JZ4770

Mobile Application Processor

User-Interfaces Programming Manual

Release Date: Aug. 15, 2011

北京君正集成电路股份有限公司
Ingenic Semiconductor Co.,Ltd.

JZ4770 Mobile Application Processor

User-Interfaces Programming Manual

Disclaimer

This documentation is provided for use with Ingenic products. No license to Ingenic property rights is granted. Ingenic assumes no liability, provides no warranty either expressed or implied relating to the usage, or intellectual property right infringement except as provided for by Ingenic Terms and Conditions of Sale.

Ingenic products are not designed for and should not be used in any medical or life sustaining or supporting equipment.

All information in this document should be treated as preliminary. Ingenic may make changes to this document without notice. Anyone relying on this documentation should contact Ingenic for the current documentation and errata.

Ingenic Semiconductor Co., Ltd.

Room 108, Building A, Information Center, Zhongguancun Software Park
8 Dongbeiwang West Road, Haidian District, Beijing, China,
Tel: 86-10-82826661
Fax: 86-10-82825845
Http: //www.ingenic.cn

CONTENTS

TABLES

FIGURES

1 LCD Controller

1.1 Overview

The JZ integrated LCD controller has the capabilities to driving the latest industry standard STN and TFT LCD panels. It also supports some special TFT panels used in consuming electronic products. The controller performs the basic memory based frame buffer and palette buffer to LCD panel data transfer through use of a dedicated DMA controller. Temporal dithering (frame rate modulation) is supported for STN LCD panels. And OSD is also supported for LCD controller.

Features:

- Basic Features
  - Support PAL/NTSC TV out. 3-components (YUV) TV out (refer TVE spec). VGA
  - Support CCIR601/656 data format
  - Single and Dual panel displays in STN mode
  - Single panel displays in TFT mode
  - Display size up to 1280x720@60Hz(BPP24)
  - Internal palette RAM 256x16 bits

- Colors Supports
  - Encoded pixel data of 1, 2, 4, 8 or 16 BPP in STN mode
  - Support 2, 4, 16 grayscales and up to 4096 colors in STN mode
  - Encoded pixel data of 1, 2, 4, 8, 16, 18 or 24 BPP in TFT mode
  - Support 65,536(65K), 262,144(260K) and up to 16,777,216 (16M) colors in TFT mode

- Panel Supports
  - Support single STN panel and dual STN panel with 1, 2, 4, 8 data output pins
  - Support 16-bit parallel TFT panel
  - Support 18-bit parallel TFT panel
  - Support 24-bit serial TFT panel with 8 data output pins
  - Support 24-bit parallel TFT panel
  - Support Delta RGB panel

- OSD Supports
  - Supports one single color background
  - Supports two foregrounds, and every size can be set for each foreground
  - Supports one transparency for the whole graphic
  - Supports one transparency for each pixel in one graphic
  - Supports color key and mask color key

- Decompressor

- Support bpp16 compressed data
- Support bpp24 compressed data with alpha
- Support bpp24 compressed data without alpha

1.2 Pin Description

Table 1-1 LCD Controller Pins Description

Name	I/O	Description
Lcd_pclk	Input/Output	Display device pixel clock
Lcd_vsync	Input/Output	Display device vertical synchronize pulse
Lcd_hsync	Input/Output	Display device horizontal synchronize pulse
Lcd_de	Output	Display device is STN: AC BIAS Pin Display device is NOT STN: data enable Pin
Lcd_d[17:0]	Output	Display device data pins
lcd_lo6_o[5:0]	Output	Display device data pins use in 24 bit parallel mode.
Lcd_spl*[1]	Output	Programmable special pin for generating control signals
Lcd_cls*[1]	Output	Programmable special pin for generating control signals
Lcd_ps*[1]	Output	Programmable special pin for generating control signals
Lcd_rev*[1]	Output	Programmable special pin for generating control signals

NOTE: The mode and timing of special pin Lcd_spl, Lcd_cls, Lcd_ps and Lcd_rev can be seen in **part 1.7 LCD Controller Pin Mapping.**

1.3 Block Diagram

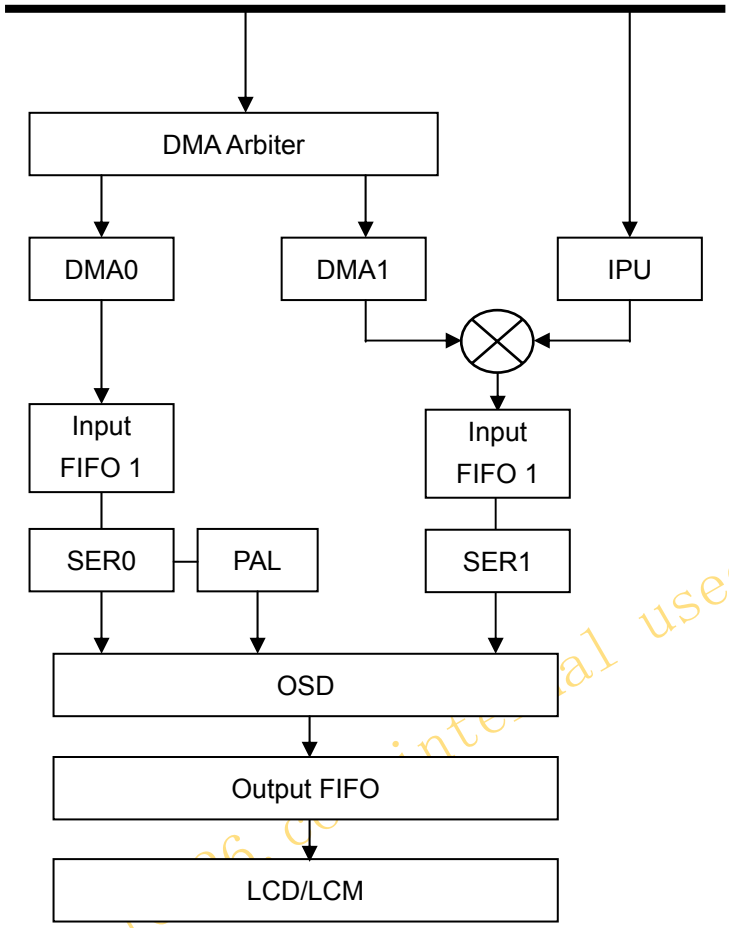


Figure 1-1 Block Diagram when use OSD mode

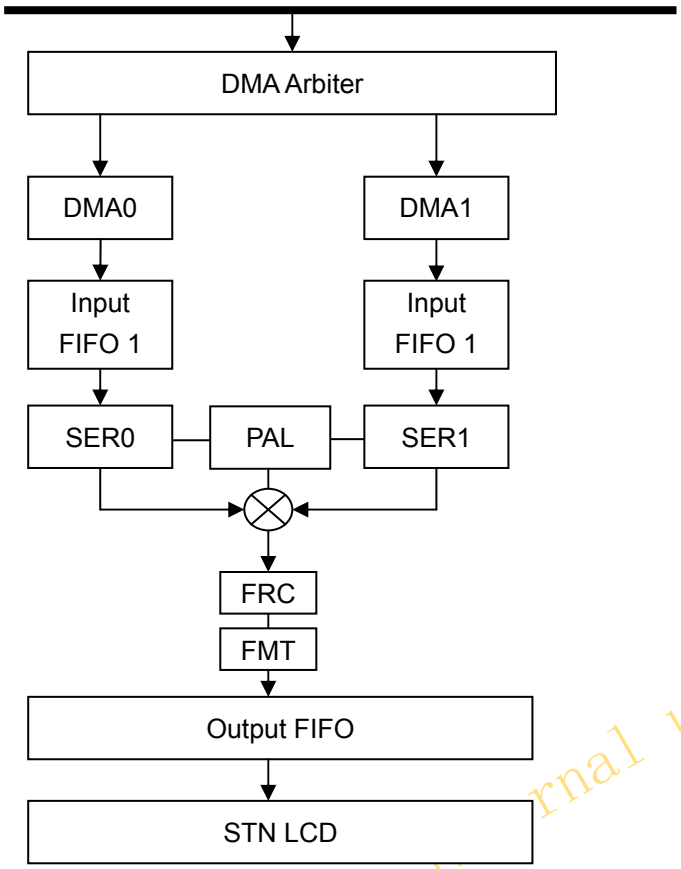


Figure 1-2 Block Diagram of STN mode (not use OSD)

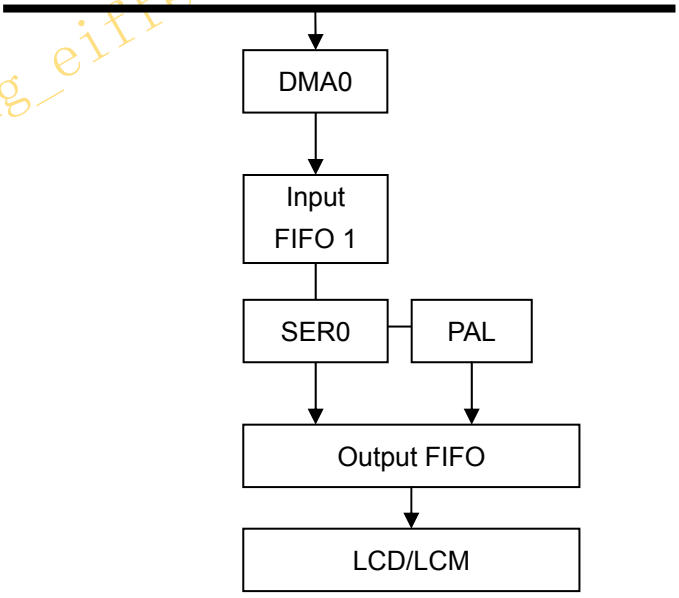


Figure 1-3 Block Diagram of TFT mode (not use OSD)

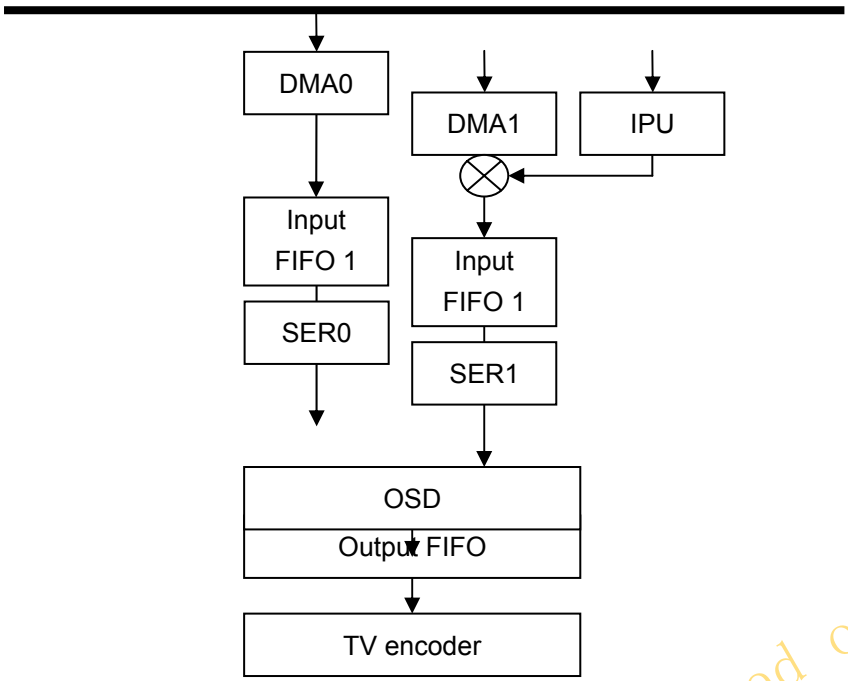


Figure 1-4 Block Diagram of TV interface

1.4 LCD Display Timing

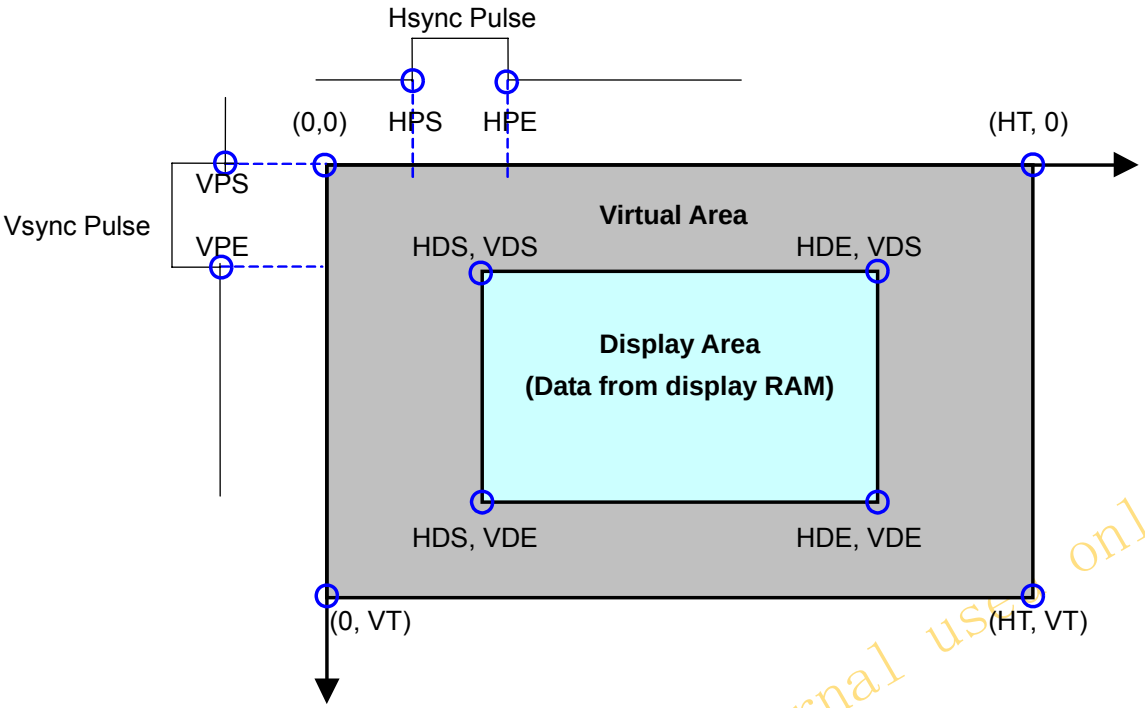


Figure 1-5 Display Parameters

NOTES:

1 VPS === 0
VSYNC pulse always start at point (0,0)

2 H: Horizontal V: Vertical T: Total
D: Display Area P: Pulse
S: Start point E: End point

In the (H, V) Coordinates:

1 The gray rectangle (0, 0) to (HT, VT) is “Virtual Area”.
2 The blue rectangle (HDS, VDS) to (HDE, VDE) is “Display Area”.
3 VPS, VPE defines the VSYNC signal timing. (VPS always be zero)
4 HPS, HPE defines the HSYNC signal timing.

All timing parameters start with “H” is measured in lcd_pclk ticks.
All timing parameters start with “V” is measured in lcd_hsync ticks.

This diagram describes the general LCD panel parameters, these can be set via the registers that describes in next section.

1.5 TV Encoder Timing

Some of Video Encoders for TV (Tele Vision) require interlaced timing interface.

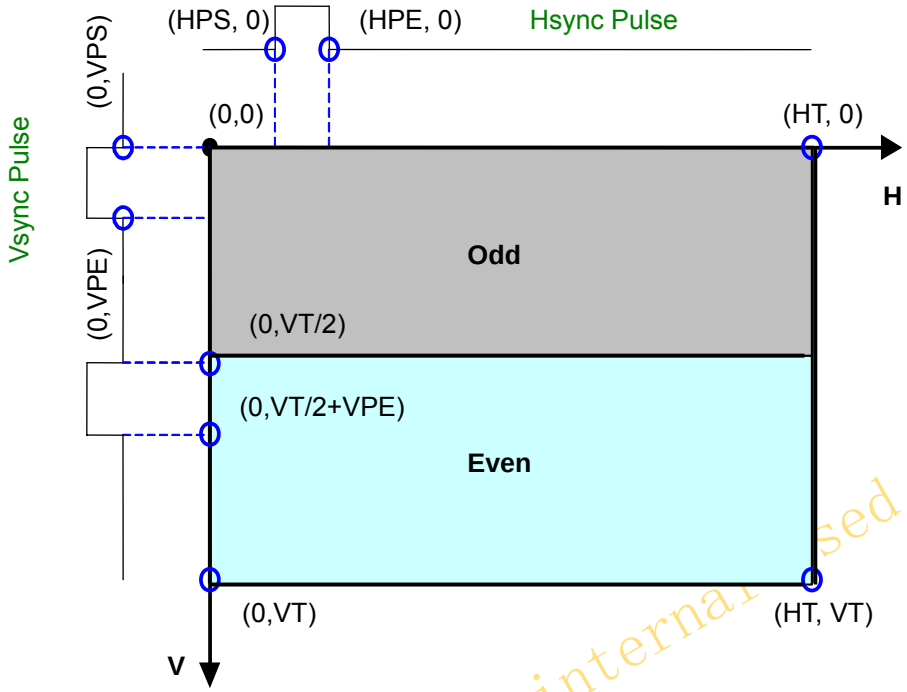


Figure 1-6 TV-Encoder Display Parameters

NOTES:

1 Even Field contains one more blank line.
e.g. For standard PAL timing, Odd filed has 312 lines while even field has 313 lines.

2 Interlace mode generate 2 vsync pulse for each field. The second vsync start at (VT/2), end at (VT/2 + VPE).

3 Display Area & Virtual Area has the same size. VDS=HDS=0, VDE=VT, HDE=HT.

1.6 OSD Graphic

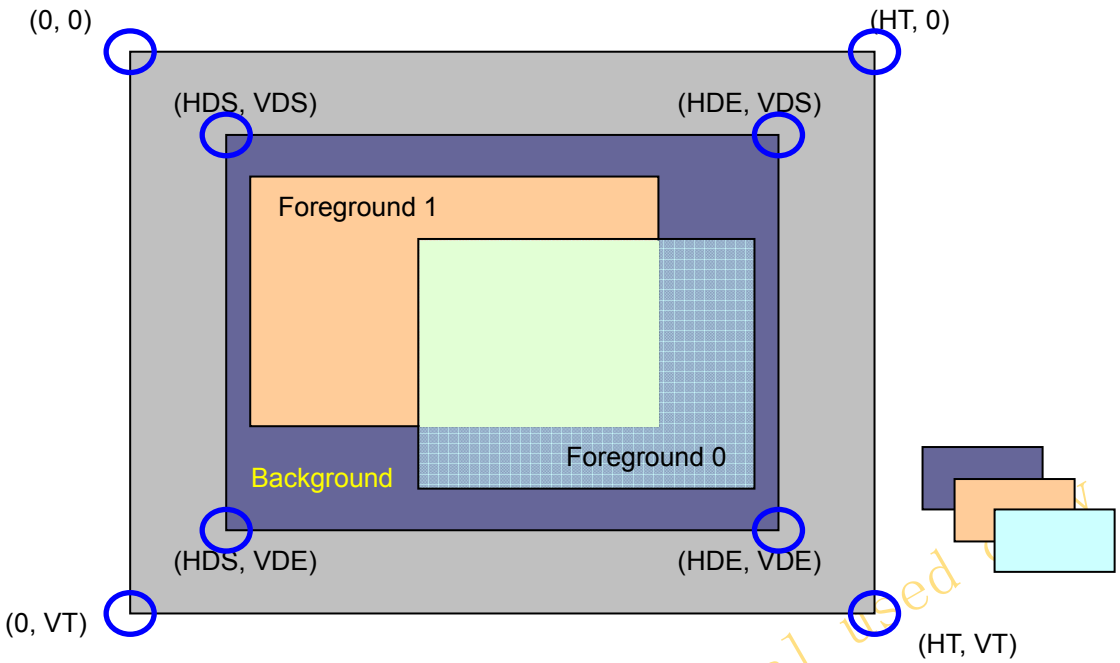


Figure 1-7 OSD Graphic

NOTES:

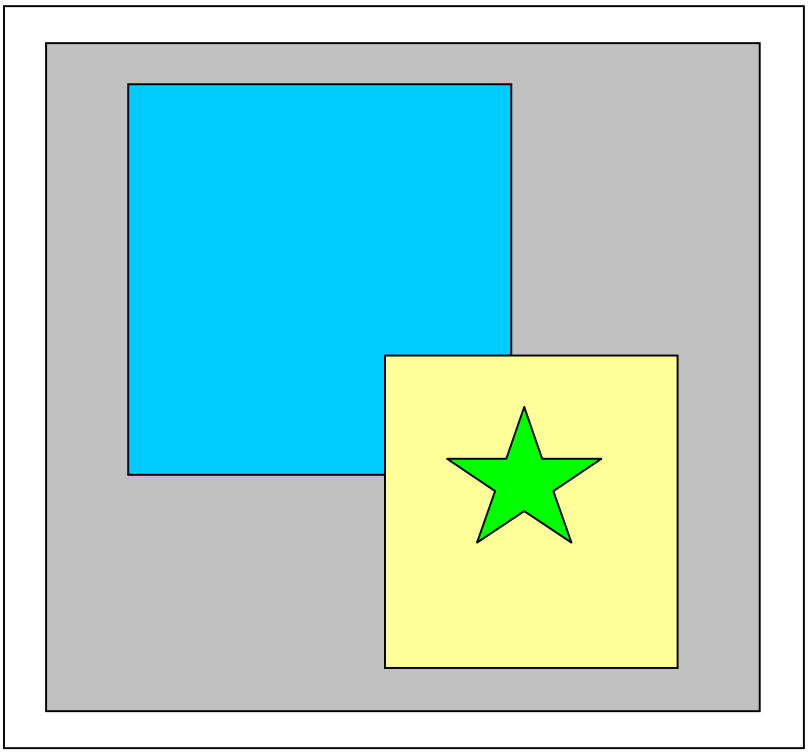
1 Background is one single color and the size is the full screen.
2 The size of foregrounds can be every size smaller than background.
3 The order of the graphic is as follows:
   a Top layer: Foreground 0.
   b Middle layer: Foreground 1.
   c Bottom layer: Background.

1.6.1 Color Key

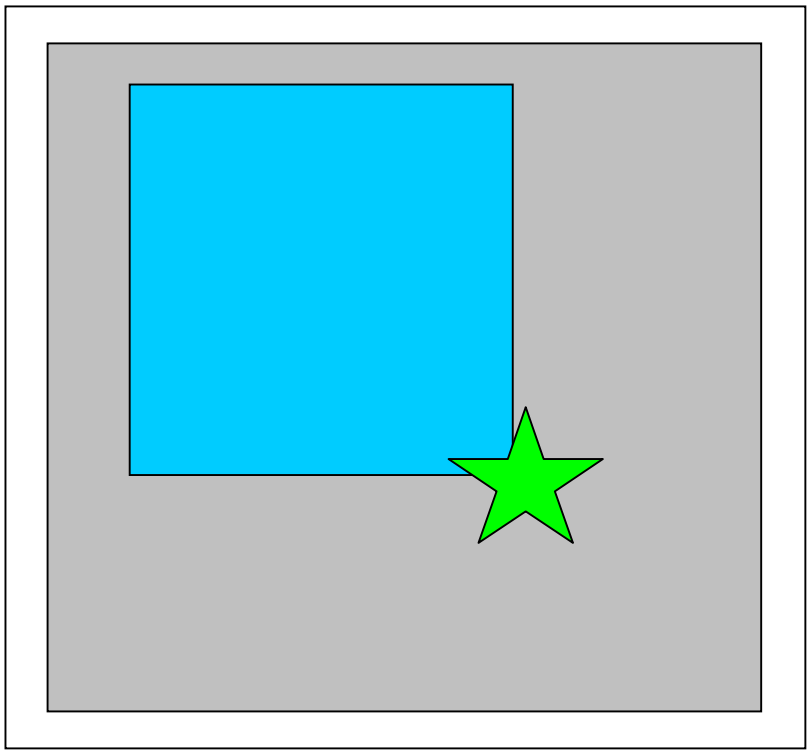
This function gives user a method to implement irregular display window. User can make foreground 0 and foreground 1 to different shape. The color key has two implements mode that called color key and mask color key.

Color Key mode is meant to mask a chosen color and show others.

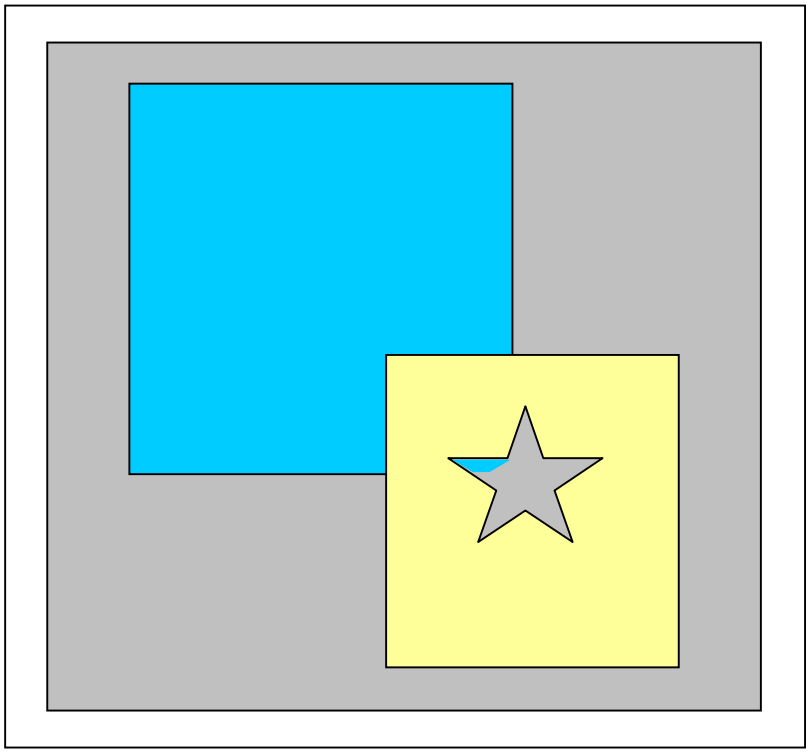
Mask Color Key mode is meant to only show a chosen color and mask others.



Not use color key function

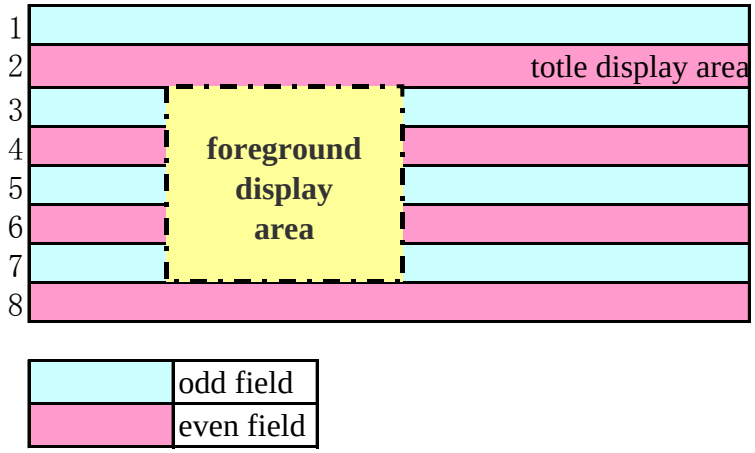


Color key mode

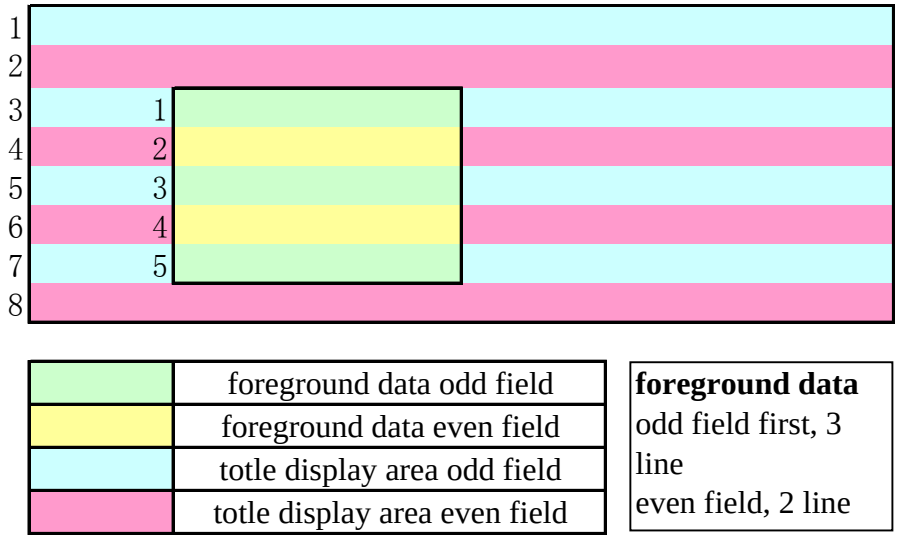


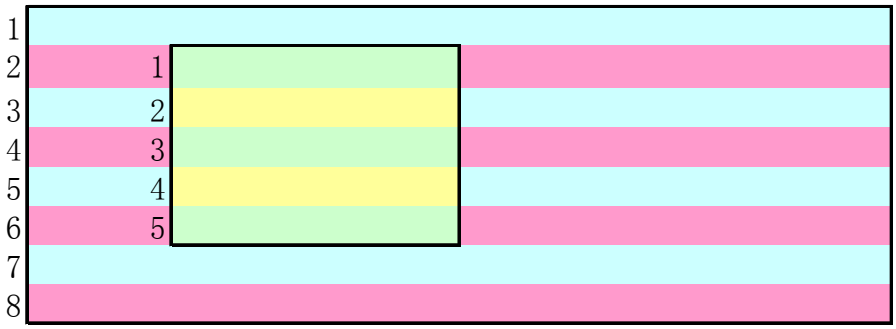
Mask color key mode

1.7 TV Graphic



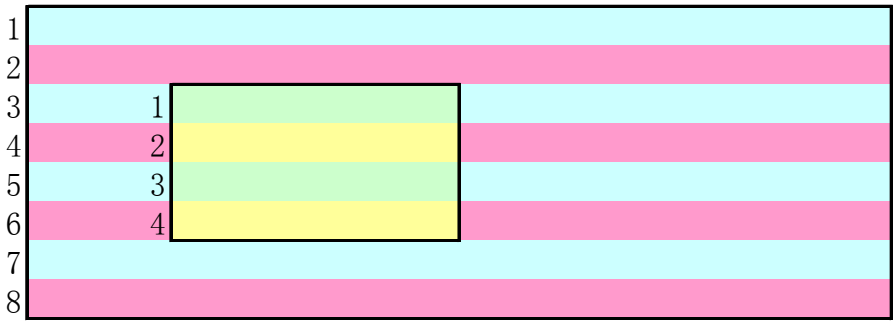
1.7.1 Different Display Field





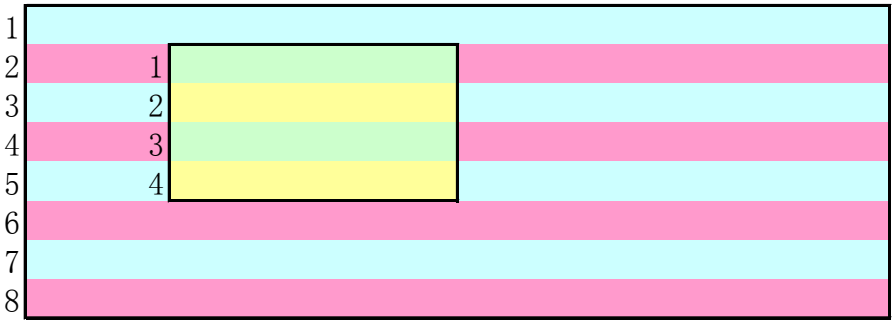
	foreground data odd field
	foreground data even field
	totle display area odd field
	totle display area even field

foreground data
even field first, 2 line
odd field, 3 line



	foreground data odd field
	foreground data even field
	totle display area odd field
	totle display area even field

foreground data
odd field first, 2 line
even field, 2 line



	foreground data odd field
	foreground data even field
	totle display area odd field
	totle display area even field

foreground data
even field first, 2 line
odd field, 2 line

1.8 Register Description

Table 1-2 LCD Controller Registers Description

Name	RW	Reset Value	Address	Access Size
LCDCFG	RW	0x00000000	0x13050000	32
LCDCTRL	RW	0x00000000	0x13050030	32
LCDSTATE	RW	0x00000000	0x13050034	32
LCDOSDC	RW	0x0000	0x13050100	16
LCDOSDCTRL	RW	0x0000	0x13050104	16
LCDOSDS	RW	0x0000	0x13050108	16
LCDBGC	RW	0x00000000	0x1305010C	32
LCDKEY0	RW	0x00000000	0x13050110	32
LCDKEY1	RW	0x00000000	0x13050114	32
LCDALPHA	RW	0x00	0x13050118	8
LCDIPUR	RW	0x00000000	0x1305011C	32
LCDRGBC	RW	0x0000	0x13050090	16
LCDVAT	RW	0x00000000	0x1305000C	32
LCDDAH	RW	0x00000000	0x13050010	32
LCDDAV	RW	0x00000000	0x13050014	32
LCDXYP0	RW	0x00000000	0x13050120	32
LCDXYP0_PART2	RW	0x00000000	0x130501F0	32
LCDXYP1	RW	0x00000000	0x13050124	32
LCDSIZE0	RW	0x00000000	0x13050128	32
LCDSIZE0_PART2	RW	0x00000000	0x130501F4	32
LCDSIZE1	RW	0x00000000	0x1305012C	32
LCDVSYNC	RW	0x00000000	0x13050004	32
LCDHSYNC	RW	0x00000000	0x13050008	32
LCDPS[*1]	RW	0x00000000	0x13050018	32
LCDCLS[*1]	RW	0x00000000	0x1305001C	32
LCDSPL[*1]	RW	0x00000000	0x13050020	32
LCDREV[*1]	RW	0x00000000	0x13050024	32
LCDIID	R	0x00000000	0x13050038	32
LCDDA0	RW	0x00000000	0x13050040	32
LCDSA0	R	0x00000000	0x13050044	32
LCDFID0	R	0x00000000	0x13050048	32
LCDCMD0	R	0x00000000	0x1305004C	32
LCDOFFS0	R	0x00000000	0x13050060	32
LCDPW0	R	0x00000000	0x13050064	32
LCDCNUM0	R	0x00000000	0x13050068	32
LCDDESSIZE0	R	0x00000000	0x1305006C	32

LCDDA1*2	RW	0x00000000	0x13050050	32
LCDSA1*2	R	0x00000000	0x13050054	32
LCDFID1*2	R	0x00000000	0x13050058	32
LCDCMD1*2	R	0x00000000	0x1305005C	32
LCDOFFS1*2	R	0x00000000	0x13050070	32
LCDPW1*2	R	0x00000000	0x13050074	32
LCDCNUM1*2	R	0x00000000	0x13050078	32
LCDDESSIZE1*2	R	0x00000000	0x1305007C	32
LCDDA0_PART2	RW	0x00000000	0x130501C0	32
LCDSA0_PART2	R	0x00000000	0x130501C4	32
LCDFID0_PART2	R	0x00000000	0x130501C8	32
LCDCMD0_PART2	R	0x00000000	0x130501CC	32
LCDOFFS0_PART2	R	0x00000000	0x130501E0	32
LCDPW_PART2	R	0x00000000	0x130501E4	32
LCDCNUM0_PART2	R	0x00000000	0x130501E8	32
LCDDESSIZE0_PART2	R	0x00000000	0x130501EC	32
LCDPCFG	RW	0x00000000	0x130502C0	32

NOTES:

1 *1: These registers are only used for SPECIAL TFT panels.

2 *2: These registers are only used for Dual Panel STN panels and use DMA channel 1 in OSD mode for TFT panels.

1.8.1 Configure Register (LCDCFG)

LCDCFG **0x13050000**

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
	LCDPIN	TVEPEH		NEWDES	PALBP	TVEN	RECOVER	DITHER	PSM	CLSM	SPLM	REVM	HSYNM	PCLKM	INVDAT	SYNDIR	PSP	CLSP	SPLP	REVP	HSP	PCP	DEP	VSP	18/16	24	PDW	MODE				
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	LCDPIN*1	LCD PIN Select bit. It is used to choose the function of LCD PINS or SLCD PINS. The function of pins is as follows. **LCDPIN** / **PIN SELECT** 0 / LCD PIN 1 / SLCD PIN	RW
30	TVEPEH	TVE PAL enable extra_halfline signal.	RW

29		**KEEP THIS BIT TO 0.**	RW
28	NEWDES	indicate use new 8 words descriptor or not. 0: use old 4 words descriptor 1: use new 8 words descriptor (add LCDOFFSx, LCDPWx, LCDCUNMx, LCDDESSIZEx) OSD mode use 8 word descriptor.	RW
27	PALBP	Indicate bypass pal in BPP8, and in OSD mode, set this bit to 1 is also bypass data format and alpha blending. 0: use PAL; 1: not use PAL.	RW
26	TVEN	Indicate the terminal is LCD panel or TV.	RW
25	RECOVER	Auto recover when output FIFO under run. 0: disable; 1: enable.	RW
24	DITHER	Dither function. (use when 24bpp data output to a 18/16bit panel) 0: disable; 1: enable. Dither function use to make the picture misty, when you show a static picture with few color, strongly recommend you not use it. When you use this function both static and dynamic picture, strongly recommend you to set the static picture with 16/18BPP color.	RW
23	PSM	PS signal mode bit. 0: enabled; 1: disabled.	RW
22	CLSM	CLS signal mode bit. 0: enabled; 1: disabled.	RW
21	SPLM	SPL signal mode bit. 0: enabled; 1: disabled.	RW
20	REVM	REV signal mode bit. 0: enabled; 1: disabled.	RW
19	HSYNM	H-Sync signal polarity choice function. 0: enabled; 1: disabled.	RW
18	PCLKM	Dot clock signal polarity choice function. 0: enabled; 1: disabled.	RW
17	INVDAT	Inverse output data. 0: normal; 1: inverse.	RW
16	SYNDIR	V-Sync and H-Sync direction. 0: output; 1: input.	RW
15	PSP	PS pin reset state.	RW
14	CLSP	CLS pin reset state.	RW
13	SPLP	SPL pin reset state.	RW
12	REVP	REV pin reset state.	RW
11	HSP	H-Sync polarity. 0: active high; 1: active low.	RW
10	PCP	Pix-clock polarity. 0: data translations at rising edge 1: data translations at falling edge	RW
9	DEP	Data Enable polarity. 0: active high; 1: active low.	RW
8	VSP	V-Sync polarity. 0: leading edge is rising edge 1: leading edge is falling edge	RW
7	18/16	18-bit TFT Panel or 16-bit TFT Panel. This bit will be available when MODE [3:2] is equal to 0 and 24[6] is equal to 0. 0: 16-bit TFT Panel 1: 18-bit TFT Panel	RW
6	24	Set this bit to 1 for 24-bit TFT Panel.	RW

<table>
<tr><td>5:4</td><td>PDW</td><td>STN pins utilization.
<table>
<tr><th></th><th>Signal Panel</th></tr>
<tr><td>00</td><td>Lcd_d[0]</td></tr>
<tr><td>01</td><td>Lcd_d[0:1]</td></tr>
<tr><td>10</td><td>Lcd_d[0:3]</td></tr>
<tr><td>11</td><td>Lcd_d[0:7]</td></tr>
<tr><th></th><th>Dual-Monochrome Panel</th></tr>
<tr><td>00</td><td>Reserved</td></tr>
<tr><td>01</td><td>Reserved</td></tr>
<tr><td>10</td><td>Upper panel: lcd_d[3:0], lower panel: lcd_d[11:8]</td></tr>
<tr><td>11</td><td>Upper panel: lcd_d[7:0], lower panel: lcd_d[15:8]</td></tr>
</table></td><td>RW</td></tr>
<tr><td>3:0</td><td>MODE</td><td>Display Device Mode Select/Output mode.
<table>
<tr><th></th><th>LCD Panel</th></tr>
<tr><td>0000</td><td>Generic 16-bit/18-bit Parallel TFT Panel</td></tr>
<tr><td>0001</td><td>Special TFT Panel Mode1</td></tr>
<tr><td>0010</td><td>Special TFT Panel Mode2</td></tr>
<tr><td>0011</td><td>Special TFT Panel Mode3</td></tr>
<tr><td>0100</td><td>Non-Interlaced TV out</td></tr>
<tr><td>0101</td><td>Reserved</td></tr>
<tr><td>0110</td><td>Interlaced TV out</td></tr>
<tr><td>0111</td><td>Reserved</td></tr>
<tr><td>1000</td><td>Single-Color STN Panel</td></tr>
<tr><td>1001</td><td>Single-Monochrome STN Panel</td></tr>
<tr><td>1010</td><td>Dual-Color STN Panel</td></tr>
<tr><td>1011</td><td>Dual-Monochrome STN Panel</td></tr>
<tr><td>1100</td><td>8-bit Serial TFT</td></tr>
<tr><td>1101</td><td>LCM</td></tr>
<tr><td>1110</td><td>Reserved</td></tr>
<tr><td>1111</td><td>Reserved</td></tr>
</table></td><td>RW</td></tr>
</table>

NOTES:

*1.

LCDPIN	PIN25	PIN24	PIN23	PIN22	PIN21	PIN20	PIN19	PIN18	PIN17-0
0	LCD PCLK	LCD VSYNC	LCD HSYNC	LCD DE	LCD REV	LCD PS	LCD CLS	LCD SPL	LCD D [17:0]
1	SLCD CLK	SLCD CS	SLCD RS	--	--	--	--	--	SLCD D [17:0]

1 The direction of PIN25 is set by register LPCDR.LCS in CPM SPEC.

2 The direction of PIN23 and PIN23 are set by register LCDCFG.SYNDIR.

1.8.2 Control Register (LCDCTRL)

LCDCTRL 0x13050030

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
	PINMD	BST			RGB	OFUP	FRC		PDD									DACTE	EOFM	SOFM	OFUM	IFUM0	IFUM1	LDDM	QDM	BEDN	PEDN	DIS	ENA	BPP		
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	PINMD	This register set Pin distribution in 16-bit parallel mode. 0: 16-bit data correspond with LCD_D[15:0] 1: 16-bit data correspond with LCD_D[17:10], LCD_D[8:1]	RW
30:28	BST	Burst Length Selection. 000: 4 word 001: 8 word 010: 16 word 011: 32 word 101: Contiue16 100: 64 word	RW
27	RGB	Bpp16 RGB mode. 0: RGB565; 1: RGB555. In OSD mode, this bit configure the foreground 0. If use parallel 18 bit，set this bit to 0.	RW
26	OFUP	Output FIFO under run protection. 0: disable; 1: enable.	RW
25:24	FRC	STN FRC Algorithm Selection. 00: 16 grayscale 01: 4 grayscale 10: 2 grayscale 11: Reserved	RW
23:16	PDD	Load Palette Delay Counter.	RW
15		keep this bit to 0.	
14	DACTE	DAC loop back test.	RW
13	EOFM	Mask end of frame interrupt. 0: INT-disabled; 1: INT-enabled.	RW
12	SOFM	Mask start of frame interrupt. 0: INT-disabled; 1: INT-enabled.	RW
11	OFUM	Mask out FIFO under run interrupt. 0: INT-disabled;1: INT-enabled.	RW
10	IFUM0	Mask in FIFO 0 under run interrupt. 0: INT-disabled; 1: INT-enabled.	RW
9	IFUM1	Mask in FIFO 1 under run interrupt. 0: INT-disabled; 1: INT-enabled.	RW
8	LDDM	Mask LCD disable done interrupt. 0: INT-disabled; 1: INT-enabled.	RW
7	QDM	Mask LCD quick disable done interrupt. 0: INT-disabled; 1: INT-enabled.	RW

6	BEDN	Endian selection. 0: same as system Endian; 1: reverse endian format.	RW
5	PEDN	Endian in byte. 0: msb first; 1: lsb first.	RW
4	DIS	Disable controller indicate bit. 0: enable; 1: in disabling or disabled.	RW
3	ENA	Enable controller. 0: disable; 1: enable.	W
2:0	BPP	Bits Per Pixel. 000: 1bpp 001: 2bpp 010: 4bpp 011: 8bpp 100: 15/16bpp 101: 18bpp/24bpp 110: 24bpp compressed 111: 30bpp In OSD mode, those bits configure the foreground 0.	RW

1.8.3 Status Register (LCDSTATE)

LCDSTATE 0x13050034

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																								QD	Reserved	EOF	SOF	OUF	IFU0	IFU1	LDD
RST	0	1	1	1	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
31:8	Reserved	Writing has no effect, read as zero.	R
7	QD	LCD Quick disable. 0: not been quick disabled; 1: quick disabled done.	RW
6	Reserved	Writing has no effect, read as zero.	R
5	EOF	End of Frame indicate bit.	RW
4	SOF	Start of Frame indicate bit.	RW
3	OUF	Out FIFO under run.	RW
2	IFU0	In FIFO 0 under run.	RW
1	IFU1	In FIFO 1 under run.	RW
0	LDD	LCD disable. 0: not been normal disabled; 1: been normal disabled.	RW

1.8.4 OSD Configure Register (LCDOSDC)

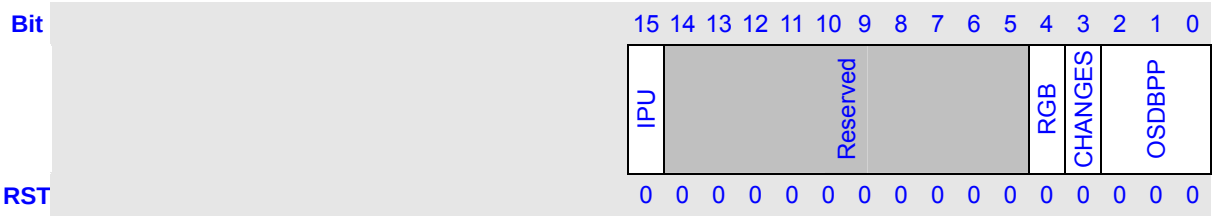
LCDOSDC **0x13050100**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SOFM1	EOFM1	Reserved	OSDIV	SOFM0	EOFM0	Reserved					F1EN	F0EN	ALPHAEN	ALPHAMD	OSDEN
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15	SOFM1	Start of frame interrupt mask for foreground 1.	RW
14	EOFM1	End of frame interrupt mask for foreground 1.	RW
13	Reserved	Writing has no effect, read as zero.	R
12	OSDIV	Not supported in this release.	RW
11	SOFM0	Start of frame interrupt mask for foreground 0.	RW
10	EOFM0	End of frame interrupt mask for foreground 0.	RW
9:5	Reserved	Writing has no effect, read as zero.	R
4	F1EN	1: Foreground 1 is enabled 0: Foreground 1 is disabled	RW
3	F0EN	1: Foreground 0 is enabled 0: Foreground 0 is disabled. *When use slcd, F0EN must set 1.	RW
2	ALPHAEN	1: Alpha blending is enabled 0: Alpha blending is disabled	RW
1	ALPHAMD	Alpha blending mode. 0: One transparency for the whole graphic, and the LCDALPHA register is used for transparency 1: One transparency for each pixel in one graphic, and the alpha value is coming from each pixel data	RW
0	OSDEN	OSD mod enable. 1: enabled. And you can use F0 F1 0: disabled	RW

1.8.5 OSD Control Register (LCDOSDCTRL)

LCDOSDCTRL **0x13050104**



Bits	Name	Description	RW
15	IPU	Indicate use IPU or DMA channel 1 to transport data to FIFO 1. This bit is only use in OSD mode. 0: use DMA channel 1 1: use IPU	RW
13:5	Reserved	Writing has no effect, read as zero.	R
4	OSDRGB	Bpp16 RGB mode. 0: RGB565; 1: RGB555. This bit only use in OSD mode to configure foreground 1.	RW
3	CHANGES	Change configure flag, when software need change the foreground0 and foreground1's enable/position/size, it need set this bit to 1. When hardware finishes the change, It will clear this bit to 0. DO NOT set this bit when you needed change size or position. AND make sure the reconfigure value is different to the old one. Only one of these (F0's position, F1's position, F0's size, F1's size) could be change in one time. Refer to 1.8.6.	RW
2:0	OSDBPP	Bits Per Pixel of OSD channel 1.(this channel cannot use palette) 000 Reserved 001 Reserved 010 Reserved 011 Reserved 100 15/16bpp 101 18bpp/24bpp 110 24bpp compressed 111 30bpp Those bits only use in OSD mode to configure display window 1.	RW

1.8.6 OSD State Register (LCDOSDS)

LCDOSDS																0x13050108
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	SOF1	EOF1	Reserved		SOF0	EOF0	Reserved									READY
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15	SOF1	Start of frame flag for foreground 1.	RW
14	EOF1	End of frame flag for foreground 1.	RW
13:12	Reserved	Writing has no effect, read as zero.	R
11	SOF0	Start of frame flag for foreground 0.	RW
10	EOF0	End of frame flag for foreground 0.	RW
9:1	Reserved	Writing has no effect, read as zero.	R
0	READY	Ready for accept the change. When this bit set 1, the software can change the descriptor's LCDDESSIZE0, 1 to change the foreground size. This bit will clear by hardware when the change is finished.	R

1.8.7 Background Color Register (LCDBGC)

LCDBGC																												0x1305010C
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								Red [7:0]								Green [7:0]								Blue [7:0]							
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
31:27	Reserved	Writing has no effect, read as zero.	R
23:16	Red	Red part or Y part of background.	RW
15:8	Green	Green part or Cb part of background.	RW
7:0	Blue	Blue part or Cr part of background.	RW

1.8.8 Foreground Color Key Register 0 (LCDKEY0)

LCDKEY0 **0x13050110**

Bit	31	30	29:27	23:16	15:8	7:0
	KEYEN	KEYMD	Reserved	Red [7:0]	Green [7:0]	Blue [7:0]
RST	0	0	0	0	0	0

Bits	Name	Description	RW
31	KEYEN	The enable bit of color key for foreground 0.	RW
30	KEYMD	Color key mod. 0: color key; 1: mask color key.	RW
29:27	Reserved	Writing has no effect, read as zero.	R
23:16	Red	Red part of color key for foreground 0.	RW
15:8	Green	Green part of color key for foreground 0.	RW
7:0	Blue	Blue part of color key for foreground 0.	RW

1.8.9 Foreground Color Key Register 1 (LCDKEY1)

LCDKEY1 **0x13050114**

Bit	31	30	29:27	23:16	15:8	7:0
	KEYEN	KEYMD	Reserved	Red [7:0]	Green [7:0]	Blue [7:0]
RST	0	0	0	0	0	0

Bits	Name	Description	RW
31	KEYEN	The enable bit of color key for foreground 1.	RW
30	KEYMD	Color key mod. 0: color key; 1: mask color key.	RW
29:27	Reserved	Writing has no effect, read as zero.	R
23:16	Red	Red part of color key for foreground 1.	RW
15:8	Green	Green part of color key for foreground 1.	RW
7:0	Blue	Blue part of color key for foreground 1.	RW

1.8.10 ALPHA Register (LCDALPHA)

LCDALPHA **0x13050118**

Bit	7	6	5	4	3	2	1	0
	ALPHA [7:0]							
RST	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
7:0	ALPHA	The alpha value for one graphic with one transparency.	RW

The formula of alpha blending is as follows:

$$NewPixel = \frac{[(256 - Alpha) * (Foreground1\_or\_background) + Alpha * Froeground0 + 128]}{256}$$

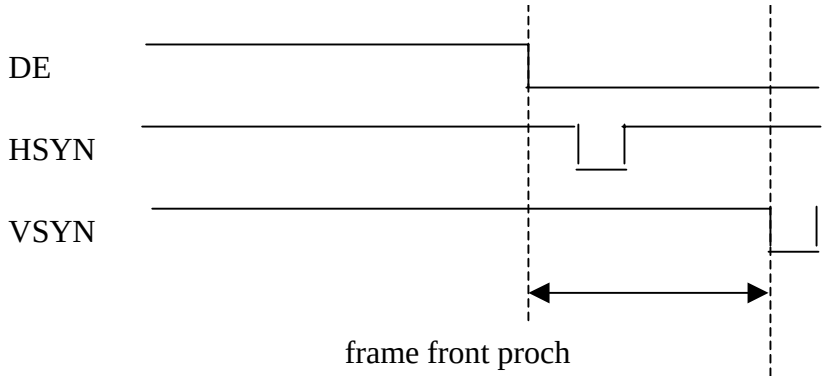
Note that foreground 1 must be overlay background.

1.8.11 IPU Restart (LCDIPUR)

LCDIPUR **0x1305011C**

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
	IPUREN	Reserved							IPUR																							
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	IPUREN	IPU restart function enable. 0:disable; 1:enable.	RW
30:24	Reserved	Writing has no effect, read as zero.	R
23:0	IPUR	This register is indicating when one frame is end, how long the panel can wait for the next frame data from IPU. In common, set this number larger than frame front porch and near to ((HT-0) X (VPE-VPS))/3. This signal only use when foreground1 work in IPU mode. Trigger IPU transfer the last frame again to avoid output FIFO under run.	RW



1.8.12 RGB Control (LCDRGBC)

LCDRGBC **0x13050090**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RGBDM	DMM	Reserved					YCC	Reserved	OddRGB			Reserved	EvenRGB		
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15	RGBDM	RGB with dummy data enable. Only useful for RGB serial mode. If this bit set to 1, the one pixel include 4 clock periods, that Red, Green, Blue and Dummy data. Dummy is equal to 0. 0: Disable; 1: Enable.	RW
14	DMM	RGB dummy mode. 0: R-G-B-Dummy 1: Dummy-R-G-B	
13:9	Reserved	Writing has no effect, read as zero.	R
8	YCC	Change RGB to YCbCr. 0: not change; 1: change to YUV. This bit only use in OSD mode. Change RGB data to YCbYCr and sent to TV encoder. Please notice that the data will be translated as 16 bits parallel. And only half of it will be transfer. (YCb or YCr in one pixel). If you not use OSD mode and TV encoder, please set this bit to 0. When use this function with IPU transfer data to an interlaced TV, please set IPU output as RGB 888, and OSDBPP to 24. or IPU output data as PACKAGE(YCbYCr) and OSDBPP to 16.	RW
7	Reserved	Writing has no effect, read as zero.	RW
6:4	OddRGB	Odd line serial RGB data arrangement, useful for RGB serial mode	RW

<table>
<tr><td></td><td></td><td>only. *Please notice that you must set 000 when use 16/18parallel mode.<table><tr><th></th><th>RGB mode</th></tr><tr><td>000</td><td>RGB</td></tr><tr><td>001</td><td>RBG</td></tr><tr><td>010</td><td>GRB</td></tr><tr><td>011</td><td>GBR</td></tr><tr><td>100</td><td>BRG</td></tr><tr><td>101</td><td>BGR</td></tr><tr><td>110</td><td>Reserved</td></tr><tr><td>111</td><td>Reserved</td></tr></table></td><td></td></tr>
<tr><td>3</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>2:0</td><td>EvenRGB</td><td>Even line serial RGB data arrangement, useful for RGB serial mode only. *Please notice that you must set 000 when use 16/18parallel mode.<table><tr><th></th><th>RGB mode</th></tr><tr><td>000</td><td>RGB</td></tr><tr><td>001</td><td>RBG</td></tr><tr><td>010</td><td>GRB</td></tr><tr><td>011</td><td>GBR</td></tr><tr><td>100</td><td>BRG</td></tr><tr><td>101</td><td>BGR</td></tr><tr><td>110</td><td>Reserved</td></tr><tr><td>111</td><td>Reserved</td></tr></table></td><td>RW</td></tr>
</table>

1.8.13 Virtual Area Setting (LCDVAT)

LCDVAT **0x1305000C**

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				HT												Reserved				VT											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	HT	Horizontal Total size. (in dot clock, sum of display area and blank space)	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	VT	Vertical Total size. (in line clock, sum of display area and blank space)	RW

1.8.14 Display Area Horizontal Start/End Point (LCDDAH)

LCDDAH **0x13050010**

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				HDS												Reserved				HDE											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	HDS	Horizontal display area start. (in dot clock)	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	HDE	Horizontal display area end. (in dot clock)	RW

1.8.15 Display Area Vertical Start/End Point (LCDDAV)

LCDDAV **0x13050014**

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				VDS												Reserved				VDE											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	VDS	Vertical display area start position. (in line clock)	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	VDE	Vertical display area end position. (in line clock)	RW

1.8.16 Foreground 0 XY Position Register (LCDXYP0)

LCDXYP0 **0x13050120**

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				YPOS												Reserved				XPOS											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	YPOS	The Y position of top-left part for foreground 0.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	XPOS	The X position of top-left part for foreground 0.	RW

1.8.17 Foreground 0 PART2 XY Position Register (LCDXYP0_PART2)

LCDXYP0_PART2 **0x130501F0**

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				YPOS												Reserved				XPOS											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	YPOS	The Y position of top-left part for foreground 0 PART2.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	XPOS	The X position of top-left part for foreground 0 PART2.	RW

1.8.18 Foreground 1 XY Position Register (LCDXYP1)

LCDXYP1 **0x13050124**

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				YPOS												Reserved				XPOS											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	YPOS	The Y position of top-left part for foreground 1.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	XPOS	The X position of top-left part for foreground 1.	RW

1.8.19 Foreground 0 Size Register (LCDSIZE0)

LCDSIZE0 **0x13050128**

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				Height												Reserved				Width											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	Height	The height of foreground 0.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	Width	The width of foreground 0.	RW

When use TVE interlaced mode, please set the area of F0 and F1 aligned with BST.

1.8.20 Foreground 0 PART2 Size Register (LCDSIZE0_PART2)

LCDSIZE0_PART2 **0x130501F4**

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				Height												Reserved				Width											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	Height	The height of foreground 0 PART2.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	Width	The width of foreground 0 PART2.	RW

1.8.21 Foreground 1 Size Register (LCDSIZE1)

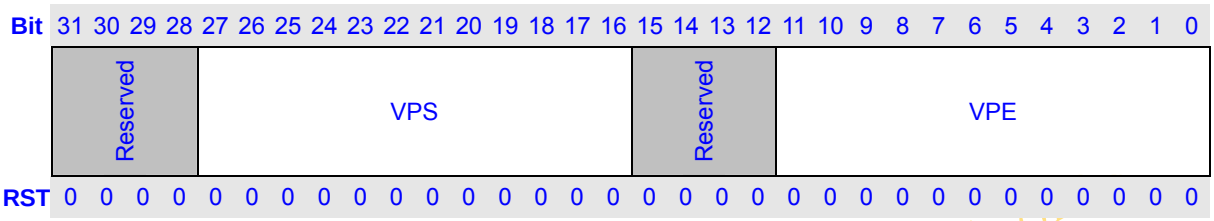
LCDSIZE1 **0x1305012c**

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				Height												Reserved				Width											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	Height	The height of foreground 1.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	Width	The width of foreground 1.	RW

1.8.22 Vertical Synchronize Register (LCDVSYNC)

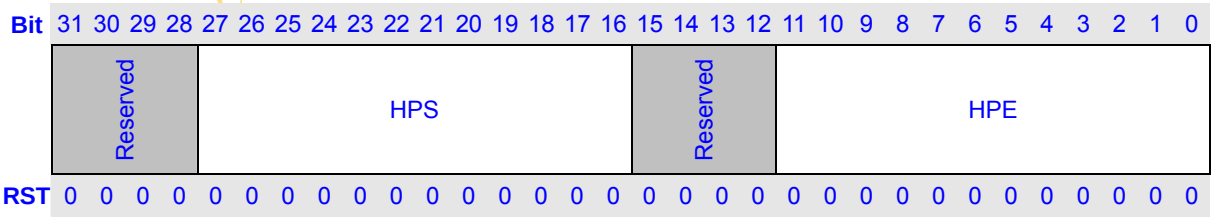
LCDVSYNC **0x13050004**



Bits	Name	Description	RW
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	VPS	V-Sync Pulse start position, fixed to 0. (in line clock)	R
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	VPE	V-Sync Pulse end position. (in line clock)	RW

1.8.23 Horizontal Synchronize Register (LCDHSYNC)

LCDHSYNC **0x13050008**



Bits	Name	Description	RW
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	HPS	H-Sync pulse start position. (in dot clock)	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	HPE	H-Sync pulse end position. (in dot clock)	RW

1.8.24 PS Signal Setting (LCDPS)

LCDPS **0x13050018**

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				PSS												Reserved				PSE											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	PSS	PS signal start position. (in dot clock) In STN mode, PS signal is ignored. But this register is used to define the AC BIAS signal. AC BIAS signal will toggle very N lines per frame. PSS defines the Toggle position.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	PSE	PS signal end position. (in dot clock) In STN mode, PSE defines N, which described in PSS.	RW

1.8.25 CLS Signal Setting (LCDCLS)

LCDCLS **0x1305001C**

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				CLSS												Reserved				CLSE											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	CLSS	CLS signal start position. (in dot clock)	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	CLSE	CLS signal end position. (in dot clock)	RW

1.8.26 SPL Signal Setting (LCDSPL)

LCDSPL **0x13050020**

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				SPLS												Reserved				SPLE											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	SPLS	SPL signal start position. (in dot clock)	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	SPLE	SPL signal end position. (in dot clock)	RW

* In test mode this register use to keep TV encoder module's output data: comp_luma([25:16]) and chroma([9:0]).

1.8.27 REV Signal Setting (LCDREV)

LCDREV **0x13050024**

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				REVS												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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	REVS	REV signal start position. (in dot clock)	RW
15:0	Reserved	Writing has no effect, read as zero.	R

1.8.28 Interrupt ID Register (LCDIID)

LCDIID is a read-only register that contains a copy of the Frame ID register (LCDFID) from the descriptor currently being processed when a start of frame (SOF) or end of frame (EOF) interrupt is generated. LCDIID is written to only when an unmasked interrupt of the above type is signaled and there are no other unmasked interrupts in the LCD controller pending. As such, the register is considered to be sticky and will be overwritten only when the signaled interrupt is cleared by writing the LCD controller status register. For dual-panel displays, LCDIID is written only when both channels have reached a given state.

LCDIID is written with the last channel to reach that state. (i.e. LCDFID of the last channel to reach SOF would be written in LCDIID if SOF interrupts are enabled). Reserved bits must be written with zeros and reads from them must be ignored.

LCDIID **0x13050038**

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
	IID																															
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
31:0	IID	A copy of Frame ID register, which transferred from Descriptor.	RW

1.8.29 Descriptor Address Registers (LCDDAx, 0_PART2)

A frame descriptor is a 4-word block, aligned on 4-word (16-byte) boundary, in external memory:

WORD [0] contains the physical address for next LCDDAx.
WORD [1] contains the physical address for LCDSAx.
WORD [2] contains the value for LCDFIDx.
WORD [3] contains the value for LCDCMDx.

Software must write the physical address of the first descriptor to LCDDAx before enabling the LCD Controller. Once the LCD Controller is enabled, the first descriptor is read, and all 4 registers are written by the DMAC. The next frame descriptor pointed to by LCDDAx is loaded into the registers for the associated DMA channel after all data for the current descriptor has been transferred.

NOTE: If only one frame buffer is used in external memory, the LCDDAx field (word [0] of the frame descriptor) must point back to itself. That is to say, the value of LCDDAx is the physical address of itself.

Read/write registers LCDDA0 and LCDDA1, corresponding to DMA channels 0 and 1, contain the physical address of the next descriptor in external memory. The DMAC fetches the descriptor at this location after finishing the current descriptor. On reset, the bits in this register are zero. The target address must be aligned to 16-byte boundary. Bits [3:0] of the address must be zero.

LCDDA0, LCDDA1, LCDDA0_PART2 **0x13050040, 0x13050050, 0x130501C0**

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
	DA0, 1																															
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
31:0	DA0, 1	Next descriptor physical address. And descriptor structure as following: WORD [0]: next descriptor physical address WORD [1]: the buffer physical address WORD [2]: the buffer ID value (Only for debug) WORD [3]: the buffer property. The value is same as LCDCMD	RW

1.8.30 Source Address Registers (LCDSAx, 0_PART2)

Registers LCDSA0 and LCDSA1, corresponding to DMA channels 0 and 1, contain the physical address of frame buffer or palette buffer in external memory. The address must be aligned on a 4, 8, or 16 word boundary according to register LCDCTRL.BST. If this descriptor is for palette data, LCDSA0 points to the memory location of the palette buffer. If this descriptor is for frame data, LCDSAx points to the memory location of the frame buffer. This address is incremented by hardware as the DMAC fetches data from memory. If desired, the Frame ID Register can be used to hold the initial frame source address.

LCDSA0, LCDSA1, LCDSA0_PART2 **0x13050044, 0x13050054, 0x130501C4**

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
	SA0, 1																															
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
31:0	SA0, 1	Buffer start address. (Only for driver debug)	R

1.8.31 Frame ID Registers (LCDFIDx, 0_PART2)

Registers LCDFID0 and LCDFID1, corresponding to DMA channels 0 and 1, contain an ID field that describes the current frame. The particular use of this field is up to the software. This ID register is copied to the LCD Controller Interrupt ID Register when an interrupt occurs.

LCDFID0, LCDFID1, LCDFID0_PART2 **0x13050048, 0x13050058, 0x130501C8**

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
	FID0, 1																															
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
31:0	FID0, 1	Frame ID. (Only for debug)	R

1.8.32 DMA Command Registers (LCDCMDx, 0_PART2)

LCDCMD0, LCDCMD1, LCDCMD0_PART2 **0x1305004C, 0x1305005C, 0x130501CC**

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
	SOFINT	EOFINT	CMD	PAL	Uncomp_en	Uncomp_md	Reserved		LEN																							
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	SOFINT	Enable start of frame interrupt. When SOFINT =1, the DMAC sets the start of frame bit (LCDSTATE.SOF) when starting a new frame. The SOF bit is set after a new descriptor is loaded from memory and before the palette/frame data is fetched. In dual-panel mode, LCDSTATE.SOF is set only when both channels reach the start of frame and both frame descriptors have SOFINT set. SOFINT must not be set for palette descriptors in dual-panel mode, since only one channel is ever used to load the palette descriptor.	R
30	EOFINT	Enable end of frame interrupt. When EOFINT =1, the DMAC sets the end of frame bit (LCDSTATE.EOF) after fetching the last word in the frame buffer. In dual-panel mode, LCDSTATE.EOF is set only when both channels reach the end of frame and both frame descriptors have EOFINT set. EOFINT must not be set for palette descriptors in dual-panel mode, since only one channel is ever used to load the palette descriptor.	R
29	CMD	It is used to distinguish command and data in lcm mode. And it is only loaded via DMA channel 0. 1: The data is command 0: The data is data	R
28	PAL	The descriptor contains a palette buffer. PAL indicates that data being fetched will be loaded into the palette RAM. If PAL =1, the palette RAM data is loaded via DMA channel 0 as follows: In bpp1, 2, 4, 8 mode, software must load the palette at least once after enabling the LCD. In bpp16 mode, PAL must be 0.	R
27	Uncomp_en	It indicate this frm is compressed or not. 0: not compressed; 1: compressed.	R
26	Uncomp_md	It indicate this compressed frm is with alpha o without alpha. 0:with alpha; 1:without alpha.	R
25:24	Reserved	Writing has no effect, read as zero.	R
23:0	LEN	The buffer length value. (in WORD) The LEN bit field determines the number of bytes of the buffer size	R

		pointed by LCDSAx. LEN = 0 is not valid. DMAC fetch data according to LEN. Each time one or more word(s) been fetched, LEN is decreased automatically. Software can read LEN. *When you use decompressed function, the LEN should be the line number, not word number.	

1.8.33 DMA OFFSIZE Registers (LCDOFFSx, 0_PART2)

LCDOFFS0, LCDOFFS1, LCDOFFS0_PART2 **0x13050060, 0x13050070, 0x130501E0**

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								OFFSIZE																							
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
23:0	OFFSIZE0, 1 OFFSIZE0_PART2	OFFSIZE value for DMA 0,1. Indicate the offset in word. *please notice that when you need OFFSIZE function, to set this reg to an un-zero value and also need to set LCDPW0, 1 to indicate how much word in one line of this frame. *When you use decompress function, you must use this to indicate of how many word of a line in the source buffer.	R

1.8.34 DMA Page Width Registers (LCDPWx, 0_PART2)

LCDPW0, LCDPW1, LCDPW_PART2 **0x13050064, 0x13050074, 0x130501E4**

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								PAGEWIDTH																							
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
23:0	PAGEWIDTH0, 1 PAGEWIDTH0_PART2	Page width for DMA 0,1. * When you set LCDOFFS.OFFSIZE0/1 to 0, you need keep the PAGEWIDTH0/1 0. *When you use decompress function, you don't need set this register, dma will get pagewidth of every line automatically.	R

1.8.35 DMA Commend Counter Registers (LCDCNUMx)

When LCDCMD.CMD = 1, **0x13050068, 0x13050078** is use as LCDCNUM0, 1 LCDCNUM0_PART2 are not used now, set it to 0.

LCDCNUM0, LCDCNUM1, LCDCNUM0_PART2 **0x13050068, 0x13050078, 0x130501E8**

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																								CNUM							
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
7:0	CNUM0,1	Commands' number in this frame transfer by DMA. (only use in Smart LCD mode)	R

1.8.36 Foreground x Size in Descriptor (LCDDESSIZEx, 0_PART2)

When LCDCMD.CMD = 0, **0x1305006C, 0x1305007C** is use as LCDDESSIZE0, 1, to indicator the next frame foreground0, 1's size.

LCDDESSIZE0, 1, LCDDESSIZE0_PART2 **0x1305006C, 0x1305007C, 0x130501EC**

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				Height												Reserved				Width											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	Height	The height of foreground 0.	R
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	Width	The width of foreground 0.	R

1.8.37 Priority level threshold configure Register (LCDPCFG)

LCDPCFG **0x130502C0**

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
	Lcd_pri_md	HP_BST			Reserved	Pcfg2											Pcfg1												Pcfg0			
RST	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The 3 thresholds cut the output fifo into 4 space,. When the entries remained triggers one of them, the priority level will be altered by hardware. And the 3 thresholds must follow the order: pcfg2 ≥ pcfg1 ≥ pcfg0.

Priority level

Priority_lv	Space
11	Empty ~ threshold0
10	Threshold0 ~ threshold1
01	Threshold1 ~ threshold2
00	Threshold2 ~ full

Bits	Name	Description	RW
31	Lcd_pri_md	Lcd priority mode. 0: use lcd dynamitic priority level; 1: use arbiter priority level.	RW
30:28	HP_BST	Highest priority Burst Length Selection. 000: 4 word 001: 8 word 010: 16 word 011: 32 word 101: Contiue16 100: 64 word 111: disable	RW
27	Reserved	Writing has no effect, read as zero.	R
26:18	Pcfg2	Threshold2: 0~511.	RW
17:9	Pcfg1	Threshold1: 0~511.	RW
8:0	Pcfg0	Threshold0: 0~511	RW

1.9 LCD Controller Pin Mapping

There are several mapping schemes for different LCD panels.

1.9.1 TFT and CCIR Pin Mapping

Pin	Generic 8-bit Serial TFT	Generic 18/16-bit Parallel TFT		Special TFT 1 18/16-bit Parallel		Special TFT 2 18/16-bit Parallel		Special TFT 3 18/16-bit Parallel		CCIR656 8-bit	CCIR601 16-bit
Lcd_pclk/ Slcd_clk	CLK	CLK		DCLK		CLK		HCLK		CLK	CLK
Lcd_vsync/ Slcd_cs	VSYNC	VSYNC		SPS		GSRT		STV		VSYNC	VSYNC
Lcd_hsync/ Slcd_rs	HSYNC	HSYNC		LP		GPCK		STH		HSYNC	HSYNC
Lcd_de	DE	DE		-		-		-		-	-
Lcd_ps	-	-		Pulse		Toggle		Toggle		-	-
Lcd_cls	-	-		Pulse		Pulse		Pulse		-	-
Lcd_rev	-	-		Toggle		Toggle		Toggle		-	-
Lcd_spl	-	-		Pulse		Pulse		**Toggle**		-	-
Lcd_dat17	-	R5	-	R5	-	R5	-	R5	-	-	-
Lcd_dat16	-	R4	-	R4	-	R4	-	R4	-	-	-
Lcd_dat15	-	R3	R5	R3	R5	R3	R5	R3	R5	-	D15
Lcd_dat14	-	R2	R4	R2	R4	R2	R4	R2	R4	-	D14
Lcd_dat13	-	R1	R3	R1	R3	R1	R3	R1	R3	-	D13
Lcd_dat12	-	R0	R2	R0	R2	R0	R2	R0	R2	-	D12
Lcd_dat11	-	G5	R1	G5	R1	G5	R1	G5	R1	-	D11
Lcd_dat10	-	G4	G5	G4	G5	G4	G5	G4	G5	-	D10
Lcd_dat9	-	G3	G4	G3	G4	G3	G4	G3	G4	-	D9
Lcd_dat8	-	G2	G3	G2	G3	G2	G3	G2	G3	-	D8
Lcd_dat7	R7/G7/B7	G1	G2	G1	G2	G1	G2	G1	G2	D7	D7
Lcd_dat6	R6/G6/B6	G0	G1	G0	G1	G0	G1	G0	G1	D6	D6
Lcd_dat5	R5/G5/B5	B5	G0	B5	G0	B5	G0	B5	G0	D5	D5
Lcd_dat4	R4/G4/B4	B4	B5	B4	B5	B4	B5	B4	B5	D4	D4
Lcd_dat3	R3/G3/B3	B3	B4	B3	B4	B3	B4	B3	B4	D3	D3
Lcd_dat2	R2/G2/B2	B2	B3	B2	B3	B2	B3	B2	B3	D2	D2
Lcd_dat1	R1/G1/B1	B1	B2	B1	B2	B1	B2	B1	B2	D1	D1
Lcd_dat0	R0/G0/B0	B0	B1	B0	B1	B0	B1	B0	B1	D0	D0

TFT 24 bit parallel mode/16 bit parallel mode2:

Pin	16 bit Parallel mode2	24 bit Parallel
Lcd_pclk/ Slcd_clk	CLK	CLK
Lcd_vsync/Slcd_cs	VSYNC	VSYNC
Lcd_hsync/Slcd_rs	HSYNC	HSYNC
Lcd_de	DE	DE
Lcd_ps	-	-
Lcd_cls	-	-
Lcd_rev	-	-
Lcd_spl	-	-
Lcd_dat17	R7	R7
Lcd_dat16	R6	R6
Lcd_dat15	R5	R5
Lcd_dat14	R4	R4
Lcd_dat13	R3	R3
Lcd_dat12	G7	R2
Lcd_dat11	G6	G7
Lcd_dat10	G5	G6
Lcd_dat9	0 (NC for panel)	G5
Lcd_dat8	G4	G4
Lcd_dat7	G3	G3
Lcd_dat6	G2	G2
Lcd_dat5	B7	B7
Lcd_dat4	B6	B6
Lcd_dat3	B5	B5
Lcd_dat2	B4	B4
Lcd_dat1	B3	B3
Lcd_dat0	0 (NC for panel)	B2
Lcd_lo6_o[5]	0	R1
Lcd_lo6_o[4]	0	R0
Lcd_lo6_o[3]	0	G1
Lcd_lo6_o[2]	0	G0
Lcd_lo6_o[1]	0	B1
Lcd_lo6_o[0]	0	B0

1.9.2 Single Panel STN Pin Mapping

Pin	Color STN	Mono STN			
	PDW=3	PDW=0	PDW=1	PDW=2	PDW=3
Lcd_pclk	CLK	CLK	CLK	CLK	CLK
Lcd_vsync	VSYNC	VSYNC	VSYNC	VSYNC	VSYNC
Lcd_hsync	HSYNC	HSYNC	HSYNC	HSYNC	HSYNC
Lcd_de	BIAS	BIAS	BIAS	BIAS	BIAS
Lcd_ps	-	-	-	-	-
Lcd_cls	-	-	-	-	-
Lcd_rev	-	-	-	-	-
Lcd_spl	-	-	-	-	-
Lcd_dat17	-	-	-	-	-
Lcd_dat16	-	-	-	-	-
Lcd_dat15	-	-	-	-	-
Lcd_dat14	-	-	-	-	-
Lcd_dat13	-	-	-	-	-
Lcd_dat12	-	-	-	-	-
Lcd_dat11	-	-	-	-	-
Lcd_dat10	-	-	-	-	-
Lcd_dat9	-	-	-	-	-
Lcd_dat8	-	-	-	-	-
Lcd_dat7	D7	-	-	-	D7
Lcd_dat6	D6	-	-	-	D6
Lcd_dat5	D5	-	-	-	D5
Lcd_dat4	D4	-	-	-	D4
Lcd_dat3	D3	-	-	D3	D3
Lcd_dat2	D2	-	-	D2	D2
Lcd_dat1	D1	-	D1	D1	D1
Lcd_dat0	D0	D0	D0	D0	D0

1.9.3 Dual Panel STN Pin Mapping

Pin	Color STN	Mono STN			
	PDW=3	PDW=0	PDW=1	PDW=2	PDW=3
Lcd_pclk	CLK	-	-	CLK	CLK
Lcd_vsync	VSYNC	-	-	VSYNC	VSYNC
Lcd_hsync	HSYNC	-	-	HSYNC	HSYNC
Lcd_de	BIAS	-	-	BIAS	BIAS
Lcd_ps	-	-	-	-	-
Lcd_cls	-	-	-	-	-
Lcd_rev	-	-	-	-	-
Lcd_spl	-	-	-	-	-
Lcd_dat17	-	-	-	-	-
Lcd_dat16	-	-	-	-	-
Lcd_dat15	UD7	-	-	-	UD7
Lcd_dat14	UD6	-	-	-	UD6
Lcd_dat13	UD5	-	-	-	UD5
Lcd_dat12	UD4	-	-	-	UD4
Lcd_dat11	UD3	-	-	UD3	UD3
Lcd_dat10	UD2	-	-	UD2	UD2
Lcd_dat9	UD1	-	-	UD1	UD1
Lcd_dat8	UD0	-	-	UD0	UD0
Lcd_dat7	LD7	-	-	-	LD7
Lcd_dat6	LD6	-	-	-	LD6
Lcd_dat5	LD5	-	-	-	LD5
Lcd_dat4	LD4	-	-	-	LD4
Lcd_dat3	LD3	-	-	LD3	LD3
Lcd_dat2	LD2	-	-	LD2	LD2
Lcd_dat1	LD1	-	-	LD1	LD1
Lcd_dat0	LD0	-	-	LD0	LD0

1.9.4 Data mapping to GPIO function.

pin name in LCD	mapping to GPIO function
Lcd_dat17/Slcd_dat17	lcd_r7
Lcd_dat16/Slcd_dat16	lcd_r6
Lcd_dat15/Slcd_dat15	lcd_r5
Lcd_dat14/Slcd_dat14	lcd_r4
Lcd_dat13/Slcd_dat13	lcd_r3
Lcd_dat12/Slcd_dat12	lcd_r2
Lcd_dat11/Slcd_dat11	lcd_g7
Lcd_dat10/Slcd_dat10	lcd_g6
Lcd_dat9/Slcd_dat9	lcd_g5
Lcd_dat8/Slcd_dat8	lcd_g4
Lcd_dat7/Slcd_dat7	lcd_g3
Lcd_dat6/Slcd_dat6	lcd_g2
Lcd_dat5/Slcd_dat5	lcd_b7
Lcd_dat4/Slcd_dat4	lcd_b6
Lcd_dat3/Slcd_dat3	lcd_b5
Lcd_dat2/Slcd_dat2	lcd_b4
Lcd_dat1/Slcd_dat1	lcd_b3
Lcd_dat0/Slcd_dat0	lcd_b2
Lcd_lo6_o[5]	lcd_r1
Lcd_lo6_o[4]	lcd_r0
Lcd_lo6_o[3]	lcd_g1
Lcd_lo6_o[2]	lcd_g0
Lcd_lo6_o[1]	lcd_b1
Lcd_lo6_o[0]	lcd_b0

1.10 Display Timing

1.10.1 General 16-bit and 18-bit TFT Timing

This section shows the general 16-bit and 18-bit TFT LCD timing diagram, the polarity of signal “Vsync”, “Hsync”, and “PCLK” can be programmed correspond to the LCD panel specification.

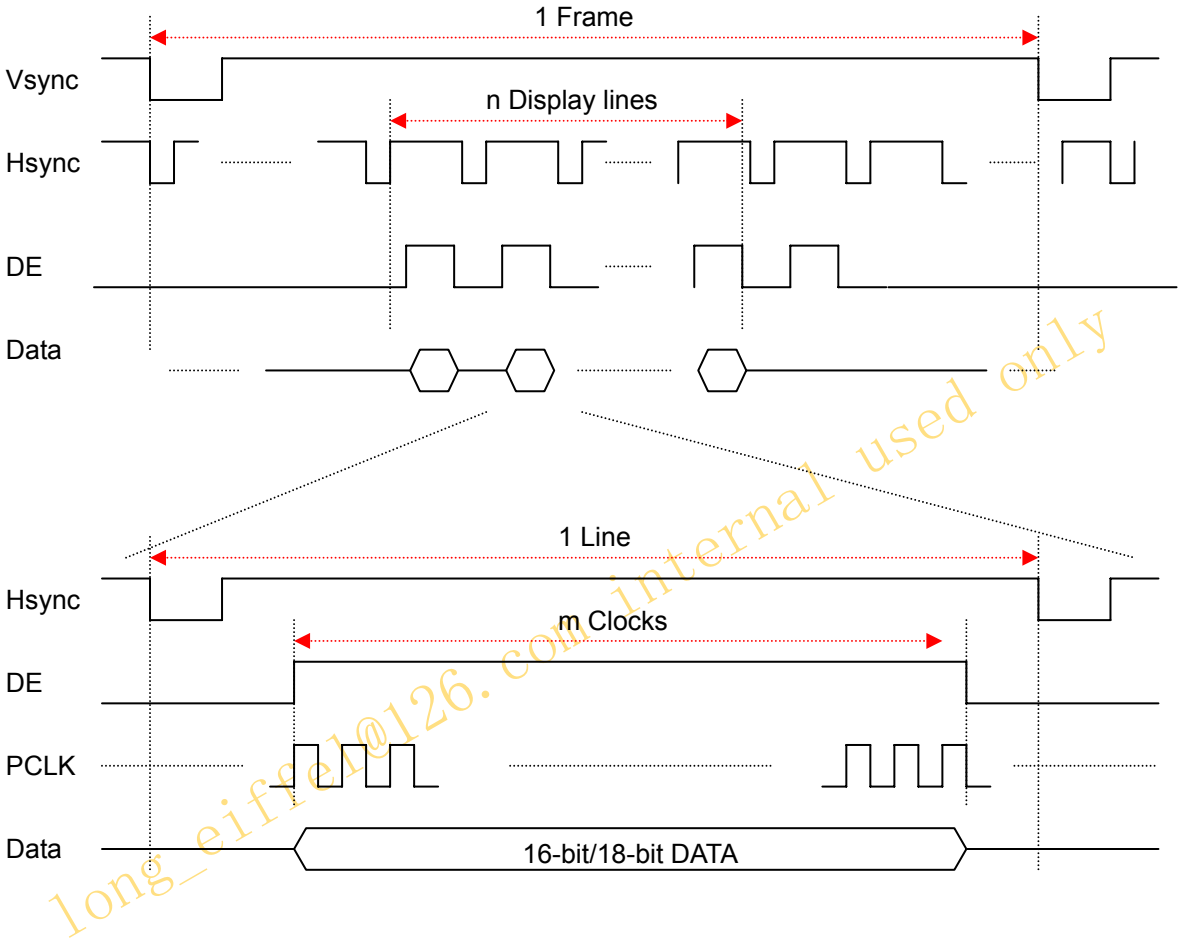


Figure 1-8 General 16-bit and 18-bit TFT LCD Timing

1.10.2 8-bit Serial TFT Timing

This section shows the 8-bit serial TFT LCD timing diagram, the polarity of signal “Vsync”, “Hsync”, and “PCLK” can be programmed correspond to the LCD panel specification.

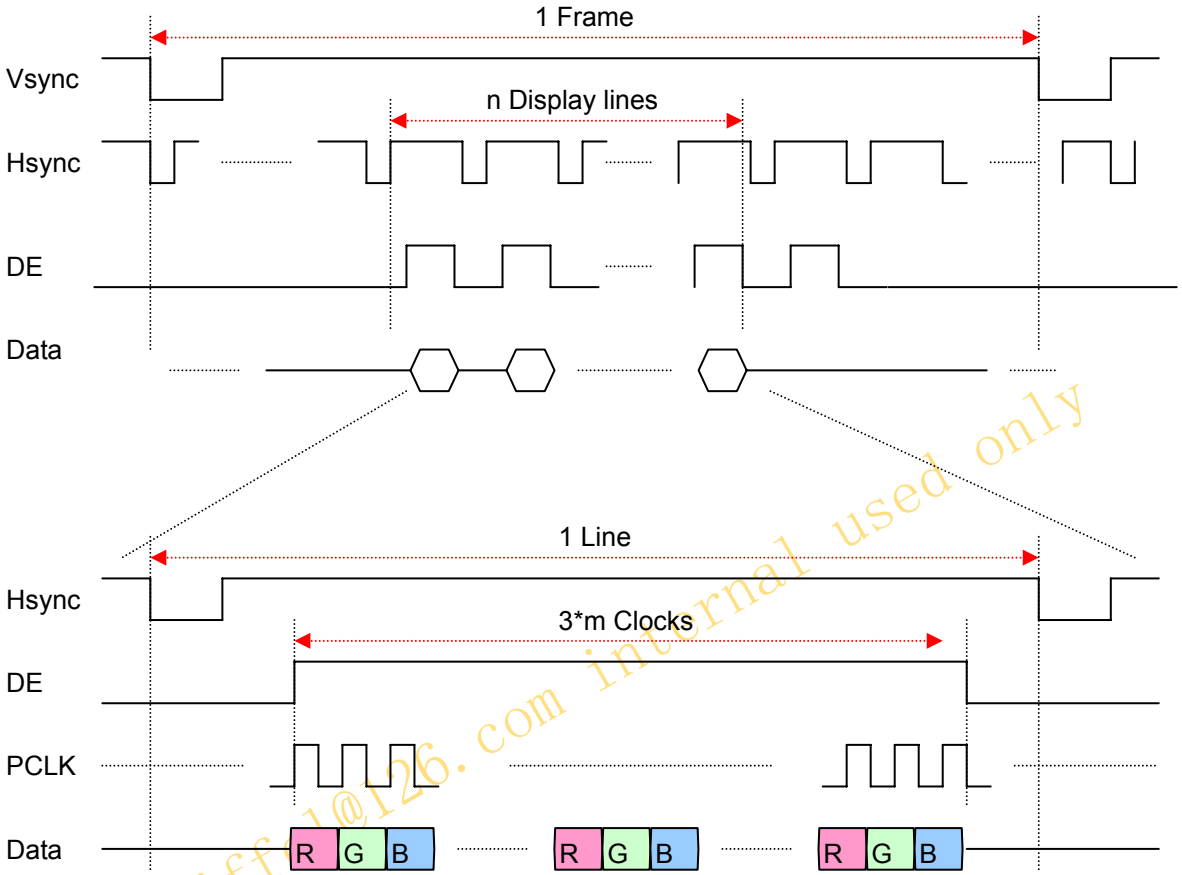


Figure 1-9 8-bit serial TFT LCD Timing (24bpp)

1.10.3 Special TFT Timing

Based on the general TFT LCD support, this controller also provides 4 special signals that can be programmed to general some special timing used for some panel. All 4 signals are worked in two modes: pulse mode and toggle mode. Signal “CLS” is fixed in pulse mode, and “REV” in toggle mode. The work mode of signals “SPL” and “PS” are defined in the special TFT LCD mode 1 to mode 3, either pulse mode or toggle mode. The position and polarity of these 4 signals can be programmed via registers. The Figures show the two modes as follows: (The toggle mode of signal “SPL” is different with the others signal. “SPL” does toggle after display line.)

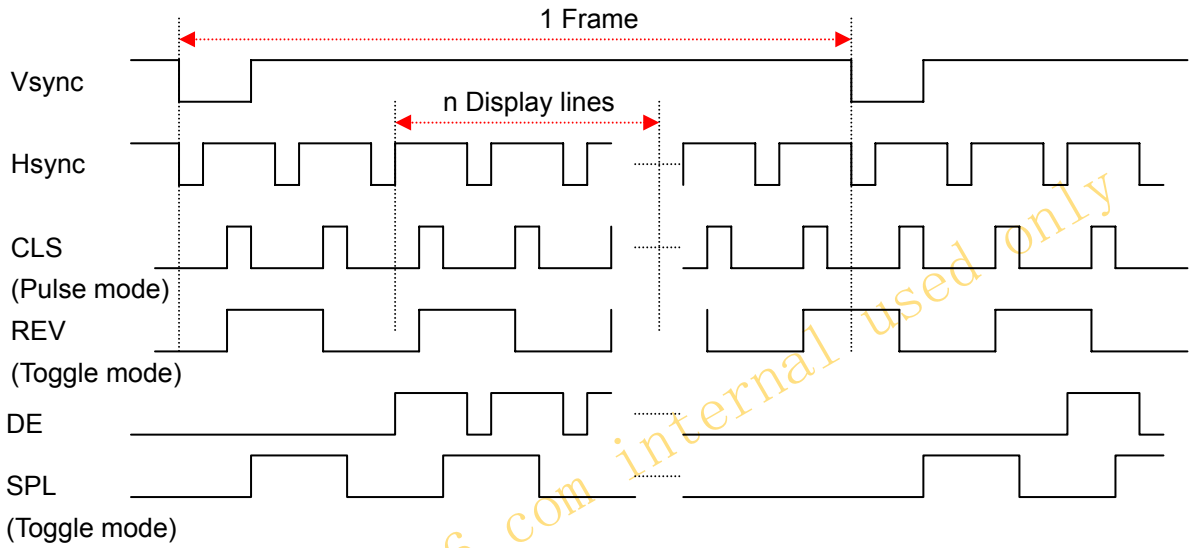


Figure 1-10 Special TFT LCD Timing 1

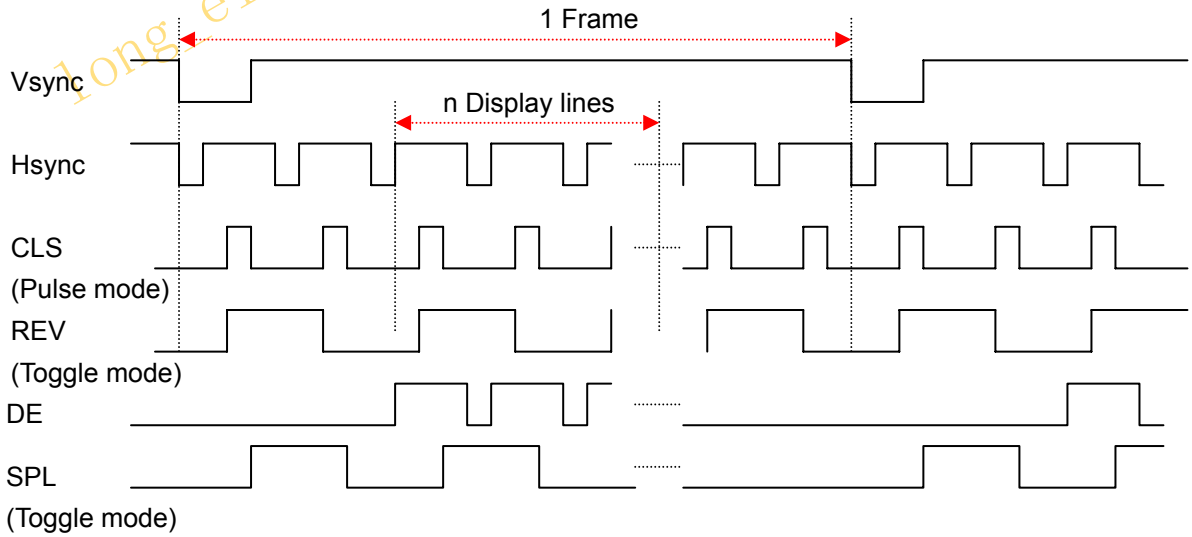


Figure 1-11 Special TFT LCD Timing 2

These two Figures show the timing of pulse mode and toggle mode, the pulse mode timing is same

and the toggle mode timing is different. Timing 1 shows the condition when the total lines in 1 frame is odd (the number of display is even and the number of blank is odd), so the phase of REV inverse at the first line of each frame and the phase of SPL dose not inverse at the first line of each frame. Timing 2 shows the condition when the total lines in 1 frame is even (the number of display is even and the number of blank is even), so the phase of REV and SPL dose not inverse at the first line of each frame.

When LCDC is enabled ,there will be a null line to be add before transferring data to LCD panel. So the toggle mode exept SPL signal of special 3 TFT mode is when reset level is high,the first valid edge will be rising edge. SPL signal of special 3 TFT mode is when reset level is high,the first valid edge will be falling edge.

1.10.4 Delta RGB panel timing

This section shows the Delta RGB timing diagram, the polarity of signal “Vsync”, “Hsync”, and “PCLK” can be programmed. And the odd/even line RGB order also can be programmed correspond to the LCD panel specification.

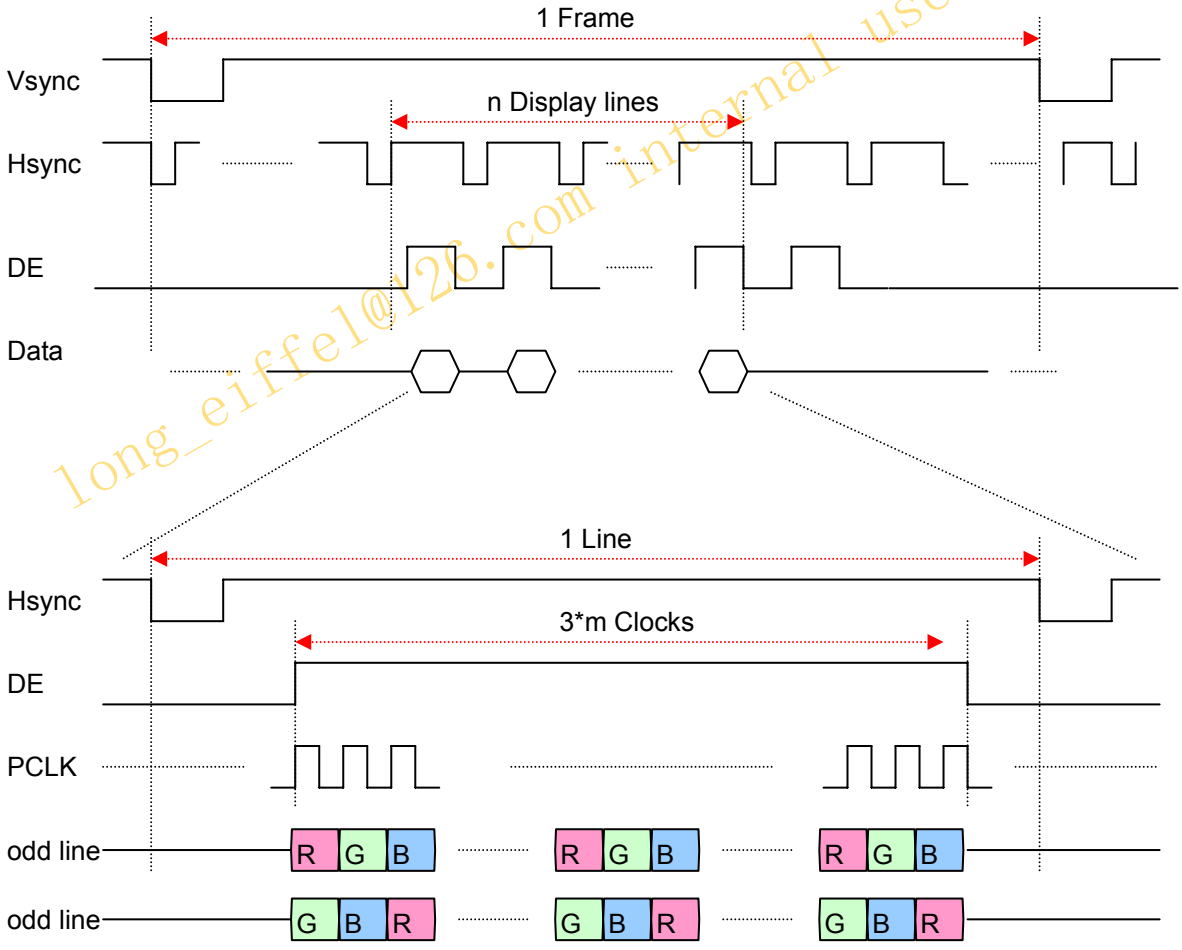
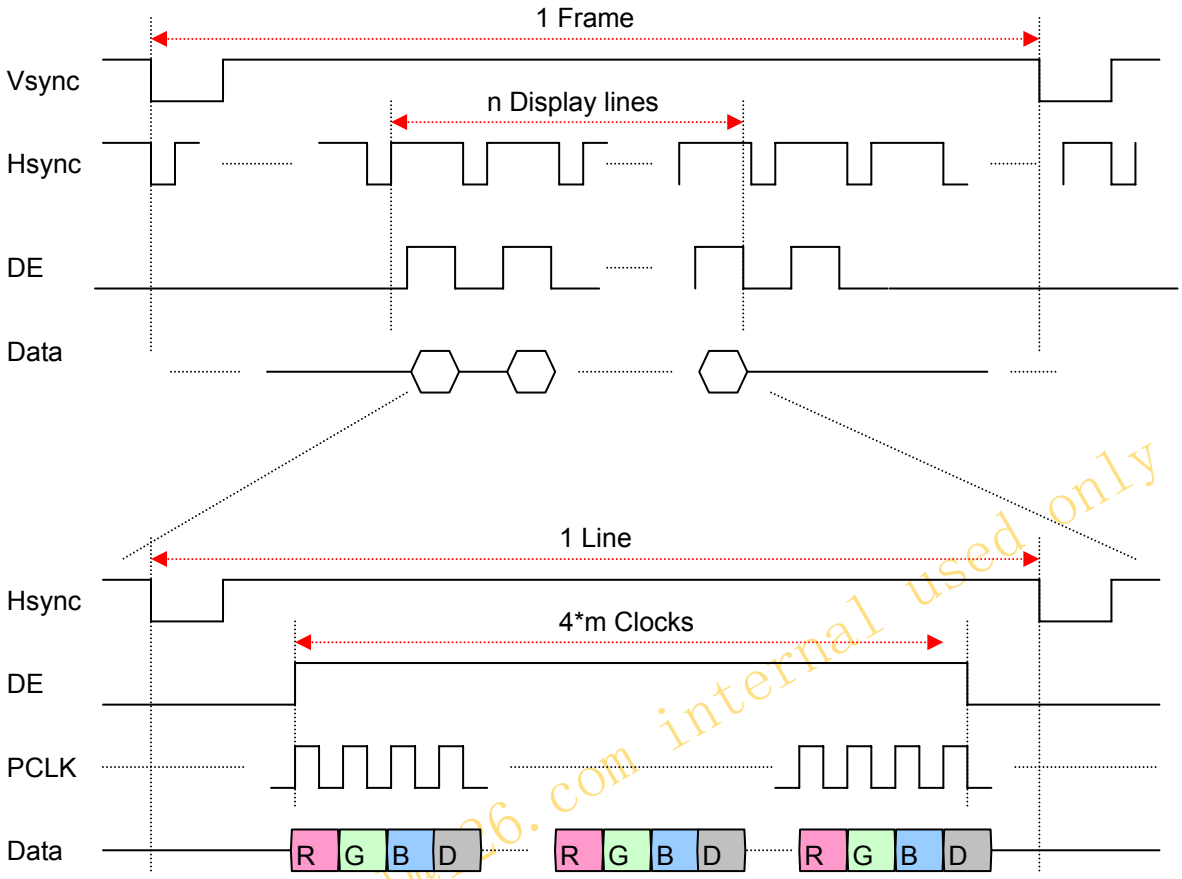


Figure 1-12 Delta RGB timing

1.10.5 RGB Dummy mode timing

This section shows the RGB Dummy diagram, the polarity of signal “Vsync”, “Hsync”, and “PCLK” can be programmed.



***Dummy = 0**

Figure 1-13 RGB Dummy timing

1.11 Format of Palette

This LCD controller contains a palette RAM with 256-entry x 16-bit used only for BPP8, BPP4, BPP2 and BPP1. Palette RAM data is loaded directly from the external memory palette buffer by DMAC channel 0. Each word of palette buffer contains 2 palette entries.

1 In 8-bpp modes, palette buffer size is128 words.
2 In 4-bpp modes, palette buffer size is 8 words.
3 In 2-bpp modes, palette buffer size is 2 words.
4 In 1-bpp modes, palette buffer size is 1 word.
5 In 16/18/24-bpp modes, has no palette buffer.

Palette buffer base address	Bit: 31 . . . 16	Bit: 15 . . . 0
Palette entry	Entry-1 bit: 15 . . . 0	Entry-0 bit: 15 . . . 0
Palette buffer base address + 4	Bit: 31 . . . 16	Bit: 15 . . . 0
Palette entry	Entry-3 bit: 15 . . . 0	Entry-2 bit: 15 . . . 0
Palette buffer base address + 8	Bit: 31 . . . 16	Bit: 15 . . . 0
Palette entry	Entry-5 bit: 15 . . . 0	Entry-4 bit: 15 . . . 0

1.11.1 STN

For STN Panel, 16-bpp pixel data is encoded with RGB 565 or RGB 555.
Please refer to register LCDCTRL.RGB.

BPP 16, RGB 565, pixel encoding for STN Panel:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0

BPP 16, RGB 555, pixel encoding for STN Panel:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	R4	R3	R2	R1	R0	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0

1.11.2 TFT

BPP 16, RGB 565, pixel encoding for TFT Panel:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0

NOTE: For BPP 16, 18, 24, palette is bypass.

1.12 Format of Frame Buffer

1.12.1 16bpp

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R4	R3	R2	R1	R0	G5	G4	G3	G2	G1	G0	B4	B3	B2	B1	B0

1.12.2 18bpp

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	R5	R4	R3	R2	R1	R0	0	0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
G5	G4	G3	G2	G1	G0	0	0	B5	B4	B3	B2	B1	B0	0	0

1.12.3 24bpp

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	R7	R6	R5	R4	R3	R2	R1	R0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0

1.12.4 16bpp with alpha

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A7	A6	A5	A4	A3	A2	A1	A0	0	0	0	0	0	0	0	0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
R5	R4	R3	R2	R1	G5	G4	G3	G2	G1	G0	B5	B4	B3	B2	B1

1.12.5 18bpp with alpha

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A7	A6	A5	A4	A3	A2	A1	A0	R5	R4	R3	R2	R1	R0	0	0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
G5	G4	G3	G2	G1	G0	0	0	B5	B4	B3	B2	B1	B0	0	0

1.12.6 24bpp with alpha

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A7	A6	A5	A4	A3	A2	A1	A0	R7	R6	R5	R4	R3	R2	R1	R0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
G7	G6	G5	G4	G3	G2	G1	G0	B7	B6	B5	B4	B3	B2	B1	B0

1.12.7 24bpp compressed

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

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RED 3 [7:0]								GLEEN 3 [7:0]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BLUE3 [7:0]								RED2 [7:0]							

1.13 Format of Data Pin Utilization

1.13.1 Mono STN

In Mono STN mode, data pin pixel ordering of one LCD screen row. Column 0 is the first pixel of a screen row.

Upper panel								
Panel data width	**Col0**	**Col1**	**Col2**	**Col3**	**Col4**	**Col5**	**Col6**	**Col7**
1 bit	D0	D0	D0	D0	D0	D0	D0	D0
2 bit	D1	D0	D1	D0	D1	D0	D1	D0
4 bit	D3	D2	D1	D0	D3	D2	D1	D0
8 bit	D7	D6	D5	D4	D3	D2	D1	D0
Lower panel (dual-panel mode)								
4 bit	D11	D10	D9	D8	D11	D10	D9	D8
8 bit	D15	D14	D13	D12	D11	D10	D9	D8

1.13.2 Color STN

In Color STN mode, data pin pixel ordering of one LCD screen row. Column 0 is the first pixel of a screen row.

Upper panel							
Col0 (R)	**Col0 (G)**	**Col0 (B)**	**Col1 (R)**	**Col1 (G)**	**Col1 (B)**	**Col2 (R)**	**Col2 (G)**
D7	D6	D5	D4	D3	D2	D1	D0
Lower panel (dual-panel mode)							
D15	D14	D13	D12	D11	D10	D9	D8

1.13.3 18-bit Parallel TFT

Col0 (RGB)																	
D17	D16	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

1.13.4 16-bit Parallel TFT

Col0 (RGB)															
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0

1.13.5 8-bit Serial TFT (24bpp)

Col0 (R)							
D7	D6	D5	D4	D3	D2	D1	D0
Col0 (G)							
D7	D6	D5	D4	D3	D2	D1	D0
Col0 (B)							
D7	D6	D5	D4	D3	D2	D1	D0

1.14 LCD Controller Operation

1.14.1 Set LCD Controller AHB Clock and Pixel Clock

The LCD Controller has 2 clock input: AHB clock and pixel clock. The both clocks are generated by CPM (Clock and Power Manager). The frequency of the 2 clocks can be set by CPM registers. lcdc 's AHB clock is equal to AHB0 clock (HCLK in CPM spec), and CPM.LPCDR set LCD pixel clock division ratio. Please refer to CPM spec for detail.

LCD AHB clock is the LCD controller's internal clock while LCD pixel clock is output to drive LCD panel. There have 2 rules for LCD clocks:

1 For TFT Panel, the frequency of LCD AHB clock must be at least 1.5 times of LCD pixel clock.
2 For STN Panel, the frequency of LCD AHB clock must be at least 3 times of LCD pixel clock.

LCD panel determines the frequency of LCD pixel clock.

1.14.2 Enabling the Controller

If the LCD controller is being enabled for the first time after system reset or sleep reset, all of the LCD registers must be programmed as follows:

1 Write the frame descriptors and, if needed, the palette descriptor to memory.
2 Program the entire LCD configuration registers except the Frame Descriptor Address Registers (LCDDAx) and the LCD Controller enable bit (LCDCTRL.ENA).
3 Program LCDDAx with the memory address of the palette/frame descriptor.
4 Enable the LCD controller by writing to LCDCTRL.ENA.

If the LCD controller is being re-enabled, there has not been a reset since the last programming; only the registers LCDDAx and LCDCTRL.ENA need to be reprogrammed. The LCD Controller Status Register (LCDSTATE) must also be written to clear any old status flags before re-enabling the LCD controller.

Once the LCD controller has been enabled, do not write new values to LCD registers except LCDCTRL.ENA or DIS or LCDDA0/1 or LCDOSDC.F0/1EN .

1.14.3 Disabling the Controller

The LCD controller can be disabled in two ways: regular and quick.

1 Regular disabling.

Regular disabling is accomplished by setting the disable bit, LCDCTRL.DIS. The other bits in LCDCTRL must not be changed — read the register, set the DIS bit, and rewrite the register. This method causes the LCD controller to stop cleanly at the end of a frame. The LCD Disable Done bit, LCDSTATE.LDD, is set when the LCD controller finishes displaying the last frame, and the enable bit, LCDCTRL.ENA, is cleared automatically by hardware.

LCDCTRL.DIS must be set zero when enabling the controller.

2 Quick disabling.

Quick disabling is accomplished by clearing the enable bit, LCDCTRL.ENA. The LCD controller will finish any current DMA transfer, stop driving the panel, setting the LCD Quick Disable bit (LCDSTATE.QD) and shut down immediately. This method is intended for situations such as a battery fault, where system bus traffic has to be minimized immediately so the processor can have enough time to store critical data to memory before the loss of power. The LCD controller must not be re-enabled until the QD bit is set, indicating that the quick shutdown is complete. Do not set the DIS bit when a quick disabling command has been issued.

NOTE: It is strongly recommended that software set the "LCD Module Stop Bit" in PMC to shut down LCDC clock supply to save power consumption after disable LCDC. Please refer to PMC for detailed information.

1.14.4 Resetting the Controller

At reset, the LCD Controller is disabled. All LCD Controller Registers are reset to the conditions shown in the register descriptions.

1.14.5 Frame Buffer & Palette Buffer

The starting address of frame buffer stored in external memory must be aligned to 4, 8 or 16 words boundary according to register LCDCTRL.BST. The length of buffer must be multiple of word (32-bit).

If LCDCTRL .BST = 0, align frame and palette buffer to 16 word boundary
If LCDCTRL .BST = 1, align frame and palette buffer to 8 word boundary
If LCDCTRL .BST = 2, align frame and palette buffer to 4 word boundary

One frame buffer contains encoded pixel data of multiple of screen lines; each line of encoded pixel data must be aligned to word boundary. If the length of a line is not the multiple of word, extra bits must be applied to reach a word boundary. It is suggested that the extra bits to be set zero.

1.14.6 CCIR601/CCIR656

CCIR601: just as 16bit-parallel output.

CCIR656: need external encoder, or software designer need give digital blanking data and timing reference signal in data buffer.

1.14.7 OSD Operation

1 Normal process.

a Configuration.

* LCDCFG and LCDCTRL
* LCDOSDC and LCDOSDCTRL

* LCDRGBC and LCDIPUR

b Set Color.
* LCDBGC, LCDKEY0, LCDKEY1, LCDALHPA

c Set Display.
* LCDVAT, LCDDAH, LCDDAV
* LCDXYP0, LCDXYP1, LCDSIZE0, LCDSIZE1
* LCDVSYNC, LCDHSYNC

d Set DMAC.
* LCDIID
* LCDDA0, LCDSA0, LCDFID0, LCDCMD0, LCDOFFS0, LCDPW0, LCDCNUM0, LCDDESSIZE0
* LCDDA1, LCDSA1, LCDFID1, LCDCMD1, LCDOFFS1, LCDPW1, LCDCNUM1, LCDDESSIZE1

e Enable LCDC.

f Check the state from register LCDSTATE and LCDOSDS.

2 Reconfigure OSD.

If foreground0 and foreground1 (enable, position, size) need to reconfigure during display process, there has two methods.

Method1: (recommend in TFT and SLCD)

a Reconfigure the relate Register after disable LCDC.
b In TFT mode, use normal disable to avoid lcd panel flicker.
c In SLCD mode, use quick disable. (smart LCD could keep the frame by its inner buffer)
d After disable LCDC, you can reconfigure any register/descriptor, but please make sure this process is quick enough in TFT mode. (less than the interval between two frame)

Method2:

Dynamic reconfigure the register:

You can reconfigure some register(LCDOSDC.F0/1EN) during display process but there some rule you must follow:

a Foreground 0 and foreground 1's data can not less than 65 words(except 0 word). Or you only can change those register after disable LCDC.

b When use TFT panel. During the display process, you can re-configure the LCDOSDC.F0EN, LCDCOSDC.F1EN; (You can not change them when use SLCD or TVE) but the new configuration will recognized by LCDC module after finished a complete frame. If you need to re-configure LCDOSDCTRL.IPU to

select IPU or DMA channel 1, you need to follow the process below:

– Quick or Normal disable LCDC. (SLCD only can use Quick disable)
– Configure the LCDOSDCTRL to set IPUEN, and then enable LCD.

To change IPU to DMA1 you can:

– Quick or Normal disable LCDC. (SLCD only can use Quick disable)
– Configure the LCDOSDCTRL to set IPUEN = 0, and then enable LCD.

3 During the display process, while foreground 1 use IPU, to change size of foreground 1 you need follow the step shown bellow:

Method1:

a Quick or Normal disable LCDC. (SLCD only can use Quick disable)
b Configure the IPU, and LCDSIZE1.
c Run IPU and enable LCDC.

Method2:

a Set LCDCOSDC.F1= 0. (follow the rule above)
b Configure the IPU.
c Change LCDSIZE1.(follow the rule change LCDSIZE1 method 2).
d Set LCDCOSDC.F1= 1. (follow the rule above)

4 You **CAN NOT** change BPP or OSDBPP during the display process. if you want to change them first you should disable LCDC, change the BPP or OSDBPP and then enable LCDC.
If you need not use Foreground0 during the whole display process. set BPP to 5.

5 You can change LCDSIZE0/1 during display process without disable LCD controller.
Method 1:

a Set LCDCOSDC.F0/1EN = 0. (follow the rule above)
b Re-configure LCDSIZE0/1 (and the relate DMA0/1 descriptor); then set LCDOSDCTRL.CHANGE = 1.
c Wait until CHANGE = 0 and then set LCDOSDC.F0/1EN = 1.

Method 2:

a Set LCDOSDCTRL.CHANGE = 1.
b Wait until LCDOSDS. READY = 1.
c Change relate DMA0/1 channel descriptor.
d Wait until LCDOSDCTRL.CHANGE = 0.

*Please notice that in TVE (not include VGA) and SLCD only use method 2.

6 You can change LCDXY0/1 during display process without disable LCD controller.
Method 1:

a Set LCDOSDC.F0/1EN = 0. (follow the rule above)
b Change LCDXYPOS0/1 and then set LCDOSDCTRL.CHANGE = 1.
c Wait until CHANGE = 0 and then set LCDOSDC.F0/1EN = 1.

Method 2:

a Set LCDOSDCTRL.CHANGE = 1.
b Wait until LCDOSDS. READY = 1.
c Change LCDXYPOS0/1.
d Wait until LCDOSDCTRL.CHANGE = 0.

*Please notice that in TVE (not include VGA) and SLCD only use method 2.

*Please notice that if you do not change foreground 0/1's size and position, keep LCDOSDCTRL.CHANGE = 0. And you can only change one of them in one time.

7 How to "close/open" foreground0 and foreground1?

Method 1:

a Set LCDOSDCTRL.CHANGE = 1.
b Wait until LCDOSDS. READY = 1.
c Direct change LCDOSDC.F0/1EN.
d Wait until LCDOSDCTRL.CHANGE = 0.

Method 2:

Change foreground0/1 size to 0 Without change LCDOSDC.F0/1EN.

Method 3: (recommend)

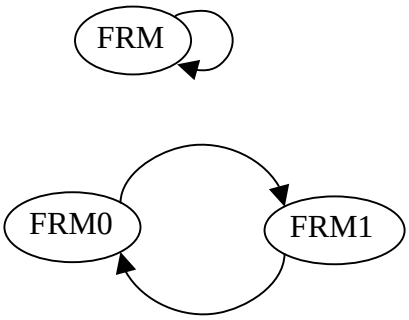
Normal disable LCDC, and change LCDOSDC.F0/1EN. Use normal disable need to wait LCDSTATE.LDD, and set relate register soon, to make sure the LCD panel are not flicker.

*Please notice that in TVE (not include VGA) and SLCD only use method 2,3. And strongly suggest that DO NOT close both foreground0 and 1 or set both foreground0 and 1 's size to 0.

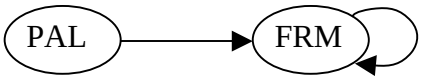
1.14.8 Descriptor Operation

1 TFT panel

Not use palette: you can use only one descriptor or several connected descriptor. As which shown below.



Use palette: add one PAL descriptor at the beginning of descriptor chain.

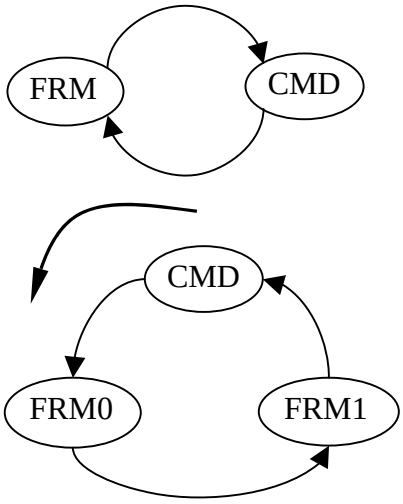


When you need to change palette during the display you need follow the steps shown below.

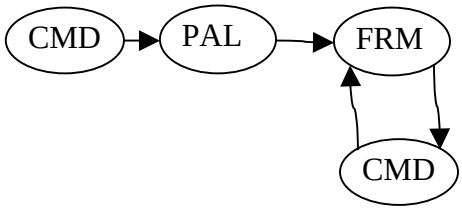
*Please notice that you **cannot** disable foreground 0 during the whole process. and also You can not change PAL when Foreground 0' s area == 0 or not enable LCDOSDC.F0EN.

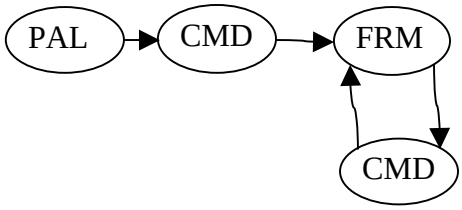
2 SLCD

Not use palette.

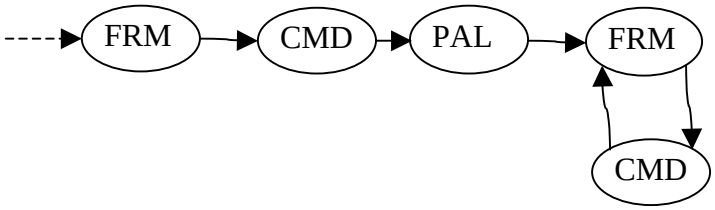


Use palette.





Change palette.



You can not change PAL when Foreground 0' s area == 0. Or not enable LCDOSDC.F0EN and during you change PAL, you can not change F0 or F1's size.

1.14.9 IPU direct connect mode

When you use IPU direct connect mode, you need to:

1. Open IPU early than LCDC.
2. Use normal disable in TFT mode, and use quick disable in SLCD/TVE mode.
3. When you use normal disable you need to wait IPU frame end flag.
4. When you use quick disable you must not wait IPU frame end flag, and must reset IPU before restart LCDC and IPU.
5. In SLCD mode, you can first wait IPU frame end flag, then quick stop LCDC. Then you need not reset IPU before restart LCDC and IPU.

* "IPU frame end flag" please refer to IPU spec.

1.14.10 VGA output

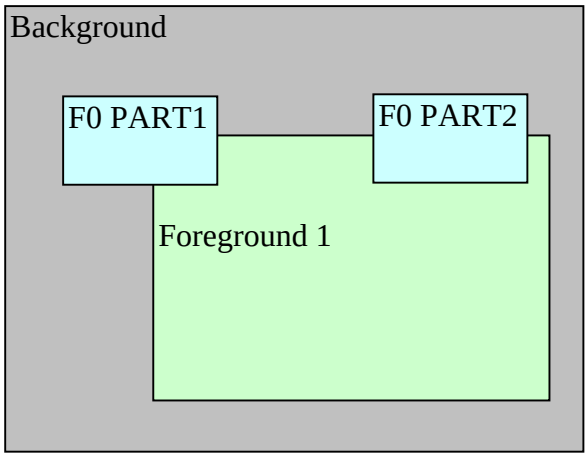
When you use VGA output you need:

1 Open all channel of DAC. (refer to TVEDAC spec)
2 Set TVEN to 0.
3 Disable LCD panel pins (except HSYNC/VSYNC) for save power. (refer to GPIO spec)

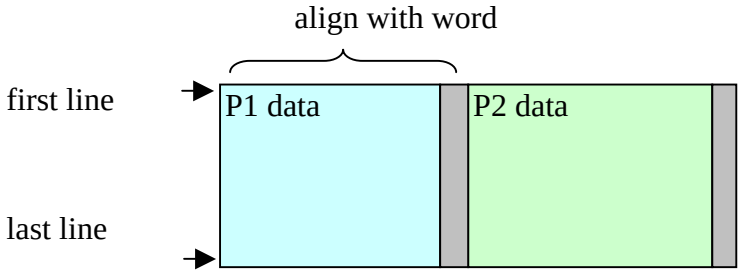
1.14.11 Foreground 0 divide mode

In divide mode the original register of foreground 0 position and size are correspond to F0 PART1, the additional (named with "_part2") registers correspond to F0 PART2.
LCDOSDC.F0EN correspond the total foreground 0 （part1 and part2）and each part has a F0PxEN to enable.
F0EN, F0P2EN, F0P1EN, and part2's position/size can be reconfigure during display process. (refer to 1.12.6)

MODE 1: LCDOSDC.F0DIVMD = 0. F0P2MD = 1.

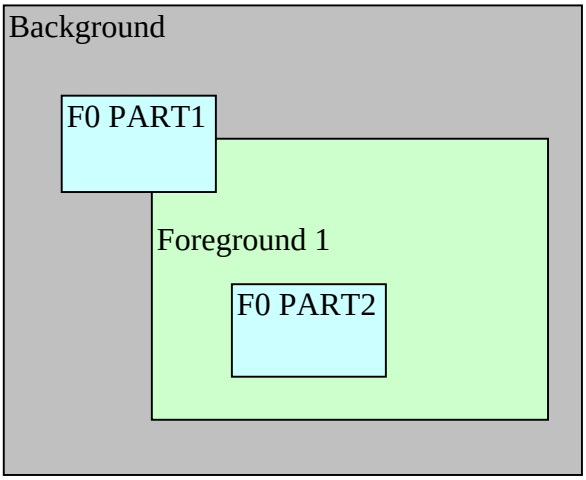


Foreground 0 divided into 2 parts, and PART1, PART2 must begin with same line and has the same height. They can have different width but cannot overlay each other.
They can use only one descriptor (connect to it self).
The two parts data must "combination" in one data buffer as follow:



*Please notice that in this mode, you need to disable LCDC before reconfigure foreground0/1's Register.

MODE 2: LCDOSDC.F0DIVMD = 1. F0P2MD = 0. F0P1EN = 1. F0P2EN = 1.



Foreground 0 divided into 2 parts, and PART1, PART2 can have different width and height but cannot overlay each other. PART2 must below PART1 they also cannot have any superposition in vertical.

PART1 and PART2 use independent descriptor refer to descriptor register with “part2”

2 Smart LCD Controller

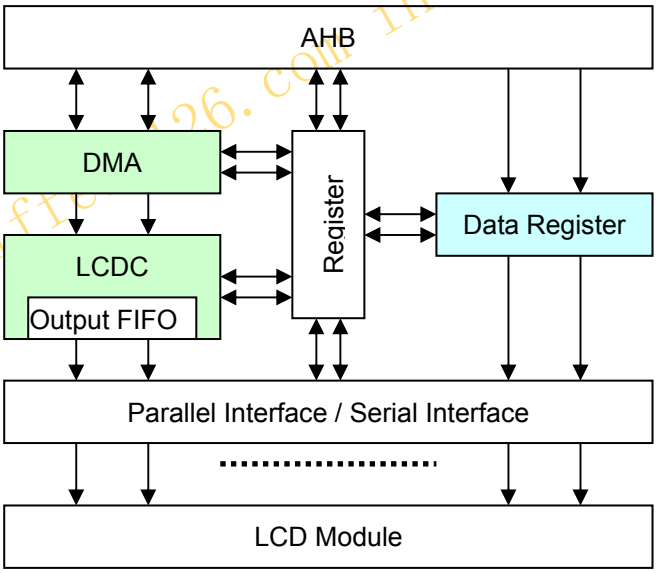
2.1 Overview

The Smart LCD Controller affords an interface to transfer data from the LCD controller to the LCD Module. It supports DMA operation and register operation.

Features:

- Supports a large variety of LCD Module from different vendors
- Supports parallel and serial interfaces
- Supports different size of display panel
- Supports different width of pixel data
- Supports internal DMA operation and register operation
- Supports Write Operation. Read Operation is not supported

2.2 Structure



*Please notice that the command only can transfer by DMA channel 0. No matter the DMA channel 1 or IPU are use or not.

2.3 Pin Description

Table 2-1 SLCD Pins Description

Name	I/O	Description	Interface
SLCD_RS	O	Command/Data Select Signal. The polarity of the signal can be programmable.	Serial: RS Parallel: RS
SLCD_CS	O	Data Sample Signal. The polarity of the signal can be programmable.	Serial: CS Parallel: Sample Data with the edge of CS
SLCD_CLK	O	The clock of SLCD. The polarity of the clock can be programmable.	Serial or not used
SLCD_DAT*1 [17:0]	O	The data of SLCD. Relate to 1.9.4 Data mapping to GPIO function.	Serial: SLCD_DAT [15] Parallel: 18bit SLCD_DAT [17:0] 16bit SLCD_DAT [15:0] 8bit SLCD_DAT [7:0]
LCD_LO6_O	O	24 bit parallel SLCD RGB (or 24 bit command) low bit ([17:16],[9:8],[1:0]) output Relate to 1.9.4 Data mapping to GPIO function.	--

NOTE:

***[1]:** SLCD_DAT [15] is also use as data pin for serial. The SLCD pins are shared with LCDC. You can see the set of register LCDCFG.LCDPIN in LCDC spec.

2.4 Register Description

In this section, we will describe the registers in Smart LCD controller. Following table lists all the registers definition. All register's 32bit address is physical address. And detailed function of each register will be described below.

Name	Description	RW	Reset Value	Address	Access Size
MCFG	SLCD Configure Register	RW	0x0000	0x130500A0	32
MCTRL	SLCD Control Register	RW	0x00	0x130500A4	8
MSTATE	SLCD Status Register	RW	0x00	0x130500A8	8
MDATA	SLCD Data Register	RW	0x00000000	0x130500AC	32

.

2.4.1 SLCD Configure Register (MCFG)

The register MCFG is used to configure SLCD.

MCFG 0x130500A0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved			DWIDTH			CWIDTH		Reserved			CSPLY	RSPLY	Reserved	CLKPLY	TYPE
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
15:13	Reserved	Writing has no effect, read as zero.	R
12:10	DWIDTH*1	Data Width.	RW

DWIDTH	Data Width
000	18-bit once Parallel/Serial
001	16-bit once Parallel/Serial
010	8-bit third time Parallel
011	8-bit twice Parallel
100	8-bit once Parallel/Serial
101	24-bit once Parallel
111	9-bit twice Parallel
110	Reserved

*Please notice that you can only use 24-bit parallel command when use 24-bit parallel data. (The command may not 24-bit but need put them as 24-bit in memory(one command use one word))

<table>
<tr><td>9:8</td><td>CWIDTH*1</td><td>Command Width.<table><tr><th>CWIDTH</th><th>Command Width</th></tr><tr><td>00</td><td>16-bit once / 9bit once</td></tr><tr><td>01</td><td>8-bit once</td></tr><tr><td>10</td><td>18-bit once</td></tr><tr><td>11</td><td>24-bit once</td></tr></table>*Please notice that you can only use 24-bit parallel command when use 24-bit parallel data. (The command may not 24-bit but need put them as 24-bit in memory (one command use one word))</td><td>RW</td></tr>
<tr><td>7:5</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>4</td><td>CSPLY</td><td>CS Polarity. (CS initial level will be different from CS Polarity)
0: Active Level is Low
1: Active Level is High</td><td>RW</td></tr>
<tr><td>3</td><td>RSPLY</td><td>RS Polarity.
0: Command RS = 0, Data RS = 1
1: Command RS = 1, Data RS = 0</td><td>RW</td></tr>
<tr><td>2</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>1</td><td>CLKPLY</td><td>LCD_CLK Polarity.
0: Active edge is Falling
1: Active edge is Rising</td><td>RW</td></tr>
<tr><td>0</td><td>TTYPE</td><td>Transfer Type.
0: Parallel
1: Serial</td><td>RW</td></tr>
</table>

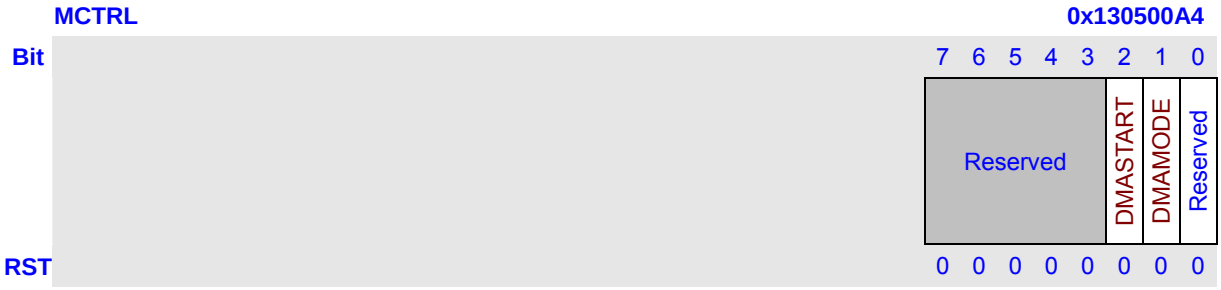
NOTE:

*1: The set of DWIDTH and CWIDTH should keep to the rules as follows:

Interface Mode	Command Width	Data Width	Color
Parallel	18-bit	18-bit once	R6G6B6
	16-bit	16-bit once	R5G6B5
		9-bit twice	--
	9-bit	9-bit twice	--
	8-bit	8-bit once	--
		8-bit twice	
		8-bit third times	--
Serial	18-bit	18-bit once	--
	16-bit	16-bit once	--
	9-bit	9bit twice	--
	8-bit	8-bit once	--
		8-bit twice	--
		8-bit third times	--

2.4.2 SLCD Control Register (MCTRL)

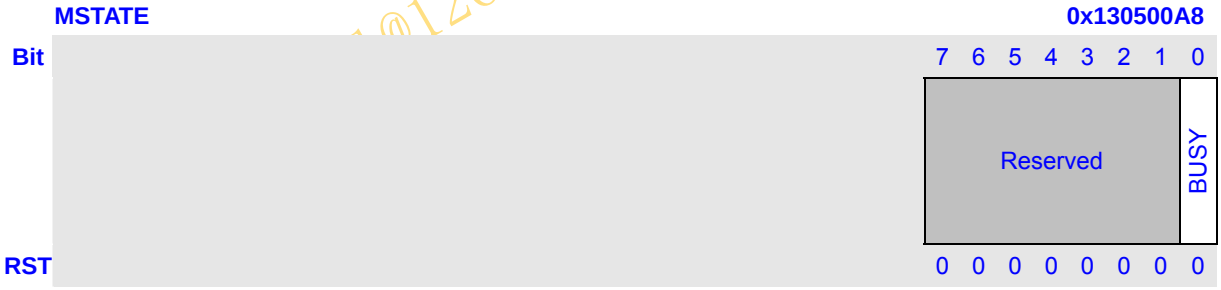
MCTRL is SLCD Control Register.



Bits	Name	Description	RW
7:3	Reserved	Writing has no effect, read as zero.	R
2	DMAMODE	SLCD descriptor DMA mode select. 0: DMA will continually transfer data follow descriptor chain 1: DMA will stop when one descriptor finished	RW
1	DMASTART	Only use when DMAMODE = 1, set 1 to restart DMA transfer.	RW
0	Reserved	Writing has no effect, read as zero.	R

2.4.3 SLCD Status Register (MSTATE)

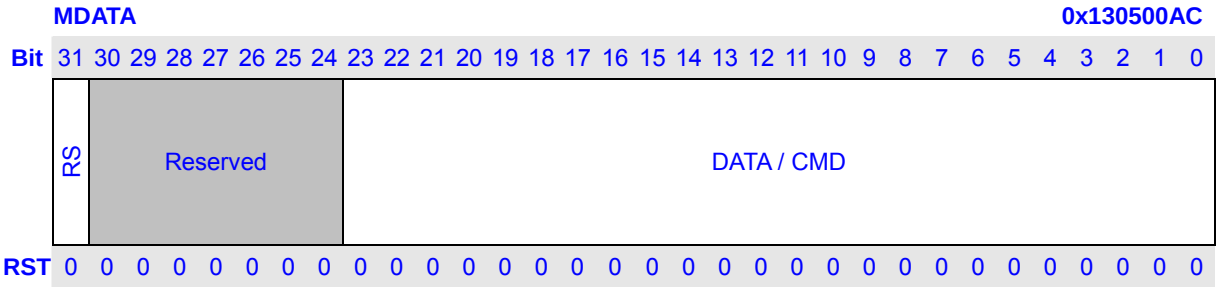
The register of MSTATE is SLCD status register.



Bits	Name	Description	RW
7:1	Reserved	Writing has no effect, read as zero.	R
0	BUSY	Transfer is working or not. This bit will be set to 1 when transfer is working. It will be cleared by hardware when transfer is finished. 0: not busy 1: busy	RW

2.4.4 SLCD Data Register (MDATA)

The register MDATA is used to send command or data to LCM. When RS=0, the low 24-bit is used as command. When RS=1, the low 24-bit is used as data.



Bits	Name	Description	RW
31	RS	The RS bit of data register is used to decide the meanings of the low 24-bit. 0: data 1: command	RW
30:24	Reserved	Writing has no effect, read as zero.	R
23:0	DATA/CMD	Data or Command Register.	RW

2.5 System Memory Format

2.5.1 Data format

You can configure these registers according to LCDC module.

2.5.2 Command Format

1 18-bit command

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
X	X	X	X	X	X	X	X	X	X	X	X	X	X	C17	C16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0

2 16-bit command

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
C15	C14	C13	C12	C11	C10	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0

3 9-bit command once

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
X	X	X	X	X	X	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
X	X	X	X	X	X	C9	C8	C7	C6	C5	C4	C3	C2	C1	C0

4 8-bit command once

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
C7	C6	C5	C4	C3	C2	C1	C0	C7	C6	C5	C4	C3	C2	C1	C0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
C7	C6	C5	C4	C3	C2	C1	C0	C7	C6	C5	C4	C3	C2	C1	C0

5 8-bit command twice (Command = command part + data part)

*Please notice that when you use this kind command, set CWIDTH as 8bit once and set the LCDCNUM.CNUM as doubled the real command number.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
D7	D6	D5	D4	D3	D2	D1	D0	C7	C6	C5	C4	C3	C2	C1	C0
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
D7	D6	D5	D4	D3	D2	D1	D0	C7	C6	C5	C4	C3	C2	C1	C0

2.6 Transfer Mode

Two transfer modes can be used: DMA/IPU Transfer Mode and Data Register Transfer Mode. In DMA/IPU mode, always transfer commands by DMA 0.

2.6.1 DMA Transfer Mode

Command and data can be recognized by RS bit coming from memory. The format of DMA transfer can be as follows:

1 Command and Data

C	D	D	C	C	D	D	D	C	C	C	D	D	D	D

*Please notice that the command only can insert between two complete frame and the number of command is 0~255.

2 Only Data

D	D	D	D	D	D	D	D	D	D	D	D	D	D	D

*You can also not use command but you still need to use a command descriptor and set the CNUM = 0.

Because DMA transfer mode only can work in OSD mode, you need to configure the panel according OSD mode:

1 Configuration.
* LCDCFG and LCDCTRL
* LCDOSDC and LCDOSDCTRL
* LCDRGBC and LCDIPUR

2 Set Color.
* LCDBGC, LCDKEY0, LCDKEY1, LCDALHPA

3 Set Display.
* LCDVAT, LCDDAH, LCDDAV
* LCDXYP0, LCDXYP1, LCDSIZE0, LCDSIZE1
* LCDVSYNC, LCDHSYNC

4 Set DMAC.
* LCDIID
* LCDDA0, LCDSA0, LCDFID0, LCDCMD0, LCDOFFS0, LCDPW0, LCDCNUM0, LCDDESSIZE0

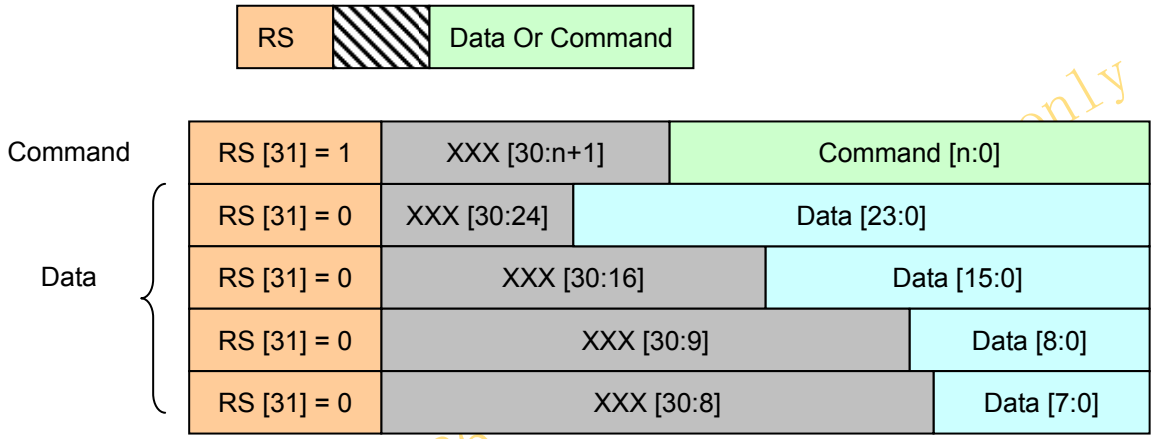
* LCDDA1, LCDSA1, LCDFID1, LCDCMD1, LCDOFFS1, LCDPW1, LCDCNUM1, LCDDESSIZE1

5 Enable slcd DMA.

6 Enable LCDC.

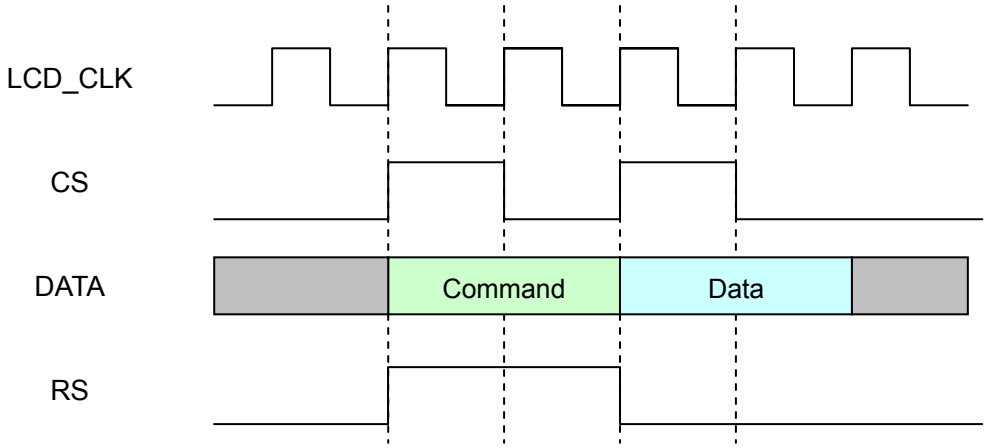
2.6.2 Register Transfer Mode

Each time you can write a command or a data to the register, then it will transfer the RS signal and data or command to LCM. Command and data can be recognized by RS bit coming from data register. The format of data register transfer can be as follows:

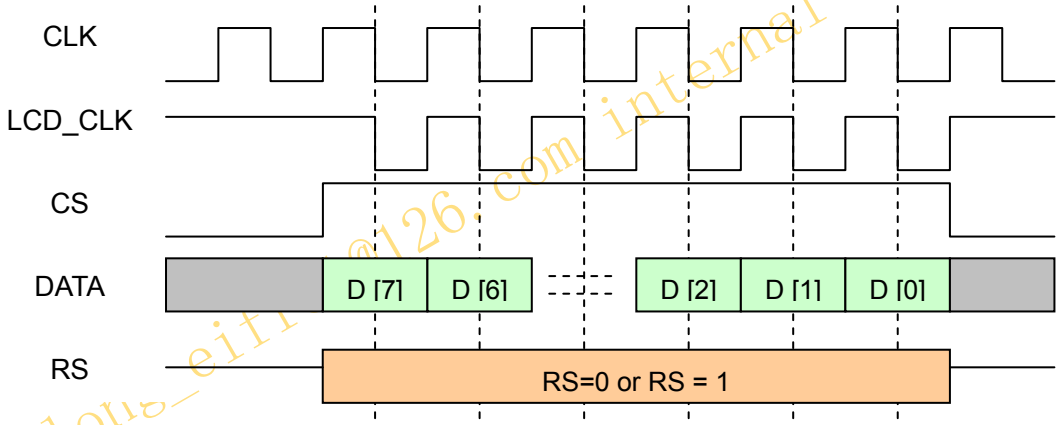


2.7 Timing

2.7.1 Parallel Timing



2.7.2 Serial Timing



2.8 Operation Guide

2.8.1 DMA Operation

1 Start DMA transfer.

a Set LCDCFG.MODE to 1101 to choose LCM.

b Set LCDCTRL.BST to choose burst length for transferring.

c Set register LCDIID0, LCDDA0, LCDSA0, LCDFID0, LCDCMD0, LCDOFFS0, LCDPW0, LCDCNUM0, LCDDESSIZE0 to initial internal DMA.

d Also set register LCDIID1, LCDDA1, LCDSA1, LCDFID1, LCDCMD1, LCDOFFS1, LCDPW1, LCDCNUM1, LCDDESSIZE1 when use DMA channel 1 in OSD mode.

e Set MCFG to configure SLCDC.

f Before starting DMA, Wait for MSTATE.BUSY == 0.

g Set MCTRL.DMATXEN to 1 to prepare DMA transfer.

Note that if you don't want to stop DMA transfer, you need not to check MSTATE.BUSY.

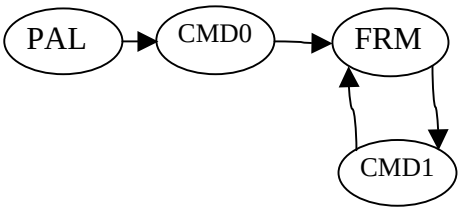
h Set LCDCCTRL.ENA to 1 to start LCDC internal DMA.

i The LCDC internal DMA will transfer data to SLCDC, and SLCDC transfer data to LCM.

Repeat this step till you want to close the SLCDC to transfer data to LCM Panel.

*please notice that use and only use DMA0 to transfer command no matter use DMA0 to transfer frame data or not.

One recommend descriptor chain (CMD0 with CNUM>0 and CMD1 with CNUM=0):



2 Stop DMA transfer.

a Set LCDCCTRL.ENA to 0 to stop LCDC internal DMA at once.

b Wait till MSTATE.BUSY is set to 0 by hardware.

MSTATE.BUSY == 1: there is data in the FIFO waited for transferring to LCM.

MSTATE.BUSY == 0: all data in the FIFO have finished transferring to LCM.

c Set MCTRL.DMATXEN to 0 to stop DMA transfer.

3 Restart DMA transfer.

When MCTRL.DMATXEN is set to 0, and then you want to restart DMA transfer at once, you should ensure that MCTRL.DMATXEN must keep 0 at least three cycles of PIXCLK.

2.8.2 Register Operation

1 Set MCFG to configure SLCD.

2 Wait for MSTATE.BUSY == 0.
3 Set MDATA register.
4 Wait for MSTATE.BUSY == 0.
5 Set MDATA register.
6 Wait for MSTATE.BUSY == 0.
7 … …

3 Decompresser

3.1 Overview

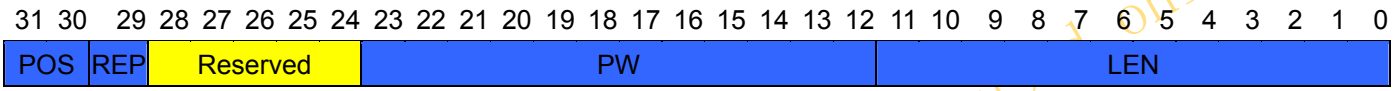
- Support bpp16 compressed data
- Support bpp24 compressed data with alpha
- Support bpp24 compressed data without alpha

3.2 Compress Method

Components

There are two kinds of Components in this method , one is command, the other is data .

Command:



Bits	Name	Description
31:30	POS	Position flag, indicates this command is in the head or middle or end of a giving line. Value / Description: 2'b00 Reserved 2'b01 This is the last command of the line 2'b10 This is sequent command of the line 2'b11 The is the first command of the line
29	REP	Repeated flag, indicates the following data is/are repeated or not , in another word , indicates the following data is/are compressed or original. 0: repeated 1: non-repeated
28:24	Reserve	All zero.
23:12	PW	Page width of current line. In the start cmd, it Indicates how many words in this line. In other cmd, it is zero.
11:0	LEN	Length , in repeated case, it indicates how many times the following data repeated; in non-repeated case , it indicates the number of following data frames. *in the case of compressed without alpha bpp24, this should be all zero.

*In the case of compressed without alpha bpp24, there is just start command , no sequent command and last command, the length is the high 8bits of every data.

Data:

In the case of compressed without alpha bpp24, the length is the high 8bits of every data , other bits are original data.

In the case of compressed with alpha bpp24, the data is ARGB8888.

In the case of compressed bpp16, the data is RGB565 or RGB555, and should be word aligned.

Example

There is an example, in next page.

The first one of the following figures is compressed with alpha;

The second one of the following figures is compressed without alpha.

That is 7 x 4 picture.

The dummies will not be fetched by DMA.

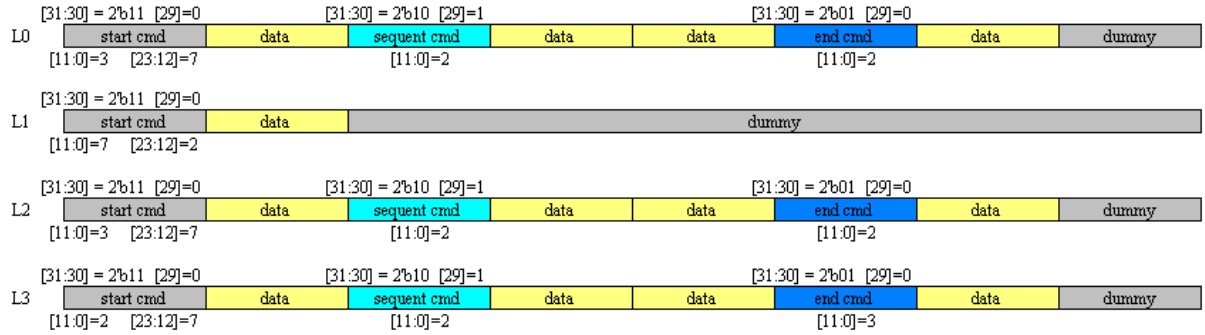
The first line contains 3 repeated pixels, 2 non-repeated pixels, and 2 repeated pixels;

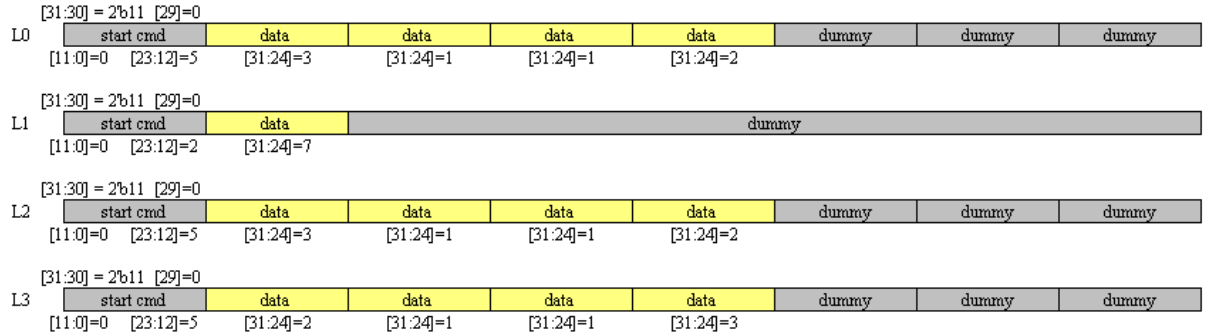
The second contains 7 repeated pixels;

The third line contains 3 repeated pixels , 2 non-repeated pixels, and 2 repeated pixels;

The last line contains 2 repeated pixels , 2 non-repeated pixels, and 3 repeated pixels.

:





3.3 Operation Guide

The routine operations are the same as normal operations except descriptor operation:

1 Uncomp_en (LCDCMDx[27])should be set if decompressing function is used.
2 Uncomp_md(LCDCMDx[26])should be set if the frame is bpp24 without alpha compressed frame, otherwise, should be clear.
3 LCDPWx don't need to set value.
4 LEN (LCDCMDx[11:0]) should be set to the LPP value of the frame.
5 LCDOFFSx should be set to how many word in per line of frame buffer for compressed frame, count in word, 64-word align or 16-word align(depend on the configuration of aosd_comp).

4 TV Encoder

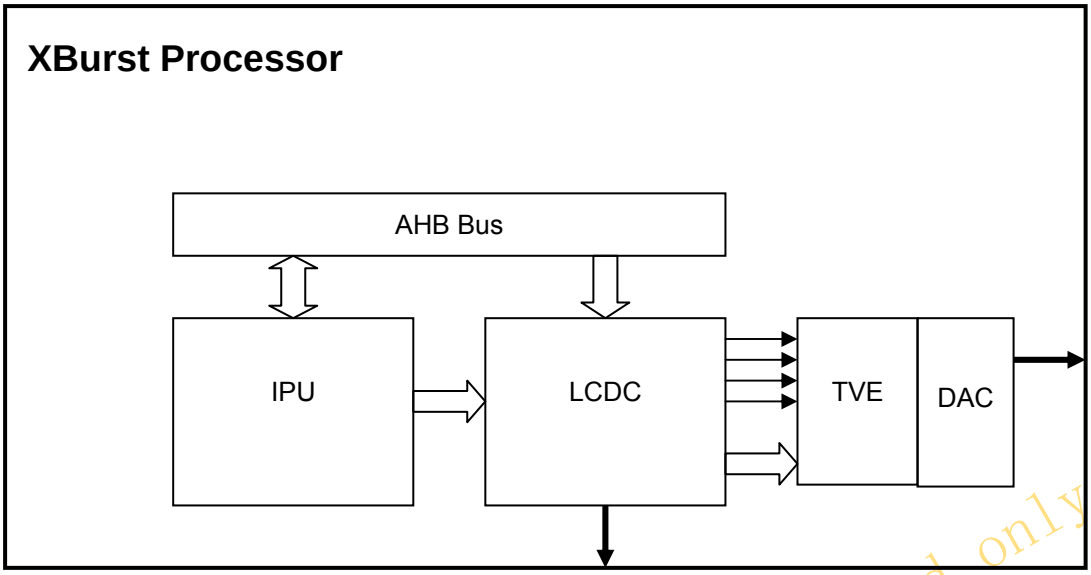
4.1 Overview

The TV Encoder enables the data for LCD panel showing in TV screen.
This release does not support S-video output.

Features:

- CVBS output
- PAL and NTSC supported

4.2 Structure



4.3 Pin Description

Table 4-1 TVE Pins Description

Name	I/O	Description	Interface
YCMP	AO	CVBS analog output	

4.4 Register Description

TVE memory mapped registers are put together with LCD controller, occupied address area of 'H13050140 ~ 'H130501FF. Following table lists all the registers definition. All register's 32bit address is physical address. And detailed function of each register will be described below.

Name	Description	RW	Reset Value	Address	Size
TVECR	TV Encoder Control register	RW	0x01040301	0x13050140	32
FRCFG	Frame configure register	RW	0x00170271	0x13050144	32
SLCFG1	TV signal level configure register 1	RW	0x0320011A	0x13050150	32
SLCFG2	TV signal level configure register 2	RW	0x012800F0	0x13050154	32
SLCFG3	TV signal level configure register 3	RW	0x00000048	0x13050158	32
LTCFG1	Line timing configure register 1	RW	0x00143F4E	0x13050160	32
LTCFG2	Line timing configure register 2	RW	0x05A0103D	0x13050164	32
CFREQ	Chrominance sub-carrier frequency configure register	RW	0x2A098ACB	0x13050170	32
CPHASE	Chrominance sub-carrier phase configure register	RW	0x00000001	0x13050174	32
CCFG	Chrominance filter configure register	RW	0x3B3B8989	0x13050178	32
WSSCR	Wide screen signal control register	RW	0x00000070	0x13050180	32
WSSCFG1	Wide screen signal configure register 1	RW	0x00000000	0x13050184	32
WSSCFG2	Wide screen signal configure register 2	RW	0x00000000	0x13050188	32
WSSCFG3	Wide screen signal configure register 3	RW	0x00000000	0x1305018C	32

4.4.1 TV Encoder Control Register (TVECR)

This register is used to control TV encoder.

TVECR **0x13050140**

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
	BLANK	SOGEN	SYNC	IBCTRL		Reserved	YUV	ECVBS	Reserved		CHSEL	DAPD	IREN	YCDLY			CGAIN		CBW		Reserved		SYNCT	PAL	FINV	ZBLACK	CR1ST	CLBAR	Reserved			SWRST
RST	0	0	1	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1

Bits	Name	Description	RW
31	BLANK	BLANK level control. 0: disable; 1: enable.	RW
30	SOGEN	Sync-On-Green enable. 0: disable; 1: enable.	RW

<table>
<tr><td>29</td><td>SYNC</td><td>SYNC level control.</td><td>RW</td></tr>
<tr><td>28:27</td><td>IBCTRL</td><td>Output current tuning. 00: -20%; 01:-10%; 10: 0%; 11:+10%.</td><td>RW</td></tr>
<tr><td>26</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>25</td><td>YUV</td><td>Keep this bit zero.</td><td>RW</td></tr>
<tr><td>24</td><td>ECVBS</td><td>Keep this bit one.</td><td>RW</td></tr>
<tr><td>23:22</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>21</td><td>CHSEL</td><td>DAC output enable. 0: disable; 1: enable.</td><td>RW</td></tr>
<tr><td>20</td><td>DAPD</td><td>DAC power down. When it is 0, power down all DACs.</td><td>RW</td></tr>
<tr><td>19</td><td>IREN</td><td>7.5 IRE pedestal control enable. 0: disable; 1: enable.</td><td>RW</td></tr>
<tr><td>18:16</td><td>YCDLY</td><td>(internal used only)</td><td>RW</td></tr>
<tr><td>15:14</td><td>CGAIN</td><td>Chrominance modulated signal gain factor setting when it is added to luminance signal in composite output format.
<table>
<tr><th>CGAIN</th><th>Description</th></tr>
<tr><td>00</td><td>1</td></tr>
<tr><td>01</td><td>1/4</td></tr>
<tr><td>10</td><td>1/2</td></tr>
<tr><td>11</td><td>3/4</td></tr>
</table></td><td>RW</td></tr>
<tr><td>13:12</td><td>CBW</td><td>Bandwidth setting for chrominance filter.
<table>
<tr><th>CBW</th><th>Description</th></tr>
<tr><td>00</td><td>Narrow band</td></tr>
<tr><td>01</td><td>Wide band</td></tr>
<tr><td>10</td><td>Extra wide band</td></tr>
<tr><td>11</td><td>Ultra wide band</td></tr>
</table></td><td>RW</td></tr>
<tr><td>11:10</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>9</td><td>SYNCT</td><td>Choose the sequence of field synchronizing pulses duration.
<table>
<tr><th>SYNCT</th><th>Description</th></tr>
<tr><td>0</td><td>The duration of sequence of field synchronizing pulses is 3 H, where H is a line period. Set SYNCT to this for NTSC TV set</td></tr>
<tr><td>1</td><td>The duration of sequence of field synchronizing pulses is 2.5 H. Set SYNCT to this for PAL TV set</td></tr>
</table></td><td>RW</td></tr>
<tr><td>8</td><td>PAL</td><td>Set this to 1 for PAL TV set, 0 for NTSC TV set.</td><td>RW</td></tr>
<tr><td>7</td><td>FINV</td><td>When this bit is 1, invert top and bottom fields.</td><td>RW</td></tr>
<tr><td>6</td><td>ZBLACK</td><td>Black of luminance (Y) input is 0. Set this bit to 1 if the input video luminance data for black is 0. Set this bit to 0 if the input video luminance data for black is 16. When this bit is 0, the Y input data will be clamped to ≥ 16.</td><td>RW</td></tr>
<tr><td>5</td><td>CR1ST</td><td>This bit described the Cb and Cr data order in input video.
<table>
<tr><th>ECVBS</th><th>Description</th></tr>
<tr><td>0</td><td>Cb comes before Cr, which is ITU656 standard</td></tr>
</table></td><td>RW</td></tr>
</table>

		1	Cr comes before Cb
4	CLBAR	Color bar mode. In this mode, a color bar picture is output to TV. **CLBAR** / **Description** 0 / Output input video to TV 1 / Output color bar to TV	RW
3:1	Reserved	Writing has no effect, read as zero.	R
0	SWRST	Software reset. When set this bit to 1, TVE is reset.	RW

4.4.2 Frame configure register (FRCFG)

This register is used to configure line in a frame.

FRCFG **0x13050144**

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								L1ST								Reserved						NLINE									
RST	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	1	0	0	1	1	1	0	0	0	1

Bits	Name	Description	RW
31:24	Reserved	Writing has no effect, read as zero.	R
23:16	L1ST	This field defines the first active video line of a field. The reset value is 23 in decimal. The frame active video line number is (NLINE – 1 – 2 * L1ST). The top and bottom field line number is a half of the frame line number.	RW
15:10	Reserved	Writing has no effect, read as zero.	R
9:0	NLINE	This field defines number of lines per-frame. The reset value is **625** in decimal.	RW

4.4.3 Signal level configure register 1, 2 and 3 (SLCFG1, SLCFG2, SLCFG3)

These registers are used to configure the TV signal level in difference phases.

SLCFG1 **0x13050150**

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								WHITEL								Reserved						BLACKL									
RST	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	0

Bits	Name	Description	RW
31:26	Reserved	Writing has no effect, read as zero.	R

25:16	WHITEL	Signal level for white color. The reset value is 800 in decimal.	RW
15:10	Reserved	Writing has no effect, read as zero.	R
9:0	BLACKL	Signal level for black color. The reset value is 282 in decimal.	RW

SLCFG2 **0x13050154**

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								VBLANKL								Reserved								BLANKL							
RST	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0

Bits	Name	Description	RW
31:26	Reserved	Writing has no effect, read as zero.	R
25:16	VBLANKL	Signal level in vertical blank period. The reset value is 296 in decimal.	RW
15:10	Reserved	Writing has no effect, read as zero.	R
9:0	BLANKL	Signal level in other blank period. The reset value is 240 in decimal.	RW

SLCFG3 **0x13050158**

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																								SYNCL							
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	1	0	0	1	0	0	0

Bits	Name	Description	RW
31:8	Reserved	Writing has no effect, read as zero.	R
7:0	SYNCL	Signal level in sync period. The reset value is 72 in decimal.	RW

4.4.4 Line timing configure register 1 and 2 (LTCFG1, LTCFG2)

These registers are used to configure timing period in a line.

LTCFG1 **0x13050160**

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											FRONTP					Reserved	HSYNCW							Reserved	BACKP						
RST	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	1	1	1	1	0	1	0	0	1	1	1	0

Bits	Name	Description	RW
31:21	Reserved	Writing has no effect, read as zero.	R

20:16	FRONTP	Front porch width, 16 cycles of 13.5MHz for 525 line system and 20 cycles for 625 line.	RW
15	Reserved	Writing has no effect, read as zero.	R
14:8	HSYNCW	HSYNC width in cycles of 13.5MHz. The reset value is 63 in decimal.	RW
7	Reserved	Writing has no effect, read as zero.	R
6:0	BACKP	Back porch width in cycles of 13.5MHz. The reset value is 78 in decimal.	RW

LTCFG2 **0x13050164**

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					ACTLIN											Reserved			PREBW					Reserved			BURSTW					
RST	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	0	1

Bits	Name	Description	RW
31:27	Reserved	Writing has no effect, read as zero.	R
26:16	ACTLIN	Active line length in cycles of 27MHz. The reset value is 1440 in decimal, which represent 720 pixels per line.	RW
15:13	Reserved	Writing has no effect, read as zero.	R
12:8	PREBW	Pre-burst width. The width after HSYNC and before the burst signals of back porch in cycles of 27MHz. The reset value is **16** in decimal.	RW
7	Reserved	Writing has no effect, read as zero.	R
6:0	BURSTW	The sub-carrier burst width inside back porch in cycles of 27MHz. The reset value is **61** in decimal.	RW

4.4.5 Chrominance configure registers (CFREQ, CPHASE, CFCFG)

This register is used to define chrominance sub-carrier frequency.

CFREQ **0x13050170**

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
	CFREQ																															
RST	0	0	1	0	1	0	1	0	0	0	0	0	1	0	0	1	1	0	0	0	1	0	1	0	1	1	0	0	1	0	1	1

Bits	Name	Description	RW
32	CFREQ	Chrominance sub-carrier frequency.	RW

This register is used to define chrominance sub-carrier phase.

CPHASE **0x13050174**

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
	INITPH								ACTPH								Reserved															CCRSTP	
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	1

Bits	Name	Description	RW
31:24	INITPH	Initial phase of chrominance sub-carrier. Corresponding to upper 8 bits of CFREQ.	RW
23:16	ACTPH	This is added to chrominance sub-carrier angle (corresponding to upper 8 bits of CFREQ) in case of active video period.	RW
15:2	Reserved	Writing has no effect, read as zero.	R
1:0	CCRSTP	Chrominance clock reset period. After the reset, chrominance clock is set to INITPH. Besides this, chrominance clock is reset also to INITPH in case of chip reset.	RW

CCRSTP	Description
00	Every 8 field
01	Every 4 fields
10	Every 2 lines
11	Never

This register is used to configure chrominance filter.

CCFG **0x13050178**

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
	CBBA								CRBA								CBGAIN								CRGAIN							
RST	0	0	1	1	1	0	1	1	0	0	1	1	1	0	1	1	1	0	0	0	1	0	0	1	1	0	0	0	1	0	0	1

Bits	Name	Description	RW
31:24	CBBA	Cb amplitude for burst period. The reset value is 59 in decimal, which corresponding to 59*4 = 236 ≈ (WHITEL – BLANKL) * 4 / 10 (±10%) = 224 ± 22 ≈ (WHITEL – BLANKL) * 3 / 7 (±3%) = 240 ± 7	RW
23:16	CRBA	Cr amplitude for burst period. The reset value is 59 in decimal. In PAL mode CRBA value is 59 in decimal and in NTSC mode CRBA value is 0 in decimal.	RW
15:8	CBGAIN	Cb gain. The reset value is 137 in decimal. CBGAIN=128 means no changing to the incoming Cb data.	RW
7:0	CRGAIN	Cr gain. The reset value is 137 in decimal. CRGAIN=128 means no changing to the incoming Cr data.	RW

4.5 Switch between LCD panel and TV set

LCD panel ➔ TV set switch

Step 1. Configure TVE (CVBS, N/P, and etc), enable DAC.

Step 2. Disable LCDC. If data is from IPU, stop IPU. Then LCD panel is turned off.

Step 3. Configure LCDC for output via TVE.

Step 4. Configure TVE and LCDC pixel clock and enable TVE clock (CPM).

Step 5. If data is from IPU, start IPU. Then start LCDC.

Step 6. Enable TVE (TVECR.SWRST=0). Then data stream from LCDC is output to TV set via TVE.

TV set ➔ LCD panel switch

Step 1. Disable TVE (TVECR.SWRST=1). Then no signal is output to TV set.

Step 2. Disable TVE clock (CPM), and disable DAC.

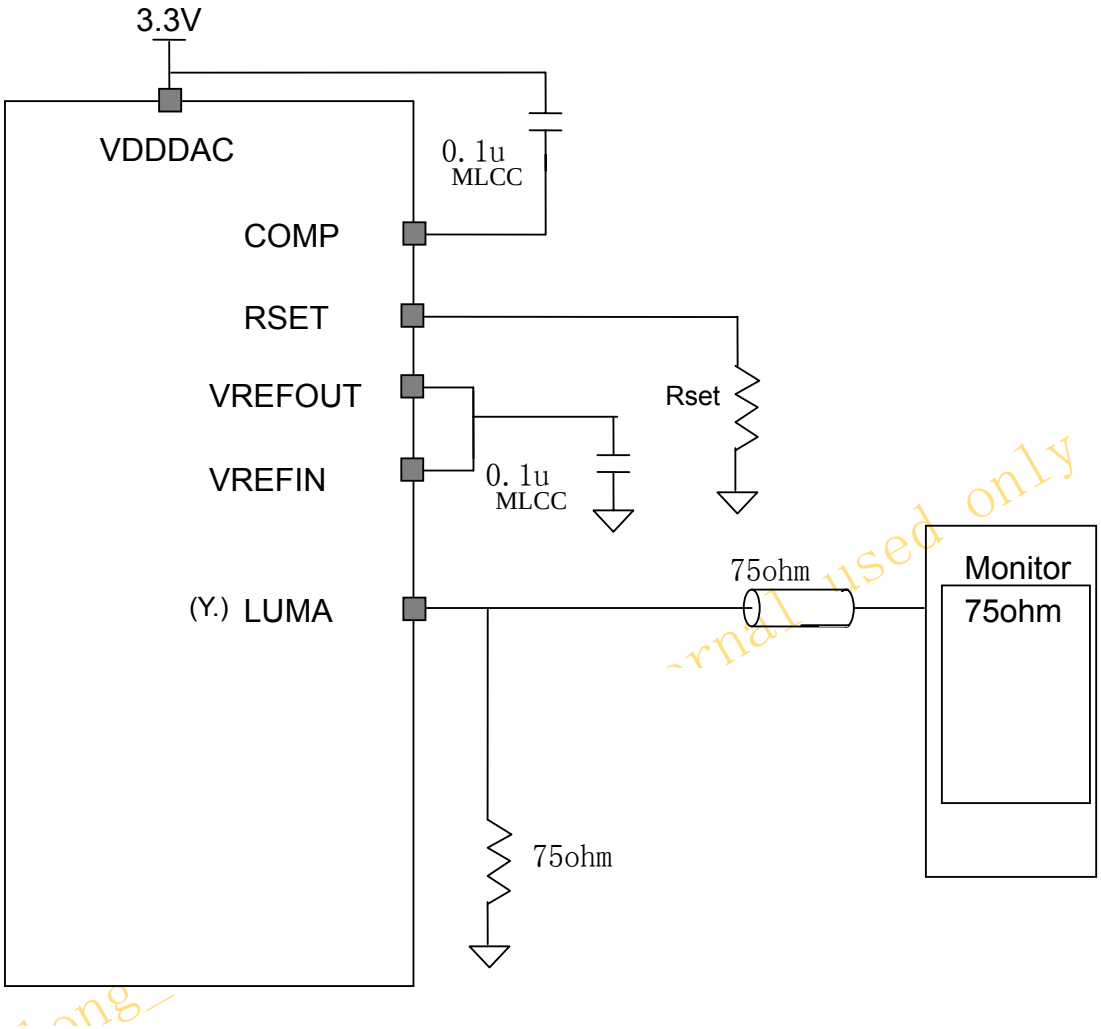
Step 3. Disable LCDC. If data is from IPU, stop IPU.

Step 4. Configure LCDC pixel clock. Configure LCDC for output to LCD panel.

Step 5. If data is from IPU, start IPU. Start LCDC. Then LCD panel is work.

4.6 DAC

4.6.1 DAC Connection



Full-Scale adjust resistor. A resistor (RSET) connected between this pin and AVSS controls the magnitude of the full-scale video signal. RSET (ohm) = 35k, where IOFS is full-scale output current per channel under the conditions of IREN =0, andSOGEN = 0.

4.6.2 DAC DC Character

VDDDAC =3.3V; DVDD=1.2V; RL=37.5ohm, CL=10Pf; Temp = 25℃.

Parameter	Symbol	Min	Type	Max	Unit
Operating voltage range	VDDDAC	3.0	3.3	3.6	V
Max output voltage	DVDD	1.08	1.20	1.32	V
DAC resolution		--	10	--	bits
Integral non-linearity error	INL	-1.2	-0.6~+0.6	+1.2	LSB
Differential non-linearity er	DNL	-1.2	-0.6~+0.6	+1.2	LSB

4.6.3 DAC Power Down Setup Time

As the output current's max value per channel is 34.1mA, keep the DAC power down when you not use TV encoder.

5 EPD Controller

5.1 Overview

The controller provides a low cost SOC solution for EPD applications.

Features:

- Supports PVI and AUO compatible EPD panels
- Supports different size up to 1024x768
- Supports 2/4/5 bits grayscale and color display
- Supports up to 8 multi-zone concurrent updating
- Supports hand-writing mode
- Supports SW LUT algorithm

5.2 EPDC Pin Mappings

Table 5-1 EPDC Pin Mapping

PVI	AUO	PIN
GDCLK	YCLK	LCD_VSYN_PC19
GDRL	UD	LCD_B0_LCD_REV_PC00
GDSP	YDIOU	LCD_B1_LCD_PS_PC01
GDOE	YOE	LCD_B2_PC02
SDCLK	XCLK	LCD_PCLK_PC08
SDOE		LCD_DE_PC09
SDLE	LD	LCD_HSYN_PC18
SDRL	RL	LCD_B3_PC03
SDCE [7]	YDIOD	LCD_R1_PC21
SDCE [6]	VCOM[1]	LCD_R0_LCD_CLS_UART4_RXD_PC20
SDCE [5]	VCOM[0]	CIM_D5_EPD_SCE5__PB15
SDCE [4]		CIM_D4_EPD_SCE4__PB14
SDCE [3]		CIM_D3_EPD_SCE3__PB1
SDCE [2]		CIM_D2_EPD_SCE2__PB12
SDCE [1]	XDIOR	LCD_G1_PC11
SDCE [0]	XDIOL	LCD_G0_LCD_SPL_UART4_TXD_PC10
SDDO [15]	DATA [15]	LCD_R7_PC27
SDDO [14]	DATA [14]	LCD_R6_PC26
SDDO [13]	DATA [13]	LCD_R5_PC25
SDDO [12]	DATA [12]	LCD_R4_PC24
SDDO [11]	DATA [11]	LCD_R3_PC23
SDDO [10]	DATA [10]	LCD_R2_PC22
SDDO [9]	DATA [9]	LCD_G7_PC17
SDDO [8]	DATA [8]	LCD_G6_PC16
SDDO [7]	DATA [7]	LCD_G5_PC15
SDDO [6]	DATA [6]	LCD_G4_PC14
SDDO [5]	DATA [5]	LCD_G3_PC13
SDDO [4]	DATA [4]	LCD_G2_PC12
SDDO [3]	DATA [3]	LCD_B7_PC07
SDDO [2]	DATA [2]	LCD_B6_PC06
SDDO [1]	DATA [1]	LCD_B5_PC05
SDDO [0]	DATA [0]	LCD_B4_PC04
PWRCOM		CIM_MCLK_EPD_PWC_PB09
PWR7	PWR7	AIC0_SDATO_EPD_PWR7_PE07
PWR6	PWR6	AIC0_SDATI_EPD_PWR6_PE06
PWR5	PWR5	LRCLK0_EPD_PWR5_PD13
PWR4	PWR4	UART3_RXD_BCLK0_EPD_PWR4_PD12

PWR3	PWR3	CIM_D7_EPD_PWR3_PB17
PWR2	PWR2	CIM_D6_EPD_PWR2_PB16
PWR1	PWR1	CIM_D1_EPD_PWR1_PB11
PWR0	PWR0	CIM_D0_EPD_PWR0_PB10
BD[3]	BD[3]	TSDI5_EPD_BD3_PB25
BD[2]	BD[2]	TSDI4_EPD_BD2_PB24
BD[1]	BD[1]	TSDI3_EPD_BD1_PB23
BD[0]	BD[0]	TSDI2_EPD_BD0_PB22

5.3 Function Block Diagram

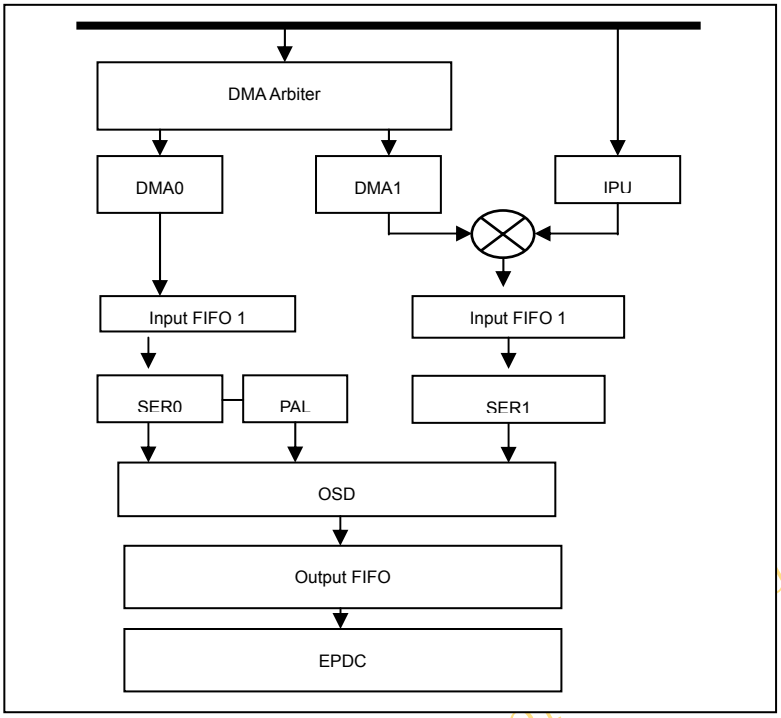


Figure 5-1 EPDC Function Diagram

5.4 EPD Controller Registers

Table 5-2 EPD Controller Registers

Name	Address	Reset Value	Access Size	RW
EPDC_CTRL	0x13050200	0x00000000	32	RW
EPDC_STA	0x13050204	0x00000000	32	R
EPDC_ISR	0x13050208	0x00000000	32	RW
EPDC_CFG0	0x1305020C	0x00000000	32	RW
EPDC_CFG1	0x13050210	0x00000000	32	RW
EPDC_PPL0	0x13050214	0x00000000	32	RW
EPDC_PPL1	0x13050218	0x00000000	32	RW
EPDC_VAT	0x1305021C	0x00000000	32	RW
EPDC_DAV	0x13050220	0x00000000	32	RW
EPDC_DAH	0x13050224	0x00000000	32	RW
EPDC_VSYN	0x13050228	0x00000000	32	RW
EPDC_HSYN	0x1305022C	0x00000000	32	RW
EPDC_GDCLK	0x13050230	0x00000000	32	RW
EPDC_GDOE	0x13050234	0x00000000	32	RW
EPDC_GDSP	0x13050238	0x00000000	32	RW
EPDC_SDOE	0x1305023C	0x00000000	32	RW
EPDC_SDSP	0x13050240	0x00000000	32	RW
EPDC_PMGR0	0x13050244	0x00000000	32	RW
EPDC_PMGR1	0x13050248	0x00000000	32	RW
EPDC_PMGR2	0x1305024C	0x00000000	32	RW
EPDC_PMGR3	0x13050250	0x00000000	32	RW
EPDC_PMGR3	0x13050254	0x00000000	32	RW
EPDC_VCOM0	0x13050258	0x00000000	32	RW
EPDC_VCOM1	0x1305025C	0x00000000	32	RW
EPDC_VCOM2	0x13050260	0x00000000	32	RW
EPDC_VCOM3	0x13050264	0x00000000	32	RW
EPDC_VCOM4	0x13050268	0x00000000	32	RW
EPDC_VCOM5	0x1305026C	0x00000000	32	RW
EPDC_BORDR	0x13050270	0x00000000	32	RW
EPDC_HWPAL	0x1305027C	0x00000000	32	RW
EPDC_PPL0_POS	0x13050280	0x00000000	32	RW
EPDC_PPL0_SIZE	0x13050284	0x00000000	32	RW
EPDC_PPL1_POS	0x13050288	0x00000000	32	RW
EPDC_PPL1_SIZE	0x1305028C	0x00000000	32	RW
EPDC_PPL2_POS	0x13050290	0x00000000	32	RW
EPDC_PPL2_SIZE	0x13050294	0x00000000	32	RW
EPDC_PPL3_POS	0x13050298	0x00000000	32	RW

EPDC_PPL3_SIZE	0x1305029C	0x00000000	32	RW
EPDC_PPL4_POS	0x130502A0	0x00000000	32	RW
EPDC_PPL4_SIZE	0x130502A4	0x00000000	32	RW
EPDC_PPL5_POS	0x130502A8	0x00000000	32	RW
EPDC_PPL5_SIZE	0x130502AC	0x00000000	32	RW
EPDC_PPL6_POS	0x130502B0	0x00000000	32	RW
EPDC_PPL6_SIZE	0x130502B4	0x00000000	32	RW
EPDC_PPL7_POS	0x130502B8	0x00000000	32	RW
EPDC_PPL7_SIZE	0x130502BC	0x00000000	32	RW

5.5 Registers Description

5.5.1 EPDC Control Registers

EPDC_CTRL **0x13050200**

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
	PPL7_FRM_INTE	PPL6_FRM_INTE	PPL5_FRM_INTE	PPL4_FRM_INTE	PPL3_FRM_INTE	PPL2_FRM_INTE	PPL1_FRM_INTE	PPL0_FRM_INTE	Reserved	FRM_VCOM_INTE	IMG_DONE_INTE	FRM_DONE_INTE	FRM_ABT_INTE	PWR_OFF_INTE	PWR_ON_INTE	DMA_DONE_INTE	PPL7_FRM_ENA	PPL6_FRM_ENA	PPL5_FRM_ENA	PPL4_FRM_ENA	PPL3_FRM_ENA	PPL2_FRM_ENA	PPL1_FRM_ENA	PPL0_FRM_ENA	IMG_RFE_ABT	IMG_REF_ENA	PWROFF	PWRON	Reserved		EPD_DMA_MODE	EPD_ENA
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	PPL7_FRM_INTE	PPL7_FRM_INT interrupt enable.	RW
30	PPL6_FRM_INTE	PPL7_FRM_INT interrupt enable.	RW
29	PPL5_FRM_INTE	PPL7_FRM_INT interrupt enable.	RW
28	PPL4_FRM_INTE	PPL7_FRM_INT interrupt enable.	RW
27	PPL3_FRM_INTE	PPL7_FRM_INT interrupt enable.	RW
26	PPL2_FRM_INTE	PPL7_FRM_INT interrupt enable.	RW
25	PPL1_FRM_INTE	PPL7_FRM_INT interrupt enable.	RW
24	PPL0_FRM_INTE	PPL7_FRM_INT interrupt enable.	RW
23	Reserved	Writing has no effect, read as zero.	R
22	FRM_VCOM_INTE	FRM_VCOM_INT interrupt enable.	RW
21	IMG_DONE_INTE	IMG_DONE_INT interrupt enable.	RW
20	FRM_DONE_INTE	FRM_DONE_INT interrupt enable.	RW
19	FRM_ABT_INTE	FRM_ABT_INT interrupt enable.	RW
18	PWR_OFF_INTE	PWR_OFF_INT interrupt enable.	RW
17	PWR_ON_INTE	PWR_ON_INT interrupt enable.	RW
16	DMA_DONE_INTE	DMA_DONE_INT interrupt enable.	RW
15	PPL7_FRM_ENA	Enable the 8th pipeline updating, cleared by HW when it finished.	RW
14	PPL6_FRM_ENA	Enable the 7th pipeline updating, cleared by HW when it finished.	RW
13	PPL5_FRM_ENA	Enable the 6th pipeline updating, cleared by HW when it finished.	RW
12	PPL4_FRM_ENA	Enable the 5th pipeline updating, cleared by HW when it finished.	RW
11	PPL3_FRM_ENA	Enable the 4th pipeline updating, cleared by HW when it finished.	RW
10	PPL2_FRM_ENA	Enable the 3rd pipeline updating, cleared by HW when it finished.	RW
9	PPL1_FRM_ENA	Enable the 2nd pipeline updating, cleared by HW when it finished.	RW
8	PPL0_FRM_ENA	Enable the 1st pipeline updating, cleared by HW when it finished.	RW
7	IMG_REF_ABT	Abort current image updating, cleared by HW when it finished.	RW
6	IMG_REF_ENA	Start to update the image, cleared by HW when it finished.	RW
5	PWROFF	Start the power off sequence, cleared by HW when it finished.	RW

4	PWRON	Start the power on sequence, cleared by HW when it finished.	RW
3:2	Reserved	Writing has no effect, read as zero.	R
1	EPD_DMA_MODE	1: The DMA will stop at the last frame data has been loaded; 0: The DMA will stop at end of each frame data has been loaded. It is available when DMAMODE in LCD MCTRL set to 1.	RW
0	EPD_ENA	Enable EPD controller.	RW

5.5.2 EPDC Status Register

EPDC_STA **0x13050204**

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																CUR_TOT_FRM																
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
31:16	Reserved	Writing has no effect, read as zero.	R
15:0	CUR_TOT_FRM	The numbers of frames have been processed in current operation, including border updating.	RW

5.5.3 EPDC ISR Register

EPD_ISR **0x13050208**

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																PPL7_FRM_INT	PPL6_FRM_INT	PPL5_FRM_INT	PPL4_FRM_INT	PPL3_FRM_INT	PPL2_FRM_INT	PPL1_FRM_INT	PPL0_FRM_INT	Reserved	FRM_VCOM_INT	IMG_DONE_INT	FRM_DONE_INT	FRM_ABT_INT	PWR_OFF_INT	PWR_ON_INT	DMA_DONE_INT
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
31:16	Reserved	Writing has no effect, read as zero.	R
15	PPL7_FRM_INT	The 8th pipeline interrupt.	RW
14	PPL6_FRM_INT	The 7th pipeline interrupt.	RW
13	PPL5_FRM_INT	The 6th pipeline interrupt.	RW
12	PPL4_FRM_INT	The 5th pipeline interrupt.	RW
11	PPL3_FRM_INT	The 4th pipeline interrupt.	RW
10	PPL2_FRM_INT	The 3rd pipeline interrupt.	RW
9	PPL1_FRM_INT	The 2nd pipeline interrupt.	RW

8	PPL0_FRM_INT	The 1st pipeline interrupt.	RW
7	Reserved	Writing has no effect, read as zero.	R
6	FRM_VCOM_INT	This interrupt will be asserted when frame counter reaches 95 at current updating. It means SW should update EPDC_VCOM0~5 if necessary. It is available when need more than 96 frames to update an image or text.	RW
5	IMG_DONE_INT	It interrupt will be asserted when all pipelines completed.	RW
4	FRM_DONE_INT	It interrupt will asserted when each frame.	RW
3	FRM_ABT_INT	This interrupt will be asserted when abort current display updating completed.	RW
2	PWR_OFF_INT	It interrupt will be asserted when perform power off sequence completed.	RW
1	PWR_ON_INT	This interrupt will be asserted when perform power on sequence completed.	RW
0	DMA_DONE_INT	This interrupt will be asserted when last frame data loaded by DMA.	RW

5.5.4 EPDC Configuration Register 0

EPDC_CFG0 0x1305020C

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
	DUAL_GATE	COLOR_MODE	COLOR_FMT			SDSP_CAS	SDSP_MODE	GDCLK_MODE	GDOE_MODE		GDUD	SDRL	GDCLK_POL	GDOE_POL	GDSP_POL	SDCLK_POL	SDOE_POL	SDSP_POL	SDCE_POL	SDLE_POL	Reserved		GDSP_CAS	Reserved					EPD_OBPP			EPD_OMODE
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	DUAL_GATE	Dual Gate Mode for some AUO EPD.	RW
30	COLOR_MODE	Color or mono mode selection.	RW
29:27	COLOR_FMT	Source Driver data format when in color mode.	RW
26	SDSP_CAS	It means output start pulse for Source Drivers cascading from XDIOR or XDIOL depending on data shift direction.	RW
25	SDSP_MODE	It means output start pulse for Source Drivers to start line a line data, or using chip enable for data sampling.	RW
24	GDCLK_MODE	0: GDCLK will be terminated when the last line data output, 1: GDCLK will last another line when outputted the last line data. For PVI EPD, it always should be set to 1; for AUO EPD, set it to 0 for normal mode and set it to 1 for fast mode.	RW
23:22	Reserved	Writing has no effect, read as zero.	R

21	GDUD	Up/Down pulse direction control and setting concatenating sequence for the Gate Driver.	RW
20	SDRL	Left/Right pulse direction control and setting concatenating sequence for the Source Driver.	RW
19	GDCLK_POL	Polarity of Gate Driver clock.	RW
18	GDOE_POL	Polarity of Gate Driver output enable.	RW
17	GDSP_POL	Polarity of Gate Driver start pulse.	RW
16	SDCLK_POL	Source Driver clock sample edge selection. 0: sample data at rising edge 1: sample data at falling edge	RW
15	SDOE_POL	Polarity of Source Driver output enable.	RW
14	SDSP_POL	Polarity of Source Driver start pulse.	RW
13	SDCE_POL	Polarity of Source Drivers chip enable.	RW
12	SDLE_POL	Polarity of Source Driver data latch signal.	RW
11:10	Reserved	Writing has no effect, read as zero.	R
9	GDSP_CAS	It means output start pulse for Gate Drivers cascading from YDIOU or YDIOD depending on data shift direction.	RW
8:4	Reserved	Writing has no effect, read as zero.	R
3:1	EPD_OBPP	Bits per pixel for Source Driver. 1: 2 bits per pixel 2: 4 bits per pixel others reserved	RW
0	EPD_OMODE	The Source Driver data bus width. 0: 8-bit 1: 16-bit	RW

5.5.5 EPDC Configuration Register 1

EPDC_CFG1 **0x13050210**

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	SDDO_REV	PDAT_SWAP	SDCE_REV	Reserved			SDOS									PDAT								SDCE_STN				SDCE_NUM			
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	Reserved	Writing has no effect, read as zero.	R
30	SDDO_REV	Set it to reverse pixels arrangement output to Source Driver. For instance, if SDDO_REV = 0, EPD_OBPP = 1, EPD_OMODE = 0, controller will output 4 pixels in each SDCLK in the format: [P3,	RW

		P2, P1, P0]. But if SDDO_REV = 1, then the output looks like [P0, P1, P2, P3].	
29	PDAT_SWAP	Swap padding data or not, using it with SDRL in following combinations: PDAT_SWAP = 0, SDRL = 0: padding after the display data PDAT_SWAP = 0, SDRL = 1: padding before the display data PDAT_SWAP = 1, SDRL = 0: padding before the display data PDAT_SWAP = 1, SDRL = 1: padding after the display data	RW
28	SDCE_REV	If Source Driver using chip enable, set it will reverse chips enable sequence, using it with SDCE_STN and SDCE_NUM. For instance, if SDCE_NUM = 4, SDCE_STN = 0, controller output chips enable [SDCE0, SDCE1, SDCE2, SDCE3] in order. But if SDCE_REV = 1, outputs will be [SDCE3, SDCE2, SDCE1, SDCE0].	RW
27:25	Reserved	Writing has no effect, read as zero.	R
24:16	SDOS	It is available when Source Driver using chip enable. SDOS = (Single Source Driver output size) / (Pixels per Clock)	RW
15:8	PDAT	Source Driver padding data, only available when Source Driver use chip enable. PDAT = (Source Driver Output Size * Number – Line Display Size) / (Pixels per Clock)	RW
7:4	SDCE_STN	Source Driver start number, only available when Source Driver use chip enables.	RW
3:0	SDCE_NUM	Source Driver total number, only available when Source Driver use chip enables.	RW

5.5.6 EPDC Pipeline Frame Register 0

EPDC_PPL0 **0x13050214**

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
	PPL3_FRM_NUM								PPL2_FRM_NUM								PPL1_FRM_NUM								PPL0_FRM_NUM							
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
31:24	PPL3_FRM_NUM	The 4th pipeline frame number.	RW
23:16	PPL2_FRM_NUM	The 3rd pipeline frame number.	RW
15:8	PPL1_FRM_NUM	The 2nd pipeline frame number.	RW
7:0	PPL0_FRM_NUM	The 1st pipeline frame number.	RW

5.5.7 EPDC Pipeline Frame Register 1

EPDC_PPL1 **0x13050218**

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
	PPL7_FRM_NUM								PPL6_FRM_NUM								PPL5_FRM_NUM								PPL4_FRM_NUM							
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
31:24	PPL7_FRM_NUM	The 8th pipeline frame number.	RW
23:16	PPL6_FRM_NUM	The 7th pipeline frame number.	RW
15:8	PPL5_FRM_NUM	The 6th pipeline frame number.	RW
7:0	PPL4_FRM_NUM	The 5th pipeline frame number.	RW

5.5.8 EPDC Virtual Display Area Setting Register

EPDC_VAT **0x1305021C**

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								VT								Reserved				HT											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	VT	The period of each frame in lines.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	HT	The period of each line in SDCLK.	RW

5.5.9 EPDC Vertical Display Area Setting Register

EPDC_DAV **0x13050220**

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								VDE								Reserved				VDS											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	VDE	The line number at which each frame displays data ends.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	VDS	The line number at which each frame displays data starts.	RW

5.5.10 EPDC Horizontal Display Area Setting Register

EPDC_DAH **0x13050224**

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				HDE												Reserved				HDS											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	HDE	The position at which each line displays data ends.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	HDS	The position at which each line displays data starts.	RW

5.5.11 EPDC Vertical Synchronous Start Pulse Setting

EPDC_VSYN **0x13050228**

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				VPE												Reserved				VPS											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	VPE	The last line number for Gate Driver generating start pulse.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	VPS	The first line number for Gate Driver generating start pulse.	RW

5.5.12 EPDC Horizontal Synchronous Start Pulse Setting

EPDC_HSYN **0x1305022C**

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				HPE												Reserved				HPS											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	HPE	The position at which Source Driver data latch signal de-asserts.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	HPS	The position at which Source Driver data latch signal asserts.	RW

5.5.13 EPDC Gate Driver Clock Setting Register

EPDC_GDCLK **0x13050230**

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	GDCLK_DIS	Reserved	GDCLK_ENA
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	GDCLK_DIS	The position at which Gate Driver clock signal de-asserts.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	GDCLK_ENA	The position at which Gate Driver clock signal asserts.	RW

5.5.14 EPDC Gate Output Enable Setting Register

EPDC_GDOE **0x13050234**

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	GDOE_DIS	Reserved	GDOE_ENA
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	GDOE_DIS	The position at which Gate Driver output enable signal de-asserts.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	GDOE_ENA	The position at which Gate Driver output enable signal asserts.	RW

5.5.15 EPDC Gate Driver Start Pulse Setting

EPDC_GDSP **0x13050238**

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	GDSP_DIS	Reserved	GDSP_ENA
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	GDSP_DIS	The position at which Gate Driver start pulse signal de-asserts.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	GDSP_ENA	The position at which Gate Driver start pulse signal asserts.	RW

5.5.16 EPDC Source Driver Output Enable Setting Register

EPD_SDOE **0x1305023C**

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				SDOE_DIS												Reserved				SDOE_ENA											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	SDOE_DIS	The position at which Source Driver output enable signal de-asserts.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	SDOE_ENA	The position at which Source Driver output enable signal asserts.	RW

5.5.17 EPDC Source Driver Start Pulse Setting Register

EPDC_SDSP **0x13050240**

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				SDSP_DIS												Reserved				SDSP_ENA											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	SDSP_DIS	The position at which Source Driver start pulse signal asserts.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	SDSP_ENA	The position at which Source Driver start pulse signal de-asserts.	RW

5.5.18 EPDC Power Management Registers 0

EPDC_PMGR0 **0x13050244**

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				PWR_DLY12												Reserved				PWR_DLY01											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	PWR_DLY12	The delay time in line between PWR [1] and PWR [2].	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	PWR_DLY01	The delay time in line between PWR [0] and PWR [1].	RW

5.5.19 EPDC Power Management Registers 1

EPDC_PMGR1 **0x13050248**

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				PWR_DLY34												Reserved				PWR_DLY23											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	PWR_DLY34	The delay time in lines between PWR [3] and PWR [4].	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	PWR_DLY23	The delay time in lines between PWR [2] and PWR [3].	RW

5.5.20 EPDC Power Management Registers 2

EPDC_PMGR2 **0x1305024C**

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				PWR_DLY56												Reserved				PWR_DLY45											
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	PWR_DLY56	The delay time in lines between PWR [5] and PWR [6].	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	PWR_DLY45	The delay time in lines between PWR [4] and PWR [5].	RW

5.5.21 EPDC Power Management Registers 3

EPDC_PMGR3 **0x13050250**

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
	VCOM_IDLE	PWRCOM_POL	UNI_POL	PPL7_BDR_ENA	BDR_LEVEL	BDR_IDLE	PWR_POL										Reserved				PWR_DLY67											
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	VCOM_IDLE	VCOM [1:0] default value when idle.	RW
29	PWRCOM_POL	The polarity of PWRCOM.	RW
28	UNIPOL	This bit choose PWRCOM or VCOM [1:0] as common voltage control signals of Source Driver. 0: VCOM [1:0] 1: PWRCOM	RW
27	PPL7_BDR_ENA	The 7th pipeline used for border updating or not.	RW
26	BDR_LEVEL	Border voltage control level setting. 0:2 bits 1:4 bits	RW
25:24	BDR_IDLE	Border voltage control signals default value when idle.	RW
23:16	PWR_POL	Polarity of PWR7~0.	R
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	PWR_DLY67	The delay time in lines between PWR [6] and PWR [7].	RW

5.5.22 EPDC Power Management Registers 4

EPDC_PMGR4 **0x13050254**

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								PWR_ENA								Reserved								PWR_VAL							
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
31:24	Reserved	Writing has no effect, read as zero.	R
23:16	PWR_ENA	These bits enable PWR7~0 individually.	RW
15:8	Reserved	Writing has no effect, read as zero.	R
7:0	PWR_VAL	The PWR [x] pin value individually if it is not enabled.	RW

5.5.23 EPDC VCOM Registers 0~5

EPDC_VCOM0, EPDC_VCOM1, EPDC_VCOM1 **0x13050258, 0x1305025C, 0x13050260**

EPDC_VCOM3, EPDC_VCOM4, EPDC_VCOM5 **0x13050264, 0x13050268, 0x1305026C**

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
	VCOM																															
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
31:0	VCOM	The VCOM [1:0] of each frame up to 16-frames, EPDC_VCOM0	RW

		contains the first 16 frames of VCOM[1:0]. EPDC_VCOM0: 16~1 EPDC_VCOM1: 32~17 EPDC_VCOM2: 48~33 EPDC_VCOM3: 64~49 EPDC_VCOM4: 80~65 EPDC_VCOM5: 96~81	

5.5.24 EPDC Border Voltage Setting Registers

	EPDC_BORDR																															0x13050270
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
	BORDR																															
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
31:0	BORDR	These bits set border voltage control signals in each frame. If BDR_LEVEL = 0, it contains up to 16 frames, but if BDR_LEVEL = 1, it contains only 8 frames.	RW

5.5.25 EPDC Handwriting Mode Setting

	EPDC_HWPAL																															0x1305027C
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
	PPL7_HW_MODE	PPL6_HW_MODE	PPL5_HW_MODE	PPL4_HW_MODE	PPL3_HW_MODE	PPL2_HW_MODE	PPL1_HW_MODE	PPL0_HW_MODE	PPL_HW_COLOR															PAL_ADDR_OFF										
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	PPL7_HW_MODE.	The 8th pipeline handwriting mode.	RW
30	PPL6_HW_MODE.	The 7th pipeline handwriting mode.	RW
29	PPL5_HW_MODE.	The 6th pipeline handwriting mode.	RW
28	PPL4_HW_MODE.	The 5th pipeline handwriting mode.	RW
27	PPL3_HW_MODE.	The 4th pipeline handwriting mode.	RW
26	PPL2_HW_MODE.	The 3rd pipeline handwriting mode.	RW
25	PPL1_HW_MODE.	The 2nd pipeline handwriting mode.	RW

24	PPL0_HW_MODE.	The 1st pipeline handwriting mode.	RW
23:9	PPL_HW_COLOR.	In color mode: PPL_HW_COLOR = { R[4:0], G[4:0], B[4:0] } In grayscale mode: PPL_HW_COLOR = { [4:0] } When a pipeline in handwriting mode, any pixels in its zone and match PPL_HW_COLOR will be updated using HW waveform.	RW
8:0	PAL_ADDR_OFF.	The address offset for dynamic changing palette.	RW

5.5.26 EPDC Pipeline 0 ~7 Position Registers

EPDC_PPL0_POS, EPDC_PPL1_POS, **0x13050280, 0x13050288**
EPDC_PPL2_POS, EPDC_PPL3_POS **0x13050290, 0x13050298**
EPDC_PPL4_POS, EPDC_PPL5_POS, **0x130502A0, 0x130502A8**
EPDC_PPL6_POS, EPDC_PPL7_POS **0x130502B0, 0x130502B8**

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	PPL_YPOS	Reserved	PPL_XPOS
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	PPL_YPOS	The top-left Y position of rectangle zone for the specific pipeline.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	PPL_XPOS	The top-left X position of rectangle zone for the specific pipeline.	RW

5.5.27 EPDC Pipeline 0~7 Size Registers

EPDC_PPL0_SIZE, EPDC_PPL1_ SIZE, **0x13050284, 0x1305028C**
EPDC_PPL2_ SIZE, EPDC_PPL3_ SIZE **0x13050294, 0x1305029C**
EPDC_PPL4_ SIZE, EPDC_PPL5_ SIZE, **0x130502A4, 0x130502AC**
EPDC_PPL6_ SIZE, EPDC_PPL7_ SIZE **0x130502B4, 0x130502BC**

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	PPL_HEIGHT	Reserved	PPL_WIDTH
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	PPL_HEIGHT	The height of rectangle zone for the specific pipeline.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	PPL_WIDTH	The width of rectangle zone for the specific pipeline.	RW

5.6 Application Guide

5.6.1 Pixel format in buffers

The format of texts and images stored in buffers is depending on how they're used. But it can be rearranged by setting PEDN and BEDN in LCDCTRL. Following tables illustrate the formats with different BPP.

Table 5-3 2 bits per pixel data buffer format

	pixel 7	**pixel 6**	**pixel 5**	**pixel 4**	**pixel 3**	**pixel 2**	**pixel 1**	**pixel 0**
FG0/1	15:14	13:12	11:10	9:8	7:6	5:4	3:2	1:0
	pixel 15	**pixel 14**	**pixel 13**	**pixel 12**	**pixel 11**	**pixel 10**	**pixel 9**	**pixel 8**
FG0/1	31:30	29:28	27:26	25:24	23:22	21:20	19:18	17:16

Table 5-4 4 bits per pixel data buffer format

	pixel 7	**pixel 6**	**pixel 5**	**pixel 4**	**pixel 3**	**pixel 2**	**pixel 1**	**pixel 0**
FG0	31:28	27:24	23:20	19:16	15:12	11:8	7:4	3:0
FG1	31:28	27:24	23:20	19:16	15:12	11:8	7:4	3:0

Table 5-5 5 bits per pixel data buffer format

		pixel 3		**pixel 2**		**pixel 1**		**pixel 0**
FG0/1	31:29	28:24	**23:21**	20:16	15:13	12:8	7:5	4:0

5.6.2 Waveform LUT Format

There are 3x512 words RAM embedded in LCD used for waveform LUT. The number of LUT frames it can store depending on COLOR_MODE, OBPP, OSDOBPP and EPD_OBPP. A frame LUT needs $2^{(\text{Bits per pixel for text or image} * 2) + (\text{Bits per pixel for source driver})}$ bits RAM in mono mode. For example, when COLOR_MODE = 0, OBPP = OSDBPP = 2, and EPD_OBPP = 1, a frame LUT needs 2^9 = 512 bits RAM. In other words, the whole RAM can store up to 96 frames LUT in this case. For multi-zone concurrent updating and handwriting mode, the topmost frame LUT reserved for "Do Noting LUT", and the second topmost frame reserved for "Handwriting Mode LUT".

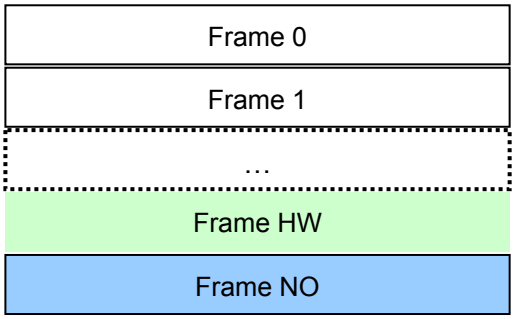


Figure 5-2 Mono mode frame LUT format

The following tables illustrate frame LUT content. The Index is generated by,
Index = {Pixel in FG1, Pixel in FG0}.For instance, when pixel in FG0 equals to 0x0 and pixel in FG1 equals to 0xf, then the green highlighted one will be addressed. Specially, the “Frame HW” and “Frame NO” index are generated by pipeline’s current frame count.

Table 5-6 Frame N LUT format

WORD	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
0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
15	255	254	253	252	251	250	249	248	247	246	245	244	243	242	241	240

Table 5-7 Frame HW/NO LUT format

WORD	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
0	Frame 15	…	…	…	…	…	…	…	…	…	…	…	…	…	…	Frame 0
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
15	Frame 255	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…

5.6.3 Power On/Off Sequence

The controller has individual bits to enable and set all power management signals’ polarity. This is useful for using high and low active power switches. All delay times are in lines, calculated by, (Delay between PWRn to PWRm) = PWR_DLYnm * (HT * (SDCLK Clock Period)).
Those power management pins delays which are not used should be set to 0. Please refer to vendor’s EPD displays panel data sheet to find out the required power on/off sequence for VDNS, VDPS, VDPG, VDNG, and VCOM_L and VCOM_H when using AUO EPD. Confirm EPD controller enabled and EPDC_PMGR0~3 settings correct before you start power on/off sequence by setting the bit EPD_PWRON in EPDC_CTRL.

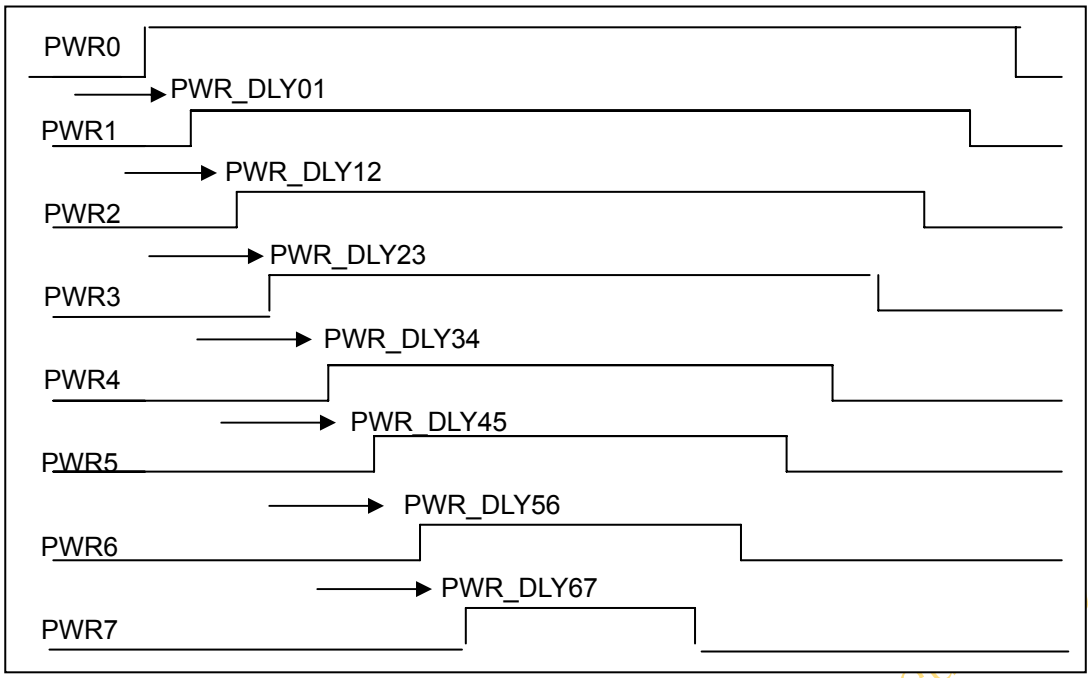


Figure 5-3 Powers On/Off Sequence

5.6.4 Display Timing Setting

The controller is designed as part of LCD controller. The frequency of pixel clock of the LCD is always double of the Source Driver clock. Typically, the frame frequency is 50 Hz in EPD display. Calculate timing parameters as followings,

Frame Rate = Line Number * Line Period, and Line number is set as VT.

Line Period = HT * (SDCLK Clock Period), SDCLK Clock Period = 2 * Pixel Clock Period.

So, first set pixel clock frequency according vendor's data sheet to get requirement SDCLK and frame rate. The following diagrams illustrate the Source and Gate Drivers' timing parameters for PVI and AUO EPD.

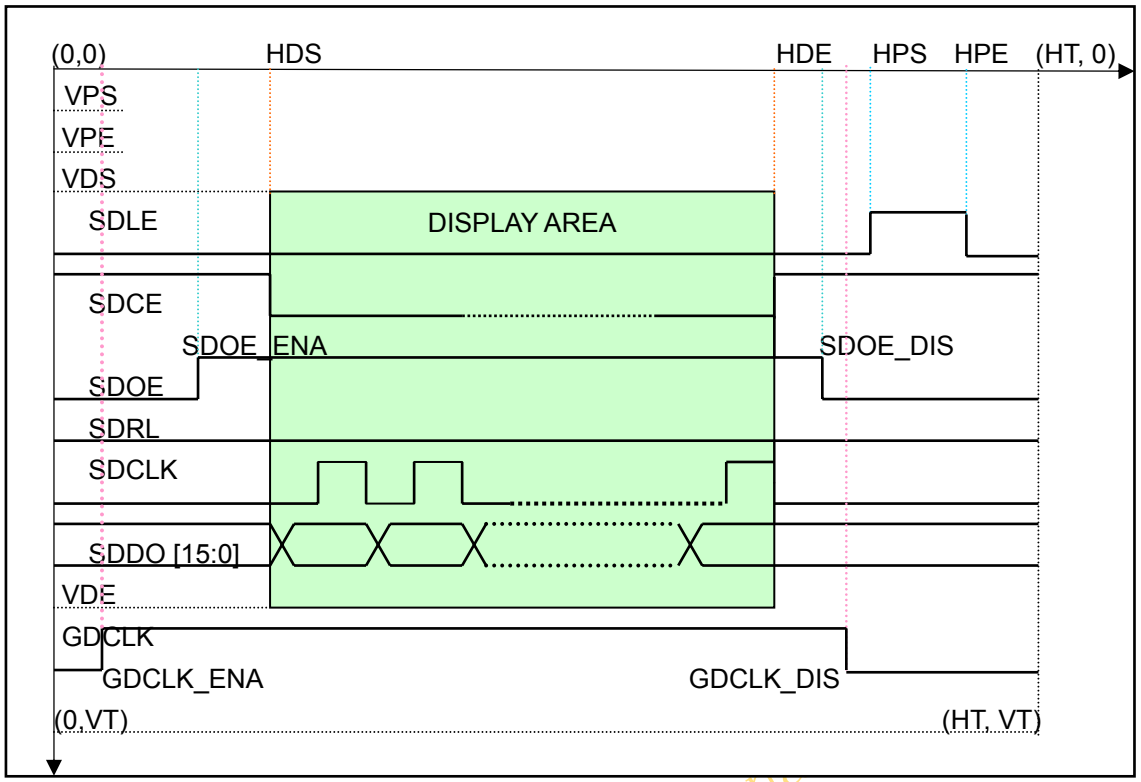


Figure 5-4 Source Drivers Reference Timing for PVI EPD

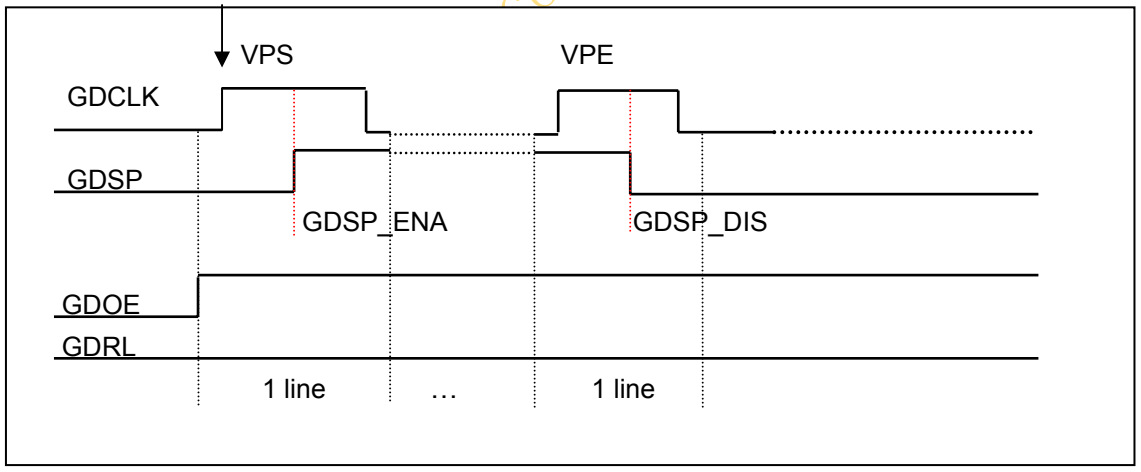


Figure 5-5 Gate Drivers Reference Timing for PVI EPD

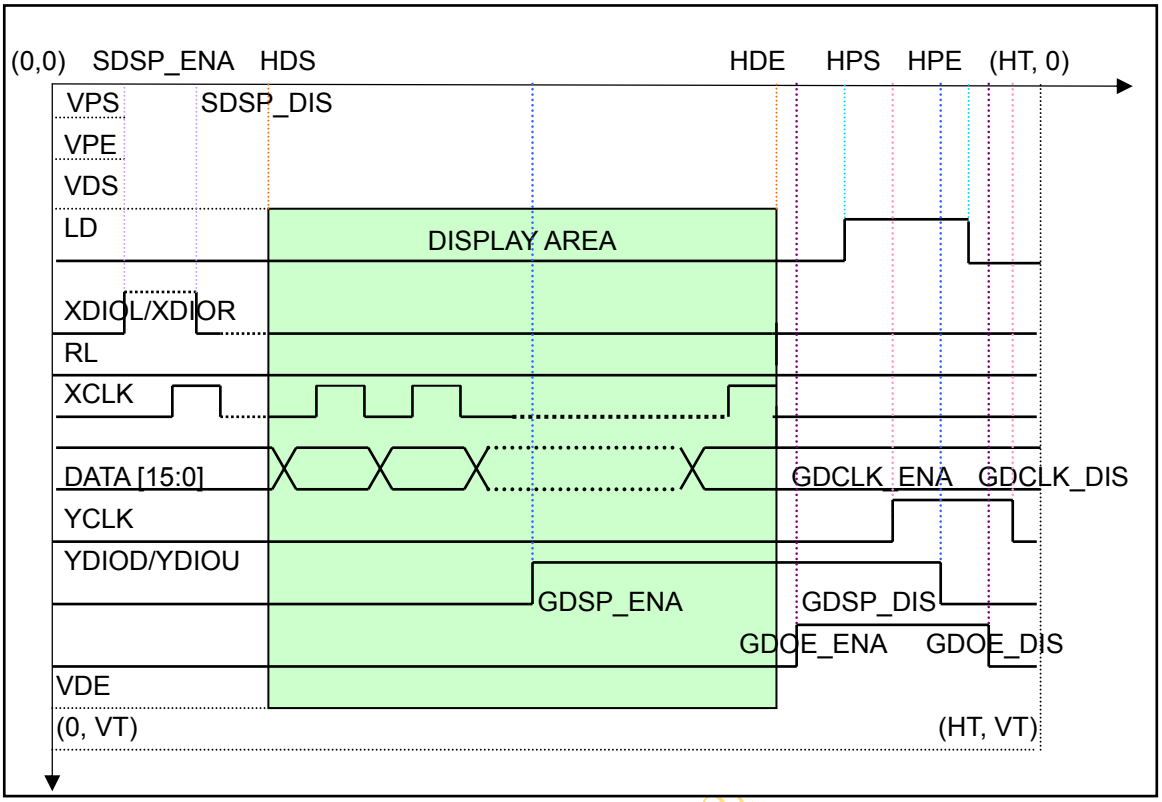


Figure 5-6 Source Drivers Reference Timing for AUO EPD

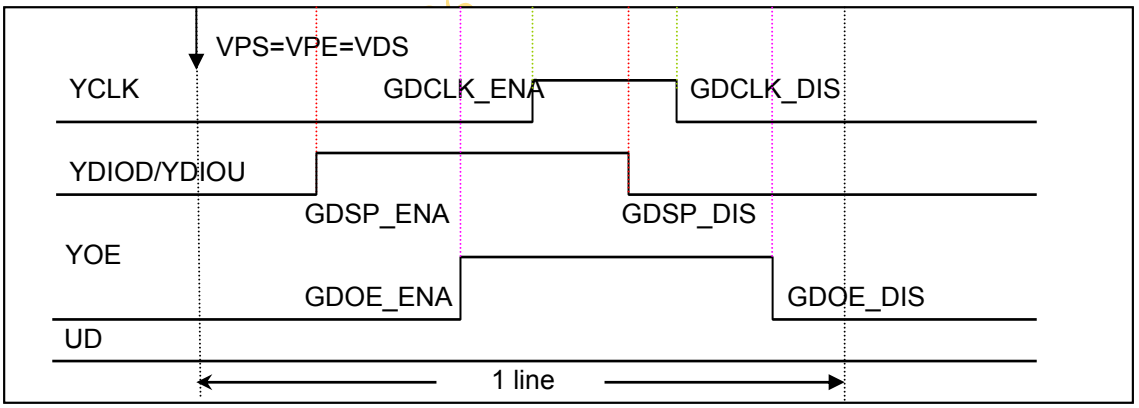


Figure 5-7 Gate Drivers Reference Timing for AUO EPD

5.6.5 Update image/text flow

1 Initialize LCD and EPD registers, such as display resolution, grayscale, and power on/off sequence requirements, source & gate driver's timings, according EPD data sheets.

2 Prepares LUT waveforms, display buffers and their descriptors relatively.

3 Set pipelines' sizes and frames if need multi-zone concurrent updating.

4 Starting EPD power on sequence, wait PWRON_INT assertion.

5 Start LCD DMA to transfer LUT and buffers data to controller and set PPL_FRM_ENA and IMG_REF_ENA to update image on EPD display.
6 If necessary, start power off sequence to reduce power.

5.6.6 Multi-zone concurrent updating

Waveform LUT for “Do Nothing LUT” frame should be filled before using multi-zone updating. If Source Driver using PWRCOM, just fill 0 to all words of it. But if using VCOM, fill it with VCOM values for all frames. If a pipeline is available, it can be start at any time by setting its frame and size. A pipeline’s frame count will increments from 0 to its PPL_FRM_NUM after it started. That is independent from others pipelines. There pipelines zones should not overlap, if not, pipeline 0 has higher priority than pipeline 1, and so on.

5.6.7 Update VCOM0~5

VCOM0~5 should be updated when need more than 96 frame to display an image or text. If FRM_VCOM_INTE enabled, a FRM_VCOM_INT interrupt will be asserted when controller has outputted 95 frame data.

5.6.8 Handwriting mode

Multi-zone pipelines can be used for handwriting. There are individual bits to enable each pipeline’s handwriting mode. If a pipeline used for handwriting, any pixels in FG1 buffer lies in its zone and matches PPL_HW_COLOR will be updated using “HW Mode LUT” waveform. If not matches PPL_HW_COLOR, it will be updated using “Do Nothing LUT”, means no change will happen for that pixel. Before starting handwriting, SW fills the “Do Nothing LUT” and “HW Mode LUT” properly, which are indexed by pipeline’s frame count.

5.6.9 Border Display

Some EPD want to display border, if necessary, the 7th pipeline can be used to update border.
Border updating can be used with images or texts display. Properly set EPDC_BORDR, BDR_LEVEL and BDR_IDLE, then set PPL7_BDR_ENA, the 7th frame number and start it.

6 Image Process Unit

6.1 Overview

IPU (Image process unit) contains Resize and CSC (color space conversion), which is used for image post processing.

6.1.1 Feature

- Location: AHB bus
- Input format
  - Separate frame: YUV /YCbCr (4:2:0, 4:2:2, 4:4:4, 4:1:1), RGB888
  - Packaged data: YUV422, RGB888, RGB565, RGB555, YUV444
  - Separate frame in block format: YUV/YCbCr 420
- Output data format
  - RGB (565, 555, 888, AAA)
  - Packaged data YUV422
- Color convention coefficient: configurable (CSC enable)
- Minimum input image size (pixel): 4x4
- Maximum input image size (pixel): 4095x4095
- Maximum output image size (pixel)
  - Width: up to 4095 (without vertical resizing)
    up to **2048** (with vertical resizing)
  - Height: up to 4095
- Image resizing
  - Support bilinear
  - 0 and bi-cube zooming mode
  - Up scaling ratios up to 1:31 in fractional steps with 1/32 accuracy
  - Down scaling ratios up to 31:1 in fractional steps with 1/32 accuracy

**For more details, refer to Special Instruction.*

6.2 Block

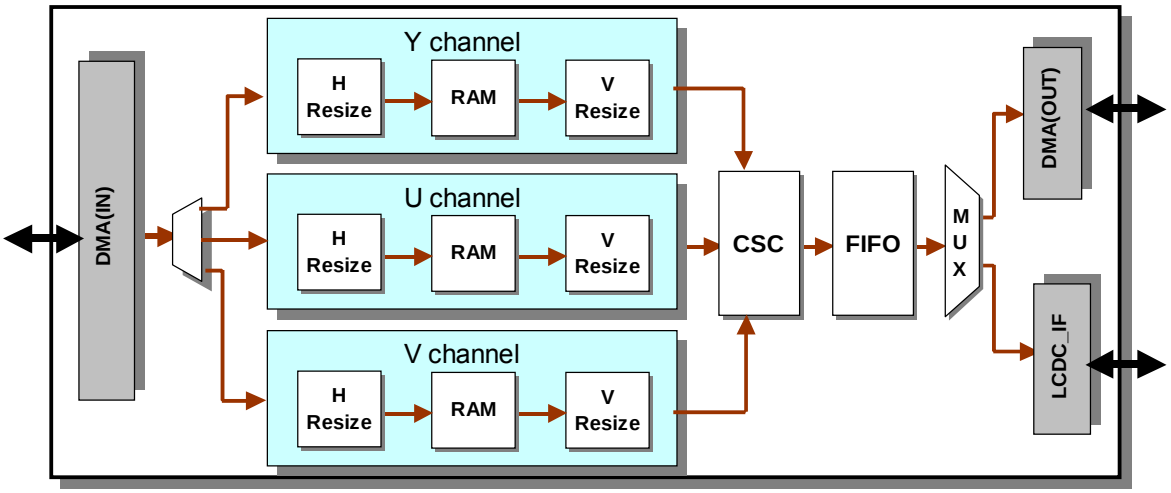


Figure 6-1 The Block about the IPUData flow

6.3 Data flow

6.3.1 Input data

- Separated YUV (or YcbCr/RGB888; the following use YUV for convenience) Frame case: Y, U, V data would be fetched from external memory by DMA burst read operation.
- Packaged YUV422 (or RGB888/ RGB565/ RGB555/YUV444) case: Packaged YUV(RGB888/ RGB565/ RGB555/YUV444) data would be fetched from external memory by DMA burst read operation.

6.3.2 Output data

- DMA output to external memory case: The output data format could be RGB (565, 555, 888) or YUV (package422), and the data would be stored to the external memory by DMA burst write operation.
- Flow into LCDC case: The output data format can be RGB or YUV (package), and the transfer would not cross AHB BUS.

6.3.3 Resize Coefficients LUT

The resize coefficients look up table is preset by software according to specific format (see for 6.4.30, 6.4.31 detail). There are 2 tables to support independent horizontal and vertical scaling. Each table has 32 entries that can accommodate up to 32 coefficient-groups.

6.4 Registers Descriptions

The physical address base for the address-mapped registers of IPU is **0x13080000.**
The following table will show all the register address.

Table 6-1 register list

NAME	Offset	Descript
IPU_CONTROL	0x0	IPU global controller
IPU_STATUS	0x4	IPU global status register
D_FMT	0x8	IPU data format register
Y_ADDR	0xC	Source Y (or R) base address
U_ADDR	0x10	Source U (or G) base address
V_ADDR	0x14	Source V (or B) base address
IN_FM_GS	0x18	Input Geometric Size (height and width)
Y_STRIDE	0x1C	Source Y frame stride
UV_STRIDE	0x20	Source U and V frame stride
OUT_ADDR	0x24	Result frame base address
OUT_GS	0x28	Result frame size (height and width)
OUT_STRIDE	0x2C	Result frame stride
RSZ_COEF_INDEX	0x30	Resize Coefficients Table Index
CSC_C0_COEF	0x34	Color conversion Coefficient
CSC_C1_COEF	0x38	Color conversion Coefficient
CSC_C2_COEF	0x3C	Color conversion Coefficient
CSC_C3_COEF	0x40	Color conversion Coefficient
CSC_C4_COEF	0x44	Color conversion Coefficient
HRSZ_COEF_LUT	0x48	Horizontal Resize Coefficients Look Up Table
VRSZ_COEF_LUT	0x4C	Vertical Resize Coefficients Look Up Table
CSC_OFSET_PARA	0x50	Color conversion offset Coefficient
SRC_TLB_ADDR	0x54	Base address of the source Y's physical address map table
DEST_TLB_ADDR	0x58	Base address of the destination's physical address map table
TLB_MONITOR	0x60	TLB monitor
IPU_ADDR_CTRL	0x64	IPU address set controller
Y_ADDR_N	0x84	Source Y base address of next frame
U_ADDR_N	0x88	Source U base address of next frame
V_ADDR_N	0x8C	Source V base address of next frame
OUT_ADDR_N	0x90	Result frame base address of next frame
SRC_TLB_ADDR_N	0x94	Base address of the source Y's physical address map table for next frame
DEST_TLB_ADDR_N	0x98	Base address of the destination's physical address map table for next frame
TLB_CTRL	0x68	TLB controller
PIC_ENC_TABLE	0x400 ~0X7FF	Picture enhance table stone.

6.4.1 IPU Control Register

IPU_CONTROL **0x0**

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							DMA_OPT	PENC_OPT	Reserved	CONF_MODE	ADDR_SEL	Reserved	ZOOM_SEL	DFIX_SEL	FIELD_SEL	FIELD_CONF	DISP_SEL	DPAGE_MAP	SPAGE_MAP	LCDC_SEL	SPKG_SEL	Reserved		IPU_STOP*4	IPU_RST	FM_IRQ_EN	CSC_EN	VRSZ_EN	HRSZ_EN	IPU_RUN	CHIP_EN
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	R/W
31:25	Reserved	Writing has no effect, read as zero.	R
24	DMA_OPT	DMA strategy evaluation: 0: disable 1: enable DMA strategy auto adjust	RW
23	PENC_OPT	Picture enhance selector: 0: no picture enhance 1: picture enhance(for YUV format)	RW
22	Reserved	Writing has no effect, read as zero.	R
21	CONF_MODE	IPU configure mode selector. 0: IPU's registers can be changed any time 1: IPU's registers only can be changed when it is not busy	RW
20	ADDR_SEL*3	IPU address mode selector. 0: IPU source and destination address only can be modified when IPU is free, just like it is in XBurst JZ4750 processor 1: IPU source and destination address can be modified anytime	RW
19	Reserved	Writing has no effect, read as zero.	R
18	ZOOM_SEL	IPU rooming mode selector. 1: bi-cube 0: bilinear	RW
17	DFIX_ SEL	Fixed destination address choose. (Valid when LCDC_SEL == 0) 0: not use the fixed address 1: use the fixed address	RW
16	FIELD_SEL *1	Destination field choose. (Valid when FIELD_CONF_EN == 1) 0: top field 1: bottom field	RW

15	FIELD_CONF_EN *1	FIELD_SEL mask. 0: do not change FIELD_SEL 1: configure FIELD_SEL Read as zero.	W
14	DISP_ SEL	Destination display choose. 0: frame display mode 1: field display mode	RW
13	DPAGE_MAP	Destination address page mapping choose. 0: not use the page mapping 1: use the page mapping	RW
12	SPAGE_MAP	Source address page mapping choose. 0: not use the page mapping 1: use the page mapping	RW
11	LCDC_SEL	Output data destination choose. 0: output to external memory 1: output to LCDC FIFO	RW
10	SPKG_SEL	Input data case choose. 0: Separated Frame 1: Packaged Frame	RW
9:8	Reserved	Writing has no effect, read as zero.	R
7	IPU_STOP*4	Stop IPU. 1: stop IPU. When stop IPU, the end flag will be pull up to 1.	W
6	IPU_RST *2	Reset IPU. Writing 1: reset IPU; 0: no effect. Read as zero.	W
5	FM_IRQ_EN	Frame process finish interrupt enable. 1: enable; 0: disable.	RW
4	CSC_EN	CSC enable. 1: enable; 0: disable.	RW
3	VRSZ_EN	Vertical Resize enable. 1: enable; 0: disable.	RW
2	HRSZ_EN	Horizontal Resize enable. 1: enable; 0: disable.	RW
1	IPU_RUN	Run the IPU. 1: run. Software just can set 1 to IPU_RUN.	RW
0	CHIP_EN	IPU chip enable. 1: enable; 0: disable.	RW

NOTES:

1 *[1]: The FIELD_SEL will work when the DISP_ SEL is 1, which indicates the IPU is under the field display mode. And the IPU will output the picture from the initial field (top or bottom) to the next field (bottom or top) automatically when the current field has been outputted or stopped. The initial field can be configured by setting the FIELD_SEL to 0 or 1 with FIELD_CONF_EN is 1. The FIELD_CONF_EN is just the trigger that controls the FIELD_SEL‘s valuation.

2 *[2]: Setting 1 to IPU_RST can reset all registers except the CHIP_EN immediately, but user must make sure the IPU is free when need to assert IPU_RST.

3 *[3]: When ADDR_SEL is set to 0, the address set method is the same as XBurst JZ4750 processor, and the frame address of IPU can be set just like the way in XBurst JZ4750 processor, which limits the address setting time to IPU none working period (after frame end-flag). When the ADD_SEL is 1, the above limitation is released. The addresses of IPU can be changed at anytime. It just needs to set the correspond bits in IPU_ADDR_CTRL to 1 to tell IPU that new address can be used, after the addresses are changed.

4 *[4]: The IPU_STOP is used to stop IPU in anytime in save mode. When the IPU_STOP has been written to 1, the IPU need some time to stop. And the user can monitor the IPU_STATUS.OUT_END to make sure the IPU has been stopped.

6.4.2 IPU Status Register

IPU_STATUS **0x4**

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																										MSTATUS		Reserved	SIZE_ERR	FMT_ERR	OUT_END
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	1	0	0	0	0	1

Bits	Name	Description	R/W
31:2	Reserved	Writing has no effect, read as zero.	R
5:4	MSTATUS	IPU main status. 00: IPU is free and waiting the CPU or LCDC to get the IPU's control 01: IPU is running now 10: IPU is under the control of CPU 11: reserved	R
3	Reserved	Writing has no effect, read as zero.	R
2	SIZE_ERR	The size error flag. 1: size error; 0: size ok.	R
1	FMT_ERR	IPU format error flag. 1: format error; 0: format OK.	R
0	OUT_END *1	Output termination flag. 1: finished; 0: not finished.	R/W

NOTE:

*[1]: If **IPU_CONTROL**.FM_IRQ_EN has been set 1, once OUT_END is set value 1 which denotes a frame's post process done, an low level active interrupt request will be issued until corresponding software handler read IPU_STATUS and clean end flag.

When the **IPU_CONTROL**.FM_LCDC_SEL has been set 1, and the IPU has finished one transfer, the LCDC and CPU need to occupy the IPU control. The IPU will monitor the request signal from LCDC and the read signal from the CPU, then it will determine whether dre-configure itself by the CPU if the CPU read first or output the same frame to LCDC again if the LCDC get the control.

6.4.3 IPU address control register

IPU_ADDR_CTRL **0x64**

Bit	31:6	5	4	3	2	1	0
	Reserved	PTD_RY	PTS_RY	D_RY	V_RY	U_RY	Y_RY
RST	0	1	0	0	0	0	1

Bits	Name	Description	R/W
31:6	Reserved	Writing has no effect, read as zero.	R
5	PTD_READY	New destination TLB base address ready. (Only used when IPU_CONTROL.ADDR_SEL==1&&IPU_CONTROL.DPAGE_MAP == 1)	RW
4	PTS_READY	New source TLB base address ready. (only used when IPU_CONTROL.ADDR_SEL==1&&IPU_CONTROL.SPAGE_MAP == 1)	RW
3	D_READY	New destination address ready.	RW
2	V_READY	New source V address ready.	RW
1	U_READY	New source U address ready.	RW
0	Y_READY	New source Y address ready.	RW

NOTES:

1 When the xx_READY bit has been set 1, the IPU will use the new corresponding address in the next frame, and use the old address value whose corresponding bit in **IPU_ADDR_CTRL** is 0.

2 **IPU_ADDR_CTRL** works when **IPU_CONTROL. ADDR_SEL** is 1.

3 When the **IPU** has fetched the new address, it will clear the **IPU_ADDR_CTRL** to 0.

6.4.4 Data Format Register

D_FMT **0x8**

Bit	31:26	25	24	23:22	21:20	19:18	17:16	15:6	5	4	3	2:1	0
	Reserved	RGB_888	RGB_OUT_OFT	OUT_FMT	YUV_PKG_OUT_OFT	Reserved	BLK_SEL	IN_OFT	IN_FMT				
RST	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	R/W
31:26	Reserved	Writing has no effect, read as zero.	R

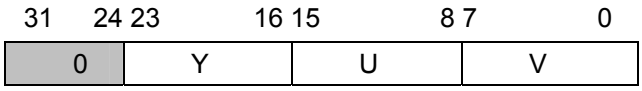
25	RGB_888_ OUT_FMT	RGB888 output format indicator. (only used in RGB888 out OUT_FMT == 010) 0: the low 24 bits will be the pixel in a word 1: the high 24 bits will be the pixel in a word	RW
24:22	RGB_OUT_ OFT	Output data packaged offset. (only used in RGB out OUT_FMT != 011) 000: RGB 001: RBG 010: GBR 011: GRB 100: BRG 101: BGR Others: reserved	RW
21:19	OUT_FMT	Indicates the destination data format. 000: RGB555 001: RGB565 010: RGB888 011: YUV422 package 100: RGBAAA(R(or G or B) is 10 bits wide)	RW
18:16	YUV_PKG_ OUT_OFT	Output data packaged offset. (only used in CSC disable case and in the source is YUV422 packaged case (**IPU_CONTROL.** SPKG_SEL == 1)) 000: Y1UY0V 001: Y1VY0U 010: UY1VY0 011: VY1UY0 100: Y0UY1V 101: Y0VY1U 110: UY0VY1 111: VY0UY1	RW
15: 6	Reserved	Writing has no effect, read as zero.	R
4	BLK_SEL	Indicate the source data format when source is YUV420	RW
3:2	IN_OFT	Input data packaged offset. (when the source is YUV422 packaged case (**IPU_CONTROL.** SPKG_SEL == 1 && IN_FMT == 01)) 00: Y1UY0V 01: Y1VY0U 10: UY1VY0 11: VY1UY0	RW
1:0	IN_FMT	Indicates the source data format. When **IPU_CONTROL.**SPKG_SEL == 0 00: YUV 4:2:0 01: YUV 4:2:2 10: YUV 4:4:4 11: YUV 4:1:1 When **IPU_CONTROL.**SPKG_SEL == 1 00: RGB555 01: YUV 4:2:2 10: RGB888 or YUV444 11: RGB565	RW

NOTES:

1 When the source frame is packed RGB, the **IPU_CONTROL.** SPKG_SEL must be 1 and

IN_FMT must be 10, and when the source frame is packed YUV, the **IPU_CONTROL.** SPKG_SEL must be 1 and IN_FMT must be 01.

2 When the source frame is packed YUV444, the data format about a pixel should be as following:



6.4.5 Input Y Data Address Register

Y_ADDR **0xc**

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
	Y_ADDR																														0	0
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	R/W
31:0	Y_ADDR *1	In separated Frame case, it indicates the source Y (or R) data buffer's start address. In source YUV422 package case, it indicates the start Address of the packaged Frame.	RW

NOTES:

1 When the **IPU_CONTROL**.SPAGE_MAP == 1, the Y_ADDR should be the start virtual address.

2 Y_ADDR should be **word align**.

6.4.6 Input U Data Address Register

U_ADDR **0x10**

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
	U_ADDR																														0	0
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	R/W
31:0	U_ADDR *1	The source U data buffer's start address of separated frame case.	RW

NOTES:

1 When the IPU_CONTROL.SPAGE_MAP == 1, the U_ADDR should be the start virtual address.

2 U_ADDR should be **word align**.

6.4.7 Input V Data Address Register

V_ADDR **0x14**

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
	V_ADDR																														0	0
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	R/W
31:0	V_ADDR	The source V data buffer's start address of separated Frame case.	RW

NOTES:

1 When the IPU_CONTROL.SPAGE_MAP == 1, the V_ADDR should be the start virtual address.

2 V_ADDR should be word align.

6.4.8 Input source TLB base address

SRC_TLB_ADDR **0x54**

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
	SRC_TLB_ADDR																														0	0
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	R/W
31:0	SRC_TLB_ ADDR	The SOURCE TLB base address. (This register will act when the IPU_CONTROL.PAGE_MAP==1)	RW

6.4.9 Destination TLB base address

DEST_TLB_ADDR **0x58**

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
	DEST_TLB_ADDR																														0	0
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	R/W
31:0	DEST_TLB_ ADDR	The destination frame's TLB base address. (This register will work when the IPU_CONTROL.DPAGE_MAP==1)	RW

6.4.10 TLB monitor

TLB_MONITOR **0x60**

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																					MIS_CNT										
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	R/W
31:11	Reserved	Writing has no effect, read as zero.	R
10:0	MIS_CNT	The TLB cache miss counter.	RW

6.4.11 TLB controller

TLB_CTRL **0x68**

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								DEST_ BURST				DEST_PAGE				Reserved								SRC_ BURST				SRC_PAGE			
RST	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Bits	Name	Description	R/W
31:25	Reserved	Writing has no effect, read as zero.	R
23:20	DEST_BURST	The destination TLB burst length.	RW

		1: 1; 2:2; 4:4; 8:8.	
19:16	DEST_PAGE	The destination TLB page size. 1: 4K; 2: 8K; 4:16K; 8:32K.	RW
15:4	Reserved	Writing has no effect, read as zero.	R
7:4	SRC_BURST	The sources TLB burst length. 1: 1; 2:2; 4:4; 8:8.	RW
3:0	SRC_PAGE	The source TLB page size. 1: 4K; 2: 8K; 4:16K; 8:32K.	RW

6.4.12 Input Y Data Address of next frame Register

Y_ADDR_N **0x84**

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
	Y_ADDR_N																														0	0
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	R/W
31:0	Y_ADDR_N	In separated Frame case, it indicates the source Y (or R) data buffer's start address of the next frame. In package case, it indicates the start address of the packaged Frame of the next frame.	R

6.4.13 Input U Data Address of next frame Register

U_ADDR_N **0x88**

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
	U_ADDR_N																														0	0
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	R/W
31:0	U_ADDR_N	The source U (G) data buffer's start address of the next frame in separated frame case.	RW

NOTE: When the IPU_CONTROL.SPAGE_MAP == 1, the U_ADDR_N will be the start virtual address.

6.4.14 Input V Data Address of next frame Register

V_ADDR_N **0x8C**

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
	V_ADDR_N																														0	0
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	R/W
31:0	V_ADDR_N	The source V (B) data buffer's start address of the next frame in separated frame case.	RW

NOTE: When the IPU_CONTROL.SPAGE_MAP == 1, the V_ADDR_N will be the start virtual address.

6.4.15 Source TLB base address of next frame

SRC_TLB_ADDR_N **0x94**

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
	SRC_TLB_ADDR_N																														0	0
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	R/W
31:0	SRC_TLB_ADDR_N	The TLB base address about the next source frame's data which will be DMA in.	R

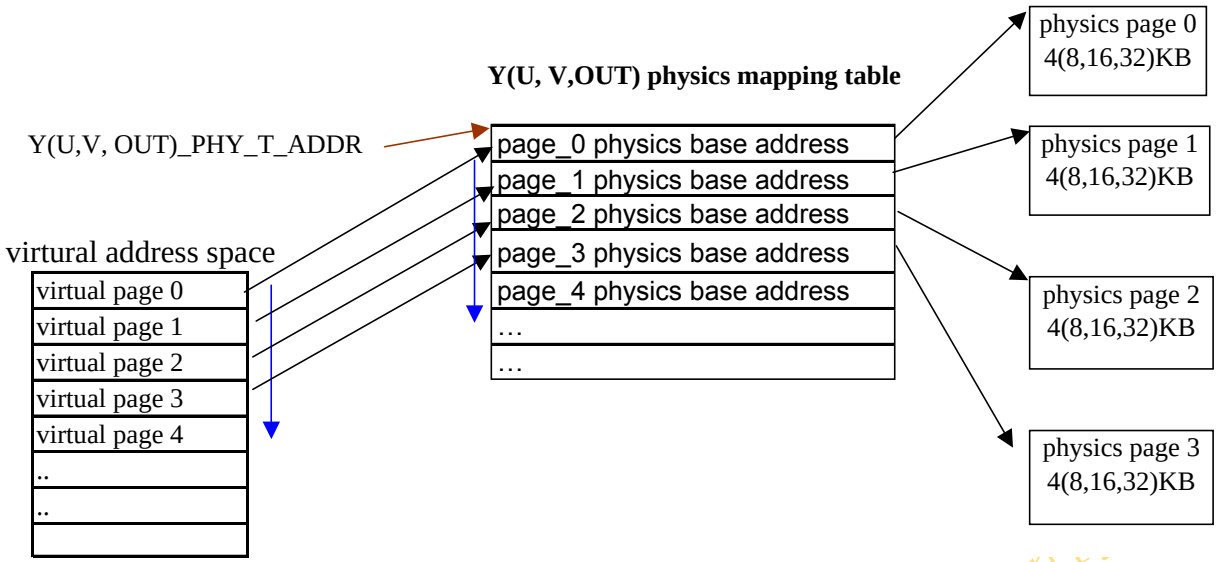
6.4.16 Destination TLB base address of next frame

DEST_TLB_ADDR_N **0x98**

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
	DEST_TLB_ADDR_N																														0	0
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	R/W
31:0	DEST_TLB_ADDR_N	The TLB base address about the next frame's data that will be DMA out.	R

6.4.17 ADDRESS Mapping



The Y (U, V, OUT)_PHY_T_ADDR should store the **base address** of the Y (U, V, OUT) physics-mapping table. In the Y (U, V, OUT) physics-mapping table, it should contain different physics page base address, and the physics page must be 4(8, 16, 32)KB align.

6.4.18 Input Geometric Size Register

IN_FM_GS **0x18**

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				IN_FM_W												Reserved				IN_FM_H											
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	R/W
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	IN_FM_W	The width of the input frame (unit: byte). Y data width is the same as this value while U/V or Cb/Cr data width should do relatively zoom in according to the source data format. And **in the source package pattern**, this value should be the Y data width also.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	IN_FM_H	The height of the input frame (unit: byte). Y data width is same as this value while U/V or Cb/Cr data width should do relatively zoom in according to the source data format.	RW

6.4.19 Input Y Data Line Stride Register

Y_STRIDE **0x1c**

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																Y_S														0	0
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	R/W
31:16	Reserved	Writing has no effect, read as zero.	R
15:0	Y_S	The line stride of the source Y data in the external memory of separated Frame case or packaged Frame stride(Unit: byte).	RW

NOTE: Y_S should be world align.

6.4.20 Input UV Data Line Stride Register

UV_STRIDE **0x20**

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							U_S									0	0	Reserved					V_S							0	0
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	R/W
31:29	Reserved	Writing has no effect, read as zero.	R
28:16	U_S	The line stride of the source U data in the external memory.	RW
15:13	Reserved	Writing has no effect, read as zero.	R
12:0	V_S	The line stride of the source V data in the external memory.	RW

NOTE: U_S and V_S should be word align and unit is byte.

6.4.21 Output Frame Start Address Register

OUT_ ADDR **0x24**

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
	OUT_ ADDR																														0	0
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	R/W
31:0	OUT_ ADDR *1	The output buffer's start address.	RW

NOTES:

1 *[1]: When the IPU_CONTROL.DPAGE_MAP == 1, the OUT_ADDR should be the start virtual address.

2 it should be word align.

6.4.22 Output Data Address of next frame Register

OUT_ ADDR_N **0x90**

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
	OUT_ ADDR_N																														0	0
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	R/W
31:0	OUT_ ADDR *1	The output buffer's start address.	RW

NOTES:

1 *[1]: When the IPU_CONTROL.DPAGE_MAP == 1, the OUT_ADDR should be the start virtual address.

2 it should be word align.

6.4.23 Output Geometric Size Register

OUT_GS **0x28**

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	OUT_FM_W															Reserved	OUT_FM_H														
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	R/W
31	Reserved	Writing has no effect, read as zero.	R
30:16	OUT_FM_W	The width of the output destination frame (unit: byte).	RW
15:13	Reserved	Writing has no effect, read as zero.	R
12:0	OUT_FM_H	The height of the output destination frame (unit: byte).	RW

NOTES:

1 In the package YUV out pattern, the OUT_FM_W should be the **pixel number** in a line.

2 In the **RGB** out pattern, the OUT_FM_W should be the **data space** width in the RAM.

3 In the out package pattern, the OUT_FM_W **must be** even number, and otherwise IPU will not run.

e.g.

Package YUV out: 480x273 (OUT_FM_W: 480 OUT_FM_H: 273)

RGB 888 (or AAA) out: 480x273 (OUT_FM_W: 480*4 OUT_FM_H: 273)

RGB 555 (or 565) out: 480x273 (OUT_FM_W: 480*2 OUT_FM_H: 273)

6.4.24 Output Data Line Stride Register

OUT_STRIDE **0x2c**

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																OUT_S														0	0
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	R/W
31:16	Reserved	Writing has no effect, read as zero.	R
15:0	OUT_S	The line stride of the destination data buffer in the external memory(Unit: byte).	RW

6.4.25 CSC C0 Coefficient Register

CSC_C0_COEF **0x34**

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																				C0_COEF											
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	R/W
31:12	Reserved	Writing has no effect, read as zero.	R
11:0	C0_COEF	The C0 coefficient of the YUV/YCbCr to RGB conversion. C0_COEF = [C0 * 1024 + 0.5].	RW

NOTE:
R = C0*(Y –LUMA_OF) + C1*(Cr-CHROM_OF)
G = C0*(Y – LUMA_OF) – C2*(Cb-CHROM_OF) – C3*(Cr-CHROM_OF)
B = C0*(Y – LUMA_OF) + C4*(Cb-CHROM_OF)

6.4.26 CSC C1 Coefficient Register

CSC_C1_COEF **0x38**

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																				C1_COEF												
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	R/W
31:12	Reserved	Writing has no effect, read as zero.	R
11:0	C1_COEF	The C1 coefficient of the YUV/YCbCr to RGB conversion. C1_COEF = [C1 * 1024 + 0.5].	RW

NOTE:
R = C0*(Y – LUMA_OF) + C1*(Cr-CHROM_OF)
G = C0*(Y – LUMA_OF) – C2*(Cb-CHROM_OF) – C3*(Cr-CHROM_OF)
B = C0*(Y – LUMA_OF) + C4*(Cb-CHROM_OF)

6.4.27 CSC C2 Coefficient Register

CSC_C2_COEF **0x3C**

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																				C2_COEF												
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	R/W
31:12	Reserved	Writing has no effect, read as zero.	R
11:0	C2_COEF	The C2 coefficient of the YUV/YCbCr to RGB conversion. C2_COEF = [C2 * 1024 + 0.5].	RW

NOTE:

R = C0*(Y – LUMA_OF) + C1*(Cr-CHROM_OF)

G = C0*(Y – LUMA_OF) – C2*(Cb-CHROM_OF) – C3*(Cr-CHROM_OF)

B = C0*(Y – LUMA_OF) + C4*(Cb-CHROM_OF)

6.4.28 CSC C3 Coefficient Register

CSC_C3_COEF **0x40**

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																				C3_COEF												
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	R/W
31:12	Reserved	Writing has no effect, read as zero.	R
11:0	C3_COEF	The C3 coefficient of the YUV/YCbCr to RGB conversion. C3_COEF = [C3 * 1024 + 0.5].	RW

NOTE:

R = C0*(Y – LUMA_OF) + C1*(Cr-CHROM_OF)

G = C0*(Y – LUMA_OF) – C2*(Cb-CHROM_OF) – C3*(Cr-CHROM_OF)

B = C0*(Y – LUMA_OF) + C4*(Cb-CHROM_OF)

6.4.29 CSC C4 Coefficient Register

CSC_C4_COEF **0x44**

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																				C4_COEF												
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	R/W
31:12	Reserved	Writing has no effect, read as zero.	R
11:0	C4_COEF	The C4 coefficient of the YUV/YCbCr to RGB conversion. C4_COEF = [C4 * 1024 + 0.5].	RW

NOTE:

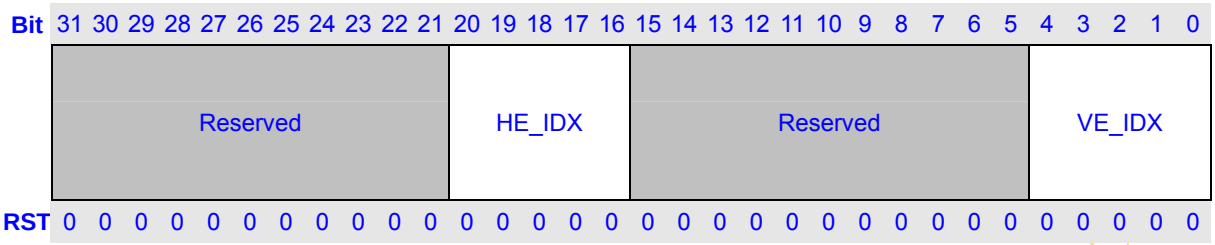
R = C0*(Y – LUMA_OF) + C1*(Cr-CHROM_OF)

G = C0*(Y – LUMA_OF) – C2*(Cb-CHROM_OF) – C3*(Cr-CHROM_OF)

B = C0*(Y – LUMA_OF) + C4*(Cb-CHROM_OF)

6.4.30 Resize Coefficients Table Index Register

RSZ_COEF_INDEX **0x30**

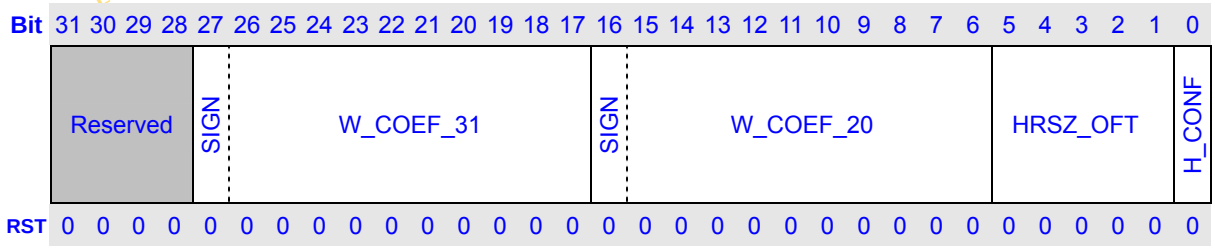


Bits	Name	Description	R/W
31:21	Reserved	Writing has no effect, read as zero.	R
20:16	HE_IDX *[1]	Indicates the end address of the horizontal resize look up table.	RW
15:5	Reserved	Writing has no effect, read as zero.	R
4:0	VE_IDX *[1]	Indicates the end address of the vertical resize look up table.	RW

NOTE: The HE_IDX (VE_IDX) should be the depth of the horizontal (vertical) resize look up table **minus 1**, and how to get HE_IDX or VE_IDX, please refer to 3.34.

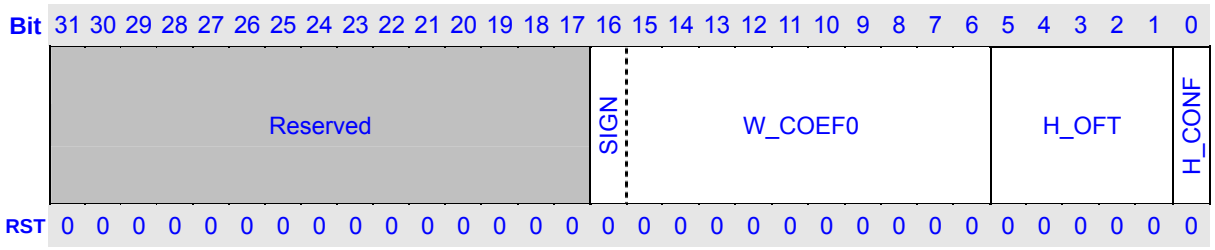
6.4.31 Horizontal Resize Coefficients Look Up Table Register group

HRSZ_COEF_LUT (bi-cube) **0x48**



Bits	Name	Description	R/W
31:28	Reserve	Writing has no effect, read as zero.	W
27:17	W_COEF_31	Weighting coefficients for pix_3 or pix_1.	W
16:6	W_COEF_20	Weighting coefficients for pix_2 or pix_0.	W
5:1	HRSZ_OFT	Horizontal Resize pixel offset.	W
0	H_CONF	Start to configure the horizontal resize table, read as zero: 1: start.	W

HRSZ_COEF_LUT (bilinear) **0x48**



Bits	Name	Description	R/W
31:17	Reserve	Writing has no effect, read as zero.	R
16:6	W_COEF0	Weighting coefficients for pix_0.	W
5:1	H_OFT	Horizontal Resize pixel offset.	W
0	H_CONF	Start to configure the horizontal resize table, read as zero: 1: start.	W

NOTES:

1 The **bilinear zooming** weighting coefficients should be calculated as following: because it is 11 bits length, a one bit is the sign-bit, so, that is to say the precision is 1/512.

For up-scaling,

$W_k = 1 – (k*n/m – [k*n/m]), k = 0, 1, … m-1.$

For down-scaling,

```
for (t=0, k=0; k < n; k++)
{
   If ([(t*n+1)/m] – k >=1) { W_k = 0;}
   else if ((t*n+1)/m – k == 0) { W_k = 1; t++;}
   else { W_k = 1 – ( (t*n+1)/m – [t*n/m]); t++;}
}
```

$W\_COEF_k = [512 * W_k]$ (stands for get the rounding integer, [20.33] = 20 while [20.66] = 21)

Here n stands for original pixel points, m stands for pixel points after resize. For example down-scaling 5:3, n = 5, m = 3. Moreover, m and n are prime, that is, for example 8:2 should be converted to 4:1.

When IPU_CONTROL.RSZ_EN set as 1 and m:n = 1:1, all coefficients should be calculated as up-scale case.

2 The **bi-cube zooming** weighting coefficients should be calculate as following: because it is 10 bits length, that is to say the precision is 1/512.

Step1:

For up-scaling,

$W_k = 1 – (k*n/m – [k*n/m]), k = 0, 1, … m-1.$

For down-scaling,

```
for (t=0, k=0; k < n; k++)
{
    If ([(t*n+1)/m] – k >=1) { W_k = 0;}
```

else if ((t*n+1)/m – k == 0) { W_k = 1; t++;}
else { W_k = 1 – ((t*n+1)/m – [t*n/m]); t++;}
}

W_COEF_k = [512 * W_k] (stands for get the rounding integer, [20.33] = 20 while [20.66] = 21)

Here n stands for original pixel points, m stands for pixel points after resize. For example down-scaling 5:3, n = 5, m = 3. Moreover, m and n are prime, that is, for example 8:2 should be converted to 4:1.

When IPU_CONTROL.RSZ_EN set as 1 and m:n = 1:1, all coefficients should be calculated as up-scale case.

Step 2:

After calculate the Wk, then using the following rule to get the coefficients:

```
double SinXDivX(double x)
{
    const float a = -1; //a can be a=-2,-1,-0.75,-0.5 and so on to control the blur level
    double x2=x*x;
    double x3=x2*x;
    if (x<=1)
    return (a+2)*x3 - (a+3)*x2 + 1;
    else if (x<=2)
    return a*x3 - (5*a)*x2 + (8*a)*x - (4*a);
    else
    return 0;
}
```

W0 = 512*SinXDivX(1+Wk) W1 = 512*SinXDivX(Wk)

W2 = 512*SinXDivX(1-Wk) W3 = 512*SinXDivX(2-Wk)

Step 3:

And then the zooming weight coefficient should set to IPU as following:

Prepare: Set H_CONF to 1

a set W4n+1 & W4n+0

b set W4n+3 & W4n+2

Index(n)	step	W0	W1	step	W2	W3
0	0	-34	129	1	100	-19
1	2	-45	45	3	77	-33
2	4	-12	122	5	94	-56
3	6	-13	230	7	123	-77
4	8	-23	11	9	45	-100
5	10	-19	87	11	69	-90
6	12	-12	79	13	148	-8

3 Calculate the H_OFT.

Step 1: calculate the line in enable flag (IN_EN) and out enable flag (OUT_EN) table.

IN_EN: In down scale case, IN_EN always equals 1.
In up scale case,
For (i=0, k=0; k < m; k++) {
If(i<= k*n/m) { IN_EN [k] = 1; i++;}
else { IN_EN [k] =0;}

OUT_EN: In up scale case, OUT_EN always equals 1.
In down scale case,
For (t=0, k=0; k < n; k++) {
If([(t*n+1)/m] – k >=1)
OUT_EN [k] = 0;
else {OUT_EN[k]=1; t++;}
}

Step 2: calculate the H_MAX_LUT.

H_MAX_LUT = max (m, n) – 1

*Step 3: C*alculate the LUT.

```
int hoft_table_buf[33];
int hcoef_table_buf[33];
int voft_table_buf[33];
int vcoef_table_buf[33];
int *hoft_table = &hoft_table_buf[1];
int *hcoef_table = &hcoef_table_buf[1];
int *voft_table = &voft_table_buf[1];
int *vcoef_table = &vcoef_table_buf[1];
int in_oft_tmp = 0;
int hcoef_real_heiht = 0   ;
int vcoef_real_heiht = 0   ;
int coef_tmp = 0   ;
j = -1 ;
 for (i=0; i<=H_MAX_LUT+1; i++)
 {
     if ( h_lut[i].out_n )
     {
          hoft_table[j] = (h_lut[i].in_n == 0)? 0: in_oft_tmp;
          hcoef_table[j] = coef_tmp;
          coef_tmp = h_lut[i].coef;
          in_oft_tmp = h_lut[i].in_n==0? in_oft_tmp : h_lut[i].in_n ;
          j++;
      }
      else
      in_oft_tmp = h_lut[i].in_n + in_oft_tmp;
 }
if ( h_lut[0].out_n )
{
```

```
        hoft_table[j] = (h_lut[0].in_n == 0)? 0: in_oft_tmp;        \
        hcoef_table[j] = coef_tmp;
    }
  j++;
  hcoef_real_heiht = j;
    RSZ_COEF_INDEX. HE_IDX = j –1;
```

Step 4: Calculate the last table of resize coefficient.

Bilinear:

```
for (cnt =0 ; cnt<hcoef_real_heiht ; cnt ++)
        HRSZ_COEF_LUT_bilinear[cnt] = (hcoef_table[cnt], hoft_table[cnt]);
```

Bicube:

```
 int sinxdivx_table_8[(2<<9)+1];
for ( i = 0 ; i < (2<<9); ++i)
 {
     sinxdivx_table_8[i] = (int)(0.5 + 512*sinxdivx(i*(1.0/512)));
 }
int u_8;
for (i = 0 ; i <hcoef_real_heiht; i++ )
{
        int au_8[4];
        u_8 = 512 - hcoef_table[i];
        cube_hcoef_table[i][0] = sinxdivx_table_8[(1<<9)+u_8];
        cube_hcoef_table[i][1] = sinxdivx_table_8[u_8];
        cube_hcoef_table[i][2] = sinxdivx_table_8[(1<<9)-u_8];
        cube_hcoef_table[i][3] = sinxdivx_table_8[(2<<9)-u_8];
   }
for (cnt =0 ; cnt<hcoef_real_heiht ; cnt ++)
        HRSZ_COEF_LUT_bicube[cnt] = (cube_hcoef_table[cnt], hoft_table[cnt]);
```


4 Following are two examples of setting LUT in bilinear scale mode.

Resize coefficients for 7:3:

W	W_COEF	IN_EN	OUT_EN	Pixel 1	Pixel 2	OUT
2/3	341	1	1	P [0]	P [1]	P [0] * 2/3 + P [1] * 1/3
0	0	1	0	P [1]	P [2]	No new pixel out
1/3	171	1	1	P [2]	P [3]	P [2] * 1/3 + P [3] * 2/3
0	0	1	0	P [3]	P [4]	No new pixel out
0	0	1	0	P [4]	P [5]	No new pixel out
1	512	1	1	P [5]	P [6]	P [5] * 1 + P [6] * 0
0	0	1	0	P [6]	P [7]	No new pixel out

Parameter set to IPU is following:

index	W	W_COEF	OFSET	Pixel1	Pixel 2	OUT
0	2/3	341	2	P [0]	P [1]	P [0] * 2/3 + P [1] * 1/3
1	1/3	171	3	P [2]	P [3]	P [2] * 1/3 + P [3] * 2/3
2	1	512	2	P [5]	P [6]	P [5] * 1 + P [6] * 0

Resize coefficients for 3:5:

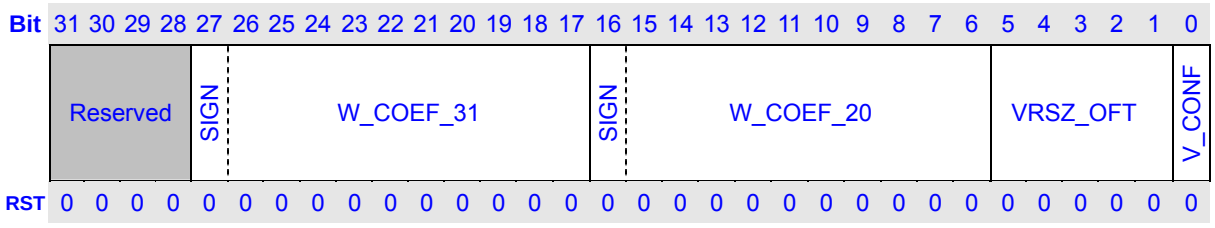
W	W_COEF	IN_EN	OUT_EN	Pixel 1	Pixel 2	OUT
1	512	1	1	P [0]	P [1]	P [0] * 1 + P [1] * 0
2/5	205	0	1	P [0]	P [1]	P [0] * 2/5 + P [1] * 3/5
4/5	410	1	1	P [1]	P [2]	P [1] * 4/5 + P [2] * 1/5
1/5	102	0	1	P [1]	P [2]	P [1] * 1/5 + P [2] * 4/5
3/5	307	1	1	P [2]	P [3]	P [2] * 3/5 + P [3] * 2/5

The parameter set to IPU is as following:

index	W	W_COEF	OFSET	Pixel 1	Pixel 2	OUT
0	1	512	0	P [0]	P [1]	P [0] * 1 + P [1] * 0
1	2/5	205	1	P [0]	P [1]	P [0] * 2/5 + P [1] * 3/5
2	4/5	410	0	P [1]	P [2]	P [1] * 4/5 + P [2] * 1/5
3	1/5	102	1	P [1]	P [2]	P [1] * 1/5 + P [2] * 4/5
4	3/5	307	1	P [2]	P [3]	P [2] * 3/5 + P [3] * 2/5

6.4.32 Vertical Resize Coefficients Look Up Table Register group

VRSZ_COEF_LUT (bi-cube) **0x4C**



Bits	Name	Description	R/W
31:28	Reserved	Writing has no effect, read as zero.	W
27:17	W_COEF_31	Weighting coefficients for pix_3 or pix_1.	W
16:6	W_COEF_20	Weighting coefficients for pix_2 or pix_0.	W
5:1	VRSZ_OFT	Vertical Resize pixel offset.	W
0	V_CONF	Start to configure the vertical resize table, read as zero: 1: start.	W

VRSZ_COEF_LUT (bilinear) **0x4C**

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															W_COEF0											V_OFT					V_CONF
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	R/W
31:17	Reserved	Writing has no effect, read as zero.	R
16:6	W_COEF0	Weighting coefficients for pix_0.	W
5:1	V_OFT	Vertical Resize pixel offset.	W
0	V_CONF	Start to configure the vertical resize table, read as zero: 1: start.	W

NOTE: refer to Horizontal HRSZ_COEF_LUT.

6.4.33 Calculation for Resized width and height

For software, to preset correct value for register OUT_GS, please refer to following formula.
Set IW stand for original input frame width, IH stand for original input frame height, OW stand for new frame width after resize, OH stand for new frame height after resize.

In Up-scale case (n < m):

If [(IW - 1) * (m/n)] * (n/m) ==(IW-1) then
    OW = [(IW - 1) * (m/n)] + 1;
Else OW = [(IW - 1) * (m/n)] + 2;
If [(IH - 1) * (m/n)] * (n/m) == (IH-1) then
    OH = [(IH - 1) * (m/n)] + 1;
Else OH = [(IH - 1) * (m/n)] + 2;

In Down-scale case (n>m):

If [(IW - 1) * (m/n)] * (n/m) ==(IW-1) then
    OW = [(IW - 1) * (m/n)];
Else OW = [(IW - 1) * (m/n)] + 1;
If [(IH - 1) * (m/n)] * (n/m) ==(IH-1) then
    OH = [(IH - 1) * (m/n)];
Else OH = [(IH - 1) * (m/n)] + 1;

For example:

A 36x46 frame with the horizontal resize ratio of 4:5 (up-scale) and vertical resize ratio of 7:6 (down-scale), by the expressions above we get its new size after resize from the following process.

For Width: [(36 - 1) * (5/4)] * (4/5) = 34.4 ≠ (36-1)

So OW = [(36 - 1) * (5/4)] + 2 = 45

For Height: [(46 - 1) * (6/7)] * (7/6) = 44.33 ≠ (46 - 1)

So OH = [(46 - 1) * (6/7)] + 1 = 39

6.4.34 CSC Offset Parameter Register

CSC_OFSET_PARA **0x50**

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								CHROM_OF								Reserved								LUMA_OF							
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	R/W
31:24	Reserved	Writing has no effect, read as zero.	R
23:16	CHROM_OF	Chroma offset value.	RW
15:8	Reserved	Writing has no effect, read as zero.	R
7:0	LUMA_OF	Luma offset value.	RW
NOTE: R = C0*(Y – LUMA_OF) + C1*(Cr-CHROM_OF) G = C0*(Y – LUMA_OF) – C2*(Cb-CHROM_OF)– C3*(Cr-CHROM_OF) B = C0*(Y – LUMA_OF) + C4*(Cb-CHROM_OF)			

6.4.35 Picture enhance table

PENC_TAB **0x7FF~0x400**

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
	C4n+3								C4n+2								C4n+1								C4n							
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

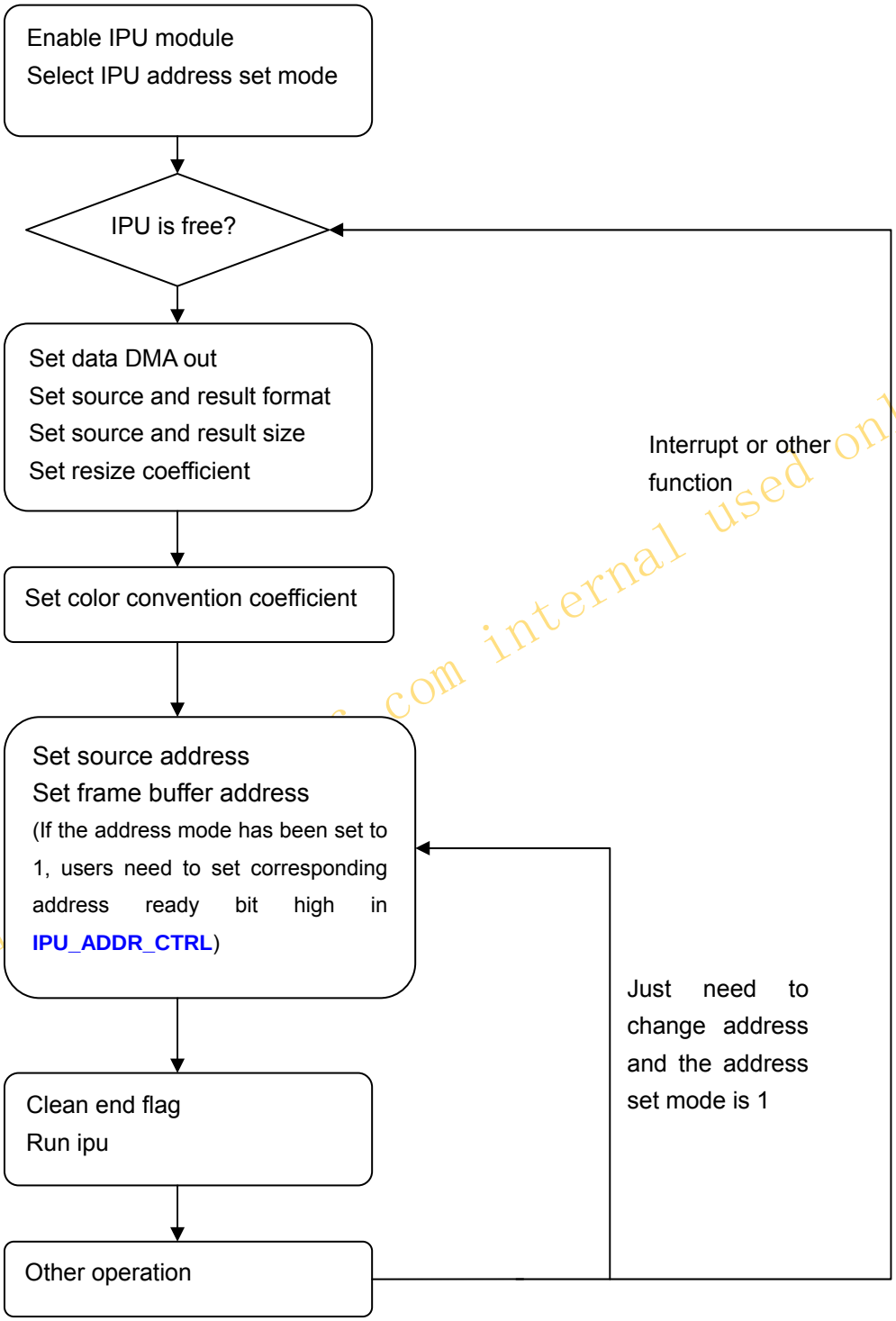
Bits	Name	Description	R/W
31:24	C4n+3	Color mapping result.	W
23:16	C4n+2	Color mapping result.	W
15:8	C4n+1	Color mapping result.	W
7:0	C4n	Color mapping result.	W

When the IPU_CONTROL.PENC_OPT == 1and the source picture is YUV, this table will act. This

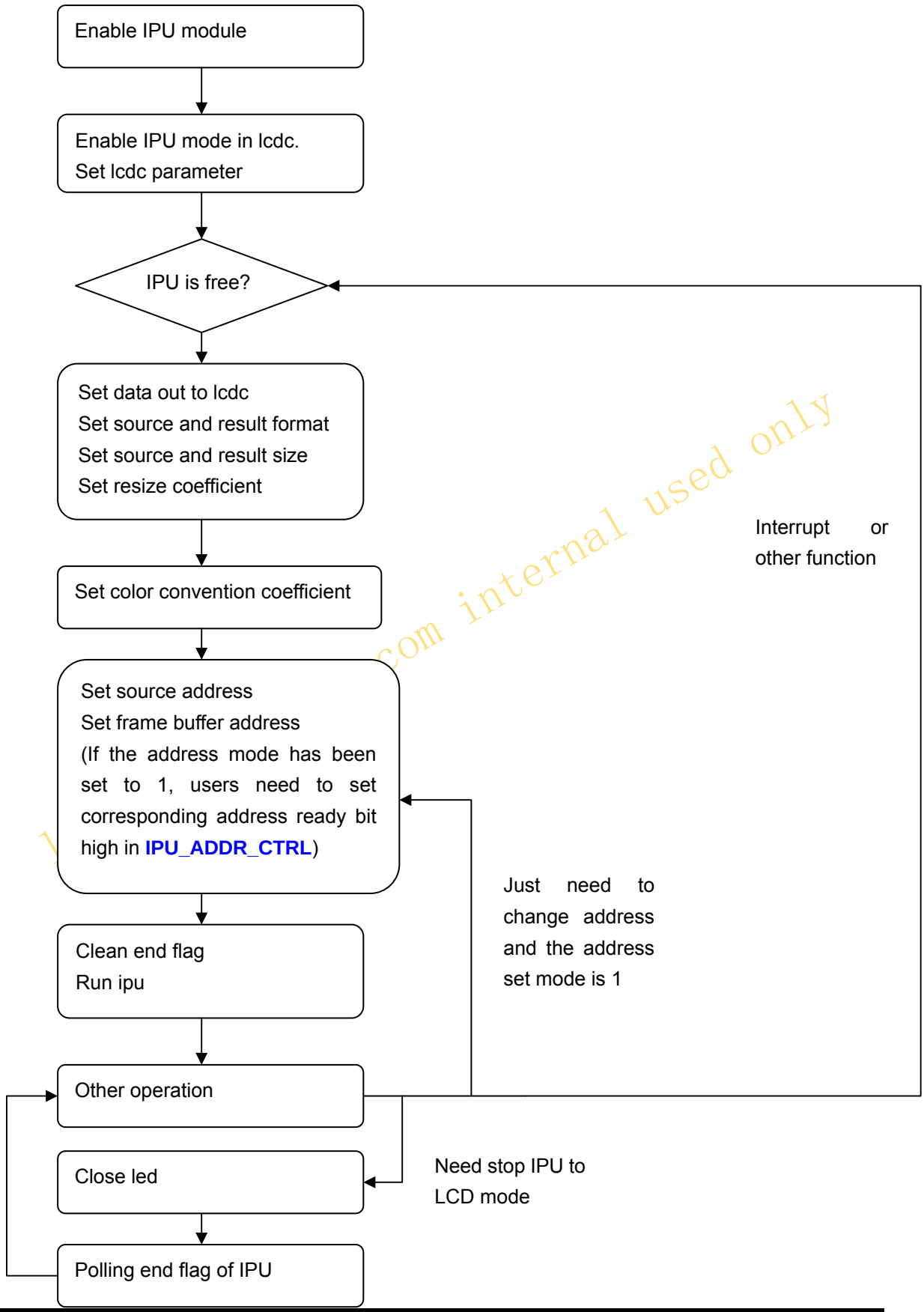
table will tell IPU how to mapping the resizing result to the final result. For an example, if the resizing result is 0x80, and the index 0x80 of this table is 0x70, so the final result will be 0x70.

6.5 IPU Operation Flow

6.5.1 Data out to frame buffer



6.5.2 Data out to lcdc



6.5.3 Operation example

Table 6-2 no mapping mode

Step	Action
Base	Chip_enable()
Base_0	ipu_addr_sel(1);
0	Do { } while {!polling_end_flag}
1	set_primary_ctrl(VRSZ_ENABLE, HRSZ_ENABLE, CSC_EN, irq_en); //
2	set_source_ctrl(source_pkg_sel, SPAGE_SEL) ;
3	set_out_ctrl(lcdc_sel, DPAGE_SEL, DISP_SEL, FIELD_SEL, FIELD_CONF_EN) ;
4	set_scale_ctrl(V_SCALE, H_SCALE);
5	set_ipu_fmt(RGB888_OUT_FMT, OUT_OFT_RGB, OUT_FMT, OUT_Y1UY0V , IN_OF_YUYV, IN_FM_YUV444);
6	set_inframe_gsize(FIN_W, FIN_H, FIN_Y_STRIDE, FIN_U_STRIDE, FIN_V_STRIDE);

7	set_y_addr((unsigned int)fin_y & 0x1FFFFFFF);
	set_u_addr((unsigned int)fin_y & 0x1FFFFFFF);
	set_v_addr((unsigned int)fin_y & 0x1FFFFFFF);

8	set_outframe_gsize(FOUT_W, FOUT_H , FOUT_STRIDE);
9	set_out_addr((unsigned int)fout & 0x00000FFF);
9A	set_addr_ready(0xFF); **NOTE:** this step is necessary when ipu address set mode is 1.

10	set_csc_c0(YUV_CSC_C0);
	set_csc_c1(YUV_CSC_C1);
	set_csc_c2(YUV_CSC_C2);
	set_csc_c3(YUV_CSC_C3);
	set_csc_c4(YUV_CSC_C4);

11	set_csc_ofset_para (128, 0) ;
12	set_rsz_lut_end(H_MAX_LUT-1, V_MAX_LUT-1);
13	start_hlut_coef_write(); **NOTE:** This step is necessary before write new LUT.

14	for (i=0;i<H_MAX_LUT;i++) {
	set_hrsz_lut_coef(h_lut[i].coef, h_lut[i].in_n, h_lut[i].out_n);
	}

15	start_vlut_coef_write(); **NOTE:** This step is necessary before write new LUT.

16	for (i=0;i<V_MAX_LUT;i++) {
	set_vrsz_lut_coef(v_lut[i].coef, v_lut[i].in_n, v_lut[i].out_n);
	}

17	Clean_end_flag(); run_ipu();

Table 6-3 mapping mode

Step	Action

Prepare	y_phy_table[0] = ((unsigned int)fin_y & 0x0FFFF000) | 0x20000000 ;
	u_phy_table[0] = ((unsigned int)fin_u & 0x0FFFF000) | 0x20000000 ;
	v_phy_table[0] = ((unsigned int)fin_v & 0x0FFFF000) | 0x20000000 ;
	out_phy_table[0] = ((unsigned int)fout & 0x0FFFF000) | 0x20000000 ;
	for (i =1; i<100; i++){
	y_phy_table[i] = y_phy_table[i-1] + 4096 ;
	u_phy_table[i] = u_phy_table[i-1] + 4096 ;
	v_phy_table[i] = v_phy_table[i-1] + 4096 ;
	out_phy_table[i] = out_phy_table[i-1] + 4096 ;
	}

Base	Chip_enable()
Base_0	ipu_addr_sel(1);
0	Do { } while {!polling_end_flag}
1	set_primary_ctrl(VRSZ_ENABLE, HRSZ_ENABLE, CSC_EN, irq_en); //
2	set_source_ctrl(source_pkg_sel, SPAGE_SEL) ;
3	set_out_ctrl(lcdc_sel, DPAGE_SEL, DISP_SEL, FIELD_SEL, FIELD_CONF_EN) ;
4	set_scale_ctrl(V_SCALE, H_SCALE);
5	set_ipu_fmt(RGB888_OUT_FMT, OUT_OFT_RGB, OUT_FMT, OUT_Y1UY0V , IN_OF_YUYV, IN_FM_YUV444);
6	set_inframe_gsize(FIN_W, FIN_H, FIN_Y_STRIDE, FIN_U_STRIDE, FIN_V_STRIDE);

7	set_y_addr((unsigned int)fin_y & 0xFFF);
	set_u_addr((unsigned int)fin_y & 0xFFF);
	set_v_addr((unsigned int)fin_y & 0xFFF);

8	set_outframe_gsize(FOUT_W, FOUT_H , FOUT_STRIDE);
9	set_out_addr((unsigned int)fout & 0x00000FFF);
10	set_y_phy_t_addr((unsigned int)y_phy_table & 0x1FFFFFFF) ; set_u_phy_t_addr((unsigned int)u_phy_table & 0x1FFFFFFF) ; set_v_phy_t_addr((unsigned int)v_phy_table & 0x1FFFFFFF) ; set_out_phy_t_addr((unsigned int)out_phy_table & 0x1FFFFFFF) ;
10A	set_addr_ready(0xFF); **NOTE:** this step is necessary when ipu address set mode is 1.

11	set_csc_c0(YUV_CSC_C0);
	set_csc_c1(YUV_CSC_C1);
	set_csc_c2(YUV_CSC_C2);
	set_csc_c3(YUV_CSC_C3);
	set_csc_c4(YUV_CSC_C4);

12	set_csc_ofset_para (128, 0) ;
13	set_rsz_lut_end(H_MAX_LUT-1, V_MAX_LUT-1);

```
14      start_hlut_coef_write();
        NOTE: This step is necessary before write new LUT.
15        for (i=0;i<H_MAX_LUT;i++) {
            set_hrsz_lut_coef(h_lut[i].coef, h_lut[i].in_n, h_lut[i].out_n);
          }
16        start_vlut_coef_write();
        NOTE: This step is necessary before write new LUT.
17        for (i=0;i<V_MAX_LUT;i++) {
            set_vrsz_lut_coef(v_lut[i].coef, v_lut[i].in_n, v_lut[i].out_n);
          }
18      Clean_end_flag();
        run_ipu();
```

6.6 Special Instruction

A1. Resizing size feature

Input size (W x H)		Output size (W x H)	
Min	4x4	Disable vertical scale	Min: 4x4
			Max: 4095x4095
Max	4095x4095	Enable vertical scale	Min: 4x4
			Max: 1280x4095

A2. Color convention feature

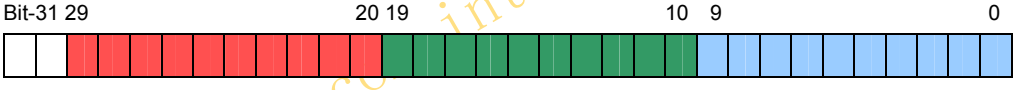
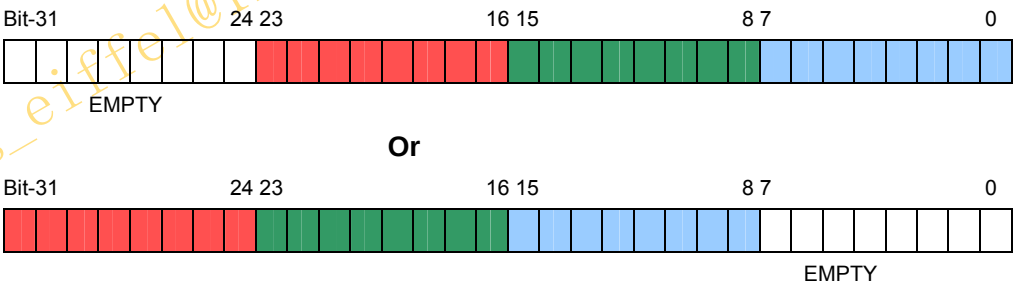
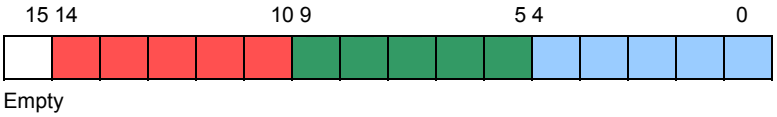
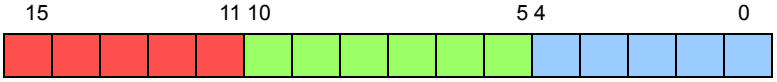
Source format	Output format	Parameter configure (necessary)
RGB	RGB	IPU_CONTROL.CSC_EN =0 IPU_CONTROL. SPKG_SEL = 0 or 1
		D_FMT. IN_FMT
		D_FMT.OUT_FMT = 2'b00, 2'b01, 2'b10
YUV	RGB	IPU_CONTROL.CSC_EN =1
		IPU_CONTROL. SPKG_SEL
		D_FMT. IN_FMT
		D_FMT. IN_OFT ***(IPU_CONTROL. SPKG_SEL == 1)***
		D_FMT.OUT_FMT = 2'b00, 2'b01, 2'b10
		D_FMT.RGB_OUT_OFT.
		CSC_C0 (1,2,3,4)_COEF, CSC_OFSET_PARA
YUV	YUV (package)	IPU_CONTROL.CSC_EN =0
		IPU_CONTROL. SPKG_SEL
		D_FMT. IN_FMT
		D_FMT. IN_OFT ***(IPU_CONTROL. SPKG_SEL == 1)***
		D_FMT.OUT_FMT = 2'b11

A3. YUV/YCbCr to RGB CSC Equations

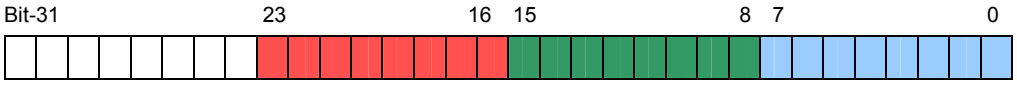
Input data	Matrix	CSC_COEF
YUV	R = C0*(Y – X0) + C1*(V-128)	CSC_C0_COEF = 0x400
	G = C`0*(Y – X0) – C2*(U-128) – C3*(V-128)	CSC_C1_COEF= 0x59C
	B = C0*(Y – X0) + C4*(U-128)	CSC_C2_COEF = 0x161
	X0: 0	CSC_C3_COEF = 0x2DC

	C0: 1 C1: 1.4026 C2: 0.3444 C3: 0.7144 C4: 1.7730	CSC_C4_COEF = 0x718
YCbCr	R = C0*(Y – X0) + C1*(Cr-128) G = C0*(Y – X0) – C2*(Cb-128) – C3*(Cr-128) B = C0*(Y – X0) + C4*(Cb-128) X0: 16 C0: 1.164 C1: 1.596 C2: 0.391 C3: 0.813 C4: 2.018	CSC_C0_COEF = 0x4A8 CSC_C1_COEF = 0x662 CSC_C2_COEF = 0x191 CSC_C3_COEF = 0x341 CSC_C4_COEF = 0x811

A4. Output data package format (RGB order)

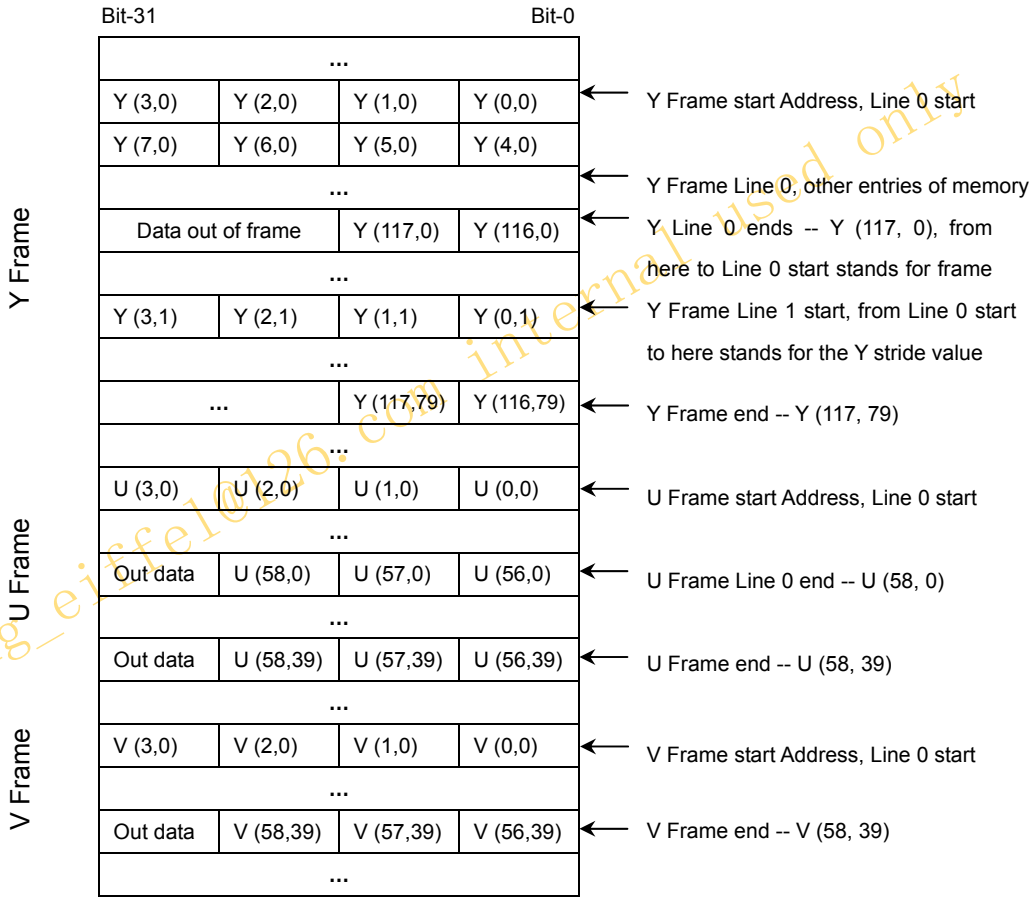
Format	Package
RGBAAA	
RGB888	
RGB555	
RGB565	
NOTE: All R/G/B data are little-endian type; all the empty bits in the above figure are filled with 0.	

A5. Input data package format (RGB order)

Format	Package
RGBPACK	

A6. Source Data storing format in external memory (separated YUV Frame)

Example: YUV420 118x80 frame



NOTES:

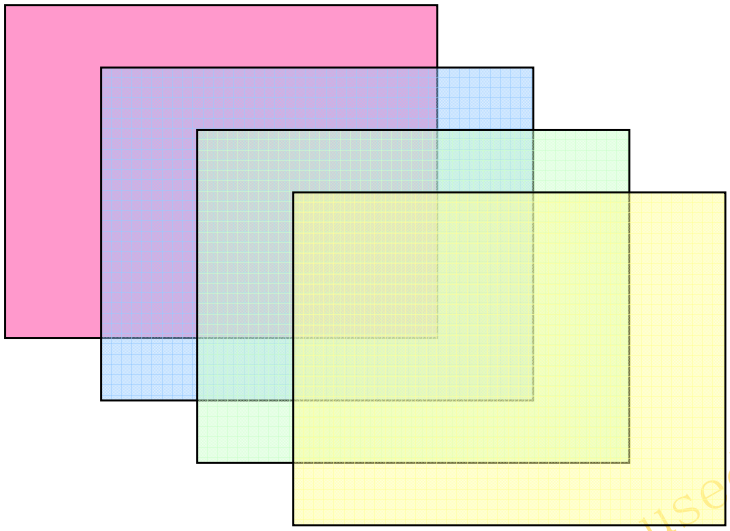
1 Every line's start address should be word aligned.

2 All pixel data should be stored as little-endian format.

3 Destination data (RGB) storing format in external memory is similar with above figure, but RGB555 and RGB565 frame's every line start address can be half-word aligned. (RGB888 frame still need word aligned)

7 Alpha_osd

7.1 Overview

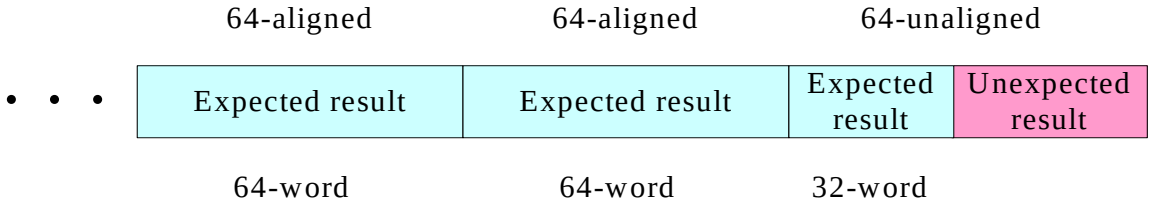


4-level overlay DMA, which can calculate Alpha Blending.

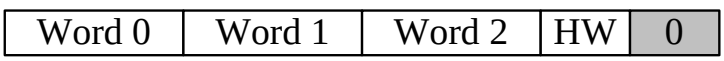
Features:

- Support ARGB8888, RGB565, RGB555 (16'h0000 means total transparent, others mean total overlaid). (this called pixel alpha mode)
- Each layer has an alpha value for all pixels (called frame alpha mode). Those value only use in a special mode, which all pixels in this layer use same alpha value
- Up to 800*480
- Software can change overlay orders
- The level of overlay can be set by software
- Software must make sure the address of source and destination are 64-word aligned. If not, alpha_osd will change the parts unaligned

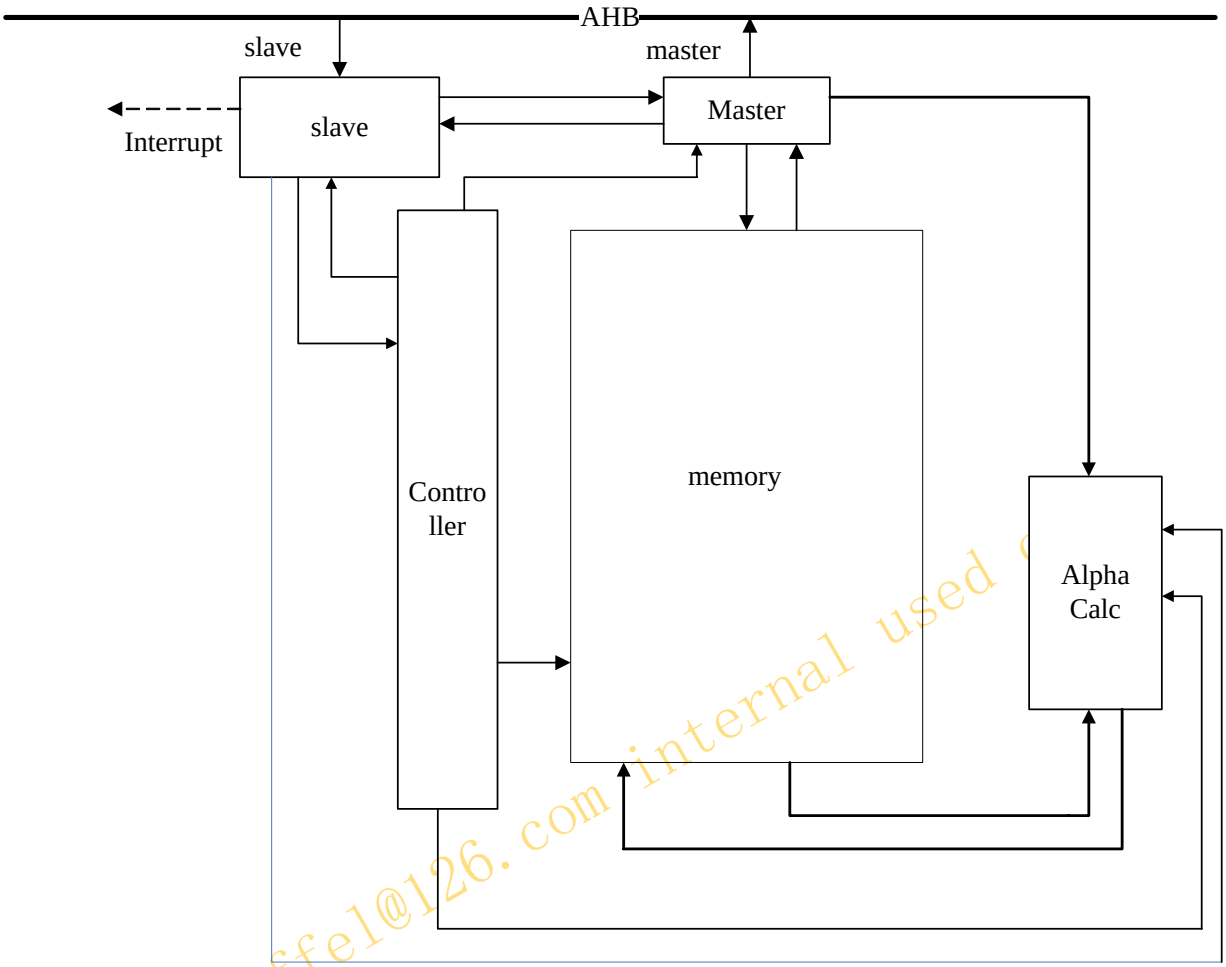
  For example:



- Support 64-burst in AHB bus
- In RGB656 & RGB555mode, software must make sure each line aligned in word. If not, software need fill the extra half-word with 16'h0000



7.2 Structure



7.3 Alpha blending function

This function is to calculate the Alpha blending. The blending is between two layers data and has 2 modes. One is the whole graph use one registered Alpha value and the other is each pixel have its own Alpha value. This function has 2 inputs, one (called foreground data in this module) is input foreground 0 data, and the other (called background data in this module) is input foreground 1 our background data determined by the display area. The calculate function is:

$$NewPixel = \frac{[(255 - Alpha) * background + Alpha * Foreground]}{255}$$

The calculate function that we used:

$$NewPixel = \frac{[(256 - Alpha) * background + Alpha * Foreground + 128]}{256}$$
$$= \frac{[256 * background + Alpha * (Foreground - background) + 128]}{256}$$

Alpha == 255 ? NewPixel = Foreground

For simplify the calculator process, we use 256 to instead the original 255, then add a 128/256 to approach the exact result. In this case we can use a shift register instead of a divider. To change the formula's format, use two full adder (in fact one of them is a subtractor) and one multiplier instead of one full-adder and two multipliers. Notice that 256 * background + 128 do not need adder or multiplier, use one 16-bit register which high 8-bit save the value of background, and low 8-bit is 8'b1000_0000.

In different mode the input, output and the Alpha have different values:

Alpha_osd_enable:

When **alpha is disable**, the output equal to foreground.

When **alpha is enable**, the output is the calculating result.

Alpha_osd_mode and **format mode**:

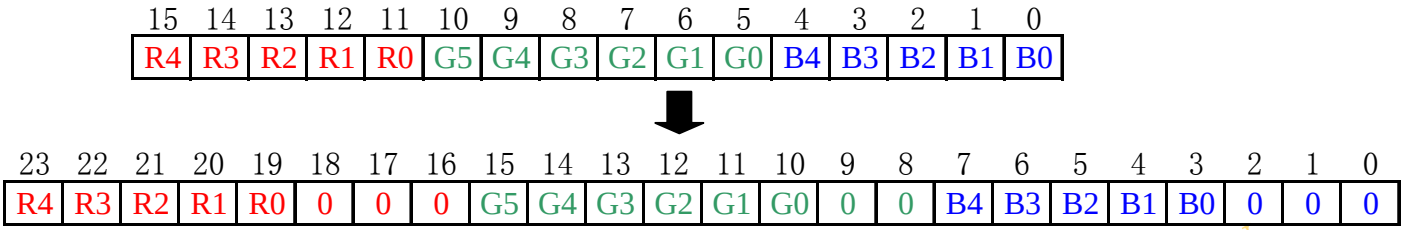
When alpha mode is **pixel alpha blending**:

Output low 24bits equal to the calculating result of CHANNEL0, high 8bits equal to the Alpha of the bottom frame, and alpha value comes from the high 8-bits of foreground multiple the corresponding alpha value form slv_reg_alphavalue register in case **format mode is ARGB8888**; output is decided by whether the value of foreground is 16'b0000 in case **format mode is RGB565 or RGB555**.

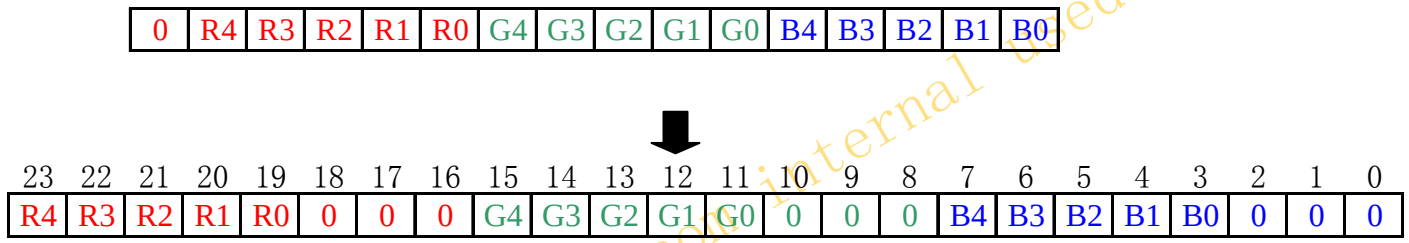
When alpha mode is **frame alpha blending**:

Output equal to the calculating result of CHANNEL0 high 8bits equal to the Alpha of the bottom frame, and alpha value comes from the slv_reg_alphavalue register in case **format mode is ARGB8888**; output is compose of results of CHANNEL0 and CHANNEL1, inputs of two channels must do format transforming as shown the figure below, the alpha value comes from the slv_reg_alphavalue register, specially foreground is all zero means total transparent in case **format mode is RGB565 or RGB555**.

RGB565



RGB555



7.4 Register Description

In this section, we will describe the registers in Alpha_osd. Following table lists all the registers definition. All register's 32bit address is physical address. And detailed function of each register will be described below.

Name	Description	RW	Reset Value	Address	Access Size
Reg_addr0	Address of DMA channel 0	RW	0x00000000	0x13070000	32
Reg_addr1	Address of DMA channel 1	RW	0x00000000	0x13070004	32
Reg_addr2	Address of DMA channel 2	RW	0x00000000	0x13070008	32
Reg_addr3	Address of DMA channel 3	RW	0x00000000	0x1307000C	32
Reg_waddr	Address of the destination of DMA	RW	0x00000000	0x13070010	32
Reg_addrlen	Length of DMA channel	RW	0x00000000	0x13070014	32
Slv_reg_ alphavalue	Alpha value of 4 frames	RW	0x00000000	0x13070018	32
CTRL*1	Ctrl register	RW	0x00060000	0x1307001C	32
int	Interrupt flag	RW	0x00000000	0x13070020	32
Clk_Gate	Control hclk gate	RW	0x00000001	0x13070048	32

NOTE:

***1:** This register must be set at last and please make sure other registers have been set correctly, if not you will get result unexpected.

7.4.1 Reg_addr0 ~Reg_addr3, Reg_waddr

These 5 registers define the address of 4 source frames and the destination respectively.

Reg_addr0 **0x13070000**

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
	Reg_addr0																															
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

Reg_addr1 **0x13070004**

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
	Reg_addr1																															
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

Reg_addr2 **0x13070008**

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
	Reg_addr2																															
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

Reg_addr3 **0x1307000C**

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
	Reg_addr3																															
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

Reg_waddr **0x13070010**

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
	Reg_waddr																															
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

7.4.2 Reg_addrlen

This register defines the length of each frame.

Reg_addrlen **0x13070014**

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
	Reg_addrlen																															
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

7.4.3 Slv_reg_alphavalue

This register defines the alpha values for each frame.

In the case of **ARGB8888 pixel alpha blending mode**, the alpha for calculating is high 8-bits of each foreground pixel multiplying corresponding alpha value in this register;

In the case of **RGB565 and RGB555 pixel alpha blending mode,** this register is unused;

In other cases, the alpha for calculating comes from corresponding alpha value in this register directly.

Slv_reg_alphavalue **0x13070018**

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
	Alpha value for addr3								Alpha value for addr2								Alpha value for addr1								Alpha value for addr0							
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

7.4.4 CTRL

This register is combined by several control bits.

CTRL **0x1307001C**

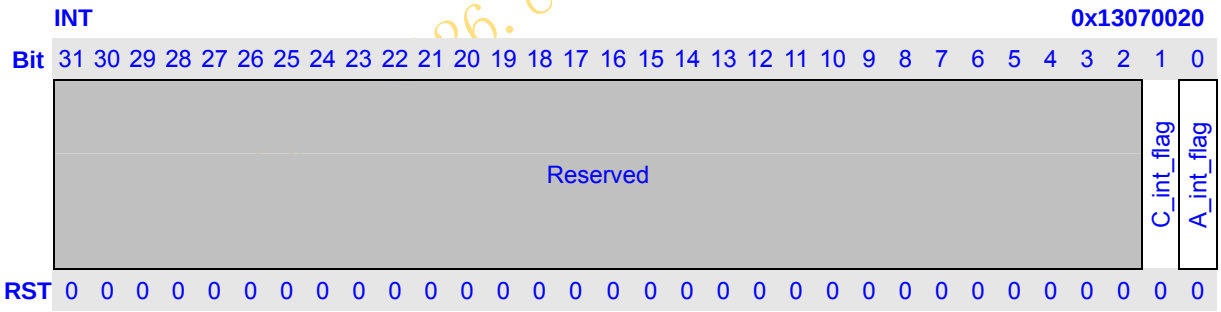
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Reserved												Frmlv		Frm_end	Alpha_start	Int_mask	Channel_level								Alpha mode					Format mode	Alpha enable	
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:20	Reserved	Writing has no effect, read as zero.	R
19:18	Frmlv	**[19:18]** / **Number of frames** 00 / invalid 01 / 2 10 / 3 11 / 4	RW
17	Frm_end	1: all frames' alpha blending are finished and send to frame buffer 0: alpha_osd is under working	R
16	Alpha_start	Writing 1 to this bit to start alpha_osd. When alpha_osd start to work, this bit is cleared by hardware automatically.	RW
15	Int_mask	Interrupt mask. 1: enable interrupt 0: disable interrupt	RW
14:7	Channel_level	2bits x 4, 4-layer order form up to down. When Frmlv was 2'b01 , [14:11] is useless; When Frmly was 2'b10, [14:13] is useless. **Bits** / **Description** 14:13 / The top frame ID 12:11 / The secondary top frame ID 10:9 / The secondary bottom frame ID 8:7 / The bottom frame ID	RW

<table>
<tr><td>6:3</td><td>Alpha_mode</td><td>1bit x 4, represent alpha mode of each frame corresponded with addr0~addr3.
0: pixel alpha blending
1: frame alpha blending
<table><tr><th>Bits</th><th>Description</th></tr><tr><td>6</td><td>Alpha mode for addr3 frame</td></tr><tr><td>5</td><td>Alpha mode for addr2 frame</td></tr><tr><td>4</td><td>Alpha mode for addr1 frame</td></tr><tr><td>3</td><td>Alpha mode for addr0 frame</td></tr></table></td><td>RW</td></tr>
<tr><td>2:1</td><td>Format mode</td><td><table><tr><th>[2:1]</th><th>Format</th></tr><tr><td>00</td><td>RGB565</td></tr><tr><td>01</td><td>RGB555</td></tr><tr><td>10</td><td>ARGB8888</td></tr><tr><td>11</td><td>Reserved</td></tr></table></td><td>RW</td></tr>
<tr><td>0</td><td>Alpha_enable</td><td>1: alpha enable
0: alpha disable</td><td>RW</td></tr>
</table>

7.4.5 INT

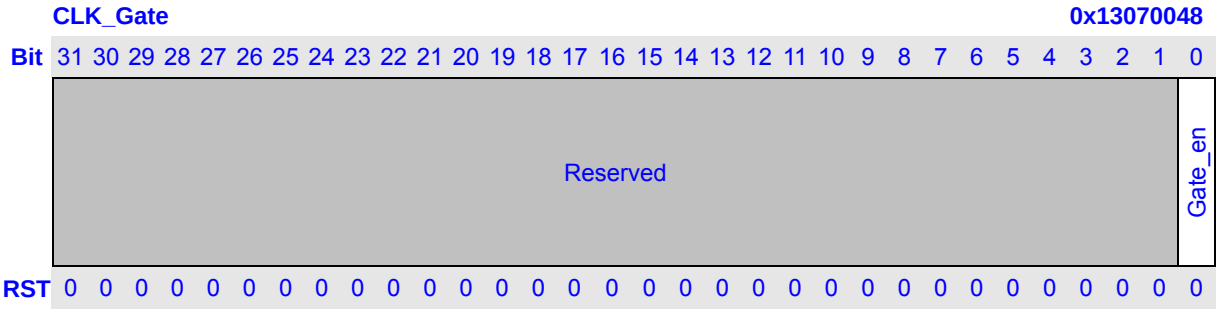
When frm_end became high, alpha_osd would generate interrupt signal if INT_mask was high, and interrupt would last until software written 1 to this register.



Bits	Name	Description	RW
31:2	Reserved	Writing has no effect, read as zero.	R
1	C_Int_flag	Compress end flag. writing 1 to this bit to clear flag. 1:interrupt 0:no interrupt ***NOTE:** this flag(c_int_flag) used in compress module.	RW
0	A_Int_flag	Alpha_osd end flag. writing 1 to this bit to clear flag. 1:interrupt 0:no interrupt	RW

7.4.6 Clk_Gate

This register set hclk gate for aosd_comp.



Bits	Name	Description	RW
31:1	Reserved	Writing has no effect, read as zero.	R
0	Gate_en	1: Enable hclk gate when controller idle	RW

7.5 Alpha_osd Operation

1 Look at frm_end.

Read CTRL and make sure frm_end is 1.

2 Configuration1.

Set Reg_addr0~Reg_addr3, Reg_waddr, Reg_addrlen and Slv_reg_alphavalue.

3 Configuration2 and start.

Set CTRL. If you want to start alpha_osd, set Alpha_start (CTRL[16]) to 1, if not set it to 0.

Be aware that Frmlv's default value is 2'b01, don't set it to 2'b00.

4 Interrupt handle.

Do configuration1 , configuration2 and start. If you want to start alpha_osd, set Alpha_start (CTRL[16]) to 1, if not set it to 0.

Clear the interrupt flag by writing 1 to Int.

8 LVDS Controller

8.1 Overview

This product is a single-Link high speed LVDS (Low-Voltage Differential Signaling) transmitter used for digital flat panel display systems. It's compatible with ANSI/TIA/EIA-644-A (LVDS) Standard. The transmitter converts 28bits parallel TTL data into four LVDS data streams. An in-phase transmit clock is transmitted in parallel with the data streams over a fifth LVDS link. It support full HDTV display up to 1920x1080p @ 60 Hz.

Feature:

- 25 to 135 MHz input clock support
- Supports VGA, SVGA, XGA , SXGA and HDTV
- Compatible with TIA/EIA-644 LVDS standard
- Support 24-bit Flat Panel Display
- Support VESA and JEIDA LVDS Data format

8.2 Register Description

In this section, we will describe the registers in LVDS controller. Following table lists all the register definitions. All registers' 32bit addresses are physical addresses. And detailed function of each register will be described below.

Table 8-1 LVDS Register Description

Name	Description	RW	Reset Value	Address	Access Size
TXCTRL	LVDS Transmitter 's Control Register	RW	0x80040060	0x130503C0	32
TXPLL0	LVDS Transmitter's PLL Control Register 0	RW	0x60001304	0x130503C4	32
TXPLL1	LVDS Transmitter's PLL Control Register 1	RW	0x61000000	0x130503C8	32

8.2.1 TXCTRL (LVDS Transmitter Control Register)

The register TXCTRL is used to control LVDS to work.

TXCTRL **0x130503C0**

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
	LVDS_MODEL_SEL	TX_PDB	TX_PDB_CK	Reserved	RESERVE_7	RESERVE_6	RESERVE_5	RESERVE_4	RESERVE_3	RESERVE_2	RESERVE_1	RESERVE_0	Reserved	TX_RSTB	TX_CKBIT_PHA_SEL	TX_CKBYTE_PHA_SEL	TX_CKOUT_PHA_S			TX_CKOUT_SET	TX_OUT_SEL	TX_DLY_SEL			TX_AMP_ADJ	TX_LVDS	TX_CR			TX_CR_CK	TX_OD_S	TX_OD_EN
RST	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

Bits	Name	Description	RW
31	LVDS_MODEL_SEL	VESA model or JEIDA model select. 0: JEIDA; 1:VESA.	RW
30	TX _PDB	Data channel Power down. 0: power down.	RW
29	TX_PDB_CK	Clock channel Power down. 0: power down.	RW
28	Reserved	Writing has no effect, read as zero.	R
27	LVDS_RESERVE_7	Reserved.	RW
26	LVDS_RESERVE_6	Reserved.	RW
25	LVDS_RESERVE_5	Reserved.	RW
24	LVDS_RESERVE_4	Reserved.	RW
23	LVDS_RESERVE_3	Reserved.	RW

22	LVDS_RESERVE_2	Reserved.	RW
21	LVDS_RESERVE_1	Reserved.	RW
20	LVDS_RESERVE_0	Reserved.	RW
19	Reserved	Writing has no effect, read as zero.	R
18	TX_RSTB	System reset signal. 0: Reset.	RW
17	TX_CKBIT_PHA_SEL	7x clock sampling edge configuration. 0: Rising edge; 1: Falling Edge.	RW
16	TX_CKBYTE_PHA_SEL	1x clock sampling edge configuration. 0: Rising edge; 1:Falling Edge.	RW
15:13	TX_CKOUT_PHA_S	Output data start-edge tuning in 1x clock output mode. 000: 0 of T_{7X} 001: 1 of T_{7X} … 111: 7 of T_{7X}	RW
12	TX_CKOUT_SET	TX clock channel output clock frequency set. 0: 1x clock output 1: 7x clock output	RW
11	TX_OUT_SEL	Transmitter output select. 0: LVDS output; 1: CMOS RGB output.	RW
10:8	TX_DLY_SEL	Input clock edge delay control, for setup/hold time fine tuning.	RW
7	TX_AMP_ADJ	LVDS output swing control. When AMP_ADJ=1, LVDS output swing is adjustable by CR<2:0>.	RW
6	TX_LVDS	Output amplitude tuning in mode of 'TX _AMP_CTRL'='0'. 0b, V_{OD}=200mV; 1b, V_{OD}=350mV.	RW
5:3	TX_CR	Digital logic input used to control output swing level.	RW
2	TX_CR_CK	Additional control bit of output swing level.	RW
1	TX_OD_S	Output level selectable pin.	RW
0	TX_OD_EN	Tx output control functions. 0: Disable ;1: Enable.	RW

8.2.2 TXPLL0 (LVDS Transmitter's PLL Control Register 0)

The register TXPLL0 is used to control PLL to work.

TXPLL0 **0x130503C4**

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
	LVDS_PLL_LOCK	PLL_DIS	BG_PWD	Reserved	PLL_SSC_EN	PLL_SSC_MODE	PLL_TEST	Reserved		PLL_POST_DIVA		PLL_POST_DIVB					Reserved	PLL_PLLN							PLL_TEST_DIV		PLL_IN_BYPASS	PLL_INDIV				
RST	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	1	0	0

Bits	Name	Description	RW
31	LVDS_PLL_LOCK	Lock detection output. 1: Lock.	R
30	PLL_DIS	PLL power down control. 1: power down.	RW
29	BG_PWD	Band-gap power down control. 1: power down.	RW
28	Reserved	Writing has no effect, read as zero.	R
27	PLL_SSC_EN	SSC function enable control. 1: Enable.	RW
26	PLL_SSC_MODE	SSC mode select. 0: Down spread 1: Center spread	RW
25	PLL_TEST	Test enable control. 1:Enable.	RW
24:23	Reserved	Writing has no effect, read as zero.	R
22:21	PLL_POST_DIVA	Post divider control bits A.	RW
20:16	PLL_POST_DIVB	N/C.	RW
15	Reserved	Writing has no effect, read as zero.	R
14:8	PLL_PLLN[6:0]	PLL feedback divider value configure.	RW
7:6	PLL_TEST_DIV	Output divider ratio control in test mode. 00: 1/2 01: 1/4 10: 1/8 11: 1/16	RW
5	PLL_IN_BYPASS	Input divider bypass.	RW
4:0	PLL_INDIV	Input divider value configure.	RW

8.2.3 TXPLL1 (LVDS Transmitter's PLL Control Register 0)

The register TXPLL1 is used to control PLL to work.

TXPLL1 **0x130503C8**

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
	PLL_ICP_SEL			Reserved	PLL_KVCO		PLL_IVCO_SEL		PLL_SSCN							PLL_COUNT													PLL_GAIN			
RST	0	1	1	0	0	0	0	1	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
31:29	PLL_ICP_SEL	Charge-pump current configure.	RW
28	Reserved	Writing has no effect, read as zero.	R
27:26	PLL_KVCO	VCO gain control.	RW
25:24	PLL_IVCO_SEL	VCO biasing current setup.	RW
23:17	PLL_SSCN	Internal divider for 30KHz clock generation.	RW
16:4	PLL_COUNT	SSC counter.	RW
3:0	PLL_GAIN	SSC counter gain.	RW

9 Camera Interface Module

9.1 Overview

The camera interface module (CIM) supports commonly available CMOS or CCD type image sensors. The CIM sources the digital image stream through a common 8-bit parallel digital protocol. The CIM can directly connect to external CMOS image sensors and ITU656 standard video decoders.

9.1.1 Features

- Input image size up to 4096x4096 pixels
- Max. VGA for image preview
- Max. VGA for video record
- Integrated DMA
- Input format
  - ITU601:YCbCr 4:4:4, YCbCr 4:2:2 and other formats
  - ITU656 (YCbCr 4:2:2)
- Output format
  - Packed : for all data format
  - YCbCr 4:4:4 Planar
  - YCbCr 4:2:2 Planar
- Configurable CIM_VSYNC and CIM_HSYNC signals: active high/low
- Configurable CIM_PCLK: active edge rising/falling
- 256x33 image data receive FIFO (RXFIFO)
- PCLK max. 120MHz
- Output format: csc mode is YCbCr 4:2:2, bypass mode is the input data format
- Configurable output order

9.1.2 Pin Description

Table 9-1 Camera Interface Pins Description

Name	I/O	Description
CIM_MCLK	O	CIM work clock
CIM_PCLK	I	Pixel clock from Image Sensor
CIM_VSYNC	I	Vertical synchronous from Image Sensor
CIM_HSYNC	I	Horizontal synchronous from Image Sensor
CIM_DATA[7:0]	I	Data bus from Image Sensor

9.2 CIM Special Register

The special registers are for CIM to configure and control the interface and DMA operation. The table below lists these registers.

Table 9-2 CIM Registers

Name	RW	Reset Value	Address	Access Size
CIMCFG	RW	0x00000000	0x13060000	32
CIMCR	RW	0x00000000	0x13060004	32
CIMST	RW	0x02020202	0x13060008	32
CIMIID	R	0x00000000	0x1306000C	32
CIMDA	RW	0x00000000	0x13060020	32
CIMFA	R	0x00000000	0x13060024	32
CIMFID	R	0x00000000	0x13060028	32
CIMCMD	R	0x00000000	0x1306002C	32
CIMSIZE	RW	0x00000000	0x13060030	32
CIMOFFSET	RW	0x00000000	0x13060034	32
CIMYFA	R	0x00000000	0x13060038	32
CIMYCMD	R	0x00000000	0x1306003C	32
CIMCBFA	R	0x00000000	0x13060040	32
CIMCBCMD	R	0x00000000	0x13060044	32
CIMCRFA	R	0x00000000	0x13060048	32
CIMCRCMD	R	0x00000000	0x1306004C	32
CIMCR2	RW	0x00000000	0x13060050	32

9.2.1 CIM Configuration Register (CIMCFG)

CIMCFG **0x13060000**

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
	EEOFEN	EXP	RF_TRIG						BW			SEP	ORDER		DF		INV_DAT	VSP	HSP	PCP	BURST_TYPE		DUMMY	E_VSYNC	LM	PACK			FP	BYPASS	DSM	
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	EEOFEN	Early EOF Mode Enable. 0: EEOF mode is disabled 1: When CIMCR.EEOF_LINE lines data has been transferred of a frame, the EEOF flag will be set, and the EEOF interrupt will occur	RW

<table>
<tr><td>30</td><td>EXP</td><td>Expand mode for CIM_DATA width.
0: CIM_DATA width = 8
1: CIM_DATA width > 8</td><td></td></tr>
<tr><td>29:24</td><td>RF_TRIG</td><td>Specifies the trigger value of RXFIFO.
<table>
<tr><th>CIMCFG.BURST_TYPE</th><th>RF_TRIG = n</th></tr>
<tr><td>INCR4</td><td>Trigger value is (n + 1) * 4</td></tr>
<tr><td>INCR8</td><td>Trigger value is (n + 1) * 8</td></tr>
<tr><td>INCR16</td><td>Trigger value is (n + 1) * 16</td></tr>
<tr><td>INCR32</td><td>Trigger value is (n + 1) * 32</td></tr>
</table></td><td>RW</td></tr>
<tr><td>23:22</td><td>BW</td><td>Bus width of CIM_DATA Interface.
When BW is n, the bus width is (n+9) bits.</td><td>RW</td></tr>
<tr><td>20</td><td>SEP</td><td>Separate frame format enable.
Used in output data format of YCbCr 4:4:4 and YcbCr 4:2:2.
0: Output is packaged frame format
1: Output is separated frame format</td><td>RW</td></tr>
<tr><td>19:18</td><td>ORDER</td><td>Input data stream order.
<table>
<tr><td></td><td>YCbCr 4:4:4</td><td>ITU656/YCbCr 4:2:2</td></tr>
<tr><td>00</td><td>YCbCr</td><td>Y_0CbY_1Cr</td></tr>
<tr><td>01</td><td>YCrCb</td><td>Y_0CrY_1Cb</td></tr>
<tr><td>10</td><td>CbCrY</td><td>CbY_0CrY_1</td></tr>
<tr><td>11</td><td>CrCbY</td><td>CrY_0CbY_1</td></tr>
</table></td><td>RW</td></tr>
<tr><td>17:16</td><td>DF</td><td>Input data format.
00: reserved
01: YCbCr 4:4:4
10: YCbCr 4:2:2
11: ITU656 YCbCr 4:2:2</td><td>RW</td></tr>
<tr><td>15</td><td>INV_DAT</td><td>Inverse every bit of input data.
0: not inverse; 1: inverse.</td><td>RW</td></tr>
<tr><td>14</td><td>VSP</td><td>VSYNC polarity selection. When VSYNC signal is input from pin CIM_VSYNC, this bit specifies the VSYNC signal active level and leading edge. When VSYNC is retrieved from SAV&EAV, this bit is ignored.
0: VSYNC signal active high, VSYNC signal leading edge is rising edge
1: VSYNC signal active low, VSYNC signal leading edge is falling edge</td><td>RW</td></tr>
<tr><td>13</td><td>HSP</td><td>Specifies the HSYNC signal active level and leading edge.
0: HSYNC signal active high, HSYNC signal leading edge is rising edge
1: HSYNC signal active low, HSYNC signal leading edge is falling edge</td><td>RW</td></tr>
<tr><td>12</td><td>PCP</td><td>Specifies the PCLK working edge.
0: Data is sampled by PCLK rising edge
1: Data is sampled by PCLK falling edge</td><td>RW</td></tr>
<tr><td>11:10</td><td>BURST_
TYPE</td><td>DMA burst type.
00: INCR4
01: INCR8</td><td>RW</td></tr>
</table>

<table>
<tr><td></td><td></td><td>10: INCR16
11: INCR32</td><td></td></tr>
<tr><td>9</td><td>DUMMY</td><td>DUMMY zero function. When DUMMY is 1, CIM hardware adds one byte zero to every 3 input data bytes to form 32-bit data.
0: DUMMY zero function disabled
1: DUMMY zero function enabled</td><td>RW</td></tr>
<tr><td>8</td><td>E_VSYNC</td><td>External / internal VSYNC selection. When DSM is ITU656Progressive Mode, VSYNC can be external (provided by sensor) or internal (retrieved from SAV&EAV). This bit only valid for ITU656Progressive Mode; In other DSM modes, this bit should always be 0.
0: Internal VSYNC mode, pin CIM_VSYNC is ignored
1: External VSYNC mode, VSYNC is provided by image sensor via pin CIM_VSYNC</td><td>RW</td></tr>
<tr><td>7</td><td>LM</td><td>Line Mode for ITU656.
0: EAV is before SAV in each line
1: SAV is before EAV in each line</td><td>RW</td></tr>
<tr><td>6:4</td><td>PACK</td><td>Data packing mode, pack 8-bit input data into 32-bit data for FIFO.
<table>
<tr><th>PACK</th><th>Bypass Mode</th><th>CSC Mode</th></tr>
<tr><td>3’b000</td><td>0x 11 22 33 44</td><td>0x Y_0 Cb Y_1 Cr</td></tr>
<tr><td>3’b001</td><td>0x 22 33 44 11</td><td>0x Cb Y_1 Cr Y_0</td></tr>
<tr><td>3’b010</td><td>0x 33 44 11 22</td><td>0x Y_1 Cr Y_0Cb</td></tr>
<tr><td>3’b011</td><td>0x 44 11 22 33</td><td>0x Cr Y_0 Cb Y_1</td></tr>
<tr><td>3’b100</td><td>0x 44 33 22 11</td><td>0x Cr Y_1 Cb Y_0</td></tr>
<tr><td>3’b101</td><td>0x 33 22 11 44</td><td>0x Y_1 Cb Y_0 Cr</td></tr>
<tr><td>3’b110</td><td>0x 22 11 44 33</td><td>0x Cb Y_0 Cr Y_1</td></tr>
<tr><td>3’b111</td><td>0x 11 44 33 22</td><td>0x Y_0 Cr Y_1 Cb</td></tr>
</table>
In this table, 0x11, 0x22, 0x33 and 0x44 mean the received data from the sensor, 0x11 is received first and 0x44 is received last, and Y0 is received before Y1.</td><td>6:4</td></tr>
<tr><td>3</td><td>FP</td><td>Field flag polarity selection. When ITU656 progressive stream is input, this bit specifies the field flag active level. When other modes are used, this bit is ignored.
0: Field flag active low
1: Field flag active high</td><td>RW</td></tr>
<tr><td>2</td><td>BYPASS</td><td>0: enable CIM CSC
1: disable CIM CSC</td><td>RW</td></tr>
<tr><td>1:0</td><td>DSM</td><td>Data sample mode. Please refer to the table below.</td><td>RW</td></tr>
</table>

			DSM	Description		
			2'b00	ITU656Progressive Mode		
			2'b01	ITU656Interlace Mode		
			2'b10	Gated Clock Mode		
			2'b11	Reserved		

9.2.2 CIM Control Register (CIMCR)

CIMCR **0x13060004**

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
	EEOF_LINE												FRC				DMA_EEOFM	WINE	VDDM	DMA_SOFM	DMA_EOFM	DMA_STOPM	RF_TRIGM	RF_OFM	DMA_SYNC	Reserved	H_SYNC	PPW		DMA_EN	RF_RST	ENA
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	EEOF_LINE	When EEOF_LINE lines data has been transferred of a frame, the EEOF flag will be set, and the EEOF interrupt will occur.	RW
19:16	FRC	CIM frame rate control. If FRC = n, CIM sampling one frame from every (n+1) frames from the sensor.	RW
15	DMA_EEOFM	The control bit to enable EEOF interrupt.	RW
14	WINE	To enable window-image. Used to indicate whether the registers CIMSIZE and CIMOFFSET work or not. 0: the value in CIMSIZE and CIMOFFSET will be ignored 1: the value in CIMSIZE and CIMOFFSET will be used	RW
13	VDDM	The control bit to enable VDD interrupt. 0: disable; 1: enable.	RW
12	DMA_SOFM	The control bit to enable DMA_SOF interrupt. 0: disable; 1: enable.	RW
11	DMA_EOFM	The control bit to enable DMA_EOF interrupt. 0: disable; 1: enable.	RW
10	DMA_STOPM	The control bit to enable DMA_STOP interrupt. 0: disable; 1: enable.	RW
9	RF_TRIGM	The control bit to enable RXF_TRIG interrupt. 0: disable; 1: enable.	RW
8	RF_OFM	The control bit to enable RX_OF interrupt. 0: disable; 1: enable.	RW
7	DMA_SYNC	The control bit to enable DMA synchronization. 0: The valid data input to CIM will be transferred by DMA to	RW

		external memory 1: When a new descriptor-DMA transfer starts with writing CIMDA, a frame synchronization will be done, and the data in RXFIFO will be ignored	
6	Reserved	Writing has no effect, read as zero.	R
5	H_SYNC	Horizontal Sync Enable. 0: disable 1: enable It is only used when CIMCFG.SEP is 1.	RW
4:3	PPW	Pixels per Word. Only used when WINE is 1. 00: 2 pixels per 1 word 01: 1 pixel per 1 word 10: 1 pixel per 2 word 11: reserved	RW
2	DMA_EN	Enable / disable the DMA function. 0: disable DMA; 1: enable DMA.	RW
1	RF_RST	RXFIFO software reset. Setting 1 to RXF_RST can reset RXFIFO immediately. Setting 0 to RXF_RST can stop resetting RXFIFO. After reset, RXFIFO is empty.	RW
0	ENA	Enable / disable the CIM module. Setting 1 to ENA can enable CIM. When CIM is working, clear ENA to 0 can stop CIM immediately. 0: CIM is not enabled, or disable CIM immediately 1: CIM is enabled, or enabling CIM	RW

9.2.3 CIM Control Register 2 (CIMCR2)

CIMCR2 **0x13060050**

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																										OP		Reserved	OPE	EMM	APM
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
31:6	Reserved	Writing has no effect, read as zero.	R

<table>
<tr><td>5:4</td><td>OP</td><td>Optional Priority Configuration. Only used when OPE is set to 1.

<table><tr><th>PG</th><th>CIM AHB Priority</th><th>Number of Data in FIFO</th></tr><tr><td>2’b00</td><td>0
1
2
3</td><td>n <= 8
8 < n <= 16
16 < n <= 32
32 < n</td></tr><tr><td>2’b01</td><td>0
1
2
3</td><td>n <= 16
16 < n <= 32
32 < n <= 64
64 < n</td></tr><tr><td>2’b10</td><td>0
1
2
3</td><td>n <= 32
32 < n <= 64
64 < n <= 96
96 < n</td></tr><tr><td>2’b11</td><td>0
1
2
3</td><td>n <= 64
64 < n <= 96
96 < n <= 128
128 < n</td></tr></table>
It is suggested to use 2’b10.</td><td>RW</td></tr>
<tr><td>3</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>2</td><td>OPE</td><td>Optional Priority Mode Enable Control. Only used when APM is 1.
0: CIM calculates the priority according to the fifo status
1: CIM calculates the priority according to OPG which is configured by software</td><td>RW</td></tr>
<tr><td>1</td><td>EME</td><td>Emergency Mode Enable Control.
0: Emergency Mode Disable
1: Emergency Mode Enable</td><td>RW</td></tr>
<tr><td>0</td><td>APM</td><td>Auto Priority Mode Enable Control.
0: Auto priority mode disable. CIM uses the priority set by arbiter
1: Auto priority mode enable. CIM can use the priority according the fifo status</td><td>RW</td></tr>
</table>

9.2.4 CIM Status Register (CIMST)

CIMST 0x13060008

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				Cr_RF_OF	Cr_RF_TRIG	Cr_RF_EMPTY	Reserved					Cb_RF_OF	Cb_RF_TRIG	Cb_RF_EMPTY	Reserved					Y_RF_OF	Y_RF_TRIG	Y_RF_EMPTY	Reserved	DMA_EEOF	DMA_SOF	DMA_EOF	DMA_STOP	RF_OF	RF_TRIG	RF_EMPTY	VDD
RST	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

Bits	Name	Description	RW
31:28	Reserved	Writing has no effect, read as zero.	R
27	Cr_RF_OF	Cr_RXFIFO over flow. When Cr_RXFIFO over flow happens, Cr_RX_OF is set 1. Can generate interrupt if CIMCR.RF_OFM bit is set. Write 0 to this bit to clear.	RW
26	Cr_RF_TRIG	Cr_RXFIFO trigger. Indicates whether Cr_RXFIFO meet the trigger value or not. Can generate interrupt if CIMCR.RF_TRIGM bit is set. 0: Cr_RXFIFO does not meets the trigger value 1: Cr_RXFIFO meets the trigger value	R
25	Cr_RF_EMPTY	Cr_RXFIFO empty. Indicates whether Cr_RXFIFO is empty or not. After reset, RXFIFO is empty, and Cr_RX_EMPTY is 1. 0: Cr_RXFIFO is not empty 1: Cr_RXFIFO is empty	R
24:20	Reserved	Writing has no effect, read as zero.	R
19	Cb_RF_OF	Cb_RXFIFO over flow. When Cb_RXFIFO over flow happens, Cb_RX_OF is set 1. Can generate interrupt if CIMCR.RF_OFM bit is set. Write 0 to this bit to clear.	RW
18	Cb_RF_TRIG	Cb_RXFIFO trigger. Indicates whether Cb_RXFIFO meet the trigger value or not. Can generate interrupt if CIMCR.RF_TRIGM bit is set. 0: Cb_RXFIFO does not meets the trigger value 1: Cb_RXFIFO meets the trigger value	R
17	Cb_RF_EMPTY	Cb_RXFIFO empty. Indicates whether Cb_RXFIFO is empty or not. After reset, Cb_RXFIFO is empty, and Cb_RX_EMPTY is 1. 0: Cb_RXFIFO is not empty 1: Cb_RXFIFO is empty	R
16:12	Reserved	Writing has no effect, read as zero.	R

11	Y_RF_OF	Y_RXFIFO over flow. When Y_RXFIFO over flow happens, Y_RX_OF is set 1. Can generate interrupt if CIMCR.RF_OFM bit is set. Write 0 to this bit to clear.	RW
10	Y_RF_TRIG	Y_RXFIFO trigger. Indicates whether Y_RXFIFO meet the trigger value or not. Can generate interrupt if CIMCR.RF_TRIGM bit is set. 0: Y_RXFIFO does not meets the trigger value 1: Y_RXFIFO meets the trigger value	R
9	Y_RF_EMPTY	Y_RXFIFO empty. Indicates whether Y_RXFIFO is empty or not. After reset, Y_RXFIFO is empty, and Y_RX_EMPTY is 1. 0: Y_RXFIFO is not empty 1: Y_RXFIFO is empty	R
8	Reserved	Writing has no effect, read as zero.	R
7	DMA_EEOF	When set to 1, indicate the DMA has transferred CIMCTRL.EEOF_LINE lines data of a frame. Write 0 to this bit to clear.	RW
6	DMA_SOF	When set to 1, Indicate the DMA start a transfer from RXFIFO to a frame buffer. Write 0 to this bit to clear.	RW
5	DMA_EOF	When set to 1, indicate the DMA complete a transfer from RXFIFO to a frame buffer. Write 0 to this bit to clear.	RW
4	DMA_STOP	When set to 1, indicate the DMA complete transferring data and stop the operation. Can generate interrupt if CIMCR.DMA_STOPM bit is set. Write 0 to this bit to clear.	RW
3	RF_OF	RXFIFO over flow. When RXFIFO over flow happens, RX_OF is set 1. Can generate interrupt if CIMCR.RF_OFM bit is set. Write 0 to this bit to clear.	RW
2	RF_TRIG	RXFIFO trigger. Indicates whether RXFIFO meet the trigger value or not. When the valid data number in RXFIFO reaches the trig value, RXF_TRIG is set 1; when the valid data number in RXFIFO do not reach the trig value, RXF_TRIG is set 0. Can generate interrupt if CIMCR.RF_TRIGM bit is set. 0: RXFIFO does not meets the trigger value 1: RXFIFO meets the trigger value	R
1	RF_EMPTY	RXFIFO empty. Indicates whether RXFIFO is empty or not. After reset, RXFIFO is empty, and RX_EMPTY is 1. 0: RXFIFO is not empty 1: RXFIFO is empty	R

0	VDD	CIM disable done. Indicate this module is disabled after clear the CIMCR.ENA bit to disable the CIM module. Can generate interrupt if CIMCR.DMA_VDDM bit is set. 0: CIM has not been disabled 1: CIM has been disabled Write 0 to this bit to clear.	RW

9.2.5 CIM Interrupt ID Register (CIMIID)

CIMIID **0x1306000C**

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
	FID																															
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
31:0	FID	Interrupt frame ID. Contains a copy of the Frame ID register (CIMFID) from the descriptor currently being processed when a DMA_SOF or DMA_EOF interrupt is generated. CIMIID is written to only when CIMCMD.SOFINT or CIMCMD.EOFINT is high. As such, the register is considered to be sticky and will be overwritten only when the associated interrupt is cleared by writing the CIM state register.	R

9.2.6 CIM Descriptor Address (CIMDA)

CIMIDA **0x13060020**

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
	NDA																															
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
31:0	NDA	Next descriptor physical address in external memory. DMAC gets the next descriptor according to it after finishing the current one. The target address Bits [3:0] must be zero to be aligned to 16-byte boundary.	RW

9.2.7 CIM Frame buffer Address Register (CIMFA)

CIMIFA **0x13060024**

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
	FPA																															
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
31:0	FPA	Frame buffer physical address in external memory when CIMCFG. SEP is 0. When starts CIM, DMA transfers data from RXFIFO to frame buffer. This address is increased by hardware automatically. Bits [6:0] must be zero to be aligned to 32-word boundary.	R

NOTE: CIMFA comes from DMA Descriptor, so here it is read-only.

9.2.8 CIM Frame ID Register (CIMFID)

CIMFID **0x13060028**

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
	FID																															
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
31:0	FID	Frame ID. The particular use of this field is up to the software. This ID will be copied to the CIMIID register when an interrupt occurs.	R

NOTE: CIMFID comes from DMA Descriptor, so here it is read-only.

9.2.9 CIM DMA Command Register (CIMCMD)

CIMFID 0x1306002C

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
	SOFINT	EOFINT	EEOFINTEn	STOP	Reserved				LEN																							
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	SOFINTEn	Interrupt enable for DMA starting a frame-buffer transfer. 1: DMA will set CIMSTATE.DMA_SOF when start of a frame-buffer transfer When one frame uses several buffers, it is suggested to set SOFINTEn of first buffer only.	R
30	EOFINTEn	Interrupt enable for DMA ending a frame-buffer transfer. 1: DMA will set CIMSTATE.DMA_EOF when CIMCMD.LEN is decreased to 0, which means end of a frame-buffer transfer When one frame uses several buffers, it is suggested to set EOFINTEn of last buffer only.	R
29	EEOFINTEn	Interrupt enable for DMA issuing an earlier eof interrupt.	R
28	STOP	DMA stop. When DMA complete transferring data, STOP bit decides whether DMA should loading next descriptor or not. 0: DMA start loading next descriptor 1: DMA stopped, and CIMSTATE.DMA_STOP bit is set 1 by hardware	R
27	OFRCVEN	Auto recovery enable when there is RXFIFO overflow. 0: No auto recovery when overflow occurs, thus the software should do something 1: Auto recovery enable, the hardware will correct the overflow	
26:24	Reserved	Writing has no effect, read as zero.	R
23:0	LEN	Length of transfer in words. Indicate the number of words to be transferred by DMA to a frame buffer. LEN = 0 is not valid. DMA transfers data according to LEN. Each time one or more word(s) been transferred, LEN is decreased automatically.	R

9.2.10 CIM Window-image Size (CIMSIZE)

CIMFID 0x13060030

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			LPF													Reserved			PPL												
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
31:29	Reserved	Writing has no effect, read as zero.	R
28:16	LPF	Lines per frame for CIM output.	RW
15:13	Reserved	Writing has no effect, read as zero.	R
12:0	PPL	PPL must be multiples of 2. In fact, the number of CIM output data in word is equal to PPL/2.	RW

NOTE:

When CIMCFG.SEP is 1, the total pixel number of window-size must be multiple of 4 or 8.

- When output data format is YCBCR4:4:4, it must be multiple of 4.
- When output data format is YCBCR4:2:2, it must be multiple of 8.

9.2.11 CIM Image Offset (CIMOFFSET)

CIMFID 0x13060034

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			V_OFFSET													Reserved			H_OFFSET												
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
31:28	Reserved	Writing has no effect, read as zero.	R
27:16	V_OFFSET	Vertical offset.	RW
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	H_OFFSET	Horizontal offset. It should be an even number.	RW

9.2.12 CIM Y Frame buffer Address Register (CIMYFA)

CIMIYFA 0x13060038

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
	YFPA																															
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
31:0	YFPA	Y Frame buffer physical address in external memory when CIMCFG. SEP is 1. When starts CIM, DMA transfers data from Y_RXFIFO to frame buffer. This address is increased by hardware automatically. Bits [6:0] must be zero to be aligned to 32-word boundary.	R

9.2.13 CIM Y DMA Command Register (CIMYCMD)

CIMYCMD 0x1306003C

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
	SOFINT	EOFINT	EEOFINTEn	STOP	Reserved												YLEN															
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	SOFINTEn	Interrupt enable for DMA starting a frame-buffer transfer. 1: DMA will set CIMSTATE.DMA_SOF when start of a frame-buffer transfer When one frame uses several buffers, it is suggested to set SOFINTEn of first buffer only.	R
30	EOFINTEn	Interrupt enable for DMA ending a frame-buffer transfer. 1: DMA will set CIMSTATE.DMA_EOF when CIMYCMD.YLEN and CIMCbCMD.CbLEN and CIMCrCMD.CrLEN are decreased to 0, which means end of a frame-buffer transfer When one frame uses several buffers, it is suggested to set EOFINTEn of last buffer only.	R
29	EEOFINTEn	Interrupt enable for DMA issuing an earlier eof interrupt.	R
28	STOP	DMA stop. When DMA complete transferring data, STOP bit decides whether DMA should loading next descriptor or not.	R

		0: DMA start loading next descriptor 1: DMA stopped, and CIMSTATE.DMA_STOP bit is set 1 by hardware	
27	OFRCVEN	Auto recovery enable when there is RXFIFO overflow. 0: No auto recovery when overflow occurs, thus the software should do something 1: Auto recovery enable, the hardware will correct the overflow DMA will do a frame synchronization, and retransfer the current descriptor.	
26:24	Reserved	Writing has no effect, read as zero.	R
23:0	YLEN	Length of transfer in words. Indicate the number of words to be transferred by DMA to a frame buffer. YLEN = 0 is not valid. DMA transfers data according to YLEN. Each time one or more word(s) been transferred, YLEN is decreased automatically.	R

9.2.14 CIM Cb Frame buffer Address Register (CIMCBFA)

CIMICBFA 0x13060040

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
	CbFPA																															
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
31:0	CbFPA	Cb Frame buffer physical address in external memory when CIMCFG.SEP is 1. When starts CIM, DMA transfers data from Cb_RXFIFO to frame buffer. This address is increased by hardware automatically. Bits [6:0] must be zero to be aligned to 32-word boundary.	R

9.2.15 CIM Cb DMA Command Register (CIMCBCMD)

CIMCBCMD 0x13060044

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								CbLEN																							
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
31:24	Reserved	Writing has no effect, read as zero.	R
23:0	CbLEN	Cb Length of transfer in words. Indicate the number of words to be transferred by DMA to a frame buffer. CbLEN = 0 is not valid. DMA transfers data according to CbLEN. Each time one or more word(s) been transferred, CbLEN is decreased automatically.	R

9.2.16 CIM Cr Frame buffer Address Register (CIMCRFA)

CIMICRFA 0x13060048

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
	CrFPA																															
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
31:0	CrFPA	Cr Frame buffer physical address in external memory when CIMCFG. SEP is 1. When starts CIM, DMA transfers data from Cr RXFIFO to frame buffer. This address is increased by hardware automatically. Bits [6:0] must be zero to be aligned to 32-word boundary.	R

9.2.17 CIM DMA Cr Command Register (CIMCRCMD)

CIMCRCMD 0x1306004C

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								CrLEN																							
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
31:24	Reserved	Writing has no effect, read as zero.	R
23:0	CrLEN	Cr Length of transfer in words. Indicate the number of words to be transferred by DMA to a frame buffer. CrLEN = 0 is not valid. DMA transfers data according to CrLEN. Each time one or more word(s) been transferred, CrLEN is decreased automatically.	R

9.3 CIM Data Sampling Modes

CIM module supports several types of data sampling mode. The modes and the corresponding signals used are shown in the following diagram:

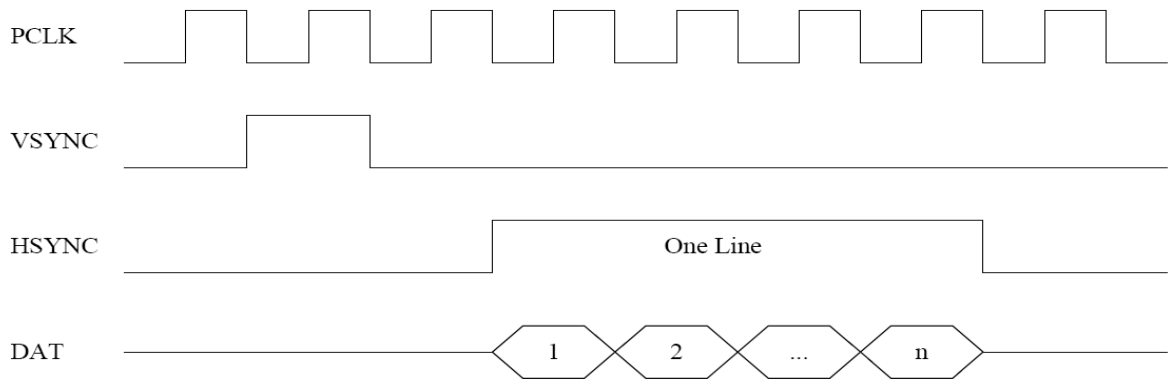
Table 9-3 The modes and the corresponding signals used

Mode \ Signals	CIM_VSYNC	CIM_HSYNC	CIM_PCLK	CIM_DATA
Gated Clock Mode	Y	Y	Y	Y
ITU656 Interlace Mode	N	N	Y	Y
ITU656 Progressive Mode	N	N	Y	Y

9.3.1 Gated Clock Mode

CIM_VSYNC, CIM_HSYNC, and CIM_PCLK signals are used in this mode.

A frame starts with VSYNC leading edge, then HSYNC goes active and holds the entire line. Data is sampled at the valid edge of PCLK when HSYNC is active; That means, HSYNC functions like “data enable” signal. Please refer to the figure below.



Gated Clock Input Timing

The VSYNC leading edge, HSYNC active HIGH or LOW, PCLK valid edges are programmable.

9.3.2 ITU656 Interlace Mode

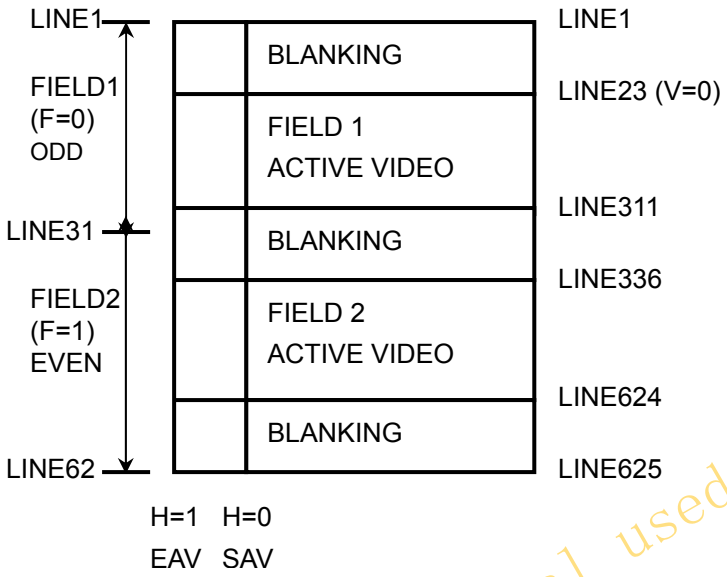
In this mode, CIM_PCLK and CIM_DAT signals are used, CIM_VSYNC, CIM_HSYNC signals are ignored.

CIM utilizes the SAV & EAV code within ITU656data stream to get active video data.

The following diagrams and tables are quoted from ITU656standard. For more information about

ITU656, please refer to ITU656 standard.

9.3.2.1 PAL Timing



LINE NUMBER	F	V	H (EAV)	H (SAV)	P0, P1, P2, P3
1-22	0 Field 1	1: blanking	1: in EAV, to indicate the end of active video	0: in SAV, to indicate the start of active video	Protection bits
23-310		0: video data			
311-312		1: blanking			
313-335	1 Field 2	1: blanking			
336-623		0: video data			
624-625		1: blanking			

Figure 9-1 Typical BT.656 Vertical Blanking Intervals for 625/50 Video Systems

9.3.2.2 Coding for Protection Bits

F	V	H	P3	P2	P1	P0
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	1	1
1	0	1	1	0	1	0
1	1	0	1	1	0	0

1	1	1	0	0	0	1

9.3.3 ITU656 Progressive Mode

CIM_PCLK and CIM_DAT signals are used in this mode. CIM_HSYNC signal is ignored.

CIM_VSYNC is optional in this mode. When the start of frame information is retrieved from SAV and EAV, it is known as internal VSYNC mode. When CIM_VSYNC is provided by sensor directly, it is known as external VSYNC mode. CIM supports both internal and external VSYNC modes.

ITU656Progressive Mode is a kind of Non-Interlace Mode. The image data are encoded within only one field. Most sensors support ITU656Progressive Mode.

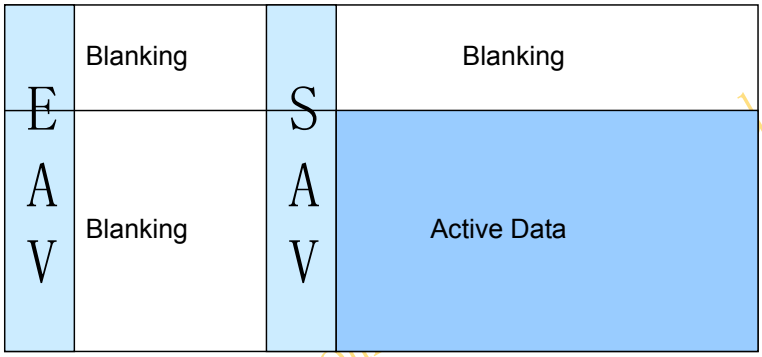


Figure 9-2 ITU656 Progressive Mode

9.4 DMA Descriptors

9.4.1 4-Word Descriptor

Used when output is packaged frame format.

A DMA descriptor is a 4-word block corresponding to the four DMA registers – CIMDA, CIMFA, CIMFID, and CIMCMD, aligned on 4-word (16-byte) boundary, in external memory:

word [0] contains the physical address for next CIMDA
word [1] contains the value for CIMFID
word [2] contains the physical address for CIMFA
word [3] contains the value for CIMCMD

Software must write the physical address of the first descriptor to CIMDA before enabling the CIM. Once the CIM is enabled, the first descriptor is read, and all 4 registers are written by the DMAC. The next DMA descriptor pointed to by CIMDA is loaded into the registers after all data for the current descriptor has been transferred.

9.4.2 8-Word Descriptor

Used when output is separated frame format.

A DMA descriptor is a 8-word block corresponding to the four DMA registers – CIMDA, CIMFA, CIMFID, and CIMCMD, aligned on 8-word (32-byte) boundary, in external memory:

word [0] contains the physical address for next CIMDA
word [1] contains the value for CIMFID
word [2] contains the physical address for CIMYFA
word [3] contains the value for CIMYCMD
word [4] contains the physical address for CIMCBFA
word [5] contains the value for CIMCBCMD
word [6] contains the physical address for CIMCRFA
word [7] contains the value for CIMCRCMD

Software must write the physical address of the first descriptor to CIMDA before enabling the CIM. Once the CIM is enabled, the first descriptor is read, and all 8 registers are written by the DMAC. The next DMA descriptor pointed to by CIMDA is loaded into the registers after all data for the current descriptor has been transferred.

NOTE: If only one frame buffer is used in external memory, the CIMDA field (word [0] of the DMA descriptor) must point back to itself. That is to say, the value of CIMDA is the physical address of itself.

9.5 Interrupt Generation

CIM has next interrupt sources:

Step 1. RXFIFO FULL Interrupt. (RF_TRIG)

When the valid data number of RXFIFO reaches trigger value, CIMST.RF_TRIG bit is set. At the same time, if RF_TRIGM is 1, RF_TRIG interrupt is generated.

Step 2. RXFIFO Over Flow Interrupt. (RF_OF)

When the valid data number of RXFIFO reaches 32 and one more data are written to RXFIFO, CIMST.RF_OF bit is set. At the same time, if RF_OFM is 1, RF_OF interrupt is generated.

Step 3. DMA Start Of Frame Data Transferring Interrupt. (DMA_SOF)

When the CIMCMD.SOFINT bit is 1 and DMA start transferring the first data from RXFIFO to frame buffer, CIMST.DMA_SOF bit is set. At the same time, if DMA_SOFM is 1, DMA_SOF interrupt is generated.

Step 4. DMA End Of Frame Data Transferring Interrupt. (DMA_EOF)

When the CIMCMD.EOFINT bit is 1 and DMA complete transferring the last data from RXFIFO to frame buffer, CIMST.DMA_EOF bit is set. At the same time, if DMA_EOFM is 1, DMA_EOF interrupt is generated.

Step 5. DMA Stop Transferring Interrupt. (DMA_STOP)

When the CIMCMD.STOP bit is 1 and DMA complete transferring the last data from RXFIFO to frame buffer, CIMST.DMA_STOP bit is set. At the same time, if DMA_STOPM is 1, DMA_STOP interrupt is generated.

Step 6. CIM Disable Done Interrupt. (VDD)

When disable the module by clearing the CIMCR.ENA, the module should be disabled after transferring current valid data. Then set the CIMST.VDD bit, at the same time, if VDDM is set, VDD interrupt is generated.

9.6 Software Operation

9.6.1 Enable CIM with DMA

Step 1. Configure register CIMCFG.
Step 2. Prepare frame buffer and descriptors.
Step 3. Configure register CIMDA.
Step 4. Clear state register: write 0 to register CIMSTATE.
Step 5. Reset RXFIFO: configure register CIMCTRL with DMA_EN=1, RXF_RST=1, ENA=0.
Step 6. Stop resetting RXFIFO: configure register CIMCTRL with DMA_EN=1, RXF_RST=0, ENA=0.
Step 7. Enable CIM: configure register CIMCTRL with DMA_EN=1, RXF_RST=0, ENA=1.

9.6.2 Enable CIM without DMA

1 Configure register CIMCFG.
2 Clear state register: write 0 to register CIMSTATE.
3 Reset RXFIFO: configure register CIMCTRL with DMA_EN=0, RXF_RST=1, ENA=0.
4 Stop resetting RXFIFO: configure register CIMCTRL with DMA_EN=0, RXF_RST=0, ENA=0.
5 Enable CIM: configure register CIMCTRL with DMA_EN=0, RXF_RST=0, ENA=1.

9.6.3 Disable CIM

Method 1:

Step 1. Configure register CIMCTRL with RXF_RST=0, ENA=0. // quick disable
Step 2. Clear state register: write 0 to register CIMSTATE.

Method 2:

When DMA is enabled, the following sequence is recommended:

Step 1. Configure descriptor with STOP = 1.
Step 2. Wait DMA_STOP interrupt, then write 0 to CIMCTRL.ENA.
Step 3. Clear state register: write 0 to register CIMSTATE.

9.6.4 CIM Priority

There are three methods to use CIM priority to transfer data from receiving fifo to external memory, called bus, module and software mode.

9.6.4.1 Bus Priority Mode

CIM uses the priority distributed by AHB bus arbiter.

Step:

Set CIMCR2.APM to 1'b0.

9.6.4.2 Module Priority Mode

CIM module chooses the priority automatically, which is according to the receiving fifo status only. It is recommended. It is recommended.

Steps:

1 Set CIMCR2.OPE to 1'b0.

2 Set CIMCR2.APM to 1'b1.

9.6.4.3 Software Priority Mode

CIM module chooses the priority according to the receiving fifo status and CIMCR2.OR.

Steps:

1 Set CIMCR2.OP to the value expected.

2 Set CIMCR2.OPE to 1'b1.

3 Set CIMCR2.APM to 1'b1.

10 Internal CODEC Interface

10.1 Overview

This chapter describes the embedded audio CODEC in the processor and related software interface.

This embedded CODEC is an I2S audio CODEC. AIC module is an interface to this CODEC in audio data replaying and recording. Several memory mapped registers are used to access this embedded CODEC, and write/read these registers could access the CODEC's internal control and configure registers that is using 12 MHz clock.

10.1.1 Features

The following are internal CODEC features:

- 24 bits ADC and DAC
- Headphone load up to 16 Ohm
- Sample frequency supported: 8k, 11.025k, 12k, 16k, 22.05k, 24k, 32k, 44.1k, 48k, and 96k
- Stereo line input
- DAC to HP path: Power consumption: 17.6mW, THD: -65dB @17.6mW /16Ohm
- DAC to stereo line output path @10kOhm: SNR: 95dB A-Weighted, THD: -80dB @FS-1dB
- Line input to ADC path: SNR: 95dB A-Weighted, THD: -80dB @FS-1dB
- Separate power-down modes for ADC and DAC path with several shutdown modes
- Reduction of audible glitches systems: Pop Reduction system, Soft Mute mode
- Output short circuit protection
- Support Capacitor-coupled and Capacitor-less mode headphone connection

TBD = parameter or document section to be defined later on
TBC = parameter or document section subject to change
TO BE COMPLETED = section to be filled or subject to change

10.1.2 Signal Descriptions

CODEC has max 13 analog signal IO pins and 4 power pins on chip. They are listed and described in the flowing table.

Table 10-1 CODEC signal IO pin description

Pin Names	IO	Pin Description	Power
MICP1	AI	Microphone mono differential analog input 1 (MIC1), positive pin.	AVDCDC
MICN1	AI	Microphone mono differential analog input 1 (MIC1), negative pin.	AVDCDC
MICP2	AI	Microphone mono differential analog input 2 (MIC2), positive pin.	AVDCDC
MICN2	AI	Microphone mono differential analog input 2 (MIC2), negative pin.	AVDCDC
MICBIAS	AO	Microphone bias.	AVDCDC
AIL	AI	Left line single-ended analog input.	AVDCDC
AIR	AI	Right line single-ended analog input.	AVDCDC
AOLOP	AO	Differential line output, positive pin.	AVDCDC
AOLON	AO	Differential line output, negative pin.	AVDCDC
AOHPL	AO	Left headphone single ended analog output.	AVDHP
AOHPR	AO	Right headphone single ended analog output.	AVDHP
AOHPM	AO	Headphone common mode output.	AVDHP
AOHPMS	AI	Headphone common mode sense input.	AVDHP
VCAP	AO	Voltage Reference Output. An 10μF ceramic or tantalum capacitor in parallel with a 0.1μF ceramic capacitor attached from this pin to AVSCDC eliminates the effects of high frequency noise.	AVDCDC
AVDHP	P	Headphone amplifier power, 2.5V.	-
AVSHP	P	Headphone amplifier ground.	-
AVDCDC	P	CODEC analog power, 2.5V, inter signal VREFP.	-
AVSCDC	P	CODEC analog ground, inter signal VREFN.	-
HPSENSE	AI	Headphone jack sense.	AVDHP
DMIC_IN	DI	Digital microphone data input pin.	AVDCDC
DMIC_CLK	DO	Digital microphone clock output pin.	AVDCDC

NOTES:

1 AVDHP = 2.5v (typ). AVDCDC= 2.5v (typ).
2 Inter signal VREFP is connected to AVDCDC, inter signal VREFN is connected to AVSCDC.
3 Please refer to data sheet of the chip for details.
4 DMIC_IN is GPIO : PB18 , MIC_CLK is GPIO : PB19. Please refer to GPIO specification for

these pins operating.

10.1.3 Block Diagram

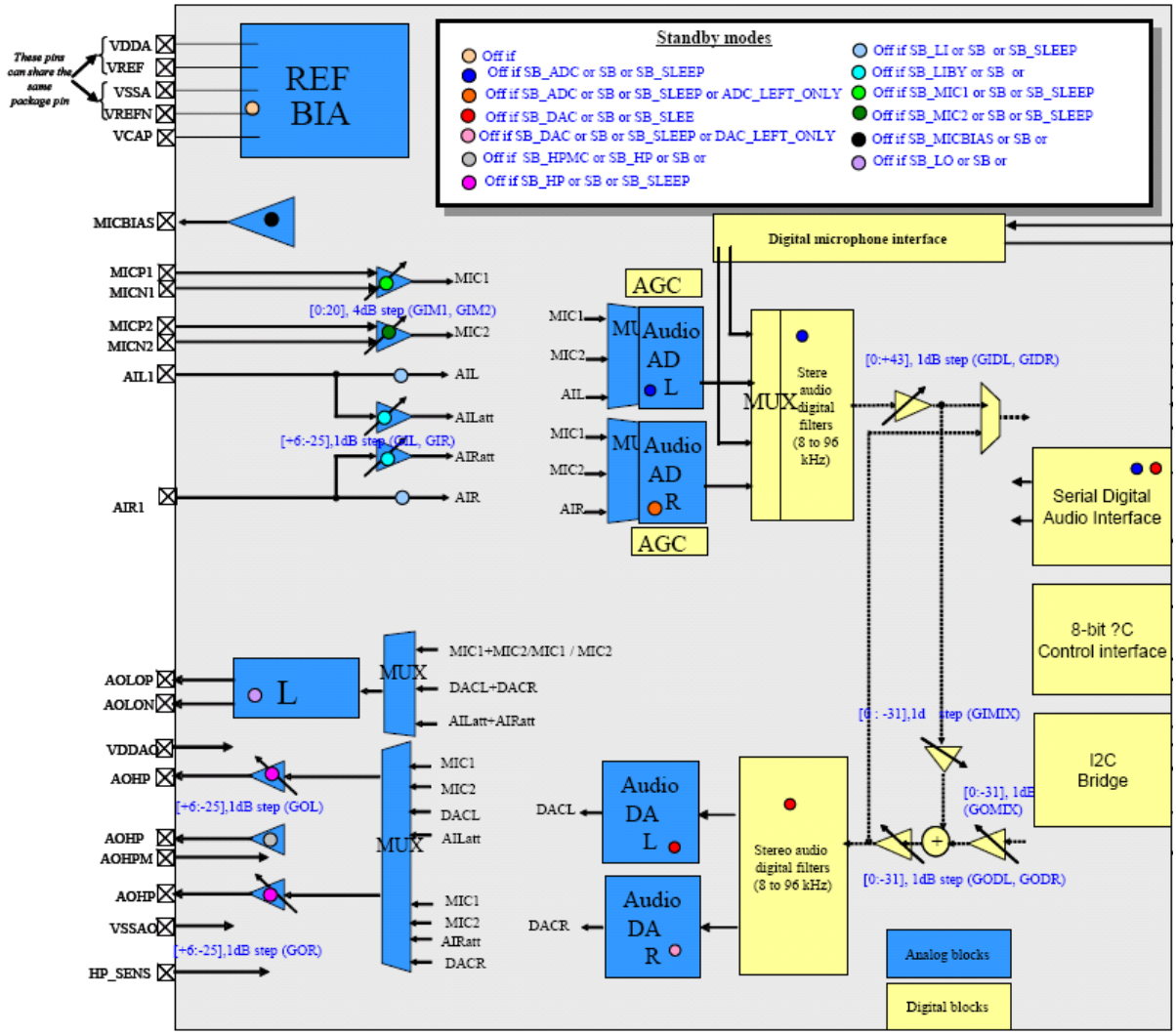


Figure 10-1 CODEC block diagram

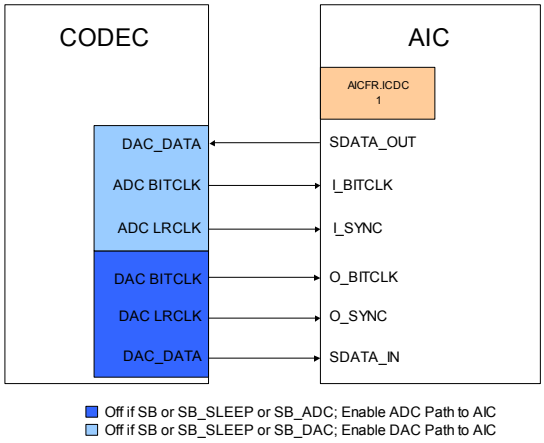


Figure 10-2 Internal CODEC works with AIC

10.2 Mapped Register Descriptions

The internal CODEC software interface includes 2 registers. They are mapped in IO memory address space of AIC module so that program can access them to control the operations of the CODEC.

Table 10-2 Internal CODEC Mapped Registers Description (AIC Registers)

Name	Description	RW	Reset value	Address	Size
RGADW	Address, data in and write command for accessing to internal registers of internal embedded CODEC.	RW	0x00000000	0x100200A4	32
RGDATA	The read out data and interrupt request status of Internal registers data in the internal embedded CODEC.	R	0x00000000	0x100200A8	32

NOTES:

1 All these registers are AIC Registers, because they are mapped in AIC IO memory address.

2 RGADW contains data, address and write command to the internal registers of the internal CODEC.

3 RGDATA returns the internal register value of the internal CODEC and interrupt request status.

10.2.1 CODEC internal register access control (RGADW)

RGADW contains address, data and write command to the internal registers of the internal embedded CODEC.

RGADW 0x100200A4

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															RGWR	Reserved	RGADDR							RGDIN							
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
31:17	Reserved	Writing has no effect, read as zero.	R
16	RGWR	Write 1 to this bit issues writing to CODEC's internal register process. This bit keeps value 1 until the current writing process is finished. A register read or a new register writing process cannot be issued before the previous writing process finished. In another word, it should not write to RGADW before RGADW.RGWR becomes 0. A writing process takes max of 0.17us plus 1 PCLK cycle. Write 0 to this bit is ignored.	RW
15	Reserved	Writing has no effect, read as zero.	R
14:8	RGADDR	When it issues a writing to CODEC's internal register command, i.e. RGWR=1, this field specifies the register's address. In addition, this field also decides the address of the register's data out at any time.	RW
7:0	RGDIN	When it issues a writing to CODEC's internal register command, i.e. RGWR=1, this field contains the data to be written to the register.	RW

NOTES:

1 It is strong suggesting verifying the data (using read RGDATA below) after writing it to internal register of CODEC. When RGDATA returns the right data which writing to the address, the writing process is finish.

2 Please notice that AIC needs SYS_CLK (refers to AIC spec), when write new value to or read from CODEC internal registers.

10.2.2 CODEC internal register data output (RGDATA)

RGDATA returns the internal register value of the internal embedded CODEC and interrupt request status.

RGDATA 0x100200A8

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																							IRQ	RGDOUT							
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
31:9	Reserved	Writing has no effect, read as zero.	R
8	IRQ	This field returns the internal embedded CODEC's interrupt request. IRQ: 0 — No CODEC's interrupt request found. IRQ: 1 — CODEC's interrupt request is pending.	R
7:0	RGDOUT	This field returns the value of the internal register in internal embedded CODEC. As the RGADW.RGADDR field specifies the register's address.	R

NOTE: AIC needs SYS_CLK (refers to AIC spec), when write new value to or read from CODEC internal registers.

10.3 Operation

The internal embedded CODEC is controlled its internal registers. These registers can be accessed by through memory-mapped registers, RGADW and RGDATA, just like L3 bus or I2C bus for an external CODEC. AIC's BITCLK and SYNC are from/to the CODEC and is controlled by CKCFG.SELAD register. The audio data transferring, i.e. audio replaying and recording, is down by AIC. AIC still takes the role of I2S controller. We will refer to many AIC operations and registers in the following audio operation descriptions, please reference to AIC spec for the details.

This is a guide for software.

10.3.1 Access to internal registers of the embedded CODEC

The embedded CODEC is controlled through its internal registers. RGADW and RGDATA are used to write to and read from these registers. Here are some examples.

Example 1. Write to a CODEC internal register.

Step 1: RGADW.RGWR == 0.

Step 2: If not, go to step 1.

Step 3: Write to RGADW and make it.

RGADW.RGDIN = <data to be written to the register>.

RGADW.RGADDR = <the register's address >.

Step 4: Write to RGADW to commit the writing operation.

RGADW.RGWR = 1.

Example 2. Read from a CODEC internal register.

Step 1: RGADW.RGWR == 0.

Step 2: If not, go to step 1.

Step 3: write to RGADW and make it.

RGADW.RGWR = 0.

RGADW.RGDIN = <don't care>.

RGADW.RGADDR = <the register's address>.

Step 4: read RGDATA.DOUT, which returns the register's content.

10.3.2 CODEC controlling and typical operations

This section is some typical operations. We are assumed the power supply of CODEC is on, and CODEC is in STANDBY mode, CRR is configured for audio Ramping system.

Before using any of these operations, make sure AIC is configured properly as list below:

1 Make AIC to use internal CODEC mode:
   AICFR.ICDC = 1; Use internal CODEC.
   AICFR.AUSEL = 1; Use I2S mode.
   AICFR.BCKD = 0; CODEC input BIT_CLK to AIC.
   AICFR.SYNCD = 0; CODEC input SYNC to AIC.
   I2SCR.AMSL = 1; Use I2S operation mode.
   I2SCR.ESCLK = 1; Open SYS_CLK to internal CODEC. (if using PLL Clock)
2 Make sure AICCR.FLUSH = 0; AICFR.RST = 0; AICCR.ENLBF = 0.
3 Clear AICSR.ROR, AICSR.TUR, AICSR.RFS, AICSR.TFS = 0 to 0.
4 Set proper value to AICCR.M2S; AICCR.ENDSW; AICCR.ASVTSU.
5 Set AICFR.ENB to 1; Open AIC.

When using DMA mode, configure AICFR.RFTH, AICCR.RDMS or AICFR.TFTH, AICCR.TDMS.
Configure TX-FIFO and interrupt means setting proper value to AICFR.TFTH, clear AICCR.ETFS to 0, and clear AICCR.ETUR to 0.
Configure RX-FIFO and interrupt means setting proper value to AICFR.RFTH, clear AICCR.ERFS to 0 and clear AICCR.EROR to 0.
When configure interrupt, software must handle all the interrupt. So all interrupt is recommended disabled as shown above.
CODEC shares the interrupt with AIC module.

The register or register bit of CODEC will use the same form as the Mapped registers, but software should use the method in the section "Mapped Register Descriptions" to access this registers.

More details are listed in the CODEC guide.

10.3.3 Power saving

There are many power modes in CODEC. In every working mode, it should close stages (parts) of CODEC for saving power.

The power diagram is shown in “CODEC Power Diagram"; please refer to “CODEC Operating modes”.

10.3.4 Pop noise and the reduction of it

Please refre to “Ramping system note” and “Anti-pop operation sequences” for details.

10.3.4.1 Reference open step

1 Init play.
  Step 0: Open DMA and two AIC modules Clocks in CPM.CLKGR.
  Step 1: Configure AIC as slave and using inter CODEC mode.
    AICFR.ICDC = 1; Use internal CODEC.
    AICFR.AUSEL = 1; Use I2S mode.
    AICFR.BCKD = 0; CODEC input BIT_CLK to AIC.
    AICFR.SYNCD = 0; CODEC input SYNC to AIC.
    I2SCR.AMSL = 1; Use I2S operation mode.
    I2SCR.ESCLK = 1; Open SYS_CLK to internal CODEC.
  Step 2: Configure DMA as slave mode using internal CODEC.
2 Open.
  Step 0: Enable DMA Channel Clock.
  Step 1: Configure AIC sample size and sample rate. Configure AIC Output FIFO Threshold.
  Step 2: Configure DMA.
  Step 3: Configure CODEC.
3 Write.
  Step 0: Enable DMA Channel Clock.
  Step 1: Configure AIC.
  Step 2: Configure DMA.
  Step 3: Configure CODEC.
4 Read.
  Step 0: Enable DMA Channel Clock.
  Step 1: Configure AIC.
  Step 2: Configure DMA.
  Step 3: Configure CODEC.
5 Close.
6 End.

NOTES:

1 SB_DAC Control the internal OBIT_CLK from CODEC to AIC, First turn it on when write data

(replay).

2 SB_ADC Control the internal IBIT_CLK from CODEC to AIC, First turns it on when read data (record).

10.4 Timing parameters

Parameter	Condition	Min.	Typ.	Max.	Unit
Tsbyu	Cext = 10uF/100nF +/-20%		250	500	ms
Tshd_adc	Cext = 10uF/100nF +/-20%		200		ms
Tshd_dac	Cext = 10uF/100nF +/-20%		400	900	ms
Tr, Tf (all inputs)	All modes			5	ns
Tr, Tf (all outputs)	All modes			5	ns

NOTES:

1 Tsbyu is the reference wake-up time after complete power down.

2 Tshd_adc is the ADC wake-up time after sleep mode.

3 Tshd_dac is DAC wake-up time after sleep mode.

10.5 AC & DC parameters

Voltages:

AVSHP and AVSCDC are connected to analog ground.

AVDHP = 2.5V (typ).

AVDCDC= 2.5V (typ).

power consumption

mode	power consumption	Unit
Sleep mode	900	uW
Playback stereo audio DAC only (capacitor coupled load configuration)	5.5	mW
Record line input only (audio ADC)	5.5	mW
Record mic stereo input only (audio ADC)	5.5	mW
Record mic mono input only (audio ADC)	3.8	mW
Bypass path (capacitor coupled load configuration)	3.05	mW

Current value is at AVDCDC = AVDHP = 2.5 V.

Chip pin Name	MAX Current across I/O @ AVDCDC = AVDHP = 2.5 V
AVDCDC	< 20 mA in normal working mode
AVSCDC	< 20 mA in normal working mode
AVDHP	< 160 mA in normal working mode
	< 1400 mA in case of short circuit
AVSHP	< 160 mA in normal working mode
	< 1400 mA in case of short circuit
VCAP	< 2 mA in normal working mode
MICP1, MICN1	< 2 mA in normal working mode
MICP2, MICN2	< 2 mA in normal working mode
MICBIAS	< 5 mA in normal working mode
AOHPL	< 80 mA in normal working mode
	< 1200 mA in case of short circuit
AOHPR	< 80 mA in normal working mode
	< 1200 mA in case of short circuit
AOHPM	< 80 mA in normal working mode
	< 1200 mA in case of short circuit
AOHPMS	< 1 mA in normal working mode
AIL, AIR	< 1 mA in normal working mode
AOLOP,AOLON	< 1 mA in normal working mode
HPSENSE	< 1 mA in normal working mode

The current in case of short circuit is the max value. This current is only sink or drawn until the short circuit detection system acts.

Please refer to Chip Datasheet for more details.

10.6 CODEC internal Registers

Register Name	Function	Address	Reset value
SR	Status Register	000000 / 0x0 / 00	h00
AICR_DAC	DAC Audio Interface Control Register	000001 / 0x1 / 01	hC3
AICR_ADC	ADC Audio Interface Control Register	000010 / 0x2 / 02	hC3
CR_LO	differential line-out Control Register	000011 / 0x3 / 03	h90
CR_HP	HeadPhone Control Register	000100 / 0x4 / 04	h98
CR_DAC	DAC Control Register	000110 / 0x6 / 06	h90
CR_MIC	Microphone Control Register	000111 / 0x7 / 07	hB1
CR_LI	Control Register for line inputs	001000 / 0x8 / 08	h11
CR_ADC	ADC Control Register	001001 / 0x9 / 09	h10
CR_MIX	Control Register for digital mixer	001010 / 0xA / 10	h00
CR_VIC	Control Register for the codec	001011 / 0xB / 11	h03
CCR	Clock Control Register	001100 / 0xC / 12	h00
FCR_DAC	DAC Frequency Control Register	001101 / 0xD / 13	h00
FCR_ADC	ADC Frequency Control Register	001110 / 0xE /14	h40
ICR	Interrupt Control Register	001111 / 0xF / 15	h00
IMR	Interrupt Mask Register	010000 / 0x10 / 16	hFF
IFR	Interrupt Flag Register	010001 / 0x11 / 17	h00
GCR_HPL	left channel headphone Control Gain Register	010010 / 0x12 / 18	h06
GCR_HPR	right channel headphone Control Gain Register	010011 / 0x13 / 19	h06
GCR_LIBYL	left channel bypass line Control Gain Register	010100 / 0x14 / 20	h06
GCR_LIBYR	right channel bypass line Control Gain Register	010101 / 0x15 / 21	h06
GCR_DACL	Left channel DAC Gain Control Register	010110 / 0x16 / 22	h00
GCR_DACR	right channel DAC Gain Control Register	010111 / 0x17 / 23	h00
GCR_MIC1	Microphone 1 Gain Control Register	011000 / 0x18 / 24	h00
GCR_MIC2	Microphone 2 Gain Control Register	011001 / 0x19 / 25	h00
GCR_ADCL	Left ADC Gain Control Register	011010 / 0x1A / 26	h00
GCR_ADCR	Right ADC Gain Control Register	011011 / 0x1B / 27	h00
GCR_MIXADC	ADC Digital Mixer Control Register	011101 / 0x1D / 29	h00
GCR_MIXDAC	DAC Digital Mixer Control Register	011110 / 0x1E / 30	h00
AGC1	Automatic Gain Control 1	011111 / 0x1F / 31	h34
AGC2	Automatic Gain Control 2	100000 / 0x20 / 32	h07

AGC3	Automatic Gain Control 3	100001 / 0x21 / 33	h44
AGC4	Automatic Gain Control 4	100010 / 0x22 / 34	h1F
AGC5	Automatic Gain Control 5	100011 / 0x23 / 35	h00

10.6.1 CODEC internal registers

10.6.1.1 SR: Status Register

Register Name: SR Register Address: 0x0

bit7-R-0	bit6-R-0	bit5-R-0	bit4-R-0	Bit3-R-0	bit2-R-0	bit1-R-0	bit0-R-0
PON_ACK	IRQ_ACK	JACK	Reserved				

Bits	Field	Description
7	PON_ACK	Acknowledge status bit after power on. Read 0 = reset value Read 1 = codec is ready to operate
6	IRQ_ACK	Acknowledge status bit after IRQ sending. Read 0 = reset value Read 1 = codec has requested an interrupt (IRQ signal activated)
5	JACK	Output Jack plug detection status. Read 0 = no jack Read 1 = output jack present
4:0	Reserved	Writing has no effect, read as zero.

10.6.1.2 AICR_DAC: Audio Interface Control Register

Register Name: AICR_DAC Register Address: 0x1

bit7-RW-1	bit6-RW-1	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-1	bit0-RW-1
DAC_ADWL		Reserved				DAC_SERIAL	DAC_I2S

Bits	Field	Description
7:6	DAC_ADWL	Audio Data Word Length: for respectively DAC and ADC paths. Read / Write 00: 16-bit word length data 01: 18-bit word length data 10: 20-bit word length data 11: 24-bit word length data
5:2	Reserved	Writing has no effect, read as zero.
1	DAC_SERIAL	Selection of DAC digital serial audio interface. Read / Write 0: Parallel interface

		1: Serial interface
0	DAC_I2S	Working mode of DAC serial mode. (only relevant when serial interface is selected) Read/Write 0: DSP mode 1: I2S mode

NOTES:

1 DAC_SERIAL should be configured to 1.

2 DAC_I2S should be configured to 1.

10.6.1.3 AICR_ADC: Audio Interface Control Register

Register Name: AICR_ADC Register Address: 0x2

bit7-RW-1	bit6-RW-1	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-1	bit0-RW-1
ADC_ADWL		Reserved				ADC_SERIAL	ADC_I2S

Bits	Field	Description
7:6	ADC_ADWL	Audio Data Word Length: for respectively DAC and ADC paths. Read / Write 00: 16-bit word length data 01: 18-bit word length data 10: 20-bit word length data 11: 24-bit word length data
5:2	Reserved	Writing has no effect, read as zero.
1	ADC_SERIAL	Selection of the ADC digital serial audio interface. Read / Write 0: Parallel interface 1: Serial interface
0	ADC_I2S	Working mode of the ADC digital serial audio interface. (only relevant when serial interface is selected) Read/Write 0: DSP mode 1: I2S mode

NOTES:

1 ADC_SERIAL should be configured to 1.

2 ADC_I2S should be configured to 1.

10.6.1.4 CR_LO: differential line-out Control Register

Register Name: CR_LO Register Address: 0x3

bit7-RW-1	bit6-RW-0	Bit5-RW-0	bit4-RW-1	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
LO_MUTE	Reserved		SB_LO	Reserved		LO_SEL	

Bits	Field	Description
7	LO_MUTE	differential line output mute mode. Read/Write 0: mute inactive, Signal applied to line output 1: no signal on line output
6:5	Reserved	Writing has no effect, read as zero.
4	SB_LO	differential line out conditioning circuitry power-down mode. Read/Write 0: active 1: power-down
3:2	Reserved	Writing has no effect, read as zero.
1:0	LO_SEL	differential line-output Amplifier input selection. Read/Write If MICSTEREO = 0 00 : Microphone 1 enabled 01 : Microphone 2 enabled 10 : Bypass path enabled 11 : DAC output enabled If MICSTEREO = 1 00 : Microphone 1 & 2 enabled 01 : Microphone 1 & 2 enabled 10 : Bypass path enabled 11 : DAC output enabled

10.6.1.5 CR_HP: HeadPhone Control Register

Register Name: CR_HP Register Address: 0x4

bit7-RW-1	Bit6-RW-0	Bit5-RW-0	Bit4-RW-1	bit3-RW-1	bit2-RW-0	bit1-RW-0	bit0-RW-0
HP_MUTE	LOAD	Reserved	SB_HP	SB_HPCM	Reserved	HP_SEL	

Bits	Field	Description
7	HP_MUTE	HeadPhone output signal disabled. Read/Write 0: Signal applied to headphone outputs 1: no signal on headphone outputs, acts as a mute signal

6	LOAD	Selection of load impedance value for ramp generation. Read/Write 0: 16 Ohm / 220 uF 1: 10 kOhm / 1 uF
5	Reserved	Writing has no effect, read as zero.
4	SB_HP	headphone output stage power-down mode. Read/Write 0: headphone output stage is active 1: power-down
3	SB_HPCM	headphone output stage common mode buffer power-down mode. Read/Write 0: active (capacitor less headphone output configuration) 1: power-down (line output configuration)
2	Reserved	Writing has no effect, read as zero.
1:0	HP_SEL	Headphone Output Amplifier input selection. Read/Write If MICSTEREO = 0 00: Microphone 1 input to left and right channels 01: Microphone 2 input to left and right channels 10: Bypass path enabled 11: DAC output enabled If MICSTEREO = 1 00: Microphone 1 input to left channel and microphone 2 input to right channel 01: Microphone 2 input to left channel and microphone 1 input to right channel 10: Bypass path enabled 11: DAC output enabled

NOTE: The LOAD register is the load of AOHP.

10.6.1.6 CR_DAC: Control Register for DAC 3

Register Name: CR_DAC Register Address: 0x6

bit7-RW-1	bit6-RW-0	bit5-RW-0	Bit4-RW-1	bit3-RW-0	bit2-RW-0	Bit1-RW-0	bit0-RW-0
DAC_MUTE	DAC_MONO	DAC_LEFT_ ONLY	SB_DAC	DAC_ LRSWAP	Reserved		

Bits	Field	Description
7	DAC_MUTE	DAC soft mute mode. Read/Write 0: mute inactive, digital input signal transmitted to the DAC

		1: puts the DAC in soft mute mode
6	DAC_MONO	Digital stereo-to-mono conversion for DAC path. Read/Write 0: stereo 1: mono When DAC_MONO=1, the left and right channels are mixed in digital part: the result is emitted on both left and right channel of DAC output. It corresponds to the average of left and right channels when DAC_MONO=0.
5	DAC_LEFT_ONLY	Left data only are considered. Read/Write 0: DAC right channel active 1: DAC left data are used for left and right channel To avoid any audible pop, it is required to put the DAC in soft mute mode before modifying the DAC_LEFT_ONLY bit.
4	SB_DAC	DAC power-down mode. Read/Write 0: active 1: power-down
3	DAC_LRSWAP	swap between Left an d right channels. Read/Write 0: left data are sent to right channel, right data to left channel (swap) 1: left data are sent to left channel, right data to left channel (do not swap)
2:0	Reserved	Writing has no effect, read as zero.

NOTE: DAC_LRSWAP should be configured to 1.

10.6.1.7 CR_MIC: Control Register for microphone inputs

Register Name: CR_MIC Register Address: 0x7

bit7-RW-1	bit6-RW-0	bit5-RW-1	Bit4-RW-1	bit3-RW-0	bit2-RW-0	Bit1-RW-0	bit0-RW-1
MIC_STEREO	MICIDFF	SB_MIC2	SB_MIC1	Reserved		MICBIAS_V0	SB_MICBIAS

Bits	Field	Description
7	MIC_STEREO	Microphone input mode selection. Read/Write 0: Microphone mono inputs 1: Microphone stereo inputs This signal affects IN_SEL, HP_SEL, LO_SEL. Refer to its description.
6	MICIDFF	Microphone input mode selection.

		Read/Write 0:Microphone single-ended inputs 1: Microphone differential inputs
5	SB_MIC2	Analog MIC2 Input conditioning circuitry power-down mode. Read/Write 0: active 1: power-down
4	SB_MIC1	Analog MIC1 Input conditioning circuitry power-down mode. Read/Write 0: active 1: power-down
3:2	Reserved	Writing has no effect, read as zero.
1	MICBIAS_V0	B-port MICBIAS stage output voltage in operating mode. Read/Write 0: 5/6*VREF output voltage 1: 4/6*VREF output voltage
0	SB_MICBIAS	Microphone biasing buffer power-down. Read/Write 0: active 1: power-down

10.6.1.8 CR_LI: Control Register for line inputs

Register Name: CR_LI Register Address: 0x8

bit7-RW-0	bit6-RW-0	bit5-RW-0	Bit4-RW-1	bit3-RW-0	bit2-RW-0	Bit1-RW-0	bit0-RW-1
Reserved			SB_LIBY	Reserved			SB_LIN

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4	SB_LIBY	Linein used for bypass path power-down. Read/Write 0: active 1: power-down
3:1	Reserved	Writing has no effect, read as zero.
0	SB_LIN	Linein used to ADC power-down. Read/Write 0: active 1: power-down

10.6.1.9 CR_ADC: Control Register for ADC

Register Name: CR_ADC Register Address: 0x9

bit7-RW-0	bit6-RW-0	bit5-RW-0	Bit4-RW-1	bit3-RW-0	bit2-RW-0	Bit1-RW-0	bit0-RW-0
DMIC_SEL	ADC_MONO	ADC_LEFT_ ONLY	SB_ADC	ADC_LRSWAP	Reserved	IN_SEL	

Bits	Field	Description
7	DMIC_SEL	digital filter input selection. Read/Write 0: ADC 1: Digital microphone
6	ADC_MONO	Digital stereo-to-mono conversion for ADC path. Read/Write 0: stereo 1: mono When ADC_MONO=1, the left and right channels are mixed in digital part: the result is emitted on both left and right channel of ADC digital output. It corresponds to the average of left and right channels when ADC_MONO=0.
5	ADC_LEFT_ONLY	Deactivation of ADC right channel. Read/Write 0: ADC right channel active 1: ADC right channel inactive Note that when ADC right channel is deactivated, left channel is emitted on both left and right channel of ADC digital output.
4	SB_ADC	ADC power down mode. Read/Write 0: active 1: power-down
3	ADC_LRSWAP	swap between Left an d right channels. Read/Write 0: left data are sent to right channel, right data to left channel (swap) 1: left data are sent to left channel, right data to right channel (do not swap)
2	Reserved	Writing has no effect, read as zero.
1:0	IN_SEL	selection of the signal converted by the ADC. Read/Write If MICSTEREO = 0 00: Microphone 1 input to left and right channels (codec automatically considers that ADC_LEFT_ONLY equals '1' to optimize power consumption)

		01: Microphone 2 input to left and right channels (codec automatically considers that ADC_LEFT_ONLY equals '1' to optimize power consumption) 10 : Line input 11 : Reserved for further use Note that when digital microphone is selected, the ADC_LEFT_ONLY is not automatically modified. If MICSTEREO = 1 00: Microphone 1 input to left channel and microphone 2 input to right channel 01: Microphone 2 input to left channel and microphone 1 input to right channel 10 : Line input 11 : Reserved for test

NOTE: ADC_LRSWAP should be configured to 1.

10.6.1.10 CR_MIX: Control Register for digital mixer

Register Name: CR_MIX Register Address: 0xA

bit7-RW-1	bit6-RW-0	bit5-RW-0	Bit4-RW-0	bit3-RW-0	bit2-RW-0	Bit1-RW-0	bit0-RW-0
Reserved				MIX_REC		DAC_MIX	

Bits	Field	Description
7:4	Reserved	Writing has no effect, read as zero.
3:2	MIX_REC	Mixer mode on ADC Path. Read/Write 00: Record input only 01: Record input + DAC 10: Reserved for further use 11: Reserved for further use
1:0	DAC_MIX	Mixer mode on DAC Path. Read/Write 00: Playback DAC only 01: Playback DAC + ADC 10: Reserved for further use 11: Reserved for further use

10.6.1.11 CR_VIC: Control Register for the codec

Register Name: CR_VIC Register Address: 0xB

bit7-RW-0	bit6-RW-0	bit5-RW-0	Bit4-RW-0	bit3-RW-0	bit2-RW-0	Bit1-RW-1	bit0-RW-1
Reserved						SB_SLEEP	SB

Bits	Field	Description
7:2	Reserved	Writing has no effect, read as zero.
1	SB_SLEEP	sleep mode. Read/Write 0: normal mode (active) 1: sleep mode
0	SB	complete power-down mode. Read/Write 0: normal mode (active) 1: complete power-down

10.6.1.12 CCR: Control Clock Register

Register Name: CCR Register Address: 0xC

bit7-RW-0	bit6-RW-0	Bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
DMIC_CLKON	Reserved			CRYSTAL			

Bits	Field	Description
7	DMIC_CLKON	Digital microphone clock (DMIC_CLK) enable. Read/Write 0: clock off 1: clock on, clock frequency varies with DMIC_RATE and MCLK
6:4	Reserved	Writing has no effect, read as zero.
3:0	CRYSTAL	Selection of the SYS_CLK frequency. Read/Write The sampling frequency value is given in the CRYSTAL table.

MCLK	CRYSTAL	DMIC_CLK frequency
12 MHz	0000	3 MHz
13 MHz	0001	3.25 MHz

CRYSTAL	Master Clock Frequency
0000	12 MHz
0001	13 MHz
….	Reserved for further use
1111	Reserved for further use

NOTE: This register should be configured to 0x00 for setting the internal 12Mhz master clock SYS_CLK (default).

10.6.1.13 FCR_DAC: DAC Frequency Control Register

Register Name: FCR_DAC Register Address: 0xD

bit7-RW-0	bit6-RW-0	Bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved				DAC_FREQ			

Bits	Field	Description
7:4	Reserved	Writing has no effect, read as zero.
3:0	DAC_FREQ	Selection of the DAC sampling rate (Fs). Read/Write The sampling frequency value is given in the FREQ table.

NOTE: Please refer to section Sample frequency: FREQ.

10.6.1.14 FCR_ADC: ADC Frequency Control Register

Register Name: FCR_ADC Register Address: 0xE

bit7-RW-0	bit6-RW-0	Bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved	ADC_HPF	Reserved		ADC_FREQ			

Bits	Field	Description
7	Reserved	Writing has no effect, read as zero.
6	ADC_HPF	ADC High Pass Filter enable. Read/Write 0: inactive 1: enables the ADC High Pass Filter
5:4	Reserved	Writing has no effect, read as zero.
3:0	ADC_FREQ	Selection of the ADC sampling rate (Fs). Read/Write The sampling frequency value is given in the FREQ table.

NOTE: Please refer to section Sample frequency: FREQ.

10.6.1.15 ICR: Interrupt Control Register

Register Name: ICR Register Address: 0xF

bit7-RW-0	bit6-RW-0	Bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
INT_FORM		Reserved					

Bits	Field	Description
7:6	INT_FORM	Waveform and polarity of the IRQ signal. Read/Write 00: The generated IRQ is a high level 01: The generated IRQ is a low level 10: The generated IRQ is a high level pulse with an 8 SYS_CLK cycles duration 11: The generated IRQ is a low level pulse with an 8 SYS_CLK cycles duration
5:0	Reserved	Writing has no effect, read as zero.

NOTE: Please refer to section Sample frequency: FREQ.

10.6.1.16 IMR: Interrupt Mask Register

Register Name: IMR Register Address: 0x10

bit7-RW-1	bit6-RW-1	Bit5-RW-1	bit4-RW-1	bit3-RW-1	bit2-RW-1	bit1-RW-1	bit0-RW-1
Reserved	SCLR_MASK	JACK_MASK	SCMC_MASK	RUP_MASK	RDO_MASK	GUP_MASK	GDO_MASK

Bits	Field	Description
7	Reserved	Writing has no effect, read as zero.
6	SCLR_MASK	Mask for the SCLR flag. Read/Write 0: interrupt enabled 1: interrupt masked (no IRQ generation)
5	JACK_MASK	Mask for the JACK_EVENT flag. 0: interrupt enabled 1: interrupt masked (no IRQ generation)
4	SCMC_MASK	Mask for the SCMC flag. 0: interrupt enabled 1: interrupt masked (no IRQ generation)
3	RUP_MASK	Mask for the RUP flag. 0: interrupt enabled 1: interrupt masked (no IRQ generation)
2	RDO_MASK	Mask for the RDO flag.

		0: interrupt enabled 1: interrupt masked (no IRQ generation)
1	GUP_MASK	Mask for the GUP flag. 0: interrupt enabled 1: interrupt masked (no IRQ generation)
0	GDO_MASK	Mask for the GDO flag. 0: interrupt enabled 1: interrupt masked (no IRQ generation)

NOTES:

1 When an interrupt is masked, the event do not generates any change on the IRQ signal, but the corresponding flag value is set to '1' in the IFR register.

2 When the IRQ signal is active on level, the IRQ signal is set to the inactive level while the bits IFR & (!IMR) equals '0'.

3 When the IRQ signal is a pulse, the IRQ signal is set to the inactive state until a new non-masked event occurs in IFR or until a masked event is unmasked.

4 SYS_CLK must not be stopped in order to propagate IRQ signal.

10.6.1.17 IFR: Interrupt Flag Register

Register Name: IFR Register Address: 0x11

bit7-RW-0	Bit6-R-0	bit5-RW-0	bit4-R-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved	SCLR	JACK_EVENT	SCMC	RUP	RDO	GUP	GDO

Bits	Field	Description
7	Reserved	Writing has no effect, read as zero.
6	SCLR	Left or Right Output short circuit detection status. Read 0 : no event 1 : event detected (due to JACK flag change to '0' or '1') Write 1 to Reset of the flag.
5	JACK_EVENT	Event on output Jack plug detection status. Read 0: no event 1: event detected (due to JACK flag change to '0' or '1'). Write 1 to Reset of the flag.
4	SCMC	Common mode buffer output short circuit detection status. Read 0: inactive 1: indicates that a short circuit has been detected by the output stage Write 1 to Update of the flag.

3	RUP	End of output stage ramp up flag. Read 1: the ramp-up sequence is completed (output stage is active). Write 1 to Reset of the flag.
2	RDO	End of output stage ramp down flag. Read 1: the ramp-down sequence is completed (output stage in stand-by mode) Write 1 to Reset of the flag.
1	GUP	End of mute gain up sequence flag. Read 1: the mute sequence is completed; the DAC input signal is transmitted to the DAC path Write 1 to Reset of the flag.
0	GDO	End of mute gain down sequence flag. Read 1: the mute sequence is completed, a 0 DC signal is transmitted to the DAC path Write 1 to Reset of the flag.

NOTES:

1 The flags RUP, RDO, GUP and GDO can be reset after 4 cycles of SYS_CLK.

2 Interpretation of any unspecified point is absolutely up to the designer of analog part, so it is need to pay an attention to using this flags in section "Anti-pop operation sequences".

10.6.1.18 GCR_HPL: left channel headphone Control Gain Register

Register Name: GCR_HPL Register Address: 0x12

bit7-RW-0	bit6-RW-0	Bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-1	Bit1-RW-1	bit0-RW-0
LRGO	Reserved			GOL			

Bits	Field	Description
7	LRGO	HP amplifier gain coupling. Read/Write 0: Left and right channels gains are independent 1: Left and right channels gain track left channel gain
6:5	Reserved	Writing has no effect, read as zero.
4:0	GOL	Left channel HP amplifier gain programming value.

NOTE: Please refer to section "Programmable attenuation: GO" for more details.

10.6.1.19 GCR_HPR: right channel headphone Control Gain Register

Register Name: GCR_HPR Register Address: 0x13

bit7-RW-0	bit6-RW-0	Bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-1	Bit1-RW-1	bit0-RW-0
Reserved			GOR				

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4:0	GOR	Right channel HP amplifier gain programming value.

NOTE: Please refer to section "Programmable attenuation: GO" for more details.

10.6.1.20 GCR_LIBYL: left channel bypass line Control Gain Register

Register Name: GCR_LIBYL Register Address: 0x14

bit7-RW-0	bit6-RW-0	Bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-1	bit1-RW-1	bit0-RW-0
LRGI	Reserved		GIL				

Bits	Field	Description
7	LRGI	analog bypass gain coupling. Read/Write 0: Left and right channels gains are independent 1: Left and right channels gain track left channel gain
6:5	Reserved	Writing has no effect, read as zero.
4:0	GIL	Left channel Line in gain programming value.

NOTE: Please refer to section "Programmable Bypass path attenuation: GI" for more details.

10.6.1.21 GCR_LIBYR: right channel bypass line Control Gain Register

Register Name: GCR_LIBYR Register Address: 0x15

bit7-RW-0	Bit6-RW-0	Bit5-RW-0	Bit4-RW-0	bit3-RW-0	bit2-RW-1	bit1-RW-1	bit0-RW-0
Reserved			GIR				

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4:0	GIR	Left channel Line in gain programming value.

NOTE: Please refer to section "Programmable Bypass path attenuation: GI" for more details.

10.6.1.22 GCR_DACL: Left channel DAC Gain Control Register

Register Name: GCR_LIBYL Register Address: 0x16

bit7-RW-0	Bit6-RW-0	Bit5-RW-0	Bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
RLGOD	Reserved			GODR			

Bits	Field	Description
7	RLGOD	DAC digital gain coupling. Read/Write 0: Left and right channels gains are independent 1: Left and right channels gain track left channel gain
6:5	Reserved	Writing has no effect, read as zero.
4:0	GODL	Left channel DAC digital gain programming value.

NOTE: Please refer to section "Programmable digital attenuation: GOD" for more details.

10.6.1.23 GCR_DACR: right channel DAC Gain Control Register

Register Name: GCR_DACR Register Address: 0x17

bit7-RW-0	Bit6-RW-0	Bit5-RW-0	Bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved				GODR			

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4:0	GODR	Right channel DAC digital gain programming value.

NOTE: Please refer to section "Programmable digital attenuation: GOD" for more details.

10.6.1.24 GCR_MIC1: Microphone 1 Gain Control Register

Register Name: GCR_MIC1 Register Address: 0x18

bit7-RW-0	Bit6-RW-0	Bit5-RW-0	Bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved					GIM1		

Bits	Field	Description
7:3	Reserved	Writing has no effect, read as zero.
2:0	GIM1	Microphone 1 boost stage gain programming value.

NOTE: Please refer to section "Programmable boost gain: GIM".

10.6.1.25 GCR_MIC2: Microphone 2 Gain Control Register

Register Name: GCR_MIC2 Register Address: 0x19

bit7-RW-0	Bit6-RW-0	Bit5-RW-0	Bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved					GIM2		

Bits	Field	Description
7:3	Reserved	Writing has no effect, read as zero.
2:0	GIM2	Microphone 2 boost stage gain programming value.

NOTE: Please refer to section "Programmable boost gain: GIM".

10.6.1.26 GCR_ADCL: Left ADC Gain Control Register

Register Name: GCR_ADCL Register Address: 0x1A

bit7-RW-0	Bit6-RW-0	Bit5-RW-0	Bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
LRGID	Reserved	GIDL					

Bits	Field	Description
7	LRGID	ADC digital gain coupling. Read/Write 0: Left and right channels gains are independent 1: Left and right channels gain track left channel gain
6	Reserved	Writing has no effect, read as zero.
5:0	GIDL	Left channel ADC digital gain programming value.

NOTE: Please refer to the section "Programmable input attenuation amplifier: GID".

10.6.1.27 GCR_ADCR: Right ADC Gain Control Register

Register Name: GCR_ADCR Register Address: 0x1B

bit7-RW-0	Bit6-RW-0	Bit5-RW-0	Bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved		GIDR					

Bits	Field	Description
7:6	Reserved	Writing has no effect, read as zero.
5:0	GIDR	Right channel ADC digital gain programming value.

NOTE: Please refer to the section "Programmable input attenuation amplifier: GID".

10.6.1.28 GCR_MIXADC: ADC Digital Mixer Control Register

Register Name: GCR_MIXADC Register Address: 0x1D

bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved				GIMIX			

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4:0	GIMIX	Mixer gain for input path. Read/Write 00000 : 0dB 00001 : -1dB … by step of 1dB 11111 : -31dB

10.6.1.29 GCR_MIXDAC: DAC Digital Mixer Control Register

Register Name: GCR_MIXDAC Register Address: 0x1E

bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved				GOMIX			

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4:0	GOMIX	Mixer gain for DAC path. Read/Write 00000 : 0dB 00001 : -1dB … by step of 1dB 11111 : -31dB

10.6.1.30 AGC1: Automatic Gain Control Register 1

Register Name: AGC1 Register Address: 0x1F

bit7-RW-0	bit6-RW-0	bit5-RW-1	bit4-RW-0	bit3-RW-0	bit2-RW-1	bit1-RW-0	bit0-RW-0
AGC_EN	AGC_STEREO	TARGET				Reserved	

Bits	Field	Description
7	AGC_EN	selection of the AGC system. Read/Write 0 : inactive

		1 : enables the automatic level control
6	AGC_STEREO	selection of the AGC system. Read/Write 0 : same gain applied to Left and Right channel 1 : different gains applied to Left and Right channel
5:2	TARGET	Target output level of the ADC. Read/Write: 0000 : -6dB 0001 : -7.5dB … by step of 1.5 dB 1111 : - 28.5dB
1:0	Reserved	Writing has no effect, read as zero.

NOTE: Please refer to section "AGC system guide" for more details.

10.6.1.31 AGC2: Automatic Gain Control Register 2

Register Name: AGC2 Register Address: 0x20

bit7-RW-0	bit6-RW-1	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-1	bit1-RW-1	bit0-RW-1
NG_EN	NG_THR			HOLD			

Bits	Field	Description
7	NG_EN	Selection of the Noise Gate system. Read/Write 0: inactive 1: enables the noise gate system
6:4	NG_THR	Noise Gate Threshold value. Input level (dB) < Noise Gate Level (dB). Read/Write 000: -72 dB 001: -66 dB … by step of 6dB 111: -30 dB
3:0	HOLD	Hold time before starting AGC adjustment to the TARGET value. Read/Write 0000: 0ms 0001: 2 ms 0010: 4 ms … Time Step x2 1111: 32.768s

NOTE: Please refer to section "AGC system guide" for more details.

10.6.1.32 AGC3: Automatic Gain Control Register 3

Register Name: AGC3 Register Address: 0x21

bit7-RW-0	bit6-RW-1	bit5-RW-0	bit4-RW-0	bit3-RW-0	bit2-RW-1	bit1-RW-0	bit0-RW-0
ATK				DCY			

Bits	Field	Description
7:4	ATK	Attack Time - Gain Ramp Down. Read/Write 0000: 32 ms 0001: 64 ms … by step of 32 ms 1111: 512 ms
3:0	DCY	Decay Time - Gain Ramp up. Read/Write 0000: 32 ms 0001: 64 ms … by step of 32 ms 1111: 512 ms

NOTES:

1 DCY and ATK registers values are delays between each step of gain.

2 Please refer to section "AGC system guide" for more details.

10.6.1.33 AGC4: Automatic Gain Control Register 4

Register Name: AGC4 Register Address: 0x22

Bit7-RW-0	bit6-RW-0	bit5-RW-0	bit4-RW-1	bit3-RW-1	bit2-RW-1	bit1-RW-1	bit0-RW-1
Reserved			AGC_MAX				

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4:0	AGC_MAX	Maximum Gain Value to apply to the ADC path.

NOTES:

1 Please refer below table for AGC_MAX setup.

AGC_MAX	Gain Value	AGC_MAX	Gain Value	AGC_MAX	Gain Value	AGC_MAX	Gain Value
00000	0	01000	12	10000	23	11000	32
00001	1.5	01001	13.5	10001	23	11001	33.5
00010	3	01010	15	10010	23	11010	35
00011	4.5	01011	16.5	10011	24.5	11011	36.5
00100	6	01100	18	10100	26	11100	38
00101	7.5	01101	19.5	10101	27.5	11101	39.5
00110	9	01110	21	10110	29	11110	41
00111	10.5	01111	22.5	10111	30.5	11111	42.5
AGC_MAX	Gain Value	AGC_MAX	Gain Value	AGC_MAX	Gain Value	AGC_MAX	Gain Value
00000	0	01000	12	10000	23	11000	32
00001	1.5	01001	13.5	10001	23	11001	33.5
00010	3	01010	15	10010	23	11010	35
00011	4.5	01011	16.5	10011	24.5	11011	36.5
00100	6	01100	18	10100	26	11100	38
00101	7.5	01101	19.5	10101	27.5	11101	39.5
00110	9	01110	21	10110	29	11110	41
00111	10.5	01111	22.5	10111	30.5	11111	42.5

2 Please refer to section "AGC system guide" for more details.

10.6.1.34 AGC5: Automatic Gain Control Register 5

Register Name: AGC5 Register Address: 0x23

bit7-RW-0	bit6-RW-0	bit5-RW-0	Bit4-RW-0	bit3-RW-0	bit2-RW-0	bit1-RW-0	bit0-RW-0
Reserved			AGC_MIN				

Bits	Field	Description
7:5	Reserved	Writing has no effect, read as zero.
4:0	AGC_MIN	Maximum Gain Value to apply to the ADC path.

NOTES:

1 Please refer to below table for AGC_MIN setup.

AGC_MIN	Gain Value	AGC_ MIN	Gain Value	AGC_ MIN	Gain Value	AGC_ MIN	Gain Value
00000	0	01000	12	10000	23	11000	32
00001	1.5	01001	13.5	10001	23	11001	33.5
00010	3	01010	15	10010	23	11010	35
00011	4.5	01011	16.5	10011	24.5	11011	36.5
00100	6	01100	18	10100	26	11100	38
00101	7.5	01101	19.5	10101	27.5	11101	39.5
00110	9	01110	21	10110	29	11110	41
00111	10.5	01111	22.5	10111	30.5	11111	42.5

2 Please refer to section ”AGC system guide” for more details.

10.7 Programmable gains

This section helps you to configure the programmable gain amplifier in the CODEC.
Internal signal VREFP is connected to AVDCDC Pin and internal signal VREFN is connected to AVSCDC Pin.

In this section, VREF equals to (VREFP – VREFN).

10.7.1 Programmable boost gain: GIM

The following table gives the relation between the gain and the input level for the microphone input amplifier when GI = 0000.

GIM	Gain value (dB)	Maximum input amplitude
000	0	0.85*VREF
001	4	0.536*VREF
010	8	0.338*VREF
011	12	0.213*VREF
100	16	0.134*VREF
101	20	0.085*VREF
110	20	0.085*VREF
111	20	0.085*VREF

NOTES:

1 Maximum analog input amplitude value is given in Vpp differential.
2 Maximum analog input amplitude is referenced as Full Scale (FS). After conversion, the corresponding digital code of the output value varies from 0x7FFF down to 0x8000 for a 16-bit word. When the analog input amplitude is greater than FS, the dynamic characteristics are not guaranteed.
3 When a change occurs on GIDi inputs, data are valid on the digital output after about 64 sample periods. If the HPF is activated, data are valid after about 64 sample periods but the offset cancellation is not still completed at this time due to its internal time constant.

10.7.2 Programmable input gain amplifier: GID

The digital gain of ADC path may be programmed through the registers bits GIDL and GIDR.
The value of the gain is programmable from 0 to 23dB with a pitch of 1dB.

The gain and input levels are obtained according to the following table:

GID	Decimal decoded	Gain (dB)	Maximum input amplitude (Vpp. Differential) (FS)
0 0 0 0 0 0	0	0	0.85*VREF

0 0 0 0 0 1	1	1	0.757*VREF
0 0 0 0 1 0	2	2	0.6021*VREF
…	…		
x y z t u v	i	i	0.85 / {10^(i/20)} * VREF
…	…		
0 1 0 1 1 1	23	23	0.06 * VREF
0 1 1 0 0 0	24	24	0.06 * VREF
…	…		
1 0 1 0 1 0	43	43	0.06 * VREF
1 1 1 1 1 1	63	43	0.06 * VREF

NOTE: The last column of the table gives the maximum analog input to be applied on the MICi inputs. The value is given in Vpp differential. These values refer to the external voltage reference VREF equals to (VREFP – VREFN). The voltage levels depend on the VREF voltage.

10.7.3 Programmable digital attenuation: GOD

The attenuation of DAC output amplifier may be programmed independently for the both channels through the registers bits GODL and GODR.

The value of the gain GODL/R is programmable from +0 to –31dB with 1 dB pitch. The gain and output levels are obtained according to the following table:

GOD					Decimal decoded value	Gain Value (dB)
0	0	0	0	0	0	0
0	0	0	0	1	1	-1
					…	
0	0	1	1	0	6	-6
					…	
1	1	1	1	0	30	-30
1	1	1	1	1	31	-31

10.7.4 Programmable attenuation: GO

The attenuation of Headphone output amplifier may be programmed independently for the both channels through the registers bits GOL and GOR.

The value of the gain GOL/R is programmable from +6 to –25dB with 1 dB pitch. The gain and output levels are obtained according to the following table:

GO					Decimal decoded value	Gain Value (dB)	Maximal PGAT input amplitude (Vpp)	Maximal PGAT output amplitude (Vpp)
0	0	0	0	0	0	+6	0.425*VREF	0.85*VREF
0	0	0	0	1	1	+5	0.478*VREF	0.85*VREF
					…		…	…
0	0	1	0	1	5	+1	0.757*VREF	0.85*VREF
0	0	1	1	0	6	0	0.85*VREF	0.85*VREF
0	0	1	1	1	7	-1	0.85*VREF	0.757*VREF
					…		…	…
1	1	1	1	0	30	-24	0.85*VREF	0.054*VREF
1	1	1	1	1	31	-25	0.85*VREF	0.048*VREF

NOTES:

1 When headphone driver is loaded by a 16 Ohm load, setting GOL/R = 0 is possible. However, set GOL/R to 9 at maximum to preserve dynamic performances. The output stage is sized to support a 70mA current and no more.

2 The last column of the table gives the analog output voltage delivered on the outputs and corresponding to a digital input at FS (Full Scale). The value is given in Vpp single-ended.

3 These values refer to the external voltage reference VREF equals to (VREFP – VREFN). The voltage levels depend on the VREF voltage.

10.7.5 Programmable Bypass path attenuation: GI

The analog input gain may be programmed through GIL/R.

The value of the gain is programmable from +6 to -25dB with a pitch of 1dB. The gain and input levels are obtained according to the following table:

GI					Decimal decoded value	Gain value (dB)	Maximum input amplitude (Vpp) (FS)
0	0	0	0	0	0	+6	0.425*VREF
0	0	0	0	1	1	+5	0.478*VREF
0	0	0	1	0	2	+4	0.536*VREF
x	y	z	t	u	i	6 - i	0.85/{10^((6 - i)/20)}*VREF
0	0	1	1	0	6	0	0.85*VREF
					…		0.85*VREF

1	1	1	1	1	31	-25	0.85*VREF

The last column of the table gives the maximum analog input to be applied on the line inputs. The value is given in Vpp. These values refer to the external voltage reference VREF equals to (VREFP – VREFN). The voltage levels depend on the VREF voltage.

10.7.6 Programmable digital mixer gain: GIMIX and GOMIX

The following table gives the relation between the gain and the input level for the microphone input amplifier when GI = 0000.

GIMIX or GOMIX	Gain value (dB)
00000	0
00001	-1
00010	-2
00011	-3
…	…
11101	-29
11110	-30
11111	-31

10.7.7 Gain refresh strategy

GI* and GO* gains are controlled through the control interface. To avoid sound artifacts, the gain increases or decreases each time the gain stage output crosses the zero value. Tcrossout time-out counter forces the gain to be updated if a zero crossing event doesn't occur. After each gain step, zero crossing events are ignored during at least Tcrossmin.

In case that gain coupling between both left and right channels is active (LRGi different of RLGi), gain stepping of each channel is independent from the other depending on zero crossing event occurrence.

The duration of Tcrossout and Tcrossmin are given below:

MCLK (MHz)	Tcrossout (ms)	Tcrossmin (ms)
12	21.8	0.171
13	20.2	0.158

10.8 Configuration of the headphone output stage

In cap-coupled connection, codec uses the LOAD register bit to control the ramping duration. Inappropriate setting will lead to a too long or too fast ramping and will create audio artifacts.

To prevent pop-up noise generation due to floating nodes when no load is plugged in the jack connector, it is required to add some resistor devices that act as pull down function (named Rhpl and Rhpr in section “Headphone connnection” and section “Required external components”).

Its value has to be determined as following:

Working Mode	Load resistor and bypass capacitor values	LOAD value	Rhpdo value
Driving Headphone	16 Ohm / 220uF	0	470 Ohm typ.
Driving Lineout	10k Ohm / 1uF	1	4.7k Ohm max.

10.9 Out-of-band noise filtering

An internal analog Low Pass Filter at the DAC output is designed to remove the out-of-band noise generated by the delta sigma modulation (Noise Shaper). The internal LPF reduces the amount of energy contained in the wide band part (> 24 kHz) of the output signal. The out-of-band noise, when not removed, can be damageable in some high quality applications.

This filter is always working and does not need configure.

10.10 Output short-circuit protection (headphone output)

Analog short-circuit protection in the output stage has been implemented to prevent excessive current from flowing through AOHPL, AOHPR output pins. This prevents the output stage from over-heating. The system detects the following cases:

- Abnormal headphone load.

10.10.1 Indication of the short circuit detection

When such an overload is detected on one of AOHPL, AOHPR output pins,

- An interrupt is sent on the IRQ pin and the SCMC flag in the IFR register is set to '1'.
- Internally to codec:

  Automatic power-down of the 2 output amplifiers (AOHPL, AOHPR signals) when a short-circuit is detected on AOHPL or AOHPR pin (SCMC flag set to '1').

10.10.2 Reset of short circuit detection

The following sequence has to be to apply:

1. Mask the interrupt by writing '1' in the Interrupt Control Register.
2. Handle the cause of short-circuit according to the events presented in following paragraphs (Capacitor-coupled headphone connection).
3. Update the short-circuit flag by writing '1' in the Interrupt Flag Register.
4. Check the reset of flag by reading the Interrupt Flag Register. The bit must be equal to '0'. If it remains at '1', that means that short-circuit is not resolved.
5. Enable the interrupt by writing '0' in the Interrupt Control Register.

10.10.3 Capacitor-coupled headphone connection

It is up to the application to put the output stage in power down mode (SB_HP = '1'), to put codec in sleep or complete power-down mode, to reset it.

The short-circuit will be solved by the following events:

- Removal of the inserted jack. (needs the use of HPSENSE pins)
- Reset of codec. (NRST signal)
- Putting the output stage in power-down mode. (SB_HP=1)
- Putting codec in sleep mode. (SB_SLEEP=1)
- Putting codec in complete power-down mode. (SB=1)

10.11 Sampling frequency: FREQ

The sampling frequency value is given in the FREQ table below.

FREQ	Sampling Rate (Fs)
0000	96kHz
0001	48kHz
0010	44.1kHz
0011	32kHz
0100	24kHz
0101	22.05kHz
0110	16kHz
0111	12kHz
1000	11.025kHz
1001	8kHz
1010	Reserved for further use
….	….
1111	Reserved for further use

10.12 Programmable data word length

The Data Word Length block (DWL) allows selecting the length of the input data and of the output data between 24-/20-/18-/16-bit thanks to AICR.DAC_ADWL and AICR.ADC_ADWL (respectively for the DAC and ADC paths) in accordance with the following table:

ADWL	Word length
0 0	16-bit word length data
0 1	18-bit word length data
1 0	20-bit word length data
1 1	24-bit word length data

The size of the buses is always 24 bits, but the input/output data only use the number of MSB programmed with ADWL. The LSB are considered as '0' in input and set to '0' in output.

The capability to use a data word length of 16 bits is kept for compatibility with standard applications.

10.13 Ramping system note

An internal mechanism is used to reduce output glitches when the headphone stage enters or leaves the power-down mode.

When the SB_HP is set to '1', the headphone output voltages (AOHPL, AOHPR) are slowly decreased in the same time from AVDHP/2 down to 0.

When the SB_HP is set to '0', the headphone output voltages (AOHPL, AOHPR) are slowly increased in the same time from 0 to AVDHP/2.

After power supplies ramp up, the CODEC start its internal initialization sequence and set SR.ACK_PON register bit once completed.
An interrupt request is sent when the ramp completes.

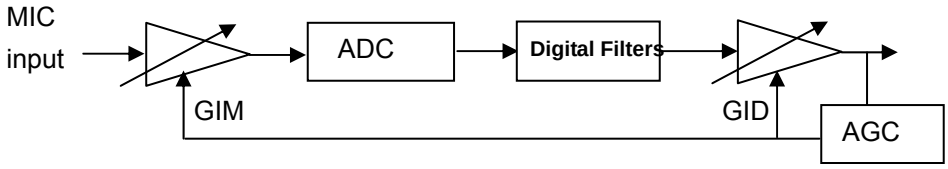
Do not change the level of SB_HP as long as the sequence due to the previous change is not complete or working not guaranteed.

In order to prevent audible glitch, it is required to power-down the output stage (SB_HP=1) before changing the output load with CR1.LOAD.

Please refer to "Anti-pop operation sequences" for details.

10.14 AGC system guide

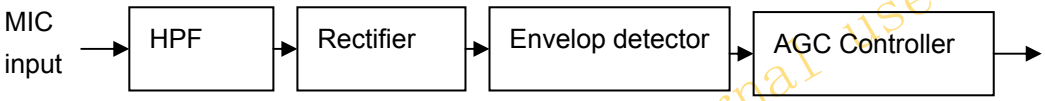
For the microphone input to ADC path, an Automatic Gain Control (AGC) system allows to optimize the signal swing at the input of the ADC.



The AGC circuit compares the output of the ADC to a level and increases or decreases the gain of the microphone preamplifier and the digital gain to compensate. The full dynamic range of the ADC can be used automatically if the audio from the microphone is to be output digitally through the ADC.

The AGC_EN register bit enables the AGC system, in this case INSEL must be equal to “00” or “01”.

AGC Block Diagram:



The AGC system is used at the MIC input.
The HPF filter characteristics: Cut Frequency =300 Hz.
In the in AGC mode, the system of gain control will directly assign the values of the gains GIDL, GIDR and GIM1 (or GIM2).

10.14.1 AGC operating mode

TARGET sets the desired ADC output range level. The AGC system adapts the gain stages (GID and GIM) in order to best reach this target. AGC_MAX and AGC_MIN fix the limits of the gain variation.
Please refer to “CODEC Operating modes” for the AGC System diagram in the “CODEC Power Diagram”.

In order that the AGC system should not alter the dynamic content of the signal (voice “tonic” for instance) by continuously adapting the gain to fit the target level, the time between two consecutives gain adjustments is modifiable by the HOLD register value.

After this delay:

- If the output level is lower than TARGET, the gain is increased step by step in accordance to the DCY register value.
- If the output level is higher than TARGET, the gain is decreased step by step in accordance to the ATK register value.

The following figure illustrates the behavior of AGC system:

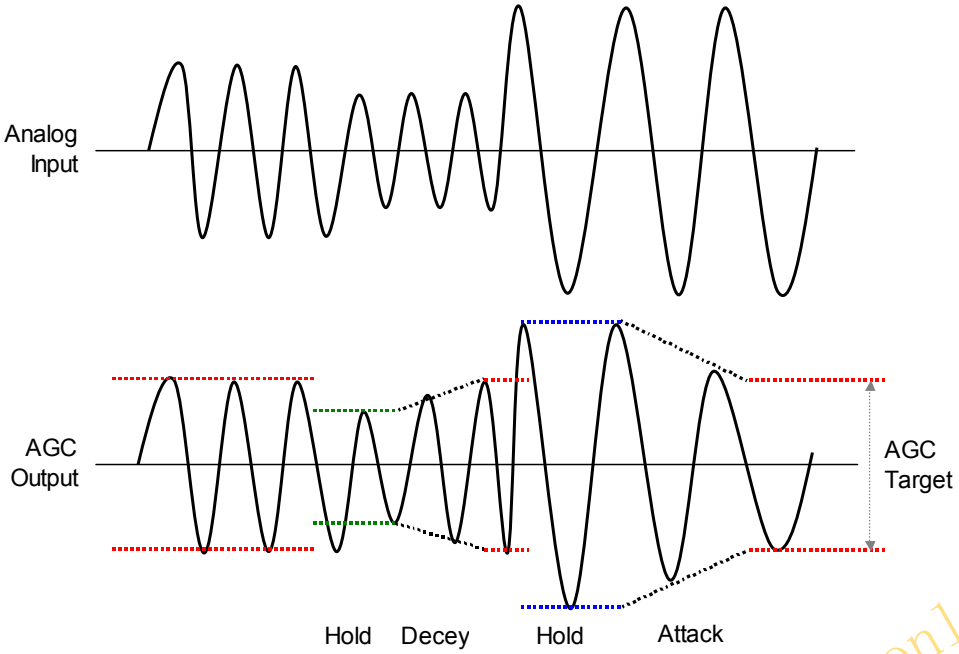


Figure 10-3 AGC adjusting waves

A noise-gating feature, enabled by the NG_EN register bit, prevents gain increases when no signal or small signal is present at the input. The noise gate threshold is set by the NG_THR register value.
The following graph shows a more detailed application.
The following graph summarizes the operations and shows more details.

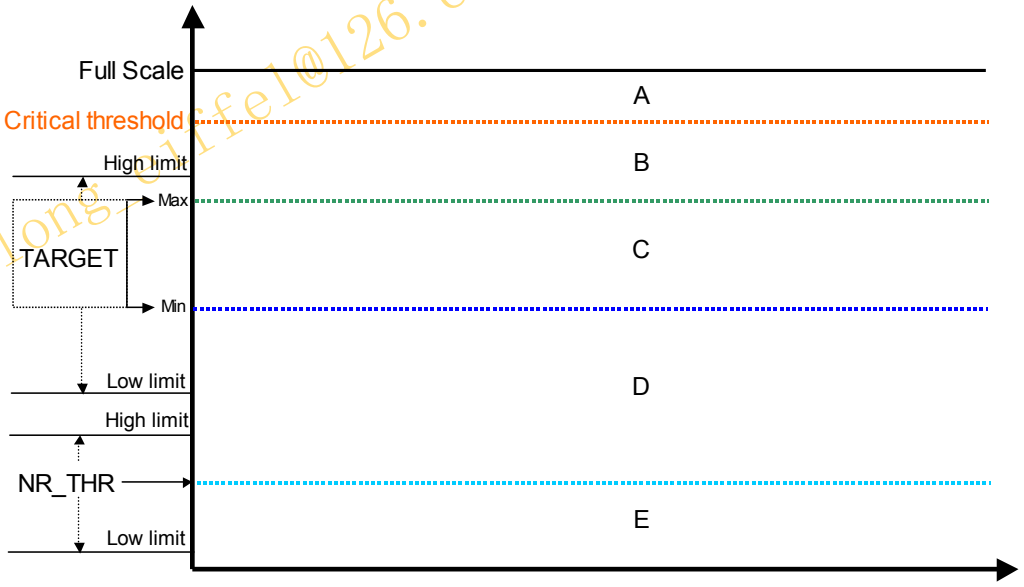


Figure 10-4 AGC adjust areas

The areas from A to E are deferent working area of AGC system, which is listing below:

A: If the signal level is in this critical area: the AGC system decreases quickly the gain at the input of the ADC until the signal goes under the critical threshold.

B: If the signal level remains in this area after the HOLD delay: the AGC system decreases the gain at the input of the ADC until the signal reaches the target area with a slope defined by AGC3.ATK register value.

C: If the signal level is in this area: the AGC system does not perform gain adjustment.

D: If the signal level remains in this area after the HOLD delay: the AGC system increases gain at the input of the ADC until the signal reach the target area with a slope defined by AGC3.DCY register value.

E: If the signal level is in this range: the AGC system considers the signal as noise and does not perform gain adjustment.

10.15 Digital Mixer description

CODEC includes a digital mixer which provides a loopback of the ADC output to the DAC and Headphone output and a loopback of the mixer output to the record path.

Two gains GIMIX and GOMIX control each input of the mixer to adapt the amplitude of the mixed signal. A zero-crossing detection is included on each gain stage to minimize the zipper noise.

A digital multiplexer allows choosing between the ADC signal and the mixer output signal on the record path.

Another digital multiplexer allows choosing between the DAC signal and the mixer output signal on the playback path.

Please refer to “CODEC Operating modes” for the digital mixer diagram in the “CODEC Power Diagram”.

10.16 Digital microphone interface

CODEC accepts bitstream from digital microphone and converts it into audio data at the sample rate (Fs) selected in FCR_ADC register. CODEC provides a clock (DMIC_CLK) and receives data on DMIC_IN at the same frequency. DMIC_CLK frequency depends on MCLK frequency selection in CCR register.

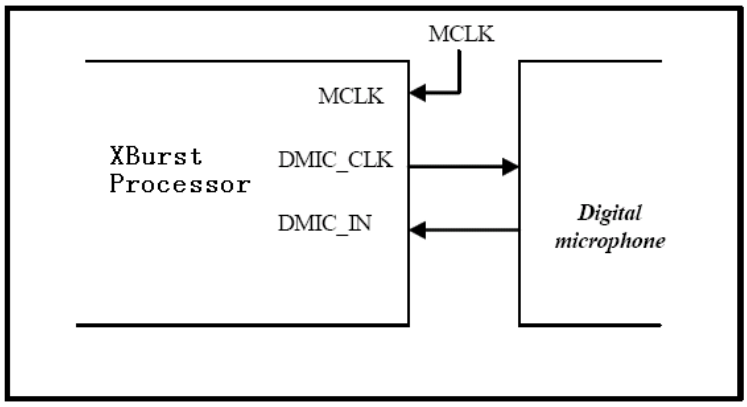


Figure 10-5 Digital microphone interface connection

After conversion, the corresponding digital code of the output value varies from 0x7FFF down to 0x8000 for a 16-bit word, coded in 2's complement.

CODEC can receive simultaneously data from two digital microphones.

10.16.1 Chronogram

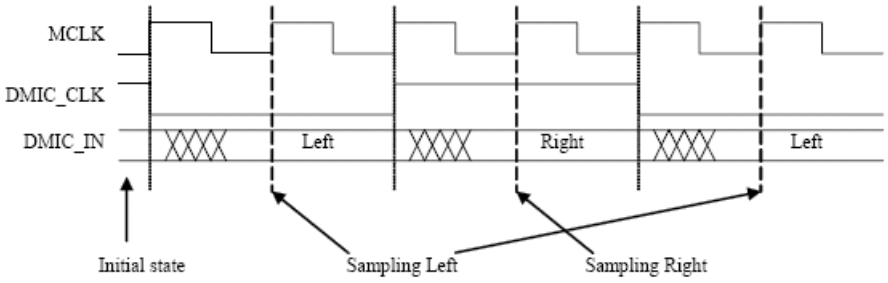
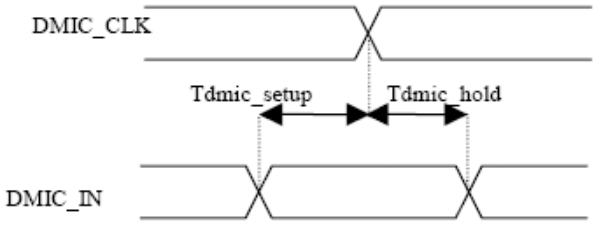


Figure 10-6 Digital microphone timing diagram at MCLK = 12 MHz

(DMIC_CLK = 3 MHz) and MCLK = 13 MHz (DMIC_CLK = 3.25 MHz)

10.16.2 Timings



Parameter	Symbol	Min	Typ	Max	Unit
DMIC_CLK frequency	F_{dmic_clk}	3	-	3.25	MHz
DMIC_CLK duty cycle	D_{dmic_clk}	0.4	0.5	0.6	-
DMIC_IN setup time	T_{dmic_setup}	T_{MCLK} + 10	-	-	ns
DMIC_IN hold time	T_{dmic_hold}	0	-	-	ns

10.16.3 Noise template (TBC)

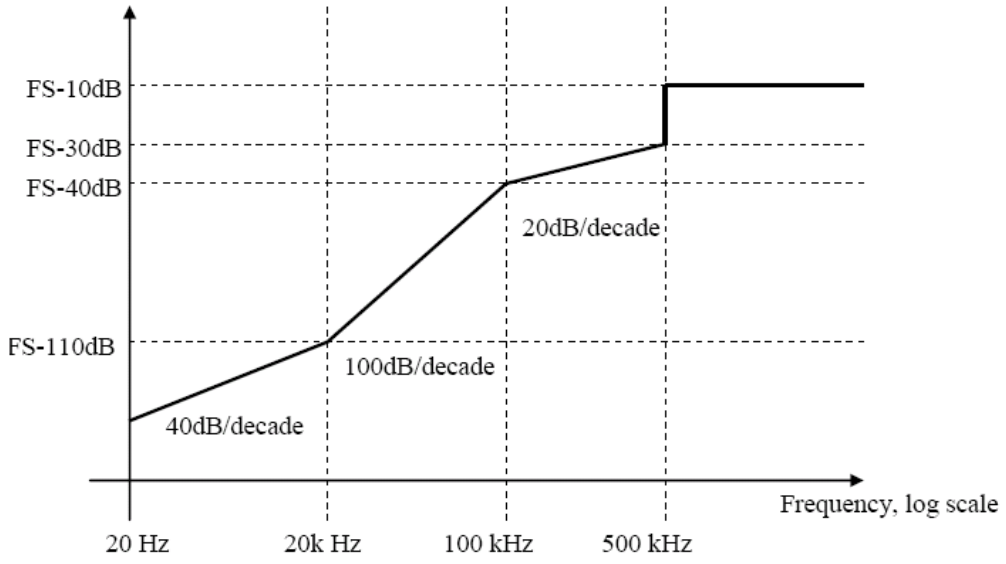


Figure 10-7 Digital microphone modulation noise reference spectrum

(with FFT resolution = 20 Hz and 7 terms Blackman-Harris windowing)

If the noise at the digital microphone output is higher than the reference in the [20 Hz – 20 kHz] bandwidth, the SNR will be limited by the digital microphone in-band noise.

If the noise at the digital microphone output is higher than the reference for frequencies beyond 20 kHz, the SNR will be limited by the aliasing of the digital microphone quantization noise.

10.17 CODEC Operating modes

Different operating modes are available:

- Power-up mode: During power on time, CODEC is in this mode.
- Reset mode: When NRST is low, CODEC is in this mode.
- Soft mute mode: When DAC_MUTE is 1, CODEC is in this mode.
- Complete Power-down mode: After RESET, CODEC is in this mode.
- SLEEP modes: When SB_SLEEP is 1, CODEC is in this mode.
- Normal mode: When CODEC is not in above mode, it is in this mode. This mode has three modes: RECORD mode, REPLAY mode, RECORD_REPLAY mode.

The power diagram is shown below.

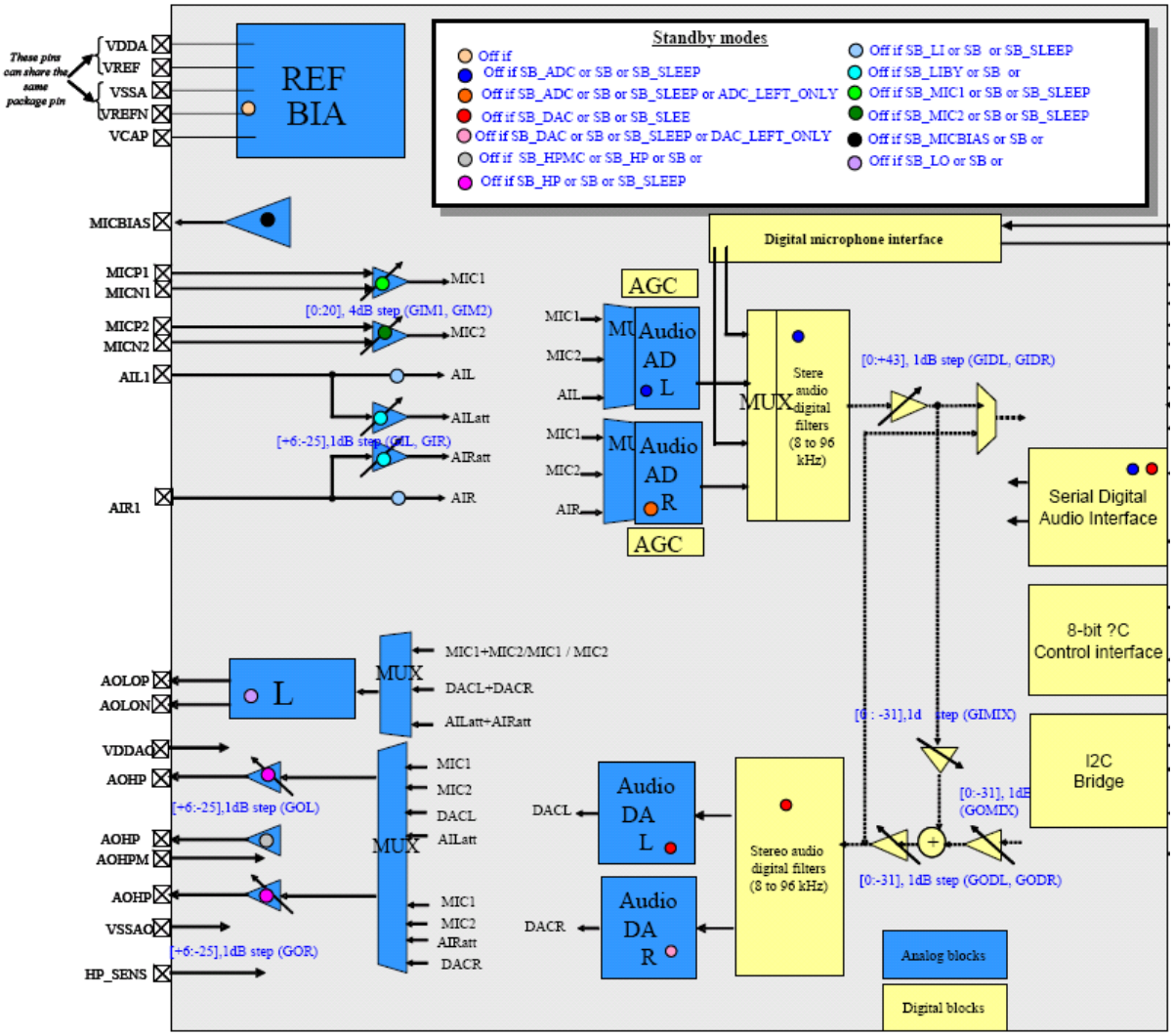


Figure 10-8 CODEC Power Diagram

There are many power parts of CODEC. Any part could be powered down independently.

10.17.1 Power-On mode and Power-Off mode

When the power supply ramps up, CODEC enters the power-on mode. During the reset, the CODEC is put in stand-by in order to reduce audible pops.

The CODEC doesn't handle the power supply ramp down on itself. The software has to turn the CODEC in complete stand-by mode before the power supply starts to ramp down.

10.17.2 RESET mode

The reset input signal is asynchronous; the reset minimum duration is one SYS_CLK cycle.
During the power-up mode and system reset, the CODEC goes into Reset mode.
After system reset the CODEC will exit Reset mode and go to STANDBY mode.

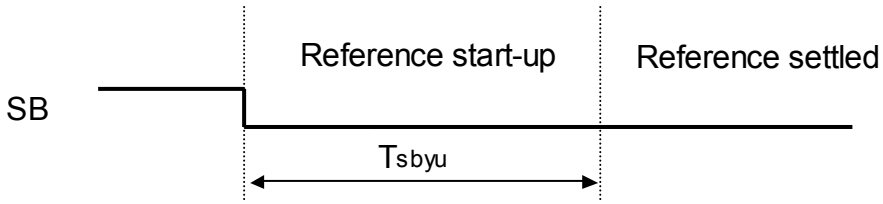
NOTES:

1 Except during the power-up mode, do NOT perform any reset in order to avoid audible pops.
2 Resetting the CODEC during normal operating mode will turn instantaneously the CODEC in STANDBY mode. This will lead to generate a large audible pop.

10.17.3 STANDBY mode

CODEC goes to STANDBY mode when the SB register bit equals 1, and all functions including ADC path, DAC path and analog references will stop and whole CODEC is shutdown for saving power. CODEC is complete down in this mode.

During the STANDBY mode, the power consumption is reduced to a minimum, so it is also called Complete Power-Down mode. When SB is set to '0', CODEC leaves the STANDBY mode. It is necessary to wait some time before the CODEC references settle. This time is called Tsbyu. When CODEC reference settled, it is in SLEEP mode.

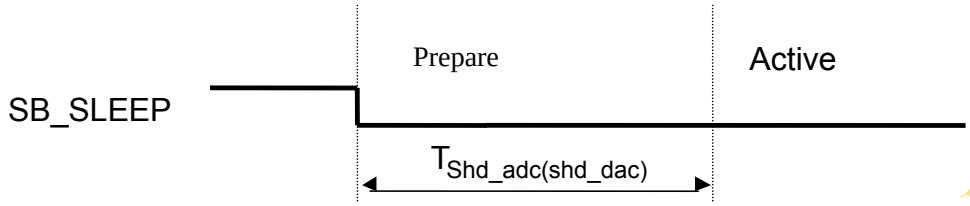


Please refer to the section "Timing parameters" for the Tsbyu Value.

10.17.4 SLEEP mode

When SB_SLEEP is set to 1, CODEC enters in sleep mode. The logical part and the analog functions, except the voltage and biasing references, enter in power-down mode. So, the power consumption is reduced without penalizing the start-up time.

When SB_SLEEP falls, CODEC leaves the corresponding STANDBY mode; it is necessary to wait some time before the CODEC reaches the normal mode. Depending on the selected mode, this time is either called Tshd_adc (SB_ADC=0) for the ADC path or Tshd_dac (SB_DAC=0) for the DAC path.



Please refer to the section "Timing parameters" for the Tshd_adc and Tshd_dac Value.

10.17.5 Soft Mute mode

Soft Mute mode is used in order to reduce audible parasites when before the DAC enters or after leaves the Normal mode. Set the DAC_MUTE register bit to 1, it will go to Soft Mute mode.

Set DAC_MUTE to 1 puts the DAC in Soft Mute mode. The CODEC decreases progressively the digital gain from 0dB to -∞. When the gain down sequence is completed, the signal of the DAC is equal to 0 whatever the value of the digital input data is. Then CODEC generates an interrupt and if ICR.GDO_MASK is 0, and set IFR.GDO register bit to 1.

During Soft Mute mode, the DAC is still converting but the output final voltages (AOLO, AOR) are equal to VREF/2, so the differential of the Headphone voltage is zero that cause no sound output.

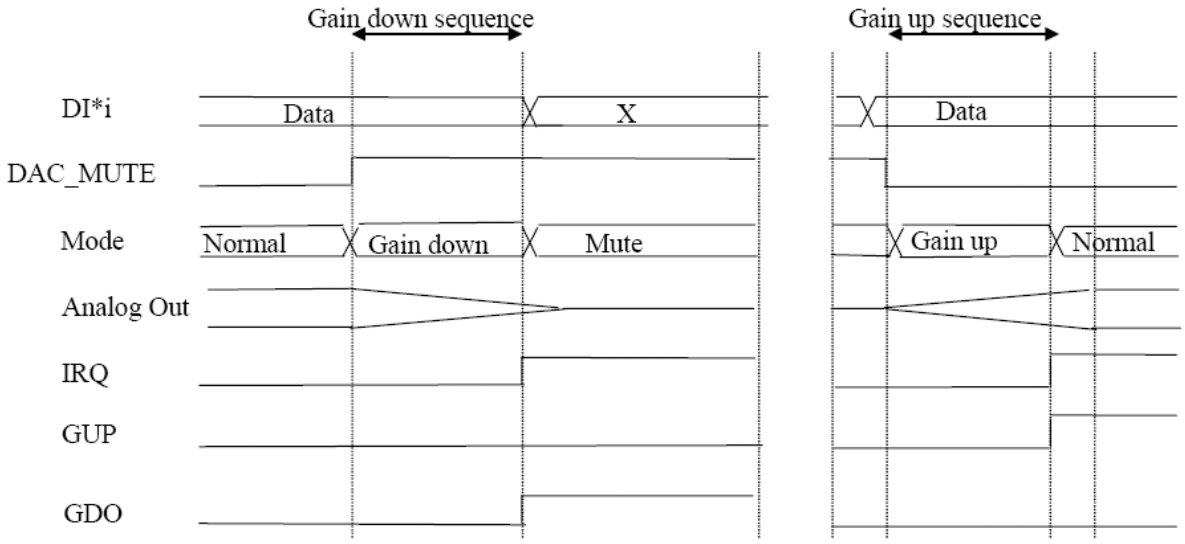


Figure 10-9 Gain up and gain down sequence

In the opposite, when DAC_MUTE is set to 0, the DAC leaves the Soft Mute mode by increasing progressively the digital gain from -∞ to 0dB. When the gain up sequence is completed, the DAC returns in Normal mode. The CODEC then generates an interrupt and if ICR.GDO_MASK is 0, and set IFR.GDO register bit to 1.
After exiting Soft Mute mode, the DAC output will flow the DAC input data, and there is sound in the Headphone.

The duration of gain down and gain up sequences are nearly independent of Fs and is about 23ms.

NOTES:

1 Do NOT change the value of DAC_MUTE while the effect of the previous change is not reached, or the working is not guaranteed.

2 Do NOT enter in stand-by mode while the gain sequence is not completed, or the working is not guaranteed.

10.17.6 Power-Down mode and ACTIVE mode

Twelve stand-by inputs allow putting independently the different parts of CODEC into Power-Down mode.

When all SB_*=1 except SB=0 and SB_SLEEP=0, the CODEC is in ACTIVE mode, it's ready for play sound or record sound. But still need follow the anti-pop start or stop sequence. Please refer to "Start up sequence" and "Shutdown sequence".

10.17.7 Working modes summary

Different working modes are sum-up in the following table (non exhaustive table):

Mode	SB	SB_SLEEP	SB_DAC	SB_HP	SB_LO	SB_ADC	SB_MICBIAS	SB_MIC1	SB_MIC2	SB_LIN	SB_LIBY	IN_SEL	HP_SEL	LO_SEL	MIC_STEREO	DAC_LEFT_ONLY	ADC_LEFT_ONLY	DAC__MUTE	HP_MUTE	LO_MUTE
Reset mode (complete power-down mode)	1	1	1	1	1	1	1	1	1	1	1	00	00	00	1	0	0	1	1	1
Complete power-down mode	1	x	x	x	x	x	x	x	x	x	x	xx	xx	xx	x	x	x	x	x	x
Sleep mode	0	1	x	x	x	x	x	x	x	x	x	xx	xx	xx	x	x	x	x	x	x
Record Mode																				
Mono MIC1 input	0	0	x	x	x	0	x	0	x	x	x	00	xx	xx	0	x	x	x	x	x
Mono MIC2 input	0	0	x	x	x	0	x	x	0	x	x	01	xx	xx	0	x	x	x	x	x
Record Mode, stereo MIC inputs,																				
MIC1 to left channel	0	0	x	x	x	0	x	0	x	x	x	00	xx	xx	1	x	0	x	x	x
MIC2 to left channel	0	0	x	x	x	0	x	x	0	x	x	01	xx	xx	1	x	0	x	x	x
Record Mode, Line input	0	0	x	x	x	0	x	x	x	0	x	10	xx	xx	x	x	0	x	x	x
Playback mode, DAC to HP	0	0	0	0	x	x	x	x	x	x	x	xx	11	xx	x	0	x	0	0	x
Playback mode, DAC to LO	0	0	0	x	0	x	x	x	x	x	x	xx	xx	11	x	x	x	x	x	0
Bypass mode, Line to HP	0	0	x	0	x	x	x	x	x	x	0	xx	10	xx	x	x	x	x	0	x
Bypass mode, Line to LO	0	0	0	x	0	x	x	x	x	x	x	xx	xx	10	x	x	x	x	x	0
Sidetone mode,																				
Mono MIC1 input to HP	0	0	x	0	x	x	x	0	x	x	x	xx	00	xx	0	x	x	x	0	x
Mono MIC2 input to HP	0	0	x	0	x	x	x	x	0	x	x	xx	01	xx	0	x	x	x	0	x
Sidetone mode, stereo MIC to HP,																				
MIC1 to left channel	0	0	x	0	x	x	x	0	x	x	x	xx	00	xx	1	x	x	x	0	x
MIC2 to left channel	0	0	x	0	x	x	x	x	x	x	x	xx	01	xx	1	x	x	x	0	x
Sidetone mode,																				
Mono MIC1 input to LO								0	x	x	x	xx	xx	00	0	x	x	x	x	0
Mono MIC2 input to LO	0	0	x	x	0	x	x	x	0	x	x	xx	xx	01	0	x	x	x	x	0
MIC1+MIC2 to LO	0	0	x	x	0	x	x	0	0	x	x	xx	xx	0x	1	x	x	x	x	0

10.18 SYS_CLK turn-off and turn-on

The main clock of CODEC is called SYS_CLK, which is generated in CPM module and called MCLK. During the SLEEP mode and the complete power-down mode, the main clock SYS_CLK may be stopped to reduce the power consumption to the leakage currents only. In other modes, the main clock SYS_CLK must not be stopped.

The main clock SYS_CLK must not be stopped until CODEC has reached the complete power-down mode and must be restarted before leaving the power-down mode.

After SYS_CLK restarts, it is required to wait 4 SYS_CLK cycles before reading or writing the registers.

When SYS_CLK is turned off (SB_SLEEP=1 or SB=1), writing on register values are not taken into account, register values are not up to date when read and interrupts not generated until SYS_CLK turns on.

10.19 Requirements on outputs and inputs selection and power-down modes

The following rules must be respected in order not to damage performances and to keep the functionality:

- If SB_MIC1 is set to 1, MICSTEREO must be equal to 0, IN_SEL, HP_SEL and LO_SEL must not be equal to ‘00’.
- If SB_MIC2 is set to 1, MICSTEREO must be equal to 0, IN_SEL, HP_SEL and LO_SEL must not be equal to ‘01’.
- If SB_LINE is set to 1, IN_SEL must not be equal to ‘10’.
- If SB_LIBY is set to 1, HP_SEL and LO_SEL must not be equal to ‘10’.
- If SB_DAC is set to 1, HP_SEL and LO_SEL must not be equal to ‘11’.

10.20 Anti-pop operation sequences

The main idea of this section is to describe the sequences to perform to minimize the audible pop to the minimum for the headphone output.

Due to the large number of stand-by combinations and to be the most flexible, the handling of the sequence from one working mode to another is left to the software. So for helping the software designer in this task, some specific sequences are automatically performed by CODEC and an interrupt mechanism (IRQ signal and associated registers) warns the application when these sequences end.

10.20.1 Initialization and configuration

To use the embedded CODEC with AIC, several AIC registers should be set up the below register of AIC before start the CODEC:

AICFR.ICDC = 1
AICFR.AUSEL = 1
AICFR.BCKD = 0
AICFR.SYNCD = 0
I2SCR.AMSL = 0
I2SCR.ESCLK = 1

10.20.2 Start up sequence (DAC)

This sequence is from Power-on mode to CODEC REPLAY mode.
The output sound is driving by DAC.
The intent of the following sequence is to prevent for large audible glitches due to the system start-up with the CODEC.

Before this sequence, setup the AIC properly.

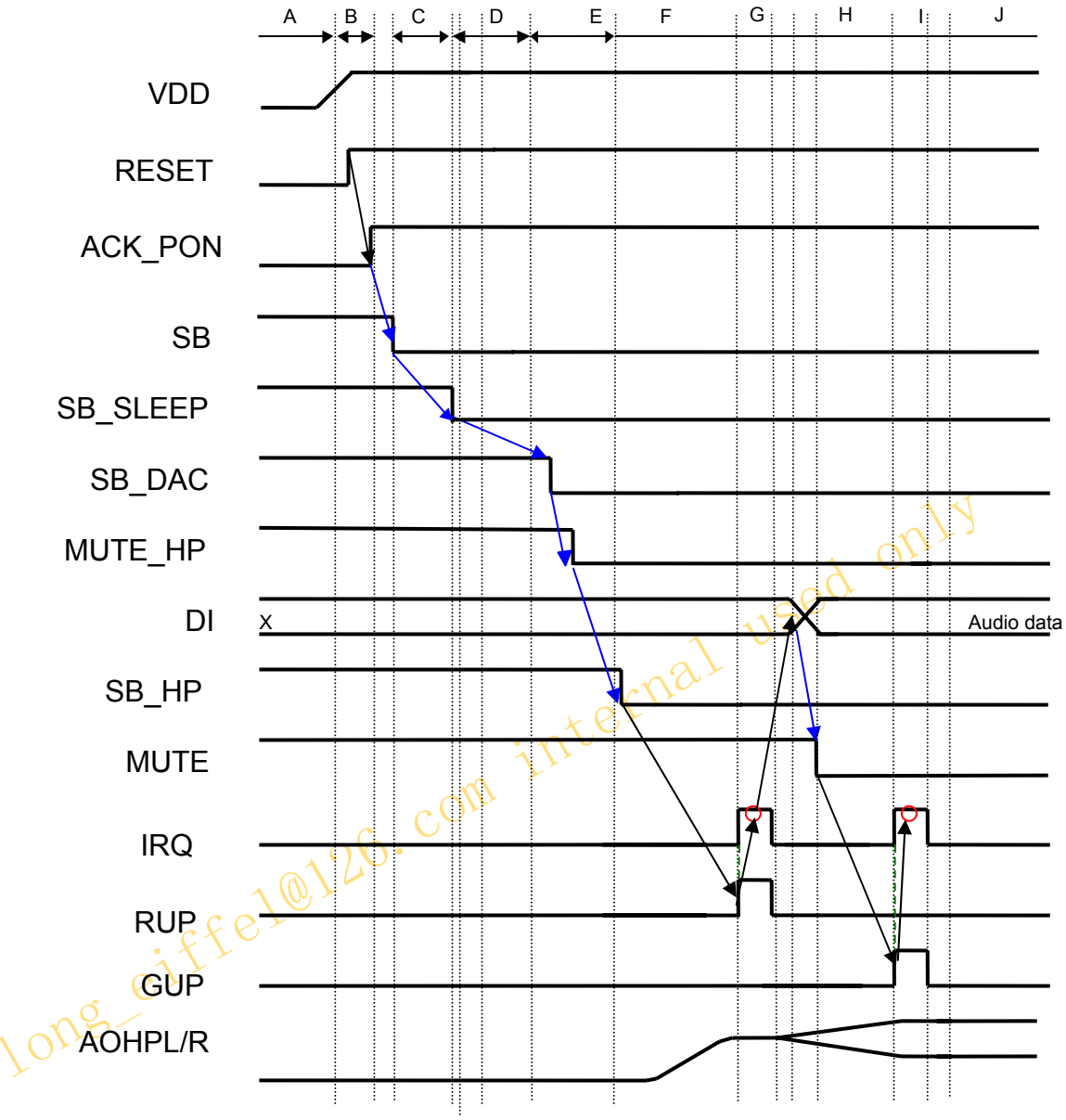


Figure 10-10 Start up sequence

NOTES:

1 The sequences in **blue** are manually handled by the software.

2 The sequences in **black** are automatically handled by the CODEC.

3 **Red** circles are interrupts automatically generated by the CODEC.

SEQUENCE:

A Initial state.
The power supply is off.

B Power supply ramp up.

The RESET of CODEC is '0' during system reset. The CODEC starts its internal initialization sequence and set ACK_PON register bit once completed.

C Starting of CODEC reference.

The software turns the CODEC on SLEEP mode by clearing SB register bit to 0. The duration equals Tsbyu. After waiting the Tsbyu duration (for example, on event generated by a timer at the application level), the CODEC is in SLEEP mode, the ADC and DAC path are ready to be turn to active mode.

D Go from SLEEP mode to active.

The application turns on the DAC by clearing SB_SLEEP register bits to 0.

E Turn on DAC.

Once after leaving SLEEP mode, the application turns on the DAC (SB_DAC=0) and after 0.5 ms switch the analog mute signal of the port to activate to 0 (MUTE_HP=0) .

F Ramp up cycle.

After waiting 1 ms, the application turns on the headphone output stage (SB_HP=0).

G Ramp up IRQ generation.

Once the ramp up cycle completes, the CODEC sets the RUP flag to 1 and generates an interrupt.

H IRQ handling and gain up cycle.

The application handles the interrupt, resets the RUP flag by writing 1 on it and releases the mute of the DAC (DAC_MUTE=0). In the same time, the application sends valid audio data to the CODEC DAC.

I Gain up IRQ generation.

Once the gain up cycle completes, the CODEC sets the GUP flag to 1 and generates an interrupt.

J IRQ handling and DAC active mode.

The application handles the interrupt and resets the GUP flag by writing 1 on. The CODEC DAC path is now fully activated.

10.20.3 Shutdown sequence (DAC)

This sequence is from CODEC REPLAY mode to STANDBY mode.
The output sound is driving by DAC.

The intent of the following sequence is to prevent for large audible glitches due to the system shutdown with the CODEC.

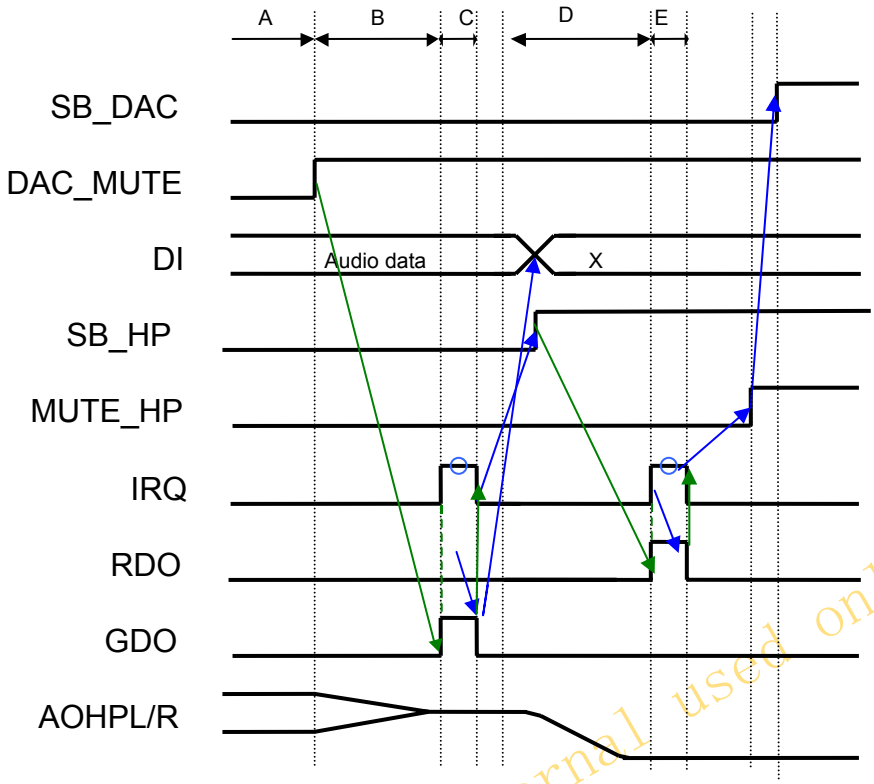


Figure 10-11 Shutdown sequence

NOTES:

1 The sequences in blue are handled by the software.

2 The sequences in black are automatically handled by the CODEC.

SEQUENCE:

A Initial state.

The power supply is on, CODEC DAC path is fully activated.

B Gain down cycle.

The application activates the mute of the DAC (DAC_MUTE=1). Once the gain down cycle completes, the CODEC sets the GDO flag to 1 and generates an interrupt.

C Gain down IRQ handling and ramp down cycle.

The application handles the interrupt and resets the GDO flag by writing 1 on it. The application can then stop sending audio data and turns off the headphone output stage (SB_HP=1).

D Ramp down IRQ generation.

Once the ramp down cycle completes, the CODEC sets the RDO flag to 1 and generates an interrupt.

E IRQ handling.

The application handles the interrupt and resets the RDO flag by writing 1 on it. Then, the application can activate the analog mute (MUTE_HP=1). Finally, the application turns off the

DAC path (SB_DAC=1) to be in sleep mode or turn off the CODEC (SB_SLEEP=1, SB=1).

10.20.4 Start up sequence (Line input)

This sequence is from Power-on mode to CODEC REPLAY mode.
The output sound is driving by Line input.
The intent of the following sequence is to prevent for large audible glitches due to the system start-up with the CODEC.

SEQUENCE:

A initial state.
DAC or Line in channel is already in use, valid analog audio signals are available at the input of the switch matrix.

B initializing output port.
The application first set the line in and headphone gain stages to their minimum value (gain automatically forced when the port is in power-down mode). This setting is taken into account in few clocks cycles. Set the MUTE_HP=0, Then the application turns on the headphone output stages (SB_HP = 0).

C Ramp up IRQ generation.
Once the ramp up cycle completes, the CODEC sets the RUP flag to 1 and generates an interrupt.

D Ramp up IRQ handling and line in stage gain up.
The application handles the interrupt and resets the RUP flag by writing 1 on it. The application then set the line in gain stage to the wished value.
The maximum duration of the gain ramping equals Trlinemax:

$$Trlinemax = N1 * Tcrossout$$

N1 is the number of line in gain steps.
Please Refer to section "Gain refresh strategy" for the value of Tcrossout.

E Headphone stage gain up.
The application set the headphone gain stage to the wished value. The maximum duration of the gain ramping equals Troutmax:

$$Troutmax = N2 * Tcrossout$$

N2 is the number of headphone gain steps.

F active mode.
The signal path is now fully activated.

10.20.5 Shutdown sequence (Line input)

This sequence is from CODEC REPLAY mode to STANDBY mode.
The output sound is driving by Line input.

The intent of the following sequence is to prevent for large audible glitches due to the system shutdown with the CODEC.

SEQUENCE:

A active mode.

The signal path is now fully activated.

B headphone stage gain down.

The application set the headphone gain stage to the minimum value. The maximum duration of the gain ramping equals Tdoutmax:

Tdoutmax = N3 * Tcrossout

N3 is the number of headphone gain steps.

Please Refer to section "Gain refresh strategy" for the value of Tcrossout.

C line in stage gain down.

The application set the line in gain stage to the minimum value. The maximum duration of the gain ramping equals Tdlinemax:

Tdlinemax = N4 * Tcrossout

N4 is the number of headphone gain steps.

D Ramp down cycle.

Then, the application can activate the analog mute (MUTE_HP=1) and turns off the headphone output stages (SB_HP=1).

E Ramp down IRQ generation.

Once the ramp up cycle completes, the CODEC sets the RDO flag to '1' and generates an interrupt.

F Ramp down IRQ handling.

The application handles the interrupt and resets the RDO flag by writing '1' on it. The signal path is now off.

10.21 Circuits design suggestions

This section lists a few PCB design suggestions with difference using mode.

10.21.1 Avoid quiet ground common currents

10.21.1.1 References pins

To work properly, CODEC requires few additional external components.

CODEC includes an internal voltage reference. To insure a correct common mode biasing of the internal components, an additional voltage VCAP is used. This requires connecting two decoupling capacitors (Cext) between the pin VCAP and AVSCDC. One 10uF low ESR (ceramic or tantalum) and one 100nF ceramic have to be used. The ceramic capacitor has to be kept as close as possible to IC package (closer than 0.2 inch).

10.21.1.2 Power supply pins

CODEC analog power supplies require external decoupling capacitors.

For each power supply pin, one 100nF ceramic capacitor has to be used. This ceramic capacitor has to be kept as close as possible to IC package (closer than 0.2 inch). One low ESR (ceramic or tantalum) capacitor has to be used to decouple the analog power supply provided to the CODEC. Its value depends on the power supply generator; its typical value is between 1uF and 10uF. Ideally use separate ground planes for analog and digital parts.

Connect all ground pins with thick traces to power plane in order to ensure lowest impedance connections.

AVSCDC must be connected to the PCB analog single point reference (star connection) ground (AGND).

10.21.2 Headphone connection (Capacitor-coupled)

Capacitor-coupled headphone and line connection

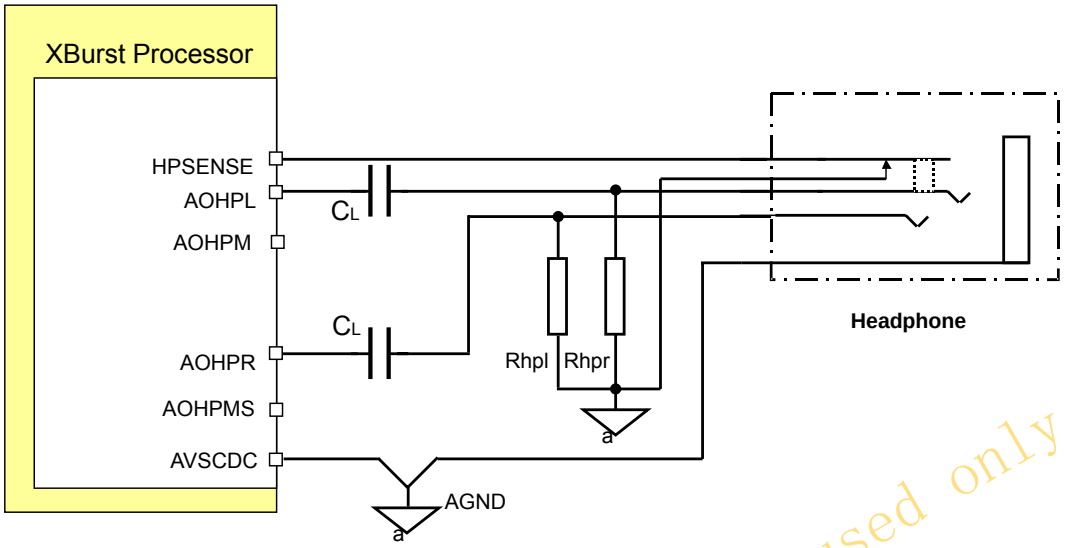


Figure 10-12 Capacitor-coupled connection

The AOHPL and AOHPR pins are connected to the headphone through an external bypass capacitor which is a DC blocking capacitors.

This capacitor is called C_L. When the headphone resistance R_L is 16 Ohm, The tantalum blocking capacitor C_L is 220 uF.

The DC value of the signal AOHPL or AOHPR equals to AVDCDC/2.

The ground of the headphone is connected to AGND, which is the PCB analog single point reference (star connection) ground.

Capacitor-less headphone connection

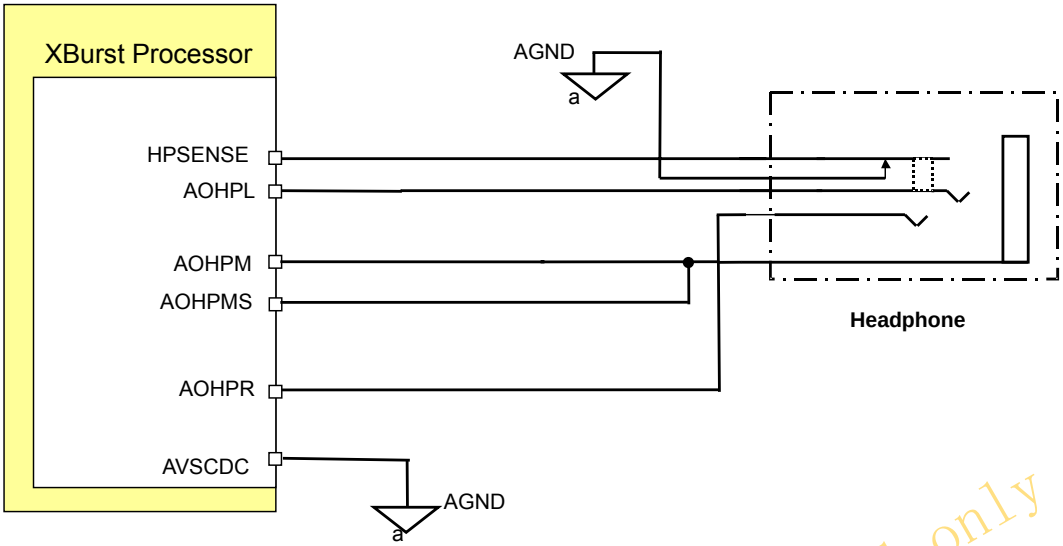


Figure 10-13 Capacitor-less connection

The signals AOHPL and AOHPR from chip are applied directly to the loads. The ground of the headphone is connected to AOHPM. The DC value of the signal AOHPi equals to VREF/2.
The measurement ground reference corresponds to the physical interconnection of AOHPM and AOHPMS.
The measurement is done between AOHPL/R and the measurement ground reference.

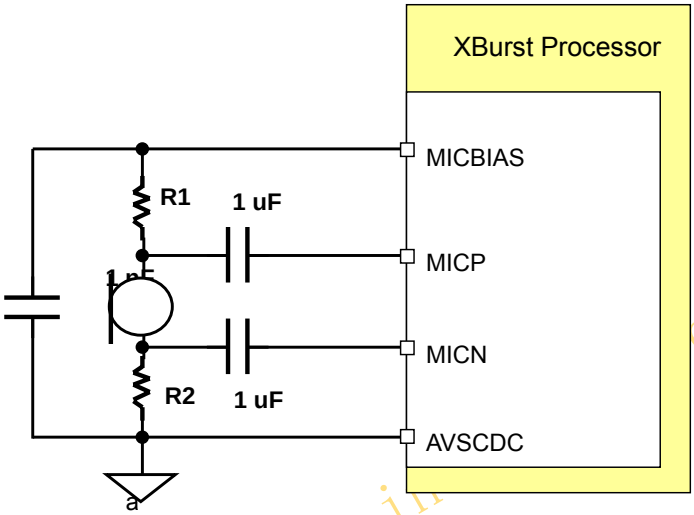
NOTE: If you want to use headphone as the antenna for FM , you had better choose this mode.

10.21.3 Microphone connection

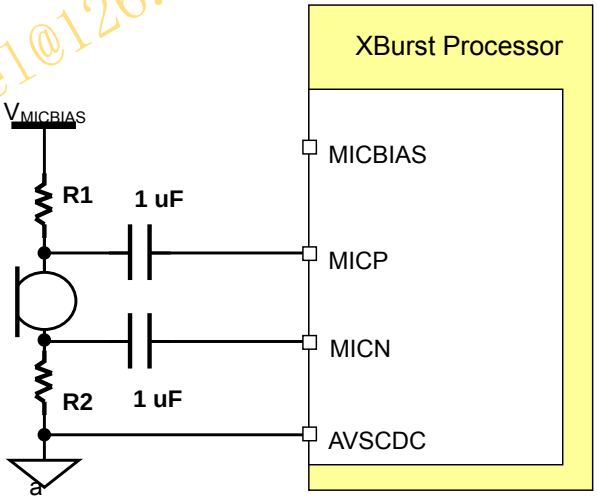
The optimal performance for the SNR is obtained in differential Microphone inputs with a FS input level corresponding to the following values: the peak-to-peak amplitude of the signal is 0.2125V, corresponding to 0.085*VREF Vpp.

We recommend customer to use differential MIC input for better performance.

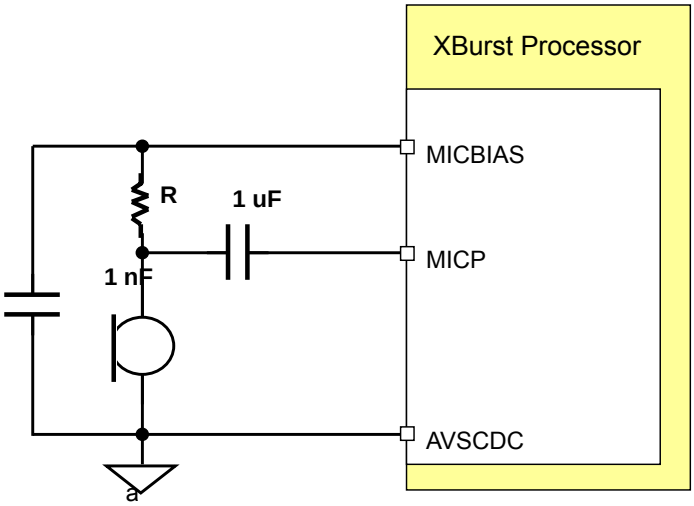
Application schematic with differential MIC input (using MICBIAS pin):



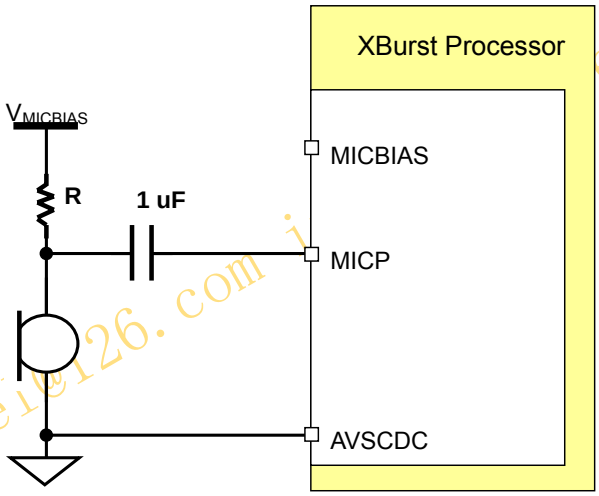
Application schematic with differential MIC input (Vmicbias generated on board):



Application schematic with single-ended MIC input (using MICBIAS pin):



Application schematic with single-ended MIC input (Vmicbias generated on board):



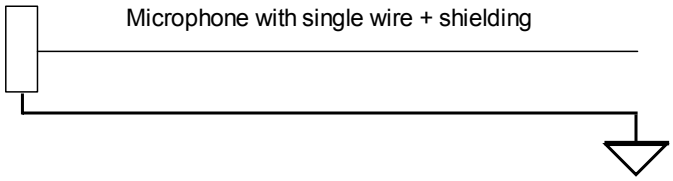
In single-ended connection, one external resistor (R) has to be used to bias the electret microphone. In differential connection, a pair of external resistor (R1, R2) has to be used to bias the electret microphone. The resistors value relation between them is R1 = R2 = R/2.

Specific value of resistor (R, commonly from 2.2k Ohm to 4.7k Ohm) and Vmicbias (if generated on board, usually from 1 to 2V or more) depends on the selected EC (Electret Condenser) microphone. The 1nf decoupling capacitance used in MICBIAS pin removes high frequency noise of the chip.

Setting SB_MIC1/SB_MIC2 to 1 will close microphone input path for saving power, also setting SB_MICBIAS to 1 will close MICBIAS stage and the MICBIAS output voltage will be zero.

MICBIAS output voltage scales with AVDCDC, equals to 5/6*AVDCDC (typical 2.08V).
MICBIAS output current is 4mA max.
MICBIAS output noise is 40uVrms max.

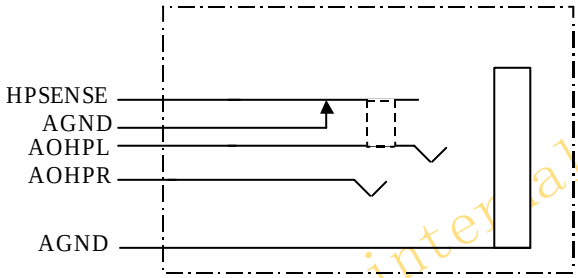
This configuration is better suited for microphone with single wire + shielding.



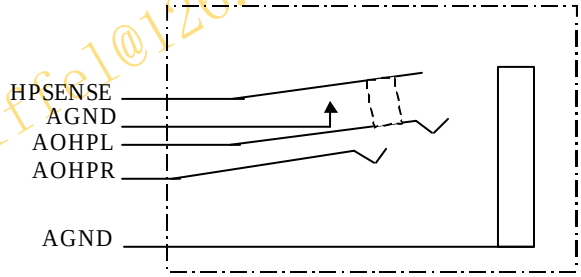
The AVSCDC Pin is connected the analog quiet reference ground in the chip (refers to Grounds and analog signal references). So the ground of MIC must be connected to AVSCDC using a star connection.

10.21.4 Description of the connections to the jack

When the jack is inserted, “sense” and “ground” are disconnected.



No jack plugged: the switch acts as a short-circuit.



Jack plugged: the switch acts as an open circuit.

10.21.4.1 Grounds and analog signal references

In order to limit the parasitic disturbances from the AVSHP output power supplies to inter VREFN analog quiet ground(which is using AVSCDC pin), should use the following principle to distribute the grounds.

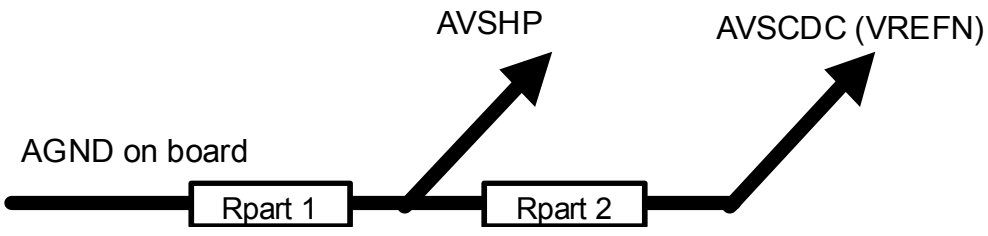


Figure 10-14 Ground distributing

Minimize the values of the connections parasitic resistance Rpar1, Rpar2.

Take a special care for Rpar1 in order to limit the disturbance from the output stages (AVSHP) to the signal reference (VREFN).

The reference of the input signals must be connected to VREFN (internal quiet ground which using the AVSCDC pin) using a star connection.

10.21.5 PCB considerations

To work properly, CODEC analog power supplies require external decoupling capacitors.

In the VCAP pin, one 10uF low ESR (ceramic or tantalum) called C2 and one 100nF ceramic called C1 have to be used. The ceramic capacitor has to be kept as close as possible to IC package (closer than 0.2 inch).

For each power supply pin, one 100nF ceramic capacitor has to be used. The capacitor used in AVDCDC pin is called C4, the capacitor used in AVDHP pin is C6. These ceramic capacitors have to be kept as close as possible to IC package (closer than 0.2 inch).

One low ESR (ceramic or tantalum) capacitor called C3 has to be used to decouple the analog power supply provided to the CODEC. Its value depends on the power supply generator; its typical value is between 1uF and 10uF. Ideally use separate ground planes for analog and digital parts.

C1, C2, C3, C4, C5, C6 are defined in section "Required external components".

The reference PCB design is shown below:

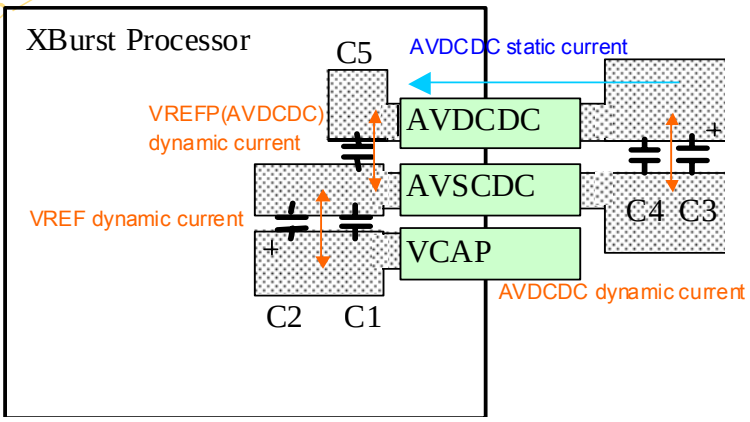


Figure 10-15 the bottom corner of chip PCB Layer

This is just an example reference diagram. You should change and select the PCB layer and route with

your design constraints.

10.21.5.1 Required external components

The following table summarizes the external components required for a proper working of CODEC, except those used for the analog input and output signals.

Name	Description	Typical Value	Unit
C1	Ceramic reference decoupling capacitor. Cext.	100	nF
C2	Tantalum reference decoupling capacitor. Cext.	10	uF
C3	Tantalum analog power supply decoupling capacitor.	1 to 10	uF
C4	Ceramic AVDCDC decoupling capacitor.	100	nF
C5	Ceramic inter signal VREFP decoupling capacitor (can be shared with C4).	100	nF
C6	Ceramic AVDHP decoupling capacitor. Not shown in section PCB considerations.	100	nF
C8	MICBIAS decoupling capacitor, Refer to section "Microphone connection".	1	nF
C9, C10	External bypass capacitor, for DC blocking, Refer to section "Headphone connection (Capacitor-coupled)".	220	uF
Rhpl, Rhpr	Headphone jack pull-down resistors, Refer to section "Headphone connection (Capacitor-coupled)".	470 or 4.7K	Ohm
R	In single-ended connection, one external resistor (R) has to be used to bias the electret microphone. Refer to section "Headphone connection (Capacitor-coupled)".	2.2K ~ 4.7K	Ohm

10.22 Main paths characteristics

10.22.1 Line input to audio ADC path

Measurement conditions: T = 25°C, AVDCDC = AVDHP = VREFP = 2.5V, input sine wave with a frequency of 1kHz, Fmclk = 12MHz, Fs = 8 to 96kHz, measurement bandwidth 20Hz – 20kHz, unless otherwise specified.					
Parameter	Test conditions	Min.	Typ	Max.	Unit
Input level	Full Scale, Gain GIDL, GIDR = 0dB (note 1)	1.89	2.12	2.39	Vpp
Input resistance		8.5			kOhm
Input capacitance	Includes 10pF for ESD, bonding and package pins capacitances			25	pF
Input bypass capacitor	Cbyline		1		uF

Note 1: The Full Scale input voltage scales with AVDCDC, equals to 0.85*VREF (typ)

10.22.2 Microphone input to audio ADC path

Measurement conditions: T = 25°C, AVDCDC = AVDHP = VREFP = 2.5V, input sine wave with a frequency of 1kHz, Fmclk = 12MHz, Fs = 8 to 96kHz, measurement bandwidth 20Hz – 20kHz, unless otherwise specified.					
Parameter	Test conditions	Min.	Typ	Max.	Unit
Input level	Full Scale, Gain GIDL, GIDR = 0dB, boost gain GIM1,GIM2 = 20dB (note 1)	0.189	0.212	0.239	Vpp
Input resistance (Differential mic configuration)	Boost gain GIM1,GIM2 = 0 dB	66	80	100	kOhm
	Boost gain GIM1,GIM2 = 20 dB	10	13	15	
Input resistance (single-ended mic configuration)	Boost gain GIM1,GIM2 = 0 dB	92	115	138	kOhm
	Boost gain GIM1,GIM2 = 20 dB	19	24	29	
Input capacitance	Includes 10pF for ESD, bonding and package pins capacitances			25	pF
Input bypass capacitor	Cbyline		1		uF

Note 1: The Full Scale input voltage scales with AVDCDC, equals to 0.085*VREF (typ)

10.22.3 Audio DAC to headphone output path

Measurement conditions: T = 25°C, AVDCDC = AVDHP = VREFP = 2.5V, input sine wave with a frequency of 1kHz, Fmclk = 12MHz, Fs = 8 to 96 kHz, measurement bandwidth 20Hz – 20kHz, unless otherwise specified.					
Parameter	Test conditions	Min.	Typ	Max.	Unit
DAC playback on 16 Ohm HeadPhone					
Output level	Full Scale, Gain GOL, GOR = -3 dB, GODL, GODR=0dB, 16 Ohm load	1.33	1.5	1.69	Vpp
Maximum output power	Rl = 16 Ohm		17.6		mW
Output resistance	R1	16			Ohm
Output bypass capacitor	Cl (Rl = 16 Ohm)			220	uF
DAC playback to 10k Ohms lineout single					
Output level	Full Scale, Gain GOL, GOR = 0 dB, GODL, GODR=0dB (note 1)	1.89	2.12	2.39	Vpp
Output resistance	R1	10k			Ohm
Output bypass capacitor	Cl (Rl = 10 kOhm)			1	uF
Common characteristics					
Output capacitance (note 2)	Cp			200	pF

Note 1: The Full Scale output voltage scales with AVDCDC, equals to 0.85*VREF. The minimum and maximum output levels are given with gain accuracy.

Note 2: Output may oscillate above specified load capacitances. The capacitance is equivalent to a 2-meter cable.

10.22.4 Audio DAC to mono line output path

Measurement conditions: T = 25°C, AVDCDC = AVDHP = VREFP = 2.5V, input sine wave with a frequency of 1kHz, Fmclk = 12MHz, Fs = 8 to 96kHz, measurement bandwidth 20Hz – 20kHz, unless otherwise specified.					
Parameter	Test conditions	Min.	Typ	Max.	Unit
Output level	Full Scale, Gain GODL, GODR = 0dB (note 1)	3.78	4.25	4.78	Vpp
Output resistance		10			kOhm
Output capacitance	Cp			100	pF
Output bypass capacitor	Cl (Rl = 10 kOhm)			1	uF

Note 1: The Full Scale output voltage scales with AVDCDC, equals to 1.7*VREF (typ)

10.22.5 Line input to headphone output path (analog bypass)

Measurement conditions: T = 25°C, AVDCDC = AVDHP = VREFP = 2.5V, input sine wave with a frequency of 1kHz, Fmclk = 12MHz, Fs = 8 to 96 kHz, measurement bandwidth 20Hz – 20kHz, unless otherwise specified.					
Parameter	Test conditions	Min.	Typ	Max.	Unit
Input level	Full Scale	1.89	2.12	2.39	Vpp
Input resistance		8.5			kOhm
bypass on 16 Ohm HeadPhone					
Output level	Full Scale, Gain GOL, GOR = -3 dB, GIL, GIR=0dB, 16 Ohm load	1.33	1.5	1.69	Vpp
Output resistance	R1	16			Ohm
bypass to 10k Ohms lineout single					
Output level	Full Scale, Gain GOL, GOR = 0 dB, GIL, GIR=0 dB (note 1)	1.89	2.12	2.39	Vpp
Common characteristics					
Input capacitance	Includes 10pF for ESD, bonding and package pins capacitances			25	pF
Input bypass capacitor	Cbyline		1		uF

Note 1: The Full Scale output voltage scales with AVDCDC, equals to 1.7*VREF (typ)

10.22.6 Microphone input to headphone output path (analog sidetone)

Measurement conditions: T = 25°C, AVDCDC = AVDHP = VREFP = 2.5V, input sine wave with a frequency of 1kHz, Fmclk = 12MHz, Fs = 8 to 96kHz, measurement bandwidth 20Hz – 20kHz, unless otherwise specified.					
Parameter	Test conditions	Min.	Typ	Max.	Unit
Input level	Full Scale, Gain GOL, GOR = 0dB, boost gain GIM1,GIM2 = 20dB (note 1)	0.189	0.212	0.239	Vpp
Output level	Full Scale, Gain GOL,GOR= 0dB, boost gain GIM1,GIM2 = 0 to 20dB, 10kOhm load (note 2)	1.89	2.12	2.39	Vpp
	Full Scale, Gain GOL,GOR= -3 dB, boost gain GIM1,GIM2 = 0 to 20dB, 16Ohm load (note 2)	1.33	1.5	1.69	Vpp

Note 1: The Full Scale input voltage scales with AVDCDC, equals to 0.085*VREF (typ)

Note 2: The Full Scale output voltage scales with AVDCDC, equals to 0.85*VREF (typ)

10.22.7 Micbias and reference

Measurement conditions: T = 25°C, AVDCDC = AVDHP = VREFP = 2.5V, input sine wave with a frequency of 1kHz, Fmclk = 12MHz, Fs = 8 to 96kHz, measurement bandwidth 20Hz – 20kHz, unless otherwise specified.					
Parameter	Test conditions	Min.	Typ	Max.	Unit
Micbias output level	(note 1)		2.08 1.66		V
Micbias output current				4	mA
Micbias decoupling capacitor	Cmic	0.75	1	1.25	nF
VCAP voltage	(note 2)		2		V

Note 1: Micbias output voltage scales with AVDCDC, equals to 5/6*VREF or 4/6*VREF (typ)

Note 2: VCAP output voltage scales with AVDCDC, equals to 0.8*VREF (typ)

11 AC97/I2S/SPDIF Controller

11.1 Overview

This chapter describes the AIC (AC'97 and I^2S Controller) included in this 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 processor's 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.

SPDIF (Sony/Philips Digital Interconnect Format) is a digital audio interface. The transmission medium can be either electrical or optical. Supports IEC60958 two-channel PCM audio and IEC61937 multi-channel compressed audio (Dolby Digital, DTS, etc.).

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.

11.1.1 Block Diagram

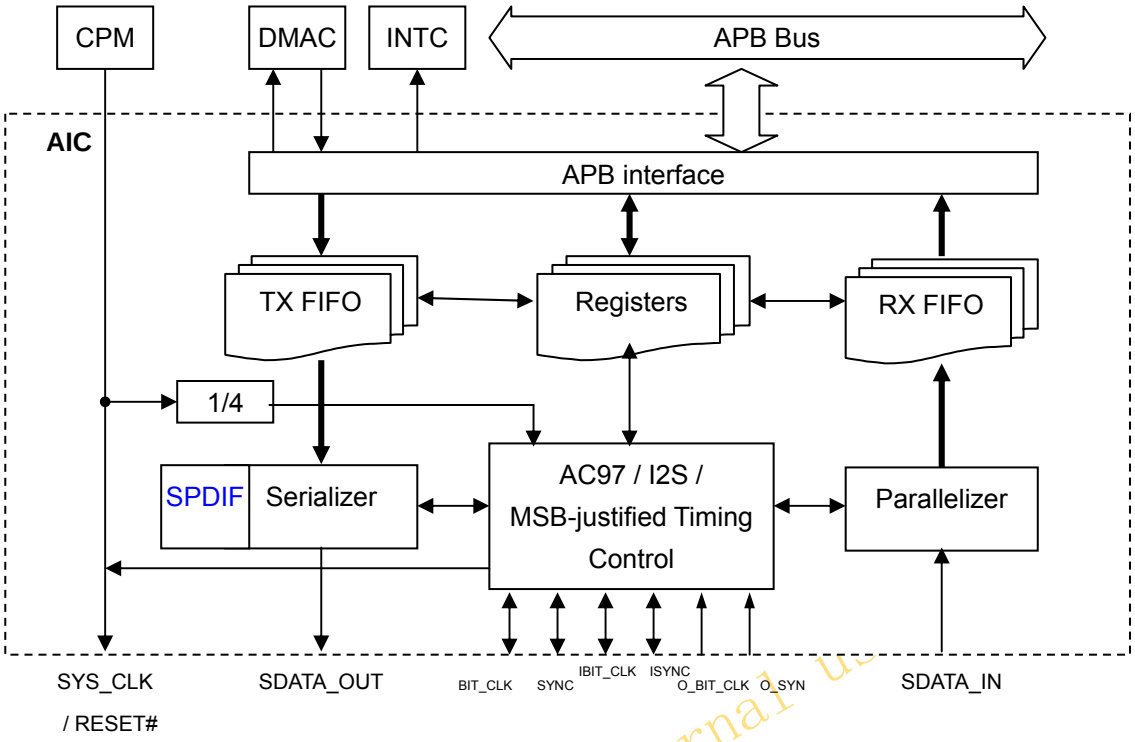


Figure 11-1 AIC Block Diagram

The O_BIT_CLK and O_SYNC ports are only used by inter CODEC.

11.1.2 Features

AIC support following AC97/I2S/SPDIF features:

AC-link (AC97) features

- Up to 20 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 normal 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

I2S features

- 8, 16, 18, 20 and 24 bit audio sample data sizes supported, 16 bits packed sample data is supported
- Up to 8 channels sample data 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 normal audio samples play back
- 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

SPDIF 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 IEC60958 two-channel PCM audio
- Support IEC61937 multi-channel compressed audio
- Support consumer mode and only support transmitter mode
- Profession mode is not supported
- The User data bit is '0'as it is not supported in the chip
- Support sampling frequency from 32kHz to 192kHz

11.1.3 Interface Diagram

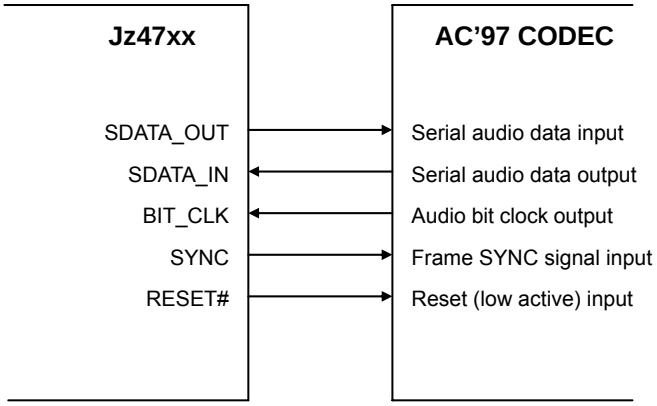


Figure 11-2 Interface to an External AC'97 CODEC Diagram

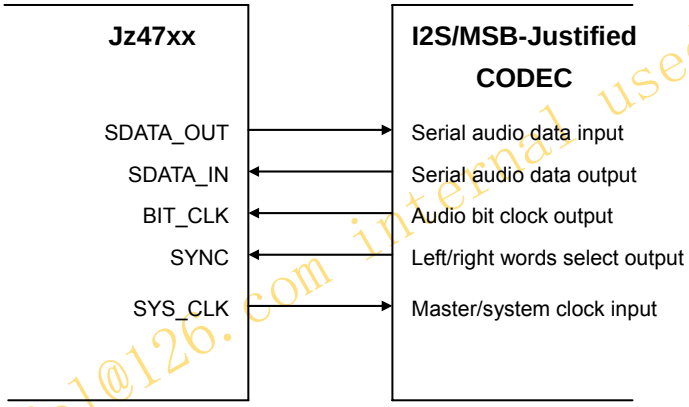


Figure 11-3 Interface to an External Master Mode I2S/MSB-Justified CODEC Diagram (Share Clock Mode)

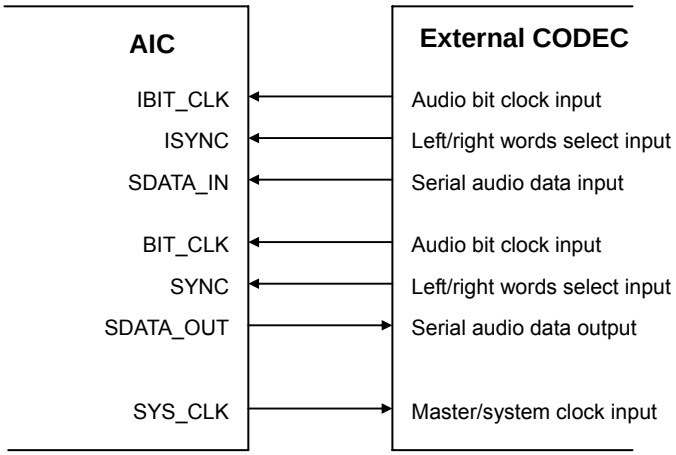


Figure 11-4 Interface to an External Master Mode I2S CODEC Diagram (Split Clock Mode)

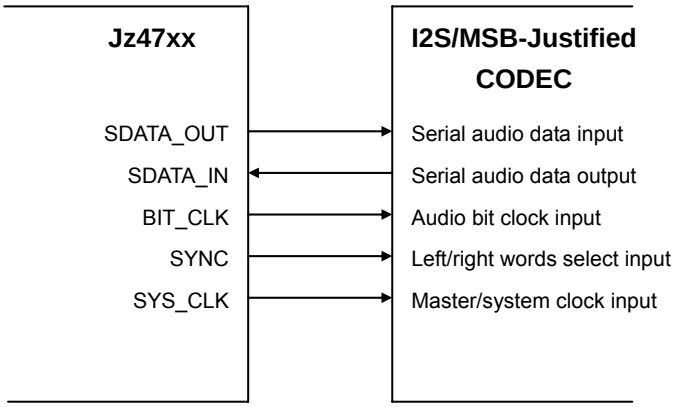


Figure 11-5 Interface to an External Slave Mode I2S/MSB-Justified CODEC Diagram (Share Clock Mode)

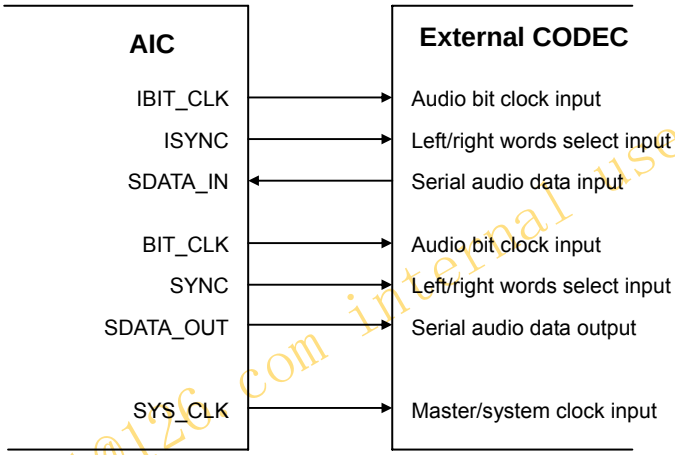


Figure 11-6 Interface to an External Slave Mode I2S CODEC Diagram (Split Clock Mode)

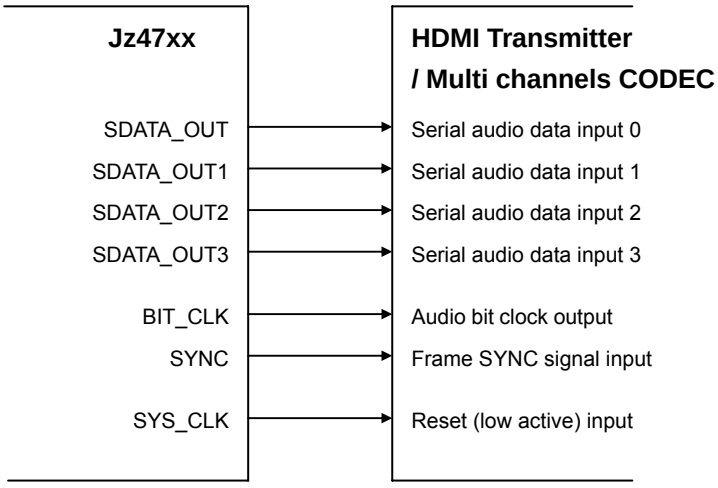


Figure 11-7 Interface to a HDMI Transmitter via I2S Diagram

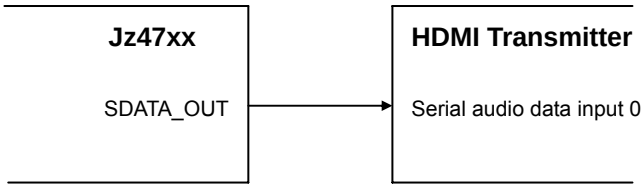


Figure 11-8 Interface to a HDMI Transmitter via SPDIF Diagram

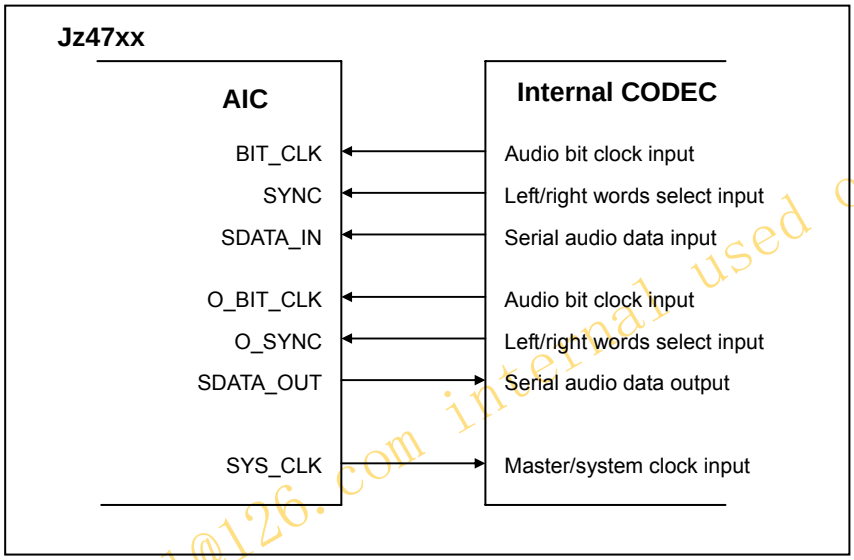


Figure 11-9 Interface to an internal Master Mode I2S CODEC Diagram

Please refer to the related CODEC specification for the details.

11.1.4 Signal Descriptions

There are all 5 pins used to connect between AIC and an external audio CODEC device. If an internal CODEC is used, these pins are not needed. Please refer to Chip Spec. They are listed and described in Table 11-1.

Table 11-1 AIC Pins Description

Function Name	PIN Name	I/O	Description
RESET# **SYS_CLK**	SCLK_RSTN	O	RESET#: AC-link format, active-low CODEC reset. SYS_CLK: I2S/MSB-Justified formats, supply system clock to CODEC.

IBIT_CLK	BCLK_AD	I/O	Sample rate dependent bit-rate clock input/output for I2S/MSB-Jistified, only input AD channel.
ISYNC	SYNC_AD	I/O	Indicates the left- or right-channel for I2S/MSB-Justified format, only input AD channel
SDATA_IN	SDATI	I	Serial audio input data from CODEC.
BIT_CLK	BCLK	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	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	SDATO	O	Serial audio output data to CODEC / I2S line 0 / SPDIF output.
SDATA_OUT1	SDATO1	O	Serial audio output data I2S line 1.
SDATA_OUT2	SDATO2	O	Serial audio output data I2S line 2.
SDATA_OUT3	SDATO3	O	Serial audio output data I2S line 3.

The O_BIT_CLK and O_SYNC signals are not connected to any pin for only using by internal CODEC.

11.1.4.1 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.

11.1.4.2 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 11-7 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.

The IBIT_CLK is for the SDATA_IN signal on division CLOCK function.

11.1.4.3 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.

The ISYNC is for the SDATA_IN signal on division CLOCK function.

11.1.4.4 SDATA_OUT Pin

SDATA_OUT is AIC output data pin, which outputs AC97/I2S serial audio data, SPDIF serial data or data of AC97 CODEC register control to an external audio CODEC device.
If in multi channels mode, it outputs the first two channels serial data.
If AIC is disabled, it retains the low.

SDATA_OUTn (n = 1,2,3) is AIC output data pin, which outputs multi channels serial audio data.

11.1.4.5 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.

11.1.4.6 IBIT_CLK Pin

The IBIT_CLK is for the SDATA_IN signal on division CLOCK function. When in Split Clock mode(AICFR.DMODE = 1), this pin is effective.

11.1.4.7 ISYNC Pin

The ISYNC is for the SDATA_IN signal on division CLOCK function. When in Split Clock mode(AICFR.DMODE = 1), this pin is effective.

11.2 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 11-2 AIC Registers Description

Name	Description	RW	Reset value	Address	Size
AICFR	AIC Configuration Register	RW	0x07100000	0x10020000	32
AICCR	AIC Common Control Register	RW	0x01240000	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
SPENA	SPDIF Enable Register	RW	0x00	0x10020080	8
SPCTRL	SPDIF Control Register	RW	0x0003	0x10020084	16
SPSTATE	SPDIF Status Register	RW	0x0000	0x10020088	16
SPCFG1	SPDIF Configure 1 Register	RW	0x00000000	0x1002008C	32
SPCFG2	SPDIF Configure 2 Register	RW	0x00000000	0x10020090	32
SPFIFO	SPDIF FIFO Register	W	0x????????	0x10020094	32
CKCFG	Clock Configure for the embedded CODEC to AIC	RW	0x00000000 0x00000002	0x100200A0	32
RGADW	Address, data in and write command for accessing to internal registers of embedded CODEC	RW	0x00000000	0x100200A4	32
RGDATA	The read out data and interrupt request status of Internal registers	R	0x00000000	0x100200A8	32

	data in the embedded CODEC				

1 AICFR is used to control FIFO threshold, AC-link or I2S/MSB-justified selection, AIC reset, master/slave selection, and AIC enable.
2 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.
3 ACCR1 is used to reflect/control valid incoming/outgoing slots of AC97.
4 ACCR2 is used to control interrupt enable, output/input sample size, and alternative control of RESET#, SYNC and SDATA_OUT pins in AC-link.
5 I2SCR is used to control BIT_CLK stop, audio sample size, I2S or MSB-justified selection in I2S/MSB-justified.
6 AICSR is used to reflect FIFOs status.
7 ACSR is used to reflect the status of the connected external CODEC in AC-link.
8 I2SSR is used to reflect AIC status in I2S/MSB-justified.
9 ACCAR and ACCDR are used to hold address and data for AC-link CODEC register read/write.
10 ACSAR and ACSDR are used to receive AC-link CODEC registers address and data.
11 I2SDIV is used to set clock divider for BIT_CLK generating in I2S/MSB-justified format.
12 AICDR is act as data input/output port to/from transmit/receive FIFO when write/read.
13 CKCFG, RGADW and RGDATA are used to access internal CODEC, please refer to CODEC Spec.

11.2.1 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				Reserved			TFTH					Reserved						IBCKD	ISYNCD	DMODE	Reserved	LSMP	ICDC	AUSEL	RST	BCKD	SYNCD	ENB
RST	0	0	0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:28	Reserved	Writing has no effect, read as zero.	R
27:24	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	RW

<table>
<tr><td></td><td></td><td>involve receive FIFO overflow. The optimum value is system dependent.</td><td></td></tr>
<tr><td>23:21</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>20:16</td><td>TFTH</td><td>Transmit FIFO threshold for interrupt or DMA request. The TFTH valid value 0 ~ 31.
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.</td><td>RW</td></tr>
<tr><td>15:11</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>10</td><td>IBCKD</td><td>The IBIT_CLK Direction. This bit specifies input/output direction of IBIT_CLK. It is only valid in I2S/MSB-justified format. When AC-link format is selected, IBIT_CLK is always input and this bit is ignored. Change this bit in case of IBIT_CLK is stopped (I2SCR.ISTPBK = 1). This is only available when DMODE = 1.
<table>
<tr><th>BCKD</th><th>BIT_CLK Direction</th></tr>
<tr><td>0</td><td>IBIT_CLK is input from an external source.</td></tr>
<tr><td>1</td><td>IBIT_CLK is generated internally and driven out to the CODEC.</td></tr>
</table></td><td>RW</td></tr>
<tr><td>9</td><td>ISYNCD</td><td>ISYNC Direction. This bit specifies input/output direction of ISYNC in I2S/MSB-justified format. When AC-link format is selected, ISYNC is always output and this bit is ignored. Change this bit in case of IBIT_CLK is stopped (I2SCR.ISTPBK = 1).
This is only available when DMODE = 1.
<table>
<tr><th>SYNCD</th><th>SYNC Direction</th></tr>
<tr><td>0</td><td>ISYNC is input from an external source.</td></tr>
<tr><td>1</td><td>ISYNC is generated internally and driven out to the CODEC.</td></tr>
</table></td><td>RW</td></tr>
<tr><td>8</td><td>DMODE</td><td>The Division Clock Mode control.
<table>
<tr><th>BCKD</th><th>BIT_CLK Direction</th></tr>
<tr><td>0</td><td>Disable. Shared clock mode.
The BIT_CLK, SYNC, SDATA_IN and SDATA_OUT are configured to one shared clock I2S channel.</td></tr>
<tr><td>1</td><td>Enable. Spilt clock mode.
The BIT_CLK, SYNC and SDATA_OUT are configured to one output I2S channel.
The IBIT_CLK, ISYNC and SDATA_IN are configured to one input I2S channel.</td></tr>
</table></td><td>RW</td></tr>
<tr><td>7</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>6</td><td>LSMP</td><td>Select between play last sample or play ZERO sample in TX FIFO underflow. ZERO sample means sample value is zero. This bit is better</td><td>RW</td></tr>
</table>

<table>
<tr><td></td><td></td><td>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></td><td></td></tr>
<tr><td>5</td><td>ICDC</td><td>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></td><td>RW</td></tr>
<tr><td>4</td><td>AUSEL</td><td>Audio Unit Select. Select between AC-link and I2S/MSB-justified. Change this bit in case of BIT_CLK is stopped (I2SCR.STPBK = 1).<table><tr><th>AUSEL</th><th>Selected</th></tr><tr><td>0</td><td>Select AC-link format.</td></tr><tr><td>1</td><td>Select I2S/MSB-justified format.</td></tr></table></td><td>RW</td></tr>
<tr><td>3</td><td>RST</td><td>Reset AIC. Write 1 to this bit reset AIC registers and FIFO except AICFR and I2SDIV register. Writing 0 to this bit has no effect and this bit is always reading 0.</td><td>W</td></tr>
<tr><td>2</td><td>BCKD</td><td>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).<table><tr><th>BCKD</th><th>BIT_CLK Direction</th></tr><tr><td>0</td><td>BIT_CLK is input from an external source.</td></tr><tr><td>1</td><td>BIT_CLK is generated internally and driven out to the CODEC.</td></tr></table></td><td>RW</td></tr>
<tr><td>1</td><td>SYNCD</td><td>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. Change this bit in case of BIT_CLK is stopped (I2SCR.STPBK = 1).<table><tr><th>SYNCD</th><th>SYNC Direction</th></tr><tr><td>0</td><td>SYNC is input from an external source.</td></tr><tr><td>1</td><td>SYNC is generated internally and driven out to the CODEC.</td></tr></table></td><td>RW</td></tr>
<tr><td>0</td><td>ENB</td><td>Enable AIC function. This bit is used to enable or disable the AIC function.<table><tr><th>ENB</th><th>Description</th></tr><tr><td>0</td><td>Disable AIC Controller.</td></tr><tr><td>1</td><td>Enable AIC Controller.</td></tr></table></td><td>RW</td></tr>
</table>

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.**

11.2.2 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		Reserved		PACK16	Reserved		CHANNEL			Reserved		OSS		ISS		RDMS	TDMS	Reserved		M2S	ENDSW	ASVTSU	TFLUSH	RFLUSH	EROR	ETUR	ERFS	ETFS	ENLBF	ERPL	EREC
RST	0	0	0	0	0	0	0	1	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
31:30	Reserved	Writing has no effect, read as zero.	R
29	Reserved	Keep this value to 0 in normal use.	R
28	PACK16	Output Sample data 16bit packed mode select. This bit reflects that one word contains two sample data or only one sample data with LSB align. The packed mode is only support 16bit sample size. **PACK16** / **Sample Size** 0: Unpacked data mode. One word only contains one 16bit sample data aligned LSB. 1: Packed data mode. One word contains two 16 bit sample data.	RW
27	Reserved	Writing has no effect, read as zero.	R
26:24	CHANNEL	Output Channel Number Select. These bits reflect output data channels from AIC to device. The data supported are: 1(mono), 2(stereo), 4, 6 and 8 channels. The sample data is LSB-justified in memory/register. **CHANNEL** / **Sample Size** 0x0: 1 channel, mono, Only SDATA0 used. 0x1: 2 channels, stereo, Only SDATA0 used. 0x2: Reserved. 0x3: 4 channels, SDATA0 and SDATA1 used.	RW

<table>
<tr><td></td><td></td><td><table>
<tr><td>0x4</td><td>Reserved.</td></tr>
<tr><td>0x5</td><td>6 channels, SDATA0 to SDATA2 used.</td></tr>
<tr><td>0x6</td><td>Reserved.</td></tr>
<tr><td>0x7</td><td>8 channels, , SDATA0 to SDATA3 used.</td></tr>
</table></td><td></td></tr>
<tr><td>23:22</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>21:19</td><td>OSS</td><td>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.<table>
<tr><th>OSS</th><th>Sample Size</th></tr>
<tr><td>0x0</td><td>8 bit.</td></tr>
<tr><td>0x1</td><td>16 bit.</td></tr>
<tr><td>0x2</td><td>18 bit.</td></tr>
<tr><td>0x3</td><td>20 bit.</td></tr>
<tr><td>0x4</td><td>24 bit.</td></tr>
<tr><td>0x5~0x7</td><td>Reserved.</td></tr>
</table></td><td>RW</td></tr>
<tr><td>18:16</td><td>ISS</td><td>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.<table>
<tr><th>ISS</th><th>Sample Size</th></tr>
<tr><td>0x0</td><td>8 bit.</td></tr>
<tr><td>0x1</td><td>16 bit.</td></tr>
<tr><td>0x2</td><td>18 bit.</td></tr>
<tr><td>0x3</td><td>20 bit.</td></tr>
<tr><td>0x4</td><td>24 bit.</td></tr>
<tr><td>0x5~0x7</td><td>Reserved.</td></tr>
</table></td><td>RW</td></tr>
<tr><td>15</td><td>RDMS</td><td>Receive DMA enable. This bit is used to enable or disable the DMA during receiving audio data.<table>
<tr><th>RDMS</th><th>Receive DMA</th></tr>
<tr><td>0</td><td>Disabled.</td></tr>
<tr><td>1</td><td>Enabled.</td></tr>
</table></td><td>RW</td></tr>
<tr><td>14</td><td>TDMS</td><td>Transmit DMA enable. This bit is used to enable or disable the DMA during transmit audio data.<table>
<tr><th>TDMS</th><th>Transmit DMA</th></tr>
<tr><td>0</td><td>Disabled.</td></tr>
<tr><td>1</td><td>Enabled.</td></tr>
</table></td><td>RW</td></tr>
<tr><td>13:12</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>11</td><td>M2S</td><td>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. Change this before replay started.<table>
<tr><th>M2S</th><th>Description</th></tr>
</table></td><td>RW</td></tr>
</table>

			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. **ENDSW** / **Description** 0 — **No change on outgoing sample data.** 1 — **Swap high and low byte for outgoing 16-bits size sample data.**		RW
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. **ASVTSU** / **Description** 0 — **No audio sample value signed ←→unsigned transfer.** 1 — **Do audio sample value signed ←→unsigned transfer.**		RW
8	TFLUSH	Transmit FIFO Flush. Write 1 to this bit flush transmit FIFOs to empty. Writing 0 to this bit has no effect and this bit is always reading 0.		W
7	RFLUSH	Receive FIFO Flush. Write 1 to this bit flush receive FIFOs to empty. Writing 0 to this bit has no effect and this bit is always reading 0.		W
6	EROR	Enable ROR Interrupt. This bit is used to control the ROR interrupt enable or disable. **EROR** / **ROR Interrupt** 0 — **Disabled.** 1 — **Enabled.**		RW
5	ETUR	Enable TUR Interrupt. This bit is used to control the TUR interrupt enable or disable. **ETUR** / **TUR Interrupt** 0 — **Disabled.** 1 — **Enabled.**		RW
4	ERFS	Enable RFS Interrupt. This bit is used to control the RFS interrupt enable or disable. **ERFS** / **RFS Interrupt** 0 — **Disabled.** 1 — **Enabled.**		RW
3	ETFS	Enable TFS Interrupt. This bit is used to control the TFS interrupt enable or disable. **ETFS** / **TFS Interrupt**		RW

			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.**	

11.2.3 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	Writing has no effect, read as zero.	R
25:16	RS	Receive Valid Slots. These bits are used to indicate which incoming slots are valid. Slot 3 is mapped to bit 16 or RS[0], slot 4 to bit 17 or RS[1] and so on. When write to this field, a bit 1 means we expect a PCM data in the corresponding slot, a bit 0 means the corresponding slot PCM data will be discarded. When read from this field, a bit 1 means we receive an expected valid PCM data in the corresponding slot. This field should be written before record started.	RW
		RS[n] Value	**Description**

			0	**Slot n+3 is invalid.**	
			1	**Slot n+3 has valid PCM data.**	
15:10	Reserved	Writing has no effect, read as zero.			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. This field should be written before record and replay started.			RW
			XS[n] Value	**Description**	
			0	**Slot n+3 is invalid.**	
			1	**Slot n+3 has valid PCM data.**	

11.2.4 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	0	0

Bits	Name	Description	RW
31:19	Reserved	Writing has no effect, read as zero.	R
18	ERSTO	Enable RSTO Interrupt. This bit is used to control the RSTO interrupt enable or disable.	RW
		ERSTO / **RSTO Interrupt**: 0 – **Disabled.**; 1 – **Enabled.**	
17	ESADR	Enable SADR Interrupt. This bit is used to control the SADR interrupt enable or disable.	RW
		ESADR / **SADR Interrupt**: 0 – **Disabled.**; 1 – **Enabled.**	
16	ECADT	Enable CADT Interrupt. This bit is used to control the CADT interrupt enable or disable.	RW

			ECADT: 0 = Disabled.; 1 = Enabled.
15:4	Reserved	Writing has no effect, read as zero.	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: 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 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.	RW

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.

11.2.5 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	31–18	17	16	15–14	13	12	11–5	4	3–1	0
	Reserved	RFIRST	SWLH	Reserved	ISTPBK	STPBK	Reserved	ESCLK	Reserved	AMSL
RST	0	0	0	0	0	0	0	0	0	0

<table>
<tr><th>Bits</th><th>Name</th><th>Description</th><th>RW</th></tr>
<tr><td>31:18</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>17</td><td>RFIRST</td><td>Send R channel first in stereo mode. This bit should only be set in 2 channels configuration. The frame is LR like or RL like. It takes effective immediately when the bit is changed.
Change this before replay started.
<table><tr><th>RFIRST</th><th>Description</th></tr><tr><td>0</td><td>Send L channel first. (LR)</td></tr><tr><td>1</td><td>Send R channel first. (RL)</td></tr></table></td><td>RW</td></tr>
<tr><td>16</td><td>SWLH</td><td>Switch LR channel in 16bit-packed stereo mode.
This bit control whether the low address data is L or R. This bit should only be set in 2 channels configuration and 16bit-packed mode. That means it can only valid with packed mode (PACK16 =1) and 2 channels (CHANNEL=0x1).
It takes effective immediately when the bit is changed. Change this before replay started.
<table><tr><th>SWLH</th><th>Description</th></tr><tr><td>0</td><td>16 bit LSB and 16bit MSB is not switched.</td></tr><tr><td>1</td><td>16 bit LSB and 16bit MSB is switched.</td></tr></table></td><td>RW</td></tr>
<tr><td>15:12</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>13</td><td>ISTPBK</td><td>Stop IBIT_CLK. It is used to stop the IBIT_CLK in I2S/MSB-justified format. When AC-link is selected, all of its operations are ignored.
<table><tr><th>STPBK</th><th>Description</th></tr><tr><td>0</td><td>IBIT_CLK is not stopped.</td></tr><tr><td>1</td><td>IBIT_CLK is stopped.</td></tr></table>
Please set this bit to 1 to stop IBIT_CLK when change AICFR.AUSEL and AICFR.IBCKD.</td><td>RW</td></tr>
<tr><td>12</td><td>STPBK</td><td>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><th>STPBK</th><th>Description</th></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.</td><td>RW</td></tr>
<tr><td>11:5</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>4</td><td>ESCLK</td><td>Enable SYSCLK output. When this bit is 1, the SYSCLK outputs to chip outside is enabled. Else, the clock is disabled.</td><td>RW</td></tr>
<tr><td>3:1</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
</table>

0	AMSL	Specify Alternate Mode (I2S or MSB-Justified) Operation. **AMSL** / **Description** 0 / Select I2S Operation Mode. 1 / Select MSB-Justified Operation Mode.	RW

11.2.6 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	Writing has no effect, read as zero.	R
29:24	RFL	Receive FIFO Level. The bits indicate the amount of valid PCM data in Receive FIFO. **RFL Value** / **Description** 0x00 ~ 0x20 / RFL valid PCM data in receive FIFO. 0x21 ~ 0x3F / Reserved.	R
23:14	Reserved	Writing has no effect, read as zero.	R
13:8	TFL	Transmit FIFO Level. The bits indicate the amount of valid PCM data in Transmit FIFO. **TFL Value** / **Description** 0x00 ~ 0x20 / TFL valid PCM data in transmit FIFO. 0x21 ~ 0x3F / Reserved.	R
7	Reserved	Writing has no effect, read as zero.	R
6	ROR	Receive FIFO Over Run. This bit indicates that receive FIFO has or has not experienced an overrun. **ROR** / **Description** 0 / When read, indicates over-run has not been found. When write, clear itself. 1 / When read, indicates data has even been written to full receive FIFO. When write, not effects.	RW
5	TUR	Transmit FIFO Under Run. This bit indicates that transmit FIFO has or has not experienced an under-run. **TUR** / **Description**	RW

		0	When read, indicates under-run has not been found.	
			When write, clear itself.	
		1	When read, indicates data has even been read from empty transmit FIFO.	
			When write, not effects.	
4	RFS	Receive FIFO Service Request. This bit indicates that receive FIFO level is or not below receive FIFO 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. **RFS** / **Description** 0 / Receive FIFO level below RFL threshold. 1 / Receive FIFO level at or above RFL threshold.		R
3	TFS	Transmit FIFO Service Request. This bit indicates that transmit FIFO level is below Transmit FIFO 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. **TFS** / **Description** 0 / Transmit FIFO level exceeds TFL threshold. 1 / Transmit FIFO level at or below TFL threshold.		R
2:0	Reserved	Writing has no effect, read as zero.		R

11.2.7 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	Writing has no effect, read as zero.	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

<table>
<tr><td>20</td><td>CRDY</td><td>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></td><td>R</td></tr>
<tr><td>19</td><td>CLPM</td><td>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></td><td>R</td></tr>
<tr><td>18</td><td>RSTO</td><td>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.</td><td>RW</td></tr>
<tr><td>17</td><td>SADR</td><td>External CODEC Registers Status Address and Data Received. This bit 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.</td><td>RW</td></tr>
<tr><td>16</td><td>CADT</td><td>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.</td><td>RW</td></tr>
<tr><td>15:0</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
</table>

11.2.8 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																										CHBSY	TBSY	RBSY	BSY	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	Writing has no effect, read as zero.	R
5	CHBSY	AIC Transmitter busy in I2S/MSB-justified format.(Multi-channel status) CHBSY: 0 – AIC Transmitter part is idle or disabled. 1 – AIC Transmitter part currently is transmitting or receiving a frame.	R
4	TBSY	AIC Transmitter busy in I2S/MSB-justified format. TBSY: 0 – AIC Transmitter part is idle or disabled. 1 – AIC Transmitter part currently is transmitting or receiving a frame.	R
3	RBSY	AIC Receiver busy in I2S/MSB-justified format. RBSY: 0 – AIC Receiver part is idle or disabled. 1 – AIC Receiver part currently is transmitting or receiving a frame.	R
2	BSY	AIC busy in I2S/MSB-justified format. BSY: 0 – AIC controller is idle or disabled. 1 – AIC controller currently is transmitting or receiving a frame.	R
1:0	Reserved	Writing has no effect, read as zero.	R

11.2.9 AIC AC97 CODEC Command Address & Data Register (ACCAR, 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 11.4.4 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	Writing has no effect, read as zero.	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. After this field is write, it should not be write again until the operation is finished.	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	Writing has no effect, read as zero.	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. After this field is write, it should not be write again until the operation is finished.	RW

11.2.10 AIC AC97 CODEC Status Address & Data Register (ACSAR, 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 11.4.4 for software flow. These registers are valid only in AC-link format and are ignored in I2S/MSB-justified format.

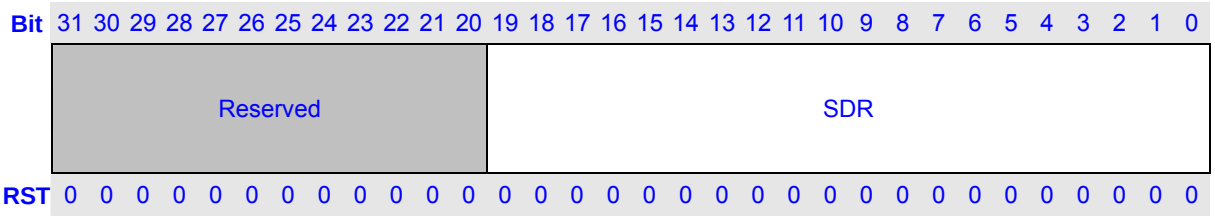
ACSAR 0x10020028

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	Writing has no effect, read as zero.	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**

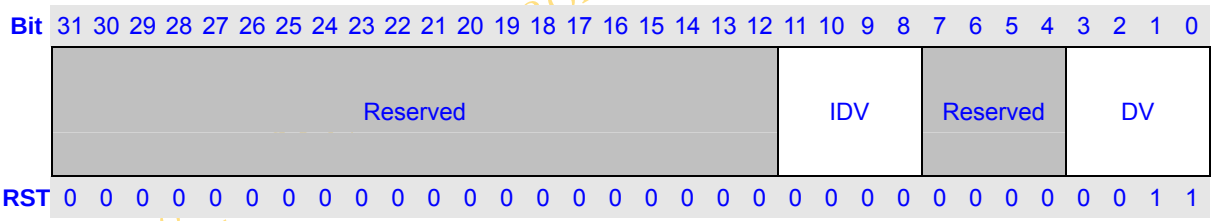


Bits	Name	Description	RW
31:20	Reserved	Writing has no effect, read as zero.	R
19:0	SDR	CODEC Status Data Register. This is used to receive 20-bit AC '97 CODEC status data from SDATA_IN slot 2. The register data of external CODEC is returned. The write operation is ignored.	R

11.2.11 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**



Bits	Name	Description	RW
31:12	Reserved	Writing has no effect, read as zero.	R
11:8	IDV	Audio IBIT_CLK clock divider value minus 1. I2SDIV.IDV is used to control the generating of IBIT_CLK from dividing SYS_CLK. The dividing value should be even and I2SDIV.IDV should be set to the dividing value minus 1. So I2SDIV.IDV bit0 is fixed to 1. IBIT_CLK frequency is fixed to 64 f_S in AIC, where f_S is the audio sample frequency. I2SDIV.IDV depends on SYS_CLK frequency f_{SYS_CLK}, which is selected according to external CODEC's requirement and internal PLL frequency. Please reference to 1.4.10 Serial Audio Clocks and Sampling Frequencies for further description.	RW
7:4	Reserved	Writing has no effect, read as zero.	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	RW

		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 1.4.10 Serial Audio Clocks and Sampling Frequencies for further description.	

11.2.12 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 sample.

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	Writing has no effect, read as zero.	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

11.2.13 SPDIF Enable Register (SPENA)

The register SPENA is used to trigger SPDIF transmitter to work.

SPENA 0x10020080

Bit	7	6	5	4	3	2	1	0
	Reserved							SPEN
RST	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
7:1	Reserved	Writing has no effect, read as zero.	R
0	SPEN	Enable / disable the SPDIF transmitter. 0: SPDIF transmitter is disabled 1: SPDIF transmitter is enabled	RW

11.2.14 SPDIF Control Register (SPCTRL)

The register SPCTRL is used to control SPDIF to work.

SPCTRL 0x10020084

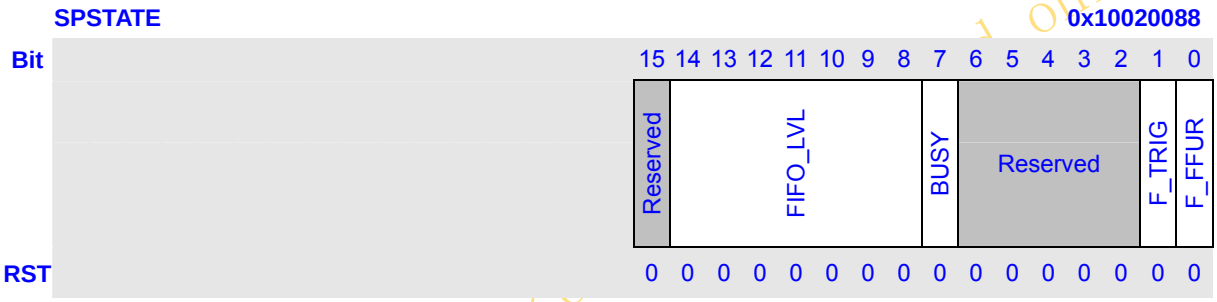
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	DMA_EN	D_TYPE	SIGN_N	INVALID	SFT_RST	SPDIF_I2S	Reserved								M_TRIG	M_FFUR
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bits	Name	Description	RW
15	DMA_EN	DMA transmitter enable bit. 0: DMA transmitter disable 1: DMA transmitter enable	RW
14	D_TYPE	If the bit number of data is less than16, the data in memory is as follows: 0: XXXXXXXXXXXXXXXX | Data 0 XXXXXXXXXXXXXXXX | Data 1 1: Data 1 | Data 0 Data 3 | Data 2	RW
13	SIGN_N	Signed to unsigned or not. If it is 1, the incoming and outgoing audio sample data will be transferred by toggle its most significant bit. 0: Not transfer 1: Do transfer	RW
12	INVALID	Data invalid bit. The data transmitted on SPDIF is valid or not. 0: Valid 1: Invalid	RW
11	SFT_RST	SPDIF FIFO software-reset. Set it to 1 and later it will be cleared by hardware auto. When SFT_RST returns back to 0, the FIFO finish reset. 0: Stop reset 1: Start reset	RW

10	SPDIF_I2S	Choose SPDIF or I2S. 0: I2S 1: SPDIF	
9:2	Reserved	Writing has no effect, read as zero.	R
1	M_TRIG	Trigger interrupt mask. 0: Enabled 1: Masked	RW
0	M_FFUR	FIFO underrun interrupt mask. 0: Enabled 1: Masked	RW

11.2.15 SPDIF State Register (SPSTATE)

The register SPSTATE is used to keep the state of SPDIF.



Bits	Name	Description	RW
15	Reserved	Writing has no effect, read as zero.	R
14:8	FIFO_LVL	FIFO level. The bits indicate the amount of valid data in FIFO.	R
7	BUSY	SPDIF busy bit. 0: SPDIF is not working 1: SPDIF is working	R
6:2	Reserved	Writing has no effect, read as zero.	R
1	F_TRIG	Trigger flag. 0: Not active 1: Active	R
0	F_FFUR	FIFO underrun flag. 0: Not active 1: Active	RW

11.2.16 SPDIF Configure 1 Register (SPCFG1)

The register SPCFG1 is used to configure SPDIF.

SPCFG1 0x1002008C

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														INIT_LVL	ZRO_VLD	Reserved		TRIG		SRC_NUM				CH1_NUM				CH2_NUMI			
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
31:18	Reserved	Writing has no effect, read as zero.	R
17	INIT_LVL	Initial level set bit. 0: SPDIF initial level is low 1: SPDIF initial level is high	
16	ZRO_VLD	The valid bit of channel state is 0 or 1 when play ZERO sample under FIFO underflow. 0: Valid 1: Invalid	RW
15:14	Reserved	Writing has no effect, read as zero.	R
13:12	TRIG	Specify the trigger value of FIFO.	RW
11:8	SRC_NUM	Source number. 0000:Unspecified 0001~1111:1~15	RW
7:4	CH1_NUM	Channel 1 number. 0000:Unspecified 0001~1111:A~O	RW
3:0	CH2_NUM	Channel 2 number. 0000:Unspecified 0001~1111:A~O	RW

TRIG	Description
00	Trigger Value is 4.
01	Trigger Value is 8.
10	Trigger Value is 16.
11	Trigger Value is 32.

11.2.17 SPDIF Configure 2 Register (SPCFG2)

The register SPCFG2 is used to configure SPDIF.

SPCFG2 0x10020090

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		FS				ORG_FRQ				SAMPL_WL			MAX_WL	CLK_ACU		CAT_CODE								CH_MD		Reserved		PRE	COPY_N	AUDIO_N	CON_PRO
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	Writing has no effect, read as zero.	R
29:26	FS	Sampling frequency. 0000:44.1kHz 0010:48kHz 0011:32kHz 1010:96kHz 1110:192kHz Others: Reference IEC60958-3	RW
25:22	ORG_FRQ	Original sampling frequency. 1111:44.1kHz 1101:48kHz 1100:32kHz 0101:96kHz 0001:192kHz Others: Reference IEC60958-3	RW
21:19	SAMPL_WL	Sample word length. When MAX_WL=1: 001:20 bit 110:21 bit 010:22 bit 100:23 bit 101:24 bit Others: reserved When MAX_WL=0: 001:16 bit 110:17 bit 010:18 bit 100:19 bit 101:20 bit Others: reserved	RW

18	MAX_WL	Maximum audio sample word length. 0:20 bit； 1:24 bit.	RW
17:16	CLK_ACU	Clock Accuracy of transmitted clock. 00: Level II 01: Level I 10: Level III 11: Interface frame rate not matched to sampling frequency	RW
15:8	CAT_CODE	Category code. Reference IEC60958-3 for full details. 00 indicates “general” mode.	RW
7:6	CH_MD	Channel mode choose bit. 00: Mode 0 01~11: Reserved	RW
5:4	Reserved	Writing has no effect, read as zero.	R
3	PRE	Pre-emphasis set bit. 0: None 1: 15us/15us	RW
2	COPY_N	Copyright set bit. 0: Copyright is asserted 1: Copyright is not asserted	RW
1	AUDIO_N	Linear PCM identification bit. 0: Audio sample word represents linear PCM samples 1: Audio sample word used for other purpose	RW
0	CON_PRO	Consumer mode and professional mode choose bit. 0: Consumer mode 1: Professional mode Professional is not supported in the chip.	RW

11.2.18 SPDIF FIFO Register (SPFIFO)

The register SPCFG1 is used to configure SPDIF.

SPFIFO **0x10020094**

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	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
31:24	Reserved	Writing has no effect, read as zero.	R
23:0	DATA	FIFO port. When write to it, data is push to the transmit FIFO. Read from it as 0.	W

11.3 Serial Interface Protocol

11.3.1 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.

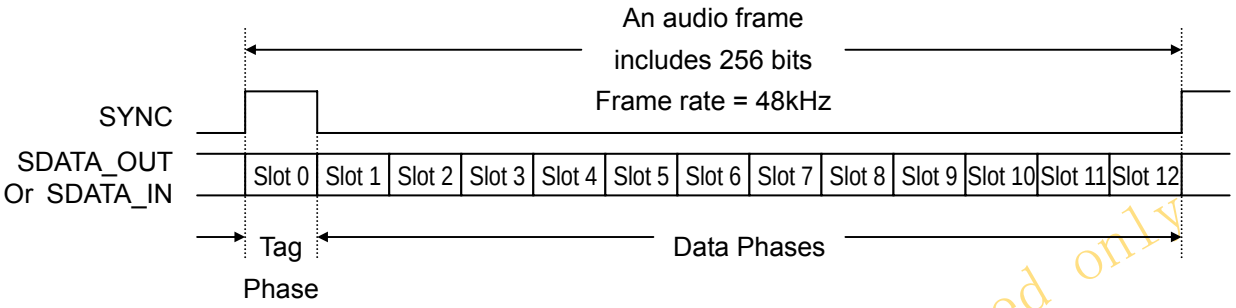


Figure 11-4 AC-link audio frame format

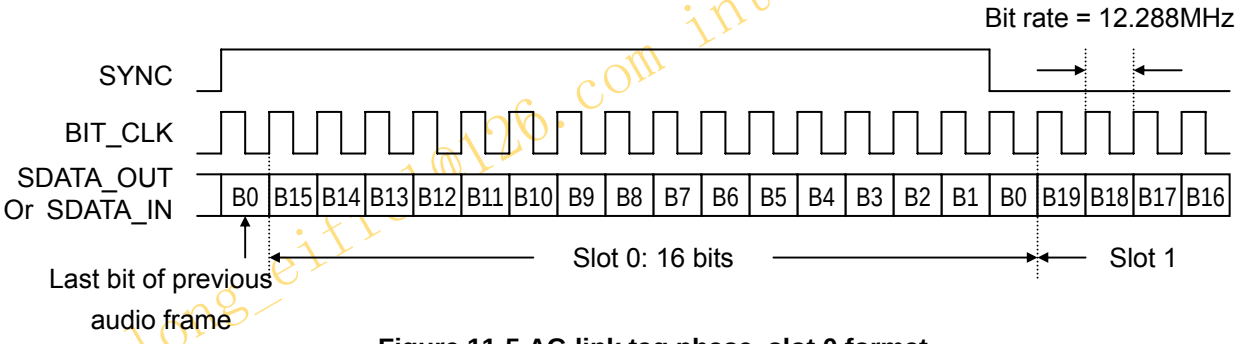


Figure 11-5 AC-link tag phase, slot 0 format

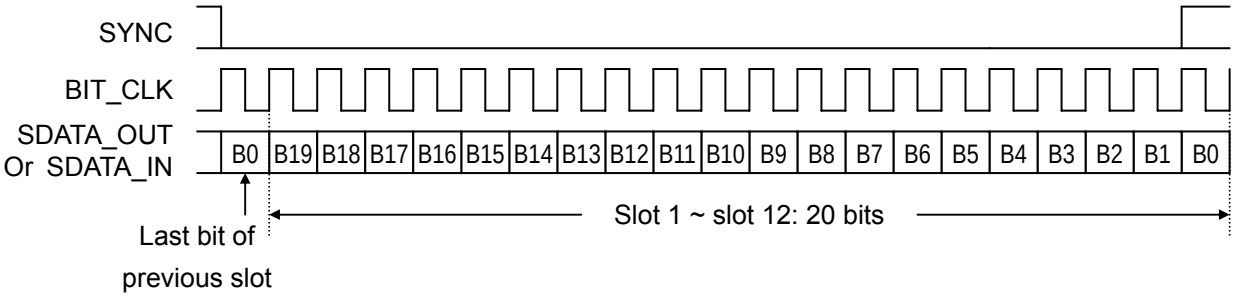


Figure 11-6 AC-link data phases, slot 1 ~ slot 12 format

11.3.2 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.

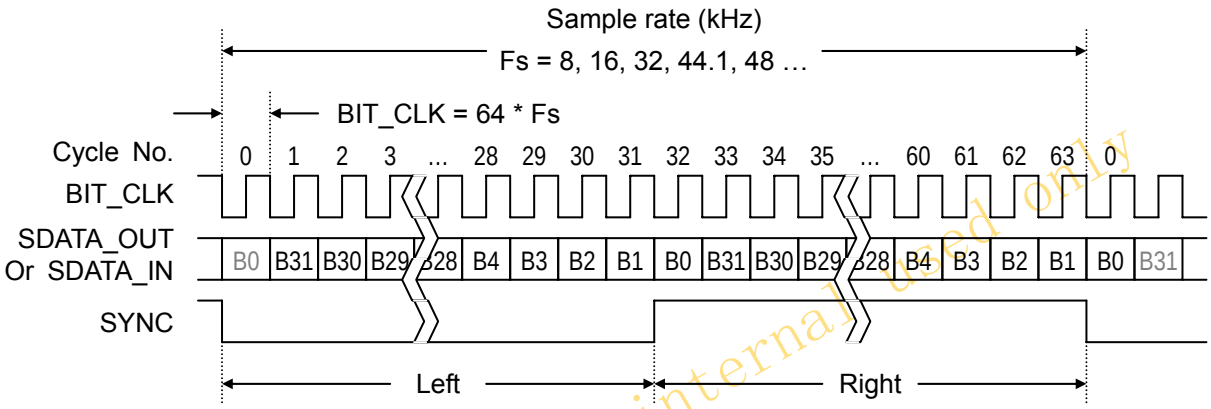


Figure 11-7 I2S data format (A: LR mode)

In the A: LR mode, first send the left channel in a stereo frame. One Left slot and one Right slot make a sample frame. It is the normal mode of I2S.

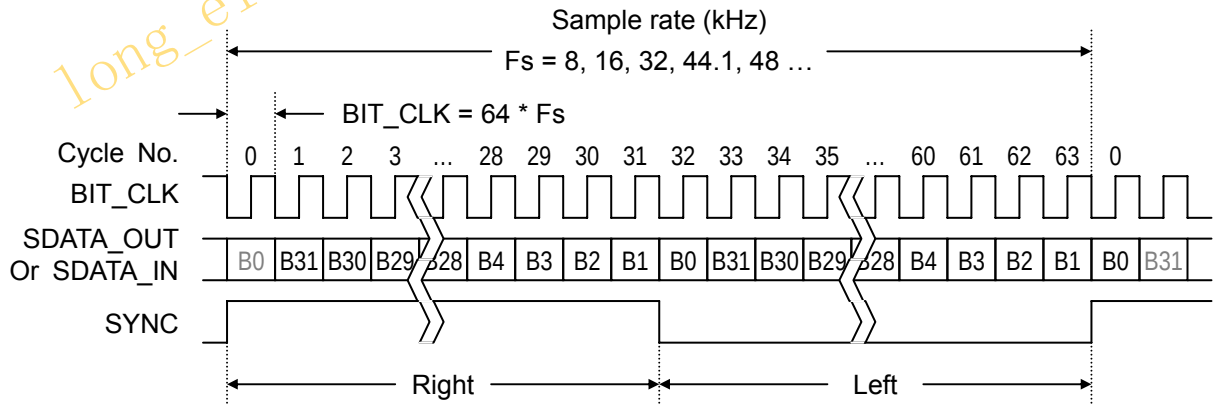


Figure 11-8 I2S data format (B: RL mode)

In the B: RL mode, first send the right channel in a stereo frame. One Right slot and one Left slot make a sample frame. It is used in same CODEC.

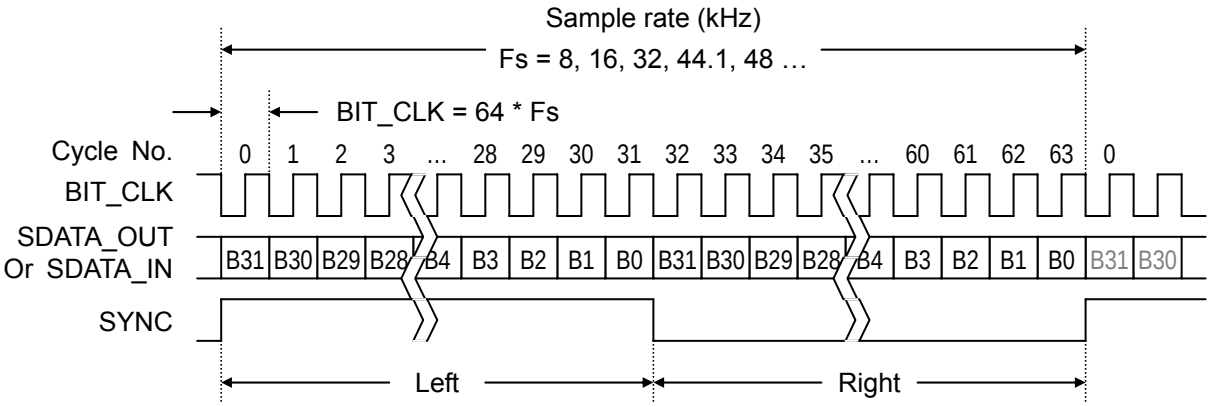


Figure 11-9 MSB-justified data format (C: LR mode)

In the C: LR mode, first send the left channel in a stereo frame. One Left slot and one Right slot make a sample frame. It is the normal mode in MSB-justified.

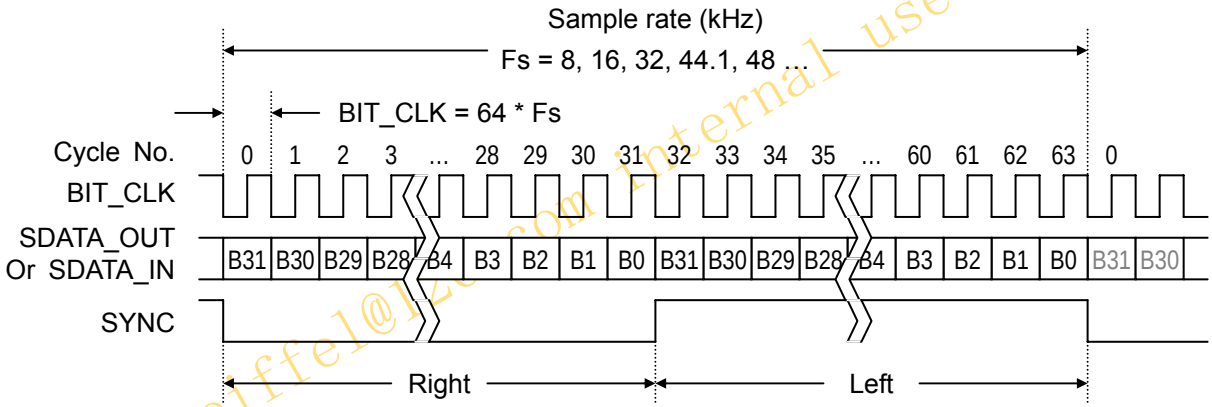


Figure 11-10 MSB-justified data format (D: RL mode)

In the D: RL mode, first send the right channel in a stereo frame. One Right slot and one Left slot make a sample frame.

Figure 11-7 and Figure 11-9 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(If BIT_CLK is generated internally). 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 also with O_BIT_CLK and O_SYNC signals are provided by the internal CODEC from the SYSCLK, which is enabled by I2SCR.ESCLK and configured to 12MHz clock using CPM.

11.3.3 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 11-3 described the how sample data bits are transferred.

Table 11-3 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

If in 16 bits packed mode, the data transferred is the same as the 16 bits normal mode as shown above. But there are two samples in one word.

11.3.4 SPDIF Protocol

SPDIF block format is shown below:

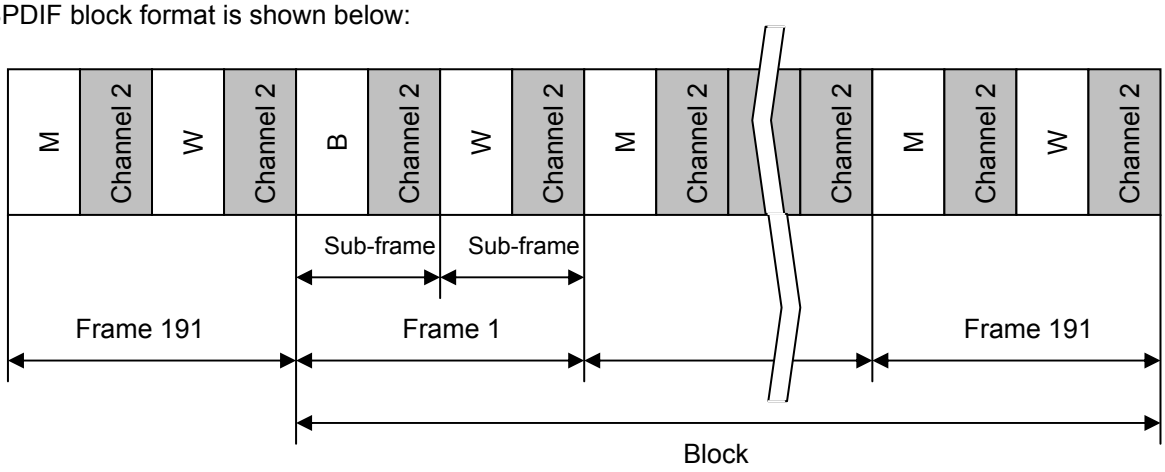


Figure 11-11 Block format

Sub-frame format in PCM mode is shown below:

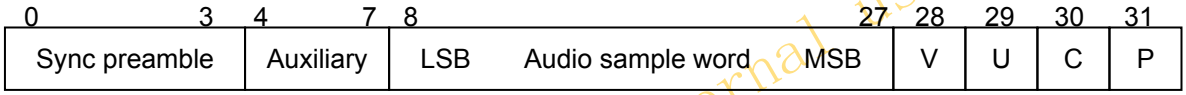


Figure 11-12 Sub-frame format in PCM mode

Sub-frame format in non-PCM mode is shown below:

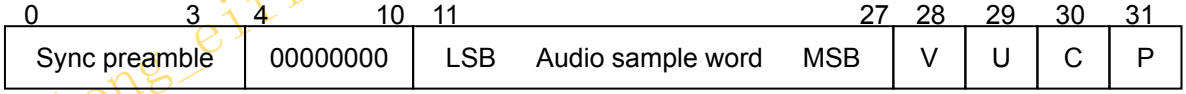


Figure 11-13 Sub-frame format in non-PCM mode

11.4 AC97/I2S Operation

The AIC can be accessed either by the processor using programmed I/O instructions or by the DMA controller. The 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 11.2.1 for further details regarding AICFR.TFTH.
- The receive FIFO request is based on the receive trigger-threshold (AICFR.RFTH) setting. See 11.2.1 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.

11.4.1 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).

4 Decide BIT_CLK direction (AICFR.BCKD) and SYNC direction (AICFR.SYNCD).
5 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 11-7. 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).
6 Enable AIC by write 1 to AICFR.ENB.
7 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.
8 In AC-link format, issue a warm or cold CODEC reset.
9 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 skipped.
10 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.

11.4.2 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).

11.4.3 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.

11.4.3.1 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.

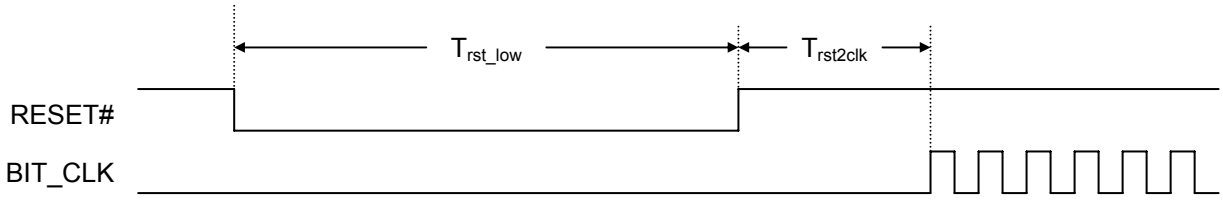


Figure 11-14 Cold AC '97 CODEC Reset Timing

Table 11-4 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

11.4.3.2 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.

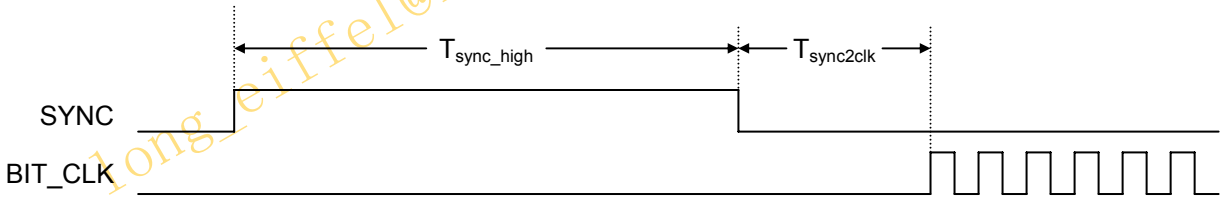


Figure 11-15 Warm AC '97 CODEC Reset Timing

Table 11-5 Warm AC '97 CODEC Reset Timing Parameters

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

11.4.4 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 by using 3 GPIO pins of the chip. 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.

11.4.5 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 the CODEC as needed.
2 Configure sample size by AICCR.OSS.
3 Configure sample channels (AICCR.CHANNEL).
4 If sample size is configured 16 bit, select packed or unpacked mode (AICCR.PACK16).
5 If two channels is configured, select the right-channel-first sample data or not (I2SCR.RFIRST).
6 If two channels is configured, select the sample data switched or not (I2SCR.SWLH).
7 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 accessing CODEC internal registers (for internal CODEC).
8 For AC-link, configure replay channels by ACCR1.XS.
9 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.
10 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).
But when using internal CODEC, write first frame of PCM data to transmit FIFO till TX FIFO under-run (AICSR.TUR is set to 1), otherwise left/right channel may be switched.
11 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 11.4.7 for detail description on FIFO.
12 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.
13 Write 0 to AICCR.ERPL.

NOTE: Before replaying Open ADC BITCLK and close it to generating Record internal circuit reset when using internal CODEC.

11.4.6 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 the 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 11.4.7 for detail description on FIFO. For AC-link, ACCR1.RS tells which channels are recorded.
When using internal CODEC, the first data should be ignored.
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.

11.4.7 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 RX FIFO is in 24 bits width and 32 entries depth, one entry for keeping one audio sample regardless of the sample size. The RX FIFO is in 32 bits width and 64 entries depth, one entry for keeping one audio sample regardless of the sample size, but in 16 bits packed mode, one entry for keeping two audio samples. 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 11-16 illustrates the FIFOs access.

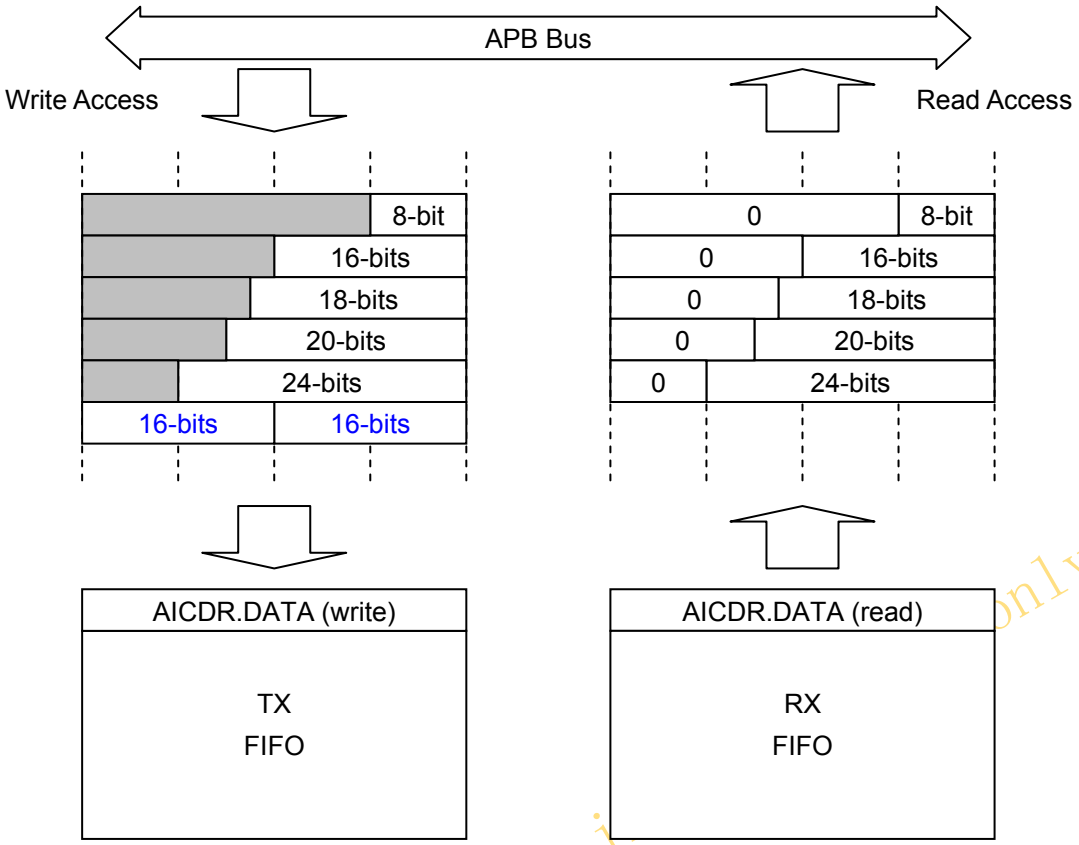


Figure 11-16 Transmitting/Receiving FIFO access via APB Bus

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, so 32-bits width DMA must be used. One 16 bits sample occupies one 16-bits half word in memory, so 16-bits width DMA must be used. One 8-bits sample occupies one byte in memory, and use 8-bits width DMA except 16bits packed mode. If in 16 bits packed mode, Two 16 bits sample occupies one 32-bits word in memory, so 32-bits width DMA must be used.

In case of processor bus initiator, any type of the audio sample must occupy one CPU general-purpose register at LSB, and read/write from/to AICDR.DATA with 32-bits load/store instruction. When process small sample size, 16-bits or 8-bits, software may need to do the data pack/unpack except 16 bits packed mode. In the 16bits packed mode, the sample data is packed, and two 16 bits audio samples occupy one CPU general-purpose register.

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 value; too small or too big are not good. When AICFR.RFTH is too small, or AICFR.TFTH is too big, 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 AICFR.RFTH is too big or AICFR.TFTH is too small, 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 default 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. In 16bits packed mode, could configure that the left channel is the first data or the right channel. By default, the 16 bits LSB is left channel, 16 bits MSB is the right channel. But it also could be switched the Left or the Right channel (I2SCR.SWLH).

11.4.8 Data Flow Control

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

11.4.8.1 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.

11.4.8.2 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.

11.4.8.3 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 the processor's DMA controller 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.

11.4.9 Audio Samples format

11.4.9.1 16 bits packed mode

One channel (mono) mode and two channels (stereo) mode:

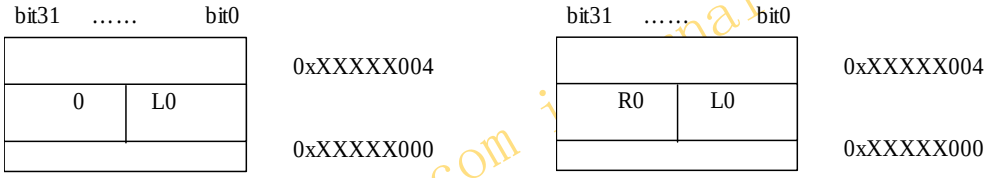


Figure 11-17 One channel (Left) and Two channels (right) mode (16 bits packed mode)

Four channels mode and six channels mode:

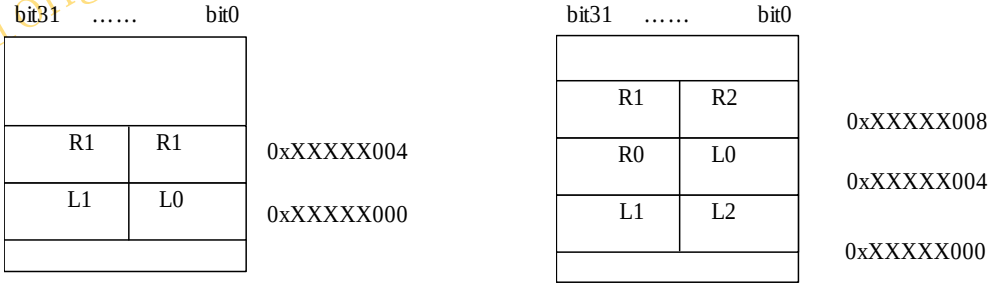


Figure 11-18 Four channels (Left) and Six channels (right) mode (16 bits packed mode)

Eight channels mode:

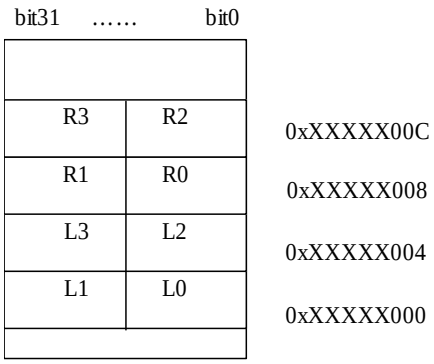


Figure 11-19 Eight channels mode (16 bits packed mode)

11.4.9.2 Normal mode.

One channel (Mono) and two channels (stereo) mode:

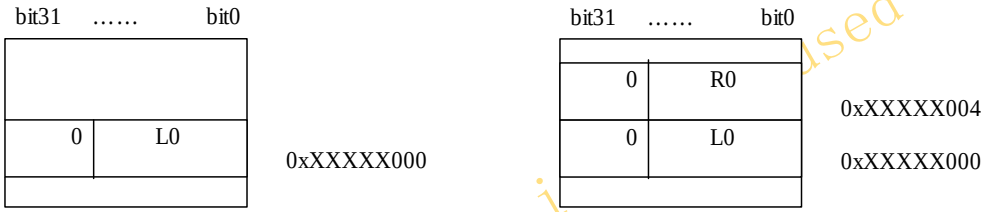


Figure 11-20 One channel (Left) and Two channels (right) mode

Four channels mode and six channels mode:

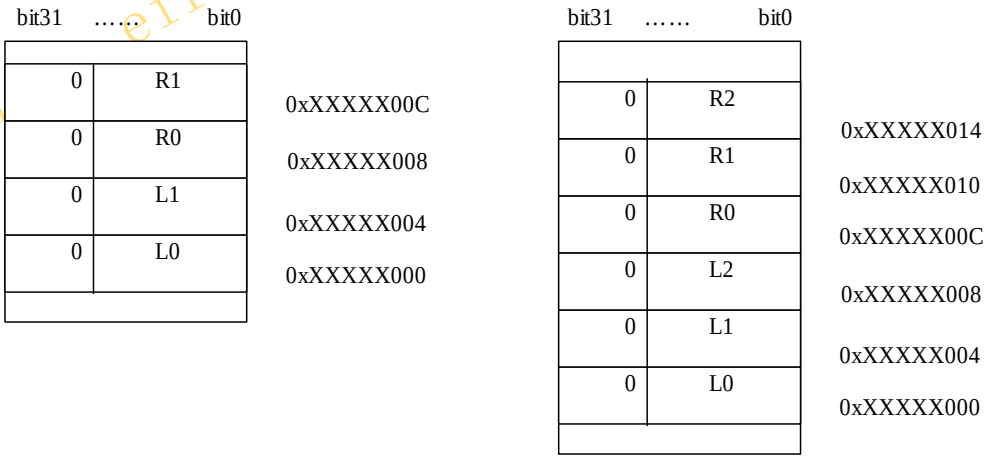


Figure 11-21 Four channels (Left) and Six channels (right) mode

Eight channel mode:

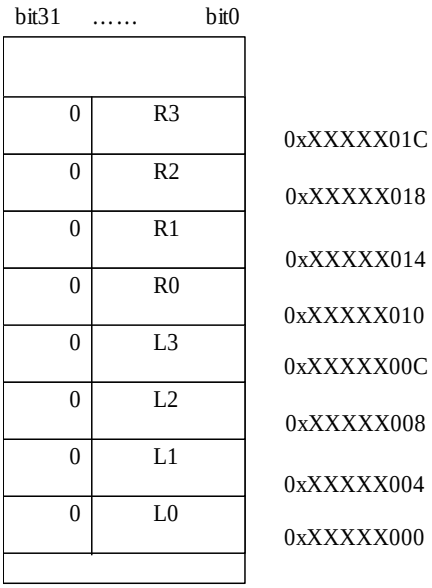


Figure 11-22 Eight channels mode

11.4.10 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 called SYS_CLK and provides I_BITCLK, O_BITCLK and I_SYNC, O_SYNC (left-right clock which is the sample rate as ADC or DAC) to AIC for outgoing and incoming audio respectively. These clocks change when change the sample rate in CODEC controlling registers. When using internal CODEC, must configure SYNC and BIT_CLK as input, more details refers to CODEC Spec.

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 out 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, $f_{BIT\_CLK} = 64\ f_{S.}$ But SYNC signal frequency is not fixed when using internal CODEC.

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 11-3 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 11-6 lists the relation between sample rate, BIT_CLK and SYS_CLK frequencies.

Table 11-6 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 this processor, 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.

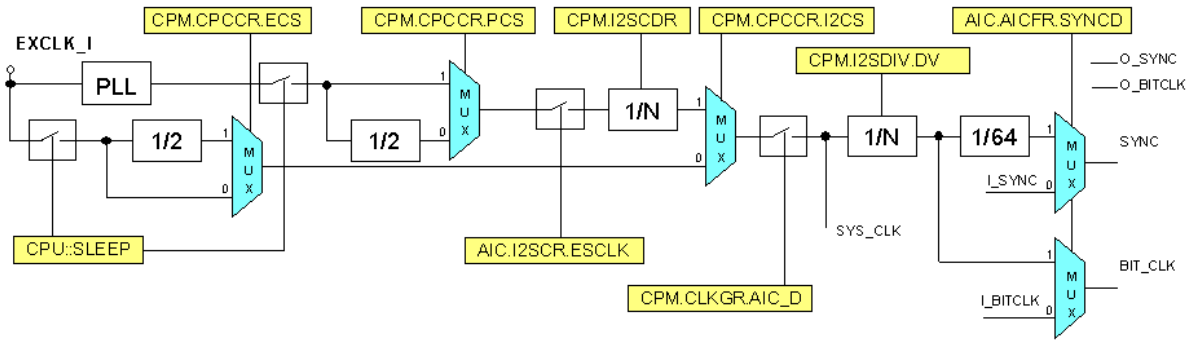


Figure 11-23 SYS_CLK, BIT_CLK and SYNC generation scheme

The setting of I2SDIV.DV is shown in Table 11-7.

Table 11-7 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 11-6, 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 11-8 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 11-8 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 11-9 lists the divider settings for various sample rates.

Table 11-9 CPM/AIC clock divider setting for various sampling rate if PLL = 270.64MHz

Sample Rate (kHz)	I2SCDR	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 11-8 as possible. Table 11-10 lists the PLL parameters and audio sample errors at different PLL frequencies for EXCLK at 12MHz.

Table 11-10 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.

11.4.11 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.

11.5 SPDIF Guide

11.5.1 Set SPDIF clock frequency

Set SPDIF clock frequency is as same as i2s clock.

11.5.2 PCM audio mode operation (Reference IEC60958)

1 Set SPCFG1 and SPCFG2 to configure SPDIF transmitter.
   a Set SPCFG2.CON_PRO to 0 to choose consumer mode.
   b Set SPCFG2. AUDIO_N to 0 to choose linear PCM audio data mode.
   c Set SPCFG1.XXX to configure SPDIF.
   d Set SPCFG2.XXX to configure SPDIF.
2 Set SPCTRL.DMA_EN to choose DMA mode or CPU mode.
3 Set SPCTRL.SIGN_N to choose whether to transfer the most significant bit by toggle or not.
4 Set SPCTRL. SFT_RST to 1 reset FIFO.
5 Wait SPCTRL. SFT_RST set to be set 0 by hardware.
6 Set SPCTRL.M_TRIG and SPCTRL.M_FFUR to enable or disable the interrupt.
7 Set SPCTRL.INVALID 1 or 0 to set the V bit of sub-frame.
8 Set SPENA.SPEN to 1 to Enable SPDIF to transmitter.

11.5.3 Non-PCM mode operation (Reference IEC61937)

1 Set SPCFG1 and SPCFG2 to configure SPDIF transmitter.
   a Set SPCFG2.CON_PRO to 0 to choose consumer mode.
   b Set SPCFG2.AUDIO_N to 1 to choose non-PCM mode.
   c Set SPCFG1.SRC_NUM to 0.
   d Set SPCFG1.CH1_NUM to 0.
   e Set SPCFG1.CH2_NUM to 0.
   f Set SPCFG2.PRE to 0.
   g Set SPCFG2.CH_MD to 0.
   h Set SPCFG2.ORG_FRQ to 0.
   i Set SPCFG2.SAMPL_WL to 0.
   j Set SPCFG2.MAX_WL to 0.
   k Set SPCFG1.XXX to configure SPDIF.
   l Set SPCFG2.XXX to configure SPDIF.
2 Set SPCTRL.DMA_EN to choose DMA mode or CPU mode.
3 Set SPCTRL.SIGN_N to choose whether to transfer the most significant bit by toggle or not.
4 Set SPCTRL. SFT_RST to 1 reset FIFO.
5 Wait SPCTRL. SFT_RST to be set to 0 by hardware.
6 Set SPCTRL.M_TRIG and SPCTRL.M_FFUR to enable or disable the interrupt.
7 Set SPCTRL.INVALID 1 or 0 to set the V bit of sub-frame.
8 Set SPENA.SPEN to 1 to Enable SPDIF to transmitter.

11.5.4 Disable operation

1 Set SPENA.SPEN to 0 to disable SPDIF to transmitter.
2 Wait SPSTATE.BUSY to be set to 0 by hardware.
3 You can do other operation.

12 PCM Interface

12.1 Overview

The PCM has the following features:

- Data starts with the frame PCMSYN or one PCMCLK later
- Support three modes of operation for PCM
  - Short frame sync mode
  - Long frame sync mode
  - Multi-slot mode
- Data is transferred and received with the MSB first
- Support master mode and slave mode
- The PCM serial output data, PCMDOUT, is clocked out using the rising edge of the PCMSCLK
- The PCM serial input data, PCMDIN, is clocked in on the falling edge of the PCMSCLK
- 8/16 bit sample data sizes supported
- DMA transfer mode supported
- Two FIFOs for transmit and receive respectively with 16 samples capacity in every direction
- Two independent PCM interface. As PCM0,PCM1

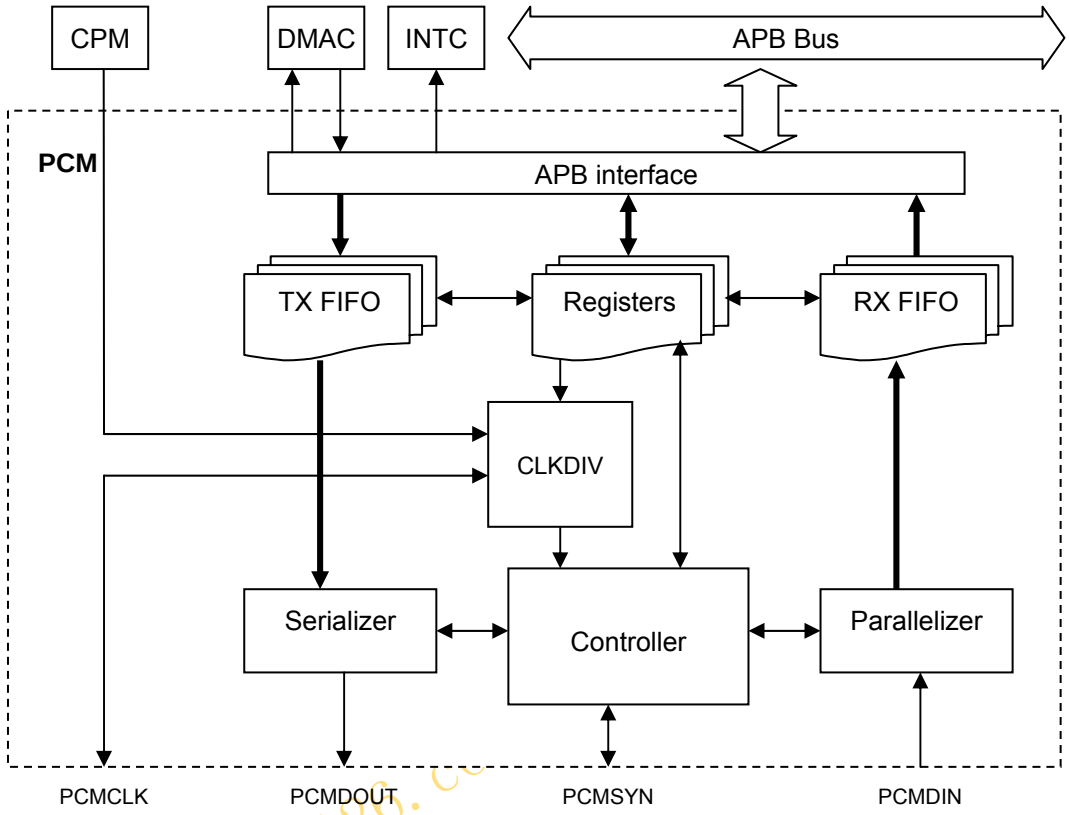
12.2 Pin Description

There are all 4 pins used to connect between PCM interface and an external device. They are listed and described in Table 12-1.

Table 12-1 PCM Interface Pins Description

Name	I/O	Description
PCMCLK	Input/Output	PCM Serial clock Line signal input/output
PCMSYN	Input/Output	PCM sync signal input/output
PCMDOUT	Output	PCM Serial data output
PCMDIN	Input	PCM Serial data input

12.3 Block Diagram



12.4 Register Description

Table 12-2 PCM0 Registers Description

Name	Description	RW	Reset Value	Address	Size
PCMCTL0	PCM Control Register	RW	0x00000000	0x10071000	32
PCMCFG0	PCM Configure Register	RW	0x00000110	0x10071004	32
PCMDP0	PCM FIFO Data Port Register	RW	0x00000000	0x10071008	32
PCMINTC0	PCM Interrupt Control Register	RW	0x00000000	0x1007100c	32
PCMINTS0	PCM Interrupt Status Register	RW	0x00000100	0x10071010	32
PCMDIV0	PCM Clock Divide Register	RW	0x00000001	0x10071014	32

Table 12-3 PCM1 Registers Description

Name	Description	RW	Reset Value	Address	Size
PCMCTL1	PCM Control Register	RW	0x00000000	0x10074000	32
PCMCFG1	PCM Configure Register	RW	0x00000110	0x10074004	32
PCMDP1	PCM FIFO Data Port Register	RW	0x00000000	0x10074008	32
PCMINTC1	PCM Interrupt Control Register	RW	0x00000000	0x1007400c	32
PCMINTS1	PCM Interrupt Status Register	RW	0x00000100	0x10074010	32
PCMDIV1	PCM Clock Divide Register	RW	0x00000001	0x10074014	32

12.4.1 PCM Control Register (PCMCTL)

PCMCTL0, PCMCTL1 **0x10071000, 0x10074000**

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																						ERDMA	ETDMA	LSMP	ERPL	EREC	FLUSH	RST	SYNEN	CLKEN	PCMEN
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
31:10	Reserved	Writing has no effect, read as zero.	R
9	ERDMA	Receive DMA Enable. This bit is used to enable or disable the DMA during receiving audio data. ERDMA: 0 = Disabled. 1 = Enabled.	RW
8	ETDMA	Transmit DMA Enable. This bit is used to enable or disable the DMA during transmit audio data.	RW

			ETDMA: Transmit DMA 0: Disabled. 1: Enabled.	
7	LSMP	Select between play last sample or play ZERO sample in TX FIFO underflow. ZERO sample means sample value is zero. LSMP: CODEC used 0: Play ZERO sample when TX FIFO underflow. 1: Play last sample when TX FIFO underflow.		RW
6	ERPL	Enable Playing Back function. This bit is used to disable or enable the audio sample data transmitting. ERPL: Description 0: PCM Playing Back Function is Disabled. 1: PCM Playing Back Function is Enabled.		RW
5	EREC	Enable Recording Function. This bit is used to disable or enable the audio sample data receiving. EREC: Description 0: PCM Recording Function is Disabled. 1: PCM Recording Function is Enabled.		RW
4	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
3	RST	Reset PCM. Write 1 to this bit reset PCM registers and FIFOs. Writing 0 to this bit has no effect and this bit is always reading 0.		W
2	Reserved	Writing has no effect, read as zero.		R
1	CLKEN	Enable the serial clock division logic. Must be HIGH for the PCM to operate.		RW
0	PCMEN	Enable PCM function. This bit is used to enable or disable the PCM function. PCMENB: Description 0: Disable PCM Controller. 1: Enable PCM Controller.		RW

12.4.2 PCM Configuration Register (PCMCFG)

PCMCFG0, PCMCFG1 **0x10071004, 0x10074004**

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																	SLOT	ISS	OSS	IMSBPOS	OMSBPOS	RFTH		TFTH							PCMMOD
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0

<table>
<tr><th>Bits</th><th>Name</th><th>Description</th><th>RW</th></tr>
<tr><td>31:15</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>14:13</td><td>SLOT</td><td>Controls the amount of valid PCM timeslot in one PCMSYN frame.</td><td>RW</td></tr>
<tr><td>12</td><td>ISS</td><td>Input Sample Size. These bits reflect input sample data size to memory or register. The data sizes supported are: 8/16bits. The sample data is LSB-justified in memory/register.
<table>
<tr><th>ISS</th><th>Sample Size</th></tr>
<tr><td>0</td><td>8 bit</td></tr>
<tr><td>1</td><td>16 bit</td></tr>
</table></td><td>RW</td></tr>
<tr><td>11</td><td>OSS</td><td>Output Sample Size. These bits reflect output sample data size from memory or register. The data sizes supported are: 8/16 bits. The sample data is LSB-justified in memory/register.
<table>
<tr><th>OSS</th><th>Sample Size</th></tr>
<tr><td>0</td><td>8 bit</td></tr>
<tr><td>1</td><td>16 bit</td></tr>
</table></td><td>RW</td></tr>
<tr><td>10</td><td>IMSBPOS</td><td>Controls the position of the MSB bit in the serial input stream relative to the PCMSYN signal.
0: MSB is captured on the falling edge of PCMCLK during the same cycle that PCMSYNC is high
1: MSB is captured on the falling edge of PCMCLK during the cycle after the PCMSYNC is high</td><td>RW</td></tr>
<tr><td>9</td><td>OMSBPOS</td><td>Controls the position of the MSB bit in the serial output stream relative to the PCMSYN signal.
0: MSB sent during the same clock that PCMSYN is high
1: MSB sent on the next PCMSCLK cycle after PCMSYNC is high</td><td>RW</td></tr>
<tr><td>8:5</td><td>RFTH</td><td>Receive FIFO threshold for interrupt or DMA request. Determines when the RFS flags go active for the RXFIFO. When the sample number in receive FIFO, indicated by PCMINTS.RFL, is great than the threshold value, PCMINTS.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.</td><td>RW</td></tr>
<tr><td>4:1</td><td>TFTH</td><td>Transmit FIFO threshold for interrupt or DMA request. Determines when the TFS flags go active for the TXFIFO. When the sample number in transmit FIFO, indicated by PCMINTS.TFL, is less than the threshold value, PCMINTS.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.</td><td>RW</td></tr>
<tr><td>0</td><td>PCMMOD</td><td>PCM mode select.
0: Master mode; 1: Slave mode.</td><td>RW</td></tr>
</table>

12.4.3 PCM FIFO DATA PORT REGISTER (PCMDP)

PCMDP0, PCMDP1 **0x10071008, 0x10074008**

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
	DATA																															
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
31:0	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

12.4.4 PCM INTERRUPT CONTROL REGISTER (PCMINTC)

PCMINTCR0, PCMINTCR1 **0x1007100c, 0x1007400c**

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																												ETFS	ETUR	ERFS	EROR
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
31:4	Reserved	Writing has no effect, read as zero.	R
3	ETFS	Enable TFS Interrupt. This bit is used to control the TFS interrupt enable or disable. ETFS / TFS Interrupt: 0 – Disabled.; 1 – Enabled.	RW
2	ETUR	Enable TUR Interrupt. This bit is used to control the TUR interrupt enable or disable. ETUR / TUR Interrupt: 0 – Disabled.; 1 – Enabled.	RW
1	ERFS	Enable RFS Interrupt. This bit is used to control the RFS interrupt enable or disable. ERFS / RFS Interrupt: 0 – Disabled.; 1 – Enabled.	RW
0	EROR	Enable ROR Interrupt. This bit is used to control the ROR interrupt	RW

		enable or disable. EROR: 0 — ROR Interrupt: Disabled. EROR: 1 — ROR Interrupt: Enabled.	

12.4.5 PCM INTERRUPT STATUS REGISTER (PCMINTS)

PCMINTS0, PCMINTS1 **0x10071010, 0x10074010**

Bit	31:15	14	13:9	8	7	6:2	1	0
	Reserved	RSTS	TFL	TFS	TUR	RFL	RFS	ROR
RST	0	0	0	1	0	0	0	0

Bits	Name	Description	RW
31:15	Reserved	Writing has no effect, read as zero.	R
14	RSTS	Soft reset / flush state. 0: Nothing / reset or flush operation has completed 1: reset or flush operation has not completed	R
13:9	TFL	Transmit FIFO Level. The bits indicate the amount of valid PCM data in Transmit FIFO.	R
8	TFS	Transmit FIFO Service Request. This bit indicates that transmit FIFO level exceeds TFL threshold which is controlled by PCMCFG.TFTH When TFS is 1, it may trigger interrupt or DMA request depends on the interrupt enable and DMA setting. TFS: 0 — Description: Transmit FIFO level exceeds TFL threshold. TFS: 1 — Description: Transmit FIFO level at or below TFL threshold.	RW
7	TUR	Transmit FIFO Under Run. This bit indicates that transmit FIFO has or has not experienced an under-run. TUR: 0 — When read, indicates under-run has not been found. TUR: 0 — When write, clear itself. TUR: 1 — When read, indicates data has even been read from empty transmit FIFO. TUR: 1 — When write, not effects.	RW
6:2	RFL	Receive FIFO Level. The bits indicate the amount of valid PCM data in Receive FIFO.	R

<table>
<tr><td>1</td><td>RFS</td><td>Receive FIFO Service Request. This bit indicates that receive FIFO level is or not below RFL threshold which is controlled by PCMCFG.RFTH. When RFS is 1, it may trigger interrupt or DMA request depends on the interrupt enable and DMA setting.
<table>
<tr><th>RFS</th><th>Description</th></tr>
<tr><td>0</td><td>Receive FIFO level below RFL threshold.</td></tr>
<tr><td>1</td><td>Receive FIFO level at or above RFL threshold.</td></tr>
</table></td><td>RW</td></tr>
<tr><td>0</td><td>ROR</td><td>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></td><td>RW</td></tr>
</table>

12.4.6 PCM CLOCK DIVIDE REGISTER (PCMDIV)

PCMDIV0, PCMDIV1 **0x10071014, 0x10074014**

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															SYNL						SYNDIV					CLKDIV					
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
31:27	Reserved	Writing has no effect, read as zero.	R
16:11	SYNL	Controls the length that the PCMSYN based upon the PCMCLK. The length of PCMSYN = (SYNL + 1) * PCMCLK cycle.	RW
10:6	SYNDIV	Controls the frequency of the PCMSYN signal based upon the PCMCLK. PCMSYN = PCMCLK / 8 (SYNDIV + 1).	RW
5:0	CLKDIV	PCMCLK clock divider value minus 1. Controls the divider used to create the PCMCLK based upon the CPM_PCM_SYSCLK. PCMCLK = CPM_PCM_SYSCLK / (CLKDIV + 1).	RW

12.5 PCM Interface Timing

The following figures show the timing relationship for the PCM transfers, Note in all cases. In master mode, the PCMCLK is derived from dividing the input clock, CPM_PCM_SYSCLK, and the PCMSYN is divided depended on the PCMCLK. In slave mode, the PCMCLK and PCMSYN are input from the external device. Data is sampled on the falling edge of the PCMCLK and sent out on the rising edge of the PCMCLK. The PCMSYN signal determines when the next data sample is to be transferred between the controller and the external device. Also, the PCMSYN signal as seen in the figure can be one bit time or a long bit time controlled by PCMDIV.SYNL. The PCMSYN frequency controlled by PCMDIV.SYNDIV is usually the sample rate. There are some variations controlled by PCMCFG.ISS, PCMCFG.OSS and PCMCFG.SLOT to accommodate 8 / 16bit sample sizes and multi-slot transmission.

12.5.1 Short Frame SYN

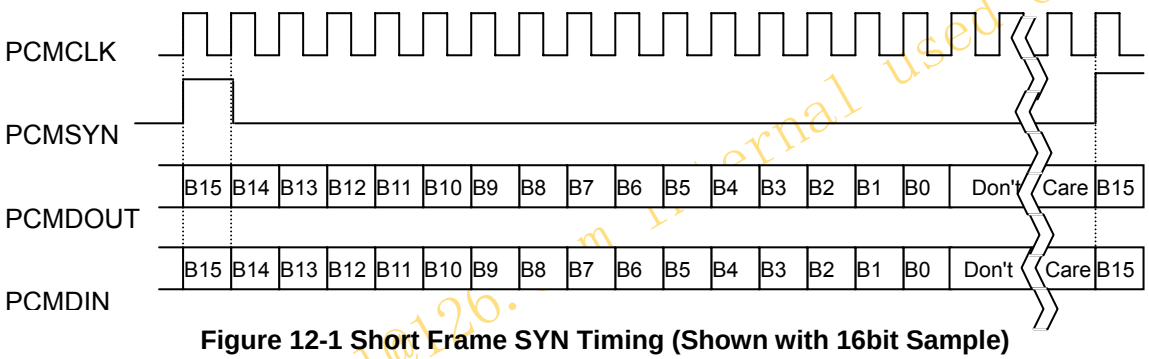


Figure 12-1 Short Frame SYN Timing (Shown with 16bit Sample)

NOTE: Figure 12-1 shows a PCM transfer with the MSB configured to be coincident with the PCMSYN.

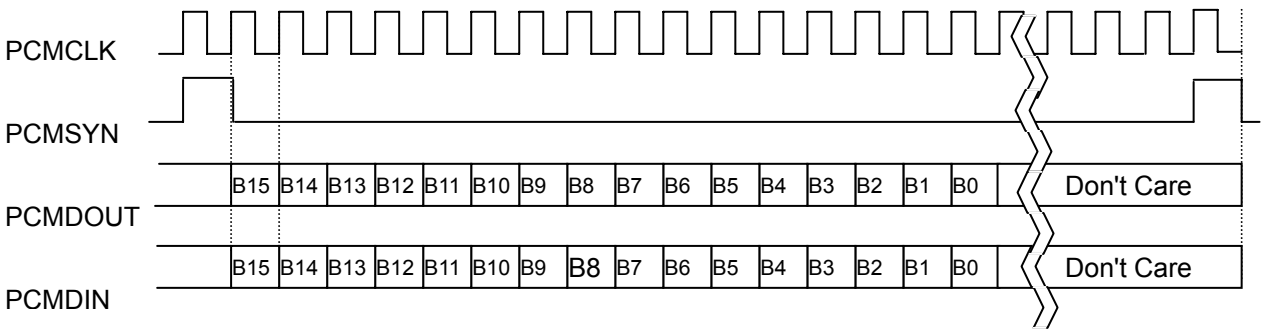


Figure 12-2 Short Frame SYN Timing (Shown with 16bit Sample)

NOTE: Figure 12-2 shows a PCM transfer with the MSB configured one shift clock after the PCMSYN.

12.5.2 Long Frame SYN

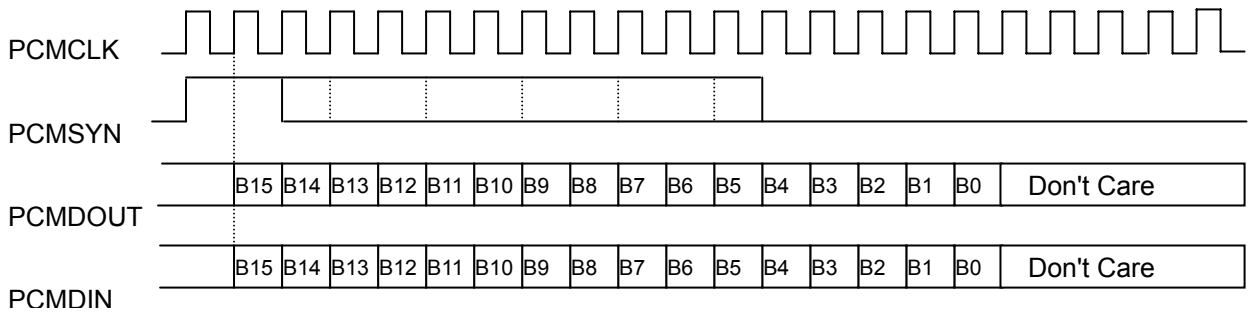


Figure 12-3 Long Frame SYN Timing (Shown with 16bit Sample)

NOTE: Figure 12-3 shows a PCM transfer with the MSB configured one shift clock after the PCMSYN.

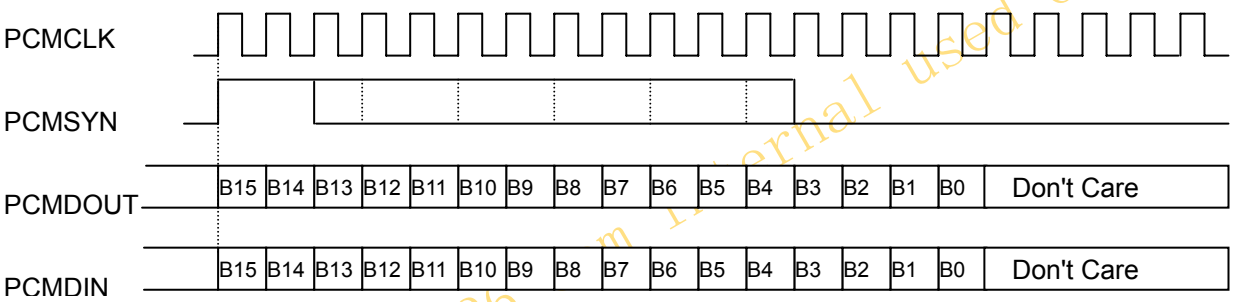


Figure 12-4 Long Frame SYN Timing (Shown with 16bit Sample)

NOTE: Figure 12-4 shows a PCM transfer with the MSB configured to be coincident with the PCMSYN.

12.5.3 Multi-Slot Operation

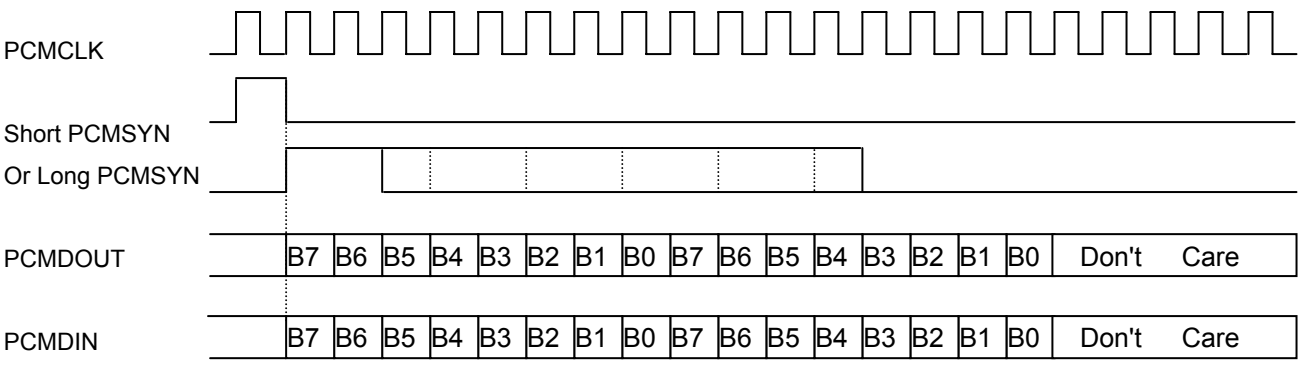
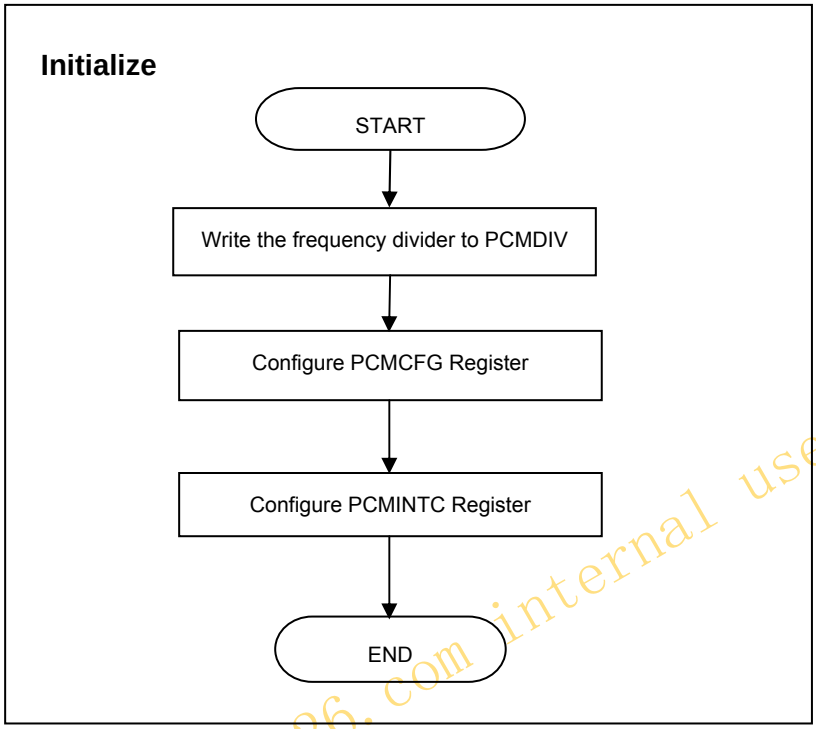


Figure 12-5 Multi-Slot Frame SYN Timing (Shown with two Slots and 8bit Sample)

12.6 PCM Operation

12.6.1 PCM Initialization

At power-on or other hardware reset (WDT and etc), PCM is disabled. Software must initiate PCM after power-on or reset.



For further details, see the corresponding register description sections.

12.6.2 Audio Replay

Outgoing audio sample data is written to PCM transmit FIFO from processor via store instruction or from memory via DMA. PCM then takes the data from the FIFO, serializes it, and sends it over the serial wire PCMDOUT to an external DEVICE.

The audio transmission is enabled automatically when the PCM is enabled by set PCMCTL.PCMEN. And PCMCTL.ERPL must be 1. If PCMCTL.ERPL is 0, value of zero is sent to external DEVICE even if there are samples in transmit FIFO. 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 external DEVICE even if PCMCTL.ERPL is 1.

Here is the audio replay flow:

1 Configure the external DEVICE as needed.
2 Initialize PCM and configure the register.
3 Write 1 to PCMCTL.PCMEN and PCMCTL.CLKEN.
4 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, PCMINTS.TUR is 1, and if PCMINTC.ETUR is 1, PCM issues an interrupt. Please reference to 12.6.4 for detail description on FIFO.

5 Write 1 to PCMCTL.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 (PCMINTS.TUR).

6 Waiting for PCMINTS.TFL change to 0. So that all samples in the transmit FIFO has been replayed, then we can have a clean start and write 0 to PCMCTL.ERPL.

12.6.3 Audio Record

Incoming audio sample data is received from PCMDIN serially and converted to parallel word and stored in PCM 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 PCM is enabled by set PCMCTL.PCMEN, And PCMCTL.EREC must be 1. If PCMCTL.EREC is 0, the received data is discarded even if there are rooms in the receive FIFO. At least one room left in the receive FIFO. If the receive FIFO is full, the received data is discarded even if PCMCTL.EREC is 1.

Here is the audio record flow:

1 Configure the external DEVICE as needed.
   a Initialize PCM and configure the register.
   b Write 1 to PCMCTL.PCMEN and PCMCTL.CLKEN.

2 Write 1 to PCMCTL.EREC. Make sure there are rooms available in the receive FIFO before set PCMCTL.EREC. Usually, it should empty the receive FIFO by fetch data from it before set PCMCTL.EREC.

3 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, PCMINTS.ROR is 1, and if PCMINTC.EROR is 1, PCM issues an interrupt. Please reference to 12.6.4 for detail description on FIFO.

4 Write 0 to AICCR.EREC.

5 Take sample data from the receive FIFO until PCMINTS.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.

12.6.4 FIFOs operation

PCM has two FIFOs, one for transmitting and one for receiving. The FIFOs are in 16 bits width and 16 entries depth, one entry for keep one sample regardless of the sample size. PCMDP.DATA provides the access point for processor/DMA to write to transmit FIFO and read from receive FIFO. One time access to PCMDP.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 6 illustrates the FIFOs access.

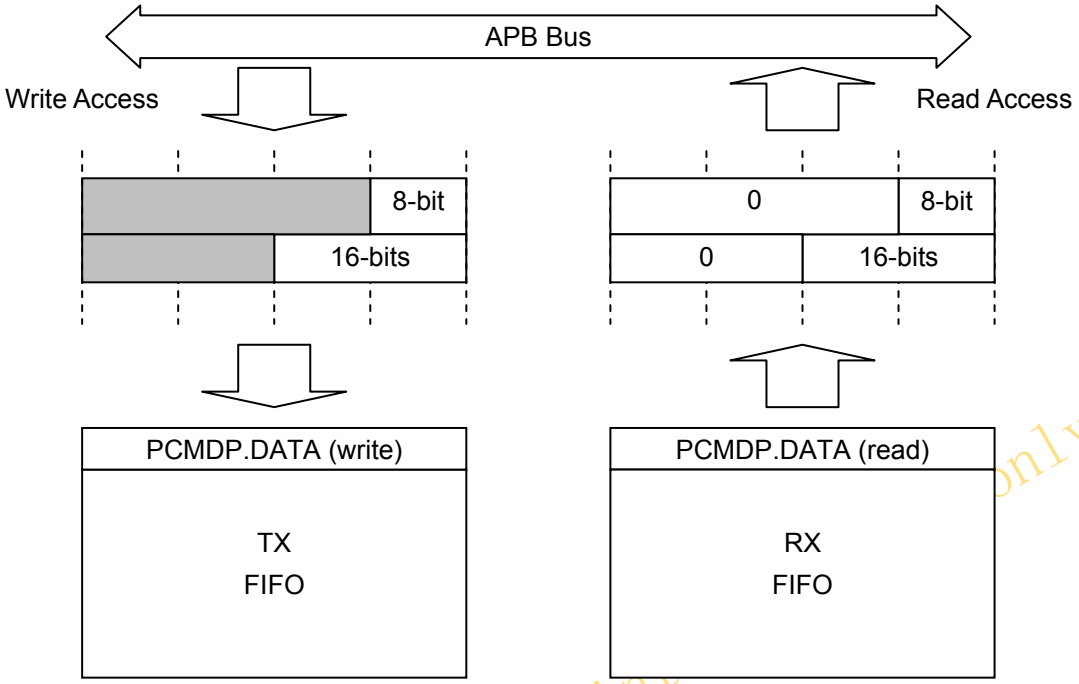


Figure 12-6 Transmitting/Receiving FIFO access via APB Bus

The software and bus initiator must guarantee the right sample placement at the bus.
In case of DMA bus initiator, One 16 bits sample occupies one 16-bits half word in memory, so 16-bits width DMA must be used. One 8-bits sample occupies one byte in memory, and use 8-bits width DMA.

12.6.5 Data Flow Control

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

12.6.5.1 Polling and Processor Access

PCMINTS.RFL and PCMINTS.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 and then load them from the RxFIFO, and when there are rooms in transmitting FIFO and then store samples to the TxFIFO.

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

12.6.5.2 Interrupt and Processor Access

Set proper values to PCMCFG.TFTH and PCMCFG.RFTH, the FIFO interrupts trig thresholds. Set PCMINTC.ETFS and/or PCMINTC.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.

12.6.5.3 DMA Access

To enable DMA operation, set PCMCTL.ERDMA and PCMCTL.ETDMA 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 DMAC spec for the details.

The PCMCFG.TFTH and PCMCFG.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.

12.6.6 PCM Serial Clocks and Sampling Frequencies

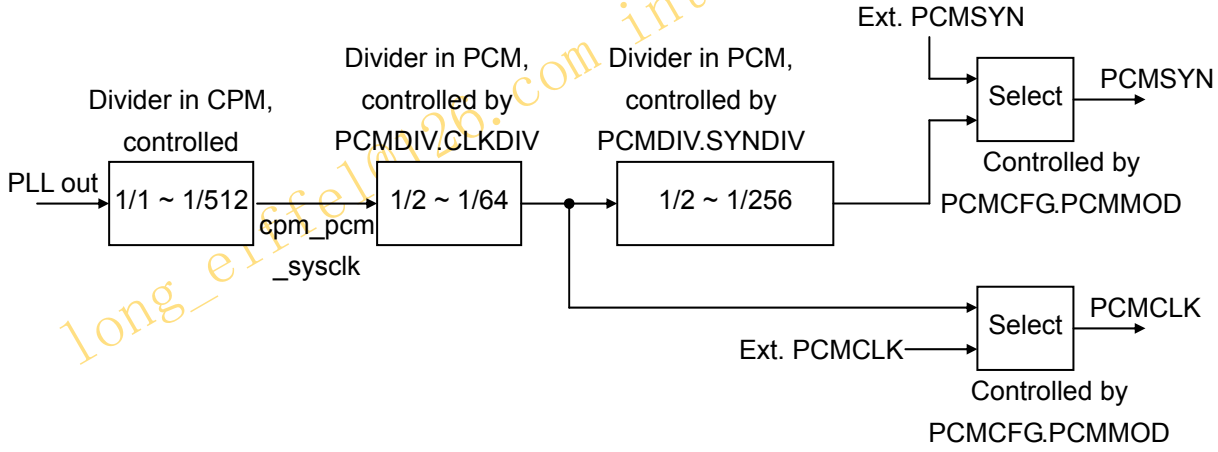


Figure 12-7 PCMCLK and PCMSYN generation scheme

12.6.7 Interrupts

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

- Receive FIFO Service (PCMINTS.RFS). It's also DMA Request.
- Transmit FIFO Service (PCMINTS.TFS). It's also DMA Request.
- Transmit Under-Run (PCMINTS.TUR).
- Receive Over-Run (PCMINTS.ROR).

For further details, see the corresponding register description sections.

13 SAR A/D Controller

13.1 Overview

The A/D in falcon is CMOS low-power dissipation 12bit touch screen SAR analog to digital converter. It operates with 3.3/1.2V power supply. It is developed as an embedded high resolution ADC targeting to the 65nm CMOS process and has wide application in portable electronic devices, high-end home entertainment center, communication systems and so on.

The SAR A/D controller is dedicated to control A/D to work at three different modes: Touch Screen (measure pen position and pen down pressure), Battery (check the battery power), and two auxiliary input. Touch Screen can transfer the data to memory though the DMA or CPU. Battery and two auxiliary input can transfer the data to memory though CPU.

Features:

- 7 Channels
- Resolution: 12-bit
- Integral nonlinearity: ±1 LSB
- Differential nonlinearity: ±0.5 LSB
- Resolution/speed: up to 2Msps
- Max Frequency: 200k
- Low power dissipation: 1.5mW(worst)
- Support 4-wire and 5-wire touch panel measurement (Through pin XP, XN, YP, YN and AUX2)
- Support multi-touch detect
- Support write control command by software
- Support voltage measurement (Through pin VBAT)
- Support two auxiliary input (Through pin AUX1, AUX2)
- Single-end and Differential Conversion Mode
- Auto X/Y, X/Y/Z1/Z2 and X/Y/Z1/Z2/X2/Y2 position measurement
- Support external touch screen controller
- Pin Description

Table 13-1 SADC Pin Description

Name	I/O	Description
XN	AI	Touch screen analog differential X- position input
YN	AI	Touch screen analog differential Y- position input
XP	AI	Touch screen analog differential X+ position input
YP	AI	Touch screen analog differential Y+ position input
VBAT	AI	VBAT direct input *[1]
AUX1	AI	Auxiliary Input

AUX2	AI	Auxiliary Input

NOTE:

*[1]: Users who already deployed resistor networks on board level can use VBAT to direct measure the battery value.

13.2 Register Description

In this section, we will describe the registers in SAR A/D controller. Following table lists all the register definitions. All registers' 32bit addresses are physical addresses. And detailed function of each register will be described below.

Table 13-2 SADC Register Description

Name	Description	RW	Reset Value	Address	Access Size
ADENA	ADC Enable Register	RW	0x80	0x10070000	8
ADCFG	ADC Configure Register	RW	0x00040000	0x10070004	32
ADCTRL	ADC Control Register	RW	0x3F	0x10070008	8
ADSTATE	ADC Status Register	RW	0x00	0x1007000C	8
ADSAME	ADC Same Point Time Register	RW	0x0000	0x10070010	16
ADWAIT	ADC Wait Time Register	RW	0x0000	0x10070014	16
ADTCH	ADC Touch Screen Data Register	RW	0x00000000	0x10070018	32
ADVDAT	ADC VBAT Data Register	RW	0x0000	0x1007001C	16
ADADAT	ADC AUX Data Register	RW	0x0000	0x10070020	16
ADCLK	ADC Clock Divide Register	RW	0x00041000	0x10070028	32
ADCMD	ADC Command Register	RW	0x00000000	0x10070024	32

13.2.1 ADC Enable Register (ADENA)

The register ADENA is used to trigger A/D to work.

ADENA 0x10070000

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																								POWER	SLP_MD	Reserved		PENDEN	TCHEN	VBATEN	AUXEN
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	1	0	0	0	1	0	0	0

Bits	Name	Description	RW
31:8	Reserved	Writing has no effect, read as zero.	R
7	POWER	SADC Power control bit. 1: SADC power down 0: SADC power on When POWER is set from 1 to 0, you should wait at least 2ms to enable SADC.	RW
6	SLP_MD	SLEEP Mode Control. 1:Enter sleep mode 0:Exit sleep mode	RW

5:4	Reserved	Writing has no effect, read as zero.	R
3	PENDEN	Pen Down Detect Enable control. 0: disable 1: enable	RW
2	TCHEN	Touch Screen Enable Control. 0: disable 1: enable	RW
1	VBATEN	VBAT Enable Control. No matter TCHEN is 1 or 0, VBATEN can be set to 1 to sample the voltage of battery, and when the value of voltage is ready, PBATEN will be cleared by hardware auto.	RW
0	AUXEN	AUX n Enable Control. No matter TCHEN is 1 or 0, AUXEN can be set to 1 to sample the voltage of AUX1, AU2 or AUX3, and when the value of voltage is ready. AUXEN will be cleared by hardware auto.	RW

NOTES:

1 TCHEN, VBATEN and AUXEN can be set to 1 at the same time. The priority of the three mode is AUX > VBAT > TCH.

2 SLP_MD, TCHEN can be set to 1 at the same time. The priority of the two mode is SLP_MD > TCH.

3 When VBATEN and AUXEN are all 0, SLP_MD can be set to 1.

13.2.2 ADC Configure Register (ADCFG)

The register ADCFG is used to configure the A/D.

ADCFG **0x10007004**

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
	SPZZ	Reserved							Wire_cel_r	Cmd_sel_r	PRU						DMA_EN	XYZ	SNUM				Reserved								CMD	
RST	0	0	0	0	0	0	0	0	0	0	0	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	SPZZ[*1]	The $X_dY_dZ_mZ_n$ of different point measure can be different. But the $X_dY_dZ_mZ_n$ of the same point measure can be same or different. 0: The $X_dY_dZ_mZ_n$ of the same point measure is all the same (X_dY_dZ1Z2, X_dY_dZ1Z2, X_dY_dZ1Z2, X_dY_dZ1Z2 … X_dY_dZ1Z2) 1: The $X_dY_dZ_mZ_n$ of the same point measure maybe different (X_dY_dZ1Z2, X_dY_dZ3Z4, X_dY_dZ3Z4, X_dY_dZ1Z2 … X_dY_dZ1Z2)	RW
30:24	Reserved	Writing has no effect, read as zero.	R

<table>
<tr><td>23</td><td>WIRE_SEL</td><td>0: use 4-wire touch panel
1: use 5-wire touch panel</td><td>RW</td></tr>
<tr><td>22</td><td>CMD_SEL</td><td>0: use hardware inter command to control touch panel
1: use software write command to control touch panel</td><td>RW</td></tr>
<tr><td>21:16</td><td>RPU</td><td>Internal Pull-up resistor for Pen Detection.
6’b111111: 64kΩ/63 = 1.02kΩ (least sensitive)
6’b111110: 64KΩ/62 = 1.03KΩ
… (pull-up = 64kΩ / binary value of RPU)
6’b000010: 64KΩ/2 = 32KΩ
6’b000001: 64kΩ/1 = 64kΩ (most sensitive) default
6’b000000: RESERVED (do not use this setting)</td><td>RW</td></tr>
<tr><td>15</td><td>DMA_EN</td><td>When A/D is used as Touch Screen , DMA_EN is used as follows:
0: The sample data is read by CPU
1: The sample data is read by DMA</td><td>RW</td></tr>
<tr><td>14:13</td><td>XYZ</td><td>When A/D is used in Touch Screen mode, XYZ is used as follows:
<table>
<tr><th>XYZ</th><th>Measure</th></tr>
<tr><td>00</td><td>$X_s \rightarrow Y_s$</td></tr>
<tr><td>01</td><td>$X_d \rightarrow Y_d$</td></tr>
<tr><td>10</td><td>$X_d \rightarrow Y_d \rightarrow Z1_d \rightarrow Z2_d$ or $X_d \rightarrow Y_d \rightarrow Z3_d \rightarrow Z4_d$</td></tr>
<tr><td>11</td><td>$X_d \rightarrow Y_d \rightarrow Z1_d \rightarrow Z2_d \rightarrow X2 \rightarrow Y2$ or
$X_d \rightarrow Y_d \rightarrow Z3_d \rightarrow Z4_d \rightarrow X2 \rightarrow Y2$</td></tr>
</table></td><td>RW</td></tr>
<tr><td>12:10</td><td>SNUM</td><td>The number of repeated sampling one point. When A/D is used as Touch Screen, SNUM is used as follows:
<table>
<tr><th>SNUM</th><th>Number</th></tr>
<tr><td>000</td><td>Reserved</td></tr>
<tr><td>001</td><td>1</td></tr>
<tr><td>010</td><td>2</td></tr>
<tr><td>011</td><td>3</td></tr>
<tr><td>100</td><td>4</td></tr>
<tr><td>101</td><td>5</td></tr>
<tr><td>110</td><td>Reserved</td></tr>
<tr><td>111</td><td>Reserved</td></tr>
</table></td><td>RW</td></tr>
<tr><td>9:2</td><td>Reserved</td><td>Writing has no effect, read as zero.</td><td>R</td></tr>
<tr><td>1:0</td><td>CMD</td><td>CMD is used to choose the current sample command when ADENA.AUXEN is set to 1.
<table>
<tr><th>CMD</th><th>Function</th></tr>
<tr><td>00</td><td>Reserved</td></tr>
<tr><td>01</td><td>Measure AUX1 voltage</td></tr>
<tr><td>10</td><td>Measure AUX2 voltage</td></tr>
<tr><td>11</td><td>Reserved</td></tr>
</table></td><td>RW</td></tr>
</table>

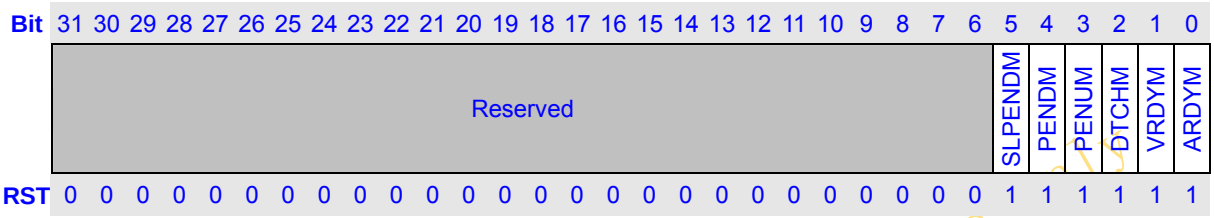
NOTE:

*1: X_s, Y_s means the reference mode of X, Y is single-end mode.
X_d, Y_d, $Z1_d$, $Z2_d$, $Z3_d$, Z4d means the reference mode of X, Y, Z1, Z2, Z3, Z4 is differential mode.
When you measure Xs you need to make sure that X-plate is driven by external DC power.
When you measure Ys you need to make sure that Y-plate is driven by external DC power.

13.2.3 ADC Control Register (ADCTRL)

The register ADCTRL is used to control A/D to work.

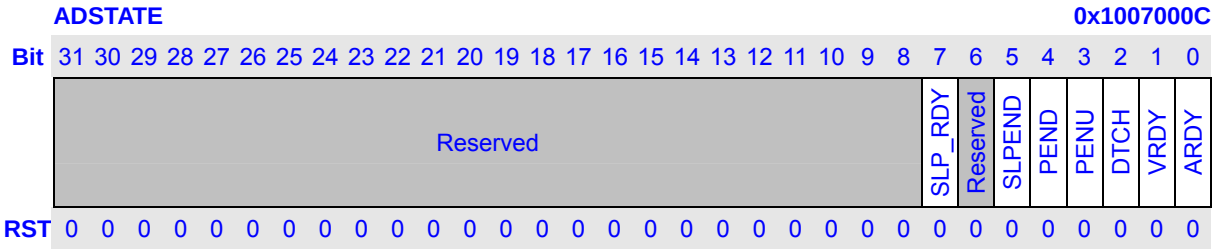
ADCTRL 0x10070008



Bits	Name	Description	RW
31:6	Reserved	Writing has no effect, read as zero.	R
5	SLPENDM	In SLEEP mode pen down interrupt mask. 0: enabled 1: masked	RW
4	PENDM	Pen down interrupt mask. 0: enabled 1: masked	RW
3	PENUM	Pen up interrupt mask. 0: enabled 1: masked	RW
2	DTCHM	Touch Screen Data Ready interrupt mask. 0: enabled 1: masked	RW
1	VRDYM	VBAT data ready interrupt mask. 0: enabled 1: masked	RW
0	ARDYM	AUX data ready interrupt mask. 0: enabled 1: masked	RW

13.2.4 ADC Status Register (ADSTATE)

The register ADSTATE is used to keep the status of A/D.



Bits	Name	Description	RW
31:8	Reserved	Writing has no effect, read as zero.	R
7	SLP_RDY	Sleep state bit. 1:The set of sleep mode is ready 0:The set of sleep mode is not ready	R
6	Reserved	Writing has no effect, read as zero.	R
5	SLPEND	In SLEEP mode pen down interrupt flag. Write 1 to this bit, the bit will clear this bit. 1: active 0: not active	RW
4	PEND	Pen down interrupt flag. Write 1 to this bit, the bit will clear this bit. 1: active 0: not active	RW
3	PENU	Pen up interrupt flag. Write 1 to this bit, the bit will clear this bit. 1: active 0: not active	RW
2	DTCH	Touch screen data ready interrupt flag. Write 1 to this bit, the bit will clear this bit. 1: active 0: not active	RW
1	VRDY	VBAT data ready interrupt flag. Write 1 to this bit, the bit will clear this bit. 1: active 0: not active	RW
0	ARDY	AUX data ready interrupt flag. Write 1 to this bit, the bit will clear this bit. 1: active 0: not active	RW

13.2.5 ADC Same Point Time Register (ADSAME)

The register ADSAME is used to store the interval time between repeated sampling the same point. The clock of the counter is clk_us.

ADSAME **0x10070010**

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																SCNT															
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

13.2.6 ADC Wait Pen Down Time Register (ADWAIT)

The register ADWAIT is used to store the interval time of wait pen down. And the register can be used as the interval time among the different point. The clock of the counter is clk_us.

ADWAIT **0x10070014**

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																WCNT															
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

13.2.7 ADC Touch Screen Data Register (ADTCH)

The read-only ADTCH is corresponded to 16x32 bit FIFO, it keep the sample data for touch screen. 0~11 bits are data, 15 bit is data type. 16~27 bits are data, 31 bit is data type. When write to the register, DATA will be clear to 0.

ADTCH **0x10070018**

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
	TYPE1	Reserved															TYPE0	Reserved														
									TDATA1																TDATA0							
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	TYPE1	Type of the Touch Screen Data1. When A/D is used as Touch Screen, ADCFG.XYZ=10 or XYZ=11: TYPE1=1: X_d→Y_d→Z1→Z2 or X_d→Y_d→Z1→Z2→X2→Y2; TYPE1=0: X_d→Y_d→Z3→Z4 or X_d→Y_d→Z3→Z4→X2→Y2. When A/D is used as Touch Screen, ADCFG.XYZ=00 or XYZ=01: TYPE1=0.	RW
30:28	Reserved	Writing has no effect, read as zero.	R
27:16	TDATA1	The concert data of touch screen A/D.	RW
15	TYPE0	Type of the Touch Screen Data2. When A/D is used as Touch Screen, ADCFG.XYZ=10 or XYZ=11: TYPE0=1: X_d→Y_d→Z1→Z2 or X_d→Y_d→Z1→Z2→X2→Y2; TYPE0=0: X_d→Y_d→Z3→Z4 or X_d→Y_d→Z3→Z4→X2→Y2. When A/D is used as Touch Screen, ADCFG.XYZ=00 or XYZ=01: TYPE0=0.	RW
14:12	Reserved	Writing has no effect, read as zero.	R
11:0	TDATA0	The concert data of touch screen A/D.	RW

NOTES:

1 When A/D is used as Touch Screen, ADCFG.XYZ=00.
The format of touch screen data is as follows:

Type1	Reserved	Data1	Type0	Reserved	Data0
0	000	Y_s	0	000	X_s

2 When A/D is used as Touch Screen, ADCFG.XYZ=01.
The format of touch screen data is as follows:

Type1	Reserved	Data1	Type0	Reserved	Data0
0	000	Y_d	0	000	X_d

3 When A/D is used as Touch Screen, ADCFG.XYZ=10.TYPE=1.
The format of touch screen data is as follows:

Type1	Reserved	Data1	Type0	Reserved	Data0
1	000	Y_d	1	000	X_d
1	000	$Z2_d$	1	000	$Z1_d$

Users need to read twice to get the whole data. The first time reading gets the data Yd and Xd. The second time reading gets the data Z2d and Z1d.

The touch pressure measurement formula is as follows: (You can use formula 1 or formula 2.)

$$R_{TOUCH} = R_{X\text{-}Plate} \bullet \frac{X\text{-}Position}{4096}\left(\frac{Z_2}{Z_1} - 1\right) \quad (1)^{*1}$$

$$R_{TOUCH} = \frac{R_{X\text{-}Plate} \bullet X\text{-}Position}{4096}\left(\frac{4096}{Z_1} - 1\right) - R_{Y\text{-}Plate} \bullet \left(1 - \frac{Y\text{-}Position}{4096}\right) \quad (2)^{*1}$$

4 When A/D is used as Touch Screen, ADCFG.XYZ=10.TYPE=0.
The format of touch screen data is as follows:

Type1	Reserved	Data1	Type0	Reserved	Data0
0	000	Y_d	0	000	X_d
0	000	$Z4_d$	0	000	$Z3_d$

Users need to read twice to get the whole data. The first time reading gets the data Y_d and X_d. The second time reading gets the data $Z4_d$ and $Z3_d$.

The touch pressure measurement formula is as follows: (You can use formula 3 or formula 4.)

$$R_{TOUCH} = R_{Y\text{-}Plate} \bullet \frac{Y\text{-}Position}{4096}\left(\frac{Z_4}{Z_3} - 1\right) \quad (3)^{*1}$$

$$R_{TOUCH} = \frac{R_{Y\text{-}Plate} \bullet Y\text{-}Position}{4096}\left(\frac{4096}{Z_3} - 1\right) - R_{X\text{-}Plate} \bullet \left(1 - \frac{X\text{-}Position}{4096}\right) \quad (4)^{*1}$$

5 When A/D is used as Touch Screen, ADCFG.XYZ=11.TYPE=1.
The format of touch screen data is as follows:

Type1	Reserved	Data1	Type0	Reserved	Data0
1	000	Y_d	1	000	X_d
1	000	$Z2_d$	1	000	$Z1_d$
1	000	Y2	1	000	X2

Users need to read thrice to get the whole data. The first time reading gets the data Y_d and X_d. The second time reading gets the data $Z2_d$ and $Z1_d$. The third time reading gets the data Y2 and X2.

6 When A/D is used as Touch Screen, ADCFG.XYZ=11.TYPE=0.
The format of touch screen data is as follows:

Type1	Reserved	Data1	Type0	Reserved	Data0
0	000	Y_d	0	000	X_d
0	000	$Z4_d$	0	000	$Z3_d$
0	000	Y2	0	000	X2

Users need to read thrice to get the whole data. The first time reading gets the data Y_d and X_d. The second time reading gets the data $Z4_d$ and $Z3_d$. The third time reading gets the data Y2 and X2.

NOTE:

*1: To determine pen or finger touch, the pressure of the touch needs to be determined. Generally, it is not necessary to have very high performance for this test; therefore, the 8-bit resolution mode is recommended (however, calculations will be shown here are in 12bit resolution mode).

$R_{X\text{-plate}}$: Total X-axis resistor value (about 200Ω~ 600Ω)

$R_{Y\text{-plate}}$: Total Y-axis resistor value (about 200Ω~ 600Ω)

X-Position: X-axis voltage sample value

Y-Position: Y-axis voltage sample value

Z1, Z2: Z1, Z2 voltage sample value

Z3, Z4: Z3, Z4 voltage sample value

X2, Y2: X2, Y2 voltage sample value

13.2.8 ADC VBAT Data Register (ADVDAT)

The read-only ADVDAT is a 16-bit register, it keep the sample data of VBAT. 0~11 bits are data.

ADBDAT 0x1007001C

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																				VDATA											
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
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	VDATA	Data of VBAT A/D convert. When write to the register, DATA will be clear to 0.	RW

The measured voltage V_{BAT} is as follows:

$$V_{BAT} = \frac{VDATA}{4096} \bullet 1.2V$$

13.2.9 ADC AUX Data Register (ADADAT)

The read-only ADADAT is a 16-bit register, it keep the sample data. 0~11 bits are data.

ADSDAT 0x10070020

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																ADATA															
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
15:12	Reserved	Writing has no effect, read as zero.	R
11:0	ADATA	Data of AUX. When write to the register, DATA will be clear to 0.	RW

The measured voltage V_{AUX} (V_{AUX1} and V_{AUX2}) is as follows:

$$V_{SAD} = \frac{ADATA}{4096} \bullet AVDD33$$

13.2.10 ADC Clock Divide Register (ADCLK)

The register ADCLK is used to set the A/D's clock dividing number.

ADCLK 0x10070028

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
	CLKDIV_MS																CLKDIV_US								CLKDIV							
RST	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Bits	Name	Description	RW
31:16	CLKDIV_MS	Dividing number to get ms clock from ADC clock. ms_clk = us_clk / (CLK_MS +1)	RW
15:8	CLKDIV_US	Dividing number to get us clock from ADC clock. us_clk = adc_clk / (CLKDIV_US+1) 0 ≤CLKDIV_10 ≤127	RW
7:0	CLKDIV	Dividing number to get ADC clock from device clock. The A/D works at the frequency between 20KHz and 200KHz. If CLKDIV = N, Then the freq of adc_clk = dev_clk / (N+1). 0 ≤ N ≤ 255	RW

13.2.11 ADC Command Register (ADCMD)

ADC Command Register ADCMD is used for write touch screen control command by software. Then, if the cmd_sel_r = 1, the controller will read ADCMD's command, then use command to control touch screen. The controller has 32x32 bit FIFO to store commands, the command format like this.

ADCMD **0x10070024**

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
	PIL	RPU					XPSUP	XNSUP	YPSUP	XPGRU	XNGRU	YNGRU	VREFNAUX	VREFNXN	VREFNXP	VREFNYN	VREFPVDD33	VREFPAUX	VREFPXN	VREFPXP	VREFPYP	XPADC	XNADC	YPADC	YNADC	WIPEADC	AUX2ADC	AUX1ADC	RPUWP	RPUXP	RPUYP	APIL
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	PIL	Current used for pressure measurement. 0: Ip = 200μA(default) 1: Ip = 400μA	RW
30:26	RPU	Internal Pull-up resistor for Pen Detection.	RW
25	XPSUP	XP to TPVDD control Switch. 0: open; 1:close.	RW
24	XNSUP	XN to TPGND control Switch. 0: open; 1:close.	RW
23	YPSUP	YP to TPVDD control Switch. 0: open; 1:close.	RW
22	XPGRU	XP to TPGND control Switch. 0: open; 1:close.	RW
21	XNGRU	XN to TPGND control Switch. 0: open; 1:close.	RW
20	YNGRU	YN to TPGND control Switch. 0: open; 1:close.	RW
19	VREFNAUX	ADC low voltage reference to AUX control switch. 0: open; 1: close.	RW
18	VREFNXN	ADC low voltage reference to XN control switch. 0: open; 1: close.	RW
17	VREFNXP	ADC low voltage reference to XP control switch. 0: open; 1: close.	RW
16	VREFNYN	ADC low voltage reference to YN control switch. 0: open; 1: close.	RW
15	VREFPVDD33	ADC high voltage reference to VDD control switch. 0: open; 1: close.	RW
14	VREFPAUX	ADC high voltage reference to AUX control switch. 0: open; 1: close.	RW
13	VREFPXN	ADC high voltage reference to XN control switch. 0: open; 1: close.	RW
12	VREFPXP	ADC high voltage reference to XP control switch. 0: open; 1: close.	RW
11	VREFPYP	ADC high voltage reference to YP control switch. 0: open; 1: close.	RW
10	XPADC	Use XP as ADC input channel control switch. 0: open; 1: close.	RW
9	XNADC	Use XN as ADC input channel control switch. 0: open; 1: close.	RW
8	YPADC	Use YP as ADC input channel control switch. 0: open; 1: close.	RW
7	YNADC	Use YN as ADC input channel control switch. 0: open; 1: close.	RW
6	WIPEADC	Use WIPE as ADC input channel control switch. 0: open; 1: close.	RW
5	AUX2ADC	Use AUX2 as ADC input channel control switch. 0: open; 1: close.	RW
4	AUX1ADC	Use AUX1 as ADC input channel control switch. 0: open; 1: close.	RW
3	RPUWP	Connect WP to RPU control switch. 0: open; 1: close.	RW
2	RPUXP	Connect XP to RPU control switch. 0: open; 1: close.	RW
1	RPUYP	Connect YP to RPU control switch. 0: open; 1: close.	RW
0	APIL	Use inter current source control switch. 0: open; 1: close.	RW

13.3 SAR A/D Controller Guide

The following describes steps of using SAR-ADC.

13.3.1 Power Down Mode

1 Then initial value of ADENA.POWER is 1, and the state of SADC is in dower down state.
2 When you want to use SADC, you should first set ADENA.POWERON to 0 to power on SADC. And you should wait for at least 2ms, then you can enable Touch Screen, VBAT and AUX.
3 When you want to power down SADC to get lower power, you should disable Touch Screen, VBAT and AUX, and then set ADENA.POWER to 1.

13.3.2 A Sample Touch Screen Operation

(Pen Down → Sample some data of several points → Pen Up)

1 Set ADCTRL to 0x1f to mask all the interrupt of SADC.
2 Set DMA_EN to choose whether to use DMA to read the sample data out or to use CPU to read the sample data out.
3 Set ADCFG.RPU to choose the Internal Pull-up resistor for Pen Detection.
4 Set ADCFG.WIRE_SEL to choose 4-wire or 5-wire mode in pendown detect.
5 Set ADCFG.CMD_SEL to choose use hardware inter command or software command control touch screen. If you want to use software command, you must write your command by ADCMD register. And if you want to use hardware inter command, please straight to set ADCFG.SPZZ.
6 Set ADCFG.SPZZ and ADCFG.XYZ to choose sample mode.
   a X_s→Y_s (Single-end X→Single-end Y).
   b X_d→Y_d (Differential X→Differential Y).
   c X_d→Y_d→$Z1_d$→$Z2_d$ or X_d→Y_d→$Z3_d$→$Z4_d$ (Reference register ADCFG.SPZZ)
   (Differential X→Differential Y→Differential Z1→Differential Z2 or
   Differential X→Differential Y→Differential Z3→Differential Z4).
   d X_d→Y_d→$Z1_d$→$Z2_d$→X2→Y2 or X_d→Y_d→$Z3_d$→$Z4_d$ →X2→Y2 (Reference register bit SPZZ).
7 Set ADCFG.SNUM to choose one point sampling times.
8 Set ADCLK.CLKDIV, ADCLK.CLKDIV_US and ADCLK.CLKDIV_MS to set A/D clock frequency.
9 Set ADWAIT to decide the wait time of pen down and the interval time between sampling different points. This time delay is necessary because when pen is put down or pen position change, there should be some time to wait the pen down signal to become stable.
10 Set ADSAME to decide the interval time between repeated sampling the same point. User can repeat sampling one point to get the most accurate data.
11 Set ADCTRL.PENDM to 0 to enable the pen down interrupt of touch panel.
12 Set ADENA.TCHEN to 1 to start touch panel.
13 When pen down interrupt happened, you should set ADCTRL.PENDM to 1 and clear

ADSTATE.PEND to close pen down interrupt. Then you should clear ADSTATE.PENDU and set ADCTRL.PENUM to 0 to enable pen up interrupt.

14 When pen down interrupt happened, the SARADC is sampling data. When ADSTATE.DTCH to 1, user must read the sample data from ADTCH. The SARADC will not sample the next point until the whole data of the one point are read (no matter by CPU or DMA). If ADCFG.XYZ is mode zero and mode one, user needs to read 1*ADCFG.SNUM times to get the whole data. In mode two, user needs to read 2*ADCFG.SNUM times to get the whole data. And in mode three, user needs to read 3*ADCFG.SNUM times to get all data.

15 Repeat 14 till pen up interrupt happened.

16 When pen up interrupt happened, you should set ADCTRL.PENUM to 1 and clear ADSTATE.PENU. Then you should clear ADSTATE.PEND and set ADCTRL.PENDM to 0 to enable pen down interrupt.

17 Wait pen down interrupt and repeat from 13.

18 When you want to shut down the touch screen, user can set the ADENA.TCHEN to 0. If the last point is not sampled completely, user can abandon it.

13.3.3 SLEEP mode Sample Operation

1 If the register ADCLK have not been set before, you should set ADCLK.CLKDIV, ADCLK.CLKDIV_US and ADCLK.CLKDIV_MS to set A/D clock frequency.

2 Clear ADSTATE.SLP_RDY, then you can set ADENA.SLP_MD to 1. When ADSTATE.SLP_RDY = 1, the Touch Screen is have entered the SLEEP mode.

3 After that you should clear ADSTATE.SLPEND and set ADCTRL.SLPENDM to 0 to enable “in SLEEP mode pen down interrupt” and mask all other interrupts. Then you can execute the SLEEP instruction to enter the SLEEP mode.

4 When “in SLEEP mode pen down interrupt” happened, it will switch from the SLEEP mode to NORMAL. Then you should set ADCTRL.SLPENDM to 1 and clear ADSTATE.SLPEND to close “in SLEEP mode pen down interrupt”. Clear ADSTATE.SLP_RDY, and you should set ADENA.SLP_MD to 0. When ADSTATE.SLP_RDY = 1, the Touch Screen is have exited the SLEEP mode.

5 Then you can do any other operations.

13.3.4 VBAT Sample Operation

1 Set ADCLK.CLKDIV, ADCLK.CLKDIV_US and ADCLK.CLKDIV_MS to set A/D clock frequency.

2 Set ADCFG.CH_MD to choose VBAT test mode channel.

3 Set ADENA.VBATEN to 1 to enable the channel.

4 When ADSTATE.VRDY = 1, you can read the sample data from ADVDAT. And the VBATEN will be set to 0 auto.

13.3.5 AUX Sample Operation

1 Set ADCFG.CMD to choose one CMD. (AUX1 or AUX2)
2 Set ADCLK.CLKDIV, ADCLK.CLKDIV_US and ADCLK.CLKDIV_MS to set A/D clock frequency.
3 Set ADENA.AUXEN to 1 to enable the channel.
4 When ADSTATE. ARDY = 1, you can read the sample data from ADADAT. And the AUXEN will be set to 0 auto.

13.3.6 Disable Touch Screen

1 When ADENA.TCHEN=1, ADENA.VBATEN=0, ADENA.AUXEN=0.
2 Set ADENA.TCHEN to 0.
3 Read ADENA.TCHEN till it is set to 0 by hardware, then Touch Screen is fully disabled.

13.3.7 Multi-touch Operation

If you want to detect multi-touch, you should follow to steps as below.

1 Set ADENA.YYZ=11.
2 Set ADCTRL.PENDM to 0 to enable the pen down interrupt of touch panel.
3 Set ADCFG.TCHEN=1 to start touch panel.
4 When pen down interrupt happened, you should set ADCTRL.PENDM to 1 and clear ADSTATE.PEND to close pen down interrupt. Then you should clear ADSTATE.PENDU and set ADCTRL.PENUM to 0 to enable pen up interrupt.
5 When ADSTATE.DTCH to 1, you can read the sample data from ADTCH. The measured data recorded as $X2_1$, $Y2_1$, you need to compare the measurement values of $X2_1$, $Y2_1$ and calibration values of X2, Y2. If $X2_1$<X2 and $Y2_1$<Y2, now is two points touch.
6 If the next measure is two points touch, the measured data recorded as $X2_2$, $Y2_2$, you can compare the $X2_1$, $y2_1$ and $X2_2$, $Y2_2$. If $X2_2$ >$X2_1$ and $Y2_2$ > $y2_1$, the touch movement state is shrinkage; if $X2_2$ <$X2_1$ and $Y2_2$ < $Y2_1$, the touch movement state is expand.

NOTE: Before in normally measurement ,you must calibration the values of X2 and Y2. When you calibration the X2, Y2 value, you need a single point of touch the touch panel, record the measurements data. Then repeat these measurements at least three times, recording measure data is $X2_1$, $X2_2$, $X2_3$ and $Y2_1$, $Y2_2$, $Y2_3$. You can use the formula (5) ,(6) to calculate the X2, Y2 value.

$$X2 = \frac{X2_1 + X2_2 + X2_3}{3} \quad (5)$$

$$Y2 = \frac{Y2_1 + Y2_2 + Y2_3}{3} \quad (6)$$

13.3.8 Use Software Command Operation

If you want to use software write command, you should follow to steps as below.

1 Set ADCFG.CMD_SEL =1.
2 Read ADCMD register once, discard the read-in data. This reading purpose is to activate the command logic.
3 Write you command to ADCMD register.
4 Write 0x00000000 to ADCMD register, indicate the written command is end.
5 Set ADENA.TCHEN=1 to start touch panel.

NOTE: The RPU values are 6bits in SARADC, but in ADCMD register , it has 5 bits only. In fact, the lowest RPU bit circuit is given as a fixed value of 1.

13.3.9 Use 5-wire touch panel Operation

If you want to use 5-wrie touch panel, you should complete the following set.

1 Set ADCFG.WIRE_SEL=1 to use 5-wire pendown detect.
2 Write all your commands to the command FIFO through ADCMD register.
3 Set ADENA.TCHEN=1 to start touch panel.

NOTE: In 5-wire mode, the ADCFG.SNUM will disable. The control logic will execution the control commands until the command value equals to 0x00000000.

13.3.10 Use External Touch Screen Controller Operation

If you want to use external touch screen controller, you should set ADCENA.TCHEN=0 and ADCENA.PEND=0, than you can use external touch screen controller freely.

NOTE: In this mode, all switches will open(default), but you can use VBAT or AUX sample operation by configure the appropriate register.

13.3.11 Use TSC to support keypad

SADC TSC function can apply to a keypad, if touch screen is not used. Suppose the keypad is a NxM matrix, where X direction has N key columns and Y direction has M key rows. Kij is used to indicate the key in ith column from left to right and jth row from bottom to top, where i=0~(N-1) and j=0~(M-1). Figure 13-1 is a 6x5 keypad circuit. The blue color is for X direction network and pink color is for Y. The networks are composed by resistors and metal line. These two networks should be connected to SADC 4 pins: XP/XN/YP/YN as illustrated in the figure. The gray circle is the key. When no key pressing, X network and Y network is open circuit. When a key is pressed, the X network and Y network is shorted under the key position.

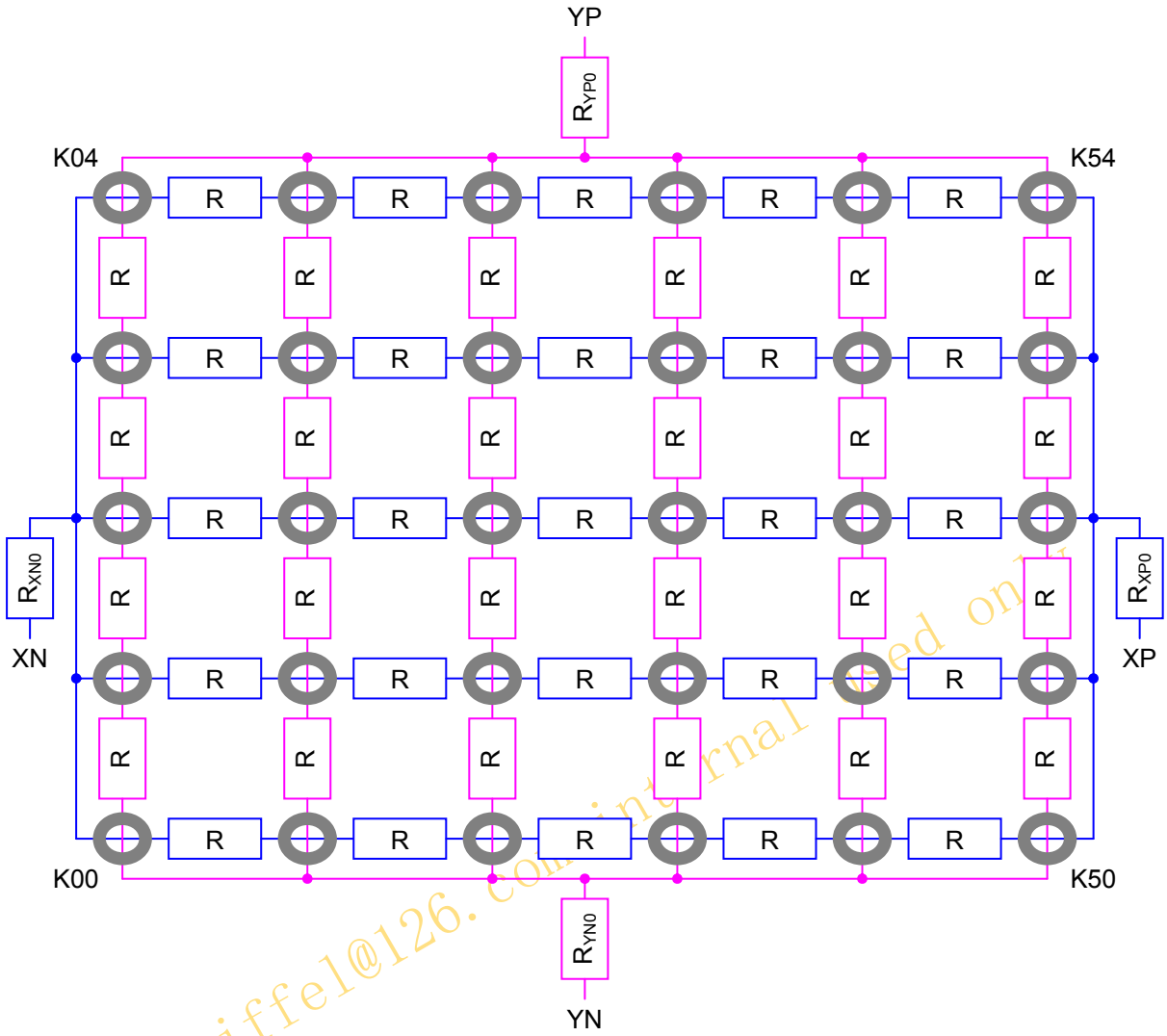


Figure 13-1 6x5 keypad circuit

When SADC is in waiting for pen-down status (C=1100), the equivalent circuit is show in Figure 13-2. When the key is not pressed, XP is open and the PEN is pulled to VDDADC, which is logic 1. When the key Kij is pressed, the circuit is: VDDADC→(10kΩ resistor)→R_{XP}→R_{YN}→VSSADC.

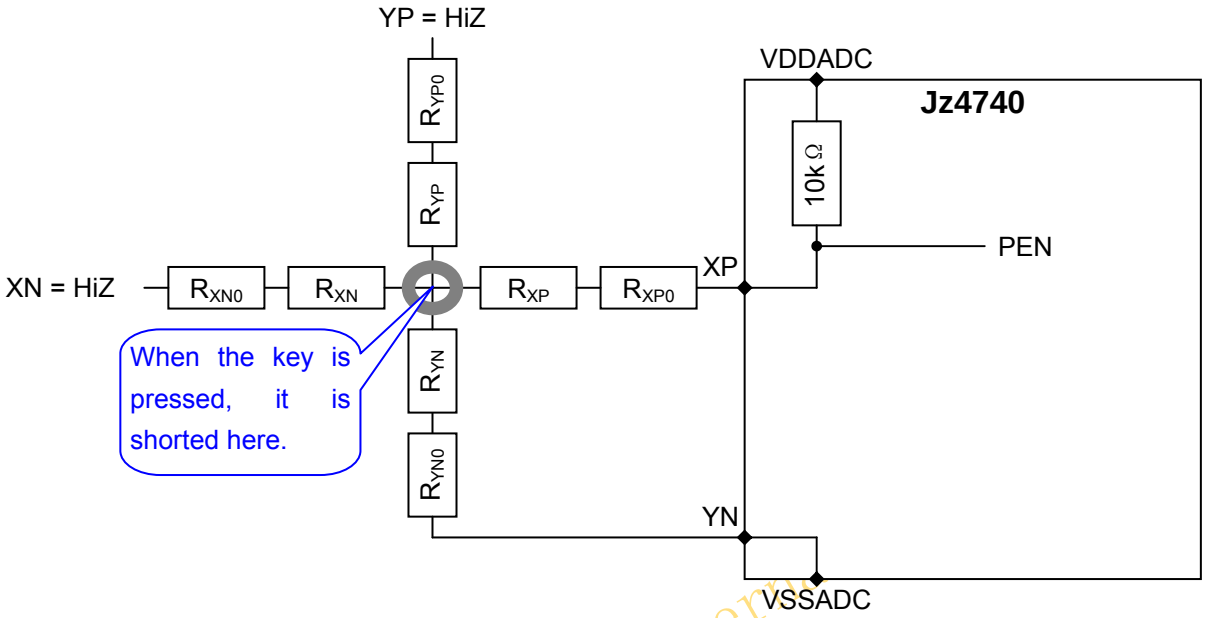


Figure 13-2 Wait for pen-down (C=1100) circuit

Where

$$R_{XP} = \frac{(N-1)^2 - i^2}{M \times (N-1-i) + 2i} \times R$$

$$R_{YN} = \frac{j \times (2M-2-j)}{N \times j + 2M - 2 - 2j} \times R$$

To ensure logic 0 at PEN in this case, following formula should be obeyed.

$$R_{XP} + R_{YN} + R_{XP0} + R_{YN0} \le 3k\Omega \quad (7)$$

It is suggested the value of N and M is as close to each other as possible. For N=2~20, M=2~20 and M=(N-1, N or N+1), we found

$$R_{XP} + R_{YN} < 2.7 \times R \quad (8)$$

After key pressing is found, the key Kij location, columns and row, should be measured by using C=0010 and C=0011 respectively. The equivalent circuits are show in Figure 13-3 and Figure 13-4, where

$$R_{X0} = \frac{N-1}{M-1} \times R$$
$$R_{Y0} = \frac{M-1}{N-1} \times R$$
$$R_{XNi} = i \times R$$
$$R_{XPi} = (N-1-i) \times R$$
$$R_{YNj} = j \times R$$
$$R_{YPj} = (M-1-j) \times R$$

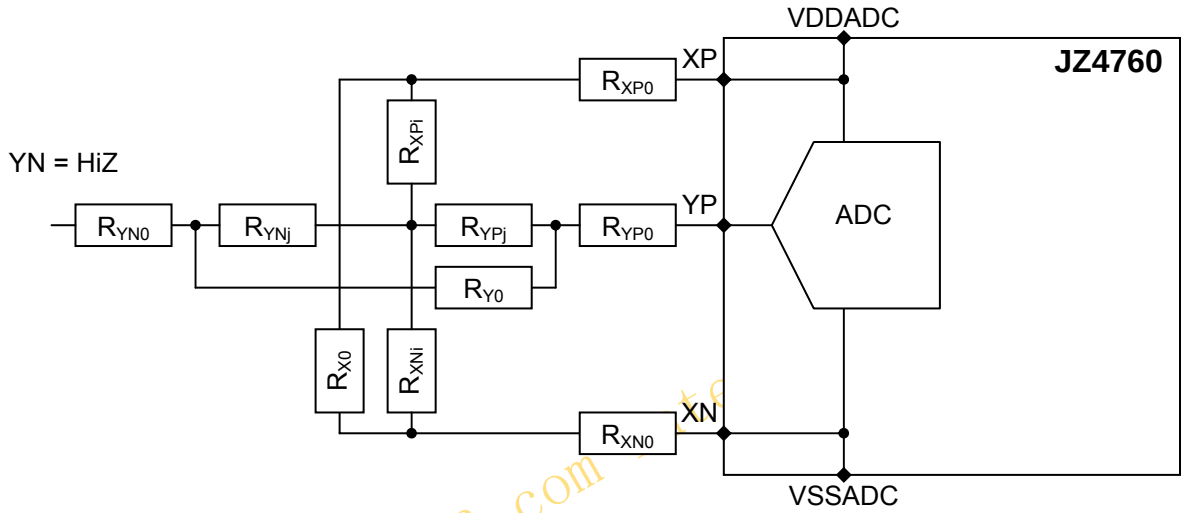


Figure 13-3 Measure X-position (C=0010) circuit

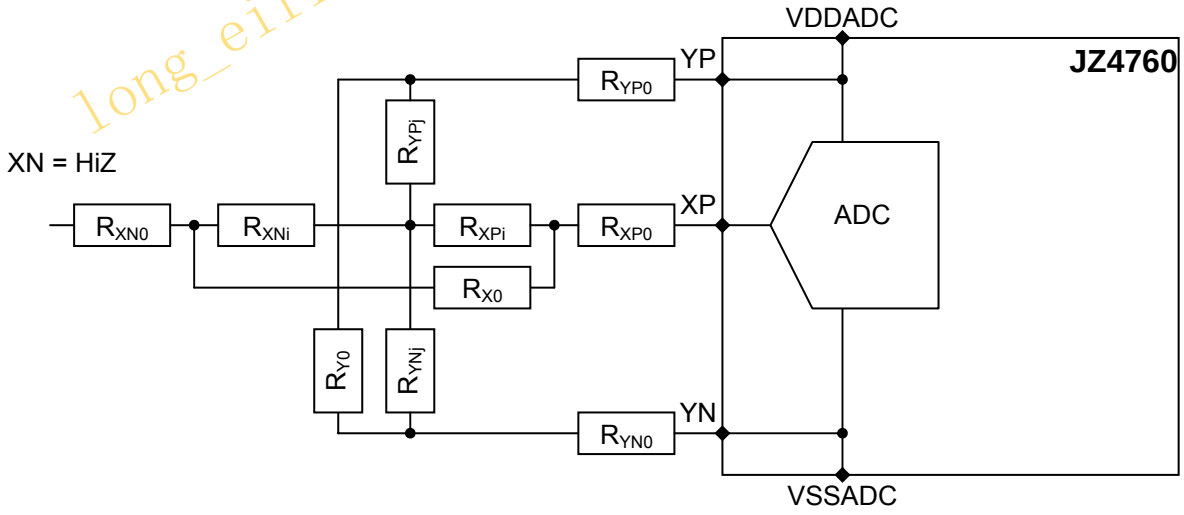


Figure 13-4 Measure Y-position (C=0011) circuit

So for Kij pressing, we should get ADC converted number Ni and Nj for i and j respectively.

$$Ni = \frac{R_{XN0} + \frac{i}{M}R}{R_{XN0} + \frac{N-1}{M}R + R_{XP0}} \times 4096$$

$$Nj = \frac{R_{YN0} + \frac{j}{N}R}{R_{YN0} + \frac{M-1}{N}R + R_{YP0}} \times 4096$$

It is required the resistor between XP and XN in case of C=0010, between YP and YN in case of C=0011, must be ≥200Ω and it better be ≥500Ω. Also consider the requirement in formula (7) and (8) above, we suggest to put R_{XP0} = R_{XN0} = R_{YP0} = R_{YN0} = 50Ω or 100Ω, put R = 500Ω ~ 1kΩ.

To use the keypad, the software should set:

ADENA.TCHEN = 1

ADCFG.XYZ = 10

The operation is similar to touch screen.