive

Step 1 Configure DMA data channel, including the transfer source address, the transfer destination address, the number of data to be transferred, and the transfer type, and so on. (For details, see section 3.9 "DMAC").

Step 2 Enable the DMA receive function of the UART by setting the register of the DMA module.

Step 3 Determine whether UART data is received completely based on the DMA status. If all data is received completely, disable the DMA receive function of the UART.

9.2.5 Register List

Module Name	Base Address
UART0	0x02500000
UART1	0x02500400
UART2	0x02500800
UART3	0x02500C00
UART4	0x02501000
UART5	0x02501400

Register Name	Offset	Description
UART_RBR	0x0000	UART Receive Buffer Register
UART_THR	0x0000	UART Transmit Holding Register
UART_DLL	0x0000	UART Divisor Latch Low Register
UART_DLH	0x0004	UART Divisor Latch High Register
UART_IER	0x0004	UART Interrupt Enable Register
UART_IIR	0x0008	UART Interrupt Identity Register
UART_FCR	0x0008	UART FIFO Control Register
UART_LCR	0x000C	UART Line Control Register
UART_MCR	0x0010	UART Modem Control Register
UART_LSR	0x0014	UART Line Status Register
UART_MSR	0x0018	UART Modem Status Register
UART_SCH	0x001C	UART Scratch Register
UART_USR	0x007C	UART Status Register
UART_TFL	0x0080	UART Transmit FIFO Level Register
UART_RFL	0x0084	UART Receive FIFO Level Register
UART_HSK	0x0088	UART DMA Handshake Configuration Register
UART_DMA_REQ_EN	0x008C	UART DMA Request Enable Register
UART_HALT	0x00A4	UART Halt TX Register
UART_DBG_DLL	0x00B0	UART Debug DLL Register
UART_DBG_DLH	0x00B4	UART Debug DLH Register
UART_A_FCC	0x00F0	UART FIFO Clock Control Register
UART_A_RXDMA_CTRL	0x0100	UART RXDMA Control Register
UART_A_RXDMA_STR	0x0104	UART RXDMA Start Register
UART_A_RXDMA_STA	0x0108	UART RXDMA Status Register
UART_A_RXDMA_LMT	0x010C	UART RXDMA Limit Register
UART_A_RXDMA_SADDRL	0x0110	UART RXDMA Buffer Start Address Low Register
UART_A_RXDMA_SADDRH	0x0114	UART RXDMA Buffer Start Address High Register
UART_A_RXDMA_BL	0x0118	UART RXDMA Buffer Length Register
UART_A_RXDMA_IE	0x0120	UART RXDMA Interrupt Enable Register
UART_A_RXDMA_IS	0x0124	UART RXDMA Interrupt Status Register
UART_A_RXDMA_WADDRL	0x0128	UART RXDMA Write Address Low Register
UART_A_RXDMA_WADDRH	0x012C	UART RXDMA Write Address high Register

Register Name	Offset	Description
UART_A_RXDMA_RADDRL	0x0130	UART RXDMA Read Address Low Register
UART_A_RXDMA_RADDRH	0x0134	UART RXDMA Read Address high Register
UART_A_RXDMA_DCNT	0x0138	UART RADMA Data Count Register

9.2.6 Register Description

9.2.6.1 0x0000 UART Receiver Buffer Register (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: UART_RBR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R	0x0	RBR Receiver Buffer Register Data byte received on the serial input port (sin) in UART mode, or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in *UART_LCR* is set. If in FIFO mode and FIFOs are enabled (The *UART_FCR*[0] is set to 1), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register can not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data are lost and an overrun error occurs.

9.2.6.2 0x0000 UART Transmit Holding Register (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: UART_THR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/

Offset: 0x0000			Register Name: UART_THR
Bit	**Read/Write**	**Default/Hex**	**Description**
7:0	W	0x0	THR Transmit Holding Register Data is transmitted on the serial output port (SOUT) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the *UART_THR* when the THRE bit (*UART_LSR*[5]) is set. If in FIFO mode and FIFOs are enabled (*UART_FCR*[0] = 1) and THRE is set, the 16 number of characters data may be written to the *UART_THR* before the FIFO is full. When the FIFO is full, any written data results in the written data being lost.

9.2.6.3 0x0000 UART Divisor Latch Low Register (Default Value: 0x0000_0000)

Offset: 0x0000			Register Name: UART_DLL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0x0	DLL Divisor Latch Low Lower 8 bits of 16 bits, read/write, Divisor Latch Register contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (*UART_LCR*[7]) is set and the UART is not busy (*UART_USR*[0] is 0). The output baud rate is equal to the serial clock (SCLK) frequency divided by sixteen times the value of the baud rate divisor, as follows: **baud rate = (serial clock freq)/(16 * divisor)**. Note that when the Divisor Latch Registers (*UART_DLL* and *UART_DLH*) are set to 0, the baud clock is disabled and no serial communications occur. Also, once the DLL is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.

9.2.6.4 0x0004 UART Divisor Latch High Register (Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: UART_DLH
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/

Offset: 0x0004			Register Name: UART_DLH
Bit	**Read/Write**	**Default/Hex**	**Description**
7:0	R/W	0x0	DLH Divisor Latch High Upper 8 bits of 16 bits, read/write, Divisor Latch Register contains the baud rate divisor for the UART. This register may only be accessed when the DLAB bit (*UART_LCR*[7]) is set and the UART is not busy (*UART_USR*[0] is 0). The output baud rate is equal to the serial clock (SCLK) frequency divided by sixteen times the value of the baud rate divisor, as follows: **baud rate = (serial clock freq)/(16 * divisor)**. Note that when the Divisor Latch Registers (*UART_DLL* and *UART_DLH*) is set to 0, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest UART clock should be allowed to pass before transmitting or receiving data.

9.2.6.5 0x0004 UART Interrupt Enable Register (Default Value: 0x0000_0000)

Offset: 0x0004			Register Name: UART_IER
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	PTIME Programmable THRE Interrupt Mode Enable This is used to enable/disable the generation of THRE Interrupt. 0: Disable 1: Enable
6:5	/	/	/
4	R/W	0x0	RS485_INT_EN RS485 Interrupt Enable 0: Disable 1: Enable
3	R/W	0x0	EDSSI Enable Modem Status Interrupt This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. 0: Disable 1: Enable

Offset: 0x0004			Register Name: UART_IER
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R/W	0x0	ELSI Enable Receiver Line Status Interrupt This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. 0: Disable 1: Enable
1	R/W	0x0	ETBEI Enable Transmit Holding Register Empty Interrupt This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third-highest priority interrupt. 0: Disable 1: Enable
0	R/W	0x0	ERBFI Enable Received Data Available Interrupt This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second-highest priority interrupt. 0: Disable 1: Enable

9.2.6.6 0x0008 UART Interrupt Identity Register (Default Value: 0x0000_0001)

Offset: 0x0008			Register Name: UART_IIR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:6	R	0x0	FEFLAG FIFOs Enable Flag This is used to indicate whether the FIFOs are enabled or disabled. 00: Disable 11: Enable
5:4	/	/	/

Offset: 0x0008			Register Name: UART_IIR
Bit	**Read/Write**	**Default/Hex**	**Description**
3:0	R	0x1	IID Interrupt ID This indicates the highest priority pending interrupt which can be one of the following types. 0000: modem status 0001: no interrupt pending 0010: THR empty 0011: RS485 Interrupt 0100: received data available 0110: receiver line status 0111: busy detect 1100: character timeout The bit 3 indicates an interrupt can only occur when the FIFOs are enabled and used to distinguish a Character Timeout condition interrupt.

Interrupt ID	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset
0001	-	None	None	-
0110	Highest	Receiver line status	Overrun/parity/framing errors or break interrupt	Reading the line status register
0011	Second	RS485 Interrupt	In RS485 mode, receives address data and match setting address	Writes 1 to addr flag to reset
0100	Third	Received data available	Receiver data available (non-FIFO mode or FIFOs disabled) or RCVR FIFO trigger level reached (FIFO mode and FIFOs enabled)	Reading the receiver buffer register (non-FIFO mode or FIFOs disabled) or the FIFO drops below the trigger level (FIFO mode and FIFOs enabled)
1100	Fourth	Character timeout indication	No characters in or out of the RCVR FIFO during the last 4 character times and there is at least 1character in it during This time	Reading the receiver buffer register

Interrupt ID	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset
0010	Fifth	Transmit holding register empty	Transmitter holding register empty (Program THRE mode disabled) or XMIT FIFO at or below threshold (Program THRE mode enabled)	Reading the IIR register (if the source of interrupt); or, writing into THR (FIFOs or THRE mode not selected or disabled) or XMIT FIFO above threshold (FIFOs and THRE mode selected and enabled).
0000	Sixth	Modem status	Clear to send or data set ready or ring indicator or data carrier detect. Note that if the autoflow control mode is enabled, a change in CTS (that is, DCTS set) does not cause an interrupt.	Reading the Modem status register
0111	Seventh	Busy detect indication	UART_16550_COMPATIBLE = NO and master has tried to write to the Line Control Register while the UART is busy (USR[0] is set to one).	Reading the UART status register

9.2.6.7 0x0008 UART FIFO Control Register (Default Value: 0x0000_0000)

Offset: 0x0008			Register Name: UART_FCR
Bit	Read/Write	Default/Hex	Description
31:8	/	/	/
7:6	W	0x0	RT RCVR Trigger This is used to select the trigger level in the receiver FIFO when the Received Data Available Interrupt is generated. In the autoflow control mode, it is used to determine when the rts_n signal is de-asserted. It also determines when the dma_rx_req_n signal is asserted in certain modes of operation. 00: 1 character in the FIFO 01: FIFO ¼ full 10: FIFO ½ full 11: FIFO-2 less than full

Offset: 0x0008			Register Name: UART_FCR
Bit	**Read/Write**	**Default/Hex**	**Description**
5:4	W	0x0	TFT TX Empty Trigger This is used to select the empty threshold level when the THRE Interrupts are generated and the mode is active. It also determines when the dma_tx_req_n signal is asserted in certain modes of operation. 00: FIFO empty 01: 2 characters in the FIFO 10: FIFO ¼ full 11: FIFO ½ full
3	W	0x0	DMAM DMA Mode 0: Mode 0 In this mode, when the PTE in UART_HALT is high and TX FIFO is enabled, the TX DMA request will be set when the TFL in UART_TFL is less than or equal to FIFO Trigger Level (otherwise it will be cleared). When the PTE is high and TX FIFO is disabled, the TX DMA request will be set only if the THR in UART_THR is empty. If the PTE is low, the TX DMA request will be set only if the TX FIFO (TX FIFO enabled) or THR (TX FIFO disabled) is empty. When the DMA_PTE_RX in UART_HALT is high and RX FIFO is enabled, the RX DRQ will be set only if the RFL in UART_RFL is equal to or more than FIFO Trigger Level, otherwise, it will be cleared. 1: Mode 1 In this mode, TX FIFO should be enabled. If the PTE in *UART_HALT* is high, the TX DMA request will be set when the TFL in *UART_TFL* is less than or equal to FIFO Trigger Level; If the PTE is low, the TX DMA request will be set when TX FIFO is empty. Once the request is set, it is cleared only when TX FIFO is full. If the RFL in UART_RFL is equal to or more than FIFO Trigger Level or there is a character timeout, the RX DRQ will be set; Once the RX DRQ is set, it is cleared only when RX FIFO (RX FIFO enabled) or RBR (RX FIFO disabled) is empty.
2	W	0x0	XFIFOR XMIT FIFO Reset The bit resets the control part of the transfer FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request. It is 'self-cleared'. It is not necessary to clear this bit.

Offset: 0x0008			Register Name: UART_FCR
Bit	**Read/Write**	**Default/Hex**	**Description**
1	W	0x0	RFIFOR RCVR FIFO Reset The bit resets the control part of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request. It is 'self-cleared'. It is not necessary to clear this bit.
0	W	0x0	FIFOE Enable FIFOs The bit enables/disables the transmitting (XMIT) and receiving (RCVR) FIFOs. Whenever the value of this bit is changed, both the XMIT and RCVR controller part of FIFOs is reset.

9.2.6.8 0x000C UART Line Control Register (Default Value: 0x0000_0000)

Offset: 0x000C			Register Name: UART_LCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	DLAB Divisor Latch Access Bit It is writable only when UART is not busy (UART_USR[0] is 0) and always readable. This bit is used to enable the reading and writing of the Divisor Latch register (UART_DLL and UART_DLH) to set the baud rate of the UART. This bit must be cleared after the initial baud rate setup in order to access other registers. 0: Select RX Buffer Register (UART_RBR)/TX Holding Register (UART_THR) and Interrupt Enable Register (UART_IER) 1: Select Divisor Latch LS Register (UART_DLL) and Divisor Latch MS Register (UART_DLM)
6	R/W	0x0	BC Break Control Bit This is used to cause a break condition to be transmitted to the receiving device. If set to 0, the serial output is forced to the spacing (logic 0) state. When not in Loopback mode, as determined by UART_MCR[4], the SOUT line is forced low until the Break bit is cleared. If SIR_MODE is enabled and active (UART_MCR[6] is set to 1), the sir_out_n line is continuously pulsed. When in Loopback mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low.

Offset: 0x000C			Register Name: UART_LCR
Bit	**Read/Write**	**Default/Hex**	**Description**
5:4	R/W	0x0	EPS Even Parity Select It is writable only when UART is not busy (UART_USR[0] is 0). This is used to select the even and odd parity when the PEN is enabled (the UART_LCR[3] is set to 1). Setting the UART_LCR[5] is unset to reverse the LCR[4]. 00: Odd Parity 01: Even Parity 1X: Reverse LCR[4] In RS485 mode, it is the 9th bit--address bit. 11: 9th bit = 0, indicates that this is a data byte. 10: 9th bit = 1, indicates that this is an address byte. **Note: When using this function, the PEN(UART_LCR[3]) must set to 1.**
3	R/W	0x0	PEN Parity Enable It is writable only when UART is not busy (UART_USR[0] is 0) and always readable. This bit is used to enable and disable parity generation and detection in transmitted and received serial characters respectively. 0: Parity disabled 1: Parity enabled
2	R/W	0x0	STOP Number of stop bits It is writable only when UART is not busy (UART_USR[0] is 0) and always readable. This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to 0, one stop bit is transmitted in the serial data. If set to 1 and the data bits are set to 5 (UART_LCR[1:0] is 0), one and a half stop bit is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit. 0: 1 stop bit 1: 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit

Offset: 0x000C			Register Name: UART_LCR
Bit	**Read/Write**	**Default/Hex**	**Description**
1:0	R/W	0x0	DLS Data Length Select It is writable only when UART is not busy (UART_USR[0] is 0) and always readable. This is used to select the count of bits in a transmitted or received frame. 00: 5 bits 01: 6 bits 10: 7 bits 11: 8 bits

9.2.6.9 0x0010 UART Modem Control Register (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: UART_MCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:6	R/W	0x0	UART_FUNCTION Select IrDA or RS485 00: UART Mode 01: IrDA SIR Mode 10: RS485 Mode 11: Reserved
5	R/W	0x0	AFCE Auto Flow Control Enable When FIFOs are enabled and the AFCE bit is set, the AutoFlow Control is enabled. 0: Auto Flow Control mode disabled 1: Auto Flow Control mode enabled

Offset: 0x0010			Register Name: UART_MCR
Bit	**Read/Write**	**Default/Hex**	**Description**
4	R/W	0x0	LOOP Loop Back Mode 0: Normal Mode 1: Loop Back Mode This is used to put the UART into a diagnostic mode for test purposes. If operating in UART mode (SIR_MODE != Enabled or not active, UART_MCR[6] is set to 0), the data on the SOUT line is held high, while serial data output is looped back to the sin line, internally. In this mode, all the interrupts are fully functional. Also, in loopback mode, the modem control inputs (dsr_n, cts_n, ri_n, dcd_n) are disconnected and the modem control outputs (dtr_n, rts_n, out1_n, out2_n) are looped back to the inputs, internally. If operating in infrared mode (SIR_MODE == Enabled AND active, UART_MCR[6] is set to 1), the data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line.
3:2	/	/	/
1	R/W	0x0	RTS Request to Send This is used to directly control the Request to Send (rts_n) output. The RTS (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. When Auto RTS Flow Control is not enabled (*UART_MCR[5]* is set to 0), the rts_n signal is set low by programming UART_MCR[1] (RTS) to a high. In Auto Flow Control, AFCE_MODE == Enabled and active (*UART_MCR[5]* is set to 1) and FIFOs enable (UART_FCR[0] is set to 1), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). The rts_n signal is de-asserted when UART_MCR[1] is set low. 0: rts_n de-asserted (logic 1) 1: rts_n asserted (logic 0) Note that in Loopback mode (UART_MCR[4] is set to 1), the rts_n output is held inactive high while the value of this location is internally looped back to an input.

Offset: 0x0010			Register Name: UART_MCR
Bit	**Read/Write**	**Default/Hex**	**Description**
0	R/W	0x0	DTR Data Terminal Ready This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n. 0: dtr_n de-asserted (logic 1) 1: dtr_n asserted (logic 0) The DTR output is used to inform the modem or data set that the UART is ready to establish communications. Note that in Loopback mode (UART_MCR[4] is set to 1), the dtr_n output is held inactive high while the value of this location is internally looped back to an input.

9.2.6.10 0x0014 UART Line Status Register (Default Value: 0x0000_0060)

Offset:0x0014			Register Name: UART_LSR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R	0x0	FIFOERR RX Data Error in FIFO When FIFOs are disabled, this bit is always 0. When FIFOs are enabled, this bit is set to “1” when there is at least one PE, FE, or BI in the RX FIFO. It is cleared by reading from the UART_LSR register, there are no subsequent errors in the FIFO.
6	R	0x1	TEMT Transmitter Empty If the FIFOs are disabled, this bit is set to "1" whenever the TX Holding Register (UART_THR) and the TX Shift Register are empty. If the FIFOs are enabled, this bit is set whenever the TX FIFO and the TX Shift Register are empty. In both cases, this bit is cleared when a byte is written to the TX data channel.

Offset:0x0014			Register Name: UART_LSR
Bit	**Read/Write**	**Default/Hex**	**Description**
5	R	0x1	THRE TX Holding Register Empty If the FIFOs are disabled, this bit is set to "1" when the TX Holding Register (UART_THR) is empty and ready to accept new data and it is cleared when the CPU writes to the TX Holding Register. If the FIFOs are enabled, this bit is set to "1" whenever the TX FIFO is empty and it is cleared when at least one byte is written to the TX FIFO.
4	R	0x0	BI Break Interrupt This is used to indicate the detection of a break sequence on the serial input data. If in UART mode (SIR_MODE == Disabled), it is set when the serial input, sir_in, is held in a logic '0' state for longer than the sum of *start time* + *data bits* + *parity* + *stop bits*. If in infrared mode (SIR_MODE == Enabled), it is set when the serial input, sir_in, is continuously pulsed to logic '0' for longer than the sum of *start time* + *data bits* + *parity* + *stop bits*. A break condition on serial input causes one and only one character, consisting of all zeros, to be received by the UART. In the FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the UART_LSR clears the BI bit. In the non-FIFO mode, the BI indication occurs immediately and persists until the UART_LSR is read.

Offset:0x0014			Register Name: UART_LSR
Bit	**Read/Write**	**Default/Hex**	**Description**
3	RC	0x0	FE Framing Error This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data. In the FIFO mode, since the framing error is associated with a character received, it is revealed when the character with the framing error is at the top of the FIFO. When a framing error occurs, the UART tries to resynchronize. It does this by assuming that the error was due to the start bit of the next character and then continues receiving the other bit i.e. data, and/or parity and stop. It should be noted that the Framing Error (FE) bit (UART_LSR[3]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (UART_LSR[4]). 0: no framing error 1: framing error Reading the UART_LSR clears the FE bit.
2	RC	0x0	PE Parity Error This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (UART_LCR[3]) is set. In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the parity error arrives at the top of the FIFO. It should be noted that the Parity Error (PE) bit (UART_LSR[2]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (UART_LSR[4]). 0: no parity error 1: parity error Reading the UART_LSR clears the PE bit.

Offset:0x0014			Register Name: UART_LSR
Bit	**Read/Write**	**Default/Hex**	**Description**
1	RC	0x0	OE Overrun Error This occurs if a new data character was received before the previous data was read. In the non-FIFO mode, the OE bit is set when a new character arrives in the receiver before the previous character was read from the UART_RBR. When this happens, the data in the UART_RBR is overwritten. In the FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the receiver. The data in the FIFO is retained and the data in the receive shift register is lost. 0: no overrun error 1: overrun error Reading the UART_LSR clears the OE bit.
0	R	0x0	DR Data Ready This is used to indicate that the receiver contains at least one character in the UART_RBR or the receiver FIFO. 0: no data ready 1: data ready This bit is cleared when the UART_RBR is read in non-FIFO mode, or when the receiver FIFO is empty, in FIFO mode. **Note: Not use when the RXDMA master is enabled (rxdma_ctrl[0] is set to 1).**

9.2.6.11 0x0018 UART Modem Status Register (Default Value: 0x0000_0000)

Offset: 0x0018			Register Name: UART_MSR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R	0x0	DCD Line State of Data Carrier Detect This is used to indicate the current state of the modem control line dcd_n. This bit is the complement of dcd_n. When the Data Carrier Detect input (dcd_n) is asserted it is an indication that the carrier has been detected by the modem or data set. 0: dcd_n input is de-asserted (logic 1) 1: dcd_n input is asserted (logic 0)

Offset: 0x0018			Register Name: UART_MSR
Bit	**Read/Write**	**Default/Hex**	**Description**
6	R	0x0	RI Line State of Ring Indicator This is used to indicate the current state of the modem control line ri_n. This bit is the complement of ri_n. When the Ring Indicator input (ri_n) is asserted it is an indication that a telephone ringing signal has been received by setting the modem or data. 0: ri_n input is de-asserted (logic 1) 1: ri_n input is asserted (logic 0)
5	R	0x0	DSR Line State of Data Set Ready This is used to indicate the current state of the modem control line dsr_n. This bit is the complement of the dsr_n. When the Data Set Ready input (dsr_n) is asserted, it is an indication that the modem or data set is ready to establish communication with UART. 0: dsr_n input is de-asserted (logic 1) 1: dsr_n input is asserted (logic 0) In Loopback Mode (UART_MCR[4] is set to 1), the DSR is the same as the DTR (UART_MCR[0]).
4	R	0x0	CTS Line State of Clear To Send This is used to indicate the current state of the modem control line cts_n. This bit is the complement of cts_n. When the Clear to Send input (cts_n) is asserted, it is an indication that the modem or data set is ready to exchange data with UART. 0: cts_n input is de-asserted (logic 1) 1: cts_n input is asserted (logic 0) In Loopback Mode (UART_MCR[4] = 1), the CTS is the same as the RTS (UART_MCR[1]).

Offset: 0x0018			Register Name: UART_MSR
Bit	**Read/Write**	**Default/Hex**	**Description**
3	RC	0x0	DDCD Delta Data Carrier Detect This is used to indicate that the modem control line dcd_n has changed since the last time the UART_MSR was read. 0: no change on dcd_n since the last read of UART_MSR 1: change on dcd_n since the last read of UART_MSR Reading the UART_MSR clears the DDCD bit. **Note: If the DDCD bit is not set and the dcd_n signal is asserted (low) and a reset occurs, then the DDCD bit is set when the reset is removed if the dcd_n signal remains asserted.**
2	RC	0x0	TERI Trailing Edge Ring Indicator This is used to indicate that a change in the input ri_n (from an active-low to an inactive-high state) has occurred since the last time the UART_MSR was read. 0: no change on ri_n since the last read of UART_MSR 1: change on ri_n since the last read of UART_MSR Reading the UART_MSR clears the TERI bit.
1	RC	0x0	DDSR Delta Data Set Ready This is used to indicate that the modem control line dsr_n has changed since the last time the UART_MSR was read. 0: no change on dsr_n since the last read of UART_MSR 1: change on dsr_n since the last read of UART_MSR Reading the UART_MSR clears the DDSR bit. In Loopback Mode (UART_MCR[4] = 1), the DDSR reflects changes on the DTR (UART_MCR[0]*)*. **Note: If the DDSR bit is not set and the dsr_n signal is asserted (low) and a reset occurs, then the DDSR bit is set when the reset is removed if the dsr_n signal remains asserted.**

Offset: 0x0018			Register Name: UART_MSR
Bit	**Read/Write**	**Default/Hex**	**Description**
0	RC	0x0	DCTS Delta Clear to Send This is used to indicate that the modem control line cts_n has changed since the last time the UART_MSR was read. 0: no change on ctsdsr_n since the last read of UART_MSR 1: change on ctsdsr_n since the last read of UART_MSR Reading the UART_MSR clears the DCTS bit. In Loopback Mode (UART_MCR[4] = 1), the DCTS reflects changes on the RTS (UART_MCR[1]). **Note: If the DCTS bit is not set and the cts_n signal is asserted (low) and a reset occurs, then the DCTS bit is set when the reset is removed if the cts_n signal remains asserted.**

9.2.6.12 0x001C UART Scratch Register (Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: UART_SCH
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0x0	SCRATCH_REG Scratch Register This register is for programmers to use as a temporary storage space. It has no defined purpose in the UART.

9.2.6.13 0x007C UART Status Register (Default Value: 0x0000_0006)

Offset: 0x007C			Register Name: UART_USR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:5	/	/	/
4	R	0x0	RFF RX FIFO Full This is used to indicate that the RX FIFO is completely full. 0: RX FIFO not full 1: RX FIFO Full This bit is cleared when the RX FIFO is no longer full.

Offset: 0x007C			Register Name: UART_USR
Bit	**Read/Write**	**Default/Hex**	**Description**
3	R	0x0	RFNE RX FIFO Not Empty This is used to indicate that the RX FIFO contains one or more entries. 0: RX FIFO is empty 1: RX FIFO is not empty This bit is cleared when the RX FIFO is empty.
2	R	0x1	TFE TX FIFO Empty This is used to indicate that the TX FIFO is completely empty. 0: TX FIFO is not empty 1: TX FIFO is empty This bit is cleared when the TX FIFO is no longer empty.
1	R	0x1	TFNF TX FIFO Not Full This is used to indicate that the TX FIFO is not full. 0: TX FIFO is full 1: TX FIFO is not full This bit is cleared when the TX FIFO is full.
0	R	0x0	BUSY UART Busy Bit 0: Idle or inactive 1: Busy

9.2.6.14 0x0080 UART Transmit FIFO Level Register (Default Value: 0x0000_0000)

Offset: 0x0080			Register Name: UART_TFL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8:0	R	0x0	TFL TX FIFO Level The bit indicates the number of data entries in the TX FIFO.

9.2.6.15 0x0084 UART Receive FIFO Level Register (Default Value: 0x0000_0000)

Offset: 0x0084			Register Name: UART_RFL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:9	/	/	/
8:0	R	0x0	RFL RX FIFO Level The bit indicates the number of data entries in the RX FIFO. **Note: Not use when the RXDMA master is enabled (**UART_RXDMA_CTRL**[0] is set to 1).**

9.2.6.16 0x0088 UART DMA Handshake Configuration Register (Default Value: 0x0000_00A5)

Offset: 0x0088			Register Name: UART_HSK
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0xA5	Handshake configuration 0xA5: DMA wait cycle mode 0xE5: DMA handshake mode

9.2.6.17 0x008C UART DMA Request Enable Register(Default Value: 0x0000_0003)

Offset: 0x008C			Register Name: UART_DMA_REQ_EN
Bit	**Read/Write**	**Default/Hex**	**Description**
31:3	/	/	/
2	R/W	0x0	DMA Timeout Enable 0: Disable 1: Enable
1	R/W	0x1	DMA TX REQ Enable 0: Disable 1: Enable
0	R/W	0x1	DMA RX REQ Enable 0: Disable 1: Enable

9.2.6.18 0x00A4 UART Halt TX Register (Default Value: 0x0000_0000)

Offset: 0x00A4			Register Name: UART_HALT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7	R/W	0x0	PTE The sending of TX_REQ In DMA1 mode (FIFO on), if the PTE is set to 1 when the TFL in UART_TFL is less than or equal to the trigger value, the controller sends the DMA request. If the PTE is set to 0, when FIFO is empty, the controller sends the DMA request. The DMA request will stop when FIFO is full. In DMA0 mode, if the PTE is set to 1 and FIFO is on, when the TFL in UART_TFL is less than or equal to the trigger value, the controller sends DMA request. If the PTE is set to 1 and FIFO off, when the THR in UART_THR is empty, the controller sends DMA request. If the PTE is set to 0, when FIFO(FIFO Enable) or THR(FIFO Enable) is empty, the controller sends DMA request. Otherwise, the DMA request is cleared.
6	R/W	0x0	DMA_PTE_RX The Transmission of RX_DRQ In DMA1 mode, when RFL is more than or equal to the trigger value, or a receive timeout has occurred, the controller sends DRQ. In DMA0 mode, when DMA_PTE_RX = 1 and FIFO is on, if RFL is more than or equal to trig, the controller sends DRQ, else DRQ is cleared. In other cases, once the received data is valid, the controller sends DRQ.
5	R/W	0x0	SIR_RX_INVERT SIR RX Pulse Polarity Invert 0: Not invert receiver signal 1: Invert receiver signal
4	R/W	0x0	SIR_TX_INVERT SIR TX Pulse Polarity Invert 0: Not invert transmit pulse 1: Invert transmit pulse
3	/	/	/

Offset: 0x00A4			Register Name: UART_HALT
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R/WAC	0x0	CHANGE_UPDATE After the user uses UART_HALT[1] to change the baud rate or LCR configuration, write 1 to update the configuration and wait this bit to self-clear to 0 to finish update process. Writing 0 to this bit has no effect. 1: Update trigger, self-clear to 0 when finish update.
1	R/W	0x0	CHCFG_AT_BUSY This is an enable bit for the user to change LCR register configuration and baud rate register (UART_DLH and UART_DLL) when the UART is busy. 1: Enable change when busy
0	R/W	0x0	HALT_TX Halt TX This register is used to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. 0 : Halt TX disabled 1 : Halt TX enabled **Note: If FIFOs are not enabled, the setting has no effect on operation.**

9.2.6.19 0x00B0 UART DBG DLL Register (Default Value: 0x0000_0000)

Offset: 0x00B0			Register Name: UART_DBG_DLL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R	0x0	DEBUG DLL

9.2.6.20 0x00B4 UART DBG DLH Register (Default Value: 0x0000_0000)

Offset: 0x00B4			Register Name: UART_DBG_DLH
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R	0x0	DEBUG DLH

9.2.6.21 0x00F0 UART FIFO Clock Control Register (Default Value: 0x0000_0003)

Offset: 0x00F0			Register Name: UART_FCC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	R	0x0	FIFO Depth Indicates the depth of TX/RX FIFO
7:3	/	/	/
2	R/W	0x0	RXFIFO Clock Mode 0: Sync mode, writing/reading clocks use apb clock 1: Sync mode, writing clock uses apb clock, reading clock uses ahb clock
1	R/W	0x1	TX FIFO Clock Enable 0: Clock disable 1: Clock enable
0	R/W	0x1	RX FIFO Clock Enable 0: Clock disable 1: Clock enable

9.2.6.22 0x0100 UART RXDMA Control Register (Default Value: 0x0000_0000)

Offset: 0x0100			Register Name: UART_RXDMA_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:8	R/W	0x0	RXDMA Timeout Threshold Unit is 1 UART bit time Note that this field is only configurable when RXDMA Busy is 0.
7	/	/	/
6	R/W	0x0	RXDMA Timeout Enable Once enable, the DMA starts a transfer even the data entries in RX FIFO do not reach BLK_SIZE. Note that this field is only configurable when RXDMA Busy is 0.

Offset: 0x0100			Register Name: UART_RXDMA_CTRL
Bit	**Read/Write**	**Default/Hex**	**Description**
5:4	R/W	0x0	RXDMA AHB Burst Mode Set for AHB port burst supported INCR8 is recommended, while INCR16 may be unsupported due to the system bus. 00: SIGNLE 01: INCR4 10: INCR8 11: INCR16 Note that this field is only configurable while RXDMA Busy is 0.
3:2	R/W	0x0	RXDMA BLK SIZE Every time when data entries in RX FIFO reach BLK_SIZE, start a DMA block transfer. It is recommended that the block size no more than RX FIFO Depth. 00: 8 bytes 01: 16 bytes 10: 32 bytes 11: 64 bytes Note that this field is only configurable while RXDMA Busy is 0.
1	R/W	0x0	RXDMA Mode 0: Continous 1: Limited When data transferred reaches the limited count set in RXDMA LIMIT, the DMA stops and the RXDMA Start bit is cleared automatically. Note that this field is only configurable while RXDMA Busy is 0.
0	R/W	0x0	RXDMA Enable 0: RXDMA Disable 1: RXDMA Enable Note that if the software turns off this bit, the RXDMA will stop after the current block transfer completes, then the software should do a reset to the RX FIFO before re-enable.

9.2.6.23 0x0104 UART RXDMA Start Register (Default Value: 0x0000_0000)

Offset: 0x0104			Register Name: UART_RXDMA_STR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:2	/	/	/
0	R/WAC	0x0	RXDMA Start Only valid when RXDMA mode is set to 1, it is auto cleared when data transferred reaches the RXDMA Limit Size.

9.2.6.24 0x0108 UART RXDMA Status Register (Default Value: 0x0000_0000)

Offset: 0x0108			Register Name: UART_RXDMA_STA
Bit	**Read/Write**	**Default/Hex**	**Description**
31:1	/	/	/
1	R	0x0	Buffer Read Address Updating 0: Buffer Read Address Register is ready for updating 1: Buffer Read Address Register is busy for updating The software should not update Buffer Read Address Register until this bit is 0.
0	R	0x0	RXDMA BUSY 0: RXDMA is idle 1: RXDMA is busy

9.2.6.25 0x010C UART RXDMA Limit Register (Default Value: 0x0000_0000)

Offset: 0x010C			Register Name: UART_RXDMA_LMT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R/W	0x0	RXDMA Limit Size Only valid when RXDMA Mode is set to 1, and the unit is byte. Note that this field is only configurable while RXDMA Busy is 0.

9.2.6.26 0x0110 UART RXDMA Buffer Start Address Low Register (Default Value: 0x0000_0000)

Offset: 0x0110			Register Name: UART_RXDMA_SADDRL
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	RXDMA Buffer Start Address [31:0] Byte address Note that this field is only configurable while RXDMA Busy is 0.

9.2.6.27 0x0114 UART RXDMA Buffer Start Address High Register (Default Value: 0x0000_0000)

Offset: 0x0114			Register Name: UART_RXDMA_SADDRH
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1:0	R/W	0x0	RXDMA Buffer Start Address [33:32] Byte address Note that this field is only configurable while RXDMA Busy is 0.

9.2.6.28 0x0118 UART RXDMA Buffer Length Register (Default Value: 0x0000_0000)

Offset: 0x0118			Register Name: UART_RXDMA_BL
Bit	Read/Write	Default/Hex	Description
31:16	/	/	/
15:0	R/W	0x0	RXDMA Buffer Length Unit is byte Note that this field is only configurable while RXDMA Busy is 0.

9.2.6.29 0x0120 UART RXDMA Interrupt Enable Register (Default Value: 0x0000_0000)

Offset: 0x0120			Register Name: UART_RXDMA_IE
Bit	Read/Write	Default/Hex	Description
31:4	/	/	/
3	R/W	0x0	RXDMA Buffer Overrun RXDMA Buffer Overrun Interrupt Enable

Offset: 0x0120			Register Name: UART_RXDMA_IE
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R/W	0x0	RXDMA Timeout Done RXDMA Timeout Done Interrupt Enable
1	R/W	0x0	RXDMA BLK Done RXDMA BLK Done Interrupt Enable
0	R/W	0x0	RXDMA Limit Done RXDMA Limit Done Interrupt Enable

9.2.6.30 0x0124 UART RXDMA Interrupt Status Register (Default Value: 0x0000_0000)

Offset: 0x0124			Register Name: UART_RXDMA_IS
Bit	**Read/Write**	**Default/Hex**	**Description**
31:4	/	/	/
3	R/W1C	0x0	RXDMA Buffer Overrun Asserted when the RXDMA buffer is overflow.
2	R/W1C	0x0	RXDMA Timeout Done Asserted when a DMA transfer caused by timeout is done.
1	R/W1C	0x0	RXDMA BLK Done Asserted when a DMA block transfer is done.
0	R/W1C	0x0	RXDMA Limit Done Asserted when data transferred reaches limit size in RXDMA Limit Mode.

9.2.6.31 0x0128 UART RXDMA Write Address Low Register (Default Value: 0x0000_0000)

Offset: 0x0128			Register Name: UART_RXDMA_WADDRL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	RXDMA Current Write Address[31:0] Updated when every DMA transfer is done It is byte address.

9.2.6.32 0x012C UART RXDMA Write Address High Register (Default Value: 0x0000_0000)

Offset: 0x012C			Register Name: UART_RXDMA_WADDRH
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1:0	R	0x0	RXDMA Current Write Address[33:32] Updated when every DMA transfer is done It is byte address.

9.2.6.33 0x0130 UART RXDMA Read Address Low Register (Default Value: 0x0000_0000)

Offset: 0x0130			Register Name: UART_RXDMA_RADDRL
Bit	Read/Write	Default/Hex	Description
31:0	R/W	0x0	RXDMA Current Read Address[31:0] Software should update this register after reading data in RXDMA Buffer in time It is byte address. The software should not update Buffer Read Address Register until UART_RXDMA_STA[1] is 0. The software should update Read Address High Register first, and then Read Address Low Register, even there is no change on Read Address High Register.

9.2.6.34 0x0134 UART RXDMA Read Address High Register (Default Value: 0x0000_0000)

Offset: 0x0134			Register Name: UART_RXDMA_RADDRH
Bit	Read/Write	Default/Hex	Description
31:2	/	/	/
1:0	R/W	0x0	RXDMA Current Read Address[33:32] Software should update this register after reading data in RXDMA Buffer in time. It is byte address. The software should not update Buffer Read Address Register until until UART_RXDMA_STA[1] is 0. The software should update Read Address High Register first and then Read Address Low Register , even there is no change on Read Address High Register.

9.2.6.35 0x0138 UART RXDMA Data Count Register (Default Value: 0x0000_0000)

Offset: 0x0138			Register Name: UART_RXDMA_DCNT
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15:0	R	0x0	RXDMA Data Count Only valid while RXDMA Mode is set to 1, it is used for counting the data transferred by RXDMA, and is cleared when reaches RXDMA Limit Size. Its unit is byte.

9.3 SPI

9.3.1 Overview

The Serial Peripheral Interface (SPI) is a full-duplex, synchronous, four-wire serial communication interface between a CPU and SPI-compliant external devices. The SPI controller contains a 64 x 8 bits receiver buffer (RXFIFO) and a 64 x 8 bits transmit buffer (TXFIFO). It can work in master mode and slave mode.

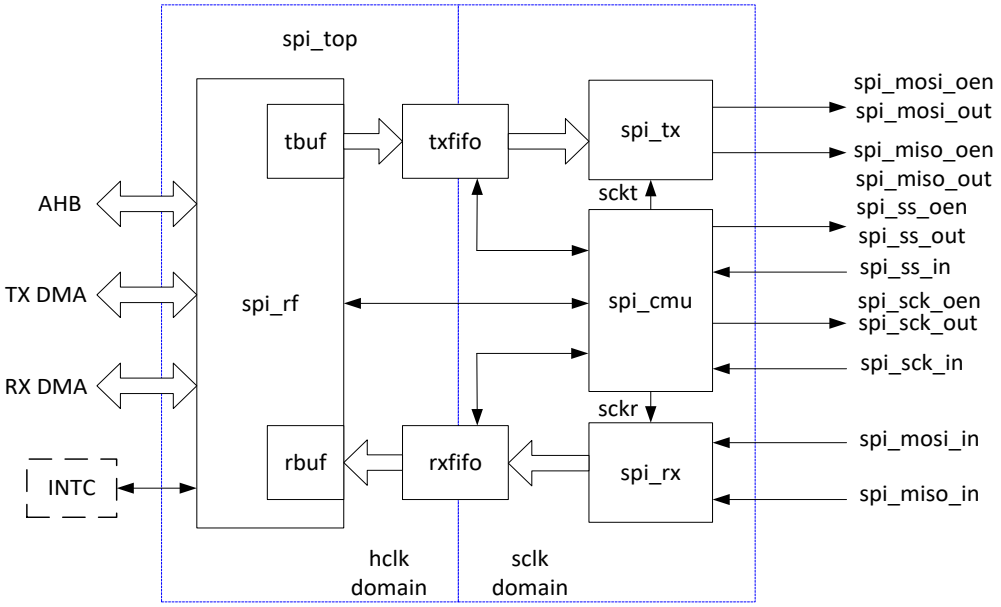
The SPI has the following features:

- Full-duplex synchronous serial interface
- Master/slave configurable
- 8-bit wide by 64-entry FIFO for both transmit and receive data
- Polarity and phase of the Chip Select (SPI_SS) and SPI Clock (SPI_SCLK) are configurable
- Supports interrupts and DMA
- Supports mode0, mode1, mode2, and mode3
- Supports 3-wire/4-wire SPI
- Supports programmable serial data frame length: 1 bit to 32 bits
- Supports standard SPI, dual-output/dual-input SPI, dual I/O SPI, quad-output/quad-input SPI
- Supports maximum IO rate of the mass production: 100 MHz

9.3.2 Block Diagram

Figure 9-20 shows a block diagram of the SPI.

Figure 9-20 SPI Block Diagram



SPI contains the following sub-blocks:

Table 9-6 SPI Sub-blocks

Sub-block	Description
spi_rf	Responsible for implementing the internal register, interrupt, and DMA Request.
spi_tbuf	The data length transmitted from AHB to TXFIFO is converted into 8 bits, then the data is written into the RXFIFO.
spi_rbuf	The block is used to convert the RXFIFO data into the reading data length of AHB.
txfifo, rxfifo	The data transmitted from the SPI to the external serial device is written into the TXFIFO; the data received from the external serial device into SPI is pushed into the RXFIFO.
spi_cmu	Responsible for implementing SPI bus clock, chip select, internal sample, and the generation of transfer clock.
spi_tx	Responsible for implementing SPI data transfer, the interface of the internal TXFIFO, and status register.
spi_rx	Responsible for implementing SPI data receive, the interface of the internal RXFIFO, and status register.

9.3.3 Functional Description

9.3.3.1 External Signals

The following table describes the external signals of SPI. The MOSI and MISO are bidirectional I/O, when SPI is as a master device, the CLK and CS are the output pin; when SPI is as a slave device, the CLK and CS are the input pin. When using SPI, the corresponding PADs are selected as SPI function via section 9.7 "GPIO".

Table 9-7 SPI External Signals

Signal	Description	Type
SPI0-CS	SPI0 chip select signal, low active When the device is not selected, data will not be accepted via the SI pin, and the SO pin will stop transmission.	I/O
SPI0-CLK	SPI0 clock signal This pin is used to provide a clock to the device and is used to control the flow of data to and from the device.	I/O
SPI0-MOSI	SPI0 master data out, slave data in	I/O
SPI0-MISO	SPI0 master data in, slave data out	I/O
SPI0-WP	Write protection and low active It also can be used for serial data input and output for SPI Quad Input or Quad Output mode.	I/O
SPI0-HOLD	When the device is selected and a serial sequence is underway, the HOLD pin can be used to temporarily pause the serial communication with the master device without deselecting or resetting the serial sequence. While the HOLD pin is asserted, the SO pin is at high impedance, and all transitions on the SCK pin and data on the SI pin are ignored. It also can be used for serial data input and output for SPI Quad Input or Quad Output mode.	I/O

9.3.3.2 Clock Sources

The SPI controller gets 5 different clock sources, users can select one of them to make SPI clock source. The following table describes the clock sources for SPI. For more details on the clock setting, configuration, and gating information, see section 3.3 "CCU".

Table 9-8 SPI Clock Sources

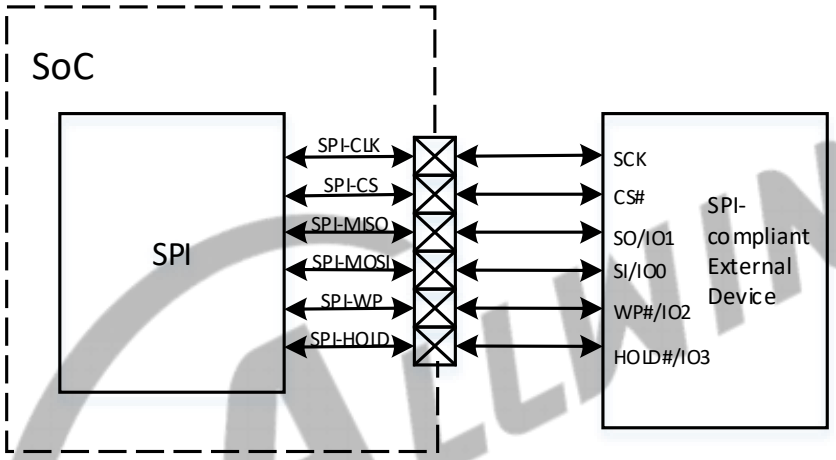
Clock Sources	Description
HOSC	24 MHz Crystal

Clock Sources	Description
PLL_PERI(1X)	Peripheral Clock, default value is 600 MHz
PLL_PERI(2X)	Peripheral Clock, default value is 1200 MHz
PLL_AUDIO0(DIV2)	Audio Clock, the default value is 1536 MHz
PLL_AUDIO0(DIV5)	Audio Clock, the default value is 614.4 MHz

9.3.3.3 Typical Application

Figure 9-21 shows the application block diagram when the SPI master device is connected to a slave device.

Figure 9-21 SPI Application Block Diagram



9.3.3.4 SPI Transmit Format

The SPI supports 4 different formats for data transfer. The software can select one of the four modes in which the SPI works by setting the bit1 (Polarity) and bit0 (Phase) of SPI_TCR. The SPI controller master uses the SPI_SCLK signal to transfer data in and out of the shift register. Data is clocked using any one of four programmable clock phase and polarity combinations.

The CPOL (SPI_TCR[1]) defines the polarity of the clock signal (SPI_SCLK). The SPI_SCLK is a high level when CPOL is '1' and it is a low level when CPOL is '0'. The CPHA (SPI_TCR[0]) decides whether the leading edge of SPI_SCLK is used to setup or sample data. The leading edge is used to setup data when CPHA is '1', and sample data when CPHA is '0'. The following table lists the four modes.

Table 9-9 SPI Transmit Format

Mode	Polarity (CPOL)	Phase (CPHA)	Leading Edge	Trailing Edge
Mode0	0	0	Sample on the rising edge	Setup on the falling edge

Mode	Polarity (CPOL)	Phase (CPHA)	Leading Edge	Trailing Edge
Mode1	0	1	Setup on the rising edge	Sample on the falling edge
Mode2	1	0	Sample on the falling edge	Setup on the rising edge
Mode3	1	1	Setup on the falling edge	Sample on the rising edge

Figure 9-22 and Figure 9-23 describe four waveforms for SPI_SCLK.

Figure 9-22 SPI Phase 0 Timing Diagram

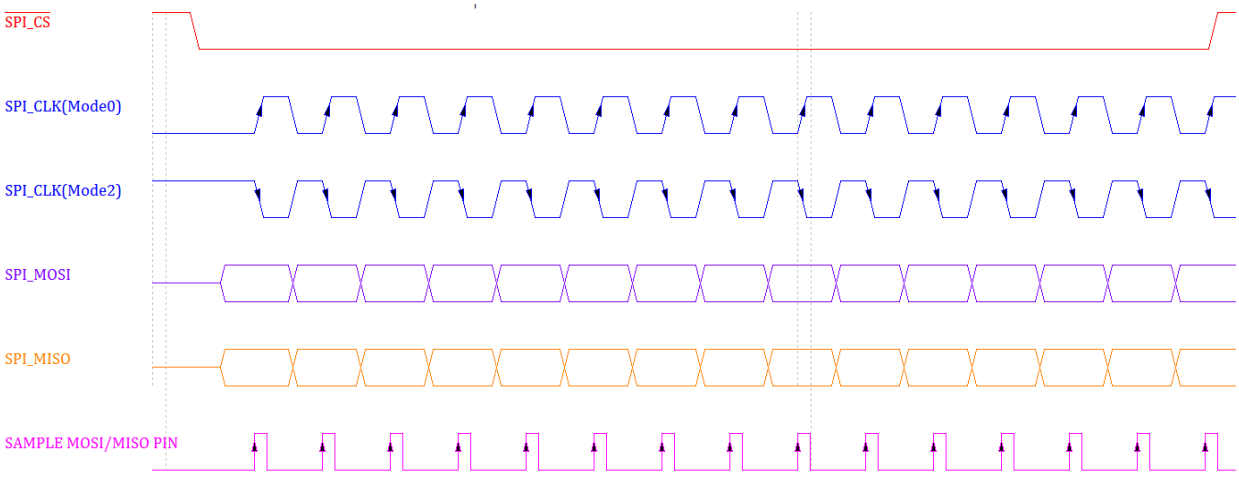
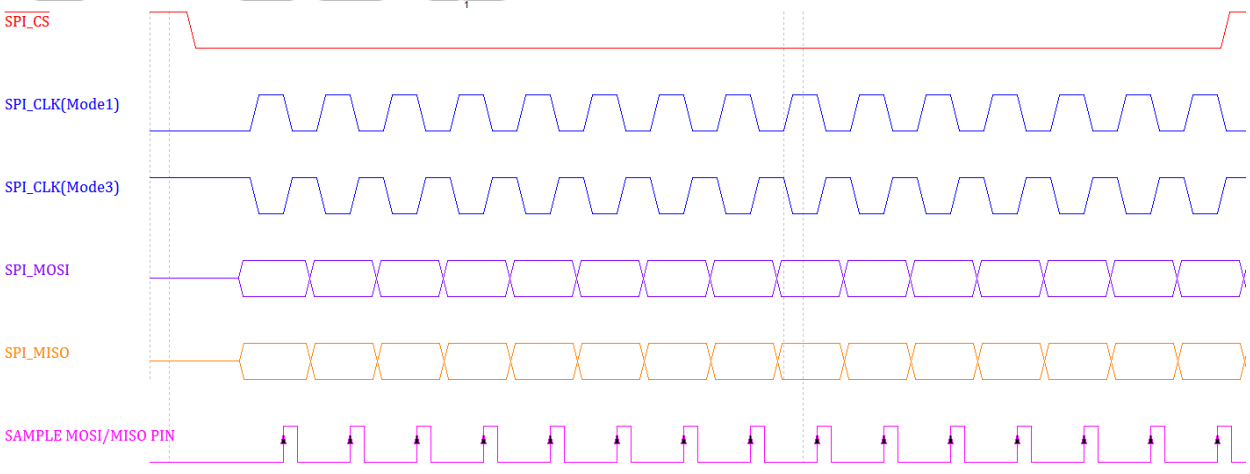


Figure 9-23 SPI Phase 1 Timing Diagram



9.3.3.5 SPI Master and Slave Mode

The SPI controller can be configured to a master or slave device. The master mode is selected by setting the MODE bit (**SPI_GCR**[1]); the slave mode is selected by clearing the MODE bit.

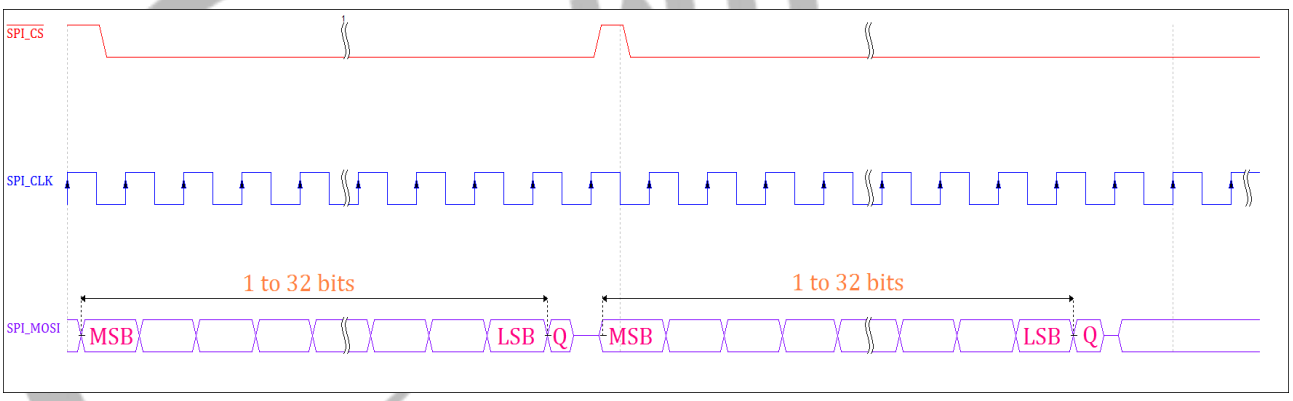
In master mode, the SPI_CLK is generated and transmitted to the external device, and the data from the TX FIFO is transmitted on the MOSI pin, the data from the slave device is received on the MISO pin and sent to RX FIFO. The Chip Select (SPI_SS) is an active low signal, and it must be set low before the data are transmitted or received. The SPI_SS can be selected the auto control mode or software manual control mode. When using the auto control, the SS_OWNER (SPI_TCR[6]) must be cleared (default value is 0); when using the manual control, the SS_OWNER must be set. And the level of SPI_SS is controlled by SS_LEVEL (SPI_TCR[7]).

In slave mode, after the software selects the MODE bit (SPI_GCR[1]) to '0', it waits for master initiate a transaction. When the master asserts SPI_SS, then SPI_CLK is transmitted to the slave device, the slave data is transmitted from TX FIFO on the MISO pin and the data from the MOSI pin is received in RX FIFO.

9.3.3.6 SPI 3-Wire Mode

The SPI 3-wire mode is only valid when the SPI controller work in master mode, and is selected when the Work Mode Select bit (SPI_BATC[1:0]) is equal to 0x2. And in the 3-wire mode, the input data and the output data use the same single data line. The following figure describes the 3-wire mode.

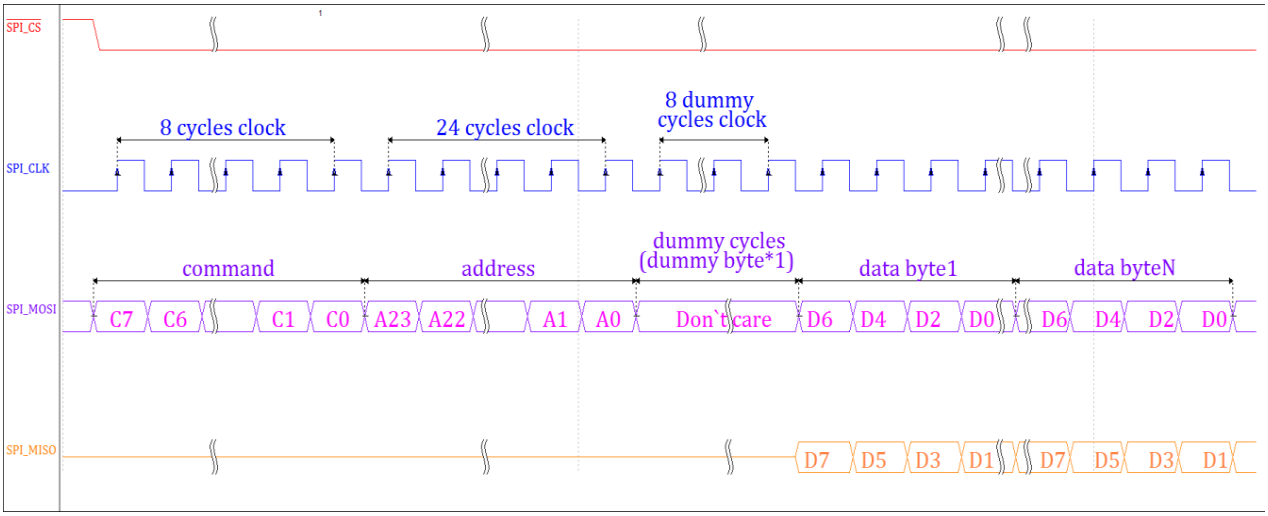
Figure 9-24 SPI 3-Wire Mode



9.3.3.7 SPI Dual-Input/Dual-Output and Dual I/O Mode

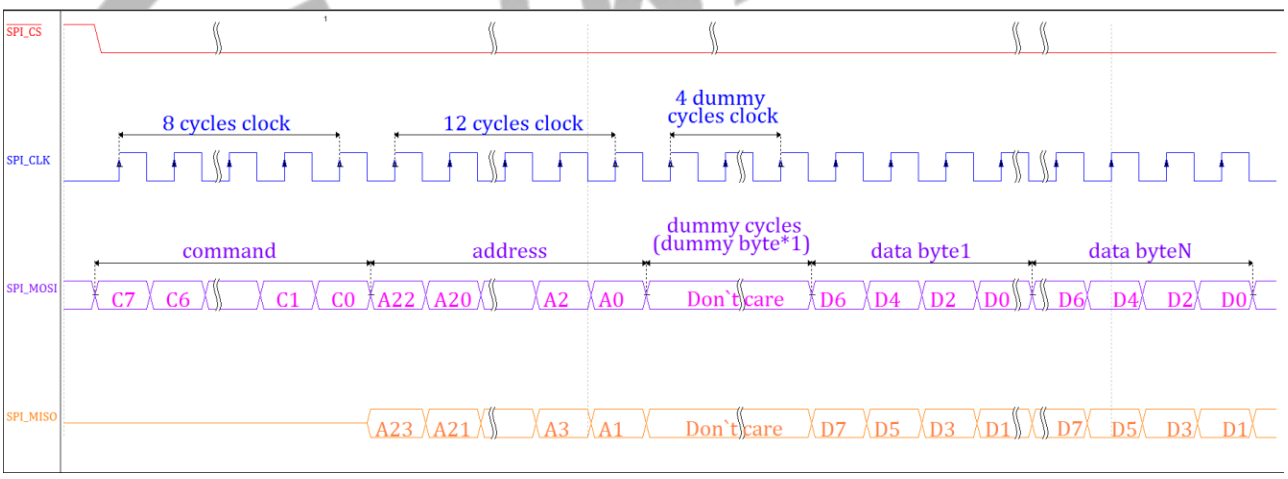
The dual read mode (SPI x2) is selected when the DRM is set in SPI_BCC[28]. Using the dual mode allows data to be transferred to or from the device at double the rate of standard single mode, the data can be read at fast speed using two data bits (MOSI and MISO) at a time. The following figure describes the dual-input/dual-output SPI (Figure 9-25) and the dual I/O SPI (Figure 9-26).

Figure 9-25 SPI Dual-Input/Dual-Output Mode



In the dual-input/dual-output SPI mode, the command, address, and the dummy bytes output in a unit of a single bit in serial mode through the SPI_MOSI line, only the data bytes are output (write) and input (read) in a unit of dual bits through the SPI_MOSI and SPI_MISO.

Figure 9-26 SPI Dual I/O Mode

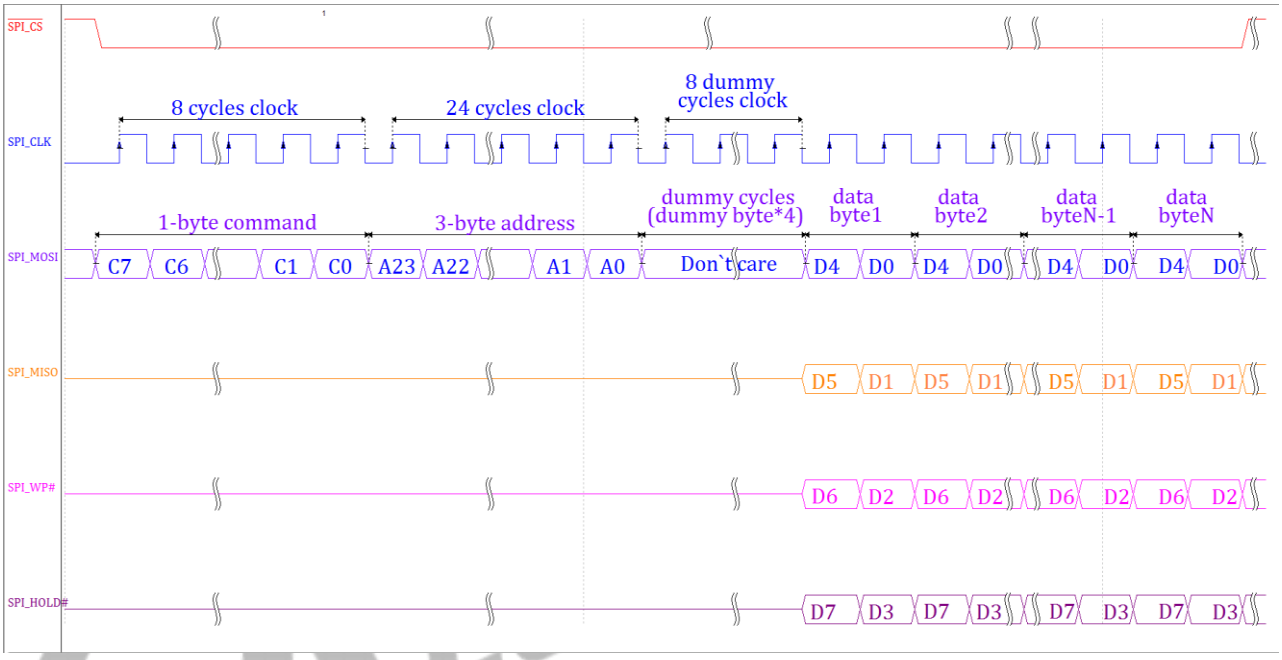


In the dual I/O SPI mode, only the command bytes output in a unit of a single bit in serial mode through the SPI_MOSI line. The address bytes and the dummy bytes output in a unit of dual bits through the SPI_MOSI and SPI_MISO. And the data bytes output (write) and input (read) in a unit of dual bits through the SPI_MOSI and SPI_MISO.

9.3.3.8 SPI Quad-Input/Quad-Output Mode

The quad read mode (SPI x4) is selected when the Quad_EN is set in **SPI_BCC**[29]. Using the quad mode allows data to be transferred to or from the device at 4 times the rate of standard single mode, the data can be read at fast speed using four data bits (MOSI, MISO, IO2 (WP#) and IO3 (HOLD#)) at the same time. The following figure describes the quad-input/quad-output SPI.

Figure 9-27 SPI Quad-Input/Quad-Output Mode



In the quad-input/quad-output SPI mode, the command, address, and the dummy bytes output in a unit of a single bit in serial mode through SPI_MOSI line. Only the data bytes output (write) and input (read) in a unit of quad bits through the SPI_MOSI, SPI_MISO, SPI_WP#, and SPI_HOLD#.

9.3.3.9 Transmission/Reception Bursts in Master Mode

In SPI master mode, the transmission and reception bursts (byte in unit) are configured before the SPI transfers the serial data between the processor and external device. The transmission bursts are written in MWTC (bit[23:0]) of the SPI Master Transmit Counter Register. The transmission bursts in single mode before automatically sending dummy bursts are written in STC (bit[23:0]) of the SPI Master Transmit Counter Register**.** For dummy data, the SPI controller can automatically send before receiving by writing DBC (bit[27:24]) in the SPI Master Transmit Counter Register. If users do not use the SPI controller to send dummy data automatically, then the dummy bursts are used as the transmission counters to write together in MWTC (bit[23:0]) of the SPI Master Transmit Counter Register. In master mode, the total burst numbers are written in MBC (bit[23:0]) of the SPI Master Transmit Counter Register. When all transmission and reception bursts are transferred, the SPI controller will send a completed interrupt, at the same time, the SPI controller will clear DBC, MWTC, and MBC.

9.3.3.10 SPI Sample Mode and Run Clock Configuration

The SPI controller runs at 3 kHz–100 MHz at its interface to external SPI devices. The internal SPI clock should run at the same frequency as the outgoing clock in the master mode. The SPI clock is selected from different clock sources, the SPI must configure different work mode. There are three work modes: normal sample mode, delay half-cycle sample mode, delay one-cycle sample mode. The Delay half-cycle sample mode is the default mode of the SPI controller. When the SPI runs at 40 MHz or below 40 MHz, the SPI can work at normal sample mode or delay half-cycle sample mode. When the SPI runs over 80 MHz, setting the SDC bit in the SPI Transfer Control Register to '1' makes the internal read sample point with a half-cycle delay of SPI_CLK, which is used in high speed read operation to reduce the error caused by the time delay of SPI_CLK between master and slave. Table 9-10 and Table 9-11 show the different configurations of the SPI sample mode.

Table 9-10 SPI Old Sample Mode and Run Clock

SPI Sample Mode	SDM(bit13)	SDC(bit11)	Run Clock
normal sample	1	0	<=24 MHz
delay half cycle sample	0	0	<=40 MHz
delay one cycle sample	0	1	>=80 MHz

CAUTION

The remaining spectrum is not recommended. Because when the output delay of SPI flash (refer to the datasheet of the manufactures for the specific delay time) is the same with the half-cycle time of SPI working clock, the variable edge of the output data for the device bumps into the clock sampling edge of the controller, so setting 1 cycle of sampling delay would cause stability problem.

Table 9-11 SPI New Sample Mode

SPI Sample Mode	SDM (bit13)	SDC (bit11)	SDC1 (bit15)
normal sample	1	0	0
delay half cycle sample	0	0	0
delay one cycle sample	0	1	0
delay 1.5 cycle sample	1	1	0
delay 2 cycle sample	1	0	1
delay 2.5 cycle sample	0	0	1
delay 3 cycle sample	0	1	1

9.3.3.11 SPI Error Conditions

If any error conditions occur, the hardware will set the corresponding status bits in the SPI Interrupt Status Register and stop the transfer. For the SPI controller, the following error scenarios can happen.

1. **TX_FIFO Underrun**

   The TX_FIFO underrun happens when the CPU/DMA reads data from TX FIFO when it is empty. In the case, the SPI controller will end the transaction and flag the error bit along with the TF_UDF bit in the SPI Interrupt Status Register. The SPI controller will generate an interrupt if interrupts are enabled. The software has to clear the error bit and the TF_UDF bit. To start a new transaction, the software has to reset the FIFO by writing to the SRST (soft reset) bit in the SPI Global Control Register.

2. **TX_FIFO Overflow**

   The TX_FIFO overflow happens when the CPU/DMA writes data into the TX FIFO when it is full. In the case, the SPI controller will end the transaction and flag the error bit along with the TF_OVF bit in the SPI Interrupt Status Register. The SPI controller will generate an interrupt if interrupts are enabled. The software has to clear the error bit and the TF_OVF bit. To start a new transaction, the software has to reset the FIFO by writing to the SRST (soft reset) bit in the SPI Global Control Register.

3. **RX_FIFO Underrun**

   The RX_FIFO underrun happens when the CPU/DMA reads data from RX FIFO when it is empty. In the case, the SPI controller will end the transaction and flag the error bit along with the RF_UDF bit in the SPI Interrupt Status Register. The SPI controller will generate an interrupt if interrupts are enabled. The software has to clear the error bit and the RF_UDF bit. To start a new transaction, the software has to reset the fifo by writing to the SRST (soft reset) bit in the SPI Global Control Register.

4. **RX_FIFO Overflow**

   The RX_FIFO overflow happens when the CPU/DMA writes data into the RX FIFO when it is full. In the case, the SPI controller will end the transaction and flag the error bit along with the RF_OVF bit in the SPI Interrupt Status Register. The SPI controller will generate an interrupt if interrupts are enabled. The software has to clear the error bit and the RF_OVF bit. To start a new transaction, the software has to reset the FIFO by writing to the SRST (soft reset) bit in the SPI Global Control Register.

9.3.4 Programming Guidelines

9.3.4.1 Writing/Reading Data Process

The SPI transfers serial data between the processor and the external device. The CPU mode and DMA mode are the two main operational modes for SPI. For each SPI, the data is simultaneously transmitted (shifted out serially) and received (shifted in serially). The SPI has 2 channels, including the TX channel and RX channel. The TX channel has the path from TX FIFO to the external device. The RX channel has the path from the external device to RX FIFO.

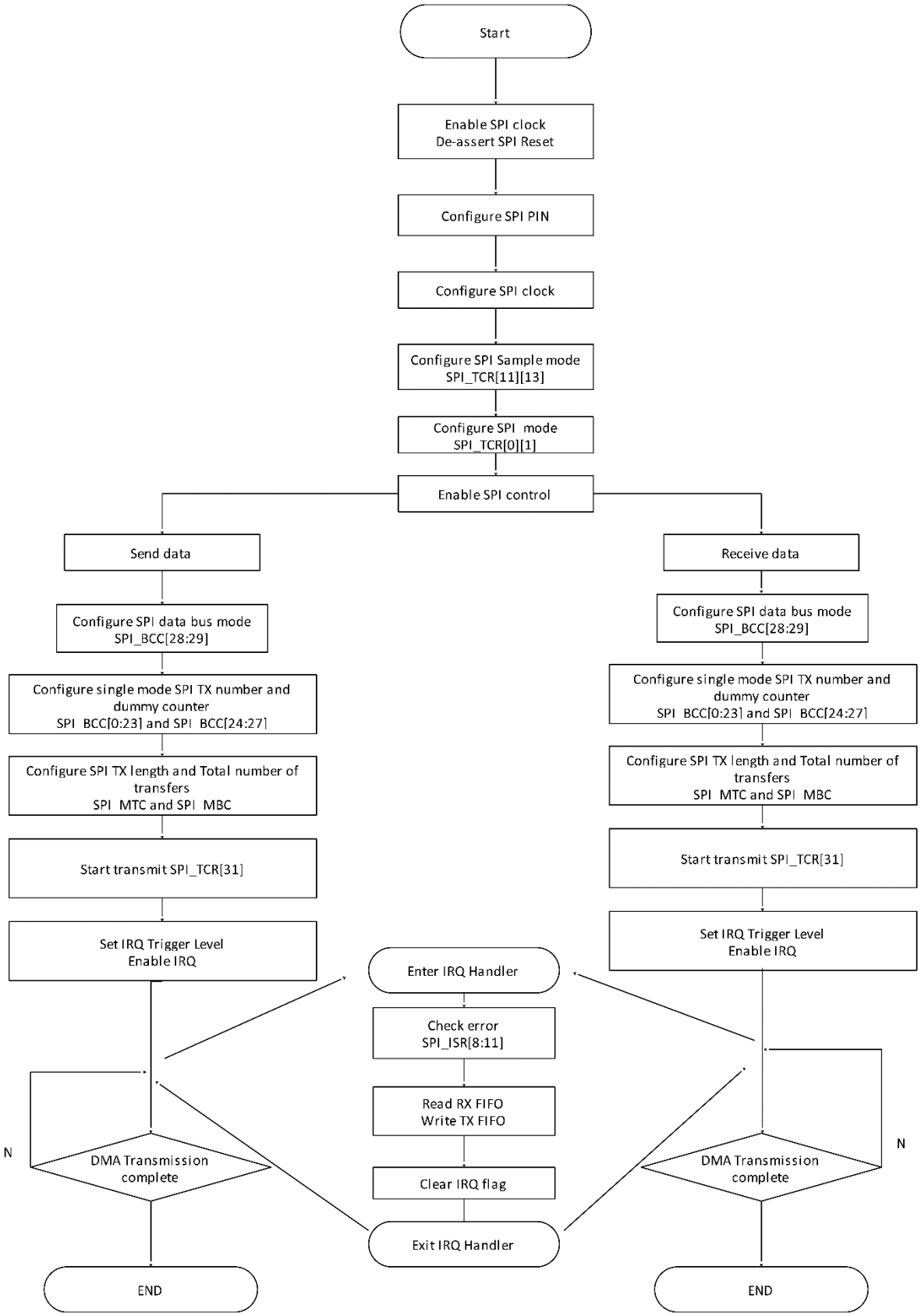
Write Data: The CPU or DMA must write data on the **SPI_TXD** register, the data on the register are automatically moved to TX FIFO.

Read Data: To read data from RX FIFO, the CPU or DMA must access the **SPI_RXD** register and the data are automatically sent to the **SPI_RXD** register.

In CPU or DMA mode, the SPI sends a completed interrupt (**SPI_ISR**[TC]) to the processor after each transmission is complete.

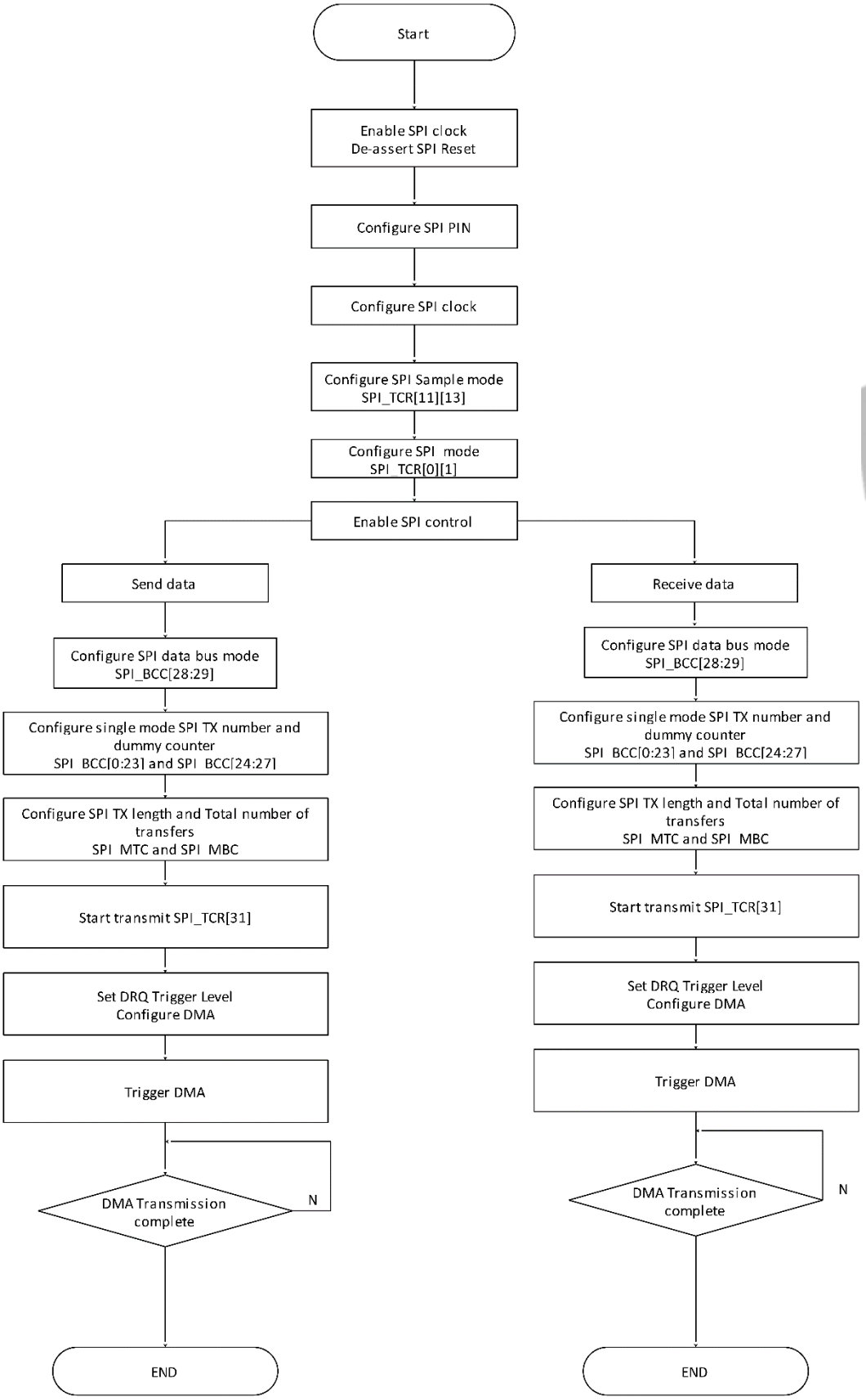
CPU Mode

Figure 9-28 SPI Write/Read Data in CPU Mode



DMA Mode

Figure 9-29 SPI Write/Read Data in DMA Mode



9.3.4.2 Calibrate Delay Chain

The SPI has one delay chain which is used to generate delay to make proper timing between the internal SPI clock signal and data signals. Delay chain is made up of 64 delay cells. The delay time of one delay cell can be estimated through delay chain calibration.

The steps to calibrate delay chain are as follows:

Step 1 Enable SPI. To calibrate the delay chain by the operation registers in SPI, the SPI must be enabled through AHB reset and AHB clock gating control registers.

Step 2 Configure a proper clock for SPI. The calibration delay chain is based on the clock for SPI from CCU.

Step 3 Set proper initial delay value. Write 0xA0 to the SPI Sample Delay Control Register to set initial delay value 0x20 to delay chain. Then write 0x0 to the SPI Sample Delay Control Register to clear this value.

Step 4 Write 0x8000 to the SPI Sample Delay Control Register to start to calibrate the delay chain.

Step 5 Wait until the flag (bit14 in the SPI Sample Delay Control Register) of calibration done is set. The number of delay cells is shown at the bit[13:8] of the SPI Sample Delay Control Register. The delay time generated by these delay cells is equal to the cycle of the SPI clock nearly. This value is the result of calibration.

Step 6 Calculate the delay time of one delay cell according to the cycle of the SPI clock and the result of calibration.

9.3.5 Register List

Module Name	Base Address
SPI0	0x04025000

Register Name	Offset	Description
SPI_GCR	0x0004	SPI Global Control Register
SPI_TCR	0x0008	SPI Transfer Control Register
SPI_IER	0x0010	SPI Interrupt Control Register
SPI_ISR	0x0014	SPI Interrupt Status Register
SPI_FCR	0x0018	SPI FIFO Control Register
SPI_FSR	0x001C	SPI FIFO Status Register
SPI_WCR	0x0020	SPI Wait Clock Register
SPI_SAMP_DL	0x0028	SPI Sample Delay Control Register
SPI_MBC	0x0030	SPI Master Burst Counter Register

Register Name	Offset	Description
SPI_MTC	0x0034	SPI Master Transmit Counter Register
SPI_BCC	0x0038	SPI Master Burst Control Register
SPI_BATCR	0x0040	SPI Bit-Aligned Transfer Configure Register
SPI_BA_CCR	0x0044	SPI Bit-Aligned Clock Configuration Register
SPI_TBR	0x0048	SPI TX Bit Register
SPI_RBR	0x004C	SPI RX Bit Register
SPI_NDMA_MODE_CTL	0x0088	SPI Normal DMA Mode Control Register
SPI_TXD	0x0200	SPI TX Data Register
SPI_RXD	0x0300	SPI RX Data Register

9.3.6 Register Description

9.3.6.1 0x0004 SPI Global Control Register (Default Value: 0x0000_0080)

Offset:0x0004			Register Name: SPI_GCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/WAC	0x0	SRST Soft reset Writing '1' to this bit will clear the SPI controller, and auto clear to '0' when reset operation completes. Writing '0' to this bit has no effect.
30:8	/	/	/
7	R/W	0x1	TP_EN Transmit Pause Enable In master mode, it is used to control the transmit state machine to stop smart burst sending when RX FIFO is full. 0: Normal operation, ignore RXFIFO status 1: Stop transmit data when RXFIFO full Cannot be written when XCH=1.
6:3	/	/	/
2	R/W	0x0	MODE_SELEC Sample Timing Mode Select 0: Old mode of Sample Timing 1: New mode of Sample Timing Cannot be written when XCH=1.

Offset:0x0004			Register Name: SPI_GCR
Bit	**Read/Write**	**Default/Hex**	**Description**
1	R/W	0x0	MODE SPI Function Mode Select 0: Slave mode 1: Master mode Cannot be written when XCH=1.
0	R/W	0x0	EN SPI Module Enable Control 0: Disable 1: Enable After transforming from bit_mode to byte_mode, it must enable the SPI module again.

9.3.6.2 0x0008 SPI Transfer Control Register (Default Value: 0x0000_0087)

Offset: 0x0008			Register Name: SPI_TCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/WAC	0x0	XCH Exchange Burst In master mode, it is used to start SPI burst. 0: Idle 1: Initiates exchange. Writing “1” to this bit will start the SPI burst, and will auto-clear after finishing the bursts transfer specified by SPI_MBC. Writing “1” to SRST (SPI_GCR[31]) will also clear this bit. Writing ‘0’ to this bit has no effect. Cannot be written when XCH=1.
30:16	/	/	/
15	R/W	0x0	SDC1 Master Sample Data Control register1 Set this bit to ‘1’ to make the internal read sample point with a delay of half-cycle of SPI_CLK. It is used in high speed read operation to reduce the error caused by the time delay of SPI_CLK between master and slave. 0: normal operation, do not delay the internal read sample point 1: delay the internal read sample point Cannot be written when XCH=1.

Offset: 0x0008			Register Name: SPI_TCR
Bit	**Read/Write**	**Default/Hex**	**Description**
14	R/W	0x0	SDDM Sending Data Delay Mode 0: Normal sending 1: Delay sending Set the bit to "1" to make the data that should be sent with a delay of half-cycle for SPI_CLK in dual IO mode of SPI mode0. Cannot be written when XCH=1.
13	R/W	0x0	SDM Master Sample Data Mode 0: Delay sample mode 1: Normal sample mode In normal sample mode, the SPI master samples the data at the correct edge for each SPI mode; In delay sample mode, the SPI master samples data at the edge that is half cycle delayed by the correct edge defined in respective SPI mode. Cannot be written when XCH=1.
12	R/W	0x0	FBS First Transmit Bit Select 0: MSB first 1: LSB first Cannot be written when XCH=1.
11	R/W	0x0	SDC Master Sample Data Control Set this bit to '1' to make the internal read sample point with a delay of half-cycle for SPI_CLK. It is used in high speed read operation to reduce the error caused by the time delay of SPI_CLK between master and slave. 0: Normal operation, do not delay the internal read sample point 1: Delay the internal read sample point Cannot be written when XCH=1.
10	R/W	0x0	RPSM Rapids Mode Select Select rapid mode for high speed write. 0: Normal write mode 1: Rapid write mode Cannot be written when XCH=1.

Offset: 0x0008			Register Name: SPI_TCR
Bit	**Read/Write**	**Default/Hex**	**Description**
9	R/W	0x0	DDB Dummy Burst Type 0: The bit value of dummy SPI burst is zero 1: The bit value of dummy SPI burst is one Cannot be written when XCH=1.
8	R/W	0x0	DHB Discard Hash Burst In master mode, it controls whether discarding unused SPI bursts 0: Receiving all SPI bursts in the BC period 1: Discard unused SPI bursts, only fetching the SPI bursts during the dummy burst period. The burst number is specified by TC. Cannot be written when XCH=1.
7	R/W	0x1	SS_LEVEL When control SS signal manually (SS_OWNER (SPI_TCR[6])==1), set this bit to '1' or '0' to control the level of SS signal. 0: Set SS to low 1: Set SS to high Cannot be written when XCH=1.
6	R/W	0x0	SS_OWNER SS Output Owner Select Usually, the controller sends the SS signal automatically with data together. When this bit is set to 1, the software must manually write SS_LEVEL (SPI_TCR[7]) to 1 or 0 to control the level of the SS signal. 0: SPI controller 1: Software Cannot be written when XCH=1.
5:4	R/W	0x0	SS_SEL SPI Chip Select Select one of four external SPI Master/Slave Devices 00: SPI_SS0 will be asserted 01: SPI_SS1 will be asserted 10: SPI_SS2 will be asserted 11: SPI_SS3 will be asserted Cannot be written when XCH=1.

Offset: 0x0008			Register Name: SPI_TCR
Bit	**Read/Write**	**Default/Hex**	**Description**
3	R/W	0x0	SSCTL In master mode, this bit selects the output waveform for the SPI_SSx signal. Only valid when SS_OWNER (SPI_TCR[6])= 0. 0: SPI_SSx remains asserted between SPI bursts 1: Negate SPI_SSx between SPI bursts Cannot be written when XCH=1.
2	R/W	0x1	SPOL SPI Chip Select Signal Polarity Control 0: Active high polarity (0 = Idle) 1: Active low polarity (1 = Idle) Cannot be written when XCH=1.
1	R/W	0x1	CPOL SPI Clock Polarity Control 0: Active high polarity (0 = Idle) 1: Active low polarity (1 = Idle) Cannot be written when XCH=1.
0	R/W	0x1	CPHA SPI Clock/Data Phase Control 0: Phase 0 (Leading edge for sample data) 1: Phase 1 (Leading edge for setup data) Cannot be written when XCH=1.

9.3.6.3 0x0010 SPI Interrupt Control Register (Default Value: 0x0000_0000)

Offset: 0x0010			Register Name: SPI_IER
Bit	**Read/Write**	**Default/Hex**	**Description**
31:14	/	/	/
13	R/W	0x0	SS_INT_EN SSI Interrupt Enable Chip select signal (SSx) from the valid state to the invalid state 0: Disable 1: Enable

Offset: 0x0010			Register Name: SPI_IER
Bit	**Read/Write**	**Default/Hex**	**Description**
12	R/W	0x0	TC_INT_EN Transfer Completed Interrupt Enable 0: Disable 1: Enable
11	R/W	0x0	TF_UDR_INT_EN TXFIFO Underrun Interrupt Enable 0: Disable 1: Enable
10	R/W	0x0	TF_OVF_INT_EN TX FIFO Overflow Interrupt Enable 0: Disable 1: Enable
9	R/W	0x0	RF_UDR_INT_EN RXFIFO Underrun Interrupt Enable 0: Disable 1: Enable
8	R/W	0x0	RF_OVF_INT_EN RX FIFO Overflow Interrupt Enable 0: Disable 1: Enable
7	/	/	/
6	R/W	0x0	TF_FUL_INT_EN TX FIFO Full Interrupt Enable 0: Disable 1: Enable
5	R/W	0x0	TX_EMP_INT_EN TX FIFO Empty Interrupt Enable 0: Disable 1: Enable
4	R/W	0x0	TX_ERQ_INT_EN TX FIFO Empty Request Interrupt Enable 0: Disable 1: Enable
3	/	/	/

Offset: 0x0010			Register Name: SPI_IER
Bit	**Read/Write**	**Default/Hex**	**Description**
2	R/W	0x0	RF_FUL_INT_EN RX FIFO Full Interrupt Enable 0: Disable 1: Enable
1	R/W	0x0	RX_EMP_INT_EN RX FIFO Empty Interrupt Enable 0: Disable 1: Enable
0	R/W	0x0	RF_RDY_INT_EN RX FIFO Ready Request Interrupt Enable 0: Disable 1: Enable

9.3.6.4 0x0014 SPI Interrupt Status Register (Default Value: 0x0000_0032)

Offset: 0x0014			Register Name: SPI_ISR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:14	/	/	/
13	R/W1C	0x0	SSI SS Invalid Interrupt When SSI is 1, it indicates that SPI_SS has changed from valid state to invalid state. Writing 1 to this bit clears it.
12	R/W1C	0x0	TC Transfer Completed In master mode, it indicates that all bursts specified by SPI_MBC have been exchanged. In other conditions, when setting, this bit indicates that all the data in TXFIFO has been loaded in the Shift register, and the Shift register has shifted out all the bits. Writing 1 to this bit clears it. 0: Busy 1: Transfer completed

Offset: 0x0014			Register Name: SPI_ISR
Bit	**Read/Write**	**Default/Hex**	**Description**
11	R/W1C	0x0	TF_UDF TXFIFO Underrun This bit is set when the TXFIFO is underrun. Writing 1 to this bit clears it. 0: TXFIFO is not underrun 1: TXFIFO is underrun
10	R/W1C	0x0	TF_OVF TXFIFO Overflow This bit is set when the TXFIFO is overflowed. Writing 1 to this bit clears it. 0: TXFIFO is not overflowed 1: TXFIFO is overflowed
9	R/W1C	0x0	RX_UDF RXFIFO Underrun When set, this bit indicates that RXFIFO is underrun. Writing 1 to this bit clears it. 0: RXFIFO is not underrun 1: RXFIFO is underrun
8	R/W1C	0x0	RX_OVF RXFIFO Overflow When set, this bit indicates that RXFIFO has overflowed. Writing 1 to this bit clears it. 0: RXFIFO is not overflowed 1: RXFIFO is overflowed
7	/	/	/
6	R/W1C	0x0	TX_FULL TXFIFO Full This bit is set when the TXFIFO is full. Writing 1 to this bit clears it. 0: TXFIFO is not Full 1: TXFIFO is Full
5	R/W1C	0x1	TX_EMP TXFIFO Empty This bit is set when the TXFIFO is empty. Writing 1 to this bit clears it. 0: TXFIFO contains one or more words 1: TXFIFO is empty

Offset: 0x0014			Register Name: SPI_ISR
Bit	**Read/Write**	**Default/Hex**	**Description**
4	R/W1C	0x1	TX_READY TXFIFO Ready 0: TX_WL > TX_TRIG_LEVEL 1: TX_WL <= TX_TRIG_LEVEL This bit will be immediately set to 1 if TX_WL <= TX_TRIG_LEVEL. Writing "1" to this bit clears it. The TX_WL is the water level of TXFIFO.
3	/	/	/
2	R/W1C	0x0	RX_FULL RXFIFO Full This bit is set when the RXFIFO is full. Writing 1 to this bit clears it. 0: Not Full 1: Full
1	R/W1C	0x1	RX_EMP RXFIFO Empty This bit is set when the RXFIFO is empty. Writing 1 to this bit clears it. 0: Not empty 1: empty
0	R/W1C	0x0	RX_RDY RXFIFO Ready 0: RX_WL < RX_TRIG_LEVEL 1: RX_WL >= RX_TRIG_LEVEL This bit is will be immediately set to 1 if RX_WL >= RX_TRIG_LEVEL. Writing "1" to this bit clears it. The RX_WL is the water level of RXFIFO.

9.3.6.5 0x0018 SPI FIFO Control Register (Default Value: 0x0040_0001)

Offset: 0x0018			Register Name: SPI_FCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/WAC	0x0	TX_FIFO_RST TX FIFO Reset Writing '1' to this bit will reset the control portion of the TX FIFO and auto clear to '0' when completing reset operation, writing to '0' has no effect.

Offset: 0x0018			Register Name: SPI_FCR
Bit	**Read/Write**	**Default/Hex**	**Description**
30	R/W	0x0	TF_TEST_ENB TX Test Mode Enable 0: Disable 1: Enable In normal mode, the TXFIFO can only be read by the SPI controller, writing '1' to this bit will switch the read and write function of the TXFIFO to AHB bus. This bit is used to test the TXFIFO, do not set in normal operation, and do not set RF_TEST and TF_TEST at the same time.
29:25	/	/	/
24	R/W	0x0	TF_DRQ_EN TXFIFO DMA Request Enable 0: Disable 1: Enable
23:16	R/W	0x40	TX_TRIG_LEVEL TXFIFO Empty Request Trigger Level
15	R/WAC	0x0	RF_RST RXFIFO Reset Writing '1' to this bit will reset the control portion of the receiver FIFO, and auto clear to '0' when completing reset operation, writing '0' to this bit has no effect.
14	R/W	0x0	RF_TEST RX Test Mode Enable 0: Disable 1: Enable In normal mode, the RXFIFO can only be written by the SPI controller, writing '1' to this bit will switch the read and write function of RXFIFO to AHB bus. This bit is used to test the RXFIFO, do not set in normal operation, and do not set RF_TEST and TF_TEST at the same time.
13:9	/	/	/
8	R/W	0x0	RF_DRQ_EN RX FIFO DMA Request Enable 0: Disable 1: Enable
7:0	R/W	0x1	RX_TRIG_LEVEL RX FIFO Ready Request Trigger Level

9.3.6.6 0x001C SPI FIFO Status Register (Default Value: 0x0000_0000)

Offset: 0x001C			Register Name: SPI_FSR
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R	0x0	TB_WR TX FIFO Write Buffer Write Enable
30:28	R	0x0	TB_CNT TX FIFO Write Buffer Counter These bits indicate the number of words in TX FIFO Write Buffer
27:24	/	/	/
23:16	R	0x0	TF_CNT TX FIFO Counter These bits indicate the number of words in TX FIFO 0: 0 byte in TX FIFO 1: 1 byte in TX FIFO … 64: 64 bytes in TX FIFO other: Reserved
15	R	0x0	RB_WR RX FIFO Read Buffer Write Enable
14:12	R	0x0	RB_CNT RX FIFO Read Buffer Counter These bits indicate the number of words in RX FIFO Read Buffer
11:8	/	/	/
7:0	R	0x0	RF_CNT RX FIFO Counter These bits indicate the number of words in RX FIFO 0: 0 byte in RX FIFO 1: 1 byte in RX FIFO … 64: 64 bytes in RX FIFO other: Reserved

9.3.6.7 0x0020 SPI Wait Clock Register (Default Value: 0x0000_0000)

Offset: 0x0020			Register Name: SPI_WCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:20	/	/	/
19:16	R/W	0x0	SWC Dual mode direction switch wait clock counter (for master mode only). These bits control the number of wait states to be inserted before starting dual data transfer in dual SPI mode. The SPI module counts SPI_SCLK by SWC for delaying the next word data transfer. 0: No wait states inserted n: n SPI_SCLK wait states inserted Cannot be written when XCH=1.
15:0	R/W	0x0	WCC Wait Clock Counter (In master mode) These bits control the number of wait states to be inserted in data transfers. The SPI module counts SPI_SCLK by WCC for delaying the next word data transfer. 0: No wait states inserted n: n SPI_SCLK wait states inserted Cannot be written when XCH=1.

9.3.6.8 0x0028 SPI Sample Delay Control Register (Default Value: 0x0000_2000)

Offset: 0x0028			Register Name: SPI_SAMP_DL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:16	/	/	/
15	R/W	0x0	SAMP_DL_CAL_START Sample Delay Calibration Start When set, the sample delay chain calibration is started. Cannot be written when XCH=1.
14	R	0x0	SAMP_DL_CAL_DONE Sample Delay Calibration Done When set, it means that sample delay chain calibration is done and the result of calibration is shown in SAMP_DL. Cannot be written when XCH=1.

Offset: 0x0028			Register Name: SPI_SAMP_DL
Bit	**Read/Write**	**Default/Hex**	**Description**
13:8	R	0x20	SAMP_DL Sample Delay It indicates the number of delay cells corresponding to the current card clock. The delay time generated by these delay cells is equal to the cycle of the card clock nearly. Generally, it is necessary to do drive delay calibration when the card clock is changed. This bit is valid only when SAMP_DL_CAL_DONE is set. Cannot be written when XCH=1.
7	R/W	0x0	SAMP_DL_SW_EN Sample Delay Software Enable When set, it indicates that enable sample delay specified at SAMP_DL_SW. Cannot be written when XCH=1.
6	/	/	/
5:0	R/W	0x0	SAMP_DL_SW Sample Delay Software The relative delay between the clock line and command line, data lines. It can be determined according to the value of SAMP_DL, the cycle of the card clock, and the input timing requirement of the device. Cannot be written when XCH=1.

9.3.6.9 0x0030 SPI Master Burst Counter Register (Default Value: 0x0000_0000)

Offset: 0x0030			Register Name: SPI_MBC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/

Offset: 0x0030			Register Name: SPI_MBC
Bit	**Read/Write**	**Default/Hex**	**Description**
23:0	R/W	0x0	MBC Master Burst Counter In master mode, this field specifies the total burst number. The total transfer data include the TXD, RXD, and dummy burst. 0: 0 burst 1: 1 burst … N: N bursts Cannot be written when XCH=1.

9.3.6.10 0x0034 SPI Master Transmit Counter Register (Default Value: 0x0000_0000)

Offset: 0x0034			Register Name: SPI_MTC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:24	/	/	/
23:0	R/W	0x0	MWTC Master Write Transmit Counter In master mode, this field specifies the burst number that should be sent to TXFIFO before automatically sending dummy bursts. For saving bus bandwidth, the dummy bursts (all zero bits or all one bits) are sent by SPI Controller automatically. 0: 0 burst 1: 1 burst … N: N bursts Cannot be written when XCH=1.

9.3.6.11 0x0038 SPI Master Burst Control Counter Register (Default Value: 0x0000_0000)

Offset: 0x0038			Register Name: SPI_BCC
Bit	**Read/Write**	**Default/Hex**	**Description**
31:30	/	/	/

Offset: 0x0038			Register Name: SPI_BCC
Bit	**Read/Write**	**Default/Hex**	**Description**
29	R/W	0x0	Quad_EN Quad Mode Enable The quad mode includes Quad-Input and Quad-Output. 0: Quad mode disable 1: Quad mode enable Cannot be written when XCH=1.
28	R/W	0x0	DRM Master Dual Mode RX Enable It is only valid when Quad_Mode_EN=0. 0: RX uses the single-bit mode 1: RX uses the dual-bit mode Cannot be written when XCH=1.
27:24	R/W	0x0	DBC Master Dummy Burst Counter In master mode, this field specifies the burst number that should be sent before receiving in dual SPI mode. The data does not care by the device. 0: 0 burst 1: 1 burst … N: N bursts Cannot be written when XCH=1.
23:0	R/W	0x0	STC Master Single Mode Transmit Counter In master mode, this field specifies the burst number that should be sent in the single mode before automatically sending dummy bursts. This is the first transmit counter in all bursts. 0: 0 burst 1: 1 burst … N: N bursts Cannot be written when XCH=1.

9.3.6.12 0x0040 SPI Bit-Aligned Transfer Configure Register (Default Value: 0x0000_00A0)

Offset: 0x0040			Register Name: SPI_BATC
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/WAC	0x0	TCE Transfer Control Enable In master mode, it is used to start to transfer the serial bit frame, it is only valid when Work Mode Select==0x10/0x11. 0: Idle 1: Initiates transfer Writing "1" to this bit will start to transfer serial bit frame (the value comes from the SPI TX Bit Register or SPI RX Bit Register), and will auto-clear after the bursts transfer completely. Writing '0' to this bit has no effect.
30	R/W	0x0	MSMS Master Sample Standard 0: Delay Sample Mode 1: Standard Sample Mode In Standard Sample Mode, the SPI master samples the data at the standard rising edge of SCLK for each SPI mode; In Delay Sample Mode, the SPI master samples data at the edge that is half cycle delayed by the standard rising edge of SCLK defined in respective SPI mode.
29:26	/	/	/
25	R/W1C	0x0	TBC Transfer Bits Completed When set, this bit indicates that the last bit of the serial data frame in SPI TX Bit Register (or SPI RX Bit Register) has been transferred completely. Writing 1 to this bit clears it. 0: Busy 1: Transfer Completed It is only valid when Work Mode Select==0x10/0x11.
24	R/W	0x0	TBC_INT_EN Transfer Bits Completed Interrupt Enable 0: Disable 1: Enable It is only valid when Work Mode Select==0x10/0x11.
23:22	/	/	/

Offset: 0x0040			Register Name: SPI_BATC
Bit	**Read/Write**	**Default/Hex**	**Description**
21:16	R/W	0x00	Configure the length of serial data frame (burst) of RX 000000: 0 bit 000001: 1 bit … 100000: 32 bits Other values: reserved It is only valid when Work Mode Select==0x10/0x11, and cannot be written when TCE (bit31) is 1.
15:14	/	/	/
13:8	R/W	0x00	TX_FRM_LEN Configure the length of serial data frame (burst) of TX 000000: 0 bit 000001: 1 bit … 100000: 32 bits Other values: reserved It is only valid when Work Mode Select==0x10/0x11, and cannot be written when TCE (bit31) is 1.
7	R/W	0x1	SS_LEVEL When control the SS signal manually, set this bit to '1' or '0' to control the level of SS signal. 0: Set SS to low 1: Set SS to high It is only valid when Work Mode Select==0x10/0x11, and only work in Mode0, cannot be written when TCE (bit31) is 1.
6	R/W	0x0	SS_OWNER SS Output Owner Select Usually, the controller sends the SS signal automatically with data together. When this bit is set to 1, the software must manually write SS_LEVEL to 1 or 0 to control the level of the SS signal. 0: SPI controller 1: Software It is only valid when Work Mode Select==0x10/0x11, and only work in Mode0, cannot be written when TCE (bit31) is 1.

Offset: 0x0040			Register Name: SPI_BATC
Bit	**Read/Write**	**Default/Hex**	**Description**
5	R/W	0x1	SPOL SPI Chip Select Signal Polarity Control 0: Active high polarity (0 = Idle) 1: Active low polarity (1 = Idle) It is only valid when Work Mode Select==0x10/0x11, and only work in Mode0, cannot be written when TCE (bit31) is 1.
4	/	/	/
3:2	R/W	0x0	SS_SEL SPI Chip Select Select one of four external SPI Master/Slave Devices 00: SPI_SS0 will be asserted 01: SPI_SS1 will be asserted 10: SPI_SS2 will be asserted 11: SPI_SS3 will be asserted It is only valid when Work Mode Select= =0x10/0x11, and only work in Mode0, cannot be written when TCE (bit31) is 1.
1:0	R/W	0x0	Work Mode Select 00: Data frame is byte aligned in standard SPI, dual-output/dual input SPI, dual IO SPI, and quad-output/quad-input SPI 01: Reserved 10: Data frame is bit aligned in 3-wire SPI 11: Data frame is bit aligned in standard SPI

9.3.6.13 0x0044 SPI Bit-Aligned CLOCK Configuration Register (Default Value: 0x0000_0000)

Offset: 0x0044			Register Name: SPI_BA_CCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:0	R/W	0x0	CDR_N Clock Divide Rate (Master Mode Only) The SPI_SCLK is determined according to the following equation: SPI_CLK = Source_CLK / (2*(CDR_N + 1)). This register is only valid when Work Mode Select==0x10/0x11.

9.3.6.14 0x0048 SPI TX Bit Register (Default Value: 0x0000_0000)

Offset: 0x0048			Register Name: SPI_TBR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	VTB The Value of the Transmit Bits This register is used to store the value of the transmitted serial data frame. In the process of transmission, the LSB is transmitted first. This register is only valid when Work Mode Select==0x10/0x11.

9.3.6.15 0x004C SPI RX Bit Register (Default Value: 0x0000_0000)

Offset: 0x004C			Register Name: SPI_RBR
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	VRB The Value of the Receive Bits This register is used to store the value of the received serial data frame. In the process of transmission, the LSB is transmitted first. This register is only valid when Work Mode Select==0x10/0x11.

9.3.6.16 0x0088 SPI Normal DMA Mode Control Register (Default Value: 0x0000_00E5)

Offset: 0x0088			Register Name: SPI_NDMA_MODE_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
31:8	/	/	/
7:6	R/W	0x11	SPI_ACT_M SPI NDMA Active Mode 00: dma_active is low 01: dma_active is high 10: dma_active is controlled by dma_request (DRQ) 11: dma_active is controlled by controller
5	R/W	0x1	SPI_ACK_M SPI NDMA Acknowledge Mode 0: active fall do not care ack 1: active fall must after detect ack is high

Offset: 0x0088			Register Name: SPI_NDMA_MODE_CTL
Bit	**Read/Write**	**Default/Hex**	**Description**
4:0	R/W	0x05	SPI_DMA_WAIT The counts of hold cycles from DMA last signal high to dma_active high

9.3.6.17 0x0200 SPI TX Data Register (Default Value: 0x0000_0000)

Offset: 0x0200			Register Name: SPI_TXD
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R/W	0x0	TDATA Transmit Data This register can be accessed in the byte, half-word, or word unit by AHB. In the byte accessing method, if there are rooms in TXFIFO, one burst data is written to TXFIFO and the depth is increased by 1. In the half-word accessing method, two SPI burst data are written and the TXFIFO depth is increased by 2. In the word accessing method, four SPI burst data are written and the TXFIFO depth is increased by 4. **Note: This address is writable-only if TF_TEST is '0', and if TF_TEST is set to '1', this address is readable and writable to test the TXFIFO through the AHB bus.**

9.3.6.18 0x0300 SPI RX Data Register (Default Value: 0x0000_0000)

Offset: 0x0300			Register Name: SPI_RXD
Bit	**Read/Write**	**Default/Hex**	**Description**
31:0	R	0x0	RDATA Receive Data This register can be accessed in the byte, half-word, or word unit by AHB. In the byte accessing method, if there are data in RXFIFO, the top word is returned and the RXFIFO depth is decreased by 1. In the half-word accessing method, two SPI bursts are returned and the RXFIFO depth is decreased by 2. In the word accessing method, the four SPI bursts are returned and the RXFIFO depth is decreased by 4. **Note: This address is readable-only if RF_TEST is '0', and if RF_TEST is set to '1', this address is readable and writable to test the RXFIFO through the AHB bus.**

9.4 SPI_DBI

9.4.1 Overview

The T113-S3 provides a 3/4 line SPI display bus interface (SPI_DBI) for video data transmission. It supports DBI mode or SPI mode. The DBI mode is compatible with multiple video data formats at the same time. The SPI mode is used for low-cost display schemes.

The SPI mode has the following features:

- Full-duplex synchronous serial interface
- Master/slave configurable
- 8-bit wide by 64-entry FIFO for both transmit and receive data
- Polarity and phase of the Chip Select (SPI_SS) and SPI Clock (SPI_SCLK) are configurable
- Supports interrupts and DMA
- Supports mode0, mode1, mode2, and mode3
- Supports 3-wire/4-wire SPI
- Supports programmable serial data frame length: 1 bit to 32 bits
- Supports standard SPI, dual-output/dual-input SPI, dual I/O SPI, quad-output/quad-input SPI
- Supports maximum IO rate of the mass production: 100 MHz

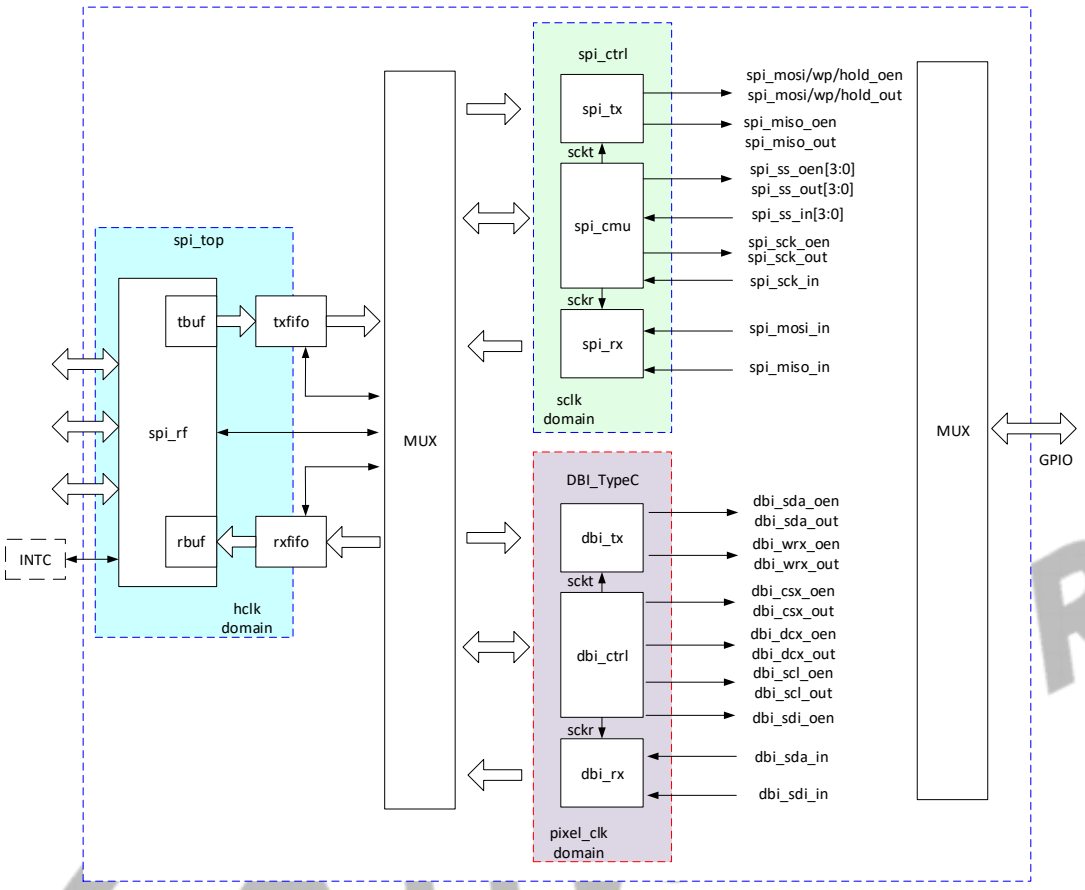
The DBI mode has the following features:

- Supports DBI Type C 3 Line/4 Line Interface Mode
- Supports 2 Data Lane Interface Mode
- Supports data source from CPU or DMA
- Supports RGB111/444/565/666/888 video format
- Maximum resolution of RGB666 240 x 320@30Hz with single data lane
- Maximum resolution of RGB888 240 x 320@60Hz or 320 x 480@30Hz with dual data lane
- Supports Tearing effect
- Supports software flexible control video frame rate

9.4.2 Block Diagram

Figure 9-30 shows a block diagram of the SPI_DBI.

Figure 9-30 SPI_DBI Block Diagram



SPI_DBI contains the following sub-blocks:

Table 9-12 SPI_DBI Sub-blocks

Sub-block	Description
spi_rf	Responsible for implementing the internal register, interrupt, and DMA Request.
spi_tbuf	The data length transmitted from AHB to TXFIFO is converted into 8 bits, then the data is written into the RXFIFO.
spi_rbuf	The block is used to convert the RXFIFO data into the reading data length of AHB.
txfifo, rxfifo	The data transmitted from the SPI to the external serial device is written into the TXFIFO; the data received from the external serial device into SPI is pushed into the RXFIFO.
spi_cmu	Responsible for implementing SPI bus clock, chip select, internal sample, and the generation of transfer clock.
spi_tx	Responsible for implementing SPI data transfer, the interface of the internal TXFIFO, and status register.

Sub-block	Description
spi_rx	Responsible for implementing SPI data receive, the interface of the internal RXFIFO, and status register.
dbi_ctrl	Responsible for implementing DBI bus clock, chip select, data command select, RGB format reshape.
dbi_tx	Responsible for implementing DBI data transfer, the interface of the internal TXFIFO, and status register.
dbi_rx	Responsible for implementing DBI data receive, the interface of the internal RXFIFO, and status register.

9.4.3 Functional Description

9.4.3.1 External Signals

The following table describes the external signals of SPI_DBI. When using SPI_DBI, the corresponding PADs are selected as SPI_DBI function via section 9.7 "GPIO".

Table 9-13 SPI_DBI External Signals

External Signal		Description	Type
DBI Mode	DBI-CSX	Chip select signal, low active	I/O
	DBI-SCLK	Serial clock signal	I/O
	DBI-SDO	Data output signal	I/O
	DBI-SDI	Data input signal, the data is sampled on the rising edge and the falling edge	I/O
	DBI-TE	Tearing effect input, it is used to capture the external TE signal edge.	I/O
	DBI-DCX	DCX pin is the select output signal of data and command. DCX = 0: register command; DCX = 1: data or parameter.	I/O
	DBI-WRX	When DBI operates in dual data lane format, the RGB666 format 2 can use WRX to transfer data	I/O
SPI Mode	SPI1-CS	SPI1 chip select signal, low active When the device is not selected, data will not be accepted via the SI pin, and the SO pin will stop transmission.	I/O
	SPI1-CLK	SPI1 clock signal This pin is used to provide a clock to the device and is used to control the flow of data to and from the device.	I/O
	SPI1-MOSI	SPI1 master data out, slave data in	I/O

External Signal		Description	Type
	SPI1-MISO	SPI1 master data in, slave data out	I/O
	SPI1-WP	Write protection and active low It also can be used for serial data input and output for SPI Quad Input or Quad Output mode.	I/O
	SPI1-HOLD	When the device is selected and a serial sequence is underway, the HOLD pin is can be used to temporarily pause the serial communication with the master device without deselecting or resetting the device serial sequence. While the HOLD pin is asserted, the SO pin is at high impedance, and all transitions on the SCK pin and data on the SI pin are ignored. It also can be used for serial data input and output for SPI Quad Input or Quad Output mode.	I/O

9.4.3.2 Clock Sources

The SPI_DBI controller gets 5 different clock sources, users can select one of them to make SPI_DBI clock source. The following table describes the clock sources for SPI_DBI. For more details on the clock setting, configuration, and gating information, see section 3.3 "CCU".

Table 9-14 SPI_DBI Clock Sources

Clock Sources	Description
HOSC	24 MHz Crystal
PLL_PERI(1X)	Peripheral Clock, the default value is 600 MHz
PLL_PERI(2X)	Peripheral Clock, the default value is 1200 MHz
PLL_AUDIO0(DIV2)	Audio Clock, the default value is 1536 MHz
PLL_AUDIO0(DIV5)	Audio Clock, the default value is 614.4 MHz

9.4.3.3 Typical Application

Figure 9-31 shows the application block diagram when the SPI master device is connected to a slave device.

Figure 9-31 SPI Application Block Diagram

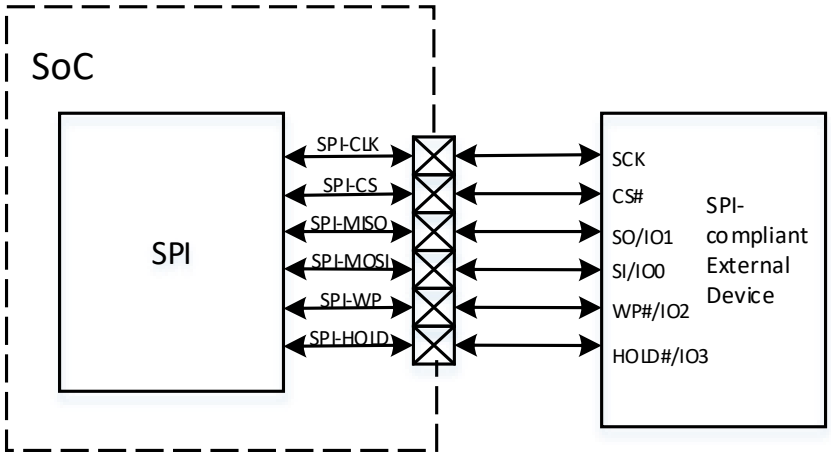
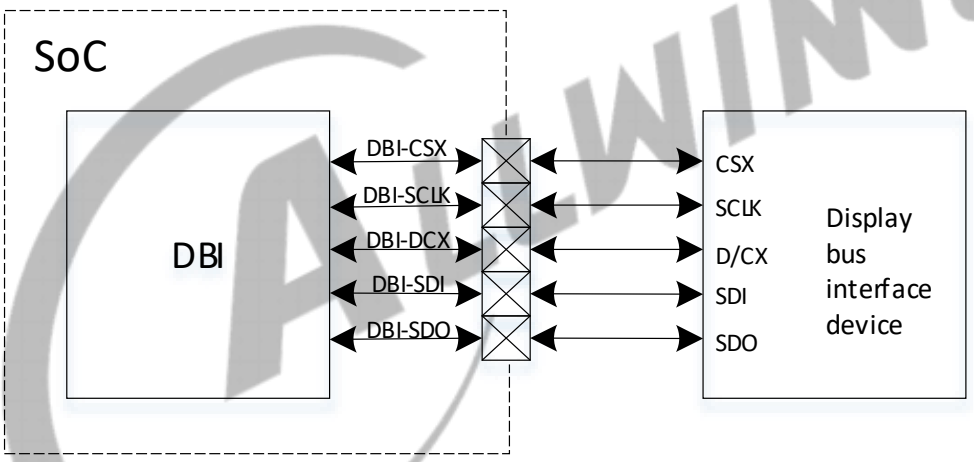


Figure 9-32 shows the application block diagram when the DBI master device is connected to a display bus interface device.

Figure 9-32 DBI Application Block Diagram



9.4.3.4 SPI Transmission Format

The SPI supports 4 different formats for data transmission. The software can select one of the four modes in which the SPI works by setting the bit1 (Polarity) and bit0 (Phase) of SPI_TCR. The SPI controller master uses the SPI_SCLK signal to transfer data in and out of the shift register. Data is clocked using any one of four programmable clock phase and polarity combinations.

The CPOL (SPI_TCR[1]) defines the polarity of the clock signal (SPI_SCLK). The SPI_SCLK is a high level when CPOL is '1' and it is a low level when CPOL is '0'. The CPHA (SPI_TCR[0]) decides whether the leading edge of SPI_SCLK is used to setup or sample data. The leading edge is used to setup data when CPHA is '1', and sample data when CPHA is '0'. The following table lists the four modes.

Table 9-15 SPI Transmit Format

SPI Mode	Polarity (CPOL)	Phase (CPHA)	Leading Edge	Trailing Edge
mode0	0	0	Sample on the rising edge	Setup on the falling edge
mode1	0	1	Setup on the rising edge	Sample on the falling edge
mode2	1	0	Sample on the falling edge	Setup on the rising edge
mode3	1	1	Setup on the falling edge	Sample on the rising edge

Figure 9-33 and Figure 9-34 describe four waveforms for SPI_SCLK.

Figure 9-33 SPI Phase 0 Timing Diagram

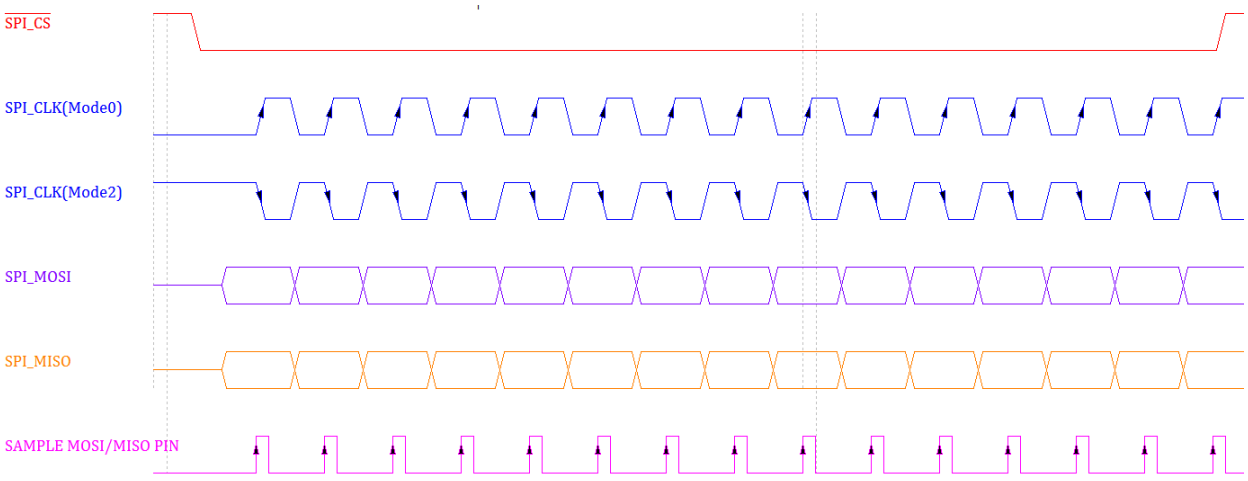
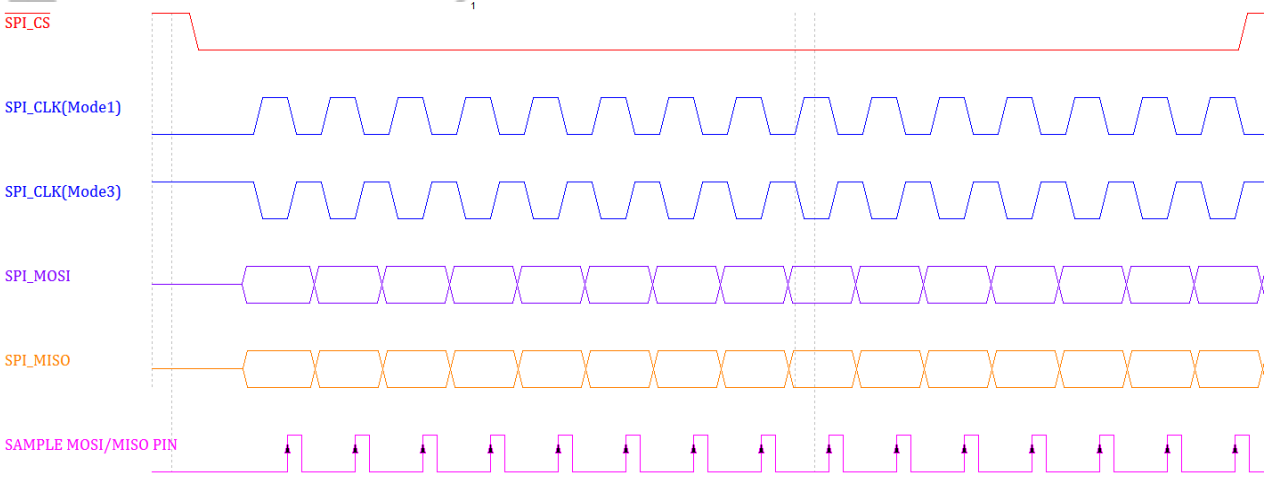


Figure 9-34 SPI Phase 1 Timing Diagram



9.4.3.5 SPI Master and Slave Mode

The SPI controller can be configured to a master or slave device. The master mode is selected by setting the MODE bit (SPI_GCR[1]); the slave mode is selected by clearing the MODE bit.

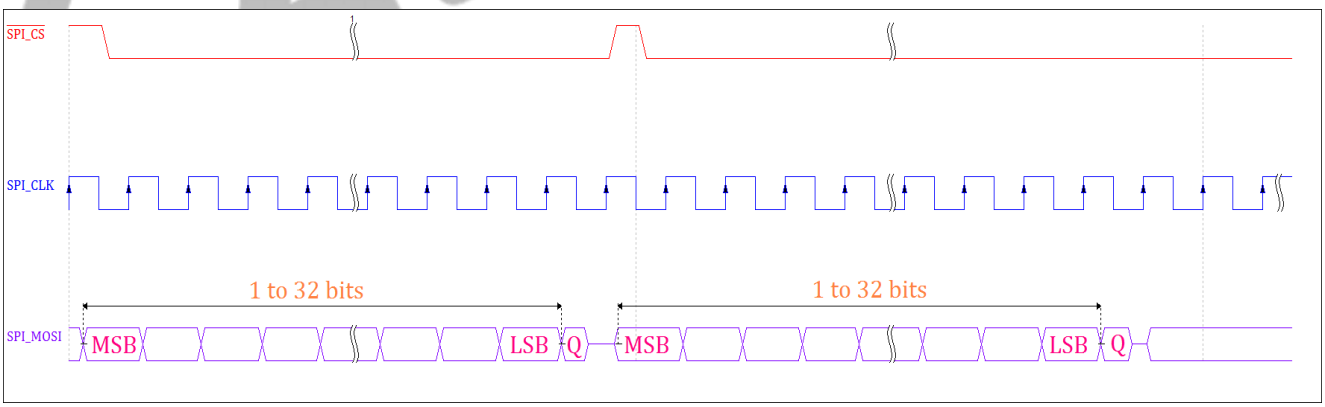
In master mode, the SPI_CLK is generated and transmitted to the external device, and the data from the TX FIFO is transmitted on the MOSI pin, the data from the slave is received on the MISO pin and sent to RX FIFO. The Chip Select (SPI_SS) is an active low signal, and it must be set low before the data are transmitted or received. The SPI_SS can be selected the auto control mode or the software manual control mode. When using auto control, the SS_OWNER (SPI_TCR[6]) must be cleared (default value is 0); when using manual control, the SS_OWNER must be set. And the level of SPI_SS is controlled by SS_LEVEL (SPI_TCR[7]).

In slave mode, after the software selects the MODE bit (SPI_GCR[1]) to '0', it waits for master initiate a transaction. When the master asserts SPI_SS, and SPI_CLK is transmitted to the slave, the slave data is transmitted from TX FIFO on the MISO pin, and the data from the MOSI pin is received in RX FIFO.

9.4.3.6 SPI 3-Wire Mode

The SPI 3-wire mode is only valid when the SPI controller work in master mode, and is selected when the Work Mode Select bit (SPI_BATC[1:0]) is equal to 0x2. And in the 3-wire mode, the input data and the output data use the same single data line. The following figure describes the 3-wire mode.

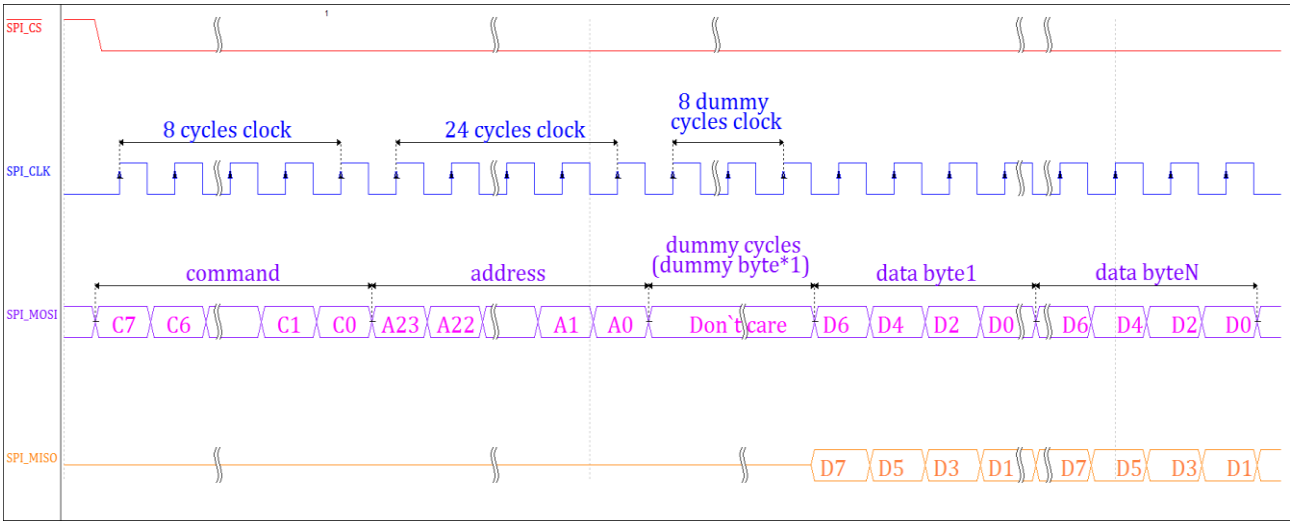
Figure 9-35 SPI 3-Wire Mode



9.4.3.7 SPI Dual-Input/Dual-Output and Dual I/O Mode

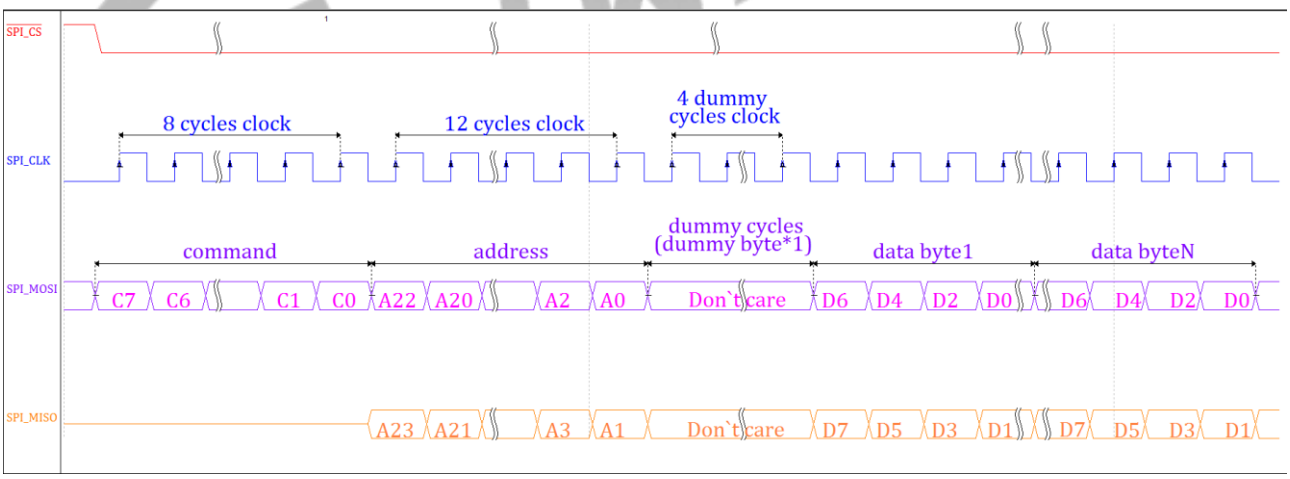
The dual read mode (SPI x2) is selected when the DRM is set in SPI_BCC[28]. Using the dual mode allows data to be transferred to or from the device at double the rate of standard single mode SPI devices, the data can be read at fast speed using two data bits (MOSI and MISO) at a time. The following figure describes the dual-input/dual-output SPI (Figure 9-36) and the dual I/O SPI (Figure 9-37).

Figure 9-36 SPI Dual-Input/Dual-Output Mode



In the dual-input/dual-output SPI mode, the command, address, and the dummy bytes output in a unit of a single bit in serial mode through the SPI_MOSI line, only the data bytes are output (write) and input (read) in a unit of dual bits through the SPI_MOSI and SPI_MISO.

Figure 9-37 SPI Dual I/O Mode

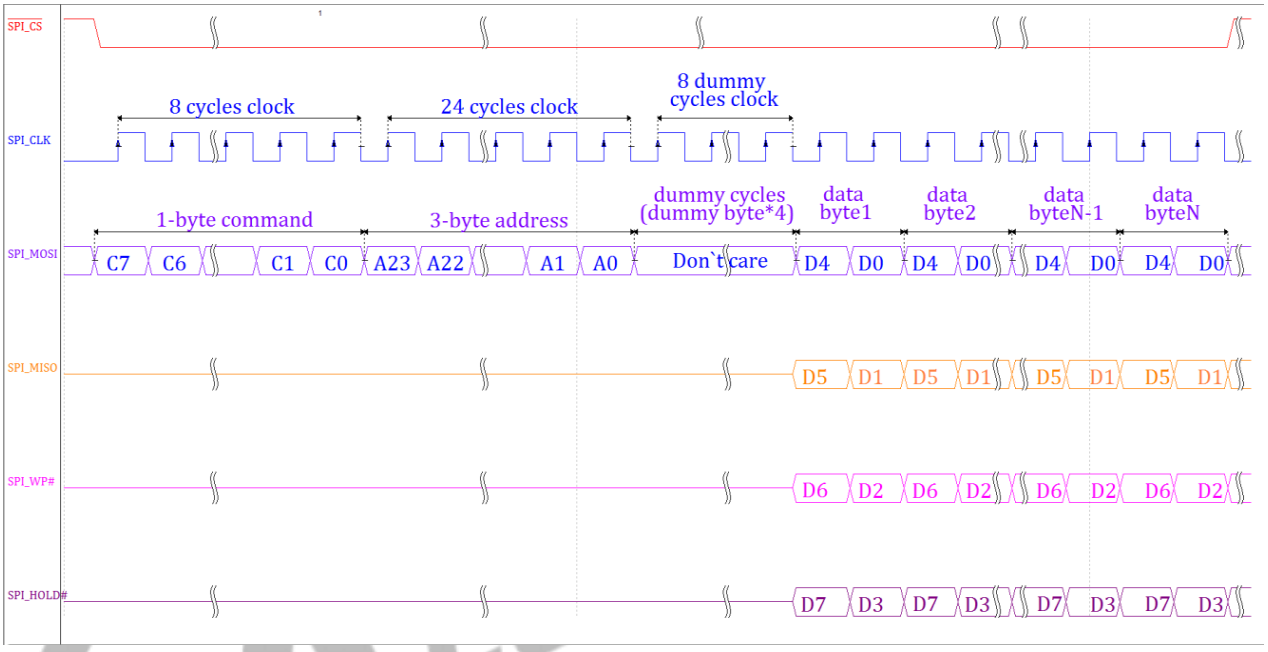


In the dual I/O SPI mode, only the command bytes output in a unit of a single bit in serial mode through the SPI_MOSI line. The address bytes and the dummy bytes output in a unit of dual bits through the SPI_MOSI and SPI_MISO. And the data bytes output (write) and input (read) in a unit of dual bits through the SPI_MOSI and SPI_MISO.

9.4.3.8 SPI Quad-Input/Quad-Output Mode

The quad read mode (SPI x4) is selected when the Quad_EN is set in **SPI_BCC**[29]. Using the quad mode allows data to be transferred to or from the device at 4 times the rate of standard single mode SPI devices, the data can be read at fast speed using four data bits (MOSI, MISO, IO2 (WP#) and IO3 (HOLD#)) at the same time. The following figure describes the quad-input/quad-output SPI.

Figure 9-38 SPI Quad-Input/Quad-Output Mode



In the quad-input/quad-output SPI mode, the command, address, and the dummy bytes output in a unit of a single bit in serial mode through the SPI_MOSI line. Only the data bytes output (write) and input (read) in a unit of quad bits through the SPI_MOSI, SPI_MISO, SPI_WP#, and SPI_HOLD#.

9.4.3.9 Transmission/Reception Bursts in Master Mode

In SPI master mode, the transmission and reception bursts (byte in unit) are configured before the SPI transfers serial data between the processor and external device. The transmission bursts are written in MWTC (bit[23:0]) of the SPI Master Transmit Counter Register**.** The transmission bursts in single mode before automatically sending dummy bursts are written in STC (bit[23:0]) of the SPI Master Transmit Counter Register**.** For dummy data, the SPI controller can automatically send before receiving by writing DBC (bit[27:24]) in the SPI Master Transmit Counter Register. If users do not use the SPI controller to send dummy data automatically, then the dummy bursts are used as the transmission counters to write together in MWTC (bit[23:0]) of the SPI Master Transmit Counter Register**.** In master mode, the total burst numbers are written in MBC (bit[23:0]) of the SPI Master Transmit Counter Register**.** When all transmission and reception bursts are transferred, the SPI controller will send a completed interrupt, at the same time, the SPI controller will clear DBC, MWTC, and MBC.

9.4.3.10 SPI Sample Mode and Run Clock Configuration

The SPI controller runs at 3 kHz–100 MHz at its interface to external SPI devices. The internal SPI clock should run at the same frequency as the outgoing clock in the master mode. The SPI clock is selected from different clock sources, the SPI must configure different work mode. There are three work modes: normal sample mode, delay half-cycle sample mode, delay one-cycle sample mode. Delay half-cycle sample mode is the default mode of the SPI controller. When the SPI runs at 40 MHz or below 40 MHz, the SPI can work at normal sample mode or delay half-cycle sample mode. When the SPI runs over 80 MHz, setting the SDC bit in the SPI Transfer Control Register to '1' makes the internal read sample point with a half-cycle delay of SPI_CLK, which is used in high speed read operation to reduce the error caused by the time delay of SPI_CLK between master and slave. Table 9-16 and Table 9-17 show the different configurations of the SPI sample mode.

Table 9-16 SPI Old Sample Mode and Run Clock

SPI Sample Mode	SDM(bit13)	SDC(bit11)	Run Clock
normal sample	1	0	<=24 MHz
delay half cycle sample	0	0	<=40 MHz
delay one cycle sample	0	1	>=80 MHz

CAUTION

The remaining spectrum is not recommended. Because when the output delay of SPI flash (refer to the datasheet of the manufactures for the specific delay time) is the same with the half-cycle time of SPI working clock, the variable edge of the output data for the device bumps into the clock sampling edge of the controller, so setting 1 cycle of sampling delay would cause stability problem.

Table 9-17 SPI New Sample Mode

SPI Sample Mode	SDM (bit13)	SDC (bit11)	SDC1 (bit15)
normal sample	1	0	0
delay half cycle sample	0	0	0
delay one cycle sample	0	1	0
delay 1.5 cycle sample	1	1	0
delay 2 cycle sample	1	0	1
delay 2.5 cycle sample	0	0	1
delay 3 cycle sample	0	1	1

9.4.3.11 DBI 3-Line Interface Writing and Reading Timing

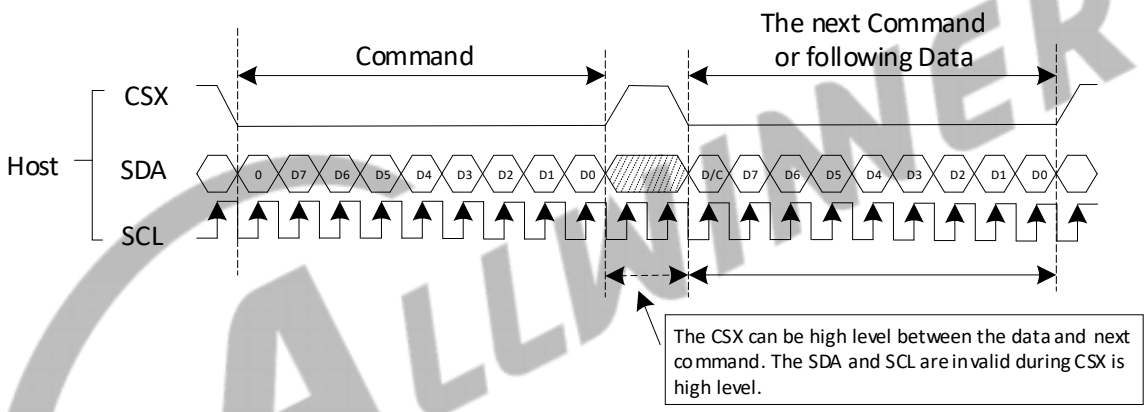
The 3-line DBI Interface I contains CSX, SDA, and SCL, where SDA shares this port for bidirectional port data input and output.

The 3-line DBI Interface II contains CSX, SDA, SCL, and SDI; Data input and output ports are independent of each other.

Since the 3-line display bus mode has no Data/Command data line indicating whether Data or Command is currently being transmitted, an extra bit is added to the data-stream before MSB to indicate whether Data or Command is currently being transmitted. (0: Command, 1: Data)

The following figure shows the writing operation format of 3-line DBI Interface I and Interface II.

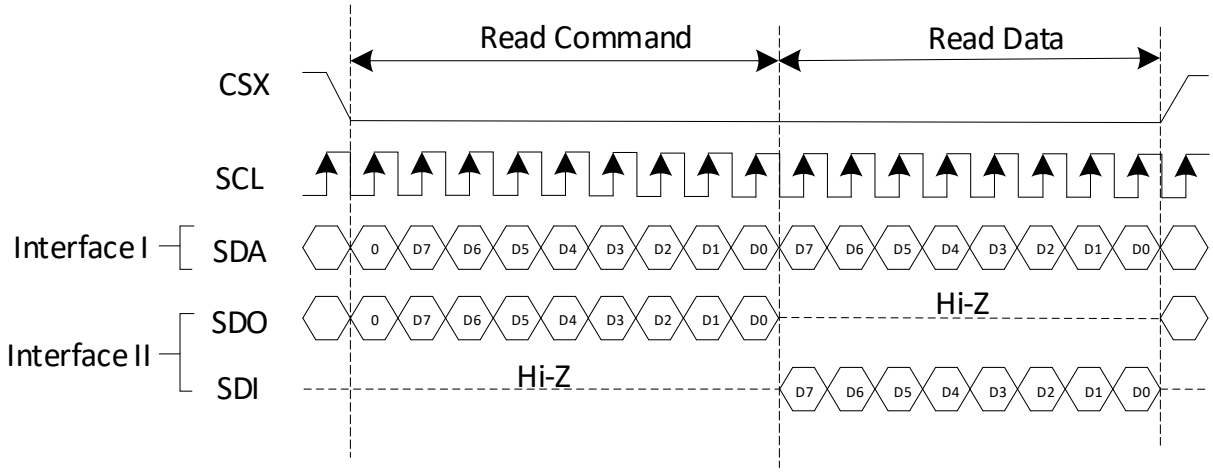
Figure 9-39 DBI 3-Line Display Bus Serial Interface Writing Operation Format



The 3-line DBI Interface I uses the SDA port as bidirectional data input and output port. There are only three cases of data reading volume, 8bits/24bits/32bits, and the first data sampled is high.

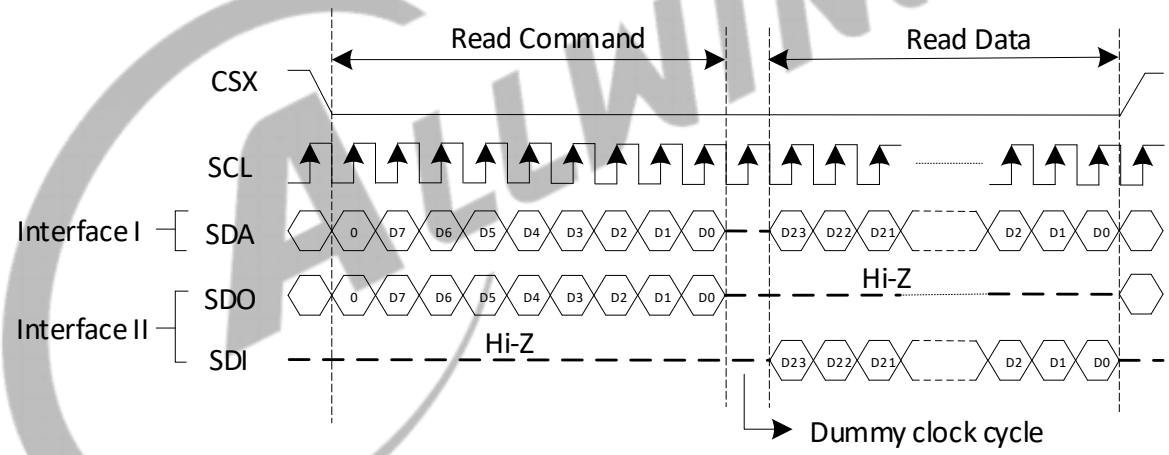
The following figure shows the 8 bits reading operation format of 3-line DBI Interface I and Interface II. After the read command is transmitted, the data is read immediately with on dummy period.

Figure 9-40 DBI 3-Line Display Bus Serial Interface 8-bit Reading Operation Format



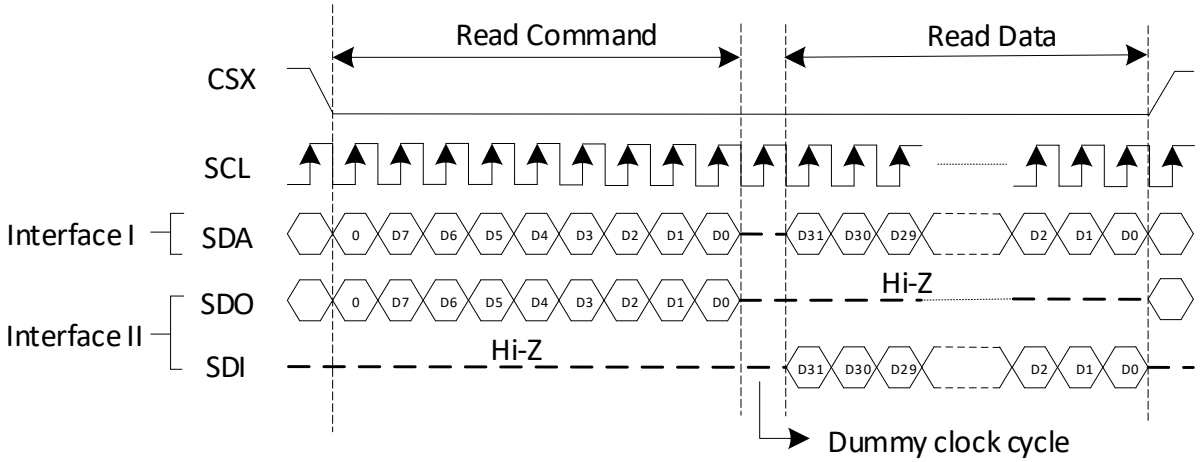
The following figure shows the 24 bits reading operation format of 3-line DBI Interface I and Interface II. After the read command is transmitted, the data is read after waiting for the dummy clock cycle.

Figure 9-41 DBI 3-Line Display Bus Serial Interface 24-bit Reading Operation Format



The following figure shows the 32 bits reading operation format of 3-line DBI Interface I and Interface II. After the read command is transmitted, the data is read after waiting for the dummy clock cycle.

Figure 9-42 DBI 3-Line Display Bus Serial Interface 32-bit Reading Operation Format



9.4.3.12 DBI 4-Line Interface Writing and Reading Timing

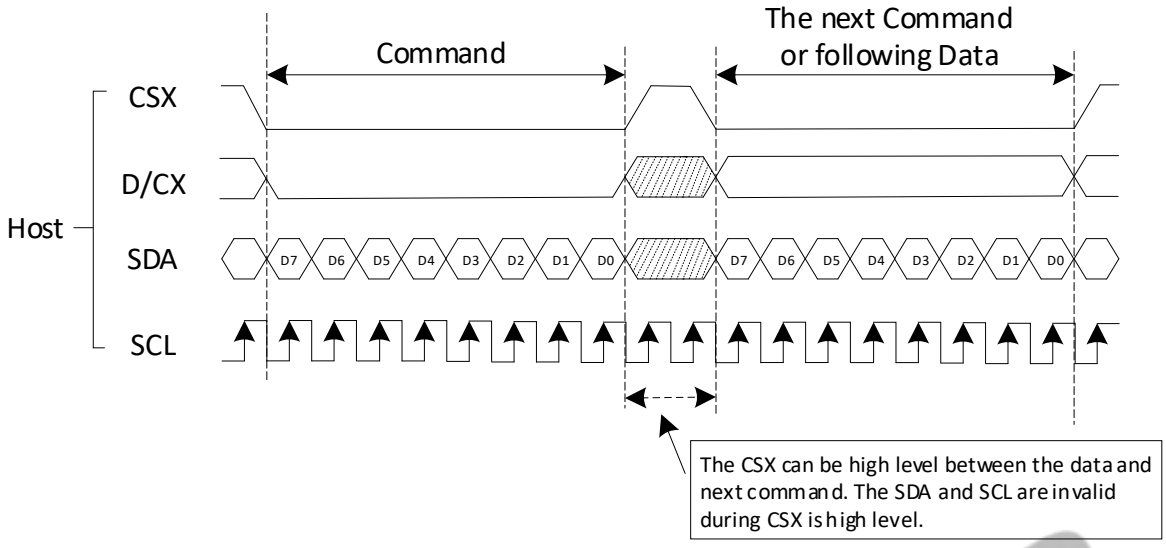
The 4-line DBI Interface I contains CSX, D/CX, SDA, and SCL, where SDA shares this port for bidirectional port data input and output.

The 4-line DBI Interface II contains CSX, D/CX, SDA, SCL, and SDI; Data input and output ports are independent of each other.

Since the 4-line display bus mode has a Data/Command data line indicating whether Data or Command is currently being transmitted (0: Command, 1: Data). So there is no need to add an extra bit to data-stream before MSB like the 3-line DBI.

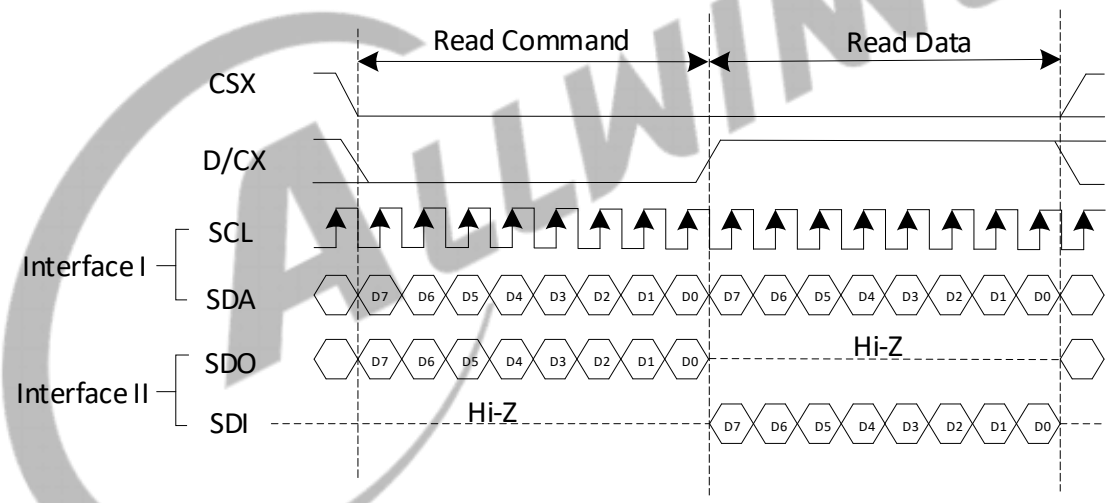
The following figure shows the writing operation format of 4-line DBI Interface I and Interface II.

Figure 9-43 DBI 4-Line Display Bus Serial Interface Writing Operation Format



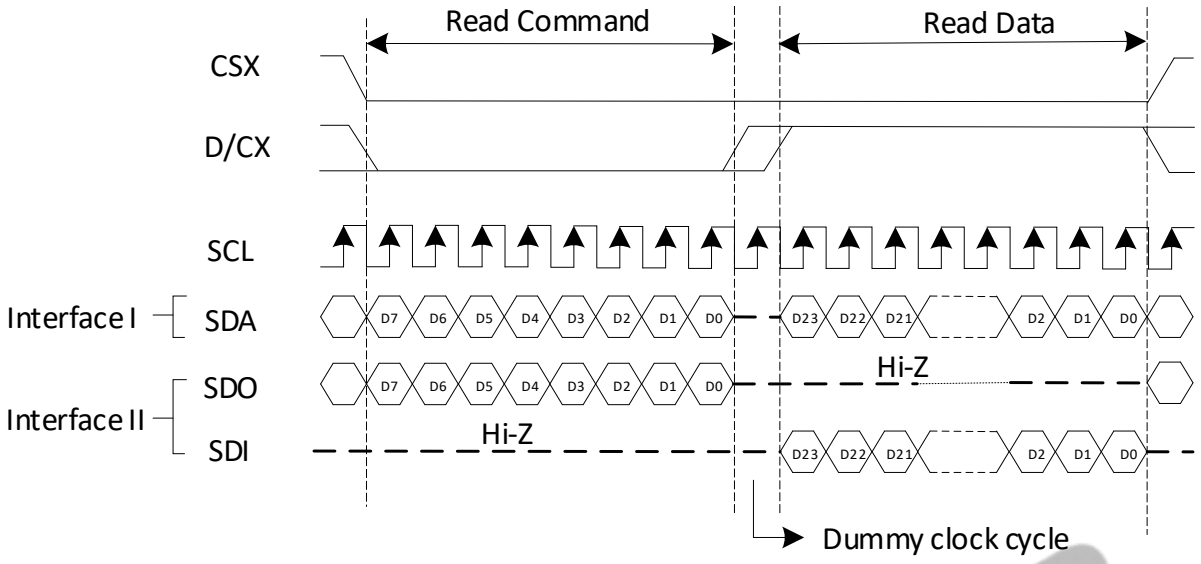
The following figure shows the 8 bits reading operation format of 4-line DBI Interface I and Interface II.

Figure 9-44 DBI 4-Line Display Bus Serial Interface 8-bit Reading Operation Format



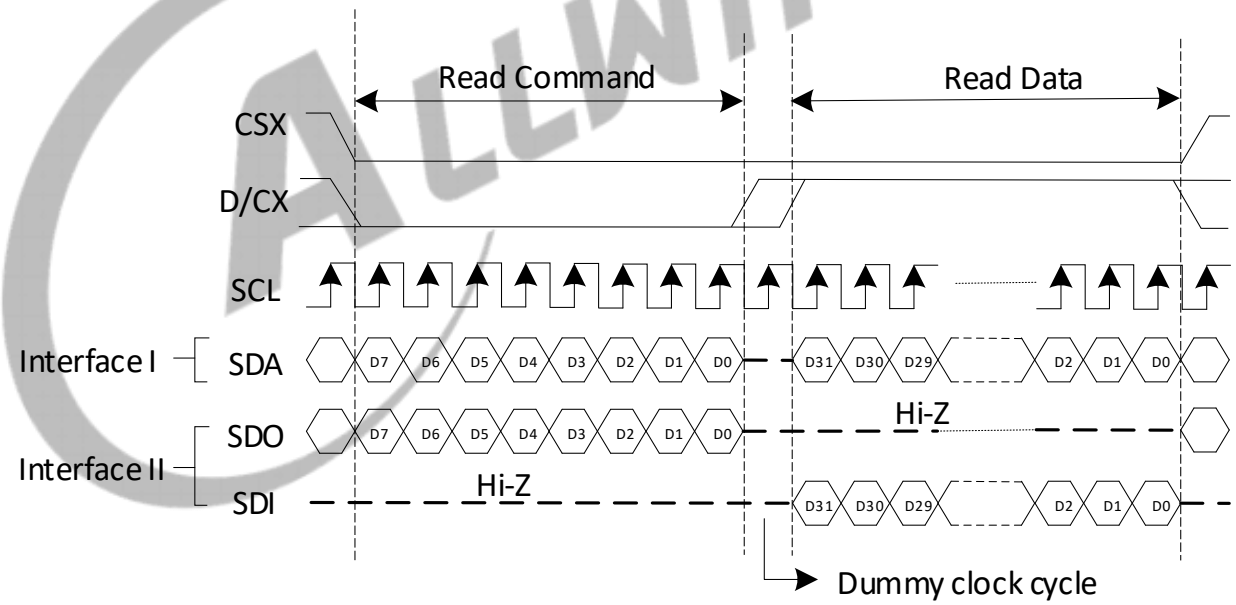
The following figure shows the 24 bits reading operation format of 4-line DBI Interface I and Interface II.

Figure 9-45 DBI 4-Line Display Bus Serial Interface 24-bit Reading Operation Format



The following figure shows the 32 bits reading operation format of 4-line DBI Interface I and Interface II.

Figure 9-46 DBI 4-Line Display Bus Serial Interface 32-bit Reading Operation Format



9.4.3.13 DBI 3-Line Interface Transmit Video Format

Figure 9-47 RGB111 3-Line Interface Transmit Video Format

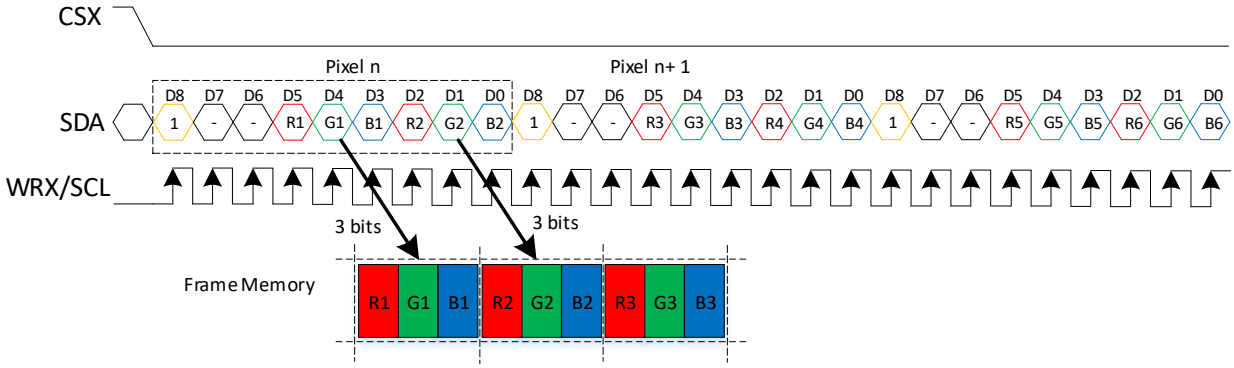
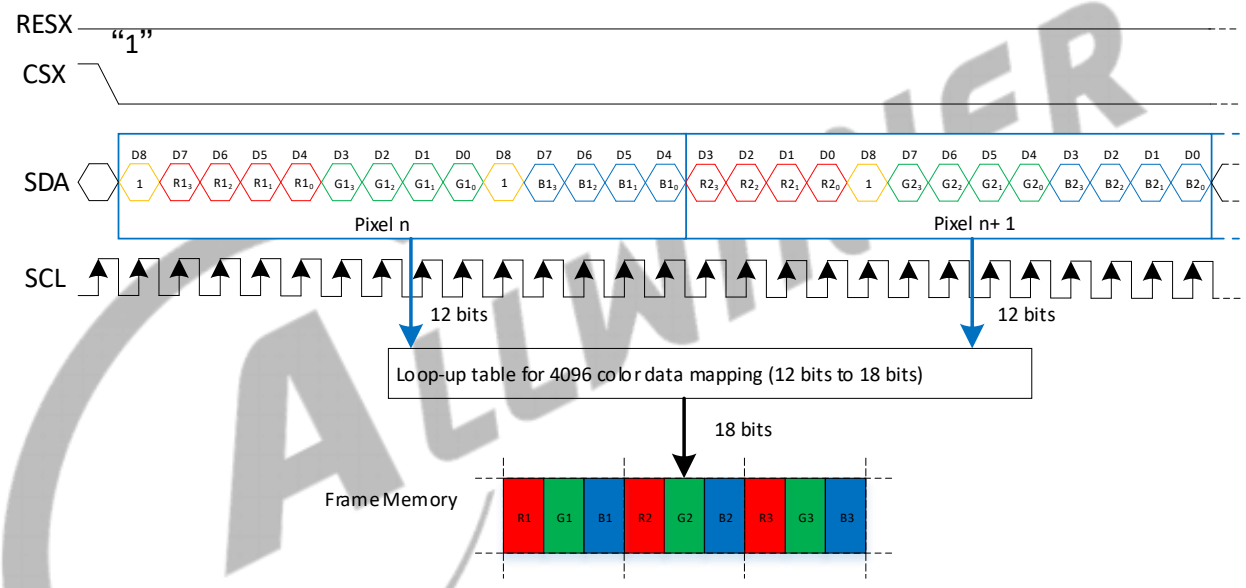
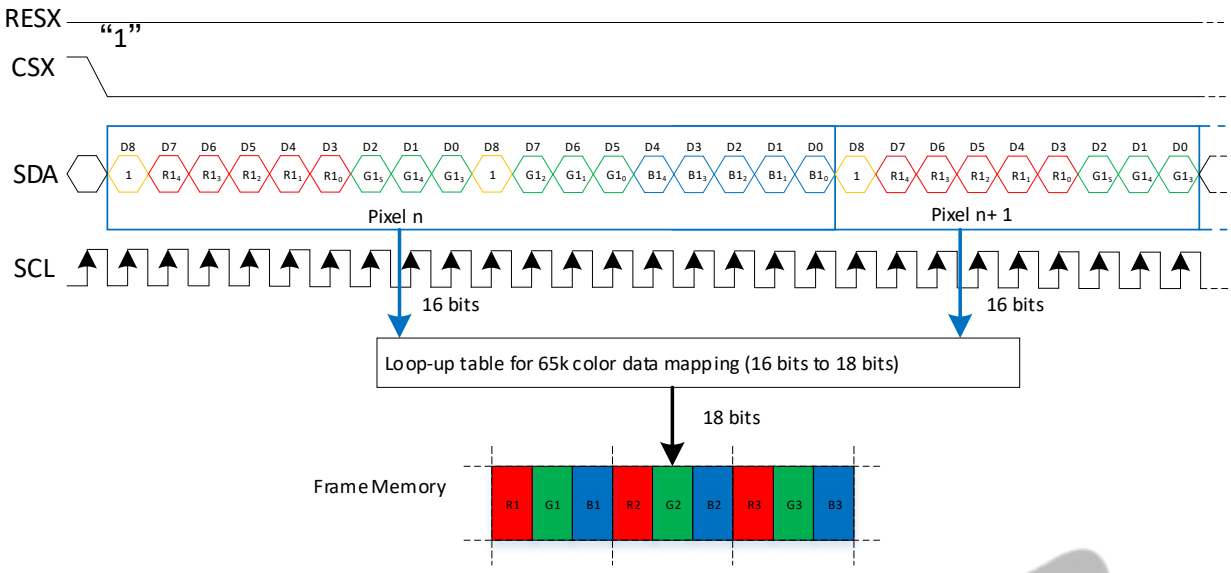


Figure 9-48 RGB444 3-Line Interface Transmit Video Format



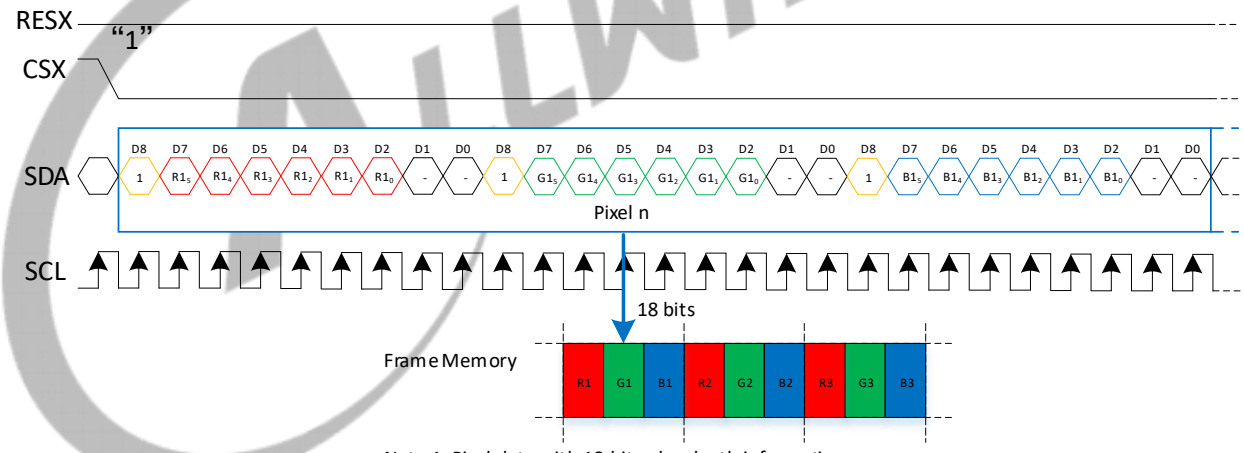
Note 1. Pixel data with 12-bit color depth information
Note 2. The most significant bits are: Rx3, Gx3 and Bx3
Note 3. The least significant bits are: Rx0, Gx0 and Bx0

Figure 9-49 RGB565 3-Line Interface Transmit Video Format



Note 1. Pixel data with 16-bit color depth information
Note 2. The most significant bits are: Rx4, Gx5 and Bx4
Note 3. The least significant bits are: Rx0, Gx0 and Bx0

Figure 9-50 RGB666 3-Line Interface Transmit Video Format



Note 1. Pixel data with 18-bit color depth information
Note 2. The most significant bits are: Rx5, Gx5 and Bx5
Note 3. The least significant bits are: Rx0, Gx0 and Bx0

9.4.3.14 DBI 4-Line Interface Transmit Video Format

Figure 9-51 RGB111 4-Line Interface Transmit Video Format

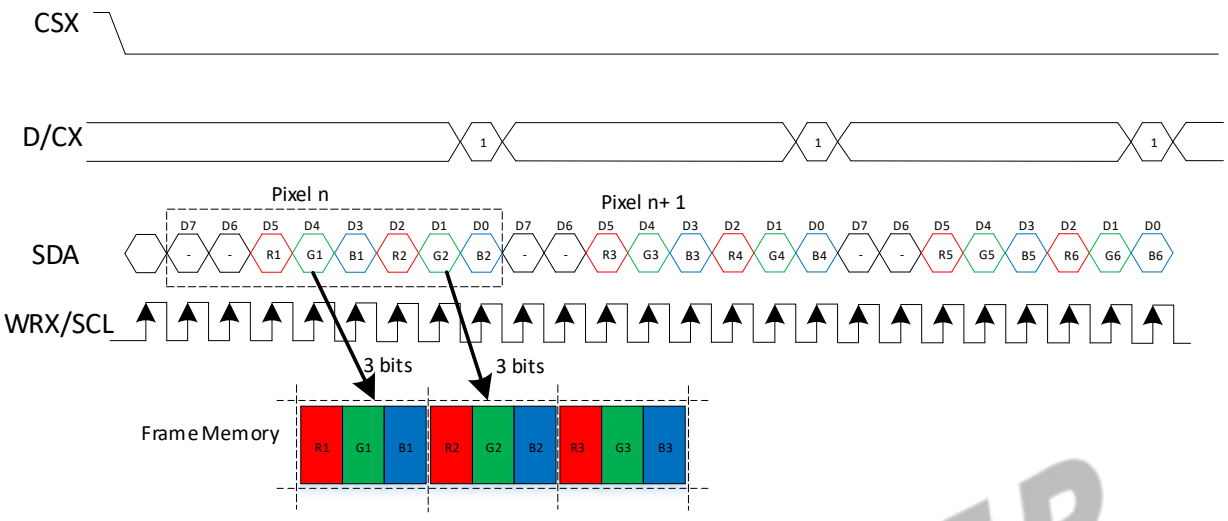
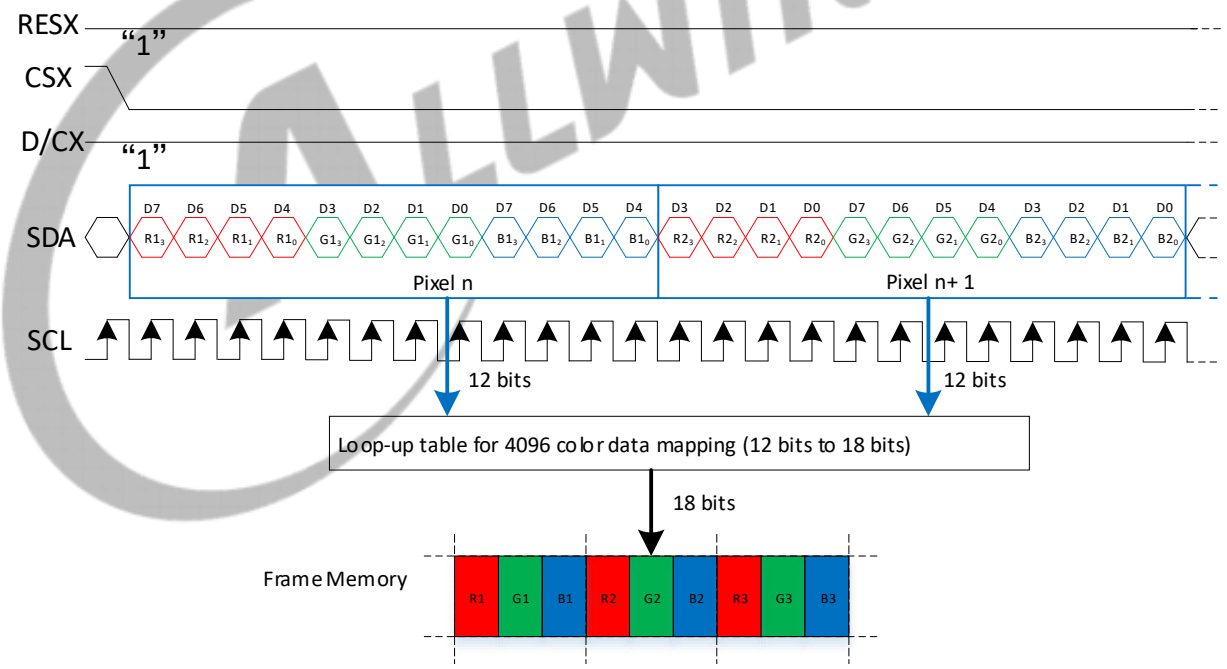
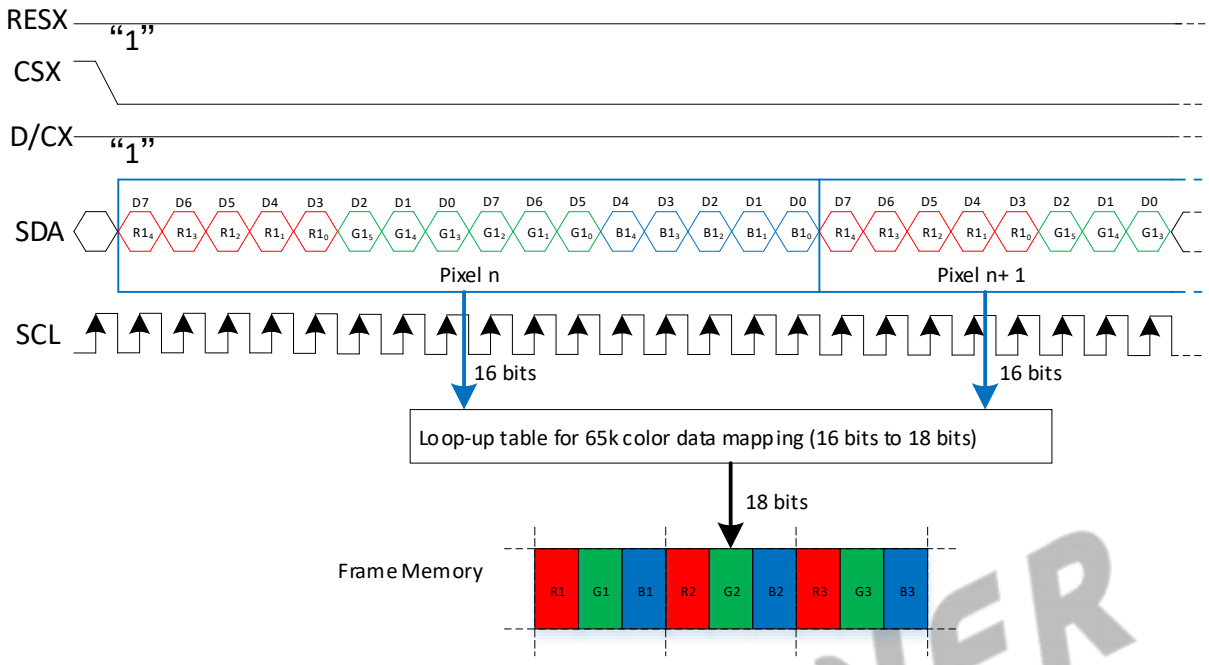


Figure 9-52 RGB444 4-Line Interface Transmit Video Format



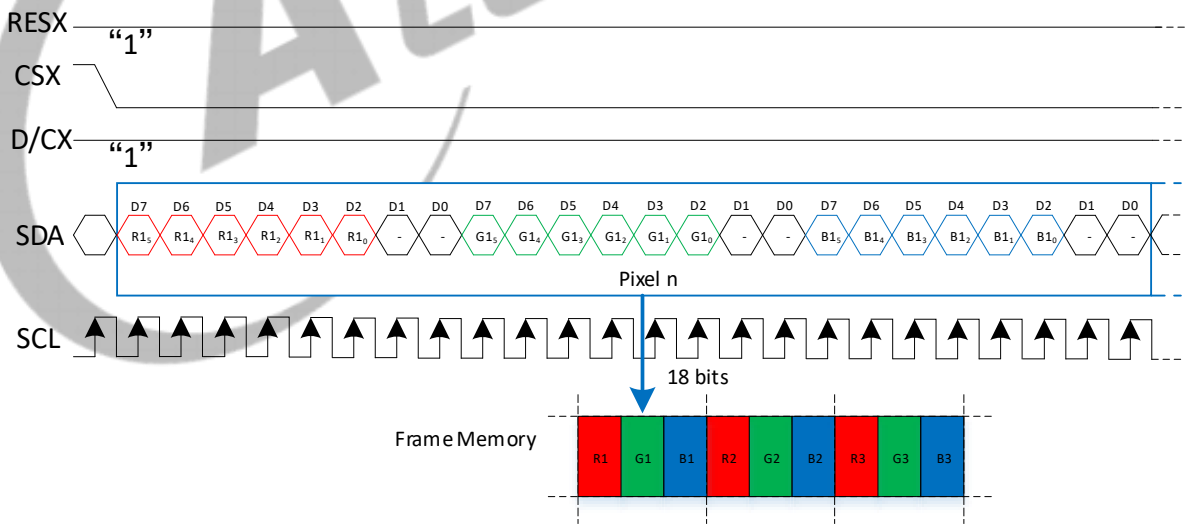
Note 1. Pixel data with 12-bit color depth information
Note 2. The most significant bits are: Rx3, Gx3 and Bx3
Note 3. The least significant bits are: Rx0, Gx0 and Bx0

Figure 9-53 RGB565 4-Line Interface Transmit Video Format



Note 1. Pixel data with 16-bit color depth information
Note 2. The most significant bits are: Rx4, Gx5 and Bx4
Note 3. The least significant bits are: Rx0, Gx0 and Bx0

Figure 9-54 RGB666 4-Line Interface Transmit Video Format

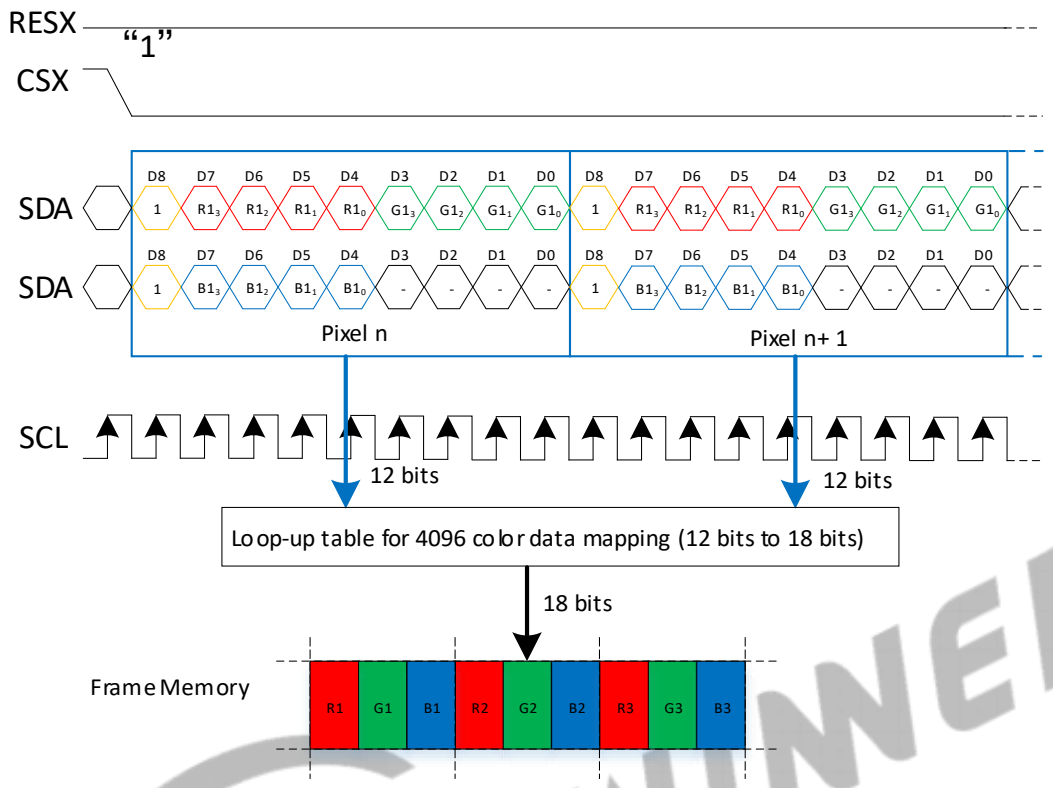


Note 1. Pixel data with 18-bit color depth information
Note 2. The most significant bits are: Rx5, Gx5 and Bx5
Note 3. The least significant bits are: Rx0, Gx0 and Bx0

9.4.3.15 DBI 2 Data Lane Interface Transmit Video Format

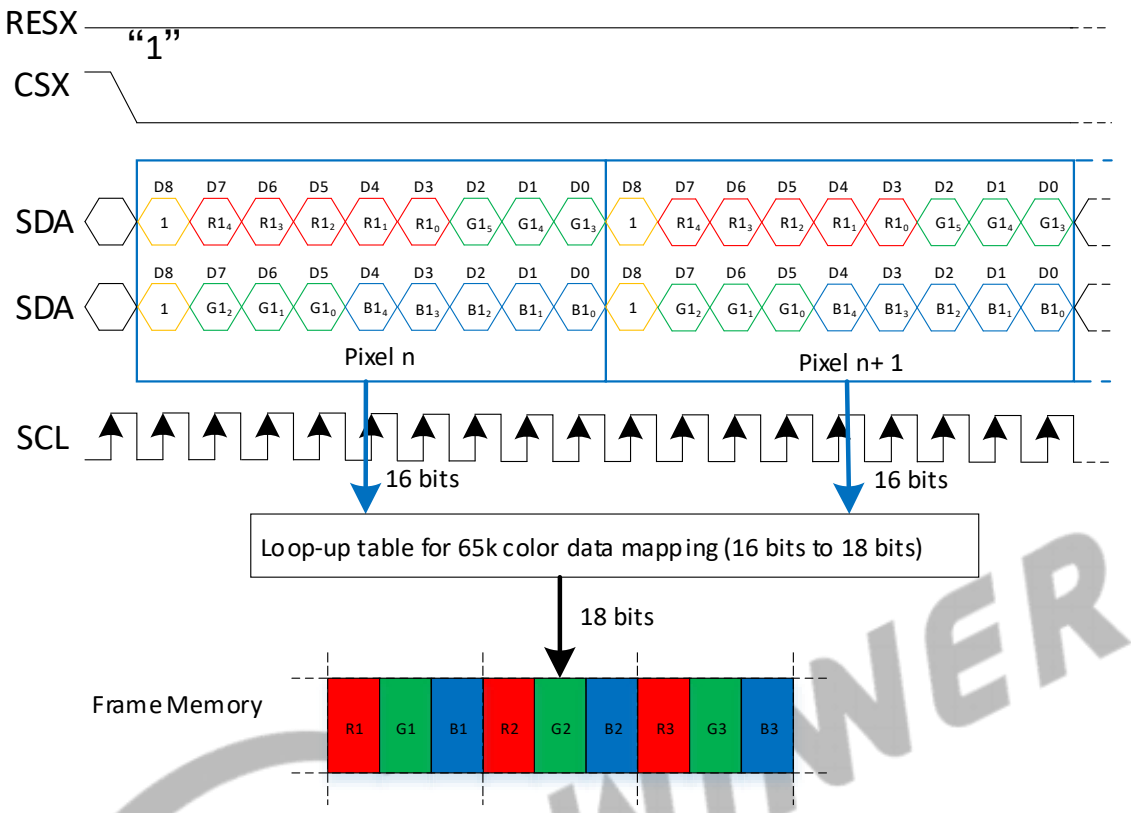
For RGB444:

Figure 9-55 RGB444 2 Data Lane Interface Transmit Video Format



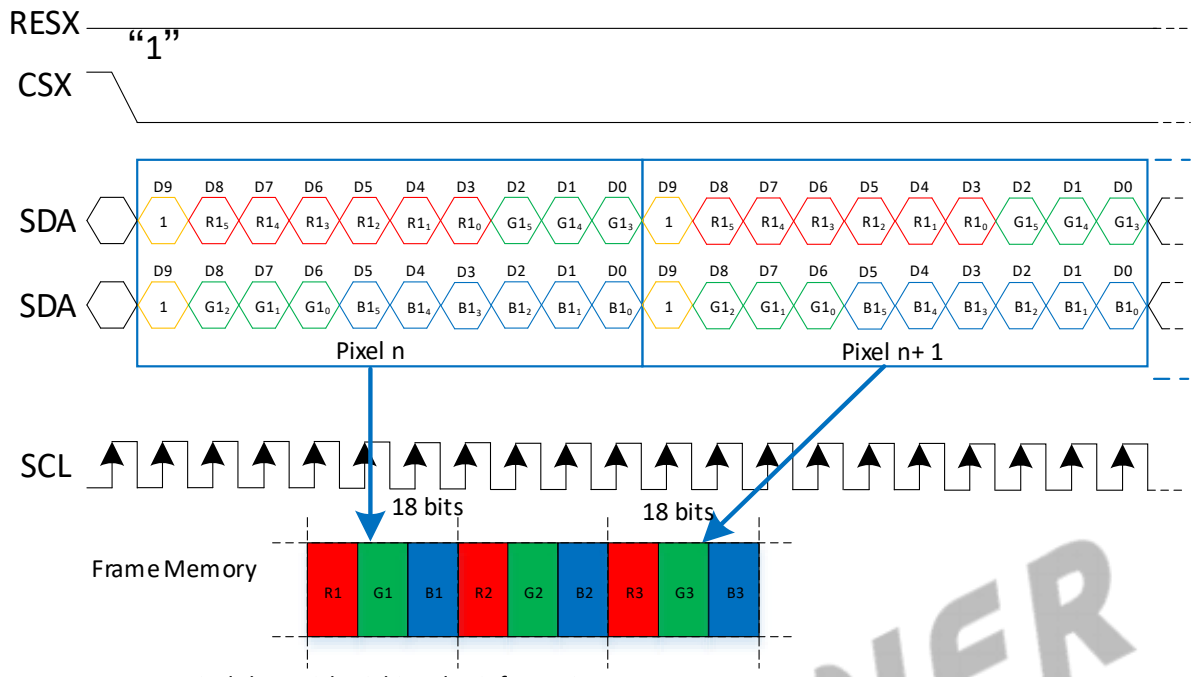
Note 1. Pixel data with 12-bit color information
Note 2. The most significant bits are: Rx3, Gx3 and Bx3
Note 3. The least significant bits are: Rx0, Gx0 and Bx0

Figure 9-56 RGB565 2 Data Lane Interface Transmit Video Format



Note 1. Pixel data with 16-bit color information
Note 2. The most significant bits are: Rx4, Gx5 and Bx4
Note 3. The least significant bits are: Rx0, Gx0 and Bx0

Figure 9-57 RGB666 2 Data Lane Interface Transmit Video Format 0



Note 1. Pixel data with 18-bit color information
Note 2. The most significant bits are: Rx5, Gx5 and Bx5
Note 3. The least significant bits are: Rx0, Gx0 and Bx0

Figure 9-58 RGB666 2 Data Lane Interface Transmit Video Format 1 (ilitek)

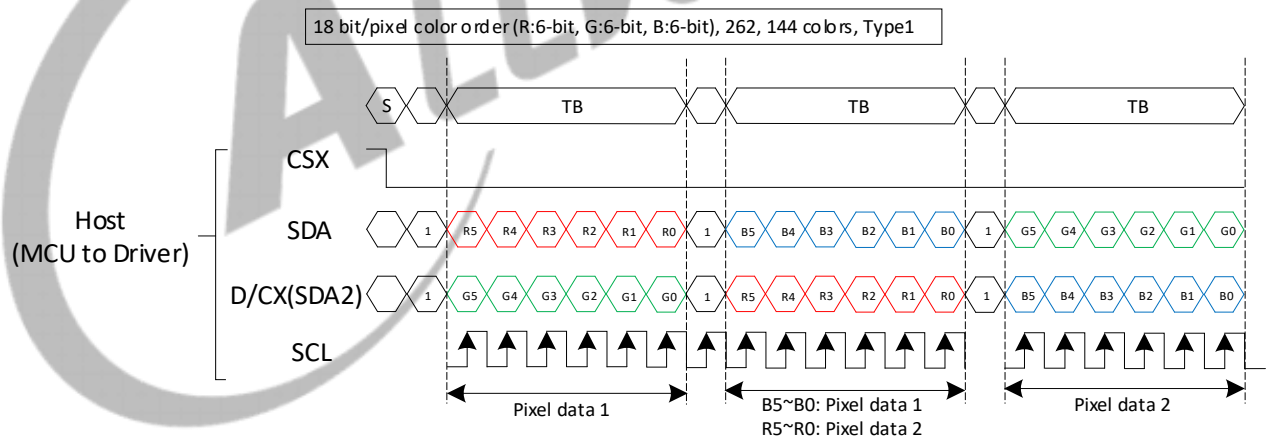


Figure 9-59 RGB666 2 Data Lane Interface Transmit Video Format 2 (New vision)

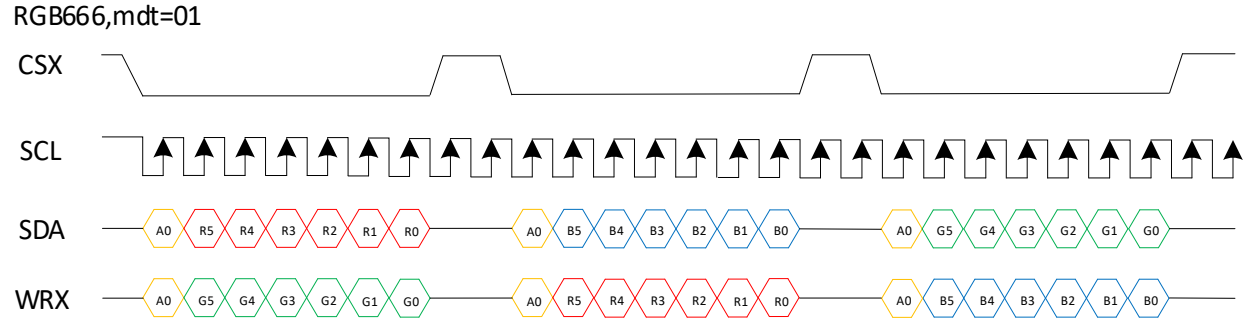
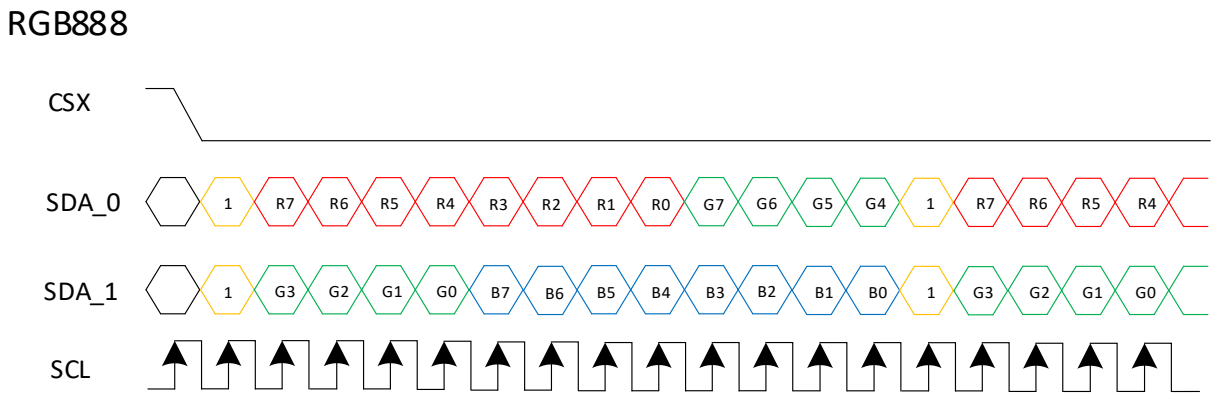


Figure 9-60 RGB888 2 Data Lane Interface Transmit Video Format



Note 1. Pixel data with 24-bit color information
Note 2. The most significant bits are: R7, G7 and B7
Note 3. The least significant bits are: R0, G0 and B0

9.4.4 Programming Guidelines

9.4.4.1 Writing/Reading Data Process Using SPI Mode

The SPI transfers serial data between the processor and the external device. CPU and DMA are the two main operational modes for SPI. For each SPI, the data is simultaneously transmitted (shifted out serially) and received (shifted in serially). The SPI has 2 channels, including the TX channel and RX channel. The TX channel has the path from TX FIFO to the external device. The RX channel has the path from the external device to RX FIFO.

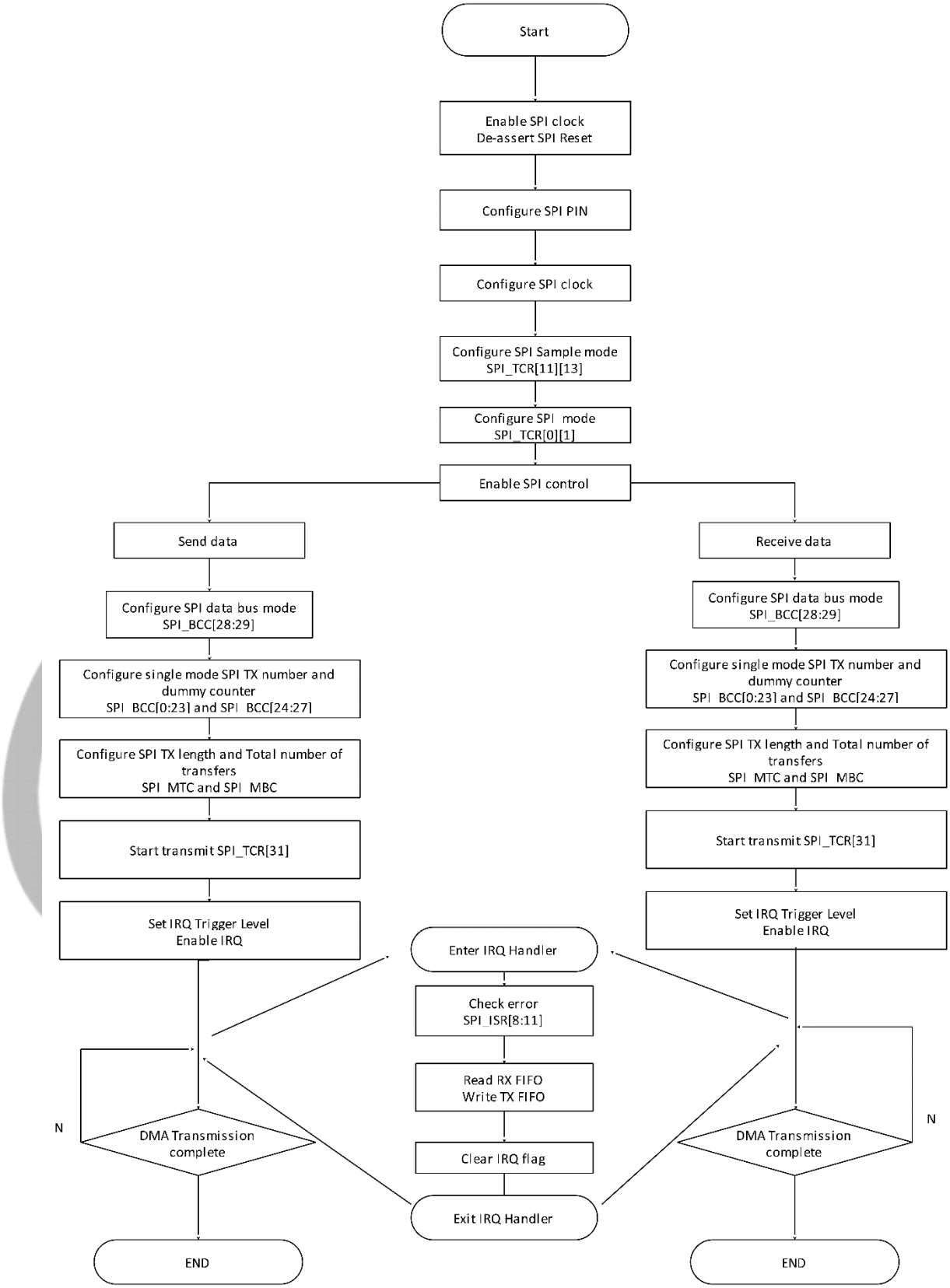
Write Data: CPU or DMA must write data on the **SPI_TXD** register, the data on the register are automatically moved to TX FIFO.

Read Data: To read data from RX FIFO, CPU or DMA must access the register **SPI_RXD** and data are automatically sent to the register **SPI_RXD**.

In CPU or DMA mode, the SPI sends a completed interrupt (**SPI_ISR**[TC]) to the processor at the end of each transfer.

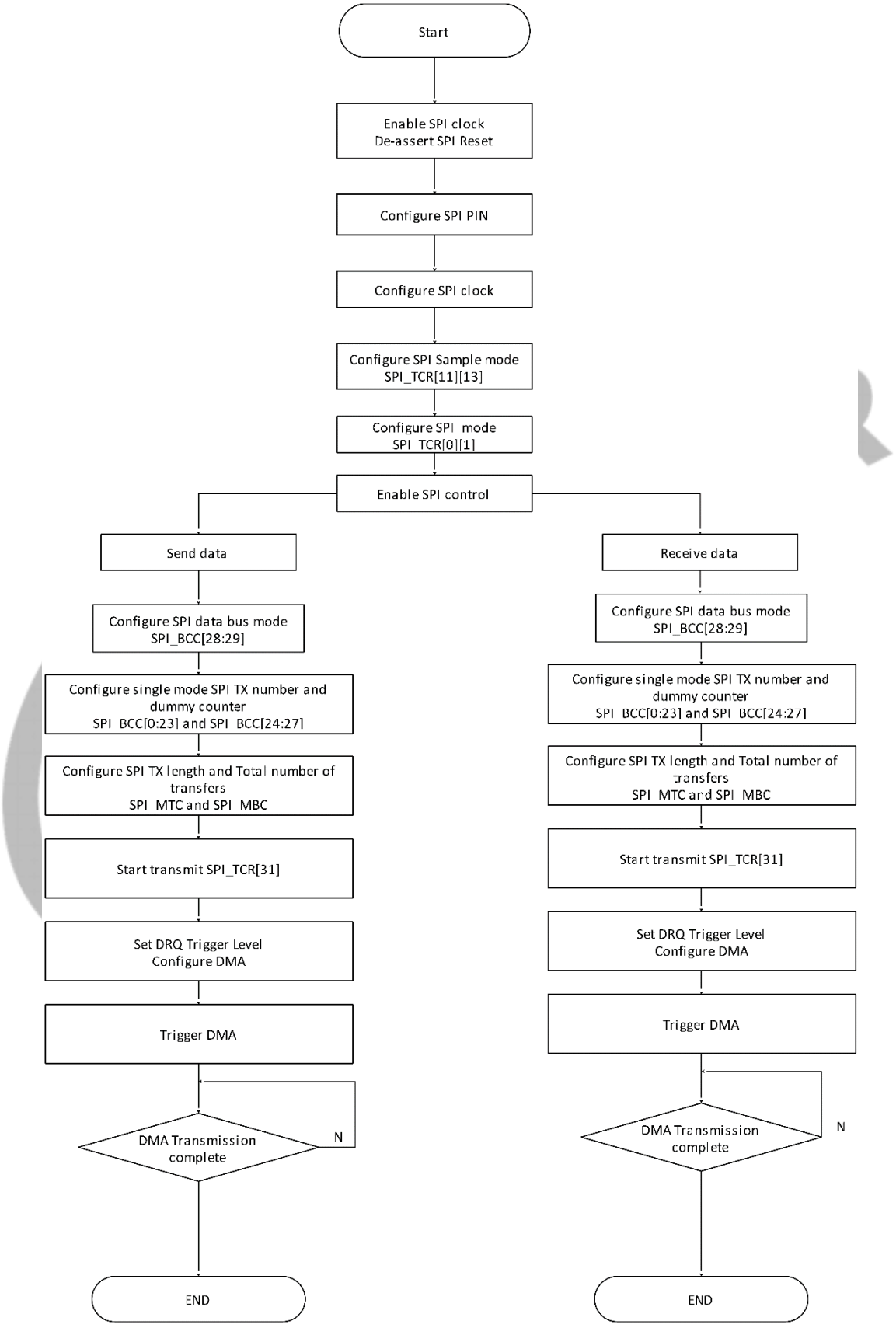
CPU Mode

Figure 9-61 SPI Write/Read Data in CPU Mode



DMA Mode

Figure 9-62 SPI Write/Read Data in DMA Mode



9.4.4.2 Calibrate Delay Chain Using SPI Mode

The SPI has one delay chain which is used to generate delay to make proper timing between the internal SPI clock signal and data signals. Delay chain is made up of 64 delay cells. The delay time of one delay cell can be estimated through delay chain calibration.

The steps to calibrate delay chain are as follows:

Step 1 Enable SPI. To calibrate the delay chain by operation registers in SPI, the SPI must be enabled through AHB reset and AHB clock gating control registers.

Step 2 Configure a proper clock for SPI. The calibration delay chain is based on the clock for SPI from CCU.

Step 3 Set proper initial delay value. Write 0xA0 to the SPI Sample Delay Control Register to set initial delay value 0x20 to delay chain. Then write 0x0 to the SPI Sample Delay Control Register to clear this value.

Step 4 Write 0x8000 to the SPI Sample Delay Control Register to start to calibrate the delay chain.

Step 5 Wait until the flag (Bit14 in the SPI Sample Delay Control Register) of calibration done is set. The number of delay cells is shown at Bit[13:8] in SPI Sample Delay Control Register. The delay time generated by these delay cells is equal to the cycle of SPI's clock nearly. This value is the result of calibration.

Step 6 Calculate the delay time of one delay cell according to the cycle of the SPI clock and the result of calibration.

9.4.4.3 Transmitting Write Command Using DBI Mode

Step 1 Set the SPI_DBI_MODE_SEL (bit3) of **SPI_GCR** (0x0004) to 1 to select DBI mode.

Step 2 Set the DBI EN MODE SEL (bit[30:29]) of **DBI_CTL_1** (0x0104) to 0 to select the trigger mode of DBI.

Step 3 Configure the **DBI_CTL_0** (0x0100).

- Set **DBI_CTL_0**[Command Type] (bit31) to 0 to configure the writing command.
- Set **DBI_CTL_0**[Write Command Dummy Cycles] (bit[30:20]) to configure the number of dummy cycles between commands.
- Set **DBI_CTL_0**[Output Data Sequence] (bit19) to select the MSB or LSB.
- Set **DBI_CTL_0**[Transmit Mode] (bit15) to 0 to select the command path.
- Set **DBI_CTL_0**[Output Data Format] (bit[14:12]) to 0 to transmit the command.
- Set **DBI_CTL_0**[DBI interface Select] (bit[10:8]) to select the DBI interface type.
- The remaining values of the **DBI_CTL_0** register remain the default value.

Step 4 Set **DBI_CTL_1**[DCX_DATA] (bit22) to 0 to send the command.

Step 5 DMA Path: Configure the **SPI_FCR** register (0x0018).

- Set **SPI_FCR**[TF_DRQ_EN] (bit24) to 1 to enable TXFIFO DMA.
- Set **SPI_FCR**[TX_TRIG_LEVEL] (bit[23:16]) to 255. It indicates the controller requests data from DMA if the remaining space of TX FIFO is greater than 255.

CPU Path: Write the command to be sent to the 0x200 address.

Step 6 Set **SPI_GCR**[DBI_EN] (bit4) to 1 to start transmitting the command.

Step 7 Wait until the TX FIFO underrun interrupt (**SPI_ISR**[TF_UDF]) is 1. It indicates that the command written to the TX FIFO is transmitted completely.

9.4.4.4 Transmitting Parameter Using DBI Mode

Step 1 Set the SPI_DBI_MODE_SEL (bit3) of **SPI_GCR** (0x0004) to 1 to select DBI mode.

Step 2 Set the DBI EN MODE SEL (bit[30:29]) of **DBI_CTL_1** (0x0104) to 0 to select the trigger mode of DBI.

Step 3 Configure the **DBI_CTL_0** register (0x0100).

- Set **DBI_CTL_0**[Command Type] (bit31) to 0 to configure the writing command.
- Set **DBI_CTL_0**[Write Command Dummy Cycles] (bit[30:20]) to configure the number of dummy cycles between commands.
- Set **DBI_CTL_0**[Output Data Sequence] (bit19) to select the MSB or LSB.
- Set **DBI_CTL_0**[Transmit Mode] (bit15) to 0 to select the command path.
- Set **DBI_CTL_0**[Output Data Format] (bit[14:12]) to 0 to transmit the command.
- Set **DBI_CTL_0**[DBI interface Select] (bit[10:8]) to select the DBI interface type.
- The remaining values of the **DBI_CTL_0** register remain the default value.

Step 4 Set **DBI_CTL_1**[DCX_DATA] (bit22) to 1 to send the parameter.

Step 5 DMA Path: Configure the C:\Users\zengjing\Downloads_Hlk49435465 - _Hlk49435792 register (0x0018).

- Set **SPI_FCR**[TF_DRQ_EN] (bit24) to 1 to enable TXFIFO DMA.
- Set **SPI_FCR**[TX_TRIG_LEVEL] (bit[23:16]) to 255. It indicates the controller requests data from DMA if the remaining space of TX FIFO is greater than 255.

CPU Path: Write the command to be sent to the 0x0200 address.

Step 6 Set **SPI_GCR**[DBI_EN] (bit4) to 1 to start transmitting the command.

Step 7 Wait until the TX FIFO underrun interrupt (**SPI_ISR**[TF_UDF]) is 1. It indicates that the command written to the TX FIFO is transmitted completely.

9.4.4.5 Transmitting Video Using DBI Mode

Set the SPI_DBI_MODE_SEL (bit3) of **SPI_GCR** (0x0004) to 1 to select DBI mode.

If the data is from the CPU path, the controller writes the command to be sent to the 0x0200 address by the AHB bus.

If the data is from the DMA path, configure **DBI_CTL_1**[DBI_FIFO_DRQ_EN] (bit15) to 1 and **DBI_CTL_1**[TX_TRIG_LEVEL] (bit[14:8]) to 64, which indicates the controller requests data from DMA if the remaining space of TX FIFO is greater than 64.

Software Trigger Mode

The software enables DBI_en_trigger when the edge interrupt of TE is detected.

After transmitting each frame image, the controller clears automatically the line_cnt, pixel_cnt and stops transmitting data.

Wait for the edge interrupt of TE, the software needs to enable DBI_en_trigger, in circulation.

The operation process is as follows.

Step 1 Set the SPI_DBI_MODE_SEL (bit3) of **SPI_GCR** (0x0004) to 1 to select DBI mode.

Step 2 Set the DBI EN MODE SEL (bit[30:29]) of **DBI_CTL_1** (0x0104) to 1 to select the software trigger mode.

Step 3 Configure the **DBI_CTL_0** register (0x0100).

- Set **DBI_CTL_0**[Command Type] (bit31) to 0 to set the writing command.
- Set **DBI_CTL_0**[Write Command Dummy Cycles] (bit[30:20]) to configure the number of dummy cycles between commands.
- Set **DBI_CTL_0**[Output Data Sequence] (bit19) to select the MSB or LSB.
- Set **DBI_CTL_0**[Transmit Mode] (bit15) to 1 to select the image path.
- Set **DBI_CTL_0**[Output Data Format] (bit[14:12]) to select RGB111//444/565/666/888.
- Set **DBI_CTL_0**[DBI interface Select] (bit[10:8]) to select the DBI interface type.
- The remaining values of the **DBI_CTL_0** register remain the default value.

Step 4 Set **DBI_CTL_1**[DCX_DATA] (bit22) to 0 to send the image data.

Step 5 Configure **DBI_Video_Size** (0x110) according to the sent image size.

Step 6 Configure **DBI_CTL_2** (0x0108) to set the TE-related parameter.

Step 7 Detect the TE interrupt of the **DBI_INT** (0x0120) register.

Step 8 Configure **DBI_CTL_1**[DBI_soft_trigger] to 1.

Timer Trigger Mode

The software configures timer_en to enable timer counting, and when the counter reaches the specified value, the DBI_EN automatically can be enabled to start transmitting data.

After transmitting each frame image, the controller clears automatically the line_cnt, pixel_cnt, and stops transmitting data.

The timer starts counting again. When the counter reaches the specified value, the controller automatically enables DBI_EN, and in circulation until the software turns off the timer_en.

The operation process is as follows.

Step 1 Set the SPI_DBI_MODE_SEL (bit3) of **SPI_GCR** (0x0004) to 1 to select DBI mode.

Step 2 Set the DBI EN MODE SEL (bit30:29) of **DBI_CTL_1** (0x0104) to 2 to select the timer trigger mode.

Step 3 Configure the **DBI_CTL_0** register (0x0100).

- Set **DBI_CTL_0**[Command Type] (bit31) to 0 to set the writing command.
- Set **DBI_CTL_0**[Write Command Dummy Cycles] (bit[30:20]) to configure the number of dummy cycles between commands.
- Set **DBI_CTL_0**[Output Data Sequence] (bit19) to select the MSB or LSB.
- Set **DBI_CTL_0**[Transmit Mode] (bit15) to 1 to select the image path.
- Set **DBI_CTL_0**[Output Data Format] (bit[14:12]) to select RGB111/444/565/666/888.
- Set **DBI_CTL_0**[DBI interface Select] (bit[10:8]) to select the DBI interface type.
- The remaining values of the **DBI_CTL_0** register remain the default value.

Step 4 Set **DBI_CTL_1**[DCX_DATA] (bit22) to 0 to send the image data.

Step 5 Configure **DBI_Video_Size** (0x110) to transmit the image size.

Step 6 Configure the related parameter of **DBI_Timer** (0x10C).

TE Trigger Mode

When the edge changes of the TE are detected (The rising and falling edges are optional), the DBI_EN automatically can be enabled to start transmitting data.

After transmitting each frame image, the controller clears automatically the line_cnt, pixel_cnt, and stops transmitting data.

When the edge changes of the TE are detected (The rising and falling edges are optional), the DBI_EN automatically can be enabled to start transmitting data until the software shuts down TE_EN or the screen no longer sends TE signals.

The operation process is as follows.

Step 1 Set the SPI_DBI_MODE_SEL (bit3) of **SPI_GCR** (0x0004) to 1 to select DBI mode.

Step 2 Set the DBI EN MODE SEL (bit30:29) of **DBI_CTL_1** (0x0104) to 3 to select the TE trigger mode.

Step 3 Configure the **DBI_CTL_0** register (0x0100).

- Set **DBI_CTL_0**[Command Type] (bit31) to 0 to set the writing command.
- Set **DBI_CTL_0**[Write Command Dummy Cycles] (bit[30:20]) to configure the number of dummy cycles between commands.
- Set **DBI_CTL_0**[Output Data Sequence] (bit19) to select the MSB or LSB.
- Set **DBI_CTL_0**[Transmit Mode] (bit15) to 1 to select the image path.
- Set **DBI_CTL_0**[Output Data Format] (bit[14:12]) to select RGB111/444/565/666/888.
- Set **DBI_CTL_0**[DBI interface Select] (bit[10:8]) to select the DBI interface type.
- The remaining values of the **DBI_CTL_0** register remain the default value.

Step 4 Configure **DBI_CTL_1**[DCX_DATA] (bit22) to 0 to send the image data.

Step 5 Configure **DBI_Video_Size** (0x0110) to transmit the image size.

Step 6 Configure **DBI_CTL_2** (0x0108) to set the TE-related parameter.

9.4.4.6 Transmitting Read Command and Read Data Using DBI Mode

Step 1 Set the SPI_DBI_MODE_SEL (bit3) of **SPI_GCR** (0x0004) to 1 to select DBI mode.

Step 2 Set the DBI EN MODE SEL (bit[30:29]) of **DBI_CTL_1** (0x0104) to 0.

Step 3 Configure the **DBI_CTL_0** register (0x0100).

- Set **DBI_CTL_0**[Command Type] (bit31) to 0 to set the reading command.
- Set **DBI_CTL_0**[Output Data Sequence] (bit19) to select the MSB or LSB.
- Set **DBI_CTL_0**[Transmit Mode] (bit15) to 0 to select the command path.
- Set **DBI_CTL_0**[Output Data Format] (bit[14:12]) to 0.
- Set **DBI_CTL_0**[DBI interface Select] (bit[10:8]) to select the DBI interface type.

- The remaining values of the DBI_CTL_0 register remain the default value.

Step 4 Configure the DBI_CTL_1 register (0x0104).

- Configure DBI_CTL_1[DCX_DATA] (bit22) to 0 to send the command.
- Configure DBI_CTL_1[Read_MSB_First] (bit20) to select whether the first bit of the read data is the highest or lowest bit of data.
- Configure DBI_CTL_1[Read Data Number of Bytes] to set the byte number to be read.
- Configure DBI_CTL_1[Read Command Dummy Cycles] to set the dummy cycle between the read command and the read data, when the dummy cycle is complete, the data starts to be sampled.

Step 5 DMA Path: Configure the SPI_FCR register (0x0018).

- Set SPI_FCR[RF_DRQ_EN] (bit8) to 1 to enable RXFIFO DMA.
- Set SPI_FCR[RX_TRIG_LEVEL] (bit[7:0]) to 32, which indicates the controller requests receiving data from DMA if the data of the RX FIFO is greater than 64.

CPU Path: Read data in RX FIFO from the 0x0300 address.

Step 6 Set SPI_GCR[DBI_EN] (bit4) to 1 to start transmitting command.

Step 7 Wait until DBI_INT[RD_DONE_INT] is 1. It indicates that the data is read completely.

9.4.5 Register List

Module Name	Base Address
SPI_DBI	0x04026000

Register Name	Offset	Description
SPI_GCR	0x0004	SPI Global Control Register
SPI_TCR	0x0008	SPI Transfer Control Register
SPI_IER	0x0010	SPI Interrupt Control Register
SPI_ISR	0x0014	SPI Interrupt Status Register
SPI_FCR	0x0018	SPI FIFO Control Register
SPI_FSR	0x001C	SPI FIFO Status Register
SPI_WCR	0x0020	SPI Wait Clock Register
SPI_SAMP_DL	0x0028	SPI Sample Delay Control Register
SPI_MBC	0x0030	SPI Master Burst Counter Register
SPI_MTC	0x0034	SPI Master Transmit Counter Register

Register Name	Offset	Description
SPI_BCC	0x0038	SPI Master Burst Control Register
SPI_BATCR	0x0040	SPI Bit-Aligned Transfer Configure Register
SPI_BA_CCR	0x0044	SPI Bit-Aligned Clock Configuration Register
SPI_TBR	0x0048	SPI TX Bit Register
SPI_RBR	0x004C	SPI RX Bit Register
SPI_NDMA_MODE_CTL	0x0088	SPI Normal DMA Mode Control Register
DBI_CTL_0	0x0100	DBI Control Register 0
DBI_CTL_1	0x0104	DBI Control Register 1
DBI_CTL_2	0x0108	DBI Control Register 2
DBI_TIMER	0x010C	DBI Timer Control Register
DBI_VIDEO_SZIE	0x0110	DBI Video Size Configuration Register
DBI_INT	0x0120	DBI Interrupt Register
DBI_DEBUG_0	0x0124	DBI BEBUG 0 Register
DBI_DEBUG_1	0x0128	DBI BEBUG 1 Register
SPI_TXD	0x0200	SPI TX Data register
SPI_RXD	0x0300	SPI RX Data register

9.4.6 Register Description

9.4.6.1 0x0004 SPI Global Control Register (Default Value: 0x0000_0080)

Offset:0x0004			Register Name: SPI_GCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/WAC	0x0	SRST Soft reset Writing ‘1’ to this bit will clear the SPI controller, and auto clear to ‘0’ when reset operation completes. Writing ‘0’ to this bit has no effect.
30:8	/	/	/

Offset:0x0004			Register Name: SPI_GCR
Bit	**Read/Write**	**Default/Hex**	**Description**
7	R/W	0x1	TP_EN Transmit Pause Enable In master mode, it is used to control transmit state machine to stop smart burst sending when RX FIFO is full. 0: Normal operation, ignore RXFIFO status 1: Stop transmit data when RXFIFO full Cannot be written when XCH=1.
6:5	/	/	/
4	R/W	0x0	DBI EN DBI Module Enable Control 0: Disable 1: Enable
3	R/W	0x0	SPI_DBI_MODE_SEL DBI Working Mode Select 0: SPI MODE 1: DBI MODE
2	R/W	0x0	MODE_SELEC Sample Timing Mode Select 0: Old mode of Sample Timing 1: New mode of Sample Timing Cannot be written when XCH=1.
1	R/W	0x0	MODE SPI Function Mode Select 0: Slave mode 1: Master mode Cannot be written when XCH=1.
0	R/W	0x0	EN SPI Module Enable Control 0: Disable 1: Enable After transforming from bit_mode to byte_mode, it must enable the SPI module again.

9.4.6.2 0x0008 SPI Transfer Control Register (Default Value: 0x0000_0087)

Offset: 0x0008			Register Name: SPI_TCR
Bit	**Read/Write**	**Default/Hex**	**Description**
31	R/WAC	0x0	XCH Exchange Burst In master mode, it is used to start SPI burst 0: Idle 1: Initiates exchange. Writing “1” to this bit will start the SPI burst, and will auto-clear after finishing the bursts transfer specified by BC. Writing “1” to SRST will also clear this bit. Writing ‘0’ to this bit has no effect. Cannot be written when XCH=1.
30:16	/	/	/
15	R/W	0x0	SDC1 Master Sample Data Control register1 Set this bit to ‘1’ to make the internal read sample point with a delay of half-cycle of SPI_CLK. It is used in high speed read operation to reduce the error caused by the time delay of SPI_CLK between master and slave. 0: Normal operation, do not delay the internal read sample point 1: Delay the internal read sample point Cannot be written when XCH=1.
14	R/W	0x0	SDDM Sending Data Delay Mode 0: Normal sending 1: Delay sending Set the bit to "1" to make the data that should be sent with a delay of half-cycle of SPI_CLK in dual IO mode for SPI mode 0. Cannot be written when XCH=1.
13	R/W	0x0	SDM Master Sample Data Mode 0: Delay sample mode 1: Normal sample mode In normal sample mode, the SPI master samples the data at the correct edge for each SPI mode; In delay sample mode, the SPI master samples data at the edge that is half cycle delayed by the correct edge defined in respective SPI mode. Cannot be written when XCH=1.

Offset: 0x0008			Register Name: SPI_TCR
Bit	**Read/Write**	**Default/Hex**	**Description**
12	R/W	0x0	FBS First Transmit Bit Select 0: MSB first. The upper bits are transmitted first. 1: LSB first. The lower bits are transmitted first. Cannot be written when XCH=1.
11	R/W	0x0	SDC Master Sample Data Control Set this bit to '1' to make the internal read sample point with a delay of half-cycle of SPI_CLK. It is used in high speed read operation to reduce the error caused by the time delay of SPI_CLK between master and slave. 0: Normal operation, do not delay the internal read sample point 1: Delay the internal read sample point Cannot be written when XCH=1.
10	R/W	0x0	RPSM Rapids Mode Select Select rapid mode for high speed write. 0: Normal write mode 1: Rapid write mode Cannot be written when XCH=1.
9	R/W	0x0	DDB Dummy Burst Type 0: The bit value of dummy SPI burst is zero 1: The bit value of dummy SPI burst is one Cannot be written when XCH=1.
8	R/W	0x0	DHB Discard Hash Burst In master mode, it controls whether discarding unused SPI bursts 0: Receiving all SPI bursts in the BC period 1: Discard unused SPI bursts, only fetching the SPI bursts during the dummy burst period. The burst number is specified by TC. Cannot be written when XCH=1.