1 AC97/I2S Controller

1.1 Overview

This chapter describes the AIC (AC'97 and I²S Controller) included in the Jz4740 processor.

The AIC supports the Audio Codec '97 Component Specification 2.3 for AC-link format and I2S or IIS (for inter-IC sound), a protocol defined by Philips Semiconductor. Both normal I2S and the MSB-justified I2S formats are supported by AIC.

AIC consists of buffers, status registers, control registers, serializers, and counters for transferring digitized audio between the Jz4740 processor system memory and an internal I2S CODEC, an external AC97 or I2S CODEC. AIC can record digitized audio by storing the samples in system memory. For playback of digitized audio or production of synthesized audio, the AIC retrieves digitized audio samples from system memory and sends them to a CODEC through the serial connection with AC-link or I2S formats. The internal or external digital-to-analog converter in the CODEC then converts the audio samples into an analog audio waveform. The audio sample data can be stored to and retrieved from system memory either by the DMA controller or by programmed I/O.

The AC-link is a synchronous, fixed-rate serial bus interface for transferring CODEC register control and status information in addition to digital audio. Where both normal I2S and MSB-justified-I2S work with a variety of clock rates, which can be obtained either by dividing the PLL clock by two programmable dividers or from an external clock source.

For I2S systems that support the L3 control bus protocol, additional pins are required to control the external CODEC. CODECs that use an L3 control bus require 3 signals: L3_CLK, L3_DATA, and L3_MODE for writing bytes into the L3 bus register. The AIC supports the L3 bus protocol via software control of the general-purpose I/O (GPIO) pins. The AIC does not provide hardware control for the L3 bus protocol.

To control the internal CODEC, internal CODEC spec can be referenced.

This chapter describes the programming model for the AIC. The information in this chapter requires an understanding of the AC'97 specification, Revision 2.3.
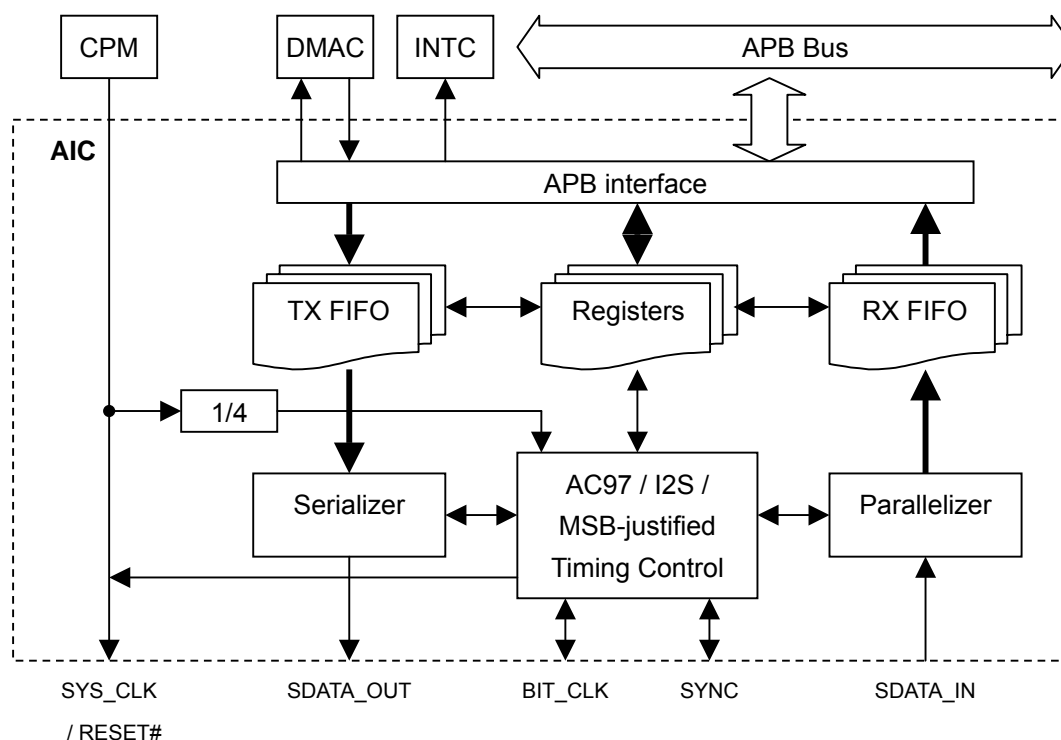
1.1.1 Block Diagram

Figure 1-1 AIC Block Diagram

1.1.2 Features

AIC support following AC97/I2S features:

● 8, 16, 18, 20 and 24 bit audio sample data sizes supported
● DMA transfer mode supported
● Stop serial clock supported
● Programmable Interrupt function supported
● Support mono PCM data to stereo PCM data expansion on audio play back
● Support endian switch on 16-bits audio samples play back
● Support variable sample rate in AC-link format
● Multiple channel output and double rated supported for AC-link format
● Power Down Mode and two Wake-Up modes Supported for AC-link format
● Internal programmable or external serial clock and optional system clock supported for I2S or MSB-Justified format
● Internal I2S CODEC supported
● Two FIFOs for transmit and receive respectively with 32 samples capacity in every direction.
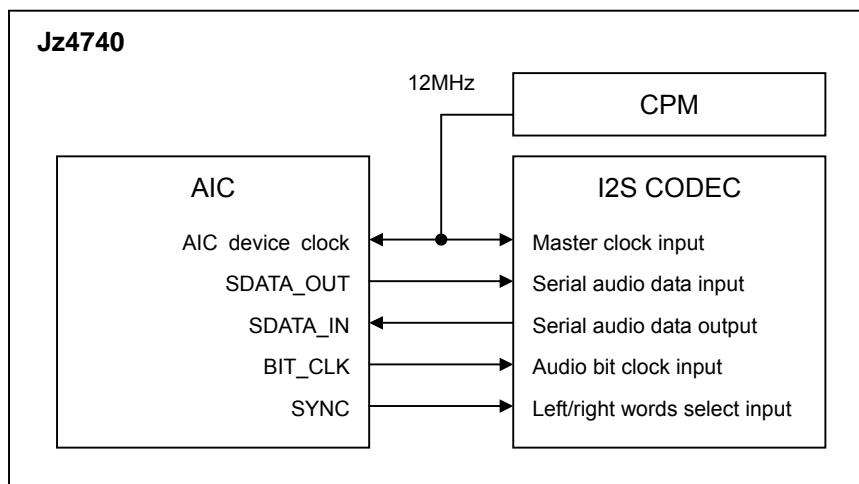
1.1.3   Interface Diagram

Figure 1-2 Jz4740 AIC Interface to the Internal I2S CODEC Diagram
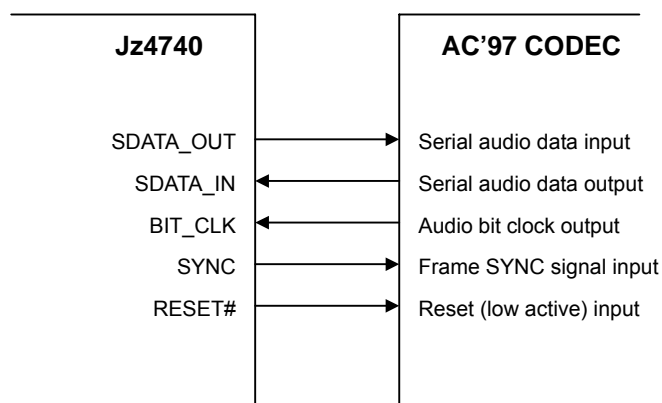

Figure 1-3 Jz4740 Interface to an External AC'97 CODEC Diagram
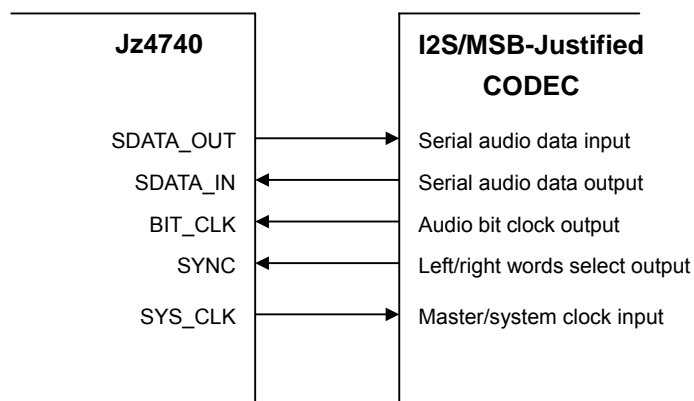

Figure 1-4 Jz4740 Interface to an External Master Mode I2S/MSB-Justified CODEC Diagram
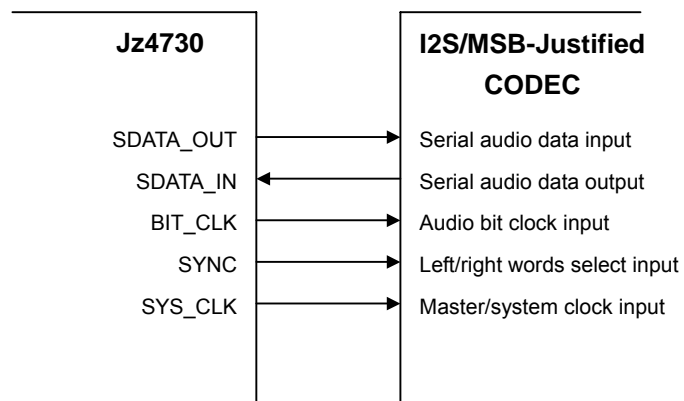
Figure 1-5 Jz4740 Interface to an External Slave Mode I2S/MSB-Justified CODEC Diagram

1.1.4 Signal Descriptions

There are all 6 pins used to connect between AIC and an external audio CODEC device. If an internal CODEC is used, these pins are not needed. They are listed and described in Table 1-1.

Table 1-1 AIC Pins Description

Name	I/O	Description
RESET# SYS_CLK	O	RESET#: AC-link format, active-low CODEC reset. SYS_CLK: I2S/MSB-Justified formats, supply system clock to CODEC
BIT_CLK	I I/O	12.288 MHz bit-rate clock input for AC-link, and sample rate dependent bit-rate clock input/output for I2S/MSB-Jistified.
SYNC	O	48-kHz frame indicator and synchronizer for AC-link format
	I/O	Indicates the left- or right-channel for I2S/MSB-Justified format
SDATA_OUT	O	Serial audio output data to CODEC
SDATA_IN	I	Serial audio input data from CODEC

1.1.5 RESET# / SYS_CLK Pin

RESET# is AC97 active-low CODEC reset, which outputs to CODEC. The CODEC's registers are reset when this RESET# is asserted. This pin is useful only in AC-link format. If AIC is disabled, it retains the high.

SYS_CLK outputs the system clock to CODEC. This pin is useful only in I2S/MSB-justified format. It generates a frequency between approximately 2.048 MHz and 24.576 MHz by dividing down the PLL clock with a programmable divisor. This frequency can be 256, 384, 512 and etc. times of the audio sampling frequency. Or it can be set to a wanted frequency. If AIC is disabled, it retains the high.

1.1.6 BIT_CLK Pin

BIT_CLK is the serial data bit rate clock, at which AC97/I2S data moves between the CODEC and the processor. One bit of the serial data is transmitted or received each BIT_CLK period. It is fixed to 12.288 MHz in AC-link format, which inputs from the CODEC. In I2S and MSB-justified format it inputs from the CODEC in slave mode and outputs to CODEC in master mode. In the master mode, the clock is generated internally that is 64 times the sampling frequency. Table 1-7 lists the available sampling frequencies based on an internal clock source. If AIC is disabled, AICFR.AUSEL and AICFR.BCKD determine the direction. And it retains the low if it is output and the state is undefined if it is input.

1.1.7 SYNC Pin

In AC-link format, SYNC provides frame synchronization, fixed to 48kHz, by specifying beginning of an audio sample frame and outputs to CODEC. In I2S/MSB-Justified formats, SYNC is used to indicate left- or right-channel sample data and toggled in sample rate frequency. It outputs to CODEC in master mode and inputs from CODEC in slave mode. If AIC is disabled, AICFR.AUSEL and AICFR.BCKD determine the direction. And it retains the low if it is output and the state is undefined if it is input.

1.1.8 SDATA_OUT Pin

SDATA_OUT is AIC output data pin, which outputs serial audio data or data of AC97 CODEC register control to an external audio CODEC device. If AIC is disabled, it retains the low.

1.1.9 SDATA_IN Pin

SDATA_IN is AIC inputs data pin, which inputs serial audio data or data of AC97 CODEC register status from an external audio CODEC device. If AIC is disabled, its state is undefined.

1.2 Register Descriptions

AIC software interface includes 13 registers and 1 FIFO data port. They are mapped in IO memory address space so that program can access them to control the operation of AIC and the outside CODEC.

Table 1-2 AIC Registers Description

Name	Description	RW	Reset value	Address	Size
AICFR	AIC Configuration Register	RW	0x00007800	0x10020000	32
AICCR	AIC Common Control Register	RW	0x00000000	0x10020004	32
ACCR1	AIC AC-link Control Register 1	RW	0x00000000	0x10020008	32
ACCR2	AIC AC-link Control Register 2	RW	0x00000000	0x1002000C	32
I2SCR	AIC I2S/MSB-justified Control Register	RW	0x00000000	0x10020010	32
AICSR	AIC FIFO Status Register	RW	0x00000008	0x10020014	32
ACSR	AIC AC-link Status Register	RW	0x00000000	0x10020018	32
I2SSR	AIC I2S/MSB-justified Status Register	RW	0x00000000	0x1002001C	32
ACCAR	AIC AC97 CODEC Command Address Register	RW	0x00000000	0x10020020	32
ACCDR	AIC AC97 CODEC Command Data Register	RW	0x00000000	0x10020024	32
ACSAR	AIC AC97 CODEC Status Address Register	R	0x00000000	0x10020028	32
ACSDR	AIC AC97 CODEC Status Data Register	R	0x00000000	0x1002002C	32
I2SDIV	AIC I2S/MSB-justified Clock Divider Register	RW	0x00000003	0x10020030	32
AICDR	AIC FIFO Data Port Register	RW	0x????????	0x10020034	32

- AICFR is used to control FIFO threshold, AC-link or I2S/MSB-justified selection, AIC reset, master/slave selection, and AIC enable.
- AICCR is used to control DMA mode, FIFO flush, interrupt enable, internal loop-back, play back and recording enable. It also controls sample size and signed/unsigned data transfer.
- ACCR1 is used to reflect/control valid incoming/outgoing slots of AC97.
- ACCR2 is used to control interrupt enable, output/input sample size, and alternative control of RESET#, SYNC and SDATA_OUT pins in AC-link.
- I2SCR is used to control BIT_CLK stop, audio sample size, I2S or MSB-justified selection in I2S/MSB-justified.
- AICSR is used to reflect FIFOs status
- ACSR is used to reflect the status of the connected external CODEC in AC-link.

- I2SSR is used to reflect AIC status in I2S/MSB-justified.

- ACCAR and ACCDR are used to hold address and data for AC-link CODEC register read/write.

- ACSAR and ACSDR are used to receive AC-link CODEC registers address and data

- I2SDIV is used to set clock divider for BIT_CLK generating in I2S/MSB-justified format.

- AICDR is act as data input/output port to/from transmit/receive FIFO when write/read.

1.2.1 AIC Configuration Register (AICFR)

AICFR contains bits to control FIFO threshold, AC-link or I2S/MSB-justified selection, AIC reset, master/slave selection, and AIC enable.

AICFR   **0x10020000**

Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12	11 10 9 8	7	6	5	4	3	2	1	0
	Reserved	RFTH	TFTH	Reserved	LSMP	ICDC	AUSEL	RST	BCKD	SYNCD	ENB
RST	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 1 1 1	1 0 0 0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:16	Reserved	Writes to these bits have no effect and always read as 0	R
15:12	RFTH	Receive FIFO threshold for interrupt or DMA request. The RFTH valid value is 0 ~ 15. This value represents a threshold value of (RFTH + 1) * 2. When the sample number in receive FIFO, indicated by AICSR.RFL, is great than or equal to the threshold value, AICSR.RFS is set. Larger RFTH value provides lower DMA/interrupt request frequency but have more risk to involve receive FIFO overflow. The optimum value is system dependent.	RW
11:8	TFTH	Transmit FIFO threshold for interrupt or DMA request. The TFTH valie value 0 ~ 15. This value represents a threshold value of TFTH * 2. When the sample number in transmit FIFO, indicated by AICSR.TFL, is less than or equal to the threshold value, AICSR.TFS is set. Smaller TFTH value provides lower DMA/interrupt request frequency but have more risk to involve transmit FIFO underflow. The optimum value is system dependent.	RW
7	Reserved	Writes to these bits have no effect and always read as 0	R
6	LSMP	Select between play last sample or play ZERO sample in TX FIFO underflow. ZERO sample means sample value is zero. This bit is better be changed while audio replay is stopped.<table><tr><th>LSMP</th><th>CODEC used</th></tr><tr><td>0</td><td>Play ZERO sample when TX FIFO underflow</td></tr><tr><td>1</td><td>Play last sample when TX FIFO underflow</td></tr></table>	RW
5	ICDC	Internal CODEC used. Select between internal or external CODEC.<table><tr><th>ICDC</th><th>CODEC used</th></tr><tr><td>0</td><td>External CODEC</td></tr><tr><td>1</td><td>Internal CODEC</td></tr></table>	RW
4	AUSEL	Audio Unit Select. Select between AC-link and I2S/MSB-justified. Change this bit in case of BIT_CLK is stopped (I2SCR.STPBK = 1)	RW

		AUSEL	Selected		
		0	Select AC-link format		
		1	Select I2S/MSB-justified format		
3	RST	Reset AIC. Write 1 to this bit reset AIC registers and FIFOs except AICFR and I2SDIV register. Writing 0 to this bit has no effect and this bit is always reading 0.			W
2	BCKD	BIT_CLK Direction. This bit specifies input/output direction of BIT_CLK. It is only valid in I2S/MSB-justified format. When AC-link format is selected, BIT_CLK is always input and this bit is ignored. Change this bit in case of BIT_CLK is stopped (I2SCR.STPBK = 1)			RW
		BCKD	BIT_CLK Direction		
		0	BIT_CLK is input from an external source.		
		1	BIT_CLK is generated internally and driven out to the CODEC.		
1	SYNCD	SYNC Direction. This bit specifies input/output direction of SYNC in I2S/MSB-justified format. When AC-link format is selected, SYNC is always output and this bit is ignored.			RW
		SYNCD	SYNC Direction		
		0	SYNC is input from an external source.		
		1	SYNC is generated internally and driven out to the CODEC.		
0	ENB	Enable AIC function. This bit is used to enable or disable the AIC function.			RW
		ENB	Description		
		0	Disable AIC Controller		
		1	Enable AIC Controller		

The BCKD bit (bit 2) and SYNCD bit (bit 1) configure the mode of I2S/MSB-justified interface. This is compliant with I2S specification.

BCKD	SYNCD	Description
0 (input)	0 (input)	AIC roles the slave of I2S/MSB-justified interface.
	1 (output)	AIC roles the master with external serial clock source of I2S/MSB-justified interface.
1 (output)	0 (input)	Reserved
	1 (output)	AIC roles the master of I2S/MSB-justified interface

Jz4740 peripheral specification, Revision 1.0

1.2.2 AIC Common Control Register (AICCR)

AICCR contains bits to control DMA mode, FIFO flush, interrupt enable, internal loop-back, play back and recording enable. It also controls sample size and signed/unsigned data transfer.

AICCR　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　　**0x10020004**

Bit	31 30 29 28 27 26 25 24 23 22	21 20 19	18 17 16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved	OSS	ISS	RDMS	TDMS	Reserved	Reserved	M2S	ENDSW	ASVTSU	FLUSH	Reserved	EROR	ETUR	ERFS	ETFS	ENLBF	ERPL	EREC
RST	0 0 0 0 0 0 0 0 0 0	1 0 0	1 0 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:22	Reserved	Writes to these bits have no effect and always read as 0	R
21:19	OSS	Output Sample Size. These bits reflect output sample data size from memory or register. The data sizes supported are: 8, 16, 18, 20 and 24 bits. The sample data is LSB-justified in memory/register. OSS / Sample Size: 0x0 — 8 bit 0x1 — 16 bit 0x2 — 18 bit 0x3 — 20 bit 0x4 — 24 bit 0x5~0x7 — Reserved	RW
18:16	ISS	Input Sample Size. These bits reflect input sample data size to memory or register. The data sizes supported are: 8, 16, 18, 20 and 24 bits. The sample data is LSB-justified in memory/register. ISS / Sample Size: 0x0 — 8 bit 0x1 — 16 bit 0x2 — 18 bit 0x3 — 20 bit 0x4 — 24 bit 0x5~0x7 — Reserved	RW
15	RDMS	Receive DMA enable. This bit is used to enable or disable the DMA during receiving audio data. RDMS / Receive DMA: 0 — Disabled 1 — Enabled	RW
14	TDMS	Transmit DMA enable. This bit is used to enable or disable the DMA during transmit audio data.	RW

		TDMS	**Transmit DMA**		
		0	Disabled		
		1	Enabled		
13:12	Reserved	Writes to these bits have no effect and always read as 0			R
11	M2S	Mono To Stereo. This bit control whether to do mono to stereo sample expansion in play back. When this bit is set, every outgoing sample data in the steam plays in both left and right channels. This bit should only be set in 2 channels configuration. It takes effective immediately when the bit is changed.			RW
		M2S	**Description**		
		0	No mono to stereo expansion		
		1	Do mono to stereo expansion		
10	ENDSW	Endian Switch. This bit control endian change on outgoing 16-bits size audio sample by swapping high and low bytes in the sample data.			RW
		ENDSW	**Description**		
		0	No change on outgoing sample data		
		1	Swap high and low byte for outgoing 16-bits size sample data		
9	ASVTSU	Audio Sample Value Transfer between Signed and Unsigned data format. This bit is used to control the signed ←→unsigned data transfer. If it is 1, the incoming and outgoing audio sample data will be transferred by toggle its most significant bit.			RW
		ASVTSU	**Description**		
		0	No audio sample value signed ←→unsigned transfer		
		1	Do audio sample value signed ←→unsigned transfer		
8	FLUSH	FIFO Flush. Write 1 to this bit flush transmit/receive FIFOs to empty. Writing 0 to this bit has no effect and this bit is always reading 0.			W
7	Reserved	Writes to these bits have no effect and always read as 0			R
6	EROR	Enable ROR Interrupt. This bit is used to control the ROR interrupt enable or disable.			RW
		EROR	**ROR Interrupt**		
		0	Disabled		
		1	Enabled		
5	ETUR	Enable TUR Interrupt. This bit is used to control the TUR interrupt enable or disable.			RW
		ETUR	**TUR Interrupt**		
		0	Disabled		
		1	Enabled		
4	ERFS	Enable RFS Interrupt. This bit is used to control the RFS interrupt enable or disable.			RW
		ERFS	**RFS Interrupt**		
		0	Disabled		
		1	Enabled		

3	ETFS	Enable TFS Interrupt. This bit is used to control the TFS interrupt enable or disable.	RW

ETFS	TFS Interrupt
0	Disabled
1	Enabled

2	ENLBF	Enable AIC Loop Back Function. This bit is used to enable or disable the internal loop back function of AIC, which is used for test only. When the AIC loop back function is enabled, normal audio replay/record functions are disabled.	RW

ENLBF	Description
0	AIC Loop Back Function is Disabled
1	AIC Loop Back Function is Enabled

1	ERPL	Enable Playing Back function. This bit is used to disable or enable the audio sample data transmitting.	RW

ERPL	Description
0	AIC Playing Back Function is Disabled
1	AIC Playing Back Function is Enabled

0	EREC	Enable Recording Function. This bit is used to disable or enable the audio sample data receiving.	RW

EREC	Description
0	AIC Recording Function is Disabled
1	AIC Recording Function is Enabled

1.2.3 AIC AC-link Control Register 1 (ACCR1)

ACCR1 contains bits to reflect/control valid incoming/outgoing slots of AC97. It is used only in AC-link format.

ACCR1   **0x10020008**

Bit	31 30 29 28 27 26	25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10	9 8 7 6 5 4 3 2 1 0
	Reserved	RS	Reserved	XS

RST 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bits	Name	Description	RW
31:26	Reserved	Writes to these bits have no effect and always read as 0	R
25:16	RS	Receive Valid Slots. These read only bits are taken from the valid bits in the incoming AC'97 tag (slot 0 of SDATA_IN) and indicate which incoming slots have valid PCM data. Slot 3 is mapped to bit 16 or RS[0], slot 4 to bit 17 or RS[1] and so on. If the corresponding bit is set it indicates that valid PCM data is in the respective slot. <table><tr><th>RS[n] Value</th><th>Description</th></tr><tr><td>0</td><td>Slot n+3 is invalid.</td></tr><tr><td>1</td><td>Slot n+3 has valid PCM data.</td></tr></table>	R
15:10	Reserved	Writes to these bits have no effect and always read as 0	R
9:0	XS	Transmit Valid Slots. These bits making up slots map to the valid bits in the AC'97 tag (slot 0 on SDATA_OUT) and indicate which outgoing slots have valid PCM data. Bit 0 or XS[0] maps to slot 3, bit 1 or XS[1] to slot 4 and so on. Setting the corresponding bit indicates to AIC to take an audio sample from transmit FIFO to fill the respective slot. And it indicates to the CODEC that valid PCM data will be in the respective slot. The number of valid bits will designate how many words will be pulled out of the FIFO per audio frame. <table><tr><th>XS[n] Value</th><th>Description</th></tr><tr><td>0</td><td>Slot n+3 is invalid.</td></tr><tr><td>1</td><td>Slot n+3 has valid PCM data.</td></tr></table>	RW

1.2.4 AIC AC-link Control Register 2 (ACCR2)

ACCR2 contains bits to control interrupt enable, output/input sample size, and alternative control of RESET#, SYNC and SDATA_OUT pins in AC-link. It is valid only in AC-link format.

ACCR2   **0x1002000C**

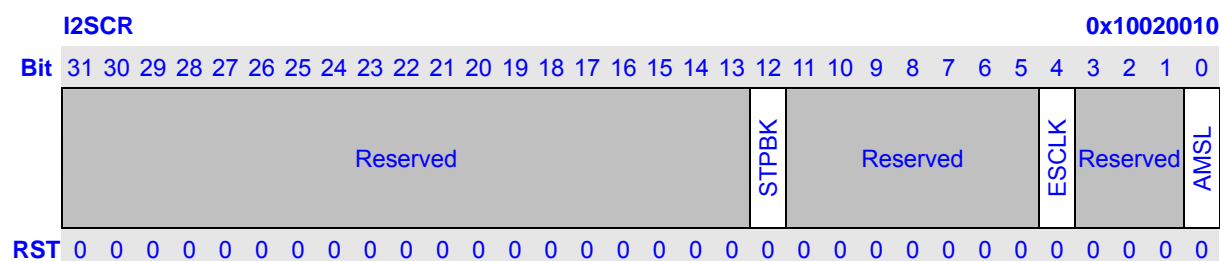| Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved													ERSTO	ESADR	ECADT	Reserved												SO	SR	SS	SA
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:19	Reserved	Writes to these bits have no effect and always read as 0	R
18	ERSTO	Enable RSTO Interrupt. This bit is used to control the RSTO interrupt enable or disable. **ERSTO** | **RSTO Interrupt** 0 | Disabled 1 | Enabled	RW
17	ESADR	Enable SADR Interrupt. This bit is used to control the SADR interrupt enable or disable. **ESADR** | **SADR Interrupt** 0 | Disabled 1 | Enabled	RW
16	ECADT	Enable CADT Interrupt. This bit is used to control the CADT interrupt enable or disable. **ECADT** | **CADT Interrupt** 0 | Disabled 1 | Enabled	RW
15:4	Reserved	Writes to these bits have no effect and always read as 0	R
3	SO	SDATA_OUT output value. When SA is 1, this bit controls SDATA_OUT pin voltage level, 0 for low, 1 for high; otherwise, it is ignored.	RW
2	SR	RESET# pin level. When AC-link is selected, this bit is used to drive the RESET# pin. **SR** | **RESET# Pin Voltage Level** 0 | High 1 | Low	RW
1	SS	SYNC value. When this bit is read, it returns the actual value of SYNC. When SA is 1, write value controls SYNC pin value. When SA is 0, write to it is ignored.	RW
0	SA	SYNC and SDATA_OUT Alternation. This bit is used to determine the	RW

driven signal of SYNC and SDATA_OUT. When SA is 0, SYNC and SDATA_OUT being driven AIC function logic; otherwise, SYNC is controlled by the SS and SDATA_OUT is controlled by the SO. The true table of SYNC is described in following.

SA	SS	Description
0	0	When read, indicated SYNC is 0
		When write, not effect
	1	When read, indicated SYNC is 1
		When write, not effect
1	0	When read, indicated SYNC is 0
		When write, SYNC is driven to 0
	1	When read, indicated SYNC is 1
		When write, SYNC is driven to 1

Jz4740 peripheral specification, Revision 1.0

1.2.5 AIC I2S/MSB-justified Control Register (I2SCR)

I2SCR contains bits to control BIT_CLK stop, audio sample size, I2S or MSB-justified selection in I2S/MSB-justified. It is valid only in I2S/MSB-justified format.

I2SCR   **0x10020010**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved																			STPBK	Reserved							ESCLK	Reserved			AMSL
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:13	Reserved	Writes to these bits have no effect and always read as 0	R
12	STPBK	Stop BIT_CLK. It is used to stop the BIT_CLK in I2S/MSB-justified format. When AC-link is selected, all of its operations are ignored. <table><tr><td>STPBK</td><td>Description</td></tr><tr><td>0</td><td>BIT_CLK is not stopped</td></tr><tr><td>1</td><td>BIT_CLK is stopped</td></tr></table> Please set this bit to 1 to stop BIT_CLK when change AICFR.AUSEL and AICFR.BCKD	RW
11:5	Reserved	Writes to these bits have no effect and always read as 0	R
4	ESCLK	Enable SYSCLK output. When this bit is 1, the SYSCLK outputs to chip outside is enabled. Else, the clock is disabled.	RW
0	AMSL	Specify Alternate Mode (I2S or MSB-Justified) Operation. <table><tr><td>AMSL</td><td>Description</td></tr><tr><td>0</td><td>Select I2S Operation Mode</td></tr><tr><td>1</td><td>Select MSB-Justified Operation Mode</td></tr></table>	RW

1.2.6   AIC Controller FIFO Status Register (AICSR)

AICSR contains bits to reflect FIFOs status. Most of the bits are read-only except two, which can be written a 0.

AICSR   **0x10020014**

Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Reserved	RFL	Reserved	TFL	Reserved	ROR	TUR	RFS	TFS	Reserved

RST	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bits	Name	Description	RW
31:30	Reserved	Writes to these bits have no effect and always read as 0	R
29:24	RFL	Receive FIFO Level. The bits indicate the amount of valid PCM data in Receive FIFO. <table><tr><th>RFL Value</th><th>Description</th></tr><tr><td>0x00 ~ 0x20</td><td>RFL valid PCM data in receive FIFO</td></tr><tr><td>0x21 ~ 0x3F</td><td>Reserved</td></tr></table>	R
23:14	Reserved	Writes to these bits have no effect and always read as 0	R
13:8	TFL	Transmit FIFO Level. The bits indicate the amount of valid PCM data in Transmit FIFO. <table><tr><th>TFL Value</th><th>Description</th></tr><tr><td>0x00 ~ 0x20</td><td>TFL valid PCM data in transmit FIFO</td></tr><tr><td>0x21 ~ 0x3F</td><td>Reserved</td></tr></table>	R

Bits	Name	Description	RW
7	Reserved	Writes to these bits have no effect and always read as 0	R
6	ROR	Receive FIFO Over Run. This bit indicates that receive FIFO has or has not experienced an overrun. <table><tr><th>ROR</th><th>Description</th></tr><tr><td rowspan="2">0</td><td>When read, indicates over-run has not been found</td></tr><tr><td>When write, clear itself</td></tr><tr><td rowspan="2">1</td><td>When read, indicates data has even been written to full receive FIFO</td></tr><tr><td>When write, not effects</td></tr></table>	RW
5	TUR	Transmit FIFO Under Run. This bit indicates that transmit FIFO has or has not experienced an under-run. <table><tr><th>TUR</th><th>Description</th></tr><tr><td rowspan="2">0</td><td>When read, indicates under-run has not been found</td></tr><tr><td>When write, clear itself</td></tr></table>	RW

			1		When read, indicates data has even been read from		
4	RFS	Receive FIFO Service Request. This bit indicates that receive FIFO level is or not below RFL threshold which is controlled by AICFR.RFTH. When RFS is 1, it may trigger interrupt or DMA request depends on the interrupt enable and DMA setting.					R

RFS	Description
0	Receive FIFO level below RFL threshold
1	Receive FIFO level at or above RFL threshold

3	TFS	Transmit FIFO Service Request. This bit indicates that transmit FIFO level exceeds TFL threshold which is controlled by AICFR.TFTH When TFS is 1, it may trigger interrupt or DMA request depends on the interrupt enable and DMA setting.	R

TFS	Description
0	Transmit FIFO level exceeds TFL threshold
1	Transmit FIFO level at or below TFL threshold

2:0	Reserved	Writes to these bits have no effect and always read as 0	R

1.2.7 AIC AC-link Status Register (ACSR)

ACSR contains bits to reflect the status of the connected external CODEC in AC-link format. Bits in this register are read-only in general, except some of them can be written a 0.

ACSR   **0x10020018**

Bit	31 30 29 28 27 26 25 24 23 22	21	20	19	18	17	16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
	Reserved	SLTERR	CRDY	CLPM	RSTO	SADR	CADT	Reserved
RST	0 0 0 0 0 0 0 0 0 0	0	0	0	0	0	0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bits	Name	Description	RW
31:22	Reserved	Writes to these bits have no effect and always read as 0	R
21	SLTERR	Hardware detects a Slot Error. This bit indicates an error in SLOTREQ bits on incoming data from external CODEC is detected. The error can be: (1) find 1 in a SLOTREQ bit, which corresponding to an inactive slot; (2) all active slots should be request in the same time by SLOTREQ, but an exception is found. All errors are accumulated to ACSR.SLTERR by hardware until software clears it. Software writes 0 clear this bit and write 1 has no effect.	RW
20	CRDY	External CODEC Ready. This bit is derived from the CODEC Ready bit of Slot 0 in SDATA_IN, and it indicates the external AC97 CODEC is ready or not.<table><tr><th>CRDY</th><th>Description</th></tr><tr><td>0</td><td>CODEC is not ready</td></tr><tr><td>1</td><td>CODEC is ready</td></tr></table>	R
19	CLPM	External CODEC Low Power Mode. This bit indicates the external CODEC is switched to low power mode or BIT_CLK is active from CODEC after wake up.<table><tr><th>CLPM</th><th>Description</th></tr><tr><td>0</td><td>BIT_CLK is active</td></tr><tr><td>1</td><td>CODEC is switched to low power mode</td></tr></table>	R
18	RSTO	External CODEC Registers Read Status Time Out. This bit indicates that the read status time out is detected or not. It is set to 1 if the data not return in 4 frames after a CODEC registers read command issued.<table><tr><th>RSTO</th><th>Description</th></tr><tr><td>0</td><td>When read, indicates time out has not occurred</td></tr><tr><td>1</td><td>When read, indicates read status time out found</td></tr></table>Write 0 clear this bit and write 1 is ignored. When RSTO is 1, it may trigger an interrupt depends on the interrupt enable setting.	RW
17	SADR	External CODEC Registers Status Address and Data Received. This bit	RW

		indicates that address and data of an external AC '97 CODEC register has or has not been received.	
		<table><tr><th>SADR</th><th>Description</th></tr><tr><td>0</td><td>When read, indicates no register address/data received.</td></tr><tr><td>1</td><td>When read, indicates address/data received.</td></tr></table> Write 0 clear this bit and write 1 is ignored. When SADR is 1, it may trigger an interrupt depends on the interrupt enable setting.	
16	CADT	Command Address and Data Transmitted. This bit indicates that a CODEC register reading/writing command transmission has completed or not. <table><tr><th>CADT</th><th>Description</th></tr><tr><td>0</td><td>When read, indicates the command has not done.</td></tr><tr><td>1</td><td>When read, indicates the command has done.</td></tr></table> Write 0 clear this bit and write 1 is ignored. When CADT is 1, it may trigger an interrupt depends on the interrupt enable setting.	RW
15:0	Reserved	Writes to these bits have no effect and always read as 0	R

1.2.8 AIC I2S/MSB-justified Status Register (I2SSR)

I2SSR is used to reflect AIC status in I2S/MSB-justified. It is a read-only register.

I2SSR   **0x1002001c**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																	Reserved													BSY	Reserved	Reserved
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:3	Reserved	Writes to these bits have no effect and always read as 0	R
2	BSY	AIC busy in I2S/MSB-justified format. <table><tr><th>BSY</th><th>Description</th></tr><tr><td>0</td><td>AIC controller is idle or disabled</td></tr><tr><td>1</td><td>AIC controller currently is transmitting or receiving a frame</td></tr></table>	R
1:0	Reserved	Writes to these bits have no effect and always read as 0	R

1.2.9 AIC AC97 CODEC Command Address Register (ACCAR) & Data Register (ACCDR)

ACCAR and ACCDR are used to hold register address and data for external AC-link CODEC register read/write operation through SDATA_OUT. The format of ACCAR.CAR and ACCDR.CDR is compliant with AC'97 Component Specification 2.3 where ACCAR.CAR[19] of "1" specifies CODEC register read operation, of "0" specifies CODEC register write operation. The write access to ACCAR and ACCDR signals AIC to issue this operation. Please reference to 1.4.4 for software flow. These registers are valid only in AC-link. It is ignored in I2S/MSB-justified format.
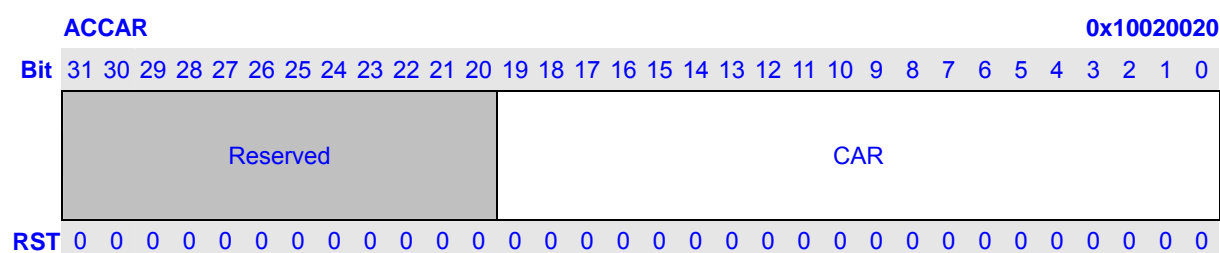
ACCAR   **0x10020020**

Bit	31 30 29 28 27 26 25 24 23 22 21 20	19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
	Reserved	CAR
RST	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bits	Name	Description	RW
31:20	Reserved	Writes to these bits have no effect and always read as 0	R
19:0	CAR	Command Address Register. This is used to hold 20-bit AC '97 CODEC register address transmitted in SDATA_OUT slot 1.	RW
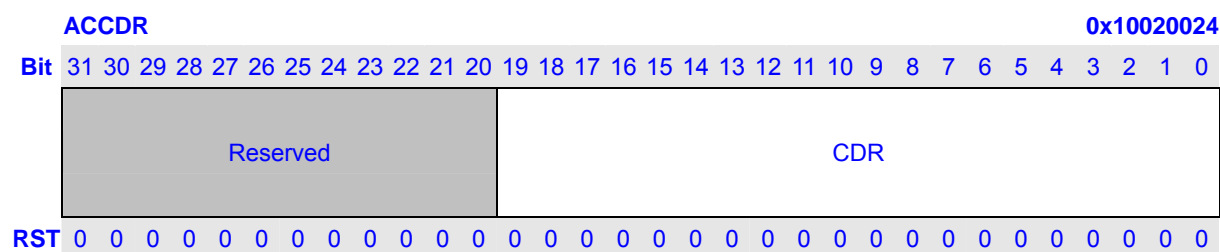
ACCDR   **0x10020024**

Bit	31 30 29 28 27 26 25 24 23 22 21 20	19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
	Reserved	CDR
RST	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bits	Name	Description	RW
31:20	Reserved	Writes to these bits have no effect and always read as 0	R
19:0	CDR	Command Data Register. This is used to hold 20-bit AC'97 CODEC register data transmitted in SDATA_OUT slot 2.	RW

1.2.10 AIC AC97 CODEC Status Address Register (ACSAR) & Data Register (ACSDR)

ACSAR and ACSDR are used to receive the external AC-link CODEC registers address and data from SDATA_IN. When AIC receives CODEC register status from SDATA_IN, it set ACSR.SADR bit and put the address and data to ACSAR.SAR and ACSDR.SDR. Please reference to 1.4.4 for software flow. These registers are valid only in AC-link format and are ignored in I2S/MSB-justified format.

ACSAR   **0x10020028**

Bit	31 30 29 28 27 26 25 24 23 22 21 20	19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
	Reserved	SAR
RST	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bits	Name	Description	RW
31:20	Reserved	Writes to these bits have no effect and always read as 0	R
19:0	SAR	CODEC Status Address Register. This is used to receive 20-bit AC '97 CODEC status address from SDATA_IN slot 1. Which reflect the register index for which data is being returned. The write operation is ignored.	R
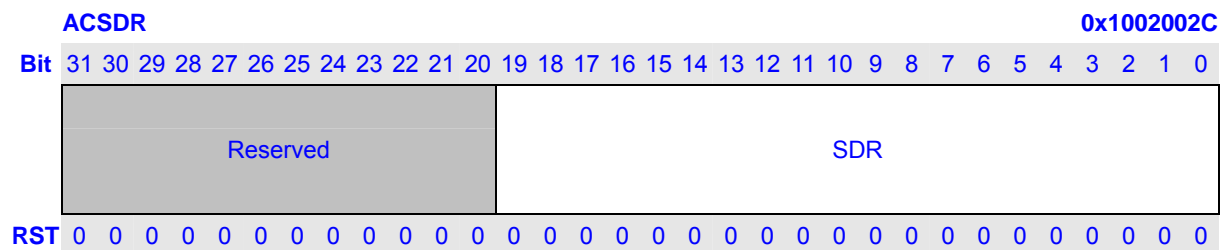
ACSDR   **0x1002002C**

Bit	31 30 29 28 27 26 25 24 23 22 21 20	19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
	Reserved	SDR
RST	0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bits	Name	Description	RW
31:20	Reserved	Writes to these bits have no effect and always read as 0	R
19:0	SDR	CODEC Status Data Register. This is used to receive 20-bit AC '97 CODEC status data from SDATA_IN slot 2. The register data of external CODEC is returned. The write operation is ignored.	R

1.2.11 AIC I2S/MSB-justified Clock Divider Register (I2SDIV)

I2SDIV is used to set clock divider to generated BIT_CLK from SYS_CLK in I2S/MSB-justified format.

I2SDIV   **0x10020030**

Bit	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4	3 2 1 0
	Reserved	DV
RST	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 1 1

Bits	Name	Description	RW
31:4	Reserved	Writes to these bits have no effect and always read as 0	R
3:0	DV	Audio BIT_CLK clock divider value minus 1. I2SDIV.DV is used to control the generating of BIT_CLK from dividing SYS_CLK. The dividing value should be even and I2SDIV.DV should be set to the dividing value minus 1. So I2SDIV.DV bit0 is fixed to 1. BIT_CLK frequency is fixed to 64 f_S in AIC, where f_S is the audio sample frequency. I2SDIV.DV depends on SYS_CLK frequency f_{SYS_CLK}, which is selected according to external CODEC's requirement and internal PLL frequency. Please reference to 1.4.9 "Serial Audio Clocks and Sampling Frequencies" for further description.	RW

1.2.12 AIC FIFO Data Port Register (AICDR)

AICDR is act as data input port to transmit FIFO when write and data output port from receive FIFO when read, one audio sample every time. The FIFO width is 24 bits. Audio sample with size N that is less than 24 is located in LSB N-bits. The sample size is specified by ACCR2.OASS and ACCR2.IASS in AC-link, and by I2SCR.WL in I2S/MSB-justified. The sample order is specified by ACCR1.XS and ACCR1.RS in AC-link. In I2S/MSB-justified, the left channel sample is prior to the right channel.

Care should be taken to monitor the status register to insure that there is room for data in the FIFO when executing a program read or write transaction. This is taken care automatically in DMA.

AICDR   **0x10020034**

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				Reserved																DATA												
RST	0	0	0	0	0	0	0	0	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?
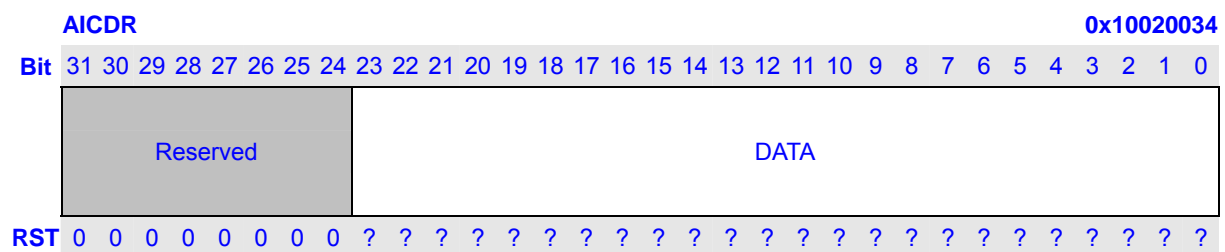
Bits	Name	Description	RW
31:24	Reserved	Writes to these bits have no effect and always read as 0	R
23:0	DATA	FIFO port. When write to it, data is push to the transmit FIFO. When read from it, data is pop from the receiving FIFO.	RW

1.3 Serial Interface Protocol

1.3.1 AC-link serial data format

Following figures are AC-link serial data format. Audio data is MSB adjusted, regardless of 8, 16, 18, 20, 24 bits sample size. When a 24-bits sample is transmitted, the LSB 4-bits are truncated. When try to record 24-bits sample, 4-bits of 0 are appended in LSB. Please reference to "AC '97 Component Specification Revision 2.3, 2002", provided by Intel Corporation, for details of AC '97 architecture and AC-link specification.
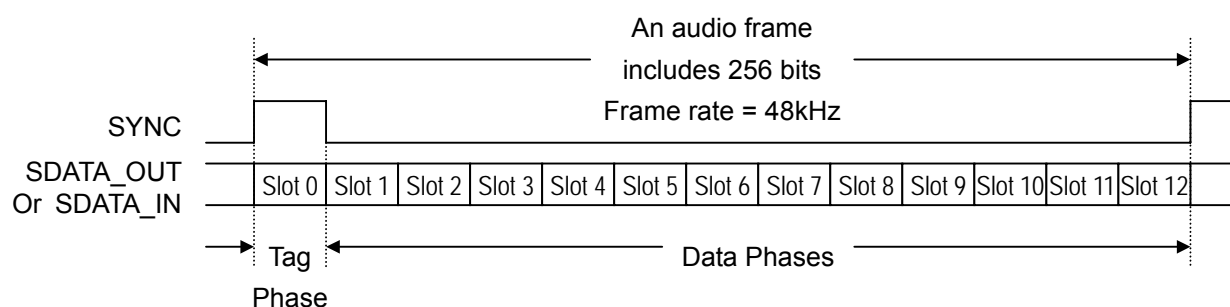
Figure 1-6 AC-link audio frame format
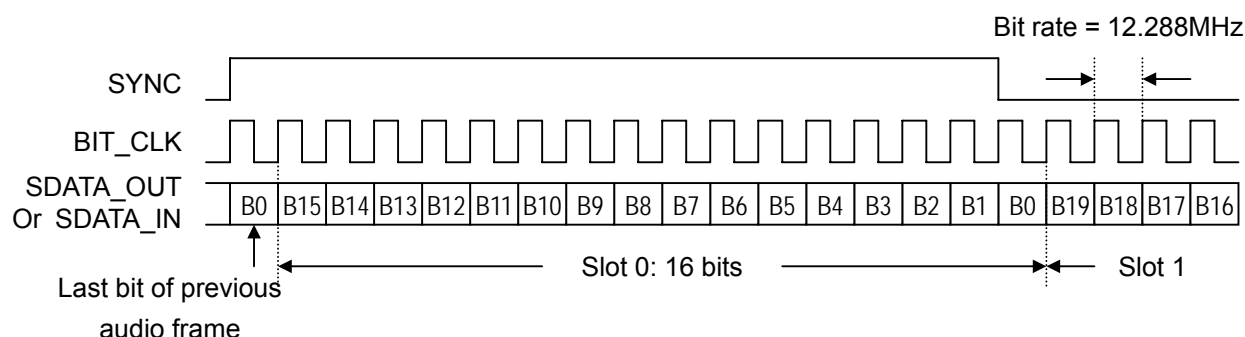
Figure 1-7 AC-link tag phase, slot 0 format
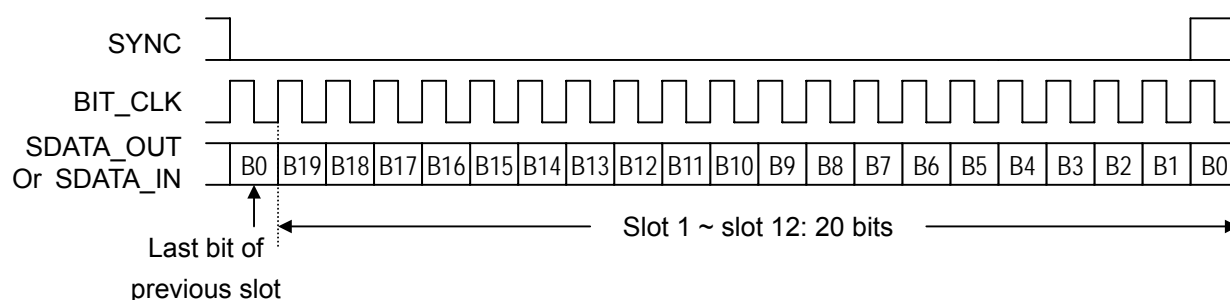
Figure 1-8 AC-link data phases, slot 1 ~ slot 12 format

1.3.2 I2S and MSB-justified serial audio format

Normal I2S and MSB-justified are similar protocols for digitized stereo audio transmitted over a serial path.

The BIT_CLK supplies the serial audio bit rate, the basis for the external CODEC bit-sampling logic. Its frequency is 64 times the audio sampling frequency. Divided by 64, the resulting 8 kHz to 48 kHz or even higher signal signifies timing for left and right serial data samples passing on the serial data paths. This left/right signal is sent to the CODEC on the SYNC pin. Each phase of the left/right signal is accompanied by one serial audio data sample on the data pins SDATA_IN and SDATA_OUT.
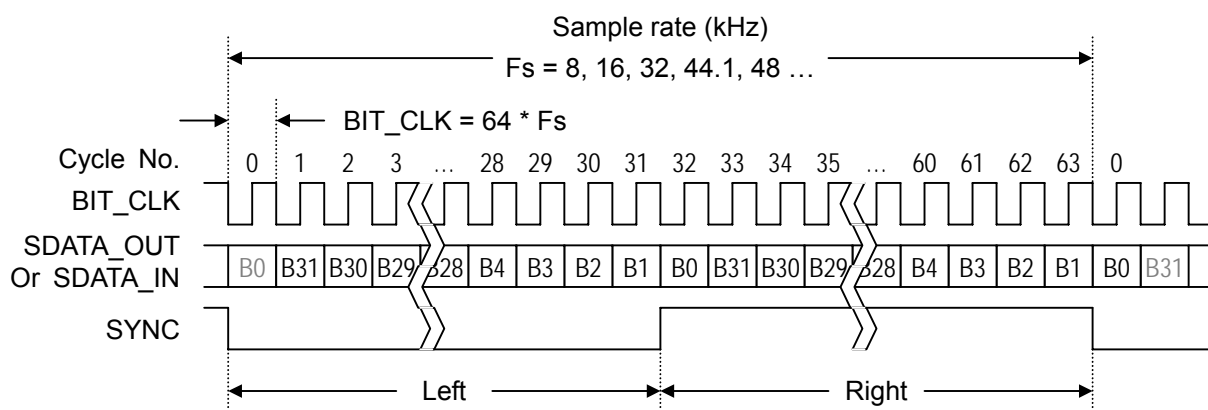
Figure 1-9 I2S data format
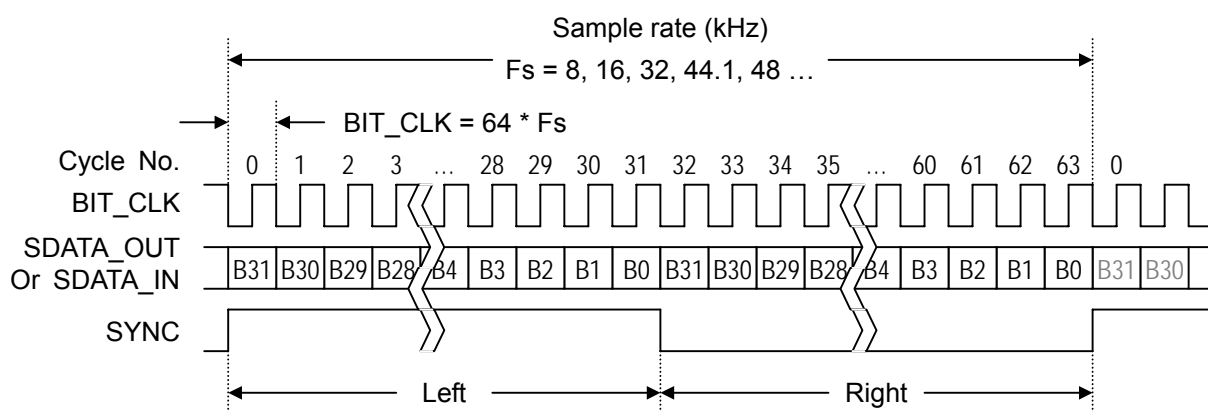
Figure 1-10 MSB-justified data format

Figure 1-9 and Figure 1-10 provide timing diagrams that show formats for the normal I2S and MSB-justified modes of operations. Data is sampled on the rising edge of the BIT_CLK and data is sent out on the falling edge of the BIT_CLK.

Data is transmitted and received in frames of 64 BIT_CLK cycles. Each frame consists of a left sample and a right sample. Each sample holds 8, 16, 18, 20 or 24 bits of valid data. The LSB other bits of each sample is padded with zeroes.

- In the normal I2S mode, the SYNC is low for the left sample and high for the right sample. Also, the MSB of each data sample lags behind the SYNC edges by one BIT_CLK cycle.
- In the MSB-justified mode, the SYNC is high for the left sample and low for the right sample. Also, the MSB of each data sample is aligned with the SYNC edges.

When use with the internal CODEC, the BIT_CLK and SYNC signals are provided by the internal CODEC from the 12MHz clock.

1.3.3 Audio sample data placement in SDATA_IN/SDATA_OUT

The placement of audio sample in incoming/outgoing serial data stream for all formats support in AIC is MSB (Most Significant Bit) justified. Suppose n bit sample composed by

S_{n-1}	S_{n-2}	…	S_2	S_1	S_0

Table 1-3 described the how sample data bits are transferred.

Table 1-3 Sample data bit relate to SDATA_IN/SDATA_OUT bit

AC-link Format						I2S/MSB-Justified Format					
SDATA IN/OUT	Audio Sample Size (bit)					SDATA IN/OUT					
	8	16	18	20	24		8	16	18	20	24
B19	S7	S15	S17	S19	S23	B31	S7	S15	S17	S19	S23
B18	S6	S14	S16	S18	S22	B30	S6	S14	S16	S18	S22
B17	S5	S13	S15	S17	S21	B29	S5	S13	S15	S17	S21
B16	S4	S12	S14	S16	S20	B28	S4	S12	S14	S16	S20
B15	S3	S11	S13	S15	S19	B27	S3	S11	S13	S15	S19
B14	S2	S10	S12	S14	S18	B26	S2	S10	S12	S14	S18
B13	S1	S9	S11	S13	S17	B25	S1	S9	S11	S13	S17
B12	S0	S8	S10	S12	S16	B24	S0	S8	S10	S12	S16
B11	0	S7	S9	S11	S15	B23	0	S7	S9	S11	S15
B10	0	S6	S8	S10	S14	B22	0	S6	S8	S10	S14
B9	0	S5	S7	S9	S13	B21	0	S5	S7	S9	S13
B8	0	S4	S6	S8	S12	B20	0	S4	S6	S8	S12
B7	0	S3	S5	S7	S11	B19	0	S3	S5	S7	S11
B6	0	S2	S4	S6	S10	B18	0	S2	S4	S6	S10
B5	0	S1	S3	S5	S9	B17	0	S1	S3	S5	S9
B4	0	S0	S2	S4	S8	B16	0	S0	S2	S4	S8
B3	0	0	S1	S3	S7	B15	0	0	S1	S3	S7
B2	0	0	S0	S2	S6	B14	0	0	S0	S2	S6
B1	0	0	0	S1	S5	B13	0	0	0	S1	S5
B0	0	0	0	S0	S4	B12	0	0	0	S0	S4
						B11	0	0	0	0	S3
						B10	0	0	0	0	S2
						B9	0	0	0	0	S1
						B8	0	0	0	0	S0
						B7~ B0	0	0	0	0	0

1.4   Operation

The AIC can be accessed either by the processor using programmed I/O instructions or by the DMA controller. Jz4740/20 processor uses programmed I/O instructions to access the AIC and can access the following types of data

- The AIC memory mapped registers data—All registers are 32 bits wide and are aligned to word boundaries.
- AIC controller FIFO data—An entry is placed into the transmit FIFO by writing to the I2S controller's Serial Audio Data register (AICDR). Writing to AICDR updates a transmit FIFO entry. Reading AICDR flushes out a receive FIFO entry.
- The external CODEC registers for I2S CODEC—CODEC registers can be accessed through the L3 bus. The L3 bus operation is emulated by software controlling three GPIO pins.
- The external CODEC registers for AC97 CODEC—An AC97 audio CODEC can contain up to sixty-four 16-bit registers. A CODEC uses a 16-bit address boundary for registers. The AIC supplies access to the CODEC registers through several registers.
- The internal CODEC registers can be accessed via memory mapped registers in the CODEC.

The DMA controller can only access the FIFOs. Accesses are made through the data registers, as explained in the previous paragraph. The DMA controller responds to the following DMA requests made by the I2S controller:

- The transmit FIFO request is based on the transmit trigger-threshold (AICFR.TFTH) setting. See 1.2.1 for further details regarding AICFR.TFTH.
- The receive FIFO request is based on the receive trigger-threshold (AICFR.RFTH) setting. See 1.2.1 for further details regarding AICFR.RFTH.

Before operation to AIC, you may need to set proper PIN function selection from GPIO using if the pin is shared with GPIO.

Please also reference to "AC '97 Component Specification Revision 2.3, 2002" when deal with AIC AC-link operations.

1.4.1 Initialization

At power-on or other hardware reset (WDT and etc), AIC is disabled. Software must initiate AIC and the internal or external CODEC after power-on or reset. If errors found in data transferring, or in other places, software must initial AIC and optional, the internal or external CODEC. Here is the initial flow.

1. Select internal or external CODEC (AICFR.ICDC).
2. If external CODEC is selected, select AC-link or I2S/MSB-Justified (AICFR.AUSEL). If internal CODEC is used, select I2S/MSB-Justified format (AICFR.AUSEL=1). If the resettlement without involving link format and architecture changing, this step can be skip.
3. If I2S/MSB-Justified is selected, select between I2S and MSB-Justified (I2SCR.AMSL), decide BIT_CLK direction (AICFR.BCKD) and SYNC direction (AICFR.SYNCD). If BIT_CLK is configured as output, BIT_CLK divider I2SDIV.DV must be set to what correspond with the values as shown in Table 1-7. And the clock selection and the divider between PLL clock out and AIC also must be set (CFCR.I2S and I2SCDR in CPM). If internal CODEC is used, select 12MHz clock input (via set proper value in CFCR.I2S and I2SCDR), I2S format (I2SCR.AMSL=0), input BIT_CLK (AICFR.BCKD=0), input SYNC (AICFR.SYNCD=0).
4. Enable AIC by write 1 to AICFR.ENB
5. If it needs to reset AIC registers and flush FIFOs, write 1 to AICFR.RST. If it need only flush FIFOs, write 1 to AICCR.FLUSH. BIT_CLK must exist here and after.
6. In AC-link format, issue a warm or cold CODEC reset.
7. In AC-link format, configure AC '97 CODEC via ACCAR and ACCDR registers. If the resettlement doesn't involving AC'97 CODEC registers changing, this step can be skip.
8. In case of external CODEC with I2S/MSB-Justified format, configure I2S/MSB-justified CODEC via the control bus connected to the CODEC, for instance I2C or L3, depends on CODEC. In case of internal CODEC, configure CODEC via CODEC's memory mapped registers. If the resettlement without involving I2S/MSB-justified CODEC or ADC/DAC function changing, this step can be skip.

1.4.2　AC '97 CODEC Power Down

AC '97 CODEC can be placed in a low power mode. When the CODEC's power-down register (26h), is programmed to the appropriate value, the CODEC will be put in a low power mode and both BIT_CLK and SDATA_IN will be brought to and held at a logic low voltage level.

Once powered down, re-activation of the AC-link via re-assertion of the SYNC signal must not occur for a minimum of four audio frame times following the frame in which the power down was triggered. When AC-link powers up it indicates readiness via the CODEC Ready bit (input slot 0, bit 15).

1.4.3　Cold and Warm AC '97 CODEC Reset

AC-link reset operations occur when the system is initially powered up, when resuming from a lower powered sleep state, and in response to critical subsystem failures that can only be recovered from with a reset.

1.4.3.1　Cold AC '97 CODEC Reset

A cold reset is achieved by asserting RESET# for the minimum specified time. By driving RESET# low, BIT_CLK, and SDATA_IN will be activated, or re-activated as the case may be, and all AC '97 CODEC registers will be initialized to their default power on reset values.

RESET# is an asynchronous AC '97 CODEC input.
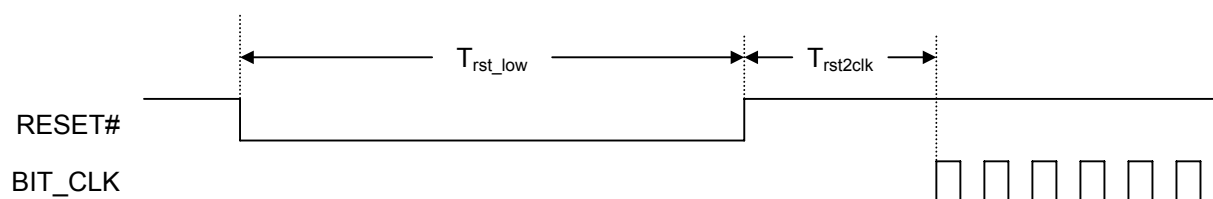
Figure 1-11 Cold AC '97 CODEC Reset Timing

Table 1-4 Cold AC '97 CODEC Reset Timing parameters

Parameter	Symbol	Min	Type	Max	Units
RESET# active low pulse width	T_{rst_low}	1.0	-	-	µs
RESET# inactive to BIT_CLK startup delay	$T_{rst2clk}$	162.8	-	-	ns

1.4.3.2 Warm AC '97 CODEC Reset

A warm AC'97 reset will re-activate the AC-link without altering the current AC'97 register values. Driving SYNC high for a minimum of 1 µs in the absence of BIT_CLK signals a warm reset.

Within normal audio frames SYNC is a synchronous AC '97 CODEC input. However, in the absence of BIT_CLK, SYNC is treated as an asynchronous input used in the generation of a warm reset to AC '97 CODEC.
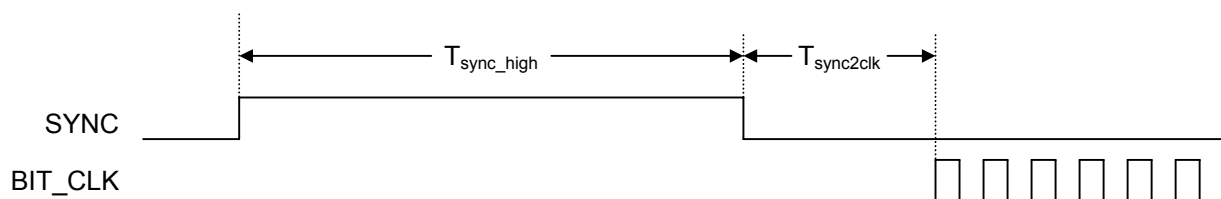

Figure 1-12 Warm AC '97 CODEC Reset Timing

Table 1-5 Warm AC '97 CODEC Reset Timing Parameters

Parameter	Symbol	Min	Type	Max	Units
SYNC active high pulse width	T_{sync_high}	1.0	-	-	µs
SYNC inactive to BIT_CLK startup delay	$T_{sync2clk}$	162.8	-	-	ns

1.4.4 External CODEC Registers Access Operation

The external audio CODEC can be configured/controlled by its internal registers. To access these registers, an I2S/MSB-justified CODEC usually employs L3 bus, SPI bus, I2C bus or other control bus. The L3 bus operation can be emulated by software controlling 3 GPIO pins in Jz4730. For AC '97, "AC '97 Component Specification" defines the CODEC register access protocol. Several registers are provided in AIC to accomplish this task.

The ACCAR and ACCDR are used to send a register accessing request command to external AC '97 CODEC. The ACSAR and ACSDR are used to receive a register's content from external AC'97 CODEC. The register accessing request and the register's content returning is asynchronous.

The AC'97 CODEC register accessing request flow:
1. If ACSR.CADT is 0, wait for 25.4µs. If no previous accessing request, this step can be skip.
2. Clear ACSR.CADT.
3. If read access, write read-command and register address to ACCAR, if write access, write write-command and register address to ACCAR and write data to ACCDR. Any order of write ACCAR and ACCDR is OK.
4. Polling for ACSR.CADT changing to 1, which means the request has been send to CODEC via AC-link.

The AC'97 CODEC register content receiving flow by polling:
1. Polling for ACSR.SADR changing to 1
2. Read the CODEC register's address from ACSAR and content from ACSDR
3. Clear ACSR.SADR

The AC'97 CODEC register content receiving flow by interrupt:
1. Before accessing request, clear ACSR.SADR and set ACCR2.ESADR.
2. Waiting for the interrupt. When the interrupt is found, check if ACSR.SADR is 1, if not, repeat this step again.
3. Read the CODEC register's address from ACSAR and content from ACSDR
4. Clear ACSR.SADR

1.4.5   Audio Replay

Outgoing audio sample data (from AIC to CODEC) is written to AIC transmit FIFO from processor via store instruction or from memory via DMA. AIC then takes the data from the FIFO, serializes it, and sends it over the serial wire SDATA_OUT to an external CODEC or over an internal wire to an internal CODEC.

The audio transmission is enabled automatically when the AIC is enabled by set AICFR.ENB. But all replay data is zero at this time except both of the following conditions are true:

1. AICCR.ERPL must be 1. If AICCR.ERPL is 0, value of zero is send to CODEC even if there are samples in transmit FIFO.
2. At least one audio sample data in the transmit FIFO. If the transmit FIFO is empty, value of zero or last sample depends on AICFR.LSMP, is send to CODEC even if AICCR.ERPL is 1.

Here is the audio replay flow:
1. Configure CODEC as needed.
2. Configure sample size by AICCR.OSS
3. Configure sample rate by clock dividers (for I2S/MSB-Justified format with BIT_CLK is provided internally) or by CODEC registers (for AC-link or BIT_CLK provided by external CODEC) or by CODEC memory mapped registers (for internal CODEC)
4. For AC-link, configure replay channels by ACCR1.XS
5. Some other configurations: mono to stereo, endian switch, signed/unsigned data transfer, transmit FIFO configuration, play ZERO or last sample when TX FIFO under-run, and etc.
6. Write 1 to AICCR.ERPL. It is suggested that at least a frame of PCM data is pre-filled in the transmit FIFO to prevent FIFO under-run flag (AICSR.TUR).
7. Fill sample data to the transmit FIFO. Repeat this till finish all sample data. In this procedure, please control the FIFO to make sure no FIFO under-run and other errors happen. When the transmit FIFO under-run, noise or pause may be heard in the audio replay, AICSR.TUR is 1, and if AICCR.ETUR is 1, AIC issues an interrupt. Please reference to 1.4.7 for detail description on FIFO.
8. Waiting for AICSR.TFL change to 0. So that all samples in the transmit FIFO has been replayed, then we can have a clean start up next time
9. Write 0 to AICCR.ERPL.

1.4.6   Audio Record

Incoming audio sample data (from CODEC to AIC) is received from SDATA_IN (for an external CODEC) or an internal wire (for an internal CODEC) serially and converted to parallel word and stored in AIC receive FIFO. Then the data can be taken from the FIFO to processor via load instruction or to memory via DMA.

The audio recording is enabled automatically when the AIC is enabled by set AICFR.ENB. But all received data is discarded at this time except both of the following conditions are true:

1. AICCR.EREC must be 1. If AICCR.EREC is 0, the received data is discarded even if there are rooms in the receive FIFO.
2. At least one room left in the receive FIFO. If the receive FIFO is full, the received data is discarded even if AICCR.EREC is 1.

Here is the audio record flow:
1. Configure CODEC as needed.
2. Configure sample size by AICCR.ISS
3. Configure sample rate by clock dividers (for I2S/MSB-Justified format with BIT_CLK is provided internally) or by CODEC registers (for AC-link or BIT_CLK provided by external CODEC) or by CODEC memory mapped registers (for internal CODEC)
4. Some other configurations: signed/unsigned data transfer, receive FIFO configuration, and etc.
5. Write 1 to AICCR.EREC. Make sure there are rooms available in the receive FIFO before set AICCR.EREC. Usually, it should empty the receive FIFO by fetch data from it before set AICCR.EREC
6. Take sample data form the receive FIFO. Repeat this till the audio finished. In this procedure, please control the FIFO to make sure no FIFO over-run and other errors happen. When the receive FIFO over-run, same recorded audio samples will be lost, AICSR.ROR is 1, and if AICCR.EROR is 1, AIC issues an interrupt. Please reference to 1.4.7 for detail description on FIFO. For AC-link, ACCR1.RS tells which channels are recorded.
7. Write 0 to AICCR.EREC.
8. Take sample data from the receive FIFO until AICSR.RFL change to 0. So that all samples in the receive FIFO has been taken away, then we can have a clean start up next time. When the receive FIFO is empty, read from it returns zero.

1.4.7 FIFOs operation

AIC has two FIFOs, one for transmit audio sample and one for receive. All AIC played/recorded audio sample data is taken from/send to transmit/receive FIFOs. The FIFOs are in 24 bits width and 32 entries depth, one entry for keep one audio sample regardless of the sample size. AICDR.DATA provides the access point for processor/DMA to write to transmit FIFO and read from receive FIFO. One time access to AICDR.DATA process one sample. The sample data should be put in LSB (Least Significant Bit) in memory or processor registers. For transmitting, bits exceed sample are discarded. For receiving, these bits are set to 0. Figure 1-13 illustrates the FIFOs access.
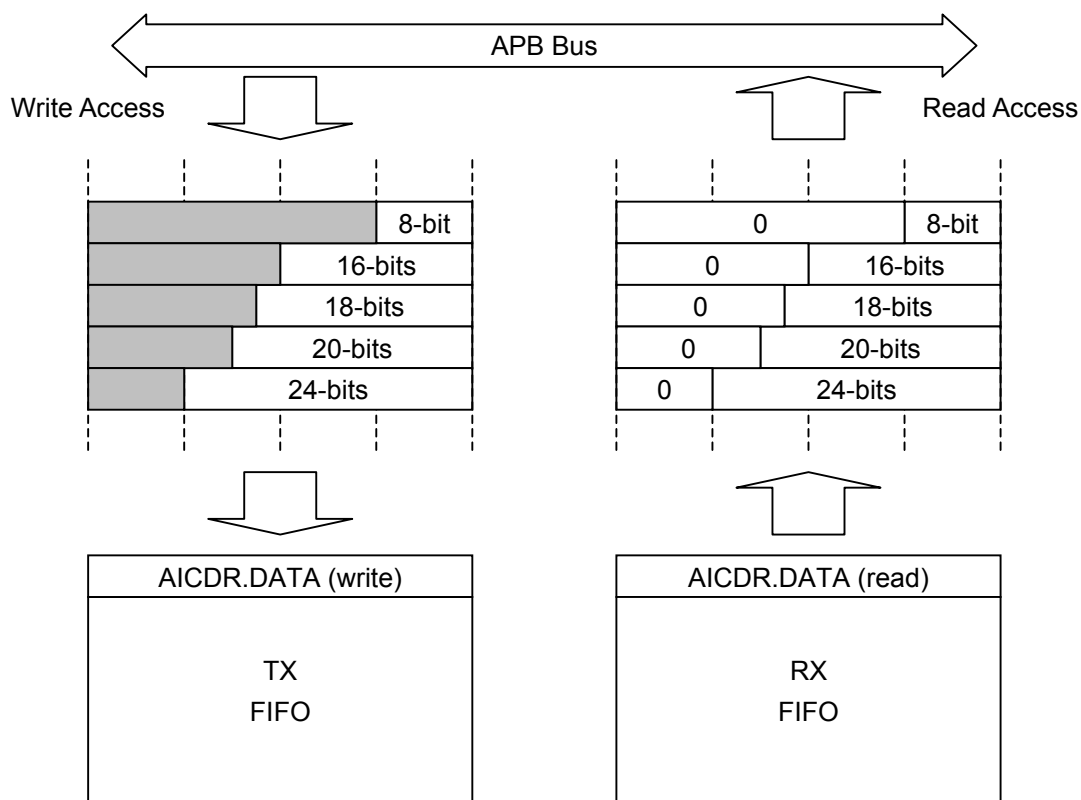
Figure 1-13 Transmitting/Receiving FIFO access via APB Bus

The software and bus initiator must guarantee the right sample placement at the bus.

In case of DMA bus initiator, one 24, 20, 18 bits audio sample must occupies one 32-bits word in memory at LSB location, and use 32-bits width DMA. One 16 bits sample occupies one 16-bits half word in memory, and use 16-bits width DMA. One 8-bits sample occupies one byte in memory, and use 8-bits width DMA.

In case of processor bus initiator, any type of the audio sample must occupies one CPU general-purpose register at LSB, and read/write from/to AICDR.DATA with 32-bits load/store

Jz4740 peripheral specification, Revision 1.0

instruction. When process small sample size, 16-bits or 8-bits, software may need to do the data pack/unpack.

The AICFR.TFTH and AICFR.RFTH are used to set the FIFO level thresholds, which are the trig levels of DMA request and/or FIFO service interrupt. The AICFR.TFTH and AICFR.RFTH should be set to proper values, too small or too big are not good. When it is too small, the DMA burst length or the number of sample can be processed by processor is too small, which harms the bus or processor efficiency. When it is too big, the bus or the interrupt latency left for under-run/over-run is too small, which may causes replay/record errors.

AICSR.TUR is set to 1 during transmit under-run conditions. If AICCR.ETUR is 1, this can trigger an interrupt. During transmit under-run conditions, zero or last sample is continuously sent out across the serial link. Transmit under-run can occur under the following conditions:
1. Valid transmit data is still available in memory, but the DMA controller/processor starves the transmit FIFO, as it is busy servicing other higher-priority tasks.
2. The DMA controller/processor has transferred all valid data from memory to the transmit FIFO.

AICSR.ROR is set to 1 during receive over-run conditions. If AICCR.EROR is 1, this can trigger an interrupt. During receive over-run conditions, data sent by the CODEC is lost and is not recorded.

When replay/record two channels data, the left channel is always the first data in FIFOs and in the serial link. If multiple channels in AC-link are used, the channel sample order is follows the slot order.

1.4.8 Data Flow Control

There are three approaches provided by Jz4730 to control/synchronize the audio incoming/outgoing data flow.

1.4.8.1 Polling and Processor Access

AICSR.RFL and AICSR.TFL reflect how many samples exist in receiving and transmitting FIFOs. Through read these register fields, processor can detect when there are samples in receiving FIFO in audio record and then load them from the RX-FIFO, and when there are rooms in transmitting FIFO in audio replay and then store samples to the TX-FIFO.

Polling approach is in very low efficiency and is not recommended.

1.4.8.2 Interrupt and Processor Access

Set proper values to AICFR.TFTH and AICFR.RFTH, the FIFO interrupts trig thresholds. Set AICCR.ETFS and/or AICCR.ERFS to 1 to enable transmitting and/or receiving FIFO level trigger interrupts. When the interrupt found, it means there are rooms or samples in the TX or RX FIFO, and processor can store or load samples to or from the FIFO.

Interrupt approach is more efficient than polling approach.

1.4.8.3 DMA Access

Audio data is real time stream, though it is in low data bandwidth, usually less than 1.2Mbps. DMA approach is the most efficient and is the recommended approach.

To enable DMA operation, set AICCR.TDMS and AICCR.RDMS to 1 for transmit and receive respectively. It also needs to allocate two channels in DMA controller for data transmitting and receiving respectively. Please reference to Jz4730 DMAC spec for the details.

The AICFR.TFTH and AICFR.RFTH are used to set the transmitting and receiving FIFO level thresholds, which determine the issuing of DMA request to DMA controller. To respond the request, DMAC initiator and controls the data movement between memory and TX/RX FIFO.

1.4.9 Serial Audio Clocks and Sampling Frequencies

For internal CODEC, CODEC module containing the audio CODEC circuit/logic and corresponding controlling registers. CODEC needs a 12MHz clock from CPM and provides BIT_CLK and LR_CLK (left-right clock which is the sample rate as SYNC) to AIC for outgoing and incoming audio respectively. These clocks change when change the sample rate in CODEC controlling registers.

For AC-link, the bit clock is input from chip external and is fixed to 12.288MHz. The sample frequency of 48kHz is supported in nature. Variable Sample Rate feature is supported in this AIC. If the CODEC supports this feature, sample rate other than 48kHz audio data can be replay directly. Otherwise, software has to do the rate transfer to replay other sample rate audio data. Double rate, 96kHz or even 88.2kHz audio is also supported with proper CODEC.

Following are for BIT_CLK/SYS_CLK configuration in I2S/MSB-Justified format with external CODEC.

The BIT_CLK is the rate at which audio data bits enter or leave the AIC. BIT_CLK can be supplied either by the CODEC or an internally PLL. If it is supplied internally, BIT_CLK is configured as output pins, and is supplied to the CODEC. If BIT_CLK is supplied by the CODEC, then it is configured as an input pin. Register bit AICFR.BCKD is used to select BIT_CLK direction.

The audio sampling frequency is the frequency of the SYNC signal, which must be 1/64 of BIT_CLK, $f_{BIT\_CLK} = 64\ f_S$.

SYS_CLK is only for CODEC. It usually takes one of the two roles, as CODEC master clock input or as CODEC over-sampling clock input. If SYS_CLK roles as CODEC master clock input, it usually should be set to a fixed frequency according to CODEC requirement but independent to audio sample rate. In this case, usually there is a PLL in the CODEC and CODEC roles master mode. See Figure 1-4 for the interface diagram. This is the recommended AIC CODEC system configuration.

If SYS_CLK roles as CODEC over-sampling clock, its frequency is usually 4, 6, 8 or 12 times of BIT_CLK frequency, which are 256, 384, 512 and 768 times of audio sample rates. Table 1-6 lists the relation between sample rate, BIT_CLK and SYS_CLK frequencies.

Table 1-6 Audio Sampling rate, BIT_CLK and SYS_CLK frequencies

Sample Rate f_S (kHz)	BIT_CLK (MHz) $f_{BIT\_CLK} = 64\ f_S$	SYS_CLK (MHz)			
		256 f_S	384 f_S	512 f_S	768 f_S
48	3.072	12.288	18.432	24.576	36.864
44.1	2.8224	11.2896	16.9344	22.5792	33.8688
32	2.048	8.192	12.288	16.384	24.576
24	1.536	6.144	9.216	12.288	18.432

22.05	1.4112	5.6448	8.4672	11.2896	16.9344
16	1.024	4.096	6.144	8.192	12.288
11.025	0.7056	2.8224	4.2336	5.6448	8.4672
8	0.512	2.048	3.072	4.096	6.144

In Jz4740, SYS_CLK can be selected from EXCLK or generated by dividing the PLL output clock in a CPM divider controlled by I2SCDR. If BIT_CLK is chosen as an output, another divider in AIC is used to divide SYS_CLK for it. Figure 1-1 illustrates the AIC clock generation scheme.

Figure 1-14 SYS_CLK, BIT_CLK and SYNC generation scheme

The setting of I2SDIV.DV is shown in Table 1-7.

Table 1-7 BIT_CLK divider setting

I2SDIV.DV	f_{SYS_CLK}	f_{BIT_CLK}	$f_{SYS\_CLK} / f_{BIT\_CLK}$
0x1	128 f_S	64 f_S	2
0x2	196 f_S	64 f_S	3
0x3	256 f_S	64 f_S	4
0x5	384 f_S	64 f_S	6
0x7	512 f_S	64 f_S	8
0xB	768 f_S	64 f_S	12

As we observe in Table 1-6, if SYS_CLK is taken as over-sampling clock by CODEC, the common multiple of all SYS_CLK frequencies is much bigger than the PLL output clock frequency. To generate all different SYS_CLK frequencies, one approach is change PLL frequency according to sample rate. This is not realistic, since frequently change PLL frequency during normal operation is not recommended.

Another approach is to found some approximate common multiples of all SYS_CLK frequencies

according to the fact that there tolerance in audio sample rate. Take f_{SYS_CLK} = 256 f_S, Table 1-8 list most frequencies, which are less than 400MHz, with relatively small sample rate errors. It is suggested to set PLL frequency as close to the frequencies listed as possible, then use clock dividers to generate different SYS_CLK/BIT_CLK for different sample rate.

Table 1-8 Approximate common multiple of SYS_CLK for all sample rates

Approximate Common Frequency (MHz)	Max Error Caused in Audio Sample Rate (%)
123.53	0.53
147.11	0.24
170.68	0.79
235.5	0.87
247.06	0.53
270.64	0.11
280.56	0.73
294.22	0.24
305.14	0.67
317.79	0.53
329.57	0.66
341.35	0.79
347	0.85
353.13	0.90
358.79	0.69
370.59	0.53
382.96	0.54
394.17	0.24

Take PLL = 270.64 MHz as an example, Table 1-9 lists the divider settings for various sample rates.

Table 1-9 CPM/AIC clock divider setting for various sampling rate if PLL = 270.64MHz

Sample Rate (kHz)	CFCR.I2S	I2SDIV.DV	Sample Rate Error (%)
48	1	11	0.11
44.1	1	12	-0.11
32	0	33	0.11
24	1	22	0.11
22.05	1	24	-0.11
16	1	33	0.11
12	1	44	0.11
11.025	1	48	-0.11
8	1	66	0.11

For an EXCLK clock frequency, try to generate PLL frequencies as close to the frequencies listed in Table 1-8 as possible. Table 1-10 lists the PLL parameters and audio sample errors at different PLL frequencies for EXCLK at 12MHz.

Table 1-10 PLL parameters and audio sample errors for EXCLK=12MHz

PLL			Max Sample Rate Error
M	N	Freq (MHz)	
103	10	123.6	0.59%
49	4	147	0.31%
128	9	170.67	0.79%
157	8	235.5	0.87%
103	5	247.2	0.59%
65	3	260	0.82%
45	2	270	0.35%
203	9	270.67	0.12%
113	5	271.2	0.32%
187	8	280.5	0.75%
237	10	284.4	0.81%
49	2	294	0.31%
178	7	305.14	0.67%
53	2	318	0.60%
302	11	329.45	0.70%
256	9	341.33	0.79%
318	11	346.91	0.88%
206	7	353.14	0.90%
299	10	358.8	0.69%
247	8	370.5	0.55%
351	11	382.91	0.55%
230	7	394.29	0.27%

The BIT_CLK should be stopped temporary when change the divider settings, or when change BIT_CLK source (from internal or external), to prevent clock glitch. Register I2SCR.STPBK is provided to assist the task. When I2SCR.STPBK = 1, BIT_CLK is disabled no matter whether it is generated internally or inputted from the external source. The operation flow is described in following.

1. Stop all replay/record by clear AICCR.ERPL and AICCR.EREC.
2. Polling I2SSR.BSY till it is 0
3. Stop the BIT_CLK by write 1 to I2SCR.STPBK
4. Operations concerning BIT_CLK
5. Resume the BIT_CLK by write 0 to I2SCR.STPBK

1.4.10 Interrupts

The following status bits, if enabled, interrupt the processor:

- Receive FIFO Service (AICSR.RFS). It's also DMA Request
- Transmit FIFO Service (AICSR.TFS). It's also DMA Request
- Transmit Under-Run (AICSR.TUR)
- Receive Over-Run (AICSR.ROR)
- Command Address and Data Transmitted, AC-link only (ACSR.CADT)
- External CODEC Registers Status Address and Data Received, AC-link only (ACSR.SADR)
- External CODEC Registers Read Status Time Out, AC-link only (ACSR.RSTO)

For further details, see the corresponding register description sections.