16/9/4-Channel Video Multiplexer & Multi-Speed MPEG4 Video SOLO6010 Data Sheets

Preliminary

May 15, 2007

SoftLogic Inc.

[Notice]

Softlogic Inc. has all rights herein contents and described this data sheet only to be used for developing purpose using SOLO6010. Without agree to Softlogic with contracting document, distribution and reproduction of all technical resources including data sheet, manual, schematic, software driver, etc may be illegal.

Softlogic Inc.
Tel) +82-31-205-9413
Fax) +82-31-205-9412

The information contained herein is subject to change without notice.

Contents

Figures

Tables

1. General Description

1.1. Overview

The SOLO6010 is a high performance, high integrated and cost effective single chip solution designed for used in storage and network surveillance applications such as a DVR (digital video recorder), network camera, network A/V server, video phone and video conference. Most of these devices require video multiplexer, multi stream MPEG4 video CODEC and multi-channel audio. The SOLO6010 contains 16 channel video multiplexer, maximum quadruple peed multi stream MPEG4 video CODEC, quadruple speed motion JPEG encoder, 20 channel G.723 voice CODEC, PCI/Host Interface and peripherals (RS232C, RS485, GPIO, SPI, IIC). The quadruple speed MPEG4 video CODEC and motion JPEG encoder can process 120 fps@D1 or 480@CIF in real time with only a single chip. Furthermore, the SOLO6010 has three kinds of option chips – for example, SOLO6010-16 for 16 channel devices, SOLO6010-9 for 9 channel devices and SOLO6010-4 for 4 channel devices. Therefore, the most high performance security systems can be designed easily and cost effectively with the SOLO6010 platform.

16 channel video multiplexer makes not only original video but also 4/9/16 scaled split video with includes asymmetric video scaling, which is possible to make a various video live output. The 16bit YUB full color OSG can be display with the video live output and the encoding OSG function enables to encode the video input with OSG to be record.

Multi-speed full hardwired MPEG4 video CODEC can encode or decode 4 channel D1 video or 16 channel CIF video in real time. Especially, the quadruple speed(four time speed) MPEG4 encoder and one time speed decoder with only a single chip is featured with 16 channel video multiplexer provides multi-channel applications such as DVRs with cost effective solution. Nowadays the networking function of DVR is very important because the internet technology has been developed very fast. The SOLO6010 has virtual MPEG4 video encoding to support two different MPEG4 video streams and motion JPEG encoding features. Therefore, the SOLO6010 enables the security system to make two stream MPEG4 stream and motion JPEG code for every video input channel. MPEG4 encoder parameters such as video size, frame interval, GOP (group of picture) size, video quality, and frame/field mode selection can be easily changed by configuring registers.

Full hardware G.723 CODEC process 20 channel encoding/decoding in real time and each channel can be used encoder or decoder according to applications.

The PCI can directly support both of PCI master and target without any other PCI bridge chip. Host interface can be also supported the CPUs which do not contain PCI. The SOLO6010 supports two UART (RS-232C, RS485) for camera pan/tilt/zoom, four IIC for video color decoder/encoder, SPI and GPIO.

The SOLO6010 can be used most of video applications such as a stand alone DVR, PCI card for PC based DVR, web monitor, network camera, network audio/video server, video conference and video home gateway.

1.2. Key Features

1.2.1. Video Input

- Accept NTSC/PAL video format
- 16/9/4 channel video input (ITU-R BT, 656 format)
- H scaler (1/2, 1/3, 2/3, 1/4, 3/4), V scaler (1/2)

1.2.2. Video Output

- 1 channel output (ITU-R BT,656 format)
- 1, 4, 9, 16, user split display
- Programmable OSG (On Screen Graphic)

1.2.3. MPEG4 Video CODEC and Motion JPEG Encoder

- ISO/IEC 1496-2 MPEG4 SOP@LEVEL 1,2,3
- Quadruple D1 Speed Full Hardwired MPEG4 Video CODEC Engine

[Encoding Only]

525/60 (NTSC) : 120 fps encoding @ 704x480 (D1)
240 fps encoding @ 704x240 (HD1)
480 fps encoding @ 352x240 (CIF)

625/50 (PAL) : 100 fps encoding @ 704x576 (D1)
200 fps encoding @ 704x240 (HD1)
400 fps encoding @ 352x288 (CIF)

[Decoding Only]

525/60 (NTSC) : 120 fps decoding @ 704x480 (D1)
240 fps decoding @ 704x240 (Half D1)
480 fps decoding @ 352x240 (CIF)

625/50 (PAL) : 100 fps decoding @ 704x576 (D1)
200 fps decoding @ 704x240 (Half D1)
400 fps decoding @ 352x288 (CIF)

[Encoding and Decoding]

525/60 (NTSC) : 120 fps encoding & 60 fps decoding @ 704x480 (D1)
240 fps encoding & 120 fps decoding @ 704x240 (HD1)
480 fps encoding & 240 fps decoding @ 352x240 (CIF)

625/50 (PAL) : 100 fps encoding & 50 fps decoding @ 704x576 (D1)
200 fps encoding & 100 fps decoding @ 704x240 (Half D1)
400 fps encoding & 200 fps decoding @ 352x288 (CIF)

- Compression parameters of each video channel can be configurable

  Frame size, frame rate, QP, GOP size, etc.

  There are not any restrictions with the SOLO6010 like as existing video multiplexer chips

- Motion JPEG encoded code can be possible for networking

525/60 (NTSC) :	120 fps encoding @ 704x480 (D1)
	240 fps encoding @ 704x240 (HD1)
	480 fps encoding @ 352x240 (CIF)
625/50 (PAL) :	100 fps encoding @ 704x576 (D1)
	200 fps encoding @ 704x240 (HD1)
	400 fps encoding @ 352x288 (CIF)

Motion JPEG is implemented with independent hardware core and its encoding performance does not affect MPEG4 video CODEC. However, its frame size should be same as MPEG4 video CODEC.

1.2.4. G.723 Voice CODEC

- 20 channel G.723 Voice CODEC
- Each channel encoding/decoding can be configured
- I2S master/slave mode interface or PCM interface

1.2.5. PCI/HOST Interface

- PCI Local Bus Specification, Rev. 2.2. 32bit/33MHz(66MHz)
- PCI Master/Target Mode.
- 32bit CPU Host Interface

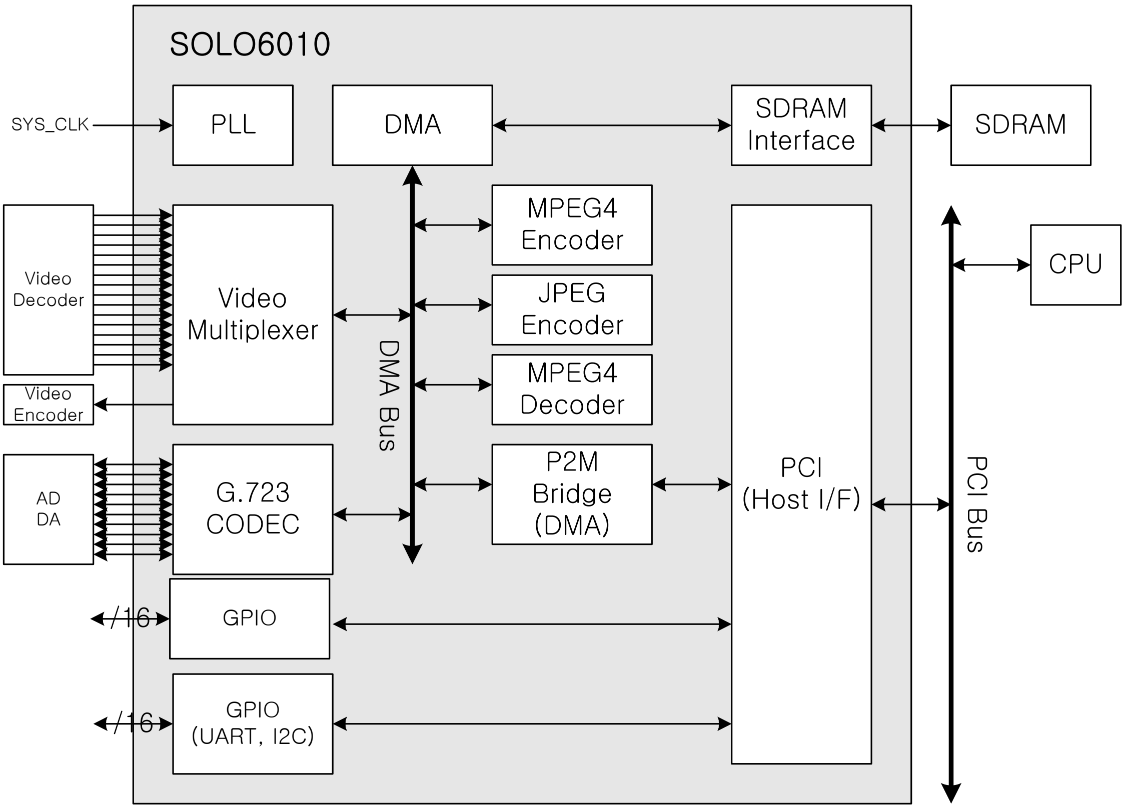
1.2.6. Peripheral

- RS485 and RS232C for camera pan/tilt/zoom
- Four I2C for video color decoders/encoders
- SPI for micro CPU communication
- 32 bit GPIO for sensor input and relay output

1.3. Device Options

Device Name	Features
SOLO6010-16	16 Channel Video Multiplexer & MPEG4 Video CODEC (16 Channel MPEG4 Video Encoding and Motion JPEG Encoding)
SOLO6010-9	16 Channel Video Multiplexer & MPEG4 Video CODEC (9 Channel MPEG4 Video Encoding and Motion JPEG Encoding)
SOLO6010-4	16 Channel Video Multiplexer & MPEG4 Video CODEC (4 Channel MPEG4 Video Encoding and Motion JPEG Encoding)

1.4. Block Diagram



2. Pin Description

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
A	GND	ioMD_0	ioMD_12	ioMD_22	ioMD_18	ioMD_25	ioMD_29	oMA_2		o_MCS		oMCLK	oMA_9	oMA_5	ioMD_34	ioMD_38	ioMD_45	ioMD_41	ioMD_50	ioMD_54	ioMD_61	GND
B	ioMD_4	GND	ioMD_11	ioMD_23	ioMD_19	ioMD_24	ioMD_28	oMA_3	oMA_10	oMBA_0	o_MWE	oMCKE	oMA_8	oMA_4	ioMD_35	ioMD_39	ioMD_44	ioMD_40	ioMD_51	ioMD_55	GND	ioMD_58
C	ioMD_5	ioMD_1	ioMD_10	ioMD_15	ioMD_20	ioMD_16	ioMD_27	ioMD_31	oMA_0	oMBA_1	o_MCAS	oMA_12	oMA_7	ioMD_32	ioMD_36	ioMD_47	ioMD_43	ioMD_48	ioMD_52	ioMD_63	ioMD_60	ioMD_57
D	ioMD_6	ioMD_2	ioMD_9	ioMD_14	ioMD_21	ioMD_17	ioMD_26	ioMD_30	oMA_1	oMDQM	o_MRAS	oMA_11	oMA_6	ioMD_33	ioMD_37	ioMD_46	ioMD_42	ioMD_49	ioMD_53	ioMD_62	ioMD_59	ioMD_56
E	ioMD_7	ioMD_3	ioMD_8	ioMD_13	ioGPIO_10	ioGPIO_11	ioGPIO_12	ioGPIO_13	ioGPIO_14	ioGPIO_15	VDD	GND	iSCAN_EN	iTEST_MODE_0	iTEST_MODE_1	iPCI_MODE	iPCI_CLKDLY_0	iPCI_CLKDLY_1	ioPCI_AD_3	ioPCI_AD_2	ioPCI_AD_1	ioPCI_AD_0
F	iVD_CLK01	iVD_CLK23	iVD_CLK45	iVD_CLK67	ioGPIO_9	VDD	VDD	VDD	VD33	VD33	VD33	VD33	VD33	VD33	VDD	VDD	VDD	iPCI_EEPROM	ioPCI_AD_7	ioPCI_AD_6	ioPCI_AD_5	ioPCI_AD_4
G	iVD_D01_0	iVD_D23_0	iVD_D45_0	iVD_D67_0	ioGPIO_8	VDD											VDD	iVDO_INIT_SELT	ioPCI_AD_10	ioPCI_AD_9	ioPCI_AD_8	io_PCI_CBE_0
H	iVD_D01_1	iVD_D23_1	iVD_D45_1	iVD_D67_1	ioGPIO_7	VDD											VDD	iSYS_CFG_0	ioPCI_AD_14	ioPCI_AD_13	ioPCI_AD_12	ioPCI_AD_11
J	iVD_D01_2	iVD_D23_2	iVD_D45_2	iVD_D67_2	ioGPIO_6	VD33			GND	GND	GND	GND	GND	GND			VD33	iSYS_CFG_1	io_PCI_SERR	ioPCI_PAR	io_PCI_CBE_1	ioPCI_AD_15
K	iVD_D01_3	iVD_D23_3	iVD_D45_3	iVD_D67_3	ioGPIO_5	VD33			GND	GND	GND	GND	GND	GND			VD33	iSYS_CFG_2	io_PCI_DEVSEL	io_PCI_STOP	i_PCI_LOCK	io_PCI_PERR
L	iVD_D01_4	iVD_D23_4	iVD_D45_4	iVD_D67_4	ioGPIO_4	VD33			GND	GND	GND	GND	GND	GND			VD33	VD33	io_PCI_CBE_2	io_PCI_FRAME	io_PCI_IRDY	io_PCI_TRDY
M	iVD_D01_5	iVD_D23_5	iVD_D45_5	iVD_D67_5	PLL_DVSS	PLL_DVDD			GND	GND	GND	GND	GND	GND			VD33	GND	ioPCI_AD_19	ioPCI_AD_18	ioPCI_AD_17	ioPCI_AD_16
N	iVD_D01_6	iVD_D23_6	iVD_D45_6	iVD_D67_6	ioGPIO_3	PLL_AVSS			GND	GND	GND	GND	GND	GND			VD33	oEE_PROM_DI	ioPCI_AD_22	ioPCI_AD_21	ioPCI_AD_20	iPCI_IDSEL
P	iVD_D01_7	iVD_D23_7	iVD_D45_7	iVD_D67_7	ioGPIO_2	PLL_AVDD			GND	GND	GND	GND	GND	GND			VD33	iEE_PROM_DO	io_PCI_CBE_3	ioPCI_AD_23	ioPCI_AD_24	ioPCI_AD_25
R	iVD_CLK89	iVD_CLKAB	iVD_CLKCD	iVD_CLKEF	ioGPIO_1	PLL_IO_VSS											VDD	oEE_PROM_CS	ioPCI_AD_29	ioPCI_AD_26	ioPCI_AD_27	ioPCI_AD_28
T	iVD_D89_0	iVD_DAB_0	iVD_DCD_0	iVD_DEF_0	ioGPIO_0	PLL_IO_VDD											VDD	oEE_PROM_SK	i_PCI_GNT	o_PCI_REQ	ioPCI_AD_31	ioPCI_AD_30
U	iVD_D89_1	iVD_DAB_1	iVD_DCD_1	iVD_DEF_1	iSYS_CLK_IN	VDD	VDD	VDD	VD33	VD33	VD33	VD33	VD33	VD33	VDD	VDD	VDD	ioPCM_SD_9			o_PCI_INT	iPCI_CLK
V	iVD_D89_2	iVD_DAB_2	iVD_DCD_2	iVD_DEF_2	i_SYS_RST_IN	o_SYS_RST_OUT	oSYS_CLK_OUT	i_TRST	iTCK	iTDI	GND	VDD	oTDO	iTMS	ioPCM_SD_5	ioPCM_SD_6	ioPCM_SD_7	ioPCM_SD_8			o_ATA_RST	ioATA_DIO_12
W	iVD_D89_3	iVD_DAB_3	iVD_DCD_3	iVD_DEF_3	iVD_D89_7	iVD_DAB_7	oVE_D_1	oVE_D_5	oVE_D_9	oVE_D_13	oVE_D_17	oVS_D_1	ioPCM_SD_2	ioPCM_SD_3	ioPCM_SD_4	iATA_INT_0	iATA_INT_1	oATA_ADR_0	ioATA_DIO_1	ioATA_DIO_5	ioATA_DIO_10	ioATA_DIO_13
Y	iVD_D89_4	iVD_DAB_4	iVD_DCD_4	iVD_DEF_4	iVD_DCD_7	iVD_DEF_7	oVE_D_2	oVE_D_6	oVE_D_10	oVE_D_14	oVE_D_18	oVS_D_2	oVS_D_5	ioPCM_SD_0	ioPCM_SD_1	iATA_IORDY_0	iATA_IORDY_1	oATA_ADR_1	ioATA_DIO_2	ioATA_DIO_6	ioATA_DIO_11	ioATA_DIO_14
AA	iVD_D89_5	GND	iVD_DAB_5	iVD_DCD_5	iVD_DEF_5	ioVE_CLK	oVE_D_3	oVE_D_7	oVE_D_11	oVE_D_15	oVS_CLK	oVS_D_3	oVS_D_6	oPCM_CLK	o_ATA_DIOR	o_ATA_CS_0	o_ATA_CS_2	oATA_ADR_2	ioATA_DIO_3	ioATA_DIO_7	GND	ioATA_DIO_15
AB	GND	iVD_D89_6	iVD_DAB_6	iVD_DCD_6	iVD_DEF_6	oVE_D_0	oVE_D_4	oVE_D_8	oVE_D_12	oVE_D_16	oVS_D_0	oVS_D_4	oVS_D_7	oPCM_SYNC	o_ATA_DIOW	o_ATA_CS_1	o_ATA_CS_3	ioATA_DIO_0	ioATA_DIO_4	ioATA_DIO_8	ioATA_DIO_9	GND

2.1. System Pins

Symbol	Pin	Type	Description
i_SYS_RST_IN	V5	I	*System Reset*. Active LOW.
iSYS_CLK_IN	U5	I	*System Clock.*
o_SYS_RST_OUT	V6	O	*System Reset Out*. Active LOW.
oSYS_CLK_OUT	V7	O	*System Clock Out.*

2.2. Configuration Pins

Symbol	Pin	Type	Description
iSCAN_EN	E13	I	*Scan Enable*. (Default LOW) The scan enable pin should be set to LOW in normal operation mode.
iTEST_MODE0	E14	I	*Test Mode*. (Default LOW) Test Mode pins should be set to LOW in normal operation mode.
iTEST_MODE1	E15		
iVDO_INIT_SELT	G18	I	*Video Initial Selection.* [0] : NTSC [1] : PAL
iPCI_MODE	E16	I	*System Clock.* [0] : HOST [1] : PCI
iPCI_CLKDLY0	E17	I	*PCI Clock Delay*. (Default LOW) The PCI clock phase will be controlled by setting these pins.
iPCI_CLKDLY1	E18	I	
iPCI_EEPROM	F18	I	*PCI EEPROM.* [0] : No use EEPROM [1] : Use EEPROM
iSYS_CFG0	H18	I	*System Configuration.* (Default HIGH) System Configuration pins should be set to HIGH.
iSYS_CFG1	J18	I	
iSYS_CFG2	K18	I	

2.3. JTAG Pins

Symbol	Pin	Type	Description
i_TRST	V8	I	*Test Reset.*
iTCK	V9	I	*Test Clock.*
iTDI	V10	I	*Test Data Input.*
oTDO	V13	O	*Test Data Output.*
iTMS	V14	I	*Test Mode Selection.*

2.4. Video Input Pins

Symbol	Pin	Type	Description
iVD_CLK01	F1	I	*Dual Video Clock Input Channel 0,1*. The rising edge is the video input clock for channel 0 and the falling edge is the video input clock of channel 1.
iVD_DAT01_0	G1	I	*Dual Video Data Input Channel 0,1*. ITU-R656 format stream data input for channel 0 and channel 1.
iVD_DAT01_1	H1		
iVD_DAT01_2	J1		
iVD_DAT01_3	K1		
iVD_DAT01_4	L1		
iVD_DAT01_5	M1		
iVD_DAT01_6	N1		
iVD_DAT01_7	P1		
iVD_CLK23	F2	I	*Dual Video Clock Input Channel 2,3*. The rising edge is the video input clock for channel 2 and the falling edge is the video input clock of channel 3.
iVD_DAT23_0	G2	I	*Dual Video Data Input Channel 2,3*. ITU-R656 format stream data input for channel 2 and channel 3.
iVD_DAT23_1	H2		
iVD_DAT23_2	J2		
iVD_DAT23_3	K2		
iVD_DAT23_4	L2		
iVD_DAT23_5	M2		
iVD_DAT23_6	N2		
iVD_DAT23_7	P2		
iVD_CLK45	F3	I	*Dual Video Clock Input Channel 4,5*.

			The rising edge is the video input clock for channel 4 and the falling edge is the video input clock of channel 5.
iVD_DAT45_0	G3	I	*Dual Video Data Input Channel 4,5*. ITU-R656 format stream data input for channel 4 and channel 5.
iVD_DAT45_1	H3		
iVD_DAT45_2	J3		
iVD_DAT45_3	K3		
iVD_DAT45_4	L3		
iVD_DAT45_5	M3		
iVD_DAT45_6	N3		
iVD_DAT45_7	P3		
iVD_CLK67	F4	I	*Dual Video Clock Input Channel 6,7*. The rising edge is the video input clock for channel 6 and the falling edge is the video input clock of channel 7.
iVD_DAT67_0	G4	I	*Dual Video Data Input Channel 6,7*. ITU-R656 format stream data input for channel 6 and channel 7.
iVD_DAT67_1	H4		
iVD_DAT67_2	J4		
iVD_DAT67_3	K4		
iVD_DAT67_4	L4		
iVD_DAT67_5	M4		
iVD_DAT67_6	N4		
iVD_DAT67_7	P4		
iVD_CLK89	R1	I	*Video Clock Input Channel 8,9*. The rising edge is the video input clock for channel 8,9.
iVD_DAT89_0	T1	I	*Video Data Input Channel 8,9*. ITU-R656 format stream data input for channel 8,9.
iVD_DAT89_1	U1		
iVD_DAT89_2	V1		
iVD_DAT89_3	W1		
iVD_DAT89_4	Y1		
iVD_DAT89_5	AA1		
iVD_DAT89_6	AB2		
iVD_DAT89_7	W5		
iVD_CLKAB	R2	I	*Video Clock Input Channel 10,11.* The rising edge is the video input clock for channel 10,11.
iVD_DATAB_0	T2	I	*Video Data Input Channel 10,11*. ITU-R656 format stream data input for channel 10,11.
iVD_DATAB_1	U2		
iVD_DATAB_2	V2		
iVD_DATAB_3	W2		
iVD_DATAB_4	Y2		
iVD_DATAB_5	AA3		
iVD_DATAB_6	AB3		
iVD_DATAB_7	W6		
iVD_CLKCD	R3	I	*Video Clock Input Channel 12,13.* The rising edge is the video input clock for channel 12,13.
iVD_DATCD_0	T3	I	*Video Data Input Channel 12,13*. ITU-R656 format stream data input for channel 12,13.
iVD_DATCD_1	U3		
iVD_DATCD_2	V3		
iVD_DATCD_3	W3		
iVD_DATCD_4	Y3		
iVD_DATCD_5	AA4		
iVD_DATCD_6	AB4		
iVD_DATCD_7	Y5		
iVD_CLKEF	R4	I	*Video Clock Input Channel 14,15*. The rising edge is the video input clock for channel 14,15.
iVD_DATEF_0	T4	I	*Video Data Input Channel 14,15*. ITU-R656 format stream data input for channel 14,15.
iVD_DATEF_1	U4		
iVD_DATEF_2	V4		
iVD_DATEF_3	W4		
iVD_DATEF_4	Y4		
iVD_DATEF_5	AA5		
iVD_DATEF_6	AB5		
iVD_DATEF_7	Y6		

2.5. Video Output Pins

Symbol	Pin	Type	Description
ioVE_CLK	AA6	I/O	*Video Clock Output*. The clock is for the video output data.
oVE _DAT_0	AB6	O	*Video Data Output*. ITU-R656 format stream data output.
oVE _DAT_1	W7		
oVE _DAT_2	Y7		
oVE _DAT_3	AA7		
oVE _DAT_4	AB7		
oVE _DAT_5	W8		
oVE _DAT_6	Y8		
oVE _DAT_7	AA8		
oVE _DAT_8	AB8	O	*Reserved.*
oVE _DAT_9	W9		
oVE _DAT_10	Y9		
oVE _DAT_11	AA9		
oVE _DAT_12	AB9		
oVE _DAT_13	W10		
oVE _DAT_14	Y10		
oVE _DAT_15	AA10		
oVE _DAT_16	AB10		
oVE _DAT_17	W11		
oVE _DAT_18	Y11		
oVS_CLK	AA11	O	*Reserved.*
oVS _DAT_0	AB11	O	*Reserved.* (Not supported the spot output.)
oVS _DAT_1	W12		
oVS _DAT_2	Y12		
oVS _DAT_3	AA12		
oVS _DAT_4	AB12		
oVS _DAT_5	Y13		
oVS _DAT_6	AA13		
oVS _DAT_7	AB13		

2.6. Audio Pins

Symbol	Pin	Type	Description
oPCM_SYNC	AB14	O	*PCM Sync*.
oPCM_CLK	AA14	O	*PCM Clock*.
ioPCM_S0	Y14	I/O	*PCM Serial Data Channel 0*.
ioPCM_S1	Y15	I/O	*PCM Serial Data Channel 1*.
ioPCM_S2	W13	I/O	*PCM Serial Data Channel 2*.
ioPCM_S3	W14	I/O	*PCM Serial Data Channel 3*.
ioPCM_S4	W15	I/O	*PCM Serial Data Channel 4*.
ioPCM_S5	V15	I/O	*PCM Serial Data Channel 5*.
ioPCM_S6	V16	I/O	*PCM Serial Data Channel 6*.
ioPCM_S7	V17	I/O	*PCM Serial Data Channel 7*.
ioPCM_S8	V18	I/O	*PCM Serial Data Channel 8*.
ioPCM_S9	U18	I/O	*PCM Serial Data Channel 9*.

2.7. Peripheral Pins

Symbol	Pin	Type	Description
ioGPIO0/SDO	T5	I/O	*General Purpose Input/Output Port0/SPI Data Out*.
ioGPIO1/SDI	R5	I/O	*General Purpose Input/Output Port1/SPI Data In*.
ioGPIO2/SCLK	P5	I/O	*General Purpose Input/Output Port2/SPI Clock*.
ioGPIO3/SSTB	N5	I/O	*General Purpose Input/Output Port3/SPI Strobe*.
ioGPIO4/SDA2	L5	I/O	*General Purpose Input/Output Port4/IIC Data2*.
ioGPIO5/SCL2	K5	I/O	*General Purpose Input/Output Port5/IIC Clock2*.
ioGPIO6/SDA3	J5	I/O	*General Purpose Input/Output Port6/IIC Data3*.
ioGPIO7/SCL3	H5	I/O	*General Purpose Input/Output Port7/IIC Clock3*.
ioGPIO8/RXD0	G5	I/O	*General Purpose Input/Output Port8/RS485 RxD*.

ioGPIO9/TXD0	F5	I/O	*General Purpose Input/Output Port9/RS485 TxD*.
ioGPIO10/nRXEN0	E5	I/O	*General Purpose Input/Output Port10/RS-485 nRxEn*.
ioGPIO11/TXEN0	E6	I/O	*General Purpose Input/Output Port11/RS-485 TxEn*.
ioGPIO12/RXD1	E7	I/O	*General Purpose Input/Output Port12/RS-232C RxD*.
ioGPIO13/TXD1	E8	I/O	*General Purpose Input/Output Port13/RS-232C TxD*.
ioGPIO14/nCTS1	E9	I/O	*General Purpose Input/Output Port14/RS-232C nCTS*.
ioGPIO15/nRTS1	E10	I/O	*General Purpose Input/Output Port15/RS-232C nRTS*.

2.8. Extended GPIO

Symbol	Pin	Type	Description
oATA_RST	V21	O	*Reserved*. (Not supported ATA IDE controller)
iATA_INT[0]	W16	I	*Reserved.*
iATA_INT[1]	W17	I	*Reserved.*
iATA_IORDY[0]	Y16	I	*Reserved.*
iATA_IORDY[1]	Y17	I	*Reserved.*
o_ATA_DIOR	AA15	O	*Reserved.*
o_ATA_DIOW	AB15	O	*Reserved.*
o_ATA_CS[0]	AA16	O	*Reserved.*
o_ATA_CS[1]	AB16	O	*Reserved.*
o_ATA_CS[2]	AA17	O	*Reserved.*
o_ATA_CS[3]	AB17	O	*Reserved.*
oATA_ADR[0]	W18	O	*Reserved.*
oATA_ADR[1]	Y18	O	*Reserved.*
oATA_ADR[2]	AA18	O	*Reserved.*
ioEXT_GPIO[16]	AB18	I/O	*General Purpose Input/Output Port16.*
ioEXT_GPIO[17]	W19	I/O	*General Purpose Input/Output Port17.*
ioEXT_GPIO[18]	Y19	I/O	*General Purpose Input/Output Port18.*
ioEXT_GPIO[19]	AA19	I/O	*General Purpose Input/Output Port19.*
ioEXT_GPIO[20]	AB19	I/O	*General Purpose Input/Output Port20.*
ioEXT_GPIO[21]	W20	I/O	*General Purpose Input/Output Port21.*
ioEXT_GPIO[22]	Y20	I/O	*General Purpose Input/Output Port22.*
ioEXT_GPIO[23]	AA20	I/O	*General Purpose Input/Output Port23.*
ioEXT_GPIO[24]	AB20	I/O	*General Purpose Input/Output Port24.*
ioEXT_GPIO[25]	AB21	I/O	*General Purpose Input/Output Port25.*
ioEXT_GPIO[26]	W21	I/O	*General Purpose Input/Output Port26.*
ioEXT_GPIO[27]	Y21	I/O	*General Purpose Input/Output Port27.*
ioEXT_GPIO[28]	V22	I/O	*General Purpose Input/Output Port28.*
ioEXT_GPIO[29]	W22	I/O	*General Purpose Input/Output Port29.*
ioEXT_GPIO[30]	Y22	I/O	*General Purpose Input/Output Port30.*
ioEXT_GPIO[31]	AA22	I/O	*General Purpose Input/Output Port31.*

2.9. SDRAM Pins

Symbol	Pin	Type	Description
oMCKE	B12	O	*SDRAM Clock Enable*.
oMCLK	A12	O	*SDRAM Clock*.
o_MCS	A10	O	*SDRAM Chip Select*.
o_MWE	B11	O	*SDRAM Chip Select*.
o_MRAS	D11	O	*SDRAM Row Strobe*.
o_MCAS	C11	O	*SDRAM Column Strobe*.
oMBA0	B10	O	*SDRAM Bank Address*.
oMBA1	C10	O	
oMA0	C9	O	*SDRAM Address*.
oMA1	D9	O	
oMA2	A8	O	
oMA3	B8	O	
oMA4	B14	O	
oMA5	A14	O	
oMA6	D13	O	
oMA7	C13	O	
oMA8	B13	O	
oMA9	A13	O	
oMA10	B9	O	
oMA11	D12	O	
oMA12	C12	O	

ioMD0	A2	I/O	*SDRAM Data*.
ioMD1	C2	I/O	
ioMD2	D2	I/O	
ioMD3	E2	I/O	
ioMD4	B1	I/O	
ioMD5	C1	I/O	
ioMD6	D1	I/O	
ioMD7	E1	I/O	
ioMD8	E3	I/O	
ioMD9	D3	I/O	
ioMD10	C3	I/O	
ioMD11	B3	I/O	
ioMD12	A3	I/O	
ioMD13	E4	I/O	
ioMD14	D4	I/O	
ioMD15	C4	I/O	
ioMD16	C6	I/O	
ioMD17	D6	I/O	
ioMD18	A5	I/O	
ioMD19	B5	I/O	
ioMD20	C5	I/O	
ioMD21	D5	I/O	
ioMD22	A4	I/O	
ioMD23	B4	I/O	
ioMD24	B6	I/O	
ioMD25	A6	I/O	
ioMD26	D7	I/O	
ioMD27	C7	I/O	
ioMD28	B7	I/O	
ioMD29	A7	I/O	
ioMD30	D8	I/O	
ioMD31	C8	I/O	
ioMD32	C14	I/O	
ioMD33	D14	I/O	
ioMD34	A15	I/O	
ioMD35	B15	I/O	
ioMD36	C15	I/O	
ioMD37	D15	I/O	
ioMD38	A16	I/O	
ioMD39	B16	I/O	
ioMD40	B18	I/O	
ioMD41	A18	I/O	
ioMD42	D17	I/O	
ioMD43	C17	I/O	
ioMD44	B17	I/O	
ioMD45	A17	I/O	
ioMD46	D16	I/O	
ioMD47	C16	I/O	
ioMD48	C18	I/O	
ioMD49	D18	I/O	
ioMD50	A19	I/O	
ioMD51	B19	I/O	
ioMD52	C19	I/O	
ioMD53	D19	I/O	
ioMD54	A20	I/O	
ioMD55	B20	I/O	
ioMD56	D22	I/O	
ioMD57	C22	I/O	
ioMD58	B22	I/O	
ioMD59	D21	I/O	
ioMD60	C21	I/O	
ioMD61	A21	I/O	
ioMD62	D20	I/O	
ioMD63	C20	I/O	
oMDQM	D10	O	*SDRAM Data Output Mask*.

2.10. PCI/Host Pins

Symbol	Pin	Type	Description
iPCI_CLK	U22	I	*PCI Clock*. 33MHz/66MHz.
ioPCI_AD0/DQ0	E22	I/O	*PCI Address/Data.* PCI Mode : PCI Address/Data. Host Mode : Data. See iPCI_MODE (Pin E16).
ioPCI_AD1/DQ1	E21	I/O	
ioPCI_AD2/DQ2	E20	I/O	
ioPCI_AD3/DQ3	E19	I/O	
ioPCI_AD4/DQ4	F22	I/O	
ioPCI_AD5/DQ5	F21	I/O	
ioPCI_AD6/DQ6	F20	I/O	
ioPCI_AD7/DQ7	F19	I/O	
ioPCI_AD8/DQ8	G21	I/O	
ioPCI_AD9/DQ9	G20	I/O	
ioPCI_AD10/DQ10	G19	I/O	
ioPCI_AD11/DQ11	H22	I/O	
ioPCI_AD12/DQ12	H21	I/O	
ioPCI_AD13/DQ13	H20	I/O	
ioPCI_AD14/DQ14	H19	I/O	
ioPCI_AD15/DQ15	J22	I/O	
ioPCI_AD16/DQ16	M22	I/O	
ioPCI_AD17/DQ17	M21	I/O	
ioPCI_AD18/DQ18	M20	I/O	
ioPCI_AD19/DQ19	M19	I/O	
ioPCI_AD20/DQ20	N21	I/O	
ioPCI_AD21/DQ21	N20	I/O	
ioPCI_AD22/DQ22	N19	I/O	
ioPCI_AD23/DQ23	P20	I/O	
ioPCI_AD24/DQ24	P21	I/O	
ioPCI_AD25/DQ25	P22	I/O	
ioPCI_AD26/DQ26	R20	I/O	
ioPCI_AD27/DQ27	R21	I/O	
ioPCI_AD28/DQ28	R22	I/O	
ioPCI_AD29/DQ29	R19	I/O	
ioPCI_AD30/DQ30	T22	I/O	
ioPCI_AD31/DQ31	T21	I/O	
ioPCI_CBE0/A2	G22	I/O	*PCI Command_Byte Enable/Address*. PCI Mode : PCI Command/Bye Enable. Host Mode : Address.
ioPCI_CBE1/A3	J21	I/O	
ioPCI_CBE2/A4	L19	I/O	
ioPCI_CBE3/A5	P19	I/O	
ioPCI_PAR/A6	J20	I/O	*PCI Parity/Address*. PCI Mode : PCI Parity. Host Mode : Address.
ioPCI_FRAME/A7	L20	I/O	*PCI Frame/Address*. PCI Mode : PCI Frame. Active LOW. Host Mode : Address.
ioPCI_IRDY/A8	L21	I/O	*PCI Master Ready/Address*. PCI Mode : PCI Master Ready. Active LOW. Host Mode : Address.
ioPCI_TRDY/A9	L22	I/O	*PCI Target Ready/Address*. PCI Mode : PCI Target Ready. Active LOW. Host Mode : Address.
ioCI_STOP/A10	K20	I/O	*PCI Stop/Address*. PCI Mode : PCI Stop. Active LOW. Host Mode : Address.
i_LOCK/i_OE	K21	I	*PCI Lock/Output Enable*. PCI Mode : PCI Lock. Active LOW. Host Mode : Output Enable. Active LOW.
i_PCI_IDSEL/i_WE	N22	I	*PCI ID Selection/Write Enable*. PCI Mode : PCI ID Selection. Host Mode : Write Enable. Active LOW.
io_PCI_DEVSEL/A11	K19	I/O	*PCI Device Selection/Address*. PCI Mode : PCI Device Selection. Active LOW. Host Mode : Address.
io_PCI_PERR/A12	K22	I/O	*PCI Parity Error/Address*. PCI Mode : PCI Parity Error. Active LOW. Host Mode : Address.
io_PCI_SERR/A13	J19	I/O	*PCI System Error/Address*.

			PCI Mode : PCI System Error. Active LOW. Host Mode : Address.
o_PCI_INT/i_INTR	U21	O	*PCI Interrupt Request/Interrupt Request*. PCI Mode : PCI Interrupt Request. Active LOW. Host Mode : Interrupt request. Active LOW.
o_PCI_REQ	T20	O	*PCI Request*. PCI Request. Active LOW.
i_PCI_GNT/i_CS	T19	I	*PCI Grant/Chip Select*. PCI Mode : PCI Grant. Active LOW. Host Mode : Chip Select. Active LOW.
oEEPROM_DI	N18	O	*PCI Configuration EEPROM Data In*.
iEEPROM_DO	P18	I	*PCI Configuration EEPROM Data Out*.
oEEPROM_CS	R18	O	*PCI Configuration EEPROM Chip Selection*.
oEEPROM_SK	T18	O	*PCI Configuration EEPROM Clock*.

2.11. Power Pins

Symbol	Pin	Type	Description
PLL_AVDD	P6	I	*PLL Analog Power.*
PLL_AVSS	N6	I	
PLL_DVDD	M6	I	
PLL_DVSS	M5	I	
PLL_IO_VDD	T6	I	
PLL_IO_VSS	R6	I	
VDD33	F9	I	*Power 3.3V.*
VDD33	F10	I	
VDD33	F11	I	
VDD33	F12	I	
VDD33	F13	I	
VDD33	F14	I	
VDD33	J6	I	
VDD33	J17	I	
VDD33	K6	I	
VDD33	K17	I	
VDD33	L6	I	
VDD33	L17	I	
VDD33	L18	I	
VDD33	M17	I	
VDD33	N17	I	
VDD33	P17	I	
VDD33	U9	I	
VDD33	U10	I	
VDD33	U11	I	
VDD33	U12	I	
VDD33	U13	I	
VDD33	U14	I	
VDD	E11	I	*Power 1.8V.*
VDD	F6	I	
VDD	F7	I	
VDD	F8	I	
VDD	F15	I	
VDD	F16	I	
VDD	F17	I	
VDD	G6	I	
VDD	G17	I	
VDD	H6	I	
VDD	H17	I	
VDD	R17	I	
VDD	T17	I	
VDD	U6	I	
VDD	U7	I	
VDD	U8	I	
VDD	U15	I	
VDD	U16	I	
VDD	U17	I	
VDD	V12	I	

VSS	A1	I	*Ground.*
VSS	A22	I	
VSS	B2	I	
VSS	B21	I	
VSS	E12	I	
VSS	J9	I	
VSS	J10	I	
VSS	J11	I	
VSS	J12	I	
VSS	J13	I	
VSS	J14	I	
VSS	K9	I	
VSS	K10	I	
VSS	K11	I	
VSS	K12	I	
VSS	K13	I	
VSS	K14	I	
VSS	L9	I	
VSS	L10	I	
VSS	L11	I	
VSS	L12	I	
VSS	L13	I	
VSS	L14	I	
VSS	M9	I	
VSS	M10	I	
VSS	M11	I	
VSS	M12	I	
VSS	M13	I	
VSS	M14	I	
VSS	M18	I	
VSS	N9	I	
VSS	N10	I	
VSS	N11	I	
VSS	N12	I	
VSS	N13	I	
VSS	N14	I	
VSS	P9	I	
VSS	P10	I	
VSS	P11	I	
VSS	P12	I	
VSS	P13	I	
VSS	P14	I	
VSS	V11	I	
VSS	AA2	I	
VSS	AA21	I	
VSS	AB1	I	
VSS	AB22	I	

3. Registers

3.1. System Clock Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0000	SYS_CONFIG	RW	[31]	*FDMA Reset.* Active HIGH. [0] : Normal operation. [1] : SDRAM controller reset. [1]→[0] : Initializing SDRAM and power-up sequence.	0
			[30]	*SDRAM Bit Width Selection.* [0] : 32bit Interface with SDRAM. [1] : 64bit Interface with SDRAM.	1
			[29:28]	*Reserved*.	0
			[27:26]	*Positive FDMA Clock Delay.* Factory Default.	0
			[25:24]	*Negative FDMA Clock Delay.* Factory Default.	0
			[23:22]	*Reserved.*	0
			[21:20]	*Reserved.*	0
			[19]	*System Clock Divider Control*	0
			[18:14]	*Input 5-bit Divider Control*. (PLL)	0
			[13:5]	*Feedback 9-bit Divider Control*. (PLL)	0
			[4:3]	*Output Divider Control.* (PLL)	0
			[2]	*Power Down Mode*. (PLL)	0
			[1]	*Bypass*. (PLL)	1
			[0]	*FOUT Enable*. (PLL)	0

3.2. FDMA Clock Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0004	FDMA_CONFIG		[15:8]	*SDRAM Refresh Cycle**. (Factory Default) SDRAM Controller will execute refresh every collapsed time setting value. The unit of setting value will be 256 times of system clock cycle.	0x10
			[7:6]	*SDRAM Size*. [00] : 4 * 16bit 64Mbit SDRAM 2 * 32bit 64Mbit SDRAM [01] : 4 * 16bit 128Mbit SDRAM 2 * 32bit 128Mbit SDRAM [10] : 4 * 16bit 256Mbit SDRAM [11] : 4 * 16bit 512Mbit SDRAM	2
			[5]	*SDRAM Clock Inversion**. (Factory Default)	1
			[4]	*FDMA Read Data Strobe Selection**. (Factory Default) The register is only effective in 32-bit FDMA interface mode.	1
			[3]	*FDMA Read Data Selection**. (Factory Default)	1
			[2]	*FDMA Read Data Clock Inversion**. (Factory Default)	0
			[1:0]	*FDMA Read Latency**. (Factory Default)	1
0x0008	DMA_CTRL1	RW	[5:2]	*Reserved. (BT Cycle)*	0
			[1]	*Reserved. (MultiMode_n)*	0
			[0]	*Reserved. (SeqMode_n)*	0

3.3. Video Clock Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x000C	VDO_CLK_CONIFG	RW	[22]	*Video Output Clock Inversion.* (Factory Default)	0
			[21:20]	*Video Output Clock Selection*. (Factory Default) [0] : SYS_CLK/4 [1] : SYS_CLK/2 [2] : SYS_CLK [3] : EXT_CLK (from pin ioVE_CLK)	0
			[19]	*Reserved. (Video Spot Output Clock Inversion*)	0

			[18]	*Video Input Clock Selection*. (Factory Default) [0] : 54MHz Dual ITU-R BT 656 format [1] : 27MHz Single ITU-R BT 656 format	0
			[17:16]	*Reserved*.	0
			[15:14]	*Video Input Clock Delay for Channel 14, 15*. (Factory Default)	0
			[13:11]	*Video Input Clock Delay for Channel 12, 13*. (Factory Default)	0
			[11:10]	*Video Input Clock Delay for Channel 10, 11*. (Factory Default)	0
			[9:8]	*Video Input Clock Delay for Channel 8, 9*. (Factory Default)	0
			[7:6]	*Video Input Clock Delay for Channel 6, 7*. (Factory Default)	0
			[5:4]	*Video Input Clock Delay for Channel 4, 5*. (Factory Default)	0
			[3:2]	*Video Input Clock Delay for Channel 2, 3*. (Factory Default)	0
			[1:0]	*Video Input Clock Delay for Channel 0, 1*. (Factory Default)	0

3.4. Interrupt Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0010	SYS_INT_STA_ACK	RW	[20]	*P2M3 Interrupt Status/Ack.* Read Mode : P2M3 Interrupt Status. Write Mode : P2M3 Interrupt Acknowledge.	0
			[19]	*P2M2 Interrupt Status/Ack.* Read Mode : P2M2 Interrupt Status. Write Mode : P2M2 Interrupt Acknowledge.	0
			[18]	*P2M1 Interrupt Status/Ack.* Read Mode : P2M1 Interrupt Status. Write Mode : P2M1 Interrupt Acknowledge.	0
			[17]	*P2M0 Interrupt Status/Ack.* Read Mode : P2M0 Interrupt Status. Write Mode : P2M0 Interrupt Acknowledge.	0
			[16]	*GPIO Interrupt Status/Ack.* Read Mode : GPIO Interrupt Status. Write Mode : GPIO Interrupt Acknowledge.	0
			[15]	*Reserved.*	0
			[14]	*Video Output Sync Interrupt Status/Ack.* Read Mode : *Video Input* Interrupt Status. Write Mode : *Video Input* Interrupt Acknowledge.	0
			[13]	*Video Motion Interrupt Status/Ack.* Read Mode : *Video Motion* Interrupt Status. Write Mode : *Video Motion* Interrupt Acknowledge.	0
			[12]	*ATA Command Interrupt Status/Ack.* Read Mode : *ATA Command* Interrupt Status. Write Mode : *ATA Command* Interrupt Acknowledge.	0
			[11]	*ATA Direction Interrupt Status/Ack.* Read Mode : *ATA Direction* Interrupt Status. Write Mode : *ATA Direction* Interrupt Acknowledge.	0
			[10]	*PCI Error Interrupt Status/Ack.* Read Mode : Fatal Interrupt Status. Write Mode : Fatal Interrupt Acknowledge.	0
			[9]	*PS2_1 Interrupt Status/Ack.* Read Mode : PS2 Interrupt Status. Write Mode : PS2 Interrupt Acknowledge.	0
			[8]	*PS2_0 Interrupt Status/Ack.* Read Mode : PCI ERROR Interrupt Status. Write Mode : PCI ERROR Interrupt Acknowledge.	0
			[7]	*SPI Interrupt Status/Ack.* Read Mode : SPI Interrupt Status. Write Mode : SPI Interrupt Acknowledge.	0

			[6]	*I2C Interrupt Status/Ack.* Read Mode : I2C Interrupt Status. Write Mode : I2C Interrupt Acknowledge.	0
			[5]	*UART1 Interrupt Status/Ack.* Read Mode : UART1 Interrupt Status. Write Mode : UART1 Interrupt Acknowledge.	0
			[4]	*UART0 Interrupt Status/Ack.* Read Mode : UART0 Interrupt Status. Write Mode : UART0 Interrupt Acknowledge.	0
			[3]	*Audio Interrupt Status/Ack.* Read Mode : Audio Interrupt Status. Write Mode : Audio Interrupt Acknowledge.	0
			[2]	*Reserved.*	
			[1]	*Decoder Interrupt Status/Ack.* Read Mode : Decoder Interrupt Status. Write Mode : Decoder Interrupt Acknowledge.	0
			[0]	*Encoder Interrupt Status/Ack.* Read Mode : *Encoder* Interrupt Status. Write Mode : *Encoder* Interrupt Acknowledge.	0
0x0014	SYS_INT_EN	RW	[20]	*P2M3 Interrupt Enable.*	0
			[19]	*P2M2 Interrupt Enable.*	0
			[18]	*P2M1 Interrupt Enable.*	0
			[17]	*P2M0 Interrupt Enable.*	0
			[16]	*GPIO Interrupt Enable.*	0
			[15]	*Reserved.*	0
			[14]	*Video Input Interrupt Enable.*	0
			[13]	*Video Motion Interrupt Enable.*	0
			[12]	*Reserved. (ATA Command Interrupt Enable)*	0
			[11]	*Reserved. (ATA Direction Interrupt Enable)*	0
			[10]	*PCI Error Interrupt Enable.*	0
			[9]	*PS2_1 Interrupt Enable.*	0
			[8]	*PS2_0 Interrupt Enable.*	0
			[7]	*SPI Interrupt Enable.*	0
			[6]	*I2C Interrupt Enable.*	0
			[5]	*UART1 Interrupt Enable.*	0
			[4]	*UART0 Interrupt Enable.*	0
			[3]	*Audio Interrupt Enable.*	0
			[2]	*Reserved.*	0
			[1]	*Decoder Interrupt Enable.*	0
			[0]	*Encoder Interrupt Enable.*	0

3.5. Test Mode Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x001C	TEST_MODE	RW	[15:8]	*Reserved.*	0
		RO	[2:0]	*Chip Option.* [111] : SOLO6010-16 [110] : SOLO6010-9 [101] : SOLO6010-4	-

3.6. EEPROM Access Register

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0060	EEPROM_ACC	RW	[7]	*PCI Configuration EEPROM Access Enable*.	0
			[3]	*PCI Configuration EEPROM Chip Select*.	0
			[2]	*PCI Configuration EEPROM Serial Clock*.	0
			[1]	*PCI Configuration EEPROM Data Output*.	0
		RO	[0]	*PCI Configuration EEPROM Data Input*.	-

3.7. PCI Error Register

Register Address	Register Name	Type	Bit Field	Description	Initial State

0x0070	PCI_ERROR	RW	[31]	*Timeout Error Enable.*	0
		RO	[28:24]	*Reserved. (FSM2 for Debugging)*	-
			[23:20]	*Reserved. (FSM1 for Debugging)*	-
			[19:16]	*Reserved. (FSM0 for Debugging)*	-
			[15:7]	*Reserved.*	-
			[6]	*P2M Descriptor Master Application.*	-
			[5]	*ATA Active in PCI Master Application.*	-
			[4]	*P2M Active as PCI Master Application.*	
			[3]	*Timeout Error.*	-
			[2]	*Target Abort Error.*	-
			[1]	*Parity Error.*	-
			[0]	*Fatal Error.*	-

3.8. P2M Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0080 0x00A0 0x00C0 0x00E0	P2M_CONFIG0 P2M_CONFIG1 P2M_CONFIG2 P2M_CONFIG3	RW	[10:6]	*DMA Waiting Interval*. (N*64 clocks) The interval between burst transaction is controlled by setting the value.	0
			[5]	*Byte Reorder*. [0] : RGB to RGB [1] : BGR to RGB	0
			[4]	16bit RGB Mode. [0] : {xrrr rrgg gggb bbbb} [1] : {rrrr rggg gggb bbbb}	0

			[3]	*Chroma Swap*.
				[0] : {Cb Y0 Cr Y1}
				[1] : {Cr Y0 Cb Y1}
					0

			[2]	*PCI Master Mode*. [0] : PCI Target/Host Mode. [1] : PCI Master Mode.	0
			[1]	*Descriptor Interrupt Mode.* [0] : Each Command Finish. [1] : Descriptor Queue Empty.	0
			[0]	*Descriptor Mode.* Active HIGH.	0
0x0084 0x00A4 0x00C4 0x00E4	P2M_DES_ADR0 P2M_DES_ADR1 P2M_DES_ADR2 P2M_DES_ADR3	RW	[31:0]	*Descriptor Base Address.* The descriptor queue will be store on CPU memory. The registers will points the addresses to be stored the 256 depth descriptor queues of which size are 4,096 bytes, respectively.	0
0x0088 0x00A8 0x00C8 0x00E8	P2M_DES_ID0 P2M_DES_ID1 P2M_DES_ID2 P2M_DES_ID3	RW	[7:0]	*Last Descriptor ID*. The registers are pointing the last descriptor. The P2M descriptor mode will be stopped after finishing the *Last Descriptor ID*.	0
0x008C 0x00AC 0x00CC 0x00EC	P2M_STATUS0 P2M_STATUS1 P2M_STATUS2 P2M_STATUS3	RO	[8]	*Command Done*. The register will be asserted at the interrupt request status.	-
			[7:0]	*Current Descriptor ID*. The register value will point the descriptor ID which is operated currently.	-
0x0090 0x00B0 0x00D0 0x00F0	P2M_CONTROL0 P2M_CONTROL1 P2M_CONTROL2 P2M_CONTROL3	RW	[31:20]	*Host Memory Increase Size (TI).* (DWORD)	0
			[19:10]	*Repeat Number (RP).*	0
			[9:7]	*Burst Size.* (Bytes) [0] : 512 [1] : 256 [2] : 128 [3] : 64 [4] : 32 [5] : 128 (2 page mode)	0

			[6]	When Color Space Conversion is asserted. *RGB Bit Width.* [0] : 24bit [1] : 16bit When Color Space Conversion is de-asserted. *VGA OSG Alpha Blending*. [0] : Off [1] : On	0
			[5:4]	When Color Space Conversion is asserted. [0] : Y[0] <=0 (Off) [1] : Y[0] <=1 [2] : Y[0] <= G[0] [3] : Y[0] <= Bit[15] When Color Space Conversion is de-asserted. [0] : Y[0] <= 0 [1] : [15] <= 1 [2] : [15] <= [5] [3] : [15] <= [15]	0
			[3]	*Color Space Conversion On.*	0
			[2]	*Reserved.*	0
			[1]	*Write Enable.* [0] : Read [1] : Write	0
			[0]	*Transaction On.*	0
0x0094 0x00B4 0x00D4 0x00F4	P2M_EXT_CFG0 P2M_EXT_CFG1 P2M_EXT_CFG2 P2M_EXT_CFG3	RW	[31:20]	*External Memory Increase Size (FI).* (DWORD)	0
			[19:0]	*External Memory Copy Size (FC).* (DWORD)	0
0x0098 0x00B8 0x00D8 0x00F8	P2M_TAR_ADR0 P2M_TAR_ADR1 P2M_TAR_ADR2 P2M_TAR_ADR3	RW	[31:0]	*Host Memory Base Address (TA).*	0
0x009C 0x00BC 0x00DC 0x00FC	P2M_EXT_ADR0 P2M_EXT_ADR1 P2M_EXT_ADR2 P2M_EXT_ADR3	RW	[31:0]	*External Memory Base Address (FA).*	0

3.9. Video Input Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0100	VI_CH_SWITCH0	RW	[29:25]	*Video Input Channel Number for Internal Video Ch5.*	5
			[24:20]	*Video Input Channel Number for Internal Video Ch4.*	4
			[19:15]	*Video Input Channel Number for Internal Video Ch3.*	3
			[14:10]	*Video Input Channel Number for Internal Video Ch2.*	2
			[19:5]	*Video Input Channel Number for Internal Video Ch1.*	1
			[4:0]	*Video Input Channel Number for Internal Video Ch0.*	0
0x0104	VI_CH_SWITCH1	RW	[29:25]	*Video Input Channel Number for Internal Video Ch11.*	11
			[24:20]	*Video Input Channel Number for Internal Video Ch10.*	10
			[19:15]	*Video Input Channel Number for Internal Video Ch9.*	9
			[14:10]	*Video Input Channel Number for Internal Video Ch8.*	8
			[19:5]	*Video Input Channel Number for Internal Video Ch7.*	7
			[4:0]	*Video Input Channel Number for Internal Video Ch6.*	6
0x0108	VI_CH_SWITCH2	RW	[24:20]	*Video Input Channel Number for Internal Spot Output.*	16
			[19:15]	*Video Input Channel Number for Internal Video Ch15.*	15
			[14:10]	*Video Input Channel Number for Internal Video Ch14.*	14
			[19:5]	*Video Input Channel Number for Internal Video Ch13.*	13
			[4:0]	*Video Input Channel Number for Internal Video Ch12.*	12
0x010C	VI_CH_ENA	RW	[15:0]	*Video Input Channel Enable.* Active HIGH. Bit[15:0] : {Ch15, Ch14, Ch13, … , Ch0}	1
0x0110	VI_CH_FORMAT	RW	[31:16]	*Video Input Format Decoder Selection1.* Bit[15:0] : {Ch15, Ch14, Ch13, … , Ch0} [0] : Primary [1] : Secondary	0

			[15:0]	*Video Input Format Decoder Selection0.*
				Bit[15:0] : {Ch15, Ch14, Ch13, … , Ch0}
				[0] : Interlace
				[1] : Progressive
				
				0x011C : VIN{Primary, Interlaced}
				0x0120 : VIN{Secondary, Interlaced}
				0x0124 : VIN{Don't care, Progressive}
					0

0x0114	VI_FMT_CFG	RW	[31]	*Vertical Line Check.* The register is effective for all video input channels. The video standard or stability is checked by counting vertical line. The response for all video input channels will be stored on register VI_STATUS1[15:0]. Refer to register address 0x012C.	0
			[30]	*Horizontal Pixel Check.* The register is effective for all video input channels. Video standard or stability is checked by counting horizontal pixel. The response for all video input channels will be stored on register VI_STATUS1[15:0]. Refer to register address 0x012C.	0
			[29]	*Reserved. (Video Standard or Unstable Check)*	0
			[28]	*Test Signal Generation.* (Only for Test)	0
			[27]	*Mosaic Horizontal Length.* [0] : 16 [1] : 32	0
			[25:22]	*Mosaic Darkness Strength.* The brightness in mosaic area is controlled by the registers. R = O >> N Where, R is the result, O is the original video data, N is the value on the register.	0
			[21:11]	*Display Horizontal Offset.* The register is effective for all video input channels.	0
			[10:0]	*Compression Horizontal Offset.* The register is effective for all video input channels.	0
0x0118	VI_PAGE_SWITCH	RW	[16]	*Reserved. (Input Synchronization on Field Index)*	0
			[15]	*Reserved. (Input Field Index for Calculation)*	0
			[14]	*Reserved. (Output Field Index for Calculation)*	0
			[1:0]	*Writing Page Distance.* Four frame buffers are used to display video output. For avoiding the frame overlapping,	2
0x011C	VI_ACT_PRI_INT	RW	[31]	*Field Index Inversion.*	0
			[30]	*Reserved. (Test Mode)*	0
			[29:21]	*Horizontal Start.*	8
			[20:11]	*Vertical Start.*	0

			[10:0]	*Vertical End.*	242(N)
					290(P)

0x0120	VI_ACT_SEC_INT	RW	[31]	*Field Index Inversion.*	0
			[30]	*Reserved. (Test Mode)*	0
			[29:21]	*Horizontal Start.*	8
			[20:11]	*Vertical Start.*	0

			[10:0]	*Vertical End.*	242(N)
					290(P)

0x0124	VI_ACT_PRI_PRG	RW	[31]	*Field Index Inversion.*	0
			[30]	*Reserved. (Test Mode)*	0
			[29:21]	*Horizontal Start.*	8
			[20:11]	*Vertical Start.*	0

			[10:0]	*Vertical End.*	242(N)
					290(P)

0x0128	VI_STATUS0	RO	[11:4]	*Reserved. (Field Free Count)*	-
			[3]	*Video Display FI.* The register shows the current displaying field index.	-
			[2:0]	*Video Display Page.* The register shows the current displaying page.	-
0x012C	VI_STATUS1	RO	[27:24]	*Reserved. (DMA Status)*	-
			[23:16]	*Reserved. (Capture Queue Status)*	-

			[15:0]	*Standard and Stability Flag.* The registers show if the video input signal is standard or loss. However, the loss detection function of the video color decoder is strongly recommended for loss detection.	-

3.10. Internal MPEG4 Video Decoder Playback Controller Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0130	VI_PB_CONFIG	RW	[1]	*Playback Video Mode.* [0] : Default [1] : User Programming	
			[0]	*Playback Video System*. [0] : NTSC [1] : PAL	
0x0134	VI_PB_RANGE_HV	RW	[21:12]	*Playback Horizontal Size.*	
			[11:0]	*Playback Vertical Size.*	
0x0138	VI_PB_ACT_H	RW	[21:12]	*Playback Horizontal Size.*	
			[11:0]	*Playback Vertical Size.*	
0x013C	VI_PB_ACT_V	RW	[21:12]	*Playback Horizontal Size.*	
			[11:0]	*Playback Vertical Size.*	

3.11. Video Input Mosaic Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0140 0x0144 0x0148 0x014C 0x0150 0x0154 0x0158 0x015C 0x0160 0x0164 0x0168 0x016C 0x0170 0x0174 0x0178 0x017C	VI_WIN_MOSAIC0 VI_WIN_MOSAIC1 VI_WIN_MOSAIC2 VI_WIN_MOSAIC3 VI_WIN_MOSAIC4 VI_WIN_MOSAIC5 VI_WIN_MOSAIC6 VI_WIN_MOSAIC7 VI_WIN_MOSAIC8 VI_WIN_MOSAIC9 VI_WIN_MOSAIC10 VI_WIN_MOSAIC11 VI_WIN_MOSAIC12 VI_WIN_MOSAIC13 VI_WIN_MOSAIC14 VI_WIN_MOSAIC15	RW	[31:24]	*Horizontal Start * 16.*	0
			[23:16]	*Horizontal End * 16.*	0
			[15:8]	*Vertical Start * 8.*	0
			[7:0]	*Vertical End * 8.*	0

3.12. Window Control Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0180 0x0184 0x0188 0x018C 0x0190 0x0194 0x0198 0x019C 0x01A0 0x01A4 0x01A8 0x01AC 0x01B0 0x01B4 0x01B8 0x01BC	VI_WIN_CTRL0 VI_WIN_CTRL1 VI_WIN_CTRL2 VI_WIN_CTRL3 VI_WIN_CTRL4 VI_WIN_CTRL5 VI_WIN_CTRL6 VI_WIN_CTRL7 VI_WIN_CTRL8 VI_WIN_CTRL9 VI_WIN_CTRL10 VI_WIN_CTRL11 VI_WIN_CTRL12 VI_WIN_CTRL13 VI_WIN_CTRL14 VI_WIN_CTRL15	RW	[31:28]	*Live Channel ID for Window.*	0
			[27]	*PIP On.*	0
			[26:24]	*Scale Mode.* [0] : Off [1] : H=1/1, V=1/1 [2] : H=1/2, V=1/1 [3] : H=1/2, V=1/2 [4] : H=1/3, V=1/3 [5] : H=1/4, V=1/4 [6] : H=2/3, V=2/3 [7] : H=3/4, V=3/4 [8]~[15] : H=1/1, V=1/1 for PIP	0
			[23:12]	*Window Horizontal Start * 4.*	0
			[11:0]	*Window Horizontal End * 4.*	0

0x01C0 0x01C4 0x01C8 0x01CC 0x01D0 0x01D4 0x01D8 0x01DC 0x01E0 0x01E4 0x01E8 0x01EC 0x01F0 0x01F4 0x01F8 0x01FC	VI_WIN_CTRL0 VI_WIN_CTRL1 VI_WIN_CTRL2 VI_WIN_CTRL3 VI_WIN_CTRL4 VI_WIN_CTRL5 VI_WIN_CTRL6 VI_WIN_CTRL7 VI_WIN_CTRL8 VI_WIN_CTRL9 VI_WIN_CTRL10 VI_WIN_CTRL11 VI_WIN_CTRL12 VI_WIN_CTRL13 VI_WIN_CTRL14 VI_WIN_CTRL15	RW	[23:12]	*Window Vertical Start.*	0
			[11:0]	*Window Vertical End.*	0
0x0200 0x0204 0x0208 0x020C 0x0210 0x0214 0x0218 0x021C 0x0220 0x0224 0x0228 0x022C 0x0230 0x0234 0x0238 0x023C	VI_WIN_ON0 VI_WIN_ON1 VI_WIN_ON2 VI_WIN_ON3 VI_WIN_ON4 VI_WIN_ON5 VI_WIN_ON6 VI_WIN_ON7 VI_WIN_ON8 VI_WIN_ON9 VI_WIN_ON10 VI_WIN_ON11 VI_WIN_ON12 VI_WIN_ON13 VI_WIN_ON14 VI_WIN_ON15	RW	[1]	*Playback On.*	0
			[0]	*Live On.*	0
0x0240	VI_WIN_SW	RW	[4:0]	*Live On Delay*. (Frame Interval)	0
0x0244	VI_WIN_MUTE	RW	[0]	*Live Auto Mute.*	0

3.13. Horizontal Expansion Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0250	VO_HOR_EXAND0	RW	[12]	*Horizontal Expansion On.*	0
			[11:0]	*Horizontal Position.*	0
0x0254	VO_HOR_EXAND1	RW	[12]	*Horizontal Expansion On.*	0
			[11:0]	*Horizontal Position.*	0
0x0258	VO_HOR_EXAND2	RW	[12]	*Horizontal Expansion On.*	0
			[11:0]	*Horizontal Position.*	0
0x025C	VO_HOR_EXAND3	RW	[12]	*Horizontal Expansion On.*	0
			[11:0]	*Horizontal Position.*	0

3.14. Video Input Motion Detection Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0260	VI_MOTION_ADR	RW	[31:16]	*Motion Detection Enable.* Bit[31:16] = {Ch15, Ch14, …, Ch0}	0
			[15:0]	*Motion Base Address.*	0
0x0264	VI_MOTION_CTRL	RW	[31:24]	*Motion Sustain Frame Count.*	0
			[21:16]	*Maximum Sample Length.*	0
			[15]	*Interrupt Mode.* [0] : Every Motion Event [1] : Start/Stop Motion Event.	0
			[14]	*Freeze Motion Data.*	0
			[13:0]	*Minimum Detected Sample Count.*	0
0x0268	VI_MOTION_CLEAR	RW	[15:0]	*Motion Flag Clear.*	0
0x026C	VI_MOTION_STA	RO	[15:0]	*Motion Detected Flag.* Bit[15:0] = {ch15, ch14, ch13, …, ch0} [0] : Non motion detected video input channel [1] : Motion detected video input channel	-

0x0270	VI_MOTION_BDR	RW	[29]	Border Y Set On.	0
			[28]	Border Y Addition On.	0
			[27]	Border CB Set On.	0
			[26]	Border CB Addition On.	0
			[25]	Border CR Set On.	0
			[24]	Border CR Addition On.	0
			[23:16]	Border Y Value.	0
			[15:8]	Border CB Value.	0
			[7:0]	Border CR Value.	0
0x0274	VI_MOTION_BAR	RW	[29]	Bar Y Set On.	0
			[28]	Bar Y Addition On.	0
			[27]	Bar CB Set On.	0
			[26]	Bar CB Addition On.	0
			[25]	Bar CR Set On.	0
			[24]	Bar CR Addition On.	0
			[23:16]	Bar Y Value.	0
			[15:8]	Bar CB Value.	0
			[7:0]	Bar CR Value.	0

3.15. Video Output Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x02F4	VO_USER656_F0	RW	[29:24]	Field Index Transition Position F0.	0
			[23:17]	VSYNC Start Position F0.	0
			[16:11]	VSYNC End Position F0.	0
			[10:0]	Number of Vertical Line F0.	0
0x02F8	VO_USER656_F1	RW	[29:24]	Field Index Transition Position F1.	0
			[23:17]	VSYNC Start Position F1.	0
			[16:11]	VSYNC End Position F1.	0
			[10:0]	Number of Vertical Line F1.	0
0x0300	VO_FMT_ENC	RW	[31]	Reserved.	0
			[30]	Video System. [0] : NTSC [1] : PAL	0
			[29]	User 656 Format Enable. [0] : Standard 656 format [1] : User Programmed 656 format	0
			[28]	Reserved.	0
			[26]	Reserved.	0
			[25]	Reserved.	0
			[23:0]	User Frame Size Clock Count. (When 26 is on)	0
			[20]	FI, V Change Enable. (When SAV)	0
			[19]	User Color Set for VSYNC.	0
			[18]	User Color Set for HSYNC.	0
			[17]	User Color Set for Non Active of H.	0
			[16]	User Color Set for Non Active of V.	0
			[15:8]	Non Active Color for Y.	0
			[7:4]	Non Active Color for CB/16.	0
			[3:0]	Non Active Color for CR/16.	0
0x0304	VO_ACT_H	RW	[31:22]	H Blank Size.	0
			[21:11]	H Start Position.	6
			[10:0]	H End Position.	710
0x0308	VO_ACT_V	RW	[31:22]	V Blank Size.	0
			[21:11]	V Start Position.	8

			[10:0]	V End Position.	248(N)
					296(P)

0x030C	VO_RANGE_HV	RW	[24]	SYNC Inversion.	0
			[23]	HSYNC Inversion.	0
			[22]	VSYNC Inversion.	0
			[21:11]	Horizontal Size.	704

			[10:0]	Vertical Size.	240(N)
					288(P)

3.16. Video Output Control Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0310	VO_DISP_CFG	RW	[31]	*Display On.*	0
			[27:24]	*Erase Frame Count. (After ease off to on)*	4
			[22]	*Double Scan.* [0] : 30Hz [1] : 60Hz	0
			[21]	*Display Single Page.*	0
			[15:0]	*Display Base Address.*	0
0x0314	VO_DISP_ERASE	RW	[1]	*Display Erase On.*	1
0x0318	VO_ZOOM_CTRL	RW	[24]	*Vertical Zoom On.*	0
			[23]	*Horizontal Zoom On.*	0
			[22]	*Vertical Compensation. (When Vertical Zoom)*	0
			[21:11]	*Zoom Offset of Horizontal*2.*	0
			[10:0]	*Zoom Vertical Offset.*	0
0x031C	VO_FREEZE_CTRL	RW	[1]	*Freeze Live.*	0
			[0]	*Freeze Live Interpolation.*	0
0x0320	VO_BGK_COLOR	RW	[23:16]	*Y.*	0x10
			[15:8]	*U.*	0x80
			[7:0]	*V.*	0x80
0x0324	VO_DEINTERLACE	RW	[19:8]	*Reserved.*	0
			[7:0]	*Reserved.*	0

3.17. Line and Bar Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0330	VO_BDG_COLOR	RW	[31:24]	*Border Ext Y.*	0x80
			[23:20]	*Border Ext U*16.*	0x8
			[19:16]	*Border Ext V*16.*	0x8
			[15:18]	*Border Cell Y.*	0x10
			[8:4]	*Border Cell U*16.*	0x8
			[3:0]	*Border Cell V*16.*	0x8
0x0334	VO_BAR_COLOR	RW	[31:24]	*Bar Ext Y.*	0x80
			[23:20]	*Bar Ext U*16.*	0x8
			[19:16]	*Bar Ext V*16.*	0x8
			[15:18]	*Bar Cell Y.*	0x10
			[8:4]	*Bar Cell U*16.*	0x8
			[3:0]	*Bar Cell V*16.*	0x8
0x0338	VO_LINE_MASK_CELL	RW	[15]	*Border On 15.*	0
			[14]	*Border On 14.*	0
			[13]	*Border On 13.*	0
			[12]	*Border On 12*	0
			[11]	*Border On 11.*	0
			[10]	*Border On 10.*	0
			[9]	*Border On 9.*	0
			[8]	*Border On 8.*	0
			[7]	*Border On 7.*	0
			[6]	*Border On 6.*	0
			[5]	*Border On 5.*	0
			[4]	*Border On 4.*	0
			[3]	*Border On 3.*	0
			[2]	*Border On 2.*	0
			[1]	*Border On 1.*	0
			[0]	*Border On 0.*	0
0x033C	VO_BAR_MASK_CELL	RW	[15]	*Bar On 15.*	0
			[14]	*Bar On 14.*	0
			[13]	*Bar On 13.*	0
			[12]	*Bar On 12.*	0
			[11]	*Bar On 11.*	0
			[10]	*Bar On 10.*	0
			[9]	*Bar On 9.*	0
			[8]	*Bar On 8.*	0
			[7]	*Bar On 7.*	0
			[6]	*Bar On 6.*	0

			[5]	*Bar On 5.*	0
			[4]	*Bar On 4.*	0
			[3]	*Bar On 3.*	0
			[2]	*Bar On 2.*	0
			[1]	*Bar On 1.*	0
			[0]	*Bar On 0.*	0
0x0340 0x0344 0x0348 0x034C 0x0350	VO_CELL_X0 VO_CELL_X1 VO_CELL_X2 VO_CELL_X3 VO_CELL_X4	RW	[9:0]	Cell Position of X1, X2, X3, X4, X5.	0
0x0354 0x0358 0x035C 0x0360 0x0364	VO_CELL_Y0 VO_CELL_Y1 VO_CELL_Y2 VO_CELL_Y3 VO_CELL_Y4	RW	[9:0]	Cell Position of Y1, Y2, Y3, Y4, Y5.	0
0x0368	VO_CELL_EXT_SET1	RW	[1]	*Ext Border* On1.	0
			[0]	*Ext* Bar On1.	0
0x036C	VO_CELL_EXT_START1	RW	[19:10]	*Ext* Start X1.	0
			[9:0]	*Ext* Start Y1.	0
0x0370	VO_CELL_EXT_END1	RW	[19:10]	*Ext* End X1.	0
			[9:0]	*Ext* End Y1.	0
0x0374	VO_CELL_EXT_SET2	RW	[1]	*Ext* Border On2.	0
			[0]	*Ext* Bar On2.	0
0x0378	VO_CELL_EXT_START2	RW	[19:10]	*Ext* Start X2.	0
			[9:0]	*Ext* Start Y2.	0
0x37C	VO_CELL_EXT_END2	RW	[19:10]	*Ext* End X2.	0
			[9:0]	*Ext* End Y2.	0

3.18. Cursor Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0380	VO_CURSOR_POS	RW	[21:11]	*Start X.*	0
			[10:0]	*Start Y.*	0
0x0384	VO_CURSOR_CLR1	RW	[31:24]	*1st Color Y.*	0x10
			[23:20]	*1st Color U*16.*	0x8
			[19:16]	*1st Color V*16.*	0x8
			[15:8]	*2nd Color Y.*	0x10
			[7:4]	*2nd Color U*16.*	0x8
			[3:0]	*2nd Color V*16.*	0x8
0x0388	VO_CURSOR_CLR2	RW	[15:8]	*3rd Color Y.*	0x10
			[7:4]	*3rd Color U*16.*	0x8
			[3:0]	*3rd Color V*16.*	0x8
0x0390 0x0394 0x0398 0x039C 0x03A0 0x03A4 0x03A8 0x03AC 0x03B0 0x03B4 0x03B8 0x03BC 0x03C0 0x03C4 0x03C8 0x03CC 0x03D0 0x03D4 0x03D8 0x03DC	VO_CUR_MASK1 VO_CUR_MASK2 VO_CUR_MASK3 VO_CUR_MASK4 VO_CUR_MASK5 VO_CUR_MASK6 VO_CUR_MASK7 VO_CUR_MASK8 VO_CUR_MASK9 VO_CUR_MASK10 VO_CUR_MASK11 VO_CUR_MASK12 VO_CUR_MASK13 VO_CUR_MASK14 VO_CUR_MASK15 VO_CUR_MASK16 VO_CUR_MASK17 VO_CUR_MASK18 VO_CUR_MASK19 VO_CUR_MASK20	RW	[31:30]	*Color of 1st Pixel on Horizontal.*	
			[29:28]	*Color of 2nd Pixel on Horizontal.*	
			[27:26]	*Color of 3rd Pixel on Horizontal.*	
			[25:24]	*Color of 4th Pixel on Horizontal.*	
			[23:22]	*Color of 5th Pixel on Horizontal.*	
			[21:20]	*Color of 6th Pixel on Horizontal.*	
			[19:18]	*Color of 7th Pixel on Horizontal.*	
			[17:16]	*Color of 8th Pixel on Horizontal.*	
			[15:14]	*Color of 9th Pixel on Horizontal.*	
			[13:12]	*Color of 10th Pixel on Horizontal.*	
			[11:10]	*Color of 11th Pixel on Horizontal.*	
			[9:8]	*Color of 12th Pixel on Horizontal.*	
			[7:6]	*Color of 13th Pixel on Horizontal.*	
			[5:4]	*Color of 14th Pixel on Horizontal.*	
			[3:2]	*Color of 15th Pixel on Horizontal.*	
			[1:0]	*Color of 16th Pixel on Horizontal.*	

3.19. OSG Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x03E0	VO_OSG_CFG	RW	[31]	*OSG On.*	0
			[28]	*Color Mute.*	0
			[27:22]	*Alpha Blending OSG Strength.*	0
			[21:16]	*Alpha Blending Video Strength.*	0
			[15:0]	*OSG Base Address.*	0
0x03E4	VO_ERASE	RW	[31:24]	*OSG Erase On.*	0
0x03E8	VO_OSG_BLINK	RW	[15:8]	*OSG Blink On.*	0
			[7:4]	*OSG Blink Interval.* [0] : 15 frame [1] : 18 frame	0

3.20. Video Capture Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0400	CAP_BASE	RW	[31:16]	*Maximum Page for Capture.*	0
			[15:0]	*Capture Base Address.*	0
0x0404	CAP_BTW	RW	[13:8]	*Bandwidth Value for Progressive Video.*	0
			[7:0]	*Maximum Bandwidth.*	0
0x0408	CAP_DIM_SCALE1	RW	[23:16]	*Vertical Picture Size for Frame1.*	
			[15:8]	*Vertical Picture Size for Field1.*	
			[7:0]	*Horizontal Picture Size1.*	
0x040C	CAP_DIM_SCALE2	RW	[23:16]	*Vertical Picture Size for Frame2.*	
			[15:8]	*Vertical Picture Size for Field2.*	
			[7:0]	*Horizontal Picture Size2.*	
0x0410	CAP_DIM_SCALE3	RW	[23:16]	*Vertical Picture Size for Frame3.*	
			[15:8]	*Vertical Picture Size for Field3.*	
			[7:0]	*Horizontal Picture Size3.*	
0x0414	CAP_DIM_SCALE4	RW	[23:16]	*Vertical Picture Size for Frame4.*	
			[15:8]	*Vertical Picture Size for Field4.*	
			[7:0]	*Horizontal Picture Size4.*	
0x0418	CAP_DIM_SCALE5	RW	[23:16]	*Vertical Picture Size for Frame5.*	
			[15:8]	*Vertical Picture Size for Field5.*	
			[7:0]	*Horizontal Picture Size5.*	
0x041C	CAP_DIM_PROG	RW	[31:24]	*Vertical Picture Size for 1/3 Horizontal Scale.*	
			[23:16]	*Horizontal Picture Size for 1/3 Vertical Scale.*	
			[15:8]	*Vertical Picture Size.*	
			[7:0]	*Horizontal Picture Size.*	
0x0420	CAP_STATUS	RO	[31:28]	*Encoding Req Queue Write Address.*	-
			[27:24]	*Encoding Req Queue Read Address.*	-
			[21:16]	*Current Bandwidth Value.*	-
			[15:0]	*Vertical Page Offset.*	-

3.21. MPEG4 Video Capture Control Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0440 0x0444 0x0448 0x044C 0x0450 0x0454 0x0458 0x045C 0x0460 0x0464 0x0468 0x046C 0x0470 0x0474 0x0478 0x047C	CAP_SCALE0 CAP_SCALE1 CAP_SCALE2 CAP_SCALE3 CAP_SCALE4 CAP_SCALE5 CAP_SCALE6 CAP_SCALE7 CAP_SCALE8 CAP_SCALE9 CAP_SCALE10 CAP_SCALE11 CAP_SCALE12 CAP_SCALE13 CAP_SCALE14 CAP_SCALE15	RW	[3]	*Frame Mode.*	0
			[2:0]	*Scale Mode.* [000] : No Encoding [001] : H=1/1, V=1/1 (Frame/Field Mode) [010] : H=1/2, V=1/1 (Only Field Mode) [011] : H=1/2, V=1/2 (Only Encoding) [100] : H=1/3, V=1/3 (Only Encoding) [101] : H=1/4, V=1/2 (Only Field Mode)	0

0x0480 0x0484 0x0488 0x048C 0x0490 0x0494 0x0498 0x049C 0x04A0 0x04A4 0x04A8 0x04AC 0x04B0 0x04B4 0x04B8 0x04BC	CAP_SCALE_E0 CAP_SCALE_E1 CAP_SCALE_E2 CAP_SCALE_E3 CAP_SCALE_E4 CAP_SCALE_E5 CAP_SCALE_E6 CAP_SCALE_E7 CAP_SCALE_E8 CAP_SCALE_E9 CAP_SCALE_E10 CAP_SCALE_E11 CAP_SCALE_E12 CAP_SCALE_E13 CAP_SCALE_E14 CAP_SCALE_E15	R/W	[0]	*Extended Encoding On.* Extended MPEG4 video encoding channel will be activated by asserting the register.	0
0x04C0 0x04C4 0x04C8 0x04CC 0x04D0 0x04D4 0x04D8 0x04DC 0x04E0 0x04E4 0x04E8 0x04EC 0x04F0 0x04F4 0x04F8 0x04FC	CAP_INTERVAL0 CAP_INTERVAL1 CAP_INTERVAL2 CAP_INTERVAL3 CAP_INTERVAL4 CAP_INTERVAL5 CAP_INTERVAL6 CAP_INTERVAL7 CAP_INTERVAL8 CAP_INTERVAL9 CAP_INTERVAL10 CAP_INTERVAL11 CAP_INTERVAL12 CAP_INTERVAL13 CAP_INTERVAL14 CAP_INTERVAL15	R/W	[9]	*Interval Mode*. [0] : Skip Mode [1] : Drop Mode	0
			[8:0]	*Interval Value*.	0
0x0500 0x0504 0x0508 0x050C 0x0510 0x0514 0x0518 0x051C 0x0520 0x0524 0x0528 0x052C 0x0530 0x0534 0x0538 0x053C	CAP_INTERVALE0 CAP_INTERVALE1 CAP_INTERVALE2 CAP_INTERVALE3 CAP_INTERVALE4 CAP_INTERVALE5 CAP_INTERVALE6 CAP_INTERVALE7 CAP_INTERVALE8 CAP_INTERVALE9 CAP_INTERVALE10 CAP_INTERVALE11 CAP_INTERVALE12 CAP_INTERVALE13 CAP_INTERVALE14 CAP_INTERVALE15	R/W	[9]	*Extended Interval Mode*. [0] : Skip Mode [1] : Drop Mode	0
			[8:0]	*Extended Interval Value*.	0

3.22. MPEG4 Video Encoder Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0610	VE_CFG0	RW	[31]	*Code Control Mode.* [0] : Multi Page Mode [1] : Two Page Mode	0
			[29:24]	*Interrupt Mode.* [0] : Interrupt asserted every frame [1] : Interrupt asserted every 2 frames [2] : Interrupt asserted every 3 frames [3] : Interrupt asserted every 4 frames	0
			[23:16]	*MPEG4 Block Size.* (64kB)	0
			[15:0]	*MPEG4 Block Base Address.* (64kB)	0
0x0614	VE_CFG1	RW	[27:24]	*Byte aligns.* [0] : 8 [1] : 16 [2] : 32 [3] : 64	0
			[18]	*Index Insertion.*	0

			[17:16]	*Motion Flag Data Insertion.* [0] : Off [1] : After index 256 bytes [2] : User Setting.	0
			[15:0]	*Motion Flag Data Base Address.*	0
0x061C	VE_WMARK_POLY	RW	[31:0]	*Watermark Generic Polynomial.*	0
0x0620	VE_WMARK_KEY	RW	[31:0]	*Watermark Initial Key Value.*	0
0x0624	VE_WMARK_CTRL	RW	[3:0]	*Watermark Strength.*	0
0x0628	VE_ENCR_POLY	RW	[31:0]	*Encryption Generic Polynomial.*	0
0x062C	VE_ENCR_KEY	RW	[31:0]	*Encryption Initial Key Value.*	0
0x0630	VE_ATTR	RW	[31]	*Byte Order.*	0
			[30]	*VOP Round.*	0
			[29:27]	*VOP F. Code.*	0
			[26:25]	*VOP Time Increment.* [00] : 60Hz (500) for NTSC field mode [01] : 30Hz (1000) for NTSC frame mode [10] : 50Hz (600) for PAL field mode [11] : 25Hz (1200) for PAL frame mode	0
			[24:21]	*VOP Time Increment Bit Width.*	0
			[20:16]	*DCT Block Interval.*	0

3.23. MPEG4 Video Encoder Status Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0640	VE_STATUS0	RW	[31]	*Progressive.*	0
			[30]	*Interlace.*	0
			[29]	*Source Field Index.*	0
			[28:24]	*Video Channel.*	0
			[23:22]	*VOP Type.*	0
			[21]	*Video Motion Flag.*	0
			[20]	*SAD Motion Flag.*	0
			[19:0]	*MPEG4 Encoded Code Size.*	0
0x0644	VE_STATUS1	RW	[31:28]	*Scale.*	0
			[19:16]	*Last Queue Position.*	0
			[15:8]	*Horizontal Picture Size.*	0
			[7:0]	*Vertical Picture Size.*	0
0x0648	VE_STATUS2	RW	[31:0]	*MPEG4 Code Data Offset.*	0
0x064C	VE_STATUS3	RW	[31:0]	*JPEG Code Data Offset.*	0
0x0650	VE_STATUS4	RW	[29:20]	*Interval.*	0
			[19:0]	*JPEG Code Size.*	0
0x0654	VE_STATUS5	RW	[31:0]	*Time.* (sec)	0
0x0658	VE_STATUS6	RW	[31:0]	*Time.* (usec)	0
0x065C	VE_STATUS7	RW	[31:0]	*Watermark Status.*	0
0x0660	VE_STATUS8	RW	[31:0]	*Encryption Status.*	0
0x0664	VE_STATUS9	RW	[4:0]	*Channel.*	0
0x0668	VE_STATUS10	RW	[19:0]	*MPEG4 Code Size.*	0
0x066C	VE_STATUS11	RW	[7:0]	*Last Queue Position*	0

3.24. Motion JPEG Encoder Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0670	VE_JPEG_QS	RW	[28:24]	*Quantization Step3.*	0
			[20:16]	*Quantization Step2.*	0
			[12:8]	*Quantization Step1.*	0
			[4:0]	*Quantization Step0.*	0
0x0674	VE_JPEG_QS_CH0	RW	[31:30]	*Quantization Step Table ID for Channel15.*	0
			[29:28]	*Quantization Step Table ID for Channel14.*	0
			[27:26]	*Quantization Step Table ID for Channel13.*	0
			[25:24]	*Quantization Step Table ID for Channel12.*	0
			[23:22]	*Quantization Step Table ID for Channel11.*	0
			[21:20]	*Quantization Step Table ID for Channel10.*	0
			[19:18]	*Quantization Step Table ID for Channel9.*	0
			[17:16]	*Quantization Step Table ID for Channel8.*	0
			[15:14]	*Quantization Step Table ID for Channel7.*	0
			[13:12]	*Quantization Step Table ID for Channel6.*	0

			[11:10]	*Quantization Step Table ID for Channel5.*	0
			[9:8]	*Quantization Step Table ID for Channel4.*	0
			[7:6]	*Quantization Step Table ID for Channel3.*	0
			[5:4]	*Quantization Step Table ID for Channel2.*	0
			[3:2]	*Quantization Step Table ID for Channel1.*	0
			[1:0]	*Quantization Step Table ID for Channel0.*	0
0x0678	VE_JPEG_QS_CH1	RW	[31:30]	*Extended Quantization Step Table ID for Channel31.*	0
			[29:28]	*Extended Quantization Step Table ID for Channel30.*	0
			[27:26]	*Extended Quantization Step Table ID for Channel29.*	0
			[25:24]	*Extended Quantization Step Table ID for Channel28.*	0
			[23:22]	*Extended Quantization Step Table ID for Channel27.*	0
			[21:20]	*Extended Quantization Step Table ID for Channel26.*	0
			[19:18]	*Extended Quantization Step Table ID for Channel25.*	0
			[17:16]	*Extended Quantization Step Table ID for Channel24.*	0
			[15:14]	*Extended Quantization Step Table ID for Channel23.*	0
			[13:12]	*Extended Quantization Step Table ID for Channel22.*	0
			[11:10]	*Extended Quantization Step Table ID for Channel21.*	0
			[9:8]	*Extended Quantization Step Table ID for Channel20.*	0
			[7:6]	*Extended Quantization Step Table ID for Channel19.*	0
			[5:4]	*Extended Quantization Step Table ID for Channel18.*	0
			[3:2]	*Extended Quantization Step Table ID for Channel17.*	0
			[1:0]	*Extended Quantization Step Table ID for Channel16.*	0
0x067C	VE_JPEG_CFG	RW	[23:16]	*JPEG Page Size.*	0
			[15:0]	*JPEG Page Base.*	
0x0680	VE_JPEG_CTRL	RW	[31:0]	*JPEG Code Enable for Channel 31~0.*	0

3.25. Video Encoder OSD Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0690	VE_OSD_CH	RW	[31:0]	*Record OSG Enable.*	0
0x0694	VE_OSD_BASE	RW	[15:0]	*Record OSG Base Address.*	0
0x0698	VE_OSD_CLR	RW	[23:16]	*Record OSG Y.*	0
			[15:8]	*Record OSG U.*	0
			[7:0]	*Record OSG V.*	0
0x069C	VE_OSD_OPT	RW	[15]	*Record OSG Horizontal Shadow.*	0
			[14]	*Record OSG Vertical Shadow.*	0
			[13:7]	*Record OSG Horizontal Offset * 16.*	0
			[6:0]	*Record OSG Vertical Offset * 16.*	0

3.26. Video Encoder Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0700 0x0704 0x0708 0x070C 0x0710 0x0714 0x0718 0x071C 0x0720 0x0724 0x0728 0x072C 0x0730 0x0734 0x0738 0x073C	VE_CH_INTL0 VE_CH_INTL1 VE_CH_INTL2 VE_CH_INTL3 VE_CH_INTL4 VE_CH_INTL5 VE_CH_INTL6 VE_CH_INTL7 VE_CH_INTL8 VE_CH_INTL9 VE_CH_INTL10 VE_CH_INTL11 VE_CH_INTL12 VE_CH_INTL13 VE_CH_INTL14 VE_CH_INTL15	RW	[1]	*Intra Picture Based Prediction.*	0
			[0]	*Interlace.*	0

0x0740	VE_CH_MOT0	RW	[15:0]	*Motion Threshold.*	0
0x0744	VE_CH_MOT1				
0x0748	VE_CH_MOT2				
0x074C	VE_CH_MOT3				
0x0750	VE_CH_MOT4				
0x0754	VE_CH_MOT5				
0x0758	VE_CH_MOT6				
0x075C	VE_CH_MOT7				
0x0760	VE_CH_MOT8				
0x0764	VE_CH_MOT9				
0x0768	VE_CH_MOT10				
0x076C	VE_CH_MOT11				
0x0770	VE_CH_MOT12				
0x0774	VE_CH_MOT13				
0x0778	VE_CH_MOT14				
0x077C	VE_CH_MOT15				
0x0780	VE_CH_QP0	RW	[4:0]	*Quantization Parameter.*	0
0x0784	VE_CH_QP1				
0x0788	VE_CH_QP2				
0x078C	VE_CH_QP3				
0x0790	VE_CH_QP4				
0x0794	VE_CH_QP5				
0x0798	VE_CH_QP6				
0x079C	VE_CH_QP7				
0x07A0	VE_CH_QP8				
0x07A4	VE_CH_QP9				
0x07A8	VE_CH_QP10				
0x07AC	VE_CH_QP11				
0x07B0	VE_CH_QP12				
0x07B4	VE_CH_QP13				
0x07B8	VE_CH_QP14				
0x07BC	VE_CH_QP15				
0x07C0	VE_CH_QP_E0	RW	[4:0]	*Extended Quantization Parameter.*	0
0x07C4	VE_CH_QP_E1				
0x07C8	VE_CH_QP_E2				
0x07CC	VE_CH_QP_E3				
0x07D0	VE_CH_QP_E4				
0x07D4	VE_CH_QP_E5				
0x07D8	VE_CH_QP_E6				
0x07DC	VE_CH_QP_E7				
0x07E0	VE_CH_QP_E8				
0x07E4	VE_CH_QP_E9				
0x07E8	VE_CH_QP_E10				
0x07EC	VE_CH_QP_E11				
0x07F0	VE_CH_QP_E12				
0x07F4	VE_CH_QP_E13				
0x07F8	VE_CH_QP_E14				
0x07FC	VE_CH_QP_E15				
0x0800	VE_CH_GOP0	RW	[7:0]	*GOP Size.*	0
0x0804	VE_CH_GOP1				
0x0808	VE_CH_GOP2				
0x080C	VE_CH_GOP3				
0x0810	VE_CH_GOP4				
0x0814	VE_CH_GOP5				
0x0818	VE_CH_GOP6				
0x081C	VE_CH_GOP7				
0x0820	VE_CH_GOP8				
0x0824	VE_CH_GOP9				
0x0828	VE_CH_GOP10				
0x082C	VE_CH_GOP11				
0x0830	VE_CH_GOP12				
0x0834	VE_CH_GOP13				
0x0838	VE_CH_GOP14				
0x083C	VE_CH_GOP15				

0x0840	VE_CH_GOP_E0	RW	[7:0]	*Extended GOP Size.*	0
0x0844	VE_CH_GOP_E1				
0x0848	VE_CH_GOP_E2				
0x084C	VE_CH_GOP_E3				
0x0850	VE_CH_GOP_E4				
0x0854	VE_CH_GOP_E5				
0x0858	VE_CH_GOP_E6				
0x085C	VE_CH_GOP_E7				
0x0860	VE_CH_GOP_E8				
0x0864	VE_CH_GOP_E9				
0x0868	VE_CH_GOP_E10				
0x086C	VE_CH_GOP_E11				
0x0870	VE_CH_GOP_E12				
0x0874	VE_CH_GOP_E13				
0x0878	VE_CH_GOP_E14				
0x087C	VE_CH_GOP_E15				
0x0880	VE_CH_REFB0	RW	[15:0]	*Reference Base Address.*	0
0x0884	VE_CH_REFB1				
0x0888	VE_CH_REFB2				
0x088C	VE_CH_REFB3				
0x0890	VE_CH_REFB4				
0x0894	VE_CH_REFB5				
0x0898	VE_CH_REFB6				
0x089C	VE_CH_REFB7				
0x08A0	VE_CH_REFB8				
0x08A4	VE_CH_REFB9				
0x08A8	VE_CH_REFB10				
0x08AC	VE_CH_REFB11				
0x08B0	VE_CH_REFB12				
0x08B4	VE_CH_REFB13				
0x08B8	VE_CH_REFB14				
0x08BC	VE_CH_REFB15				
0x08C0	VE_CH_REFB_E0	RW	[15:0]	*Extended Reference Base Address.*	0
0x08C4	VE_CH_REFB_E1				
0x08C8	VE_CH_REFB_E2				
0x08CC	VE_CH_REFB_E3				
0x08D0	VE_CH_REFB_E4				
0x08D4	VE_CH_REFB_E5				
0x08D8	VE_CH_REFB_E6				
0x08DC	VE_CH_REFB_E7				
0x08E0	VE_CH_REFB_E8				
0x08E4	VE_CH_REFB_E9				
0x08E8	VE_CH_REFB_E10				
0x08EC	VE_CH_REFB_E11				
0x08F0	VE_CH_REFB_E12				
0x08F4	VE_CH_REFB_E13				
0x08F8	VE_CH_REFB_E14				
0x08FC	VE_CH_REFB_E15				
0x0A00	VE_MPEG4_QUE	RW	[31]	*Motion Flag.*	0
0x0A08			[30:29]	*VOP Type.*	0
0x0A10			[28:24]	*Channel.*	0
0x0A18			[23:0]	*MPEG4 Code Address Offset.*	0
0x0A20					
0x0A28					
0x0A30					
0x0A38					
0x0A40					
0x0A48					
0x0A50					
0x0A58					
0x0A60					
0x0A68					
0x0A70					
0x0A78					

0x0A04 0x0A0C 0x0A14 0x0A1C 0x0A24 0x0A2C 0x0A34 0x0A3C 0x0A44 0x0A4C 0x0A54 0x0A5C 0x0A64 0x0A6C 0x0A74 0x0A7C	VE_JPEG_QUE	RW	[23:0]	*JPEG Code Address Offset.*	0

3.27. MPEG4 Video Decoder Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0900	VD_CFG0	RW	[24]	*No Write if No Window.*	0
			[23]	*Busy Wait When Code Busy.*	0
			[22]	*Busy Wait When Res Write.*	0
			[21]	*Busy Wait When Ref Read.*	0
			[20]	*Busy Wait When Macroblock Read.*	0
			[18]	*Decode Mode* [0] : Multi [1] : Single	0
			[17]	*Display Control Mode.* [0] : Auto [1] : User	0
			[16]	*Display Page Control Mode.* [0] : 2 page auto [1] : User	0
			[15]	*Byte Order.*	0
			[14]	*Decode Start Field Index.*	0
			[13]	*Decode Lock When Error.*	0
			[12]	*Error Interrupt Enable.*	0
			[11:8]	*Time Width.*	0
			[7:0]	*DCT Interval.*	0

3.28. MPEG4 Video Decoder Control Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x090C	VD_CODE_ADDR	RW	[31:0]	*Code Address.*	0
0x0910	VD_CTRL	RW	[31]	*Decoder On.*	0
			[11:0]	*Max Item Count When Multiple Decode Mode.*	0
0x0920	VD_STATUS0	RO	[22]	*Interrupt by Command Ack.*	-
			[21]	*Interrupt by Command Empty.*	-
			[20]	*Interrupt by Decode Error.*	-
			[11:0]	*Rest Number of Reserved Decoding.* Only for multiple Decode Mode.	-
0x0924	VD_STATUS1	RO	[23:20]	*Internal VLD Error Flags.*	-
			[19:0]	*Decoded Byte Count.*	
0x0930	VD_IDX0	RW	[31]	*Progressive.*	0
			[30]	*Interlace.*	0
			[29]	*Source FI.*	0
			[27:24]	*Source Channel ID.*	0
			[23:22]	*VOP Type.*	0
			[21]	*Decode Page Stop.*	0
			[20]	*Sync Start.*	0
			[19:0]	*MPEG4 Code Size.*	0
0x0934	VD_IDX1	RW	[31:28]	*Source Scale.*	0
			[27:24]	*Target Window ID.*	0
			[16]	*Frame Interpolation On.*	0

			[15:8]	*Horizontal Macroblock Size.*	0
			[7:0]	*Vertical Macroblock Size.*	0
0x0938	VD_IDX2	RW	[31]	*Ref Address Horizontal Offset.* [0] : 0 [1] : 512 (When use CIF)	0
			[15:0]	*Ref Base Address.*	0
0x093C	VD_IDX3	RW	[31:28]	*User Set Macro Scale Mode.*	0
			[27]	*User Set Macro Interlace Write.*	0
			[26]	*User Set Macro Read Interpolation.*	0
			[25]	*User Set Display Horizontal Zoom.*	0
			[24]	*Reserved.*	0
			[23:12]	*User Set Display SX.*	0
			[11:0]	*User Set Display SY.*	0
0x0940	VD_IDX4	RW	[7:0]	*User Set Decode Write Page.*	0
			[2]	*User Set Display Read Page.*	0
0x0944	VD_IDX5	RW	[31:0]	*Next Code Address*	0

3.29. GPIO Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0B00	GPIO_CONFIG0	RW	[31:30]	*GPIO[15] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL7 [3] : RS_nRTS1	0
			[29:28]	*GPIO[14] Configuration*. [0] : Input [1] : Output [2] : I2C_SDA7 [3] : RS_nCTS1	0
			[27:26]	*GPIO[13] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL6 [3] : RS_TXD1	0
			[25:24]	*GPIO[12] Configuration*. [0] : Input [1] : Output [2] : I2C_SDA6 [3] : RS_nRXD1	0
			[23:22]	*GPIO[11] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL5 [3] : RS_TxEn0	0
			[21:20]	*GPIO[10] Configuration*. [0] : Input [1] : Output [2] : I2C_SDA5 [3] : RS_RxEn0	0
			[19:18]	*GPIO[9] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL4 [3] : RS_TxD0	0
			[17:16]	*GPIO[8] Configuration*. [0] : Input [1] : Output [2] : I2C_SDA4 [3] : RS_RxD0	0
			[15:14]	*GPIO[7] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL3 [3] : PS2_C1	0

			[13:12]	*GPIO[6] Configuration*. [0] : Input [1] : Output [2] : I2C_SDA3 [3] : PS2_D1	0
			[11:10]	*GPIO[5] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL2 [3] : PS2_C0	0
			[9:8]	*GPIO[4] Configuration*. [0] : Input [1] : Output [2] : I2C_SDA2 [3] : PS2_D0	0
			[7:6]	*GPIO[3] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL1 [3] : SPI_STB	0
			[5:4]	*GPIO[2] Configuration*. [0] : Input [1] : Output [2] : I2C_SDA1 [3] : SPI_CLK	0
			[3:2]	*GPIO[1] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL0 [3] : SPI_SDI	0
			[1:0]	*GPIO[0] Configuration*. [0] : Input [1] : Output [2] : I2C_SDA0 [3] : SPI_SDO	0
0x0B04	GPIO_ CONFIG1	RW	[31:16]	*GPIO[31:16] Enable*. [0] : ATA DIO[15:0] [1] : GPIO[31:16]	0
			[15]	*GPIO[31] Configuration*. [0] : Input [1] : Output	0
			[14]	*GPIO[30] Configuration*. [0] : Input [1] : Output	0
			[13]	*GPIO[29] Configuration*. [0] : Input [1] : Output	0
			[12]	*GPIO[28] Configuration*. [0] : Input [1] : Output	0
			[11]	*GPIO[27] Configuration*. [0] : Input [1] : Output	0
			[10]	*GPIO[26] Configuration*. [0] : Input [1] : Output	0
			[9]	*GPIO[25] Configuration*. [0] : Input [1] : Output	0
			[8]	*GPIO[24] Configuration*. [0] : Input [1] : Output	0
			[7]	*GPIO[23] Configuration*. [0] : Input [1] : Output	0
			[6]	*GPIO[22] Configuration*. [0] : Input [1] : Output	0

			[5]	*GPIO[21] Configuration*. [0] : Input [1] : Output	0
			[4]	*GPIO[20] Configuration*. [0] : Input [1] : Output	0
			[3]	*GPIO[19] Configuration*. [0] : Input [1] : Output	0
			[2]	*GPIO[18] Configuration*. [0] : Input [1] : Output	0
			[1]	*GPIO[17] Configuration*. [0] : Input [1] : Output	0
			[0]	*GPIO[16] Configuration*. [0] : Input [1] : Output	0
0x0B08	GPIO_DATA_OUT	RW	[31:0]	*GPIO Output Data*.	0xffff_ffff
0x0B0C	GPIO_DATA_IN	RW	[31:0]	*GPIO Input Data*.	-
0x0B10	GPIO_INT_STA/ACK	RW	[31:0]	*GPIO Interrupt Status/Ack.* Read Mode : GPIO[31:0] Interrupt Status. Write Mode : GPIO[31:0] Interrupt Ack.	0
0x0B14	GPIO_INT_ENA	RW	[31:0]	*GPIO Interrupt Enable*. GPIO[31:0] Interrupt Enable.	0
0x0B18	GPIO_INT_CFG0	RW	[31:30]	*GPIO[15] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[29:28]	*GPIO[14] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[27:26]	*GPIO[13] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[25:24]	*GPIO[12] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[23:22]	*GPIO[11] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[21:20]	*GPIO[10] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[19:18]	*GPIO[9] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[17:16]	*GPIO[8] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[15:14]	*GPIO[7] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0

			[13:12]	*GPIO[6] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[11:10]	*GPIO[5] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[9:8]	*GPIO[4] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[7:6]	*GPIO[3] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[5:4]	*GPIO[2] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[3:2]	*GPIO[1] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[1:0]	*GPIO[0] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
0x0B1C	GPIO_INT_CFG1	RW	[31:30]	*GPIO[31] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[29:28]	*GPIO[30] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[27:26]	*GPIO[29] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[25:24]	*GPIO[28] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[23:22]	*GPIO[27] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[21:20]	*GPIO[26] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[19:18]	*GPIO[25] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0

			[17:16]	*GPIO[24] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[15:14]	*GPIO[23] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[13:12]	*GPIO[22] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[11:10]	*GPIO[21] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[9:8]	*GPIO[20] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[7:6]	*GPIO[19] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[5:4]	*GPIO[18] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[3:2]	*GPIO[17] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[1:0]	*GPIO[16] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0

3.30. I2C Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0B20	IIC_CFG	RW	[8]	*I2C Module Enable*.	0
			[7:0]	*Clock Pre-Scale*. (SYS_CLK/5) / (PR[7:0] * 16+1)	8
0xB24	IIC_CTRL	RO	[20]	*Auto Clear Request*. The register sets to “1” to clear START, STOP, READ and WRITE.	-
			[19]	*Received Acknowledge from Slave*. [0] : ACK received [1] : ACK not received	-
			[18]	*I2C bus Busy*. [0] : after STOP signal detected [1] : after START signal detected	-
			[17]	*Arbitration Lost*. a) STOP Signal is detected, but none requested. b) The master drives SDA high, but SDA low.	-
			[16]	*Transfer in Progress*. [0] : Complete [1] : Transferring	-

		RW	[7:5]	*Channel Select*. [000] : Channel 0 [001] : Channel 1 [010] : Channel 2 [011] : Channel 3 [100] : Channel 4 [101] : Channel 5 [110] : Channel 6 [111] : Channel 7	0
			[4]	*ACK*. It should be set to appropriate value if receiver is set to ACK=0 or ACK=1.	0
			[3]	*START*. The register is set or reset by user or reset by Auto Clear Request.	0
			[2]	*STOP*. The register is set or reset by user or reset by Auto Clear Request.	0
			[1]	*READ*. The register is set or reset by user or reset by Auto Clear Request.	0
			[0]	*WRITE*. The register is set or reset by user or reset by Auto Clear Request.	0
0x0B28	IIC_TXD	RW	[7:0]	*Tx Data*.	0
0x0B2C	IIC_RXD	RO	[7:0]	*Rx Data*.	-

3.31. SPI Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0B40	SPI_TXD0	R/W	[31:0]	*SPI Tx Data[31:0]*.	0
0x0B44	SPI_TXD1	R/W	[31:0]	*SPI Tx Data[63:32]*.	0
0x0B48	SPI_TXD2	R/W	[15:0]	*SPI Tx Data[79:64]*.	0
0x0B50	SPI_RXD0	R	[31:0]	*SPI Rx Data[31:0]*.	-
0x0B54	SPI_RXD1	R	[31:0]	*SPI Rx Data[63:32]*.	-
0x0B58	SPI_RXD2	R	[15:0]	*SPI Rx Data[79:64]*.	-

3.32. PS2 Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0B60 0x0B80	PS2_CFG0 PS2_CFG1	RW	[31:24]	*PS2 Tx Timeout Limitation.* Sampling Period * (n+1)	0x10
		RW	[23:16]	*PS2 Rx Timeout Limitation.* Sampling Period * (n+1)	0x08
		RW	[15:8]	*PS2 Tx Clock Holding Time.* Sampling Period * (n+1)	0x10
		RW	[6:4]	*PS2* Sampling Period. Clock Dividing : 2^(n+6) * SYS_CLK period Where, 0 <= n <= 5	0x04
		RW	[3]	*PS2 Tx Data Interrupt Enable.*	0
		RW	[2]	*PS2 Rx Data Interrupt Enable.*	0
		RW	[1]	*PS2 Error Interrupt Enable.*	0
		RW	[0]	*PS2 Enable.*	0
0x0B64 0x0B84	PS2_STA0 PS2_STA1	RO	[26:24]	*PS2 Tx Buffer Status.* [0] : Full [4] : Empty	-
		RO	[23]	*PS2 Tx Data End.*	-
		RO	[22]	*PS2 Tx Buffer Full.*	-
		RO	[21]	*PS2 Tx Busy.*	-
		RO	[20]	*PS2 Tx Timeout.* Cleared by Core Reset.	-
		RO	[19:16]	*PS2 Tx Status of FSM.*	-
		RO	[15:10]	*PS2 Rx Buffer Status.* [0] : Full [4] : Empty	-

		RO	[9]	*PS2 Rx Data End.*	-
		RO	[8]	*PS2 Rx Buffer Empty.*	-
		RO	[7]	*PS2 Rx Busy.*	-
		RO	[6]	*PS2 Rx Timeout.* Cleared by Core Reset.	-
		RO	[5]	*PS2 Rx Overflow.* Cleared by Core Reset.	-
		RO	[4]	*PS2 Rx Parity Error.* Cleared by Core Reset.	-
		RO	[3:0]	*PS2 Rx Status of FSM.*	-
0x0B68 0x0B88	PS2_TXD0 PS2_TXD1	RW	[8]	*PS2 Tx Push.*	0
		RW	[7:0]	*PS2 Tx Data[7:0].*	0
0x0B6C 0x0B8C	PS2_RXD0 PS2_RXD1	RW	[8]	*PS2 Rx Push.*	0
		RO	[7:0]	*PS2 Rx Data[7:0].*	-

3.33. UART Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0BA0 0x0BC0	UART_CTRL0 UART_CTRL1	RW	[28:24]	*Divide Number*. [0x07] : System Clock = 27MHz [0x09] : System Clock = 33MHz [0x0f] : System Clock = 54MHz (Default) [0x13] : System Clock = 66MHz	0
			[20]	*CTS & RTS Enable*.	0
			[18]	*Parity Error Data Drop Enable*.	0
			[17]	*Error Interrupt Enable*.	0
			[16]	*Rx Interrupt Enable*.	0
			[15]	*Tx Interrupt Enable*.	0
			[14]	*Rx Module Enable*.	0
			[13]	*Tx Module Enable*.	0
			[12]	*Half Duplex*.	0
			[11]	*Loop Back*.	0
			[10:9]	*Baud Rate1*. The register is only effective when Baud Rate0 register is set to [110]. [00] : 4800 [01] : 2400 [10] :1200 [11] : 300	0
			[8:6]	*Baud Rate0*. [000] : 230400 [001] : 115200 [010] :57600 [011] : 38400 [100] : 19200 [101] : 9600 [110] : below 9600	0
			[5:4]	*Data Size*. [00] : 5 bit [01] : 6 bit [10] : 7 bit [11] : 8 bit	0
			[3:2]	*Stop Bit Size*. [00] : 1 bit [01] : 2 bit [10], [11]: 1.5 bit	0
			[1:0]	*Parity Mode*. [00], [01] : NONE [10] : Even [11] : Odd	0
0x0BA4 0x0BC4	UART_STA0 UART_STA1	RO	[15]	*CTS*.	-
			[14]	*Rx Busy*.	-
			[13]	*Overrun.* (Rx Buffer Full & Rx Frame Input)	-
			[12]	*Frame Error*.	-
			[11]	*Parity Error*.	-

			[10:6]	*Rx Buffer State*. The register indicates the number of remained data in Rx Buffer. Zero means empty.	-
			[5]	*Tx Line Busy*.	-
			[4:0]	*Tx Buffer State*. The register indicates the number of remained data in Tx Buffer. Zero means full.	8
0x0BA8 0x0BC8	UART_TXD0 UART_TXD1	RW	[7:0]	*Tx Data*. Push signal is generated automatically when CPU writes Tx data.	0
0x0BAC 0x0BCC	UART_RXD0 UART_RXD1	RO	[7:0]	*Rx Data*. Pop Signal is generated automatically when CPU reads Rx data.	-

3.34. Timer Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0BE0	TIMER_CLK_NUM	RW	[8:0]	Clock Number for uSec. (SYS_CLK / 1MHz) -1	107
0x0BE4	TIMER_WATCHDOG	RW	[8]	*Watchdog Enable.*	1
			[7:0]	*Watchdog Second Count.* Write : Set Read : Status	0xff
0x0BE8	TIMER_USEC	RW	[9:0]	*Watchdog Second Count.* Write : Set Read : Status	107999
0x0BEC	TIMER_SEC	RW	[31:0]	*Watchdog Second Count.* Write : Set Read : Status	0

3.35. ATA Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0C00	ATA_CMD	RW	[31]	*Macro Mode Enable*.	0
			[30]	*Slave Mode.* [0] : Master Mode [1] : Slave Mode	0
			[29]	*Write Enable.*	0
			[28]	*Macro Command Skip.*	0
			[27:26]	*Mode.* [0] : DMA (PIO4 Protocol) [3] : Direct	0
			[25:24]	*Multi Selector.* [0] : 8 [1] : 16 [2] : 24 [3] : 128	0
			[23]	*IDE Bus Selection.* [0] : CS0, CS1 [1] : CS2, CS3	0
			[22]	*ATA Path.* [0] : PCI to ATA [1] : FDMA to ATA	0
			[21]	*48Bit LBA Mode.*	0
			[20:19]	*Clock Divide.* [0] : x1 [1] : x2 [2] : x4 [3] : x8	0
			[18]	*Slow Data.* Doubling PM_LEN, PM_TNS, PM_LTH values.	0
			[17:16]	*Timeout.* [0] : Off [1] : 32M clocks [2] : 64M clocks [3] : 96M clocks	0

			[15:0]	*Sector Counter.*	0
0x0C04	ATA_TIMING_CFG	RW	[31:28]	CMD_CS_LEN. Length of CS0, CS.	0xf
			[27:24]	CMD_CS_FALL. Falling Point of CS0, CS1 (at CS_LEN rising edge)	0x2
			[23:20]	CMD_RW_FALL. Falling Point of RD, WR.	0x9
			[19:16]	CMD_RW_RISE. Rising Point of RD, WR.	0xd
			[15:12]	CMD_RD_LTH. Storing Point of DIO (Input Data).	0xb
			[11:8]	DAT_PM_LEN. Period of RD, WR.	0x7
			[7:4]	DAT_PM_TNS. Rising Point of RD, WR	0x6
			[3:0]	DAT_PM_LTH. Storing Point of DIO (Input Data)	0x6
0x0C08	ATA_TAR_ADR	RW	[31:0]	PCI Target Start Address.	0
0x0C0C	ATA_EXT_ADR	RW	[31:0]	*External Memory Start Address.*	0
0x0C10	ATA_STR_SEC0	RW	[31:0]	*ATA Start Sector[31:0].*	0
0x0C14	ATA_STR_SEC1	RW	[31:0]	*ATA Start Sector[47:32]*	0
0x0C18	ATA_DIR_STA	RO	[31]	*Timeout.*	-
			[30]	HDD Error State.	-
			[29]	Command End.	-
			[28:26]	Reserved.	-
			[25]	INT pin. (IDE BUS 1)	-
			[24]	INT pin. (IDE BUS 0)	-
		WO	[23]	RESET pin.	-
			[22]	READ pin.	-
			[21]	WRITE pin.	-
			[20]	CS1 pin.	-
			[19]	CS0 pin.	-
			[18]	ADR2 pin.	-
			[17]	ADR1 pin.	-
			[16]	ADR0 pin.	-
			[15:0]	DATA bus.	-
0x0C1C	ATA_FSM_STA	RO	[18:13]	DMA IO Status.	-
			[12:9]	DMA Control Status.	-
			[8:4]	Command Status.	-
			[3:0]	Control Status.	-

3.36. Audio Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0D00	ADO_CTRL0	RW	[31]	*Audio CODEC Enable*.	0
			[30]	*Master Mode.* [0] : CLK/SYNC IN [1] : CLK/SYNC OUT	0
			[29]	*Interface Mode.* [0] : u-Law [1] : I2S	0
			[28]	*Reserved.*	0
			[27]	*Left-Right Word Swap*. Only for I2S mode.	0
			[26]	*Bit Width*. Only for I2S. [0] : 16bit [1] : 8bit	0
			[25:24]	*Multi Channel*. Only for I2S slave mode. [0] : 2 Channel [1] : 4 Channel [2] : 8 Channel [3] : 16 Channel	0
			[23]	*Mix Enable for Ch9 ~ Ch0*.	0
			[22:20]	*Decoding Volume Control for Ch9 ~ Ch0.*	0
			[19]	*Mix Enable for Ch19 ~ Ch10*.	0
			[18:16]	*Decoding Volume Control for Ch19 ~ Ch10.*	0

			[9:0]	*Channel I/O Mode.* PCM mode : [Ch9, Ch8, Ch7, …, Ch0] Ch(x) = 0 : Encoding Ch(x) = 1 : Decoding I2S mode : [{Ch19, Ch18}, {Ch17, Ch16}, {Ch15, Ch14}, …, {Ch1, Ch0}] {Ch(x1),Ch(x0)} = 0 : Encoding {Ch(x1),Ch(x0)} = 1 : Decoding	0
0x0D04	ADO_SAMPLING	RW	[31]	*Short Mode for Simulation*. Only for Test.	0
			[30]	*EE Mode for Testing*. Only for Test.	0
			[29:25]	*EE Mode Encoding Channel for Testing*. Only for Test	0
			[24:16]	*Serial Clock Number*. (Only for I2S Master Mode) M (Clock Frequency / Sampling)	0x18
			[8:0]	*Clock Divider*. (Only for I2S Master Mode) N (Clock Frequency = SYS_CLK / (N*2) Hz)	0x1A5
0x0D08	ADO_FDMA_INTR	RW	[31:19]	*DMA Wait Interval.*	
			[18:16]	*Interrupt Page Count Mode.* The range will be allowed 0~4. 2^n pages will be applied.	
			[15:0]	*External Memory Base Address*.	0
0x0D0C	AUDIO_EVOL0	RW	[29:27]	*Encoding Volume Control for Ch9*. [000] : x1 [100] : x1 [001] : x2 [101] : x1/2 [010] : x4 [110] : x1/4 [011] : x8 [111] : x1/8	0
			[26:24]	*Encoding Volume Control for C8*.	0
			[23:21]	*Encoding Volume Control for Ch7*.	0
			[20:18]	*Encoding Volume Control for Ch6*.	0
			[17:15]	*Encoding Volume Control for Ch5*.	0
			[14:12]	*Encoding Volume Control for Ch4*.	0
			[11:9]	*Encoding Volume Control for Ch3*.	0
			[8:6]	*Encoding Volume Control for Ch2*.	0
			[5:3]	*Encoding Volume Control for Ch1*.	0
			[2:0]	*Encoding Volume Control for Ch0*.	0
0x0D10	AUDIO_EVOL1	RW	[29:27]	*Encoding Volume Control for Ch19*. [000] : x1 [100] : x1 [001] : x2 [101] : x1/2 [010] : x4 [110] : x1/4 [011] : x8 [111] : x1/8	0
			[26:24]	*Encoding Volume Control for Ch18*.	0
			[23:21]	*Encoding Volume Control for Ch17*.	0
			[20:18]	*Encoding Volume Control for Ch16*.	0
			[17:15]	*Encoding Volume Control for Ch15*.	0
			[14:12]	*Encoding Volume Control for Ch14*.	0
			[11:9]	*Encoding Volume Control for Ch13*.	0
			[8:6]	*Encoding Volume Control for Ch12*.	0
			[5:3]	*Encoding Volume Control for Ch11*.	0
			[2:0]	*Encoding Volume Control for Ch10*.	0
0x0D14	ADO_STA	RW	[31:10]	*Reserved*. (FSM for Debugging.)	0
			[9:5]	*Channel Count.* (Accessing Channel 0 ~ 19)	0
			[4:0]	*Page Count.* (Accessing Page 0 ~ 31)	0

4. Functional Description

4.1. System Signals

4.1.1. System Reset & Clock Sources

The asynchronous system reset pin i_SYS_RST_IN is active LOW. Although the SOLO6010 is designed with full hardwired logic and has no restrictions for the width of system reset pulse to reset internal flip flops, it should be delayed enough to guarantee stable clock sources to initialize the SDRAM with predefined power up sequence and to synchronize internal PLL. 1 msec delay is recommended after power-up. Fig. 4-1 shows the timing diagram of the system clock iSYS_CLK_IN and the asynchronous system reset i_SYS_RST_IN.

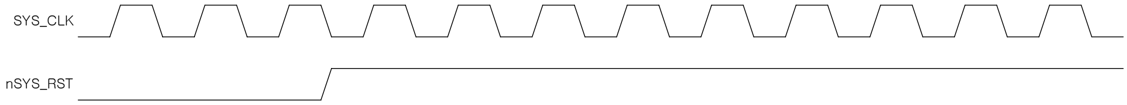


Fig. 4-1 Timing diagram of system reset

The SOLO6010 contains the internal PLL based clock generation module which is doubling input system clock iSYS_CLK_IN. In initial mode, input system clock 27MHz or 54MHz iSYS_CLK_IN will be bypass the PLL, but the output of PLL based clock generation module can be used for the internal system clock by configuring the PLL configuration register [0x0000]. More detail operation of the PLL based clock generation module will be described in the next section.

In normal operation, iSYS_CLK_IN accepts 27MHz or 54MHz frequency oscillator clock sources with 50% HIGH and 50% LOW clock duty and internal 108MHz system clock will be generated by PLL based clock generation module using accepted iSYS_CLK_IN. Most glue logic of the SOLO6010 is designed to be used internal 108MHz system clock (SYS clock), and this feature enables the performance of quadruple speed MPEG4 video CODEC. The video output for the live video output and PCI module is designed to be used 27MHz video output clock (VOUT clock) and 33MHz/66MHz PCI clock (PCI clock), respectively.

Table 4-1 System Clock Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0000	SYS_CONFIG	RW	[31]	*FDMA Reset.* Active HIGH. [0] : Normal operation. [1] : SDRAM controller reset. [1]→[0] : Initializing SDRAM and power-up sequence.	0
			[30]	*SDRAM Bit Width Selection.* [0] : 32bit Interface with SDRAM. [1] : 64bit Interface with SDRAM.	1
			[29:28]	*Reserved.*	0
			[27:26]	*Positive FDMA Clock Delay.* Factory Default.	0
			[25:24]	*Negative FDMA Clock Delay.* Factory Default.	0

			[23:22]	*Reserved.*	0
			[21:20]	*Reserved.*	0
			[19]	*System Clock Divider Control*	0
			[18:14]	*Input 5-bit Divider Control.* (PLL)	0
			[13:5]	*Feedback 9-bit Divider Control.* (PLL)	0
			[4:3]	*Output Divider Control.* (PLL)	0
			[2]	*Power Down Mode.* (PLL)	0
			[1]	*Bypass.* (PLL)	1
			[0]	*FOUT Enable.* (PLL)	0

As shown in Table 4-1, *System Configuration Registers* [0x0000] control the SDRAM interface and the internal PLL clock generation for the system clock. *FDMA Reset Register* [0x0000] is active HIGH. The SDRAM controller will be reset when it is asserted and initialize power-up sequence on external SDRAM when it is deasserted. The external SDRAM of the SOLO6010 is interfaced with 64bit or 32bit. *SDRAM Bit Width Selection Register* [0x0000] configures the external SDRAM interface bit width, which is 64bit or 32bit if *SDRAM Bit Width Selection Register* [0x0000] is HIGH or LOW, respectively. 64bit SDRAM interface configuration will bring to system with quadruple speed MPEG4 video encoding or decoding performance. 32bit SDRAM interface configuration will bring to system with cost effective solution. The FDMA clock phase regarding to SYS clock is controlled by *Positive FDMA Clock Delay Register* [0x0000] and *Negative FDMA Clock Delay Register* [0x0000]. Proper values on these registers are defined in the SOLO6010 driver software. According to applications, SYS clock is needed to 54MHz and SYS clock will be set to 54MHz by asserting *System Clock Divider Control Register* [0x0000].

4.1.2. PLL Based Clock Generation Module

Fig. 4-2 shows the block diagram of PLL based clock generation module.

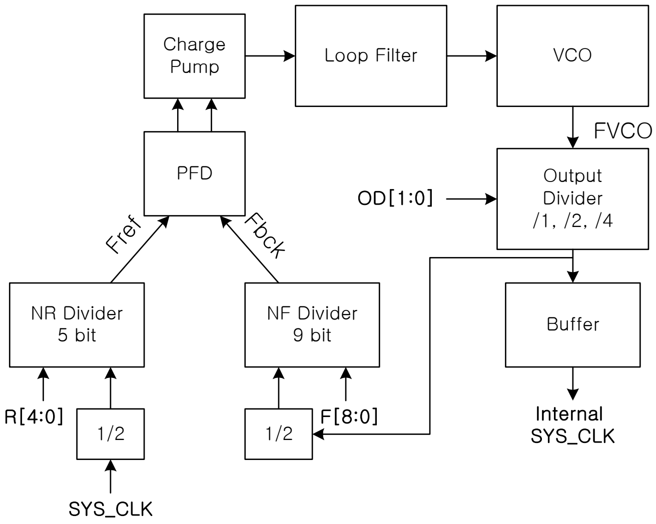


Fig. 4-2 Block diagram of PLL Based Clock Generation Module

There are 3 operation modes in this macrocell : Power Down Mode, Bypass Mode and Normal mode (see Table 4-2). Power Down Mode forces FOUT to Low and make PLL in low power consumption state. Bypass Mode provides FOUT with the same frequency as FIN. Normal Mode synthesizes FOUT by programming suitable divider values. It needs a Tready time (Pull_in Time + Locking Time) for PLL to re-lock the FIN clock when PLL wakes up from Power Down Mode operation to Normal Mode. In general, it should be reserved a Tready time for re-locking when PLL is changed to Normal Mode from Power Down or Bypass Mode, or when any divider setting is changed.

Table 4-2 Operation Modes

Operation Mode Setting			FOUT	Description
PD	BP	OE		
1	-	-	0	Power Down Mode
0	1	1	FIN	Bypass Mode
0	0	0	-	Normal Mode

In Normal Mode operation, it is necessary to set suitable divider values to make sure PLL functional :

- **PLL Divider Value Setting**

  There are 3 divider values (NR, NF, NO) to set the PLL output clock frequency FOUT :

  - Input Divider Value NR

    NR = 16*R4+8*R3+4*R2+2*R1+R0 +2

  - Feedback Divider Value NF

    NF = 256*F8+128*F7+64*F6+32*F5+16*F4+8*F3+4*F2+2*F1+F0 +2

  - Output Divider Value(NO)

Table 4-3 Output Divider Value

Output Divider Setting	NO (Output Divider Value)
OD0=0, OD1=0	1
OD0=0, OD1=1*	2
OD0=1, OD1=0*	2
OD0=1, OD1=1*	4

Note* : When OD0=1 or OD1=1, FOUT would have around 50% duty-cycle.

- PLL Output Clock Frequency Setting

  FOUT = [FIN * (NF * 2)] / [(NR * 2) * NO]

  Meanwhile, the following constraints must be followed :

  - 3.2MHz < FIN < 150MHz

- o 800KHz < Fref < 8MHz
- o 200MHz < FVCO < 500MHz, FVCO > 250MHz is preferred

where

PLL's comparison frequency Fref = FIN/(NR*2).

PLL's VCO frequency FVCO = FIN*(NF*2)/(NR*2).

Note : There is no guaranteed phase relationship between FIN and FOUT even in the case when the FOUT frequency is integer multiple of the FIN frequency.

For example, if we want to get a 108MHz output frequency clock(when using the 54MHz crystal oscillator as input), we need the NF/NR=512/64, NO=4, So, we get

NF = 256, F8:F0 = 254 -----------> 0 1 1 1 1 1 1 1 0

(F8..................... .F0)

NR = 32, R4:R0 = 30 ----------> 1 1 1 1 0 (R4........ R0)

NO = 4, OD0=1, OD1=1

The following examples (Table 2) show how to program the PLL's control pin and divider values in Normal Mode.

Table 4-4 Normal Mode Examples of PLL

FIN	FOUT	FVCO	NF/NR	OD0	OD1	PD	BP	OE	R[4:0]	F[8:0]	Note
-	0	-	-	-	-	1	-	-	-	-	Power Down
-	0	-	-	-	-	-	-	1	-	-	Fout Disable
-	FIN	-	-	-	-	0	1	0	-	-	Bypass PLL
-	FVCO	-	-	0	0	0	0	0	-	-	PLL operation VCO=Fin*NF/NR Fout=FVCO/NO
-	FVCO/2	-	-	1	0	0	0	0	-	-	
-	FVCO/2	-	-	0	1	0	0	0	-	-	
-	FVCO/4	-	-	1	1	0	0	0	-	-	
54MHz	108MHz	432MHz	256/32	1	1	0	0	0	11110	011111110	
27MHz	108MHz	432MHz	512/32	1	1	0	0	0	11001	111111110	
13.5MHz	108MHz	432MHz	512/16	1	1	0	0	0	01110	111111110	
33MHz	108MHz	432MHz	432/33	1	1	0	0	0	11111	110101110	

The PLL gives good performance based on proper divider settings. However, whether the divider setting is proper or not is determined by application. VCO stability and loop bandwidth play important roles on the quality of PLL's output clock. Of cause one should give VCO (and whole PLL) clean power, and pay more place-and-route consideration to avoid noise polluting. On the other hand, loop bandwidth selection (divider setting selection) may not be so strait-forward. Narrower loop bandwidth (larger NF divider value, and usually with lower comparison frequency) makes PLL less sensitive to input clock jitter, leaves the PLL's performance simply determined by VCO. It usually gives good short-term jitter performance if the VCO is in a quiet circumstance. But in some applications the long-term jitter performance is of main concern. It is known that increasing the loop bandwidth (set lower NF value and increase the comparison frequency) could give good long-term jitter result. So there are different "proper settings" in different applications and situation (power, noise, and input clock quality). Please find the proper divider setting value according to application, implement condition, and our silicon report.

4.1.3. SDRAM Controller & Fast DMA

All data for 16chnannel video multiplexer, MPEG4 video encoder, MPEG4 video decoder and 20channel G.723 audio CODEC are stored on the external SDRAM. The SOLO6010 contains only one high bandwidth SDRAM controller, which can reduce the number of SDRAM components as well as the SDRAM memory space with efficient memory management. SDRAM should be powered up and initialized in a predefined manner as shown in Fig. 4-3. Once power is applied to VDD and VDDQ simultaneously and the clock is stable, the SDRAM requires a 100us delay prior to issuing any command than a COMMAND INHIBIT or a NOP. Once the 100us delay has been satisfied with at least one COMMAND INHIBIT or NOP command having been applied, a PRECHARGE command should be applied. All banks must be precharged, thereby placing the device in the all banks idle state.

Once in the idle state, two auto refresh cycles must be performed. After the auto refresh cycles are complete, the SDRAM is ready for mode register programming. Because the mode register will power up in an unknown state, it should be loaded prior to applying any operational command.

The mode register is used to define the specific mode of operation of the SDRAM. Although the SDRAM is designed to be used for various applications, the SOLO6010 includes the selection of a burst length(4), a burst type(sequential), a CAS latency(3 or 2), an operating mode(standard operation) and a write burst mode(programmed burst length).

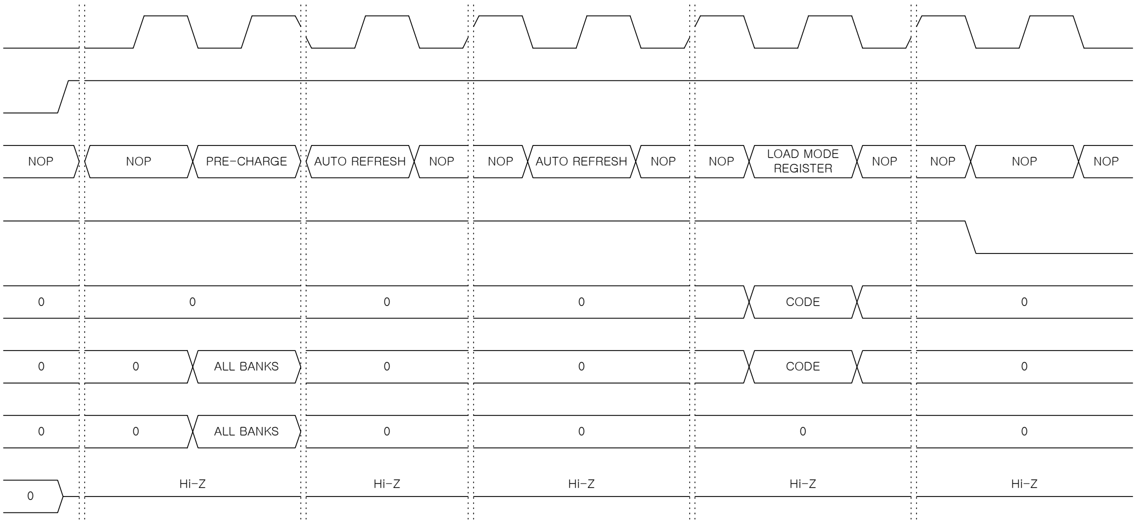


Fig. 4-3 Timing diagram of power up sequence

The SOLO6010 initialize SDRAM with power up sequence automatically when i_SYS_RST_IN is de-asserted. (See Fig. 4-1.) Host CPU is able to initialize SDRAM with power up sequence using *FDMA Reset Register* [0x0000].

Fig. 4-4 shows timing diagram of SDRAM read cycle. As shown in Fig. 4-4, the SOLO6010 uses 4 non-interleaved burst. Fig. 4-5 shows timing diagram of SDRAM write cycle. As shown in Fig. 4-5, the SOLO6010 uses 4 non-interleaved burst.

The ACTIVE command is used to open (or activate) a row in a particular bank for a subsequent access. The READ command is used to initiate a burst read access to an active row. The WRITE command is used to initiate a burst write access to an active row.

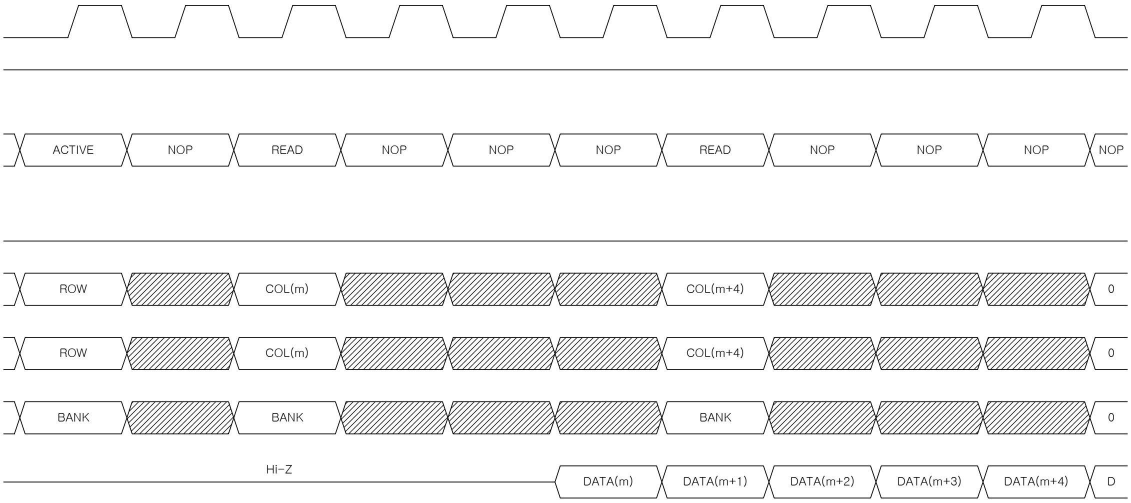


Fig. 4-4 Timing diagram of SDRAM read cycle

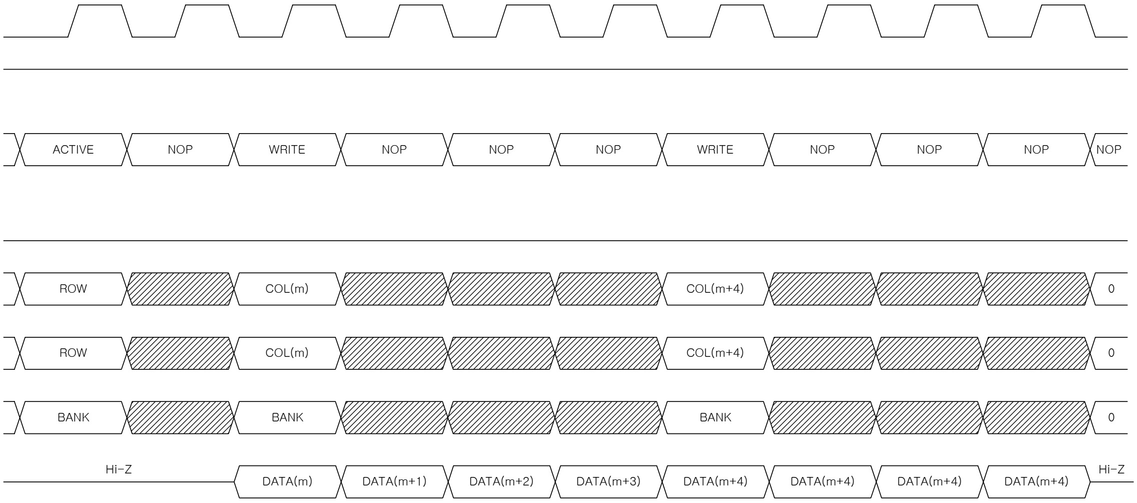


Fig. 4-5 Timing diagram of SDRAM write cycle

Table 4-5 FDMA Clock Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0004	FDMA_CONFIG		[15:8]	*SDRAM Refresh Cycle**. (Factory Default) SDRAM Controller will execute refresh every collapsed time setting value. The unit of setting value will be 256 times of system clock cycle.	0x10
			[7:6]	*SDRAM Size*. [00] : 4 * 16bit 64Mbit SDRAM 2 * 32bit 64Mbit SDRAM [01] : 4 * 16bit 128Mbit SDRAM 2 * 32bit 128Mbit SDRAM [10] : 4 * 16bit 256Mbit SDRAM [11] : 4 * 16bit 512Mbit SDRAM	2
			[5]	*SDRAM Clock Inversion**. (Factory Default)	1
			[4]	*FDMA Read Data Strobe Selection**. (Factory Default) The register is only effective in 32-bit FDMA interface mode.	1
			[3]	*FDMA Read Data Selection**. (Factory Default)	1
			[2]	*FDMA Read Data Clock Inversion**. (Factory Default)	0
			[1:0]	*FDMA Read Latency**. (Factory Default)	1
0x0008	DMA_CTRL1	RW	[5:2]	*Reserved. (BT Cycle)*	0
			[1]	*Reserved. (MultiMode_n)*	0
			[0]	*Reserved. (SeqMode_n)*	0

As shown in Table 4-5, *System FDMA Registers* [0x0004~0x0008] are accessed on high frequency clock source (108MHz) with the SOLO6010. The SOLO6010 generated predefined auto refresh cycle automatically during normal operation. The auto refresh period will be controlled by the value of *SDRAM Refresh Cycle Register* [0x0004] and its value is defined in the SOLO6010 software driver. *SDRAM Size Register* [0x0004] will define the SDRAMs which is interface with the SOLO6010. The SOLO6010 is interface 64bit data bus or 32bit data bus and the available SDRAMs are listed in Table 4-6. *SDRAM*

Clock Inversion Register [0x0004], *FDMA Read Data Strobe Selection Register* [0x0004], *FDMA Read Data Clock Inversion Register* [0x0004] and *FDMA Read Latency Register* [0x0004] are used for stable SDRAM interface. The appropriate values will be defined in the SOLO6010 driver software.

Table 4-6 SDRAM Size

SDRAM Size	Memory Space	Data Bus Width	Address Bus Width	Column Size	SDRAM(No.)	
					64bit interface	32bit Interface
00	64Mbit	16bit	12bit	8bit	4	2
	64Mbit	32bit	11bit	8bit	2	1
01	128Mbit	16bit	12bit	9bit	4	2
	128Mbit	32bit	11bit	9bit	2	1
02	256Mbit	16bit	13bit	9bit	4	2
03	512Mbit	16bit	13bit	10bit	4	2

4.1.4. Video Clock Configuration

The SOLO6010 includes the internal PLL which is generating the internal 108MHz SYS clock (SYS_CLK) in normal operation. The VOUT clock for the video color encoder can be generated to pin ioVE_CLK by the SYS clock or accepted from pin ioVE_CLK by the external clock module.

As shown in Table 4-7, the video input clock and output clock are controlled by *Video Clock Configuration Registers* [0x000C]. *Video Output Clock Selection Register* [0x000C] will decide the generating method of VOUT clock. In normal operation, the frequency of SYS clock is 108MHz and the frequency of VOUT clock will be 27MHz by setting 0 on *Video Output Clock Selection Register* [0x000C]. If the frequency of SYS clock is set to 54MHz, 27MHz according to special application, the frequency of VOUT clock will always be 27MHz by setting 1, 2 on *Video Output Clock Selection Register* [0x000C], respectively. If the frequency of SYS clock is set to other frequency expect 108MHz, 54MHz, 27Mhz, VOUT clock will be accepted the external 27MHz clock from pin ioVE_CLK by setting 3 on *Video Output Clock Selection Register* [0x000C]. *Video Output Clock Inversion Register* [0x000C] is used for stable interface with video color encoder chips and its value will be defined in the SOLO6010 software driver. The SOLO6010 has 8 video input ports to accept the video input, which will accept 16 channel video input with 54MHz Dual ITU-R BT 656 format. In case of accept 27MHz Single ITU-R BT 656 format, the SOLO6010 will accept only 8 channel video input by asserting 1 on *Video Input Clock Selection Register* [0x000C]. *Video Input Clock Delay Register* [0x000C] for each video input channel is used for stable interface with video color decoder and their value will be defined in the SOLO6010 driver software.

Table 4-7 Video Clock Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x000C	VDO_CLK_CONIFG	RW	[22]	*Video Output Clock Inversion.* (Factory Default)	0
			[21:20]	*Video Output Clock Selection*. (Factory Default) [0] : SYS_CLK/4 [1] : SYS_CLK/2 [2] : SYS_CLK [3] : EXT_CLK (from pin ioVE_CLK)	0
			[19]	*Reserved. (Video Spot Output Clock Inversion*)	0
			[18]	*Video Input Clock Selection*. (Factory Default) [0] : 54MHz Dual ITU-R BT 656 format [1] : 27MHz Single ITU-R BT 656 format	0
			[17:16]	*Reserved*.	0
			[15:14]	*Video Input Clock Delay for Channel 14, 15*. (Factory Default)	0
			[13:11]	*Video Input Clock Delay for Channel 12, 13*. (Factory Default)	0
			[11:10]	*Video Input Clock Delay for Channel 10, 11*. (Factory Default)	0
			[9:8]	*Video Input Clock Delay for Channel 8, 9*. (Factory Default)	0
			[7:6]	*Video Input Clock Delay for Channel 6, 7*. (Factory Default)	0
			[5:4]	*Video Input Clock Delay for Channel 4, 5*. (Factory Default)	0
			[3:2]	*Video Input Clock Delay for Channel 2, 3*. (Factory Default)	0
			[1:0]	*Video Input Clock Delay for Channel 0, 1*. (Factory Default)	0

4.1.5. Interrupt Configuration

As shown in Table 4-8, the Interrupt controller of the SOLO6010 has *System Interrupt Status Registers* [0x0010], *System Interrupt Acknowledge Registers* [0x0010] and *System Interrupt Enable Registers* [0x0014] for 21 different system interrupt sources. Each interrupt source can be active by asserting *System Interrupt Enable Register* [0x0014] bit. *System Interrupt Status* and *Acknowledge Registers* [0x0010] are interrupt status when they are read and interrupt acknowledge when they are written. The host CPU writes 1 on *System Interrupt Acknowledge Registers* [0x0010] to clear interrupt

Table 4-8 Interrupt Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0010	SYS_INT_STA_ACK	RW	[20]	*P2M3 Interrupt Status/Ack.* Read Mode : P2M3 Interrupt Status. Write Mode : P2M3 Interrupt Acknowledge.	0
			[19]	*P2M2 Interrupt Status/Ack.* Read Mode : P2M2 Interrupt Status. Write Mode : P2M2 Interrupt Acknowledge.	0
			[18]	*P2M1 Interrupt Status/Ack.* Read Mode : P2M1 Interrupt Status. Write Mode : P2M1 Interrupt Acknowledge.	0
			[17]	*P2M0 Interrupt Status/Ack.* Read Mode : P2M0 Interrupt Status. Write Mode : P2M0 Interrupt Acknowledge.	0
			[16]	*GPIO Interrupt Status/Ack.* Read Mode : GPIO Interrupt Status. Write Mode : GPIO Interrupt Acknowledge.	0

			[15]	*Reserved.*	0
			[14]	*Video Output Sync Interrupt Status/Ack.* Read Mode : *Video Input* Interrupt Status. Write Mode : *Video Input* Interrupt Acknowledge.	0
			[13]	*Video Motion Interrupt Status/Ack.* Read Mode : *Video Motion* Interrupt Status. Write Mode : *Video Motion* Interrupt Acknowledge.	0
			[12]	*ATA Command Interrupt Status/Ack.* Read Mode : *ATA Command* Interrupt Status. Write Mode : *ATA Command* Interrupt Acknowledge.	0
			[11]	*ATA Direction Interrupt Status/Ack.* Read Mode : *ATA Direction* Interrupt Status. Write Mode : *ATA Direction* Interrupt Acknowledge.	0
			[10]	*PCI Error Interrupt Status/Ack.* Read Mode : Fatal Interrupt Status. Write Mode : Fatal Interrupt Acknowledge.	0
			[9]	*PS2_1 Interrupt Status/Ack.* Read Mode : PS2 Interrupt Status. Write Mode : PS2 Interrupt Acknowledge.	0
			[8]	*PS2_0 Interrupt Status/Ack.* Read Mode : PCI ERROR Interrupt Status. Write Mode : PCI ERROR Interrupt Acknowledge.	0
			[7]	*SPI Interrupt Status/Ack.* Read Mode : SPI Interrupt Status. Write Mode : SPI Interrupt Acknowledge.	0
			[6]	*I2C Interrupt Status/Ack.* Read Mode : I2C Interrupt Status. Write Mode : I2C Interrupt Acknowledge.	0
			[5]	*UART1 Interrupt Status/Ack.* Read Mode : UART1 Interrupt Status. Write Mode : UART1 Interrupt Acknowledge.	0
			[4]	*UART0 Interrupt Status/Ack.* Read Mode : UART0 Interrupt Status. Write Mode : UART0 Interrupt Acknowledge.	0
			[3]	*Audio Interrupt Status/Ack.* Read Mode : Audio Interrupt Status. Write Mode : Audio Interrupt Acknowledge.	0
			[2]	*Reserved.*	
			[1]	*Decoder Interrupt Status/Ack.* Read Mode : Decoder Interrupt Status. Write Mode : Decoder Interrupt Acknowledge.	0
			[0]	*Encoder Interrupt Status/Ack.* Read Mode : *Encoder* Interrupt Status. Write Mode : *Encoder* Interrupt Acknowledge.	0
0x0014	SYS_INT_EN	RW	[20]	*P2M3 Interrupt Enable.*	0
			[19]	*P2M2 Interrupt Enable.*	0
			[18]	*P2M1 Interrupt Enable.*	0
			[17]	*P2M0 Interrupt Enable.*	0
			[16]	*GPIO Interrupt Enable.*	0
			[15]	*Reserved.*	0
			[14]	*Video Input Interrupt Enable.*	0
			[13]	*Video Motion Interrupt Enable.*	0
			[12]	*Reserved. (ATA Command Interrupt Enable)*	0
			[11]	*Reserved. (ATA Direction Interrupt Enable)*	0
			[10]	*PCI Error Interrupt Enable.*	0
			[9]	*PS2_1 Interrupt Enable.*	0
			[8]	*PS2_0 Interrupt Enable.*	0
			[7]	*SPI Interrupt Enable.*	0
			[6]	*I2C Interrupt Enable.*	0
			[5]	*UART1 Interrupt Enable.*	0
			[4]	*UART0 Interrupt Enable.*	0
			[3]	*Audio Interrupt Enable.*	0
			[2]	*Reserved.*	0
			[1]	*Decoder Interrupt Enable.*	0
			[0]	*Encoder Interrupt Enable.*	0

4.1.6. Chip Option

As shown in Table 4-9, the value of the *Chip Option Register* [0x001C] will represent 7 for SOLO6010-16, 6 for SOLO6010-9, 5 for SOLO6010-4, respectively. The SOLO6010 enables to develop 16, 8/9, 4 channel DVR or NVS with only one platform easily. Therefore, only one application software using *Chip Option register* [0x001C] is possible to develop 16, 8/9, 4 channel DVR or NVS. This feature will bring to reduce system development and maintenance cost.

Table 4-9 Test Mode Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x001C	TEST_MODE	RW	[15:8]	*Reserved.*	0
		RO	[2:0]	*Chip Option.* [111] : SOLO6010-16 [110] : SOLO6010-9 [101] : SOLO6010-4	-

4.2. PCI and Local Host Interface

The SOLO6010 supports PCI or 32bit local host interface by the pin iPCI_MODE configuration. iPCI_MODE pin is set to HIGH to active PCI or LOW to active 32bit local host interface. The PCI of the SOLO6010 supports very powerful four channel DMA functions in PCI master mode. Therefore, the host CPUs with PCI are strongly recommended to supporting the SOLO6010 which has the quadruple speed MPEG4 video CODEC.

4.2.1. PCI

The PCI of the SOLO6010 is designed to be satisfied with the protocol and electrical features of the PCI Local Bus Specification, Revision 2.2 32bit/33MHz (66MHz). The PCI interface requires a minimum of 47 pins for a target-only device and 49 pins for a master to handle data and addressing, interface control, arbitration, and system functions. The direction indication on signals in Fig. 4-6 assumes a combination master/target device.

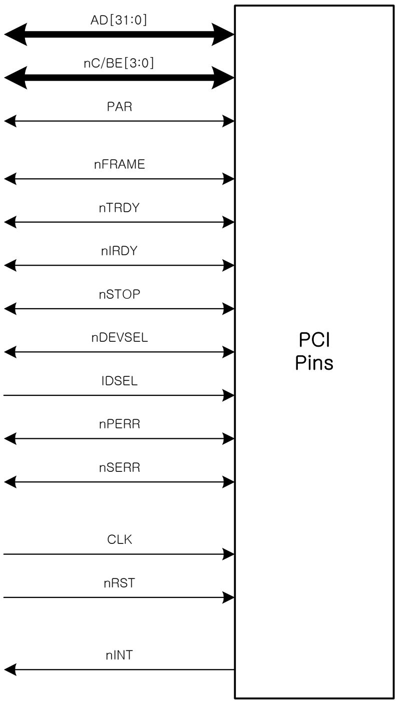


Fig. 4-6 PCI Pin List

PCI signal is defined in Table 4-10.

in *Input* is a standard input-only signal.

out *Totem Pole Output* is a standard active driver.

t/s *Tri-State* is a bi-directional, tri-state input/output pin.

s/t/s *Sustained Tri-State* is an active low tri-state signal owned and driven by one and only one agent at a time. The agent that drives an s/t/s pin low must drive it high for at least one clock before letting it float. A new agent cannot start driving a s/t/s signal any sooner than one clock after the previous owner tri-states it. A pullup is required to sustain the inactive state until another agent drives it and must be provided by the central resource.

o/d *Open Drain* allows multiple devices to share as a wire-OR. A pull-up is required to sustain the inactive state until another agent drives it and must be provided by the central resource.

Table 4-10 PCI Pin Type Definition

System Pins		
CLK	in	*Clock* provides timing for all transactions on PCI and is an input to every PCI device. All other PCI signals, except **nRST**, and **nINTD**, are sampled on the rising

		edge of **CLK** and all other timing parameters are defined with respect to this edge. PCI operates up to 33 MHz and, in general, the minimum frequency is DC (0 Hz).
nRST	in	*Reset* is used to bring PCI-specific registers, sequencers, and signals to a consistent state. What effect **nRST** has on a device beyond the PCI sequencer is beyond the scope of this specification, except for reset states of required PCI configuration registers. A device that can wake the system while in a powered down bus state has additional requirements related to **nRST**. Refer to the *PCI Power Management Interface Specification* for details. Anytime **nRST** is asserted, all PCI output signals must be driven to their benign state. In general, this means they must be asynchronously tri-stated. **nREQ** and **nGNT** must both be tristated (they cannot be driven low or high during reset). To prevent **AD**, **nC/BE**, and **PAR** signals from floating during reset, the central resource may drive these lines during reset (bus parking) but only to a logic low level; they may not be driven high. **nRST** may be asynchronous to **CLK** when asserted or deasserted. Although asynchronous, deassertion is guaranteed to be a clean, bounce-free edge. Except for configuration accesses, only devices that are required to boot the system will respond after reset.
Address and Data Pins		
AD[31:0]	t/s	*Address* and *Data* are multiplexed on the same PCI pins. A bus transaction consists of an address2 phase followed by one or more data phases. PCI supports both read and write bursts. The address phase is the first clock cycle in which **nFRAME** is asserted. During the address phase, **AD[31:0]** contain a physical address (32 bits). For I/O, this is a byte address; for configuration and memory, it is a DWORD address. During data phases, **AD[07:0]** contain the least significant byte (lsb) and **AD[31:24]** contain the most significant byte (msb). Write data is stable and valid when **nIRDY** is asserted; read data is stable and valid when **nTRDY** is asserted. Data is transferred during those clocks where both **nIRDY** and **nTRDY** are asserted.
nC/BE[3:0]	t/s	*Bus Command* and *Byte Enables* are multiplexed on the same PCI pins. During the address phase of a transaction, **nC/BE[3:0]** define the bus command. During the data phase, **nC/BE[3:0]** are used as Byte Enables. The Byte Enables are valid for the entire data phase and determine which byte lanes carry meaningful data. **nC/BE[0]** applies to byte 0 (lsb) and **nC/BE[3]** applies to byte 3 (msb).
PAR	t/s	*Parity* is even3 parity across **AD[31:0]** and **nC/BE[3:0]**. Parity generation is required by all PCI agents. **PAR** is stable and valid one clock after each address phase. For data phases, **PAR** is stable and valid one clock after either **nIRDY** is asserted on a write transaction or **nTRDY** is asserted on a read transaction. Once **PAR** is valid, it remains valid until one clock after the completion of the current data phase. (**PAR** has the same timing as **AD[31:0]**, but it is delayed by one clock.) The master drives **PAR** for address and write data phases; the target drives **PAR** for read data phases.
Interface Control Pins		
nFRAME	s/t/s	*Cycle Frame* is driven by the current master to indicate the beginning and duration of an access. **nFRAME** is asserted to indicate a bus transaction is beginning. While **nFRAME** is asserted, data transfers continue. When **nFRAME** is deasserted, the transaction is in the final data phase or has completed.
nIRDY	s/t/s	*Initiator Ready* indicates the initiating agent's (bus master's) ability to complete the current data phase of the transaction. **nIRDY** is used in conjunction with **nTRDY**. A data phase is completed on any clock both **nIRDY** and **nTRDY** are asserted. During a write, **nIRDY** indicates that valid data is present on **AD[31:0]**. During a read, it indicates the master is prepared to accept data. Wait cycles are inserted until both **nIRDY** and **nTRDY** are asserted together.
nTRDY	s/t/s	*Target Ready* indicates the target agent's (selected device's) ability to complete the current data phase of the transaction. **nTRDY** is used in conjunction with **nIRDY**. A data phase is completed on any clock both **nTRDY** and **nIRDY** are asserted. During a read, **nTRDY** indicates that valid data is present on **AD[31:0]**. During a write, it indicates the target is prepared to accept data. Wait cycles are inserted until both **nIRDY** and **nTRDY** are asserted together.
nSTOP	s/t/s	*Stop* indicates the current target is requesting the master to stop the current transaction.
nLOCK	s/t/s	*Lock* indicates an atomic operation to a bridge that may require multiple transactions to complete. When **nLOCK** is asserted, non-exclusive transactions may proceed to a bridge that is not currently locked. A grant to start a transaction on PCI does not guarantee control of **nLOCK**. Control of **nLOCK** is obtained under its own protocol in conjunction with **nGNT**. It is possible for different agents to use PCI while a single master retains ownership of **nLOCK**. Locked transactions may be initiated only by host bridges, PCI-to-PCI bridges, and expansion bus bridges. Refer to Appendix F for details on the requirements of **nLOCK**.

IDSEL	in	*Initialization Device Select* is used as a chip select during configuration read and write transactions.
nDEVSEL	s/t/s	*Device Select*, when actively driven, indicates the driving device has decoded its address as the target of the current access. As an input, **nDEVSEL** indicates whether any device on the bus has been selected.
Arbitration Pins (Bus Masters Only)		
nREQ	t/s	*Request* indicates to the arbiter that this agent desires use of the bus. This is a point-to-point signal. Every master has its own **nREQ** which must be tri-stated while **nRST** is asserted.
nGNT	t/s	*Grant* indicates to the agent that access to the bus has been granted. This is a point-to-point signal. Every master has its own **nGNT** which must be ignored while **nRST** is asserted.
Error Reporting Pins		
nPERR	s/t/s	*Parity Error* is only for the reporting of data parity errors during all PCI transactions except a Special Cycle. The **nPERR** pin is sustained tri-state and must be driven active by the agent receiving data (when enabled) two clocks following the data when a data parity error is detected. The minimum duration of **nPERR** is one clock for each data phase that a data parity error is detected. (If sequential data phases each have a data parity error, the **nPERR** signal will be asserted for more than a single clock.) **nPERR** must be driven high for one clock before being tri-stated as with all sustained tri-state signals.
nSERR	o/d	*System Error* is for reporting address parity errors, data parity errors on the Special Cycle command, or any other system error where the result will be catastrophic. If an agent does not want a non-maskable interrupt (NMI) to be generated, a different reporting mechanism is required. **nSERR** is pure open drain and is actively driven for a single PCI clock by the agent reporting the error. The assertion of **nSERR** is synchronous to the clock and meets the setup and hold times of all bused signals. However, the restoring of **nSERR** to the deasserted state is accomplished by a weak pullup (same value as used for s/t/s) which is provided by the central resource not by the signaling agent. This pullup may take two to three clock periods to fully restore **nSERR**. The agent that reports **nSERR** to the operating system does so anytime **nSERR** is asserted.
Interrupt Pins (Optional)		
nINT	o/d	*Interrupt* is used to request an interrupt.

Exactly how the IDSEL pin is driven is left to the discretion of the host/memory bridge or system designer. This signal has been designed to allow its connection to one of the upper 21 address lines, which are not otherwise used in a configuration access. However, there is no specified way of determining IDSEL from the upper 21 address bits. Therefore, the IDSEL pin must be supported by all targets. Devices must not make an internal connection between an AD line and an internal IDSEL signal in order to save a pin. The only exception is the host bridge, since it defines how IDSELs are mapped. IDSEL generation behind a PCI-to-PCI bridge is specified in the PCI-to-PCI Bridge Architecture Specification.

How a system generates IDSEL is system specific; however, if no other mapping is required, the following example may be used. The IDSEL signal associated with Device Number 0 is connected to AD[16], IDSEL of Device Number 1 is connected to AD[17], and so forth until IDSEL of Device Number 15 is connected to AD[31]. For Device Number 17-31, the host bridge should execute the transaction but not assert any of the AD[31:16] lines but allow the access to be terminated with Master-Abort.

Twenty-one different devices can be uniquely selected for configuration accesses by connecting a different address line to each device an asserting one of the AD[31:11] lines at a time. The issue with connecting one of the upper 21 AD lines to IDSEL to the appropriate AD line. This does, however, create a very slow slew rate on IDSEL, causing it to be in an invalid logic state most of the time, as shown in Fig. 4-7 with “XXXX” marks.

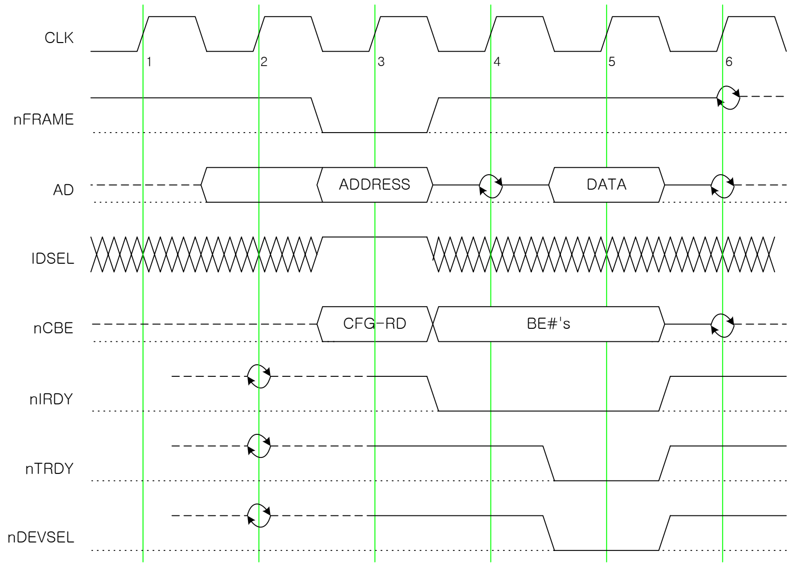


Fig. 4-7 Configuration Read

Fig. 4-8 illustrates a real transaction and starts with an address phase which occurs when nFRAME is asserted for the first time and occurs on clock2. During the address phase, AD[31:0] contain a valid address and nC/BE[3:0] contain a valid bus command.

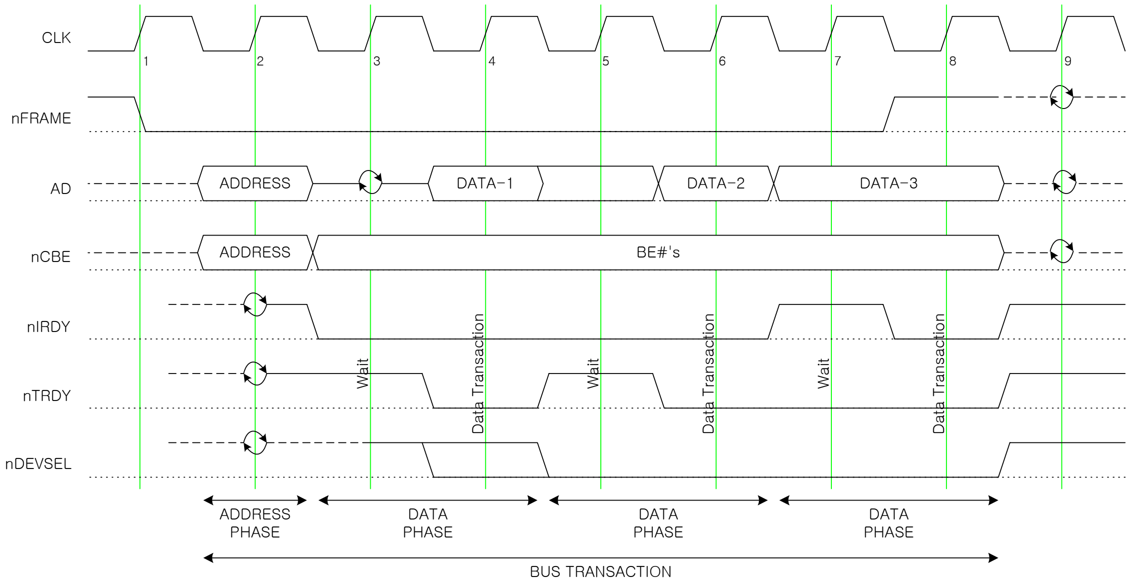


Fig. 4-8 Basic Read Operation

The first clock of the first data phase is clock3. During the data phase, nC/BE indicate which byte lanes are involved in the current data phase. A data phase may consist of wait cycles and a data transfer. The nC/BE output buffers must remain enabled (for both read and writes) form the fist clock of the data phase through the end of the transaction. This ensures nC/BE are not left floating for long intervals. The nC/BE lines contain valid byte enable information during the entire data phase independent of the state of nIRDY. The nC/BE lines contain the byte enable information for data phase N+1 on the clock following the completion of the data phase N. This is not shown in Fig. 4-8 because a burst read transaction typically has all byte enables asserted; however, it is shown in Fig. 4-9. Notice on clock 5 in Fig. 4-9, the master inserted a wait state by deasserting nIRDY. However, the byte enables for data phase 3 are valid on clock 5 and remain valid until the data phase completes on clock 8.

The fist data phase on a read transaction requites a turnaround-cycle (enforced by the target via nTRDY). In this case, the address is valid on clock2 and then the master stops driving AD. The earliest the target can provide valid data is clock 4. The target must drive the AD lines following the turnaround cycle when nDEVSEL is asserted. Once enabled, the output buffers must stay enabled through the end of the transaction. (This ensures that the AD lines are not left floating for lone intervals.)

One way for a data phase to complete is when data is transferred, which occurs when both nIRDY and nTRDY are asserted on the same rising clock edge. There are other conditions that complete a data phase. (nTRDY cannot be driven until nDEVSEL is asserted.) When either nIRDY or nTRDY is deasseted, a wait cycle is inserted and no data is transferred. As noted in Fig. 4-8, data is successfully transferred on clocks 4, 6, and 8 and wait cycles are inserted on clock 3, 5, and 7. The first data phase completes in the minimum time for a read transaction. The second data phase is extended on clock 5 because nTRDY is deasserted. The last data phase is extended because nIRDY was deasserted on clock 7.

The master knows at clock 7 that the next data phase is the last. However, because the master is not ready to complete the last transfer (nIRDY is deasserted on clock 7), nFRAME stays asserted. Only when nIRDY is asserted can nFRAME be deasserted as occurs on clock 8, indicating to the target that this is the last data phase of the transaction.

Fig. 4-9 illustrates a write transaction. The transaction starts when nFRAME is asserted for the first time which occurs on clock 2. A write transaction is similar to a read transaction except no turnaround cycle is required following the address phase because the master provides both address and data. Data phases work the same for both read and write transactions.

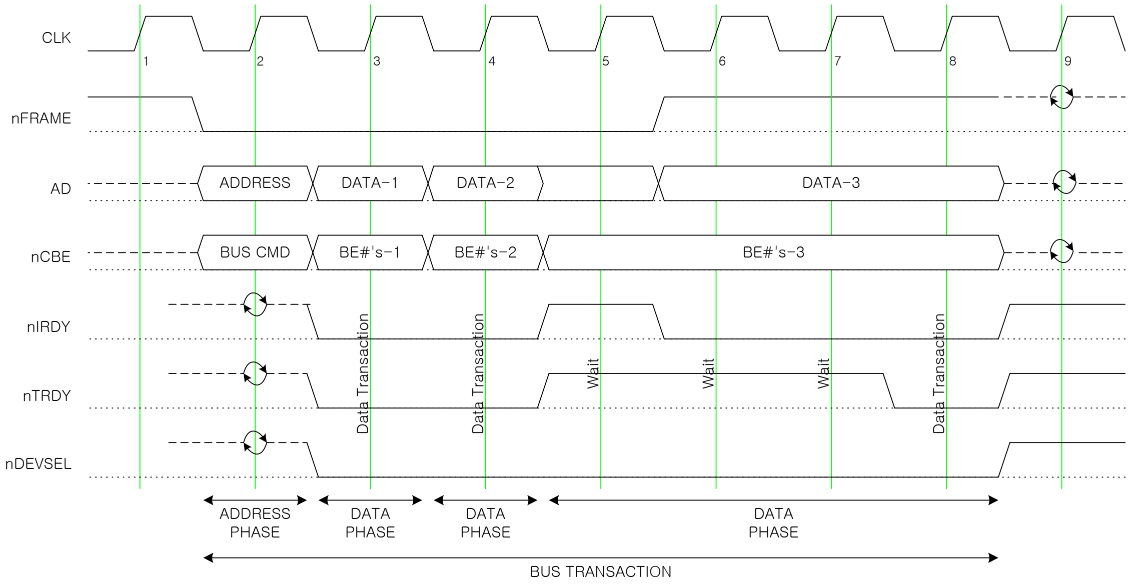


Fig. 4-9 Basic Write Operation

In Fig. 4-9, the first and second data phases complete with zero wait cycles. However, the third data phase has three wait cycles inserted by the target. Notice both agents insert a wait cycle on clock 5. nIRDY must be asserted when nFRAME is deasserted indicating the last data phase.

The date transfer was delayed by the master on clock 5 because nIRDY was deasserted. The last data phase is signaled by the master on clock 6, but it does not complete until clock 8.

The PCI configuration registers of the SOLO6010 have default values and are updated values using PCI configuration EEPROM. iPCI_EEPROM pin is set to HIGH to use PCI configuration EEPROM and the PCI configuration registers are loaded after the pin i_SYS_RST_IN is reset. As shown in Table 4-11, *PCI Configuration EEPROM Access Enable Register* [0x0060] should be set to 1 to access the PCI configuration EEPROM. *PCI Configuration EEPROM Chip Selection Register* [0x0060], *PCI Configuration EEPROM Serial Clock* [0x0060], *PCI Configuration EEPROM Data Output Register* [0x0060] and *PCI Configuration EEPROM Data Input Register* [0x0060] are used to interface with PCI configuration through oEEPROM_CS, oEEPROM_SK, oEEPROM_DO and iEEPROM_DI, respectively.

Table 4-11 EEPROM Access Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0060	EEPROM_ACC	RW	[7]	*PCI Configuration EEPROM Access Enable.*	0
			[3]	*PCI Configuration EEPROM Chip Select.*	0
			[2]	*PCI Configuration EEPROM Serial Clock.*	0
			[1]	*PCI Configuration EEPROM Data Output.*	0
		RO	[0]	*PCI Configuration EEPROM Data Input.*	-

Table 4-12 shows PCI configuration registers, which is listed up initial values. Device ID, Vendor ID, Subsys ID, Subsys Vendor ID, Class Code, Rev ID, Header Type, Max Lat, Min Gnt and Int Pin of PCI configuration registers can be update by only PCI configuration EEPROM. Command, Status, Latency Timer, Cache Line Size, Memory Base Address, Interrupt Line, Retry Timeout and TRDY Timeout of PCI configuration registers can be update by PCI configuration write as shown as Fig. 4-7. In this case, PCI command needs to be change as PCI configuration write instead of PCI configuration read. In addition, PCI configuration read is used to read PCI configuration registers for checking initial value and updated value.

Table 4-12 PCI Configuration Registers

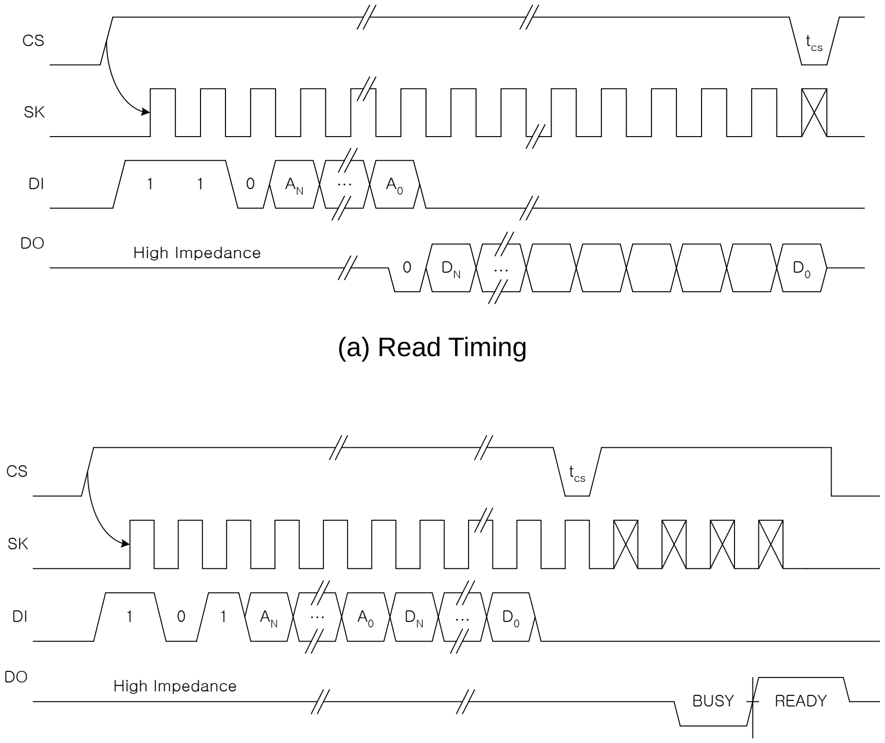
31		16 15	0	Offset	Initial Value
Device ID		Vendor ID		00h	6010_9413h
Status		Command		04h	02a0_0000h
Class Code			Revision ID	08h	040000_00h
BIST	Header Type	Latency Timer	Cache line Size	0Ch	00_00_20_00h
Memory Base Address				10h	00_00000_8h
Reserved				14h	
Reserved				18h	
Reserved				1Ch	
Reserved				20h	
Reserved				24h	
Reserved				28h	
Subsystem ID		Subsystem Vendor ID		2Ch	0x6010_9413
Reserved				30h	
Reserved				34h	
Reserved				38h	
Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line	3Ch	0x01_00_01_00
Reserved		Retry Timeout	TRDY Timeout	40h	0x0000_80_80

Table 4-13 Contents of PCI configuration EEPROM

Address	Contents
00h	Vendor ID
04h	Device ID
58h	Subsystem Vendor ID

5Ch	Subsystem ID	
78h	Interrupt Pin	
7Ch	Max_Lat	Min_Gnt

Pin iPCI_EEPROM should be set to HIGH to update the PCI configuration registers from PCI configuration EEPROM. The configuration registers are updated automatically after system reset. PCI configuration EEPROM can be access using EEPROM_ACC register. The PCI Configuration EEPROM Access Enable should be set to 1 to access PCI configuration EEPROM. Fig. 4-10 shows an example of PCI configuration EEPROM timing.



(a) Read Timing

(b) Write Timing

Fig. 4-10 PCI Configuration EEPROM Timing (AT93C46 of ATMEL)

Fig. 4-11 shows the address mapping on PCI memory address region. The host CPU accesses the registers to control the SOLO6010 and the P2M buffer to exchange many amount of data efficiently with the SOLO6010. The PCI master mode operation will be explained in clause 4.2.3.

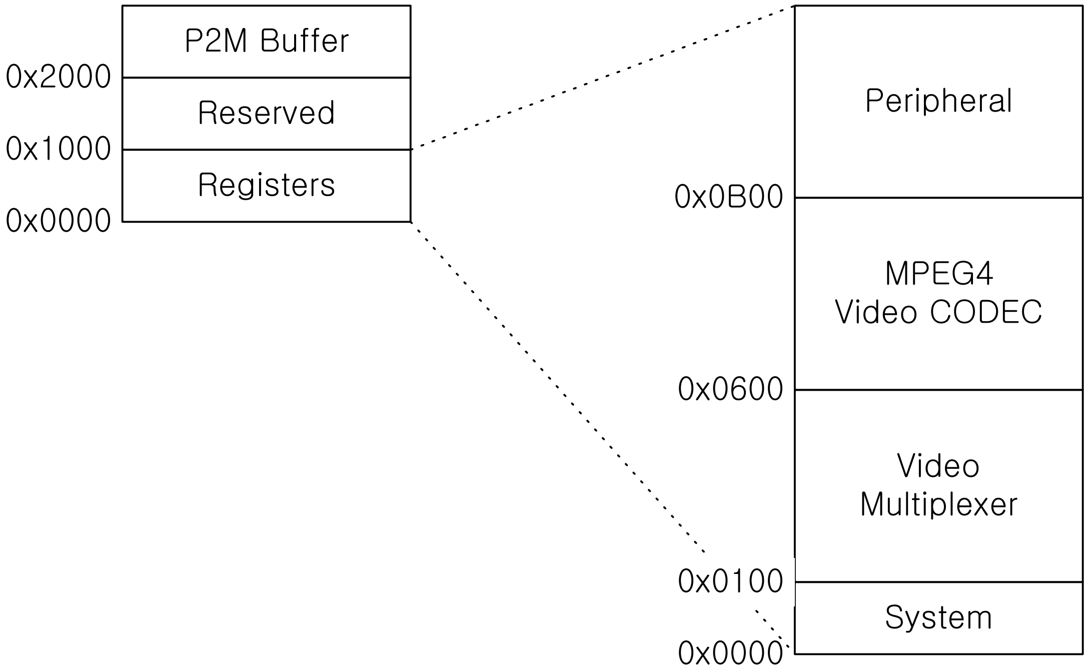


Fig. 4-11 Address Mapping on PCI memory address region

If the PCI error is occurred, the SOLO6010 will assert the PCI error interrupt on the *PCI Error Interrupt Status/Ack Register* [0x0010]. In case of error interrupt, *PCI Error Registers* [0x0070] will store PCI error statuses which are classified as PCI fatal error, PCI parity error, PCI target abort error and PCI timeout error. The SOLO6010 driver software will include the handling methods for these PCI errors.

When the SOLO6010 is operated in master mode, the SOLO6010 can meet a PCI fatal error. In this case, the SOLO6010 will assert *PCI fatal Error interrupt Register* [0x0010]. The PCI fatal error will occur in the following cases:

1. If the requested PCI target does not respond. This error usually indicates an invalid address. The SOLO6010 will execute a Master Abort cycle.
2. If the requested PCI target terminates the cycle with a target abort.
3. If the requested PCI target responds with a PCI_DEVSEL#, but does not follow with PCI_TRDY or PCI_STOP to allow the cycle to complete. The SOLO6010 provides a programmable timer. TRDY_TIMEOUT to determinate at what point the master should abandon the cycle. (Bus Lookup Prevention)
4. If the PCI target device retries beyond the retry count. The SOLO6010 provides a programmable timer, RETRY_TIMEOUT to determine at what point the master should abandon the cycle. (Bus Lookup Prevention)

The PCI master parity error can be occurred during the SOLO6010 is writing in the PCI master operation. The SOLO6010 will assert io_PCI_PERR for the PCI data parity error and io_PCI_SERR for the PCI address parity error in the PCI target operation.

The SOLO6010 will assert the PCI Target Abort Error if the PCI address parity error or bytes enable error was occurred.

The SOLO6010 will wait 65,536 cycles in PCI master operation until occupying the PCI bus. The SOLO6010 will assert the PCI timeout error if it cannot occupy the PCI bus during 65,536 cycles. If the Timeout Error Enable register is asserted, the SOLO6010 will assert the interrupt as other error conditions.

Table 4-14 PCI Error Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0070	PCI_ERROR	RW	[31]	*Timeout Error Enable.*	0
		RO	[28:24]	*Reserved. (FSM2 for Debugging)*	-
			[23:20]	*Reserved. (FSM1 for Debugging)*	-
			[19:16]	*Reserved. (FSM0 for Debugging)*	-
			[15:7]	*Reserved.*	-
			[6]	*P2M Descriptor Master Application.*	-
			[5]	*ATA Active in PCI Master Application.*	-
			[4]	*P2M Active as PCI Master Application.*	
			[3]	*Timeout Error.*	-
			[2]	*Target Abort Error.*	-
			[1]	*Parity Error.*	-
			[0]	*Fatal Error.*	-

4.2.2. Host Interface

Fig. 4-12 shows host writing timing. The host interface is designed by synchronized on internal SYS clock. Therefore, control signal, address and data need to be maintained at least three clock cycles for two clock cycles for the setup time and one clock cycle for the hold time. For example, one clock cycle is about 9.25nsec if SYS clock is set to 108MHz (clock period is about 9.25nsec) clock source.

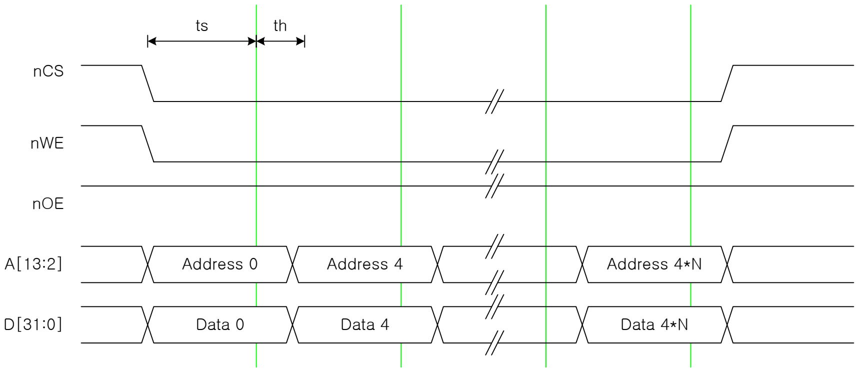


Fig. 4-12 Host Writing Timing

Fig. 4-13 show host reading timing. Control signal, address and data need to be maintained at least four clock cycles for two clock cycles for the setup time and two clock cycles for the hold time.

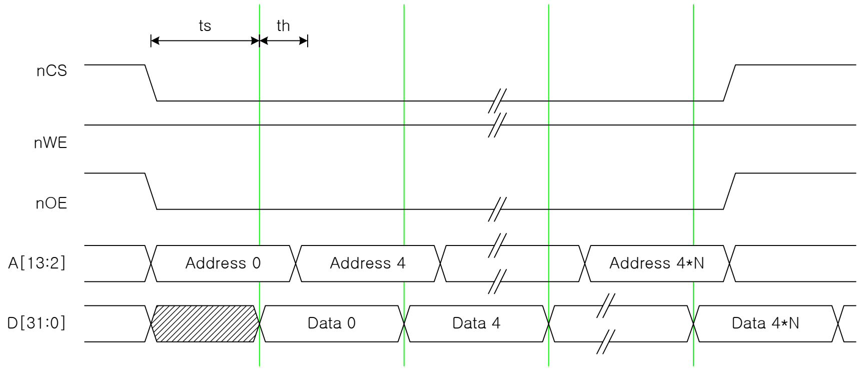


Fig. 4-13 Host Reading Timing

4.2.3. P2M Bridge

The SOLO6010 includes the P2M (PCI to Memory) bridge which exchanges the data between the host CPU memory and the SOLO6010 external memory. It will enable to transfer data very efficiently using DMA (Direct Memory Access) in the PCI master operation mode. However, it is impossible to use the DMA in the PCI target operation mode or in the host interface mode. The powerful DMA of the SOLO6010 will bring reducing the relative CPU cost as well as the upgraded system performance. In other word, it is very important to support the quadruple speed MPEG4 CODEC without any interface restrictions in PCI master mode applications.

As shown in Table 4-15, the SOLO6010 supports four DMA channels for the P2M bridge and includes four set of registers to configure *P2M Configuration Registers* [0x0080~0x00FC]. Each P2M bridge channel can be allocated for the MPEG4 video encoder, the MPEG4 video decoder, the OSG write and the live image, respectively. Therefore, the entire system performance regarding the DMA will be increased because four different DMA will be operated simultaneously.

Table 4-15 P2M Configuration Register

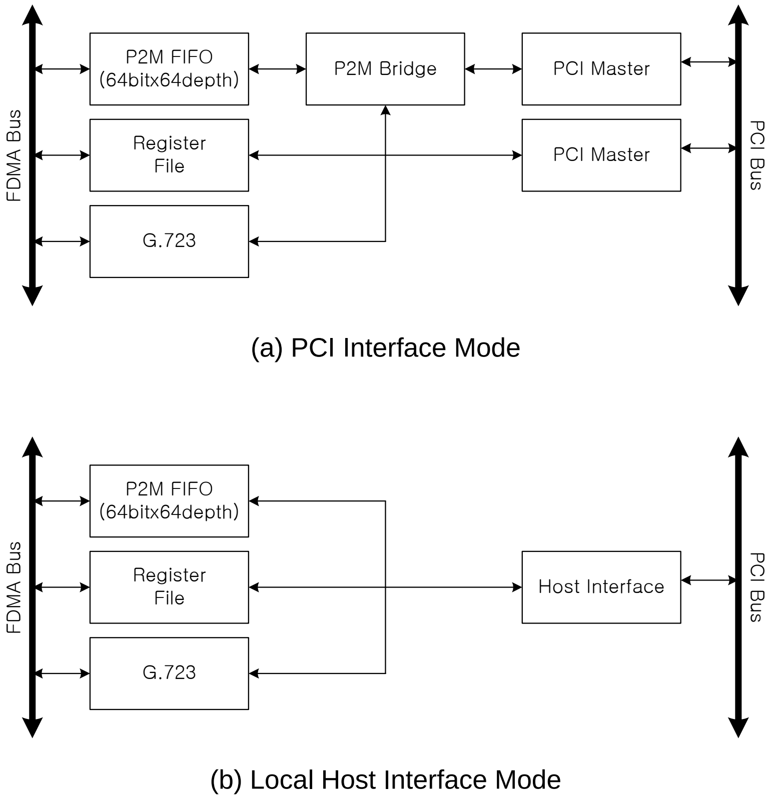
Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0080 0x00A0 0x00C0 0x00E0	P2M_CONFIG0 P2M_CONFIG1 P2M_CONFIG2 P2M_CONFIG3	RW	[10:6]	*DMA Waiting Interval*. (N*64 clocks) The interval between burst transaction is controlled by setting the value.	0
			[5]	*Byte Reorder*. [0] : RGB to RGB [1] : BGR to RGB	0
			[4]	16bit RGB Mode. [0] : {xrrr rrgg gggb bbbb} [1] : {rrrr rggg gggb bbbb}	0

			[3]	*Chroma Swap*.
				[0] : {Cb Y0 Cr Y1}
				[1] : {Cr Y0 Cb Y1}
					0

			[2]	*PCI Master Mode*. [0] : PCI Target/Host Mode. [1] : PCI Master Mode.	0
			[1]	*Descriptor Interrupt Mode.* [0] : Each Command Finish. [1] : Descriptor Queue Empty.	0
			[0]	*Descriptor Mode.* Active HIGH.	0
0x0084 0x00A4 0x00C4 0x00E4	P2M_DES_ADR0 P2M_DES_ADR1 P2M_DES_ADR2 P2M_DES_ADR3	RW	[31:0]	*Descriptor Base Address.* The descriptor queue will be store on CPU memory. The registers will points the addresses to be stored the 256 depth descriptor queues of which size are 4,096 bytes, respectively.	0
0x0088 0x00A8 0x00C8 0x00E8	P2M_DES_ID0 P2M_DES_ID1 P2M_DES_ID2 P2M_DES_ID3	RW	[7:0]	*Last Descriptor ID*. The registers are pointing the last descriptor. The P2M descriptor mode will be stopped after finishing the *Last Descriptor ID*.	0
0x008C 0x00AC 0x00CC 0x00EC	P2M_STATUS0 P2M_STATUS1 P2M_STATUS2 P2M_STATUS3	RO	[8]	*Command Done*. The register will be asserted at the interrupt request status.	-
			[7:0]	*Current Descriptor ID*. The register value will point the descriptor ID which is operated currently.	-
0x0090 0x00B0 0x00D0 0x00F0	P2M_CONTROL0 P2M_CONTROL1 P2M_CONTROL2 P2M_CONTROL3	RW	[31:20]	*Host Memory Increase Size (TI).* (DWORD)	0
			[19:10]	*Repeat Number (RP).*	0
			[9:7]	*Burst Size.* (Bytes) [0] : 512 [1] : 256 [2] : 128 [3] : 64 [4] : 32 [5] : 128 (2 page mode)	0
			[6]	When Color Space Conversion is asserted. *RGB Bit Width.* [0] : 24bit [1] : 16bit When Color Space Conversion is de-asserted. *VGA OSG Alpha Blending*. [0] : Off [1] : On	0
			[5:4]	When Color Space Conversion is asserted. [0] : Y[0] <=0 (Off) [1] : Y[0] <=1 [2] : Y[0] <= G[0] [3] : Y[0] <= Bit[15] When Color Space Conversion is de-asserted. [0] : Y[0] <= 0 [1] : [15] <= 1 [2] : [15] <= [5] [3] : [15] <= [15]	0
			[3]	*Color Space Conversion On.*	0
			[2]	*Reserved.*	0
			[1]	*Write Enable.* [0] : Read [1] : Write	0
			[0]	*Transaction On.*	0
0x0094 0x00B4 0x00D4 0x00F4	P2M_EXT_CFG0 P2M_EXT_CFG1 P2M_EXT_CFG2 P2M_EXT_CFG3	RW	[31:20]	*External Memory Increase Size (FI).* (DWORD)	0
			[19:0]	*External Memory Copy Size (FC).* (DWORD)	0
0x0098 0x00B8 0x00D8 0x00F8	P2M_TAR_ADR0 P2M_TAR_ADR1 P2M_TAR_ADR2 P2M_TAR_ADR3	RW	[31:0]	*Host Memory Base Address (TA).*	0
0x009C 0x00BC 0x00DC 0x00FC	P2M_EXT_ADR0 P2M_EXT_ADR1 P2M_EXT_ADR2 P2M_EXT_ADR3	RW	[31:0]	*External Memory Base Address (FA).*	0

The SOLO6010 supports the DMA descriptor, which can reduce the waiting time between the P2M transaction and increase the transaction efficiency. This feature is only effective in PCI master operation. This feature will make more powerful system performance for the OSG or the live video overlay of PC based DVR because the discontinuous data on the external SDRAM addressing will be transacted like as the continuous data. As shown in Fig. 4-11, the buffer memory for the P2M bridge is allocation on 0x2000 in PCI addressing region.

Fig. 4-14 shows the block diagram of P2M bridge. In PCI target mode and local host interface mode, the host CPUs accesses the registers, P2M buffer and G.723 code buffer. The DMA in P2M bridge transfers data in burst mode during the SOLO6010 is operated in PCI master mode.



(a) PCI Interface Mode

(b) Local Host Interface Mode

Fig. 4-14 Block Diagram of the P2M Bridge

PCI Target and Host Mode Operation

As shown in Fig. 4-15, the host CPU reads or writes SDRAM data of the SOLO6010 by the unit of the burst size at one transaction in PCI target mode or local host mode operation. The CPU can acquire the end of transaction by using P2M transaction done register with polling mode or P2M interrupt status register with interrupt mode. That feature may reduce transaction efficiency and increase CPU load.

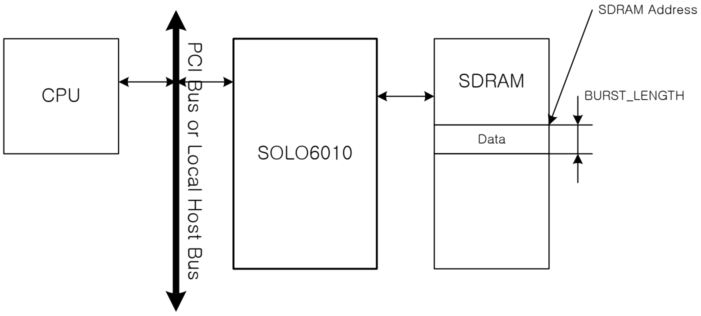


Fig. 4-15 Operation of PCI Target Mode or Local Host Mode

In PCI target and Local Host Mode operation, two pages can be used when *Burst Size Register* [0x0090, 0x00B0, 0x00D0, 0x00F0] is set to 5. The P2M FIFO in Fig. 4-14 will be split to two buffers. One is used to buffer data from SDRAM while the other is used to buffer data to CPU. The transaction efficiency will be better about 10% than that of one page modes.

PCI Master Mode Operation

As shown in Fig. 4-16, the SOLO6010 transfers a large amount of data by the unit of the burst size repeatedly in PCI master mode operation. The CPU can acquire the end of transaction by using P2M transaction done register with polling mode or P2M interrupt status register with interrupt mode. That feature can increase transaction efficiency and reduce CPU load.

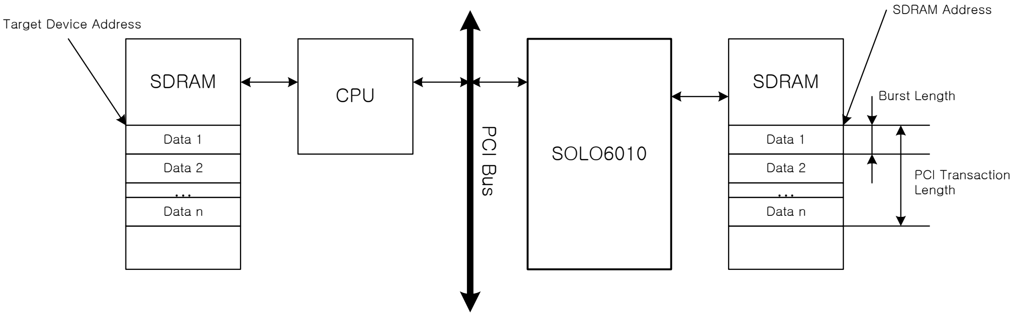


Fig. 4-16 Operation of PCI Master Mode

In normal operation, the data will be transacted in continuous memory address. However, the video live read or OSG data write will be transacted line by line. The SOLO6010 includes the P2M bridge transact the data line by line. Fig. 4-17 illustrates DMA transaction line by line.

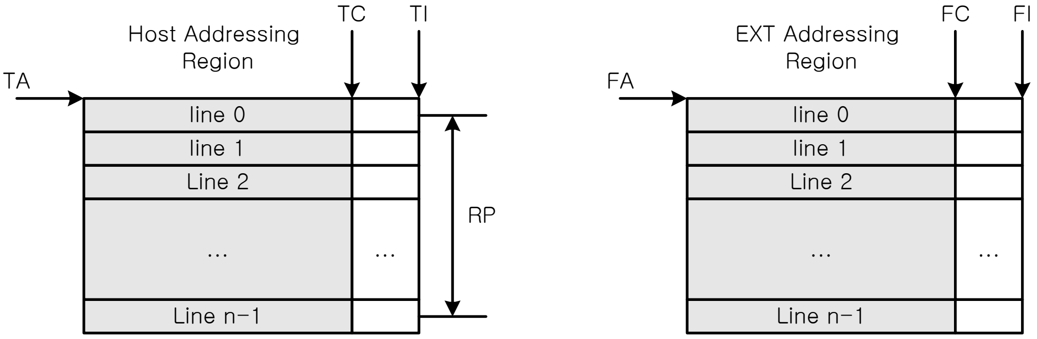


Fig. 4-17 DMA Transaction Line by Line

FA : External Memory Base Address (P2M_EXT_ADR)

TA : Host Memory Base Address (P2M_TAR_ADR)

FC : External Memory Copy Size (P2M_EXT_CFG[19:0]), DWORD

TC : Host Memory Copy Size, (Not defined) DWORD

FI : External Memory Increase Size (P2M_EXT_CFG[31:20]), DWORD

TI : Host Memory Increase Size (P2M_CTRL[31:20]), DWORD

RP : Repeat Number (P2M_CTROL[19:10])

When the user set the registers, the set values should be sufficient the following conditions.

1) FI >= FC, TI >= TC

2) If CSC Mode, FC=TC*2/3

Otherwise, FC=TC

Although the SOLO6010 supports the DMA operation, it will take quite a long for the host CPU to response the interrupt and to set the next DMA operation register. The SOLO6010 supports the description mode to reduce the interval between transactions. If the host CPU stores the P2M descriptors in the P2M description queue on the host addressing region, and starts the description operation, the SOLO6010 will get the P2M descriptor on the PCI master mode and process the P2M descriptor one by one until the description queue is empty. Each P2M description queue size will be 4k bytes and store 256 P2M descriptors which are 16 bytes as shown in Table 4-16. This feature will improve the entire system performance because it needs not to wait finishing every P2M command and to set the next command.

Table 4-16 The P2M Descriptor

Offset	31 20	19 10	9 0
0x00	Host Memory Increase Number[TI] (DWORD)	Repeat Number (Line Mode)	Controls

0x04	External Memory Increase Number[FI] (DWORD)	External Memory Copy Size [FC] (DWORD)
0x08	Host Memory Base Address [TA]	
0x0C	Ext Memory Base Address [FA]	

Controls

9	8	7	6	5	4	3	2	1	0
Burst			16	A Mode		CSC	INT	WR	ON

Color Space Conversion (CSC)

The P2M bridge includes the RGB to YUV color space conversion which converts RGB to YUV format automatically during writing the RGB OSB data on the external memory region. Fig. 4-18 shows the block diagram of the color space conversion.

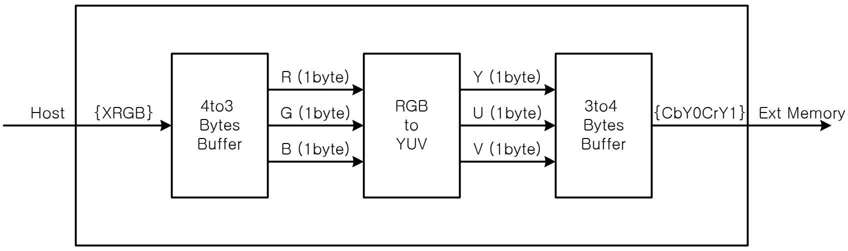


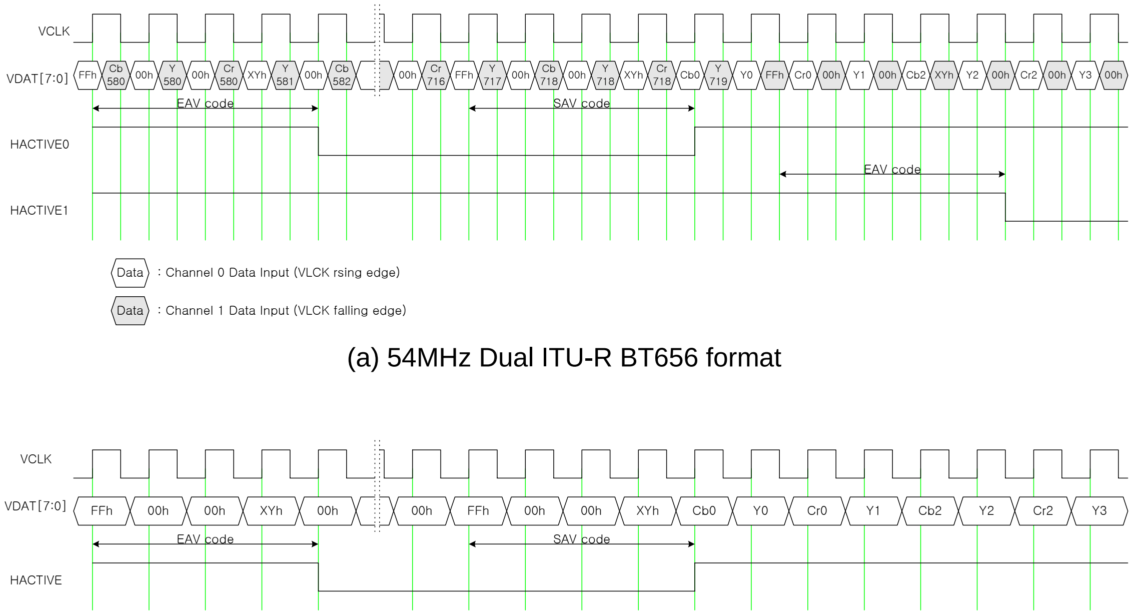
Fig. 4-18 The Block Diagram of the Color Space Conversion

4.3. Video Multiplexer

4.3.1. Video Input Interface

The SOLO6010 has 8 video input ports which accept 16 video input streams with 54MHz dual ITU-R BT 656 format. Dual ITU-R BT 656 format enables two video input channel data to be accepted through one video input port using time multiplexing. Therefore, dual ITU-R BT 656 format is very useful for security applications which need multi video input ports because video input ports can be reduced.

Fig. 4-19(a) shows 54MHz dual ITU-R BT 656 format for video input port 0 to 15 and Fig. 4-19(b) shows normal ITU-R BT 656 format which is used for the video output port. As shown in Fig. 4-19(a), the video input channel 0 is accepted at the rising edge of the video clock and the video input channel 1 is accepted at the falling edge of the video clock. The video input clock can be delayed to avoid setup/hold timing violation by setting *Video Clock Configuration Registers* [0x000C]. The appropriate value will be defined in the SOLO6010 driver software.



(a) 54MHz Dual ITU-R BT656 format

(b) 27HMz ITU-R 656 format

Fig. 4-19 Waveform of Multiplexed Video Input Format

The SOLO6010 contains the video matrix switch between the video input ports and the video multiplexer input ports. The video matrix enables the video input channel to be changed virtually and to be used to produce the multiple MPEG4 video streams. Fig. 4-21 shows the structure of the video matrix switch of the SOLO6010.

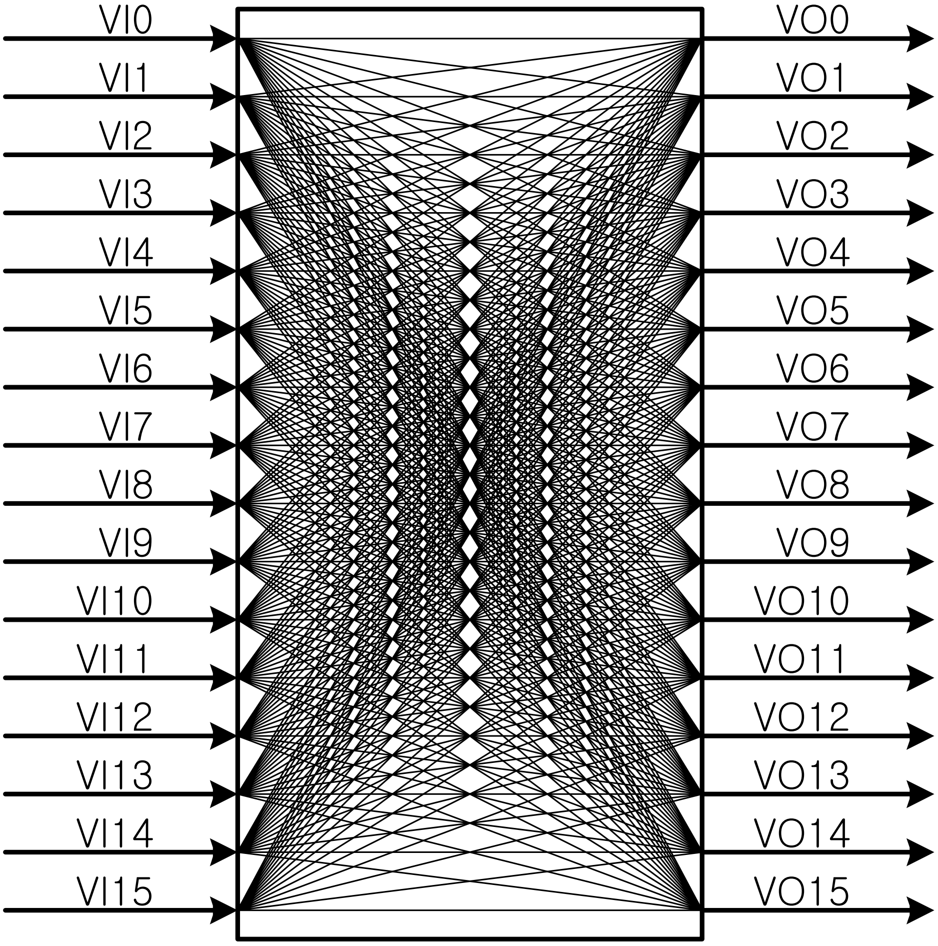
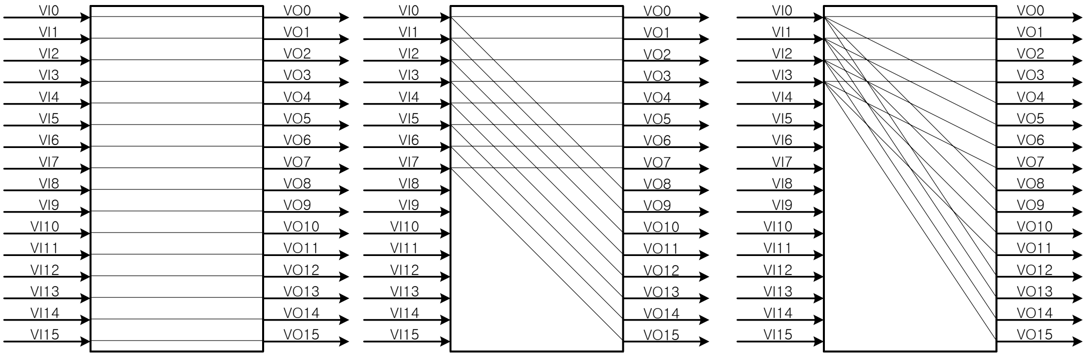


Fig. 4-20 Video Matrix Switch

Fig. 4-21 shows examples of the video matrix switch. Fig. 4-21(a), (b) and (c) illustrate a normal connection at the initial condition, double connection using SOLO6010-9 and quadruple connection using SOLO6010-4. The multiple connections enable to produce more various MPEG4 video streams with virtual video channel encoding which is very important to support the network clients on various internet environments. Although SOLO6010-9 and SOLO6010-4 are restricted to encoding video channel, it is possible to encode 16channel MPEG4 encoding using the multiple connections. As shown in Table 4-17, *Internal Input Video Ch0 Registers* to *Internal Input Video Ch15 Registers* [0x0100~0x0108] is used to control the video matrix switch.



(a) (b) (c)

Fig. 4-21 Examples of the Video Matrix Switch

Table 4-17 Video Input Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0100	VI_CH_SWITCH0	RW	[29:25]	*Video Input Channel Number for Internal Video Ch5.*	5
			[24:20]	*Video Input Channel Number for Internal Video Ch4.*	4
			[19:15]	*Video Input Channel Number for Internal Video Ch3.*	3
			[14:10]	*Video Input Channel Number for Internal Video Ch2.*	2
			[19:5]	*Video Input Channel Number for Internal Video Ch1.*	1
			[4:0]	*Video Input Channel Number for Internal Video Ch0.*	0
0x0104	VI_CH_SWITCH1	RW	[29:25]	*Video Input Channel Number for Internal Video Ch11.*	11
			[24:20]	*Video Input Channel Number for Internal Video Ch10.*	10
			[19:15]	*Video Input Channel Number for Internal Video Ch9.*	9
			[14:10]	*Video Input Channel Number for Internal Video Ch8.*	8
			[19:5]	*Video Input Channel Number for Internal Video Ch7.*	7
			[4:0]	*Video Input Channel Number for Internal Video Ch6.*	6
0x0108	VI_CH_SWITCH2	RW	[24:20]	*Video Input Channel Number for Internal Spot Output.*	16
			[19:15]	*Video Input Channel Number for Internal Video Ch15.*	15
			[14:10]	*Video Input Channel Number for Internal Video Ch14.*	14
			[19:5]	*Video Input Channel Number for Internal Video Ch13.*	13
			[4:0]	*Video Input Channel Number for Internal Video Ch12.*	12
0x010C	VI_CH_ENA	RW	[15:0]	*Video Input Channel Enable.* Active HIGH. Bit[15:0] : {Ch15, Ch14, Ch13, … , Ch0}	1
0x0110	VI_CH_FORMAT	RW	[31:16]	*Video Input Format Decoder Selection1.* Bit[15:0] : {Ch15, Ch14, Ch13, … , Ch0} [0] : Primary [1] : Secondary	0

			[15:0]	*Video Input Format Decoder Selection0.*
				Bit[15:0] : {Ch15, Ch14, Ch13, … , Ch0}
				[0] : Interlace
				[1] : Progressive
				0x011C : VIN{Primary, Interlaced}
				0x0120 : VIN{Secondary, Interlaced}
				0x0124 : VIN{Don't care, Progressive}
					0

0x0114	VI_FMT_CFG	RW	[31]	*Vertical Line Check.* The register is effective for all video input channels. The video standard or stability is checked by counting vertical line. The response for all video input channels will be stored on register VI_STATUS1[15:0]. Refer to register address 0x012C.	0

			[30]	*Horizontal Pixel Check.* The register is effective for all video input channels. Video standard or stability is checked by counting horizontal pixel. The response for all video input channels will be stored on register VI_STATUS1[15:0]. Refer to register address 0x012C.	0
			[29]	*Reserved. (Video Standard or Unstable Check)*	0
			[28]	*Test Signal Generation.* (Only for Test)	0
			[27]	*Mosaic Horizontal Length.* [0] : 16 [1] : 32	0
			[25:22]	*Mosaic Darkness Strength.* The brightness in mosaic area is controlled by the registers. R = O >> N Where, R is the result, O is the original video data, N is the value on the register.	0
			[21:11]	*Display Horizontal Offset.* The register is effective for all video input channels.	0
			[10:0]	*Compression Horizontal Offset.* The register is effective for all video input channels.	0
0x0118	VI_PAGE_SWITCH	RW	[16]	*Reserved. (Input Synchronization on Field Index)*	0
			[15]	*Reserved. (Input Field Index for Calculation)*	0
			[14]	*Reserved. (Output Field Index for Calculation)*	0
			[1:0]	*Writing Page Distance.* Four frame buffers are used to display video output. For avoiding the frame overlapping,	2
0x011C	VI_ACT_PRI_INT	RW	[31]	*Field Index Inversion.*	0
			[30]	*Reserved. (Test Mode)*	0
			[29:21]	*Horizontal Start.*	8
			[20:11]	*Vertical Start.*	0

			[10:0]	*Vertical End.*	242(N)
					290(P)

0x0120	VI_ACT_SEC_INT	RW	[31]	*Field Index Inversion.*	0
			[30]	*Reserved. (Test Mode)*	0
			[29:21]	*Horizontal Start.*	8
			[20:11]	*Vertical Start.*	0

			[10:0]	*Vertical End.*	242(N)
					290(P)

0x0124	VI_ACT_PRI_PRG	RW	[31]	*Field Index Inversion.*	0
			[30]	*Reserved. (Test Mode)*	0
			[29:21]	*Horizontal Start.*	8
			[20:11]	*Vertical Start.*	0

			[10:0]	*Vertical End.*	242(N)
					290(P)

0x0128	VI_STATUS0	RO	[11:4]	*Reserved. (Field Free Count)*	-
			[3]	*Video Display FI.* The register shows the current displaying field index.	-
			[2:0]	*Video Display Page.* The register shows the current displaying page.	-
0x012C	VI_STATUS1	RO	[27:24]	*Reserved. (DMA Status)*	-
			[23:16]	*Reserved. (Capture Queue Status)*	-
			[15:0]	*Standard and Stability Flag.* The registers show if the video input signal is standard or loss. However, the loss detection function of the video color decoder is strongly recommended for loss detection.	-

Table 4-18 shows *Internal MPEG4 Video Decoder Controller Registers* [0x0130~0x013C].

Table 4-18 Internal MPEG4 Video Decoder Controller Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State

0x0130	VI_PB_CONFIG	RW	[1]	*Playback Video Mode.* [0] : Default [1] : User Programming	
			[0]	*Playback Video System.* [0] : NTSC [1] : PAL	
0x0134	VI_PB_RANGE_HV	RW	[21:12]	*Playback Horizontal Size.*	
			[11:0]	*Playback Vertical Size.*	
0x0138	VI_PB_ACT_H	RW	[21:12]	*Playback Horizontal Size.*	
			[11:0]	*Playback Vertical Size.*	
0x013C	VI_PB_ACT_V	RW	[21:12]	*Playback Horizontal Size.*	
			[11:0]	*Playback Vertical Size.*	

The SOLO6010 supports mosaic for privacy area. Table 4-19 shows *Video Input Mosaic Registers* [0x0140~0x017C]. Fig. 4-22 illustrates the video input mosaic, which is only one mosaic area effective for each video input channel.

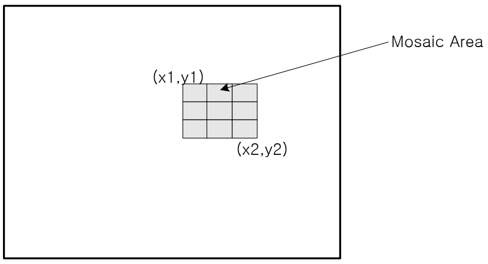


Fig. 4-22 Video Input Mosaic

*Horizontal Start * 16 Register* [0x0140, 0x0144, 0x0148, 0x014C, 0x0150, 0x0154, 0x0158, 0x015C, 0x0160, 0x0164, 0x0168, 0x016C, 0x0170, 0x0174, 0x0178, 0x017C] points x1 in Fig. 4-22. *Horizontal End * 16 Register* [0x0140, 0x0144, 0x0148, 0x014C, 0x0150, 0x0154, 0x0158, 0x015C, 0x0160, 0x0164, 0x0168, 0x016C, 0x0170, 0x0174, 0x0178, 0x017C] points x2 in Fig. 4-22. *Vertical Start * 8 Register* [0x0140, 0x0144, 0x0148, 0x014C, 0x0150, 0x0154, 0x0158, 0x015C, 0x0160, 0x0164, 0x0168, 0x016C, 0x0170, 0x0174, 0x0178, 0x017C] points y1 in Fig. 4-22. *Vertical End * 8 Register* [0x0140, 0x0144, 0x0148, 0x014C, 0x0150, 0x0154, 0x0158, 0x015C, 0x0160, 0x0164, 0x0168, 0x016C, 0x0170, 0x0174, 0x0178, 0x017C] points y2 in Fig. 4-22.

Table 4-19 Video Input Mosaic Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0140 0x0144 0x0148	VI_WIN_MOSAIC0 VI_WIN_MOSAIC1 VI_WIN_MOSAIC2	RW	[31:24]	*Horizontal Start * 16.*	0
			[23:16]	*Horizontal End * 16.*	0
			[15:8]	*Vertical Start * 8.*	0

0x014C 0x0150 0x0154 0x0158 0x015C 0x0160 0x0164 0x0168 0x016C 0x0170 0x0174 0x0178 0x017C	VI_WIN_MOSAIC3 VI_WIN_MOSAIC4 VI_WIN_MOSAIC5 VI_WIN_MOSAIC6 VI_WIN_MOSAIC7 VI_WIN_MOSAIC8 VI_WIN_MOSAIC9 VI_WIN_MOSAIC10 VI_WIN_MOSAIC11 VI_WIN_MOSAIC12 VI_WIN_MOSAIC13 VI_WIN_MOSAIC14 VI_WIN_MOSAIC15		[7:0]	*Vertical End * 8.*	0

Table 4-20 shows *Windows Control Registers* [0x0180~0x0244]. The display window layout can be displayed as examples in Fig. 4-23. Each display windows contains channel ID and scale mode, and display only one of live or playback video for each video input channel. Therefore, each live or playback video channel will be displayed fitted to given window scale mode of given channel ID. However, PIP window contains totally independent window and should be set to its own position and scale mode.

Yellow circles and the number in center points channel ID and video scale mode in Fig. 4-23, respectively.

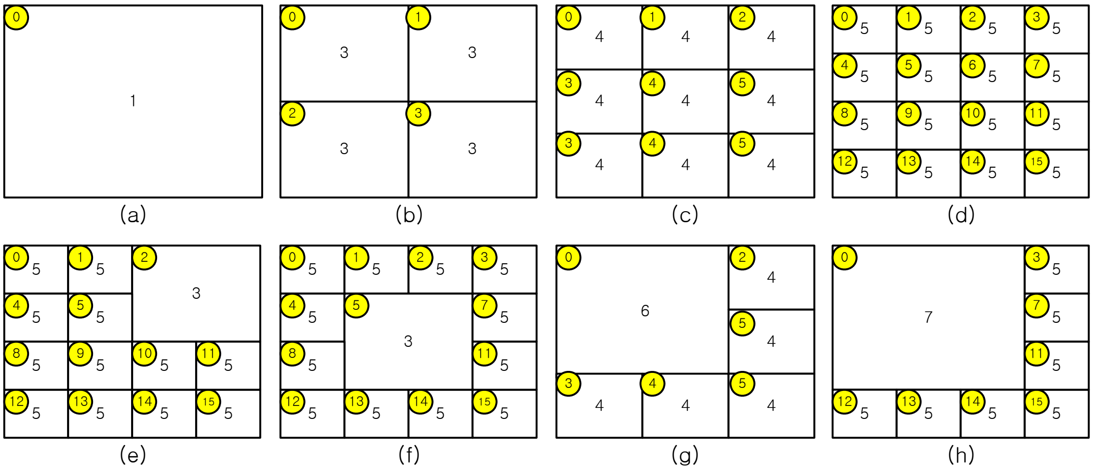


Fig. 4-23 Examples of Display Control

Fig. 4-24 shows examples of PIP. Fig. 4-24(a) and (b) illustrate 1/3 scale and 1/4 scale PIP display, respectively.

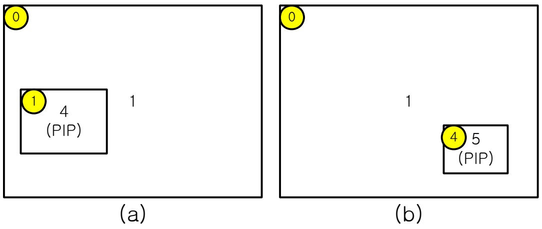


Fig. 4-24 Examples of PIP

Live Channel ID for Window Register [0x0180, 0x0184, 0x0188, 0x018C, 0x0190, 0x0194, 0x0198, 0x019C, 0x01A0, 0x01A4, 0x01A8, 0x01AC, 0x01B0, 0x01B4, 0x01B8, 0x01BC] means channel ID as shown in Fig. 4-23 and Fig. 4-24. *PIP On Register* [0x0180, 0x0184, 0x0188, 0x018C, 0x0190, 0x0194, 0x0198, 0x019C, 0x01A0, 0x01A4, 0x01A8, 0x01AC, 0x01B0, 0x01B4, 0x01B8, 0x01BC] is needed to be set to 1 to activate PIP mode as shown in Fig. 4-24. *Scale Mode Register* [0x0180, 0x0184, 0x0188, 0x018C, 0x0190, 0x0194, 0x0198, 0x019C, 0x01A0, 0x01A4, 0x01A8, 0x01AC, 0x01B0, 0x01B4, 0x01B8, 0x01BC] determines the picture size on video display output by controlling video scale. *Window Horizontal Start * 4 Register* [0x0180, 0x0184, 0x0188, 0x018C, 0x0190, 0x0194, 0x0198, 0x019C, 0x01A0, 0x01A4, 0x01A8, 0x01AC, 0x01B0, 0x01B4, 0x01B8, 0x01BC] and *Window Horizontal End *4 Register* [0x0180, 0x0184, 0x0188, 0x018C, 0x0190, 0x0194, 0x0198, 0x019C, 0x01A0, 0x01A4, 0x01A8, 0x01AC, 0x01B0, 0x01B4, 0x01B8, 0x01BC] points the horizontal position as shown in Fig. 4-23 and Fig. 4-24. *Window Vertical Start * 4 Register* [0x01C0, 0x01C4, 0x01C8, 0x01CC, 0x01D0, 0x01D4, 0x01D8, 0x01DC, 0x01E0, 0x01E4, 0x01E8, 0x01EC, 0x01F0, 0x01F4, 0x01F8, 0x01FC] and *Window Vertical End *4 Register* [0x01C0, 0x01C4, 0x01C8, 0x01CC, 0x01D0, 0x01D4, 0x01D8, 0x01DC, 0x01E0, 0x01E4, 0x01E8, 0x01EC, 0x01F0, 0x01F4, 0x01F8, 0x01FC] points the vertical position as shown in Fig. 4-23 and Fig. 4-24.

Playback On Register [0x0200, 0x0204, 0x0208, 0x020C, 0x0210, 0x0214, 0x0218, 0x021C, 0x0220, 0x0224, 0x0228, 0x022C, 0x0230, 0x0234, 0x0238, 0x023C] and *Live On Register* [0x0200, 0x0204, 0x0208, 0x020C, 0x0210, 0x0214, 0x0218, 0x021C, 0x0220, 0x0224, 0x0228, 0x022C, 0x0230, 0x0234, 0x0238, 0x023C] are for window display source selection. One of live or playback can be displayed on each window.

Live On Delay Register [0x0240] is used to delay prior to display live video output when switching it because the garbage data can be displayed when switching the live video output. *Live Auto Mute Register* [0x0244] will force to mute the live video output for nonstandard or unstable video input signal when it is set to 1. However, it cannot detect video loss when the video decoder chip is set to auto free running. For video loss detection, the video loss detection function of the video decoder is strongly recommended.

Table 4-20 Windows Control Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0180 0x0184 0x0188 0x018C 0x0190 0x0194 0x0198 0x019C 0x01A0 0x01A4 0x01A8 0x01AC 0x01B0 0x01B4	VI_WIN_CTRL0 VI_WIN_CTRL1 VI_WIN_CTRL2 VI_WIN_CTRL3 VI_WIN_CTRL4 VI_WIN_CTRL5 VI_WIN_CTRL6 VI_WIN_CTRL7 VI_WIN_CTRL8 VI_WIN_CTRL9 VI_WIN_CTRL10 VI_WIN_CTRL11 VI_WIN_CTRL12 VI_WIN_CTRL13	RW	[31:28]	*Live Channel ID for Window.*	0
			[27]	*PIP On.*	0
			[26:24]	*Scale Mode.* [0] : Off [1] : H=1/1, V=1/1 [2] : H=1/2, V=1/1 [3] : H=1/2, V=1/2 [4] : H=1/3, V=1/3 [5] : H=1/4, V=1/4 [6] : H=2/3, V=2/3 [7] : H=3/4, V=3/4 [8]~[15] : H=1/1, V=1/1 for PIP	0
			[23:12]	*Window Horizontal Start * 4.*	0

0x01B8 0x01BC	VI_WIN_CTRL14 VI_WIN_CTRL15		[11:0]	*Window Horizontal End * 4.*	0
0x01C0 0x01C4 0x01C8 0x01CC 0x01D0 0x01D4 0x01D8 0x01DC 0x01E0 0x01E4 0x01E8 0x01EC 0x01F0 0x01F4 0x01F8 0x01FC	VI_WIN_CTRL0 VI_WIN_CTRL1 VI_WIN_CTRL2 VI_WIN_CTRL3 VI_WIN_CTRL4 VI_WIN_CTRL5 VI_WIN_CTRL6 VI_WIN_CTRL7 VI_WIN_CTRL8 VI_WIN_CTRL9 VI_WIN_CTRL10 VI_WIN_CTRL11 VI_WIN_CTRL12 VI_WIN_CTRL13 VI_WIN_CTRL14 VI_WIN_CTRL15	RW	[23:12]	*Window Vertical Start.*	0
			[11:0]	*Window Vertical End.*	0
0x0200 0x0204 0x0208 0x020C 0x0210 0x0214 0x0218 0x021C 0x0220 0x0224 0x0228 0x022C 0x0230 0x0234 0x0238 0x023C	VI_WIN_ON0 VI_WIN_ON1 VI_WIN_ON2 VI_WIN_ON3 VI_WIN_ON4 VI_WIN_ON5 VI_WIN_ON6 VI_WIN_ON7 VI_WIN_ON8 VI_WIN_ON9 VI_WIN_ON10 VI_WIN_ON11 VI_WIN_ON12 VI_WIN_ON13 VI_WIN_ON14 VI_WIN_ON15	RW	[1]	*Playback On.*	0
			[0]	*Live On.*	0
0x0240	VI_WIN_SW	RW	[4:0]	*Live On Delay*. (Frame Interval)	0
0x0244	VI_WIN_MUTE	RW	[0]	*Live Auto Mute.*	0

Fig. 4-25 shows the video input motion detection. The motion grid is 32 pixels by 32 pixels and the motion detection flags are 32 samples by 32 samples. Therefore, the motion detection engine will cover 1,024 pixels by 1,024 pixels sized picture for each 16 channel video input. The left figure in Fig. 4-25 represents the memory allocation map for the external SDRAM. The memory structure is composed of the motion flag table area (512 bytes per channel), threshold table (8,192 bytes per channel) and working cache (8,192 bytes per channel) as shown in Fig. 4-25. The structure of each motion flag is 32 samples by 32 samples as shown as the right figure in Fig. 4-25. Therefore, 22 samples by 15 samples will be needed for NTSC video input, and 22 samples by 18 samples will be needed for PAL video input.

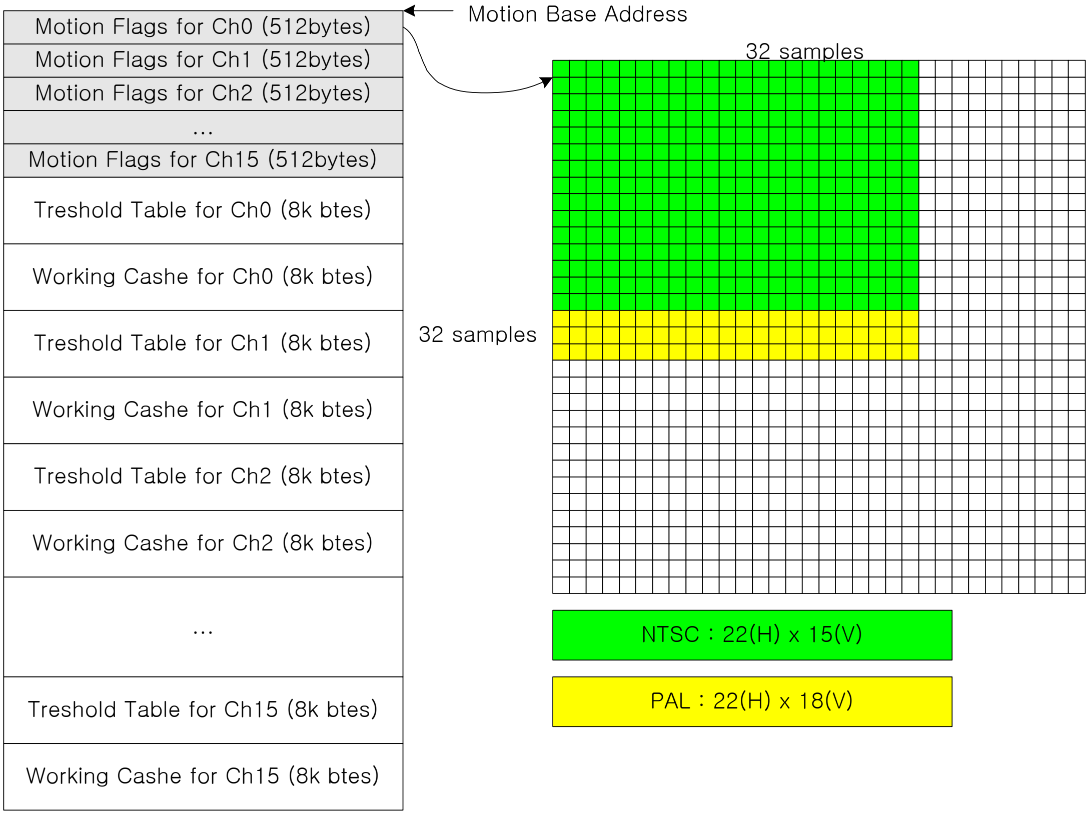
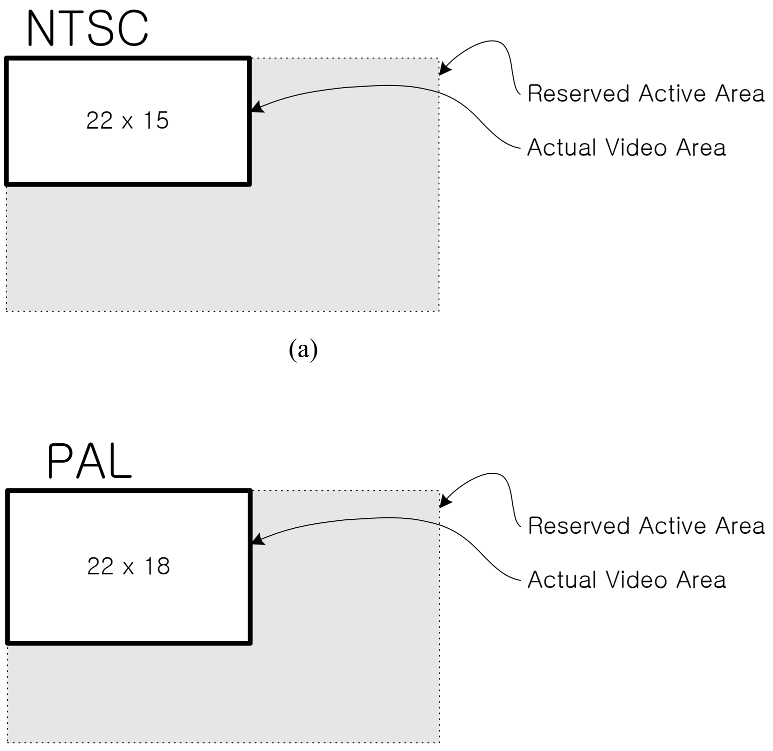

Motion Flags for Ch0 (512bytes)
Motion Flags for Ch1 (512bytes)
Motion Flags for Ch2 (512bytes)
…
Motion Flags for Ch15 (512bytes)
Treshold Table for Ch0 (8k btes)
Working Cashe for Ch0 (8k btes)
Treshold Table for Ch1 (8k btes)
Working Cashe for Ch1 (8k btes)
Treshold Table for Ch2 (8k btes)
Working Cashe for Ch2 (8k btes)
…
Treshold Table for Ch15 (8k btes)
Working Cashe for Ch15 (8k btes)
Motion Base Address
32 samples
32 samples
NTSC : 22(H) x 15(V)
PAL : 22(H) x 18(V)

Fig. 4-25 Video Input Motion Detection

The motion threshold table is reserved to 32 samples by 32 samples fixed size as mentioned above. However, the actual working size for a system is smaller as shown in Fig. 4-26.



(a)

(b)

Fig. 4-26 Motion Threshold Table Map

Fig. 4-27 shows an example of motion threshold table programming. The range of motion threshold is 0 to 65,536. The smaller value is needed to be set for higher motion sensitivity. Each motion threshold can have different value for configuring different sensitivity for any motion area.

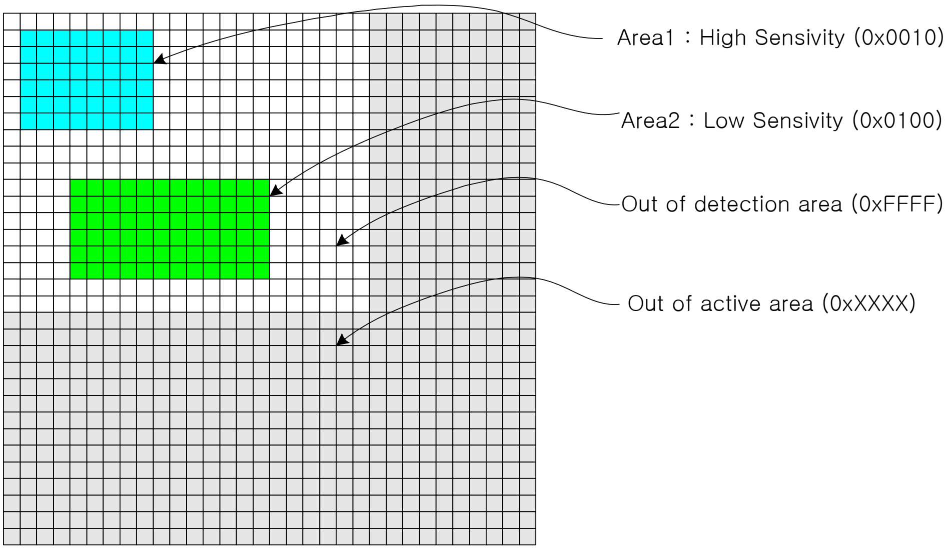

Area1 : High Sensivity (0x0010)
Area2 : Low Sensivity (0x0100)
Out of detection area (0xFFFF)
Out of active area (0xXXXX)

Fig. 4-27 Example of Motion Threshold Table Programming

Fig. 4-28 shows motion interrupt mode. Motion detection interrupt will be asserted for every motion event when Interrupt Mode Register [0x0264] is set to 0. Motion detection interrupt will be asserted for the first motion event and sustained during the predefined frame sustain period after the last motion event. The predefined frame sustain period is defined on *Motion Sustain Frame Count Register* [0x0264].

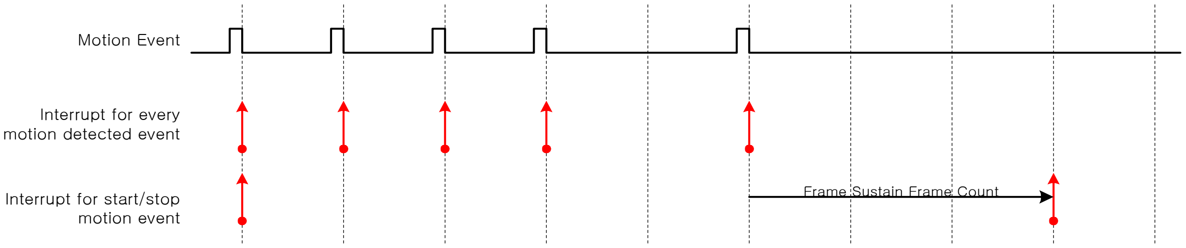

Motion Event
Interrupt for every motion detected event
Interrupt for start/stop motion event
Frame Sustain Frame Count

Fig. 4-28 Motion Interrupt Mode

Table 4-21 shows *Video Input Motion Detection Registers* [0x0260~0x0274]. *Motion Detection Enable Register* [0x0260] is set to 1 to activate video input motion detection function for each video input channel. *Motion Base Address Register* [0x0260] points the base address on the external memory that the motion detected flags are stored. When *Interrupt Mode Register* [0x0264] is set to 1, the detected motion data are sustained as the value of *Motion Sustain Frame Count Register* [0x0264] for CPU to read them stably. *Maximum Sample Length Register* [0x0264] should be set as horizontal picture size/32. When *Interrupt Mode Register* [0x0264] is set to 0, the motion detection interrupt will be asserted for every motion event. When *Interrupt Mode Register* [0x0264] is set to 1, the motion detection interrupt will be asserted and sustained during the time as long as the value of *Motion Sustain Frame Count Register* [0x0264]. *When Freeze Motion Data Register* [0x0264] is set to 1, the motion data will be kept after motion event. *Minimum Detected Sample Count Register* [0x0264] points the threshold value to avoid misdetection by noise. *Motion Flag Clear Register* [x0268] is used to clear the motion detected data for each video input channel. *Motion Detected Flag Register* [0x026C] will the motion detection result for each video input channel.

Motion detection cells are consisted of 32 by 32 pixels. Motion detection result can be displayed with border and bar. When *Border Y Set On Register* [0x0270], *Border CR Set On Register* [0x0270] and *Border CB Set On Register* [0x0270] are set to 1, the border will be displayed with the values of *Border Y Value Register* [0x0270], *Border CB Value Register* [0x0270] and *Border CB Value Register* [0x0270]. When *Border Y Addition On Register* [0x0270], *Border CR Addition On Register* [0x0270] and *Border CB Addition On Register* [0x0270] are set to 1, the border will be displayed with adding the values of *Border Y Value Register* [0x0270], *Border CB Value Register* [0x0270] and *Border CB Value Register* [0x0270].

When *Bar Y Set On Register* [0x0270], *Bar CR Set On Register* [0x0270] and *Bar CB Set On Register* [0x0270] are set to 1, the border will be displayed with the values of *Bar Y Value Register* [0x0270], *Bar CB Value Register* [0x0270] and *Bar CB Value Register* [0x0270]. When *Bar Y Addition On Register* [0x0270], *Bar CR Addition On Register* [0x0270] and *Bar CB Addition On Register* [0x0270] are set to 1, the border will be displayed with adding the values of *Bar Y Value Register* [0x0270], *Bar CB Value Register* [0x0270] and *Bar CB Value Register* [0x0270].

Table 4-21 Video Input Motion Detection Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0260	VI_MOTION_ADR	RW	[31:16]	*Motion Detection Enable.* Bit[31:16] = {Ch15, Ch14, …, Ch0}	0
			[15:0]	*Motion Base Address.*	0
0x0264	VI_MOTION_CTRL	RW	[31:24]	*Motion Sustain Frame Count.*	0
			[21:16]	*Maximum Sample Length.*	0
			[15]	*Interrupt Mode.* [0] : Every Motion Event [1] : Start/Stop Motion Event.	0
			[14]	*Freeze Motion Data.*	0
			[13:0]	*Minimum Detected Sample Count.*	0
0x0268	VI_MOTION_CLEAR	RW	[15:0]	*Motion Flag Clear.*	0
0x026C	VI_MOTION_STA	RO	[15:0]	*Motion Detected Flag.* Bit[15:0] = {ch15, ch14, ch13, …, ch0} [0] : Non motion detected video input channel [1] : Motion detected video input channel	-
0x0270	VI_MOTION_BDR	RW	[29]	*Border Y Set On.*	0
			[28]	*Border Y Addition On.*	0
			[27]	*Border CB Set On.*	0
			[26]	*Border CB Addition On.*	0
			[25]	*Border CR Set On.*	0
			[24]	*Border CR Addition On.*	0
			[23:16]	*Border Y Value.*	0

			[15:8]	*Border CB Value.*	0
			[7:0]	*Border CR Value.*	0
0x0274	VI_MOTION_BAR	RW	[29]	*Bar Y Set On.*	0
			[28]	*Bar Y Addition On.*	0
			[27]	*Bar CB Set On.*	0
			[26]	*Bar CB Addition On.*	0
			[25]	*Bar CR Set On.*	0
			[24]	*Bar CR Addition On.*	0
			[23:16]	*Bar Y Value.*	0
			[15:8]	*Bar CB Value.*	0
			[7:0]	*Bar CR Value.*	0

Fig. 4-29 shows the waveform of 27MHz ITU-R 656 format. Pin iVDO_INIT_SELT (G18) is needed to set to 0 or 1 for NTSC or PAL system, respectively. The memory size for the video output requires 4 display buffer pages. Each display buffer size is 480 x 2,048 for NTSC system or 576 x 2,048 for PAL system. The video output includes background auto erase to erase all display memory with background color, digital x2 zoom with vertical video quality compensation, freeze with interpolation, and interpolation for the live freeze and the decoded video.

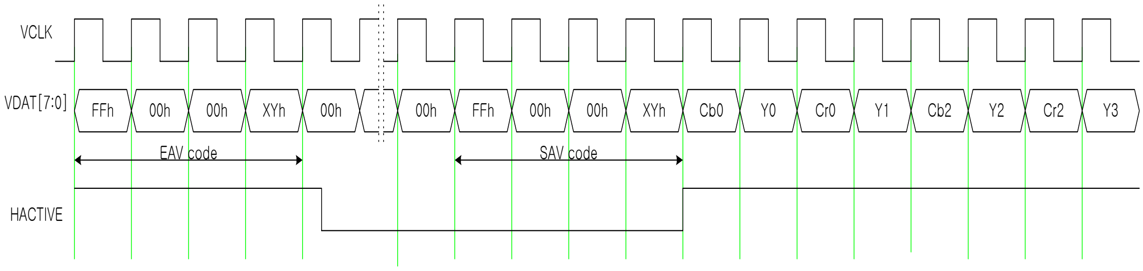


Fig. 4-29 Waveform of 27MHz ITU-R 656 format

Fig. 4-30 shows the video and layers of the video output. At the video point, the layer order is Cursor, OSG, Border, PIP Video and Normal Video.

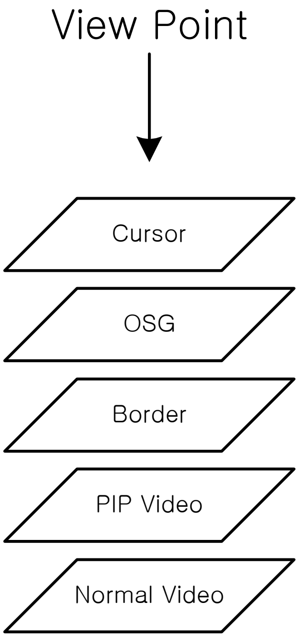


Fig. 4-30 Video and Graphic Layers

Table 4-22 shows *Video Output Configuration Registers* [0x02F4~0x030C]. *Field Index Transition Position F0 Register* [0x02F4], *VSYNC Start Position F0 Register* [0x02F4], *VSYNC End Position F0* [0x02F4] and *Number of Vertical Line F0* [0x02F4] will determine the waveform for field0. *Field Index Transition Position F1 Register* [0x02F4], *VSYNC Start Position F1 Register* [0x02F4], *VSYNC End Position F1* [0x02F4] and *Number of Vertical Line F1* [0x02F4] will determine the waveform for field1. These values are only effective when *User 656 Format Register* [0x0300] is set to 1 and the appropriate values are defined in the SOLO6010 driver software. *Video System Register* [0x0300] is set to 0 for NTSC system or 1 for PAL system. *FI, V Change Enable Register* [0x0300] is set to 1 to change the position of field0 and field1. *User Color Set for VSYNC Register* [0x0300], *User Color Set for VSYNC Register* [0x0300], *User Color Set for Non Active of H Register* [0x0300] and *User Color Set for Non Active of V Register* [0x0300] are set to 1 to define the color as *Non Active Color for Y Register* [0x0300], *Non Active Color for CB/16 Register* [0x0300] and *Non Active Color for CR/16 Register* [0x0300] is configured. *H Blank Size Register* [0x0304], *H Start Position Register* [0x0304], *H End Position Register* [0x0304], *V Blank Size Register* [0x0308], *V Start Position Register* [0x0308] and *V End Position Register* [0x0308] are used to define the vertical blank sizes and the horizontal picture positions. *SYNC Inversion Register* [0x030C], *HSYNC Inversion Register* [0x030C] and *VSYNC Inversion Register* [0x030C] are used to inverse the polarity of the video synchronization signals and effective for VGA interface.

Table 4-22 Video Output Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x02F4	VO_USER656_F0	RW	[29:24]	*Field Index Transition Position F0.*	0
			[23:17]	*VSYNC Start Position F0.*	0
			[16:11]	*VSYNC End Position F0.*	0
			[10:0]	*Number of Vertical Line F0.*	0
0x02F8	VO_USER656_F1	RW	[29:24]	*Field Index Transition Position F1.*	0
			[23:17]	*VSYNC Start Position F1.*	0
			[16:11]	*VSYNC End Position F1.*	0
			[10:0]	*Number of Vertical Line F1.*	0
0x0300	VO_FMT_ENC	RW	[31]	*Reserved.*	0
			[30]	*Video System.* [0] : NTSC [1] : PAL	0
			[29]	*User 656 Format Enable.* [0] : Standard 656 format [1] : User Programmed 656 format	0
			[28]	*Reserved.*	0
			[26]	*Reserved.*	0
			[25]	*Reserved.*	0
			[23:0]	*User Frame Size Clock Count.* (When 26 is on)	0
			[20]	*FI, V Change Enable.* (When SAV)	0
			[19]	*User Color Set for VSYNC.*	0
			[18]	*User Color Set for HSYNC.*	0
			[17]	*User Color Set for Non Active of H.*	0
			[16]	*User Color Set for Non Active of V.*	0
			[15:8]	*Non Active Color for Y.*	0
			[7:4]	*Non Active Color for CB/16.*	0
			[3:0]	*Non Active Color for CR/16.*	0
0x0304	VO_ACT_H	RW	[31:22]	*H Blank Size.*	0
			[21:11]	*H Start Position.*	6
			[10:0]	*H End Position.*	710
0x0308	VO_ACT_V	RW	[31:22]	*V Blank Size.*	0
			[21:11]	*V Start Position.*	8

			[10:0]	*V End Position.*	248(N)
					296(P)

0x030C	VO_RANGE_HV	RW	[24]	*SYNC Inversion.*	0
			[23]	*HSYNC Inversion.*	0
			[22]	*VSYNC Inversion.*	0
			[21:11]	*Horizontal Size.*	704

			[10:0]	*Vertical Size.*	240(N)
					288(P)

Table 4-23 shows *Video Output Control Registers* [0x0310~0x0324]. *Display On Register* [0x0310] is set to 1 to activate the video output. *Erase Frame Count Register* [0x0310] is used to delay after erasing display using *Display Erase On Register* [0x0314]. *Display Base Address Register* [0x0310] points the base address on external SDRAM. *Vertical Zoom On Register* [0x0318] and *Horizontal Zoom On Register* [0x0318] is set to 1 for the vertical and horizontal digital x2 zoom. When *Vertical Compensation Register* [0x0318] is set to 1, the vertical video compensation algorithm will be applied to improve the vertical video quality automatically. *Zoom Offset of Horizontal*2 Register* [0x0318] and *Zoom Offset of Vertical Register* [0x0318] will determine the position of the zoomed video. *Freeze Live Register* [0x031C] is set to freeze the video. When *Freeze Live Interpolation Register* [0x031C] is set to 1, the video output will be interpolated using one field to remove the tremble effect by time difference of two video fields. *Background Color Y Register* [0x0320], *Background Color U Register* [0x0320] and *Background Color V Register* [0x0320] will determine the background video color.

Table 4-23 Video Output Control Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0310	VO_DISP_CFG	RW	[31]	*Display On.*	0
			[27:24]	*Erase Frame Count. (After ease off to on)*	4
			[22]	*Double Scan.* [0] : 30Hz [1] : 60Hz	0
			[21]	*Display Single Page.*	0
			[15:0]	*Display Base Address.*	0
0x0314	VO_DISP_ERASE	RW	[1]	*Display Erase On.*	1
0x0318	VO_ZOOM_CTRL	RW	[24]	*Vertical Zoom On.*	0
			[23]	*Horizontal Zoom On.*	0
			[22]	*Vertical Compensation. (When Vertical Zoom)*	0
			[21:11]	*Zoom Offset of Horizontal*2.*	0
			[10:0]	*Zoom Offset of Vertical.*	0
0x031C	VO_FREEZE_CTRL	RW	[1]	*Freeze Live.*	0
			[0]	*Freeze Live Interpolation.*	0
0x0320	VO_BGK_COLOR	RW	[23:16]	*Background Color Y.*	0x10
			[15:8]	*Background Color U.*	0x80
			[7:0]	*Background Color V.*	0x80
0x0324	VO_DEINTERLACE	RW	[19:8]	*Reserved.*	0
			[7:0]	*Reserved.*	0

Table 4-24 shows *Line and Bar Registers* [0x0330~0x037C]. *Border Ext Y Register* [0x0330], *Border Ext U*16 Register* [0x0330] and *Border Ext V*16 Register* [0x0330] will determine the color of external borders. *Bar Cell Y Register* [0x0330], *Bar Cell U*16 Register* [0x0330] and *Bar Cell V*16 Register* [0x0330] will determine the color of cell borders.

Border On 15 Register [0x0338], *Border On 14 Register* [0x0338], *Border On 13 Register* [0x0338], *Border On 12 Register* [0x0338], *Border On 11 Register* [0x0338], *Border On 10 Register* [0x0338], *Border On 9 Register* [0x0338], *Border On 8 Register* [0x0338], *Border On 7 Register* [0x0338], *Border On 6 Register* [0x0338], *Border On 5 Register* [0x0338], *Border On 4 Register* [0x0338], *Border On 3 Register* [0x0338], *Border On 2 Register* [0x0338], *Border On 1 Register* [0x0338] and *Border On 0 Register* [0x0338] are set to 1 to display cell borders of window video channel 15, window video channel 14, window video channel 13, window video channel 12, window video channel 11, window video channel 10, window video channel 9, window video channel 8, window video channel 7, window video channel 6, window video channel 5, window video channel 4, window video channel 3, window video channel 2, window video channel 1 and window video channel 0, respectively.

Bar On 15 Register [0x0338], *Bar On 14 Register* [0x0338], *Bar On 13 Register* [0x0338], *Bar On 12 Register* [0x0338], *Bar On 11 Register* [0x0338], *Bar On 10 Register* [0x0338], *Bar On 9 Register* [0x0338], *Bar On 8 Register* [0x0338], *Bar On 7 Register* [0x0338], *Bar On 6 Register* [0x0338], *Bar On 5 Register* [0x0338], *Bar On 4 Register* [0x0338], *Bar On 3 Register* [0x0338], *Bar On 2 Register* [0x0338], *Bar On 1 Register* [0x0338] and *Bar On 0 Register* [0x0338] are set to 1 to display cell bar of window video channel 15, window video channel 14, window video channel 13, window video channel 12, window video channel 11, window video channel 10, window video channel 9, window video channel 8, window video channel 7, window video channel 6, window video channel 5, window video channel 4, window video channel 3, window video channel 2, window video channel 1 and window video channel 0, respectively.

Cell Position of X1, X2, X3, X4, X5 Register [0x0340, 0x0344, 0x0348, 0x034C, 0x0350] points the cell horizontal positions from left to right. *Cell Position of Y1, Y2, Y3, Y4, Y5 Register* [0x0340, 0x0344, 0x0348, 0x034C, 0x0350] points the cell vertical positions from top to bottom.

Two external borders and two external bars are supported to draw outside border and PIP border or highlight for particular position. *Ext Border On1 Register* [0x0368] and *Ext Bar On1 Register* [0x0368] are set to 1 to draw the first external border and bar. *Ext Start X1 Register* [0x036C] and *Ext Start Y1 Register* [0x036C], *Ext End X1 Register* [0x0370] and *Ext Y1 Register* [0x0370] determined the potion of the first external border and bar. *Ext Border On2 Register* [0x0374] and *Ext Bar On2 Register* [0x0374] are set to 1 to draw the second external border and bar. *Ext Start X2 Register* [0x0378], *Ext Start Y2 Register* [0x0378], *Ext End X2 Register* [0x037C] and *Ext Y2 Register* [0x037C] determined the potion of the second external border and bar.

Fig. 4-31 shows an example of windows layout and border/bar draw. Fig. 4-31 (a), (b) and (c) show window layout, live or playback on/off and display, respectively. Fig. 4-31 (d), (e) and (f) show cell layout, border or bar mask draw and external border draw, respectively. At this example, cell border and cell bar should be set to 0 for cell 0, 1, 2, 4, 5, 6, 8, 9 and 10. Cell border should be set to 1 and cell bar should be set to 0 for cell 3, 7, 11, 12, 13 and 15. Cell border should be set to 1 and cell bar should be set to 1 for cell 14. Finally, the video output will display the picture as Fig. 4-31 (g).

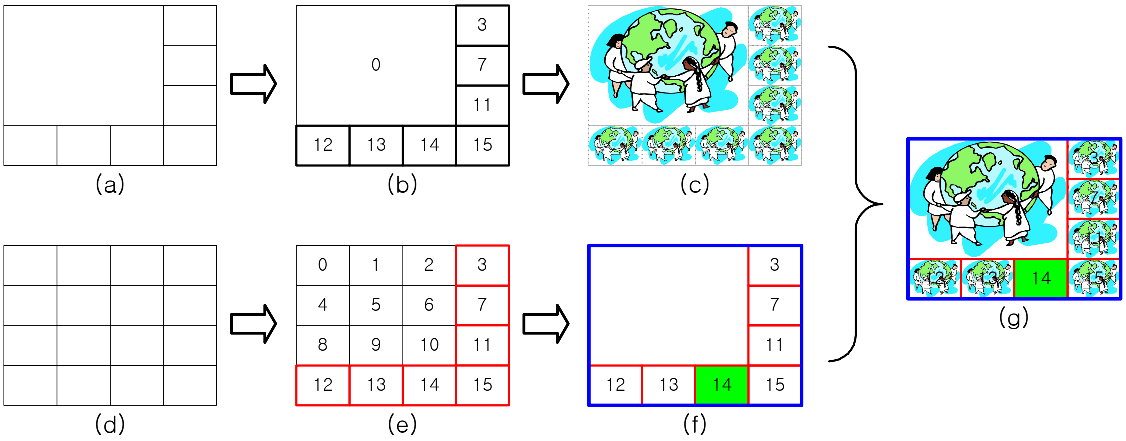


Fig. 4-31 Example of Windows Layout and Border/Bar Draw

Table 4-24 Line and Bar Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0330	VO_BDG_COLOR	RW	[31:24]	*Border Ext Y.*	0x80
			[23:20]	*Border Ext U*16.*	0x8
			[19:16]	*Border Ext V*16.*	0x8
			[15:18]	*Border Cell Y.*	0x10
			[8:4]	*Border Cell U*16.*	0x8
			[3:0]	*Border Cell V*16.*	0x8
0x0334	VO_BAR_COLOR	RW	[31:24]	*Bar Ext Y.*	0x80
			[23:20]	*Bar Ext U*16.*	0x8
			[19:16]	*Bar Ext V*16.*	0x8
			[15:18]	*Bar Cell Y.*	0x10
			[8:4]	*Bar Cell U*16.*	0x8
			[3:0]	*Bar Cell V*16.*	0x8
0x0338	VO_LINE_MASK_CELL	RW	[15]	*Border On 15.*	0
			[14]	*Border On 14.*	0
			[13]	*Border On 13.*	0
			[12]	*Border On 12*	0
			[11]	*Border On 11.*	0
			[10]	*Border On 10.*	0
			[9]	*Border On 9.*	0
			[8]	*Border On 8.*	0
			[7]	*Border On 7.*	0
			[6]	*Border On 6.*	0
			[5]	*Border On 5.*	0
			[4]	*Border On 4.*	0
			[3]	*Border On 3.*	0
			[2]	*Border On 2.*	0
			[1]	*Border On 1.*	0
			[0]	*Border On 0.*	0
0x033C	VO_BAR_MASK_CELL	RW	[15]	*Bar On 15.*	0
			[14]	*Bar On 14.*	0
			[13]	*Bar On 13.*	0
			[12]	*Bar On 12.*	0
			[11]	*Bar On 11.*	0
			[10]	*Bar On 10.*	0
			[9]	*Bar On 9.*	0
			[8]	*Bar On 8.*	0
			[7]	*Bar On 7.*	0
			[6]	*Bar On 6.*	0
			[5]	*Bar On 5.*	0
			[4]	*Bar On 4.*	0
			[3]	*Bar On 3.*	0
			[2]	*Bar On 2.*	0
			[1]	*Bar On 1.*	0
			[0]	*Bar On 0.*	0
0x0340 0x0344 0x0348 0x034C 0x0350	VO_CELL_X0 VO_CELL_X1 VO_CELL_X2 VO_CELL_X3 VO_CELL_X4	RW	[9:0]	Cell Position of X1, X2, X3, X4, X5.	0
0x0354 0x0358 0x035C 0x0360 0x0364	VO_CELL_Y0 VO_CELL_Y1 VO_CELL_Y2 VO_CELL_Y3 VO_CELL_Y4	RW	[9:0]	Cell Position of Y1, Y2, Y3, Y4, Y5.	0
0x0368	VO_CELL_EXT_SET1	RW	[1]	*Ext Border* On1.	0
			[0]	*Ext* Bar On1.	0
0x036C	VO_CELL_EXT_START1	RW	[19:10]	*Ext* Start X1.	0
			[9:0]	*Ext* Start Y1.	0
0x0370	VO_CELL_EXT_END1	RW	[19:10]	*Ext* End X1.	0
			[9:0]	*Ext* End Y1.	0

0x0374	VO_CELL_EXT_SET2	RW	[1]	*Ext* Border On2.	0
			[0]	*Ext* Bar On2.	0
0x0378	VO_CELL_EXT_START2	RW	[19:10]	*Ext* Start X2.	0
			[9:0]	*Ext* Start Y2.	0
0x37C	VO_CELL_EXT_END2	RW	[19:10]	*Ext* End X2.	0
			[9:0]	*Ext* End Y2.	0

Table 4-25 shows *Cursor Registers* [0x0380~0x03DC]. The cursor is composed of 16 pixels by 20 lines with 3 different colors. *Position X Register* [0x0380] and *Position Y Register* [0x0380] point the position of the cursor. *1st Color Y Register* [0x0384], *1st Color U*16 Register* [0x0384] and *1st Color V*16 Register* [0x0384] store the first cursor pixel color. *2nd Color Y Register* [0x0384], *2nd Color U*16 Register* [0x0384] and *2nd Color V*16 Register* [0x0384] store the second cursor pixel color. *3rd Color Y Register* [0x0388], *3rd Color U*16 Register* [0x0388] and *3rd Color V*16 Register* [0x0388] store the third cursor pixel color.

Color of 1st Pixel on Horizontal Register [0x0390~0x03DC], *Color of 2nd Pixel on Horizontal Register* [0x0390~0x03DC], *Color of 3rd Pixel on Horizontal Register* [0x0390~0x03DC], *Color of 4th Pixel on Horizontal Register* [0x0390~0x03DC], *Color of 5th Pixel on Horizontal Register* [0x0390~0x03DC], *Color of 6th Pixel on Horizontal Register* [0x0390~0x03DC], *Color of 7th Pixel on Horizontal Register* [0x0390~0x03DC], *Color of 8th Pixel on Horizontal Register* [0x0390~0x03DC], *Color of 9th Pixel on Horizontal Register* [0x0390~0x03DC], *Color of 10th Pixel on Horizontal Register* [0x0390~0x03DC], *Color of 11th Pixel on Horizontal Register* [0x0390~0x03DC], *Color of 12th Pixel on Horizontal Register* [0x0390~0x03DC], *Color of 13th Pixel on Horizontal Register* [0x0390~0x03DC], *Color of 14th Pixel on Horizontal Register* [0x0390~0x03DC], *Color of 15th Pixel on Horizontal Register* [0x0390~0x03DC], *Color of 16th Pixel on Horizontal Register* [0x0390~0x03DC], *Color of 17th Pixel on Horizontal Register* [0x0390~0x03DC], *Color of 18th Pixel on Horizontal Register* [0x0390~0x03DC], *Color of 19th Pixel on Horizontal Register* [0x0390~0x03DC] and *Color of 20th Pixel on Horizontal Register* [0x0390~0x03DC] determine the shape and color of the cursor.

Table 4-25 Cursor Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0380	VO_CURSOR_POS	RW	[21:11]	*Position X.*	0
			[10:0]	*Position Y.*	0
0x0384	VO_CURSOR_CLR1	RW	[31:24]	*1st Color Y.*	0x10
			[23:20]	*1st Color U*16.*	0x8
			[19:16]	*1st Color V*16.*	0x8
			[15:8]	*2nd Color Y.*	0x10
			[7:4]	*2nd Color U*16.*	0x8
			[3:0]	*2nd Color V*16.*	0x8
0x0388	VO_CURSOR_CLR2	RW	[15:8]	*3rd Color Y.*	0x10
			[7:4]	*3rd Color U*16.*	0x8
			[3:0]	*3rd Color V*16.*	0x8
0x0390 0x0394 0x0398 0x039C 0x03A0	VO_CUR_MASK1 VO_CUR_MASK2 VO_CUR_MASK3 VO_CUR_MASK4 VO_CUR_MASK5	RW	[31:30]	*Color of 1st Pixel on Horizontal.*	
			[29:28]	*Color of 2nd Pixel on Horizontal.*	
			[27:26]	*Color of 3rd Pixel on Horizontal.*	
			[25:24]	*Color of 4th Pixel on Horizontal.*	
			[23:22]	*Color of 5th Pixel on Horizontal.*	

0x03A4 0x03A8 0x03AC 0x03B0 0x03B4 0x03B8 0x03BC 0x03C0 0x03C4 0x03C8 0x03CC 0x03D0 0x03D4 0x03D8 0x03DC	VO_CUR_MASK6 VO_CUR_MASK7 VO_CUR_MASK8 VO_CUR_MASK9 VO_CUR_MASK10 VO_CUR_MASK11 VO_CUR_MASK12 VO_CUR_MASK13 VO_CUR_MASK14 VO_CUR_MASK15 VO_CUR_MASK16 VO_CUR_MASK17 VO_CUR_MASK18 VO_CUR_MASK19 VO_CUR_MASK20		[21:20]	*Color of 6th Pixel on Horizontal.*	
			[19:18]	*Color of 7th Pixel on Horizontal.*	
			[17:16]	*Color of 8th Pixel on Horizontal.*	
			[15:14]	*Color of 9th Pixel on Horizontal.*	
			[13:12]	*Color of 10th Pixel on Horizontal.*	
			[11:10]	*Color of 11th Pixel on Horizontal.*	
			[9:8]	*Color of 12th Pixel on Horizontal.*	
			[7:6]	*Color of 13th Pixel on Horizontal.*	
			[5:4]	*Color of 14th Pixel on Horizontal.*	
			[3:2]	*Color of 15th Pixel on Horizontal.*	
			[1:0]	*Color of 16th Pixel on Horizontal.*	

The video output supports 16 bit full color YUV 4:2:2 OSG which is same size as the video display. The alpha blending supports 64 levels and its relationship is as following equation.

D = if O is zero, then Y (no OSG)

= if O is even, then O (OSG without alpha blending)

= else, then $(V * V_A) / 64 + (O * O_A) /64$ (OSG with alpha blending)

Where,

D : Displayed output data

O : OSG source data

V : Video source data

O_A : Alpha blending OSG strength

Y_A : Alpha blending video strength

Table 4-26 shows *OSG Registers* [0x03E0~0x03E8]. *OSG On Register* [0x03E0] is set to 1 to activate the video output OSG. *Color Mute Register* [0x03E0] is set to 1 to display black and white OSG. *Alpha Blending OSG Strength Register* [0x03E0] and *Alpha Blending Video Strength Register* [0x03E0] are configured with the values as the above equation. *OSG Base Address Register* [0x03E0] points the base address on external SDRAM. *OSG Erase On Register* [0x03E4] is used to erase the OSG automatically. *OSG Blink On Register* [0x03E8] is used to blink OSG display and its blink interval is configured using *OSG Blink Interval Register* [0x03E8].

Table 4-26 OSG Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x03E0	VO_OSG_CFG	RW	[31]	*OSG On.*	0
			[28]	*Color Mute.*	0
			[27:22]	*Alpha Blending OSG Strength.*	0
			[21:16]	*Alpha Blending Video Strength.*	0
			[15:0]	*OSG Base Address.*	0
0x03E4	VO_ERASE	RW	[31:24]	*OSG Erase On.*	0
0x03E8	VO_OSG_BLINK	RW	[15:8]	*OSG Blink On.*	0
			[7:4]	*OSG Blink Interval.* [0] : 15 frame [1] : 18 frame	0

4.4. MPEG4 Video Encoder

4.4.1. Overview

MPEG4 is an ISO/IEC standard developed by MPEG (Moving Picture Experts Group). MPEG4 video is standardized for video storage and transmission in many multimedia environments. MPEG4 video of the SOLO6010 is developed full hardwired core. Full search motion estimation, DCT, quantization and VLC (variable length coding) can process one pixel every clock cycle with pipeline. These features enable to process 4 channels @ D1, 8 channels @ HD1 and 16 channels @ CIF in real time.

FEATURE

- ISO/IEC MPEG4 advanced simple profile @ LEVEL 3,2,1
- Quadruple D1 speed MPEG4 video CODEC
- Full hardwired core
- Full search motion estimation with half-pixel resolution for optimal compression efficiency
- Scaled video size : D1, HD1, CIF, QCIF (only for encoding)
- Video interval : N, N/2, N/3, …, N/255 or N-1, N-2, N-3, … where N=30 in NTSC and N=25 PAL
- Adaptive DCT type selection for interlaced video
- Quantization parameter control : 1 to 31
- Key picture interval(GOP size) Control : 1-255
- Watermarking & compressed video encryption

The video encoder of the SOLO6010 supports motion JPEG as well as MPEG4 for various network applications. The MPEG4 video stream which has GOP structure needs long recovery duration against network transmission errors. Therefore, the MPEG4 video client should wait until receiving the next I picture. Using the SOLO6010, the low bit rate MPEG4 video stream can be used for the storage and the motion JPEG can be transmitted for the network. Moreover the SOLO6010 supports the virtual video channel which encodes two different MPEG4 video bit stream for each video channel under guaranteed MPEG4 encoding and decoding specification. Therefore, it is possible to use one MPEG4 video stream for storage and the other MPEG4 video stream for network in DVR applications. In this case, recording MPEG4 video stream and networking MPEG4 video stream can be controlled independently. The SOLO6010 also support I picture based prediction for the next solution. However, the virtual MPEG4 video encoding and motion JPEG features will be better solution with the SOLO6010.

4.4.2. MPEG4 Video Encoder Configuration

Fig. 4-32 shows GOP (group of picture) structure, which means the interval between I-pictures. Its configuration range is 1 to 255. Although the SOLO6010 supports only P-picture, the MPEG4 encoders supporting B-pictures need a great amount of memory space to store P-picture and all B-picture for each video channel. Therefore, MPEG4 encoders supporting B-Picture may require 3-4times external SDRAM space, and increase the cost of system. However, the SOLO6010 is implemented with full searching motion estimation algorithm to reduce bit rate without wasting memory space and reducing frame delay to encode MPEG4 video. Full search algorithm will result the optimal compression ratio because its results are same as software simulation results theoretically. Therefore, the SOLO6010 has merits for security solutions which need multi stream MPEG4 video encoder.

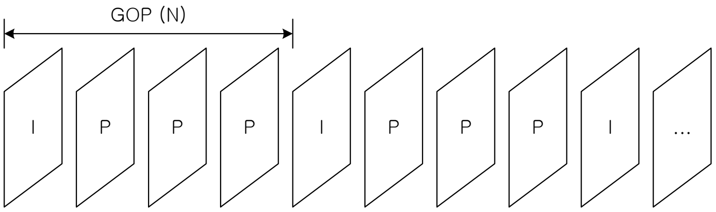
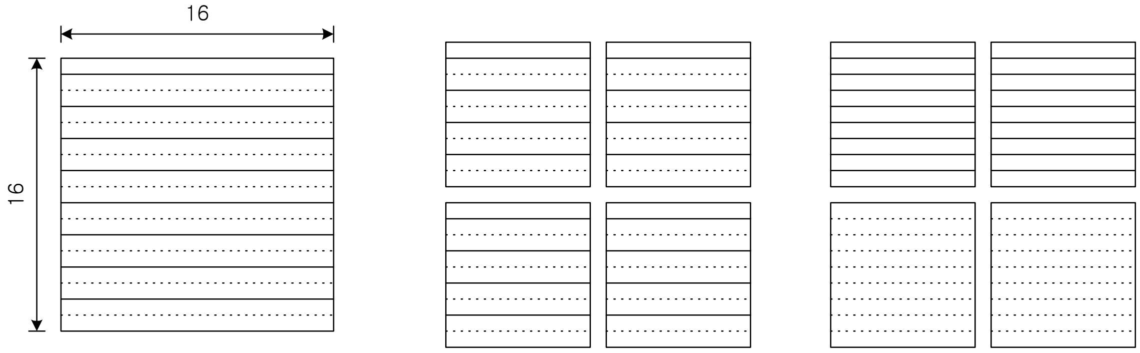


Fig. 4-32 Group of Picture

Picture Interval register controls the picture interval of security applications to save memory space for DVR or to fit bandwidth for network server or camera. Picture Interval should be set to 0 to encode real time frame mode pictures with 30frame per second, and 1 to encode field mode pictures with 30field per second. In frame mode, 2 or 3 will encode pictures with 10frame per second or 7.5frame per second, respectively. In field mode, 2 or 3 will encode pictures with 15field per second or 10field per second, respectively.

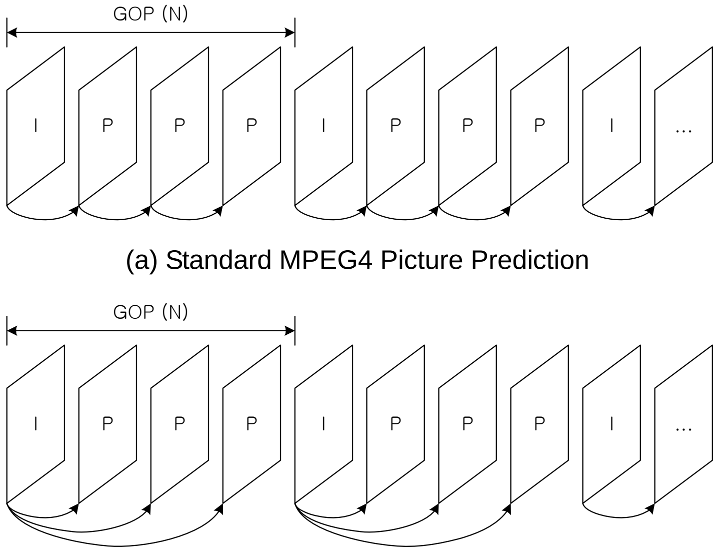
Adaptive frame/field DCT type decision will be processed automatically when Interlaced Mode is asserted in frame mode as shown in Fig. 4-33. Macroblocks with small motion or with big motion will be selected to frame DCT type or field DCT type to reduce bit rate. This feature should be deasserted for advanced simple profile clients because it is supported for main profile.



(a) macroblock (b) frame DCT type (c) field DCT type

Fig. 4-33 Frame/Field DCT

Intra Picture Based Prediction is not MPEG4 standard, but most commercial MPEG4 video CODECs support this feature to avoid picture drop on network transmission. However, its inheritance – non standard – cannot support commercial MPEG4 standard video clients and compression ratio is worse for big motion pictures. Although the SOLO6010 supports this feature, it is not recommend because the most important reason of MPEG4 video CODEC used for system is compression efficiency and compatibility for commercial MPEG4 standard video clients. Fig. 4-34 shows standard MPEG4 picture prediction and intra picture based picture prediction.



(a) Standard MPEG4 Picture Prediction

(b) Intra Picture Based Picture Prediction

Fig. 4-34 Picture Prediction Method

Extended MPEG4 video encoding and motion JPEG encoding features of SOLO6010 are good solution for network transmission for network sever or network camera or DVR. That features mean the SOLO6010 produces two MPEG4 video streams and motion JPEG. One MPEG4 video bit stream can be used for local network clients or the other MPEG4 video bit stream can be used for remote network clients on internet for network application. Networking function of recent DVR system is very important feature. Therefore, one MPEG4 video streams can be stored in storage and the other MPEG4 video bit stream can be transmitted for network clients.

Fig. 4-35 shows the SOLO6010 encoding scheme. For using extended MPEG4 video stream, the SOLO6010 needs more extended SDRAM storage and rooms of total quadruple encoding performance. However, the encoding performance of motion JPEG is independent MPEG4 video encoding performance because the SOLO6010 contains the independent motion JPEG hardware core.

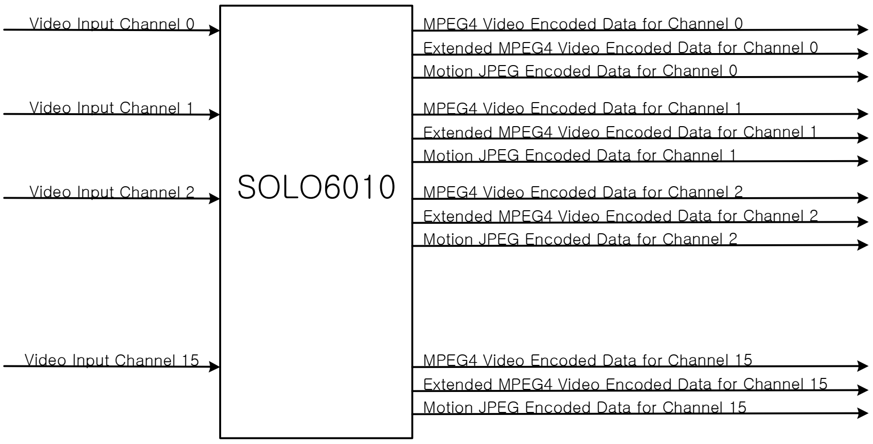

Video Input Channel 0
Video Input Channel 1
Video Input Channel 2
Video Input Channel 15
SOLO6010
MPEG4 Video Encoded Data for Channel 0
Extended MPEG4 Video Encoded Data for Channel 0
Motion JPEG Encoded Data for Channel 0
MPEG4 Video Encoded Data for Channel 1
Extended MPEG4 Video Encoded Data for Channel 1
Motion JPEG Encoded Data for Channel 1
MPEG4 Video Encoded Data for Channel 2
Extended MPEG4 Video Encoded Data for Channel 2
Motion JPEG Encoded Data for Channel 2
MPEG4 Video Encoded Data for Channel 15
Extended MPEG4 Video Encoded Data for Channel 15
Motion JPEG Encoded Data for Channel 15

Fig. 4-35 SOLO6010 Encoding Scheme

Fig. 4-36 shows the conceptual control flow of the video encoder. The video encoder capture part controls the interval (frame rate) for the encoding video channels. The Video encoder scaler part controls the resolution (frame size) for the encoding video channels. The MPEG4 video encoder part controls the quantization parameter (picture quality) and GOP size (I picture interval), and the motion JPEG encoder part controls the picture quality.

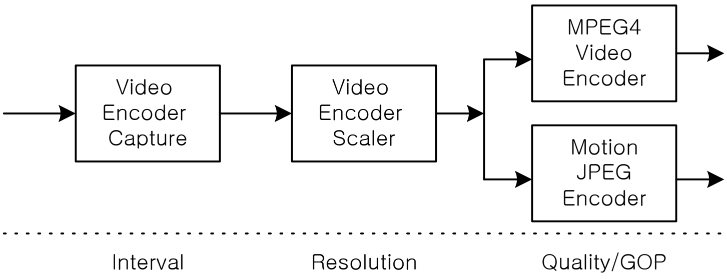


Fig. 4-36 Control Flow of the Video Encoder

As shown in Table 4-28, *Video Capture Configuration Registers* [0x0400~0x0420] are used to define the video dimensions to be encoded. *Maximum Page for Capture Register* [0x0400] defines the memory space to store the pictures to be encoded. *Capture Base Address* [0x0400] points the base address to store the pictures on the extended SDRAM of the SOLO6010. The memory space to store one pixel needs two bytes composed of luminance and chrominance data. Therefore, the memory space to store one line needs 1,440 bytes (720*2), but SOLO6010 needs 2,048 bytes to reduce design complexity. As a result, memory space to store one PAL frame needs 1,779,648 bytes. The base address pointer of the SOLO6010 is used higher 16bits. Therefore, 18(1,779,648/65,536) memory segments are needed to store one PAL frame by the unit of 64kbyte memory block. As similar method, 9, 15 and 8 memory segments are needed to store PAL field, NTSC frame and NTSC field, respectively. In addition, 5(1024*288/65,536) and

4(1024*240/65,536) memory segments are needed to store PAL and NTSC CIF field. When the SOLO6010 is operated with changing frame, field and CIF, the biggest segment should be configured. If it is not configured with enough memory space, each channel data can be mixed. This calculation method can be applied to configure all memory offset (or base address) to store pictures on external SDRAM.

The SOLO6010 contains 5 different video scale mode for frame and field mode, respectively, where 0 is set not to encode corresponding video for each channel. Each video scale mode can have 5 horizontal video sizes and 10 vertical video sizes because horizontal video size is only different according to frame or field mode. Each video size is preset on these compression dimension registers. During operating, Compression Video Scale register can be updated for application. The video sizes regarding Compression Video Scale register are referenced on compression dimension register. The video sizes are divided by 16 and its divided values are configured on this registers. However, only D1, HD1 or CIF picture size are available for play back display in DVR applications. Table 4-27 shows the configuration example of the picture size registers [0x0408~0x0418]. The detail application of these registers will be described in *Scale Mode Register* [0x0440, 0x0444, 0x0448, 0x044C, 0x0450, 0x0454, 0x0458, 0x045C, 0x0460, 0x0464, 0x0468, 0x046C, 0x0470, 0x0474, 0x0478, 0x047C].

Table 4-27 Configuration Example of the Picture Size Registers

Scale Mode	Picture Size Register	NTSC	PAL
0	No Encoding	-	-
1	Vertical Picture Size for Frame1	480/16	576/16
	Vertical Picture Size for Field1	240/16	288/16
	Horizontal Picture Size1	704/16	704/16
2	Vertical Picture Size for Frame2	480/16	576/16
	Vertical Picture Size for Field2	240/16	288/16
	Horizontal Picture Size2	352/16	352/16
3	Vertical Picture Size for Frame3	240/16	288/16
	Vertical Picture Size for Field3	112/16	144/16
	Horizontal Picture Size3	352/16	352/16
4	Vertical Picture Size for Frame4	160/16	192/16
	Vertical Picture Size for Field4	80/16	96/16
	Horizontal Picture Size4	224/16	224/16
5	Vertical Picture Size for Frame5	240/16	288/16
	Vertical Picture Size for Field5	112/16	144/16
	Horizontal Picture Size5	176/16	176/16

Progressive Vertical Picture Size for 1/3 Horizontal Scale Register [0x041C], *Progressive Horizontal Picture Size for 1/3 Vertical Scale Register* [0x041C], *Progressive Vertical Picture Size Register* [0x041C] and *Progressive Horizontal Picture Size Register* [0x041C] are reserved for the future applications.

Table 4-28 Video Capture Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0400	CAP_BASE	RW	[31:16]	*Maximum Page for Capture.*	0
			[15:0]	*Capture Base Address.*	0
0x0404	CAP_BTW	RW	[13:8]	*Bandwidth Value for Progressive Video.*	0
			[7:0]	*Maximum Bandwidth.*	0
0x0408	CAP_DIM_SCALE1	RW	[23:16]	*Vertical Picture Size for Frame1.*	
			[15:8]	*Vertical Picture Size for Field1.*	
			[7:0]	*Horizontal Picture Size1.*	
0x040C	CAP_DIM_SCALE2	RW	[23:16]	*Vertical Picture Size for Frame2.*	
			[15:8]	*Vertical Picture Size for Field2.*	
			[7:0]	*Horizontal Picture Size2.*	
0x0410	CAP_DIM_SCALE3	RW	[23:16]	*Vertical Picture Size for Frame3.*	
			[15:8]	*Vertical Picture Size for Field3.*	
			[7:0]	*Horizontal Picture Size3.*	
0x0414	CAP_DIM_SCALE4	RW	[23:16]	*Vertical Picture Size for Frame4.*	
			[15:8]	*Vertical Picture Size for Field4.*	
			[7:0]	*Horizontal Picture Size4.*	
0x0418	CAP_DIM_SCALE5	RW	[23:16]	*Vertical Picture Size for Frame5.*	
			[15:8]	*Vertical Picture Size for Field5.*	
			[7:0]	*Horizontal Picture Size5.*	
0x041C	CAP_DIM_PROG	RW	[31:24]	*Vertical Picture Size for 1/3 Horizontal Scale.*	
			[23:16]	*Horizontal Picture Size for 1/3 Vertical Scale.*	
			[15:8]	*Vertical Picture Size.*	
			[7:0]	*Horizontal Picture Size.*	
0x0420	CAP_STATUS	RO	[31:28]	*Encoding Req Queue Write Address.*	-
			[27:24]	*Encoding Req Queue Read Address.*	-
			[21:16]	*Current Bandwidth Value.*	-
			[15:0]	*Vertical Page Offset.*	-

As shown in Table 4-30, *MPEG4 Video Capture Control Registers* [0x0440~0x053C] are used to control frame size or frame interval to be encoded. *Frame Mode Register* [0x0440, 0x0444, 0x0448, 0x044C, 0x0450, 0x0454, 0x0458, 0x045C, 0x0460, 0x0464, 0x0468, 0x046C, 0x0470, 0x0474, 0x0478, 0x047C] is set to 0 for field or 1 for frame encoding for each video input channel.*Scale Mode Register* [0x0440, 0x0444, 0x0448, 0x044C, 0x0450, 0x0454, 0x0458, 0x045C, 0x0460, 0x0464, 0x0468, 0x046C, 0x0470, 0x0474, 0x0478, 0x047C] is set to 0 for no encoding and non zero for MPEG4 video encoding for each video input channel. When Scale Mode Register [0x0440, 0x0444, 0x0448, 0x044C, 0x0450, 0x0454, 0x0458, 0x045C, 0x0460, 0x0464, 0x0468, 0x046C, 0x0470, 0x0474, 0x0478, 0x047C] is set to 1, 2, 3, 4 or 5 for each video input channel, the picture size is referred as defined on scale mode1, scale mode2, scale mode3, scale mode4 or scale mode5 in Table 4-27, respectively.

Extended Encoding On Register [0x0480, 0x0484, 0x0488, 0x048C, 0x0490, 0x0494, 0x0498, 0x049C, 0x04A0, 0x04A4, 0x04A8, 0x04AC, 0x04B0, 0x04B4, 0x04B8, 0x04BC] is set to 1 to activate extended MPEG4 video encoding.

For the frame rate control, the video encoder capture control has two modes. Skip mode captures skip sampling the encoding picture except for N-1'th frames or fields. Drop mode captures drop every N-1'th frames or fields. *Interval Mode Register* [0x04C0, 0x04C4, 0x04C8, 0x04CC, 0x04D0, 0x04D4, 0x04D8, 0x04DC, 0x04E0, 0x04E4, 0x04E8, 0x04EC, 0x04F0, 0x04F4, 0x04F8, 0x04FC] and *Extended Interval Mode Register* [0x0500, 0x0504, 0x0508, 0x005C, 0x0510, 0x0514, 0x0518, 0x051C, 0x0520, 0x0524, 0x0528, 0x052C, 0x0530, 0x53F4, 0x0538, 0x053C] are set to 0 for skip mode and 1 for drop mode.

Interval Value Register [0x04C0, 0x04C4, 0x04C8, 0x04CC, 0x04D0, 0x04D4, 0x04D8, 0x04DC, 0x04E0, 0x04E4, 0x04E8, 0x04EC, 0x04F0, 0x04F4, 0x04F8, 0x04FC] and *Extended Interval Value Register* [0x0500, 0x0504, 0x0508, 0x005C, 0x0510, 0x0514, 0x0518, 0x051C, 0x0520, 0x0524, 0x0528, 0x052C, 0x0530, 0x53F4, 0x0538, 0x053C] are configure the values to control the frame rate as following equations. Table 4-29 shows the frame rate and frame period.

Skip mode frame rate : $T = M / (N + P)$

Drop mode frame rate : $T = M * (N/(N + 1))$

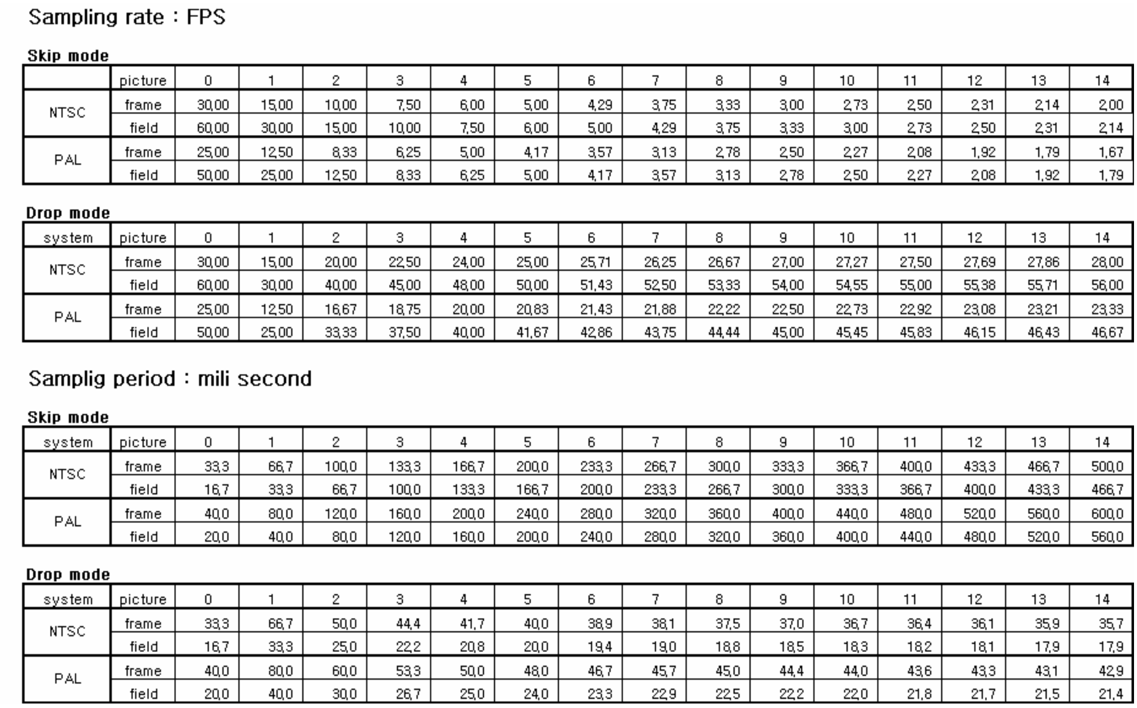
Where, T = frame rate

N = Interval

M = 30 for NTSC, 25 for PAL

P = 1 for frame, 0 for field

Table 4-29 Frame rate and Frame Period



Sampling rate : FPS

Skip mode

	picture	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
NTSC	frame	30,00	15,00	10,00	7,50	6,00	5,00	4,29	3,75	3,33	3,00	2,73	2,50	2,31	2,14	2,00
	field	60,00	30,00	15,00	10,00	7,50	6,00	5,00	4,29	3,75	3,33	3,00	2,73	2,50	2,31	2,14
PAL	frame	25,00	12,50	8,33	6,25	5,00	4,17	3,57	3,13	2,78	2,50	2,27	2,08	1,92	1,79	1,67
	field	50,00	25,00	12,50	8,33	6,25	5,00	4,17	3,57	3,13	2,78	2,50	2,27	2,08	1,92	1,79

Drop mode

system	picture	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
NTSC	frame	30,00	15,00	20,00	22,50	24,00	25,00	25,71	26,25	26,67	27,00	27,27	27,50	27,69	27,86	28,00
	field	60,00	30,00	40,00	45,00	48,00	50,00	51,43	52,50	53,33	54,00	54,55	55,00	55,38	55,71	56,00
PAL	frame	25,00	12,50	16,67	18,75	20,00	20,83	21,43	21,88	22,22	22,50	22,73	22,92	23,08	23,21	23,33
	field	50,00	25,00	33,33	37,50	40,00	41,67	42,86	43,75	44,44	45,00	45,45	45,83	46,15	46,43	46,67

Samplig period : mili second

Skip mode

system	picture	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
NTSC	frame	33,3	66,7	100,0	133,3	166,7	200,0	233,3	266,7	300,0	333,3	366,7	400,0	433,3	466,7	500,0
	field	16,7	33,3	66,7	100,0	133,3	166,7	200,0	233,3	266,7	300,0	333,3	366,7	400,0	433,3	466,7
PAL	frame	40,0	80,0	120,0	160,0	200,0	240,0	280,0	320,0	360,0	400,0	440,0	480,0	520,0	560,0	600,0
	field	20,0	40,0	80,0	120,0	160,0	200,0	240,0	280,0	320,0	360,0	400,0	440,0	480,0	520,0	560,0

Drop mode

system	picture	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
NTSC	frame	33,3	66,7	50,0	44,4	41,7	40,0	38,9	38,1	37,5	37,0	36,7	36,4	36,1	35,9	35,7
	field	16,7	33,3	25,0	22,2	20,8	20,0	19,4	19,0	18,8	18,5	18,3	18,2	18,1	17,9	17,9
PAL	frame	40,0	80,0	60,0	53,3	50,0	48,0	46,7	45,7	45,0	44,4	44,0	43,6	43,3	43,1	42,9
	field	20,0	40,0	30,0	26,7	25,0	24,0	23,3	22,9	22,5	22,2	22,0	21,8	21,7	21,5	21,4

Table 4-30 MPEG4 Video Capture Control Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0440 0x0444 0x0448 0x044C 0x0450 0x0454 0x0458 0x045C 0x0460 0x0464 0x0468 0x046C 0x0470 0x0474 0x0478 0x047C	CAP_SCALE0 CAP_SCALE1 CAP_SCALE2 CAP_SCALE3 CAP_SCALE4 CAP_SCALE5 CAP_SCALE6 CAP_SCALE7 CAP_SCALE8 CAP_SCALE9 CAP_SCALE10 CAP_SCALE11 CAP_SCALE12 CAP_SCALE13 CAP_SCALE14 CAP_SCALE15	RW	[3]	*Frame Mode*.	0
			[2:0]	*Scale Mode*. [000] : No Encoding [001] : H=1/1, V=1/1 (Frame/Field Mode) [010] : H=1/2, V=1/1 (Only Field Mode) [011] : H=1/2, V=1/2 (Only Encoding) [100] : H=1/3, V=1/3 (Only Encoding) [101] : H=1/4, V=1/2 (Only Field Mode)	0
0x0480 0x0484 0x0488 0x048C 0x0490 0x0494 0x0498 0x049C 0x04A0 0x04A4 0x04A8 0x04AC 0x04B0 0x04B4 0x04B8 0x04BC	CAP_SCALE_E0 CAP_SCALE_E1 CAP_SCALE_E2 CAP_SCALE_E3 CAP_SCALE_E4 CAP_SCALE_E5 CAP_SCALE_E6 CAP_SCALE_E7 CAP_SCALE_E8 CAP_SCALE_E9 CAP_SCALE_E10 CAP_SCALE_E11 CAP_SCALE_E12 CAP_SCALE_E13 CAP_SCALE_E14 CAP_SCALE_E15	R/W	[0]	*Extended Encoding On.* Extended MPEG4 video encoding channel will be activated by asserting the register.	0
0x04C0 0x04C4 0x04C8 0x04CC 0x04D0 0x04D4 0x04D8 0x04DC 0x04E0 0x04E4 0x04E8 0x04EC 0x04F0 0x04F4 0x04F8 0x04FC	CAP_INTERVAL0 CAP_INTERVAL1 CAP_INTERVAL2 CAP_INTERVAL3 CAP_INTERVAL4 CAP_INTERVAL5 CAP_INTERVAL6 CAP_INTERVAL7 CAP_INTERVAL8 CAP_INTERVAL9 CAP_INTERVAL10 CAP_INTERVAL11 CAP_INTERVAL12 CAP_INTERVAL13 CAP_INTERVAL14 CAP_INTERVAL15	R/W	[9]	*Interval Mode*. [0] : Skip Mode [1] : Drop Mode	0
			[8:0]	*Interval Value*.	0
0x0500 0x0504 0x0508 0x050C 0x0510 0x0514 0x0518 0x051C 0x0520 0x0524 0x0528 0x052C 0x0530 0x0534 0x0538 0x053C	CAP_INTERVALE0 CAP_INTERVALE1 CAP_INTERVALE2 CAP_INTERVALE3 CAP_INTERVALE4 CAP_INTERVALE5 CAP_INTERVALE6 CAP_INTERVALE7 CAP_INTERVALE8 CAP_INTERVALE9 CAP_INTERVALE10 CAP_INTERVALE11 CAP_INTERVALE12 CAP_INTERVALE13 CAP_INTERVALE14 CAP_INTERVALE15	R/W	[9]	*Extended Interval Mode*. [0] : Skip Mode [1] : Drop Mode	0
			[8:0]	*Extended Interval Value*.	0

Fig. 4-37 shows the code buffer of MPEG4 video encoder and motion JPEG encoder. 2 page mode generates interrupt whenever the video encoder is done and uses only 2 code buffer. However, multi page mode generates interrupt whenever the video encoder is done after processing predefine frames and uses unlimited external SDRAM memory space up to the remained memory size.

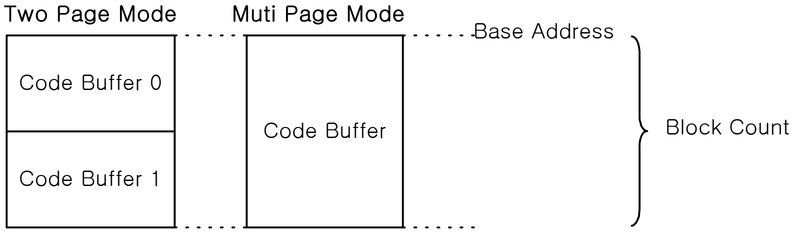
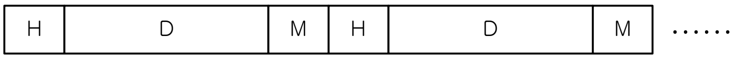


Fig. 4-37 Code Buffer of MPEG4 Video and Motion JPEG Encoder

Fig. 4-38 shows the MPEG4 stream format. Header includes index structure based stream information and stuffing is appended for internal cash buffer allocation.



Header : 64 bytes
Data : N bytes
Stuffing (only for multi page mode) : 32, 64, 128, 256 bytes will be appended at the end of Data

Fig. 4-38 MPEG4 Stream Format

MPEG4 Video Encoder Configuration Registers [0x0610~0x0630] is used to control MPEG4 video encoder, watermark and code encryption. *Code Control Mode Register* [0x0610] is set to 0 for multi frame page mode and 1 for two page mode. *Interrupt Mode Register* [0x0610] is set to 0, 1, 2 and 3 for interrupt asserted every frame, 2 frames, 3 frames and 4 frames, respectively. *MPEG4 Block Size Register* [0x0610] is used to define the code block size. *MPEG4 Block Base Address Register* [0x0610] points the base address on the external SDRAM. *Byte Aligns Register* [0x0614] is set to 0, 1, 2, and 3 to align 8 bytes, 16 bytes, 32 bytes and 64 bytes. *Motion Flag Data Insertion Register* [0x0614] is set to 0 for no motion flag, 1 for inserting after 256 bytes and 2 for user setting. In case of user setting, *Motion Flag Data Base Address Register* [0x0614] points the address on external SDRAM to be stored the motion flags.

The SOLO6010 supports MPEG4 video encoded code encryption and watermark. *Watermark Generic Polynomial Register* [0x061C], *Watermark Initial Key Value Register* [0x0620], and *Watermark Strength Register* [0x0624] need to be set for activate watermark function. *Watermark Generic Polynomial Register* [0x061C] and *Watermark Initial Key Value Register* [0x0620] can be used as system secure value. *Watermark Strength Register* [0x0624] can be used to embedding strength. The SOLO6010 watermark feature enables system to decide if MPEG4 video encoded picture is manipulated. The pass rate of the watermark detection algorithm will be higher but the picture quality will be lower if the value of Watermark Strength Register [0x0624] increased. Otherwise, the watermark pass rate will be lower and the picture quality will be higher.

Encryption of MPEG4 video coded data is also important for secure system when encoded video data is transmitted on internet. Fig. 4-39 shows concept of MPEG4 video code encryption. As shown in Fig. 4-39, 0x23(0010 0011), 0xA0(1010 0000), 0x17(0001 0111), and 0x0E(0000 1110) are assumed as original input sequence to transmit. 0x11(0001 0001), 0x012(0001 0010), 0x13(0001 0011), and 0xF0(1111 0000) are assumed as key sequence. Original input sequence is operated with bitwise exclusive OR by encryption block. 0x32(0011 0010), 0xB2(1011 0010), 0x04(0000 0100), and 0xFE(1111 1110) are encrypted sequence. The encrypted sequence is transmitted on internet and is operated with bitwise exclusive OR by the same key sequence which is used in the encryption. The decrypted output sequence is resulted to same sequence as original sequence. If key sequence is unknown in decryption, it is impossible to decrypt the encrypted sequence. Therefore, it is very important to generate the key generation algorithm.

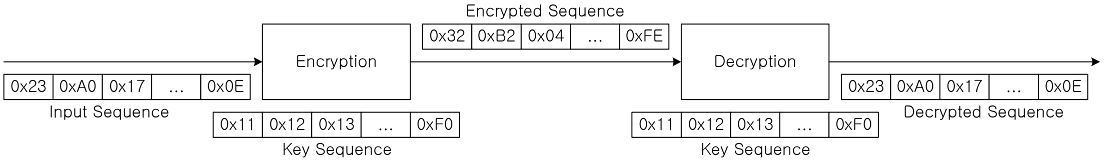


Fig. 4-39 MPEG4 Video Code Encryption

The SOLO6010 contains two logics for generating shift register sequences, which is used for embedding watermark and encrypting MPEG4 video coded data, respectively. The shift register sequences are very simple to implement hardware logic and satisfy random property. Random property of the key sequence is very important for secure system because the key sequence should not be analyzed. If the key sequence is analyzed easily, the encrypted sequence would be decrypted easily by an unauthorized listener.

The sequence which is generated shift register is pseudo random sequence. A truly random signal cannot be predicted. Its future variations can only be described in a statistical sense. However, a pseudorandom signal is not random at all. It is a deterministic periodic signal that is known to both the transmitter and receiver. Even though the signal is deterministic, it appears to have the statistical properties of sampled white noise. It appears, to an unauthorized listener, to be a truly random signal.

Fig. 4-40 shows linear feedback shift register example. It is made up of a four-stage register for storage and shifting, a modulo-2 adder, and a feedback path from the adder to the input of the register.

The shift register operation is controlled by a sequence register is shifted one stage to the right.

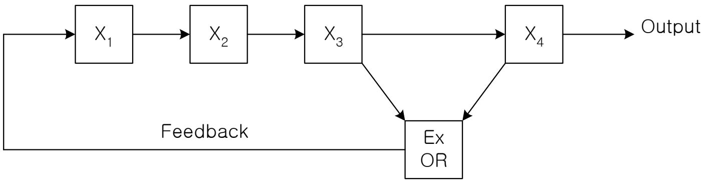


Fig. 4-40 Linear feedback shift register example

Assume that {X_1, X_2, X_3, X_4} are filled with {1, 0, 0, 0}, that is, the initial state of the register is {1, 0, 0, 0}. We can see that the succession of register states will be {1, 0, 0, 0}, {0, 1, 0, 0}, {0, 0, 1, 0}, {1, 0, 0, 1}, {1, 1, 0, 0}, {0, 1, 1, 0}, {1, 0, 1, 1}, {0, 1, 0, 1}, {1, 0, 1, 0}, {1, 1, 0, 1}, {1, 1, 1, 0}, {1, 1, 1, 1}, {0, 1, 1, 1}, {0, 0, 1, 1}, {0, 0, 0, 1}, and {1, 0, 0, 0}.

Since the last state, {1, 0, 0, 0}, corresponds to the initial state, we see that the register repeats the foregoing sequence after 15 clock pulses. The shift register generator produces sequences that depend on the number of stages, the feedback tap connections, and initial conditions. The maximal length sequences have the property that for an n-stage linear feedback shift register the sequence repetition period in clock pulses p is $p = 2^n - 1$.

The shift register of the SOLO6010 is composed of 32 stage shift register, that is, the maximal length sequences is 4,294,967,295 (2^{32}-1). The generic polynomial and initial key value are set by user.

Watermarking is an important issue for security system to detect malice manipulation for legal evidence. Fig. 4-41 shows watermark embedding scheme which is used in the SOLO6010. The watermark embedding process using pseudo noise is shown in (1).

$$W(x,y) = O(x,y) + \alpha \cdot N(x,y) \qquad (1)$$

Where, $O(x,y)$ is the original signal, $N(x,y)$ is the PN(pseudo random noise) sequence, α is the embedding strength.

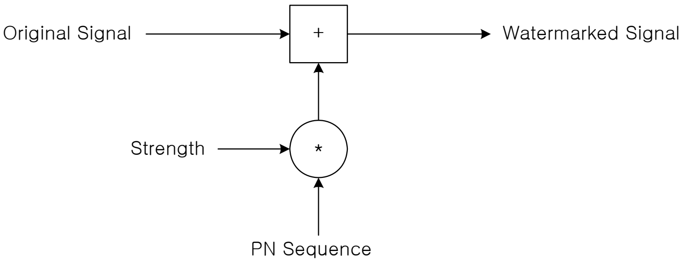


Fig. 4-41 Watermark Embedding Scheme

Fig. 4-42 shows the watermark extraction process that can extract the embedded watermark using the PN sequence. The SOLO6010 includes only watermark embedding scheme but users need to understand watermark extraction scheme for application software.

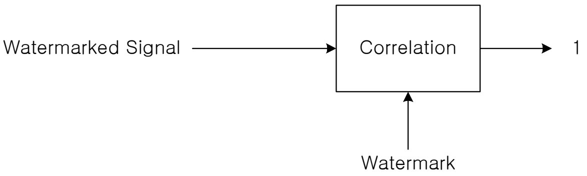


Fig. 4-42 Watermark Extraction Scheme

In watermark extraction scheme, the correlation process acquires synchronization using correlation of the watermarked data and the PN sequence. As in (2), the extraction algorithm uses cross correlation of the watermarked signal $W(x,y)$ with the same PN signal $N(x,y)$ that has been used in embedding process.

$$\begin{aligned} R_{wn} &= E\{[W(x,y)-\mu_w][N(x,y)-\mu_n]\} \\ &= E\{W(x,y)\cdot N(x,y)\}-\mu_w\mu_n \end{aligned} \tag{2}$$

where, $\mu_w = E\{W(x,y)\}$, $\mu_n = E\{N(x,y)\}$. From (1), we can define the relation between μ_w and μ_n as follows.

$$\mu_w = \mu_o + \alpha \cdot I \cdot \mu_n \tag{3}$$

By substituting (1) and (3) into (2), the cross correlation can be rewritten as follows.

$$\begin{aligned} R_{wn} &= E\{O(x,y)\cdot N(x,y)\}-\mu_w\mu_n+\alpha\cdot I\cdot E\{N^2(x,y)\} \\ &= E\{[O(x,y)-\mu_o][N(x,y)-\mu_o]\} \\ &\quad +\alpha\cdot I\cdot E\{N^2(x,y)\}+\alpha\cdot I\cdot\mu_n^2 \end{aligned} \tag{4}$$

Let assume that the original signal $O(x,y)$ is orthogonal to the PN sequence $N(x,y)$. Then, the cross correlation of the original signal with the PN sequence becomes zero as follows.

$$E\{[O(x,y)-\mu_o][N(x,y)-\mu_n]\}=0 \tag{5}$$

As the PN sequence $N(x,y)$ is the same as that used in embedding process, the expectation of square of the PN sequence is always one.

$$E\{N^2(x,y)\}=1 \tag{6}$$

The SOLO6010 internal bus width is 64bit and the codeword can be aligned by *Code Byte Order Register* [0x0630].

The SOLO6010 produces VOP header of MPEG4 video stream and the SOLO6010 driver software appends VOL header of MPEG4 video stream. *VOP Time Increment Register* [0x0630] and *VOP Time Increment Bit Width Register* [0x0630] are used for time stamp for VOP header automatically. Time information is important to decode MPEG4 video stream commercial client or viewer for compatibility.

VOP Round Register [0x0630] should be set to 0, *VOP F. Code Register* [0x0630] should be set to 1, and *VOP Time Increment Bit Width Register* [0x0630] should be set to 15, respectively for normal applications. The MPEG4 video encoder of the SOLO6010 can process the quadruple D1 speed MPEG4 video encoding performance when the internal system clock is set to 108MHz and *DCT Block Interval Register* [0x0630] is set to 9. In network camera or network video server applications, the internal system clock can be set to 54MHz for reducing the SOLO6010 surface temperature. In this case, the MPEG4 video encoder will process the twice D1 speed MPEG4 video encoding performance.

Table 4-31 MPEG4 Video Encoder Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0610	VE_CFG0	RW	[31]	*Code Control Mode.* [0] : Multi Page Mode [1] : Two Page Mode	0
			[29:24]	*Interrupt Mode.* [0] : Interrupt asserted every frame [1] : Interrupt asserted every 2 frames [2] : Interrupt asserted every 3 frames [3] : Interrupt asserted every 4 frames	0
			[23:16]	*MPEG4 Block Size.* (64kB)	0
			[15:0]	*MPEG4 Block Base Address.* (64kB)	0
0x0614	VE_CFG1	RW	[27:24]	*Byte aligns.* [0] : 8 [1] : 16 [2] : 32 [3] : 64	0
			[18]	*Index Insertion.*	0
			[17:16]	*Motion Flag Data Insertion.* [0] : Off [1] : After index 256 bytes [2] : User Setting.	0
			[15:0]	*Motion Flag Data Base Address.*	0
0x061C	VE_WMARK_POLY	RW	[31:0]	*Watermark Generic Polynomial.*	0
0x0620	VE_WMARK_KEY	RW	[31:0]	*Watermark Initial Key Value.*	0
0x0624	VE_WMARK_CTRL	RW	[3:0]	*Watermark Strength.*	0
0x0628	VE_ENCR_POLY	RW	[31:0]	*Encryption Generic Polynomial.*	0
0x062C	VE_ENCR_KEY	RW	[31:0]	*Encryption Initial Key Value.*	0
0x0630	VE_ATTR	RW	[31]	*Byte Order.*	0
			[30]	*VOP Round.*	0
			[29:27]	*VOP F. Code.*	0
			[26:25]	*VOP Time Increment.* [00] : 60Hz (500) for NTSC field mode [01] : 30Hz (1000) for NTSC frame mode [10] : 50Hz (600) for PAL field mode [11] : 25Hz (1200) for PAL frame mode	0
			[24:21]	*VOP Time Increment Bit Width.*	0
			[20:16]	*DCT Block Interval.*	0

As shown in Table 4-32, *MPEG4 Video Status Registers* [0x0640~0x066C] indicate information such as Video Channel, VOP Type, MPEG4 Encoded Code Size, etc when the picture is encoded. *Progressive Register* [0x0640] indicates if the MPEG4 encoded picture is progressive mode for the current encoded picture. *Interlace Register* [0x0640] indicates if the MPEG4 encoded picture is interface or not for the current encoded picture. *Source Field Index Register* [0x0640] indicates if the MPEG4 encoded picture is odd or even field for the current encoded picture. *Video Channel Register* [0x0640] indicates the video input channel identification of the encoded MPEG4 video stream for the current encoded picture. *VOP Type Register* [0x0640] indicates 0 for I Picture or 1 for P Picture for the current encoded picture.

Video Motion Flag Register [0x0640] indicates if the video motion is detected for the current encoded picture. *SAD Motion Flag Register* [0x0640] indicates if the SAD motion is detected for the current encoded picture. *MPEG4 Encoded Code Size Register* [0x0640] is the code size for the current encoded picture. *Scale Register* [0x0644] indicates the scale for the current encoded picture.

Horizontal Picture Size Register [0x0644] indicates the picture horizontal size for the current encoded picture. *Vertical Picture Size Register* [0x0644] indicates the picture vertical size for the current encoded picture. *MPEG4 Code Data Offset Register* [0x0648] indicates the MPEG4 code data offset on the external SDRAM for the current encoded picture. *JPEG Code Data Offset Register* [0x064C] indicates the motion JPEG code data offset on the external SDRAM for the current encoded picture. *Interval Register* [0x0650] indicates the interval for the current encoded picture. *JPEG Code Size Register* [0x0650] indicates the JPEG code size for the current encoded picture. *Time Second Register* [0x0654] and *Time Micro Second Register* [0x0658] indicate the time information for the current encoded picture. *Watermark Status Register* [0x065C] indicates watermark synchronization status information for the current encoded picture. *Encryption Status Register* [0x0660] indicates encryption synchronization status information for the current encoded picture. *Channel Register* [0x0664] indicates the channel number for the current encoded picture. *MPEG4 Code Size Register* [0x0668] indicates the MPEG4 code size for the current encoded picture.

Table 4-32 MPEG4 Video Encoder Status Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0640	VE_STATUS0	RO	[31]	*Progressive.*	0
			[30]	*Interlace.*	0
			[29]	*Source Field Index.*	0
			[28:24]	*Video Channel.*	0
			[23:22]	*VOP Type.*	0
			[21]	*Video Motion Flag.*	0
			[20]	*SAD Motion Flag.*	0
			[19:0]	*MPEG4 Encoded Code Size.*	0
0x0644	VE_STATUS1	RO	[31:28]	*Scale.*	0
			[19:16]	*Last Queue Position.*	0
			[15:8]	*Horizontal Picture Size.*	0
			[7:0]	*Vertical Picture Size.*	0
0x0648	VE_STATUS2	RO	[31:0]	*MPEG4 Code Data Offset.*	0
0x064C	VE_STATUS3	RO	[31:0]	*JPEG Code Data Offset.*	0
0x0650	VE_STATUS4	RO	[29:20]	*Interval.*	0
			[19:0]	*JPEG Code Size.*	0
0x0654	VE_STATUS5	RO	[31:0]	*Time.* (sec)	0
0x0658	VE_STATUS6	RO	[31:0]	*Time.* (usec)	0
0x065C	VE_STATUS7	RO	[31:0]	*Watermark Status.*	0
0x0660	VE_STATUS8	RO	[31:0]	*Encryption Status.*	0
0x0664	VE_STATUS9	RO	[4:0]	*Channel.*	0
0x0668	VE_STATUS10	RO	[19:0]	*MPEG4 Code Size.*	0
0x066C	VE_STATUS11	RO	[7:0]	*Last Queue Position*	0

As shown in Table 4-33, *Video Encoder Configuration Registers* [0x0700~0x0A04] are used to control MPEG4 video encoder parameters. Intra picture based prediction will be activated for each video input channel by asserting *Intra Picture Based Prediction register* [0x0700, 0x0704, 0x0708, 0x070C, 0x0710, 0x0714, 0x0718, 0x071C, 0x0720, 0x0724, 0x0728, 0x072C, 0x0730, 0x0734, 0x0738, 0x073C].

Adaptive frame/field DCT type decision of each video input channel will be applied by asserting *Interlace register* [0x0700, 0x0704, 0x0708, 0x070C, 0x0710, 0x0714, 0x0718, 0x071C, 0x0720, 0x0724, 0x0728, 0x072C, 0x0730, 0x0734, 0x0738, 0x073C]. The SOLO6010 MPEG4 video encoder support MPEG4 video advanced simple profile. However, adaptive frame/field DCT type is applied in MPEG4 video main profile. Therefore, adaptive frame/field DCT type should be deasserted in advanced simple profile applications.

The SOLO6010 supports motion detection using SAD (sum of absolute difference) in ME (motion estimation) as well as video multiplexer motion detection. The motion sensitivity can be controlled independently for each video input channel. *Motion Threshold register* [0x0740, 0x0744, 0x0748, 0x074C, 0x0750, 0x0754, 0x0758, 0x075C, 0x0760, 0x0764, 0x0768, 0x076C, 0x0770, 0x0774, 0x0778, 0x077C] controls the motion sensitivity for each video input channel.

Quantization Parameter register [0x0780, 0x0784, 0x0788, 0x078C, 0x0790, 0x0794, 0x0798, 0x079C, 0x07A0, 0x07A4, 0x07A8, 0x07AC, 0x07B0, 0x07B4, 0x07B8, 0x07BC] controls the video picture quality or bit rate of MPEG4 video encoder for each video input channel. *Extended Quantization Parameter register* [0x0780, 0x0784, 0x0788, 0x078C, 0x0790, 0x0794, 0x0798, 0x079C, 0x07A0, 0x07A4, 0x07A8, 0x07AC, 0x07B0, 0x07B4, 0x07B8, 0x07BC] is applied for virtual MPEG4 video stream for each video input channel.

GOP Size register [0x0800, 0x0804, 0x0808, 0x080C, 0x0810, 0x0814, 0x0818, 0x081C, 0x0820, 0x0824, 0x0828, 0x082C, 0x0830, 0x0834, 0x0838, 0x083C] controls the interval of I pictures for each video input channel. *Extended GOP Size register* [0x0840, 0x0844, 0x0848, 0x084C, 0x0850, 0x0854, 0x0858, 0x085C, 0x0860, 0x0864, 0x0868, 0x086C, 0x0870, 0x0874, 0x0878, 0x087C] is applied for virtual MPEG4 video stream for each video input channel.

Reference Base Address register [0x0880, 0x0884, 0x0888, 0x088C, 0x0890, 0x0894, 0x0898, 0x089C, 0x08A0, 0x08A4, 0x08A8, 0x08AC, 0x08B0, 0x08B4, 0x08B8, 0x08BC] points the base address to be stored previous picture for encoding P picture for each video input channel. *Extended Reference Base Address register* [0x0880, 0x0884, 0x0888, 0x088C, 0x0890, 0x0894, 0x0898, 0x089C, 0x08A0, 0x08A4, 0x08A8, 0x08AC, 0x08B0, 0x08B4, 0x08B8, 0x08BC] is applied for virtual MPEG4 video stream for each video input channel.

Table 4-33 Video Encoder Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0700	VE_CH_INTL0	RW	[1]	*Intra Picture Based Prediction.*	0

0x0704	VE_CH_INTL1		[0]	*Interlace.*	0
0x0708	VE_CH_INTL2				
0x070C	VE_CH_INTL3				
0x0710	VE_CH_INTL4				
0x0714	VE_CH_INTL5				
0x0718	VE_CH_INTL6				
0x071C	VE_CH_INTL7				
0x0720	VE_CH_INTL8				
0x0724	VE_CH_INTL9				
0x0728	VE_CH_INTL10				
0x072C	VE_CH_INTL11				
0x0730	VE_CH_INTL12				
0x0734	VE_CH_INTL13				
0x0738	VE_CH_INTL14				
0x073C	VE_CH_INTL15				
0x0740	VE_CH_MOT0	RW	[15:0]	*Motion Threshold.*	0
0x0744	VE_CH_MOT1				
0x0748	VE_CH_MOT2				
0x074C	VE_CH_MOT3				
0x0750	VE_CH_MOT4				
0x0754	VE_CH_MOT5				
0x0758	VE_CH_MOT6				
0x075C	VE_CH_MOT7				
0x0760	VE_CH_MOT8				
0x0764	VE_CH_MOT9				
0x0768	VE_CH_MOT10				
0x076C	VE_CH_MOT11				
0x0770	VE_CH_MOT12				
0x0774	VE_CH_MOT13				
0x0778	VE_CH_MOT14				
0x077C	VE_CH_MOT15				
0x0780	VE_CH_QP0	RW	[4:0]	*Quantization Parameter.*	0
0x0784	VE_CH_QP1				
0x0788	VE_CH_QP2				
0x078C	VE_CH_QP3				
0x0790	VE_CH_QP4				
0x0794	VE_CH_QP5				
0x0798	VE_CH_QP6				
0x079C	VE_CH_QP7				
0x07A0	VE_CH_QP8				
0x07A4	VE_CH_QP9				
0x07A8	VE_CH_QP10				
0x07AC	VE_CH_QP11				
0x07B0	VE_CH_QP12				
0x07B4	VE_CH_QP13				
0x07B8	VE_CH_QP14				
0x07BC	VE_CH_QP15				
0x07C0	VE_CH_QP_E0	RW	[4:0]	*Extended Quantization Parameter.*	0
0x07C4	VE_CH_QP_E1				
0x07C8	VE_CH_QP_E2				
0x07CC	VE_CH_QP_E3				
0x07D0	VE_CH_QP_E4				
0x07D4	VE_CH_QP_E5				
0x07D8	VE_CH_QP_E6				
0x07DC	VE_CH_QP_E7				
0x07E0	VE_CH_QP_E8				
0x07E4	VE_CH_QP_E9				
0x07E8	VE_CH_QP_E10				
0x07EC	VE_CH_QP_E11				
0x07F0	VE_CH_QP_E12				
0x07F4	VE_CH_QP_E13				
0x07F8	VE_CH_QP_E14				
0x07FC	VE_CH_QP_E15				

0x0800 0x0804 0x0808 0x080C 0x0810 0x0814 0x0818 0x081C 0x0820 0x0824 0x0828 0x082C 0x0830 0x0834 0x0838 0x083C	VE_CH_GOP0 VE_CH_GOP1 VE_CH_GOP2 VE_CH_GOP3 VE_CH_GOP4 VE_CH_GOP5 VE_CH_GOP6 VE_CH_GOP7 VE_CH_GOP8 VE_CH_GOP9 VE_CH_GOP10 VE_CH_GOP11 VE_CH_GOP12 VE_CH_GOP13 VE_CH_GOP14 VE_CH_GOP15	RW	[7:0]	*GOP Size.*	0
0x0840 0x0844 0x0848 0x084C 0x0850 0x0854 0x0858 0x085C 0x0860 0x0864 0x0868 0x086C 0x0870 0x0874 0x0878 0x087C	VE_CH_GOP_E0 VE_CH_GOP_E1 VE_CH_GOP_E2 VE_CH_GOP_E3 VE_CH_GOP_E4 VE_CH_GOP_E5 VE_CH_GOP_E6 VE_CH_GOP_E7 VE_CH_GOP_E8 VE_CH_GOP_E9 VE_CH_GOP_E10 VE_CH_GOP_E11 VE_CH_GOP_E12 VE_CH_GOP_E13 VE_CH_GOP_E14 VE_CH_GOP_E15	RW	[7:0]	*Extended GOP Size.*	0
0x0880 0x0884 0x0888 0x088C 0x0890 0x0894 0x0898 0x089C 0x08A0 0x08A4 0x08A8 0x08AC 0x08B0 0x08B4 0x08B8 0x08BC	VE_CH_REFB0 VE_CH_REFB1 VE_CH_REFB2 VE_CH_REFB3 VE_CH_REFB4 VE_CH_REFB5 VE_CH_REFB6 VE_CH_REFB7 VE_CH_REFB8 VE_CH_REFB9 VE_CH_REFB10 VE_CH_REFB11 VE_CH_REFB12 VE_CH_REFB13 VE_CH_REFB14 VE_CH_REFB15	RW	[15:0]	*Reference Base Address.*	0
0x08C0 0x08C4 0x08C8 0x08CC 0x08D0 0x08D4 0x08D8 0x08DC 0x08E0 0x08E4 0x08E8 0x08EC 0x08F0 0x08F4 0x08F8 0x08FC	VE_CH_REFB_E0 VE_CH_REFB_E1 VE_CH_REFB_E2 VE_CH_REFB_E3 VE_CH_REFB_E4 VE_CH_REFB_E5 VE_CH_REFB_E6 VE_CH_REFB_E7 VE_CH_REFB_E8 VE_CH_REFB_E9 VE_CH_REFB_E10 VE_CH_REFB_E11 VE_CH_REFB_E12 VE_CH_REFB_E13 VE_CH_REFB_E14 VE_CH_REFB_E15	RW	[15:0]	*Extended Reference Base Address.*	0
0x0A00	VE_MPEG4_QUE	RO	[31]	*Motion Flag.*	0
			[30:29]	*VOP Type.*	0
			[28:24]	*Channel.*	0
			[23:0]	*MPEG4 Code Address Offset.*	0
0x0A04	VE_JPEG_QUE	RO	[23:0]	*JPEG Code Address Offset.*	0

Table 4-34 shows *Motion JPEG Encoder Configuration Registers* [0x0670~0x0680]. The motion JPEG code size is controlled by quantization step size. The SOLO6010 supports 5 (1 to 5) level quantization step size control. Quantization step sizes are predefined on *Quantization Step0 register* [0x0670], *Quantization Step1 register* [0x0670], *Quantization Step2 register* [0x0670] and *Quantization Step3 register* [0x0670], respectively. Quantization step for each video input channel is set on *Quantization Step Table ID0 register* [0x0674], *Quantization Step Table ID1 register* [0x0674], *Quantization Step Table ID3 register* [0x0674], *Quantization Step Table ID4 register* [0x0674], *Quantization Step Table ID5 register* [0x0674], *Quantization Step Table ID6 register* [0x0674], *Quantization Step Table ID7 register* [0x0674], *Quantization Step Table ID8 register* [0x0674], *Quantization Step Table ID9 register* [0x0674], *Quantization Step Table ID10 register* [0x0674], *Quantization Step Table ID11 register* [0x0674], *Quantization Step Table ID12 register* [0x0674], *Quantization Step Table ID13 register* [0x0674], *Quantization Step Table ID14 register* [0x0674] and *Quantization Step Table ID15 register* [0x0674], respectively. Furthermore, the SOLO6010 produces two different quality motion JPEG codes for each video input channel. Quantization step for each extended video input channel is set on *Extended Quantization Step Table ID16 register* [0x0678], *Extended Quantization Step Table ID16 register* [0x0678], *Extended Quantization Step Table ID16 register* [0x0678], *Extended Quantization Step Table ID16 register* [0x0678], *Extended Quantization Step Table ID16 register* [0x0678], *Extended Quantization Step Table ID16 register* [0x0678], *Extended Quantization Step Table ID16 register* [0x0678], *Extended Quantization Step Table ID16 register* [0x0678], *Extended Quantization Step Table ID16 register* [0x0678], *Extended Quantization Step Table ID16 register* [0x0678], *Extended Quantization Step Table ID16 register* [0x0678], *Extended Quantization Step Table ID16 register* [0x0678], *Extended Quantization Step Table ID16 register* [0x0678], *Extended Quantization Step Table ID16 register* [0x0678], *Extended Quantization Step Table ID16 register* [0x0678] and *Extended Quantization Step Table ID16 register* [0x0678], respectively.

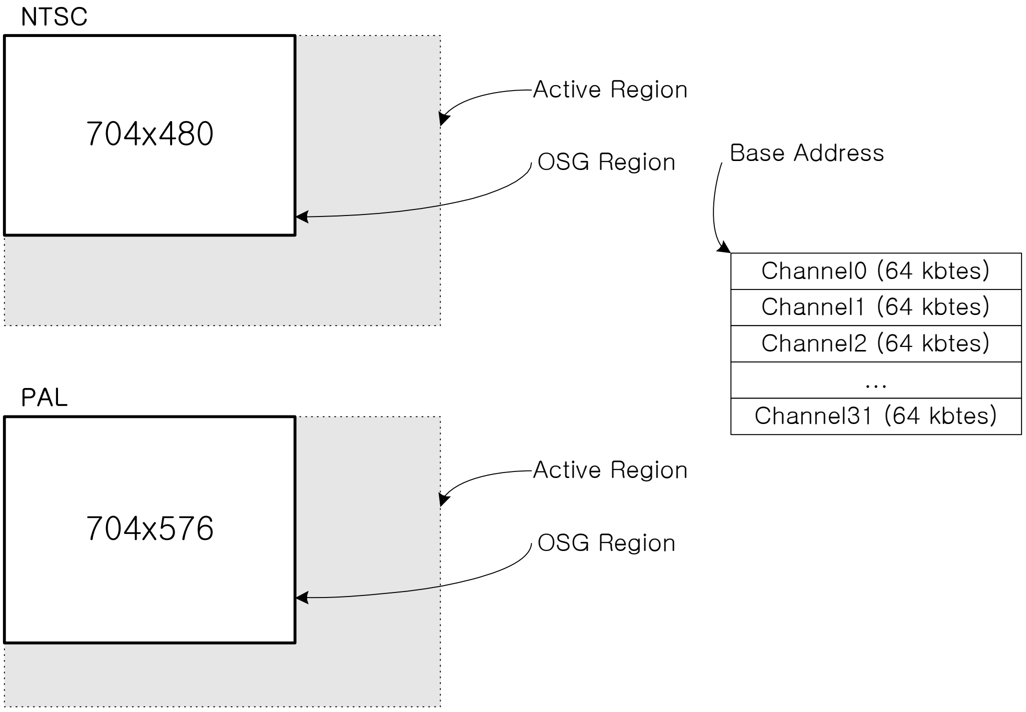
JPEG Page Size Register [0x067C] is configured with the JPEG page size. *JPEG Page Base Register* [0x067C] points the base address on the external SDRAM. *JPEG Encoder Enable Register* [0x0680] is set to 1 to activate motion JPEG encoder for video input channel0 to 31.

Table 4-34 Motion JPEG Encoder Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0670	VE_JPEG_QS	RW	[28:24]	*Quantization Step3.*	0
			[20:16]	*Quantization Step2.*	0
			[12:8]	*Quantization Step1.*	0
			[4:0]	*Quantization Step0.*	0
0x0674	VE_JPEG_QS_CH0	RW	[31:30]	*Quantization Step Table ID15.*	0
			[29:28]	*Quantization Step Table ID14.*	0
			[27:26]	*Quantization Step Table ID13.*	0
			[25:24]	*Quantization Step Table ID12.*	0
			[23:22]	*Quantization Step Table ID11.*	0
			[21:20]	*Quantization Step Table ID10.*	0
			[19:18]	*Quantization Step Table ID9.*	0

			[17:16]	*Quantization Step Table ID8.*	0
			[15:14]	*Quantization Step Table ID7.*	0
			[13:12]	*Quantization Step Table ID6.*	0
			[11:10]	*Quantization Step Table ID5.*	0
			[9:8]	*Quantization Step Table ID4.*	0
			[7:6]	*Quantization Step Table ID3.*	0
			[5:4]	*Quantization Step Table ID2.*	0
			[3:2]	*Quantization Step Table ID1.*	0
			[1:0]	*Quantization Step Table ID0.*	0
0x0678	VE_JPEG_QS_CH1	RW	[31:30]	*Extended Quantization Step Table ID31.*	0
			[29:28]	*Extended Quantization Step Table ID30.*	0
			[27:26]	*Extended Quantization Step Table ID29.*	0
			[25:24]	*Extended Quantization Step Table ID28.*	0
			[23:22]	*Extended Quantization Step Table ID27.*	0
			[21:20]	*Extended Quantization Step Table ID26.*	0
			[19:18]	*Extended Quantization Step Table ID25.*	0
			[17:16]	*Extended Quantization Step Table ID24.*	0
			[15:14]	*Extended Quantization Step Table ID23.*	0
			[13:12]	*Extended Quantization Step Table ID22.*	0
			[11:10]	*Extended Quantization Step Table ID21.*	0
			[9:8]	*Extended Quantization Step Table ID20.*	0
			[7:6]	*Extended Quantization Step Table ID19.*	0
			[5:4]	*Extended Quantization Step Table ID18.*	0
			[3:2]	*Extended Quantization Step Table ID17.*	0
			[1:0]	*Extended Quantization Step Table ID16.*	0
0x067C	VE_JPEG_CFG	RW	[23:16]	*JPEG Page Size.*	0
			[15:0]	*JPEG Page Base.*	
0x0680	VE_JPEG_CTRL	RW	[31:0]	*JPEG Encoder Enable.*	0

Fig. 4-43 shows the record OSG. The record OSG with 16 by 16 font format with 1 bit resolution is supported for 32 video input channels including 16 extended video input channels for the virtual channels. The dimension for a character is 16 x16 pixels (32 bytes). Each video input channel is covered the dimension with 1,024 x 512. Therefore, the memory space for each video input channel is 64 k bytes.



Channel0 (64 kbtes)
Channel1 (64 kbtes)
Channel2 (64 kbtes)
…
Channel31 (64 kbtes)

Fig. 4-43 Record OSG

Table 4-35 shows *Video Record OSG Registers* [0x0690~0x069C]. *Record OSG Enable Register* [0x0690] is used to activate the video record OSG for each video input channel. *Record OSG Base Address Register* [0x0694] points the base address on the external SDRAM. *Record OSG Y Register* [0x0698], *Record OSG U Register* [0x0698] and *Record OSG V Register* [0x0698] determine the record OSG color.

Table 4-35 Video Record OSG Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0690	VE_OSG_CH	RW	[31:0]	*Record OSG Enable.*	0
0x0694	VE_OSG_BASE	RW	[15:0]	*Record OSG Base Address.*	0
0x0698	VE_OSG_CLR	RW	[23:16]	*Record OSG Y.*	0
			[15:8]	*Record OSG U.*	0
			[7:0]	*Record OSG V.*	0
0x069C	VE_OSG_OPT	RW	[15]	*Record OSG Horizontal Shadow.*	0
			[14]	*Record OSG Vertical Shadow.*	0
			[13:7]	*Record OSG Horizontal Offset * 16.*	0
			[6:0]	*Record OSG Vertical Offset * 16.*	0

Fig. 4-44 shows the conceptual flow of the video decoder. Although the video encoder supports D1 (704 x 480 for NTSC, 704 x 576 for PAL), HD1 (704 x 240 for NTSC, 704 x 288 for PAL), CIF (352 x 240 for NTSC, 352 x 288 for PAL) and QCIF (176 x 120 for NTSC, 176 x 144 for PAL), the vide decoder supports only D1, HD1 and CIF resolution video because of the video display resolution. The video decoder scaler in Fig. 4-44 includes the auto adaptive scale processor by comparing source scale with target scale. The video display processor in Fig. 4-44 includes the interpolation, deinterlaced filter and horizontal x2 scaler.

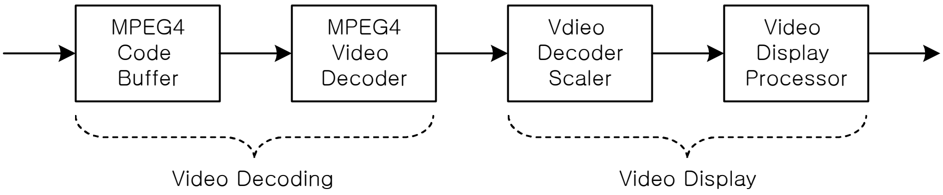


Fig. 4-44 Conceptual Flow of the Video Decoder

Fig. 4-45 (a), (b) and (c) show example when source scale is D1 real time, HD1 less than 30fps for NTSC or 25fps for PAL, CIF less than 30fps for NTSC or 25fps for PAL, respectively. Fig. 4-45 (b) should be work with interlace write and display interpolation for target scale 1, and Fig. 4-45 (c) should be worked with interlace write, display interpolation and display horizontal x2 for target scale 1, respectively.

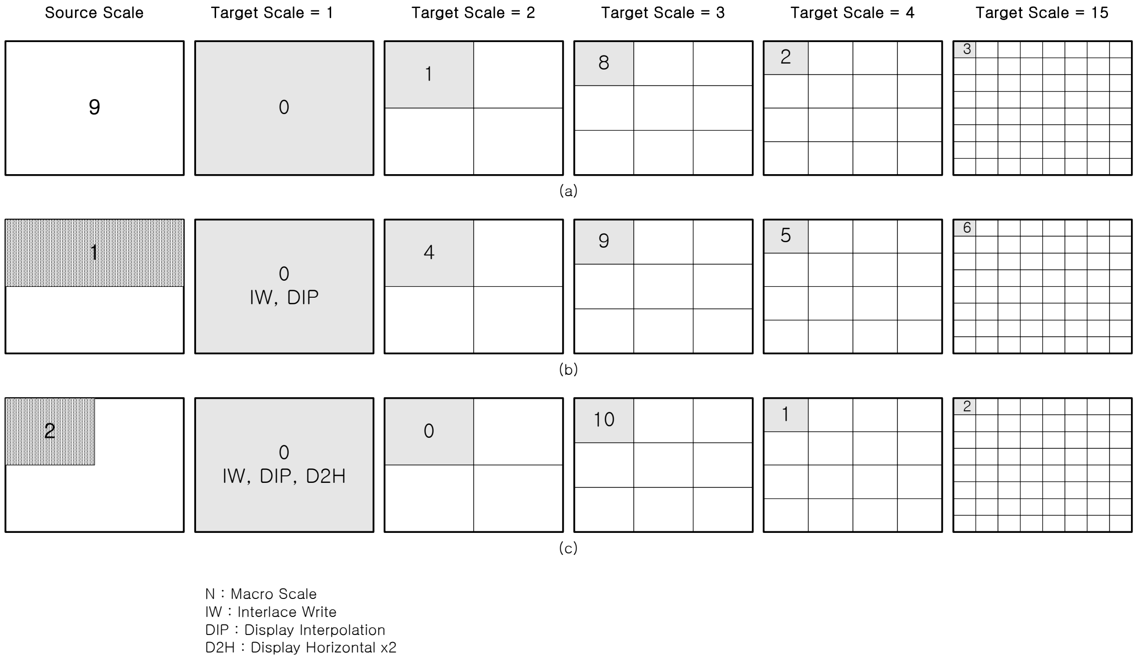


Fig. 4-45 Examples for Display Scale

Table 4-36 shows *MPEG4 Video Decoder Configuration Registers* [0x0900~0x0908]. *No Write if No Window Register* [0x0900] is set to 1 to protect decoding if no window ID. *Busy Wait When Code Busy Register* [0x0900], *Busy Wait When Res. Write Register* [0x0900], *Busy Wait When Ref. Read Register* [0x0900] and *Busy Wait When Macroblock Read Register* [0x0900] are used to distribute the FDMA bandwidth. *Decode Mode Register* [0x0900] is set to 1 to decode single MPEG4 video stream and 0 to decode multi MPEG4 video stream. *Display Control Mode Register* [0x0900] is set to 0 for auto display control and 1 for user defined display control. *Display Page Control Mode Register* [0x0900] is set to 0 for 2 page display mode and 1 for user defined page display mode. *Byte Order Register* [0x0900] should be set 0. *Decode Start Field Index Register* [0x0900] defines the field index to be decoded MPEG4 video stream. *Decode Lock When Error Register* [0x0900] is set to 1 for locking MPEG4 video decoder when decoder error is occurred. *Error Interrupt Enable Register* [0x0900] is set to 1 to use error interrupt mode. *Time Width Register* [0x0900] defined the bit width of MPEG4 VOP header time information. *DCT Interval Register* [0x0900] defines the decoding interval between macroblocks.

Table 4-36 MPEG4 Video Decoder Configuration Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0900	VD_CFG0	RW	[24]	*No Write if No Window.*	0
			[23]	*Busy Wait When Code Busy.*	0
			[22]	*Busy Wait When Res Write.*	0
			[21]	*Busy Wait When Ref Read.*	0
			[20]	*Busy Wait When Macroblock Read.*	0
			[18]	*Decode Mode* [0] : Multi [1] : Single	0

			[17]	*Display Control Mode.* [0] : Auto [1] : User	0
			[16]	*Display Page Control Mode.* [0] : 2 page auto [1] : User	0
			[15]	*Byte Order.*	0
			[14]	*Decode Start Field Index.*	0
			[13]	*Decode Lock When Error.*	0
			[12]	*Error Interrupt Enable.*	0
			[11:8]	*Time Width.*	0
			[7:0]	*DCT Interval.*	0

Table 4-37 shows *MPEG4 Video Decoder Control Registers* [0x090C~0x0940]. *Code Address Register* [0x090C] points the address to be decoded MPEG4 currently video code on the external SDRAM. *Decode On Register* [0x0910] is set to 1 to activate the MPEG4 video decoder. *Max Item Count When Multi Decode Mode Register* [0x0910] will determine the number of picture decoding at one time when multi page mode. *Interrupt by Command Ack. Register* [0x0920] will be asserted when the MPEG4 decoder is done. *Interrupt by Command Empty Register* [0x0920] will be asserted when the code buffer is empty. *Interrupt by Decode Error Register* [0x0920] will be asserted when MPEG4 video decoder meets errors. *Internal VLD Error Flags Register* [0x0924] will be asserted when internal VLD errors are occurred. *Decoded Byte Count Register* [0x0924] shows the actual decoded byte count. *Progressive Register* [0x0930] indicates progressive for the MPEG4 video decoding currently.

Interlace Register [0x0930] is set to 1 for interlaced video decoding. *Source FI Register* [0x0930] set to 0 or 1 for field0 or field1, respectively. *Source Channel ID Register* [0x0930] is configured with source video channel ID. *VOP Type Register* [0x0930] is configured with VOP type for the MPEG4. *Decode MPEG4 Code Size Register* [0x0930] is configured with MPEG4 code size. *Source Scale Register* [0x0934] is configured with MPEG4 source scale. *Target Window ID Register* [0x0934] is configured with target window channel ID. *Frame Interpolation On Register* [0x0934] is set 1 to process interpolation. *Horizontal Macroblock Size Register* [0x0934] and *Vertical Macroblock Size Register* [0x0934] are configured with the horizontal Macroblock size and vertical Macroblock size of the picture. *Ref. Address Horizontal Offset Register* [0x0938]. *Ref. Base Address Register* [0x0938]. *User Set Macro Scale Mode Register* [0x093C]. *User Set Macro Interlace Write Register* [0x093C]. *User Set Macro Read Interpolation Register* [0x093C]. *User Set Display Horizontal Zoom Register* [0x093C]. *User Set Display SX Register* [0x093C]. *User Set Display SY Register* [0x093C]. *User Set Decode Write Page Register* [0x0940]. *User Set Decode Read Page Register* [0x0940]. *Next Code Address Register* [0x0944]

Table 4-37 MPEG4 Video Decoder Control Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x090C	VD_CODE_ADDR	RW	[31:0]	*Code Address.*	0
0x0910	VD_CTRL	RW	[31]	*Decoder On.*	0
			[11:0]	*Max Item Count When Multiple Decode Mode.*	0
0x0920	VD_STATUS0	RO	[22]	*Interrupt by Command Ack.*	-
			[21]	*Interrupt by Command Empty.*	-

			[20]	*Interrupt by Decode Error.*	-
			[11:0]	*Rest Number of Reserved Decoding.* Only for multiple Decode Mode.	-
0x0924	VD_STATUS1	RO	[23:20]	*Internal VLD Error Flags.*	-
			[19:0]	*Decoded Byte Count.*	
0x0930	VD_IDX0	RW	[31]	*Progressive.*	0
			[30]	*Interlace.*	0
			[29]	*Source FI.*	0
			[27:24]	*Source Channel ID.*	0
			[23:22]	*VOP Type.*	0
			[21]	*Decode Page Stop.*	0
			[20]	*Sync Start.*	0
			[19:0]	*MPEG4 Code Size.*	0
0x0934	VD_IDX1	RW	[31:28]	*Source Scale.*	0
			[27:24]	*Target Window ID.*	0
			[16]	*Frame Interpolation On.*	0
			[15:8]	*Horizontal Macroblock Size.*	0
			[7:0]	*Vertical Macroblock Size.*	0
0x0938	VD_IDX2	RW	[31]	*Ref Address Horizontal Offset.* [0] : 0 [1] : 512 (When use CIF)	0
			[15:0]	*Ref Base Address.*	0
0x093C	VD_IDX3	RW	[31:28]	*User Set Macro Scale Mode.*	0
			[27]	*User Set Macro Interlace Write.*	0
			[26]	*User Set Macro Read Interpolation.*	0
			[25]	*User Set Display Horizontal Zoom.*	0
			[24]	*Reserved.*	0
			[23:12]	*User Set Display SX.*	0
			[11:0]	*User Set Display SY.*	0
0x0940	VD_IDX4	RW	[7:0]	*User Set Decode Write Page.*	0
			[2]	*User Set Display Read Page.*	0
0x0944	VD_IDX5	RW	[31:0]	*Next Code Address*	0

4.5. G.723 Voice CODEC

The SOLO6010 has two full hardware 10 channel G.723 voice CODEC engines to support total 20 channels which are an ITU-T recommendation for voice CODECs that use the ADPCM method and provides quality audio at 20 or 40 Kbps. Fig. 4-46 shows G.723 simplified block diagram. The voice CODEC engines support I2S slave/master interface and u-Law interface according to register configuration. When it is used on I2S master interface, multi channel interface is possible though only channel 0 (pin ioPCM_S0). The sampling rate should be less than 10 kHz on 54MHz system clock and 20 kHz on 108MHz system clock.

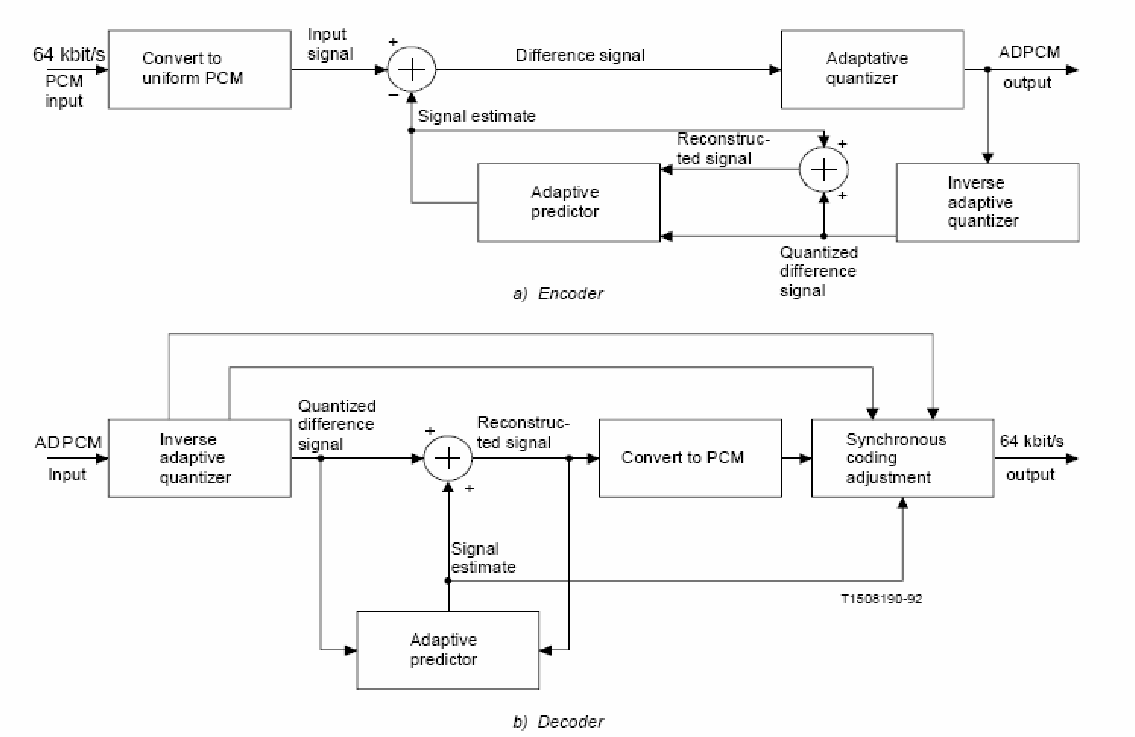


Fig. 4-46 G.723 Simplified Block Diagram

Fig. 4-47 shows the timing diagram of u-Law, where ioPCM_SYNC, ioPCM_CLK and ioPCM_SD represent synchronization, serial clock signal and serial data signal at w-Law, respectively. In case of u-Law, one serial data pin can interface with one channel. Therefore, only 10 channels are supported at u-Law interface. Otherwise, 20 channels are supported at I2S interface because one data pin interface with two channels (right and left channels).

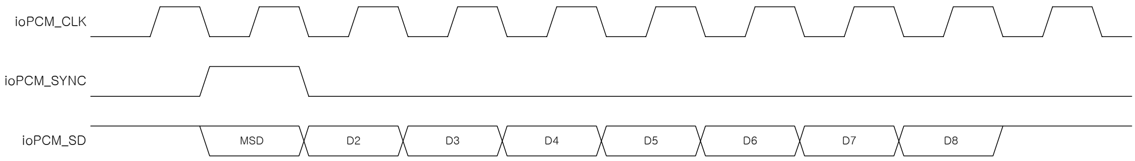


Fig. 4-47 Timing Diagram of u-Law

Fig. 4-48 shows the simple system configurations and basic interface timing.

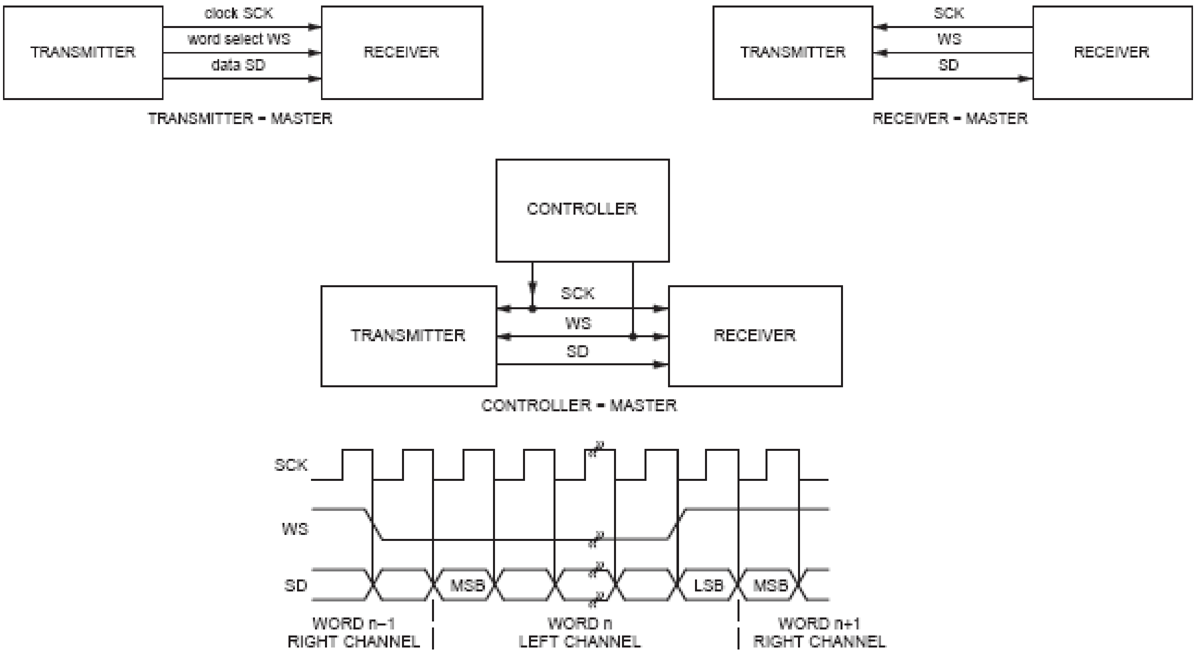
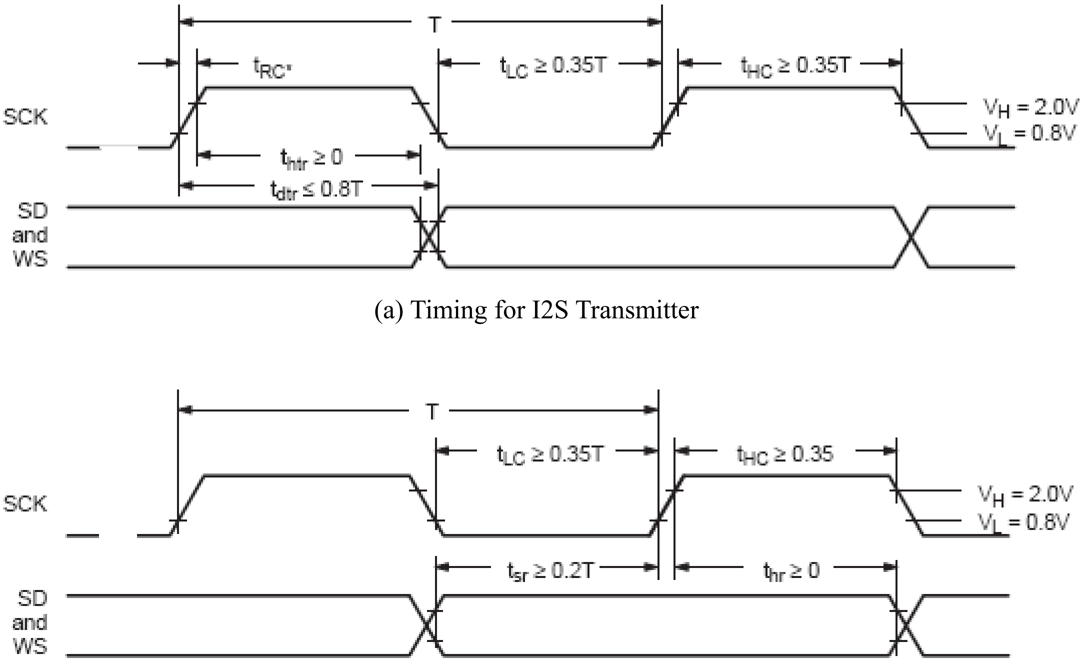


Fig. 4-48 I2S Simple System Configurations and Basic Interface Timing

Fig. 4-49 shows the timing diagram for I2S interface.



(a) Timing for I2S Transmitter

(b) Timing for I2S Receiver

Fig. 4-49 Timing Diagram for I2S interface

Fig. 4-50 shows the code address structure of 20 channel G.723 voice CODEC. The memory space allocated on the external SDRAM is 64 kyte. The encoding code area is allocated on lower 32 kByte and the decoding code area is allocated on upper 32 kByte. Each area is composed of 32 pages and its size is 1 kByte. Each page includes 960 Byte (20 channel * 48 Byte) and 64 Byte dummy space. One page size is the code size for every interrupt when *Interrupt Page Count Mode Register* [0x0D08] is set to 0. In other word, 48 byte is the code size for 128 code (16msec on 8 kHz sampling, 8msec on 16 kHz sampling).

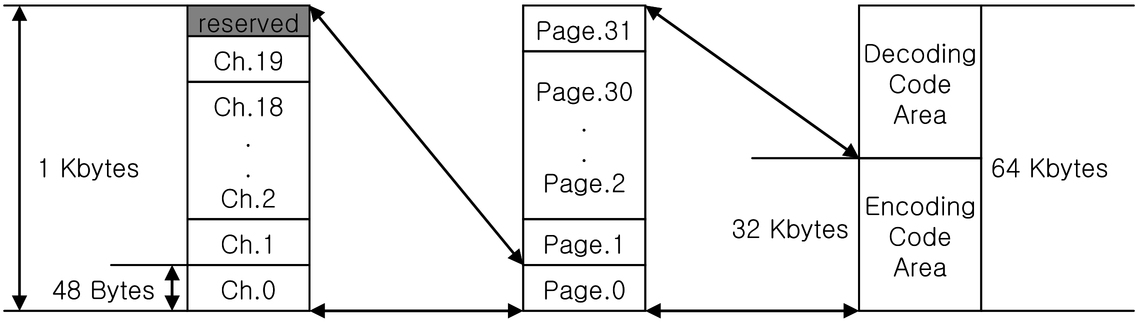


Fig. 4-50 Code Address Structure

Table 4-39 shows *Audio Registers* [0x0D00~0x0D14]. *Audio CODEC Enable Register* [0x0D00] should be set to 1 to activate G.723 Voice CODEC. *Master Mode Register* [0x0D00] is set to 0 when pin ioPCM_CLK and ioPCM_SYNC are used as input pin for I2S slave mode, and 1 when pin ioPCM_CLK and ioPCM_SYNC are used as output pin for I2S master mode. *Interface Mode Register* [0x0D00] is set to 0 for u-Law interface and 1 for I2S interface. *Left-Right Word Swap Register* [0x0D00] is effective only for I2S interface and used to swap left right word at input and output interface. *Bit Width Register* [0x0D00] will decide the word width as 16 bit for 0 and 8 bit for 1. However, u-Law interface supports only 8 bit interface. *Multi Channel Register* [0x0D00] is only effective IS2 slave encoding mode. Table 4-38 shows multi channel mode.

Table 4-38 I2S Multi Channel Mode

IO PIN	Ch. (Code)			
	[25:24] = 0	[25:24] = 1	[25:24] = 2	[25:24] = 3
ioPCM_S0	0, 1	0, 1, 2, 3	0 ~ 7	0 ~ 15
ioPCM_S1	2, 3	-	-	-
ioPCM_S2	4, 5	4, 5, 6, 7	-	-
ioPCM_S3	6, 7	-	-	-
ioPCM_S4	8, 9	8, 9, 10, 11	8 ~ 15	-
ioPCM_S5	10, 11	-	-	-
ioPCM_S6	12, 13	12, 13, 14, 15	-	-
ioPCM_S7	14, 15	-	-	-
ioPCM_S8	16, 17	16, 17	16, 17	16, 17
ioPCM_S9	18, 19	18, 19	18, 19	18, 19

Mix Enable for Ch9~Ch0 Register [0x0D00] can be set to 1 to mix Ch0 to Ch9 all audio signal to output Ch9 when Ch9 is set as decoding mode. *Decoding Volume Control for Ch19~Ch10 Register* [0x0D00] is used to control the audio output volume for Ch0 to Ch9. *Mix Enable for Ch19~Ch10 Register* [0x0D00] can be set to 1 to mix Ch19 to Ch10 all audio signal to output Ch19 when Ch19 is set as decoding mode. *Decoding Volume Control for Ch19~Ch10 Register* [0x0D00] is used to control the audio output volume for Ch19 to Ch10. Channel I/O Mode Register [0x0D00] configures input or output mode.

Serial Clock Number Register [0x0D04] is only effective for I2S master mode and its value decides the clock number for one period ioPCM_SYNC signal. *Clock Divider Register* [0x0D04] is only effective for I2S master mode and used to adjust the sampling rate which can be set to 8 kHz or 16 kHz. The sample rate should be set as less than 10 kHz at 54 MHz system clock mode and 20 kHz at 108 MHz system clock mode. The serial clock range is different according to audio AD/DA chips. The serial clock can be configured to the appropriate value and the clock number can be defined after determining the serial clock. Theses value should be set to the appropriate value to operating normally. The calculation examples are shown as followings.

Ex1) Sampling frequency = 16 kHz
M = 384 bit => Clock frequency = 16 * 384 = 6.144 MHz
N = 108/9.184 = 8.78 ≈ 9
Ex2) Sampling frequency = 8 kHz
M = 256 bit => Clock frequency = 8 * 256 = 2.048 MHz
N = 108/4.092 = 26.39 ≈ 26
Ex3) Sampling frequency = 8 kHz
Clock frequency = 200 kHz => M = 200/8 = 25
N = 108/0.4 = 270

DMA Wait Interval Register [0x0D08] will define the interval (clock number) between each channel to transact code data with external SDRAM. *Interrupt Page Count Mode Register* [0x0D08] decide the number page. The interrupt interval will be longer and the length of transaction at one time will be more according to bigger value. *External Memory Base Address Register* [0x0D08] points the base address on external SDRAM.

Encoding Volume Control for Ch0~Ch9 [0x0D0C] and *Encoding Volume Control for Ch10~Ch19* [0x0D10] are used to control the volume for each channel. *Channel Count Register* [0x0D14] and *Page Count Register* [0x0D14] will inform the current access channel and page, respectively.

Table 4-39 Audio Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0D00	ADO_CTRL0	RW	[31]	*Audio CODEC Enable*.	0
			[30]	*Master Mode.* [0] : CLK/SYNC IN [1] : CLK/SYNC OUT	0
			[29]	*Interface Mode.* [0] : u-Law [1] : I2S	0
			[28]	*Reserved.*	0
			[27]	*Left-Right Word Swap*. Only for I2S mode.	0
			[26]	*Bit Width*. Only for I2S. [0] : 16bit [1] : 8bit	0
			[25:24]	*Multi Channel*. Only for I2S slave mode. [0] : 2 Channel [1] : 4 Channel [2] : 8 Channel [3] : 16 Channel	0
			[23]	*Mix Enable for Ch9 ~ Ch0*.	0
			[22:20]	*Decoding Volume Control for Ch9 ~ Ch0.*	0

			[19]	*Mix Enable for Ch19 ~ Ch10*.	0
			[18:16]	*Decoding Volume Control for Ch19 ~ Ch10.*	0
			[9:0]	*Channel I/O Mode.* PCM mode : [Ch9, Ch8, Ch7, …, Ch0] Ch(x) = 0 : Encoding Ch(x) = 1 : Decoding I2S mode : [{Ch19, Ch18}, {Ch17, Ch16}, {Ch15, Ch14}, …, {Ch1, Ch0}] {Ch(x1),Ch(x0)} = 0 : Encoding {Ch(x1),Ch(x0)} = 1 : Decoding	0
0x0D04	ADO_SAMPLING	RW	[31]	*Short Mode for Simulation*. Only for Test.	0
			[30]	*EE Mode for Testing*. Only for Test.	0
			[29:25]	*EE Mode Encoding Channel for Testing*. Only for Test	0
			[24:16]	*Serial Clock Number*. (Only for I2S Master Mode) M (Clock Frequency / Sampling)	0x18
			[8:0]	*Clock Divider*. (Only for I2S Master Mode) N (Clock Frequency = SYS_CLK / (N*2) Hz)	0x1A5
0x0D08	ADO_FDMA_INTR	RW	[31:19]	*DMA Wait Interval.*	
			[18:16]	*Interrupt Page Count Mode.* The range will be allowed 0~4. 2^n pages will be applied.	
			[15:0]	*External Memory Base Address*.	0
0x0D0C	AUDIO_EVOL0	RW	[29:27]	*Encoding Volume Control for Ch9*. [000] : x1 [100] : x1 [001] : x2 [101] : x1/2 [010] : x4 [110] : x1/4 [011] : x8 [111] : x1/8	0
			[26:24]	*Encoding Volume Control for C8*.	0
			[23:21]	*Encoding Volume Control for Ch7*.	0
			[20:18]	*Encoding Volume Control for Ch6*.	0
			[17:15]	*Encoding Volume Control for Ch5*.	0
			[14:12]	*Encoding Volume Control for Ch4*.	0
			[11:9]	*Encoding Volume Control for Ch3*.	0
			[8:6]	*Encoding Volume Control for Ch2*.	0
			[5:3]	*Encoding Volume Control for Ch1*.	0
			[2:0]	*Encoding Volume Control for Ch0*.	0
0x0D10	AUDIO_EVOL1	RW	[29:27]	*Encoding Volume Control for Ch19*. [000] : x1 [100] : x1 [001] : x2 [101] : x1/2 [010] : x4 [110] : x1/4 [011] : x8 [111] : x1/8	0
			[26:24]	*Encoding Volume Control for Ch18*.	0
			[23:21]	*Encoding Volume Control for Ch17*.	0
			[20:18]	*Encoding Volume Control for Ch16*.	0
			[17:15]	*Encoding Volume Control for Ch15*.	0
			[14:12]	*Encoding Volume Control for Ch14*.	0
			[11:9]	*Encoding Volume Control for Ch13*.	0
			[8:6]	*Encoding Volume Control for Ch12*.	0
			[5:3]	*Encoding Volume Control for Ch11*.	0
			[2:0]	*Encoding Volume Control for Ch10*.	0
0x0D14	ADO_STA	RW	[31:10]	*Reserved*. (FSM for Debugging.)	0
			[9:5]	*Channel Count.* (Accessing Channel 0 ~ 19)	0
			[4:0]	*Page Count.* (Accessing Page 0 ~ 31)	0

4.6. Peripherals

4.6.1. GPIO

SOLO6010 supports two 16bit GPIO for sensor input or relay output. ioGPIO[15:0] is designed overlapped with I2C, UART, PS2 or SPI. ioGPIO[31:16] is designed for only extended GPIO. Table 4-40 shows *GPIO Registers* [0x0B00~0x0B1C]. The configuration method of ioGPIO[15:0] is explained detail in *GPIO[15:0] Configuration Register* [0x0B00] of Table 4-40. *GPIO[31:16] Enable Register* [0x0B04] should be set to 1 for each bit in order to use ioGPIO[31:16]. The configuration method of ioGPIO[31:16] is explained in *GPIO[31:16] Configuration Register* [0x0B04] of Table 4-40.

GPIO Output Data Register [0x0B08] is used to write GPIO output data. *GPIO Input Data Register* [0x0B0C] is used to read GPIO input data. *GPIO Interrupt Status/Ack Register* [0x0B10] is worked as status when read and acknowledge when written. *GPIO Interrupt Enable Register* [0x0B14] needs to be set to use GPIO interrupt. The GPIO interrupt events are configured as rising edge, falling edge, both edge or low level. *GPIO[15:0] Interrupt Configuration Register* [0x0B18] and *GPIO[31:16] Interrupt Configuration Register* [0x0B1C] is used for GPIO interrupt configuration as shown in Table 4-40.

Table 4-40 GPIO Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0B00	GPIO_CONFIG0	RW	[31:30]	*GPIO[15] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL7 [3] : RS_nRTS1	0
			[29:28]	*GPIO[14] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL7 [3] : RS_nCTS1	0
			[27:26]	*GPIO[13] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL6 [3] : RS_TXD1	0
			[25:24]	*GPIO[12] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL6 [3] : RS_nRXD1	0
			[23:22]	*GPIO[11] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL5 [3] : RS_TxEn0	0
			[21:20]	*GPIO[10] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL5 [3] : RS_RxEn0	0
			[19:18]	*GPIO[9] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL4 [3] : RS_TxD0	0

			[17:16]	*GPIO[8] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL4 [3] : RS_RxD0	0
			[15:14]	*GPIO[7] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL3 [3] : PS2_C1	0
			[13:12]	*GPIO[6] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL3 [3] : PS2_D1	0
			[11:10]	*GPIO[5] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL2 [3] : PS2_C0	0
			[9:8]	*GPIO[4] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL2 [3] : PS2_D0	0
			[7:6]	*GPIO[3] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL1 [3] : SPI_STB	0
			[5:4]	*GPIO[2] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL1 [3] : SPI_CLK	0
			[3:2]	*GPIO[1] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL0 [3] : SPI_SDI	0
			[1:0]	*GPIO[0] Configuration*. [0] : Input [1] : Output [2] : I2C_SCL0 [3] : SPI_SDO	0
0x0B04	GPIO_ CONFIG1	RW	[31:16]	*GPIO[31:16] Enable*. [0] : ATA DIO[15:0] [1] : GPIO[31:16]	0
			[15]	*GPIO[31] Configuration*. [0] : Input [1] : Output	0
			[14]	*GPIO[30] Configuration*. [0] : Input [1] : Output	0
			[13]	*GPIO[29] Configuration*. [0] : Input [1] : Output	0
			[12]	*GPIO[28] Configuration*. [0] : Input [1] : Output	0
			[11]	*GPIO[27] Configuration*. [0] : Input [1] : Output	0
			[10]	*GPIO[26] Configuration*. [0] : Input [1] : Output	0
			[9]	*GPIO[25] Configuration*. [0] : Input [1] : Output	0

			[8]	*GPIO[24] Configuration*. [0] : Input [1] : Output	0
			[7]	*GPIO[23] Configuration*. [0] : Input [1] : Output	0
			[6]	*GPIO[22] Configuration*. [0] : Input [1] : Output	0
			[5]	*GPIO[21] Configuration*. [0] : Input [1] : Output	0
			[4]	*GPIO[20] Configuration*. [0] : Input [1] : Output	0
			[3]	*GPIO[19] Configuration*. [0] : Input [1] : Output	0
			[2]	*GPIO[18] Configuration*. [0] : Input [1] : Output	0
			[1]	*GPIO[17] Configuration*. [0] : Input [1] : Output	0
			[0]	*GPIO[16] Configuration*. [0] : Input [1] : Output	0
0x0B08	GPIO_DATA_OUT	RW	[31:0]	*GPIO Output Data*.	0xffff_ffff
0x0B0C	GPIO_DATA_IN	RW	[31:0]	*GPIO Input Data*.	-
0x0B10	GPIO_INT_STA/ACK	RW	[31:0]	*GPIO Interrupt Status/Ack.* Read Mode : GPIO[31:0] Interrupt Status. Write Mode : GPIO[31:0] Interrupt Ack.	0
0x0B14	GPIO_INT_ENA	RW	[31:0]	*GPIO Interrupt Enable*. GPIO[31:0] Interrupt Enable.	0
0x0B18	GPIO_INT_CFG0	RW	[31:30]	*GPIO[15] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[29:28]	*GPIO[14] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[27:26]	*GPIO[13] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[25:24]	*GPIO[12] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[23:22]	*GPIO[11] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[21:20]	*GPIO[10] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[19:18]	*GPIO[9] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0

			[17:16]	*GPIO[8] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[15:14]	*GPIO[7] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[13:12]	*GPIO[6] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[11:10]	*GPIO[5] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[9:8]	*GPIO[4] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[7:6]	*GPIO[3] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[5:4]	*GPIO[2] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[3:2]	*GPIO[1] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[1:0]	*GPIO[0] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
0x0B1C	GPIO_INT_CFG1	RW	[31:30]	*GPIO[31] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[29:28]	*GPIO[30] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[27:26]	*GPIO[29] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[25:24]	*GPIO[28] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[23:22]	*GPIO[27] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0

			[21:20]	*GPIO[26] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[19:18]	*GPIO[25] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[17:16]	*GPIO[24] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[15:14]	*GPIO[23] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[13:12]	*GPIO[22] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[11:10]	*GPIO[21] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[9:8]	*GPIO[20] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[7:6]	*GPIO[19] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[5:4]	*GPIO[18] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[3:2]	*GPIO[17] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0
			[1:0]	*GPIO[16] Interrupt Configuration*. [0] : Rising Edge [1] : Falling Edge [2] : Both Edge [3] : Low Level	0

4.6.2. I2C

The SOLO6010 contains 8 I2C channels which are able to support only one channel at one time because the I2C core is extended 8 channels. In order to activate I2C pins, refer to *GPIO[15:0] Configuration Register* [0x0B00] in Table 4-40. The SOLO6010 can transfer one byte at one time using control signals (Ack, Start, Stop, Read, Write) and data signals (Tx Data, Rx Data) though I2C. The host CPU can use the interrupt to indicate arbitration lost when one byte transfer is finished. Fig. 4-51 and Fig. 4-52 show the block diagram and the waveform of IIC.

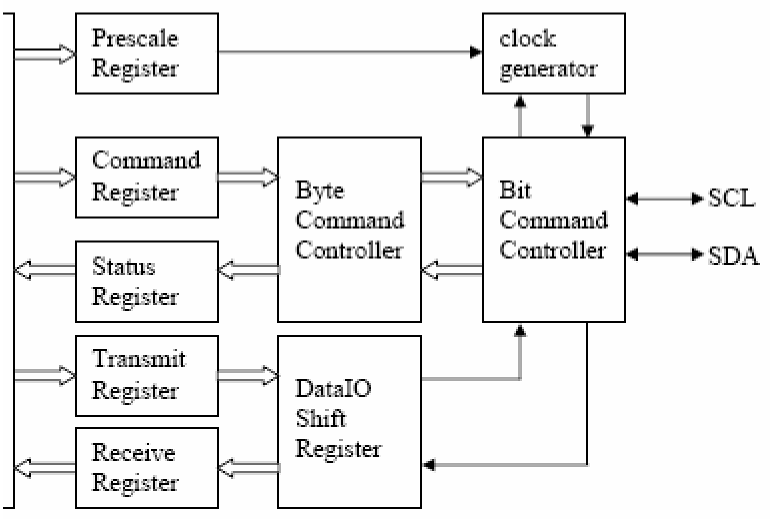


Fig. 4-51 The Block Diagram of I2C

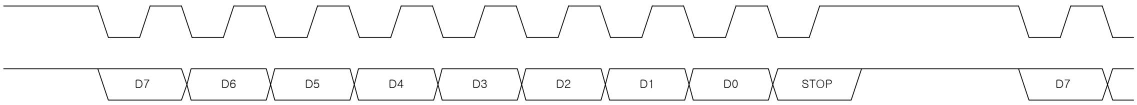


Fig. 4-52 The Waveform of I2C

I2C Module Enable Register [0x0B20] is use to activate I2C core. *Clock Pre-Scale Register* [0x0B20] is use to control I2C clock frequency. *Auto Clear Request Register* [0x0B24], *Received Acknowledge from Slave Register* [0x0B24], *I2C Bus Busy Register* [0x0B24], *Arbitration Lost Register* [0x0B24] and *Transfer in Progress Register* [0x0B24] are use to read I2C core status. *ACK Register* [0x0B24], *START Register* [0x0B24], *STOP Register* [0x0B24], *READ Register* [0x0B24] and *WRITE Register* [0x0B24] are use to control I2C core. *Tx Data Register* [0x0B28] is used to send data and *Rx Data Register* [0x0B2C] is used to receive data.

Table 4-41 I2C Register

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0B20	IIC_CFG	RW	[8]	*I2C Module Enable*.	0
			[7:0]	*Clock Pre-Scale*. (SYS_CLK/5) / (PR[7:0] * 16+1)	8
0xB24	IIC_CTRL	RO	[20]	*Auto Clear Request*. The register sets to “1” to clear START, STOP, READ and WRITE.	-

			[19]	*Received Acknowledge from Slave*. [0] : ACK received [1] : ACK not received	-
			[18]	*I2C bus Busy*. [0] : after STOP signal detected [1] : after START signal detected	-
			[17]	*Arbitration Lost*. a) STOP Signal is detected, but none requested. b) The master drives SDA high, but SDA low.	-
			[16]	*Transfer in Progress*. [0] : Complete [1] : Transferring	-
		RW	[7:5]	*Channel Select*. [000] : Channel 0 [001] : Channel 1 [010] : Channel 2 [011] : Channel 3 [100] : Channel 4 [101] : Channel 5 [110] : Channel 6 [111] : Channel 7	0
			[4]	*ACK*. It should be set to appropriate value if receiver is set to ACK=0 or ACK=1.	0
			[3]	*START*. The register is set or reset by user or reset by Auto Clear Request.	0
			[2]	*STOP*. The register is set or reset by user or reset by Auto Clear Request.	0
			[1]	*READ*. The register is set or reset by user or reset by Auto Clear Request.	0
			[0]	*WRITE*. The register is set or reset by user or reset by Auto Clear Request.	0
0x0B28	IIC_TXD	RW	[7:0]	*Tx Data*.	0
0x0B2C	IIC_RXD	RO	[7:0]	*Rx Data*.	-

4.6.3. SPI

The SOLO6010 supports SPI salve. In order to activate SPI pins, refer to *GPIO[15:0] Configuration Register* [0x0B00] in Table 4-40. Fig. 4-53 shows the waveform of SPI. SPI_STB, SPI_CK, SPI_DO and SPI_DI are strobe, clock, output and input, respectively.

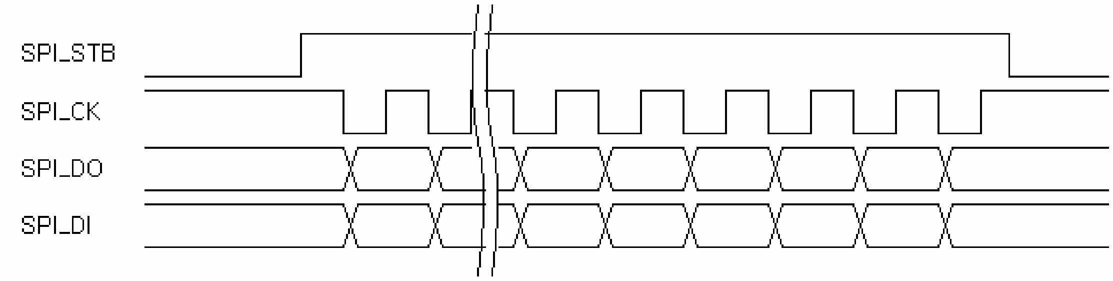


Fig. 4-53 The Waveform of SPI

SPI Tx Data[31:0] Register [0x0B40], *SPI Tx Data[63:32] Register* [0x0B44] and *SPI Tx*

Data[79:64] Register [0x0B48] are use to send 80bit data. *SPI Rx Data[31:0] Register* [0x0B50], *SPI Rx Data[63:32] Register* [0x0B54] and *SPI Rx Data[79:64] Register* [0x0B58] are use to send 80bit data.

Table 4-42 SPI Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0B40	SPI_TXD0	R/W	[31:0]	*SPI Tx Data[31:0].*	0
0x0B44	SPI_TXD1	R/W	[31:0]	*SPI Tx Data[63:32].*	0
0x0B48	SPI_TXD2	R/W	[15:0]	*SPI Tx Data[79:64].*	0
0x0B50	SPI_RXD0	R	[31:0]	*SPI Rx Data[31:0].*	-
0x0B54	SPI_RXD1	R	[31:0]	*SPI Rx Data[63:32].*	-
0x0B58	SPI_RXD2	R	[15:0]	*SPI Rx Data[79:64].*	-

4.6.4. PS2

The SOLO6010 supports two PS2 channels. In order to activate PS2 pins, refer to *GPIO[15:0] Configuration Register* [0x0B00] in Table 4-40. There are differences between two PS2 channels. PS2 channel0 contains 32 depth Rx Buffer and RS2 channel1 contains 8 depth Rx Buffer. Therefore, PS2 channel0 is strongly recommended for mouse interface and PS2 channel1 is recommended for key board interface.

Fig. 4-54 shows the timing diagram of PS2 Interface. PS2 clock frequency should be 10 to 16.7kHz. The following timing specification should be kept.

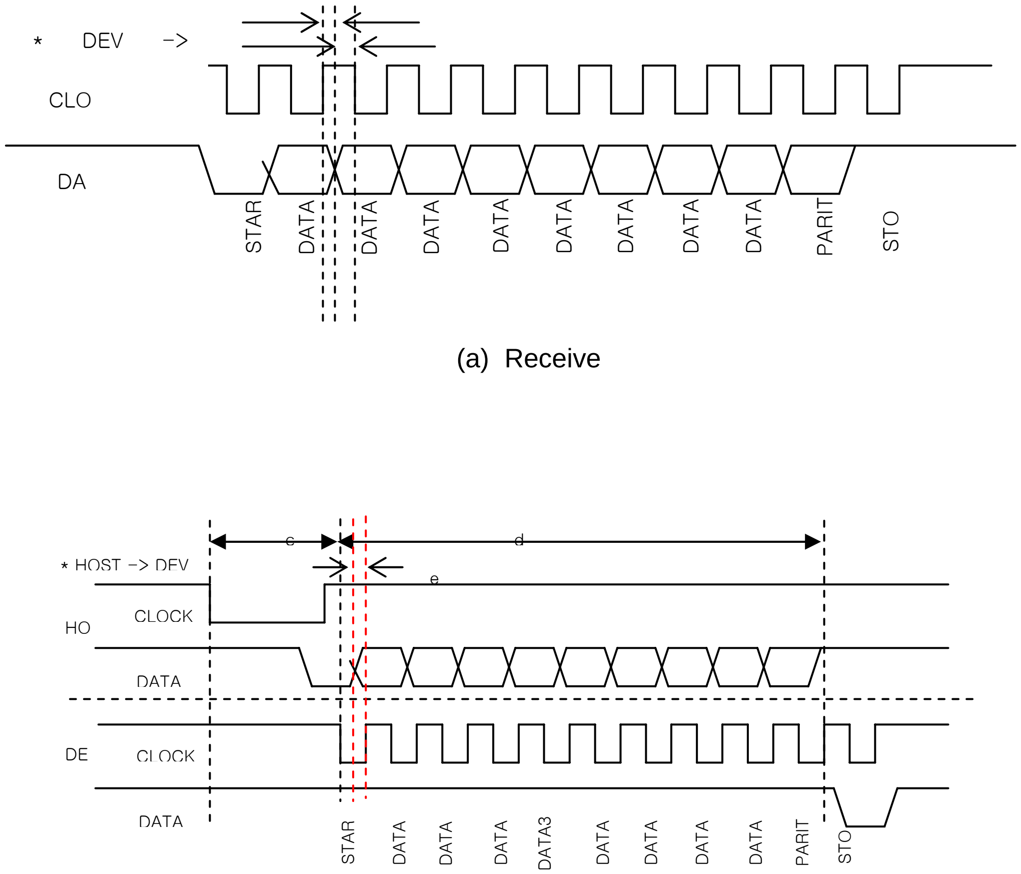
a > 5 usec

5usec < b < 25usec

100usec < c < 15msec

d < 2msec

e > 15usec (experimentally)



(a) Receive

(b) Transmit

Fig. 4-54 Timing Diagram of PS2 Interface

Table 4-43 shows *PS2 Enable Registers* [0x0B60~0x0B8C]. *PS2 Enable Register* [0x0B60, 0x0B80] should be set to 1 to activate PS2. Rx Data End, Tx Data End and Error interrupts are possible for PS2 interface. For detecting PS2 error, Tx Timeout, Rx Timeout, Rx Overflow and Rx Parity Error are able to be detected using PS2 status registers. *PS2 Tx Data Interrupt Enable Register* [0x0B60, 0x0B80], *PS2 Rx Data Interrupt Enable Register* [0x0B60, 0x0B80] and *PS2 Error Interrupt Register* [0x0B60, 0x0B80] should be set to 1 to use interrupt.

PS2 Sampling Period Register [0x0B60, 0x0B80] is used to determine the clock enable period of PS2 core. As shown in Fig. 4-54, the frequency range should be controlled from 10 kHz to 16.7 kHz. The value of the register is 0x3 for 54MHz system clock or 0x4 for 108MHz system clock.

The setting values for *PS2 Tx Clock Holding Time Register* [0x0B60, 0x0B80], *PS2 Rx Timeout Limitation Register* [0x0B60, 0x0B80] and *PS2 Tx Timeout Limitation Register* [0x0B60, 0x0B80] are recommended as 150usec, 300usec and 300usec, respectively. The appropriate configuration values are coded in the SOLO6010 driver software.

The status registers of the PS2 are defined as *PS2 Tx Buffer Status Register* [0x0B64, 0x0B84], *PS2 Rx Data End Register* [0x0B64, 0x0B84], *PS2 Tx Buffer Full Register* [0x0B64, 0x0B84], *PS2 Tx Busy Register* [0x0B64, 0x0B84], *PS2 Tx Timeout Register* [0x0B64, 0x0B84], *PS2 Rx Buffer Status Register* [0x0B64, 0x0B84], *PS2 Rx Data End Register* [0x0B64, 0x0B84], *PS2 Rx Buffer Empty Register* [0x0B64, 0x0B84], *PS2 Rx Busy Register* [0x0B64, 0x0B84], *PS2 Rx Timeout Register* [0x0B64, 0x0B84], *PS2 Rx Overflow Register* [0x0B64, 0x0B84] and *PS2 Rx Parity Error Register* [0x0B64, 0x0B84]. These status registers inform the end of operation or error status and the error status will be remained until *PS2 Enable Registers* [0x0B60] is set to 0.

The data will be stored on the internal buffer after writing data *PS2 Tx Data Register* [0x0B68, 0x0B88], and setting and resetting on *PS2 Tx Push Register* [0x0B68, 0x0B88]. Similarly, the data will be read from the internal buffer after reading the value of *PS2 Rx Data Register* [0x0B6C, 0x0B8C], and setting and resetting on *PS2 Rx Push Register* [0x0B6C, 0x0B8C].

Table 4-43 PS2 Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0B60 0x0B80	PS2_CFG0 PS2_CFG1	RW	[31:24]	*PS2 Tx Timeout Limitation.* Sampling Period * (n+1)	0x10
		RW	[23:16]	*PS2 Rx Timeout Limitation.* Sampling Period * (n+1)	0x08
		RW	[15:8]	*PS2 Tx Clock Holding Time.* Sampling Period * (n+1)	0x10
		RW	[6:4]	*PS2 Sampling Period.* Clock Dividing : 2^(n+6) * SYS_CLK period Where, 0 <= n <= 5	0x04
		RW	[3]	*PS2 Tx Data Interrupt Enable.*	0
		RW	[2]	*PS2 Rx Data Interrupt Enable.*	0
		RW	[1]	*PS2 Error Interrupt Enable.*	0
		RW	[0]	*PS2 Enable.*	0
0x0B64 0x0B84	PS2_STA0 PS2_STA1	RO	[26:24]	*PS2 Tx Buffer Status.* [0] : Full [4] : Empty	-
		RO	[23]	*PS2 Tx Data End.*	-
		RO	[22]	*PS2 Tx Buffer Full.*	-
		RO	[21]	*PS2 Tx Busy.*	-
		RO	[20]	*PS2 Tx Timeout.* Cleared by Core Reset.	-
		RO	[19:16]	*PS2 Tx Status of FSM.* Only for Debugging.	-
		RO	[15:10]	*PS2 Rx Buffer Status.* [0] : Full [4] : Empty	-
		RO	[9]	*PS2 Rx Data End.*	-
		RO	[8]	*PS2 Rx Buffer Empty.*	-
		RO	[7]	*PS2 Rx Busy.*	-
		RO	[6]	*PS2 Rx Timeout.* Cleared by Core Reset.	-
		RO	[5]	*PS2 Rx Overflow.* Cleared by Core Reset.	-
		RO	[4]	*PS2 Rx Parity Error.* Cleared by Core Reset.	-
		RO	[3:0]	*PS2 Rx Status of FSM.* Only for Debugging.	-
0x0B68	PS2_TXD0	RW	[8]	*PS2 Tx Push.*	0

0x0B88	PS2_TXD1	RW	[7:0]	*PS2 Tx Data.*	0
0x0B6C 0x0B8C	PS2_RXD0 PS2_RXD1	RW	[8]	*PS2 Rx Push.*	0
		RO	[7:0]	*PS2 Rx Data.*	-

4.6.5. UART

The SOLO6010 contains two UARTs for camera pan/tilt/zoom. UART0 supports for RS485 (TxEn, RxEn) and UART1 supports for RS232C (RTS, CTS). The FIFO depth for transmitting or receiving is 8, respectively. In order to activate UART pins, refer to *GPIO[15:0] Configuration Register* [0x0B00] in Table 4-40.

Fig. 4-55 shows the timing diagram of UART frame which is transmitted or received as start, data, parity and stop. Start bit is level HIGH, and Stop bit level LOW. The number of data bit, parity bit and the number of stop bit can be configured using *UART Registers* [0x0BA0~0x0BCC]. Fig. 4-56 shows the block diagram of UART.

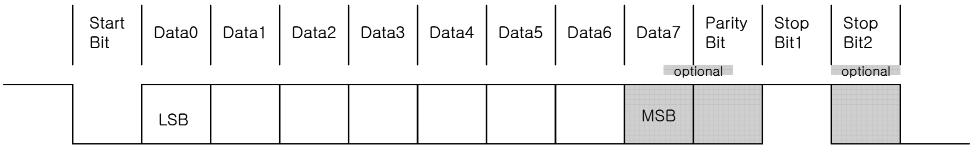


Fig. 4-55 Timing Diagram of UART Frame

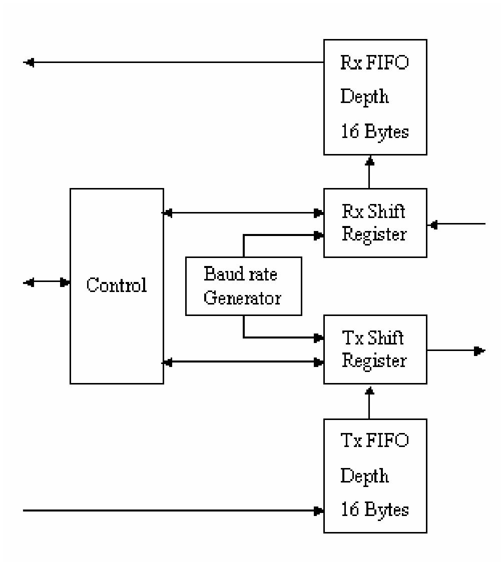


Fig. 4-56 Block Diagram of UART

Table 4-44 shows *UART Registers* [0x0BA0~0x0BCC]. Each UART supports Rx Data, Tx Data Error interrupts by activating *Rx Interrupt Enable Register* [0x0BA0, 0x0BC0], *Tx Interrupt Enable Register* [0x0BA0, 0x0BC0], and *Error Interrupt Enable Register* [0x0BA0, 0x0BC0]. Using error interrupt, Rx Parity Error, Rx Frame Error and Overflow Error can be detected. Baud Rate is configured by setting

Baud Rate1 Register [0x0BA0, 0x0BC0] and *Baud Rate0 Register* [0x0BA0, 0x0BC0].

The data will be stored on the internal buffer after writing data *Tx Data Register* [0x0BA8, 0x0BC8], and setting and resetting on *Tx Push Register* [0x0BA8, 0x0BC8]. Similarly, the data will be read from the internal buffer after reading the value of *Rx Data Register* [0x0BAC, 0x0BCC], and setting and resetting on *Rx Push Register* [0x0BAC, 0x0BCC].

Table 4-44 UART Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0BA0 0x0BC0	UART_CTRL0 UART_CTRL1	RW	[28:24]	*Divide Number*. [0x07] : System Clock = 27MHz [0x09] : System Clock = 33MHz [0x0f] : System Clock = 54MHz (Default) [0x13] : System Clock = 66MHz	0
			[20]	*CTS & RTS Enable*.	0
			[18]	*Parity Error Data Drop Enable*.	0
			[17]	*Error Interrupt Enable*.	0
			[16]	*Rx Interrupt Enable*.	0
			[15]	*Tx Interrupt Enable*.	0
			[14]	*Rx Module Enable*.	0
			[13]	*Tx Module Enable*.	0
			[12]	*Half Duplex*.	0
			[11]	*Loop Back*.	0
			[10:9]	*Baud Rate1*. The register is only effective when Baud Rate0 register is set to [110]. [00] : 4800 [01] : 2400 [10] :1200 [11] : 300	0
			[8:6]	*Baud Rate0*. [000] : 230400 [001] : 115200 [010] :57600 [011] : 38400 [100] : 19200 [101] : 9600 [110] : below 9600	0
			[5:4]	*Data Size*. [00] : 5 bit [01] : 6 bit [10] : 7 bit [11] : 8 bit	0
			[3:2]	*Stop Bit Size*. [00] : 1 bit [01] : 2 bit [10], [11]: 1.5 bit	0
			[1:0]	*Parity Mode*. [00], [01] : NONE [10] : Even [11] : Odd	0
0x0BA4 0x0BC4	UART_STA0 UART_STA1	RO	[15]	*CTS*.	-
			[14]	*Rx Busy*.	-
			[13]	*Overrun.* (Rx Buffer Full & Rx Frame Input)	-
			[12]	*Frame Error*.	-
			[11]	*Parity Error*.	-
			[10:6]	*Rx Buffer State.* The register indicates the number of remained data in Rx Buffer. Zero means empty.	-
			[5]	*Tx Line Busy*.	-

			[4:0]	*Tx Buffer State.* The register indicates the number of remained data in Tx Buffer. Zero means full.	8
0x0BA8 0x0BC8	UART_TXD0 UART_TXD1	RW	[8]	*Tx Push.*	0
		RW	[7:0]	*Tx Data.*	0
0x0BAC 0x0BCC	UART_RXD0 UART_RXD1	RW	[8]	*Rx Pop.*	0
		RO	[7:0]	*Rx Data.*	-

4.6.6. Timer

Table 4-45 shows *Timer Registers* [0x0BE0~0x0BEC]. Timer function which composes 32bit *Second Counter Register* [0x0BEC] and 20bit *Micro-Second Counter Register* [0x0BE8]. *Time Second Register* [0x0654] and *Time Micro-Second Register* [0x0658] at MPEG4 Video Encoder Status Registers refer these registers to represent time information for MPEG4 video encoder.

To support watchdog reset out using pin o_SYS_RST_OUT, *Micro-Second Clock Number Register* [0x0BE0], *Watchdog Enable Register* [0x0BE4] and *Watchdog Second Counter Register* [0x0BE4] should be set. System watchdog is very useful to the security applications, which improves reliability and stability of embedded system. System watchdog monitors if the host CPU is operated in normal and restarts the host CPU if it has a fatal error. When *Watchdog Enable Register* [0x0BE4] is enabled, *Watchdog Second Counter Register* [0x0BE4] will be decreased every one second automatically and the CPU should update the value not to be zero. If the CPU has a fatal error and cannot update the register value, pin o_SYS_RST_OUT will be LOW to reset the CPU. *Watchdog Enable Register* [0x0BE4] is set to 1 and *Watchdog Second Counter Register* [0x0BE4] is set to 255 by default. Therefore, *Watchdog Enable Register* [0x0BE4] should be updated to 0 in 255 seconds if watchdog is not used.

Table 4-45 Timer Registers

Register Address	Register Name	Type	Bit Field	Description	Initial State
0x0BE0	TIMER_CLK_NUM	RW	[8:0]	Micro-Second Clock Number. (SYS_CLK / 1MHz) -1	107
0x0BE4	TIMER_WATCHDOG	RW	[8]	*Watchdog Enable.*	1
			[7:0]	*Watchdog Second Counter.* Write : Set Read : Status	0xff
0x0BE8	TIMER_USEC	RW	[19:0]	*Micro-Second Counter.* Write : Set Read : Status	107999
0x0BEC	TIMER_SEC	RW	[31:0]	*Second Counter.* Write : Set Read : Status	0

5. Electrical Characteristics

5.1. Operating Range

Symbol	Parameter	MIN.	TYP.	MAX.	UNIT
VDD	Core Supply voltage	1.62	1.8	1.98	V
	I/O Supply Voltage	3.0	3.3	3.6	V
T_{amb}	Operating ambient temperature	0	25	70	ºC

5.2. DC Characteristics

		0 ºC	100 ºC	Conditions	
Symbol	Parameter	Minimum	Maximum	VDD	
VIL	Low level input voltage	-0.5 V	0.3 x VDD	2.7V to 3.6V	Guaranteed Input Low Voltage
VIH	High level input voltage	0.7 x VDD	VDD + 0.5V	2.7 to 3.6V	Guaranteed Input High Voltage
VOL	Low level output voltage		VSS + 0.1V	2.7V	IOL =0.8mA
VOH	High level output voltage	VDD-0.1V		2.7V	IOH =0.8mA
I_I	Input current at maximum voltage		1mA	2.7V to 3.6V	Input = 5.5V

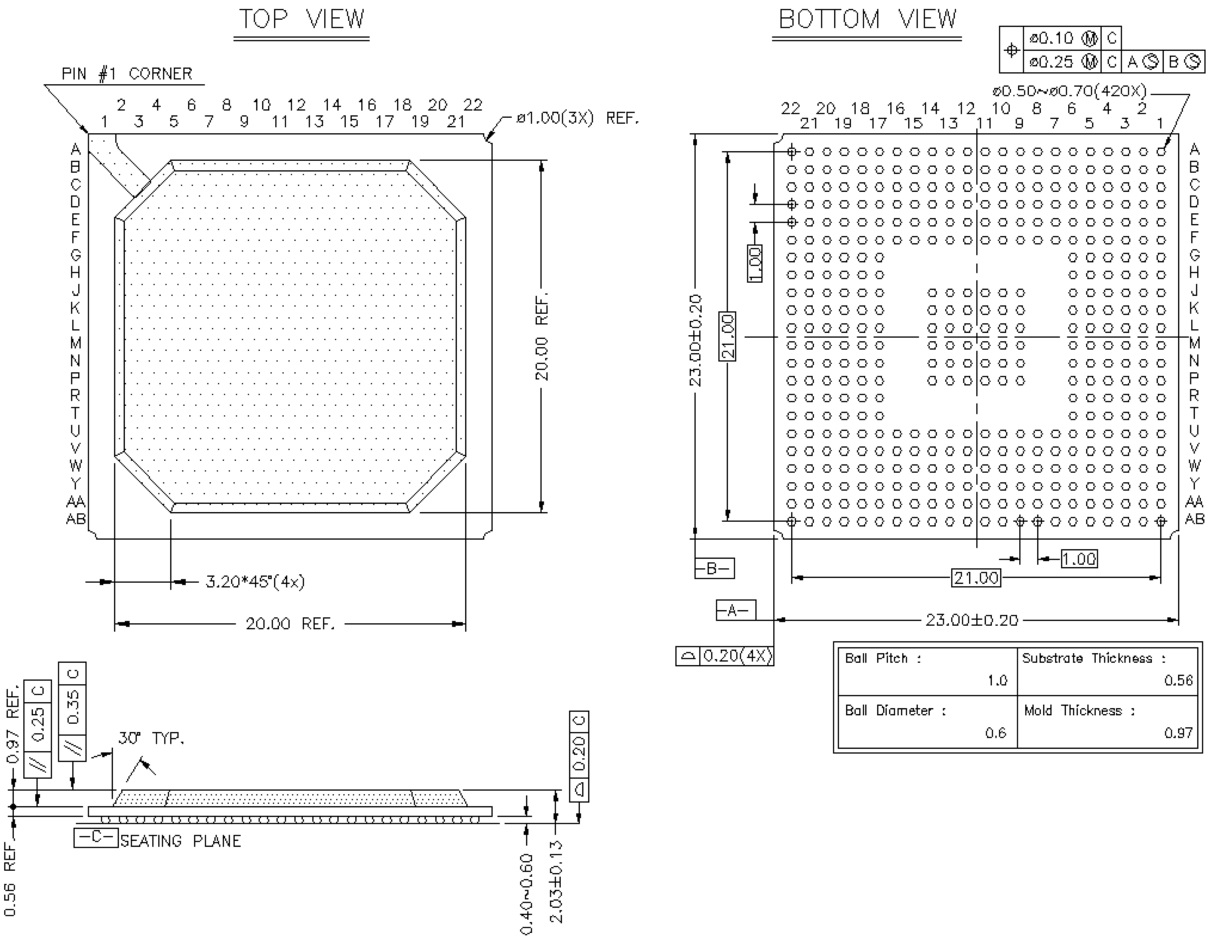
5.3. Power Dissipation

Maximum 1.28W @ (1.8V, core supply voltage), 108MHz(core speed)

Maximum about 70degree surface temperature @ 25 degree environment

Operating Condition	108 MHz			54 MHz	
	Idle	D1 quadruple speed MPEG4 Encoding	D1 quadruple speed MPEG4 Encoding & D1 dual speed MPEG Decoding	Idle	D1 dual speed Encoding
1.8V	500mA	600mA	620mA	280mA	350mA
3.3V	40mA	50mA	50mA	40mA	40mA
Surface Temperature	62degree	70degree	70degree	53degree	55degree

6. Mechanical Specifications



Ball Pitch :	Substrate Thickness :
1.0	0.56
Ball Diameter :	Mold Thickness :
0.6	0.97

7. References

[1] ISO/IEC JTC1/SC29/WG11, “Information Technology – Coding of Audio-Visual Objects – Part 2: Visual,” ISO/IEC 14496-2, International Standard, 2nd Edition, Dec. 2001.

[2] ITU-R, BT 656.

[3] ITU-T, “Extensions of Recommendation G.721 adaptive differential pulse code modulation to 24 and 40 kbit/s for digital circuit multiplication equipment application,” International Recommendation, 2nd Edition, Nov. 1988. The content of 1988 edition of ITU-T G.723 is now covered by ITU-T G.726.

[4] PCI Special Interest Group, “PCI Local Bus Specification,” Rev. 2.2, Dec. 1998.

[5] TSMC, “500MHz PLL-Based Clock Generator,” TSMC Macrocells Data Sheet PG18E1 Sep. 2002.