RTD2473AD/2483AD-CG

Multi-Function Display Controller

Specification

Version 1.00
Last updated: 2011/11/29

1. Features

General
- Embedded 3 DDC with DDC1/2B/CI
- Zoom scaling up and down
- Embedded one MCU with SPI flash controller.
- It contains 4 ADCs in key pad application
- Require only one crystal to generate all timing.
- Programmable internal low-voltage-reset (LVR)
- High resolution 6 channels PWM output, and wide range selectable PWM frequency.
- Support input format up to FHD.
- Support 27MHz/24MHz/14.318MHz crystal type

Analog RGB Input Interface
- 1 Analog input supported
- Integrated 8-bit triple-channel 210MHz ADC/PLL
- Embedded programmable Schmitt trigger of HSYNC
- Support Sync-On-Green (SOG) and various kinds of composite sync modes
- On-chip high-performance hybrid PLLs
- High resolution true 64 phase ADC PLL
- YPbPr support up to HDTV 1080p resolution

HDMI 1.4a Compliant Digital Input Interface with HDCP
- HDMI Input with embedded high speed switch
- Single link on-chip TMDS receiver up to 225MHz.
- Support long cable
- Adaptive algorithm for TMDS capability
- Data enable only mode support
- High-Bandwidth Digital Content Protection (HDCP 1.3)
- Enhanced protection of HDCP secret key
- Capable of 8-channel I2S/SPDIF output in HDMI application
- ATC Lab certification pass HDMI1.4a compliance test
- Support DVI 1.0

Embedded MCU
- Industrial standard 8051 core with external serial flash
- Low speed ADC for various application
- I2C Master hardware supported

Auto Detection /Auto Calibration
- Input format detection
- Compatibility with standard VESA mode and support user-defined mode
- Smart engine for Phase/Image position/Color calibration

Audio
- 5-band Equalizer
- AVC (Auto Volume Control)
- Output: IIS , SPDIF
- Embedded 2ch Audio DAC
- Embedded headphone amp

Scaling
- Fully programmable zoom ratios
- Independent horizontal/vertical scaling
- Advanced zoom algorithm provides high image quality
- Sharpness/Smooth filter enhancement
- Support non-linear scaling from 4:3 to 16:9 or 16:9 to 4:3

Color Processor
- True 10 bits color processing engine
- xvYCC supported
- sRGB compliance
- Advanced dithering logic for 18-bit panel color depth enhancement
- Dynamic overshoot-smear canceling engine
- Brightness and contrast control
- Programmable 10-bit gamma support
- Peaking/Coring function for video sharpness

VividColor™
- Independent color management (ICM)
- Dynamic contrast control (DCC)
- Precise color mapping (PCM)
- Active adaptive power saving(IAPS)

Output Interface
- Fully programmable display timing generator
- Flexible data pair swapping for easier system design.
- Display clock supports up to186MHz
- LVDS -output interface on single PCB
- Support 8-bit LVDS output
- Spread-Spectrum DPLL to reduce EMI
- Fixed Last Line output for perfect panel capability

Embedded OSD
- Embedded 20K SRAM dynamically stores OSD command and fonts
- Support multi-color RAM font, 1, 2 and 4-bit per pixel
- 64 color palette
- Maximum 18 window with alpha-blending/
- gradient / gradient target color / gradient reversed color/ dynamic fade-in/fade-out, bordering/ shadow/3D window type

- Rotary 90,180,270 degree
- Independent row shadowing/bordering
- Programmable blinking effects for each character
- OSD-made internal pattern generator for factory mode
- Support 12x18~4x18 proportional font
- Hardware decompression for OSD font
- Support OSD scrolling
- Support 2 independent font based OSD

Power Supply

- 3.3V / 1.2V power supply
- Low standby current (<4mA)

2. Ordering Information

Part No.	VGA	DVI	HDMI	DP	HDCP	Audio DAC	OD	FRC	Max. Resolution	Output	PKG
RTD2473AD-CG	Yes (210MHz)	Yes	Yes	No	Yes	Yes	No	No	1680*1050	Dual-LVDS (186MHz)	LQFP128 (green package)
RTD2483AD-CG	Yes (210MHz)	Yes	Yes	No	Yes	Yes	No	No	1920*1080	Dual-LVDS (186MHz)	LQFP128 (green package)

3. Chip Data Path Block Diagram

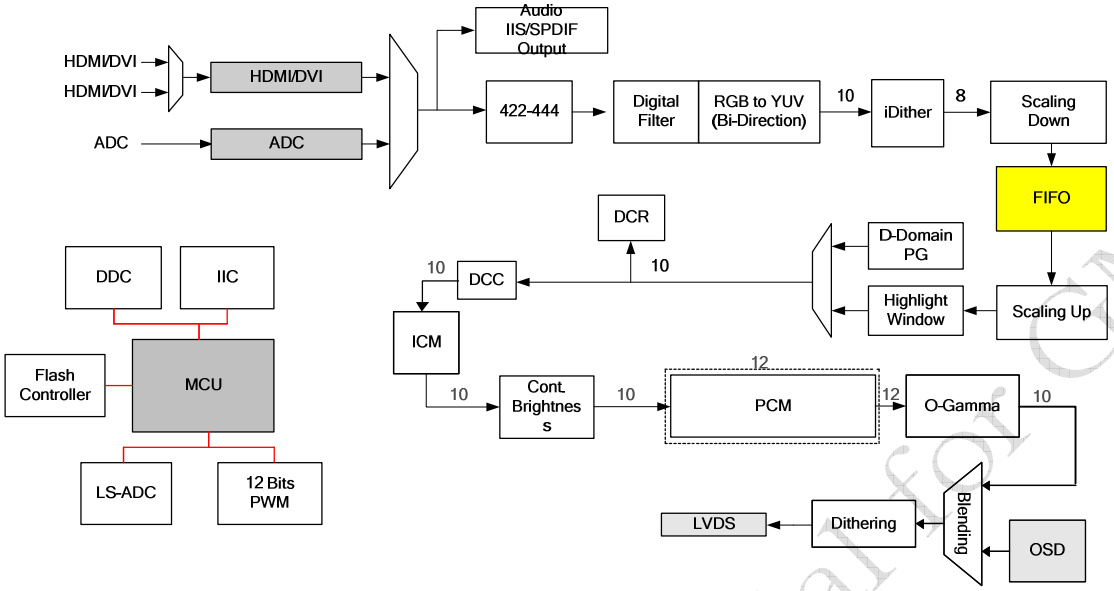
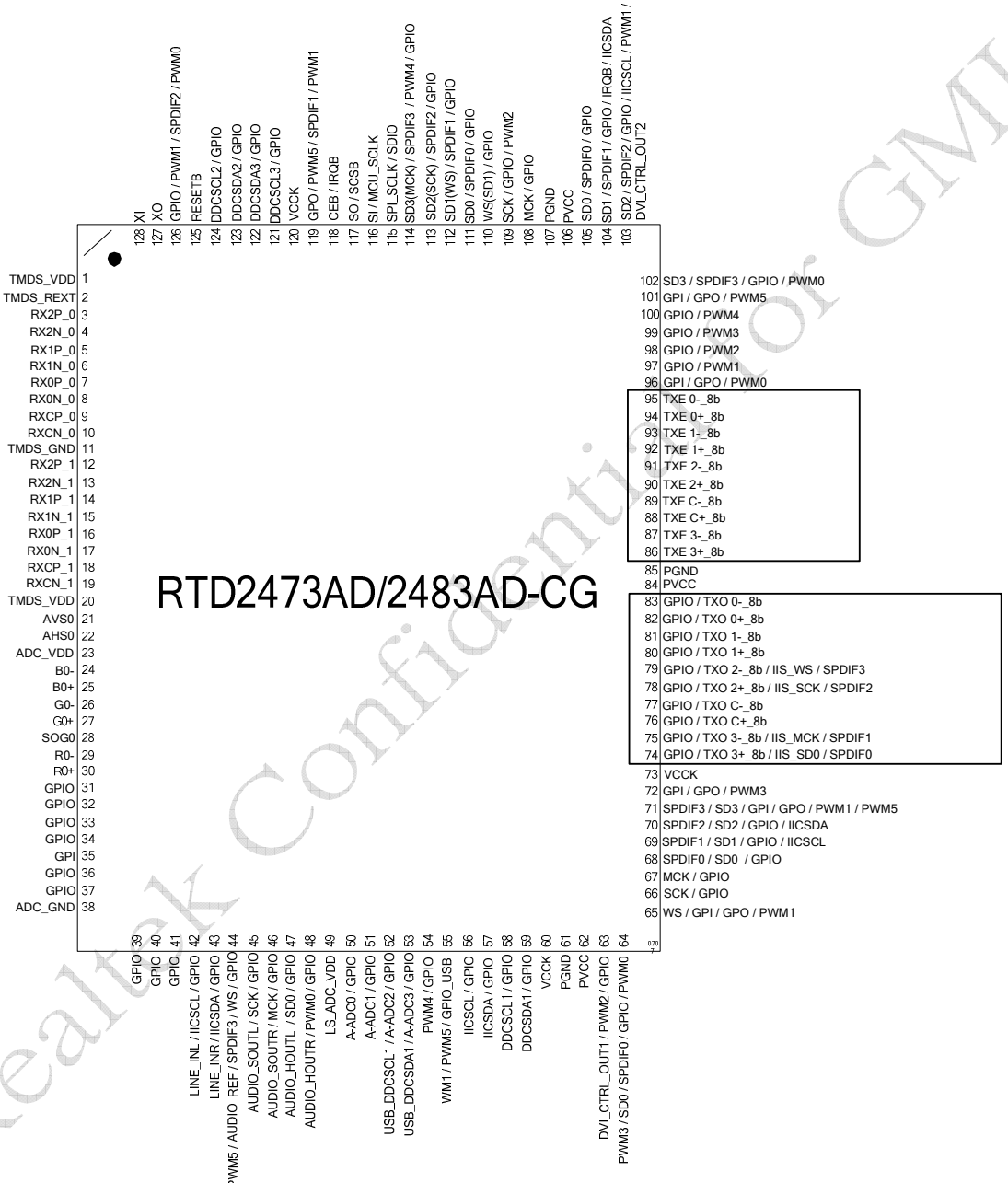


Figure1

4. Pin Diagram

128 Pin LQFP



(Pin 119 : Power on latch Pin)

(when AC Power On , Power on latch pin must be “High”)

Table of Pin Assignment

(I/O Legend: A = Analog, I = Input, O = Output, P = Power, G = Ground)

Name	I/O	Pin #	Description	Note
TMDS_VDD	AP	1	TMDS power	(3.3 V)
TMDS_REXT	AI	2	Impedance Match Reference Resistor For Scan mode,it should be pulled high Scan mode: SI[7:0] is assigned to {124~121,114~111} SO[7:0] is assigned to {110~108,105~101} SE is assigned to 100.	Ref value: 6.2 K ohm (Reference to VCC)
RX2P_0	AI	3	TMDS Differential signal Input	
RX2N_0	AI	4	TMDS Differential signal Input	
RX1P_0	AI	5	TMDS Differential signal Input	
RX1N_0	AI	6	TMDS Differential signal Input	
RX0P_0	AI	7	TMDS Differential signal Input	
RX0N_0	AI	8	TMDS Differential signal Input	
RXCP_0	AI	9	TMDS Differential signal Input	
RXCN_0	AI	10	TMDS Differential signal Input	
TMDS_GND	AG	11	TMDS ground	
RX2P_1	AI	12	TMDS Differential signal Input	
RX2N_1	AI	13	TMDS Differential signal Input	
RX1P_1	AI	14	TMDS Differential signal Input	
RX1N_1	AI	15	TMDS Differential signal Input	
RX0P_1	AI	16	TMDS Differential signal Input	
RX0N_1	AI	17	TMDS Differential signal Input	
RXCP_1	AI	18	TMDS Differential signal Input	
RXCN_1	AI	19	TMDS Differential signal Input	
TMDS_VDD	AP	20	TMDS power	(3.3 V)
AVS0	I	21	ADC vertical sync input	5V Tolerance when power off
AHS0	I	22	ADC horizontal sync input AVS0 and AHS0 could be used to select one of three scan chain. AHS0/AVS0: 2'b00: {i_chain[2:0], mcu_chain[1:0], vbi_chain[2:0]} 2'b01: d_chain 2'b10: vdec_chain Other are reserved	5V Tolerance when power off
ADC_VDD	AP	23	ADC Power	(1.2V)
B0-	AI	24	Negative BLUE analog input (Pb-)	
B0+	AI	25	Positive BLUE analog input (Pb+)	
G0-	AI	26	Negative GREEN analog input (Y-)	
G0+	AI	27	Positive GREEN analog input (Y+)	
SOG0	AI	28	Sync-On-Green	
R0-	AI	29	Negative RED analog input (Pr-)	
R0+	AI	30	Positive RED analog input (Pr+)	
GPIO	IO	31	MCU GPIO	3.3V Tolerance
GPIO	IO	32	MCU GPIO	3.3V Tolerance
GPIO	IO	33	MCU GPIO	3.3V Tolerance
GPIO	IO	34	MCU GPIO	3.3V Tolerance

GPI	IO	35	MCU GPI	3.3V Tolerance
GPIO	IO	36	MCU GPIO	3.3V Tolerance
GPIO	IO	37	MCU GPIO	3.3V Tolerance
ADC_GND	AG	38	ADC ground	
GPIO	IO	39	MCU GPIO	5V Tolerance when power off
GPIO	IO	40	MCU GPIO	5V Tolerance when power off
GPIO	IO	41	MCU GPIO	5V Tolerance when power off
GPIO/IICSCL/ LINE_INL	IO	42	MCU GPIO/ IIC BUS/LINE-IN	3.3 V tolerance
GPIO/IICSDA/ LINE_INR	IO	43	MCU GPIO/IIC BUS/LINE-IN	3.3 V tolerance
GPIO/WS/SPDIF3/ AUDIO_REF/PWM5	IO	44	MCU GPIO/IIS-WS/SPDIF3/Audio Reference Resistance/PWM	3.3 V tolerance
GPIO/SCK/ AUDIO_SOUTL	IO	45	MCU GPIO/ IIS-SCK/Speaker Output	3.3 V tolerance
GPIO/MCK/ AUDIO_SOUTR	IO	46	MCU GPIO/IIS-MCK/Speaker Output	3.3 V tolerance
GPIO/SD0/ AUDIO_HOUTL	IO	47	MCU GPIO/IIS-SD0/Audio Headphone Output	3.3 V tolerance
GPIO/PWM0/ AUDIO_HOUTR	IO	48	MCU GPIO/PWM/Audio Headphone Output	3.3 V tolerance
LS_ADC_VDD	AP	49	Low Speed ADC POWER	(3.3V)
A-ADC0/GPIO	IO	50	8-bit MCU ADC Input /MCU GPIO	5V Tolerance when power on
A-ADC1/GPIO	IO	51	8-bit MCU ADC Input/MCU GPIO	5V Tolerance when power on
A-ADC2/GPIO /USB_DDCSCL1	IO	52	8-bit MCU ADC Input /MCU GPIO /USB_DDCSCL1 When (Page 10, 0xA2[0] = 1) && (pin55 = 1), disable DDC function of pin 58, 59 and swap to pin 52, 53	5V Tolerance when power on
A-ADC3/GPIO /USB_DDCSDA1	IO	53	8-bit MCU ADC Input/MCU GPIO /USB_DDCSDA1 When (Page 10, 0xA2[0] = 1) && (pin55 = 1), disable DDC function of pin 58, 59 and swap to pin 52, 53	5V Tolerance when power on
GPIO/PWM4	IO	54	MCU GPIO/PWM	5V Tolerance when power off
GPIO_USB/PWM1/PWM 5	IO	55	MCU GPIO_USB Ctrl/PWM	5V Tolerance when power off
GPIO/IICSCL	IO	56	MCU GPIOD/IIC BUS	5V Tolerance when power off
GPIO/IICSDA	IO	57	MCU GPIO/IIC BUS	5V Tolerance when power off
DDCSCL1/GPIO	IO	58	DDC1(Open drain I/O)/MCU GPIO	5V Tolerance when power off

DDCSDA1/GPIO	IO	59	DDC1(Open drain I/O)/MCU GPIO	5V Tolerance when power off
VCCK	P	60	Digital Power	(1.2V)
PGND	G	61	Pad ground	
PVCC	P	62	Pad power	(3.3V)
GPIO/PWM2 [1] / DVI_CTRL_OUT1	IO	63	MCU GPIO/PWM	5V Tolerance when power off
SD0/SPDIF0 /GPIO/ PWM0/PWM3	IO	64	IIS-SD0/SPDIF0 /MCU GPIO/PWM	5V Tolerance when power off
WS/GPI/GPO /PWM1	IO	65	IIS-WS/MCU GPI/GPO/PWM	5V Tolerance when power off
SCK/GPIO	IO	66	IIS-SCK/MCU GPIO	5V Tolerance when power off
MCK/GPIO	IO	67	IIS-MCK/MCU GPIO	5V Tolerance when power off
SD0/SPDIF0/GPIO/	IO	68	IIS-SD0/SPDIF0 /MCU GPIO	5V Tolerance when power off
SPDIF1/SD1/GPIO/IICS CL	IO	69	SPDIF1/IIS-SD1 /MCU GPIO/IIC BUS	5V Tolerance when power off
SPDIF2/SD2 /GPIO/IICSDA	IO	70	SPDIF2 /IIS-SD2 /MCU GPI/GPO/IIC bus	5V Tolerance when power off
SPDIF3/SD3/GPI/GPO/P WM1/PWM5	IO	71	SPDIF3 /IIS-SD3 /MCU GPI/GPO/PWM	5V Tolerance when power off
GPI/GPO/PWM3	IO	72	MCU GPI/GPO/PWM	5V Tolerance when power off
VCCK	P	73	Digital Power	(1.2V)
TXO3+_8b/GPIO IIS_SD0/SPDIF0	IO	74	LVDS 8bit/MCU GPIO/IIS_SD0/SPDIF0	3.3V Tolerance
TXO3-_8b/GPIO IIS_MCK/SPDIF1	IO	75	LVDS 8bit/MCU GPIO/IIS_MCK/SPDIF1	3.3V Tolerance e
TXOC+_8b/GPIO	IO	76	LVDS 8bit/MCU GPIO	3.3V Tolerance
TXOC-_8b/GPIO	IO	77	LVDS 8bit/MCU GPIO	3.3V Tolerance
TXO2+_8b/GPIO II S_SCK/SPDIF2	IO	78	LVDS 8bit/MCU GPIO/IIS_SCK/SPDIF2	3.3V Tolerance
TXO2-_8b/GPIO IIS_WS/SPDIF3	IO	79	LVDS 8bit/MCU GPIO/IIS_WS/SPDIF3	3.3V Tolerance
TXO1+_8b/GPIO	IO	80	LVDS 8bit/MCU GPIO	3.3V Tolerance
TXO1-_8b/GPIO	IO	81	LVDS 8bit/MCU GPIO	3.3V Tolerance
TXO0+_8b/GPIO	IO	82	LVDS 8bit/MCU GPIO	3.3V Tolerance
TXO0-_8b/GPIO	IO	83	LVDS 8bit/MCU GPIO	3.3V Tolerance
PVCC	P	84	Pad power	3.3V
PGND	G	85	Pad ground	
TXE3+_8b	O	86	LVDS 8bit	3.3V Tolerance
TXE3-_8b	O	87	LVDS 8bit	3.3V Tolerance

TXEC+_8b	O	88	LVDS 8bit	3.3V Tolerance
TXEC-_8b	O	89	LVDS 8bit	3.3V Tolerance
TXE2+_8b	O	90	LVDS 8bit	3.3V Tolerance
TXE2-_8b	O	91	LVDS 8bit	3.3V Tolerance
TXE1+_8b	O	92	LVDS 8bit	3.3V Tolerance
TXE1-_8b	O	93	LVDS 8bit	3.3V Tolerance
TXE0+_8b	O	94	LVDS 8bit	3.3V Tolerance
TXE0-_8b	O	95	LVDS 8bit	3.3V Tolerance
GPI/GPO/PWM0	IO	96	MCU GPI/GPO/PWM	3.3V Tolerance
GPIO/PWM1	IO	97	MCU GPIO/PWM	3.3V Tolerance
GPIO/PWM2/	IO	98	MCU GPIO/PWM	3.3V Tolerance
GPIO/PWM3	IO	99	MCU GPIO/PWM	3.3V Tolerance
GPIO/PWM4	IO	100	MCU GPIO/PWM	5V Tolerance when power off
GPI/GPO/PWM5	IO	101	MCU GPI/GPO/PWM	5V Tolerance when power off
SD3/SPDIF3/GPIO/PWM 0	IO	102	IIS-SD3/SPDIF3 /MCU GPIO/ PWM	5V Tolerance when power off
SD2/SPDIF2/GPIO/IICS CL/PWM1/DVI_CTRL_ OUT2	IO	103	IIS-SD2/SPDIF2/MCU GPIO/IICSCL/PWM1	5V Tolerance when power off
SD1/SPDIF1/GPIO /IRQB/IICSDA	IO	104	IIS-SD1/SPDIF1/MCU GPIO/IRQ Bar/IICSDA	5V Tolerance when power off
SD0/SPDIF0/GPIO	IO	105	IIS-SD0/SPDIF0 /MCU GPIO	5V Tolerance when power off
PVCC	P	106	Pad 3.3V power	3.3V
PGND	G	107	Pad 3.3V GND	
MCK/GPIO	IO	108	IIS-MCK/MCU GPIO	5V Tolerance when power off
SCK/GPIO /PWM2	IO	109	IIS-SCK /MCU GPIO/PWM	5V Tolerance when power off
WS(SD1)/GPIO	IO	110	IIS-WS(IIS-SD1) /MCU GPIO	5V Tolerance when power off
SD0/SPDIF0/GPIO	IO	111	IIS-SD0/SPDIF0 /MCU GPIO	5V Tolerance when power off
SD1(WS)/SPDIF1/GPIO	IO	112	IIS-SD1(IIS-WS)/SPDIF1 /MCU GPIO	5V Tolerance when power off

SD2(SCK)/SPDIF2/GPIO	IO	113	IIS-SD2(IIS-SCK)/SPDIF2 /MCU GPIO	5V Tolerance when power off
SD3(MCK)/SPDIF3/ GPIO/PWM4	IO	114	IIS-SD3(IIS-MCK)/SPDIF3/ MCU GPIO/PWM	5V Tolerance when power off
SPI_SCLK/SDIO	IO	115	SPI flash serial clock /external MCU serial control I/F data in	3.3 V tolerance
SI/MCU_SCLK	IO	116	SPI flash serial data input /external MCU serial control I/F clock	3.3 V tolerance
SO/SCSB	IO	117	SPI flash serial data output /external MCU serial control I/F chip select	3.3 V tolerance
CEB/IRQB	IO	118	SPI flash chip enable bar/IRQ Bar Note:It should be pulled down to 0 v or pulled up to 3.3 v in order to designate the MCU type(Internal MCU(3.3 volts) or External MCU(0 volts)).	3.3 V tolerance
GPO/PWM5/SPDIF1/ PWM1	IO	119	MCU GPO/PWM/SPDIF1/PWM **(Power on latch Pin .)** **(when AC Power On , Power on latch Pin must be "High")**	5V Tolerance when power off
VCCK	P	120	Digital 1.2V Power	1.2V
DDCSCL3/GPIO	IO	121	DDC3(Open drain I/O)/MCU GPIO	5V Tolerance when power off
DDCSDA3/GPIO	IO	122	DDC3(Open drain I/O)/MCU GPIO	5V Tolerance when power off
DDCSDA2/GPIO	IO	123	DDC2(Open drain I/O)/MCU GPIO	5V Tolerance when power off
DDCSCL2/GPIO	IO	124	DDC2(Open drain I/O)/MCU GPIO	5V Tolerance when power off
RESETB	I	125	Chip Reset Bar	5V Tolerance when power off
GPIO/PWM1/SPDIF2/P MW0	I/O	126	MCU GPIO/PWM/SPDIF2	5V Tolerance when power on
XO	AO	127	Crystal Output	
XI	AI	128	Crystal Input	

MCU GPIO Assignment

PIN No.	MCU GPIO Name	PIN No.	MCU GPIO Name	PIN No.	MCU GPIO Name
31	PD.7	70	P1.6	122	P7.2
32	PD.6	71	P1.7	123	P7.1
33	PD.5	72	PC.2	124	P7.0
34	PD.4	74	P9.0	126	PC.0
35	PD.3	75	P9.1		
36	PD.2	76	P9.2		
37	PD.1	77	P9.3		
39	PD.0	78	P9.4		
40	PC.4	79	PA.0		
41	PB.7	80	PA.1		
42	PB.6	81	PA.2		
43	PB.5	82	PA.3		
44	PB4	83	PA.4		
45	PB.3	94	P5.0 (removed)		
46	PB.2	95	P5.1 (removed)		
47	PB.1	96	P5.2		
48	PB.0	97	P5.3		
50	P6.0	98	P5.4		
51	P6.1	99	P5.5		
52	P6.2	100	P5.6		
53	P6.3	101	P5.7		
54	P6.4	102	P7.6		
55	P6.5	103	P7.5		
56	P6.6	104	P7.4		
57	P6.7	105	P8.0		
58	P3.0/RXD(I/O)	108	P8.1/CLKO1(O)		
59	P3.1/TXD(O)	109	P3.2/INT0(I)		
63	PC.3 /INT0(I)	110	P3.3/INT1(I)		
64	P1.0/T2(I) /INT1(I)	111	P3.4/T0		
65	P1.1/T2EX(I)	112	P3.5(BS)/T1		
66	P1.2/CLKO2(O)	113	P3.6		
67	P1.3	114	P3.7		
68	P1.4	119	PC.1		
69	P1.5	121	P7.3		

5. Register Description

Global Event Flag

Register::ID_Reg				0x00	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ID	7:0	**R**	0x71	**MSB 4 bits:** 0111 product code **LSB 4 bits:** 0001 rev. code	

Register:: Host_ctrl				0x01	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config.**
XTAL_PWDN	7	R/W	0	**Power Gated XTAL (active high)**	
Reset_chk	6	R/W	0	**Reset Check** Once scalar is reset, this value will be cleared to 0.The purpose of it is to check if LVR has been triggered. It should be written to 1 ahead, then read it..LVR has been triggered if the value is 0,else LVR has not.	
Rev	5:3	---	---	**Reserved**	
PD_EN	2	R/W	1	**Power Down Mode Enable** 0: Normal 1: Enable power down mode(Default) Turn off ADC RGB Channel/ ADC Band-gap/ SOG/ DPLL/ LVDS/ADC PLL/ SYNC- PROC/ TMDS / HDMI-Audio PLL/AUTO SOY ADC/m2pll Note: For LVDS Power Control, refer to following table.	
PS_EN	1	R/W	1	**Power Saving Mode Enable** 0: Normal 1: Enable power saving mode (Default) Turn off ADC RGB channel/ DPLL/ LVDS/ ADC PLL/ m2pll When power down or power saving function is enabled, internal mcu clock is forced to crystal clock. Note: For LVDS Power Control, refer to following table.	
Sft_Reset	0	R/W	0	**Software Reset Whole Chip (Low pulse at least 8ms)** 0: Normal (Default) 1: Reset All registers are reset to default except HOST_CTRL and power-on-latch.	

- Power Down/Power Saving control only effective when LVDS Display Output is double.

DISP_TYPE CR 8C-00[1:0]	DATA_TYPE CR 28[2]	Port	Power Control
LVDS [01]	**Double** [1]	**LVDS Even**	**Power Down/Power Saving CR01 [2]/CR 01[1]**
LVDS [01]	**Double** [1]	**LVDS Odd**	
LVDS [01]	**Single** [0]	**LVDS Even**	**Power Up LVDS Even-Port CR8C-A0 [5]**
LVDS [01]	**Single** [0]	**LVDS Odd**	**Power Up LVDS Odd-Port CR8C-A0 [4]**

Register:: STATUS0	0x02

Name	Bit	R/W	Default	Description	Config.
ADCPLL_nonlock	7	R	0	**ADC_PLL Non-Lock** If the ADC_PLL non-lock occurs, this bit is set to "1".	
IVS_error	6	R	0	**Input VSYNC Error** If the input vertical sync occurs within the programmed active period, this bit is set to "1".	
IHS_error	5	R	0	**Input HSYNC Error** If the input horizontal sync occurs within the programmed active period, this bit is set to "1".	
ODD_Occur	4	R	0	**Input ODD Toggle Occur** (For internal field odd toggle, refer to CR1A[5]) If the ODD signal (From SAV/EAV or V16_ODD) toggle occurs, this bit is set to "1".	
V8HV_Occur	3	R	0	**Video8 Input Vertical/Horizontal Sync Occurs** If the YUV input V or H sync edge occurs, this bit is set to "1".	
ADCHV_Occur	2	R	0	**ADC Input Vertical/Horizontal Sync Occurs** Input V or H sync edge occurs; this bit is set to "1".	
Buffer_Ovf1	1	R	0	**Input Overflow Status** (Frame Sync Mode) *[1] If an overflow in the input data capture buffer occurs, this bit is set to "1".	
Buffer_Udf1	0	R	0	**Line Buffer Underflow Status** (Frame Sync Mode) If an underflow in the line-buffer occurs, this bit is set to "1".	

Write to clear status.

Register:: STATUS1				**0x03**	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config.**
Buffer_Ovf2	7	R	0	**Line Buffer Overflow Status** 1: Line Buffer overflow has occurred since the last status cleared	
Buffer_Udf2	6	R	0	**Line Buffer Underflow Status** 1: Line Buffer underflow has occurred since the last status cleared	
DENA_Stop	5	R	0	**DENA Stop Event Status** 1: If the DENA stop event occurred since the last status cleared	
DENA_Start	4	R	0	**DENA Start Event Status** 1: If the DENA start event occurred since the last status cleared as an interrupt source	
DVS_Start	3	R	0	**DVS Start Event Status** 1: If the DVS start event occurred since the last status cleared	
IENA_Stop	2	R	0	**IENA Stop Event Status** 1: If the IENA stop event occurred since	

*[1]Only first event of input overflow/underflow is recorded if both of them occurs.

				the last status cleared	
IENA_Start	1	R	0	**IENA Start Event Status** 1: If the IENA start event occurred since the last status cleared	
IVS_Start	0	R	0	**IVS Start Event Status** 1: If the IVS start event occurred since the last status cleared	

Write to clear status.

Register::IRQ_CTRL0				**0x04**	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config.**
IRQ_EN	7	R/W	0	**Internal IRQ Enable: (Global)** 0: Disable these interrupt. 1: Enable these interrupt.	
IRQ_ADCPLL	6	R/W	0	**IRQ (ADC_PLL Non-Lock)** 0: Disable the ADC_PLL non-lock error event as an interrupt source 1: Enable the ADC_PLL non-lock error event as an interrupt source	
IRQ_IHV	5	R/W	0	**IRQ (Input VSYNC/HSYNC Error) (DEN across Vsync or Hsync)** 0: Disable the Input VSYNC/HSYNC error event as an interrupt source 1: Enable the Input VSYNC/HSYNC error event as an interrupt source	
IRQ_ODD	4	R/W	0	**IRQ (Input ODD Toggle Occur) (EAV/SAV from Video8)** 0: Disable Input ODD toggle event as an interrupt source 1: Enable the Input ODD toggle event as an interrupt source	
IRQ_V8_HV	3	R/W	0	**IRQ (Video8 Input Hsync/Vertical Sync Occurs)** 0: Disable the Video8 Input Hsync or Vsync event as an interrupt source 1: Enable the Video8 Input Hsync or Vsync event as an interrupt sourc	
IRQ_ADC_HV	2	R/W	0	**IRQ (ADC Input Hsync/Vertical Sync Occurs)** 0: Disable the ADC Input Hsync or Vsync event as an interrupt source 1: Enable the ADC Input Hsync or Vsync event as an interrupt source	
IRQ_Buffer	1	R/W	0	**IRQ (Line Buffer Underflow/Overflow Status)** 0: Disable the Line Buffer underflow/overflow event as an interrupt source 1: Enable the Line Buffer underflow/overflow event as an interrupt source	
IRQ_IENA	0	R/W	0	**IRQ (Input ENA Start Event Occurred Status)** 0: Disable IENA start as interrupt source 1: Enable IENA start as interrupt source	

Register:: HDMI_STATUS0	**0x05**

Name	Bit	R/W	Default	Description	Config.
HDMI status 0	7:0	R	---	Reference to CRCB for HMDI Function (Page 2)(write 1 clear)	

Register:: HDMI_STATUS1				0x06	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config.**
HDMI status 1	7:0	R	---	Reference to CRCC for HMDI Function (Page 2)(write 1 clear)	

Register:: New_added_status0				0x07	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config.**
Wstate	7	R	---	Wait state status	
New_m_state	6	R	---	New mode state	
Change_m_happen	5	R	---	Change mode happen (it will not be triggered while VGIP active signal is low)	
Wstate_IRQ_en	4	R/W	0	IRQ enable of Wait state status 0:disable 1:enable	
New_m_state_IRQ_en	3	R/W	0	IRQ enable of New mode status 0:disable 1:enable	
Change_m_happen_IRQ_en	2	R/W	0	IRQ enable of change mode happen status 0:disable 1:enable	
Reserved	1	R	---	Reserved	
VBI_Status	0	---	---	reseved to 0	

Register:: New_added_status1				0x08	
Name	**Bits**	**R/W**	**Default**	**Description**	**Config**
Reserved	7:6	---	---	Reserved	
Reserved	5:2	---	---	Reserved	
ihvs_timeout_IRQ_en	1	R/W	0	(Input HSYNC/VSYNC overflow or Input VSYNC Time-Out) detected IRQ enable 0: disable 1: enable	
ihvs_timeout_IRQ	0	R	0	(Input HSYNC/VSYNC overflow or Input VSYNC Time-Out) detected IRQ status	Wclr_out

Power Control

Register:: Power_Ctrl0				0x09	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config.**
Powoff_EO	7	R	-	Top power off block power cut cell strong control signal flag. When all power cut strong control enable, the flag will be 0.	
Powoff_EWO	6	R	-	Top power off block power cut cell weak control signal flag. When all power cut weak control enable, the flag will be 0.	
Powoff_EI_FW	5	R/W	1	Top power off block power cut cell strong control enable by F/W. Not control power cut cell directly, decided by 0x0B[7]	
Powoff_EWI	4	R/W	1	Top power off block power cut cell weak control enable.	
Xtal clock select	3	R/W	0	X'tal clock select 0: external xtal 1: internal OSC	
Xtal power off	2	R/W	0	0: normal mode 1: disable xtal	
Top_Pwr_off_blk_iso ctrl	1	R/W	0	Top power off block isolation control 0: normal mode 1: isolation mode	
Top_pwr_off_blk_soft_rst	0	R/W	0	0: normal 1: reset	

Register:: Power_Ctrl1				0x0A	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config.**
GDIoff_EO	7	R	-	GDI power off block power cut cell strong control signal flag. When all power cut strong control enable, the flag will be 0.	
GDIoff_EWO	6	R	-	GDI power off block power cut cell weak control signal flag. When all power cut weak control enable, the flag will be 0.	
GDIoff_EI_FW	5	R/W	1	GDI power off block power cut cell strong control enable by F/W. Not control power cut cell directly, decided by 0x0B[6]	
GDIoff_EWI	4	R/W	1	GDI power off block power cut cell weak control enable.	
Reserved	3	R/W	0	Reserved	
Analog_blk_pwr_off ctrl	2	R/W	0	analog block power off control , LVDSPLL only 0: normal mode 1: isolation mode	
GDI_pwr_off_blk_iso_ctrl	1	R/W	0	GDI power off block isolation control 0: normal mode 1: isolation mode	
GDI_pwr_off_blk _soft_rst	0	R/W	0	GDI power off block soft reset 0: normal 1: reset	

Register:: Power_Ctrl2				0x0B	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config.**
Powoff_EI_sel	7	R/W	0	Top power off block power cut cell strong	

				control enable select 0: controlled by F/W 0x09[5] 1: controlled by power cut cell weak control flag, so the flow will be weak on -> strong on	
GDIoff_EI_sel	6	R/W	0	GDI power off block power cut cell strong control enable select 0: controlled by F/W 0x0A[5] 1: controlled by power cut cell weak control flag, so the flow will be weak on -> strong on	
Reserved	5:3	R/W	00	Reserved	
ldo_on	2	R	--	1: LDO controlled by Page0, 0xDF, 0xE0 0: LDO is disabled in this chip	
sync_pro_sync_sel	1	R/W	0	Select Sync processor V-sync2/H-sync2 source 0: from PAD_AVS1(pin40) / PAD_AHS1(pin39) 1: from d-domain DVS/DHS	
OSC_27M_EN	0	R/W	0	EMB_OSC 27MHz and 14.318MHz select 0: 14.318MHz 1: 27MHz	

Watch Dog

Address: 0C **WATCH_DOG_CTRL0** **Default: 00h**

Bit	Mode	Function
7	R/W	**Auto Switch When Input HSYNC/VSYNC Error** 0: Disable (Default) 1: Enable (See CR02[6] and CR02[5])
6	R/W	**Auto Switch When Input HSYNC/VSYNC Timeout or Overflow** 0: Disable (Default) 1: Enable (See CR52[4] and CR54[5:4])
5	R/W	**Auto Switch When Display VSYNC Timeout** 0: Disable (Default) 1: Enable
4	R/W	**Auto Switch When ADC-PLL Unlock** 0: Disable (Default) 1: Enable
3	R/W	**Auto Switch When Overflow or Underflow (for Frame-Sync Display)** 0: Disable (Default) 1: Enable
2	R/W	**Watch-Dog Action if Event Happened (for Display Timing)** 0: Disable (Default) 1: Free Run
1	R/W	**Watch-Dog Action if Event Happened (for Display Data)** 0: Disable (Default) 1: Background (Turn off overlay function and switch to background display simultaneously)
0	R	**Display VSYNC Timeout Flag (for CR0C[5])** 0: DVS is present 1: DVS is timeout The line number of Display HS is equal to Display Vertical Total; this bit is set to “1”. (Write to clear status).

Address: 0D **WATCH_DOG_CTRL1** **Default: 00h**

Bit	Mode	Function
7	R/W	**Auto Switch When Input HSYNC Changed** 0: Disable (Default) 1: Enable (See CR58[3])
6	R/W	**Auto Switch When Input VSYNC Changed** 0: Disable (Default) 1: Enable (See CR58[2])
5	R/W	**Wstate WD enable** 0:Disable(Default) 1:enable
4	R/W	**New_m_state** 0:Disable(Default) 1:enable
3	R/W	**Change_mode_happen** **0:Disable(Default)** **1:enable**
2:0	---	**Reserved**

Address: 0E~0F Reserved

Input Video Capture

Address: 10 **VGIP_CTRL (Video Graphic Input Control Register) Default: 00h**

Bit	Mode	Function
7	R/W	**8 bit Random Generator** 0: Disable(Default) 1: Enable
6	R/W	**Input Test Mode:** 0: Disable (Default) 1: Test data input will go through RGB channel, AVS=>IVS, AHS=>IHS, VCLK=>ICLK
5	R/W	**VGIP Double Buffer Ready** 0: Not Ready to Apply 1: Ready to Apply When the list table of **CR10[4]** is set, then enable **CR10[5]**. Finally, hardware will auto load these values into VGIP double buffer registers as the trigger event happens and clear **CR10[5]** to 0.
4	R/W	**VGIP Double Buffer Mode Enable** (Each register described below has its own double buffer) 0: Disable (Original- Write instantly by MCU write cycles) 1: Enable (Double Buffer Function Write Mode)

Register	Trigger Event
PLLPHASE(CRB3,CRB4) Add 1-clk Delay to IHS Delay (CR12[4]) HSYNC Synchronize Edge (CR12[3])	Falling edge of Ivactive
IPH_ACT_STA (CR14[2:0],CR15)	Falling edge of Ivactive
IPV_ACT_STA (CR18[2:0],CR19) IV_DV_LINES (CR40)	Falling edge of Ivactive
IVS_DELAY (for capture) (CR1C,CR1E[1])	Falling edge of Ivactive
IHS_DELAY (for capture) (CR1D, CR1E[0])	Falling edge of Ivactive

3:2	R/W	**Input Pixel Format** 00: Embedded ADC (ADC_HS)(Default) 01: Embedded TMDS 10: est Mode 11: Reserved
1	R/W	**Input Graphic/Video Mode** 0: From analog input (input captured by 'Input Capture Window') (Default) 1: From digital input (captured start by 'enable signal', but sill stored in 'capture window size')
0	R/W	**Input Sampling Run Enable** 0: No data is transferred (Default) 1: Sampling input pixels

Address: 11 **VGIP_SIGINV (Input Control Signal Inverted Register) Default: 00h**

Bit	Mode	Function
7	R/W	**Safe Mode** 0: Normal (Default) 1: Safe Mode Enable, mask 1 frame IVS of every 2 frame IVS, slow down input frame rate.
6	R/W	**IVS Sync with IHS Control** (Avoid VS bouncing) 0: Enable (Default) 1: Disable
5	R/W	**HS Signal Inverted for Field Detection** 0: Negative Edge (Default) 1: Positive Edge
4	R/W	**Input Video ODD Signal Invert Enable** 0: Not inverted (ODD = positive polarity) (Default) 1: Inverted (ODD = negative polarity)
3	R/W	**Input VS Signal Polarity Inverted** 0: Not inverted (VS = positive polarity) (Default) 1: Inverted (VS = negative polarity)

2	R/W	**Input HS Signal Polarity Inverted** 0: Not inverted (HS = positive polarity) (Default) 1: Inverted (HS = negative polarity)
1	R/W	**Input ENA Signal Polarity Inverted** 0: Not inverted (input high active) (Default) 1: Inverted (while input low active)
0	R/W	**Video Input Clock Polarity** 0: Rising edge latched (Default) 1: Falling edge latched

Address: 12 **VGIP_DELAY_CTRL** **Default: 00h**

Bit	Mode	Function
7	R	**6-Iclk-delay HS Level Latched by VS Rising Edge**
6	R	**HS Level Latched by VS Rising Edge**
5	R	**HS Level Latched by 6-Iclk-delay VS Rising Edge**
4	R/W/D	**Add One Clock Delay to IHS Delay** 0: Disable (Default) 1: Enable
3	R/W/D	**HSYNC Synchronize Edge** 0: HSYNC is synchronized by the positive edge of the input clock 1: HSYNC is synchronized by the negative edge of the input clock (HSYNC source is selected by CR48[0] and then synchronized)
2	R/W	**VSYNC Synchronize Edge** 0: Latch VS by the negative edge of input HSYNC (Default) 1: Latch VS by the positive edge of input HSYNC
1:0	R/W	**Video Input Clock Delay Control:** 00: Normal (Default) 01: 1ns delay 10: 2ns delay 11: 3ns delay

Address: 13 **VGIP_ODD_CTRL (Video Graphic Input ODD Control Register)** **Default: 00h**

Bit	Mode	Function
7	R/W	**ODD Inversion for ODD-Controlled-IVS-Delay** 0: Not Invert (Default) 1: Invert
6	R/W	**ODD-Controlled-IVS-Delay One-Line Enable** 0: Disable (Default) 1: Enable (Both for Auto and Capture)
5	R/W	**Safe Mode ODD Inversion** 0: Not inverted (Default) 1: Inverted
4	R/W	**Force ODD Toggle Enable (Without ODD/EVEN Toggle Select in Safe Mode)** 0: Disable (Default) 1: Enable
3	R/W	**422 to 444 Conversion by duplicate style** 0: Disable(Default) 1: Enable
2	R/W	Reserved to 0
1	R/W	Reserved to 0
0	R/W	**Internal ODD Signal Selection** 0: ODD signal from EAV or SAV or HDMI (Default) 1: Internal Field Detection ODD signal by CR1A[5] (Also support under VGA, DVI input)

Input Frame Window

(All capture window setting unit is 1)

Address: 14 **IPH_ACT_STA_H (Input Horizontal Active Start)** **Default: 00h**

Bit	Mode	Function
7:4	R/W/D	**Input Video Horizontal Active Width** -- High Byte [11:8]
3:0	R/W/D	**Input Video Horizontal Active Start** -- High Byte [11:8]

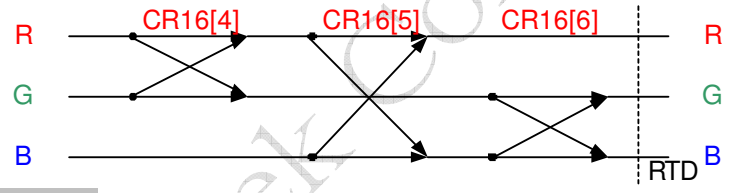
Address: 15 **IPH_ACT_STA_L (Input Horizontal Active Start Low)** **Default: 00h**

Bit	Mode	Function
7:0	R/W/D	**Input Video Horizontal Active Start** -- Low Byte [7:0]

- In analog mode, **IPH_ACT_STA** means the delay number of pixel clock from the leading edge of HS to the first pixel of each active line. Actual delay number of pixel clock **= IPH_ACT_STA(>=2) +2,**
- In digital mode, **IPH_ACT_STA** means the delay number of pixel clock from the leading edge of DE to the first pixel of each active line. Actual delay number of pixel clock = **IPH_ACT_STA(>=0)**

Address: 16 **IPH_ACT_WID_H (Input Horizontal Active Width High)** **Default: 00h**

Bit	Mode	Function
7	R/W	**Test Data Input Latch Bus MSB to LSB Swap Control:** 0: Normal (Default) 1: Swap **Test Data** MSB to LSB sequence into LSB to MSB
6	R/W	**ADC Input G/B Swap** 0: No Swap 1: Swap
5	R/W	**ADC Input R/B Swap** 0: No Swap 1: Swap
4	R/W	**ADC Input R/G Swap** 0: No Swap 1: Swap
3	R/W	**Double Clock Input** 0: Single Clock 1: Double Clock
2	R/W	**Drop next frame when APLL double buffer is triggered** 0: Disable(Default) 1: Enable You’d better wait for IVS before enable fast PLL mechanism And APLL setting if drop next frame.
1:0	R/W	**Reserved**



Address: 17 **IPH_ACT_WID_L (Input Horizontal Active Width Low)** **Default: 00h**

Bit	Mode	Function
7:0	R/W	**Input Video Horizontal Active Width** -- Low Byte [7:0]

This register defines the number of active pixel clocks to be captured.

Address: 18 **IPV_ACT_STA_H (Input Vertical Active Start High)** **Default: 00h**

Bit	Mode	Function
7:4	R/W	**Input Video Vertical Active Lines** – High Byte [11:8]
3:0	R/W/D	**Input Video Vertical Active Start** – High Byte [11:8]

Address: 19 **IPV_ACT_STA_L (Input Vertical Active Start Low)** **Default: 00h**

Bit	Mode	Function
7:0	R/W/D	**Input Video Vertical Active Start** – Low Byte [7:0]

The numbers of lines from the leading edge of selected input video VSYNC to the first line of the active window.
The value above should be larger than 1.
\

Address: 1A **IPV_ACT_LEN_H (Input Vertical Active Lines)** **Default: 00h**

Bit	Mode	Function
7	R	**SAV/EAV 2-Bit Error Happened** (Set if happened and write to clear)
6	R	**SAV/EAV 1-Bit Error Happened** (Set if happened and write to clear)
5	R	**Internal Field Detection ODD Toggle Happened** (Set if happened and write to clear) The function should be worked under no input clock
4:3	R	**Number of Input HS between 2 Input VS** (LSB bit [1:0])
2:0	R/W	Reserved

Address: 1B **IPV_ACT_LEN_L (Input Vertical Active Lines)** **Default: 00h**

Bit	Mode	Function
7:0	R/W	**Input Video Vertical Active Lines** – Low Byte [7:0]

This register defines the number of active lines to be captured.

Address: 1C **IVS_DELAY (Internal Input-VS Delay Control Register)** **Default: 00h**

Bit	Mode	Function
7:0	R/W/D	**Input VSYNC Delay for Capture[7:0]** (Counted by Input HSYNC) It's IVS delay for capture and digital filter, not for auto function

Address: 1D **IHS_DELAY (Internal Input-HS Delay Control Register)** **Default: 00h**

Bit	Mode	Function
7:0	R/W/D	**Input HSYNC Delay for Capture [7:0]** (Counted by Input Pixel Clock) It's IHS delay for capture and digital filter, not for auto function

Address: 1E **VGIP_HV_DELAY** **Default: 00h**

Bit	Mode	Function
7:6	R/W	**Input HSYNC Delay for Auto Function** (Counted by Input Pixel Clock) 00: No delay 01: 32 pixels 10: 64 pixels 11: 96 pixels
5:4	R/W	**Input VSYNC Delay for Auto Function** (Counted by Input HSYNC) 00: No delay 01: 3 line 10: 7 line 11: 15 line
3	R/W	**Select DataEnable or HSync to adjust clock phase** 0: use DataEable to adjust clock phase (Default) 1: use HSync to adjust clock phase (while input source as ADC)
2	---	Reserved
1	R/W/D	**Input VSYNC Delay for Capture[8]** (Counted by Input HSYNC)
0	R/W/D	**Input HSYNC Delay for Capture[8]** (Counted by Input Pixel Clock)

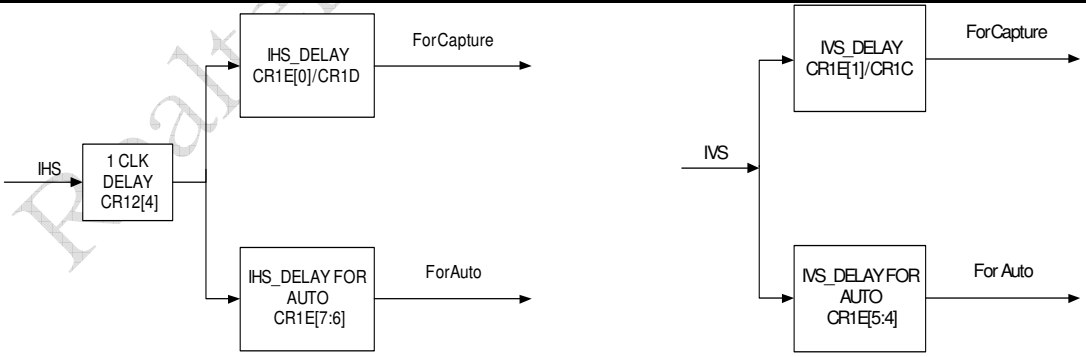


Figure 15: Input HSYNC/VSYNC Delay Path Diagram

Address: 1F **V8 Source Select & YUV422 to YUV444Conversion** **Default: 00h**

Bit	Mode	Function

7	**R/W**	**Reorder the data flow** **0: dfilter -> color_conversion -> dithering -> HSD (Default)** **1: dfilter -> dithering -> color_conversion -> HSD**
6	R/W	**Test Data source select** 0: From Pin 31~39 (excluding pin 38) 1: From Pin 41~48
5:4	R/W	**Reserved to 2'b10**
3	R/W	**Video 4:2:2->4:4:4 Enable before Scale-Down** 0: Disable (Default) 1: Enable (This bit should be always enable when in HDMI YUV422 mode.)
2	R/W	**Video 4:2:2->4:4:4 Mode Select** 0: Interpolation (Default) 1: Duplicate (This bit would be work only while CR1F[3] is enable)
1	R/W	**Output 444 Format** (only work in **Interpolation Mode**) 0: $Y_0U_0V_0$, Y_1 $(U_0+U_2)/2$ $(V_0+V_2)/2$, $Y_2U_2V_2$, Y_3 $(U_2+U_4)/2$ $(V_2+V_4)/2$… 1: $Y_0U_0V_1$, Y_1 $(U_0+U_2)/2$ V_1, Y_2U_2 $(V_1+V_3)/2$, Y_3 $(U_2+U_4)/2$ V_3…
0	R/W	**UV Swap (for YUV422 to YUV444)** (only work in **Interpolation Mode**) 0: Sequence 444 result: Y, U, V 1: Sequence 444 result: Y, V, U

Address: 20 **V8CLK_SEL (v8clk selection setting)** **Default: 00h**

Bit	Mode	Function
7:6	---	Reserved
5:4	R/W	**I-Domain clk divider:** 00: div 2 (Default) 01: div 4 10: div 8 11: reserved
3	---	Reserved
2:0	R/W	**I-Domain clk_phase:** 000: phase 0 (Default) 001: phase 1 010: phase 2 (not work while div2) 011: phase 3 (not work while div2) 100: phase 4 (not work while div2 & div4) 101: phase 5 (not work while div2 & div4) 110: phase 6 (not work while div2 & div4) 111: phase 7 (not work while div2 & div4)

FIFO Frequency

Address: 22 **FIFO Frequency** **Default: 50h**

Bit	Mode	Function
7	R/W	**Test Mode** 0: Disable 1: Input data of VGIP Replaced by Background Color in CR6D
6:3	R/W	**Divider of M2PLL Clock. Active when bit2 is set to 1.** **0000 : Div = 2;** **0001 : Div = 2;** **0010 : Div = 2;** **…..** **…..** **1010 : Div = 10; (Default)** **……** **1111 : Div = 15;**
2	R/W	**Internal Xtal Frequency** 0: F_{xtal} **1: F_{xtal} * M2PLL_M / M2PLL_N / Div. (refer to bit6~3)**
1:0	R/W	**FIFO Frequency** 00: M2PLL 01: ICLK 10: DCLK 11: Test CLK

Scaling Down Control

Address: 23 **SCALE_DOWN_CTRL (Scale Down Control Register)** **Default:00h**

Bit	Mode	Function
7	R/W	**Vertical scale down function mode selection:** 0: Use line interpolation mode (Default) 1: Use drop line mode **(Note: This bit is only valid while CR23[0]=1'b1.)**
6	R	**Bist for Line Buffer one & two ok** 0: Fail 1: Ok
5	R/W	**FIFO Line Buffer Bist Function Start (Auto clear to 0 when finish)** 0: Finish 1: Start
4	R/W	**Reserved**
3	R/W	**Horizontal non-linear scale down** 0: linear 1: non-linear
2	R/W	**Vertical Scale-Down Compensation** 0: Disable (Default) 1: Enable
1	R/W	**Horizontal scale down function enable:** 0: Disable scale down function (Default) 1: Enable scale down function
0	R/W	**Vertical scale down function enable:** 0: Disable scale down function (Default) 1: Enable scale down function (Note: There is a bit to select **interpolation** or **dropping** for vertical scale down at CR23[7].)

Address: 24 **Scale_Down_Access_Port Control** **Default: 00h**

Bit	Mode	Function
7	R/W	**Enable scale-down access port**
6:5	--	**Reserved to 0**
4:0	R/W	**Scale-down port address**

Address: 25-00 **V_SCALE_INIT**

Bit	Mode	Function
7:6	--	**Reserved**
5:0	R/W	**Vertical Scale Down Initial Select [5:0]**

- Scale Down Initial Point Select: for example, if the value is 43, we select the initial point is 43/64

Address: 25-01 **V_SCALE_DH (Vertical scale down factor register)**

Bit	Mode	Function
7:3	R/W	**Reserved**
2:0	R/W	**Vertical Scale Down Factor** [18:16]

Address: 25-02 **V_SCALE_DM (Vertical scale down factor register)**

Bit	Mode	Function
7:0	R/W	**Vertical Scale Down Factor** [15:8]

Address: 25-03 **V_SCALE_DL (Vertical scale down factor register)**

Bit	Mode	Function
7:0	R/W	**Vertical Scale Down Factor** [7:0]

- Registers {V_SCALE_DH, V_SCALE_DM, V_SCALE_DL} = (Yi/Ym)*(2^17).
- The largest scale down ratio is 1/4 (integer part 2 bits)
- Meanwhile, Yi = vertical input length; Ym=vertical memory write length

Address: 25-04 **H_SCALE_INIT**

Bit	Mode	Function
7:6	--	**Reserved**

5:0	R/W	**Horizontal Scale Down Initial Select [5:0]**

- Scale Down Initial Point Select: for example, if the value is 43, we select the initial point is 43/64

Address: 25-05 **H_SCALE_DH**

Bit	Mode	Function
7:0	R/W	**Horizontal Scale Down Factor** [23:16]

Address: 25-06 **H_SCALE_DM**

Bit	Mode	Function
7:0	R/W	**Horizontal Scale Down Factor** [15:8]

Address: 25-07 **H_SCALE_DL**

Bit	Mode	Function
7:0	R/W	**Horizontal Scale Down Factor** [7:0]

- For linear scale down, registers {H_SCALE_DH, HSCALE_DM, HSCALE_DL} = (Xi/Xm)*(2^20).
- Meanwhile, Xi = vertical input length; Xm=vertical memory write length

Address: 25-08 **H_SCALE_ACCH**

Bit	Mode	Function
7	--	**Reserved**
6:0	R/W	**Horizontal Scale Down Accumulated Factor** [14:8]

Address: 25-09 **H_SCALE_ACCL**

Bit	Mode	Function
7:0	R/W	**Horizontal Scale Down Accumulated Factor** [7:0]

Address: 25-0A **SD_ACC_WIDTHH**

Bit	Mode	Function
7:2	--	**Reserved**
1:0	R/W	**Horizontal Scale Down Accumulated Width** [9:8]

Address: 25-0B **SD_ACC_WIDTHL**

Bit	Mode	Function
7:0	R/W	**Horizontal Scale Down Accumulated Width** [7:0]

Address: 25-0C **SD_FLAT_WIDTHH**

Bit	Mode	Function
7:3	--	**Reserved**
2:0	R/W	**Horizontal Scale Down Flat Width** [10:8]

Address: 25-0D **SD_ FLAT _WIDTHL**

Bit	Mode	Function
7:0	R/W	**Horizontal Scale Down Flat Width** [7:0]

Address: 25-0E, 25-0F reserved

Address: 25-10 **Input Pattern Generator Ctrl 0** **Default: 8'h00**

Bit	Mode	Function
7	R/W	Pattern reset to initial value 0 : 1 frame 1 : 16 frame
6	R/W	Random generator mode **0 : x^9 + x^3 + 1** **1 : x^29+x^6+x^4+x+1 (Green, Blue, Red)**
5	R/W	Data update (RED) 0 : reference data enable(pixel base) 1: reference horizontal data enable end(line base)
4	R/W	Data update (GREEN) 0 : reference data enable 1: reference horizontal data enable end

3	R/W	Data update (BLUE) 0 : reference data enable 1: reference horizontal data enable end
2	R/W	Pattern generator mode (RED) 0 : random generator (ref. CR25-10[6] 1 : pattern generator (reg. CR25-11[2])
1	R/W	Pattern generator mode (GREEN) 0 : random generator (ref. CR25-10[6] 1 : pattern generator (reg. CR25-11[1])
0	R/W	Pattern generator mode (BLUE) 0 : random generator (ref. CR25-10[6] 1 : pattern generator (reg. CR25-11[0])

Address: 25-11 **Input Pattern Generator Ctrl 1** **Default: 8'h00**

Bit	Mode	Function
7-3	R/W	Reserved to 0
2	R/W	Pattern generator (RED) 0 : Out(n) = Out(n-1) 1: Out(n) = Out(n-1) + 1
1	R/W	Pattern generator (GREEN) 0 : Out(n) = Out(n-1) 1: Out(n) = Out(n-1) + 1
0	R/W	Pattern generator (BLUE) 0 : Out(n) = Out(n-1) 1: Out(n) = Out(n-1) + 1

Address: 25-12 **Input Pattern Generator RED Initial Value** **Default: 8'h01**

Bit	Mode	Function
7-0	R/W	RED Initial Value [7:0]

Address: 25-13 **Input Pattern Generator GREEN Initial Value** **Default: 8'h01**

Bit	Mode	Function
7-0	R/W	Green Initial Value [7:0]

Address: 25-14 **Input Pattern Generator BLUE Initial Value** **Default: 8'h01**

Bit	Mode	Function
7-0	R/W	BLUE Initial Value [7:0]

Address: 25-15 **Input Pattern Generator RED/GREEN/BLUE Initial Value** **Default: 8'h00**

Bit	Mode	Function
7-6	R/W	Reserved to 0
5-4	R/W	RED Initial Value [9:8]
3-2	R/W	GREEN Initial Value [9:8]
1-0	R/W	BLUE Initial Value [9:8]

Other Setting for Auto Function

Address: 25-1D **HSYNC_SYNC_ EDGE_CHANGED_ PHASE_START** **Default: 00h**

Bit	Mode	Function
7:6	R/W	Reserved to 0
5:0	R/W	**Phase Start For The Range Of Hsync Sync Edge Changed**

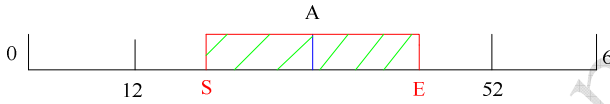
Address: 25-1E **HSYNC_SYNC_ EDGE_CHANGED_ PHASE _END** **Default: 00h**

Bit	Mode	Function
7:6	R/W	Reserved to 0
5:0	R/W	**Phase End For The Range Of Hsync Sync Edge Changed**

Address: 25-1F **HSYNC_SYNC_EDGE_ CHANGED_CTRL** **Default: 00h**

Bit	Mode	Function
7:2	R/W	**Reserved to 0**
1	R/W	**HSYNC Synchronize Edge by input clock in Select Range** 0: Negative(Default) 1: Positive When current phase is in range of start and end point, HSYNC is synchronized by selected edge, and synchronized by opposite edge for other range.
0	R/W	**Auto Change Hsync Sync Edge Enable** 0: Disable(Default) 1: Enable(Only effective when auto phase) Update the register HSYNC Synchronize Edge(CR12[3]) according to current phase before disable the function.

E.g. At phase A = 24 (12 <= A < 52), where the leading edge of input clock and HSync superpose. It is recommend that we avoid the superposition by ±12(FW Define) range from A,



thus the start and end position should be assigned as blow,
S: CR25_1D[5:0] = A – 12 = 24 – 12 = 12
E: CR25_1E[5:0] = A + 12 = 24 + 12 = 36
CR25_1F[1] = 0
CR25_1F[0] = 1

E.g. At phase A = 4 (0 <=A< 12),

the start and end position should be assigned as blow,
S: CR25_1D[5:0] = A + 12 = 4 + 12 = 16
E: CR25_1E[5:0] = 64 – 12 + A = 64 – 12 + 4 = 56
CR25_1F[1] = 1
CR25_1F[0] = 1

E.g. At phase A= 56 (52 <= A < 64),

the start and end position should be assigned as blow,
S: CR25_1D[5:0] =A + 12 – 64 = 56 + 12 – 64 = 4
E: CR25_1E[5:0] = A – 12 = 56 - 12 = 44
CR25_1F[1] = 1
CR25_1F[0] = 1

Address: 26~27 Reserved

Display Format

Address: 28 **VDIS_CTRL (Video Display Control Register)** **Default: 20h**

Bit	Mode	Function
7	R/W	**Force Display Timing Generator Enable: (Should be set when in Free-Run mode)** 0: wait for input IVS trigger 1: force enable
6	R/W	**Display Data Output Inverse Enable** 0: Disable (Default) 1: Enable (only when data bus clamp to 0)
5	R/W	**Display Output Force to Background Color** 0: Display output operates normally 1: Display output is forced to the color as selected by background color (CR6D) (Default)
4	R/W	**Display 18 bit RGB Mode Enable** 0: All individual output pixels are full 24-bit RGB (Default) 1: All individual output pixels are truncated to 18-bit RGB (LSB 2 bits = 0)
3	R/W	**Frame Sync Mode Enable** 0: Free running mode (Default) 1: Frame sync mode
2	R/W	**Display Output Double Port Enable** 0: Single port output (Default) (Not effective if CR8C-A0[1]=1'b1) 1: Double port output
1	R/W	**Display Output Run Enable** 0: DHS, DVS, DEN & DATA bus are clamped to "0" (Default) 1: Display output normal operation.
0	R/W	**Display Timing Run Enable** 0: Display Timing Generator is halted, Zoom Filter halted (Default) 1: Display Timing Generator and Zoom Filter enabled to run normally

Steps to disable output: First set CR28[1]=0, set CR28[6], then set CR28[0]=0 to disable output.

Address: 29 **VDISP_SIGINV (Display Control Signal Inverted)** **Default: 00h**

Bit	Mode	Function
7	R/W	**DHS Output Format Select (only available in Frame Sync)** 0: The first DHS after DVS is active (Default) 1: The first DHS after DVS is inactive
6	R/W	**Display Data Port Even/Odd Data Swap** 0: Disable (Default) 1: Enable
5	R/W	**Display Data Port Red/Blue Data Swap** 0: Disable (Default) 1: Enable
4	R/W	**Display Data Port MSB/LSB Data Swap** 0: Disable (Default) 1: Enable
3	R/W	**Skew Display Data Output** 0: Non-skew data output (Default) 1: Skew data output
2	R/W	**Display Vertical Sync (DVS) Output Invert Enable:** 0: Display Vertical Sync output normal active high logic (Default) 1: Display Vertical Sync output inverted logic
1	R/W	**Display Horizontal Sync (DHS) Output Invert Enable:** 0: Display Horizontal Sync output normal active high logic (Default) 1: Display Horizontal Sync output inverted logic
0	R/W	**Display Data Enable (DEN) Output Invert Enable:** 0: Display Data Enable output normal active high logic (Default) 1: Display Data Enable output inverted logic

Address: 2A **DISP_ADDR (Display Format Address Port)**

Bit	Mode	Function
7	R/W	**Display Setting Double buffer enable** **0 : Disable** **1 : Enable**

		Register: DH_TOTAL — Trigger Event: DVS Rising Register: **ODD_FIXED_LAST EVEN_FIXED_LAST** — Trigger Event: DVS Rising	
6	R/W	**Display Double Buffer Ready** 0: Not Ready to Apply 1: Ready to Apply When the list table of DISP_ADDR[7] is set, then enable DISP_ADDR[6], finally, hardware will auto load these value into RTD as the trigger event happens and clear DISP_ADDR[6] to 0.	
5:0	R/W	**Display Format Address**	

Address: 2B **DISP_DATA (Display Format Data Port)**

Bit	Mode	Function
7:0	R/W	**Display Format Data**

Address: 2B-00 **DH_TOTAL_H (Display Horizontal Total Pixels)**

Bit	Mode	Function
7	R/W	**Series four line buffer**
6:4	--	**Reserved to 0**
3:0	R/W	**Display Horizontal Total Pixel Clocks**: High Byte[11:8]

Address: 2B-01 **DH_TOTAL_L (Display Horizontal Total Pixels)**

Bit	Mode	Function
7:0	R/W	**Display Horizontal Total Pixel Clocks:** Low Byte[7:0]

Real DH_Total (Target value)= DH_Total (Register value)+ 4

Address: 2B-02 **DH_HS_END (Display Horizontal Sync End)**

Bit	Mode	Function
7:0	R/W	**Display Horizontal Sync End[7:0]:** Determines the width of DHS pulse in DCLK cycles

Address: 2B-03 **DH_BKGD_STA_H (Display Horizontal Background Start)**

Bit	Mode	Function
7:4	--	**Reserved to 0**
3:0	R/W	**Display Horizontal Background Start**: High Byte [11:8]

Address: 2B-04 **DH_BKGD_STA_L (Display Horizontal Background Start)**

Bit	Mode	Function
7:0	R/W	**Display Horizontal Background Start**: Low Byte [7:0] (Minimun value of horizontal start is 0x0D)

Determines the number of DCLK cycles from leading edge of DHS to first pixel of Background region.
Real DH_BKGD_STA (Target value)= DH_BKGD_STA (Register value)+ 10

Address: 2B-05 **DH_ACT_STA_H (Display Horizontal Active Start)**

Bit	Mode	Function
7:4	--	**Reserved to 0**
3:0	R/W	**Display Horizontal Active Region Start**: High Byte [11:8]

Address: 2B-06 **DH_ACT_STA_L (Display Horizontal Active Start)**

Bit	Mode	Function
7:0	R/W	**Display Horizontal Active Region Start**: Low Byte [7:0] (Minimun value of horizontal start is 0x0D)

Determines the number of DCLK cycles from leading edge of DHS to first pixel of Active region.
Real DH_ACT_STA (Target value)= DH_ACT_STA (Register value)+ 10

Address: 2B-07 **DH_ACT_END_H (Display Horizontal Active End)**

Bit	Mode	Function
7:4	--	**Reserved to 0**
3:0	R/W	**Display Horizontal Active End**: High Byte [11:8]

Address: 2B-08 **DH_ACT_END_L (Display Horizontal Active End)**

Bit	Mode	Function
7:0	R/W	**Display Horizontal Active End**: Low Byte [7:0]

Determines the number of DCLK cycles from leading edge of DHS to the pixel of background region.
Real DH_ACT_END (Target value)= DH_ACT_END (Register value)+ 10

Address: 2B-09 **DH_BKGD_END_H (Display Horizontal Background End)**

Bit	Mode	Function
7:4	--	**Reserved to 0**
3:0	R/W	**Display Horizontal Background end**: High Byte [11:8]

Address: 2B-0A **DH_BKGD_END_L (Display Horizontal Background End)**

Bit	Mode	Function
7:0	R/W	**Display Horizontal Background end**: Low Byte [7:0]

Real DH_BKGD_END (Target value) = DH_BKGD_END (Register value)+ 10

Address: 2B-0B **DV_TOTAL_H (Display Vertical Total Lines)**

Bit	Mode	Function
7:4	--	**Reserved to 0**
3:0	R/W	**Display Vertical Total**: High Byte [11:8]

Address: 2B-0C **DV_TOTAL_L (Display Vertical Total Lines)**

Bit	Mode	Function
7:0	R/W	**Display Vertical Total:** Low Byte [7:0]

CR2B-0B, CR2B-0C are used as watch dog reference value in ***frame sync*** mode, the event should be the line number of display HS is equal to DV Total.

Address: 2B-0D **DVS_END (Display Vertical Sync End)**

Bit	Mode	Function
7:5	**--**	**Reserved to 0**
4:0	R/W	**Display Vertical Sync End[4:0]:** Determines the duration of DVS pulse in lines

Address: 2B-0E **DV_BKGD_STA_H (Display Vertical Background Start)**

Bit	Mode	Function
7:4	--	**Reserved to 0**
3:0	R/W	**Display Vertical Background Start:** High Byte [11:8] Determines the number of lines from leading edge of DVS to first line of background region.

Address: 2B-0F **DV_BKGD_STA_L (Display Vertical Background Start)**

Bit	Mode	Function
7:0	R/W	**Display Vertical Background Start:** Low Byte [7:0]

Address: 2B-10 **DV_ACT_STA_H (Display Vertical Active Start)**

Bit	Mode	Function
7:4	--	**Reserved to 0**
3:0	R/W	**Display Vertical Active Region Start:** High Byte [11:8] Determines the number of lines from leading edge of DVS to first line of active region.

Address: 2B-11 **DV_ACT_STA_L (Display Vertical Active Start)**

Bit	Mode	Function
7:0	R/W	**Display Vertical Active Region Start:** Low Byte [7:0]

Address: 2B-12 **DV_ACT_END_H (Display Vertical Active End)**

Bit	Mode	Function
7:4	--	**Reserved to 0**
3:0	R/W	**Display Vertical Active Region End:** High Byte [11:8]

Address: 2B-13 **DV_ACT_END_L (Display Vertical Active End)**

Bit	Mode	Function
7:0	R/W	**Display Vertical Active Region End**: Low Byte [7:0]

Determine the number of lines from leading edge of DVS to the line of following background region.

Address: 2B-14 **DV_BKGD_END_H (Display Vertical Background End)**

Bit	Mode	Function
7:4	--	**Reserved to 0**
3:0	R/W	**Display Vertical Background end:** High Byte [11:8]

Address: 2B-15 **DV_BKGD_END_L (Display Vertical Background End)**

Bit	Mode	Function
7:0	R/W	**Display Vertical Background End:** Low Byte [7:0]

Determine the number of lines from leading edge of DVS to the line of start of vertical blanking.

Address: 2B-16~2B-1F Reserved

Display Fine Tune

Address: 2B-20 **DIS_TIMING (Display Clock Fine Tuning Register)** **Default: 00h**

Bit	Mode	Function
7	R/W	**Reserved to 0**
6:4	R/W	**Display Output Clock Fine Tuning Control:** 000: DCLK rising edge correspondents with output display data 001: 1ns delay 010: 2ns delay 011: 3ns delay 100: 4ns delay 101: 5ns delay 110: 6ns delay 111: 7ns delay
3	---	**Reserved**
2	---	**Reserved**
1	R/W	**DCLK Output Enable** 0: Disable 1: Enable
0	R/W	**DCLK Polarity Inverted** 0: Disable 1: Enable

Address: 2B-21 **OSD_REFERENCE__DEN** **Default: 00h**

Bit	Mode	Function
7:0	R/W	**Position Of Reference DEN for OSD[7:0]** When Page12 CRA0[7:6] = 11,this register can't be 0x00.

Address: 2B-22 **NEW_DV_CTRL** **Default: 00h**

Bit	Mode	Function
7	R/W	**New Timing Enable** 0: Disable 1: Enable
6	R/W	**Line Compensation Enable** 0: Disable 1: Enable
5	R/W	**Pixel Compensation Enable** 0: Disable 1: Enable
4	R/W	**Reserve to 0**
3:0	R/W	**DCLK_Delay[11:8]**

Address: 2B-23 **NEW_DV_DLY** **Default: 00h**

Bit	Mode	Function
7:0	R/W	**DCLK_Delay[7:0]**

When CR2B-22[7]=1, DCLK_Delay[11:0] can't be 0.

Address: 2B-24 **SSCG_NEW_Timing_Mode Setting** **Default: 00h**

Bit	Mode	Function
7	R/W	**SSCG New Timing Mode Even/Odd last line setting iverse** **0: no inverse** **1: inverse**
6	R/W	**SSCG New Timing Mode Even/Odd last line setting enable** **0: disable** **1: enable**
5:0	R/W	**Reserve**

Cyclic-Redundant-Check

Address: 2c **OP_CRC_CTRL (Output CRC Control Register)** **Default: 00h**

Bit	Mode	Function
7:6	R/W	**CRC Selector** 0x: CRC after scale-down 10: CRC after all-processing 11: reserved
5:1	--	**Reserved to 0**
0	R/W	**Output CRC Control:** 0: Stop or finish (Default) 1: Start

CRC function = X^24 + X^7 + X^2 + X + 1.

Address: 2d **OP _CRC_CHECKSUM (Output CRC Checksum)**

Bit	Mode	Function
7:0	R/W	1st read=> Output CRC-24 bit 23~16 2nd read=> Output CRC-24 bit 15~8 3rd read=> Out put CRC-24 bit 7~0

- The read pointer should be reset when 1. OP_CRC_BYTE is written 2. Output CRC Control starts.
- The read back CRC value address should be auto-increase, the sequence is shown above

FIFO Window

Address: 30 **FIFO_WIN_ADDR (FIFO Window Address Port)**

Bit	Mode	Function
7:5	--	**Reserved to 0**
4:0	R/W	**FIFO Window Address Port**

Address: 31 **FIFO_WIN_DATA (FIFO Window Data Port)**

Bit	Mode	Function
7:0	R/W	**FIFO Window Data Port**

- *Port address will increase automatically after read/write.*

Address: 31-00 **DRL_H_BSU (Display Read High Byte Before Scaling-Up)** **Default: 00h**

Bit	Mode	Function
7:4	R/W	**Display window read width before scaling up:** High Byte [11:8]
3:0	R/W	**Display window read length before scaling up:** High Byte [11:8]

Address: 31-01 **DRW_L_BSU (Display Read Width Low Byte Before Scaling-Up)** **Default: 00h**

Bit	Mode	Function
7:0	R/W	**Display window read width before scaling up:** Low Byte [7:0] the value must be even

Address: 31-02 **DRL_L_BSU (Display Read Length Low Byte Before Scaling-Up)** **Default: 00h**

Bit	Mode	Function
7:0	R/W	**Display window read length before scaling up:** Low Byte [7:0]

- The setting above should be use 2 as unit
- The setting above should be use 2 as unit

Scaling Up Function

Address: 32 **SCALE_CTRL (Scale Control Register)** **Default: 00h**

Bit	Mode	Function
7	R/W	**Video mode compensation:** 0: Disable (Default) 1: Enable
6	R/W	**Internal ODD-signal inverse for video-compensation** 0: No invert (Default) 1: invert
5	R	**Display Line Buffer Ready** 0: Busy 1: Ready
4	R/W	**Enable Full Line buffer:** 0: Disable (Default) 1: Enable
3	R/W	**Vertical Line Duplication** 0: Disable 1: Enable
2	R/W	**Horizontal pixel Duplication** 0: Disable 1: Enable
1	R/W	**Enable the Vertical Filter Function:** 0: By pass the vertical filter function block (Default) 1: Enable the vertical filter function block
0	R/W	**Enable the Horizontal Filter Function:** 0: By pass the horizontal filter function block (Default) 1: Enable the horizontal filter function block

- When using H/V duplication mode, FIFO window width set original width, but FIFO window height should be 2X the original height.

Address: 33 **SF_ACCESS_Port** **Default: 00h**

Bit	Mode	Function
7	R/W	**Enable scaling-factor access port**
6	R/W	Merge two line buffer to 1 line buffer when scale up (for resolution over line buffer size)
5	R/W	I domain Share display line buffer & display MAC when vertical scale down
4:0	R/W	**Scaling factor port address (Active when bit7 = 1)**

- When disable scaling factor access port, the access port pointer will reset to 0

Address: 34-00 **HOR_SCA_H (Horizontal Scale Factor High)**

Bit	Mode	Function
7:4	--	**Reserved**
3:0	R/W	**Bit [19:16] of horizontal scale factor**

Address: 34-01 **HOR_SCA_M (Horizontal Scale Factor Medium)**

Bit	Mode	Function
7:0	R/W	**Bit [15:8] of horizontal scale factor**

Address: 34-02 **HOR_SCA_L (Horizontal Scale Factor Low)**

Bit	Mode	Function
7:0	R/W	**Bit [7:0] of horizontal scale factor**

Address: 34-03 **VER_SCA_H (Vertical Scale Factor High)**

Bit	Mode	Function
7:4	--	**Reserved**
3:0	R/W	**Bit [19:16] of vertical scale factor**

Address: 34-04 **VER_SCA_M (Vertical Scale Factor Medium)**

Bit	Mode	Function
7:0	R/W	**Bit [15:8] of vertical scale factor**

Address: 34-05 **VER_SCA_L (Vertical Scale Factor Low)**

Bit	Mode	Function
7:0	R/W	**Bit [7:0] of vertical scale factor**

This scale-up factor includes a 20-bit fraction part to present a vertical scaled up size over the stream input. For example, for 600-line original picture scaled up to 768-line, the factor should be as follows:
(600/768) x 2^20 = 0.78125 x 2^20 = 819200 = C8000h = 0Ch, 80h, 00h.

Address: 34-06 **Horizontal Scale Factor Segment 1 Pixel** **Default: 00h**

Bit	Mode	Function
7:3	--	**Reserved**
2:0	R/W	Bit [10:8] of Scaling Factor Segment 1 pixel

Address: 34-07 **Horizontal Scale Factor Segment 1 Pixel** **Default: 00h**

Bit	Mode	Function
7:0	R/W	Bit [7:0] of Scaling Factor Segment 1 pixel

Address: 34-08 **Horizontal Scale Factor Segment 2 Pixel** **Default: 00h**

Bit	Mode	Function
7:3	--	**Reserved**
2:0	R/W	Bit [10:8] of Scaling Factor Segment 2 pixel

Address: 34-09 **Horizontal Scale Factor Segment 2 Pixel** **Default: 00h**

Bit	Mode	Function
7:0	R/W	Bit [7:0] of Scaling Factor Segment 2 pixel

Address: 34-0A **Horizontal Scale Factor Segment 3 Pixel** **Default: 00h**

Bit	Mode	Function
7:3	--	**Reserved**
2:0	R/W	Bit [10:8] of Scaling Factor Segment 3 pixel

Address: 34-0B **Horizontal Scale Factor Segment 3 Pixel** **Default: 00h**

Bit	Mode	Function
7:0	R/W	Bit [7:0] of Scaling Factor Segment 3 pixel

Address: 34-0C **Horizontal Scale Factor Delta 1** **Default: 00h**

Bit	Mode	Function
7:5	--	**Reserved**
4:0	R/W	Bit [12:8] of Horizontal Scale Factor delta 1

Address: 34-0D **Horizontal Scale Factor Delta 1** **Default: 00h**

Bit	Mode	Function
7:0	R/W	Bit [7:0] of Horizontal Scale Factor delta 1

Address: 34-0E **Horizontal Scale Factor Delta 2** **Default: 00h**

Bit	Mode	Function
7:5	--	**Reserved**
4:0	R/W	Bit [12:8] of Horizontal Scale Factor delta 2

Address: 34-0F **Horizontal Scale Factor Delta 2** **Default: 00h**

Bit	Mode	Function
7:0	R/W	Bit [7:0] of Horizontal Scale Factor delta 2

Address: 34-10 **Horizontal Filter Coefficient Initial Value** **Default: C4h**

Bit	Mode	Function
7:0	R/W	Accumulate Horizontal filter coefficient initial value

Address: 34-11 **Vertical Filter Coefficient Initial Value** **Default: C4h**

Bit	Mode	Function
7:0	R/W	Accumulate Vertical filter coefficient initial value

Address: 35 **FILTER_CTRL (Filter Control Register)** **Default: 00h**

Bit	Mode	Function
7	R/W	**Enable Filter Coefficient Access** 0: Disable (Default) 1: Enable
6	R/W	**Select H/V User Defined Filter Coefficient Table for Access Channel** 0: 1st coefficient table (Default) 1: 2nd coefficient table
5	R/W	**Select Horizontal user defined filter coefficient table** 0: 1st Horizontal Coefficient Table (Default) 1: 2nd Horizontal Coefficient Table
4	R/W	**Select Vertical user defined filter coefficient table** 0: 1st Vertical Coefficient Table (Default) 1: 2nd Vertical Coefficient Table
3:0	---	**Reserved to 0**

- The User Defined Filter Coefficient Table can be modified on-line. Only the non-active coefficient-table can be modified, and then switch it to active.

Address: 36 **FILTER_PORT (User Defined Filter Access Port)** **Default: 00h**

Bit	Mode	Function
7:0	W	**Access port for user defined filter coefficient table**

- When enable filter coefficient accessing, the first write byte is stored into the LSB(bit[7:0]) of coefficient #1 and the second byte is into MSB (bit[8:11]). Therefore, the valid write sequence for this table is c0-LSB, c0-MSB, c1-LSB, c1-MSB, c2-LSB, c2-MSB … c63-LSB & c63-MSB, totally 64 * 2 cycles. Since the 128 taps is symmetric, we need to fill the 64-coefficient sequence into table only.

Address: 37~3F Reserved

Frame Sync Fine Tune

Address: 40 **IVS2DVS_DEALY_LINES (IVS to DVS Lines)** **Default: 00h**

Bit	Mode	Function
7:0	R/W	**IVS to DVS Lines:** (Only for FrameSync Mode) The number of input HS from IVS to DVS. Should be double buffer by CR10[5:4]

Address: 41 **IV_DV_DELAY_CLK_ODD (Frame Sync Delay Fine Tuning)** **Default: 00h**

Bit	Mode	Function
7:0	R/W	**Frame Sync Mode Delay Fine Tune [7:0]** Applied to all fields when Interlaced_FS_Delay_Fine_Tuning is disabled (CR43[1] = 0) Only for odd-field when Interlaced_FS_Delay_Fine_Tuning is enabled (CR43[1] = 1)

In Frame Sync Mode , CR41[7:0] represents output VS delay fine-tuning. It delays the number of (CR41 [7:0] *16 + 16) input clocks if CR41[7:0] is not equal to 0. (No delay fine-tune if CR41[7:0] = 0)

Address: 42 **IV_DV_DELAY_CLK_EVEN (Frame Sync Delay Fine Tuning)** **Default: 00h**

Bit	Mode	Function
7:0	R/W	**Frame Sync Mode Delay Fine Tune [7:0] "00" to disable** **Only for even-field when Interlaced_FS_Delay_Fine_Tuning is enabled (CR43[1] = 1)**

Address: 43 **FS_DELAY_FINE_TUNING** **Default: 00h**

Bit	Mode	Function
7	R/W	**Enable measure last line by field** **0 : disable** **1: enable**
6	R/W	**Reference field in last line measure** **0 : 0dd** **1 : Even**
5:2	R/W	**Reserved to 0**
1	R/W	**Interlaced_FS_Delay_Fine_Tuning** 0: Disable (Default) 1: Enable
0	R/W	**Internal ODD-signal inverse for Interlaced_FS_Delay_Fine_Tuning** 0: No invert (Default) 1: Invert

Address: 44 **LAST_LINE_H** **Default: 00h**

Bit	Mode	Function
7	R/W	**Last-line-width / DV-Total Selector :** 0: CR44 [3:0] and CR45 indicate last-line width counted by display clock (Default) 1: CR44 [3:0] and CR45 indicate DHS total number between 2 DVS.
6	R/W	**DV sync with 4X clock** 0: Disable 1: Enable
5	R/W	**BIST Test Enable** 0: Disable 1: Enable (Auto clear when finish)
4	R/W	**BIST Test Result** 0: Fail 1: Ok
3:0	R	**DV Total or Last Line Width[11:8] Before Sync in Frame Sync Mode**

Address: 45 **LAST_LINE_L**

Bit	Mode	Function
7:0	R	**DV Total or Last Line Width[7:0] Before Sync in Frame Sync Mode**

Address: 46 Reserved as page selector for new sync-processor feature

Sync Processor

Address: 47 **SYNC_SELECT** **Default: 00h**

Bit	Mode	Function
7	R/W	**On line Sync Processor Power Down (Stop Crystal Clock In)** 0: Normal Run (Default) 1: Power Down
6	R/W	**HSYNC Type Detection Auto Run** 0: manual (Default) 1: automatic
5	R/W	**De-composite circuit enable** 0: Disable (Default) 1: Enable
4	R/W	**Input Sync. Source selection** 0: HS_RAW(SS/CS) (Default) 1: SOG/SOY
3	R/W	**SOG Source Selection** 0: SOG0/SOY0 (Default) 1: SOG1/SOY1 (Useless)
2	R/W	**VGA-ADC HS/VS Source** 0: 1ST HS/VS (Default) (Original HS/VS of ADC source) 1: 2ND HS/VS (HS/VS from D domain)
1	R/W	**Measured by Crystal Clock (Result shown in CR59) (in Digital Mode)** 0: Input Active Region (Vertical IDEN start to IDEN stop) (Resulst pop up at IDEN STOP) (Default) 1: Display Active Region(Vertical DEN start to DEN stop) (Resulst pop up at DEN STOP) The function should work correctly when IVS or DVS occurs and enable by CR50[4].
0	R/W	**HSYNC & VSYNC Measured Mode** 0: HS period counted by crystal clock & VS period counted by HS (Analog mode) (Default) 1: H resolution counted by input clock & V resolution counted by ENA (Digital mode) (Get the correct resolution which is triggered by enable signal, ENA)

Address: 48 **SYNC_INVERT** **Default: 00h**

Bit	Mode	Function
7	R/W	**COAST Signal Invert Enable:** 0: Not inverted (Default) 1: Inverted
6	R/W	**COAST Signal Output Enable:** 0: Disable (Default) 1: Enable
5	R/W	**HS_OUT Signal Invert Enable:** 0: Not inverted (Default) 1: Inverted
4	R/W	**HS_OUT Signal Output Enable:** 0: Disable (Default) 1: Enable
3	R/W	**CS_RAW Inverted Enable** 0: Normal (Default) 1: Invert
2	R/W	**CLAMP Signal Output Enable** 0: Disable (Default) 1: Enable
1	R/W	**HS Recovery in Coast** 0: Disable (Default) (SS/SOY) 1: Enable (CS or SOG)
0	R/W	**HSYNC Synchronize source** 0: AHS (Default) 1: Feedback HS

Address: 49 **SYNC_CTRL (SYNC Control Register)** **Default: 00h**

Bit	Mode	Function
7	R/W	**CLK Inversion to latch Feedback HS for Coast Recovery** **(Coast Recovery means HS feedback to replace input HS)**

		0: Non Inversion (Default) 1: Inversion
6	R/W	**Select HS_OUT Source Signal** 0: Bypass (SeHs)(Use in Separate Mode) 1: Select De-Composite HS out(DeHs) (In Composite mode)
5	R/W	**Select ADC_VS Source Signal (Auto switch in Auto Run Mode)** 0: VS_RAW 1: DeVS
4	R/W	**CLK Inversion to latch ADC HS for Clamp** 0: Non Inversion (Default) 1: Inversion
3	R/W	**Inversion of HSYNC to measure VSYNC** 0: Non Inversion (Default) 1: Inversion
2	R/W	**HSYNC Measure Source(ADC_HS1)** 0: Select ADC_HS(Default) 1: Select SeHS or DeHS by CR49[6]
1:0	R/W	**Measure HSYNC/VSYNC Source Select:** 00: TMDS (Default) 01: VIDEO8 10: ADC_HS1/ADC_VS 11: CS_RAW/VS_RAW

Address: 4A **STABLE_HIGH_PERIOD_H** **Default: 00h**

Bit	Mode	Function
7	R	**Even/Odd Field of Measure Input Source (Reference CR49[1:0])**(By Line-Count Mode) 0: Even 1: Odd
6	R	**The Toggling of Polarity of Measure Input Source (Reference CR49[1:0]) Field Happened** (By Line-Count Mode) 0: No toggle 1: Toggle
5	R	**Even/Odd Field of Measure Input Source (Reference CR49[1:0])** (By VS-Position Mode) 0: Even 1: Odd
4	R	**The Toggling of Polarity of Measure Input Source (Reference CR49[1:0]) Field Happened** (By VS-Position Mode) 0: No toggle 1: Toggle
3	R/W	**Odd Detection Mode** 0: Line-Count Mode (Default) 1: VS-Position Mode
2:0	R	**Stable High Period[10:8]** Compare each line's high pulse period, if we get continuous 64 lines with the same one, the period is updated as the stable period.

Address: 4B **STABLE_HIGH_PERIOD_L**

Bit	Mode	Function
7:0	R	**Stable High Period[7:0]** Compare each line's high pulse period, if we get continuous 64 lines with the same one, the period is updated as the stable period.

Address: 4C **VSYNC_COUNTER_LEVEL_MSB** **Default: 03h**

Bit	Mode	Function
7	R	**HSYNC Type Detection Auto Run Result ready**
6:4	R	**HSYNC Type Detection Auto Run Result** 000: No Signal 001: Not Support 010: YPbPr 011: Serration Composite SYNC 100: XOR/OR-Type Composite SYNC with Equalizer

		101: XOR/OR-Type Composite SYNC without Equalizer 110: HSYNC with VS_RAW (Separate HSYNC) 111: HSYNC without VS_RAW (HSYNC only) Reference when Hsync type detection auto run result ready (CR4C[7])
3	R/W	Syncprocess Clock Select 0: Fxtal (Default) 1: Fxtal * M2PLL_M / M2PLL_N / Div. (refer to CR5D-0A[7:4])
2:0	R/W	**VSYNC counter level count [10:8] MSB** VSYNC detection counter start value.

Address: 4D **VSYNC_COUNTER_LEVEL_LSB** **Default: 00h**

Bit	Mode	Function
7:0	R/W	**VSYNC counter level count [7:0] LSB**

Address: 4E **HSYNC_TYPE_DETECTION_FLAG**

Bit	Mode	Function
7	R	**HSYNC Overflow (16-bits)**
6	R	**Stable Period Change (write clear when CR4E[6]=1 or CR4F[0]=1)**
5	R	**Stable Polarity Change (write clear when CR4E[5]=1 or CR4F[0]=1)**
4	R	**VS_RAW Edge Occurs (write clear when CR4E[4]=1 or CR4F[0]=1)** If VS_RAW edge occurs, this bit is set to "1".
3	R	**Detect Capture Window Unlock Repeated 32 Times (write clear when CR4E[3]=1 or CR4F[0]=1)**
2	R	**HSYNC with Equalization (write clear when CR4E[2]=1 or CR4F[0]=1)**
1	R	**HSYNC Polarity Change (write clear when CR4E[1]=1 or CR4F[0]=1)**
0	R	**Detect Capture Window Unlock (write clear when CR4E[0]=1 or CR4F[0]=1)**

Address: 4F **STABLE_MEASURE** **Default: 00h**

Bit	Mode	Function
7	R	**Stable Flag** 0: Period or polarity can't get continuous stable status. 1: Both polarity and period are stable.
6	R	**Stable Polarity** 0: Negative 1: Positive Compare each line's polarity; if we get continuous ~~N~~ 64 lines with the same one, the polarity is updated as the stable polarity.
5:4	R/W	**Feedback HSYNC High Period Select by ADC Clock:** 00: 32 (Default) 01: 64 10: 96 11: 128
3	R/W	**Stable Period Tolerance** 0: ±2 crystal clks (Default) 1: ±4 crystal clks
2	R/W	**VSYNC measure invert Enable** 0: Disable (Default) 1: Enable
1	R/W	**Pop Up Stable Value** 0: No Pop Up (Default) 1: Pop Up Result, (CR4A[2:0], CR4B[7:0], CR4F[6], CR50[2:0], CR51[7:0])
0	R/W	**Stable Measure Start** 0 : Stop (Default) 1 : Start

Address: 50 **Stable_Period_H** **Default: 00h**

Bit	Mode	Function
7	R	**Measure One Frame Status** 0: Auto clear to 0 after measure finished in one frame (Both H/V measure and active region measure finished in one frame whenever on-line measure (CR52[7]) opened or not. 1: Measuring Now
6	R	**CS_RAW Inverted by Auto Run Mode**

		0: Not inverted 1: Inverted
5	R/W	**HS_OUT Bypass PLL into VGIP** 0: Disable (Default) 1: Enable
4	R/W	**Active Region Measure Enable** 0: Disable (Default) 1: Enable
3	R/W	**ADC_VS Source Select in Test Mode** 0: Select ADC_VS Source in Normal Mode or Auto Mode by CR47[6] (Default) 1: Select ADC_VS Source in Test Mode (Select VS_RAW or DeVS by CR49[5])
2:0	R	**Stable Period[10:8]** Compare each line's period, if we get continuous 64 lines with the same one, the period is updated as the stable period.

Address: 51 **Stable_Period_L**

Bit	Mode	Function
7:0	R	**Stable Period[7:0]** Compare each line's period, if we get continuous 64 lines with the same one, the period is updated as the stable period.

Address: 52 **MEAS_HS_PER_H (HSYNC Period Measured Result) Default: 8'b000xxxxx**

Bit	Mode	Function
7	R/W	**Auto Measure Enable** 0: Disable (Default) 1: Enable
6	R/W	**Pop Up Period Measurement Result** 0: No Pop Up (Default) 1: Pop Up Result
5	R/W	**Start a HS & VS period / H & V resolution & polarity measurement (on line monitor)** 0: Finished/Disable (Default) 1: Enable to start a measurement, auto cleared after finished
4	R	**Over-flow bit of Input HSYNC Period Measurement** 0: No Over-flow occurred 1: Over-flow occurred
3:0	R	**Input HSYNC Period Measurement Result:** High Byte[11:8]

Address: 53 **MEAS_HS_PER_L (HSYNC Period Measured Result)**

Bit	Mode	Function
7:0	R	**Input HSYNC Period Measurement Result:** Low Byte[7:0]

- The result is expressed as the average number of crystal clocks (CR47[0]=0), or input clocks (CR47[0]=1) between 2 HSYNC.
- The result is the total number of crystal/input clocks inside 16-HSYNC periods divided by 16.
- Fractional part of measure result is stored in CR56[3:0].

Address: 54 **MEAS_VS_PER_H (VSYNC Period Measured Result)**

Bit	Mode	Function
7	R	**Input VSYNC Polarity Indicator** 0: negative polarity (high period is longer than low one) 1: positive polarity (low period is longer than high one)
6	R	**Input HSYNC Polarity Indicator** 0: negative polarity (high period is longer than low one) 1: positive polarity (low period is longer than high one)
5	R	**Time-Out bit of Input VSYNC Period Measurement (No VSYNC occurred)** 0: No Time Out 1: Time Out occurred (512 time_clk ref CR5D-18[7:6])
4	R	**Over-flow bit of Input VSYNC Period Measurement** 0: No Over-flow occurred 1: Over-flow occurred
3:0	R	**Input VSYNC Period Measurement Result:** High Byte[11:8]

Address: 55 **MEAS_VS_PER_L (VSYNC Period Measured Result)**

Bit	Mode	Function
7:0	R	**Input VSYNC Period Measurement Result:** Low Byte[7:0]

- This result is expressed in terms of input HS pulses.
- When measured digitally, the result is expressed as the number of input ENA signal within a frame.

Address: 56 **MEAS_HS&VS_HI_H (HSYNC&VSYNC High Period Measured Result)**

Bit	Mode	Function
7:4	R	**Input HSYNC High Period Measurement Result: High Byte[11:8]** (CR58[0] = 0) **Input VSYNC High Period Measurement Result: High Byte[11:8]** (CR58[0] = 1)
3:0	R	**Input HSYNC Period Measurement Fractional Result** (See CR52,53)

Address: 57 **MEAS_HS&VS_HI_L (HSYNC&VSYNC High Period Measured Result)**

Bit	Mode	Function
7:0	R	**Input HSYNC High Period Measurement Result: Low Byte[7:0]** (CR58[0] = 0) **Input VSYNC High Period Measurement Result: Low Byte[7:0]** (CR58[0] = 1)

- This result of HSYNC high-period is expressed in terms of crystal clocks. When measured digitally, the result of HSYNC high-period is expressed as the number of input clocks inside the input enable signal.
- This result of VSYNC high-period is expressed in terms of input HS pulses

Address: 58 **MEAS_HS&VS_HI_SEL (VSYNC High Period Measured Result) Default:00h**

Bit	Mode	Function
7:6	R/W	**HSYNC_MAX_DELTA** 00: Don't care (CR58[3] will never go high) 01: 4-clock 10: 8-clock 11: 16-clock
5:4	R/W	**VSYNC_MAX_DELTA** 00: Don't care (CR58[2] will never go high) 01: 2-HSYNC 10: 4-HSYNC 11: 8-HSYNC
3	R	**HSYNC_OVER_RANGE** Set to 1 if variation of HSYNC larger than HSYNC_MAX_DELTA is detected by on-line measurement (CR52[7]=1). Write to clear this flag.
2	R	**VSYNC_OVER_RANGE** Set to 1 if variation of VSYNC larger than VSYNC_MAX_DELTA is detected by on-line measurement (CR52[7]=1). Write to clear this flag.
1	R/W	**Start Measurement after Mode Detection Auto-mode** 0: Disable (Default) 1: Enable
0	R/W	**HSYNC/VSYNC High Period Measurement Result Select** 0: HSYNC 1: VSYNC (See CR56~CR57)

Address: 59 **MEAS_ACTIVE_REGION_H (Active Region Measured by CRSTL_CLK Result)**

Bit	Mode	Function
7:0	R/W	**Active Region Measured By Crystal Clock** **1st read: Measurement Result:** High Byte[23:16] **2nd read: Measurement Result:** High Byte[15:8] **3rd read: Measurement Result:** High Byte[8:0] Read pointer is auto increase, if write, the pointer is also reset to 1st result.

Address: 5A **SYNC_TEST_MISC** **Default: 00h**

Bit	Mode	Function
7	R/W	Clamp Reference Source Selection 0: Clamp source from normal HS 1: Clamp source from CS_RAW
6	R/W	**HS/ENA source swap for measurement** 0: HS for analog mode, ENA for digital mode 1: HS for digital mode, ENA for analog mode

5:4	R/W	**Active Region Measurement Option** 00: Active Data Region (first ENA rising to last ENA falling, not include low level region of ENA in every line) 01: Whole Frame (VS rising to VS rising) 10: Back Porch (VS rising to first EAN rising) 11: Front Porch (last ENA falling to VS Rising)
3	R/W	vs/vs_org select option 0: vs/vs_org 1: vs_raw/vs_raw
2:0	R	**The Number of Input HS between 2 Input VSYNC**. LSB bit [2:0] for YPbPr

Address: 5B **HS_DETECT** **Default: 00h**

Bit	Mode	Function
7	R/W	**HS detected flag, write clear.** 0: no detect result 1: HS detected. (enable by CR5D-19[7])
6	R/W	**SOG detected flag, write clear.** 0: no detect result 1: SOG detected. (enable by CR5D-19[6])
5	R/W	**VS detected flag, write clear.** 0: no detect result 1: VS detected. (enable by CR5D-19[4])
4	R/W	**Coast Signal Generate Reference Source** 0: capture window (Default) 1: Normal (DeHs)
3	R/W	**Miss DeHs In DeVS Region If Unlock Occurred Enable** 0: Disable.(Default) 1: Enable
2	R/W	**Miss DeHs In Coast Region Enable** 0: Disable.(Default) 1: Enable
1:0	R/W	**DeVS Counter Level Select** 00: Stable Period x 1/2 (Default) 01: Stable high period x 2 10: Stable high period x 4 11: Normal (Referrence CR4C[2:0]~CR4D)

Address: 5C **SYNC_PROC_PORT_ADDR** **Default: 00h**

Bit	Mode	Function
7:5	R/W	**Reserved**
4:0	R/W	**Sync Processor Access Port Address**

Address: 5D **SYNC_PROC_PORT_DATA** **Default: 00h**

Bit	Mode	Function
7:0	R/W	**Sync Processor Access Port Data**

- *Port address will increase automatically after read/write.*

Address: 5D-00 **G_CLAMP_START (Clamp Signal Output Start)** **Default: 04h**

Bit	Mode	Function
7:0	R/W	**Start of Output Clamp_L Signal Pulse for Y/G Channel[7:0]:** Determine the number of input double-pixel between the trailing edge of input HSYNC and the start of the output CLAMP signal. (High Byte [11:8] at 5D-0x0D[7:4])

Address: 5D-01 **G_CLAMP_END (Clamp Signal Output End)** **Default: 10h**

Bit	Mode	Function
7:0	R/W	**End of Output Clamp_L Signal Pulse for Y/G Channel [7:0]:** Determine the number of input double-pixel between the trailing edge of input HSYNC and the end of the output CLAMP signal.

		(High Byte [11:8] at 5D-0x0D[3:0])

Address: 5D-02 **BR_CLAMP_START (Clamp Signal Output Start)** **Default: 04h**

Bit	Mode	Function
7:0	R/W	**Start of Output Clamp_L Signal Pulse for B/Pb and R/Pr Channel [7:0]:** Determine the number of input double-pixel between the trailing edge of input HSYNC and the start of the output CLAMP signal. (High Byte [11:8] at 5D-0x0E[7:4])

Address: 5D-03 **BR_CLAMP_END (Clamp Signal Output End)** **Default: 10h**

Bit	Mode	Function
7:0	R/W	**End of Output Clamp_L Signal Pulse for B/Pb and R/Pr Channel [7:0]:** Determine the number of input double-pixel between the trailing edge of input HSYNC and the end of the output CLAMP signal. (High Byte [11:8] at 5D-0x0E[3:0])

Address: 5D-04 **CLAMP _CTRL0** **Default:00h**

Bit	Mode	Function
7	R/W	**Clamp Trigger Edge Inverse for Y/G Channel** 0: Trailing edge (Default) 1: Leading edge
6	R/W	**Clamp Trigger Edge Inverse for B/Pb and R/Pr Channel** 0: Trailing edge (Default) 1: Leading edge
5:0	R/W	**Mask Line Number before DeVS [5:0]**

Address: 5D-05 **CLAMP _CTRL1** **Default: 00h**

Bit	Mode	Function
7	R/W	**Clamp Mask Enable** 0: Disable (Default) 1: Enable
6	R/W	**Select Clamp Mask as De VS** 0: Disable 1: Enable
5:0	R/W	**Mask Line Number after DeVS [5:0]**

CR5D-04[5:0] and CR5D-05[5:0] will set number of Mask Line before/after DeVS for Clamp Mask.

Address: 5D-06 **CLAMP_CTRL2** **Default: 00h**

Bit	Mode	Function
7	R/W	**Clamp Clock Source** 0: ADC_Clock (Default) 1: Crystal Clock
6	R/W	**Clamp Counter Unit (0x5D-00 – 0x5D-03)** 0: Double Pixels (Default) 1: Single Pixel
5	R/W	**ADC1_clamp_enable** 0: Disable (Default) 1: Enable **(useless, because there is no ADC1)**
4	R/W	**ADC0_clamp_enable** 0: Disable (Default) 1: Enable
3	R/W	**ADC-3 Clamp Source** 0: Clamp-G (Default) 1: Clamp-BR
2	R/W	**ADC-2 Clamp Source** 0: Clamp-G (Default) 1: Clamp-BR
1	R/W	**ADC-1 Clamp Source** 0: Clamp-G (Default) 1: Clamp-BR
0	R/W	**ADC-0 Clamp Source**

		0: Clamp-G (Default) 1: Clamp-BR

Address: 5D-07 **COAST_CTRL** **Default: 21h**

Bit	Mode	Function
7:4	R/W	**Start of COAST before DeVS Leading Edge [3:0]**
3:0	R/W	**End of COAST after DeVS Trailing Edge [3:0]**

Address: 5D-08 **CAPTURE_WINDOW_SETTING** **Default: 04h**

Bit	Mode	Function
7	R/W	**Coast_sel** 0: de_coast (Default) 1: coast_org
6	R/W	**Capture Miss Limit during Hsync Extraction** 0: 32 (Default) 1: 16
5	R/W	**Capture Window add step as Miss Lock** 0: ±1 crystal clks (Default) 1: ±2 crystal clks
4:0	R/W	Capture Window Tolerance 5'h00: ±6 crystal clks for capture window 5'h01 ~ 5'b1F : ±1 ~ ±31 crystal clks for capture window

Address: 5D-09 **DETECTION_TOLERANCE_SETTING** **Default: 04h**

Bit	Mode	Function
7	R/W	**Clamp Signal Inverted Enable** 0: Not inverted (Default) 1: Invert
6:5	R/W	**Stable Period Tolerance Extension** 00: Use 0x4F[3] Setting (Default) 01: ±4 crystal clks 10: ±8 crystal clks 11: ±16 crystal clks
4:0	R/W	H-sync for De-composite De-bounce Length 5'h00: Disable De-bounce Function 5'h01 ~ 5'h1F : De-bounce 1 ~ 31 crystal clks for de-composite

Address: 5D-0A **DEVS_CAP_NUM_H** **Default: 90h**

Bit	Mode	Function
7:4	**R/W**	**Divider of Fxtal * M2PLL_M / M2PLL_N Clock. Enable when CR4C[3] is set to 1.** **0000 : Div = 1(Reserved)** **0001 ~ 0111 : Div 1~Div7(Reserved)** **1000 : Div = 8** **1001 : Div = 9 (Default)** **……** **1111 : Div = 15**
3:0	R	The munber of Capture window between DeVs high period: High Byte[11:8]

Address: 5D-0B **DEVS_CAP_NUM_L** **Default: 00h**

Bit	Mode	Function
7:0	R	The munber of Capture window between DeVs high period: High Byte[7:0]

Address: 5D-0C **HSYNC_GLITCH_COUNTER_CTRL** **Default: 00h**

Bit	Mode	Function
7	R/W	**Enable Hsync Glitch Counter** 0: Disable 1: Enable.(Counter results pop up to bit3~0)
6:4	R/W	Reserved to 0
3:0	R/W	**Glitch Counter Number,Write Clear By Write Bit7 To 0. (Active when bit7 enabled)** 0~15: The glitch counter number between two hs, max glitch number is 15 if glitch number greater than 15.

Address: 5D-0D **G_CLAMP_START_H (Clamp Signal Output Start/End) Default: 00h**

Bit	Mode	Function
7:4	R/W	**Start of Output Clamp Signal Pulse for Y/G Channel[11:8]:** Determine the number of input double-pixel between the trailing edge of input HSYNC and the start of the output CLAMP signal.
3:0	R/W	**End of Output Clamp Signal Pulse for Y/G Channel[11:8]:** Determine the number of input double-pixel between the trailing edge of input HSYNC and the start of the output CLAMP signal.

Address: 5D-0E **BR_CLAMP_START_H (Clamp Signal Output Start/End) Default: 00h**

Bit	Mode	Function
7:4	R/W	**Start of Output Clamp Signal Pulse for B/Pb and R/Pr Channel [11:8]:** Determine the number of input double-pixel between the trailing edge of input HSYNC and the start of the output CLAMP signal.
3:0	R/W	**End of Output Clamp Signal Pulse for B/Pb and R/Pr Channel [11:8]:** Determine the number of input double-pixel between the trailing edge of input HSYNC and the end of the output CLAMP signal.

Address: 5D-0F **HV_DELAY_CTRL Default: 05h**

Bit	Mode	Function
7	R	**Original VS level be latched by HS rising edge which N Xtal delayed.(N reference bit2:0)**
6	R	**VS level be latched by original HS rising edge.**
5	R	**VS level which N Xtal delayed be latched by original HS rising edge.(N reference bit2:0)**
4	R/W	**Delay HS Or Original HS Out Select For Measurement** 0: Original.(Default) 1: Delay HS
3	R/W	**Delay VS Or Original VS Out Select For Measurement** 0: Original.(Default) 1: Delay VS
2:0	R/W	**HS Or VS Delay Crystal (or IOSC) Clock Counter** 101: (Default 6 XTAL) 000 ~ 111: 1~8 crystal clk for HS or VS delay counter.

Highlight window

Address: 60 **Highlight Window Access Port control** **Default: 00h**

Bit	Mode	Function
7	R/W	**Enable highlight window access port**
6	R/W	**Enable highlight window**
5:4	--	**Reserved**
3:0	R/W	**Highlight-window port address (Active when bit7 = 1)**

Address: 61-00 **Highlight Window Horizontal Start**

Bit	Mode	Function
7:4	--	**Reserved**
3:0	R/W	**Highlight window horizontal start[11:8]**

Address: 61-01 **Highlight Window Horizontal Start**

Bit	Mode	Function
7:0	R/W	**Highlight window horizontal start[7:0]**

Address: 61-02 **Highlight Window Horizontal End**

Bit	Mode	Function
7:4	--	**Reserved**
3:0	R/W	**Highlight window horizontal end[11:8]**

Address: 61-03 **Highlight Window Horizontal End**

Bit	Mode	Function
7:0	R/W	**Highlight window horizontal end[7:0]**

Address: 61-04 **Highlight Window Vertical Start**

Bit	Mode	Function
7:4	--	**Reserved**
3:0	R/W	**Highlight window vertical start[11:8]**

Address: 61-05 **Highlight Window Vertical Start**

Bit	Mode	Function
7:0	R/W	**Highlight window vertical start[7:0]**

Address: 61-06 **Highlight Window Vertical End**

Bit	Mode	Function
7:4	--	**Reserved**
3:0	R/W	**Highlight window vertical end[11:8]**

Address: 61-07 **Highlight Window Vertical End**

Bit	Mode	Function
7:0	R/W	**Highlight window vertical end[7:0]**

Highlight window horizontal/vertical reference point is DEN (display background start).

Address: 61-08 **Highlight Window Border**

Bit	Mode	Function
7:4	--	**Reserved**
3:0	R/W	**Highlight window border width**

Address: 61-09 **Highlight Window Border Color**

Bit	Mode	Function
7:6	--	**Reserved**
5:0	R/W	**Highlight window border red color MSB 6bit (red color 2-bit LSB = 00)**

Address: 61-0A **Highlight Window Border Color**

Bit	Mode	Function
7:6	--	**Reserved**

5:0	R/W	**Highlight window border green color MSB 6bit (green color 2-bit LSB = 00)**

Address: 61-0B **Highlight Window Border Color**

Bit	Mode	Function
7:6	--	**Reserved**
5:0	R/W	**Highlight window border blue color MSB 6bit (blue color 2-bit LSB = 00)**

Setup of highlight:
border width=15
border start = 79
border end = 143

Address: 61-0C **Highlight Window Control 0** **Default : 00h**

Bit	Mode	Function
7:6	R/W	**Contrast / brightness application control** 00: Set A used on full region 01: Set B used inside highlight window 10: Set A used outside highlight window 11: Set A used outside highlight window, and Set B used inside highlight window

Contrast (CR62[1])	Application control	Inside window	Outside window
0	X	bypass	bypass
1	CR61-0C[7:6]=00 || CR60[6]=0	Set A	Set A
1	CR61-0C[7:6]=01 && CR60[6]=1	Set B	bypass
1	CR61-0C[7:6]=10 && CR60[6]=1	bypass	Set A
1	CR61-0C[7:6]=11 && CR60[6]=1	Set B	Set A

Brightness (CR62[0])	Application control	Inside window	Outside window
0	X	bypass	bypass
1	CR61-0C[7:6]=00 || CR60[6]=0	Set A	Set A
1	CR61-0C[7:6]=01 && CR60[6)=1	Set B	bypass
1	CR61-0C[7:6]=10 && CR60[6]=1	bypass	Set A
1	CR61-0C[7:6]=11 && CR60[6]=1	Set B	Set A

5:4	R/W	**Gamma application control** 00: gamma used on full region 01: gamma used inside window 10: gamma used outside window 11: reserved

Gamma (CR67[6])	Application control	Inside window	Outside window
0	X	bypass	bypass
1	CR61-0C[5:4]=00 || CR60[6]=0	Gamma	Gamma
1	CR61-0C[5:4]=01 && CR60[6]=1	Gamma	bypass
1	CR61-0C[5:4]=10 && CR60[6]=1	bypass	Gamma

3:2	R/W	**DCC/ICM application control** 00: DCC/ICM used on full region 01: DCC/ICM used inside window 10: DCC/ICM used outside window 11: DCC/ICM is effective in inside window, but DCC counts whole image

ICM (CRD0[7])	Application control	Inside window	Outside window

		0	X	bypass	bypass
		1	CR61-0C[3:2]=00 || CR60[6]=0	ICM	ICM
		1	CR61-0C[3:2]=01 && CR60[6]=1	ICM	bypass
		1	CR61-0C[3:2]=10 && CR60[6]=1	bypass	ICM
		1	CR61-0C[3:2]=11 && CR60[6]=1	ICM	bypass

DCC (CRC7[7])	Application control	Inside window	Outside window
0	X	bypass	bypass
1	CR61-0C[3:2]=00 || CR60[6]=0	DCC	DCC
1	CR61-0C[3:2]=01 && CR60[6]=1	DCC	bypass
1	CR61-0C[3:2]=10 && CR60[6]=1	bypass	DCC
1	CR61-0C[3:2]=11 && CR60[6]=1	DCC	bypass

1:0	R/W	Reserved

Address: 61-0D **Highlight Window Control 1** **Default : 00h**

Bit	Mode	Function
7:6	R/W	**sRGB application control** 00: sRGB used on full region 01: sRGB used inside highlight window 10: sRGB used outside highlight window 11: Reserved

sRGB (CR62[2])	Application control	Inside window	Outside window
0	X	bypass	bypass
1	CR61-0D[7:6]=00 || CR60[6]=0	sRGB	sRGB
1	CR61-0D[7:6]=01 && CR60[6]=1	sRGB	bypass
1	CR61-0D[7:6]=10 && CR60[6]=1	bypass	sRGB

5:4	R/W	**DCR/CABC application control** 00: DCR/CABC used on full region. 01: DCR/CABC used inside highlight window. 10: DCR/CABCused outside highlight window. 11: Reserved.

DCR(Page 7 CRD8[0])	Application control	Inside window	Outside window
0	X	bypass	bypass
1	CRB6-0D[5:4]=00 || CR60[6]=0	DCR/CABC	DCR/CABC
1	CRB6-0D[5:4]=01 && CR60[6]=1	DCR/CABC	bypass
1	CRB6-0D[5:4]=10 && CR60[6]=1	bypass	DCR/CABC

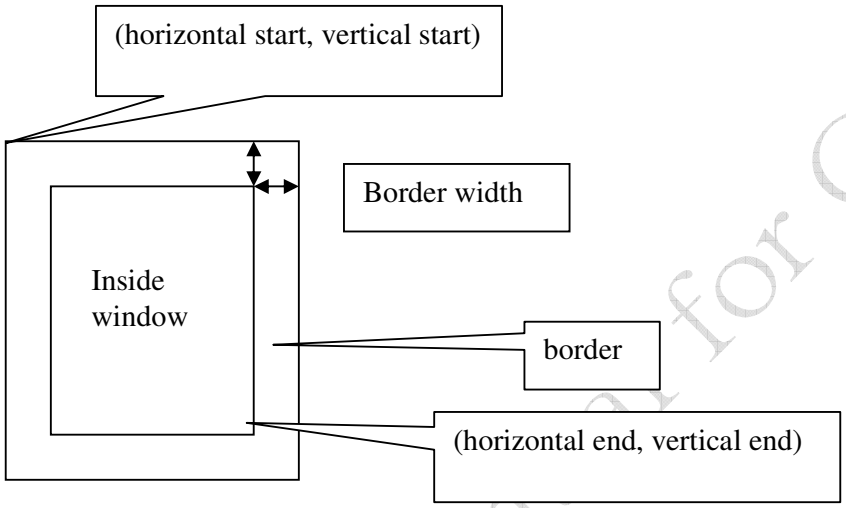
3:2	R/W	**D domain DLCTI_CTRL** 00: DLCTI used on full region. 01: DLCTI used inside highlight window. 10: DLCTI used outside highlight window. 11: Reserved.

DLCTI(Page11 CRF0[5]/CRF4[0])	Application control	Inside window	Outside window
0	X	bypass	bypass
1	CR61-0D[5:4]=00 || CR60[6]=0	DLCTI	DLCTI
1	CR61-0D[5:4]=01 && CR60[6]=1	DLCTI	bypass
1	CR61-0D[5:4]=10 && CR60[6]=1	bypass	DLCTI

1:0	R/W	**Sharpness_CTRL** 00: SHP used on full region.

01: SHP used inside highlight window.
10: SHP used outside highlight window.
11: Reserved.

DCR(Page 12 CRA0[06])	Application control	Inside window	Outside window
0	X	bypass	bypass
1	CR61-0D[5:4]=00 || CR60[6]=0	SHP	SHP
1	CR61-0D[5:4]=01 && CR60[6]=1	SHP	bypass
1	CR61-0D[5:4]=10 && CR60[6]=1	bypass	SHP



Inside window left-top point = (horizontal start + border width, vertical start + border width)
Inside window right-bottom point = (horizontal end, vertical end)
Border window left-top point = (horizontal start, vertical start)
Border window right-bottom point = (horizontal end+ border width, vertical end + border width)
Border = border window – inside window
Outside window = screen – border window

Color Processor Control

Address: 62 **COLOR_CTRL (Color Control Register)** **Default: 00h**

Bit	Mode	Function
7	R/W	**sRGB Coefficient Write Ready** 0: Not ready or cleared after finished 1: Ready to write (wait for DVS to apply)
6	R/W	**sRGB Precision** 0: Normal (Default) 1: Multiplier Coefficient Bit Left Shift
5:3	R/W	**sRGB Coefficient Write Enable** 000: Disable 001: Write R Channel (RRH,RRL,RGH,RGL,RBH,RBL) (address reset to 0 when written) 010: Write G Channel (GRH,GBL,GGH,GGL,GBH,GBL) (address reset to 0 when written) 011: Write B Channel (BRH,BRL,BGH,BGL,BBH,BBL) (address reset to 0 when written) 100: R Offset (RoffsetH, RoffsetL) (address reset to 0 when written) 101: G Offset (GoffsetH, GoffsetL) (address reset to 0 when written) 110: B Offset (BoffsetH, BoffsetL) (address reset to 0 when written)
2	R/W	**Enable sRGB Function** 0: Disable (Default) 1: Enable
1	R/W	**Enable Contrast Function:** 0: disable the coefficient (Default) 1: enable the coefficient
0	R/W	**Enable Brightness Function:** 0: disable the coefficient (Default) 1: enable the coefficient

Address: 63 **SRGB_ACCESS_PORT**

Bit	Mode	Function
7:0	W	sRGB_COEF[7:0]

- For Multiplier coefficient: 9 bit: 1 bit sign, 8 bit fractional part
- For filling multiplier coefficient, the sequence should be SIGN bit (High Byte), 8 bit fractional (Low Byte)
- For Offset Coefficient: 1 sign, 8 integer, 4 bit fractional part
- R G : 8 bit integer, 2bit fractional
- sRGB output saturation to 1023 and Clamp to 0
- sRGB Output is 10 bit

$$\begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} 1+RR & RG & RB \\ GR & 1+GG & GB \\ BR & BG & 1+BB \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} Roffset \\ Goffset \\ Boffset \end{bmatrix}$$

New SRGB(RL6093/RTD2382)	
{62[6], 68[6:5]}	Bit Definition
100	1-bit shift-left
101	2-bit shift-left (S1.7)
110	3-bit shift-left (S2.6)
111	0-bit shift-left (S0.08)
0xx	0-bit shift-left (S0.08)
Old SRGB (RTD2472D,RTD2472RD)	
{62[6], 68[6]}	Bit Definition
10	1-bit shift-left (S0.8)
11	2-bit shift-left (S1.7)
0x	0-bit shift-left (S0.08)

Contrast/Brightness Coefficient

Address: 64 **Contrast /Brightness Access Port Control** **Default: 00h**

Bit	Mode	Function
7	R/W	**Enable Contrast /Brightness access port**
6	R/W	**Contrast/Brightness and sRGB swap** **0: sRGB before Contrast/Brightness(Default)** **1: Contrast/Brightness before sRGB (for PCM)**
5:0	R/W	**Contrast /Brightness port address**

Access data port continuously will get address auto increase.

Address: 65-00 **BRI_RED_COE (Set A)**

Bit	Mode	Function
7:0	R/W	**Brightness Red Coefficient:** Valid range: -128(00h) ~ 0(80h) ~ +127(FFh)

Address: 65-01 **BRI_GRN_COE (Set A)**

Bit	Mode	Function
7:0	R/W	**Brightness Green Coefficient: Valid range:** Valid range: -128(00h) ~ 0(80h) ~ +127(FFh)

Address: 65-02 **BRI_BLU_COE (Set A)**

Bit	Mode	Function
7:0	R/W	**Brightness Blue Coefficient:** Valid range: -128(00h) ~ 0(80h) ~ +127(FFh)

Address: 65-03 **CTS_RED_COE (Set A)**

Bit	Mode	Function
7:0	R/W	**Contrast Red Coefficient:** Valid range: 0(00h) ~ 1(80h) ~ 2(FFh)

Address: 65-04 **CTS_GRN_COE (Set A)**

Bit	Mode	Function
7:0	R/W	**Contrast Green Coefficient:** Valid range: 0(00h) ~ 1(80h) ~ 2(FFh)

Address: 65-05 **CTS_BLU_COE (Set A)**

Bit	Mode	Function
7:0	R/W	**Contrast Blue Coefficient:** Valid range: 0(00h) ~ 1(80h) ~ 2(FFh)

Address: 65-06 **BRI_RED_COE (Set B)**

Bit	Mode	Function
7:0	R/W	**Brightness Red Coefficient:** Valid range: -128(00h) ~ 0(80h) ~ +127(FFh)

Address: 65-07 **BRI_GRN_COE (Set B)**

Bit	Mode	Function
7:0	R/W	**Brightness Green Coefficient: Valid range:** Valid range: -128(00h) ~ 0(80h) ~ +127(FFh)

Address: 65-08 **BRI_BLU_COE (Set B)**

Bit	Mode	Function
7:0	R/W	**Brightness Blue Coefficient:** Valid range: -128(00h) ~ 0(80h) ~ +127(FFh)

Address: 65-09 **CTS_RED_COE (Set B)**

Bit	Mode	Function
7:0	R/W	**Contrast Red Coefficient:** Valid range: 0(00h) ~ 1(80h) ~ 2(FFh)

Address: 65-0A **CTS_GRN_COE (Set B)**

Bit	Mode	Function
7:0	R/W	**Contrast Green Coefficient:** Valid range: 0(00h) ~ 1(80h) ~ 2(FFh)

Address: 65-0B **CTS_BLU_COE (Set B)**

Bit	Mode	Function
7:0	R/W	**Contrast Blue Coefficient:** Valid range: 0(00h) ~ 1(80h) ~ 2(FFh)

When highlight window is disable, coefficient set A is used.

Address: 65-0C **CTS_RED2_COE (Set A)** **Default: 80h**

Bit	Mode	Function
7:0	R/W	**Contrast Red Coefficient:** Valid range: 0(00h) ~ 1(80h) ~ 2(FFh)

Address: 65-0D **CTS_GRN2_COE (Set A)** **Default: 80h**

Bit	Mode	Function
7:0	R/W	**Contrast Green Coefficient:** Valid range: 0(00h) ~ 1(80h) ~ 2(FFh)

Address: 65-0E **CTS_BLU2_COE (Set A)** **Default: 80h**

Bit	Mode	Function
7:0	R/W	**Contrast Blue Coefficient:** Valid range: 0(00h) ~ 1(80h) ~ 2(FFh)

Address: 65-0F **CTS_RED2_COE (Set B)** **Default: 80h**

Bit	Mode	Function
7:0	R/W	**Contrast Red Coefficient:** Valid range: 0(00h) ~ 1(80h) ~ 2(FFh)

Address: 65-10 **CTS_GRN2_COE (Set B)** **Default: 80h**

Bit	Mode	Function
7:0	R/W	**Contrast Green Coefficient:** Valid range: 0(00h) ~ 1(80h) ~ 2(FFh)

Address: 65-11 **CTS_BLU2_COE (Set B)** **Default: 80h**

Bit	Mode	Function
7:0	R/W	**Contrast Blue Coefficient:** Valid range: 0(00h) ~ 1(80h) ~ 2(FFh)

Address: 65-12

Register:: CTS_Threshold (Set A)					
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
CTS_Threshold	7:0	R/W	0xFF	Threshold of GAIN1 and GAIN2 of Set A	

Address: 65-13

Register:: CTS_Threshold (Set B)					
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
CTS_Threshold	7:0	R/W	0xFF	Threshold of GAIN1 and GAIN2 of Set B	

If input value > threshold, the CTS_ADDING_VALUE will be applied.

Address: 65-14

Register:: CTS_R_GAIN_ADDING_VALUE_H (Set A)					
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
CTS_R GAIN_ADDING_VALUE_H	7:0	R/W	0x00	The adding value High byte [11:8] of GAIN2 region	

Address: 65-15

Register:: CTS_R_GAIN_ADDING_VALUE_L (Set A)					
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
CTS_R GAIN_ADDING_VALUE_L	7:0	R/W	0x00	The adding value Low byte [7:0] of GAIN2 region	

Address: 65-16

Register:: CTS_G_GAIN_ADDING_VALUE_H (Set A) 0x5B -0C					
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
CTS_G_GAIN_ADDING_VALUE_H	7:0	R/W	0x00	The green channel adding value High byte [11:8] of GAIN2 region	

Address: 65-17

Register:: CTS_G_GAIN_ADDING_VALUE_L (Set A)					
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
CTS_G_GAIN_ADDING_VALUE_L	7:0	R/W	0x00	The green channel adding value Low byte [7:0] of GAIN2 region	

Address: 65-18

Register:: CTS_B_GAIN_ADDING_VALUE_H (Set A)					
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
CTS_B_GAIN_ADDING_VALUE_H	7:0	R/W	0x00	The blue channel adding value High byte [11:8] of GAIN2 region	

Address: 65-19

Register:: CTS_B_GAIN_ADDING_VALUE_L (Set A)					
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
CTS_B_GAIN_ADDING_VALUE_L	7:0	R/W	0x00	The blue channel adding value Low byte [7:0] of GAIN2 region	

Address: 65-1A

Register:: CTS_GAIN_ADDING_VALUE_H (Set B)					
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
CTS_R GAIN_ADDING_VALUE_H	7:0	R/W	0x00	The adding value High byte [11:8] of GAIN2 region	

Address: 65-1B

Register:: CTS_GAIN_ADDING_VALUE_L (Set B)					
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
CTS_R GAIN_ADDING_VALUE_L	7:0	R/W	0x00	The adding value Low byte [7:0] of GAIN2 region	

Address: 65-1C

Register:: CTS_G_GAIN_ADDING_VALUE_H (Set B)					
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
CTS_G_GAIN_ADDING_VALUE_H	7:0	R/W	0x00	The green channel adding value High byte [11:8] of GAIN2 region	

Address: 65-1D

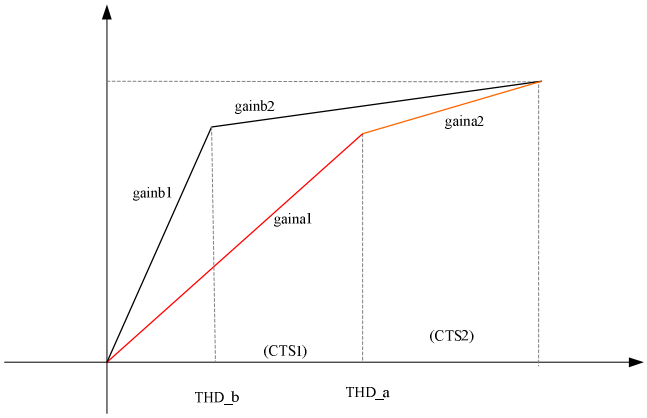
Register:: CTS_G_GAIN_ADDING_VALUE_L (Set B)					
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
CTS_G_GAIN_ADDING_VALUE_L	7:0	R/W	0x00	The green channel adding value Low byte [7:0] of GAIN2 region	

Address: 65-1E

Register:: CTS_B_GAIN_ADDING_VALUE_H (Set B)					
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
CTS_B_GAIN_ADDING_VALUE_H	7:0	R/W	0x00	The blue channel adding value High byte [11:8] of GAIN2 region	

Address: 65-1F

Register:: CTS_B_GAIN_ADDING_VALUE_L (Set B)					
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
CTS_B_GAIN_ADDING_VALUE_L	7:0	R/W	0x00	The blue channel adding value Low byte [7:0] of GAIN2 region	



InWin:
Yout= Yin*CT1 + BR
Yout= (Yin-THD_a)*CT2 + THD_a*CT1
= Yin*CT2 + BR + (CT1-CT2)*THD_a

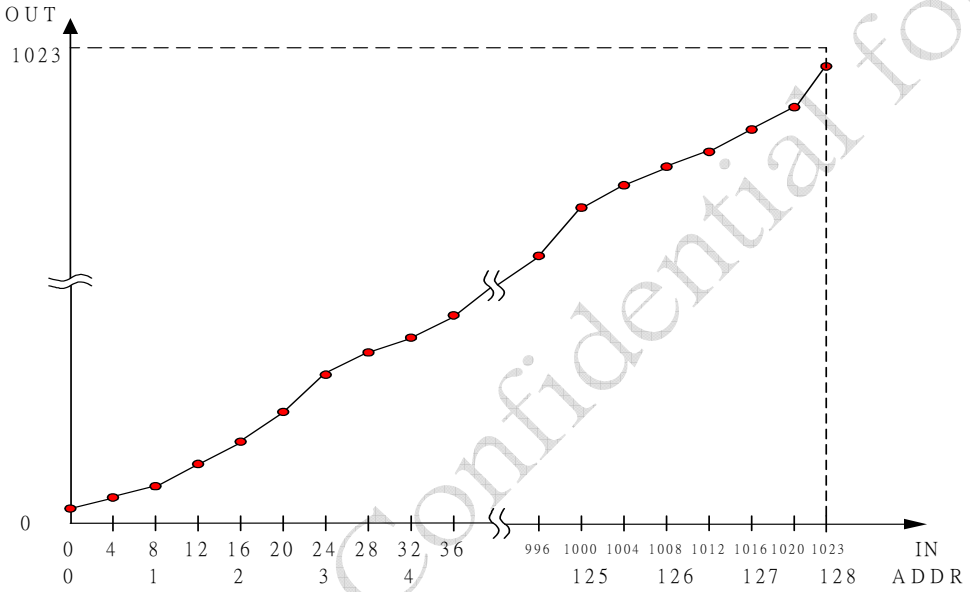
gain_adding_value a = (gaina1 – gaina2) * THD_a
u1.7 * u8 = u9.7
-> truncate lsb 4bit => u9.3
gain_adding_value b = (gainb1 – gainb2)* THD_b

Gamma Control

Address: 66 **GAMMA_PORT**

Bit	Mode	Function
7:0	R/W	**Access port for gamma correction table**

- The Gamma Table written to this port should follow the sequences as expressed below:
  {2'b0, g0[9:4]}, {g0[3:0]}, 2'b0, g4[9:8]}, {g4[7:0]}, <- addr = 0
  {2'b0, g8[9:4]}, {g8[3:0]}, 2'b0, g12[9:8]}, {g12[7:0]}, <- addr = 1
  …,
  {2'b0, g1016[9:4]}, {g1016[3:0]}, 2'b0, g1020[9:8]}, {g1020[7:0]}, <- addr = 127
  {2'b0, g1023[9:4]}, {g1023[3:0]}, 4'b0}, {8'b0} <- addr = 128
- When CR67[3] is set to 1, we can directly specify the initial address of Gamma Table in this port.
- When CR67[3] is set to 1, the value of this port is the address of Gamma Table that you are going to R/W.
- When CR67[3] is set to 0, we can read the value of Gamma Table in the following order.
  {2'b0, g_4*2n [9:4]}, {g_4*2n [3:0]}, 2'b0, g4*(2n+1)[9:8]}, {g4*(2n+1)[7:0]},
  {2'b0, g_4*(2n+2)[9:4]}, {g_4*(2n+2) [3:0]}, 2'b0, g4*(2n+3) [9:8]}, {g4*(2n+3)[7:0]},
  …,
  {2'b0, g_1023[9:4]}, {g_1023*(2n+2) [3:0]}, 4'b0}, {8'b0}



When support PCM, the table setting will be as below :
To Latch
1. {2'b00, gamma_data0[9:4]}, {gamma_data0[3:0], 2'b00, gamma_data1[9:8]}, { gamma_data1[7:0]}.
2. {2'b00, gamma_data2[9:4]}, {gamma_data2[3:0], 2'b00, gamma_data3[9:8]}, { gamma_data3[7:0]}.
3. {2'b00, gamma_data4[9:4]}, {gamma_data4[3:0], 4'h0}, {8'h00}.
4. {2'b00, gamma_data6[9:4]}, {gamma_data6[3:0], 2'b00, gamma_data5[9:8]}, { gamma_data5[7:0]}.
5. {8'h00}, {6'h00, gamma_data7[9:8] }, { gamma_data7[7:0] }.
6. {2'b00, gamma_data8[9:4]}, {gamma_data8[3:0], 2'b00, gamma_data9[9:8]}, { gamma_data9[7:0]}
7. {2'b00, gamma_data10[9:4]}, {gamma_data10[3:0], 2'b00, gamma_data11[9:8]}, { gamma_data11[7:0]}
8. {2'b00, gamma_data12[9:4]}, {gamma_data12[3:0], 2'b00, gamma_data13[9:8]}, { gamma_data13[7:0]}

To SDRAM
1. {8'h00}, {8'h00}, {8'h00}
2. {2'b00, gamma_data14[9:4]}, {gamma_data14[3:0], 2'b00, gamma_data15[9:8]}, { gamma_data15[7:0]}.
3. {2'b00, gamma_data16[9:4]}, {gamma_data16[3:0], 2'b00, gamma_data17[9:8]}, { gamma_data17[7:0]}.
……

Address: 67 **GAMMA_CTRL** **Default: 00h**

Bit	Mode	Function

7	R/W	**Enable Access Channels for Gamma Correction Coefficient:** 0: disable these channels (Default) 1: enable these channels
6	R/W	**Gamma table enable** 0: by pass (Default) 1: enable
5:4	R/W	**Color Channel of Gamma Table** 00: Red Channel (Default) 01: Green Channel 10: Blue Channel 11: Red/Green/Blue Channel (R/G/B Gamma are the same)
3	R/W	**Gamma Port Address Access Enable** 0: Normal function. (Default) 1: Gamma Port is used as specifying initial address.
2	R/W	**Write Table to SRAM or Latch Select** 0: to SRAM (Default) 1: to Latch
1:0	--	**Reserved to 0**

Address: 68 **GAMMA_BIST (Color Control Register)** **Default: 60h**

Bit	Mode	Function

7	R/W	**Test_mode**
		0: Disable, dither_out = dither_result[9:2]; // truncate to integer number (Default)
		1: Enable, dither_out = dither_result[7:0]; // propagate decimal part for test

6:5	R/W	**sRGB multiplier coefficient precision** **00: 1-bit Shift-left** **10: 3-bit Shift-left** **01: 2-bit Shift-left** **11: 0-bit shift-left (Default)**
4:3	--	**Reserved to 0**
2	R/W	**Gamma BIST_En** 0: BIST disable (Default) 1: BIST enable
1	R	**Gamma BIST_Period Progress** 0: BIST is done (Default) 1: BIST is running
0	R	**Gamma BIST Test Result (It will go low first during BIST period)** 0: SRAM Fail 1: SRAM OK

Dithering Control(For Display Domain)

Register:: DITHERING_DATA_ACCESS				0x69	
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
DITHERING_DATA_ACCESS	7:0	W	0	Refer to following description	

A. When CR6A[7:6] is 2'b01, dithering sequence table access is enabled:

- There are three set of dithering sequence table, each table contains 32 elements, s0, s1, … , s31. Each element has 2 bit to index one of 4 dithering table.
- Input data sequence is {sr3,sr2,sr1,sr0}, {sr7,sr6,sr5,sr4}, … , {sr31,sr30,sr29,sr28}, {sg3,sg2,sg1,sg0}, … , {sg31,sg30,sg29,sg28}, {sb3,sb2,sb1,sb0}, … , {sb31,sb30,sb29,sb28} for red, green and blue channel.
- R + (2R+1) * C choose sequence element, where R is Row Number / 2, and C is Column Number / 2.

B. When CR6A[7:6] is 2'b10, dithering table access is enabled:

- For dithering table access, the red, green, blue each channel has 4 dithering table, each table is 2x2 elements, and one element has 4 bit for 10B/8B, the elements should fill 0 to 3, for 10B/6B, the elements should fill 0 to 15.
- Input data sequence is [Dr00 Dr01],[Dr02,Dr03], … , [Dr30,Dr31],[Dr32,Dr33], [Dg00,Dg01],[Dg02,Dg03], … , [Dg30,Dg31],[Dg32,Dg33], [Db00,Db01],[Db02,Db03], … , [Db30,Db31],[Db32,Db33].

D00	D01	D10	D11	D20	D21	D30	D31
D02	D03	D12	D13	D22	D23	D32	D33

C. When CR6A[7:6] is 2'b11, temporal offset access is enabled:

- There are 16 element for temporal offset table, t0, t1, … , t15. Each element has 2 bit to index one of 4 temporal offset.
- Input data sequence is {t3,t2,t1,t0}, {t7,t6,t5,t4}, {t11,t10,t9,t8}, {t15,t14,t13,t12}.

Register:: DITHERING_CTRL1				0x6A	
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
Dither_Access	7:6	R/W	0	**Enable Access Control** 00: disable (Default) 01: enable access dithering sequence table 10: enable access dithering table 11: enable access temporal offset	
Dither_en	5	R/W	0	**Enable Dithering Function** 0: disable (Default) 1: enable	
Dither_temp	4	R/W	0	**Temporal Dithering** 0: Disable (Default) 1: Enable	
Dither_table	3	R/W	0	**Dithering Table Value Sign** 0: unsigned 1: signed (2's complement)	
Dither_mode	2	R/W	0	**Dithering Mode** 0: New (Default) 1: Old	
Dither_V_Fram_M	1	R/W	0	**Vertical Frame Modulation** 0: Disable (Default) 1: Enable	
Dither_VH_Fram_M	0	R/W	0	**Horizontal Frame Modulation** 0: Disable (Default) 1: Enable	

Dithering Control (Display Domain)

Register:: DITHERING_CTRL2				0x6B	
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
DITHER_THD_SEL_EN	7	R/W	0x0	**Dither RGB level threshold select enable** 0: disable 1: enable	
DITHER_THD_SEL	6:4	R/W	0x00	**Dither RGB level threshold select (0x66)**	

				000: {6'h0, R_THD[9:8]} 001: R_THD[7:0] 010: {6'h0, G_THD[9:8]} 011: G_THD[7:0] 100: {6'h0, B_THD[9:8]} 101: B_THD[7:0] 110: {6'h0, ALL R/G/B[9:8]} 111: ALL R/G/B[7:0]	
RESERVED	3	--	0	Reserved	
DITHER_SEQ_SET	2	R/W	0	**Sequence table select** 0: all select setting0 1: select setting1 only if R/G/B < THD	
Reserved	1	R/W	0	Reserved	
Dither_Table_Ref	0	R/W	1	**Table reference** 0: By VS/HS 1: By DEN (Default)	

Overlay/Color Palette/Background Color Control

Address: 6C **OVERLAY_CTRL (Overlay Display Control Register)** **Default: 00h**

Bit	Mode	Function
7:6	--	**Reserved to 0**
5	R/W	**Background color access enable** 0: Disable(Reset CR6D Write Pointer to R) 1: Enable
4:2	R/W	**Alpha blending level (Also enable OSD frame control register 0x003 byte 1[3:2]** 000: Disable (Default) 001 ~111: 1/8~ 7/8
1	R/W	**Overlay Sampling Mode Select:** 0: single pixel per clock (Default) 1: dual pixels per clock (The OSD will be zoomed 2X in horizontal scan line)
0	R/W	**Overlay Port Enable:** 0: Disable (Default) 1: Enable **Turn off overlay enable and switch to background simultaneously when auto switch to background.**

Address: 6D **BGND_COLOR_CTRL** **Default: 00h**

Bit	Mode	Function
7:0	R/W	**Background color RGB 8-bit value[7:0]**

- There are 3 bytes color select of background R, G, B, once we enable Background color access channel(CR6C[5] and the continuous writing sequence is R/G/B

Address: 6E **OVERLAY_LUT_ADDR (Overlay LUT Address)** **Default: 00h**

Bit	Mode	Function
7	R/W	**Enable Overlay Color Plate Access:** 0: Disable (Default) 1: Enable
6	R/W	**Double buffer write enable (auto clear)** 0: disable 1: enable
5:0	R/W	**Overlay-64*24 Look-Up-Table Write Address [5:0]**

- Auto-increment while every accessing "Overlay LUT Access Port".

- While OSDON, enable double buffer write is optional for modifying color palette LUT.
  If double buffer write is enable, the flow below must be followed:
  - Write [6] to be 1 and set LUT write address first
  - Write 3 data R[7:0], G[7:0], B[7:0] to 6F
  - Wait for [6] auto clear to be 0, then repeat the flow until the procedure is done

Address: 6F **COLOR_LUT_PORT (LUT Access Port)**

Bit	Mode	Function
7:0	W	**Color Palette 64*24 Look-Up-Table access port [7:0]**

- Using this port to access overlay color plate which addressing by the above registers.
- The writing sequence into LUT is [R0, G0, B0, R1, G1, B1, …R63, G63, and B63] and the address counter will be automatic increment and circular from 0 to 63.

Image Auto Function

Address: 70 **H_BOUNDARY_H**

Bit	Mode	Function
7:4	R/W	**Horizontal Boundary Start:** High Byte [11:8]
3:0	R/W	**Horizontal Boundary End:** High Byte [11:8]

Address: 71 **H_BOUNDARY_STA_L**

Bit	Mode	Function
7:0	R/W	**Horizontal Boundary Start:** Low Byte [7:0]

Address: 72 **H_BOUNDARY_END_L**

Bit	Mode	Function
7:0	R/W	**Horizontal Boundary End:** Low Byte [7:0]

Address: 73 **V_BOUNDARY_H**

Bit	Mode	Function
7:4	R/W	**Vertical Boundary Start:** High Byte [11:8]
3:0	R/W	**Vertical Boundary End:** High Byte [11:8]

Vertical boundary search should be limited by Vertical boundary start.

Address: 74 **V_BOUNDARY_STA_L**

Bit	Mode	Function
7:0	R/W	**Vertical Boundary Start:** Low Byte [7:0]

Address: 75 **V_BOUNDARY_END_L**

Bit	Mode	Function
7:0	R/W	**Vertical Boundary End:** Low Byte [7:0]

Address: 76 **RED_NOISE_MARGIN (Red Noise Margin Register)**

Bit	Mode	Function
7:2	R/W	**Red pixel noise margin setting register**
1:0	--	**Reserved to 0**

Address: 77 **GRN_NOISE_MARGIN (Green Noise Margin Register)**

Bit	Mode	Function
7:2	R/W	**Green pixel noise margin setting register**
1:0	--	**Reserved to 0**

Address: 78 **BLU_NOISE_MARGIN (Blue Noise Margin Register)**

Bit	Mode	Function
7:2	R/W	**Blue pixel noise margin setting register**
1	R/W	**Auto phase result address write** 0: CR-86 read only 1: CR-86 for result index
0	R/W	**Auto phase result mode** 0: old mode (default) 1: new mode

Address: 79 **DIFF_THRESHOLD**

Bit	Mode	Function
7:0	R/W	**Difference Threshold** (Threshold for DIFF no matter CR7D[2] = 0 or 1)

Address: 7A **AUTO_ADJ_CTRL0** **Default: 00h**

Bit	Mode	Function
7	R/W	**Field_Select_Enable:** Auto-Function only active when Even or Odd field. 0: Disable (Default) 1: Enable
6	R/W	**Field_Select:** Select Even or Odd field. Active when **Field_Select_Enable**.

		0: Active when ODD signal is “0” (Default) 1: Active when ODD signal is “1”
5	**R/W**	**Low Pass Filter (121-LPF)** 0: Disable (Default) 1: Enable
4	R/W	**Auto Function Acceleration :** 0: Disable (Default) 1: Enable For auto-balance (CR7D[1]=0), this function must be disabled.
3:2	R/W	**Vertical boundary search:** 00: 1 pixel over threshold (Default) 01: 2 pixel over threshold 10: 4 pixel over threshold 11: 8 pixel over threshold
1:0	R/W	**Color Source Select for Detection:** 00: B color (Default) 01: G color 10: R color 11: ALL (the result will be divided by 2)

Address: 7B **HW_AUTO_PHASE_CTRL0** **Default: 00h**

Bit	Mode	Function
7:3	**R/W**	**Number of Auto-Phase Step** (Valut+1) **(How many times (steps reference CR7B[2:0]) jumps when using Hardware Auto)**
2:0	**R/W**	**Hardware Auto Phase Step** 000: Step =1 (Default) 001 Step =2 010: Step =4 011: Step =8 1xx: Step =16

Address: 7C **HW_AUTO_PHASE_CTRL1** **Default: 00h**

Bit	Mode	Function
7	**R/W**	**Hardware Auto Phase Select Trigger** 0: IVS 1: Vertical Boundary End
6:0	**R/W**	**Initial phase of Auto-Phase** (0~127)

Address: 7D **AUTO_ADJ_CTRL1** **Default: 00h**

Bit	Mode	Function
7	R/W	**Measure Digital Enable Info when boundary search active** 0: Normal Boundary Search (Default) 1: Digital Enable Info Boundary Search.(Digital mode)
6	R/W	**Hardware / Software Auto Phase Switch** 0: Software (Default) 1: Hardware
5	R/W	**Color Max or Min Measured Select:** 0: MIN color measured (Only when Balance-Mode, result must be complemented) (Default) 1: MAX color measured
4	R/W	**Accumulation or Compare Mode** 0: Compare Mode (Default) 1: Accumulation Mode
3	R/W	**Mode Selection For SOD** 0: SOD Edge Mode (Default) 1: SOD Edge + Pulse Mode
2	R/W	**Type Selection For DIFF** 0: DIFF 1: (DIFF/4) * (DIFF/4) Total result for each color is divided by 8 if this bit is 1.
1	R/W	**Function (Phase/Balance) Selection** 0: Auto-Balance (Default) 1: Auto-Phase

0	R/W	**Start Auto-Function Tracking Function:** 0: stop or finished (Default) 1: start

Control Table/ Function	Sub-Function	CR7D.6	CR7D.5	CR7D.4	CR7D.3	CR7D.1	CR7C
Auto-Balance	**Max pixel**	X	1	0	0	0	X
	Min pixel	X	0	0	0	0	X
Auto-Phase Type	**Mode1**	1	1	1	0	1	Th
	Mode2	1	1	1	1	1	Th
Accumulation	**All pixel**	1	1	1	0	0	0

Table 1 Auto-Tracking Control Table

Address: 7E **VER_START_END_H (Active region vertical start Register)**

Bit	Mode	Function
7:4	R	**Active region vertical START measurement result:** bit[11:8]
3:0	R	**Active region vertical END measurement result:** bit[11:8]

Address: 7F **VER_START_L (Active region vertical start Register)**

Bit	Mode	Function
7:0	R	**Active region vertical start measurement result:** bit[7:0]

Address: 80 **VER_END_L (Active region vertical end Register)**

Bit	Mode	Function
7:0	R	**Active region vertical end measurement result:** bit[7:0]

Address: 81 **H_START_END_H (Active region horizontal start Register)**

Bit	Mode	Function
7:4	R	**Active region horizontal START measurement result:** bit [11:8]
3:0	R	**Active region horizontal END measurement result:** bit[11:8]

Address: 82 **H_START_L (Active region horizontal start Register)**

Bit	Mode	Function
7:0	R	**Active region horizontal start measurement result:** bit[7:0]

Address: 83 **H_END_L (Active region horizontal end Register)**

Bit	Mode	Function
7:0	R	**Active region horizontal end measurement result:** bit[7:0]

Address: 84 **AUTO_PHASE_3 (Auto phase result byte3 register)**

Bit	Mode	Function
7:0	R	**Auto phase measurement result:** bit[31:24]

Address: 85 **AUTO_PHASE_2 (Auto phase result byte2 register)**

Bit	Mode	Function
7:0	R	**Auto phase measurement result:** bit[23:16]

Address: 86 **AUTO_PHASE_1 (Auto phase result byte1 register)**

Bit	Mode	Function
7:0	R	**Auto phase measurement result:** bit[15:8] When CR-78[1] set to 1, auto phase result read index 0~255 indicates for maximum 64 auto phase results (4 Bytes every result)

Address: 87 **AUTO_PHASE_0 (Auto phase result byte0 register)**

Bit	Mode	Function
7:0	R	**Auto phase measurement result:** bit[7:0] The measured value of R or G or B color max or min. (Auto-Balance) When CR-78[0] set to 1, read CR-87 continuously for auto phase results

- CR-87 can be read continously for totally 256 Bytes (64 results).

The multiple results, Address: 84~87, are stored in mcu's xdata space.

SOD Xdata space Range is set by SOD_DATA_LOCATION(0xFFEF[6:5]).
Enable SOD Access Xram by SOD_ACCESS_DISABLE(0xFFE4[7]) .
Xdata access method is suggested method.

- Set CR-78[1] and CR-86 for read index before read CR-87.
- Once CR-87 has been read 4 times, read index in CR-86 will increase automatically.

When input is 2560x1600, there will be three case for Register 0x84~0x87:

a. Only SOD + Pulse for RGB
   2560x1600x255x2x3 = 6266880000 need 33 bits to indicate.
   CR 84~87 will give bit [32:1].
b. (SOD/4)^2 / 8 + Pulse for RGB
   2560x1600x(255/4)^2 /8 x2x3 = 12484800000 need 34 bits to indicate.
   CR 84~87 will give bit [33:2]
c. (SOD/4)^2 /8 + Pulse only for one color
   2560x1600x(255/4)^2 /8 x2 = 4161600000 need 32 bits to indicate.
   CR 84~87 will give bit [31:0]

Dithering Control (For Input Domain)

Register:: I_DITHERING_DATA_ACCESS				0x88	
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
DITHERING_DATA_ACCESS	7:0	W	0	Refer to following description	

A. When CR88[7:6] is 2'b01, dithering sequence table access is enabled:

- There are three set of dithering sequence table, each table contains 32 elements, s0, s1, … , s31. Each element has 2 bit to index one of 4 dithering table.
- Input data sequence is {sr3,sr2,sr1,sr0}, {sr7,sr6,sr5,sr4}, … , {sr31,sr30,sr29,sr28}, {sg3,sg2,sg1,sg0}, … , {sg31,sg30,sg29,sg28}, {sb3,sb2,sb1,sb0}, … , {sb31,sb30,sb29,sb28} for red, green and blue channel.
- R + (2R+1) * C choose sequence element, where R is Row Number / 2, and C is Column Number / 2.

B. When CR88[7:6] is 2'b10, dithering table access is enabled:

- For dithering table access, the red, green, blue each channel has 4 dithering table, each table is 2x2 elements, and one element has 4 bit for 10B/8B, the elements should fill 0 to 3, for 10B/6B, the elements should fill 0 to 15.
- Input data sequence is [Dr00 Dr01],[Dr02,Dr03], … , [Dr30,Dr31],[Dr32,Dr33], [Dg00,Dg01],[Dg02,Dg03], … , [Dg30,Dg31],[Dg32,Dg33], [Db00,Db01],[Db02,Db03], … , [Db30,Db31],[Db32,Db33].

D00	D01	D10	D11	D20	D21	D30	D31
D02	D03	D12	D13	D22	D23	D32	D33

C. When CR88[7:6] is 2'b11, temporal offset access is enabled:

- There are 16 element for temporal offset table, t0, t1, … , t15. Each element has 2 bit to index one of 4 temporal offset.
- Input data sequence is {t3,t2,t1,t0}, {t7,t6,t5,t4}, {t11,t10,t9,t8}, {t15,t14,t13,t12}.

Register:: I_DITHERING_CTRL1				0x89	
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
Dither_Access	7:6	W	0	**Enable Access Control** 00: disable (Default) 01: enable access dithering sequence table 10: enable access dithering table 11: enable access temporal offset	
Dither_en	5	R/W	0	**Enable Dithering Function** 0: disable (Default) 1: enable	
Dither_temp	4	R/W	0	**Temporal Dithering** 0: Disable (Default) 1: Enable	
Dither_table	3	R/W	0	**Dithering Table Value Sign** 0: unsigned 1: signed (2's complement)	
Dither_mode	2	R/W	0	**Dithering Mode** 0: New (Default) 1: Old	
Dither_V_Fram_M	1	R/W	0	**Vertical Frame Modulation** 0: Disable (Default) 1: Enable	
Dither_VH_Fram_M	0	R/W	0	**Horizontal Frame Modulation** 0: Disable (Default) 1: Enable	

Register:: I_DITHERING_CTRL2				0x8A	
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
Reserved	7:2	R/W	**0**	**Reserved**	
Reg_Dither_Test_en	1	R/W	**0**	**Test_mode** 0: Disable, dither_out = dither_result[9:2]; // truncate to integer number (Default)	

				1: Enable, dither_out = dither_result[7:0]; // propagate decimal part for test	
Dither_Table_Ref	0	R/W	1	**Table reference** 0: By VS/HS 1: By DEN (Default)	

Global Control for LVDS

Register:: LVDS_ADDR_PORT				0x8B	
Name	**Bits**	**R/W**	**Default**	**Name**	**Config**
LVDS_ADDR_PORT	7:0	R/W	0x0	**Address port for embedded LVDS access**	

Register:: LVDS_DATA_PORT				0x8C	
Name	**Bits**	**R/W**	**Default**	**Name**	**Config**
LVDS_DATA_PORT	7:0	R/W	0x0	**Data port for embedded LVDS access**	

Register:: LVDS_CTRL15				0x 8C-00	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:2	--	--	Reserved	
DISP_TYPE	1:0	R/W	0x1	**DISP_TYPE** 00: Reserved 01: LVDS (Default) 10: Hi-Z 11: Reserved	

Register:: LVDS_CTRL16				0x 8C-01	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	--	--	Reserved	

Register:: LVDS_CTRL17				0x 8C-02	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	--	--	Reserved	

Register:: LVDS_CTRL18				0x 8C-03	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	--	--	Reserved	

Register:: LVDS_CTRL19				0x 8C-04	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LVDS_SLVDSIL	7:6	R/W	0x2	LVDS driver current option I=([1]*20uA+[0]*10uA+50uA)*SL[2:0]. See Table 1.	
LVDS_SLVDSECKIL	5:4	R/W	0x2	LVDS driver current option for Even port C I=([1]*20uA+[0]*10uA+50uA)* SECKL[2:0]. See Table 1.	
LVDS_SLVDSOCKIL	3:2	R/W	0x2	LVDS driver current option for Odd port D I=([1]*20uA+[0]*10uA+50uA)* SOCKL[2:0]. See Table 1	
LVDS_DRVING_SELECT	1	R/W	0x1	**0:** Type A → ECK driver current for TXEC (Pin92、Pin93), OCK driver current for TXOC(Pin80、Pin81) ,Data driver current for (pin74~79、pin82、pin83、pin86~91、pin94、pin95) **1:** Type B → ECK driver current for TXEC (Pin88、Pin89) , OCK driver current for TXOC (Pin76、Pin77) ,Data driver current for (pin74、pin75、pin78~83、pin86、pin87、pin90~95)	
Reserved	0	--	--	Reserved	

Register:: LVDS_CTRL20				0x 8C-05	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LVDS_RESL[7:6]	**7:6**	**R/W**	**0x3**	00: x0 PMOS driver can't be used. 01: x3 PMOS driver 10: x2 PMOS driver 11: x5 PMOS driver	
LVDS_RESL[5:4]	**5:4**	**R/W**	**0x03**	00: x0 NMOS driver can't be used. 01: x1 NMOS driver 10: x2 NMOS driver 11: x3 NMOS driver	

LVDS_RESL[3]	**3**	**R/W**	**0x0**	Reserved	
LVDS_RESL[2]	**2**	**R/W**	**0x0**	0: ck1x rising edge sample 1: ck1x falling edge sample	
LVDS_RESL[1:0]	**1:0**	**R/W**	**0x0**	00: ck7x 0 delay, 01:ck7x delay 3stages 10:ck7x delay 6stages 11:ck7x delay 9stage,	

Register:: LVDS_CTRL21				**0x 8C-06**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:6	--	--	Reserved.	
LVDS_SL	5:3	R/W	0x6	LVDS driver current multiplier (See Table 1) 000 : x14 001 : x20 010 : x22 011 : x30 100 : x33 101 : x37 110 : x50 111 : x75	
LVDS_SECKL	2:0	R/W	0x6	LVDS driver current multiplier/ Even port C (See Table 1) 000 : x14 001 : x20 010 : x22 011 : x30 100 : x33 101 : x37 110 : x50 111 : x75	

Register:: LVDS_CTRL22				**0x 8C-07**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:6	--	--	Reserved.	
LVDS_SOCKL	5:3	R/W	0x6	LVDS driver current multiplier Odd port D (See Table 1) 000 : x14 001 : x20 010 : x22 011 : x30 100 : x33 101 : x37 110 : x50 111 : x75	
Reserved	2:0	--	--	Reserved.	

Register:: LVDS_CTRL23				**0x 8C-08**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	--	--	Reserved	

Register:: LVDS_CTRL24				**0x 8C-09**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LVDS_EDALAGL	7:3	R/W	0x0	Select CK/Data align or lag T/N for even port (N= 14 for LVDS) 0: align 1: lag T/N 10000: (Pin86、Pin87) 01000: (Pin88、Pin89) 00100: (Pin90、Pin91) 00010: (Pin92、Pin93) 00001: (Pin94、Pin95) LVDS_EDALAGL[4] → Pin86、Pin87(EA)	

				LVDS_EDALAGL[3] → Pin88、Pin89(EB) LVDS_EDALAGL[2] → Pin90、Pin91(EC) LVDS_EDALAGL[1] → Pin92、Pin93(ED) LVDS_EDALAGL[0] → Pin94、Pin95(EE)	
Reserved	2:0	--	--	Reserved	

Register:: LVDS_CTRL25				**0x 8C-0A**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LVDS_ODALAGL	7:3	R/W	0x0	Select CK/Data align or lag T/N for odd port (N= 14 for LVDS) 0: align 1: lag T/N 10000: (Pin74、Pin75) 01000: (Pin76、Pin77) 00100: (Pin78、Pin79) 00010: (Pin80、Pin81) 00001: (Pin82、Pin83) LVDS_ODALAGL[4] → Pin74、Pin75(OA) LVDS_ODALAGL[3] → Pin76、Pin77(OB) LVDS_ODALAGL[2] → Pin78、Pin79(OC) LVDS_ODALAGL[1] → Pin80、Pin81(OD) LVDS_ODALAGL[0] → Pin82、Pin83(OE)	
Reserved	2:0	--	--	Reserved	

Register:: LVDS_CTRL26				**0x 8C-0B**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:6	--	--	Reserved.	
LVDS_PRE_SR_ENL	5	R/W	0x0	Enable /Disable pre-emphasis or slew rate control function 0: Disable 1: Enable	
LVDS_PRE_SR_SELL	4	R/W	0x0	Pre-emphasis/ Slew rate control selection 0: pre-emphasis 1: slew rate control	
LVDS_PRE_SR_SL	3:0	R/W	0	LVDS pre-emphasis driving current multiplier for data port / LVDS slew rate control for Even/Odd data port 0000: X4 0001: X8 0011: X12 0111: X16 1111: X20 Ref Table 2	

Register:: LVDS_CTRL27				**0x 8C-0C**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LVDS_PRE_SECKL	7:4	R/W	0x0	LVDS pre-emphasis driving current multiplier for Even clock port (C port) / LVDS slew rate control 0000: X4 0001: X8 0011: X12 0111: X16 1111: X20 Ref Table 2	
LVDS_PRE_SOCKL	3:0	R/W	0x0	LVDS pre-emphasis driving current multiplier for Odd clock port (D port) / LVDS slew rate control 0000: X4 0001: X8 0011: X12 0111: X16 1111: X20 Ref Table 2	

Register:: LVDS_CTRL28			0x 8C-0D		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LVDS_RESVREL	7:4	R/W	0x0	LVDS even port reserve pin	
LVDS_RESVREL	3:2	R/W	0x0	00: Even port output 0pF 01: Even port output 1.62pF 10: Even port output 3pF 11: Even port output 4.62pF	
LVDS_RESVREL	1:0	R/W	0x0	LVDS even port reserve pin	

Register:: LVDS_CTRL29			0x 8C-0E		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LVDS_RESVROL	7:0	R/W	0x0	LVDS odd port reserve pin	
LVDS_RESVROL	3:2	R/W	0x0	00: Odd port output 0pF 01: Odd port output 1.62pF 10: Odd port output 3pF 11: Odd port output 4.62pF	
LVDS_RESVROL	1:0	R/W	0x0	LVDS odd port reserve pin	

Register:: LVDS_CTRL30			0x 8C-0F		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LVDS_PLL_LVNL	7:6	R/W	0x2	Select LVDS P2S ratio 0X: Reserved 10: 7X for LVDS 11: Reserved	
LVDS_PLL_DIVOL	5	R/W	0x0	PLL Output divider 0:Div1 1:Div2	
LVDS_PLL_EPI_ENL	4	R/W	0x0	Enable/disable even port clk pair phase interpolation 0: Disable 1: Enable	
LVDS_PLL_OPI_ENL	3	R/W	0x0	Enable/disable odd port clk pair phase interpolation 0: Disable 1: Enable	
LVDS_PLL_DIVNL	2:1	R/W	0x2	Feedback Divider Ratio 0x: 6 10: 7 11: 8	
Reserved	0	--	--	Reserved	

Register:: LVDS_CTRL31			0x 8C-10		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LVDS_PLL_RSL	7:5	R/W	0x3	PLL Loop Filter Resister Control 000:16K 001:18K 010:20K 011:22K 100:24K 101:26K 110:28K 111:30K	
LVDS_PLL_CSL	4:3	R/W	0x2	PLL Loop Filter Capacitor Control CS= 00:18p, 01:20p, 10:24p, 11:28p	
LVDS_PLL_CPL	2	R/W	0x0	CP Control 0:CP=1.77pF 1:CP=2.1pF	
LVDS_PLL_VSET_SEL	1:0	R/W	0x01	set VCO vc voltage 00:0.24V 01:0.48V 10:0.72V 11:0.96V	

Register:: LVDS_CTRL32				0x 8C-11	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LVDS_PLL_IPL	7:5	R/W	0x02	PLL Charge Pump Current Control Icp=(2.5uA+2.5uA* [0]+5uA* [1]+10uA* [2])	
LVDS_PLL_SEL_FBKL	4:3	R/W	0x01	sel PFD input frequency 00: fref 01: fbk 10: 1.2V(Reserved) 11:1.2V(Reserved)	
Reserved	2	--	--	Reserved	
Reserved	1	--	--	Reserved	
LVDS_PLL_VCORSTBL	0	R/W	0x1	RESET VCO (active Low)	

Register:: LVDS_CTRL33				0x 8C-12	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LVDS_PLL_VCOMDL	7:6	R/W	0x3	PLL VCO Default band mode 00: VCO slowest, 11: VCO fastest	
LVDS_PLL_CALBPL	5	R/W	0x0	PLL bypass calibration(active high) 0: calibration 1: no calibration	
LVDS_PLL_CALSWL	4	R/W	0x0	calibration validated (go high after power on 1200us)	
LVDS_PLL_CALLCHL	3	R/W	0x0	latch calibration (go high after power on 1100us)	
LVDS_PLL_CMPENL	2	R/W	0x0	cmp enable (go high after power on 1000us)	
LVDS_PLL_VO2	1	R	0x0	DPLL CAL OUT2	
LVDS_PLL_VO1	0	R	0x0	DPLL CAL OUT1	

Register:: LVDS_CTRL34				0x 8C-13	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LVDS_PLL_CAL	7:6	R	0x0	DPLL calibrated VCO code	
Reserved	5	--	--	Reserved	
LVDS_PLL_RESERVE1L	4	R/W	0x0	LVDS PLL SERVED	
LVDS_PLL_RESERVED	3:0	R/W	0x0	LVDS PLL SERVED	

Register:: LVDS_CTRL35				0x 8C-14	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LVDS_PLL_SEL_PSCTRLVBPL	7:6	R/W	0x0	Select Phase interpolation bias option 00: Max, PI bias 11: Min, PI bias	
Reserved	5:4	--	--	Reserved	
LVDS_PLL_PSCTRLEL[11:8]	3:0	R/W	0x0	LVDS even port clock phase selection. Ref.table 3	

Register:: LVDS_CTRL36				0x 8C-15	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LVDS_PLL_PSCTRLEL[7:0]	7:0	R/W	30	LVDS even port clock phase selection. Ref. table 3	

Register:: LVDS_CTRL37				0x 8C-16	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:4	--	--	Reserved	
LVDS_PLL_PSCTRLOL[11:8]	3:0	R/W	0x0	LVDS odd port clock phase selection. Ref. table 3	

Register:: LVDS_CTRL38				0x 8C-17	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LVDS_PLL_PSCTRLOL[7:0]	7:0	R/W	30	LVDS odd port clock phase selection. Ref. table 3	

Register:: LVDS_CTRL39				0x 8C-18	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**

Reserved	7:0	--	--	Reserved.	

Register:: LVDS_CTRL40				**0x 8C-19**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LVDS_PLL_EPS_ENABLE	7:3	R/W	0x08	Enable even port phase selection 0: disable, use phase 0 1: enable, use LVDS_PLL_PSCTRLEL 10000: (Pin86、Pin87) EA 01000: (Pin88、Pin89) EB 00100: (Pin90、Pin91) EC 00010: (Pin92、Pin93) ED 00001: (Pin94、Pin95) EE	
Reserved	2:0	--	--	Reserved	

Register:: LVDS_CTRL41				**0x 8C-1A**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LVDS_PLL_OPS_ENABLE	7:3	R/W	0x08	Enable odd port phase selection 0: disable, use phase 0 1: enable, use LVDS_PLL_PSCTRLOL 10000: (Pin74、Pin75) OA 01000: (Pin76、Pin77) OB 00100: (Pin78、Pin79) OC 00010: (Pin80、Pin81) OD 00001: (Pin82、Pin83) OE	
Reserved	2:0	--	--	Reserved	

Table1

SLVDSIL[1:0]				
SL[2:0]	00	01	10	11
000(x14)	0.7mA	0.84mA	0.98mA	1.12mA
001(x20)	1.00mA	1.20mA	1.40mA	1.60mA
010(x22)	1.10mA	1.32mA	1.54mA	1.76mA
011(x30)	1.50mA	1.80mA	2.10mA	2.40mA
100(x33)	1.65mA	1.98mA	2.31mA	2.64mA
101(x37)	1.85mA	2.22mA	2.59mA	2.96mA
110(x50)	2.50mA	3.00mA	3.50mA	4.00mA
111(x75)	3.75mA	4.50mA	5.25mA	6.00mA

Table 2

SR control				
SLVDSIL[1:0]				
PRE_SR_SL[3:0]	00	01	10	11
0000(x 4)	-/+0.2mA	-/+0.24mA	-/+0.28mA	-/+0.32mA
0001(x 8)	-/+0.4mA	-/+0.48mA	-/+0.56mA	-/+0.64mA
0011(x12)	-/+0.6mA	-/+0.72mA	-/+0.84mA	-/+0.96mA
0111(x16)	-/+0.8mA	-/+0.96mA	-/+1.12mA	-/+1.28mA
1111(x20)	-/+1mA	-/+1.2mA	-/+1.4mA	-/+1.6mA
Pre-emphasis				
0000(x 4)	+/-0.2mA	+/-0.24mA	+/-0.28mA	+/-0.32mA
0001(x 8)	+/-0.4mA	+/-0.48mA	+/-0.56mA	+/-0.64mA
0011(x12)	+/-0.6mA	+/-0.72mA	+/-0.84mA	+/-0.96mA
0111(x16)	+/-0.8mA	+/-0.96mA	+/-1.12mA	+/-1.28mA
1111(x20)	+/-1mA	+/-1.2mA	+/-1.4mA	+/-1.6mA

Table 3

Phase	MSB[7:0]	LSB[3:0]
0	0000 0011	0000
	0000 0011	0001
	0000 0011	0011
	0000 0011	0111
4	0000 0110	1111
	0000 0110	1110
	0000 0110	1100
	0000 0110	1000
8	0000 1100	0000
	0000 1100	0001
	0000 1100	0011
	0000 1100	0111
12	0001 1000	1111
	0001 1000	1110
	0001 1000	1100
	0001 1000	1000
16	0011 0000	0000
	0011 0000	0001
	0011 0000	0011
	0011 0000	0111
20	0110 0000	1111
	0110 0000	1110
	0110 0000	1100
	0110 0000	1000
24	1100 0000	0000
	1100 0000	0001
	1100 0000	0011
	1100 0000	0111
28	1000 0001	1111
	1000 0001	1110
	1000 0001	1100
	1000 0001	1000

Register:: LVDS_CTRL42				**0x 8C-1B**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	--	--	Reserved	

Register:: LVDS_CTRL43				**0x 8C-1C**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	--	--	Reserved	

Register:: LVDS_CTRL2				**0x 8C-1D**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reseved	7:0	R/W	0xAA	Reserved	

Register:: LVDS_CTRL44				**0x 8C-1E**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LVDS_CLK	7:0	R/W	0xE3	LVDS CLK timing (default : 11100011)	

Register:: LVDS_CTRL45				**0x 8C-1F**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**

Reserved	7:0	--	--	Reserved	

TCLK+

LVDS	Bit 1	Bit 0	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit 6	Bit 5
TXE0	ER1	ER0	EG0	ER5	ER4	ER3	ER2	ER1	ER0	EG0	ER5
TXE1	EG2	EG1	EB1	EB0	EG5	EG4	EG3	EG2	EG1	EB1	EB0
TXE2	EB3	EB2	DEN	VS	HS	EB5	EB4	EB3	EB2	DEN*6	VS*5
TXE3	ER7	ER6	RSV	EB7	EB6	EG7	EG6	ER7	ER6	RSV*7	EB7
TXO0	OR1	OR0	OG0	OR5	OR4	OR3	OR2	OR1	OR0	OG0	OR5
TXO1	OG2	OG1	OB1	OB0	OG5	OG4	OG3	OG2	OG1	OB1	OB0
TXO2	OB3	OB2	DEN	VS	HS	OB5	OB4	OB3	OB2	DEN*2	VS*1
TXO3	OR7	OR6	RSV	OB7	OB6	OG7	OG6	OR7	OR6	RSV*3	OB7

TABLE 1 Bit-Mapping 6bit(5~0)+2bit(7~6)

TCLK+

LVDS	Bit 1	Bit 0	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit 6	Bit 5
TXE0	ER3	ER2	EG2	ER7	ER6	ER5	ER4	ER3	ER2	EG2	ER7
TXE1	EG4	EG3	EB3	EB2	EG7	EG6	EG5	EG4	EG3	EB3	EB2
TXE2	EB5	EB4	DEN	VS	HS	EB7	EB6	EB5	EB4	DEN*6	VS*5
TXE3	ER1	ER0	RSV	EB1	EB0	EG1	EG0	ER1	ER0	RSV*7	EB1
TXO0	OR3	OR2	OG2	OR7	OR6	OR5	OR4	OR3	OR2	OG2	OR7
TXO1	OG4	OG3	OB3	OB2	OG7	OG6	OG5	OG4	OG3	OB3	OB2
TXO2	OB5	OB4	DEN	VS	HS	OB7	OB6	OB5	OB4	DEN*2	VS*1
TXO3	OR1	OR0	RSV	OB1	OB0	OG1	OG0	OR1	OR0	RSV*3	OB1

TABLE 2 Bit-Mapping 6bit(7~2)+2bit(1~0)

Register:: DIS_FORMAT1				**0x 8C-20**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:5	--	--	Reserved	
Reserved	4	R/W	0x00	Reserved	
D_SRAM_BIST_EN	3	R/W	0x0	Dual3 SRAM BIST enable 0:Disable 1:Enable	
D_SRAM_BIST_DONE	2	R	0x0	Dual3 SRAM BIST Test Finished 0:Running 1:Done	
D_SRAM_BIST_FAIL	1	R	0x0	Dual3 SRAM BIST Test fail flag 0:Ok 1:Failed	
DISP_CONV_TEST_SEL	0	R/W	0x0	~~Enable test switch test pin out~~ ~~0:~~ ~~1:~~ ~~Test function~~ Switch teset pins of disp_conv 0: [29:0] = {mem_f_web, mem_b_web, w_half_back, w_reverse, w_cnt[3], w_cnt[2], w_cnt[1], w_cnt[0], r_reverse, r_cnt[3], r_cnt[2],	

				r_cnt[1], r_cnt[0], remapping_vs_mux, remapping_hs_mux, remapping_pre_mux, remapping_ena_mux, disp_vs, disp_hs, disp_ena, disp_pre2, lvds_mode, m_clk_div2, m_clk_out, dout_en, dual_port_en, stuff_en, remap_en,} 1: [29:0] = {hwidth_cnt[2], hwidth_cnt[1], hwidth_cnt[0], equal_half, gxcnt1_clr, ft_clk_cnt[2], ft_clk_cnt[1], ft_clk_cnt[0], reset_region, blanking_flag_ro, wd2_flag_ro, den_out(LVDS), e_rsv, e_den, e_vs, e_hs, reg_sgl3p, reg_sgl6p, reg_dual3p, vs_out, hs_out, de_out}	

Register:: LVDS_CTRL46				**0x 8C-21**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
reg_ft_mode	7	R/W	0	1: FT test mode, force EE, OE pair output data to verify LVDS analog 0: nornal function	
LVDS_Reserved01	6:0	R/W	0x00	Reserved	

Register:: LVDS_CTRL47				**0x 8C-22**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LVDS_Reserved02	7:0	R/W	0x00	Reserved	

Register:: LVDS_CTRL48				**0x 8C-23**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LVDS_Reserved03	7:0	R/W	0x00	Reserved	

Register:: LVDS_CTRL49				**0x 8C-24**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	--	--	Reserved	

Register:: LVDS_CTRL50				0x 8C-25	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	--	--	Reserved	

Control for LVDS

Register:: LVDS_CTRL0				0x 8C-A0	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LVDS_ IBPOWL	7	R/W	0x0	Power Up LVDS IBGEN 0: Power down (Default) 1: Normal	
Reserved	6	--	--	Reserved	
LVDS_ELVDSPOWL	5	R/W	0x0	Power Up **LVDS** Even-Port **(pin86~95)** 0: Power down (Default) 1: Normal	
LVDS_OLVDSPOWL	4	R/W	0x0	Power Up **LVDS** Odd-Port **(pin74~83)** 0: Power down (Default) 1: Normal	
LVDS_PLL_WDRSTL	3	R/W	0x0	DPLL WD Reset 0: Normal 1: Reset	
LVDS_PLL_ WDSETL	2	R/W	0x0	DPLL WD Set 0: Normal 1: Set	
Reserved	1	--	--	**Reserved**	
LVDS_PLL_WDO	0	R	0x0	**LVDS Watch Dog** **0: Normal** **1: Abnormal**	

Register:: LVDS_CTRL1				0x 8C-A1	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7	--	--	Reserved	
LVDS_SVOCML[3]	6	R/W	0x0	VOCM option[3]	
LVDS_STSTIL	5:3	R/W	0x2	TSTPAD output control 000 : PLL_WDO 001 : VOCM 010 : IB40U 011 : IBVOCM 100 : PLLTST-fin 101 : PLLTST-fbak 110 : LVTST-LVDSIN[6] 111 : LVTST-CKDIN	
LVDS_SVOCML[2:0]	2:0	R/W	0x4	VOCM option[2:0] VCM option set by (8C-A1[6]+8C-A1[2:0]) 0000 : 1.07v 0001 : 1.12v 0010 : 1.17v 0011 : 1.22v 0100 : 1.29v 0101 : 1.33v 0110 : 1.38v 0111 : 1.43v 1000 : 0.5v 1001 : 0.55v 1010 : 0.6v 1011 : 0.65v 1100 : 0.7v 1101 : 0.75v 1110 : 0.8v 1111 : 0.85v	

Register:: LVDS_CTRL2				0x 8C-A2	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:6	--	--	Reserved	
LVDS_ENIB40UX2L	5	R/W	0x0	ENIB40UX2L: Double the LVDS output swing 0: 1X 1: 2X	
LVDS_SIBXL	4	R/W	0x1	SIBXL: Select the LVDS driving/vocm biasing current source 0: From ADC internal bandgap 1: From GDI refer to external resistor	
LVDS_PLL_POLAR	3	R/W	0x0	The phase of clock when PLL latch data 0: Postive 1: Negative	
LVDS_SIBGENL	2:0	R/W	0x3	SIBGENL (LVDS Current Source correction) 000 : 25uA/62.5uA 001 : 30uA/75uA 010 : 35uA/87.5uA 011 : 40uA/100uA (Default) 100 : 45uA/112.5uA 101 : 50uA/125uA 110 : 55uA/137.5uA 111 : 60uA/150uA	

Register:: LVDS_CTRL3				0x 8C-A3	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7	--	--	Reserved	
LVDS_ MIRROR	6	R/W	0x0	LVDS Mirror 0: Normal (TXE3+, TXE3-, TXEC+, TXEC-, TXE2+, TXE2-, TXE1+, TXE1-, TXE0+, TXE0-, TXO3+, TXO3-, TXOC+, TXOC-, TXO2+, TXO2-, TXO1+, TXO1-, TXO0+, TXO0-) 1: Mirror (TXE0-, TXE0+, TXE1-, TXE1+, TXE2-, TXE2+, TXEC-, TXEC+, TXE3-, TXE3+, TXO0-, TXO0+, TXO1-, TXO1+, TXO2-, TXO2+, TXOC-, TXOC+, TXO3-, TXO3+)	
Reserved	5:1	--	--	Reserved	
LVDS_BMTS	0	R/W	0x0	BMTS: Bit-Mapping Table Select 0: Table 1 for 8bit LVDS (Default) 1: Table 2 for 8bit LVDS	

Register:: LVDS_CTRL4				0x 8C-A4	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LVDS_RSV_E	7:6	R/W	0x2	**E_RSV**: even port reserve signal select 11: Always '1' 10: Always '0' 01: Reserved 00: PWM_0	
LVDS_DEN_E	5:4	R/W	0x0	**E_DEN**: even port data enable signal select 11: Always '1' 10: Always '0' 01: Reserved 00: DENA	
LVDS_VS_E	3:2	R/W	0x0	**E_VS**: even port VS signal select 11: Always '1' 10: DENA 01: Reserved 00: DVS	
LVDS_HS_E	1:0	R/W	0x0	**E_HS**: even port HS signal select 11: Always '1' 10: DENA 01: Reserved 00: DHS	

Register:: LVDS_CTRL5	0x 8C-A5

Name	Bits	R/W	Default	Comments	Config
LVDS_RSV_O	7:6	R/W	0x2	**O_RSV**: odd port reserve signal select 11: Always '1' 10: Always '0' 01: Reserved 00: PWM_1	
LVDS_DEN_O	5:4	R/W	0x0	**O_DEN**: odd port data enable signal select 11: Always '1' 10: Always '0' 01: Reserved 00: DENA	
LVDS_VS_O	3:2	R/W	0x0	**O_VS**: odd port VS signal select 11: Always '1' 10: DENA 01: Reserved 00: DVS	
LVDS_HS_O	1:0	R/W	0x0	**O_HS**: odd port HS signal select 11: Always '1' 10:DENA 01: Reserved 00: DHS	

Register:: LVDS_CTRL6				**0x 8C-A6**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PN_SWAP	7	R/W	0x0	LVDS Differential pair PN swap (data) (Also refer to CR50[2:0]) 0: No Swap (Default) 1: Swap	
Reserved	6:1	--	--	Reserved	
LVDS_PLL_POWL	0	R/W	0x0	Power Up **LVDS** PLL 0: Power down (Default) 1: Normal	

Register:: LVDS_CTRL7				**0x 8C-A7**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	--	--	**Reserved**	

Register:: LVDS_CTRL8				**0x 8C-A8**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	--	--	**Reserved**	

Register:: LVDS_CTRL9				**0x 8C-A9**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	--	--	**Reserved**	

Register:: LVDS_CTRL10				**0x 8C-AA**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	--	--	Reserved	

Register:: LVDS_CTRL11				**0x 8C-AB**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	--	--	Reserved	

Register:: LVDS_CTRL12				**0x 8C-AC**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	--	--	Reserved	

Register:: LVDS_CTRL13				**0x 8C-AD**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	--	--	Reserved	

Register:: LVDS_CTRL14				0x 8C-AE	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	--	--	Reserved	

Test function

Register::Pin_config_Addr_Port					0x8D
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
Pin_config_Addr_Port	7:0	**R/W**	00	**Address port for pin configuration control access**	

Register::Pin_config_Data_Port					0x8E
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
Pin_config_Data_Port	7:0	**R/W**	00	**Data port for pin configuration control access**	

Name	8D-00[7] = 1'b0	8D-00[7] = 1'b1	PLL clock	scan	72pin	128pin
PAD_GPIO119				scan_mode	62	119
PAD_TXO3P			audio_pll	si[0]	37	74
PAD_TXO3N				si[1]	38	75
PAD_TXO2P	tst[0]	tst[0]	ck108_pll27x	si[2]	39	78
PAD_TXO2N	tst[1]	tst[1]		si[3]	40	79
PAD_TXO1P	tst[2]	tst[16]	fav4	si[4]	41	80
PAD_TXO1N	tst[3]	tst[17]		si[5]	42	81
PAD_TXO0P	tst[4]	tst[18]	xclk	si[6]	43	82
PAD_TXO0N	tst[5]	tst[19]		si[7]	44	83
PAD_TXE3P	tst[6]	tst[20]	dpll	so[0]	45	86
PAD_TXE3N	tst[7]	tst[21]		so[1]	46	87
PAD_TXECP	tst[8]	tst[22]	test2out	so[2]	47	88
PAD_TXECN	tst[9]	tst[23]		so[3]	48	89
PAD_TXE2P	tst[10]	tst[24]	dpll_status	so[4]	49	90
PAD_TXE2N	tst[11]	tst[25]		so[5]	50	91
PAD_TXE1P	tst[12]	tst[26]	test1out	so[6]	51	92
PAD_TXE1N	tst[13]	tst[27]		so[7]	52	93
PAD_TXE0P	tst[14]	tst[28]	m2pll		53	94
PAD_TXE0N	tst[15]	tst[29]			54	95

72 pin	128 pin		GPIO	Scan	Test Mode	Clock Out
71	1	TMDS_VDD				
72	2	PAD_REXT				
1	3	PAD_RX2P0				
2	4	PAD_RX2N0				
3	5	PAD_RX1P0				
4	6	PAD_RX1N0				
5	7	PAD_RX0P0				
6	8	PAD_RX0N0				
7	9	PAD_RXCP0				
8	10	PAD_RXCN0				
	11	TMDS_GND				
	12	PAD_RX2P1				
	13	PAD_RX2N1				
	14	PAD_RX1P1				
	15	PAD_RX1N1				
	16	PAD_RX0P1				
	17	PAD_RX0N1				
	18	PAD_RXCP1				
	19	PAD_RXCN1				
	20	TMDS_VDD				
9	21	PAD_AVS0			AVS_I	
10	22	PAD_AHS0			AHS_I	
11	23	ADC_VDD12_GDI				
11	23	ADC_VDD12				
12	24	PAD_BIN0N				
13	25	PAD_BIN0P				
14	26	PAD_GIN0N				
15	27	PAD_GIN0P				
	28	PAD_SOGIN0				
16	29	PAD_RIN0N				
17	30	PAD_RIN0P				
	31	PAD_BIN1N	PD.7		V8[7]_I	
	32	**PAD_32**	PD.6		V8[6]_I	
	33	**PAD_33**	PD.5		V8[5]_I	
	34	**PAD_34**	PD.4		V8[4]_I	
	35	**PAD_35**	PD.3		V8[3]_I	
	36	**PAD_36**	PD.2		V8[2]_I	
	37	**PAD_37**	PD.1	si[0]	V8[1]_I	
18	38	ADC_GND12		si[1]		
18	38	ADC_GNDOFF				
18	38	ADC_AUD_GNDOFF				
18	38	Audio_ADC_GND				
19	39	**PAD_39**	PD.0	si[2]	V8[0]_I	
20	40	**PAD_40**	PC.4	si[3]		
	41	PAD_VIN0P	PB.7	si[4]		
	42	PAD_VIN0N	PB.6	si[5]		
	43	PAD_VIN1P	PB.5	si[6]		
	44	PAD_VIN1N	PB4	si[7]		
	45	PAD_VIN2P	PB.3	so[0]		
	46	PAD_VIN2N	PB.2	so[1]		
	47	PAD_VIN3P	PB.1	so[2]		
	48	PAD_VIN3N	PB.0	so[3]		
	49	AVDD_SARADC_APAD		so[4]		
	49	AVDD_BB0_APAD				
	49	AVDD_BB1_DACVREF_APAD				
21	50	PAD_ADCA0	P6.0	so[5]		
22	51	PAD_ADCA1	P6.1	so[6]		
23	52	PAD_ADCA2	P6.2	so[7]		
24	53	PAD_ADCA3	P6.3			
	54	PAD_ADCA4	P6.4			
25	55	PAD_ADCB0	P6.5		DCLK_I	
26	56	PAD_ADCB1	P6.6		ADC_CLK_I	
27	57	PAD_ADCB2	P6.7		M2PLL_CLK_I	
28	58	PAD_DDCSCL1	P3.0/RXD(I/O)		HDMI_CP_ACLK_I	
72 pin	128 pin		GPIO	Scan	Test Mode	Clock Out
29	59	PAD_DDCSDA1	P3.1/TXD(O)		HDMI_CP_CLK_I	
30	60	PAD_VCCK				
	61	PAD_GND				
31	62	PAD_PVCC		scan_mode		
32	63	PAD_GPIO63	PC.3 /INT0(I)	scan_clk		
33	64	PAD_TCON0	P1.0/T2(I) /INT1(I)	scan_en		clk_pll27x

72 pin	128 pin		GPIO	Scan	Test Mode	Clock Out
29	59	PAD_DDCSDA1	P3.1/TXD(O)		HDMI_CP_CLK_I	
30	60	PAD_VCCK				
	61	PAD_GND				
31	62	PAD_PVCC		scan_mode		
32	63	PAD_GPIO63	PC.3 /INT0(I)	scan_clk		
33	64	PAD_TCON0	P1.0/T2(I) /INT1(I)	scan_en		clk_pll27x
34	65	PAD_TCON1	P1.1/T2EX(I)			
35	66	PAD_TCON2	P1.2/CLKO2(O)			
36	67	PAD_TCON5	P1.3			
	68	PAD_TCON13	P1.4			
	69	PAD_TCON3	P1.5			
	70	PAD_TCON9	P1.6			
	71	PAD_TCON8	P1.7			
	72	PAD_TCON6	PC.2			
	73	PAD_VCCK				
37	74	PAD_TXO3P	P9.0			audio_pll
38	75	PAD_TXO3N	P9.1			
	76	PAD_TXOCP	P9.2			
	77	PAD_TXOCN	P9.3			
39	78	PAD_TXO2P	P9.4		Test_IO[0]	ck108_pll27x
40	79	PAD_TXO2N	PA.0		Test_IO[1]	
41	80	PAD_TXO1P	PA.1		Test_IO[2]	fav4
42	81	PAD_TXO1N	PA.2		Test_IO[3]	
43	82	PAD_TXO0P	PA.3		Test_IO[4]	xclk
44	83	PAD_TXO0N	PA.4		Test_IO[5]	
	84	PAD_PVCC				
	85	PAD_PGND				
45	86	PAD_TXE3P			Test_IO[6]	dpll
46	87	PAD_TXE3N			Test_IO[7]	
47	88	PAD_TXECP			Test_IO[8]	test2out
48	89	PAD_TXECN			Test_IO[9]	
49	90	PAD_TXE2P			Test_IO[10]	dpll_status
50	91	PAD_TXE2N			Test_IO[11]	
51	92	PAD_TXE1P			Test_IO[12]	test1out
52	93	PAD_TXE1N			Test_IO[13]	
53	94	PAD_TXE0P			Test_IO[14]	m2pll
54	95	PAD_TXE0N			Test_IO[15]	
	96	PAD_PWM0	P5.2		Test_IO[16]	
	97	PAD_PWM1	P5.3		Test_IO[17]	
	98	PAD_PWM2	P5.4		Test_IO[18]	
	99	PAD_PWM3	P5.5		Test_IO[19]	
	100	PAD_PWM4	P5.6		Test_IO[20]	
	101	PAD_PWM5	P5.7		Test_IO[21]	
	102	PAD_SPDIF3	P7.6		Test_IO[22]	
55	103	PAD_SPDIF2	P7.5		Test_IO[23]	
56	104	PAD_SPDIF1	P7.4		Test_IO[24]	
	105	PAD_SPDIF0	P8.0		Test_IO[25]	
57	106	PAD_PVCC				
	107	PAD_PGND				
	108	PAD_MCK	P8.1/CLKO1(O)		Test_IO[26]	
	109	PAD_SCK	P3.2/INT0(I)		Test_IO[27]	
	110	PAD_WS	P3.3/INT1(I)		Test_IO[28]	
	111	PAD_SD0	P3.4/T0		Test_IO[29]	
	112	PAD_SD1	P3.5(BS)/T1			
	113	PAD_SD2	P3.6			
	114	PAD_SD3	P3.7			
58	115	PAD_SPI_SCLK				
59	116	PAD_SI				
60	117	PAD_SO				
61	118	PAD_CS				
62	119	PAD_GPIO119	PC.1			
63	120	PAD_VCCK				

72 pin	128 pin		GPIO	Scan	Test Mode	Clock Out
63	120	PAD_VCCK				
64	121	PAD_DDCSCL3	P7.3			
65	122	PAD_DDCSDA3	P7.2			
66	123	PAD_DDCSDA2	P7.1		ISP_DDCSCL	
67	124	PAD_DDCSCL2	P7.0		ISP_DDCSDA	
68	125	PAD_RESETB				
	126	PAD_126	PC.0			
69	127	PAD_XO				
70	128	PAD_XI				

Register::TEST_MODE					0x00
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
Select_data_test_mode	7	**R/W**	0	Select Data Test mode LSB **A. for 72 pin, pin54~45, 44-39** 0 : select Data test mode [15:0] 1: select Data test mode [29:16],[1:0] **B. for 128 pin, pin95~86, 83~78** 0 : select Data test mode [15:0] 1 : select Data test mode [29:16],[1:0] **pin 111~108, 105~96** is Data test mode [29:16]	
Test mode select	6:5	**R/W**	00	00:Normal 01:test_output mode Others are Reserved	
Test_output_Mode	4	**R/W**	0	0:Select Data test mode Select Data test output to 128pin 95~86, 83~78 or 64pin 49~40, 38~33 depend on bit7, bit3~bit0 1:PLL test mode {dpll, m2pll,audio_pll, tie to GND , IOSC, mpll, dpllstatus,test1out, test2out, fav4, xclk, dpll(digital pad)} will be outputted to 128 pin {86, 94, 74, X, 64, 78, 90, 92, 88, 80, 82, 126} or 64 pin {40, 48, 31, X, 30, 33, 44, 46, 42, 35, 37} when set to 1, clock frequency of some test pin could be divided by assigning its corresponding TST_CLK_CTRL	
Data_Test_mode	3:0	**R/W**	0	**0000:** 1'b0, Z0TST[3:0], pclk_tst, Red[7:0], Green[7:0], Blue[7:0] through VGIP **0001:** 1'b0, Z0TST[3:0], adc_clk, Red[7:0], Green[7:0], Blue[7:0] After Scale Down **0010:** Z0TST[3:0], adc_clk, IVS_DLY, IHS_DLY, IFD_ODD, IENA, VSD_DEN, VSD_ACT,Auto_hs, Auto_vs, auto_field, 1'b0, COAST, test_s1, test_s2, CLAMP_G, CLAMP_BR, SOG_IN0, SOG_IN1, FAV4,final_pe_com, t_s[1:0], pe_extrab, high_88, recur_delay_chain_en, high_127 **0011:** adc_clk, MCUWR, MCURD, MCU_ADR_INC, MIN[7:0], MCUWR, MCURD, MADR[7:0], MPLL_SDMOUT_TST[3:0], SDMOUT_TST[3:0] **0100:** 1'b0, adc_clk, RAW_VS, RAW_HS, RAW_ODD, RAW_DEN, SDMOUT_TST[3:0], Green[9:0], Red[9:0] through VGIP **0101:** 1'b0, adc_clk, Red[9:0], Green[9:0], raw_vs, raw_hs, en_flag, meas_ihs, HSOUT_sync_proc, coast, CLAMP_G, CLAMP_BR **0110:** 1'b0, adc_clk, raw_vs, raw_hs, test_s1, test_s2, raw_filed, Blue[9:0], Green[9:0], hs0_schmitt, hs1_schmitt, ~apll_por **0111:** 3'b0, adc_clk, Green[9:0], iclk_tst, raw_vs, raw_hs, raw_filed, fifo_clk, internal_crystal, test_s1, test_s2,	

				sync_pro_tst[7:0] **1000:** AUDIO_DAC enable signal test pin: dac_2ch_otpin[29:0] **1001:** VSDMAIN test mode: pclk_tst,, vsd_pr[7:0], vsd_y[7:0],vsd_pb[7:0,den,dvs,dhs,ch1_ivs, vsd_den, ~~**1010:** Auto_soy test mode~~ **1010:** misc, ref CR-8E-0D[3:0] **1011:** avc_eq_test[29:0] **1100:** Embedded MCU test out mode ~~**1101:** HDMI test in mode HDMI_TST_IN [0:29] will be assigned to {124~121, 114~108, 105~100, 72~64, 54~51}~~ **1110:** HDMI test out mode HDMI_TST_OUT [0:29] will be assigned to {124~121, 114~108, 105~100, 72~64, 54~51} **1111:** 6'b0, new_fifo_odata_g, new_fifo_odata_b, new_fifo_odata_r When set to 0010/0110/0111,test_s1 & test_s2 can be assigned by "Select_Tst_s1s2" **Others are reserved**	

Register::TST_CLK_CTRL0				**0x01**	
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
DPLL_OEN	7	R/W	0	DPLL frequency output enable 0: output disabled 1: output enabled	
M2pll_OEN	6	R/W	0	M2PLL frequency output enable 0: output disabled 1: output enabled	
Audio_pll_OEN	5	R/W	0	Audio_PLL frequency output enable 0: output disabled 1: output enabled	
Reserved	4	R/W	0	Reserved to 0	
CLK108_PLL27X_OEN	3	R/W	0	CLK108_PLL27X frequency output enable (M_domain clk, refer to CR_22[1:0]) 0: output disabled 1: output enabled	
Test1out_OEN	2	R/W	0	Test1out frequency output enable 0: output disabled 1: output enabled	
Test2out_OEN	1	R/W	0	Test2out frequency output enable 0: output disabled 1: output enabled	
Fav4_OEN	0	R/W	0	Fav4 frequency output enable 0: output disabled 1: output enabled	

Register::TST_CLK_CTRL1				**0x02**	
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
XCLK_OEN	7	R/W	0	XCLK frequency output enable 0: output disabled 1: output enabled	
CKT_PLL27X_OEN	6	R/W	0	CKT_PLL27X frequency output enable 0: output disabled 1: output enabled	
Rev	5:0	---	---	Reserved	

Register::TST_CLK_CTRL2					0x03
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
Rev	7:6	**---**	---	Reserved	
DPLL_DIV_CTRL	5:4	**R/W**	00	DPLL frequency is divided by 00:1 01:2 10:4 11:8	
M2pll_DIV_CTRL	3:2	**R/W**	11	M2PLL frequency is divided by 00:1 01:2 10:4 11:8	
Audio_pll_DIV_CTRL	1:0	**R/W**	00	Audio PLL frequency is divided by 00:1 01:2 10:4 11:8	

Register:: TST_CLK_CTRL3					0x04
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
Fav4 _DIV_CTRL	7:6	**R/W**	00	Fav frequency is divided by 00:1 01:2 10:4 11:8	
Test1out _DIV_CTRL	5:4	**R/W**	00	Test1out frequency is divided by 00:1 01:2 10:4 11:8	
Test2out _DIV_CTRL	3:2	**R/W**	00	Test2out frequency is divided by 00:1 01:2 10:4 11:8	
CLK108_pll27xDIV_CTRL	1	**R/W**	0	(M_domain clk DIV) 0:Divided by 1 1:Divided by 4	
Ckt_pll27x DIV_CTRL	0	**R/W**	0	0:divided by 1 1:divided by4	

Register:: Select_Tst_s1s2					0x05
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
Reserved	7	**R/W**	0	**Reserved**	
Select_Tst_s1	6:4	**R/W**	001	**Select test function of test_s1** **3'b000: DPLL clock (TIE LOW NOW)** **3'b001: PLLS fbk clock** **3'b010: CKOAD2(High Speed)** **3'b011: PLL status** **3'b100: HSOUT** **3'b101: ADC clock(from PLLS)(High Speed)** **3'b110: Empty Flag** **3'b111: BVS(Video8)**	
Reserved	3	**R/W**	0	**Reserved**	
Select_Tst_s2	2:0	**R/W**	010	**Select test function of test_s1** **3'b000: PLLS phase swallow clock (High speed)** **3'b001: DPLL status(TIE LOW NOW)**	

				3'b010: PLLS phase0 clock(High speed) **3'b011: M2PLL clock(Not in APLL)** **3'b100: HSFB** **3'b101: TP2_MX5** **3'b110: Full Flag** **3'b111: BHS(Video8)**	

Register:: Select_Tstinclock				**0x06**	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
DPLL_TST_IN	7	**R/W**	0	**0:Normal** **1:DCLK enter from pin ~~55~~ 50**	
ADCPLL_TST_IN	6	**R/W**	0	**0:Normal** **1:ADC CLK enter from pin ~~56~~ 51**	
M2PLL_TST_IN	5	**R/W**	0	**0:Normal** **1:M2PLL CLK enter from pin ~~57~~ 52**	
HDMI_CP_ACLK _TST_IN	4	**R/W**	0	**0:Normal** **1:HDMI_CP_ACLK enter from pin 58**	
HDMI_CP_CLK _TST_IN	3	**R/W**	0	**0:Normal** **1:HDMI_CP_CLK enter from pin 59**	
SCAN_CLK_TST_IN	2	**R/W**	1	**0:Normal** **1:SCAN CLK enter from pin ~~98~~63**	
DPLL_NDIV2_EN	1	**R/W**	0	**DPLL Test Mode Divider Enable** **0:use pin 50 div2 as dclk** **1:use pin 50 as dclk**	
MPLL_TST_IN	0	**R/W**	0	**0:Normal** **1:MPLL CLK enter from pin 51(128pin) or pin?? (68pin)**	

TEST MODE in FIFO

Register:: ADC TEST MODE				**0x07**	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_TEST_MODE	7	**R/W**	0	**0:Disable** **1:Enable**	
ADC_TEST_MODE_2	6	**R/W**	0	**Useless**	
FIFO_TEST_MODE	5	**R/W**	0	0:Disable 1:Enable test the CRC from FIFO，and open the Pattern Gen to d domain. Pattern Gen Seed (R = 01,G=00,B=00)	
ADC_TEST_START	4	**R**	0	Under ADC_TEST_MODE = 1, ADC_TEST_START will high when the new fifo is full ， then read out data from FIFO by sending DCLK from outside test pin.	
Rev	3:0	**---**	---	Reserved	

Register:: ADC TEST MODE ADDR MSB				**0x08**	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_TEST_ADDR[1:0]	7:6	**R/W**	0X00	Read the FIFO initial Addr.	
Rev	5:0	**---**	---	Reserved	

Register:: ADC TEST MODE ADDR LSB				**0x09**	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_TEST_ADDR[7:0]	7:0	**R/W**	0X04	Read the FIFO initial Addr.	

Register:: ADC FIFO CRC	**0x0A**

Name	Bit	R/W	Default	Description	Config
NEW_FIFO_CRC[23:16]	7:0	**R**	0	NEW FIFO CRC	

Register:: ADC FIFO CRC				**0x0B**	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
NEW_FIFO_CRC[15:8]	7:0	**R**	0	NEW FIFO CRC	

Register:: ADC FIFO CRC				**0x0C**	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
NEW_FIFO_CRC[7:0]	7:0	**R**	0	NEW FIFO CRC	

Register:: Auto_soy_test				**0x0D**	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
Auto_soy_test_mode	7	**R/W**	0	0: Normal 1: Test mode clk will be from pin51 & 6 bits data will be from {pin 32~pin37}	
Auto_soy_test_sel	6:4	**R/W**	0	Test function sel	
Test Signal Misc	3:0	**R/W**	0	**0000:** 5'b0, nr_db_en, force_avg_db_en, hist_db_en, vsync_ion, den_in, vact_in, vsync_out, den_out, vact_out, line_end_update_hist, mem_wr_sel, tone_red[2:0], tone_grn[2:0], tone_blu[2:0], raw_hist_red, raw_hist_grn, raw_hist_blu, rst_dum_hist_rst_mode, rst_main_hist_rst_mode **0001:** color processing test out **0010:** disp conv test out **0011:** idclti test out, refer to page 0x11, CR-A1[4] **0100:** vivid color test out, refer to page 0x11, CR-F0p7:6] **0101:** autosoy test **Others are reserved.**	

Embedded OSD

Address: 90 **OSD_ADDR_MSB (OSD Address MSB 8-bit)**

Bit	Mode	Function
7:0	R/W	**OSD MSB 8-bit address**

Address: 91 **OSD _ADDR_LSB (OSD Address LSB 8-bit)**

Bit	Mode	Function
7:0	R/W	**OSD LSB 8-bit address**

Address: 92 **OSD_DATA_PORT (OSD Data Port)**

Bit	Mode	Function
7:0	W	**Data port for embedded OSD access**

Refer to the embedded OSD application note for the detailed.

Address: 93 **OSD_SCRAMBLE** **Default: 05h**

Bit	Mode	Function
7	R/W	**BIST Start** 0: stop (Default) 1: start (auto clear)
6	R	**BIST Result** 0: fail (Default) 1: success

5	~~R/W~~ --	~~**BIST SELECT**~~ ~~0: 20K byte FONT SRAM~~ ~~1: 192 byte LUT SRAM~~ Reserved (Default 0)
4	R	**Double_Buffer_Write_Status** 0: double buffer write out is finish, or data write to double buffer is not ready, or no double buffer function. 1: after data write to dbuf and before dbuf write out, such that double buffer is busy.
3	R/W	**OSDADRHSB** **0: If initial address lower than or equal to 12K** **1: If initial address higher than 12K** The bit will be designed to control 20K bytes SRAM. However it will have no effect for WINDOW setting. Also please remember to set {OSDADRHSB, OSDADRMSB(CR90), OSDADRLSB(CR91) } again while you like to R/W a new address.
2:0	R/W	**Double buffer depth (Default=6)** **000~101=>1~6**

Address: 94 **OSD_TEST**

Bit	Mode	Function
7:0	R/W	**Testing Pattern**

Digital Filter

Address: 98 **DIGITAL_FILTER_CTRL** **Default: 00h**

Bit	Mode	Function
7:4	R/W	**Access Port Write Enable** 0000: disable 0001: phase access port 0010: negative smear access port 0011: positive smear access port 0100: negative ringing access port 0101: positive ringing access port 0110: mismatch access port 0111: Y(B)/Pb(G)/Pr(R) channel digital filter enable 1xxx: noise reduction access port
3:2	R/W	**Two condition occur continuous** (ringing to smear) 00: disable(hardware is off , depend on firmware) 01: only reduce ringing condition 10: only reduce smear condition 11: no adjust (hardware is on, but do nothing)
1	R/W	**When noise reduction and mismatch occur, select** 0: mismatch 1: noise reduction
0	--	**Reserved to 0**

Address: 99 **DIGITAL_FILTER_PORT** **Default: 00h**

DIGITAL_FILTER_CTRL[7:4] =-0111		
Bit	**Mode**	**Function**
7	R/W	**Y EN (G)**: function enable 0: function disable 1: function enable
6	R/W	**Pb EN (B)** : function enable 0: function disable 1: function enable
5	R/W	**Pr EN (R)** : function enable 0: function disable 1: function enable
4	R/W	**Initial value**: 0: raw data 1: extension
3:0	--	**Reserved to 0**

DIGITAL_FILTER_CTRL[7:4] = 0000 ~0 110 or 1xxx		
Bit	**Mode**	**Function**
7	R/W	**EN**: function enable 0: function disable 1: function enable
6:4	R/W	**THD_OFFSET** Threshold value of phase and mismatch and noise reduction or offset value of smear and ringing
3:2	R/W	**DIV:** divider value of phase, smear and ringing 00: 0 01: 1 10: 2 11: 3
1:0	--	**Reserved to 0**

THD_OFFSET define:

The THD value definition of phase enhance function

Bit6~4	000	001	010	011	100	101	110	111
Value	112	128	144	160	176	192	208	224

The offset value definition of smear and ringing reduce function

Bit6~4	000	001	010	011	100	101	110	111
Value	no use	16	32	48	64	80	96	112

The THD value definition of mismatch enhance function

Bit6~4	000	XX1
Value	1	2

The THD value definition of noise reduction function

Bit6~4	000	001	010	011	100	101	110	111
Value	0	1	2	3	4	5	6	7

Video Color Space Conversion(Input Domain)
YUV2RGB

Address: 9C **YUV_RGB_CTRL (YUV <-> RGB Control Register)** **Default: 10h**

Bit	Mode	Function
7	R/W	Y_OUT Shift 0: Bypass 1: Y_Out+64
6	R/W	CbCr_Out_Shift: 0 : Bypass 1: Cb_Out+512, Cr_Out+512
5	---	Reserved
4	R/W	Color Conversion Type 0: YUV->RGB 1: RGB->YUV (U,V are translated to unsigned 10-bit number)
3	R/W	Enable YUV/RGB coefficient Access: 0: Disable 1: Enable If this bit is set to 0, the address of the data port will reset to 0, and continuously writes 18 bytes
2	R/W	Cb Cr Clamp 0: Bypass 1: Cb-512, Cr-512 (MSB Inversion)
1	R/W	Y Gain/Offset: 0 : Bypass 1: (Y-64)*1.164
0	R/W	Enable YUV <-> RGB Conversion: 0: Disable YUV<->RGB conversion (Default) 1: Enable YUV<->RGB conversion

Address: 9D **YUV_RGB_COEF_DATA**

Bit	Mode	Function
7:0	W	COEF_DATA[7:0]

YUV->RGB matrix : (CR9C[4] = 0)

- CR9C[1] = 0, CR9C[2] = 1,

R = h00*Y + h01*(Cb-512) + h02*(Cr-512)
G = h10*Y + h11*(Cb-512) + h12*(Cr-512)
B = h20*Y + h21*(Cb-512) + h22*(Cr-512)

- CR9C[1] = 1, CR9C[2] = 1,

R = h00*(1.164*(Y-64)) + h01*(Cb-512) + h02*(Cr-512)
G = h10*(1.164*(Y-64)) + h11*(Cb-512) + h12*(Cr-512)
B = h20*(1.164*(Y-64)) + h21*(Cb-512) + h22*(Cr-512)

The output, Rout / Gout / Bout, should be calculated by offset / gain operation

Rout = (R' + Roffset) * Rgain
Gout = (G' + Goffset) * Ggain
Bout = (B' + Boffset) * Bgain

RGB->YUV matrix : (CR9C[4] = 1, CR9C[2:1] = 00)

Y = h00*R + h01*G + h02*B
Cb = h10*R + h11*G + h12*B
Cr = h20*R + h21*G + h22*B

All 'h' coefficients are expressed as 2's complement with 4-bit signed-extension, 2-bit integer and 10-bit fractional number. (0x0400 means 1.0)

When color conversion setting is YUV->RGB (CR9C[4]=0), h00, h10 and h20 is not effective(force to 1.0 internally).

Integer part is only effective for h02, h21. For other coefficients, integer part must be the same as signed- extension.

Offsets are S(14,4) 14bits, 10-bits signed integer and 4-bit fractional bits

Gains are U(10,9) 10bits, 1-bit integer and 9-bit fractional bits.

Coefficient Sequence (30-byte) : h00 (High-byte), h00 (Low-byte), h01 (High-byte), h01 (Low-byte), h02 (High-byte), h02 (Low-byte), h10 (High-byte), h10 (Low-byte), h11 (High-byte), h11 (Low-byte), h12 (High-byte), h12 (Low-byte), h20 (High-byte), h20 (Low-byte), h21 (High-byte), h21 (Low-byte), h22 (High-byte), h22 (Low-byte).Roffset (Hight-byte), Roffset(Low-byte), Goffset (Hight-byte), Goffset(Low-byte), Boffset (Hight-byte), Boffset(Low-byte)Rgain (Hight-byte), Rgain (Low-byte), Ggain (Hight-byte), Ggain (Low-byte), Bgain (Hight-byte), Bgain (Low-byte)

Default value:
h00=0105h,h01=0202,h02=0063h,h10=ff69h,h11=fed8h,h12=01c0h,h20=01c0h,h21=fe89h,h22=ffb8h,offset=0x000h,gain=0x0200h

Paged Control Register

Address: 9F **PAGE_SEL** **Default: 00h**

Bit	Mode	Function
7:5	R/W	**Reserved to 0**
4:0	R/W	**Page Selector (CRA0~CRFF)** **Page 0: Embedded ADC/ABL/LVR/Schmitt trigger** **Page 1: PLL** **Page 2: HDMI** **Page 3: LiveShow** **Page 4: SDRAM control** **Page 5: FRC** **Page 6: IOSC AudioDAC** **Page 7: Vivid color/DCC/ICM** **Page 8: Audio volume control** **Page 9: Input Gamma** **Page A: Reserved** **Page B: GDI PHY** **Page C: GDI MAC** **Page D: MCU** **Page E: MCU** **Page F: MCU** **Page 10: Pin Share Control** **Page 11:Reserved** **Page 12: Reserved** **Page 13: Reserved** **Page 14: Reserved** **Page 15: Reserved** **Others: Reserved**

Embedded ADC (Page 0)

Register::ADC_POWER_SOG_SOY_CONTROL[7:0]					0xBA
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
Reserved	7:1	R/W	0x10	Reserved	
ADC_VIDEO	0	R/W	0b0	~~Video8 input control (0: analog input, 1: video8 input) (pin 31~40)~~ ~~**(useless, because there is no ADC2)**~~ **Not used**	

Register:: ADC_2X_SAMPLE[7:0]					0xBB
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_2X_SAMPLE[7]	7	**R/W**	0b0	ADC 2x over sample (0:1x 1:2x)	
ADC_2X_SAMPLE[6]	6	**R/W**	0b1	2x Clock Polarity ~~(0:Normal 1:Inverted)~~ **(0: Negative 1: Positive)**	
ADC_2X_SAMPLE[5]	5	**R/W**	0b1	1x Clock Polarity ~~(0:Normal 1:Inverted)~~ **(0: Negative 1: Positive)**	
ADC_2X_SAMPLE[4]	4	**R/W**	0b0	common-mode output voltage source selection for pipeline stages 0: from VCC (default), 1: from BG	
ADC_2X_SAMPLE[3]	3	**R/W**	0b0	'refgen' ckt P-side cascode on/off selection 0: off (default), 1:on	
ADC_2X_SAMPLE[2]	2	**R/W**	0b0	clock input select (0:from CKOAD_V33, 1:from CKOAD_V12)	
ADC_2X_SAMPLE [1]	1	**R/W**	0b0	'refgen' ckt N-side gate resistor short enable 0: disable (default), 1: enable	
ADC_2X_SAMPLE [0]	0	**R/W**	0b0	'refgen' ckt P-side gate resistor short enable 0: disable (default), 1: enable	

Register:: ADC_CLOCK[7:0]					0xBC
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_CLOCK[7]	7	**R/W**	0b0	Input Clock Polarity (0:Negative 1:Positive)	
ADC_CLOCK[6]	6	**R/W**	0b0	Output Divider Clock Polarity (0:Normal 1:Inverted)	
ADC_CLOCK[5:4]	5:4	**R/W**	0b0	ADC_OUT_PIXEL Delay (00:1.05n 01:1.39n 10:1.69n 11:1.97n)	
ADC_CLOCK[3]	3	**R/W**	0b0	1X or 2X from APLL (0:1X 1:2X)	
ADC_CLOCK[2]	2	**R/W**	0b0	Single Ended or Diff. Clock from APLL (0:Diff. 1:Single Ended)	
ADC_CLOCK[1:0]	1:0	**R/W**	0b1	Duty Staiblizer(00: 48% 01:50% 10: 51% 11:52%)	

Register:: ADC_TEST[7:0]					0xBD
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_TEST[7]	7	**R/W**	0b0	Reserved (move to ADC_SOG_CMP[4])	
ADC_TEST[6:4]	6:4	**R/W**	0b000	**TEST OUPUT SELECTION** **SOGIN0 (000:X 001:gnd 010:vrefn 011:vcm 100:vrefp 101:vdd 110:vdd 111:vdd)**	
Reserved	3:2	**R/W**	0b0	Reserved	
ADC_TEST[1:0]	1:0	**R/W**	0b00	Clock Output Divider (00: 1/1 01: 1/2 10: 1/3 11: 1/4)	

Register::RGB gain_LSB					0xBE
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**

Reserved	7:6	**R/W**	0b0	Reserved	
ADC_GAI_RED[1:0]	5:4	**R/W**	0x0	Red Channel Gain Adjust[1:0]	
ADC_GAI_GRN[1:0]	3:2	**R/W**	0x0	Green Channel Gain Adjust[1:0]	
ADC_GAI_BLU[1:0]	1:0	**R/W**	0x0	Blue Channel Gain Adjust[1:0]	

Register::RGB offset_LSB					**0XBF**
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
Reserved	7:6	**R/W**	0b0	Reserved	
ADC_OFF_RED[1:0]	5:4	**R/W**	0x0	Red Channel Offset Adjust[1:0]	
ADC_OFF_GRN[1:0]	3:2	**R/W**	0x0	Green Channel Offset Adjust[1:0]	
ADC_OFF_BLU[1:0]	1:0	**R/W**	0x0	Blue Channel Offset Adjust[1:0]	

Register::red gain_MSB					**0xC0**
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_GAI_RED[9:2]	7:0	**R/W**	0x80	Red Channel Gain Adjust[9:2]	

Register::green gain_MSB					**0xC1**
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_GAI_GRN[9:2]	7:0	**R/W**	0x80	Green Channel Gain Adjust[9:2]	

Register::blue gain_MSB					**0xC2**
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_GAI_BLU[9:2]	7:0	**R/W**	0x80	Blue Channel Gain Adjust[9:2]	

Register::RED OFFSET_MSB					**0xC3**
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_OFF_RED[9:2]	7:0	**R/W**	0x80	Red Channel Offset Adjust[9:2]	

Register::GREEN OFFSET_MSB					**0xC4**
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_OFF_GRN[9:2]	7:0	**R/W**	0x80	Green Channel Offset Adjust[9:2]	

Register::BLUE OFFSET_MSB					**0xC5**
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_OFF_BLU[9:2]	7:0	**R/W**	0x80	Blue Channel Offset Adjust[9:2]	

Register:: ADC_POWER[7:0]					**0xC6**
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_POWER[7]	7	R/W	0b0	ADC VCM Power On (0: Power Down, 1: Power On) for RL6093 ADC 1.2V leakage	
ADC_POWER[6]	6	R/W	0b0	ADC clock Power On (0: Power Down 1: Power On)	
ADC_POWER[5]	5	**R/W**	0b0	SOG_ADC0 Power On (0: Power Down 1: Power On)	
ADC_POWER[4]	4	**R/W**	0b0	ADC Regulator Power On (0: Power Down 1: Power On)	
ADC_POWER[3]	3	**R/W**	0b0	Bandgap/IBIAS[4:0] Power On (0: Power Down 1: Power On)	
ADC_POWER[2]	2	**R/W**	0b0	Red Channel ADC Power On (0: Power Down 1: Power On)	
ADC_POWER[1]	1	**R/W**	0b0	Green Channel ADC Power On (0: Power Down 1: Power On)	
ADC_POWER[0]	0	**R/W**	0b0	Blue Channel ADC Power On (0: Power Down 1: Power On)	

Register:: ADC_IBIAS0[7:0]					**0xC7**
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**

ADC_IBIAS0[7:6]	7:6	**R/W**	0b01	[7:6]: PGA Input GM Current, High 2 bits Low 2 bits=ADC_IBIAS1<7:6> (0000: 0u, 0001: 50u, 0010: 100u, 0011: 150u 0100: 200u, 0101: 250u, 0110: 300u, 0111: 350u 1000: 400u, 1001: 450u, 1010: 500u, 1011: 550u 1100: 600u, 1101: 650u, 1110: 700u, 1111: 750u)	
ADC_IBIAS0[5:4]	5:4	**R/W**	0b01	~~Bias Current of RSDS20U (00:16u 01:20u 10:24u 11:28u)~~	
ADC_IBIAS0[3:2]	3:2	**R/W**	0b01	Bias Current of LVDS20U (00:16u 01:20u 10:24u 11:28u)	
ADC_IBIAS0[1:0]	1:0	**R/W**	0b01	Bias Current of RSDS_TTL20U (00:16u 01:20u 10:24u 11:28u)	

Register:: ADC_IBIAS1[7:0]					**0xC8**
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_IBIAS1[7:6]	7:6	**R/W**	0b01	[7:6]: PGA Input GM Current, Low 2 bits High 2 bits=ADC_IBIAS0<7:6>	
ADC_IBIAS1[5:4]	5:4	**R/W**	0b01	Settings for common-mode output voltage from BG If ADC_2X_SAMPLE[4]=0(default), doesn't matter If ADC_2X_SAMPLE[4]=1, ADC_IBIAS1[5:4]= 00: 630mV, 01:673mV (default), 10: 694mV, 11: 706mV	
ADC_IBIAS1[3:2]	3:2	**R/W**	0b01	Reserved	
ADC_IBIAS1[1:0]	1:0	**R/W**	0b01	Reserved	

Register:: ADC_IBIAS2[7:0]					**0xC9**
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_IBIAS2[7:6]	7:6	**R/W**	0b00	Reserved	
ADC_IBIAS2[5]	5	**R/W**	0b0	Reserved	
ADC_IBIAS2[4:2]	4:2	**R/W**	0b001	Bias Current of SOG DAC (000:22.5u 001:25u 010:27.5u 011:30u) (100:32.5u 101:35u 110:17.5u 111:20u)	
ADC_IBIAS2[1:0]	1:0	**R/W**	0b01	Bias Current of Audio_DAC (00:16u 01:20u 10:24u 11:28u)	

Register:: ADC_IBIAS3[7:0]					**0xCA**
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_IBIAS3[7:6]	7:6	**R/W**	0b01	Bias Current of ADC_SF (00:15u 01:20u 10:25u 11:30	
ADC_IBIAS3[5:3]	5:3	**R/W**	0b011	Bias Current of 1.2v mbias (000:17.5u 001:20u 010:22.5u 011:25u 100:27.5u 101:30u 110:32.5u 111:35u)	
ADC_IBIAS3[2:0]	2:0	**R/W**	0b001	Bias Current of SH,MDAC (000:12u 001:18u 010:24u 011:27u 100:30u 101:33u 110:39u 111:45u),	

				SUBADC (000:10u 001:10u 010:10u 011:10u 100:20u 101:20u 110:20u 111:20u)	

Register:: ADC_IBIAS4[7:0]				**0xCB**	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_IBIAS4[7:6]	7:6	**R/W**	0b01	Bias Current of DPLL20U/ DPLL_AUD/ DPLL_GDI (00:16u 01:20u 10:24u 11:28u)	
ADC_IBIAS4[5:4]	5:4	**R/W**	0b01	Bias Current of APLL_IB60U (00:48u 01:60u 10:72u 11:84u)	
ADC_IBIAS4[3]	3	**R/W**	0b0	SF LPF enable 0: disable (default), 1: enable If ADC_IBIAS4[3]=0, SF LPF is disabled. If ADC_IBIAS4[3]=1, SF LPF is enabled, the setting is as follows: ADC_CLAMP_CTRL2[5]=0: ADC_DCRESTORE_CTRL[2:1]= 00:73 MHz, 01:158 MHz, 10:250 MHz, 11:340 MHz ADC_CLAMP_CTRL2[5]=1: ADC_DCRESTORE_CTRL[2:1]= 00:38 MHz, 01:63 MHz, 10:96 MHz, 11:131 MHz	
ADC_IBIAS4[2]	2	**R/W**	0b1	Reserved	
ADC_IBIAS4[1:0]	1:0	**R/W**	0b01	Reserved	

Register:: ADC_VBIAS0[7:0]				**0XCC**	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_VBIAS0[7:6]	7:6	**R/W**	0b00	PGA Input Ioffset Current, High 2 bits Voffset=Ioffset*2.4K Low 2 bits=ADC_VBIAS0<3:2> (0000: 0u, 0001: 40u, 0010: 80u, 0011: 120u 0100: 160u, 0101: 200u, 0110: 240u, 0111: 280u 1000: 320u, 1001: 360u, 1010: 400u, 1011: 440u 1100: 480u, 1101: 520u, 1110: 560u, 1111: 600u)	
ADC_VBIAS0[5:4]	5:4	**R/W**	0b01	Select output voltage of ADC Regulator (00:1.2 01:1.3 10:1.4 11:1.5)	
ADC_VBIAS0[3:2]	3:2	**R/W**	0b00	PGA Input Ioffset Current, Low 2 bits High 2 bits=ADC_VBIAS0<7:6>	
ADC_VBIAS0[1:0]	1:0	**R/W**	0b01	Bandgap Voltage (00:1.15 01:1.25 10:1.34 11:1.42)	

Register:: ADC_VBIAS1[7:0]					**0xCD**
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_VBIAS1[7]	7	**R/W**	0b0	Reserved (move to ADC_SOG_CMP[7])	
ADC_VBIAS1[6]	6	**R/W**	0b0	R Channel Clamp to -300mV (0: 0mV	

				1:-300mV)	
ADC_VBIAS1[5]	5	**R/W**	0b0	G Channel Clamp to -300mV (0: 0mV 1:-300mV)	
ADC_VBIAS1[4]	4	**R/W**	0b0	B Channel Clamp to -300mV (0: 0mV 1:-300mV)	
ADC_VBIAS1[3]	3	**R/W**	0b1	SH boot enable (0:no boost, 1: boost)	
ADC_VBIAS1[2]	2	**R/W**	0b0	SH boot adjuest (0:0.8, 1:0.85)	
ADC_VBIAS1[1]	1	**R/W**	0b0	Reserved	
ADC_VBIAS1[0]	0	**R/W**	0b1	PGA Input GM Power On (0: Power Down, 1: Power On)	

Register:: ADC_CTL_RGB[7:0]					**0xCE**
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_CTL_RGB[7:4]	7:4	**R/W**	0b1000	SH gain(0000:0.95, 0001:1, 0010:1.05, 0011:1.1, 0100:1.15, 0101:1.2, 0110:1.25, 0111:1.3, 1000:1.35, 1001:1.4, 1010:1.45)	
ADC_CTL_RGB[3]	3	**R/W**	0b0	Dual (0: Input0 1: force to ground)	
ADC_CTL_RGB[2]	2	**R/W**	0b1	Single Ended or Diff. Input (0: Single Ended 1: Diff)	
ADC_CTL_RGB[1:0]	1:0	**R/W**	0b10	PGA LPF BW control (ADC_CLAMP_CTRL0[1], ADC_CTL_RGB[1:0] work together) ADC_CLAMP_CTRL0[1]=0: ADC_CTL_RGB[1:0]= 00:75MHz, 01:150MHz, 10: 250MHz, 11:340MHz ADC_CLAMP_CTRL0[1]=1: ADC_CTL_RGB[1:0]= 00:45MHz, 01:62MHz, 10: 97MHz, 11:122MHz	

Register:: ADC_CTL_RED[7:0]					**0xCF**
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_CTL_RED[7]	7	**R/W**	0b0	RGB/YPrPb Clamp (0: RGB 1:YPrPB) //ADC_VBIAS1[6]==0	
ADC_CTL_RED[6:4]	6:4	**R/W**	0b100	Clamp Voltage (0V~700mV, Step=100mV)	
ADC_CTL_RED[3]	3	**R/W**	0b0	Offset Depends on Gain (0: RGB Yes, YPrPb No 1:RGB No, YPrPb No)	
ADC_CTL_RED[2:0]	2:0	**R/W**	0b000	Red Channel ADC Fine Tune Delay, Step=90ps	

Register:: ADC_CTL_GRN[7:0]					**0XD0**
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_CTL_GRN[7]	7	**R/W**	0b0	RGB/YPrPb Clamp (0: RGB 1:YPrPB) //ADC_VBIAS1[6]==0	
ADC_CTL_GRN[6:4]	6:4	**R/W**	0b100	Clamp Voltage (0V~700mV, Step=100mV)	
ADC_CTL_GRN[3]	3	**R/W**	0b0	Offset Depends on Gain (0: RGB Yes, YPrPb No 1:RGB No, YPrPb No)	
ADC_CTL_GRN[2:0]	2:0	**R/W**	0b0	Green Channel ADC Fine Tune Delay, Step=90ps	

Register:: ADC_CTL_BLU[7:0]					**0xD1**
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_CTL_BLU[7]	7	**R/W**	0b0	RGB/YPrPb Clamp (0: RGB 1:YPrPB) //ADC_VBIAS1[6]==0	
ADC_CTL_BLU[6:4]	6:4	**R/W**	0b100	Clamp Voltage (0V~700mV, Step=100mV)	
ADC_CTL_BLU[3]	3	**R/W**	0b0	Offset Depends on Gain (0: RGB Yes, YPrPb No 1:RGB No, YPrPb No)	

ADC_CTL_BLU[2:0]	2:0	**R/W**	0b0	Blue Channel ADC Fine Tune Delay, Step=90ps	

Register:: ADC_SOG_CMP[7:0]				**0xD2**	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_SOG_CMP[7]	7	**R/W**	0	ADC Gain Calibration, OFF PAD to Circuit Path (0: Normal 1: Calibration)	
ADC_SOG_CMP[6]	6	**R/W**	0	~~Reserved~~ Select Absolute VDD: 0: from 3.3V 1: from 1.2V	
ADC_SOG_CMP[5]	5	**R/W**	0	ADC Gain Calibration, Always Enable Clamp Signal (0: ADC_CLAMP[7:0]= from sync processor 1: ADC_CLAMP[7:0]=H)	
ADC_SOG_CMP[4]	4	**R/W**	0	ADC Gain Calibration Voltage Source (0: from RGB+Bandgap, 1: from Gain_Cal_DAC3B) Refer to ADC_SOG_CMP[2:0]	
ADC_SOG_CMP[3]	3	**R/W**	0b0	ADC Gain Calibration Voltage Source of Gain_Cal_DAC3B (0: from Absolute VDD, 1:from Bandgap)	
ADC_SOG_CMP[2:0]	2:0	**R/W**	0b000	**ADC Gain Calibration** **Gain Cal DAC3B Voltage,(Measure method is the same as VMID, VOFFSET)** **Positive side:** **(000:0.4V 001:0.50V 010:0.6V 011:0.7V 100:0.8V 101:0.9V 110:1.00V 111:1.10V)** **Negative side: 0.7V**	

Register:: ADC_DCR_CTRL[7:0]				**0xD3**	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_DCR_CTRL[7]	7	**R/W**	0b0	**Reserved**	
ADC_DCR_CTRL[6]	6	**R/W**	0b0	**Reserved**	
ADC_DCR_CTRL[5]	5	**R/W**	0b0	**Reserved**	
ADC_DCR_CTRL[4]	4	**R/W**	0b0	SOG0 DC Restore Enable(0:Disable 1:Enable)	
ADC_DCR_CTRL[3]	3	**R/W**	0b0	**Reserved**	
ADC_DCR_CTRL[2:1]	2:1	**R/W**	0b00	SF LPF BW control (ADC_CLAMP_CTRL2[5] works together with ADC_DCRESTORE_CTRL[2:1]) If ADC_IBIAS4[3]=0, SF LPF is disabled. If ADC_IBIAS4[3]=1, SF LPF is enabled, the setting is as follows: ADC_CLAMP_CTRL2[5]=0: ADC_DCRESTORE_CTRL[2:1]= 00:73 MHz, 01:158 MHz, 10:250 MHz, 11:340 MHz ADC_CLAMP_CTRL2[5]=1: ADC_DCRESTORE_CTRL[2:1]= 00:38 MHz, 01:63 MHz, 10:96 MHz, 11:131 MHz	
ADC_DCR_CTRL[0]	0	**R/W**	0b0	**Reserved**	

Register:: ADC_CLAMP_CTRL0[7:0]				**0xD4**	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_CLAMP_CTRL0[7:6]	7:6	**R/W**	0b00	R channel Common-mode control ckt selection for PGA input	

				00: op amp type (default), 01: resistor type I, 10: resistor type II, 11: resistor type III	
ADC_CLAMP_CTRL0[5]	5	**R/W**	0b0	**Reserved**	
ADC_CLAMP_CTRL0[4]	4	**R/W**	0b0	SOG0 Clamp Enable(0:Disable 1:Enable)	
ADC_CLAMP_CTRL0[3:2]	3:2	**R/W**	0b00	G channel Common-mode control ckt selection for PGA input 00: op amp type (default), 01: resistor type I, 10: resistor type II, 11: resistor type III	
ADC_CLAMP_CTRL0[1]	1	**R/W**	0b0	PGA LPF BW control (ADC_CLAMP_CTRL0[1], ADC_CTL_RGB[1:0] work together) ADC_CLAMP_CTRL0[1]=0: ADC_CTL_RGB[1:0]= 00:75MHz, 01:150MHz, 10: 250MHz, 11:340MHz ADC_CLAMP_CTRL0[1]=1: ADC_CTL_RGB[1:0]= 00:45MHz, 01:62MHz, 10: 97MHz, 11:122MHz	
ADC_CLAMP_CTRL0[0]	0	**R/W**	0b0	**Reserved**	

Register:: ADC_CLAMP_CTRL1[7:0]				**0xD5**	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_CLAMP_CTRL1[7:6]	7:6	**R/W**	0b00	B channel Common-mode control ckt selection for PGA input 00: op amp type (default), 01: resistor type I, 10: resistor type II, 11: resistor type III	
ADC_CLAMP_CTRL1[5]	5	**R/W**	0b0	**Reserved**	
ADC_CLAMP_CTRL1[4]	4	**R/W**	0b0	ADC 1.2V Generated by Resistor, PS/PD mode (0: Disable, 1: Enable)	
ADC_CLAMP_CTRL1[3]	3	**R/W**	0b0	**ADC Regulator Output Voltage Switch (0: 1.3V not short to ground 1: 1.3V short to ground)** **If use internal ADC Regulator, PS/PD mode set ADC_CLAMP_CTRL1[3]=1**	
ADC_CLAMP_CTRL1[2]	2	**R/W**	0b0	**SOG0 Bias Current Source (0: from POR, 1: from ADC)**	
ADC_CLAMP_CTRL1[1]	1	**R/W**	0b0	SOG0 clamp -300mV (0: noraml clamp 1:clamp -300m) //IR	
ADC_CLAMP_CTRL1[0]	0	**R/W**	0b0	**Bias Current of SOG (0: Normal Mode, 1: Low Standby Mode)**	

Register:: ADC_CLAMP_CTRL2[7:0]				**0XD6**	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
ADC_CLAMP_CTRL2[7:6]	7:6	**R/W**	0b01	SOG0 DC restore resister (00:open 01:500k 10:1M 11:5M) 500k/1M are Poly R	
ADC_CLAMP_CTRL2[5]	5	**R/W**	0b0	SF LPF BW control (ADC_CLAMP_CTRL2[5] works together with ADC_DCRESTORE_CTRL[2:1]) If ADC_IBIAS4[3]=0, SF LPF is disabled. If ADC_IBIAS4[3]=1, SF LPF is enabled, the setting is as follows: ADC_CLAMP_CTRL2[5]=0: ADC_DCRESTORE_CTRL[2:1]= 00:73	

				MHz, 01:158 MHz, 10:250 MHz, 11:340 MHz ADC_CLAMP_CTRL2[5]=1: ADC_DCRESTORE_CTRL[2:1]= 00:38 MHz, 01:63 MHz, 10:96 MHz, 11:131 MHz	
ADC_CLAMP_CTRL2[4]	4	**R/W**	0b01	Reserved	
ADC_CLAMP_CTRL2[3]	3	**R/W**	1	RGB Input Range Adjust 0: 0.5V-1.0V, 1: 0.25-1.25V,	
ADC_CLAMP_CTRL2[2]	2	**R/W**	0	Red channel clamp to top (0: noraml 1: top)	
ADC_CLAMP_CTRL2[1]	1	**R/W**	0	Green channel clamp to top (0: noraml 1: top)	
ADC_CLAMP_CTRL2[0]	0	**R/W**	0	Blue channel clamp to top (0: noraml 1: top)	

Register::ADC_SOG_DAC_SOY_CONTROL[7:0]				**0xD7**	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
Reserved	7:6	---	0b00	Reserved	
ADC_SOG0_DAC[5:0]	5:0	R/W	0b100000	SOG0 DAC input	

Address:D8 **PTNPOS_H** **Default: 00h**

Bit	Mode	Function
7:4	**R/W**	**Test Pattern V Position Register [11:8]** Assign the test pattern digitized position in line after V_Start.
3:0	**R/W**	**Test Pattern H Position Register [11:8]** Assign the test pattern digitized position in pixel after H_Start.

Address: D9 **PTNPOS_V_L**

Bit	Mode	Function
7:0	R/W	**Test Pattern V Position Register [7:0]** Assign the test pattern digitized position in line after V_Start..

Address:DA **PTNPOS_H_L**

Bit	Mode	Function
7:0	R/W	**Test Pattern H Position Register [7:0]** Assign the test pattern digitized position in line after H_Start..

Use PTNPOS to assign the pixel position after HSYNC leading edge that input signal digitized. Each time the PTNPOS is written, the digitized results will be loaded into PTNRD, PTNGD and PTNBD. For test issue, make the input signal a fixed pattern before PTNPOS is written. Then the same digitized output will be got.

Address: DB **PTNRD**

Bit	Mode	Function
7:0	R	**Test Pattern Red-Channel Digitized Result.**

Address: DC **PTNGD**

Bit	Mode	Function
7:0	R	**Test Pattern Green-Channel Digitized Result.**

Address:DD **PTNBD**

Bit	Mode	Function
7:0	R	**Test Pattern Blue-Channel Digitized Result.**

Address:DE **TEST_PATTERN_CTRL** **Default: 00h**

Bit	Mode	Function
7	**R/W**	**Enable Test** 0: Finish (and result sequence is R-G-B) (Default) 1: Start
6:0	**--**	**Reserved to 0**

Embedded LDO(Page 0)

Register:: EBD_REGLATOR_DB_CTRL				0XE1	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
POWLDOX_L	7	R/W	1	**LDO_XTAL enable:** 0: disable LDO(output tri-state) 1: enable LDO (default value)	
TUNELDOX_VREF_L	6:4	R/W	0x4	LDO_XTAL output voltage control： 000：2.00V 001：1.65V 010：1.70V 011：1.75V 100：1.80V (default) 101：1.85V 110：1.90V 111：1.95V	
Reserved	3:0	R/W	0x0	Reserved	

Note: all Page6 registers reset to default value only when external reset/por happen

ABL(Page 0)

Address: E2 **AUTO_BLACK_LEVEL_CTRL1** **Default: 10h**

Bit	Mode	Function
7	R/W	**ABL Mode** 0: RBG (Default) 1: YPbPr
6	R/W	**On-line/Off-line ABL Mode** 0: Off-line (Default) 1: On-line
5:4	R/W	**Width of ABL region in each line** 00: 16 pixels 01: 32 pixels (Default) 10: 64 pixels 11: 4 pixels
3	R	**R/Pr Channel ABL Result (write clear)** 0: not equal 1: equal (Black Level = Target Value) On-line mode: |Black Level - Target Value| <=LOCK_MGN Off-line mode: |Black Level - Target Value| <= EQ_MGN
2	R	**G/Y Channel ABL Result (write clear)** 0: not equal 1: equal (Black Level = Target Value) On-line mode: |Black Level - Target Value| <= LOCK _MGN Off-line mode: |Black Level - Target Value| <= EQ_MGN
1	R	**B/Pb Channel ABL Result (write clear)** 0: not equal 1: equal (Black Level = Target Value) On-line mode: |Black Level - Target Value| <= LOCK _MGN Off-line mode: |Black Level - Target Value| <= EQ_MGN
0	R/W	**Auto Black Level Enable (write 0 force stop)** 0: Finished/Disable (Default) 1: Enable to start ABL, auto cleared after finished Cleared to 0 when off-line mode completes.

- Parameters can only be changed when EN_ABL is 0
- The on-line mode never stops unless EN_ABL is 0.
- Off-line mode completes when MAX_FRAME is measured or the result is equal.
- ABL must be disabled before switching On-line/Off-line mode and then enable again.

Address: E3 **AUTO_BLACK_LEVEL_CTRL2** **Default: 08h**

Bit	Mode	Function
7:5	R/W	**Line averaged for each ABL adjustment** 000 : 2 (Default) 001 : 4 010 : 8 011 : 16 100 : 32 101 : 64 110: Reserved 111: Reserved
4:0	R/W	**Start Vertical Position of ABL in each line** Determine the start line of auto-black-level after the leading edge of Vsync

Address: E4 **AUTO_BLACK_LEVEL_CTRL3** **Default: 10h**

Bit	Mode	Function
7:4	R/W	**Y/R/G/B Target value** 0000: 1 0001: 2 (Default) 0010: 3 0011: 4

		…. 1111:16 (Pb/Pr Target level is fixed 128)
3:2	R/W	**Lock Margin** 00: 1 (Default) 01: 2 10: 4 11: 6
1:0	R/W	**End Vertical Position of ABL measurement region [9:8]** Determine the last line of auto-black-level measurement for every frame/field countd by double line

- Off-line mode rule:
  Measures once for each field / frame, and the offset is the delta.
- On-line mode rule:
  If (delta <= EQ_MGN) offset = 0
  Else if (delta < L_MGN) offset = +/-1
  Else offset = +/-L_MGN
- ADC offset is updated immediately.

Address: E5 **AUTO_BLACK_LEVEL_CTRL4** **Default: 0Ah**

Bit	Mode	Function
7:0	R/W	**End Vertical Position of ABL measurement region [7:0]** Determine the last line of auto-black-level measurement for every frame/field counted by double line.

- Note: ABL will fail if End Vertical Position < Start Vertical Position + Average Line(CRC1[7:6])

Address: E6 **AUTO_BLACK_LEVEL_CTRL5** **Default: 80h**

Bit	Mode	Function
7:0	R/W	**Start Position of ABL in Each Line** Determine the start position of auto-black-level after the trailing edge of reference signal. (When ABL mode in YPbPr, the reference signal is input Hsync. In RGB mode, the reference signal is clamp signal.) High Byte at P0_CREF[3:0]

- In each region, hardware compare the average value in the target region (fixed 16 input pixels after start position of ABL) with target value and add +1/-1 or +L_MGN /- L_MGN to ADC offset. (+ for greater than target value, - for smaller than target value).

Address: E7 **AUTO_BLACK_LEVEL_CTRL6** **Default: F8h**

Bit	Mode	Function
7:6	R/W	**Large Error Margin (L_MGN) (For on-line Mode)** 00: 2 01: 4 10: 6 11: 8 (Default)
5:4	R/W	**Max. Frame/Field Count (For off-line mode)** 00: 4 01: 5 10: 6 11: 7 (Default)
3	R/W	**Setting Width of ABL region in each line be multiplied (P0_CRE2[5:4])** 0 : x1 1 : x4 (Default)
2:0	R/W	**Lines delayed between each measurement region (For on-line Mode)** 000: 16 (Default) 001: 32 010: 64 011: 128 100: 192 101: 256 110: 384 111: 640

Address: E8 **AUTO_BLACK_LEVEL_CTRL7** **Default: 60h**

Bit	Mode	Function

7	--	**Reserved**
6	R/W	**Equal Condition (Off-line mode)** 0: To trigger status until measurement achieve Max Frame/Field Count. Reference by P0_CRE7[5:4]. 1: To trigger status once if |Black Level - Target Value| <= EQ_MGN. When Equal Condition is met, ABL will stop and auto clear P0_CRE2[0]. (Default)
5	R/W	**Measure Pixels Method** 1: Minimum value (Default) 0: Average value
4	R/W	**Measure Error Flag Reset** 0: Normal 1: Reset
3	R	**Measure Error Flag** 0: Normal 1: Error (This flag is occurred when Hsync trailing edge is met during measurement.)
2	R/W	**Hsync Start Reference Select** 0: HS leading edge (Default) 1: HS trailing edge
1:0	R/W	**Equal margin (EQ_MGN)** 00: 0 (Default) 01: 1 10: 2 11: 3

Address: E9 **AUTO_BLACK_LEVEL_RED_VALUE**

Bit	Mode	Function
7:0	R	**Minimum/Average Value of Red Channel in Test Mode (only show MSB 8bit.)**

Address: EA **AUTO_BLACK_LEVEL_GREEN_VALUE**

Bit	Mode	Function
7:0	R	**Minimum/Average Value of Green Channel in Test Mode (only show MSB 8bit.)**

Address: EB **AUTO_BLACK_LEVEL_BLUE_VALUE**

Bit	Mode	Function
7:0	R	**Minimum/Average Value of Blue Channel in Test Mode (only show MSB 8bit.)**

Address: EC **AUTO_BLACK_LEVEL_NOISE_VALUE_OF_RED_CHANNEL**

Bit	Mode	Function
7:0	R	**Noise Value of Red Channel in Test Mode after Equal status is triggered. (only show MSB 8bit.)**

Address: ED **AUTO_BLACK_LEVEL_NOISE_VALUE_OF_GREEN_CHANNEL**

Bit	Mode	Function
7:0	R	**Noise Value of Green Channel in Test Mode after Equal status is triggered. (only show MSB 8bit.)**

Address: EE **AUTO_BLACK_LEVEL_NOISE_VALUE_OF_BLUE_CHANNEL**

Bit	Mode	Function
7:0	R	**Noise Value of Blue Channel in Test Mode after Equal status is triggered. (only show MSB 8bit.)**

Address: EF **AUTO_BLACK_LEVEL_CTRL8** **Default: 01h**

Bit	Mode	Function
7:4	--	**Reserved**
3:0	R/W	**Start Position of ABL in Each Line [11:8]** Determine the start position of auto-black-level after the trailing edge of reference signal. (When ABL mode in YPbPr, the reference signal is input Hsync. In RGB mode, the reference signal is clamp signal.) Low Byte at P0_CRE6[7:0]

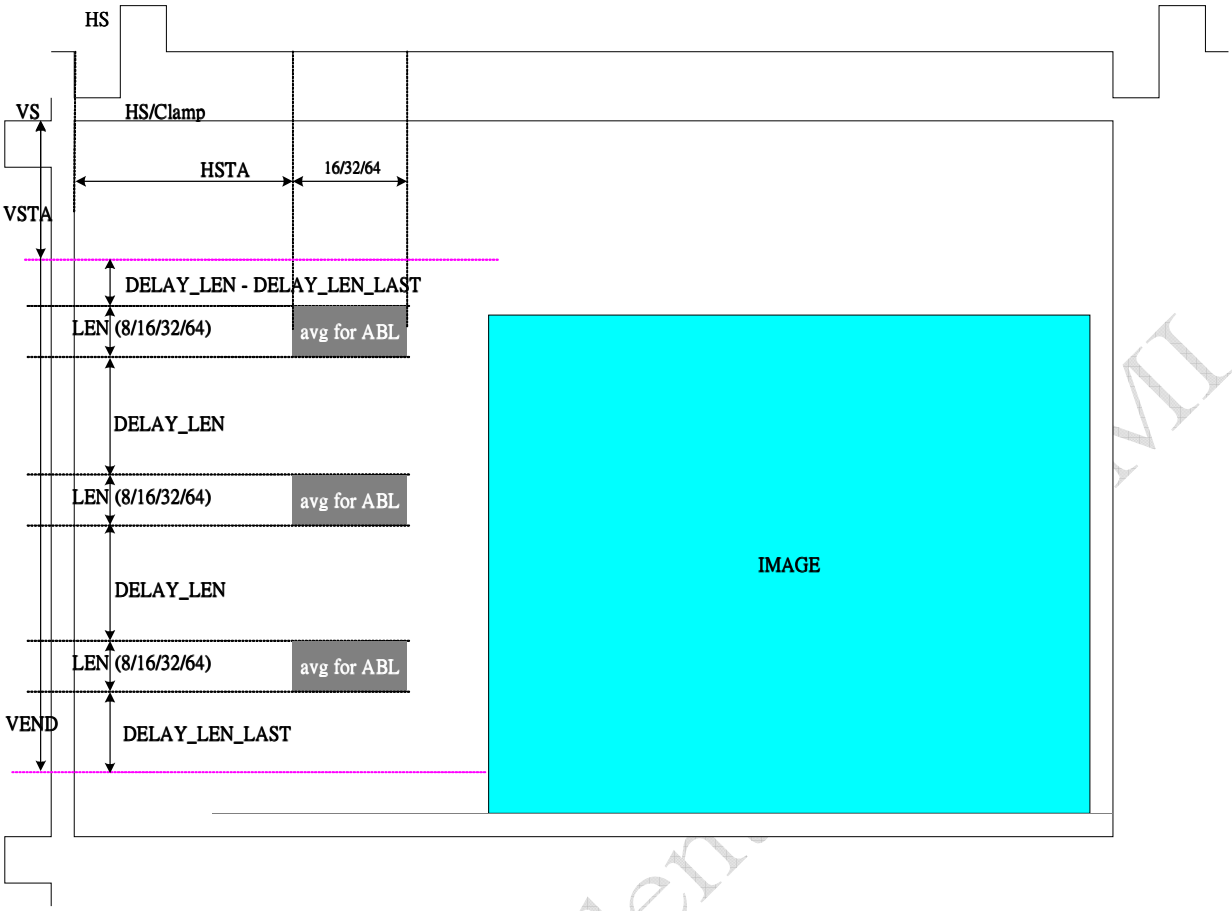


Figure-1: Auto Black Level active region – case 1

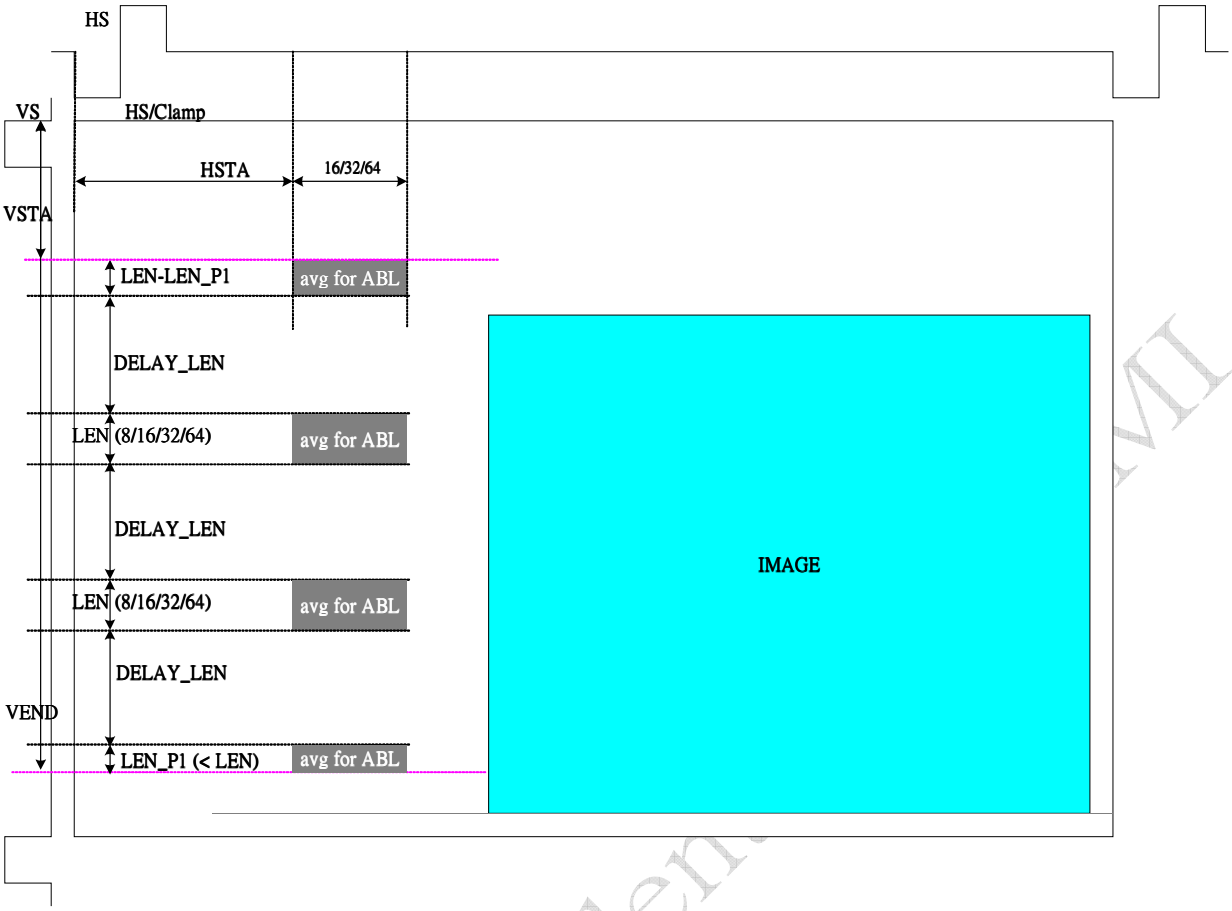


Figure-2: Auto Black Level active region – case 2

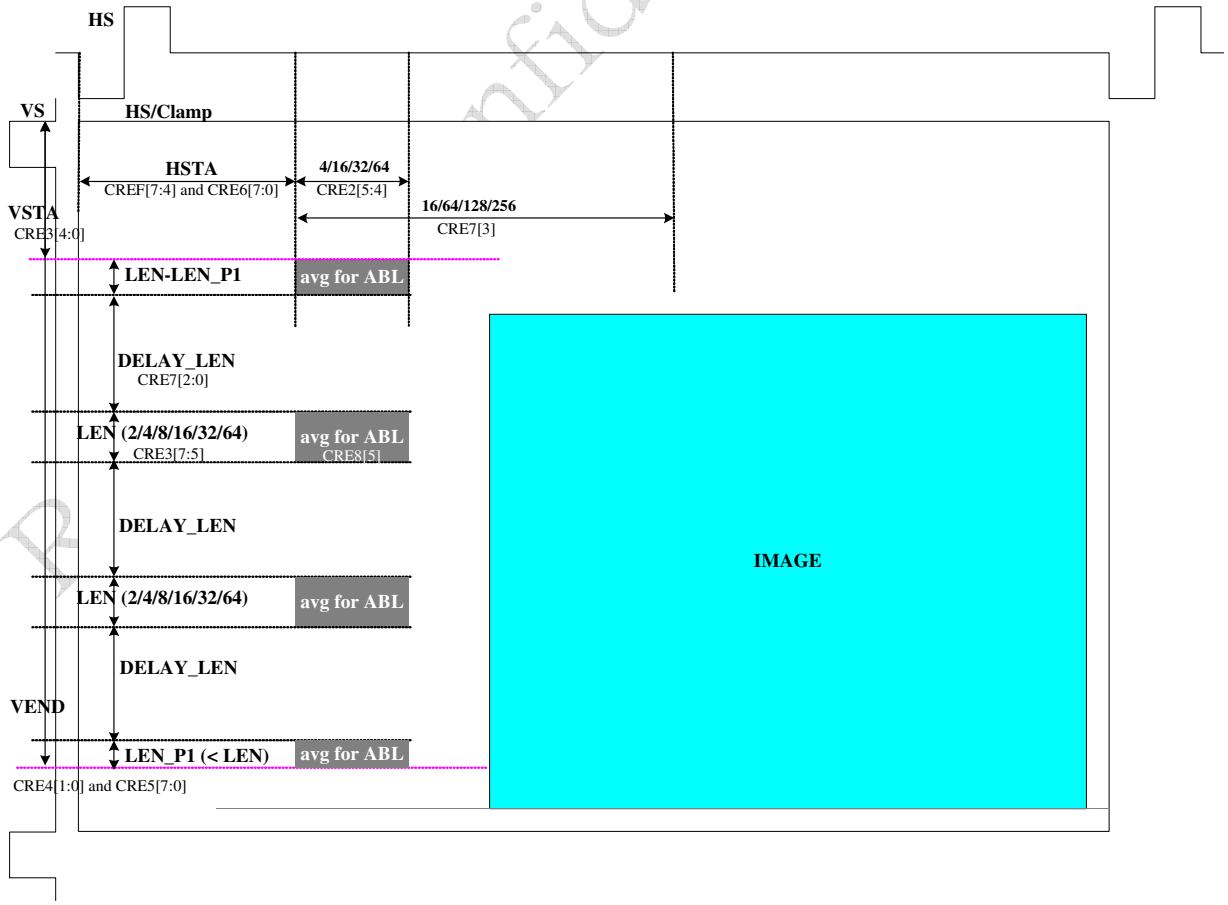
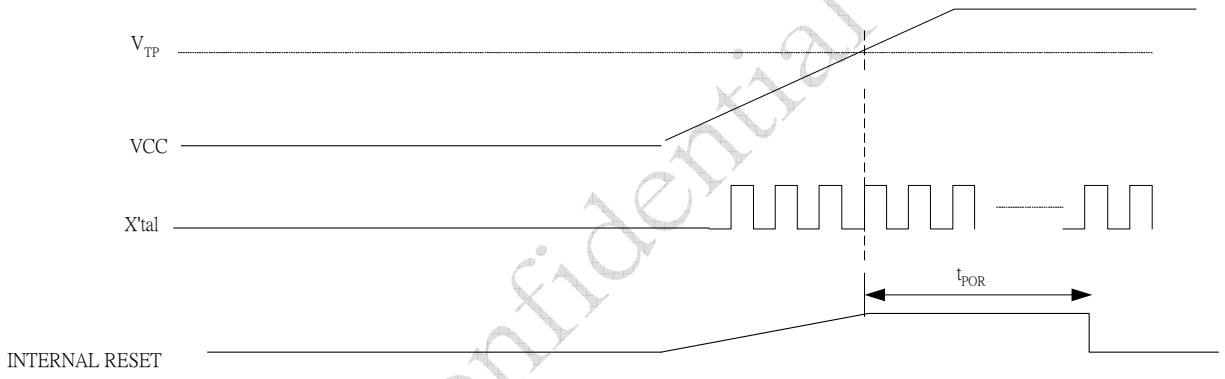


Figure-3: Auto Black Level active region – case 3

LVR(Page 0)

Register:: POWER_ON_RESET			0xF3		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Porvset	7:6	R/W	0x3	**Negative Threshold Value For Power on Reset** (POR reset time=160ms) (VTP=2.8V) 00:2.4V 01: 2.6V 10: 2.5V 11:2.7V(Default)	
Pormcuvset	5:4	R/W	0x0	**PORMCUVSET (LVR Threshold Voltage) (VTN=0.66, VTP=0.88)** **(Core LVR voltage: tied to 2'b00 by hardware)**	
Reserved	3	R/W	0x0	**Reserved**	
Reserved	2	R/W	0x0	**Reserved**	
Adc_rglt_vol	1:0	R/W	0x0	**XI/XO Pad Driving** 00: Strong (default) 01: Medium 10: Medium 11: Weak	



V_{TP}: 2.8v

t_{por}: 65536*16*4* X'tal cycles

Schmitt trigger (Page 0)

Register:: HS_SCHMITT_TRIGGE_CTRL			0xF4		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Hsync3ven	7	R/W	0x0	**HSYNC Schmitt Power Down (Only for Schmitt trigger new mode)** 0: Power down (Default) 1: Normal	
Hsyncpol	6	R/W	0x1	**Polarity Select** 0: Negative HSYNC (high level) 1: Positive HSYNC (low level) (Default)	
Hsyncmode	5	R/W	0x0	**Schmitt Trigger Mode** 0: Old mode 1: New mode	
Hsyncvos	4	R/W	0x0	**Threshold Voltage Fine Tune (only for Schmitt trigger new mode)** 0: 0V (Default) 1: -0.1V	
Hsyncvtp	3:2	R/W	0x0	**Positive Threshold Voltage**	
Hsyncvtn	1:0	R/W	0x1	**Negative Threshold Voltage**	

- There is a mode of the HSYNC Schmitt trigger.
  1. New mode: Fully programmable Schmitt trigger.
     The following table will determine the Schmitt Trigger positive and negative voltage:

bit[6]=1 (Positive HSYNC)				**bit[6] = 0** (Negative HSYNC)			
bit[3:2]	**V_t^+**	**bit[1:0]**	**V_t^-**	**bit[3:2]**	**V_t^+**	**bit[1:0]**	**V_t^-**
00	1.4V	00	V_t^+ - 1.2V	00	1.8V	00	V_t^+ - 1.2V
01	1.6V	01	V_t^+ - 1.0V	01	2.0V	01	V_t^+ - 1.0V
10	1.8V	10	V_t^+ - 0.8V	10	2.2V	10	V_t^+ - 0.8V
11	2.0V	11	V_t^+ - 0.6V	11	2.4V	11	V_t^+ - 0.6V

- After we get the threshold voltage by the table, we still can fine tune it:
  Final Positive Threshold Voltage = V_t^+ - 0.1* bit[4]
  Final Negative Threshold Voltage = V_t^- - 0.1* bit[4]

ADC PLL (Page 1)

Address: A0 **PLL_DIV_CTRL** **Default: 08h**

Bit	Mode	Function
7	R/W	**DDS Tracking Edge** 0: HS positive edge (Default) 1: HS negative edge
6	R/W	Tracking direction inversion 0: if HS leads HSFB => phase lead => m, k ↑ (Default) 1: if HS lags HSFB => phase lag => m, k ↓
5:4	R/W	Waiting HS lines to start counting divider for Fast Lock function 00: 4 (default) 01: 3 10: 2 11: 1
3:2	R/W	**Delay Compensation Mode** 00: Mode 0 No delay from PLL phase0 to DDS pfd input 01: Mode 1 Delay the path from PLL phase0 to DDS pfd input to be around 4.2 ns 10: Mode 2 (default) Delay the path from PLL phase0 to DDS pfd input to be around 4.6 ns 11: Mode 3 Delay the path from PLL phase0 to DDS pfd input to be around 5 ns
1	R/W	**Reserved to 0**
0	R/W	**Reserved to 0**

Address: A1 **I_CODE_M** **Default: 01h**

Bit	Mode	Function
7	R/W	**Reserved to 0**
6:0	R/W	**I_CODE[14:8]**

Address: A2 **I_CODE_L** **Default: 04h**

Bit	Mode	Function
7:0	R/W	**I_CODE[7:0]**

Address: A3 **P_CODE** **Default: 20h**

Bit	Mode	Function
7:0	R/W	**P_CODE[7:0]**

Address: A4 **PFD_CALIBRATED_RESULTS** **Default: 8'b0xxxxxxx**

Bit	Mode	Function
7	R/W	**PFD Calibration Enable** (auto clear when finished) Overwrite 0 to 1 return a new PFD calibrated value.
6:4	R/W	**Reserved to 0**
3:0	R	**PFD Calibrated Results [11:8]**

Address: A5 **PFD_CALIBRATED_RESULTS** **Default: 8'bxxxxxxxx**

Bit	Mode	Function
7:0	R	**PFD Calibrated Results [7:0]**

Address: A6 **PE_MEARSURE** **Default: 8'b0xxxxxxx**

Bit	Mode	Function
7	R/W	**PE Measure Enable** (auto clear when finished) 0: Disable (Default) 1: Start PE Measurement, clear after finish.
6:4	R/W	Reserved to 0
3:0	R	**PE Value Result [11:8]**

Address: A7 **PE_MEARSURE** **Default: 8'bxxxxxxxx**

Bit	Mode	Function
7:0	R	**PE Value Result [7:0]**

Address: A8 **PE_MAX_MEASURE** **Default: 8'b0xxxxxxx**

Bit	Mode	Function
7	R/W	**PE Max. Measure Enable** 0: Disable (Default) 1: Start PE Max. Measurement
6:4	R/W	Reserved to 0
3:0	R	**PE Max Value [11:8]**

Address: A9 **PE_MAX_MEASURE** **Default: 8'bxxxxxxxx**

Bit	Mode	Function
7:0	R	**PE Max Value [7:0]**

Address: AA **FAST_PLL_CTRL** **Default: 00h**

Bit	Mode	Function
7	R/W	**PE Max. Measure Clear** 0: clear (Default) 1: write '1' to clear PE Max. Value
6	R/W	**Enable APLL Setting** 0: Disable (Default) 1: Enable (Auto clear when finished) When CRAA[5] enabled, enable this bit will write P_CODE, I_CODE, PLL M/N, PLL K, PLLDIV and DDS SUM_I at the end of input vertical data enable
5	R/W	**Enable Fast PLL Mechanism** 0: Disable (Default) 1: Enable (Auto clear when finished)
4	R/W	**Force APLL Setting Enable** **Force to write PLL M/N, K, PLLDIV and SUM_I while got no V_ACTIVE signal** 0: Disable (Default) 1: Enable (Auto clear when finished)
3	R/W	**DDS SUM_I Setting Updated Enable** 0: Disable (Default) 1: Enable (Auto clear when finished)
2	R/W	**Measure SUM_I** 0: Disable 1: Enable (Auto clear after finish)
1	R/W	**Enable Port AB** 0: Disable Port **AB** Access 1: Enable Port **AB** Access When this bit is 0, port address will be reset to 00, and will auto increase when read or write
0	R/W	**Select SUM_I for Read** 1: Select SUM_I_NOW [26:0] for read 0: Select SUM_I_PRE [26:0] for read

Address: AB **FAST_PLL_SUM_I**

Bit	Mode	Function
7:0	R/W	**SUM_I_PRE (Auto Increase)** **1st [00000, SUM_I [26:24]]** **2nd SUM_I [23:16]** **3rd SUM_I [15:8]** **4th SUM_I [7:0]**

SUM_I [26] is the signed bit

The operation steps are as following:
SUM_I Access Port Indexing=0,
SUM_I Access Port Indexing=1,
SUM_I selection =1, **Fast Lock Function=1**

Latch SUM_I_NOW=1

Read SUM_I_NOW from SUM_I_ACCESS_PORT for 4 times:
SUM_I_NOW [26:24]
SUM_I_NOW [23:16]
SUM_I_NOW [15:8]
SUM_I_NOW [7:0]
Calculate new freq. SUM_I_PRE and write to SUM_I_ACCESS_PORT for 4 times:
SUM_I_PRE [26:24]
SUM_I_PRE [23:16]
SUM_I_PRE [15:8]
SUM_I_PRE [7:0]
SUM_I_PRE_SET =1
Write PLL2 M/N code and DDS feed back divider
Write New P/I code
Setting Auto Load =1
Wait for next frame start or polling Reg [2E].6

Address: AC **PLL_M (M Parameter Register)** **Default: 19h**

Bit	Mode	Function
7:0	R/W	**PLLM[7:0] (PLL DPM value – 3)**

Address: AD **PLL_N (N Parameter Register)** **Default: 20h**

Bit	Mode	Function
7:4	R/W	**PLLSPHNEXT[3:0] (K) (default is 0000)**
3	R/W	**PLLSNBP** 0: N is followed by the value of REGAD [3:1] 1: N is always 1
2:0	R/W	**PLLN[2:0] (PLL DPN value – 2) (default is 000)** **It is supposed to be always bigger than 2**

- PLL1_N modify to only 4-bit.
- Assume PLL1_M=0x0B, P1M=0x0B+3=14; PLL1_N=0x03, P1N=0x03+2=5; K=7; F_IN = 24.576MHz. F_PLL = F_IN x ((P1M+7/16) / P1N) = 24.576 x 14.4375 / 5 = 70.9632MHz
- If the target frequency is F_ADC, the constraint of F_PLL is *(M+-7/16)/N * XTCLK < F_PLL1 < (M+8/16)/N *XTCLK*
- Although the new dds provides +15/-16 phase margin for tracking. However it is better not to set M, N and K to be some freq. that PLL has to swallow +15/-16 phases. Because under that condition, SDM will get saturation problem.
- For NO shrink IC => PLLN setting will have no limitation
- For shrink IC and timing factor predicted as 0.8 => crystal clock 27 MHZ => PLLN can't be 0 while APLL VCO is lower than 167MHZ
  crystal clock 24.576 MHZ => PLLN can't be 0 while APLL VCO is lower than 84 MHZ
- For shrink IC and timing factor predicted as 0.9 => crystal clock 27 MHZ => PLLN can't be 0 while APLL VCO is lower than 74 MHZ
  crystal clock 24.576 MHZ => PLLN can't be 0 while APLL VCO is lower than 52 MHZ

Address: AE **PLL_CRNT (PLL Current/Resistor Register)** **Default: 6Fh**

Bit	Mode	Function
7:5	R/W	**PLLVR [2:0] (PLL Loop Filter Resister Control)** 000: 7K 001: 9.5K 010: 12K 011: 14.5K(Default) 100: 17K 101: 19.5K 110: 22K 111: 24.5K
4:0	R/W	**PLLSI [4:0] (PLL Charger Pump Current IchDpll) (Default: 00011b)** Icp = 2.5uA+2.5uA*bit[0]+5uA*bit[1]+10uA*bit[2]+20uA*bit[3]+30uA*bit[4]

- **Keep Icp/DPM constant**

Address: AF **PLL_WD (PLL Watch Dog Register)** **Default: 09h**

Bit	Mode	Function
7	R	**PLLSTATUS (PLL WD Status)** 0: Normal (Default) 1: Abnormal

6	R/W	**PLLWDRST (PLL WD Reset)** 0: Normal (Default) 1: Reset
5	R/W	**PLLWDSET (PLL WD Set)** 0: Normal (Default) 1: Set
4:3	R/W	**PLLWDVSET[1:0] (PLL WD Voltage Set)** 00: 2.46V 01: 1.92V(Default) 10: 1.36V 11: 1.00V
2	R/W	**HS_dds2synp latch edge** 0: falling edge (Default) 1: rising
1	R/W	**Reset DDS** 0: normal (Default) 1: reset whole DDS
0	R/W	**PLLPWDN (PLL Power Down)** 0: Normal Run 1: Power Down (Default)

- **HSFB_dds2synp & HS_dds2synp will be both sampled by AF [2]**

Address: B0 **PLL_MIX** **Default: 8'b0000_000x**

Bit	Mode	Function
7	R/W	**PLLSVR3**
6	---	**Reserved to 0**
5	R/W	**PLLSVC3**
4	---	**Reserved to 0**
3	---	**Reserved to 0**
2:1	R/W	**ADCKMODE [1:0] (ADC Input Clock Select Mode)** 00: Single Clock Mode (Default) 01: Single Inverse-Clock Mode 10: External Clock Mode 11: Dual Clock Mode (1x and 2x Clock)
0	R	**Swallow phase enable (K mask disabled)** **The pll can't enable swallow phase function while pll just be power up. Waiting for 64 clock cycles then start to enable phase swallow function.** **While power down, the counter will be reset. While power up, the counter start to work**

Address: B1 **PLLDIV_H** **Default: 45h**

Bit	Mode	Function
7	**---**	**Reserved to 0**
6	**R/W**	**Phase_Select_Method** 0: Manual 1: Look-Up-Table (default)
5	**R/W**	**PLLPH0PATH** 0: Short Path (Default) 1: Long Path (Compensate PLL_ADC path delay)
4	**R/W**	**PLLD2** 0:ADC CLK=1/2 VCO CLK (Default) 1:ADC CLK=1/4 VCO CLK
3:0	**R/W**	**PLL Divider Ratio Control. High-Byte [11:8]. (Default: 5h)**

Address: B2 **PLLDIV_L** **Default: 2Eh**

Bit	Mode	Function
7:0	R/W	**PLL Divider Ratio Control**. **Low-Byte [7:0]**. PLLDIV should be double buffered when PLLDIV_LO changes and IDEN_STOP occurs.

- This register determines the number of output pixel per horizontal line. PLL derives the sampling clock and data output clock (DCLK) from input HSYNC. *The real operation Divider Ratio = PLLDIV+1*

- The power up default value of PLLDIV is 053Fh(=1343, VESA timing standard, 1024x768 60Hz, Horizontal time).
- The setting of PLLDIV must include sync, back-porch, left border, active, right border, and front-porch times.
- Control-Register B1 & B2 will filled in when Control-Register B2 is written.

Address: B3 **PLLPHASE_CTRL0 (Select Phase to A/D)** **Default: 30h**

Bit	Mode	Function
7	R/W	**PLLD2X control** (Default=0)
6	R/W	**PLLD2Y control** (Default=0)
5	R/W	**PLLX (PLL X Phase control)** (Default=1)
4	R/W	**PLLY (PLL X Phase control)** (Default=1)
3:0	R/W	**PLLSCK [4:1] (PLL 32 Phase Pre-Select Control)** (Default=0h)

Address: B4 **PLLPHASE_CTRL1 (Select Phase to A/D)** **Default: 00h**

Bit	Mode	Function
7	R/W	**PLLSCK [0] (PLL 32 Phase Pre-Select Control)** (Default=0)
6	R/W	**MSB of 128 phase** (Only for ADC CLK=1/4 VCO CLK) (Default=0)
5:0	R/W	**Phase Select the index of Look-Up-Table[5:0]** (Default=0)

- **When Phase_Select_Method=1, Phase is selected by CRB4[6:0].**
- **When Phase_Select_Method=0, PLLD2X, PLLD2Y, PLLX, PLLY, PLLSCLK[4:0] Should be double buffered when PLLSCK[0] is updated**

Address: B5 **PLL_PHASE_INTERPOLATION** **Default: 50h**

Bit	Mode	Function
7:6	R/W	**PLL Phase Interpolation Control Load (Default: 01)**
5:3	R/W	**PLL Phase Interpolation Control Source (Default: 010)**
2:1	R/W	**PLL Add Phase Delay** 00: Original phase selected by X,Y and 16-phase pre-select 01-11: Add 1-3 delay to Original phase selected by X,Y and 32-phase pre-select
0	R/W	**Reserved to 0**

Phase	[XY^^^^^]	Phase	[XY ^^^^^]	Phase	[XY ^^^^^]	Phase	[XY ^^^^^]
0	[11 00000]	**16**	[01 10000]	**32**	[10 00000]	**48**	[00 10000]
1	[11 00001]	**17**	[01 10001]	**33**	[10 00001]	**49**	[00 10001]
2	[11 00010]	**18**	[01 10010]	**34**	[10 00010]	**50**	[00 10010]
3	[11 00011]	**19**	[01 10011]	**35**	[10 00011]	**51**	[00 10011]
4	[11 00100]	**20**	[01 10100]	**36**	[10 00100]	**52**	[00 10100]
5	[11 00101]	**21**	[00 10101]	**37**	[10 00101]	**53**	[00 10101]
6	[11 00110]	**22**	[00 10110]	**38**	[10 00110]	**54**	[00 10110]
7	[11 00111]	**23**	[01 10111]	**39**	[10 00111]	**55**	[00 10111]
8	[11 01000]	**24**	[01 11000]	**40**	[10 01000]	**56**	[00 11000]
9	[11 01001]	**25**	[01 11001]	**41**	[10 01001]	**57**	[00 11001]
10	[01 01010]	**26**	[10 11010]	**42**	[10 01010]	**58**	[11 11010]
11	[01 01011]	**27**	[10 11011]	**43**	[10 01011]	**59**	[11 11011]
12	[01 01100]	**28**	[10 11100]	**44**	[00 01100]	**60**	[11 11100]
13	[01 01101]	**29**	[10 11101]	**45**	[00 01101]	**61**	[11 11101]
14	[01 01110]	**30**	[10 11110]	**46**	[00 01110]	**62**	[11 11110]
15	[01 01111]	**31**	[10 11111]	**47**	[00 01111]	**63**	[11 11111]

Address: B6 **P_CODE mapping methods** **Default: 18h**

Bit	Mode	Function
7:6	R/W	**Mapping method:** 00: normal mapping P_CODE x G value (default) 01: nonlinear mapping I smaller than Q(PE) 2 4 8 16 32 64 P_CODE x 1 2 4 8 32 128 128 10: nonlinear mapping II P_CODE x 1 2 2 8 32 256 256 11: nonlinear mapping III P_CODE x 1 2 8 16 32 128 512
5:2	R/W	**G value** 0000: 0

		0001: 1 0010: 4 0011: 16 0100: 64 0101: 128 0110: 256 (default) 0111: 512 1000: 1/4 1001: 1/16 1010: 1/64 1011: reserved to 0 1100: reserved to 0 1101: reserved to 0 1110: reserved to 0 1111: reserved to 0
1	R/W	***Adaptive tracking enable for I_CODE*** 0: disable to use adaptive I_CODE (default) 1: enable to use adaptive I_CODE
0	R/W	***Adaptive tracking enable for P_CODE*** 0: disable to use adaptive P_CODE (default) *1: enable to use adaptive P_CODE*

Address: B7 **PE tracking method** **Default: 02h**

Bit	Mode	Function
7:6	R/W	Threshold value of Q (PE) to decide if starting adaptive tracking 00: 2 (default) 01: 4 10: 8 11: 15
5:4	R/W	Threshold times to decide if starting adaptive tracking while Q(PE) < Threshold value successively 00: 3 (default) 01: 7 10: 11 11: 15
3	R/W	Mask high speed testing pins (test1out, test2out, fav4) 0: normal 1: mask
2	R/W	Adaptive tracking enable => refer to B6 [1:0] to decide if I_CODE or P_CODE enables adaptive tracking or not 0: disable (default) 1: enable
1:0	R/W	Decrease ratio for adaptive tracking Adaptive tracking will be enabled while getting Q (PE) <=2 for over 8 times, and it will be triggered only under delay-chain mode 00: 1/2 01: 1/4 10: 1/8 (default) 11: 1/16

Address: B8 **DDS_MIX_1** **Default: 06h**

Bit	Mode	Function
7:6	R	DDS tracking state [1:0] 00: not lock 01: lock 10: unlock but not using new tracking mode yet 11: unlock & using new tracking mode
5:4	R/W	Reserved to 0
3:1	R/W	Judge threshold lock already => while Q (PE) keep smaller than threshold for 32 HS 000: 2 001: 4 010: 6 011: 8 (default)

		100: 16 101: 32 110: 64 111: 120
0	R	PLL lock already 0: not lock already 1: lock already

Address: B9 **DDS_MIX_2** **Default: 00h**

Bit	Mode	Function
7:0	R/W	**P_code_max[16:9]** **Set p_code_max value to clamp the GAIN of APLL**

Address: BA **DDS_MIX_3** **Default: 00h**

Bit	Mode	Function
7:0	R/W	**P_code_max[8:1]** **Set p_code_max value to clamp the GAIN of APLL**

Address: BB **DDS_MIX_4** **Default: 1Bh**

Bit	Mode	Function
7	R/W	**P_code_max[0]** **Set p_code_max value to clamp the GAIN of APLL**
6	R/W	**New mode enable** **0: disable new mode tracking (default)** **1: enable new mode tracking**
5:3	R/W	**New mode enable threshold** **000: 8** **001: 20** **010: 60** **011: 120 (default)** **100: 200** **101: 450** **110: 800** **111: 1200**
2:0	R/W	**New mode lock threshold=> while Q (PE) keep smaller than threshold for 32 HS** **000: 2** **001: 4** **010: 6** **011: 8 (default)** **100: 16** **101: 32** **110: 64** **111: 120**

- **New mode enable threshold should be larger than new mode lock threshold, otherwise, the track state will always be at lock state and new mode function will not be enabled while new mode enable threshold < Q (PE) < new mode lock threshold**

Address: BC **DDS_MIX_5** **Default: A0h**

Bit	Mode	Function
7:6	R/W	**Delay chain length select (only valid while new mode enable and track state is 01 10 11)** **00: cnt=7 => 59.6ns** **01: cnt=15 => 117ns** **10: cnt=23 => 184.4ns (default)** **11: cnt=31 => 246.8ns**
5:4	R/W	**Phase error sample period choose (only valid while new mode enable and track state is 01 10 11)** **00: every 1 cycle sample** **01: every 2 cycle sample** **10: every 3 cycle sample (default)** **11: every 4 cycle sample**
3	R/W	**Delay chain reset period select** **0: short reset (2ns) (default)** **1: long reset (1 fbck)**

2	R/W	**Reset delay chain saturation flag** **0: normal (default)** **1: reset flag**
1	R	**Delay chain saturation flag** **0: not saturate** **1: saturate => it need to enlarge the sample period or set bigger N code**
0	R/W	**APLL_free_run enable** **0: normal state (default)** **1: force APLL to free run state**

- While we got delay chain saturation flag 1'b1, that means that the big jitter is bigger than what we image and we have to reset the delay chain length setting BC [7:6]. Also we have to enlarge the sampling period & delay chain length
- The choice for sampling period will be set by the rule as following:
  (Delay chain length * 78 +50) * each tap delay + 10(ns) must be < N * T_{XCLK} * sample period
  if delay chain saturation flag goes high, then we must enlarge the delay chain length & set bigger sampling period
- While we enable free run mode, DDS will keep reset status until disable free run

Address: BD **DDS_MIX_6**

Bit	Mode	Function
7:0	R	**Final M code to APLL**

- While we like to read final M code & K code, we have to enable measure PE 9E[7] first. Otherwise we will get glitch value

Address: BE **DDS_MIX_7** **Default: 00h**

Bit	Mode	Function
7:4	R	**Final K code to APLL**
3:1	R/W	**Change mode threshold => triggered by any Q (PE) > threshold** **000: 600 (default)** **001: 850** **010: 1100** **011: 1350** **100: 1600** **101: 1850** **110: 2100** **111: 2350**
0	R/W	**new_mode_i_code_en** **0: while new mode enable, I code will have no effect on SUM_I. All phase error will be compensated by P code (default)** **1: while new mode enable, I code will be operated as normal state**

- For APLL interrupt status that include 4 different types:
  No lock: initial is 1 => over lock threshold B8 [3:1] => 1
  Wait state: initial is 1 => valid only while u enable new mode => over new mode enable threshold BB [5:3] => 1
  New mode state: initial is 1 => valid only while u enable new mode => over new mode lock threshold BB [2:0] => 1
  Change mode happen state: initial is 1 => over change mode threshold BE [3:1] => 1

DISPLAY PLL (Page 1)

Register::DPLL_M				0xBF	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DPLL_M[7:0]	7:0	R/W	4E	**DPLL DPM value - 2**	

Register::DPLL_N				0xC0	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DPLL_RESERVE1	7	R/W	0	**Reserved**	
DPLL_BPN	6	R/W	0	**DPLLBPN** 0: N divider enable. 1: N divider disable, OUT=ckxtal.	
DPLL_O[1:0]	5:4	R/W	1	**DPLL Output Divider** 00: Div1 01: Div2 (Default) 10: Div4 11: Div8	
DPLL_N[3:0]	3:0	R/W	3	**DPLL DPN value - 2**	

- Assume DPLL_M=0x7D, DPM=0x7D+2=127; DPLL_N=0x0A, DPN=0x0A+2=12; Divider=1/4, F_IN = 24.576MHz.
  F_DPLL = F_IN x DPM / DPN x Divider = 24.576 x 127 / 12 / 4 = 65.024MHz.

CRBF~CRC0 are double buffer.

Register::DPLL_CRNT				0xC1	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DPLL_RS[2:0]	7:5	R/W	3	**DPLL Loop Filter Resister Control** 000: 16K 001: 18K 010: 20K 011: 22K (Default) 100: 24K 101: 26K 110: 28K 111: 30K	
DPLL_CS[1:0]	4:3	R/W	2	**DPLL Loop Filter Capacitor Control** 00: 18p 01: 20p 10: 24p (Default) 11: 28p	
DPLL_IP[2:0]	2:0	R/W	2	**DPLL Charger Pump Current Control** Icp=(2.5uA+2.5uA*bit[0]+5uA*bit[1]+10uA*bit[2]) Icp=0.5*M/N*14.318/27	

DCLK Spread Spectrum (Page 1)

Register::DPLL_WD				0xC2	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DPLL_WDO	7	R	0	**DPLL WD Status** 0: Normal 1: Abnormal Write 1 to clear.(WD enable when DPLL power on)	wclr_out
DPLL_WDRST	6	R/W	0	**DPLL WD Reset** 0: Normal (Default) 1: Reset	
DPLL_WDSET	5	R/W	0	**DPLL WD Set** 0: Normal (Default) 1: Set	
Revered	4	R/W	0	**Reserved**	
DPLL_STOP	3	R/W	1	**DPLL Freqency Tuning** 0: Disable 1: Enable (Default)	
DPLL_FREEZE	2	R/W	0	**DPLL Output Freeze** 0: Normal (Default) 1: Freeze Active high.	
DPLL_VCORSTB	1	R/W	0	**Reset VCO** 0: Normal (Default) 1: Reset Active high.	
DPLL_PWDN	0	R/W	1	**Power Down DPLL** 0: Power on 1: Power down(Default) Active high.	

Register::DPLL_CAL				0xC3	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DPLL_VCOMD[1:0]	7:6	R/W	3	**DPLL VCO Default Mode** 00: VCO slowest 11: VCO fastest (Default)	
DPLL_CALBP	5	R/W	0	**DPLL Bypass Calibration** 0: Reference by Calibration result(Default) 1: Reference by CRC3[7:6] Active high.	
DPLL_CALSW	4	R/W	0	**Calibration Validated** Go high after power on 1200us. 0: Reference by CRC3[7:6] 1: Refernect by cal result	
DPLL_CALLCH	3	R/W	0	**Latch Calibration** Go high after power on 1100us. 0: Disable Latch 1: Enable Latch	
DPLL_CMPEN	2	R/W	0	**CMP Enable** Go high after power on 1000us. 0: Diable CMPEN 1: Enable CMPEN	
DPLL_CP	1	R/W	0	**CP Control** 0: 1.77pF` 1: 2.1pF	
DPLL_RESERVE	0	R/W	1	**Reserved for DPLL Phase Swallow Circuit** 0: Path0 1: Path1	

Register:: Initial DCLK_FINE_TUNE_OFFSET_MSB				0xC4	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DPLL_LINEAR_CHANGE	7	R/W	0	**Linear change offset value function** 0 : disable 1: enable (auto clear when finish) It should work on DDS Spread Spectrum Output function enable. When function is done, the initial offset and DPLLUPDN value would be the target offset and DPLLUPDN value.	
DPLL_EVEN_OLD_EN	6	R/W	0	**Only Even / Odd Field Mode Enable** 0: Disable (Default) 1: Enable	
DPLL_EVEN_OD_SEL	5	R/W	0	**Even / Odd Field Select** 0: Even (Default) 1: Odd	
DPLL_FUPDN	4	R/W	1	**DPLLFUPDN (DPLL Frequency Tuning)** 0: Freq Up 1: Freq Down (Default)	
DCLK_OFFSET[11:8]	3:0	R/W	0	Initial **DCLK Offset** [11:8] in Fixed Last Line DVTOTAL & DHTOTAL	

Register:: Initial DCLK_FINE_TUNE_OFFSET_LSB				0xC5	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DCLK_OFFSET[7:0]	7:0	R/W	0	**Initial DCLK Offset** [7:0] in Fixed Last Line DVTOTAL & DHTOTAL	

Register:: DCLK_SPREAD_SPECTRUM				0xC6	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DCLK_SPREAD_RANGE	7:4	R/W	0	**DCLK Spreading range** (0.0~7.5%) The bigger setting, the spreading range will bigger, but not uniform	
DCLK_FMDIV	3	R/W	0	**Spread Spectrum FMDIV** (SSP_FMDIV)//(0) 0: 33K 1: 66K	
DCLK_READY	2	R/W	0	**Spread Spectrum Setting Ready for Writing** (Auto Clear) 0: Not ready 1: Ready to write	
FREQ_SYNTHESIS_SEL	1:0	R/W	0	**Frequency Synthesis Select** (F & F-N*dF) 00~11: N=1~4	

- The "Spread Spectrum Setting Ready for Writing" means 4 kinds of registers will be set after this bit is set:
  1. DCLK spreading range
  2. Spread spectrum FMDIV
  3. DCLK offset setting
  4. Frequency synthesis select

Register:: EVEN_FIXED_LAST_LINE_MSB				0xC7	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
EVEN_FIXED_LAST_LINE[11:8]	6:4	R/W	3	**Even Fixed Last Line Length** [11:8]	
EVEN_FIXED_DVTOTAL[11:8]	3:0	R/W	0	**Even Fixed DVTOTAL** [11:8]	

Register:: EVEN_FIXED_LAST_LINE_LSB				0xC8	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
EVEN_FIXED_DVTOTAL[7:0]	7:0	R/W	0	**Even Fixed DVTOTAL** [7:0]	

Register:: EVEN FIXED_LAST_LINE_ LENGTH_LSB				0xC9	
Name	Bits	R/W	Default	Comments	Config
EVEN_FIXED_DVTOTAL[7:0]	7:0	R/W	0	**Even Fixed Last Line Length** [7:0]	

- If Even / Odd mode disable, we use EVEN_FIXED_LAST only.
- If Even/Odd mode enable, the even / odd field would be reference different setting.
- Fixed last line value can't be zero, and can't smaller than DH_Sync width.

Register:: FIXED_LAST_LINE_CTRL				0xCA	
Name	Bits	R/W	Default	Comments	Config
DEGLITCH	7	R/W	0	**Deglitch sscg asynchronous interface data function** **0: disable** **1: enable**	
RSV_CA_6	6	--	0	**Reserved to 0**	
MEASURE_PHASE	5	R/W	0	**Measure the Phase about Fixed DVTOTAL & Last Line DHTOTAL Function** 0 : Disable 1 : Enable (Auto clear when finish)	
MARK_PHASE_TRACKING	4	R/W	0	**Mark Phase tracking about Fixed DVTOTAL & Last Line DHTOTAL Function** 0 : Disable 1 : Enable	
NED_FIXED_LAST_LINE_MODE	3	R/W	0	**Enable New Design Function in Fixed Last Line Mode** 0: Disable (Default) 1: Enable	
DCLK_DDS	2	R/W	0	**DDS Spread Spectrum Test Enable** 0: Disable (Default) 1: Enable	
DCLK_FIXED_LAST_LINE_EN	1	R/W	0	**Enable the Fixed DVTOTAL & Last Line DHTOTAL Function** 0: Disable (Default) 1: Enable	
DCLK_DDS_EN	0	R/W	0	**Enable DDS Spread Spectrum Output Function** 0: Disable (Default) 1: Enable	

Procedure:

- First, we have set M/N code and then we need to tune DCLK OFFSET to achieve frame-sync, every step of offset frequency is DCLK/ 2^{15} .
- When we finished the frame-sync, we turn on CRCA[1] to let the system running in to free-run mode, at this time, the CRC7,CRC8,CRC9 are the reference DV and DH total and Fixed last Line Length.
- But the free-run mode DVS' should be close to frame-sync mode DVS to achieve pseudo-frame-sync(actually, it is free run mode now)
- Then we use CRC6[1:0] (F-N*dF) to keep DVS' and DVS very closely to achieve pseudo-frame-sync.

Notice:

- In RTD2485XD, when all the setting above is ready, then we open spread spectrum function, the DCLK OFFSET will shift, please keep the DCLK OFFSET keeps steady when we open spread spectrum function.
- In Real free-run mode, the DV_TOTAL refers to CR2B-0B/CR2B-0C, and in Fixed-Last-Line mode, and disable "Even/Odd mode" then the free-run timing DV_TOTAL refers to CRC7/CRC8, at this time CR2B-0B/CR2B-0C serve for Vsync-timeout watch dog reference.

Register:: ODD_FIXED_LAST_LINE_MSB				0xCB	
Name	Bits	R/W	Default	Comments	Config
ODD_FIXED_LAST_LINE_LENG[11:8]	6:4	R/W	0	**ODD Fixed Last Line Length** [11:8]	
ODD_FIXED_DVTOTAL[11:8]	3:0	R/W	0	**ODD Fixed DVTOTAL** [11:8]	

Register:: ODD_FIXED_LAST_LINE_DVTOTAL _LSB				0xCC	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ODD_FIXED_DVTOTAL[7:0]	7:0	R/W	0	**ODD Fixed DVTOTAL** [7:0]	

Register:: ODD_FIXED_LAST_LINE_ LENGTH_LSB				0xCD	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ODD_FIXED_LAST_LINE_LENG[7:0]	7:0	R/W	0	**ODD Fixed Last Line Length** [7:0]	

Register:: DCLK_SPREAD_SPECTRUM				0xCE	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
RSV_CE_72	7:2	---	0	Reserved	
FIXED_LAST_LINE_HIT	1	R	0	Fixed Last Line Tracking time hit Field one 0:hit field zero (If field zero is odd → hit odd field) 1:hit field one (If field one is even → hit even field)	
IVS_LEAD_LAG_TO_DVS	0	R	0	IVS Lead/Lag to DVS 0 : Lag 1 : Lead	

Register:: PHASE_RESULT_MSB				0xCF	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
RSV_CF_7	7	---	0	Reserved	
PHASE_LINE[11:8]	6:4	R	0	**Phase Line** [11:8]	
PHASE_PIXEL[11:8]	3:0	R	0	**Phase Pixel** [11:8]	

Register:: PHASE_LINE_LSB				0xD0	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PHASE_LINE[7:0]	7:0	R	0	**Phase Line** [7:0]	

Register:: PHASE_PIXEL_PIXEL				0xD1	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PHASE_PIXEL[7:0]	7:0	R	0	**Lead Phase Pixel** [7:0]	

Register:: TARGET_DCLK_FINE_TUNE_OFFSET_MSB				0xD2	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
RSV_D2_76	7:6	--	0	Reserved	
No_dbrdy_rst_profile	5	R/W	0	0: reset profile when CRC6[2] set to 1 1: no reset profile when CRC6[2] set to 1	
TARGET_DPLLUPDB	4	R/W	0	**Target DPLLUPDN (DPLL Frequency Tuning Up/Down)** 0: Freq Up(Default) 1: Freq Down	
TARGET_DCLK_OFFSET[11:8]	3:0	R/W	0	**Target DCLK Offset** [11:8] in Fixed Last Line DVTOTAL & DHTOTAL	

Register:: TARGET_DCLK_FINE_TUNE_OFFSET_LSB	0xD3

Name	Bits	R/W	Default	Comments	Config
TARGET_DCLK_OFFSET[7:0]	3:0	R/W	0	**Target DCLK Offset** [7:0] in Fixed Last Line DVTOTAL & DHTOTAL	

Register::DPLL_RESULT				**0xD4**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DPLL_REF_CLK_SEL	7	R/W	0	**DPLL reference clk select:** **0: xtal_clk** **1: m2pll_clk/Div, Div is reference to CR_22[6:3]**	
RSV_D4_74	6:4	---	0	**Reserved**	
DPLL_VO2	3	R	0	**DPLL CAL OUT2**	
DPLL_VO1	2	R	0	**DPLL CAL OUT1**	
DPLL_CAL[1:0]	1:0	R	0	**DPLL calibrated VCO code**	

Register::SSC_REF_SEL				**0xD5**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	**Reserved**	
DPLL_SSC_REF_SEL	0	R/W	0	**DPLL_SSC Referance clock select:** **0: xtal_clk,** **1: m2pll_clk/Div, Div is reference to CR_22[6:3]**	

Multiply PLL for Input Cyrstal (Page 1)

Register::M2PLL_M			0xE0		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
M2PLL_M[7:0]	7:0	R/W	0x42	**M2PLL DPM value – 2 (M)** * PLL output=input*(M/P)	

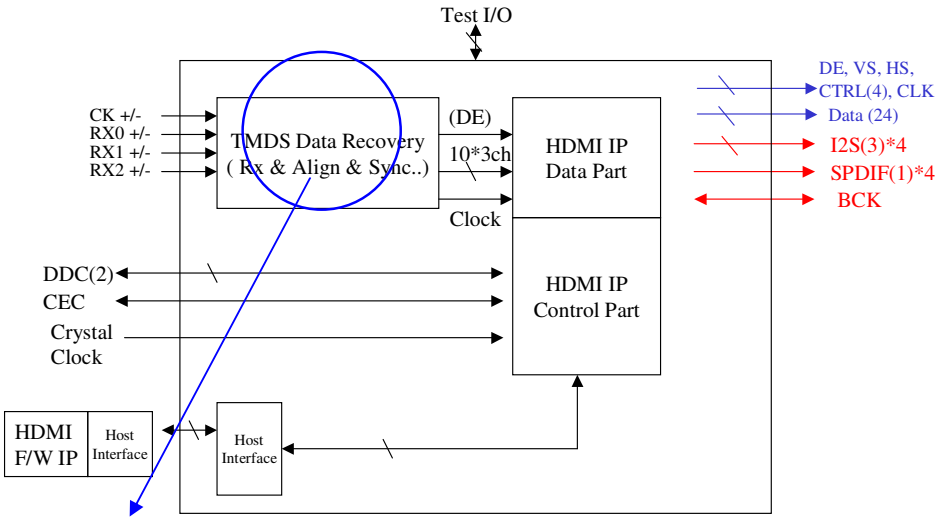
Register::M2PLL_N			0xE1		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
M2PLL_CP	7	R/W	0	**CP Control** 0:CP=1.77pF 1:CP=2.1pF	
M2PLL_BPN	6	R/W	0	**M2PLLBPN=0 , N divder enable** **M2PLLBPN=1, N divder disable , OUT=ckxtal**	
M2PLL_O[1:0]	5:4	R/W	1	**M2PLL Output divider** 00:Div1, 01:Div2, 10:Div4, 11:Div8	
M2PLL_N[3:0]	3:0	R/W	0	**M2PLL DPN value - 2**	

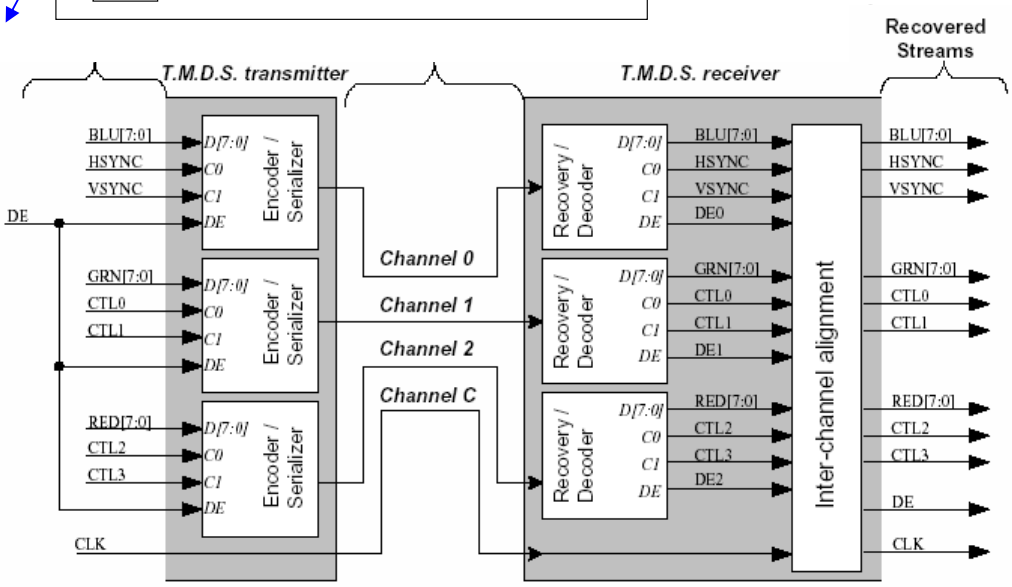
Note: CRE0~E1 are double buffer
CRE2~E3 are not controlled by software reset.

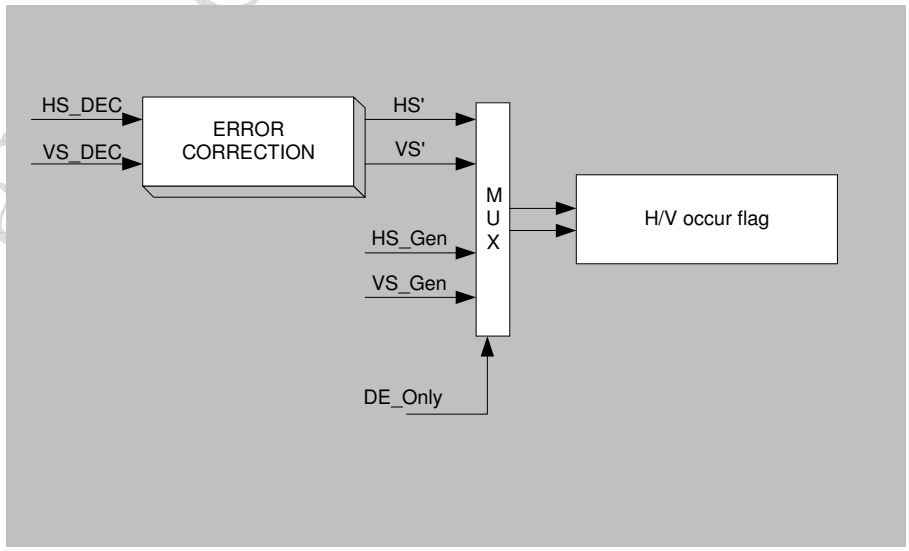
Register::M2PLL_CRNT			0xE4		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
M2PLL_RS[2:0]	7:5	R/W	3	**M2PLL Loop Filter Resister Control(Rs)** 000:16K, 001:18K, 010:20K, 011:22K 100: 24K, 101: 26K, 110:28K, 111:30K	
M2PLL_CS[1:0]	4:3	R/W	2	**M2PLL Loop Filter Capacitor Control(Cs)** 00:18p, 01:20p, 10:24p, 11:28p	
M2PLL_IP[1:0]	2:0	R/W	6	**M2PLL Charge Pump Current Control** Icp=(2.5uA+2.5uA*bit[0]+5uA*bit[1]+10uA*bit[2]) Icp=0.5*M/N*14.318/27	

Register::M2PLL_WD			0xE5		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
M2PLL_WDO	7	R	0	**M2PLL WD Status register** 0:Normal 1:Abnormal Write 1 to clear.(WD enable when M2PLL power on)	wclr_out
M2PLL_WDRST	6	R/W	0	**M2PLL WD Reset** 0:Normal 1:Reset	
M2PLL_WDSET	5	R/W	0	**M2PLL WD Set** 0:Normal 1:Set	
M2PLL_VCOMD[1:0]	4:3	R/W	3	**M2PLL VCO Default mode** 00: VCO slowest 11: VCO fastest	
M2PLL_FREEZE	2	R/W	0	**M2PLL Output Freeze** 0:Normal 1:Freeze (active high)	
M2PLL_VCORSTB	1	R/W	0	**RESET VCO (active high)**	
M2PLL_PWDN	0	R/W	0	**Power Down M2PLL (active high)**	

Overall HDMI System Function Block (Page 2)







Register: TMDS_MSR			0XA1	
Name	**Bits**	**R/W**	**Reset**	**Comments**

			State	
TMM	7	R/W	0	**Transition measurement method of hsync or data enable** 0: old method: Measure the number of transition for N-clock duration (TMDS_NCP[3:0]: [3:0]*(1/12) ms) 1: new method: Measure the number of transition smaller than 16 or 64 clock period (TMDS_CTC: 0xA2[0]) for 1-frame duration (vsync)
MT	6:4	R/W	0	**Measure times(exponential of 2)** 000: 1 001: 2 010: 4 011: 8 100: 16 101: Not available 110: Not available 111: Not available This function will do bit [6:4] times, each time lasts for bit [3:0]/12 ms.
NCP	3:0	R/W	0	**Numbers of Clock Period, measurement duration (where clock frequency is 12Khz)** 0000: 16 0001: 1 0010: 2 0011: 3 ……. 1111: 15 This function will do bit [6:4] times, each time lasts for bit [3:0]/12 ms.

Register: TMDS_MRR0				**0XA2**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
TMS	7	R/W	0	**Transition Measurement** 0: Stop measure, Cleared after finish (Default) 1: Start measure
MRS	6:5	R/W	0	**Measure Result Select** 00: AVE Value (Default) 01: Max Value 10: Min Value
MS	4:3	R/W	0	**The Selection of transition measurement of Hsync and Data_enable (select Hsync or Data enable)** 00: Measure Hsync transition times before error correction. 01: Measure Hsync transition times after error correction. 10: Measure Data Enable transition times before error correction. 11: Measure Data Enable transition times after error correction.
DE_INV_DISABLE	2	R/W	0	**Disable The Inversion of RGB channel Data Enable from data align (RGB channel together change polarity)** 0: Invert Data Enable 1: Keep the original polarity of Data Enable
CRC_NONSTABLE	1	R	0	**Check CRC is stable or not (write 1 clear)** 0: means CRC is stable. 1: means CRC is not stable.
CTC	0	R/W	0	**Criterion of Transition Count(de-bounce times)**, duration smaller than 0: 16 clock 1: 64 clock

Register: TMDS_MRR1				**0XA3**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
CRC_DONE	7	R	0	**CRC Calculation Finished** (Calculation starts when 0xA4 bit 0 setting to 1)

				0: not finished. 1: finished.
VMR	6:0	R	0	**Transition measure result** [6:0] **of hsync or data_enable** (Item refer to 0xA2[4:3])

Register:: TMDS_CTRL			**0XA4**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
BCD	7	R	x	**Blue-Channel Detect whether data_enable is low 128 clk** (DE low 128 clock) (DE low 128 clock)(write 1 clear) 0:no 1:yes
GCD	6	R	x	**Green-Channel Detect whether data_enable is low 128 clk** (DE low 128 clock)(write 1 clear) 0: no 1:yes
RCD	5	R	x	**Red-Channel Detect whether data_enable is low 128 clk** (DE low 128 clock)(write 1 clear) 0: no 1: yes
HO	4	R	x	**The source of Hsync is from HDCP.** **Detect whether Hsync Occurs** (write 1 clear) 0: no 1: yes
YO	3	R	x	**The source of Vsync is from HDCP.** **Detect whether Vsync Occurs** (write 1 clear) 0: no 1: yes
CRCTS	2:1	R/W	0	**The selection of CRC calculation type** 00: do CRC calculation only with DE 01: do CRC calculation only with DIEN (Data Island Enable) 10: do CRC calculation with both DE and DIEN 11: reserved
CRCC	0	R/W	0	**CRC calculation enable** 0: disable CRC calculation 1: enable CRC calculation

Register:: TMDS_CRCOB2			**0XA5**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
CRCOB2	7:0	R	--	1st read=> Output CRC-48 bit 47~40 2nd read=> Output CRC-48 bit 39~32 3rd read=> Out put CRC-48 bit 31~24 4th read=> Out put CRC-48 bit 23~16 5th read=> Out put CRC-48 bit 15~8 6th read=> Out put CRC-48 bit 7~0

- The read back CRC value address should be auto-increase, the sequence is shown above
-

Register:: TMDS_OUTCTL			**0XA6**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
AOE	7	R/W	0	**Auto Output Enable (video output fsm mode)** 0: Disable (Default), manual mode, determined by 0xA6 [6:3] 1: Enable, auto mode, determined by de_low128 from channel status
TRCOE	6	R/W	0	**TMDS R Channel Output Enable** 0: Disable (Default)

				1: Enable
TGCOE	5:	R/W	0	**TMDS G Channel Output Enable** 0: Disable (Default) 1: Enable
TBCOE	4	R/W	0	**TMDS B Channel Output Enable** 0: Disable (Default) 1: Enable
OCKE	3	R/W	0	**OCLK Enable** 0: Disable (Default) 1: Enable
OCKIE	2	R/W	0	**OCLK Invert Enable** 0: Normal (Default) 1: Enable
de_err_pulse_en	1	R/W	0	**Enable reset de_low128 according to data_enable error pulse, data_enable error pulse comes from data_align function** 0: disable 1: enable
CLK25XINV	0	R/W	0	**Input 1x Clock Invert** 0: No Invert (Default) 1: Invert

Register: TMDS_PWDCTL			**0xA7**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
DEO	7	R/W	0	**DE-only: Generate VS/HS from DE signal** 0: Disable (Default) 1: Enable
BRCW	6	R/W	0	**B/R channel swap** 0: No swap (Default) 1: Swap
PNSW	5	R/W	0	**P/N Swap** 0:No swap(Default) 1:swap
ICCAF	4	R/W	0	**Input Channel control by auto function** 0: Manual 1: Auto (Default)
ECC	3	R/W	0	**Enable Clock channel: turn on clock channel PLL (For manual use)** 0: Disable (Default) 1: Enable
ERIP	2	R/W	0	**Enable Red input port (For manual use, cut off 50ohm internal resistor)** 0: Disable (Default) 1: Enable
EGIP	1	R/W	0	**Enable Green input port (For manual use, cut off 50ohm internal resistor)** 0: Disable (Default) 1: Enable
EBIP	0	R/W	0	**Enable Blue input port (For manual use, cut off 50ohm internal resistor)** 0: Disable (Default) 1: Enable

Register:: TMDS_ACC0			**0XA8**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
TUNE_UP_DOWN	7	R/W	0	**AVMUTE Window Range Tune Up or Tune Down** 0: Tune Down 1: Tune Up
TUNE_RANGE	6:0	R/W	0	**AVMUTE Window Range**

				If tune up, valid window = 384 cycle + 2*AVMUTE window range. If tune down, valid window = 384 cycle - 2*AVMUTE window range.

Register:: TMDS_ACC1			**0XA9**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
WR_AKSV_FLAG	7	R	0	**Write AKSV Flag (Write 1 Clear)** When TX writing AKSV done, this flag will be asserted.
RD_RI_FLAG	6	R	0	**Read Ri Flag (Write 1 Clear)** When TX reading HDCP's Ri, this fl;ag will be asserted.
RD_BKSV_FLAG	5	R	0	**Read BKSV Flag (Write 1 Clear)** When TX reading HDCP's BKSV, this flag will be asserted.
HDCP_KEY_ACCESS_MODE	4	R/W	0	**HDCP Key SRAM Access Right Control Mode** 0: old mode 1: new mode (Auto switch SRAM Access Right when Aksv is received)
Reserved	3	R	0	**Reseved**
Reserved	2:0	--	0	**Reserved**

Register:: TMDS_ABC			**0xAA**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Reserved	7	--	--	Reserved
HDCP_ENC_DIS	6	R/W	0	**HDCP ENC DIS:** 0: Original VS 1: Vertical Blanking Start
Reserved	5	R/W	--	**Reserved to 0**
IRQ_AKSV_EN	4	R/W	0	**IRQ Control For AKSV (Flag refer to 0xA9)** 0: Disable 1: Enable
IRQ_RI_EN	3	R/W	0	**IRQ Control For Ri (Flag refer to 0xA9)** 0: Disable 1: Enable
IRQ_BKSV_EN	2	R/W	0	**IRQ Control For BKSV (Flag refer to 0xA9)** 0: Disable 1L Enable
Reserved	1:0	--	--	**Reserved to 0**

Register:: TMDS_ACC2			**0XAB**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
REG_HS_WIDTH_SEL	7	R/W	0	**Hsnc Pulse Width Selection** 0: 72 clock cycle width 1: 8 clock cycle width
HDCP_VS_SEL	6	R/W	0	**HDCP Vsnc Selection** 0: Choose original Vsnc. 1: Choose Virtual Vsnc.
BLAN K_KEEP_ EN	5	R/W	1	**Activated READ process of Audio doesn't pause even when Audio Packet assertion until the end of READ process.** 0: Disable 1: Enable
BLAN K_EN	4	R/W	1	**Block any READ request when Audio Packet assertion.** 0: Disable 1: Enable
BLAN K_PRE_EN	3	R/W	1	**Pre_block any READ request when Audio Packet assertion.** 0: Disable

				1: Enable
Reserved	2:0	--		**Reserved.**

Register:: TMDS_Z0CC2				**0xAC**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
DDCDBNC	7	R/W	1	**HDCP DDC DEBOUNCE** 0: Disable 1: Enable
HDE	6	R/W	0	**HDMI/DVI function enable** (HDCP enable is moved to HDCP) 0: Disable, gated clock and cut off TMDS pull up resistor for saving power. 1: Enable,
DBNC_LEVEL	5	R/W	1	**HDCP Debounce Level Selection** 0: Crystal period * 4 1: Crystal period * 8 When using 14.318 Mhz crystal clock, debounce level should be set to 0.
KM_CLK_SEL	4	R/W	0	**Clock Selection for KM Calculation** 0: Choose EMCU non-stop clock 1: Choose crystal clock
Reserved	3:0	R/W	--	**Reserved to 0**

Register:: TMDS_CPS				**0xAD**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
PLL_DIV2_EN	7	R/W	0	HDMI output clock div 2 (enable this register if 2x clock is needed) 0: disable 1: enable
CLKV_MEAS_SEL	6:5	R/W	01	**Input Clock Source Selection.** 00: Red clock (Lane 0 clock) 01: Blue clock (Lane 2 clock)) 10: Green clock (Lane 1 clock) 11: TMDS clock (Lane 3 clock)
CLKC_MEAS_SEL	4	R/W	1	**Detection Clock Source Selection.** 0: Use TMDS clock as reference clock to detect input clock frequency. 1: Use crystal clock as reference clock to detect input clock frequency.
CLR_INFOFRAME_DVI	3	R/W	0	**Clear info-frame data when DVI mode** 0: Disabled. 1: Enabled.
AUTO_DVI2HDMI	2	R/W	0	**In HDMI/DVI auto detection mode, auto switch to DVI mode when no signal(clk) being detected.** 0: Disabled. 1: Enabled
Reserved	1	--		**Reserved.**
CLKV_SEL	0	R/W	0	**The video clock selection of frequency detection circuit in power-off region:** 0: determined by 0xAD[6:5] 1: hdmi_cp_clk

Register:: TMDS_RPS				**0XAE**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
AVMUTE_FLAG	7	R/W	0	**Ignore AVMUTE Flag** 0: Disable 1: Enable (CLR_AVMUTE)
Reserved	6:2	--	0	**Reserved to 0**
HDCP_KEY_RDY	1	R/W	0	**HDCP KEY Ready Flag** This flag is for F/W to indicate HW that HDCP key is READY!

				0: HDCP key not ready. 1: HDCP key ready.
AUTHST_MODE_SEL	0	R/W	0	**Selection of AUTHST mode.** 0: Original mode. 1: New mode, H/W will keep authst signal until HDCP key is ready.

Register:: TMDS_WDC			**0xAF**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Reserved	7:0	R/W	0	**Reserved to 0**

~~**Register:: TMDS_CDR0**~~					~~**0xB0**~~
~~**Name**~~	~~**Bits**~~	~~**R/W**~~	~~**Default**~~	~~**Comments**~~	~~**Config**~~
~~**reserved**~~	~~7:4~~	~~--~~	~~0~~		
~~adj_gain~~	~~3:2~~	~~R/W~~	~~0~~	~~**ADJ_GAIN<1:0> Phase adjust gain.** One UP/DOWN could mean to change the phase by 1~4 minimum step sizes.~~	
~~lpf_sel~~	~~1:0~~	~~R/W~~	~~2~~	~~**LPF<1:0>** LPF selection 00: ACCUMULATION type x1: CONSECUTIVE type, 10: CASCADE type. (Consecutive →accumulation) (Default)~~	

~~**Register:: TMDS_CDR1**~~					~~**0xB1**~~
~~**Name**~~	~~**Bits**~~	~~**R/W**~~	~~**Default**~~	~~**Comments**~~	~~**Config**~~
~~thr_acc~~	~~7:0~~	~~R/W~~	~~A~~	~~**THR_ACC<7:0>:** Threshold to assert UP/DOWN in accumulation LPF~~	

~~**Register:: TMDS_CDR2**~~					~~**0xB2**~~
~~**Name**~~	~~**Bits**~~	~~**R/W**~~	~~**Default**~~	~~**Comments**~~	~~**Config**~~
~~thr_cse~~	~~7:0~~	~~R/W~~	~~A~~	~~**THR_CSC<7:0>:** Threshold to assert UP/DOWN in concessive LPF~~	

~~**Register:: TMDS_CDR3**~~					~~**0xB3**~~
~~**Name**~~	~~**Bits**~~	~~**R/W**~~	~~**Default**~~	~~**Comments**~~	~~**Config**~~
~~wait_time~~	~~7:0~~	~~R/W~~	~~0x14~~	~~**WAIT_TIME<7:0>:** The minimum period between two phases adjusts. (phase change responding time)~~	

Register:: TMDS_DPC0			**0XB4**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
dpc_pp	7:4	R	0	PP value of HDMI 1.3 Deep color mode. (If dpc_auto(0xB8[2]) ==0, this bit is R/W; otherwise, it is read-only)
dpc_cd	3:0	R	0	CD value of HDMI 1.3 Deep color mode. (If dpc_auto(0xB8[2]) ==0, this bit is R/W; otherwise, it is read-only)

Register:: TMDS_UDC0			**0XB5**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
dpc_bypass_dis	7	R/W	0	Disable Deep Color Mode 0: disable 1: enable
reserved	6:4	--	0	Reserved.
CP_DPLLFDDS	3	R/W	0	CP Test mode: 0: dpllfddst_gate = reg_cptest ? bckin : dpllfdds 1: dpllfddst_gate can switch to dpllfdds when reg_cptest = 1
CPTEST	2	R	0	**CPTEST** 0: normal mode, in which clock and data from analog are used. 1: select TSTCKIN/TSTDIN as input 2X5 clock and data respectively, for TESTING.

HMTM	1:0	R/W	0	**HDCP MP TESTING MODE** Force CTL[3:0] always equal to 00:Original 01:CTRL=1001 10:CTRL=1000 11:CTRL=0000

Register:: TMDS_UDC1				**0xB6**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
no_clk_in	7	R	0	No clock input. 0: normal, 1: no clock
cdr_rdy_red	6	R	0	CDR ready of red channel
cdr_rdy_grn	5	R	0	CDR ready of green channel
cdr_rdy_blu	4	R	0	CDR ready of blue channel
reserved	3:0	--	0	Reserved.

Register:: TMDS_UDC2				**0XB7**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
NL	7:5	R/W	0(should be 3)	**The selection of error correction, ERRC_SEL<2:0>** 000: original signal 001: 1 cycle debouncing 010: 1+8 cycle debouncing 011: 1+8 cycle debouncing + de masking transition of vs/hs 100: 1+8 cycle debouncing + de masking transition of vs/hs + masking first 8-line de
NLFW	4:0	R/W	0	**The selection of debug signal, total has 32 kinds of debug signals.** 00000: dbg_out_0 00001: dbg_out_1 …….. 11111: dbg_out_31

Register:: TMDS_DPC1				**0XB8**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
reserved	7:4	--	0	Reserved.
dpc_clk_source	3	R/W	0	Select the reference clock of deep color pll 0: recovered tmds clock 1: original tmds clock
dpc_auto	2	R/W	1	0: manual mode (CD/PP/default_ phase fields are specified by FW) 1: auto mode (CD/PP/default_ phase are directly decoded by HW)
dpc_default_ph	1	R/W	0	Default Phase of HDMI 1.3 Deep color mode. (If dpc_auto(0xB8[2]) ==0, this bit is R/W; otherwise, it is read-only)
dpc_pp_valid	0	R/W	0	Phase valid of HDMI 1.3 Deep color mode. (If dpc_auto(0xB8[2]) ==0, this bit is R/W; otherwise, it is read-only)

Register:: TMDS_OUT_CTRL				**0xB9**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
TMDS_BYPASS	7	R/W	1	**TMDS Bypass Control** 0: RGB values will be assigned by register's setting when it is out of data enable region 1: RGB values will be set to 16'b0 when it is out of the data enable region
Rev	6:0	---		**Reserved**

Register:: TMDS_ROUT_HIGH_BYTE				**0xBA**
Name	**Bits**	**R/W**	**Reset**	**Comments**

			State	
TMDS_ROUT_H	7:0	R/W	0	**TMDS Rout High Byte Register Value** High Byte [15:8]

~~**Register:: TMDS_ROUT_LOW_BYTE**~~				~~**0xBB**~~
~~**Name**~~	~~**Bits**~~	~~**R/W**~~	~~**Reset State**~~	~~**Comments**~~
~~TMDS_ROUT_L~~	~~7:0~~	~~R/W~~	~~0~~	~~**TMDS Rout Low Byte Register Value**~~ ~~Low Byte [7:0]~~

Register:: TMDS_GOUT_HIGH_BYTE				**0xBC**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
TMDS_GOUT_H	7:0	R/W	0	**TMDS Gout High Byte Register Value** High Byte [15:8]

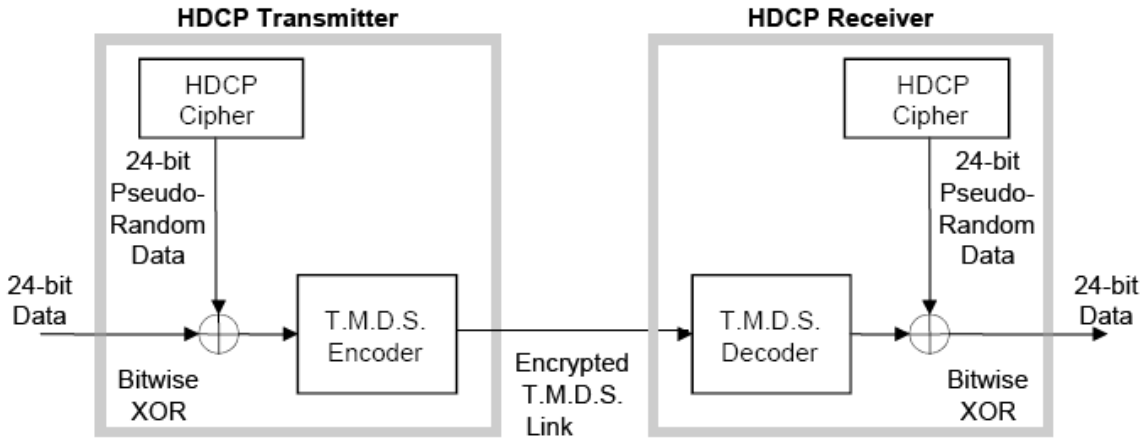
Register:: TMDS_GOUT_LOW_BYTE				**0xBD**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
TMDS_GOUT_L	7:0	R/W	0	**TMDS Gout Low Byte Register Value** Low Byte [7:0]

Register:: TMDS_BOUT_HIGH_BYTE				**0xBE**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
TMDS_BOUT_H	7:0	R/W	0	**TMDS Bout High Byte Register Value** High Byte [15:8]

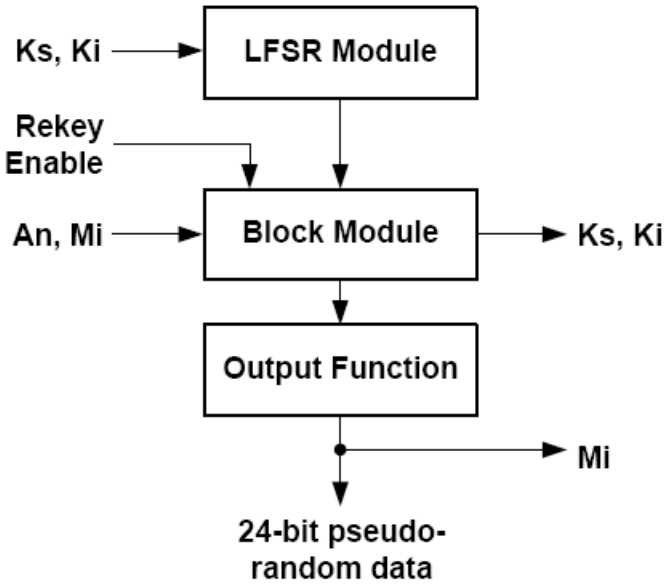
Register:: TMDS_BOUT_LOW_BYTE				**0xBF**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
TMDS_BOUT_L	7:0	R/W	0	**TMDS Bout Low Byte Register Value** Low Byte [7:0]

HDCP 1.3 (Page 2)

HDCP Encryption and Decryption:



HDCP Cipher:



Register:: HDCP_CR			0XC0	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
AC_MOD	7	R/W	1	Advance Cipher Mode 0: Old mode (AC = HDMI or (DVI & Ainfo[1])) 1: New mode (AC = Ainfo[1])
Reserve	6	R	0	**Reserved.**
IVSP	5	R	0	**Indicate VSYNC Polarity** 0: Positive, which means VS pulse is high. 1: Negative
INVVS	4	R/W	0	**Invert VSYNC for HDCP** High: Inverted Low: Not Inverted
IVSPM	3	R/W	0	**Indicate VSYNC Polarity Mode:** High: manual, decided by INVVS

				Low: auto, indicate by IVSP
MADDF	2	R/W	0	**MCU Access DDC data first** **(when download key from MCU, force only MCU could work)** 0: enable DDC channel and MCU access only when DDC is not busy 1: disable DDC channel and MCU access only
DKAPDE	1	R/W	0	**Device Key Access Port download enable (download key from MCU)** High: enable Low: disable, this would reset the address of Device Key Access Port to 0.
Enable	0	R/W	0	**HDCP Enable** High: Auto Enable HDCP function, when Tx I2C write Aksv, Low: Disable HDCP, except for output.

Register:: HDCP_DKAP			**0XC1**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
DKAP	7:0	R/W	0	When enable device key accessing 40x56 table, the 56-bit key table will be transferred to 64-bit pseudo data with 7th, 15th, 23rd, 31st, 39th, 47th, 55th bits inserted. The inserted data are '0'.And the write sequence is: {D0-Byte0, D0-Byte1, D0-Byte2, D0-Byte3,D0-Byte4, D0-Byte5, D0-Byte6, D0-Byte7}, {D1-Byte0, D1-Byte1, 1-Byte2,D1-Byte3, D1-Byte4, D1-Byte5, D1-Byte6, D1-Byte7}, …… Accessing this port must be coded/decoded by REALTEK protection code.

Register:: HDCP_PCR			**0xC2**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Rev	7:5	---		**Reserved**
ENC_TOG	4	R	0	**ENC toggled.**
AVMUTE_DIS	3	R/W	1	**Auto enc_dis when AVMUTE** 0: non active 1: active
DDCSEL	2:1	R/W	0	**DDC Channel SEL for Key Access** 00: DDCSCL2/DDCSDA2(pin 123~124) 01: DDCSCL3/DDCSDA3(pin 121~122) 10: non-useful 11: non-useful
APAI	0	R/W	0	**HDCP Accessing Port Auto Increase** (For Host Side) 0: auto increase 1: keep in the same address.

Register:: HDCP_AP			**0XC3**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
AP	7:0	R/W	0	**Address port for embedded HDCP access**, auto increase after DATA_PORT being accessed. (For Host Side controlled by APAI)

Register:: HDCP_DP			**0XC4**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
DP	7:0	R/W	0	**Data port for embedded HDCP access**

I2C Control Register Map (DVI DDC side) device address: 0x74/0x75
Following register is assigned by "HDCP-address port", "HDCP-data port" (From HDCP 1.1)

Hex Address	*Write/ Read*	*Size in Bytes*	*Register Name*	*Function*
0x C4-00	R	5	BKSV	HDCP Receiver KSV. This value may be used to determine that the receiver is HDCP capable. Valid KSVs contain 20 ones and 20 zeros, a characteristic that must be verified by HDCP Transmitters before encryption is enabled. This value must be available any time the HDCP Receiver's HDCP hardware is ready to operate.
0x C4-05	R	3	Reserved	All bytes read as 0x00
0x C4-08	R	2	Ri'	Link verification response. Upon completion of the authentication computations, this register contains the R0' value. Following that, it is updated upon completion of HDCPBlockCipher if (i mod 128) == 0. It is recommended that HDCP Transmitters protect against errors in the I2C transmission by re-reading this value when unexpected values are received, though care must be taken to avoid missing legitimate mis -match conditions. This value must be available at all times between updates. R0' must be available less than 100 ms after A*ksv* is received. Subsequent Ri' values must be available a maximum of 128 pixel clocks following the Encryption Enable detection (ENC_EN).
0x C4-0A	R	1	Pj'	Enhanced Link Verification Response. Updated upon receipt of first video pixel received when frame counter value (j mod 16) == 0. The value is the XOR of the decrypted byte on channel zero of the first video pixel with the least significant byte of Rj. Rj is derived from the output function in the same manner as Ri, but is captured every 16 th counted frame (rather than every 128 th counted frame).
0x C4-0B	R	5	Reserved	All bytes read as 0x00
0x C4-10	R/W	5	AKSV	HDCP transmitter KSV. Writes to this multi-byte value are written least significant byte first. The final write to 0x14 triggers the authentication sequence in the HDCP Receiver, and the current Ainfo value is copied from the port, takes effect, and the port is reset to the default value of zero.
0x C4-15	R/W	1	Ainfo	Bits 7-2: Reserved zeros. Bit 1: ENABLE_1.1_FEATURES. This bit enables the Advance Cipher option. If in DVI mode, it also enables the Enhanced Encryption Status Signaling (EESS) (in HDMI mode, EESS is enabled regardless of this bit setting). This bit resets to default zero when the HDCP Receiver becomes attached or active, or is reset, or the last byte of Aksv is written. A write to the last byte of Aksv copies the port value and causes it to take effect, and then resets the port value to the default value of zero. Thus the options must be explicitly enabled prior to each authentication. Bit 0: Reserved (must be zero).
0x C4-16	R	2	Reserved	All bytes read as 0x00
0x C4-18	R/W	8	An	Session random number. This multi-byte value must be written by the HDCP Transmitter before the KSV is written.
0x C4-20	R	20	Reserved	All bytes read as 0x00
0x C4-34	R	12	Reserved	All bytes read as 0x00
0x C4-40	R	1	Bcaps	Bit 7: HDMI_RESERVED Use of this bit is reserved. HDCP Receivers not capable of supporting HDMI must clear this bit to 0. Bit 6: REPEATER, HDCP Repeater capability. When set to one, this HDCP Receiver supports downstream connections as permitted by the Digital Content Protection LLC license. This bit does not change while the HDCP Receiver is active. Bit 5: READY, KSV FIFO ready. When set to one, this HDCP Repeater has built the list of attached KSVs and computed the verification value *V*'. This value is always zero during the computation of *V*'. Bit 4: FAST. When set to one, this device supports 400 KHz transfers. When zero, 100 KHz is the maximu m transfer rate supported. Note that 400KHz transfers are not permitted to any device unless all devices on the I 2 C bus are capable of 400KHz transfer. The transmitter may not be able to determine if the EDID ROM, present on the HDCP Receiver, is capable of 400KHz operation. This bit does not change while the HDCP

				Receiver is active. Bits 3-2: Reserved (must be zero). Bit 1: 1.1_FEATURES. When set to one, this HDCP Receiver supports Enhanced Encryption Status Signaling (EESS), Advance Cipher, and Enhanced Link Verification options. For the HDMI protocol, Enhanced Encryption Status Signaling (EESS) capability is assumed regardless of this bit setting. This bit does not change while the HDCP Receiver is active. Bit 0: FAST_REAUTHENTICATION. When set to 1, the receiver is capable of receiving (unencrypted) video signal during the session re-authentication. All HDMI-capable receivers shall be capable of performing the fast re-authentication even if this bit is not set. This bit does not change while the HDCP Receiver is active.
0x C4-41	R	2	Bstatus	Refer to Table 1
0x C4-43	R	1	KSV FIFO	Key selection vector FIFO. This device is not a repeater. All byte read as 0x00 for HDCP Receivers that are not HDCP Repeaters(REPEATER==0).
0x C4-44	R	124	Reserved	All bytes read as 0x00

Name	***Bit Field***	***Read/Write***	***Description***
Reserved	15:14	R	Read as zero.
HDMI_RESERVED_2	13	R	Reserved for future possible HDMI use.
HDMI_MODE	12	R	HDMI Mode. When set to one, the HDCP Receiver has transitioned from DVI Mode to HDMI Mode. This has occurred because the HDCP Receiver has detected HDMI bus conditions on the link. This bit must not be cleared when the HDCP Transmitter and HDCP Receiver are connected and both are operating in an active HDMI mode. This bit must be cleared upon power-up, reset, unplug or plug of an HDCP Transmitter or anytime that the HDCP Receiver has not seen at least one Data Island within 30 video frames.
	11:0	R	Read as zero.

Table 1 (Address 0x41)

Note :
1. When accessing this DDC register map by DDC, the address should increase automatically, except for the first accessing address is KSV_FIFO, 0x43.

Register:: HDCP_BCAPS			**0xC4-40**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
HDMI_RESERVED	7	R	0	**HDMI_RESERVED** Use of this bit is reserved. HDCP Receivers not capable of supporting HDMI must clear this bit to 0.
REAPEATER	6	R	0	**REPEATER** HDCP Repeater capability. When set to one, this HDCP Receiver supports downstream connections as permitted by the Digital Content Protection LLC license. This bit does not change while the HDCP Receiver is active.
READY	5	R	0	**READY** KSV FIFO ready. When set to one, this HDCP Repeater has built the list of attached KSVs and computed the verification value V'. This value is always zero during the computation of V'. See states C0 and C2.
FAST	4	R	0	**FAST** When set to one, this device supports 400 KHz transfers. When zero, 100 KHz is the maximum transfer rate supported. Note that 400KHz transfers are not permitted to any device unless all devices on the I2C

				bus are capable of 400KHz transfer. The transmitter may not be able to determine if the EDID ROM, present on the HDCP Receiver, is capable of 400KHz operation. This bit does not change while the HDCP Receiver is active.
Reserved	3:2	R	0	**Reserved to 0**
1.1_FEATURES	1	R	0	**1.1_FEATURES** When set to one, this HDCP Receiver supports Enhanced Encryption Status Signaling (EESS), Advance Cipher, and Enhanced Link Verification options. For the HDMI protocol, Enhanced Encryption Status Signaling (EESS) capability is assumed regardless of this bit setting. This bit does not change while the HDCP Receiver is active.
FAST_REAUTHENTICATION	0	R	0	**FAST_REAUTHENTICATION** When set to 1, the receiver is capable of receiving (unencrypted) video signal during the session re-authentication. All HDMI-capable receivers shall be capable of performing the fast re-authentication even if this bit is not set. This bit does not change while the HDCP Receiver is active.

Register:: HDCP_FCR			**0xC4-C0**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Reserved	7	R	--	**Reserved**
FC	6:0	R	0	**HDCP_frame counter**[6:0]

Register:: HDCP_SIR			**0xC4-C1**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
AST	7	R	0	**Authst (Means bksv of RTD pass Tx authorization, Tx is ready to do HDCP transaction)**
AKM	6	R	0	**Authkm (Means RTD finish computing KM, ri) //Hidden**
ADNE	5	R	0	**Authdone (means TX admitted ri value, start to do HDCP transmission)**
REA	4	R/W	0	**RE_AUTH**
ENCM	3	R/W	0	**ENC_Method**
ENCE	2	R	0	**ENC_ERROR (Receive ENC enable and disable at the same time)**
NC	1	R	0	**NO CTRL(HDCP1.0: no ctrl[3], HDCP1.1: ctrl is not 1001 nor 0001)**
IB	0	R	0	**Internal buffer for Ainfo[1].** Since Ainfo[1] in DDC port is 0 at most of time, we need to know what Tx wrote.

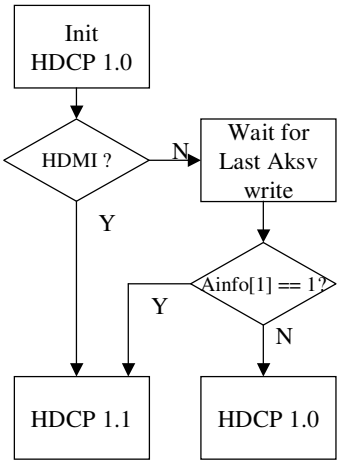
Register:: HDCP_SLAVE_ADD			**0xC4-C2**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Reserved	7	R/W	0	Reserved
Slave_ADDR	6:0	R/W	3A	HDCP Slave_Addr[6:0] Ex. HDCP address = 0x74, 8'b 0111 0100 Reg_Slave_Addr[6:0] = 0111 010 … the last bit is ignored.

HDCP 1.1/1.0 decide flow.

1. If HDMI conditions happen, HDCP 1.1 is used.
2. When last byte of Aksv is written, Ainfo[1] indicates HDCP 1.0/1.1 mode.

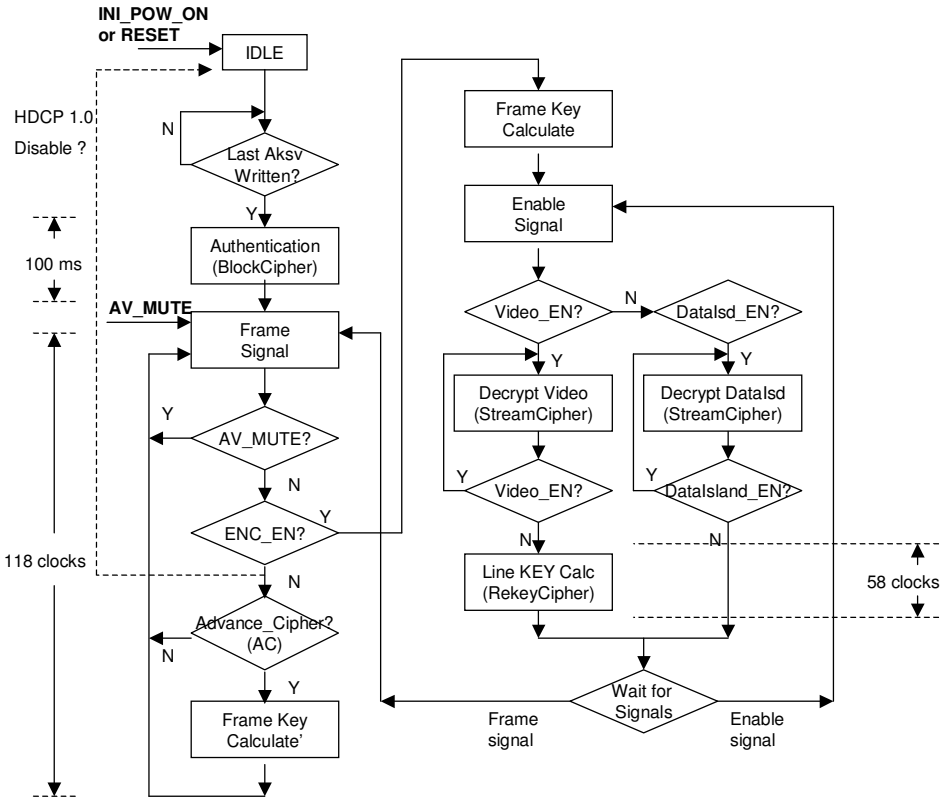
OESS is the same as HDCP 1.0. We could tell it by Ainfo[1] in DDC.

HDCP 1.0/1.1 decide flow
(Before Auth)



Initial flow.

HDCP Total Flow



HDCP 1.0/1.1 difference

Item	Description	HDCP 1.0	HDCP 1.1
1	Fast Reset	No constraint in 1.0	It must be done
2	DDC : Ainfo	Useless	Double buffer
3	DDC : Pj	No this feature	Update per 16 frames
4	DDC : Bcaps[1]	No this feature	It is used to tell if Rx supports 1.1

5	DDC : Bstatus	No this feature	HDMI mode mapping
6	DDC : short read	Read Ri.	Read Ri & Pj.
7	OESS/EESS	Only OESS compatible	Depend on DDC info. Sync.
8	Support protocol	DVI (DE only)	DVI & HDMI (DE & DIEN)
9	CTLx position	CTL3 follows VS	All info must be in opp. window.
10	Error correction	No the requirement	Error correction for ENC_EN/DIS
11	VS polarity distinguishment	No clear description	1. init is neg. 2. VS debouncing befor DE. 3. VS por for open opp window.

Frame counter

HDCP 1.0 : Increase by VS(CTL3).

HDCP 1.1 : In OESS mode, increase by ENC_EN

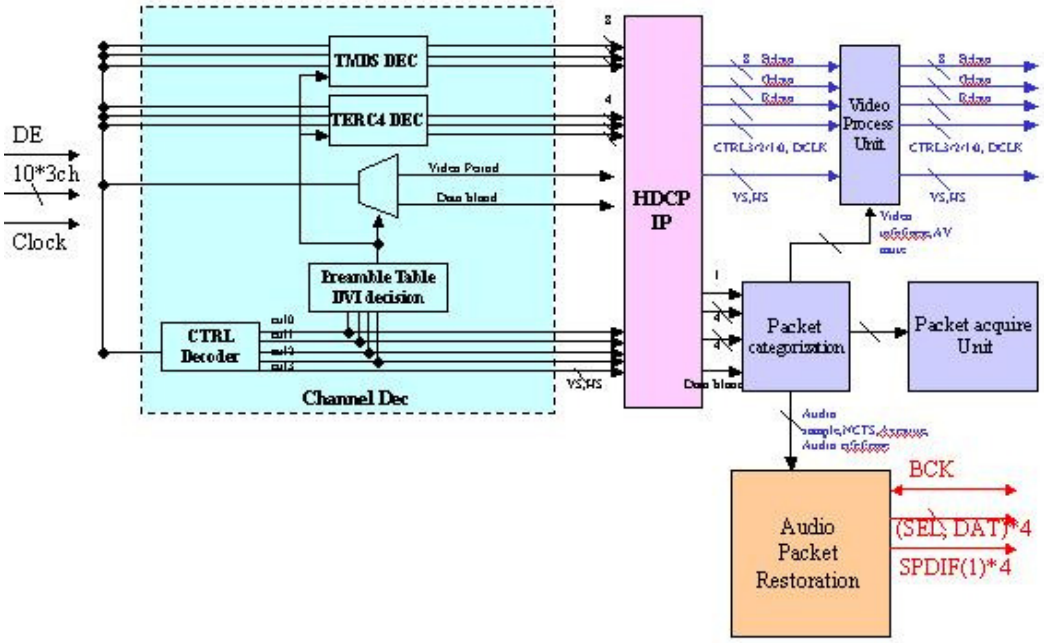
In EESS mode, increase when a. AV_MUTE = false. & b. AC = 1 or ENC_EN = 1.

NOTE :

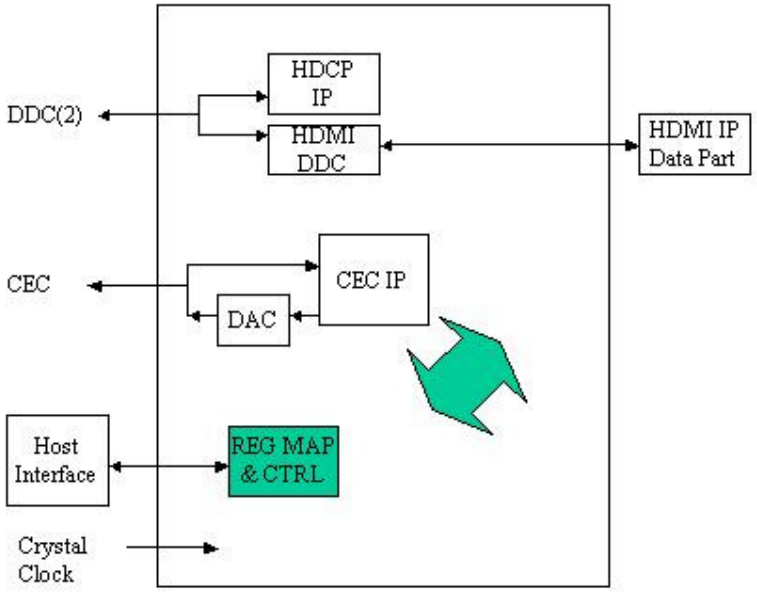
1. HDCP output must be always enable for DVI/HDMI.
2. The sub-descriptions i of Ri & j of Pj are the same.

HDMI Video and Audio Part (Page 2)

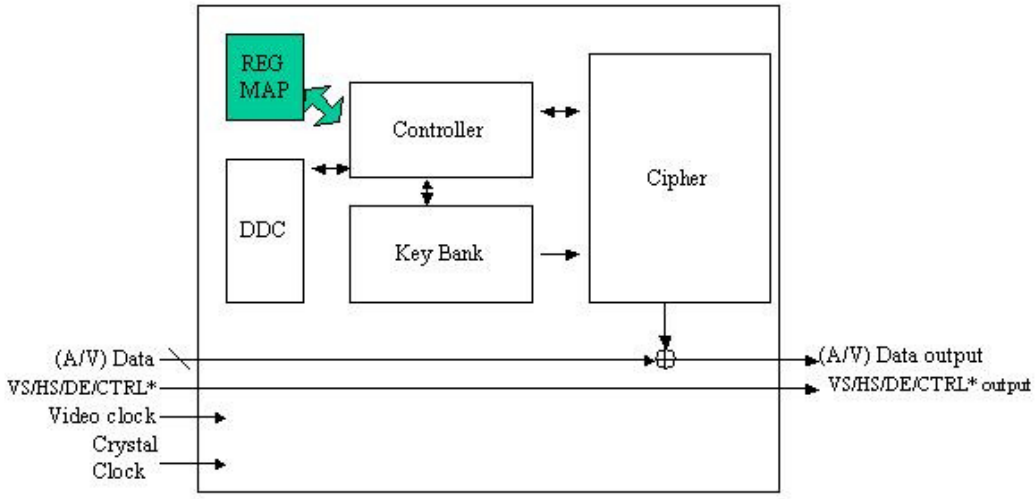
HDMI IP Data Part



HDMI IP Control Part



HDCP IP



Register:: HDMI_APC			0xC8	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Reserved	7:1	R/W	0	**Reserved to 0**
AAIF	0	R/W	0	**Address auto increase function** 0: If read/write "HDMI data port" continuously without assign "HDMI address port", address would be not added by one automatically. 1: If read/write "HDMI data port" continuously without assign "HDMI address port", address would be added by one automatically.

Register:: HDMI_AP			0xC9	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
AP	7:0	R/W	0	**Address port for HDMI**

Register:: HDMI_DP			0xCA	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
DP	7:0	R/W	0	**Data port for HDMI**

HDMI Register in Address Data Port

Access Addr.	Name	Description
0x00	HDMI_SCR	System Control
0x01	HDMI_N_VAL	N times of Condition A
0x02	HDMI_BCHCR	BCH Control Bits
0x03	HDMI_AFCR	Audio Flow Control
0x04	HDMI_AFSR	Audio FIFO Status
0x05	HDMI_MAGCR	Manual Audio Gain Coefficient
0x06	HDMI_AAGCR	Auto Audio Gain Control
0x07	HDMI_MAG_M_FINAL	Audio Final Gain Control
0x08	HDMI_MAG_L_FINAL	HDMI Left Channel Final Gain
0x09	HDMI_MAG_R_FINAL	HDMI Right Channel Final Gain
0x0A	AUDIO_LD_P_TIME_M	Audio Waiting Time

0x0B	AUDIO_LD_P_TIME_N	Audio Waiting Time
0x0C	ZCD_CTRL	Zero Crossing Control of Final Gain
0x0D	ZCD_TIME_OUT	Time Out to Force Final Gain
0x0E	ZCD_STATUS	Zero Crossing Detect Status
0x10	HDMI_CMCR	Clock MUX Control
0x11	HDMI_MCAPR	M Code of Audio PLL
0x12	HDMI_SCAPR	S Code of Audio PLL
0x13	HDMI_DCAPR0	MSB of D Code of Audio PLL
0x14	HDMI_DCAPR1	LSB of D Code of Audio PLL
0x15	HDMI_PSCR	Phase Swallow Control
0x16	HDMI_FDDR	FIFO Depth at DE Rising
0x17	HDMI_FDDF	FIFO Depth at DE Falling
0x18	HDMI_MFDDR	Maximum FIFO Depth at DE Rising
0x19	HDMI_MFDDF	Minimum FIFO Depth at DE Falling
0x1A	HDMI_FTR	FIFO Trend Register
0x1B	HDMI_FBR	FIFO Boundary Register
0x1C	HDMI_ICPSNCR0	I Code of Phase Swallow and N/CTS Register 0
0x1D	HDMI_ICPSNCR1	I Code of Phase Swallow and N/CTS Register 1
0x1E	HDMI_PCPSNCR0	P Code of Phase Swallow and N/CTS Register 0
0x1F	HDMI_PCPSNCR1	P Code of Phase Swallow and N/CTS Register 1
0x20	HDMI_ICTPSR0	I Code of Trend for Phase Swallow Register 0
0x21	HDMI_ICTPSR1	I Code of Trend for Phase Swallow Register 1
0x22	HDMI_PCTPSR0	P Code of Trend for Phase Swallow Register 0
0x23	HDMI_PCTPSR1	P Code of Trend for Phase Swallow Register 1
0x24	HDMI_ICBPSR0	I Code of Boundary for Phase Swallow Register 0
0x25	HDMI_ICBPSR1	I Code of Boundary for Phase Swallow Register 1
0x26	HDMI_PCBPSR0	P Code of Boundary for Phase Swallow Register 0
0x27	HDMI_PCBPSR1	P Code of Boundary for Phase Swallow Register 1
0x28	HDMI_NTx1024TR0	Number of Tx in 1024 Tv Register 0
0x29	HDMI_PCBPSR1	Number of Tx in 1024 Tv Register 1
0x2A	HDMI_STBPR	Stop Time for Boundary PE Register
0x2B	HDMI_NCPER	N and CTS Phase Error Register
0x2C	HDMI_PETR	Phase Error Threshold Register
0x2D	HDMI_AAPNR	Action for Audio PLL Non-Lock Register
0x2E	HDMI_APDMCR	Audio PLL Debug Mode Control Register
0x30	HDMI_AVMCR	Audio and Video Mute Control Register
0x31	HDMI_WDCR0	Watch Dog Control Register 0
0x32	HDMI_WDCR1	Watch Dog Control Register 1
0x33	HDMI_WDCR1	Watch Dog Control Register 2
0x34	HDMI_DBCR	HDMI Double Buffer Control Register
0x35	HDMI_APTMCR0	Audio PLL Test Mode Control Register 0
0x36	HDMI_APTMCR1	Audio PLL Test Mode Control Register 1
0x38	HDMI_DPCR0	DPLL Control Register 0
0x39	HDMI_DPCR1	DPLL Control Register 1
0x3A	HDMI_DPCR2	DPLL Control Register 2
0x3B	HDMI_DPCR3	DPLL Control Register 3
0x3C	HDMI_SUMCM	SDM SumC
0x3D	HDMI_SUMCL	SDM SumC
0x40	HDMI_AWDSR	Audio Watch Dog Status Register
0x41	HDMI_VWDSR	Video Watch Dog Status Register
0x42	HDMI_PAMICR	Packet Acquire Mechanism Interrupt Control Register
0x43	HDMI_PTRSV1	Packet Type of RSV1 Packet

0x44	HDMI_PTRSV2	Packet Type of RSV2 Packet
0x45	HDMI_PVGCR0	Packet Variation Global Control Register 0
0x46	HDMI_PVGCR1	Packet Variation Global Control Register 1
0x47	HDMI_PVGCR2	Packet Variation Global Control Register 2
0x48	HDMI_PVSR0	Packet Variation Status Register 0
0x49	HDMI_PVSR1	Packet Variation Status Register 1
0x4A	HDMI_PVSR2	Packet Variation Status Register 2
0x50	HDMI_VCR	Video Control Register
0x51	HDMI_ACRCR	ACR Control Register
0x52	HDMI_ACRSR0	ACR Status Register 0
0x53	HDMI_ACRSR1	ACR Status Register 1
0x54	HDMI_ACRSR2	ACR Status Register 2
0x55	HDMI_ACRSR3	ACR Status Register 3
0x56	HDMI_ACRSR4	ACR Status Register 4
0x57	HDMI_ACS0	Audio Channel Status 0
0x58	HDMI_ACS1	Audio Channel Status 1
0x59	HDMI_ACS2	Audio Channel Status 2
0x5A	HDMI_ACS3	Audio Channel Status 3
0x5B	HDMI_ACS4	Audio Channel Status 4
0x60	HDMI_INTCR	HDMI Interrupt Control Register
0x61	HDMI_ALCR	Audio Layout Control Register
0x62	HDMI_AOCR	Audio Output Control Register
0x70	HDMI_BCSR	HDMI Basic Coding Status Register
0x71	HDMI_ASR0	Audio Status Register 0
0x72	HDMI_ASR1	Audio Status Register 1
0x80	TMDS_DPC_SET0	Deep Color Setting 0
0x81	TMDS_DPC_SET1	Deep Color Setting 1
0x82	TMDS_DPC_SET2	Deep Color Setting 2
0x83	TMDS_DPC_SET3	Deep Color Setting 3
0x84	TMDS_DET_0	TMDS Decoding Error Detection 0
0x85	TMDS_DET_1	TMDS Decoding Error Detection 1
0x86	TMDS_DET_2	TMDS Decoding Error Detection 2
0x87	TMDS_DET_3	TMDS Decoding Error Detection 3
0x88	TMDS_DET_4	TMDS Decoding Error Detection 4
0x90	AUDIO_FREQDET	Audio Frequency Detect
0x91	AUDIO_FREQDET_RESULT_M	Audio Frequency Detect Result
0x92	AUDIO_FREQDET_RESULT_L	Audio Frequency Detect Result
0x93	XTAL_DIV	Crystal Divider
0x94	RANGE0_M	Threshold of Range 0 Detection
0x95	RANGE0_L	Threshold of Range 0 Detection
0x96	RANGE1_L	Threshold of Range 1 Detection
0x97	RANGE2_M	Threshold of Range 2 Detection
0x98	RANGE2_L	Threshold of Range 2 Detection
0x99	RANGE3_L	Threshold of Range 3 Detection
0x9A	RANGE4_M	Threshold of Range 4 Detection
0x9B	RANGE4_L	Threshold of Range 4 Detection
0x9C	RANGE5_L	Threshold of Range 5 Detection
0x9D	PRESET_S_CODE_0	S Code of Range 0
0x9E	PRESET_S_CODE_1	S Code of Range 1
0x9F	PRESET_S_CODE_2	S Code of Range 2
0xA0	PRESET_S_CODE_3	S Code of Range 3
0xA1	PRESET_S_CODE_4	S Code of Range 4

0xA2	PRESET_S2_CODE	S1 MSB Code of All Range
0xA3	AFSM_MOD	Audio FSM Mode
0xA4	HDMI_DYNAMIC_I_CTRL	Dynamic I Code Control
0xA5	HDMI_INI_ICB_M	Initial I Code
0xA6	HDMI_INI_ICB_L	Initial I Code

Register:: HDMI_SR			**0xCB**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
AVMUTE_BG	7	R	0	**AV_MUTE Flag under Background (write 1 clear)** 1: Means HW receive Set_AVMUTE flag of General Control packect
AVMUTE	6	R	0	**AV_MUTE flag of General Control Packet** 0: If HW receive Clear _AVMUTE flag of General Control Packet ,this bit shall assign to 0 until HW receive Set _AVMUTE 1: If HW receive Set_AVMUTE flag of General Control Packet ,this bit shall assign to 1 until HW receive Clear_AVMUTE **Note :** If HW never receives "General Control Packet", this bit shall set to 0. If HW receive "General Control Packet" with Clear_AVMUTE flag = 0 & Set_AVMUTE flag = 0, this bit shall keep previous value. If HW receive "General Control Packet" with Clear_AVMUTE flag = 1 & Set_AVMUTE flag = 1, this bit shall keep previous value, but set "General Control Packet error flag".
VIC	5	R	0	**If VIC(In AVI Infoframe) is different with pervious value ,this bit would be assigned to 1 until clear this bit. (write 1 clear for each bit)**
SPDIFTYPE	4	R	0	**SPDIF coding type** 0: LPCM 1: Non-LPCM
PLLSTS	3	R	0	**PLL status. This bit is global status, we could watch more detail information in PLL detail status byte.** (write 1 clear for each bit) 1: non-lock 0: lock
AFIFOOF	2	R	0	**0: Audio FIFO isn't overflow for X samples** **1: Audio FIFO is overflow for X sample** **(write 1 clear for each bit)** If audio FIFO has stayed at overflow state for X-sample periods, this bit would be set to '1' until F/W clear this bit.
AFIFOUF	1	R	0	**0: Audio FIFO isn't underflow for Y samples** **1: Audio FIFO is underflow for Y sample** **(write 1 clear for each bit)** If audio FIFO has stayed at underflow state for Y-sample periods, this bit would be set to '1' until F/W clear this bit.
MODE	0	R	0	**HDMI/DVI mode detected by auto function, even in manual mode, this bit could indicate decision of auto function.** 0: DVI 1: HDMI

FW should read "PLL status" after 0.66ms~3 ms from FW clear this bit.

Register:: HDMI_GPVS			**0xCC**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
NPS	7	R	0	**Null Packet Status**
PIS	6:5	R	0	**Packet Input Status** 6: RSV1 received 5: RSV0 received
PVS	4:0	R	0	**Packet Variation Status** 0: AVI infoframe 1: Audio infoframe 2: ACP

				3: ISRC1 4: MPEG infoframe

Note. Write 1 Clear

"Packet variation status":
1. "Packet variation status" means packet content variation, bit4 ~ bit 0 corresponds to AVI info-frame, audio info-frame, ACP, ISRC1, and MPEG info-frame respectively.
2. Before FW process the corresponding action item, FW should clear the corresponding bit of "Global Packet variation status".
3. Then FW read the content of the corresponding packet, polling "Global Packet variation status", check if corresponding bit of "Global Packet variation status" is 0, and execute follow-up action item if .this bit is 0.
4. Jump to step 2 if this bit is 1.
5. The variation result appears in "Global Packet variation status" after the corresponding packet finish transmitting.

"Packet input status":
1. "Packet input status" represents updated status of RSV1, RSV0 respectively. If it is updated, "Packet input status" is assigned to 1 until F/W clear this bit.
2. "Null Packet status" :When receive null packet , "Null Packet status" is assigned to 1until F/W clear this bit
3. If one bit of "Packet variation status" is cleared, the corresponding bit of "local variation flag for detail info" is also cleared.

Register:: HDMI_PSAP			**0xCD**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
APSS	7:0	R/W	0	**Address for Packet Storage SRAM**

Register:: HDMI_PSDP			**0xCE**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
DPSS	7:0	R	0	**Data Port for Packet Storage SRAM**

BCH is stored in the 1st address of each packet type, its content is stated as following;
Bit0: 2-bit error for bch header (0: 2-bit error doesn't occur; 1: 2-bit error occurs)
Bit1: 2-bit error for bch block 0 (0: 2-bit error doesn't occur; 1: 2-bit error occurs)
Bit2: 2-bit error for bch block 1 (0: 2-bit error doesn't occur; 1: 2-bit error occurs)
Bit3: 2-bit error for bch block 2 (0: 2-bit error doesn't occur; 1: 2-bit error occurs)
Bit4: 2-bit error for bch block 3 (0: 2-bit error doesn't occur; 1: 2-bit error occurs)
Bit5: checksum result (0: checksum error doesn't occur; 1: checksum error occurs)

Packet Type and Address

Packet type	Variation status	Storage (byte) (+ means BCH)	Address needed (8 bits/add)	Address
AVI info	9+1(global)	16+	17	0~16
Audio info	4+1	8+	9	17~25
ACP	3+1	4+	5	26~30
ISRC1	1+1	18+	19	31~49
ISRC2	X	18+	19	50~68
MPEG info	3+1	8+	9	69~77
RSV0	1, only global	30+	31	78~108
RSV1	1, only global	30+	31	109~139

Table 2 Packet Type and Address SRAM map Table

Following register is assigned by "HDMI-address port", "HDMI-data port"

Register:: HDMI_SCR			**0xCA-00**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Reserved	7:6	--	0	Reserved to 0
DVI_ACLK_MODE	5	R/W	1	**When switch to DVI mode, select audio clock control mode** **0: Old mode (Gate audio PLL output clk, FIFO R/W & Data output)** **1: New mode (Audio PLL output clk remains, but FIFO R/W & Data**

				output are gated)
REG_PACKET_IGNORE	4	R/W	0	**HDMI/DVI Mode Detection Method Selection** 0: Detect by both Video and Data Island Guard Band. 1: Detect by only Video Guard Band. (Can't be used when auto mode)
MODE	3	R/W	0	**HDMI/DVI switch mode** 0: Auto detect flow is as Fig.1 1: Manual, determined by bit[2]
MSMODE	2	R/W	0	**When manual mode, select HDMI or DVI** 0: DVI 1: HDMI
CABS	1	R/W	0	**DVI/HDMI condition A, B select** 0: **condition A**: Detect data island preamble + data island guard band (appear count is decided by "N") **condition B**: Detect if data island preamble + data island guard band appear in continuous 30 or 2 frames(decide by bit 0) 1: **condition A**: Detect data island preamble + data island guard band & video preamble + video guard band(appear count is decided by "N") **condition B**: Detect if data island preamble + data island guard band & video preamble + video guard band appear in continuous 30 or 2 frames(decide by bit 0)
FCDDIP	0	R/W	0	**Frame count to detect data island packet (Condition B)** 0: 2 frames 1: 30 frames

1. HDMI/DVI auto switch mode , the information must be passed to HDCP :
DVI/HDMI decision flow is shown as below.

DVI/ HDMI decide flow

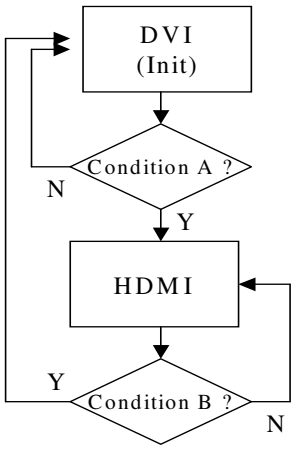


Fig 1

2. Power Saving for HDMI/HDCP :
In Power saving mode, TMDS channel Green/Red are always turn off. HDMI is power down.
There are only TMDS clock input frequency detect and channel blue DE decoder working.
The channel blue DE decoder is active after clock frequency is OK.

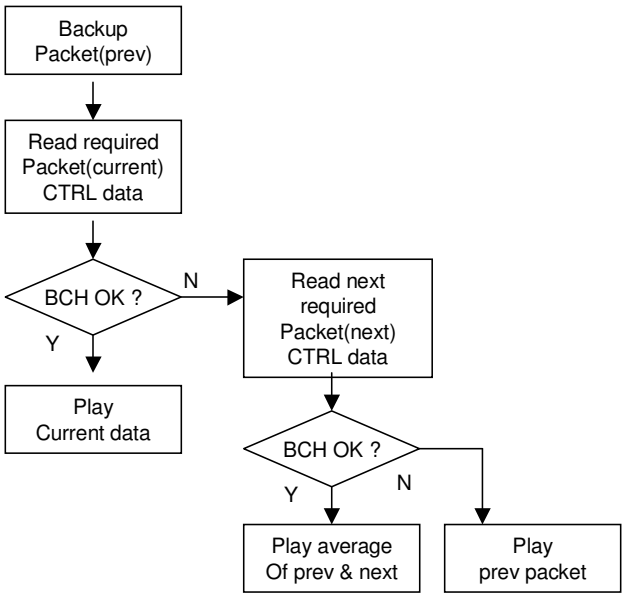
Register:: HDMI_N_VAL				**0xCA-01**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
NVAL	7:0	R/W	1	**The N value used to detect guard band (refer to 0xC9-00)** **N=** 00 : X 01 : 1 FF : 255 N = 1 ~ 255 , N can't be assigned to 0x00

Register:: HDMI_BCHCR				**0xCA-02**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Reserved	7:6	--	--	Reserved
BCH_FLAG_CLR_DVI	6	R/W	0	In DVI mode, clear BCH error flag (0x02[2:1]) 0: keep old method (write 0 to clear) 1: auto clear BCH flag
SPCSS	5	R/W	0	**SPDIF preamble channel status Source** (Can be used when PLL is non-lock) 0: Input audio sample (normal) 1: Internal system
ENRWE	4	R/W	0	**Enable noise reduction when BCH error is greater than one.** 1: Enable noise reduction 0: Disable noise reduction
BCHE	3	R/W	1	**BCH function enable** 1: Enable BCH function 0: Disable BCH function, bit[2:1] are always 2'b00.
BCHES	2	R	0	**BCH function's result, one bit error. It is set by this case, and cleared by write 0.** This bit is the result of ORing 5 bits BCH 1 bit error. 1: One bit error occurs. 0: No error occurs Note: If BCH detect 1-bit error, this bit would be assigned to 1 until clear this bit

BCHES2	1	R	0	**BCH function's result, two bits error. It is set by this case, and cleared by write 0.** This bit is the result of ORing 5 bits BCH 1 bit error. 1: 2-bit error occurs 0: 2-bit error don't occurs If BCH detect 2-bit error, this bit would be assigned to 1 until clear this bit
PE	0	R/W	0	**The processing for Packet with two or more BCH error (not include Audio packet)** 1: Block Info frame message 0: As correct frame, decided by F/W NOTE! Audio samples always go to FIFO

Register:: HDMI_AFCR				**0xCA-03**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
TST_I2S_SW	7	R/W	0	**Switch of Embedded Test Signal for I2S and DAC** 0: Old version, using test_cnt[5:1] as reference counter. 1: New version, using test_cnt[4:0] as reference counter.
AOEM	6	R/W	1	**Audio Output Enable mode** 1: Auto audio output flow, bit[5:0] could be assigned by HW, but couldn't be assigned by FW. 0: Manual audio output flow, bit[5:0] could be assigned by FW, but couldn't be assigned by HW.
AOC	5	R	0	**Audio output on/off control** 0: Audio output off, cut off audio output immediately in "manual audio output flow" , and audio output is turned on by auto audio output flow gradually in "auto audio output flow". 1: Audio output on, switch on audio output immediately in "manual audio output flow", and audio output is turned on by auto audio output flow gradually in "auto audio output flow".
AUDIO_TEST_ENABLE	4	R/W	1	**0:Disable** **1:Generate sine wave to IIS/SPDIF internally** This is assigned to "1" in IIS/SPDIF test mode, but it is assigned to "0" in normal mode.
MGC	3	R/W	0	**Manual Gain contro**l 1: Enable gain control, gain is decided by "Manual Audio Gain coefficient" 0: Disable gain control, gain = 1
AFIFOWE	2	R/W	0	**Audio FIFO write enable (FIFO pointer will be reset when disable)** 0: Disable, no audio sample would go in audio FIFO. This bit would clear Audio FIFO status, including read/write address, ovfl, unfl, and etc. 1: Enable FIFO audio Write, and enable bit[1:0] function, read control . (If buffer write to target depth, new data read out action is controlled by bit1).
AFIFORE	1:0	R/W	0	**Audio FIFO read enable, this bit is only active when bit[2] = 1,** 00: No audio frequency read, only drop old data when new data in. 01: Audio sample which read form FIFO repeats previous sample, only drop old data when new data in. 1x: Use audio frequency to read out FIFO.

Audio noise reduction 1



Register:: HDMI_AFSR				**0xCA-04**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Reserved	7:6	---	0	**Reserved**
BISTR	5	R	1	**Audio FIFO BIST Result** 0: fail 1: success
BISTS	4	R/W	0	**Audio FIFO BIST Start** (embedded test pattern) 0: stop 1: start(auto clear)
AFIFOF	3	R	0	**Audio FIFO Full (write clear)** 0: Indicate FIFO is not full. 1: Indicate FIFO is full.
AFIFOE	2	R	0	**Audio FIFO Empty(write clear)** 0: Indicate FIFO is not empty. 1: Indicate FIFO is empty.
AFIFO_DEPTH_REACH	1	R	0	**Current Aufio FIFO Depth Reach Target Depth** **0: Not yet reached** **1: Reached** **(Target Depth is set at 0xCA-1B[7:3])**
Reserved	0 ~~1:0~~	---	0	**Reserved to 0**

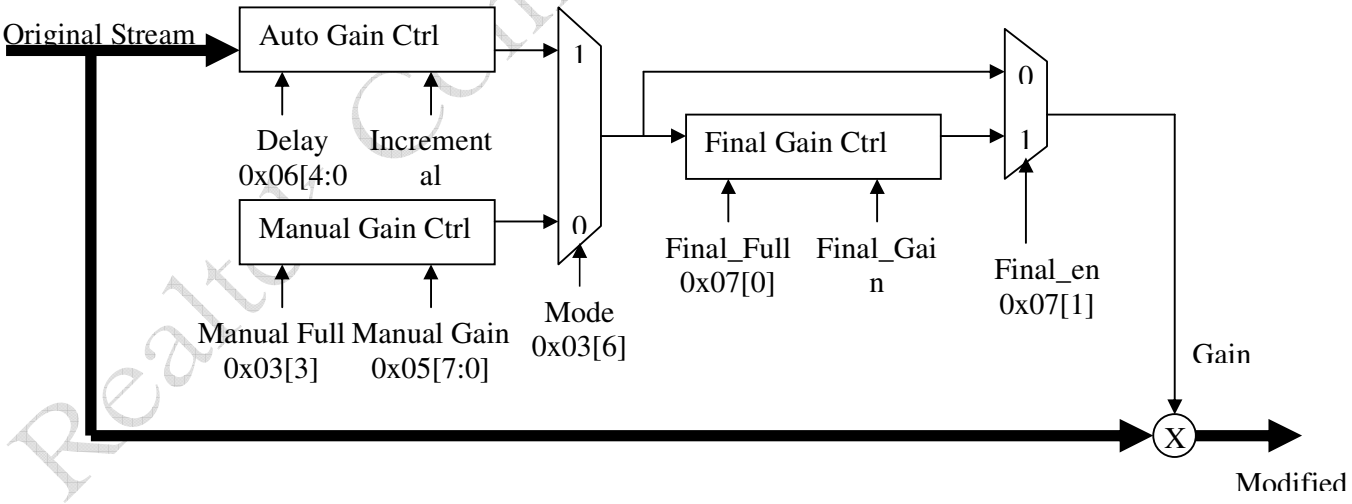
Register:: HDMI_MAGCR				**0xCA-05**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
MG	7:0	R/W	0	**Manual Gain**. Unsigned floating. NOTE, gain value here is always less than 1**.** 8'h00 = 0 8'hFF = 1 – 2^-8

Only valid when "Manual Gain control" is enabled in "manual audio output flow"

Register:: HDMI_MAGCR	**0xCA-06**

Name	Bits	R/W	Reset State	Comments
AGI	7:5	R/W	4	**Auto Gain Incremental** 000 : 2^ -8 001 : 2^ -7 010 : 2^ -6 … 111 : 2^ -1
AGD	4:0	R/W	4	**Auto Gain Delay** 00000 : 2^0 sample 00001 : 2^1 samples 00010 : 2^2 samples … 00111 : 2^7 samples 01000 : 2^8 samples 01001 : 2^9 samples 01010 : 2^10 samples … 01111: 2^15 samples 10000: 2^16 samples 10001: 2^17 samples 10010: 2^18 samples 10011: 2^19 samples 10100: 2^20 samples Others: Not used. The total meanings of this byte are: When Auto audio function is on, gain increase from 0 to 1 with 'incremental' per 'delay'. When Auto audio function is off, gain decrease from 1 to 0 with '-inc' per 'delay'. So that the default value means increase 2^-5 per 16 samples.

Only valid in "auto audio output flow"



Register:: HDMI_MAG__M_FINAL			0x CA-07	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Reserved	7:4	--	0	Reserved.
AUTO_DLY_MOD	3	R/W	1	Automatically modify the "Auto Gain Delay"

				0: Keep original "Auto Gain Delay" 1: According the Auto Audio Sampling Rate Detection Function to adjust the "Auto Gain Delay" automatically. If AFREQ_MEAS_RANGE = = 000: Auto Gain Delay' = Auto Gain Delay 001: Auto Gain Delay' = Auto Gain Delay/4 010: Auto Gain Delay' = Auto Gain Delay/4 011: Auto Gain Delay' = Auto Gain Delay/2 100: Auto Gain Delay' = Auto Gain Delay 101: Auto Gain Delay' = Auto Gain Delay Others: Auto Gain Delay' = Auto Gain Delay
FG_EN	2	R/W	1	Enable Final Gain control 1'b0: Original Design 1'b1: Enable Final Gain Control
FFG_L	1	R/W	1	**Left Channel F Gain**. Full gain 1'b0: Gain is controlled by 0x08 1'b1: Full gain.
FFG_R	0	R/W	1	**Right Channel F Gain**. Full gain 1'b0: Gain is controlled by 0x09 1'b1: Full gain.

Register:: HDMI_MAG_L_FINAL			**0x CA-08**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
FG_L	7:0	R/W	0	**F Gain**. Unsigned floating. NOTE, gain value here is always less than 1. 8'h00 = 0 8'hFF = 1 – 2^-8

Register:: HDMI_MAG_R_FINAL			**0x CA-09**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
FG_R	7:0	R/W	0	**F Gain**. Unsigned floating. NOTE, gain value here is always less than 1. 8'h00 = 0 8'hFF = 1 – 2^-8

Register:: AUDIO_LD_P_TIME_M			**CA-0x0A**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
RESERVED	7:3	--	0	Reserved.
LDP_TIME_MODE	2	R/W	1	The mode of Wait-TIME for loading parameter 0: Old mode: 1024*Xclk 1: New mode: LDP_TIME * 200us
LDP_TIME[9:8]	1:0	R/W	0	Wait-time for HW getting stable. (Base clock is 5kHz), During this period, there'll be no audio outputted. 0: 200us 1: 400us … 1023: 0.2048s

Note: the LDP_TIME[9:8] will be double-buffered when 0x0B is written.

Register:: AUDIO_LD_P_TIME_N			**CA-0x0B**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
LDP_TIME[7:0]	7:0	R/W	2	Wait-time for HW getting stable. (Base clock is 5kHz) 0: 200us 1: 400us … 1023: 0.2048s

Zero-Crossing Detection & Application of Final Gain

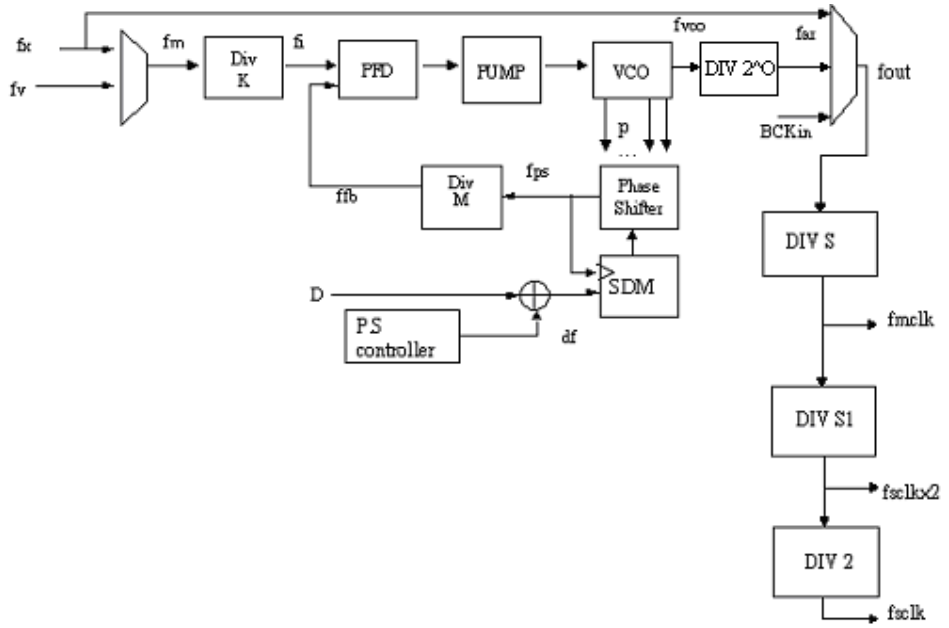
Register:: ZCD_CTRL			0x CA-0C	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
ZCD_EN	7	R/W	1	Final Gain controlled by zero crossing detection 0: Disabled. 1: Enabled. (0x07[1:0] & 0x08 & 0x09 are applied when zero-crossing)
ZCD_SEP_8CH	6	R/W	1	Final Gain (8-ch) 0: 8-channel volumes are controlled at the same time. (Once any one of 8-channel detects zero-crossing, then apply the final gain to all channels) 1: Determined by .0x0C[5]
ZCD_SEP_ST	5	R/W	1	Final Gain (Stereo) 0: L/R volumes are controlled at the same time. (Once L or R detects zero-crossing, then apply the final gain to both channels) 1: L/R volumes are controlled separately.
RESERVED	4:0	R/W	0	Reserved

Register:: ZCD_TIMEOUT			0x CA-0D	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
RESERVED	7	R/W	0	Reserved
ZCD_TIMEOUT	6:0	R/W	3	Force to apply final gain when time-out. Interval = Xtal * 512 Timeout = Interval * {(ZCD_TIMEOUT +1), 3'b000}

ZCD_CHx_DONE is cleared when new gain value is set.

Register:: ZCD_STATUS			0x CA-0E	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
ZCD_CH7_DONE	7	R	--	ZCD of Channel 7 has been detected and Gain has been also applied. ~~Write 1 to clear.~~
ZCD_CH6_DONE	6	R	--	ZCD of Channel 6 has been detected and Gain has been also applied. ~~Write 1 to clear.~~
ZCD_CH5_DONE	5	R	--	ZCD of Channel 5 has been detected and Gain has been also applied. ~~Write 1 to clear.~~
ZCD_CH4_DONE	4	R	--	ZCD of Channel 4 has been detected and Gain has been also applied. ~~Write 1 to clear.~~
ZCD_CH3_DONE	3	R	--	ZCD of Channel 3 has been detected and Gain has been also applied. ~~Write 1 to clear.~~
ZCD_CH2_DONE	2	R	--	ZCD of Channel 2 has been detected and Gain has been also applied. ~~Write 1 to clear.~~
ZCD_CH1_DONE	1	R	--	ZCD of Channel 1 has been detected and Gain has been also applied. ~~Write 1 to clear.~~
ZCD_CH0_DONE	0	R	--	ZCD of Channel 0 has been detected and Gain has been also applied. ~~Write 1 to clear.~~

Audio Clock Regeneration



Definition :

fx : frequency of crystal	fps : frequency after P.S
fv : frequency of video	ffb : feed back frequency
fa : audio frequency	P.S : Phase Swallow
fout : 128 * fa	p : number of phase
far : recovered 128 * fa	D : P.S density, shift D phase per cycle
fm : freq. of mux-clock	df : fine tune of D
fvco : frequency after VCO	T* : Period of f*

NOTE!!! Signed number and detail procedures are not ready.

Register:: HDMI_CMCR			**0x CA-10**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
ICMUX	7	R/W	0	**Input Clock MUX** 1: use video clock as input 0: use crystal clock as input
OCS	6:5	R/W	2	**Output Clock Select** 00: use crystal clock as output clock. 01: use BCKin as output clock 1X: use generated clock, far, as output clock (must set when power-saving)
DBDCB	4	R/W	0	**Double Buffer Download Control Bit** Enable is also triggered by HW, ref. “Phase error mode”. 1: write current data to active buffer. 0: after write done, this bit would be cleared automatically. When set this bit to 1, “ K”, “S”, “S1”, “M”, “D”, “O”, “DPLLBPN”, “In/out clk mux”, “Phase tracking enable control bits” would fill in after finish current audio PLL cycle and then set this bit to 0.
KCAPLL	3:0	R/W	3	**K Code of Audio PLL**, the value set here adding 1 is real div value 0000: div 2 1111: div 17 If “DPLLBPN” == 1'b1, no div, else, div number is decided by these four bits.

NOTE:

1. When reading the registers with double buffers, the read-out value is the value in the 2nd buffer, not the value just written.

2. The meaning of default value of registers with double buffers is that default values of both 1st registers and 2nd buffer are the value written in spec.

Register:: HDMI_MCAPR			0x CA-11	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
MC	7:0	R/W	4E	**M Code** 00: div 2 FF: div 257

Register:: HDMI_SCAPR			0x CA-12	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
SLC	7	R/W	0	**S1 code** 0: div 1 1: div 2
SC	6:0	R/W	5	**S/2 code**

Register:: HDMI_DCAPR0			0x CA-13	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
DCAPR	7:0	R/W	0	**D[15:8]**

Register:: HDMI_DCAPR1			0x CA-14	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
DCAPR	7:0	R/W	0	**D[7:0]**

Register:: HDMI_PSCR			CA-0x15	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
FDINT	7:5	R/W	7	**When max. FIFO depth increase for n times or min. FIFO depth decrease for n times, turn FIFO tracking mechanism** 000 : xx 001 : n=2,don't use this value for normal case 010 : n=3 011 : n=4 100 : n=5 101 : n=6 110 : n=7 111 : n=8
ETCN	4	R/W	0	**Enable tracking of CTS & N** 0: disable. 1: enable.
ETFD	3	R/W	0	**Enable tracking of the trend of FIFO depth** 0: disable. 1: enable.
ETFBC	2	R/W	0	**Enable tracking of FIFO boundary condition** (This bit is suggested to be 1) 0: disable. 1: enable.
PECS	1:0	R/W	1	**Phase error count source(CTS & N)** 00 : phase error counted by video clock 01 : phase error counted by crystal clock 10 : phase error counted by fps/4, fdds 11 : It is too fast, about 500MHz,to be used

Note. Phase tracking control bits is bit4~bit2.

Total FIFO depth is 32 samples

Register:: HDMI_FDDR			0x CA-16	
Name	**Bits**	**R/W**	**Reset**	**Comments**

			State	
FDDR	7:0	R	0	**FIFO depth at DE rising**, this unit is number of samples,

Register:: HDMI_FDDF				**0x CA-17**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
FDDF	7:0	R	0	**FIFO depth at DE falling**

Register:: HDMI_MFDDR				**0x CA-18**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
MFDDR	7:0	R	0	**Max. FIFO depth at DE rising**. Auto clear to 0x00 when up-trend is confirmed and frequency up is triggered. Write 1 to clear this byte as 0x00.the clear action needs video clock to work.

Register:: HDMI_MFDDF				**0x CA-19**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
MFDDF	7:0	R	0	**Min. FIFO depth at DE falling**. Auto clear to 0xFF when down-trend is confirmed and frequency down is triggered. Write 1 to clear this byte as 0x00.the clear action needs video clock to work. Write 1 to clear.

Register:: HDMI_FTR				**0x CA-1A**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
TL2DER	7:6	R	0	**Trend of latest 2 DE rising**. 0X: the same 10: trend down, which means FIFO depth goes lower and lower. 11: trend up, which means FIFO depth goes larger and larger.
TL2DEF	5:4	R	0	**Trend of latest 2 DE falling**. 0X: the same 10: trend down, which means FIFO depth goes lower and lower. 11: trend up, which means FIFO depth goes larger and larger.
TT	3:0	R/W	7	**Target times for summation of one trend to decide the trend**. Times = value set + 1 0000 : 1, 1111 : 16

Register:: HDMI_FBR				**0x CA-1B**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
TFD	7:3	R/W	E	**Target FIFO depth**, the unit is 4 address, and 16 bits in one address.
BAD	2:0	R/W	2	**Boundary address distance** for triggering Audio PLL tracking where boundary address= value set * 4, and 16 bits per address. 4 bytes*16 bits is one sample. When the value is 2,number of sample is 0,1,31,and 32 will trigger boundary condition. Value 0 can't be used.

PI Code is double buffered when LSB is written

Register:: HDMI_ICPSNCR0				**0x CA-1C**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
IC	7:0	R/W	0	**I code of N/CTS [15:8]**

Register:: HDMI_ICPSNCR1				**0x CA-1D**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
IC	7:0	R/W	0	**I code of N/CTS [7:0]**

Register:: HDMI_PCPSNCR0	**0x CA-1E**

Name	Bits	R/W	Reset State	Comments
PC	7:0	R/W	0	**P code of N/CTS [15:8]**

Register:: HDMI_PCPSNCR1				**0x CA-1F**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
PC	7:0	R/W	0	**P code of N/CTS [7:0]**

Register:: HDMI_ICTPSR0				**0x CA-20**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
ICT	7:0	R/W	0	**I code of trend [15:8]**

Register:: HDMI_ICTPSR1				**0x CA-21**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
ICT	7:0	R/W	0	**I code of trend [7:0]**

Register:: HDMI_PCTPSR0				**0x CA-22**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
PCT	7:0	R/W	0	**P code of trend [15:8]**

Register:: HDMI_PCTPSR1				**0x CA-23**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
PCT	7:0	R/W	0	**P code of trend [7:0]**

Register:: HDMI_ICBPSR0				**0x CA-24**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
ICB	7:0	R/W	0	**I code of boundary [15:8]**

Register:: HDMI_ICBPSR1				**0x CA-25**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
ICB	7:0	R/W	0	**I code of boundary [7:0]**

Register:: HDMI_PCBPSR0				**0x CA-26**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
PCB	7:0	R/W	0	**P code of boundary [15:8]**

Register:: HDMI_PCBPSR1				**0x CA-27**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
PCB	7:0	R/W	0	**P code of boundary [7:0]**

Register:: HDMI_NTx1024TR0				**0x CA-28**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Reserved	7:5	---	0	**Reserved to 0**
S1_CODE_MSB	4	R/W	0	**S1 Code MSB bit** 0: Depends on LSB bit 1: div 4
RM	3	R/W	0	**Restart measure**. Measure the length of 1024 Tv by crystal. The result is readable from the following bits. 1: enable measure. Writing 1 would clear the answer. This bit would be auto cleared after measure done.

				0: indicating measure is done.
NT	2:0	R	0	**Number of Tx for 1024 Tv [10:8]**, (How many Tx = 1024 * Tv)

Register:: HDMI_NTx1024TR1				**0x CA-29**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
NT	7:0	R/W	0	**Number of Tx for 1024 Tv [7:0]**, (How many Tx = 1024 * Tv)

Register:: HDMI_STBPR				**0x CA-2A**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
FTB	7:0	R/W	0	**The fast time for boundary df repeating**. The unit is 16 crystal clock. 8'h00: 16 crystal clock. 8'h7F: 128 * 16 crystal clock.

Register:: HDMI_NCPER				**0x CA-2B**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
NCPER	7:0	R	0	**Phase error equals how many numbers of measuring clock, PE[7:0]**

NOTE!! The active PI code of CTS&N would have proportional alike relation with Phase error.
The value of this byte is record of the maximum value after last write.
Write this byte when fpec exists would clear the value to 0.
When "pe_mode"==1, delay mode, the max value of phase error is 40.
When "pe_mode"==1, clock mode, the max value of phase error is FF.

Register:: HDMI_PETR				**0x CA-2C**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
PETR	7:0	R/W	FF	**Phase error threshold of audio PLL non-lock**

If "Phase error" is greater than phase error threshold, "PLL status" would be automatically assigned to 1 until FW clear it.

Register:: HDMI_AAPNR				**0x CA-2D**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
CMVTC	7	R/W	0	**Clear max value (18, 19) when trend condition is true.** 1: Enable trend to clear max value 0: Disable this function
CMVBC	6	R/W	0	**Clear max value (18, 19) when boundary condition is true.** 1: Enable boundary to clear max value 0: Disable this function
SSDMOU	5	R	0	**Flag of sum_r of SDM overflow/underflow (Read only)** 1: Overflow or underflow happened 0: No overflow, no underflow
TEF	4	R/W	0	**Trend Error Flag** 1: Detect up and down at the same time. Clear only when disable SDM (2D[1] = 0) 0: Trend is ok.
W1C5	3	W	0	**Write 1 to clear bit [5]**
PEM	2	R/W	0	**Phase Error Mode**, 1: Use delay to calculate, each unit is 0.1 ns. 0: Use clock to calculate, the clock select is at "PE count source".
ESDM	1	R/W	0	**Enable SDM**(phase swallow) 1: Enable 0: Disable, there won't be phase swallow operating in the loop of PLL.
Reserved	0	---	0	**Reserved**

Register:: HDMI_APDMCR				**0x CA-2E**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Reserved	7:6	---	0	**Reserved**
EDM	5	R/W	0	**Enable Debug Mode**

				0: Normal run 1: Enable when test mode
PST	4	R/W	0	**Phase swallow trend** 0: Fast direction 1: Slow direction
PSC	3:0	R/W	0	**Phase Swallow Cycle.** Any bit is set to 1 for swallow, 0 for hold.

Behavior description of audio PLL non-clock

When system receive new audio or video timing , audio PLL would non-lock ,and watch dog mechanism would force audio output to mute state(I2S DAC: MCLK,SCLK, and LRCK normal output, but SDATA output zero),so system should provide a stable fout to I2S DAC in audio mute state.

In the transition form normal fout to mute fout , fout frequency couldn't change too much, for this reason ,HW provide double buffers of mechanism of " K", "S", "S1", "M", "D", "O", "DPLLBPN", "In/out clk mux", "Phase tracking enable control bits".

For initial state, a stable fv input to audio PLL, and audio PLL would lock by use suitable "KMSDO"&PI code.The suitable "KMSDO" could be named as "KMSDO1",and it would save in 2nd buffer(The value of 2nd buffer could be applied to audio PLL directly , and that of 1st buffer is used to backup, when "double buffer download control bit" is assigned to 1,the value of 1st buffer would be downloaded to 2nd buffer).F/W should calculate "KMSDO2" of crystal clock input to produce a fout which is the same as present fout ,then save KMSDO2 in 1st buffer of KMSDO, F/W also assign "phase tracking control bits" to 000'b in 1st buffer, and assign "input clock mux" to "crystal input" in 1st budder.

When audio PLL is non-lock(change audio frequency or video frequency),the 1st buffer content of "KMSDO", "phase tracking control bits", and "input clock mux" would download to their corresponding 2nd buffers. Then audio PLL would switch input to crystal in, apply KMSDO2,and disable phase tracking at the same time, and provide a stable fout to I2SDAC foe mute state.

In mute state, F/W calculate KMSDO(KMSDO3) of new audio or video timing, assign KMSDO3 in 1st buffer of KMSDO, F/W also assign "enable setting" in 1st buffer of phase tracking enable control, and assign "video input" to second buffer of input clock mux.

Assign PI code, then double buffer download control bit is assigned to 1, audio PLL would switch input to video in, apply KMSDO3, and disable phase tracking at the same time, and provide a fout for new video and audio timing.

Register:: HDMI_AVMCR			**0x CA-30**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
AVMUTE_WIN_EN	7	R/W	0	**Avmute Window Enable** 1: Enable 0: Disable If this bit is enabled, avmute signal will be blocked when in the invalid region. The valid region depends on the setting of register 0xA8
AOC	6	R/W	0	**Audio output enable/disable control** 1: Enable 0: Disable If this bit is enabled, audio output signal would be controlled by bit4. When FW set this bit to 1, then HW will return this bit to 0 if audio PLL non-lock if audio PLL non-lock.
AOMC	5	R/W	0	**Audio Output Mute Control** 1: Normal output 0: Mute If bit 5 is 0, output of I2S & SPDIF shall be disabled regardless of 1 or 0 in this bit for "auto audio output flow". When FW set this bit to 1,then HW will return this bit to 0 if audio PLL is non-lock
AWD	4	R/W	0	**If Audio Watch Dog event occur, audio output would be** 0: Mute 1: Disable

VE	3	R/W	0	**Video clock output Enable. When video watch dog occurs, this bit will be reset to 0.** 1: Enable video clock output 0: disable video clock output
AMPIC	2	R/W	0	**Audio Mute Pin Invert Control, execute when mute/disable happens**. 0: when event (audio mute or disable) occur, set this pin to low voltage, others maintain high. 1: when event (audio mute or disable) occur, set this pin to high voltage, others maintain low
VDPIC	1	R/W	0	**Video Disable Pin Invert Control** 0: when event (video disable) occurred, set this pin to low voltage, others maintain high. 1: when event (video disable) occurred, set this pin to high voltage, others maintain low.
NFPSS	0	R/W	0	**IRQ Output Pin Polarity Inverse** 0: no inverse, which means H : IRQ, L : no IRQ 1: inverse, which means H : no IRQ L : IRQ

Definition:

Disable Video Assign "DE pins", "VS pin", "HS pin", "CTRL(4) pins", "CLK pin", "Data(24) pins" to zero , refer to "Global System"

Mute Audio
1. In I2S application, keep MCLK*4, SCLK*4, and LRCK*4 to normal output,but cut SDATA*4 to zero.
2. In SPDIF application,keep preamble(M,B,W) to normal output, but cut other bits to zero.

Disable Audio I2S => assign MCLK*4, SCLK*4, and LRCK*4 and SDATA*4 to zero.
SPDIF => Assign all bits to zero.

Register:: HDMI_WDCR0			**0x CA-31**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
ASMFE	7	R/W	1	**Auto SET_AVMUTE function enable** 0: If HW receives SET_AVMUTE flag, don't mute/disable audio & disable video by HW. 1: If HW receives SET_AVMUTE flag, mute/disable audio & disable video by HW. *Note:* If "CLEAR_AVMUTE" and "SET_AVMUTE" of the General Control Packet are all 1, keep previous A/V output state, and pull up "General Control Packet error flag"
load_d_code	6	R/W	0	**Enable Load DCode** 0: disable 1: enable
audio_wd_by_vic	5	R/W	0	**In audio FIFO tracking process(trend and boundary tracking), when phase error occurs, the behavior of I/P code:** 0: I/P code doesn't work, i.e., I/P code of HW use = 16'b0 1: I/P code work, i.e., I/P code of HW use = register setting
AWDCT	4	R/W	0	**Audio watch dog for audio coding type(Decode from SPDIF, code type only include LPCM or Non-LPCM)** 0: If coding type is different with previous type, don't mute/disable audio by HW. 1: If coding type is different with previous type, mute/disable audio by HW.
AWDAP	3	R/W	0	**Audio Watch dog enable for audio PLL** 0: If audio PLL is non-lock, don't mute/disable audio by HW. 1: If audio PLL is non-lock, mute audio , mute/disable audio by HW.
AWDFO	2	R/W	0	**Audio watch dog function for audio FIFO overflow for "X" sample.** 0: If audio FIFO is overflow for X samples , don't mute/disable audio by HW. 1: If audio FIFO is overflow for X samples, mute/disable audio by HW.
AWDFU	1	R/W	0	**Audio watch dog function for audio FIFO underflow for "Y" sample.** 0: If audio FIFO is underflow for Y samples, don't mute/disable audio by HW.

				1: If audio FIFO is underflow for Y samples, mute/disable audio by HW.
CT	0	R/W	0	**"SPDIF coding type" is decoded by** 0: Channel status bit 1 1: Valid bit

Audio/Video watch dog for "packet acquire mechanism" is listed in Packet acquire mechanism Unit.

Register:: HDMI_WDCR1			**0x CA-32**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
AWDCK	7	R/W	0	**Audio Watch Dog For TMDS clock** 1: If TMDS clock disappears, mute or disable audio. 0**:** If TMDS clock disappears, doesn't mute or disable audio.
AWDLF	6	R/W	0	**Audio Watch Dog For Layout Field Of Audio Sample Packet** 1: If layout field is different with previous value, mute or disable audio. 0: If layout field is different with previous value, don't mute or disable audio.
~~Rev~~ VWDMOD	5	R/W	0	~~**Reserved**~~ **Video Watch Dog Mode** 0: Old Mode (Gated Clk) 1: New Mode (RGB Output Disable)
VWDACT	4	R/W	0	**Video Watch Dog For Audio Coding Type** 1: If coding type is different with previous type, disable video 0: If coding type is different with previous type, don't disable video
XV	3:0	R/W	0	**X Value** 0000: 1 0001: 3 ~ 1111: 31

Register:: HDMI_WDCR2			**0x CA-33**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
VWDAP	7	R/W	0	**Video Watch dog enable for audio PLL** 1: If audio PLL is non-lock, disable video 0: If audio PLL is non-lock, don't disable video
VWDLF	6	R/W	0	**Video watch dog for layout field of audio sample packet** 1: If layout field is different with previous value, disable Video. 0: If layout field is different with previous value, don't disable Video.
VWDAFO	5	R/W	0	**Video watch dog function for audio FIFO overflow.** 1: If audio FIFO is overflow for "X" samples, disable Video. 0: If audio FIFO is overflow for "X" samples, don't disable Video.
VWDAFU	4	R/W	0	**Video watch dog function for audio FIFO underfloww .** 1: If audio FIFO is underflow for "Y" samples, disable Video. 0: If audio FIFO is overflow for "Y" samples, don't disable Video
YV	3:0	R/W	0	**Y value** 0000:1 0001:3 ~ 1111:31

Register:: HDMI_DBCR			**0x CA-34**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Reserved	7:4	R/W	0	**Reserved**
ALDBFv	3	R/W	0	**Auto Load Double Buffer** when TMDS clock disappear (0xC9-10[4]) 0: If TMDS clock disappear , don't assign "double buffer download control bit" to 1 by HW 1: If TMDS clock disappear, assign "double buffer download control bit" to 1 by HW. *Note:* If this bit is 0, "phase tracking control bits" shall be downloaded to 2nd buffer by assigned "double buffer download control bit" to 1. If this bit is 1, "phase tracking control bits" shall be downloaded to 2nd buffer by

				HW if fv < 25MHz or fv > 165MHz.
ALDBFO	2	R/W	0	**Auto Load Double Buffer when FIFO overflow is for X samples.** 0: If audio FIFO is overflow for X samples, don't assign "double buffer download control bit" to 1 by HW 1: If FIFO is overflow for X samples, assign "double buffer download control bit" to 1 by HW. *Note:* If this bit is 0, "phase tracking control bits" shall be downloaded to 2nd buffer by assigned "double buffer download control bit" to 1. If this bit is 1, "phase tracking control bits" shall be downloaded to 2nd buffer by HW if FIFO is overflow for X samples.
ALDBFU	1	R/W	0	**Auto Load Double Buffer when FIFO underflow is for Y samples.** 0: If audio FIFO is underflow for Y samples, don't assign "double buffer download control bit" to 1 by HW 1: If FIFO is underflow for Y samples, assign "double buffer download control bit" to 1 by HW. *Note:* If this bit is 0, "phase tracking control bits" shall be downloaded to 2nd buffer by assigned "double buffer download control bit" to 1. If this bit is 1, "phase tracking control bits" shall be downloaded to 2nd buffer by HW if FIFO is underflow for Y samples.
ALDBPN	0	R/W	0	**Auto Load Double Buffer when PLL non-lock.** This function needs crystal clock to work, which means it can't work when power down. After PLL non-lock, 0: If audio PLL non-lock occurred, don't assign "double buffer download control bit" to 1 by HW 1: If audio PLL non-lock occurred, assign "double buffer download control bit" to 1 by HW. *Note:* If this bit is 0, "phase tracking control bits" shall be downloaded to 2nd buffer by assigned "double buffer download control bit" to 1. If this bit is 1, "phase tracking control bits" shall be downloaded to 2nd buffer by HW if "PLL status" is non-lock.

Register:: HDMI_ APTMCR0			**0x CA-35**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
FPS	7:4	R/W	0	**1st phase shift amount for a step**
SPS	3:0	R/W	0	**2nd phase shift amount for a step**

Register:: HDMI_ APTMCR1			**0x CA-36**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Reserved	7	---	0	**Reserved**
PLLTM	6	R/W	0	**PLL test mode enable** 1: enable 0: disable
FPSD	5	R/W	0	**1st phase shift direction** 0: upwards 1: downwards
SPSD	4	R/W	0	**2nd phase shift direction** 0: upwards 1: downwards
NFPSS	3:0	R/W	0	**Number of 1st phase shift step**

In test mode, PLL shift its phase by 16 steps periodically. The steps which are performed in 1st phase each 16 steps could be assigned by "Number of 1st phase shift step", remaining steps are performed in 2nd phase.

Register:: HDMI_DPCR0			**0x CA-38**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
DPLLC2	7	R/W	1	**DPLLPWDN**

				0: power up 1: power down
DPLLC1	6	R/W	0	**DPLLFREEZE** 0: normal 1: freeze
DPLLC0	5:4	R/W	1	**DPLLO** div 2^(DPLLO)
DPLL_CALBP	3	R/W	0	**DPLL bypass calibration(active high)**
DPLL_CALSW	2	R/W	0	**calibration validated (go high after power on 1200us)**
DPLL_CALLCH	1	R/W	0	**latch calibration (go high after power on 1100us)**
DPLL_CMPEN	0	R/W	0	**cmp enable (go high after power on 1000us)**

Register:: HDMI_DPCR1				**0x CA-39**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
DPLL_RS	7:5	R/W	3	**DPLL Loop Filter Resister Control** RS: 000:16K 001:18K 010:20K 011:22K 100: 24K 101: 26K 110:28K 111:30K
DPLL_CS	4:3	R/W	2	**DPLL Loop Filter Capacitor Control** CS= 00:18p, 01:20p, 10:24p, 11:28p
DPLL_IP	2:0	R/W	2	**DPLL Charge Pump Current Control** Icp=(2.5uA+2.5uA*bit[0]+5uA*bit[1]+10uA*bit[2]) Keep DPM/Icp constant=10.67

Register:: HDMI_DPCR2				**0x CA-3A**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
DPLLSTATUS	7	R	0	**DPLLSTATUS(DPLL WD Status)** 0:Normal 1:Abnormal Write 1 to clear.(WD enable when Audio PLL power on)
DPLLWDRST	6	R/W	0	**DPLLWDRST(DPLL WD Reset)** 0:Normal 1:Reset
DPLLWDSET	5	R/W	0	**DPLLWDSET(DPLL WD Set)** 0:Normal 1:Set
DPLLVCOMD	4:3	R/W	3	**DPLL VCO Default mode** 00: VCO slowest, 11: VCO fastest
DPLLRESERVE	2	R/W	1	**DPLLRESERVE, phase swallow circuit clock select** 0: fvco, default is 1 1: fps
DPLLSTOP	1	R/W	1	**DPLLSTOP(DPLL Frequency Tuning Enable)** 0:Disable 1:Enable
DPLL_CP	0	R/W	0	**CP Control** 0:CP=1.77pF 1:CP=2.1pF

Register:: HDMI_DPCR3				**0x CA-3B**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
DPLL_VO2	7	R	0	**DPLL CAL OUT2**
DPLL_VO1	6	R	0	**DPLL CAL OUT1**
DPLL_CAL	5:4	R	0	**DPLL calibrated VCO code**
RESERVED	3	R/W	0	**Reserved.**
DPLLBPN	2	R/W	0	**DPLLBPN** 0: divider K enable 1: divider K disable(K=1)

DPLL_RESERVE1	1	R/W	0	**DPLL_RESERVE1**
DPLLVCORSTB	0	R/W	0	**RESET VCO (active high)**

Register:: HDMI_SUMCM			**0x CA-3C**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
SUMC[15:8]	7:0	R	-	**SDM SUMC[15:8]**

Register:: HDMI_SUMCL			**0x CA-3D**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
SUMC[7:0]	7:0	R		**SDM SUMC[7:0]**

Packet Acquire Mechanism

Register:: HDMI_AWDSR				0x CA-40
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Reserved	7:5	---	0	**Reserved to 0**
AWDPVSB	4:0	R/W	0	**Audio watch dog for Packet variation status bit**

If a bit is assigned to 1 and the corresponding bit of "Global Packet variation status" is 1, audio output will be disabled/muted.

Register:: HDMI_VWDSR				0x CA-41
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Reserved	7:5	---	0	**Reserved to 0**
VWDPVSB	4:0	R/W	0	**Video watch dog for Packet variation status bit**

If a bit is assigned to 1 and the corresponding bit of "Global Packet variation status" is 1, video output will be disabled.

Register:: HDMI_PAMICR				0x CA-42
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Reserved	7:5	---	0	**Reserved to 0**
ICPVSB	4:0	R/W	0	**IRQ control for Packet variation status bit**

If a bit is assigned to 1 and the corresponding bit of "Global Packet variation status" is 1, issue IRQ signal.
Note: The corresponding bit of "Global Packet variation status" means bit0 maps to bit 0 of "Global Packet variation status ,bit1" maps to bit 1 of "Global Packet variation status",…etc.

Register:: HDMI_PTRSV1				0x CA-43
Name	**Bits**	**R/W**	**Reset State**	**Comments**
PT	7:0	R/W	0	**Packet Type of RSV1 packet**

Register:: HDMI_PTRSV2				0x CA-44
Name	**Bits**	**R/W**	**Reset State**	**Comments**
PT	7:0	R/W	0	**Packet Type of RSV2 packet**

Register:: HDMI_PVGCR0				0x CA-45
Name	**Bits**	**R/W**	**Reset State**	**Comments**
PVSEF	7:0	R/W	FF	**Bit7 ~ Bit0 of packet variation status enable flag**

Register:: HDMI_PVGCR1				0x CA-46
Name	**Bits**	**R/W**	**Reset State**	**Comments**
PVSEF	7:0	R/W	FF	**Bit15 ~ Bit8 of packet variation status enable flag**

Register:: HDMI_PVGCR2				0x CA-47
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Reserved	7:4	---	0	**Reserved**
PVSEF	3:0	R/W	F	**Bit19 ~ Bit16 of packet variation status enable flag**

When the bits of enable "Packet Variation Global Control Register" are set, the corresponding "Packet Variation Status Register" bits will OR to "Packet Variation Global Control Register".

Register:: HDMI_PVSR0				0x CA-48
Name	**Bits**	**R/W**	**Reset State**	**Comments**
PVS	7:0	R	0	**Bit7 ~ Bit0 of packet variation status**

Register:: HDMI_PVSR1	0x CA-49

Name	Bits	R/W	Reset State	Comments
PVS	7:0	R	0	**Bit15 ~ Bit8 of packet variation status**

Register:: HDMI_PVSR2				**0x CA-4A**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Reserved	7:4	---	0	**Reserved**
PVS	3:0	R	0	**Bit19 ~ Bit16 of packet variation status**

There are 20 bits "Enable flags to global Packet variation". Each bit is set to watching a standard type of received packet content, and checking if it changed from the previous received packet.
If received packet content changed from previous received one, the relative bit in "local variation flag for detail info." register will be set, and it will trigger the "global packet variation status" set.
The following table presents the detail of "local variation flag for detail info."

InfoFrame	Bit	Description
AVI	0	Y0Y1change
	1	A0,R0,R1,R2,R3 change
	2	S0,S1 any bit change
	3	C0,C1 change
	4	M0,M1 change
	5	VIC0 ~ VIC6 change
	6	PR0 ~ PR6 change
	7	SC1,SC0 change
	8	B0,B1,Top bar, bottom bar, left bar , right bar change
Audio	9	CC0~CC3 change
	10	CA0~CA7 change
	11	LSV0~LSV3 change
	12	DM_INH any bit change
ACP	13	ACP_Type change
	14	DVD-audio_type_dependent_generation change
	15	Copy_Permission, Copy_Number,Quality,& Transaction change
ISRC1	16	ISRC_status change
MPEG	17	MB#3~MB#0 change
	18	FR0 change
	19	MF1,MF0 change

Register:: HDMI_VCR				**0x CA-50**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
EOI	7	R/W	0	**EVEN/ODD Inverse** 0: Normal 1: Inverse
EOT	6	R	0	**EVEN/ODD Toggle (write 1 clear)** 0: Progressive 1: Interlace
SE	5	R	0	**EVEN/ODD signal error (write 1 clear)** 0: Normal 1: Error
RS	4	R/W	0	**The reference signal for executing Info-frame automatically.** 0: DEN 1: VSYNC
DSC	3:0	R/W	0	**Down sample control (only valid if Video Down Sampling Auto Mode Disable)** 0000: pixel down sample for 1 time(no down sample) 0001: pixel down sample for 2 times 0010: pixel down sample for 3 times 0011: pixel down sample for 4 times 0100: pixel down sample for 5 times 0101: pixel down sample for 6 times

				0110: pixel down sample for 7 times 0111: pixel down sample for 8 times 1000: pixel down sample for 9 times 1001: pixel down sample for 10 times others : XXX

Register:: HDMI_ACRCR				**0x CA-51**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
HDIRQ	7	R/W	0	**HDMI/DVI change interrupt enable** **0:disable** **1:enable**
CSAM	6	R/W	1	**Color Space Translation** 0: Manual 1: Auto
CSC	5:4	R/W	0	**Color Space Control (if CSAM=1, CSC will be read-only)** 00: RGB 01: YCrCb-422 10: YCrCb-444 11: Reserved
Reserved	3	--	0	**Reserved to 0**
PRDSAM	2	R/W	1	**Pixel Repetition down sampling auto mode** 1: auto, the circuit resolve the repeat number, and enable it in next frame. The result could be read in bits for repeat number. 0: manual, F/W sets repeat number, the number is set in bits for repeat number.
PUCNR	1	R/W	0	**Pop up CTS&N result** 0: No pop up 1: Pop up result (Pop up CTS&N which is acquired completely. If present N&CTS is acquiring, pop up previous complete N&CTS) If the info is updating, HW will refuse this command.
PUCSR	0	R/W	0	**Pop up channel status result** 0: No pop up 1: Pop up result (Pop up channel status which is acquired completely. If present channel status is acquiring, pop up previous complete channel status)

Register:: HDMI_ACRSR0				**0x CA-52**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
CTS	7:0	R	0	**CTS in usage, CTS[19:12]**

Register:: HDMI_ACRSR1				**0x CA-53**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
CTS	7:0	R	0	**CTS in usage, CTS[11:4]**

Register:: HDMI_ACRSR2				**0x CA-54**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
CTS	7:4	R	0	**CTS in usage, CTS[3:0]**
N	3:0	R	0	**N in usage, N[19:16]**

Register:: HDMI_ACRSR3				**0x CA-55**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
N	7:0	R	0	**N in usage, N[15:8]**

Register:: HDMI_ACRSR4				**0x CA-56**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
N	7:0	R	0	**N in usage, N[7:0]**

Register:: HDMI_ACS0				0x CA-57
Name	**Bits**	**R/W**	**Reset State**	**Comments**
CS	7:0	R	0	**Channel status bit7~ bit0**

Register:: HDMI_ACS1				0x CA-58
Name	**Bits**	**R/W**	**Reset State**	**Comments**
CS	7:0	R	0	**Channel status bit 15~ bit 8**

Register:: HDMI_ACS2				0x CA-59
Name	**Bits**	**R/W**	**Reset State**	**Comments**
CS	7:0	R	0	**Channel status bit23~ bit 16**

Register:: HDMI_ACS3				0x CA-5A
Name	**Bits**	**R/W**	**Reset State**	**Comments**
CS	7:0	R	0	**Channel status bit 31~ bit 24**

Register:: HDMI_ACS4				0x CA-5B
Name	**Bits**	**R/W**	**Reset State**	**Comments**
CS	7:0	R	0	**Channel status bit 39~ bit 32**

Register:: HDMI_INTCR				0x CA-60
Name	**Bits**	**R/W**	**Reset State**	**Comments**
PENDING	7	R	0	**When IRQ occurred, this bit would be assigned to 1 by HW, and IRQ would be pended until FW clear this bit. (write 1 clear) (to MCU)**
AVMUTE	6	R/W	0	**If get General control packet and the corresponding Set_AVMUTE flag & Clear_AVMUTE flag is different with previous values** 0: IRQ don't occur. 1: IRQ occur.
FIFOD	5	R/W	0	**If FIFO depth reach Target** (Used for manual audio flow) 0: IRQ don't occur 1: IRQ occur
ACT	4	R/W	0	**Audio Coding Type** 0: If audio coding type is different with previous value, IRQ doesn't occur. 1: If audio coding type is different with previous value, IRQ occurs.
APLL	3	R/W	0	**Audio PLL** 0: If audio PLL is non-lock, IRQ doesn't occur 1: If audio PLL is non-lock, IRQ occurs
AFIFOO	2	R/W	0	**Audio FIFO Overflow** 0: If audio FIFO is overflow for X samples , IRQ doesn't occur. 1: If audio FIFO is overflow for X samples , IRQ occurs.
AFIFOU	1	R/W	0	**Audio FIFO Underflow** 0: If audio FIFO is underflow for Y samples , IRQ doesn't occur. 1: If audio FIFO is underflow for Y samples , IRQ occurs.
VC	0	R/W	0	0: If video clock is broken, IRQ doesn't occur. 1: If video clock is broken, IRQ occur.

Register:: HDMI_ALCR				0x CA-61
Name	**Bits**	**R/W**	**Reset State**	**Comments**
LO1	7:6	R/W	0	**Speaker location of I2S #1 & SPDIF OUT#1** 00: from SubPacket0 of Audio Sample Packet 01: from SubPacket1 of Audio Sample Packet 10: from SubPacket2 of Audio Sample Packet 11: from SubPacket3 of Audio Sample Packet

LO2	5:4	R/W	1	**Speaker location of I2S #2 & SPDIF OUT #2**
LO3	3:2	R/W	2	**Speaker location of I2S #3 & SPDIF OUT #3**
LO4	1:0	R/W	3	**Speaker location of I2S #4 & SPDIF OUT #4**

Register:: HDMI_AOCR			**0x CA-62**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
SPDIFO1	7	R/W	0	**SPDIF 1 Output Switch** 0: cutoff 1: normal
SPDIFO2	6	R/W	0	**SPDIF 2 Output Switch**
SPDIFO3	5	R/W	0	**SPDIF 3 Output Switch**
SPDIFO4	4	R/W	0	**SPDIF 4 Output Switch**
I2SO1	3	R/W	0	**I2S 1 Output Switch** 0: cutoff 1: normal
I2SO2	2	R/W	0	**I2S 2 Output Switch**
I2SO3	1	R/W	0	**I2S 3 Output Switch**
I2SO4	0	R/W	0	**I2S 4 Output Switch**

Register:: HDMI_BCSR			**0x CA-70**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Reserved	7:6	---	0	**Reserved to 0**
NVLGB	5	R	0	**Video No Leading Guard Band** If no leading GB after video preamble (It is only triggered in HDMI mode), this bit would be assigned to 1 until clear this bit Write 1 to clear.
NALGB	4	R	0	**Audio No Leading Guard Band** If no leading GB after audio preamble (It is only triggered in HDMI mode), this bit would be assigned to 1 until clear this bit Write 1 to clear.
NATGB	3	R	0	**Audio No Trailing Guard Band** If audio packets without trailing GB, this bit would be assigned to 1 until clear this bit. Write 1 to clear.
NGB	2	R	0	**No Guard Band** If any type of GB is not synchronous in 3 channels(audio is only 2 channel), this bit would be assigned to 1 until clear this bit. Write 1 to clear.
PE	1	R	0	**Packet Error** If size of Data Island Packet is not times of 32, this bit would be assigned to 1 until clear this bit. Write 1 to clear.
GCP	0	R	0	**General Control Packet error flag**: If HW receive General Control Packet with Clear_AVMUTE=1 & Set_AVMUTE=1 ,assign this bit to 1 until clear this bit Write 1 to clear.

Register:: HDMI_ASR0			**0x CA-71**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Reserved	7:3	---	0	**Reserved to 0**
FsRE	2	R	0	**Fs Regeneration Error** If CTS & N received 0, this bit would be assigned to 1 until clear this bit Write 1 to clear.
FsIF	1	R	0	**Fs from InfoFrame** If audio frequency from InfoFrame ready, this bit would be assigned to 1 until clear this bit Write 1 to clear.
FsCS	0	R	0	**Fs from Channel Status**

				If audio frequency from Channel Status ready, this bit would be assigned to 1 until clear this bit Write 1 to clear.

Register:: HDMI_ASR1			**0x CA-72**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Reserved	7	---	0	**Reserved**
FBIF	6:4	R	0	**Frequency bits from info frame** 000: refer to channel status bits 001: 32k 010: 44.1k 011: 48k 100: 88.2k 101: 96k 110: 176.4k 111: 192k
FBCS	3:0	R	0	**Frequency bits from channel status**. (pop up with channel status simultaneously) 0100: 22.05k 0000: 44.1k 1000: 88.2k 1100: 176.4k 0110: 24k 0010: 48k 1010: 96k 1110: 192k 0011: 32k 0001: Sampling frequency not indicated

Register:: TMDS_DPC_SET0			**0x CA-80**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
dpc_en	7	R/W	0	Enable deep color mode
phase_errcnt_in	6:4	R/W	0	Max. times of dpc pll phase error to rise error flag 3'b000 → count 8 times 3'b001~3'b111→ count 1~7 times
phase_clrcnt_in	3:1	R/W	0	Max. times of sync. Signal to clear the phase error counter according to "phase_clr_sel" 3'b000 → count 8 times 3'b001~3'b111→ count 1~7 times
phase_clr_sel	0	R/W	0	Unit of "phase_clrcnt_in" 0: Use V sync 1: Use H sync

Register:: TMDS_DPC_SET1			**0x CA-81**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
set_full_noti	7:4	R/W	0	Set full notify level (recommend: 3'd7) 3'b000~3'b111→ set 0~7
set_empty_noti	3:0	R/W	0	Set empty notify level (recommend: 3'd3) 3'b000~3'b111→ set 0~7

Register:: TMDS_DPC_SET2			**0x CA-82**	
Name	**Bits**	**R/W**	**Reset State**	**Comments**
fifo_errcnt_in	7:5	R/W	0	Max. times of FIFO error to rise error flag

				3'b000 → count 8 times 3'b001~3'b111→ count 1~7 times
clr_phase_flag	4	R/W	0	Clear phase error flag
clr_fifo_flag	3	R/W	0	Clear FIFO error flag
dpc_phase_ok	2	R	0	Phase locking OK
dpc_phase_err_flag	1	R	0	Become 1 when phase error than "phase_errcnt_in" number
dpc_fifo_err_flag	0	R	0	Become 1 when fifo error than "fifo_errcnt_in" number

Register:: TMDS_DPC_SET3				**0x CA-83**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
dpc_fifo_over_flag	7	R	0	Become 1 when internal FIFO receive writing signal while it is full.
dpc_fifo_under_flag	6	R	0	Become 1 when internal FIFO receive reading signal while it is empty.
dpc_fifo_over_xflag	5	R	0	Become 1 when internal FIFO receive writing signal while it is full. If (fifo_under_xflag=1), this flag is not active.
dpc_fifo_under_xflag	4	R	0	Become 1 when internal FIFO receive reading signal while it is empty. If (fifo_over_xflag=1), this flag is not active.
Reserved	3:0	--	0	reserved

TMDS Decoding Error Detection

Register:: TMDS_DET_0				**0xCA-84**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Reserved	7	---		Reserved
DE_SEL	6	R/W	0	**Source of DE Selection** 0: Choose DE from Post Stage (After LPF) 1: Choose DE from Pre-Stage
POS_DE_LOWBD	5:0	R/W	36	**DE =1 Low Bound Setting** Setting Positive DE Period Low Bound Value 0~63

Register:: TMDS_DET_1				**0xCA-85**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
NEG_DE_LOWBD	7:1	R/W	12	**DE =0 Low Bound Setting** Setting Negative DE Period Low Bound Value 0~127
RED_TRAN_ERR_FLAG	0	R	0	**RED Channel Transition Error Flag (write 1 clear)** 0: Means No Transition Error Occurs at Red Channel 1: Means Transition Error occurs

Register:: TMDS_DET_2				**0xCA-86**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
GRN_TRAN_ERR_FLAG	7	R	0	**Green Channel Transition Error Flag (write 1 clear)** 0: Means No Transition Error Occurs at Green Channel 1: Means Transition Error occurs
BLU_TRAN_ERR_FLAG	6	R	0	**Blue Channel Transition Error Flag (write 1 clear)** 0: Means No Transition Error Occurs at Blue Channel 1: Means Transition Error occurs
RED_POS_DE_ERR_FLAG	5	R	0	**Red Channel Positive DE Period Measurement Flag (write 1 clear)** 0: Means DE Period is Valid under Red Channel 1: Means DE Period is Invalid under Red Channel
GRN_POS_DE_ERR_FLAG	4	R	0	**Green Channel Positive DE Period Measurement Flag (write 1 clear)** 0: Means DE Period is Valid under Green Channel

				1: Means DE Period is Invalid under Green Channel
BLU_POS_DE_ERR_FLAG	3	R	0	**Blue Channel Positive DE Period Measurement Flag (write 1 clear)** 0: Means DE Period is Valid under Blue Channel 1: Means DE Period is Invalid under Blue Channel
RED_NEG_DE_ERR_FLAG	2	R	0	**Red Channel Negative DE Period Measurement Flag (write 1 clear)** 0: Means DE Period is Valid under Red Channel 1: Means DE Period is Invalid under Red Channel
GRN_NEG_DE_ERR_FLAG	1	R	0	**Green Channel Negative DE Period Measurement Flag (write 1 clear)** 0: Means DE Period is Valid under Green Channel 1: Means DE Period is Invalid under Green Channel
BLU_NEG_DE_ERR_FLAG	0	R	0	**Blue Channel Negative DE Period Measurement Flag (write 1 clear)** 0: Means DE Period is Valid under Blue Channel 1: Means DE Period is Invalid under Blue Channel

Register:: TMDS_DET_3				**0xCA-87**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
TRAN_ERR_THRD	7:4	R/W	0	**Transition Error Count Threshold Value Setting** ~~0~15~~ 0-14 Zero value means at least one error
POS_DE_ERR_THRD	3:0	R/W	0	**Positive DE Error Count Threshold Value Setting** ~~0~15~~ 0-14 Zero value means at least one error

Register:: TMDS_DET_4				**0xCA-88**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
Rev	7:4	---		**Reserved**
NEG_DE_ERR_THRD	3:0	R/W	0	**Negative DE Error Count Threshold Value Setting** ~~0~15~~ 0-14 Zero value means at least one error

Register:: AUDIO_FREQDET				**0x CA-90**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
AUDIO_FREQ_DETECT	7	R/W	0	Enable the detection of the audio sampling rate.
AUTO_LOAD_SCODE	6	R/W	0	Auto load s-code.
AWD_BY_FREQCHANGE	5	R/W	0	Audio watch dog triggered by audio sampling rate changed.
AWD_BY_NOAUDIO	4	R/W	0	Audio watch dog triggered by no_audio_in.
IRQ_BY_ FREQCHANGE	3	R/W	0	IRQ triggered by audio sampling rate changed.
IRQ_BY_NOAUDIO	2	R/W	0	IRQ triggered by no_audio_in.
FREQCHANGE	1	R	0	Audio sampling rate changed, including change to no audio(Write 1 to clear)
NOAUDIO	0	R	0	No any audio packet transmitted. Need 2ms for measurement to be asserted (Write 1 to clear)

By measuring the number of received audio packets within 1ms (xtal clock divided by "32*XTAL_DIV") to determine the audio sampling rate. The measured result should be the same for 2 times (2ms) for sampling rate determination.

Register:: AUDIO_FREQDET_RESULT_M				**0x CA-91**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
AFREQ_MEAS_RANGE	7:5	R	0	Automatic detection of audio sampling rate. 3'b000: Range 0 (Default for no_audio) 3'b001: Range 1 (Default for 32kHz) 3'b010: Range 2 (Default for 44.1kHz & 48kHz) 3'b011: Range 3 (Default for 88.2kHz & 96kHz) 3'b100: Range 4 (Default for 176.4kHz & 192kHz)

				3'b101: Range 5 (For future extention) Other: Not used.
POPUP_AFREQ_MEAS_RESULT	4	R/W	0	Pop-up the current number of audio packets in 1ms
AFREQ_MEAS_RESULT[11:8]	3:0	R	0	The number of audio packets in 1ms

Register:: AUDIO_FREQDET_RESULT_L				**0x CA-92**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
AFREQ_MEAS_RESULT[7:0]	7:0	R	0	The number of audio packets in 1ms

Register:: XTAL_DIV				**0x CA-93**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
XTAL_DIV	7:0	R/W	0x98	Select the most proper divider that the output clock will approximate 5kHz: Eq: Fx/32/Xtal_div ~= 5kHz For example: 14.318MHz: Xtal_div = 89 24.000MHz: Xtal_div = 150 24.300MHz: Xtal_div = 152 24.576MHz: Xtal_div = 154 27.000MHz: Xtal_div = 169

Register:: RANGE0_M				**0x CA-94**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
RANGE0_TH[11:8]	7:4	R/W	0	The number of audio packets in 1ms less than RANGE0_TH or larger than RANGE5_TH will be regarded as Range0
RANGE1_TH[11:8]	3:0	R/W	0	The number of audio packets in 1ms larger than RANGE0_TH and less than RANGE1_TH will be regarded as Range1

Register:: RANGE0_L				**0x CA-95**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
RANGE0_TH[7:0]	7:0	R/W	0x10	The number of audio packets in 1ms less than RANGE0_TH or larger than RANGE5_TH will be regarded as Range0

Register:: RANGE1_L				**0x CA-96**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
RANGE1_TH[7:0]	7:0	R/W	0x26	The number of audio packets in 1ms larger than RANGE0_TH and less than RANGE1_TH will be regarded as Range1

Register:: RANGE2_M				**0x CA-97**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
RANGE2_TH[11:8]	7:4	R/W	0	The number of audio packets in 1ms larger than RANGE1_TH and less than RANGE2_TH will be regarded as Range2
RANGE3_TH[11:8]	3:0	R/W	0	The number of audio packets in 1ms larger than RANGE2_TH and less than RANGE3_TH will be regarded as Range3

Register:: RANGE2_L				**0x CA-98**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
RANGE2_TH[7:0]	7:0	R/W	0x44	The number of audio packets in 1ms larger than RANGE1_TH and less than RANGE2_TH will be regarded as Range2

Register:: RANGE3_L				0x CA-99
Name	**Bits**	**R/W**	**Reset State**	**Comments**
RANGE3_TH[7:0]	7:0	R/W	0x88	The number of audio packets in 1ms larger than RANGE2_TH and less than RANGE3_TH will be regarded as Range3

Register:: RANGE4_M				0x CA-9A
Name	**Bits**	**R/W**	**Reset State**	**Comments**
RANGE4_TH[11:8]	7:4	R/W	0	The number of audio packets in 1ms larger than RANGE3_TH and less than RANGE4_TH will be regarded as Range4
RANGE5_TH[11:8]	3:0	R/W	0x1	The number of audio packets in 1ms larger than RANGE4_TH and less than RANGE5_TH will be regarded as Range5

Register:: RANGE4_L				0x CA-9B
Name	**Bits**	**R/W**	**Reset State**	**Comments**
RANGE4_TH[7:0]	7:0	R/W	0xF0	The number of audio packets in 1ms larger than RANGE3_TH and less than RANGE4_TH will be regarded as Range4

Register:: RANGE5_L				0x CA-9C
Name	**Bits**	**R/W**	**Reset State**	**Comments**
RANGE5_TH[7:0]	7:0	R/W	0xA4	The number of audio packets in 1ms larger than RANGE4_TH and less than RANGE5_TH will be regarded as Range5

Register:: PRESET_S_CODE0				0x CA-9D
Name	**Bits**	**R/W**	**Reset State**	**Comments**
PRE_SET_S1_CODE0	7	R/W	0	Once the determination of audio sampling rate is Range 0, the S1 code will be automatically replaced by this value.
PRE_SET_S_CODE0	6:0	R/W	0x06	Once the determination of audio sampling rate is Range 0, the S code will be automatically replaced by this value.

Register:: PRESET_S_CODE1				0x CA-9E
Name	**Bits**	**R/W**	**Reset State**	**Comments**
PRE_SET_S1_CODE1	7	R/W	0	Once the determination of audio sampling rate is Range 1, the S1 code will be automatically replaced by this value.
PRE_SET_S_CODE1	6:0	R/W	0x04	Once the determination of audio sampling rate is Range 1, the S code will be automatically replaced by this value.

Register:: PRESET_S_CODE2				0x CA-9F
Name	**Bits**	**R/W**	**Reset State**	**Comments**
PRE_SET_S1_CODE2	7	R/W	0	Once the determination of audio sampling rate is Range 2, the S1 code will be automatically replaced by this value.
PRE_SET_S_CODE2	6:0	R/W	0x02	Once the determination of audio sampling rate is Range 2, the S code will be automatically replaced by this value.

Register:: PRESET_S_CODE3				0x CA-A0
Name	**Bits**	**R/W**	**Reset State**	**Comments**
PRE_SET_S1_CODE3	7	R/W	0	Once the determination of audio sampling rate is Range 3, the S1 code will be automatically replaced by this value.
PRE_SET_S_CODE3	6:0	R/W	0x01	Once the determination of audio sampling rate is Range 3, the S code will be automatically replaced by this value.

Register:: PRESET_S_CODE4	0x CA-A1

Name	Bits	R/W	Reset State	Comments
PRE_SET_S1_CODE4	7	R/W	0	Once the determination of audio sampling rate is Range 4, the S1 code will be automatically replaced by this value.
PRE_SET_S_CODE4	6:0	R/W	0x00	Once the determination of audio sampling rate is Range 4, the S code will be automatically replaced by this value.

Register:: PRESET_S2_CODE				**0x CA-A2**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
PRE_SET_SM_CODE0	7	R/W	1	Once the determination of audio sampling rate is Range 0, the S1_MSB code will be automatically replaced by this value.
PRE_SET_SM_CODE1	6	R/W	1	Once the determination of audio sampling rate is Range 1, the S1_MSB code will be automatically replaced by this value.
PRE_SET_SM_CODE2	5	R/W	1	Once the determination of audio sampling rate is Range 2, the S1_MSB code will be automatically replaced by this value.
PRE_SET_SM_CODE3	4	R/W	1	Once the determination of audio sampling rate is Range 3, the S1_MSB code will be automatically replaced by this value.
PRE_SET_SM_CODE4	3	R/W	1	Once the determination of audio sampling rate is Range 4, the S1_MSB code will be automatically replaced by this value.
AFREQ_RESERVED0	2:0	R/W	0	Reserved.

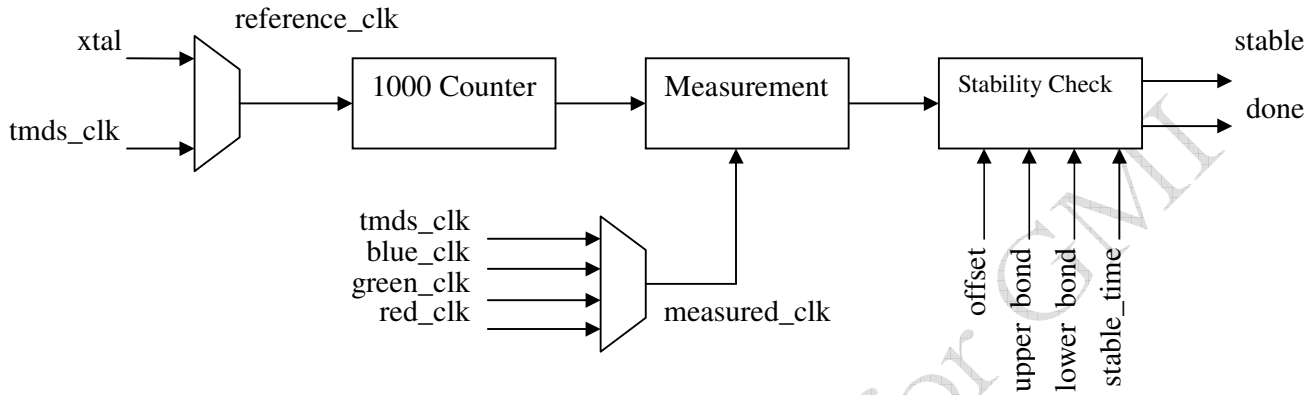
Register:: AFSM_MOD				**0x CA-A3**
Name	**Bits**	**R/W**	**Reset State**	**Comments**
DP_ABUF_WR_MOD_EN	7	R/W	1	Before HW confirms there has been any audio-input, HW stops the dp-abuf write process.
AUTO_STOP_TRK_EN	6	R/W	1	HW automatically pause the FIFO tracking process right after no audio has been detected.
TRK_MOD _EN	5	R/W	1	FIFO tracking (trend & boundary) is activated only when Audio FSM is within normal state.
AFSM_MOD_EN	4	R/W	1	As long as no audio has been detected, HW automatically turn down the audio volume gradually.
AFREQ_RESERVED1	3:0	R/W	0	Reserved.

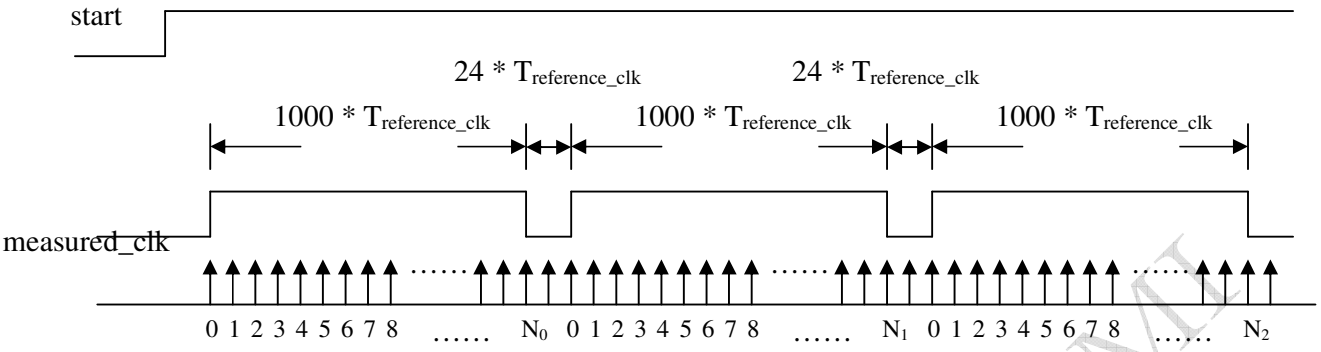
Power Saving Mode (Page 2)

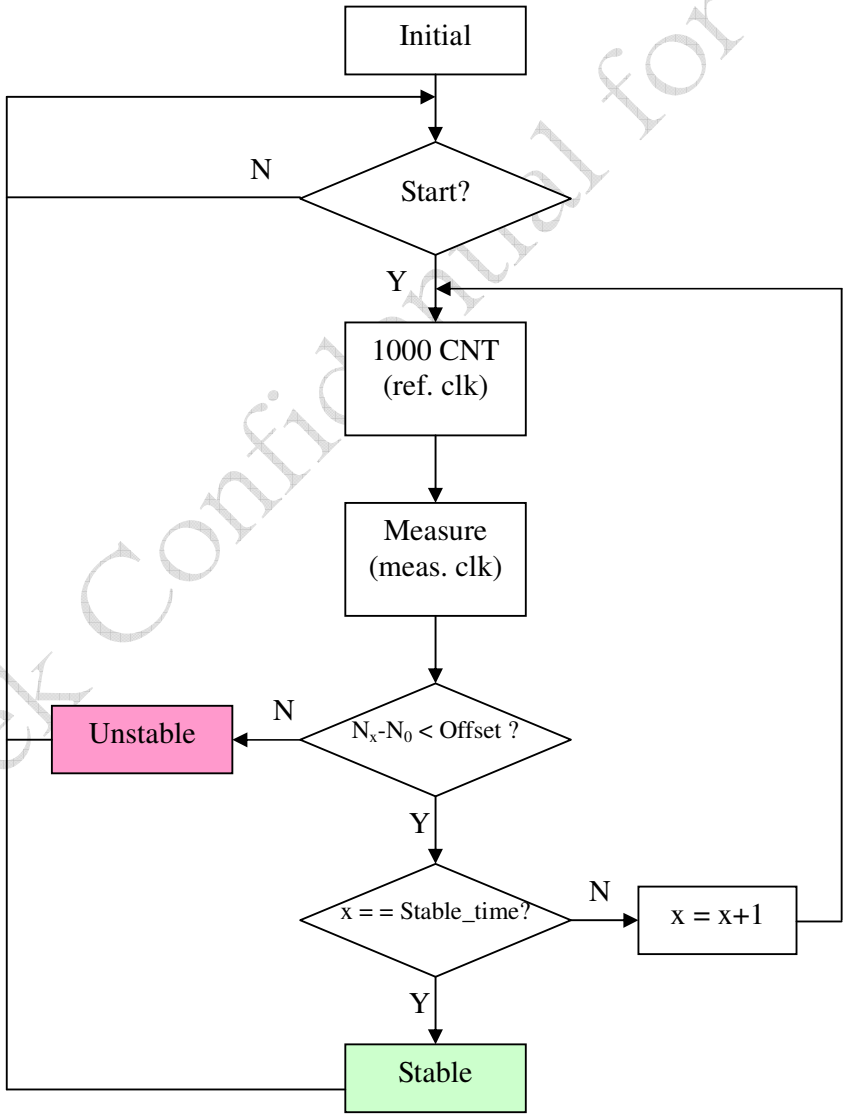
Register:: PS_REPLY			0xE1		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:6	R/W	0	Reserved	
Xtal_sel4gdi	5	R/W	1	1: from m2pll 0 : from xtal	
Reserved	4:0	R/W	0	Reserved	

Power-Saving Mode Domain
Input Video clock frequency detection

Frequency detection formula: $1000 \times T_{ref\_clk} = counter \times T_{measure\_clk}$







Register:: HDMI_FREQDET_CTRL				0xE6
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
FREQDET_START	7	R/W	0	Start the frequency detection process. Once the bit has been set, this process won't stop util disabling this bit. 0: Diabled 1: Started	
FREQDET_DONE	6	R	0	The first frequency detection process has done. 0: Not done yet. 1: Done.	
FREQ_SOURCE	5:4	R/W	0	Choose which clock to be measured. 00: TMDS clock. 01: Blue channel clock. 10: Green channel clock. 11: Red channel clock.	
FREQ_REFERENCE	3	R/W	0	Choose which clock as reference clock. 0: Xtal clock. (14.318~27MHz are preferred) 1: TMDS clock.	
FREQ_UNSTABLE	2	R	0	The freq status has been unstable. (Write 1 to clear) 0: Disabled 1: Enabled	
IRQ_FREQ_UNSTABLE	1	R/W	0	Interrupt as long as the freq is unstable. 0: Disabled 1: Enabled	
VWD_FREQ_UNSTABLE	0	R/W	0	Trigger video watch-dog as long as the freq is unstable. 0: Disabled 1: Enabled	WportRport

Register:: HDMI_FREQDET_OFFSET				0xE7
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
MEAS_CLK_SEL	7	R/W	0	The selection of measure clk 0: determined by 0xE6[5:4] 1: determined by hdmi_cp_clk	
FREQDET_OFFSET	6:0	R/W	2	Determine the pre-set threshold, within which means the frequency is still stable. 0: No variation is allowed. 1: ± Fx / 1024 2: ± 2* Fx / 1024 3: ± 3* Fx / 1024 … 127: ± 127* Fx / 1024, where Fx is frequency of xtal clock.	

Register:: HDMI_FREQDET_UPPER_M				0xE8
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
UPPER_BOUND[15:8]	7:0	R/W	2D	Once the frequency counter exceeds this value, then it will be treated as unstable.	

Register:: HDMI_FREQDET_UPPER_L				0xE9
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
UPPER_BOUND[7:0]	7:0	R/W	ED	Once the frequency counter exceeds this value, then it will be treated as unstable.	

Register:: HDMI_FREQDET_LOWER_M				0xEA
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
LOWER_BOUND[15:8]	7:0	R/W	2	Once the frequency counter is smaller than this value, then it will be treated as unstable.	

Register:: HDMI_FREQDET_LOWER_L				0xEB
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
LOWER_BOUND[7:0]	7:0	R/W	F6	Once the frequency counter is smaller than this value, then it will be treated as unstable.	

Register:: HDMI_FREQDET_STABLE				0xEC	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
RESULT_POPUP	7	W	0	POP UP the counter value in HW The result will be shown in 0xED[7], 0xED[6:0], 0xEE[7:0]	**Wclr_out**
STABLE_TIME	6:0	R/W	A	Determine how much times that freq needs to be within the pre-set threshold then the freq can be regarded as stable. 0: 1 time. 1: 2 time. 2: 3 times. … 127: 128 times.	

Register:: HDMI_FREQDET_RESULT_M				0xED	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
NO_CLOCK	7	R	0	No input clock	
FREQDET_RESULT [14:8]	6:0	R		The result of frequency counter (bit 15 is in 0xEF[5])	

Register:: HDMI_FREQDET_RESULT_L				0xEE	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
FREQDET_RESULT [7:0]	7:0	R		The result of frequency counter	

Register:: HDMI_ERROR_TH				0xEF	
Name	**Bit**	**R/W**	**Default**	**Description**	**Config**
FREQ_STABLE_IRQ_EN	7	R/W	0	Interrupt as long as the freq is stable. 0: Disabled 1: Enabled	
FREQ_STABLE	6	R	0	Determine if clock frequency is stable in measure process. 0: Not stable yet 1: Stable at least once. (write 1 clear)	**write 1 clear**
FREQDET_RESULT[15]	5	R/W	0	The result of frequency counter	
FREQ_ERROR_TH	4:0	R/W	0	Error Count Upper Limit When accumulated error below this value, frequency unstable will not occur. 0: means 1 error 1: means 2 errors ……. 31: means 32 errors.	

Internal OSC (Page 6)

Register::OSC_TRIM_CTRL0				0xA0	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
TRIM_EN	7	R/W	0x0	Trimming Enable 0: Disable 1: Enable	
TRIM_DONE	6	R	0x1	Trimming is done 0: In progress 1: Done	
TRIM_RESOLUTION	5	R/W	0x0	Trimming Counter Set (counted by Tref) 0: 1000 1: 2000(higher resolution)	
TRIM_WCNT_1ms	4	R/W	0x1	Wait time for “Initial OSC stable” 0: 1* 2^10 *Tref 1: 15* 2^10 *Tref (1ms at Tref=70ns)	
TRIM_WCNT	3:2	R/W	0x1	Wait time for trimming” 00: 1* 2^10 *Tref 01: 2* 2^10 *Tref 10: 3 * 2^10 *Tref 11: 4 * 2^10 *Tref	
IOSC_DIV	1:0	R/W	0	IOSC Divider 00: Div 1 (default) 01: Div 2 10: Div 4 11: Div 8	

Register::OSC_TRIM_CTRL1				0xA1	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
TRIM_TARGET_L	7:0	R/W	0xE8	Trimming Comparison Target (default when Tref = 70ns)	

Register::OSC_TRIM_CTRL2				0xA2	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
TRIM_MANUAL_EN	7	R/W	0	Trimming Manual mode Enable 0: Trimming Auto Mode 1: Trimming Manual Mode	
TRIM_LT	6	R	0	0: Trimming Counter Result more than target 1: Trimming Counter Result less than target	
TRIM_VC_RESULT	5:3	R	0	Trimming VC Result	
TRIM_MOS_RESULT	2:0	R	0	Trimming MOS Result	

Register::EMBEDDED_OSC_CTRL				0xA3	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
OSC_EN	7	R/W	1	EMB OSC ENABLE, (need waiting 1ms after enable) 0: DISABLE 1: ENABLE	
OSC_AUTO	6	R/W	1	0: manual mode (by A2[5:0]) 1: auto switch to trimming result	
OSC_VR_VC	5:3	R/W	0	bit[5:3] OSC VC Tune LSB active when manual mode (bit6=0)	
OSC_VR_MOS	2:0	R/W	0x3	bit[1:0] OSC MOS Tune MSB-bit 2 is useless active when manual mode (bit6=0)	

Register:: OSC_TRIM_CNT				0xA4	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
TRIM_CNT_L	7:0	R	--	Trimming Counter(Low byte), to compare with the target clock	

Register:: EMB_BGRES				0xA5	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
TRIM_CNT_H	7:4	R	--	Trimming Counter(High byte), to compare with the target clock	
TRIM_TARGET_H	3:0	R/W	0x3	Trimming Comparison Target (default when Tref = 70ns)	

Register:: EMB_BGRES				0xA6	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reversed	7:2	--	--	Reversed	
REG_EMB_BGRES[1:0]	1:0	R/W	0x0	Value of temperature factor (01 for 14.318M, 10 for 27M) 00: smallest 11: largest	

Address 0xA7 ~ 0xAF reserved

Audio DAC (Page 6)

Register:: BB_POWER0			0xB0		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BB_POW_AIN	7	R/W	0	Power down control for AIN buffer (0:power down, 1:power on)	
BB_POW_AINVOL	6	R/W	0	Power down control for AIN volume control (0:power down, 1:power on)	
BB_POW_AOUT	5	R/W	0	Power down control for AOUT amplifier (0:power down, 1:power on)	
BB_POW_DAC	4	R/W	0	Power down control for DAC (0:power down, 1:power on)	
BB_POW_DACVOL	3	R/W	0	Power down control for DAC volume control (0:power down, 1:power on)	
BB_POW_DACVREF	2	R/W	0	Power down control for DAC ref. voltage buffer (0:power down, 1:power on)	
BB_POW_DF2SE	1	R/W	0	Power down control for DF2SE (0:power down, 1:power on)	
BB_POW_HPOUT	0	R/W	0	Power down control for HPOUT amplifier (0:power down, 1:power on)	

Register:: BB_POWER1			0xB1		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Rev_b1_7_3	7:3	R/W	0	Reserved	
BB_EN_AIN	2	R/W	0	Enable AIN (0:disable, 1:enable)	
BB_POW_MBIAS	1	R/W	0	Power down control for bias generator (0:power down, 1:power on)	
BB_POW_VREF	0	R/W	0	Power down control for analog ground generator (0:power down, 1:power on)	

Register:: BB_ZCD_ANA			0xB2		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BB_MBIAS_ZCD	7:6	R/W	2	Bias current selection for zero crossing comparator (00b:10u, 01b:15u, 10b:20u, 11b:30u)	
BB_POW_ZCD	5	R/W	0	Power down ctrl for all zero crossing comparator (0:power down, 1:power on)	
BB_POW_ZCDCOMP	4:1	R/W	0	Power down ctrl for VOL zero crossing comparator (0:power down, 1:power on) Bit 4: AINVOL_L, Bit3: AINVOL_R, Bit2: DACVOL_L, Bit1: DACVOL_R	
EN_POCD	0	R/W	0	Not Used	

Register:: AOUT_CONTROL			0xB3		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BB_MUTE_AOUT_L	7	R/W	1	Mute control for AOUT_L (0:unmute, 1:mute) If C2 [4]=0, write this address could be read out immediately. Else, this value would be the final volume apply value	Rport wport

BB_MUTE_AOUT_R	6	R/W	1	Mute control for AOUT_R (0:unmute, 1:mute) If C2 [4]=0, write this address could be read out immediately. Else, this value would be the final volume apply value	Rport wport
BB_MUX_AOUT	5	R/W	0	Source selection for AOUT (0:from DAC, 1:from AIN)	
BB_OUTEN_AOUT	4	R/W	0	Output enable for AOUT (0: input mode, for start up de-pop) (1: output mode, for normal operation)	
Rev_b4_3_0	3:0	R/W	0	Reserved	

P.S Before set B3[5] & B3[3], make sure C2[5] is disable.

Register:: HPOUT_CONTROL 0xB4

Name	Bits	R/W	Default	Comments	Config
BB_MUTE_HPOUT_L	7	R/W	1	Mute control for HPOUT_L (0:unmute, 1:mute) If C2 [5]=0, write this address could be read out immediately. Else, this value would be the final volume apply value.	Rport wport
BB_MUTE_HPOUT_R	6	R/W	1	Mute control for HPOUT_R (0:unmute, 1:mute) If C2 [5]=0, write this address could be read out immediately. Else, this value would be the final volume apply value	Rport wport
BB_MUX_HPOUT	5	R/W	0	ZCD Reference Input (Should be aligned with CA) 0: From DAC 1: From AIN	
BB_OUTEN_HPOUT	4	R/W	0	Output enable for HPOUT (0: input mode, for start up de-pop) (1: output mode, for normal operation)	
Rev_b5_3_0	3:0	R/W	0	Reserved	

P.S Before set B4[5] & B4[3], make sure C2[4] is disable.

Register:: MBIAS_CONTROL0 0xB5

Name	Bits	R/W	Default	Comments	Config
BB_MBIAS_AMP	7:6	R/W	2	Bias current selection for output amplifier (00b:5u, 01b:10u, 10b:20u, 11b:30u)	
BB_MBIAS_DACVREF	5:4	R/W	2	Bias current selection for DACVREF (00b:10u, 01b:15u, 10b:20u, 11b:30u)	
BB_MBIAS_DAOP	3:2	R/W	2	Bias current selection for DAOP (00b:10u, 01b:15u, 10b:20u, 11b:30u)	
BB_MBIAS_DAREFBUF	1:0	R/W	2	Bias current selection for DAREFBUF (00b:10u, 01b:15u, 10b:20u, 11b:30u)	

Register:: MBIAS_CONTROL1 0xB6

Name	Bits	R/W	Default	Comments	Config
BB_MBIAS_DF2SE	7:6	R/W	2	Bias current selection for DF2SE (00b:10u, 01b:15u, 10b:20u, 11b:30u)	
BB_MBIAS_IN_MC3	5:4	R/W	2	Bias current selection for input buffer (00b:10u, 01b:15u, 10b:20u, 11b:30u)	
BB_MBIAS_VOL	3:2	R/W	2	Bias current selection for volume control (00b:10u, 01b:15u, 10b:20u, 11b:30u)	

BB_MBIAS_VREF	1:0	R/W	2	Bias current selection for analog ground generator (00b:5u, 01b:10u, 10b:20u, 11b:30u)	

Register:: VREF_CONTROL			**0xB7**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BB_VREF_VAG	7:6	R/W	1	Analog ground voltage selection (00b:1.717 ,01b:1.65, 10b:1.58, 11b:1.51)	
BB_DACVREF_MODE	5	R/W	1	DAC reference voltage source (0:internal generation, 1:external given)	
Rev_b8_4_0	4:0	R/W	0	Reserved	

Register:: DAC_VOL_L			**0xB8**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BB_DACVOL_L	7:0	R/W	0	Volume control for DAC output (00h:-58.5dB~37h:24dB) If C2 [7]=0, write this address could be read out immediately. Else, this value would be the final volume apply value. Note : double buffered by B9	Rport wport

PS. Before setting this byte, make sure corresponding "Done L" signal (C4[7]) is 1.

Register:: DAC_VOL_R			**0xB9**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BB_DACVOL_R	7:0	R/W	0	Volume control for DAC output (00h:-58.5dB~37h:24dB) If C2 [7]=0, write this address could be read out immediately. Else, this value would be the final volume apply value.	Rport wport

PS. Before setting this byte, make sure corresponding "Done L & R " signals (C4[7:6]) are both 1.
Note: B8 value is double buffered by B9

Register:: AIN_VOL_L			**0xBA**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BB_AINVOL_L	7:0	R/W	0	Volume control for AIN (00h:-58.5dB~37h:24dB) If C2[6]=0, write this address could be read out immediately. Else, this value would be the final volume apply value. Note : double buffered by BB	Rport wport

PS. Before setting this byte, make sure corresponding "Done L" signal (C4[5]) is 1.

Register:: AIN_VOL_R			**0xBB**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BB_AINVOL_R	7:0	R/W	0	Volume control for AIN (00h:-58.5dB~37h:24dB) If C2[6]=0, write this address could be read out immediately. Else, this value would be the final volume apply value.	Rport wport

PS. Before setting this byte, make sure corresponding "Done L & R " signals (C4[5:4]) are both 1.
Note: BA value is double buffered by BB

Register:: ANA_RESERVE1			0xBC		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ANA_REV1	7:2	R/W	0	reserved	
ANA_REV1	1	R/W	0	0: As depop power down, ramp switch OFF 1: As depop power down, ramp switch ON	
BB_IMP_SEL<0>	0	R/W	0	It is only used in tradition de-pop mode (input mode, then output mode.) (According the value of external de-coupling capacitor , set different input impedance) 0: input impedance is 0.5K ohms. 1: input impedance is 3.5K ohms.	

Register:: ANA_RESERVE2			0xBD		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ANA_REV2	7:0	R/W	0	reserved	

Register:: MODULATOR_CONTROL			0xBE		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BB_MOD_CLK_RATE	7:6	R/W	00	00:mclk(256fs) 01:aclk(128fs) 10:sclk(64fs)	
BB_MOD_RST_N	5	R/W	1	for second time to reset sigma-delta modulator(after reset up sample filter about 22*(1/fs)) 0: Reset 1: No Reset	
BB_DEBUG_EN	4	R/W	0	Debug Mode Enable 1:Enable 0:Disable	
BB_DEBUG_MODE	3:1	R/W	0	Support 8 sets debug mode.	
BB_OUT_L_R_SEL	0	R/W	0	Debug Mode, Adding L or R output 16bits in digital function 0: L 1: R	

Register:: BIST_CONTROL			0xBF		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BB_BIST_MODE	7	R/W	0	1: Enable 0: Disable	
BB_BIST _RST_N	6	R/W	1	0: Reset 1: No Reset	
BB_BIST_DONE	5	R	0	1: BIST is done 0: BIST running	
BB_BIST_FAIL	4	R	0	1: SRAM fail 0: SRAM ok	
BB_FT_EN	3	R/W	0	For FT and test performance。 1:Iinput PCM data from test pin in. 0:PCM data from digital circuit.	
BB_48PIN_MODE	2	R/W	0	1:16bits TEST IN (PCM DATA). 0:22bits TEST IN. (PCM DATA)	

SDM_OUT_TEST_EN	1	R/W	0	SDM test enable. When set to 1, the DAC input will be forced SDM_OUT_TEST_VALUE	
HPF_EN	0	R/W	1	High pass filter enable 1 : Enable 0 : Disable	

Register:: SDM_OUT_VALUE_MSB 0xC0					
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
SDM_OUT_TEST_VALUE[15:8]	7:0	R/W	0xAA	Active when BF[1] = 1 , for verify	

Register:: SDM_OUT_VALUE_LSB 0xC1					
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
SDM_OUT_TEST_VALUE[7:0]	7:0	R/W	0xAA	Active when BF[1] = 1 , for verify	

Register:: ZERO_CROSSING_CTRL 0xC2					
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ZCD_DAC_VOL_EN	7	R/W	1	DAC 1:Volume control by zero crossing enable 0: disable	
ZCD_AIN_VOL_EN	6	R/W	1	AIN 1:Volume control by zero crossing enable 0: disable	
ZCD_HPOUT _MUTE_EN	5	R/W	0	Head phone Including un-mute and mute 1:mute control by zero crossing enable 0:disable	
ZCD_AOUT_MUTE_EN	4	R/W	0	Speaker Including un-mute and mute 1:mute control by zero crossing enable 0:disable	
ZCD_DAC_VOL_ SEP	3	R/W	1	DAC volume 1: LR controlled separately 0: LR controlled volume at the same time (if L or R encounter zero crossing, apply volume/mute to another one)	
ZCD_AIN_VOL_ SEP	2	R/W	1	Ain volume 1: LR controlled separately 0: LR controlled volume at the same time (if L or R encounter zero crossing, apply volume/mute to another one)	
ZCD_HP_MUTE_ SEP	1	R/W	1	Head phone mute 1: LR controlled separately 0: LR controlled volume at the same time (if L or R encounter zero crossing, apply volume/mute to another one)	
ZCD_AOUT_MUTE_SEP	0	R/W	1	Speaker mute 1: LR controlled separately 0: LR controlled volume at the same time (if L or R encounter zero crossing, apply volume/mute to another one)	

Register:: ZCD_TIMEOUT 0xC3					
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**

ZCD_TEST_INPUT	7	R/W	0	Test input. When F7[4] is 1, the input square wave will be switched to this register bit.	
ZCD_TIMEOUT	6:0	R/W	3	Setting range 0~7E Clock_base is xtal * 512 = 35.84us Timeout target is clock_base* {(ZCD_TIMEOUT+1),3'b0}	

Register:: ZCD_STATUS			**0xC4**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ZCD_DAC_L_VOL_DONE	7	R	--	Active when ZCD_DAC_VOL_EN = 1 ~~Write F9 to clear~~	
ZCD_DAC_R_VOL_DONE	6	R	--	Active when ZCD_DAC_VOL_EN = 1 ~~Write FA to clear~~	
ZCD_AIN_L_VOL_DONE	5	R	--	Active whe ZCD_AIN_VOL_EN = 1 ~~Write FB to clear~~	
ZCD_AIN_R_VOL_DONE	4	R	--	Active whe ZCD_AIN_VOL_EN = 1 ~~Write FC to clear~~	
ZCD_HPOUT_L_MUTE_DONE	3	R	--	Active when ZCD_HPOUT_MUTE_EN = 1 ~~Write F3[7] to clear~~	
ZCD_HPOUT_R_MUTE_DONE	2	R	--	Active when ZCD_HPOUT_MUTE_EN = 1 ~~Write F3[6] to clear~~	
ZCD_AOUT_L_MUTE_DONE	1	R	--	Active whe ZCD_AOUT_MUTE_EN = 1 ~~Write F2[7] to clear~~	
ZCD_AOUT_R_MUTE_DONE	0	R	--	Active whe ZCD_AOUT_MUTE _EN = 1 ~~Write F2[6] to clear~~	

Register:: DIG_RESERVE1			**0xC5**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DIG_REV1	7:0	R/W	0	reserved	

Register:: DIG_RESERVE2			**0xC6**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DA_CLK_EN	7	R/W	1	da_clk enable	
DIG_REV2	6:0	R/W	FF	reserved	

Register:: DIG_RESERVE_RONLY			**0xC7**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DIG_REV3	7:0	R	0	reserved	

Register::DEPOP_CTRL0			**0xC8**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BB_POW_DEPOP	7	R/W	0	Power down control for De-pop (0:power down, 1:power on)	
BB_POW_DEPOP_CORE	6	R/W	0	Power down control for De-pop Core (0:power down, 1:power on)	
BB_POW_DEPOP_OP	5	R/W	0	Power down control for De-pop OP (0:power down, 1:power on)	

BB_POW_DEPOP_CKGEN	4	R/W	0	Power down control for De-pop CKGEN (0:power down, 1:power on)	
BB_AOUT_NORM	3	R/W	0	Transfer De-pop mode to AOUT normal operation mode. 0: disable 1: Normal operation mode.	
BB_INEN_AOUT	2	R/W	0	Enable input mode at AOUT 0 Disable 1: Enable input mode.	
BB_AOUT_COPY<1:0>	1:0	R/W	0	Select Channel of AOUT 00b: Normal Stereo<Default> 01b: Normal Stereo 10b: L copy to R 11b: R copy to L	

Register::DEPOP_CTRL1			**0xC9**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BB_EN_AMP	7	R/W	0	Enable HPOUT AMP 0: disable(10k loading) 1. enable(32 loading)	
BB_HP_NORM	6	R/W	0	Transfer De-pop mode to HPOUT normal operation mode. 0: disable 1: Normal operation mode.	
BB_INEN_HPOUT	5	R/W	0	Enable input mode at HPOUT 0 Disable 1: Enable input mode.	
BB_EN_HPOUT_COPY<1:0>	4:3	R/W	0	Select Channel of HPOUT **(NO USE)** 00b: Normal Stereo<Default> 01b: Normal Stereo 10b: L copy to R 11b: R copy to L	
Reserved	2:0	R/W	0	Reserved to 0	

Register::DEPOP_CTRL2			**0xCA**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BB_MUX_HPOUT_L<3:0>	7:4	R/W	0	Source selection for HPOUT_L 1110: From DAC_L 1101: From AIN_L 1011: From DAC_R 0111: From AIN_R	
BB_MUX_HPOUT_R<3:0>	3:0	R/W	0	Source selection for HPOUT_R 1110: From DAC_R 1101: From AIN_R 1011: From DAC_L 0111: From AIN_L	

Register::DEPOP_CTRL4			**0xCB**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BB_MBIAS_DEPOP[1:0]	7:6	R/W	2	Bias current selection for DEPOP (00b:10u, 01b:15u, 10b:20u, 11b:30u)	
BB_MBIAS_DEPOP_OP1[1:0]	5:4	R/W	2	Bias current selection for DEPOP OP1(00b:10u, 01b:15u, 10b:20u, 11b:30u)	
BB_MBIAS_DEPOP_OP2[1:0]	3:2	R/W	2	Bias current selection for DEPOP OP2 (00b:10u, 01b:15u, 10b:20u, 11b:30u)	
BB_VCM_READY	1	R/W	0	When VCM starts to charged to (AVDD/2), REG_BB_VCM_READY changes from L to	

				H.	
Reserved	0	R/W	0	Reserved to 0	

Register::DEPOP_CTRL5			0xCC		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BB_DEPOP_SEL<1:0>	7:6	R/W	1	De-pop type selection 00: DEPOP1-original 01: DEPOP2-cap(Default) 10: DEPOP3-cap 11: DEPOP4-cap - high impedance	
BB_DEPOP_CUR_SEL<2:0>	5:3	R/W	2	Select the current at charge pump of the ramp wave generator. 000: 0.3125uA 001: 0.625uA 010: 1.25uA (Default) 011: 2.5uA 100: 5uA 101: 10uA	
BB_DEPOP_CAP_SEL<1:0>	2:1	R/W	3	Select the capacitor at charge pump of the ramp wave generator. 00: 6pF+6pF 01: 12pF+24pF 10: 18pF+24pF 11: 24pF+24pF (Default)	
Reserved	0	R/W	0	Reserved to 0	

Register::DEPOP_CTRL6			0xCD		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BB_CK_DEPOP_CK_DIV	7:4	R/W	0x4	Clock to De-pop circuit (14.318M / 64) 0000: 14.318MHz/64 = 223kHz 0001: 14.318MHz/128 = 111kHz 0010: 14.318MHz/256 = 55.75kHz (transition time=100ms) 0011: 14.318MHz/512 = 27.875kHz 0100: 14.318MHz/1024 = 13.9kHz(Default) 0101: 14.318MHz/2048 = 6.97kHz 0110: 14.318MHz/4096 = 3.48kHz (transition time=400ms) 0111: 14.318MHz/8192 = 1.74kHz 1000: 14.318MHz/16384 = 0.87kHz 1001: 14.318MHz/32768 = 0.435kHz	
DEPOP_CK_EN	3	R/W	0	Depop clock enable 1: Enable, 0: Disable	
Reserved	2:0	R/W	0	Reserved to 0	

Register::LFSR_CTRL			0xCE		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LFSR_EN	7	R/W	0	24-bit LFSR (Random generator) in SDM input 1: enable 0: output -> all “1’b1”	
BIT2_LFSR_EN	6	R/W	0	1:SDM input[2] replaced by LFSR[23] 0:orginal input	

BIT1_LFSR_EN	5	R/W	0	1:SDM input[1] replaced by LFSR[15] 0:orginal input	
BIT0_LFSR_EN	4	R/W	0	1:SDM input[0] replaced by LFSR[7] 0:orginal input	
SMPL_LFSR_EN	3	R/W	0	24-bit LFSR in HPF input 1: enable 0: output -> all “1’b1”	
SMPL_BIT2_LFSR_EN	2	R/W	0	1: HPF input[2] replaced by LFSR[23] 0:orginal input	
SMPL_BIT1_LFSR_EN	1	R/W	0	1: HPF input[1] replaced by LFSR[15] 0:orginal input	
SMPL_BIT0_LFSR_EN	0	R/W	0	1: HPF input[0] replaced by LFSR[7] 0:orginal input	

Register:: AUDIO_RES			**0xCF**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
AUDIO_RES	7:0	R/W	0	ANALOG reserved	

Vivid color-DCC (Page 7)

Register:: DCC_CTRL_0			0xc7		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DCC_EN	7	R/W	0	**DCC_ENABLE** 0: Disable 1: Enable	
Y_FORMULA	6	R/W	0	**Y_FORMULA** 0: Y = (2R+5G+B)/8 1: Y = (5R+8G+3B)/16	
SC_EN	5	R/W	0	**SOFT_CLAMP** 0: Disable 1: Enable	
DCC_MODE	4	R/W	0	**DCC_MODE** 0: Auto Mode 1: Manual Mode	
SCG_EN	3	R/W	0	**SCENE_CHANGE** 0: Disable Scene-Change Function 1: Enable Scene-Change Function in Auto Mode	
BWL_EXP	2	R/W	0	**BWL_EXP** 0: Disable Black/White Level Expansion 1: Enable Black/White Level Expansion in Auto Mode	
PAGE_SEL	1:0	R/W	0	**DCC_PAGE_SEL** 00: Page 0 (for Histogram / Ymin-max / Soft-Clamping / Scene-Change) 01: Page 1 (for Y-Curve / WBL Expansion) 10: Page 2 (for Calculation Parameter) 11: Page 3 (for Testing and Debug)	

Register:: DCC_CTRL_1			0xc8		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
GAIN_EN	7	R/W	0	**DCC gain control enable** 0: Disable 1: Enable Note: DCC gain control enable must delay MOV_AVG_LEN frame after DCC enable.	
DCC_FLAG	6	R	0	1: time to write highlight window position & normalized factor, write to clear	
SAT_COMP_EN	5	R/W	0	**Saturation Compensation Enable** 0: Disable 1: Enable	
BLD_MODE	4	R/W	0	**Blending Factor Control Mode** 0: old mode 1: new mode (diff. regions have diff. blending factor)	
Reserved	3:0	--	0x00	Reserved to 0	

Register:: DCC Address Port			0xc9		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DCC_ADDR	7:0	R/W	0x00	DCC address	

Register:: DCC Data Port			0xca		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DCC_DATA	7:0	R/W	0x00	DCC data	

Register:: NOR_FACTOR_H (page0)			(ACCESS[C9,CA]) 0x00		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:6	--	--	Reserved	
NOR_FAC_H	5:0	R/W	0x00	Bit[21:16] of Normalized Factor; NF=(255/N)*(2^22)	

Register:: NOR_FACTOR_M (page0)			(ACCESS[C9,CA]) 0x01		

Name	Bits	R/W	Default	Comments	Config
NOR_FAC_M	7:0	R/W	0x00	Bit[15:8] of Normalized Factor; NF=(255/N)*(2^22)	

Register:: NOR_FACTOR_L (page0)			(ACCESS[C9,CA]) 0x02		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
NOR_FAC_L	7:0	R/W	0x00	Bit[7:0] of Normalized Factor; NF=(255/N)*(2^22)	

Register:: BBE_CTRL (page0)			(ACCESS[C9,CA]) 0x03		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BBE_EN	7	R/W	0	**BBE_ENA** 0: Disable Black-Background Exception 1: Enable Black-Background Exception	
Reserved	6:4	--	--	Reserved	
BBE_THD	3:0	R/W	0x4	**BBE_THD** 8-bit RGB Threshold for Black-Background Exception	

Register:: NFLT_CTRL (page0)			(ACCESS[C9,CA]) 0x04		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
HNFLT_EN	7	R/W	0	**HNFLT_ENA** 0: Disable Histogram Noise Filter 1: Enable Histogram Noise Filter	
HNFLT_THD	6:4	R/W	0	**HNFLT_THD** Threshold for Histogram Noise Filter	
YNFLT_EN	3	R/W	0	**YNFLT_ENA** 0: Disable Ymax / Ymin Noise Filter 1: Enable Ymax / Ymin Noise Filter	
YNFLT_THD	2:0	R/W	0	**YNFLT_THD** Threshold for Ymax/Ymin Noise Filter (= 4*YNFLT_THD)	

Register:: HIST_CTRL (page0)			(ACCESS[C9,CA]) 0x05		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
RH0_LIMITER	7	R/W	0	**RH0_LIMITER** 0: Disable RH0 Limiter 1: Enable RH0 Limiter	
RH1_LIMITER	6	R/W	0	**RH1_LIMITER** 0: Disable RH1 Limiter 1: Enable RH1 Limiter	
REAL_MA_LEN	5:3	R	--	**Real MOV_AVG_LEN** may be different with MOV_AVG_LEN, if SCG enable	
MOV_AVG_LEN	2:0	R/W	0	**MOV_AVG_LEN** 000: Histogram Moving Average Length = 1 001: Histogram Moving Average Length = 2 010: Histogram Moving Average Length = 4 011: Histogram Moving Average Length = 8 100: Histogram Moving Average Length = 16 101~111: reserved	

Register:: SOFT_CLAMP (page0)			(ACCESS[C9,CA]) 0x06		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
SOFT_CLAMP	7:0	R/W	0xB0	**Slope of Soft-Clamping (= SOFT_CLAMP / 256)**	

Register:: Y_MAX_LB (page0)			(ACCESS[C9,CA]) 0x07		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Y_MAX_LB	7:0	R/W	0xFF	**Lower Bound of Y_MAX (= 4*Y_MAX_LB)**	

Register:: Y_MIN_HB (page0)			(ACCESS[C9,CA]) 0x08		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Y_MIN_HB	7:0	R/W	0x00	**Higher Bound of Y_MIN (= 4*Y_MIN_HB)**	

Register:: SCG_PERIOD (page0)			(ACCESS[C9,CA]) 0x09		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
SCG_MODE	7	R/W	0	**Scene-Change Control Mode** 0: old mode (2553V) 1: new mode (2622)	
Reserved	6:5	--	--	Reserved	
SCG_PERIOD	4:0	R/W	0x10	**Scene-Change Mode Period = 1~32.** Note: SCG_PERIOD >= MOV_AVG_LEN, CRED-05[2:0](page0)	

Register:: SCG_LB (page0)			(ACCESS[C9,CA]) 0x0A		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
SCG_LB	7:0	R/W	0x00	**SCG_DIFF Lower Bound for Exiting Scene-Change Mode**	

Register:: SCG_HB (page0)			(ACCESS[C9,CA]) 0x0B		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
SCG_HB	7:0	R/W	0xFF	**SCG_DIFF Higher Bound for Exiting Scene-Change Mode**	

Register:: POPUP_CTRL (page0)			(ACCESS[C9,CA]) 0x0C		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	--	Reserved	
POPUP_BIT	0	R	--	Reg[0D]~Reg[16] are updated every frame. Once POPUP_BIT is read, the value of Reg[0D] ~ Reg[16] will not be updated until Reg[16] is read.	

Register:: SCG_DIFF (page0)			(ACCESS[C9,CA]) 0x0D		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
SCG_DIFF	7:0	R	--	**= (Histogram Difference between Current Frame and Average) / 8**	

Register:: Y_MAX_VAL (page0)			(ACCESS[C9,CA]) 0x0E		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Y_MAX_VAL	7:0	R	--	**= Max { Y_MAX_LB, (Y Maximum in Current Frame / 4) }**	

Register:: Y_MIN_VAL (page0)			(ACCESS[C9,CA]) 0x0F		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Y_MIN_VAL	7:0	R	--	**= Min { Y_MIN_HB, (Y Minimum in Current Frame / 4) }**	

Register:: S0_VALUE (page0)			(ACCESS[C9,CA]) 0x10		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
S0_VALUE	7:0	R	--	**Normalized Histogram S0 Value**	

Register:: S1_VALUE (page0)			(ACCESS[C9,CA]) 0x11		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
S1_VALUE	7:0	R	--	**Normalized Histogram S1 Value**	

Register:: S2_VALUE (page0)			(ACCESS[C9,CA]) 0x12		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
S2_VALUE	7:0	R	--	**Normalized Histogram S2 Value**	

Register:: S3_VALUE (page0)			(ACCESS[C9,CA]) 0x13		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
S3_VALUE	7:0	R	--	**Normalized Histogram S3 Value**	

Register:: S4_VALUE (page0)			(ACCESS[C9,CA]) 0x14		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**

S4_VALUE	7:0	R	--	**Normalized Histogram S4 Value**	

Register:: S5_VALUE (page0)				**(ACCESS[C9,CA]) 0x15**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
S5_VALUE	7:0	R	--	**Normalized Histogram S5 Value**	

Register:: S6_VALUE (page0)				**(ACCESS[C9,CA]) 0x16**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
S6_VALUE	7:0	R	--	**Normalized Histogram S6 Value**	

Register:: YHL_THD (page0)				**(ACCESS[C9,CA]) 0x17**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
YHL_THD	7:0	R/W	0x00	**Y_H and Y_L Theshold** When DIFF[10:0] < YHL_THD[7:0], Y_H and Y_L keep the previous values	

Register:: DEF_CRV[01] (page1)				**(ACCESS[C9,CA]) 0x00**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DEF_CRV01	7:0	R/W	0x10	**Pre-Defined Y-Curve; Keep DEF_CRV[N] ≥ DEF_CRV[N-1]**	

Register:: DEF_CRV[02] (page1)				**(ACCESS[C9,CA]) 0x01**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DEF_CRV02	7:0	R/W	0x20	**Pre-Defined Y-Curve; Keep DEF_CRV[N] ≥ DEF_CRV[N-1]**	

Register:: DEF_CRV[03] (page1)				**(ACCESS[C9,CA]) 0x02**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DEF_CRV03	7:0	R/W	0x30	**Pre-Defined Y-Curve; Keep DEF_CRV[N] ≥ DEF_CRV[N-1]**	

Register:: DEF_CRV[04] (page1)				**(ACCESS[C9,CA]) 0x03**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DEF_CRV04	7:0	R/W	0x40	**Pre-Defined Y-Curve; Keep DEF_CRV[N] ≥ DEF_CRV[N-1]**	

Register:: DEF_CRV[05] (page1)				**(ACCESS[C9,CA]) 0x04**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DEF_CRV05	7:0	R/W	0x50	**Pre-Defined Y-Curve; Keep DEF_CRV[N] ≥ DEF_CRV[N-1]**	

Register:: DEF_CRV[06] (page1)				**(ACCESS[C9,CA]) 0x05**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DEF_CRV06	7:0	R/W	0x60	**Pre-Defined Y-Curve; Keep DEF_CRV[N] ≥ DEF_CRV[N-1]**	

Register:: DEF_CRV[07] (page1)				**(ACCESS[C9,CA]) 0x06**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DEF_CRV07	7:0	R/W	0x70	**Pre-Defined Y-Curve; Keep DEF_CRV[N] ≥ DEF_CRV[N-1]**	

Register:: DEF_CRV[08] (page1)				**(ACCESS[C9,CA]) 0x07**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DEF_CRV08	7:0	R/W	0x80	**Pre-Defined Y-Curve; Keep DEF_CRV[N] ≥ DEF_CRV[N-1]**	

Register:: DEF_CRV[09] (page1)				**(ACCESS[C9,CA]) 0x08**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DEF_CRV09	7:0	R/W	0x90	**Pre-Defined Y-Curve; Keep DEF_CRV[N] ≥ DEF_CRV[N-1]**	

Register:: DEF_CRV[10] (page1)				**(ACCESS[C9,CA]) 0x09**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DEF_CRV10	7:0	R/W	0xA0	**Pre-Defined Y-Curve; Keep DEF_CRV[N] ≥ DEF_CRV[N-1]**	

Register:: DEF_CRV[11] (page1)			(ACCESS[C9,CA]) 0x0A		
Name	Bits	R/W	Default	Comments	Config
DEF_CRV11	7:0	R/W	0xB0	**Pre-Defined Y-Curve; Keep DEF_CRV[N] ≥ DEF_CRV[N-1]**	

Register:: DEF_CRV[12] (page1)			(ACCESS[C9,CA]) 0x0B		
Name	Bits	R/W	Default	Comments	Config
DEF_CRV12	7:0	R/W	0xC0	**Pre-Defined Y-Curve; Keep DEF_CRV[N] ≥ DEF_CRV[N-1]**	

Register:: DEF_CRV[13] (page1)			(ACCESS[C9,CA]) 0x0C		
Name	Bits	R/W	Default	Comments	Config
DEF_CRV13	7:0	R/W	0xD0	**Pre-Defined Y-Curve; Keep DEF_CRV[N] ≥ DEF_CRV[N-1]**	

Register:: DEF_CRV[14] (page1)			(ACCESS[C9,CA]) 0x0D		
Name	Bits	R/W	Default	Comments	Config
DEF_CRV14	7:0	R/W	0xE0	**Pre-Defined Y-Curve; Keep DEF_CRV[N] ≥ DEF_CRV[N-1]**	

Register:: DEF_CRV[15] (page1)			(ACCESS[C9,CA]) 0x0E		
Name	Bits	R/W	Default	Comments	Config
DEF_CRV15	7:0	R/W	0xF0	**Pre-Defined Y-Curve; Keep DEF_CRV[N] ≥ DEF_CRV[N-1]**	

Register:: DEF_CRV[16] (page1)			(ACCESS[C9,CA]) 0x0F		
Name	Bits	R/W	Default	Comments	Config
DEF_CRV16	7:0	R/W	0x00	**Pre-Defined Y-Curve; Keep DEF_CRV[N] ≥ DEF_CRV[N-1]** **Note : default = 0x00 means 0x100 (256)**	

When y-curve boundary is changed (DEF_CRV[16] != 0x00), disable histogram noise filter.

Registers below is effective only when auto mode is disable and black/white level expansion is enabled.
When auto mode is enabled (DCC_MODE=0), Y_BL_BIAS and Y_WL_BIAS are read-only.

Register:: Y_BL_BIAS (page1)			(ACCESS[C9,CA]) 0x10		
Name	Bits	R/W	Default	Comments	Config
Y_BL_BIAS	7:0	R/W	0x00	**Y Offset for Black-Level Expansion (Y_L' = 4*Y_BL_BIAS)**	

Register:: Y_WL_BIAS (page1)			(ACCESS[C9,CA]) 0x11		
Name	Bits	R/W	Default	Comments	Config
Y_WL_BIAS	7:0	R/W	0x00	**Y Offset for While-Level Expansion (1023-Y_H' = 4*Y_WL_BIAS)**	

Load double buffer CRED-00 ~ CRED-11 (page1) after write CRED-11 when DCC enable

Register:: SAT_FACTOR (page1)			(ACCESS[C9,CA]) 0x12		
Name	Bits	R/W	Default	Comments	Config
Reserved	7:6	--	--	Reserved	
SAT_FACTOR	5:0	R/W	0x00	**Saturation Compensation Factor = 0 ~ 32.**	

Registers below is effective only when auto mode is enabled.
In manual mode (DCC_MODE=1), BLD_VAL will be fixed to 0. It means Y-curve is fully determined by DEF_CUR[01~15]

Register:: BLD_UB (page1)			(ACCESS[C9,CA]) 0x13		
Name	Bits	R/W	Default	Comments	Config
BLD_UB	7:0	R/W	0x00	**Upper Bound of Blending Factor**	

Register:: BLD_LB (page1)			(ACCESS[C9,CA]) 0x14		
Name	Bits	R/W	Default	Comments	Config
BLD_LB	7:0	R/W	0x00	**Lower Bound of Blending Factor**	

Register:: DEV_FACTOR (page1)			(ACCESS[C9,CA]) 0x15		

Name	Bits	R/W	Default	Comments	Config
DEV_FACTOR	7:0	R/W	0x00	**Deviation Weighting Factor**	

Register:: BLD_VAL_SEL (page1)			**(ACCESS[C9,CA])**	**0x16**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
WL_RANGE	7:6	R/W	0x00	**White-Level Range** 00: Yi = 512 (Z8) 01: Yi = 576 (Z9) 10: Yi = 640 (Z10) 11: Yi = 704 (Z11)	
WL_BLD_VAL	5:4	R/W	0x00	**White-Level Blending Factor** 00: 0 (user-defined curve) 01: R/2 10: R 11: 2R	
BL_RANGE	3:2	R/W	0x00	**Black-Level Range** 00: Yi = 448 (Z7) 01: Yi = 384 (Z6) 10: Yi = 320 (Z5) 11: Yi = 256 (Z4)	
BL_BLD_VAL	1:0	R/W	0x00	**Black-Level Blending Factor** 00: 0 (user-defined curve) 01: R/2 10: R 11: 2R	

Register:: BLD_VAL (page1)			**(ACCESS[C9,CA])**	**0x17**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BLD_VAL	7:0	R	--	**= Max{ BLD_UB – [(DEV_VAL*DEV_FACTOR)/256], BLD_LB}**	

Register:: DEV_VAL_HI (page1)			**(ACCESS[C9,CA])**	**0x18**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DEV_VAL_HI	7:0	R	--	**Bit[8:1] of Deviation Value**	

Register:: DEV_VAL_LO (page1)			**(ACCESS[C9,CA])**	**0x19**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DEV_VAL_LO	7	R	--	**Bit[0] of Deviation Value**	
Reserved	6:0	--	--	Reserved	

Register:: SRAM initial value (page2)			**(ACCESS[C9,CA])**	**0x00~0x8F**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
SRAM_XX	7:0	W	--	**Addr 00: SRAM_00** **Addr 01: SRAM_01** **….** **Addr 8F : SRAM_8F**	

Register:: SRAM_BIST (page3)			**(ACCESS[C9,CA])**	**0x00**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BIST_EN	7	R/W	0	**BIST_EN** 0: disable 1: enable	
RAM_Mode	6	R/W	0	**RAM_Mode** 0: dclk domain mode (normal mode, BIST) 1: MCU domain mode (SCG test)	
Reserved	5:2	--	--	Reserved	
BIST_PERIOD	1	R	--	**BIST_Period** 0: BIST is done	

				1: BIST is running	
BIST_OK	0	R	--	**BIST_OK** 0: SRAM fail 1: SRAM ok	

ICM(Page 7)

Address: D0 **ICM Control** **Default: 00h**

Bit	Mode	Function
7	R/W	**ICM Enable** 0: Disable 1: Enable
6	R/W	**Y Correction Mode** 0: dY = (8dU+dV)/8 1: dY = (6dU+dV)/8
5	R/W	**ICM U/V Delta Range:** 0: Original -128~+127 1: Double -256~254
4	R/W	**CM0 Enable** 0: Disable 1: Enable
3	R/W	**CM1 Enable** 0: Disable 1: Enable
2	R/W	**CM2 Enable** 0: Disable 1: Enable
1	R/W	**CM3 Enable** 0: Disable 1: Enable
0	R/W	**CM4 Enable** 0: Disable 1: Enable

Address: D1 **ICM_SEL** **Default: 00h**

Bit	Mode	Function
7:5	R/W	**ICM Test Mode** 000: disable 001: bypass U, V result 010: bypass hue/saturation result 011: bypass dU, dV value 1xx: R,B as LUT input, and bypass LUT output to R/G/B output
4	R/W	**delta U/V range extend** **0: delta U/V x1 (default)** **1: delta U/V x4**
3	R/W	**CM5 Enable** 0: Disbale 1: Enable
2:0	R/W	**CM Select** 000: Select Chroma Modifier 0 for Accessing Through Data Port 001: Select Chroma Modifier 1 for Accessing Through Data Port 010: Select Chroma Modifier 2 for Accessing Through Data Port 011: Select Chroma Modifier 3 for Accessing Through Data Port 100: Select Chroma Modifier 4 for Accessing Through Data Port 101: Select Chroma Modifier 5 for Accessing Through Data Port 110~111: reserved

Address: D2 **ICM_ADDR** **Default: 00h**

Bit	Mode	Function
7	R/W	**Negative Compensate Enable (1 : enable / 0 : disable)**
6	-	**Reserved**
5:0	R/W	**ICM port address**

Address: D3 **ICM_Data**

Bit	Mode	Function
7:0	R/W	ICM port data

ICM_ADDR will be increased automatically after each byte of ICM_DATA has been accessed.

Address: D3-00 MST_HUE_HB **Default: x0h**

Bit	Mode	Function
7:4	--	Reserved
3:0	W	High Byte[11:8] of Master Hue for Chroma Modifier N.

Address: D3-01 MST_HUE_LB **Default: 00h**

Bit	Mode	Function
7:0	W	Low Byte[7:0] of Master Hue for Chroma Modifier N.

Address: D3-02 HUE_SET **Default: 00h**

Bit	Mode	Function
7:6	W	**CM[N]_LWID** 00: CM[N] left width = 64 01: CM[N] left width = 128 10: CM[N] left width = 256 11: CM[N] left width = 512
5:4	W	**CM[N]_LBUF** 00: CM[N] left Buffer = 0 01: CM[N] left Buffer = 64 10: CM[N] left Buffer = 128 11: CM[N] left Buffer = 256
3:2	W	**CM[N]_RWID** 00: CM[N] right width = 64 01: CM[N] right width = 128 10: CM[N] right width = 256 11: CM[N] right width = 512
1:0	W	**CM[N]_RBUF** 00: CM[N] right Buffer = 0 01: CM[N] right Buffer = 64 10: CM[N] right Buffer = 128 11: CM[N] right Buffer = 256

Address: D3-03~32 U/V Offset **Default: 00h**

Bit	Mode	Function
7:0	W	**Addr 03: U Offset 00, -128~127** **Addr 04: V Offset 00, -128~127** **Addr 05: U Offset 01, -128~127** **Addr 06: V Offset 01, -128~127** **Addr 07: U Offset 02, -128~127** **Addr 08: V Offset 02, -128~127** **Addr 09: U Offset 03, -128~127** **Addr 0A: V Offset 03, -128~127** **Addr 0B: U Offset 04, -128~127** **Addr 0C: V Offset 04, -128~127** **Addr 0D: U Offset 05, -128~127** **Addr 0E: V Offset 05, -128~127** **Addr 0F: U Offset 06, -128~127** **Addr 10: V Offset 06, -128~127** **Addr 11: U Offset 07, -128~127** **Addr 12: V Offset 07, -128~127** **Addr 13: U Offset 10, -128~127** **Addr 14: V Offset 10, -128~127** **Addr 15: U Offset 11, -128~127** **Addr 16: V Offset 11, -128~127** **Addr 17: U Offset 12, -128~127** **Addr 18: V Offset 12, -128~127** **Addr 19: U Offset 13, -128~127**

		Addr 1A: V Offset 13, -128~127 **Addr 1B: U Offset 14, -128~127** **Addr 1C: V Offset 14, -128~127** **Addr 1D: U Offset 15, -128~127** **Addr 1E: V Offset 15, -128~127** **Addr 1F: U Offset 16, -128~127** **Addr 20: V Offset 16, -128~127** **Addr 21: U Offset 17, -128~127** **Addr 22: V Offset 17, -128~127** **Addr 23: U Offset 20, -128~127** **Addr 24: V Offset 20, -128~127** **Addr 25: U Offset 21, -128~127** **Addr 26: V Offset 21, -128~127** **Addr 27: U Offset 22, -128~127** **Addr 28: V Offset 22, -128~127** **Addr 29: U Offset 23, -128~127** **Addr 2A: V Offset 23, -128~127** **Addr 2B: U Offset 24, -128~127** **Addr 2C: V Offset 24, -128~127** **Addr 2D: U Offset 25, -128~127** **Addr 2E: V Offset 25, -128~127** **Addr 2F: U Offset 26, -128~127** **Addr 30: V Offset 26, -128~127** **Addr 31: U Offset 27, -128~127** **Addr 32: V Offset 27, -128~127**

DCR(Page 7)

Register::DCR Address Port			0xD8		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DCR_ADDR	7:2	--	0	DCR address	
RESULT_READ	1	R/W	0	0: Disable Read to refresh measure result. 1: Read DCR measure result.	
MEASURE_START	0	R/W	0	0: Finish or disable 1: Start DCR computation.	

Register:: DCR Data Port			0xD9		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DCR_DATA	7:0	R/W	0x00	DCR data	

Register:: DCR_THESHOLD1			(ACCESS[D8,D9]) 0x00		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
THRESHOLD1_VALUE	7:0	R/W	0x08	DCR threshold1. (R+G+B)*0.75	

If we want to set threshold1 = 200. THRESHOLD1_VALUE = 200*0.75 = 150.

Register:: DCR_THESHOLD2			(ACCESS[D8,D9]) 0x01		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
THRESHOLD2_VALUE	7:0	R/W	0x60	DCR threshold2. (threshold2 > threshold1) (R+G+B)*0.75	

If we want to set threshold2 = 200. THRESHOLD2_VALUE = 200*0.75 = 150.

If we expect the target_thd = 10b'R+10'bG+10'bB /3
And in firmware that THD*4 = R+G+B
so that → 3* target_thd = 4* THD , => THD = 3/4 target_thd.

Register::DCR_ABOVE_TH1_NUM_2			(ACCESS[D8,D9]) 0x02		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ABOVE_TH1_NUM_2	7:0	R	0	Total pixel number above threshold1: bit[23:16]	

Register::DCR_ABOVE_TH1_NUM_1			(ACCESS[D8,D9]) 0x03		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ABOVE_TH1_NUM_1	7:0	R	0	Total pixel number above threshold1: bit[15:8]	

Register::DCR_ABOVE_TH1_NUM_0			(ACCESS[D8,D9]) 0x04		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ABOVE_TH1_NUM_0	7:0	R	0	Total pixel number above threshold1: bit[7:0]	

Register::DCR_ABOVE_TH1_VAL_3			(ACCESS[D8,D9]) 0x05		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ABOVE_TH1_VAL_3	7:0	R	0	Total sum (R+G+B) of pixel value above threshold1: bit[31:24]	

Register::DCR_ABOVE_TH1_VAL_2			(ACCESS[D8,D9]) 0x06		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ABOVE_TH1_VAL_2	7:0	R	0	Total sum (R+G+B) of pixel value above threshold1: bit[23:16]	

Register::DCR_ABOVE_TH1_VAL_1			(ACCESS[D8,D9]) 0x07		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ABOVE_TH1_VAL_1	7:0	R	0	Total sum (R+G+B) of pixel value above threshold1: bit[15:8]	

Register::DCR_ABOVE_TH1_VAL_0			(ACCESS[D8,D9]) 0x08		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ABOVE_TH1_VAL_0	7:0	R	0	Total sum (R+G+B) of pixel value above threshold1: bit[7:0]	

Register::DCR_ABOVE_TH2_NUM_2			(ACCESS[D8,D9]) 0x09		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ABOVE_TH2_NUM_2	7:0	R	0	Total pixel number above threshold2: bit[23:16]	

Register::DCR_ABOVE_TH2_NUM_1			(ACCESS[D8,D9]) 0x0A		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ABOVE_TH2_NUM_1	7:0	R	0	Total pixel number above threshold2: bit[15:8]	

Register::DCR_ABOVE_TH2_NUM_0			(ACCESS[D8,D9]) 0x0B		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ABOVE_TH2_NUM_0	7:0	R	0	Total pixel number above threshold2: bit[7:0]	

Register::DCR_ABOVE_TH2_VAL_3			(ACCESS[D8,D9]) 0x0C		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ABOVE_TH2_VAL_3	7:0	R	0	Total sum (R+G+B) of pixel value above threshold2: bit[31:24]	

Register::DCR_ABOVE_TH2_VAL_2			(ACCESS[D8,D9]) 0x0D		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ABOVE_TH2_VAL_2	7:0	R	0	Total sum (R+G+B) of pixel value above threshold2: bit[23:16]	

Register::DCR_ABOVE_TH2_VAL_1			(ACCESS[D8,D9]) 0x0E		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ABOVE_TH2_VAL_1	7:0	R	0	Total sum (R+G+B) of pixel value above threshold2: bit[15:8]	

Register::DCR_ABOVE_TH2_VAL_0			(ACCESS[D8,D9]) 0x0F		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ABOVE_TH2_VAL_0	7:0	R	0	Total sum (R+G+B) of pixel value above threshold2: bit[7:0]	

Register::DCR_HIGH_LV_NUM_R_1			(ACCESS[D8,D9]) 0x10		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
HIGH_LV_NUM_R_1	7:0	R	0	Dynamically detect highest level pixel number of red channel. RMAX_NUM[15:8]	

Register::DCR_HIGH_LV_NUM_R_0				(ACCESS[D8,D9]) 0x11	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
HIGH_LV_NUM_R_0	7:0	R	0	Dynamically detect highest level pixel number of red channel. RMAX_NUM[7:0]	

Register::DCR_LOW_LV_NUM_R_1				(ACCESS[D8,D9]) 0x12	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LOW_LV_NUM_R_1	7:0	R	0	Dynamically detect the lowest level pixel number of red channel. RMIN_NUM[15:8]	

Register::DCR_LOW_LV_NUM_R_0				(ACCESS[D8,D9]) 0x13	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LOW_LV_NUM_R_0	7:0	R	0	Dynamically detect the lowest level pixel number of red channel. RMIN_NUM[7:0]	

Register::DCR_HIGH_LV_VAL_R				(ACCESS[D8,D9]) 0x14	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
HIGH_LV_VAL_R	7:0	R	0	Dynamically detect highest level value of red channel.	

Register::DCR_LOW_LV_VAL_R				(ACCESS[D8,D9]) 0x15	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LOW_LV_VAL_R	7:0	R	0	Dynamically detect the lowest level value of red channel.	

Register::DCR_HIGH_LV_NUM_G_1				(ACCESS[D8,D9]) 0x16	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
HIGH_LV_NUM_G_1	7:0	R	0	Dynamically detect the highest level pixel number of green channel. GMAX_NUM[15:8]	

Register::DCR_HIGH_LV_NUM_G_0				(ACCESS[D8,D9]) 0x17	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
HIGH_LV_NUM_G_0	7:0	R	0	Dynamically detect the highest level pixel number of green channel. GMAX_NUM[7:0]	

Register::DCR_LOW_LV_NUM_G_1				(ACCESS[D8,D9]) 0x18	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LOW_LV_NUM_G_1	7:0	R	0	Dynamically detect the lowest level pixel number of green channel. GMIN_NUM[15:8]	

Register::DCR_LOW_LV_NUM_G_0				(ACCESS[D8,D9]) 0x19	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LOW_LV_NUM_G_0	7:0	R	0	Dynamically detect the lowest level pixel number of green channel. GMIN_NUM[7:0]	
Register::DCR_HIGH_LV_VAL_G				**(ACCESS[D8,D9]) 0x1A**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**

HIGH_LV_VAL_G	7:0	R	0	Dynamically detect the highest level value of green channel.	

Register::DCR_LOW_LV_VAL_G			(ACCESS[D8,D9]) 0x1B		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LOW_LV_VAL_G	7:0	R	0	Dynamically detect the lowest level value of green channel.	

Register::DCR_HIGH_LV_NUM_B_1			(ACCESS[D8,D9]) 0x1C		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
HIGH_LV_NUM_B_1	7:0	R	0	Dynamically detect the highest level pixel number of blue channel. BMAX_NUM[15:8]	

Register::DCR_HIGH_LV_NUM_B_0			(ACCESS[D8,D9]) 0x1D		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
HIGH_LV_NUM_B_0	7:0	R	0	Dynamically detect the highest level pixel number of blue channel. BMAX_NUM[7:0]	

Register::DCR_LOW_LV_NUM_B_1			(ACCESS[D8,D9]) 0x1E		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LOW_LV_NUM_B_1	7:0	R	0	Dynamically detect the lowest level pixel number of blue channel. BMIN_NUM[15:8]	

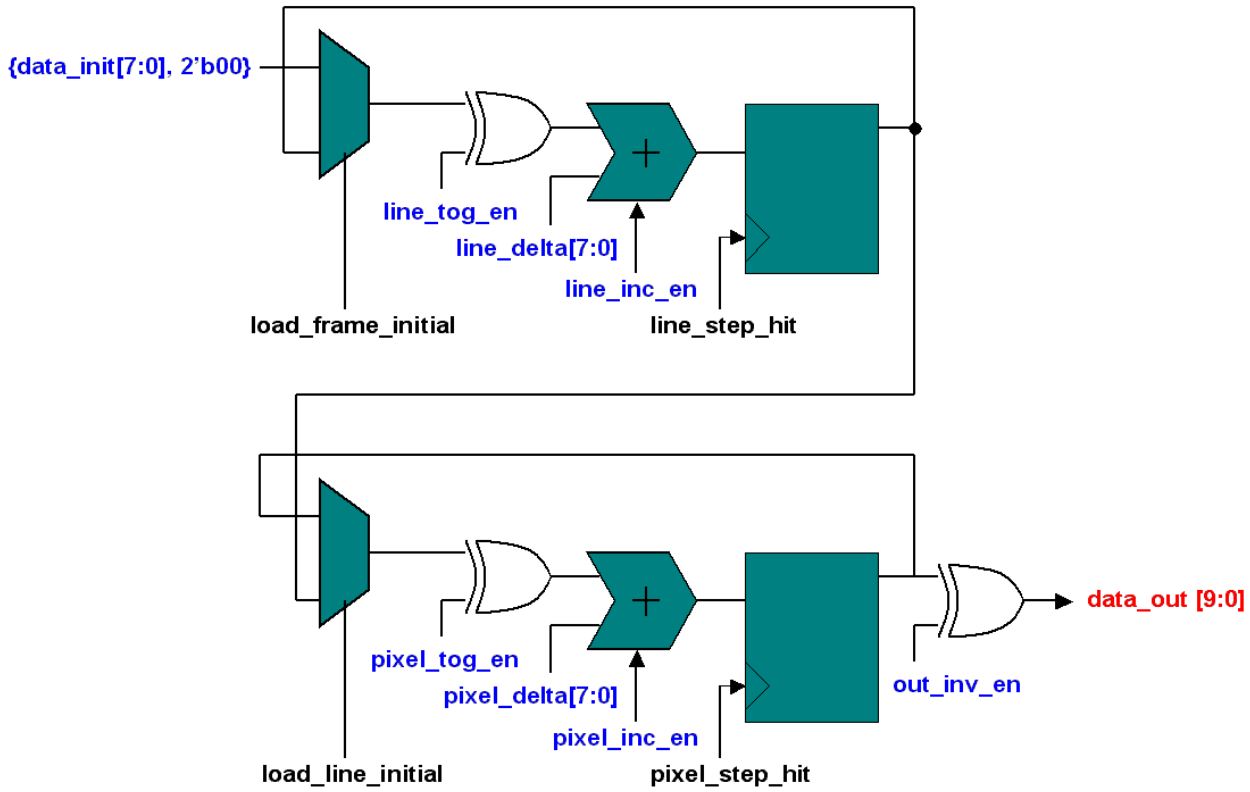
Register::DCR_LOW_LV_NUM_B_0			(ACCESS[D8,D9]) 0x1F		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LOW_LV_NUM_B_0	7:0	R	0	Dynamically detect the lowest level pixel number of blue channel. BMIN_NUM[7:0]	

Register:: DCR_HIGH_LV_VAL_B			(ACCESS[D8,D9]) 0x20		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
HIGH_LV_VAL_B	7:0	R	0	Dynamically detect the highest level value of blue channel.	

Register:: DCR_LOW_LV_VAL_B			(ACCESS[D8,D9]) 0x21		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LOW_LV_VAL_B	7:0	R	0	Dynamically detect the lowest level value of blue channel.	

Pattern Generator in D-Domain (Page 7)

RTD2485XD supports programmable patterns, such as gray-level, chessboard, dot-pattern, etc., for display image testing.



Register::DISP_PG_R_CTRL				0xF0	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PG_ENABLE	7	R/W	0	Dispaly Pattern Gen. Function Enable	
PG_R_CTRL_DUM	6	R/W	0	Dummy	
PG_ROUT_INV_EN	5	R/W	0	Inverse Data Output	
PG_R_CLAMP_EN	4	R/W	0	Adder result clamp to 10'h3FFF	
LINE_R_TOG_EN	3	R/W	0	Data toggled in each line -step	
LINE_R_INC_EN	2	R/W	0	Data increment in each line-step	
PIXEL_R_TOG_EN	1	R/W	0	Data toggled in each pixel-step	
PIXEL_R_INC_EN	0	R/W	0	Data incremented in each pixel-step	

Register::DISP_PG_G_CTRL				0xF1	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**

PG_G_CTRL_DUM	7:6	R/W	0	Dummy	
PG_GOUT_INV_EN	5	R/W	0	Inverse Data Output	
PG_G_CLAMP_EN	4	R/W	0	Adder result clamp to 10'h3FFF	
LINE_G_TOG_EN	3	R/W	0	Data toggled in each line -step	
LINE_G_INC_EN	2	R/W	0	Data increment in each line-step	
PIXEL_G_TOG_EN	1	R/W	0	Data toggled in each pixel-step	
PIXEL_G_INC_EN	0	R/W	0	Data incremented in each pixel-step	

Register::DISP_PG_B_CTRL				**0xF2**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PG_B_CTRL_DUM	7:6	R/W	0	Dummy	
PG_BOUT_INV_EN	5	R/W	0	Inverse Data Output	
PG_B_CLAMP_EN	4	R/W	0	Adder result clamp to 10'h3FFF	
LINE_B_TOG_EN	3	R/W	0	Data toggled in each line -step	
LINE_B_INC_EN	2	R/W	0	Data increment in each line-step	
PIXEL_B_TOG_EN	1	R/W	0	Data toggled in each pixel-step	
PIXEL_B_INC_EN	0	R/W	0	Data incremented in each pixel-step	

Register::DISP_PG_R_Initial				**0xF3**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PG_R_INIT	7:0	R/W	0	Initial Pattern Value for Red Data [9:2]	

Register::DISP_PG_G_Initial				**0xF4**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PG_G_INIT	7:0	R/W	0	Initial Pattern Value for Green Data [9:2]	

Register::DISP_PG_B_Initial				**0xF5**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PG_B_INIT	7:0	R/W	0	Initial Pattern Value for Blue Data [9:2]	

Register::DISP_PG_Pixel_Delta				**0xF6**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PG_PIXEL_DELTA	7:0	R/W	0	Pixel Delta value for incremental	

Register::DISP_PG_Line_Delta				**0xF7**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PG_LINE_DELTA	7:0	R/W	0	Line Delta value for incremental	

Register::DISP_PG_Pixel_Step_MSB				**0xF8**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**

PG_PIXEL_STEP_M	7:0	R/W	01h	Pixel Step for toggle/incremental, can not be 0	

Register::DISP_PG_Line_Step_MSB				**0xF9**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PG_LINE_STEP_M	7:0	R/W	01h	Line Step for toggle/incremental, can not be 0	

Register::DISP_PG Step_LSB				**0xFA**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LINE_STEP_DUM	7:6	R/W	0	Dummy	
PG_LINE_STEP_L	5:4	R/W	0	Decimal part for Line-step	
PIXEL_STEP_DUM	3:2	R/W	0	Dummy	
PG_PIXEL_STEP_L	1:0	R/W	0	Decimal part for Pixel-step	

Ex: If the pattern is 256 gray level in 640 pixels, the wanted pixel_step is 640/256 = 2.5. Hence, PG_PIXEL_STEP_M = 2h and PG_PIXEL_STEP_L = 2'b10.

({PG_PIXEL_STEP_M, PG_PIXEL_STEP_L} = 2.5).

Audio Control (Page 8)

Auto Volume Control

Register::Auto_Volume_Control_0				0xA0	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
AVC_EN	7	R/W	0	H/W AVC Enable 0b:Disable(clear internal state) 1b:Enable	
AVC_NOISE_GAIN	6:5	R/W	0	00: -6dB 01: -12dB 10: -18dB 11: -24dB	
reserved	4	--	--	reserved	
AVC_DOWN_DB	3:2	R/W	1	00: -1dB (gain1=gain1*7/8, gain2=gain2-1/8) 01: -1/2dB (gain1=gain1*15/16, gain2=gain2-1/16) 10: -1/4dB (gain1=gain1*31/32, gain2=gain2-1/32) 11: -1/8dB (gain1=gain1*63/64, gain2=gain2-1/64)	
AVC_UP_DB	1:0	R/W	1	00: +1dB (gain1=gain1*9/8, gain2=gain2+1/8) 01: +1/2dB (gain1=gain1*17/16, gain2=gain2+1/16) 10: +1/4dB (gain1=gain1*33/32, gain2=gain2+1/32) 11: +1/8dB (gain1=gain1*65/64, gain2=gain2+1/64)	

0xA1 is reserved

Register:: AVC_NOISE_THRE				0xA2	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
AVC_NOISE_THRE	7:0	R/W	0x10	real_noise_threshold = avc_noise_thre/2^15 no avc_noise action when avc_noise_thre = 8'h0	

Register::AVC_LEVEL_MAX				0xA3	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
AVC_LEVEL_MAX	7:0	R/W	0x10	real_level_max = avc_level_max/2^7	

Register::AVC_LEVEL_MIN_0				0xA4	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
AVC_LEVEL_MIN	7:0	R/W	0x0C	real_level_min = avc_level_min/2^7	

Register::AVC_CNT_THRE				0xA5	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
AVC_CNT_THRE	7:0	R/W	0xc8	if(counter>avc_cnt_thre) gain = gain*avc_down_db else if(counter<-avc_cnt_thre) gain = gain*avc_up_db else gain = gain	

A6 is reserved

Register::AVC_WINDOW_CONTROL_0				0xA6	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
AVC_ZC_WIN[11:8]	7:4	R/W	0x2	AVC_ZC_WIN[11:8] : zero-crossing time-out window size (unit: pcm samples)	
AVC_MON_WIN[11:8]	3:0	R/W	0x8	AVC_MON_WIN[11:8] monitor window size (unit: pcm samples)	

Register::AVC_WINDOW_CONTROL_1				0xA7	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
AVC_ZC_WIN[7:0]	7:0	R/W	0	AVC_ZC_WIN[7:0] : zero-crossing time-out window size (unit: pcm samples)	

Register::AVC_WINDOW_CONTROL_2				0xA8	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
AVC_MON_WIN[7:0]	7:0	R/W	0	AVC_MON_WIN[7:0]	

				monitor window size (unit: pcm samples)	

EQUALIZER

Register::EQ_CONTROL				**0xAA**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
EQ_EN	7	R/W	0	Digital EQ Enable	
eq_band5_enable	6	R/W	0	EQ_Band-6 Control. 0: Disable 1: Enable.❶	
eq_band4_enable	5	R/W	0	EQ_Band-5 Control. 0: Disable 1: Enable.❶	
eq_band3_enable	4	R/W	0	EQ_Band-4 Control. 0: Disable 1: Enable.❶	
eq_band2_enable	3	R/W	0	EQ_Band-3 Control. 0: Disable 1: Enable.❶	
eq_band1_enable	2	R/W	0	EQ_Band-2 Control. 0: Disable 1: Enable.❶	
eq_lp_enable	1	R/W	0	EQ_Band-1 Control. 0: Disable 1: Enable.❶	
eq_hp_enable	0	R/W	0	EQ_Band-0 Control. 0: Disable 1: Enable.❶	

Register::EQ_STATUS				**0xAB**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_db_en	7	R/W	0	EQ Coefficient Double Buffer Enable (AC-D5) Write 1 to clear	Wclr_out
reserved	6:0	--	--	reserved	

Below EQ bypass filters(AC-D5) should be double buffer by AB[7]

Register:: EQ_a1_0				**0xAC**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_lp_a1[15:8]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a1_1				**0xAD**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_lp_a1[7:0]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a1_2				**0xAE**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_bp1_a1[15:8]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a1_3				**0xAF**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_bp1_a1[7:0]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a1_4				**0xB0**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_bp2_a1[15:8]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a1_5				**0xB1**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_bp2_a1[7:0]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a1_6				**0xB2**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_bp3_a1[15:8]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a1_7	**0xB3**

Name	Bits	R/W	Default	Comments	Config
eq_bp3_a1[7:0]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a1_8				0xB4	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_bp4_a1[15:8]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a1_9				0xB5	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_bp4_a1[7:0]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a1_A				0xB6	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_bp5_a1[15:8]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a1_B				0xB7	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_bp5_a1[7:0]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a1_C				0xB8	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_hp_a1[15:8]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a1_D				0xB9	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_hp_a1[7:0]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a2_0				0xBA	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_bp1_a2[15:8]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a2_1				0xBB	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_bp1_a2[7:0]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a2_2				0xBC	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_bp2_a2[15:8]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a2_3				0xBD	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_bp2_a2[7:0]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a2_4				0xBE	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_bp3_a2[15:8]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a2_5	0xBF

Name	Bits	R/W	Default	Comments	Config
eq_bp3_a2 [7:0]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a2_6				0xC0	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_bp4_a2[15:8]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a2_7				0xC1	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_bp4_a2[7:0]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a2_8				0xC2	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_bp5_a2[15:8]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_a2_9				0xC3	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_bp5_a2[7:0]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -2 ~ 1.99.	

Register:: EQ_h0_0				0xC4	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
reserved	7:1	--	--	reserved	
eq_hp_slope	0	R/W	0	0:20dB/decade slope 1: 40dB/decade slope (2 cascaded HP6)	

Register:: EQ_h0_1				0xC5	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_lp_h0[15:8]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -5 ~ 4.99. (Gain=-20dB ~ +20dB)	

Register:: EQ_h0_2				0xC6	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_lp_h0[7:0]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -5 ~ 4.99. (Gain=-20dB ~ +20dB)	

Register:: EQ_h0_3				0xC7	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_bp1_h0[15:8]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -5 ~ 4.99. (Gain=-20dB ~ +20dB)	

Register:: EQ_h0_4				0xC8	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_bp1_h0[7:0]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -5 ~ 4.99. (Gain=-20dB ~ +20dB)	

Register:: EQ_h0_5				0xC9	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_bp2_h0[15:8]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -5 ~ 4.99. (Gain=-20dB ~ +20dB)	

Register:: EQ_h0_6				0xCA	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
eq_bp2_h0[7:0]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -5 ~ 4.99. (Gain=-20dB ~ +20dB)	

Register:: EQ_h0_7				0xCB	
Name	Bits	R/W	Default	Comments	Config
eq_bp3_h0[15:8]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -5 ~ 4.99. (Gain=-20dB ~ +20dB)	

Register:: EQ_h0_8				0xCC	
Name	Bits	R/W	Default	Comments	Config
eq_bp3_h0[7:0]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -5 ~ 4.99. (Gain=-20dB ~ +20dB)	

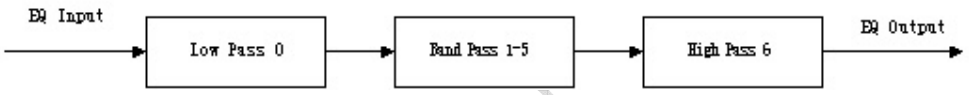
❶Gain is between –20dB ~ +20dB, the Ho should be between –1 ~4.99.

Register:: EQ_h0_9				0xCD	
Name	Bits	R/W	Default	Comments	Config
eq_bp4_h0[15:8]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -5 ~ 4.99. (Gain=-20dB ~ +20dB)	

Register:: EQ_h0_A				0xCE	
Name	Bits	R/W	Default	Comments	Config
eq_bp4_h0[7:0]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -5 ~ 4.99. (Gain=-20dB ~ +20dB)	

Register:: EQ_h0_B				0xCF	
Name	Bits	R/W	Default	Comments	Config
eq_bp5_h0[15:8]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -5 ~ 4.99. (Gain=-20dB ~ +20dB)	

Register:: EQ_h0_C				0xD0	
Name	Bits	R/W	Default	Comments	Config
eq_bp5_h0[7:0]	7:0	R/W	0	16-bit 2's complement coefficient. Range= -5 ~ 4.99. (Gain=-20dB ~ +20dB)	



Register:: EQ_over_under_flow				0xD1	
Name	Bits	R/W	Default	Comments	Config
reserved	7:5	--	--	reserved	
Ovud_clear	4	R/W	0	Write 1 to clear the ovfl udfl flags below	Wclr_out
eq_l_ovfl	3	R	0	EQ L output overflow	
eq_l_udfl	2	R	0	EQ L output underflow	
eq_r_ovfl	1	R	0	EQ R output overflow	
eq_r_udfl	0	R	0	EQ_R output underflow	

Register:: Filter_over_under_flow				0xD2	
Name	Bits	R/W	Default	Comments	Config
Fileter_Ovud_clear	7	R/W	0	Write 1 to clear the ovfl udfl flags below	Wclr_out
eq_band5_of	6	R	0	EQ_Band-6 Status. 0: Normal 1: Overflow. This bit is set if overflow had ever occurred. Write 1 to clear it.	
eq_band4_of	5	R	0	EQ_Band-5 Status. 0: Normal 1: Overflow. This bit is set if overflow had ever occurred. Write 1 to clear it.	
eq_band3_of	4	R	0	EQ_Band-4 Status. 0: Normal 1: Overflow. This bit is set if overflow had ever occurred. Write 1 to	

				clear it.	
eq_band2_of	3	R	0	EQ_Band-3 Status. 0: Normal 1: Overflow. This bit is set if overflow had ever occurred. Write 1 to clear it.	
eq_band1_of	2	R	0	EQ_Band-2 Status. 0: Normal 1: Overflow. This bit is set if overflow had ever occurred. Write 1 to clear it.	
eq_lp_of	1	R	0	EQ_Band-1 Status. 0: Normal 1: Overflow. This bit is set if overflow had ever occurred. Write 1 to clear it.	
eq_hp_of	0	R	0	EQ_Band-0 Status. 0: Normal 1: Overflow. This bit is set if overflow had ever occurred. Write 1 to clear it.	

PEAK DETECTION

Register:: PEAK_CTRL_0				**0xD3**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Pdetect_enable	7	R/W	0	Peak detect enable 0: disable 1: enable	
Rev_peak_d3	6:4	R/W	0	Debug mode	
Peak_ovfl_det_l_irq_en	3	R/W	0	Left channel peak overflow detected IRQ enable 0: disable 1: enable	
Peak_ovfl_det_r_irq_en	2	R/W	0	Right channel peak overflow detected IRQ enable 0: disable 1: enable	
Peak_ovfl_det_l	1	R	0	Left channel peak overflow detected	Wclr_out
Peak_ovfl_det_r	0	R	0	Right channel peak overflow detected	Wclr_out

Register:: PEAK_CTRL_1				**0xD4**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Debounce_cnt_l[7:5]	7:5	R/W	0x1	The de-bouncing threshold to judge if peak volume exceeds the threshold debounce_cnt 000: 2^0 001: 2^1 010: 2^2 ……. 111: 2^7	
Rev_peak_d4_4	4	R/W	0	Reserved	
Debounce_cnt_r[3:1]	3:1	R/W	0x1	The de-bouncing threshold to judge if peak volume exceeds the threshold debounce_cnt 000: 2^0 001: 2^1 010: 2^2 ……. 111: 2^7	
Rev_peak_d4_0	0	R/W	0	Reserved	

Register:: PEAK_CTRL_L_0				**0xD5**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**

Ovf_th_l[23:16]	7:0	R/W	0x20	the overflow threshold of left channel[23:16]	

Register:: PEAK_CTRL_L_1				0xD6	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Ovf_th_l[15:8]	7:0	R/W	0	the overflow threshold of left channel[15:8]	

Register:: PEAK_CTRL_L_2				0xD7	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Ovf_th_l[7:0]	7:0	R/W	0	the overflow threshold of left channel[7:0]	

Register:: PEAK_CTRL_L_3				0xD8	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Monitor_window_l[15:8]	7:0	R/W	0x1	Monitor period of left channel for peak detection [15:8] Unit: ~~1/512~~fs	

Register:: PEAK_CTRL_L_4				0xD9	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Monitor_window_l[7:0]	7:0	R/W	0x80	Monitor period of left channel for peak detection [7:0] Unit: ~~1/512~~fs	

Register:: PEAK_CTRL_R_0				0xDA	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Ovf_th_r[23:16]	7:0	R/W	0x20	the overflow threshold of right channel[23:16]	

Register:: PEAK_CTRL_R_1				0xDB	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Ovf_th_r[15:8]	7:0	R/W	0	the overflow threshold of right channel[15:8]	

Register:: PEAK_CTRL_R_2				0xDC	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Ovf_th_r[7:0]	7:0	R/W	0	the overflow threshold of right channel[7:0]	

Register:: PEAK_CTRL_R_3				0xDD	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Monitor_window_r[15:8]	7:0	R/W	0x1	Monitor period of right channel for peak detection [15:8] Unit: ~~1/512~~fs	

Register:: PEAK_CTRL_R_4				0xDE	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Monitor_window_r[7:0]	7:0	R/W	0x80	Monitor period of right channel for peak detection [7:0] Unit: ~~1/512~~fs	

I2S OUT

Register:: I2S_OUT_0				0xE0	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
reserved	7:0	--	--	reserved	

SPDIF OUT

Register:: SPDIF_OUT_0				0xF0	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
reserved	7:3	--	--	reserved	
spd_output_sel	2:1	R/W	0	00: normal i2s_out (pcm from FIFO) 01: from internal HDMI others: disable i2s_out	
validity_ctrl	0	R/W	0	0: valid (v_bit = 0) 1: invalid (v_bit = 1)	

.

Register:: SPDIF_CS_0	0xF1

Name	Bits	R/W	Default	Comments	Config
cs_db_en	7	R/W	0	Channel Status Double buffer (used in F1-F4) Write 1 to clear	Wclr_out
reserved	6	--	--	reserved	
clock_accuracy	5:4	R/W	0	clock accuracy ref to spdif spec	
sampling_freq	3:0	R/W	0	0000: 44.1k 0010: 48k 0011: 32k 0100: 22.05k 0110: 24k 1000: 88.2k 1010: 96k 1100: 176.4k 1110: 192k others: reserved for future application	

Register:: SPDIF_CS_1				0xF2	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
channel_number	7:4	R/W	0	channel number, ref to spdif spec	
source_number	3:0	R/W	0	source number, ref to spdif spec	

Register:: SPDIF_CS_3				0xF3	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
category_code	7:0	R/W	0	The indication of the kind of equipment	

Register:: SPDIF_CS_4				0xF4	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
mode	7:6	R/W	0	channel status formats	
pre_emphasis	5:3	R/W	0	000: 2 audio channels without pre-emphasis 001: 2 audio channels with 50us/15us pre-emphasis others: reserved for future application	
copyright	2	R/W	0	0: copyright is asserted 1: no copyright is asserted	
data_type	1	R/W	0	0: linear pcm data 1: non-pcm data	
professional	0	R/W	0	0: consumer use 1: professional use	

TEST FUNCTION

Register:: TEST_FUNCTION				0xFA	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
reserved	7:0	--	--	reserved	

Register:: TEST_OUT_FUNCTION				0xFB	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Hard_gain	7:4	R/W	0	0000: 0dB 0001: 6dB 0010: 12dB 0011: 18dB 0100: 24dB 0101: -6dB 0110: -12dB 0111: -18dB 1000: -24dB	
test_out_en	3	R/W	0	0:disable 1:enable	
test_out_sel	2:0	R/W	0	000:i2s_2_pcm 001:gcounter 010:avc	

				011:eq1 100:eq2 101:i2s_tx 110:spd_tx 111:avc_eq_out	

Reserved Register

Register::reserve1				0xFC	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
valid_source	7	R/W	0	0: use valid bit from register setting F0 1: use valid bit from hdmi	
Reserve1	6:0	R/W	0	Reserved	

Register::reserve2				0xFD	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserve2	7:0	R/W	FF	Reserved2	

Register::reserve3				0xFE	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserve3	7:0	R/W	0	Reserved3	

Register::reserve4				0xFF	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserve4	7:0	R/W	FF	Reserved4	

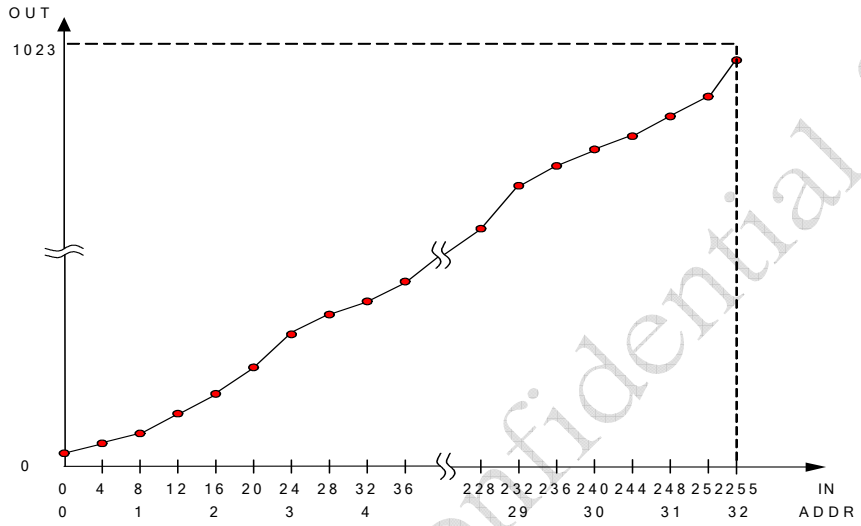
Input Gamma Control (Page 9)

Address: A0 **INPUT_GAMMA_PORT**

Bit	Mode	Function
7:0	R/W	**Access port for input gamma correction table**

- The Gamma Table written to this port should follow the sequences as expressed below:
  {2'b0, g0[9:4]}, {g0[3:0]}, 2'b0, g16[9:8]}, {g16[7:0]}, <- addr = 0
  {2'b0, g32[9:4]}, {g32[3:0]}, 2'b0, g48[9:8]}, {g48[7:0]}, <- addr = 1
  …,
  {2'b0, g992[9:4]}, {g992[3:0]}, 2'b0, g1008[9:8]}, {g1008[7:0]}, <- addr = 31
  {2'b0, g1023[9:4]}, {g1023[3:0]}, 4'b0}, {8'b0} <- addr = 32

- There are two thresholds divide Input Gamma Table into three divisions.
  - From 0 to threshold 1 is the first division.
  - From (threshold 1 + 1) to threshold 2 is the second division.
    Above threshold 2 is the third division



Address: A1 **INPUT_GAMMA_CTRL** **Default: 00h**

Bit	Mode	Function
7	R/W	**Enable Access Channels for Input Gamma Correction Coefficient:** 0: disable these channels (Default) 1: enable these channels
6	R/W	**Input Gamma table enable** 0: by pass (Default) 1: enable
5:0	--	**Reserved to 0**

Address: A2 **INPUT_GAMMA_LOW_THRE** **Default: 11h**

Bit	Mode	Function
7	R/W	Reserved to 0
6:0	R/W	Input Gamma low threshold value

Both threshold values are restricted in 0~64

Address: A3 **INPUT_GAMMA_HIGH_THRE** **Default: 22h**

Bit	Mode	Function
7	R/W	Reserved to 0
6:0	R/W	Input Gamma high threshold value

High threshold value has to be greater than low threshold value.

Both threshold values are restricted in 0~64

GDI PHY(Page B)

PHY CONTROL

Register::SIG_DET			0XA0		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:4	R/W	0	RESERVED	
REXT2VDD	3	R/W	1	1: bandgap external resistor 6.2K connect to VDD 0: connect to GND	
REG_ITUNE_LVDS[1:0]	2:1	R/W	01	tuning ibhx20u to LVDS 00:15u 01:20u 10:25u 11:30u Active when LVDS bias current select to GDI.	
en_sigdet	0	R/W	0	1: Enable signal detection 0: Disable signal detection	

Register:: FLD_FSM_0			0XA1		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:6	R/W	0	Reserved	
fld_st_lane1	5:3	R	0	lane1 FLD state	
fld_st_lane0	2:0	R	0	lane0 FLD state	

Register:: FLD_FSM_1			0xA2		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:6	R/W	0	Reserved	
fld_st_lane3	5:3	R	0	lane3 FLD state	
fld_st_lane2	2:0	R	0	lane2 FLD state	

Register:: Reserved_1			0xA3		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	R/W	0	Reserved	

Register::CDR_2			0XA4		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
lpf_cp1p_sel	7	R/W	0	select CDR lpf cp 1pF	
lpf_cs60p_sel	6	R/W	0	select CDR lpf cs 46pF	
lpf_cp05p_sel	5	R/W	0	select CDR lpf cp 0.5pF	
rev_reg[4:0]	4:0	R/W	0		

Register::CDR_3			0XA5		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
vco_div	7:6	R/W	0	VCO Divider Mcode: 00=VCO/1, 01=VCO/2, 10=VCO/4, 11=VCO/8	
sel_bb	5	R/W	1	0=linear PD 1=Binary PD	
sel_half	4	R/W	1	0=Full rate PD 1=Half rate PD ,	
sel_vcmfb	3	R/W	0	Charge pump common mode voltage Setting 0: set charge pump common mode voltage to CPVREF, i.e. 0.7/0.65/0.6/0.55V 1: set charge pump common mode voltage to Vdd/2	
vco_vc_manual	2	R/W	0	0: VCO in CDR loop 1:force VCO control voltage to +/-100mV by vco_vc if cp_en_manual =1'b0	
kvco	1:0	R/W	1	Kvco boost: 00=388MHz/V, 01=526MHz/V, 10=640MHz/V,11=832MHz/V	

Register::CDR_4			0xA6		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
cp_sel	7:4	R/W	1	Charge pump current 0000: 3.75uA 0001: 7.5uA 0010: 11.25uA 0011: 15uA 0100: 7.5uA 0101: 15uA 0110: 22.5uA 0111: 30uA 1000:11.25uA 1001: 22.5uA 1010:33.75uA 1011: 45uA 1100: 15uA 1101: 30uA 1110: 45uA 1111: 60uA	
cpsr	3:1	R/W	1	Loop resister: 100=0.5k 101=1k 000=2k 110=3k 001=4k 010=8k 011=10k 111=12k	
Reserved	0	R/W	0	Reserved	

A5[7:4], A5[1:0],A6, AA[0] enable when AF[6]=1

Register:: CDR_5			0XA7		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
dvco_div	7:6	R	0	DVCO_DIV<1:0>	
dsel_bb	5	R	0	DSEL_BB	
dsel_half	4	R	0	DSEL_HALF	
Reserved	3:1	R/W	0	Reserved	
en_vco_biasr	0	R/W	0	0: VCO bias large R controlled by FLD 1: VCO bias using large RC(better phase noise).	

Register::CDR_6			0xA8		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
data_rdy	7:4	R	0	CDR[3:0] Data Ready of Each Lane 0: unlock, 1:lock	
Reserved	3:2	R/W	0	RESERVED	
Reserved	1:0	R	0	Reserved	

Register::CDR_7			0XA9		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
bg_psm	7	R/W	0	Bandgap power saving mode	
loop_ok_in_regb	6	R/W	0	1'b0:loop_ok=1 when reg_loop_f_sel=1	
reg_loop_f_sel	5	R/W	0	1'b0:select as RL6049E mode det output	

vr_reg[3]	4	R/W	0	sel_cktp1 1'b0=select from FSM(default), 1'b1=select from en_cktp1	
vr_reg[2:1]	3:2	R/W	3	reserved register	
vr_reg[0]	1	R/W	1	1'b0=disable tp1 input 1'b1=enable tp1 input (default),	
reg_xtal_sel	0	R/W	1	1'b1=FLD xtal divider select /8	

Register:: CDR_8			**0XAA**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
bpdgain	7	R/W	1	BPD gain	
sel_cont3	6	R/W	1	1=binary PD UP/DN pulse div3 0=binary PD UP/DN pulse div1	
sel_ibx	5	R/W	0	VCO bias current 0: ibn, 1: ibx	
sel_ib90	4:3	R/W	1	90u bias current: 00=80u, 01=90u, 10=100u, 11=110u	
sel_ib25	2:1	R/W	2	25u bias current: 00=15u, 01=20u, 10=25u, 11=30u.	
cpsc	0	R/W	0	Loop capacitor: 0=2p disable, 1=2p enable	

Register:: CDR_9			**0XAB**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7	R/W	0	Reserved	
xtal_en	6	R/W	1	1=crystal input 0=non crystal, use trimmed clock	
rxcom_sel	5:4	R/W	1	rx amplifier input common mode voltage: 2'b00=0.9V, 2'b01=0.95V(default), 2'b10=1V, 2'b11=1.05V	
rstb_z0	3	R/W	1	Reset 50 Ohm calibration	
z0_manual	2	R/W	0	1: Manual select Z0 calibration reset signal, ,from A4[3]rstb_z0 0:reset signal from rxen	
en_wide_temp	1	R/W	0	en_wide_temp: FLD 0 : disable vco_vc control . Vc=0 (default) 1 : enable vco vc control. (in bit 0)	
vco_vc	0	R/W	0	vco control voltage when vco_vc_manual=1'b1 0=-0.1V,1=0.1V	

Register::CDR_A			**0XAC**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:5	R/W	0	Reseved	
dp_mode	4	R/W	0	0=hmdi mode 1=Reserved	
Reserved	3:0	R/W	0	Reserved	

Register::RXMISC_0			**0XAD**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
cdrcpop_en	7	R/W	0	1=enable CDR charge pump charge sharing OP	
eqcpop_en	6	R/W	0	1=enable EQ charge pump charge sharing OP	
sel_csop_in	5	R/W	1	0=CDR CP CSOP input from Rs 1=CDR CP CSOP input from Cs	
cp_adj_en	4	R/W	1	enable adjust charge pump current when tracking	

adp_eq_off	3	R/W	1	1:Passive 0:Adaptive Active when auto_mode 0xAF[7] = 1	
adp_time	2:0	R/W	7	adaptive equalizer turn on time delay	

Register:: RXMISC_1			**0XAE**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
cpvref_sel	7:6	R/W	0	CDR CP Vref: 00: 0.55V, 01: 0.60V, 10: 0.65V, 11: 0.50V	
eqvref_sel	5:4	R/W	0	EQ CP Vref: 00: 0.55V, 01: 0.60V, 10: 0.65V, 11: 0.50V	
eqvc_selini	3	R/W	1	EQ VC Initial Value 1: Boost higher , Vcp-Vcn=0.2V 0: Boost lower , Vcp=Vcn	
Reserved	2:1	R/W	0	RESERVED	
sel_sync	0	R/W	1	Div5 & div2p5 out phase: 0: clock edge at middle point of data, 1:clock edge align with data edge	

Register::RXMISC_2			**0XAF**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
auto_mode	7	R/W	1	FLD auto mode: 0=manual, 1=auto set by FLD,	
loop_manual	6	R/W	0	Parameter manual set: 0=auto set by mode detect, 1=manual set (divider,PD)	
cp_en_manual	5	R/W	0	Enable only at FLD manual mode CP enable when auto_mode=0 0=by auto, 1=manual enable CP	
band_rst	4	R/W	0	VCO band reset: 0=not reset when FLD detect unlock, 1=reset when FLD detect lock	
calib_manual	3	R/W	0	data_rdy <= calib_manual when auto_mode=0	
calib_time	2:0	R/W	7	date_rdy output delay time	

Register:: EQ_0			**0xB0**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	R/W	0	Reserved	
eq_autoselrange	0	R/W	0	1: Enable auto range selection (Adaptive) 0: Disable	

Register:: EQ_1			**0xB1**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
en_boo1	7	R/W	1	Enable Boost Stage1	
en_boo2	6	R/W	1	Enable Boost Stage2	
en_boo3	5	R/W	1	Enable Boost Stage3	
en_boo4	4	R/W	1	Enable Boost Stage4	

en_adap_eq	3	R/W	0	1: Auto Set Booster (Adaptive) 0: Manual Set Booster3 ,4 (Passive)	
en_single_vc	2	R/W	0	0: equalizer Vc is differetial mode 1: equalizer Vc is single-ended mode。	
eq_en_buf	1:0	R/W	0	Bit1 : Control Booster 4 Bit 0 :Control Booster 3 0: DC gain =0 1: Add DC gain	

Register:: EQ_2			**0XB2**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
adp_en_manual	7	R/W	0	Adaptive Equalizer hand mode enable: 0: disable 1: enable	
koffset_en	6	R/W	1	Adaptive Equalizer offset calibration enable: 0: disable 1: enable	
offset_autok	5	R/W	1	Adaptive Equalzier offset calibration using auto mode: 0: disable 1: enable	
Reserved	4:3	R/W	0	Reserved	
eqsc	2	R/W	1	Adaptive Equalizer Loop filter C: 0: 5p, 1: 10p	
eqcp_sel	1:0	R/W	2	Adaptive Equalizer Loop Charge Pump current: 00: 2.5u, 01: 7.5u, 10:5u(default), 11: 15u	

Register:: Reserved_2			**0xB3**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	R/W	0	Reserved	

Register:: Power_0			**0XB4**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
cmu_en	7:4	R/W	0	CMU_EN[7:0], power control for VCO of lane3..0 0:disable 1: enable	
rx_en	3:0	R/W	0	RX_EN[3:0], power control for Rx channel 3..0, 0:disable 1: enable	

Register:: 2D_switch			**0XB5**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
reserved	7:6	R/W	0	RESERVED	
freqadd	5	R/W	0	1=VCO frequency shift up	
freqsub	4	R/W	0	1=VCO frequency shift down	
reserved	3	R/W	0	reserved	
din_sel	2	R/W	0	0=port 0; 1=port 1	
z0_autok	1	R/W	1	50 Ohm calibration 0: Register, 1: Auto-Calibration	
rstb_fsm	0	R/W	1	reset mode_det and FLD 0=reset,1=power on	

Register:: Z0_0			**0XB6**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
p0_z0_en	7:4	R/W	0	0 = disable 50 Ohm on of port0, 1 = Z0 turn ON	
z0_adjr	3:0	R/W	8	Manual Control for 50 ohm resistor (both port 0 & port 1)	

Register:: Z0_1			0XB7		
Name	Bits	R/W	Default	Comments	Config
p1_z0_en	7:4	R/W	0	0 = disable 50 Ohm on of port1, 1 = Z0 turn ON	
Reserved	3:0	R/W	0	Reserved	

Register:: Z0_2			0XB8		
Name	Bits	R/W	Default	Comments	Config
Reserved	7:5	R/W	0	Reserved	
z0_tst	4:0	R	0	50 Ohm termination debug output	

Register:: Reserved_3			0xB9		
Name	Bits	R/W	Default	Comments	Config
Reserved	7:0	R/W	0	Reserved	

Register::EQ_3			0XBA		
Name	Bits	R/W	Default	Comments	Config
offset_tst_lane3[4]	7	R	0	1: Calibration finished. Lane3	
offset_tst_lane2[4]	6	R	0	1: Calibration finished. Lane2	
offset_tst_lane1[4]	5	R	0	1: Calibration finished. Lane1	
offset_tst_lane0[4]	4	R	0	1: Calibration finished. Lane0	
offset_idiv	3:2	R/W	1	Offset calibration current range: 2'b00=+/-22.5u, 2'b01=+/-45u, (default) 2'b10=+/-45u,2'b11=+/-90u,	
bypass_ok	1	R/W	0	bypass calibration ok signal 1'b1=bypass,1'b0=auto use calibration ok (default)	
en_sel_tst	0	R/W	0	0: offset_tst monitor input offset results 1:offset_tst monitor adaptive equalzier status	

Register:: BandGap			0XBB		
Name	Bits	R/W	Default	Comments	Config
bg_en	7	R/W	1	BG_EN	
bg_db	6:4	R/W	4	3'b000=1.155V, 3'b001=1.167V, 3'b010=1.178V 3'b011=1.19V, 3'b100=1.202V, 3'b101=1.213V 3'b110=1.225V, 3'b111=1.237V,	
reserved	3	R/W	0	reserved	
psm	2	R/W	0	1=CDR in power saving mode,only output ref clock	
reserved	1:0	R/W	0	reserved	

lane 01 Calibration value

Register:: EQ_4			0xBC		
Name	Bits	R/W	Default	Comments	Config
offset_tst_lane0[3:0]	7:4	R	0	Calibration offset value of lane 0	
offset_tst_lane1[3:0]	3:0	R	0	Calibration offset value of lane 1	

lane 23 Calibration value

Register:: EQ_5			0xBD		
Name	Bits	R/W	Default	Comments	Config
offset_tst_lane2[3:0]	7:4	R	0	Calibration offset value of lane 2	

offset_tst_lane3[3:0]	3:0	R	0	Calibration offset value of lane 3	

Register:: EQ_6			0xBE		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
offset_adjr0	7:4	R/W	8	OFFSET_ADJR<3:0> of lane0	
offset_adjr1	3:0	R/W	8	OFFSET_ADJR<3:0> of lane1	

Register:: EQ_7			0xBF		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
offset_adjr2	7:4	R/W	8	OFFSET_ADJR<3:0> of lane2	
offset_adjr3	3:0	R/W	8	OFFSET_ADJR<3:0> of lane3	

0xC0~0xC7 HDMI counter For Mode Detect (Useless)

Register::TMDS_CONT_0			0XC0		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
r_cont_hi[8]	7	R/W	1		
r_cont_mhi[8]	6	R/W	1		
r_cont_mid[8]	5	R/W	0		
r_cont_mlo[8]	4	R/W	0		
r_cont_lo[8]	3	R/W	0		
reserved	2:0	R/W	0		

Register:: TMDS_CONT_1			0XC1		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
r_cont_hi[7:0]	7:0	R/W	D9		

Register:: TMDS_CONT_2			0XC2		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
r_cont_mhi[7:0]	7:0	R/W	06		

Register:: TMDS_CONT_3			0XC3		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
r_cont_mid[7:0]	7:0	R/W	83		

Register:: TMDS_CONT_4			0XC4		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
r_cont_mlo[7:0]	7:0	R/W	42		

Register:: TMDS_CONT_5			0XC5		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
r_cont_lo[7:0]	7:0	R/W	19		

Register:: TMDS_CONT_6			0XC6		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7	R/W	0	Reserved	
r_cont_divider	6:0	R/W	20	Refer to PPT	

Register:: TMDS_CONT_7			0XC7		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:3	R/W	0	Reserved	
r_cont_timer	2:0	R/W	2	Refer to PPT	

FLDcounter →

$$\frac{Ck\_ref}{Fld\_ref\_cont} = \frac{Ck\_sys}{Fld\_cdr\_cont}$$

Fld_cdr_cont →

1. HDMI

	Ck_ref	24.3MHz	Bin	Hex
HDMI	Fld_ref_cont0	320	1 01000000	1 40
	Fld_cdr_cont0	320	1 01000000	1 40

Register:: FLD_CONT_0			0XC8		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
fld_cdr_cont0[8]	7	R/W	1		
fld_ref_cont0[8]	6	R/W	1		
reserved	5	R/W	0		
fld_cdr_delta	4:0	R/W	5		

Register:: FLD_CONT_1			0XC9		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
fld_cdr_cont0[7:0]	7:0	R/W	40		

gister:: FLD_CONT_2			0XCA		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
fld_ref_cont0[7:0]	7:0	R/W	40		

Register:: FLD_CONT_3			0XCB		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
fld_cdr_cont1[8]	7	R/W	1		
fld_ref_cont1[8]	6	R/W	0		
fld_cdr_cont2[8]	5	R/W	1		
fld_ref_cont2[8]	4	R/W	0		

Reserved	3:0	R/W	0	Reserved	

Register:: FLD_CONT_4			0XCC		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
fld_cdr_cont1[7:0]	7:0	R/W	2D		

Register:: FLD_CONT_5			0XCD		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
fld_ref_cont1[7:0]	7:0	R/W	1A		

Register:: FLD_CONT_6			0XCE		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
fld_cdr_cont2[7:0]	7:0	R/W	2D		

Register:: FLD_CONT_7			0XCF		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
fld_ref_cont2[7:0]	7:0	R/W	1A		

Register:: RXMISC_3			0XD0		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
LANE_MUX_SEL	7:0	R/W	E4	select the data mapping between physical lane and MAC lane [7:6]: lane3 data comes from physical lane # 2'b00: physical lane0 2'b01: physical lane1 2'b10: physical lane2 2'b11: physical lane3 (default) [5:4]: lane2 data comes from physical lane # 2'b00: physical lane0 2'b01: physical lane1 2'b10: physical lane2 (default) 2'b11: physical lane3 [3:2]: lane1 data comes from physical lane # 2'b00: physical lane0 2'b01: physical lane1 (default) 2'b10: physical lane2 2'b11: physical lane3 [1:0]: lane0 data comes from physical lane # 2'b00: physical lane0 (default) 2'b01: physical lane1 2'b10: physical lane2 2'b11: physical lane3	

Register:: RXMISC_4			0XD1		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
SAMPLE	7:4	R/W	0	SAMPLE edge selector of each lane 0:postive clock edge 1:negative clock edge	
REG_PN_SWAP	3:0	R/W	0	REG_PN_SWAP, swap pn to avoid pn mismatch (physical lane)	

Register:: Reserved_4			0xD2		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	R/W	0	Reserved	

Register:: Reserved_5			0xD3		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**

Reserved	7:0	R/W	0	Reserved	

Register:: Reserved_6			0xD4		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	R/W	0	Reserved	

Register:: Reserved_7			0xD5		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	R/W	0	Reserved	

Register:: Reserved_8			0xD6		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	R/W	0	Reserved	

Register:: Reserved_9			0xD7		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	R/W	0	Reserved	

Register:: VCO_0			0XD8		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
reserved	7	R/W	0	Reserved	
dfvco_lane0	6:0	R	0	lane0 VCO auto calibration result	

Register:: VCO_1			0XD9		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
reserved	7	R/W	0	Reserved	
dfvco_lane1	6:0	R	0	lane1 VCO auto calibration result	

Register:: VCO_2			0XDA		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
reserved	7	R/W	0	Reserved	
dfvco_lane2	6:0	R	0	lane2 VCO auto calibration result	

Register:: VCO_3			0XDB		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
reserved	7	R/W	0	Reserved	
dfvco_lane3	6:0	R	0	lane3 VCO auto calibration result	

Register::VCO_4			0xDC		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7	R/W	0	Reserved	
rfvco_lane0	6:0	R/W	3F	Manual set Lane0 vco band selection=0~127	

Register:: VCO_5			0XDD		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
reserved	7	R/W	0	Reserved	
rfvco_lane1	6:0	R/W	3F	Manual set Lane1vco band selection=0~127	

Register::VCO_6			0xDE		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7	R/W	0		

rfvco_lane2	6:0	R/W	3F	Manual set Lane2 vco band selection=0~127	

Register:: VCO_7			0XDF		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
reserved	7	R/W	0		
rfvco_lane3	6:0	R/W	3F	Manual set Lane3 vco band selection=0~127	

Register:: DIG_00			0XE0		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
CTL_SEL	7	R/W	1	CTL_SEL, select primary control signal 0:from primary input 1:from SI	
SI_X1X2_SEL	6	R/W	1	SI_X1X2_SEL, interface clock rate selector 0:16 bits 1:8 bits	
SI_ENCH_MODE	5	R/W	0	SI_ENCH_MODE, enhanced control mode enable	
SI_HD_SEL	4	R/W	1	SI_HD _SEL, HDMI selector 0: Reserved, 1: HDMI	
HDMI_X4_SEL	3:0	R/W	0	HDMI_X4_SEL HDMI clock divided by 4 mode	

Register:: DIG_01			0XE1		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
CLK_INV	7:4	R/W	0	_CLK_INV, inverse clk clock	
CLKX2_INV	3:0	R/W	0	CLKX2_INV, inverse clkx2 clock	

Register:: DIG_02			0XE2		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
SCRAMB_DIS	7	R/W	0	SCRAMB_DIS, disable de-scrambling	
COMMA_DIS	6	R/W	0	COMMA_DIS, disable comma detection	
COMSEL	5:3	R/W	0	COMSEL, comma number selector 000: 3 comma 001: 4 comma 010: 6 comma 011: 8 comma 100: 10 comma 101: 12 comma 110: 14 comma 111: 16 comma	
PRBS_REVERSE	2	R/W	0	PRBS_REVERSE, reverse PRBS7 pattern generator	
Reserve_dig_00	1	R/W	0	1: 4 symbol match {SR,BF/3C,BF/3C,SR} →scrambler reset 0 : 3 symbol match → scrambler reset	
RXBIST_CLK	0	R/W	0	RXBIST CLK selector 0: from analog 1:form TX clock	

Register:: DIG_03			0XE3		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ERROR_COUNT_CLEAR	7	R/W	0	Error count clear	
RXBIST_SOURCE	6:5	R/W	0	RXBIST_SOURCE, rx bist source selection 00: from PMA(normal function) 01: from comdet input 10: from 10b8b input 11: from de-scrambling input	

RXBIST_DEST	4:3	R/W	0	RXBIST_DEST, rx bist destination selection 00: disable rxbist 01: to comdet output 10: to 10b8b output 11: to de-scrambling output	
PRBS_SEL	2	R/W	0	PRBS_SEL, prbs pattern selection 0: prbs 7 1: prbs31	
Reserved	1:0	R/W	0	Reserved	

Register:: DIG_04			**0xE4**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BIST_SEED	7:0	R/W	0	BIST_SEED, bist seed pattern	

Register:: DIG_05			**0xE5**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
RX_DEBUG	7:0	R/W	0	RX digital debug	

Register:: DIG_06			**0xE6**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BYTE_COUNT_MSB	7:0	R	0	BYTE_COUNT[15:8], received packet number	

Register:: DIG_07			**0xE7**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BYTE_COUNT_LSB	7:0	R	0	BYTE_COUNT[7:0], received packet number	

Register:: DIG_08			**0xE8**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
RXBIST_ERR_CNT	7:0	R	0	RXBIST_ERR_CNT, received error counter	

0xE9~0xEF for HDMI Tx Test Chip

Register:: SSCG_DIG			**0XE9**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
SSC_EN	7	R/W	0	Enable SSCG function	
ORDER	6	R/W	0	ORDER, 0:2-nd order, 1:1-st order	
Reverse_sscg_dig	5	R/W	0	Reserved	
WAVE_SEL	4:3	R/W	0	WAVE_SEL 00: -0.6 01: -0.5 10: -0.4 11: -0.3	
DITHER_SEL	2	R/W	0	DITHER selection0 00: disabled 01:2^-5 10:2^-4 11:2^-3	
SI_ANA_EN	1	R/W	0	ENABLE SI_ANA function	
TEST_CHIP	0	R/W	0	TEST CHIP TX/RX selector 0: RX selected 1: TX selected	

Register:: DIG_TX_00			**0XEA**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
TXBIST_MODE	7:6	R/W	0	Tx bist mode selector: 00: normal 01: 8bit mode	

				10: 10 bit mode	
TX_PRBS_SEL	5	R/W	0	TX prbs_sel	
Reserved	4:0	R/W	0	RESERVED	

Register:: DIG_TX_01				**0xEB**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PACKET_LENGTH	7:0	R/W	0	PACKET_LENGTH, bist packet length	

Register:: DIG_TX_02				**0xEC**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PACKET_NUM	7:0	R/W	0	PACKET_NUMBER, bist packet number	

Register:: DIG_TX_03				**0xED**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ERR_PERIOD	7:0	R/W	0	ERR_PERIOD, force error period	

Register:: Reserved_A				**0xEE**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	R/W	0	Reserved	

Register::DIG_TX_04				**0xEF**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
P0_IE	7:0	R/W	FF	P0_IE[7:0]: GPI pad input enable	

TMDS MAC (Page C)

GLOBAL CONTROL

Register:: DP_CTRL			0xA1		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reservd	7:6	R/W	0	Reserved	
HD_SEL	5	R/W	1	HDMI Seletor 0: Reserved 1: HDMI	
Reserved	4:0	R/W	0	Reserved	

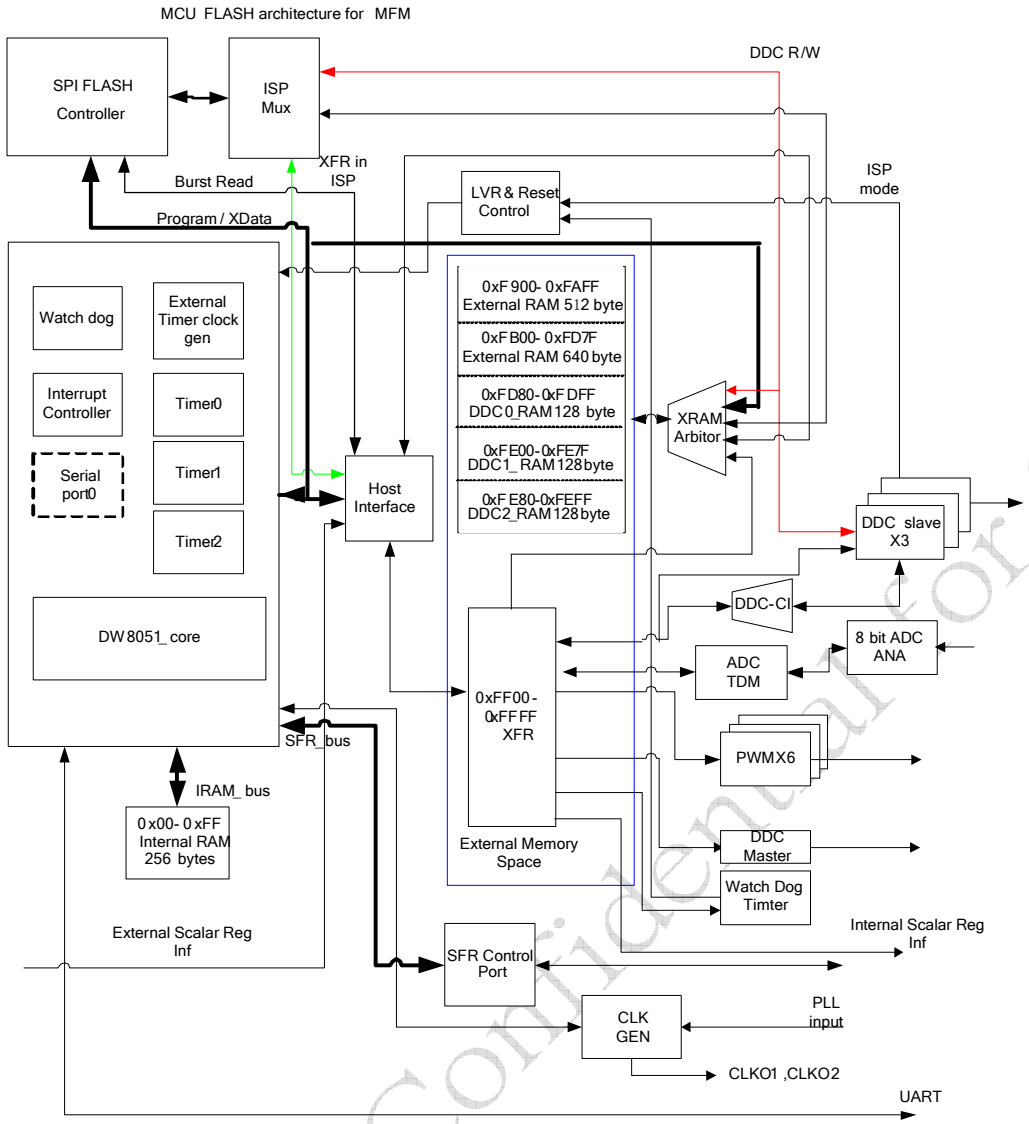
MCU register 1(page D)

Embedded MCU Function

Designware DW8051 of Synopsys is integrated into this chip and is compatible with other industry 8051 series. A lot of peripherals are integrated and accessed by XFR (eXternal Function Register). When embedded MCU is used, scalar-related registers are access via XFR, too. The program code is stored in external serial FLASH. If the external MCU is used, the integrated peripherals can be accessed via special page in scalar's register map.

Features

- 8051 core, CPU operating frequency up to 50MHz
- 256-byte IRAM, 1.5K byte shared XRAM
- external serial FLASH, support GPIO bank switching for 128K byte and up to 16M bytes for XFR bank switching
- Compliant with VESA DDC1/2B/2Bi/CI
- Embedded triple ports DDC RAM(0~768 bytes shared with XRAM)
- Six channels of PWM DAC
- Watchdog timer with programmable interval
- Three 16-bit counters/timers (T0, T1, and ET2)
- Programmable frequency clock output, 2 clock output ports
- One full-duplex serial port
- Six interrupt sources with 2 external interrupts
- Four channels of 8-bit ADC
- Hardware ISP, no boot code required
- Built-in Low voltage reset circuit



DW8051 micro-processor

The DW8051 is compatible with industry standard 803x/805x and provides the following design features and enhancements to the standard 8051 microcontroller:

High speed architecture

Compared to standard 8051, the DW8051 processor core provides increased performance by executing instructions in a 4-clock bus cycle, as opposed to the 12-clock bus cycle in the standard 8051. The shortened bus timing improves the instruction execution rate for most instructions by a factor of three over the standard 8051 architectures. The average speed improvement for the entire instruction set is approximately 2.5X.

Stretch Memory Cycles

The stretch memory cycle feature enables application software to adjust the speed of data memory access. The DW8051 can execute the MOVX instruction in as little as 2 instruction cycles. However, it is

sometimes desirable to stretch this value; for example, to access slow memory or slow memory-mapped peripherals such as UARTs or LCDs.

The three LSBs of the Clock Control Register (at SFR location 8Eh) control the stretch value. You can use stretch values between zero and seven. A stretch value of zero adds zero instruction cycles, resulting in MOVX instructions executing in two instruction cycles. A stretch value of seven adds seven instruction cycles, resulting in MOVX instructions executing in nine instruction cycles. The stretch value can be changed dynamically under program control.

By default, the stretch value resets to one (three cycle MOVX). For full-speed data memory access, the software must set the stretch value to zero. The stretch value affects only data memory access. The only way to reduce the speed of program memory (ROM) access is to use a slower clock.

Dual Data Pointers

The DW8051 employs dual data pointers to accelerate data memory block moves. The standard 8051 data pointer (DPTR) is a 16-bit value used to address external data RAM or peripherals. The DW8051 maintains the standard data pointer as DPTR0 at SFR locations 82h and 83h. It is not necessary to modify code to use DPTR0.

The DW8051 adds a second data pointer (DPTR1) at SFR locations 84h and 85h. The SEL bit in the DPTR Select register, DPS (SFR 86h), selects the active pointer. When SEL = 0, instructions that use the DPTR will use DPL0 and DPH0. When SEL = 1, instructions that use the DPTR will use DPL1 and DPH1. SEL is the bit 0 of SFR location 86h. No other bits of SFR location 86h are used.

All DPTR-related instructions use the currently selected data pointer. To switch the active pointer, toggle the SEL bit. The fastest way to do so is to use the increment instruction (INC DPS). This requires only one instruction to switch from a source address to a destination address, saving application code from having to save source and destination addresses when doing a block move.

Using dual data pointers provides significantly increased efficiency
when moving large blocks of data.

Timer Rate Control

One important difference exists between the DW8051 and 80C32 regarding timers. The original 80C32 used a 12 clock per cycle scheme for timers and consequently for some serial baud rates(depending on the mode). The DW8051 architecture normally runs using 4 clocks per cycle. However, in the area of timers, it will default to a 12 clock per cycle scheme on a reset. This allows existing code with real–time dependencies such as baud rates to operate properly. If an application needs higher speed timers or serial baud rates, the timers can be set to run at the 4 clock rate.
The Clock Control register (CKCON – 8Eh) determines these timer speeds. When the relevant CKCON bit is a logic 1, the device uses 4 clocks per cycle to generate timer speeds. When the control bit is set to a zero, the device uses 12 clocks for timer speeds. The reset condition is a 0. CKCON.5 selects the speed of Timer 2. CKCON.4 selects Timer 1 and CKCON.3 selects Timer zero. Note that unless a user desires very fast timing, it is unnecessary to alter these bits. Note that the timer controls are independent.

RESET

There are five reset sources.

- RST pin
  The external reset is high active and its pulse width must be larger than 8 clock cycles. The RST pin can reset the DW8051
- Low voltage reset(LVR) and power on reset(POR)
- Software can use SOF_RST register to reset whole chip
- Watchdog can reset DW8051
- When entering ISP mode, DW8051 will in reset state. When exiting ISP mode, DW8051 will also asserting a reset, too.

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Addr
SP									81h
DPL0									82h
DPH0									83h
DPL1									84h
DPH1									85h
DPS	0	0	0	0	0	0	0	SEL	86h
PCON	SMOD 0		1	1	GF1	GF0	STOP	IDLE	87h
	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	88h
TMOD	GATE	C/T	M1	M0	GATE	C/T	M1	M0	89h
TL0									8Ah
TL1									8Bh
TH0									8Ch
TH1									8Dh
CKCON			T2M	T1M	T0M	MD2	MD1	MD0	8Eh
SPC_FNC	0	0	0	0	0	0	0	WRS	8Fh
P1_W	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0	90h
MPAGE									92h
~~P1_R~~	~~P1.7~~	~~P1.6~~	~~P1.5~~	~~P1.4~~	~~P1.3~~	~~P1.2~~	~~P1.1~~	~~P1.0~~	~~93h~~
SCON0	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	98h
SBUF0									99h
P2	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0	A0h
IE	EA	0	ET2	ES0	ET1	EX1	ET0	EX0	A8h
P3_W	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0	B0h
~~P3_R~~	~~P3.7~~	~~P3.6~~	~~P3.5~~	~~P3.4~~	~~P3.3~~	~~P3.2~~	~~P3.1~~	~~P3.0~~	~~B3h~~
IP	1	0	PT2	PS0	PT1	PX1	PT0	PX0	B8h
T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/T2	CP/ RL2	C8h
RCAP2L									CAh
RCAP2H									CBh
TL2									CCh
TH2									CDh
PSW	CY	AC	F0	RS1	RS0	OV	F1	P	D0h
ACC									E0h
B									F0h

SFR reset value

Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Addr
SP	0	0	0	0	0	1	1	1	81h
DPL0	0	0	0	0	0	0	0	0	82h

DPH0	0	0	0	0	0	0	0	0	83h
DPL1	0	0	0	0	0	0	0	0	84h
DPH1	0	0	0	0	0	0	0	0	85h
DPS	0	0	0	0	0	0	0	0	86h
PCON	0	0	1	1	0	0	0	0	87h
	0	0	0	0	0	0	0	0	88h
TMOD	0	0	0	0	0	0	0	0	89h
TL0	0	0	0	0	0	0	0	0	8Ah
TL1	0	0	0	0	0	0	0	0	8Bh
TH0	0	0	0	0	0	0	0	0	8Ch
TH1	0	0	0	0	0	0	0	0	8Dh
CKCON	0	0	0	0	0	0	0	1	8Eh
SPC_FNC	0	0	0	0	0	0	0	0	8Fh
P1_W	1	1	1	1	1	1	1	1	90h
MPAGE	0	0	0	0	0	0	0	0	92h
SCON0	0	0	0	0	0	0	0	0	98h
SBUF0	0	0	0	0	0	0	0	0	99h
P2	0	0	0	0	0	0	0	0	A0h
IE	0	0	0	0	0	0	0	0	A8h
P3_W	1	1	1	1	1	1	1	1	B0h
IP	1	0	0	0	0	0	0	0	B8h
T2CON	0	0	0	0	0	0	0	0	C8h
RCAP2L	0	0	0	0	0	0	0	0	CAh
RCAP2H	0	0	0	0	0	0	0	0	CBh
TL2	0	0	0	0	0	0	0	0	CCh
TH2	0	0	0	0	0	0	0	0	CDh
PSW	0	0	0	0	0	0	0	0	D0h
ACC	0	0	0	0	0	0	0	0	E0h
B	0	0	0	0	0	0	0	0	F0h

DW8051 user-modifiable parameters

ram_256	1
timer2	1
rom_addr_size	0
Serial	1
extd_intr	0

REG_XFR_WRAPPER is the eXternal Function Register defined in 0xFF00 ~ 0xFFFF in DW8051's XDATA memory space. There are three possible input sources for REG_XFR_WRAPPER. In embedded MCU mode, DW8051 has the power of control.

	PAGE D						PAGE E						PAGE F			
	A	B	C	D	E	F	A	B	C	D	E	F	A	B	C	D
0xFFXX	0X	1X	2X	3X	4X	5X	6X	7X	8X	9X	AX	BX	CX	DX	EX	FX
X0	IRQ	ADC	ADC_DDC	DDC-CI	PWM	PWM	SPI-FLASH	SPI-FLASH	SPI-FLASH	PWM	GPIO	WD	GPIO	GPIO	GPIO	SPI_WP

X1	IRQ	ADC	ADC_DDC	DDC-CI	PWM	PWM	SPI-FLASH	SPI-FLASH	SPI-FLASH	PWM	GPIO		GPIO	GPIO	GPIO	
X2	VS_IRQ	ADC	VSYNC_Sel	DDC-CI	PWM	PWM	SPI-FLASH	SPI-FLASH	SPI-FLASH	PWM	GPIO		GPIO	GPIO	SFR	PWM_I2C
X3	OSD turbo	ADC	DDC-CI	DDC-CI	PWM	PWM	SPI-FLASH	SPI-FLASH	I2C	PWM	GPIO		GPIO	GPIO	SFR	SCA
X4	OSD turbo	ADC	DDC-CI	DDC-CI	PWM	PWM	SPI-FLASH	SPI-FLASH	I2C	PWM	GPIO		GPIO	GPIO	EDID_IRQ	SCA
X5	OSD turbo	ADC	DDC-CI	DDC-CI	PWM	I2C	SPI-FLASH	SPI-FLASH	I2C	PWM	Flash		GPIO	GPIO	EDID_IRQ	SCA
X6	OSD turbo	ADC	DDC-CI	PWM	PWM	I2C	SPI-FLASH	SPI-FLASH	I2C	PWM	Flash		GPIO	GPIO	GPIO	SCA
X7	OSD turbo	ADC	DDC-CI	PWM	PWM	I2C	SPI-FLASH	GPIO	I2C	PWM	Flash		GPIO	GPIO		SCA
X8	ADC	OSD turbo	DDC-CI	PWM	PWM	I2C	SPI-FLASH	GPIO	I2C	PWM	Flash		GPIO	GPIO		SCA
X9	ADC	ADC_DDC	DDC-CI	PWM	PWM	I2C	SPI-FLASH	GPIO	GPIO	PWM	Flash		GPIO	GPIO	Dummy	SCA
XA	ADC	ADC_DDC	DDC-CI	PWM	PWM	I2C	SPI-FLASH	GPIO	GPIO	PWM	WD		GPIO	GPIO	WD	SCA
XB	ADC	ADC_DDC	DDC-CI	PWM	PWM	I2C	SPI-FLASH	GPIO	GPIO	PWM	WD		GPIO	GPIO	WD	SCA
XC	ADC	ADC_DDC	DDC-CI	PWM	PWM	I2C	SPI-FLASH	GPIO	GPIO	PWM	WD		GPIO	GPIO	ISP	Bank-Switch
XD	ADC	ADC_DDC	DDC-CI	PWM	PWM	I2C	SPI-FLASH	GPIO	GPIO	Flash	WD		GPIO	GPIO	ISP	Bank-Switch
XE	ADC	ADC_DDC	DDC-CI	PWM	PWM	I2C	SPI-FLASH	GPIO	GPIO	Flash	WD		GPIO	GPIO	ISP	Bank-Switch
XF	ADC	ADC_DDC	DDC-CI	PWM	PWM	I2C	SPI-FLASH	GPIO	GPIO	Flash	WD		GPIO	GPIO	ISP	Bank-Switch

0xFF00	(Page D) 0xA0	0xFF10	(Page D) 0xB0	0xFF20	(Page D) 0xC0	0xFF30	(Page D) 0xD0	0xFF40	(Page D) 0xE0	0xFF50	(Page D) 0xF0
0xFF01	(Page D) 0xA1	0xFF11	(Page D) 0xB1	0xFF21	(Page D) 0xC1	0xFF31	(Page D) 0xD1	0xFF41	(Page D) 0xE1	0xFF51	(Page D) 0xF1
0xFF02	(Page D) 0xA2	0xFF12	(Page D) 0xB2	0xFF22	(Page D) 0xC2	0xFF32	(Page D) 0xD2	0xFF42	(Page D) 0xE2	0xFF52	(Page D) 0xF2
0xFF03	(Page D) 0xA3	0xFF13	(Page D) 0xB3	0xFF23	(Page D) 0xC3	0xFF33	(Page D) 0xD3	0xFF43	(Page D) 0xE3	0xFF53	(Page D) 0xF3
0xFF04	(Page D) 0xA4	0xFF14	(Page D) 0xB4	0xFF24	(Page D) 0xC4	0xFF34	(Page D) 0xD4	0xFF44	(Page D) 0xE4	0xFF54	(Page D) 0xF4
0xFF05	(Page D) 0xA5	0xFF15	(Page D) 0xB5	0xFF25	(Page D) 0xC5	0xFF35	(Page D) 0xD5	0xFF45	(Page D) 0xE5	0xFF55	(Page D) 0xF5
0xFF06	(Page D) 0xA6	0xFF16	(Page D) 0xB6	0xFF26	(Page D) 0xC6	0xFF36	(Page D) 0xD6	0xFF46	(Page D) 0xE6	0xFF56	(Page D) 0xF6
0xFF07	(Page D) 0xA7	0xFF17	(Page D) 0xB7	0xFF27	(Page D) 0xC7	0xFF37	(Page D) 0xD7	0xFF47	(Page D) 0xE7	0xFF57	(Page D) 0xF7
0xFF08	(Page D) 0xA8	0xFF18	(Page D) 0xB8	0xFF28	(Page D) 0xC8	0xFF38	(Page D) 0xD8	0xFF48	(Page D) 0xE8	0xFF58	(Page D) 0xF8
0xFF09	(Page D) 0xA9	0xFF19	(Page D) 0xB9	0xFF29	(Page D) 0xC9	0xFF39	(Page D) 0xD9	0xFF49	(Page D) 0xE9	0xFF59	(Page D) 0xF9
0xFF0A	(Page D) 0xAA	0xFF1A	(Page D) 0xBA	0xFF2A	(Page D) 0xCA	0xFF3A	(Page D) 0xDA	0xFF4A	(Page D) 0xEA	0xFF5A	(Page D) 0xFA
0xFF0B	(Page D) 0xAB	0xFF2B	(Page D) 0xBB	0xFF2B	(Page D) 0xCB	0xFF3B	(Page D) 0xDB	0xFF4B	(Page D) 0xEB	0xFF5B	(Page D) 0xFB
0xFF0C	(Page D) 0xAC	0xFF1C	(Page D) 0xBC	0xFF2C	(Page D) 0xCC	0xFF3C	(Page D) 1xDC	0xFF4C	(Page D) 0xEC	0xFF5C	(Page D) 0xFC
0xFF0D	(Page D) 0xAD	0xFF1D	(Page D) 0xBD	0xFF2D	(Page D) 0xCD	0xFF3D	(Page D) 1xDD	0xFF4D	(Page D) 0xED	0xFF5D	(Page D) 0xFD
0xFF0E	(Page D) 0xAE	0xFF1E	(Page D) 0xBE	0xFF2E	(Page D) 0xCE	0xFF3E	(Page D) 2xDE	0xFF4E	(Page D) 0xEE	0xFF5E	(Page D) 0xFE
0xFF0F	(Page D) 0xAF	0xFF1F	(Page D) 0xBF	0xFF2F	(Page D) 0xCF	0xFF3F	(Page D) 2xDF	0xFF4F	(Page D) 0xEF	0xFF5F	(Page D) 0xFF

0xFF60	(Page E) 0xA0	0xFF70	(Page E) 0xB0	0xFF80	(Page E) 0xC0	0xFF90	(Page E) 0xD0	0xFFA0	(Page E) 0xE0	0xFFB0	(Page E) 0xF0
0xFF61	(Page E) 0xA1	0xFF71	(Page E) 0xB1	0xFF81	(Page E) 0xC1	0xFF91	(Page E) 0xD1	0xFFA1	(Page E) 0xE1	0xFFB1	(Page E) 0xF1
0xFF62	(Page E) 0xA2	0xFF72	(Page E) 0xB2	0xFF82	(Page E) 0xC2	0xFF92	(Page E) 0xD2	0xFFA2	(Page E) 0xE2	0xFFB2	(Page E) 0xF2
0xFF63	(Page E) 0xA3	0xFF73	(Page E) 0xB3	0xFF83	(Page E) 0xC3	0xFF93	(Page E) 0xD3	0xFFA3	(Page E) 0xE3	0xFFB3	(Page E) 0xF3
0xFF64	(Page E) 0xA4	0xFF74	(Page E) 0xB4	0xFF84	(Page E) 0xC4	0xFF94	(Page E) 0xD4	0xFFA4	(Page E) 0xE4	0xFFB4	(Page E) 0xF4
0xFF65	(Page E) 0xA5	0xFF75	(Page E) 0xB5	0xFF85	(Page E) 0xC5	0xFF95	(Page E) 0xD5	0xFFA5	(Page E) 0xE5	0xFFB5	(Page E) 0xF5
0xFF66	(Page E) 0xA6	0xFF76	(Page E) 0xB6	0xFF86	(Page E) 0xC6	0xFF96	(Page E) 0xD6	0xFFA6	(Page E) 0xE6	0xFFB6	(Page E) 0xF6
0xFF67	(Page E) 0xA7	0xFF77	(Page E) 0xB7	0xFF87	(Page E) 0xC7	0xFF97	(Page E) 0xD7	0xFFA7	(Page E) 0xE7	0xFFB7	(Page E) 0xF7
0xFF68	(Page E) 0xA8	0xFF78	(Page E) 0xB8	0xFF88	(Page E) 0xC8	0xFF98	(Page E) 0xD8	0xFFA8	(Page E) 0xE8	0xFFB8	(Page E) 0xF8
0xFF69	(Page E) 0xA9	0xFF79	(Page E) 0xB9	0xFF89	(Page E) 0xC9	0xFF99	(Page E) 0xD9	0xFFA9	(Page E) 0xE9	0xFFB9	(Page E) 0xF9
0xFF6A	(Page E) 0xAA	0xFF7A	(Page E) 0xBA	0xFF8A	(Page E) 0xCA	0xFF9A	(Page E) 0xDA	0xFFAA	(Page E) 0xEA	0xFFBA	(Page E) 0xFA
0xFF6B	(Page E) 0xAB	0xFF7B	(Page E) 0xBB	0xFF8B	(Page E) 0xCB	0xFF9B	(Page E) 0xDB	0xFFAB	(Page E) 0xEB	0xFFBB	(Page E) 0xFB
0xFF6C	(Page E) 0xAC	0xFF7C	(Page E) 0xBC	0xFF8C	(Page E) 0xCC	0xFF9C	(Page E) 0xDC	0xFFAC	(Page E) 0xEC	0xFFBC	(Page E) 0xFC
0xFF6D	(Page E) 0xAD	0xFF7D	(Page E) 0xBD	0xFF8D	(Page E) 0xCD	0xFF9D	(Page E) 0xDD	0xFFAD	(Page E) 0xED	0xFFBD	(Page E) 0xFD
0xFF6E	(Page E) 0xAE	0xFF7E	(Page E) 0xBE	0xFF8E	(Page E) 0xCE	0xFF9E	(Page E) 0xDE	0xFFAE	(Page E) 0xEE	0xFFBE	(Page E) 0xFE
0xFF6F	(Page E) 0xAF	0xFF7F	(Page E) 0xBF	0xFF8F	(Page E) 0xCF	0xFF9F	(Page E) 0xDF	0xFFAF	(Page E) 0xEF	0xFFBF	(Page E) 0xFF

0xFFC0	(Pege F) 0xA0	0xFFD0	(Page F) 0xB0	0xFFE0	(Pege F) 0xC0	0xFFF0	(Page F) 0xD0				
0xFFC1	(Pege F) 0xA1	0xFFD1	(Page F) 0xB1	0xFFE1	(Pege F) 0xC1	0xFFF1	(Page F) 0xD1				
0xFFC2	(Pege F) 0xA2	0xFFD2	(Page F) 0xB2	0xFFE2	(Pege F) 0xC2	0xFFF2	(Page F) 0xD2				
0xFFC3	(Pege F) 0xA3	0xFFD3	(Page F) 0xB3	0xFFE3	(Pege F) 0xC3	0xFFF3	(Page F) 0xD3				
0xFFC4	(Pege F) 0xA4	0xFFD4	(Page F) 0xB4	0xFFE4	(Pege F) 0xC4	0xFFF4	(Page F) 0xD4				
0xFFC5	(Pege F) 0xA5	0xFFD5	(Page F) 0xB5	0xFFE5	(Pege F) 0xC5	0xFFF5	(Page F) 0xD5				
0xFFC6	(Pege F) 0xA6	0xFFD6	(Page F) 0xB6	0xFFE6	(Pege F) 0xC6	0xFFF6	(Page F) 0xD6				
0xFFC7	(Pege F) 0xA7	0xFFD7	(Page F) 0xB7	0xFFE7	(Pege F) 0xC7	0xFFF7	(Page F) 0xD7				
0xFFC8	(Pege F) 0xA8	0xFFD8	(Page F) 0xB8	0xFFE8	(Pege F) 0xC8	0xFFF8	(Page F) 0xD8				
0xFFC9	(Pege F) 0xA9	0xFFD9	(Page F) 0xB9	0xFFE9	(Pege F) 0xC9	0xFFF9	(Page F) 0xD9				
0xFFCA	(Pege F) 0xAA	0xFFDA	(Page F) 0xBA	0xFFEA	(Pege F) 0xCA	0xFFFA	(Page F) 0xDA				
0xFFCB	(Pege F) 0xAB	0xFFDB	(Page F) 0xBB	0xFFEB	(Pege F) 0xCB	0xFFFB	(Page F) 0xDB				
0xFFCC	(Pege F) 0xAC	0xFFDC	(Page F) 0xBC	0xFFEC	(Pege F) 0xCC	0xFFFC	(Page F) 0xDC				
0xFFCD	(Pege F) 0xAD	0xFFDD	(Page F) 0xBD	0xFFED	(Pege F) 0xCD	0xFFFD	(Page F) 0xDD				
0xFFCE	(Pege F) 0xAF	0xFFDE	(Page F) 0xBE	0xFFEE	(Pege F) 0xCF	0xFFFE	(Page F) 0xDE				
0xFFCF	(Pege F) 0xAF	0xFFDF	(Page F) 0xBF	0xFFEF	(Pege F) 0xCF	0xFFFF	(Page F) 0xDF				

In external MCU mode, external host interface has the privilege to access REG_XFR_WRAPPER. The total 256 addresses are separated into 3 pages. 0xFF00 ~ 0xFF5F is Page D, 0xFF60 ~0xFFBF is Page E and 0xFFC0 ~ 0xFFFF is Page F. External host control signals are passed to scalar register interface. Host interface must integrate the output of scalar interface and REG_XFR_WRAPPER depends on page selection.

When ISP is activated, REG_XFR_WRAPPER is accessed by ISP interface, it has the highest priority. DW8051 and external host is selected by power-on-latch signal, ext_host_sel.

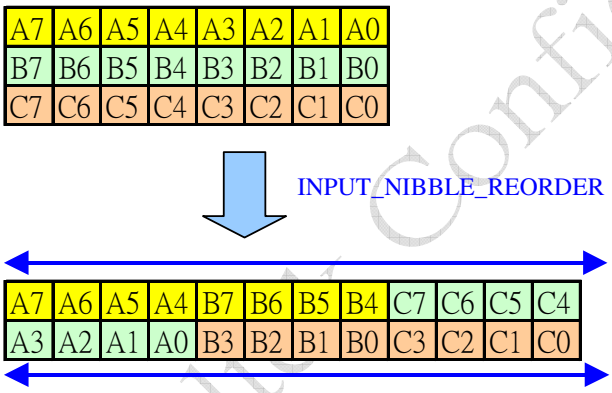
Interrupt Control

Register::IRQ_Status			0xFF00		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7	--	--	Reserved	
M2PLL_IRQ_EVENT	6	R/W	0	M2PLL-abnormal Event Status 1. Select M2PLL as clock source, but M2PLL power down, power saving or output disable, clear this bit to disable the interrupt	Rport Wport
Reserved	5	R/W	0	Reserved	Rport Wport
SCA_IRQ_EVENT	4	R/W	0	**Scalar-related Event Status** 1. IF Scalar integrated IRQ event occurred since the last status cleared	Rport Wport
I2CM_IRQ_EVENT	3	R/W	0	**I2C module Event Status** 1. IF I2C module IRQ event occurred since the last status cleared	Rport Wport
ADC_IRQ_EVENT	2	R/W	0	**ADC Event Status** 1: If the ADC IRQ event occurred since the last status cleared	Rport Wport
WDT_IRQ_EVENT	1	R/W	0	WDT_IRQ event status 1: If the WDT_IRQ event occurred since the last status cleard	Rport Wport
DDC_IRQ_EVENT	0	R/W	0	**DDC Event Status** 1: If the DDC IRQ event occurred since the last status cleared	Rport Wport

Register::IRQ_Priority			0xFF01		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7	--	--	Reserved	
M2PLL_IRQ_PRI	6	R/W	0	M2PLL-abnormal IRQ Priority 0: Connected to int0 1: Connected to int1	
Reserved	5	R/W	0	Reserved	
SCA_IRQ_PRI	4	R/W	0	**Scalar integrated IRQ Priority** 0: Connected to int0 1: Connected to int1	
I2CM_IRQ_PRI	3	R/W	0	**I2C module IRQ Priority** 0: Connected to int0 1: Connected to int1	
ADC_IRQ_PRI	2	R/W	0	**ADC IRQ Priority** 0: Connected to int0 1: Connected to int1	
WDT_IRQ_PRI	1	R/W	0	**WDT IRQ Priority** 0: Connected to int0 1: Connected to int1	
DDC_IRQ_PRI	0	R/W	0	**DDC IRQ Priority** 0: Connected to int0 1: Connected to int1	

Register:: VSYNC_IRQ~~REV_DUMMY1~~			0xFF02		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
VS1_IRQ_PRI	7	R/W	1	**VS1 IRQ Priority** 0: Connected to int0 1: Connected to int1	
VS0_IRQ_PRI	6	R/W	0	**VS0 IRQ Priority** 0: Connected to int0 1: Connected to int1	
VS1_IRQ_EVENT	5	R/W	0	VS1 Event Status, write 0 to clear	
VS0_IRQ_EVENT	4	R/W	0	VS0 Event Status, , write 0 to clear	
VS1_INT_EN	3	R/W	0	VS1 Interrupt enable 0: disable 1 :enable	
VS1_trigger_type	2	R/W	0	VS1 trigger type 0: posedge trigger 1: bypass mode (level trigger)	
VS0_INT_EN	1	R/W	0	VS0 Interrupt enable 0: disable 1: enable	
VS0_trigger_type	0	R/W	0	VS0 trigger type 0: posedge trigger 1: bypass mode (level trigger)	

OSD reorder



Register::Triple_Bytes_Operation			0xFF03		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
GLOBAL_NIBBLE_REORDER_EN	7	R/W	0	1: The input bit sequence of three bytes, A[7:0], B[7:0] C[7:0] will be reordered to A[7:4], B[7:4], C[7:4], A[3:0], B[3:0], C[3:0] before any operation and change back to input sequence at the end 0: No modification	
INPUT_NIBBLE_REORDER_EN	6	R/W	1	1: The input bit sequence of three bytes, A[7:0], B[7:0], C[7:0] will be reordered to {A[7:4], B[7:4], C[7:4]}, {A[3:0], B[3:0], C[3:0]} before any operation 0: No modification	
OUTPUT_NIBBLE_REORDER_EN	5	R/W	0	1: The output bit sequence of three bytes, A[7:4], B[7:4], C[7:4], A[3:0], B[3:0],	

				C[3:0] will be reordered to A[7:0], B[7:0], C[7:0] before output 0: No modification	
TBO_dummy	4	R/W	0	Dummy Bit	
FIRST_BYTE_SHIFT_DIRECTION	3	R/W	1	1: LEFT SHIFT 0: RIGHT SHIFT	
LEFT_BYTE_SHIFT_DIRECTION	2	R/W	0	1: LEFT SHIFT 0: RIGHT SHIFT	
RESTART_FROM_LAST	1	R/W	0	1: Restart from last three bytes (auto_clear)	rport WPORT
RESTART_FROM_FIRST	0	R/W	0	1: Restart from input first bytes (auto_clear)	RPORT WPORT

Register::Shift_Bits_Number				**0xFF04**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
FIRST_BITS_SHIFT	7:4	R/W	0	The left/right bit shift times.	
SECOND_BITS_SHIFT	3:0	R/W	0	The left/right bit shift times.	

Register::Byte_Address				**0xFF05**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
THIRD_BITS_SHIFT	7:4	R/W	0	The left/right bit shift times.	
LEFT_BITS_SHIFT	3:0	R/W	0	The left/right bit shift times.	

Register::Input_Triple_Bytes				**0xFF06**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
INPUT_OP_BYTES	7:0	R/W	0	Three bytes input by sequence here.	Rport Wport

Register::Result_Triple_Bytes				**0xFF07**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
OUTPUT_OP_BYTES	7:0	R/W	0	Output bytes input by sequence here.	RPORT WPORT

Register::Byte_Address				**0xFF18**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ALL_ADDR_RST	7	R/W	0	1: Write 1 to reset input/result triple bytes address to 0	RPORT WPORT
Reserved	6:3	R/W	--	Reserved	
INPUT_OUTPUT_CONTROL	2	R/W	0	1: Write BYTE_ADDR to control address of result triple bytes 0: Write BYTE_ADDR to control address of input triple bytes	Rport Wport
BYTE_ADDR	1:0	R/W	0	Byte Address for Current Input or Result bytes	Rport Wport

ADC

A/D Converter

RTD2485XD has embedded 4 channels of analog-to-digital converter by A_circuit. The ADCs_ACKT convert analog input voltage on the four A/D input pins to four 8-bit digital data stored in XFRs (FF09 ~ FF0C) sequentially.

The ADC conversion range is from GND to VDD and the conversion is linear and monotonic with no missing codes. To start A/D conversion, set STRT_ADCx = 1 and the conversion will be completed in less than 12us for 4 channels.

ADC_Ackt block diagram (for Key Sensing, D-connector)

Register::ADC_Acontrol				0xFF08	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
STRT_ADC_ACKT	7	R/W	0	Write 1 to start the A/D conversion. Auto clear when A/D Conversion has been completed. 0:A/D Conversion has been completed 1:A/D Conversion is not completed yet	Rport Wport
ADC_ATEST	6	R/W	0	0: Normal operation 1: ADC test mode	
Reserved	5:3	R/W	0	Reserved	
ADC_A_BIAS_ADJ	2:1	R/W	1	ADC bias current adjust 00: 15u 01: 20u 10: 25u 11: 30u	
ADC_A_CK_SEL	0	R/W	0	Inverse ADC input clock pos/neg 0: pos 1: neg	

Register::ADC_A0_convert_result				0xFF09	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ADC_A0_DATA	7:0	R	FF	Converted data of ADC_A0	

Register::ADC_A1_convert_result				0xFF0A	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ADC_A1_DATA	7:0	R	FF	Converted data of ADC_A1	

Register::ADC_A2_convert_result				0xFF0B	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ADC_A2_DATA	7:0	R	FF	Converted data of ADC_A2	

Register::ADC_A3_convert_result				0xFF0C	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ADC_A3_DATA	7:0	R	FF	Converted data of ADC_A3	

Register::ADC_CLK_DIV				0xFF0D	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ADC_CLK_SEL	7	R/W	0	ADC clk from 0: Xtal/IOSC 1: M2PLL	
ADC_CLK_DIV	6:0	R/W	0x04	ADC clk divider 0x00 is DIV1, 0x01 is DIV1, 0x02 is DIV2 and so on.	

				ADC clk target is 3Mhz.	

Register:: Auto_Mode_Ctrl01				0xFF0E	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ADC_INT_flag	7	R/W	0	Interrupt flag Write "1" to clear	Rport Wport
Reserved	6:3	R/W	0	Reserved	
ADC_Auto_Mode_EN	2	R/W	0	0: Disable 1: Enable auto mode	
ADC_EN_DEBU	1	R/W	1	0: disable de-bounce 1: enable de-bounce. meas 3 times (based on waiting time)	
ADC_INT_EN	0	R/W	0	(L_Threshold < Meas < H_Threshold), Meas_Cnt +1, Meas_Cnt > debounce times, into ISR 0: disable INT 1: enable INT	

Register::ADC0_THRESHOLD_H				0xFF0F	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ADC0_HI_THRESHOLD	7:0	R/W	00	High threshold value for auto mode	

Register::ADC0_THRESHOLD_L				0xFF10	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ADC0_LO_THRESHOLD	7:0	R/W	00	Low threshold value for auto mode	

Register::ADC1_THRESHOLD_H				0xFF11	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ADC1_HI_THRESHOLD	7:0	R/W	00	High threshold value for auto mode	

Register::ADC1_THRESHOLD_L				0xFF12	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ADC1_LO_THRESHOLD	7:0	R/W	00	Low threshold value for auto mode	

Register::ADC2_THRESHOLD_H				0xFF13	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ADC2_HI_THRESHOLD	7:0	R/W	00	High threshold value for auto mode	

Register::ADC2_THRESHOLD_L				0xFF14	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ADC2_LO_THRESHOLD	7:0	R/W	00	Low threshold value for auto mode	

Register::ADC3_THRESHOLD_H				0xFF15	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ADC3_HI_THRESHOLD	7:0	R/W	00	High threshold value for auto mode	

Register::ADC3_THRESHOLD_L				0xFF16	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ADC3_LO_THRESHOLD	7:0	R/W	00	Low threshold value for auto mode	

Register::CTRL0_WAIT_TIME_VALUE				0xFF17	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ADC_Wait_Value	7:0	R/W	0x01	Wait time= wait_value * 1.3ms (**0.75Khz**), 1.3ms < wait time < 340ms	

DDC

RTD2485XD has three DDC ports. The MCU can access the following three DDC interface:

- DDC_RAM1 (FD80~FDFF) through pin ASDL and ASDA by ADC DDC channel.
- DDC_RAM2 (FE00~FE7F) through pin DSDL and DSDA by DVI DDC channel.
- DDC_RAM3 (FE80~FEFF) through pin HSDL and HSDA by HDMI DDC channel.

Besides, the DDC_RAM1, DDC_RAM2, DDC_RAM3 can be assigned from 128 to 256bytes. The actual sizes of each DDC_RAM are determined by the combination of ADDCRAM_ST, DDDCRAM_ST, and HDDCRAM_ST. The DDC RAMs are shared with MCU's XSRAM, configuration must be take care for reserving XSRAM for programming. For example, Set ADDCRAM_ST = 0x2, DDDCRAM_ST = 0x3, HDDCRAM_ST = 0x2 and disable , DVI DDC. The XSRAM for MCU is 512 bytes and ADC DDC/HDMI DDC is used with 256 bytes.

The DDC of RTD2485XD is compliant with VESA DDC standard. All DDC slaves are in DDC1 mode after reset. When a high to low transition is detected on ASCL/DSCL/HSCL pin, the DDC slave will enter DDC2 transition mode. The DDC slave can revert to DDC1 mode if the SCL signal keeps unchanged for 128 VSYNC periods in DDC2 transition mode and RVT_A_DDC1_EN / RVT_D_DDC1_EN / RVT_H_DDC1_EN = 1. In DDC2 transition mode, the DDC slave will lock in DDC2 mode if a valid control byte is received. Furthermore, user can force the DDC slave to operate DDC2 mode by setting A_DDC2 / D_DDC2/ H_DDC2 = 1.

Register:: ADC_SEGMENT_ADDRESS				**0XFF19**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ADC_SEG_ADDR	7:1	R/W	0x30	ADC slave address for segment control	
Reserved	0	--	--	Reserved	

Register:: ADC_SEGMENT_DATA				**0XFF1A**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ADC_SEG_DATA	7:0	R/W	0x00	Data Access for Slave ID, ADC_SEGMENT_ADDRESS, in ADC DDC	Rport Wport

Register::ADC_DDC_enable				**0xFF1B**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
A_DDC_ADDR	7:5	R/W	0	ADC DDC Channel Address Least Significant 3 Bits (The default DDC channel address MSB 4 Bits is "A")	
A _SCL_DBN_SEL	4	R/W	0	SCL Debounce Clock Selection 0: De-bounce clock (after clock divider) 1: De-bounce reference clock	
A_DDC_W_STA	3	R/W	0	ADC DDC Write Status (for external DDC access only) It is cleared after write. (No matter what the data are)	Rport wport
A_DDCRAM_W_EN	2	R/W	0	ADC DDC SRAM Write Enable (for external DDC access only) 0: Disable 1: Enable	
A_DBN_EN	1	R/W	1	ADC DDC De-bounce Enable 0: Disable 1: Enable (with crystal/4)	
A_DDC_EN	0	R/W	0	ADC DDC Channel Enable Bit 0: MCU access Enable 1: DDC channel Enable	

Register::ADC_DDC_control_1				**0xFF1C**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
A_DBN_CLK_SEL	7:6	R/W	0	De-bounce clock divider 00: 1/1 reference clock 01: 1/2 reference clock	

				1X: 1/4 reference clock	
A_STOP_DBN_SEL	5:4	R/W	3	De-bounce sda stage 00: no latch stage 01: latch one stage 10: latch two stage 11: latch three stage	
A_SYS_CK_SEL	3	R/W	0	De-bounce reference clock 0: crystal clock 1. Serial flash clock (M2PLL / Flash_DIV)	
A_DDC2	2	R/W	0	Force to ADC DDC to DDC2 mode 0: Normal operation 1: DDC2 is active	
RST_A_DDC	1	R/W	0	Reset ADC DDC circuit 0: Normal operation 1: reset (auto cleared)	Rport wport
RVT_A_DDC1_EN	0	R/W	0	ADC DDC revert to DDC1 enable(SCL idle for 128 VSYNC) 0: Disable 1: Enable	

Register::ADC_DDC_control_2				**0xFF1D**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
A_SEG_WR_EN	7	R/W	0	Enable interrupt of ADC segment address write 0: Disable 1: Enable	
Reserved	6:5	--	--	Reserved	
ADC_DDCCI_EN	4	R/W	1	ADC DDC-CI Channel Enable Switch 0: Disable 1: Enable	
ADC_DDCISP_EN	3	R/W	1	ADC DDC ISP Channel Enable Switch 0: Disable 1: Enable If all of DDC ISP Channel are disable, ADC DDC ISP Channel will be enable.	
ADC_DDCSEG_EN	2	R/W	1	ADC DDC Segment Channel Enable Switch 0: Disable 1: Enable	
A_SEG_WR	1	R/W	0	ADC DDC Segment Write Status 0: no external write after clear 1: new external write after clear It is cleared after write	Wport Rport
A_FORCE_SCL_L	0	R/W	0	Force external SCL bus low 1: Driving SCL = 0 after external SCL = 0 0: Release SCL	

Register::DVI_DDC_enable				**0xFF1E**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
D_DDC_ADDR	7:5	R/W	0	DVI DDC Channel Address Least Significant 3 Bits (The default DDC channel address MSB 4 Bits is "A")	
D_SCL_DBN_SEL	4	R/W	0	SCL Debounce Clock Selection 0: De-bounce clock (after clock divider) 1: De-bounce reference clock	
D_DDC_W_STA	3	R/W	0	DVI DDC External Write Status (for external DDC access only) It is cleared after write.	Wport rport

D_DDCRAM_W_EN	2	R/W	0	DVI DDC External Write Enable (for external DDC access only) 0: Disable 1: Enable	
D_DBN_EN	1	R/W	1	DVI DDC Debounce Enable 0: Disable 1: Enable (with crystal/4)	
D_DDC_EN	0	R/W	0	DVI DDC Channel Enable Switch 0: MCU access Enable 1: External DDC access Enable	

Register::DVI_DDC_control_1				**0xFF1F**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
D_DBN_CLK_SEL	7:6	R/W	0	De-bounce clock divider 00: 1/1 reference clock 01: 1/2 reference clock 1X: 1/4 reference clock	
D_STOP_DBN_SEL	5:4	R/W	3	De-bounce sda stage 00: no latch stage 01: latch one stage 10: latch two stage 11: latch three stage	
D_SYS_CK_SEL	3	R/W	0	De-bounce reference clock 0: crystal clock 1. Serial flash clock (M2PLL / Flash_DIV)	
D_DDC2	2	R/W	0	Force to DVI DDC to DDC2 mode 0: Normal operation 1: DDC2 is active	
RST_D_DDC	1	R/W	0	Reset DVI DDC circuit 0: Normal operation 1: reset (auto cleared)	Rport wport
RVT_D_DDC1_EN	0	R/W	0	DVI DDC revert to DDC1 enable(SCL idle for 128 VSYNC) 0: Disable 1: Enable	

Register::DVI_DDC_control_2				**0xFF20**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
D_SEG_WR_EN	7	R/W	0	Enable interrupt of DVI segment address write 0: Disable 1: Enable	
Reserved	6:5	--	--	Reserved	
DVI_DDCCI_EN	4	R/W	1	DVI DDC-CI Channel Enable Switch 0: Disable 1: Enable	
DVI_DDCISP_EN	3	R/W	1	DVI DDC ISP Channel Enable Switch 0: Disable 1: Enable	
DVI_DDCSEG_EN	2	R/W	1	DVI DDC Segment Channel Enable Switch 0: Disable 1: Enable	
D_SEG_WR	1	R/W	0	DVI DDC Segment Write Status 0: no external write after clear 1: new external write after clear It is cleared after write	Wport Rport
D_FORCE_SCL_L	0	R/W	0	Force external SCL bus low 1: Driving SCL = 0 after external SCL = 0 0: Release SCL	

Register::DDCRAM_partition				0xFF21	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ISP_DDC2B_SWITCH	7	R/W	0	0：Not Allow DDC2B access in ISP Mode，ddc can program 1：Allow DDC2B access in ISP Mode，but can not program	
FORCE_DDCRAM_ST	6	R/W	0	Force ADC/DVI/HDMI DDC RAM Start Address to be 0xFD00 1: enable 0: disable (RAM start address decided by bit[5:4], bit[3:2], bit[1:0])	
ADDCRAM_ST	5:4	R/W	0x3	ADDC RAM Start Address is 0xFC00 + ADDCRAM_ST*0x80, ADDCRAM SIZE = DDDCRAM_ST – ADDCRAM_ST	
DDDCRAM_ST	3:2	R/W	0x3	DDDC RAM Start Address is 0xFC80 + DDDCRAM_ST*0x80, DDDCRAM SIZE = HDDCRAM_ST – DDDCRAM_ST	
HDDCRAM_ST	1:0	R/W	0x3	HDDC RAM Start Address is 0xFD00 + ADDCRAM_ST*0x80, HDDCRAM SIZE = 0xFF00 – HDDCRAM_ST	

XRAM Arbitor	Byte	Start	End	
External RAM	1152	0xF900	0xFD7F	xData
DDC0_RAM	128	0xFD80	0xFDFF	VGA
DDC1_RAM	128	0xFE00	0xFE7F	DVI_1
DDC2_RAM	128	0xFE80	0xFEFF	DVI_2
External RAM	896	0xF900	0xFC7F	xData
DDC0_RAM	128	0xFC80	0xFCFF	VGA
DDC1_RAM	256	0xFD00	0xFDFF	HDMI_1
DDC2_RAM	256	0xFE00	0xFEFF	HDMI_2

Register::VSYNC_Sel				0xFF22	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:4	-			
VS_CON0	3:2	R/W	0	00: VSYNC0 signal is connected to ADC DDC 01: VSYNC0 signal is connected to DVI DDC 1x: VSYNC0 signal is connected to HDMI DDC	
~~VS_CON1~~	1:0	R/W	0	~~00: VSYNC1 signal is connected to ADC DDC~~ ~~01: VSYNC1 signal is connected to DVI DDC~~ ~~1x: VSYNC1 signal is connected to HDMI DDC~~	

DDC-CI

Register::IIC_set_slave				0xFF23	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
IIC_ADDR	7:1	R/W	37	IIC Slave Address to decode	
CH_SEL	0	R/W	0	Channel Select, overridden by HCH_SEL(0xFF2B[0]) = 1 0: from ADC DDC 1: from DVI DDC	

pin121 / pin122 → DDCSCL3 / DDCSDA3, (HDMI DDC) → **0xFF2B** Bit 0 Enable
pin123/ pin124 → DDCSCL2 / DDCSDA2, (DVI DDC), → **0xFF23** Bit 0 Enable

Register::IIC_sub_in				0xFF24	
Name	Bits	R/W	Default	Comments	Config
IIC_SUB_ADDR	7:0	R	00	IIC Sub-Address Received	

Register::IIC_data_in				0xFF25	
Name	Bits	R/W	Default	Comments	Config
IIC_D_IN	7:0	R	00	IIC data received. 16-bytes depth read in buffer mode	RPORT

Register::IIC_data_out				0xFF26	
Name	Bits	R/W	Default	Comments	Config
IIC_D_OUT	7:0	W	00	IIC data to be transmitted	Rport Wport

Register::IIC_status				0xFF27	
Name	Bits	R/W	Default	Comments	Config
A_WR_I	7	R/W	0	If ADC DDC detects a STOP condition in write mode, this bit is set to “1” . Write 0 to clear.	Rport Wport
D_WR_I	6	R/W	0	If DVI DDC detects a STOP condition in write mode, this bit is set to “1” . Write 0 to clear.	Rport Wport
DDC_128VS0_I	5	R/W	0	In DDC2 Transition mode, SCL idle for 128 VSYNC. Write 0 to clear.	Rport Wport
STOP_I	4	R/W	0	If IIC detects a STOP condition(slave address must match), this bit is set to “1” . Write 0 to clear.	Rport Wport
D_OUT_I	3	R	0	If IIC_DATA_OUT loaded to serial-out-byte, this bit is set to “1”. Write IIC_data_out (FF26) to clear.	
D_IN_I	2	R	0	If IIC_DATA_IN latched, this bit is set to “1” . Read IIC_data_in (FF25) to clear.	
SUB_I	1	R/W	0	If IIC_SUB latched, this bit is set to “1” Write 0 to clear.	Rport Wport
SLV_I	0	R/W	0	If IIC_SLAVE latched, this bit is set to “1” Write 0 to clear.	Rport Wport

Register::IIC_IRQ_control				0xFF28	
Name	Bits	R/W	Default	Comments	Config
AWI_EN	7	R/W	0	0: Disable the A_WR_I signal as an interrupt source 1: Enable the A_WR_I signal as an interrupt source	
DWI_EN	6	R/W	0	0: Disable the D_WR_I signal as an interrupt source 1: Enable the D_WR_I signal as an interrupt source	
DDC_128VSI0_EN	5	R/W	0	0: Disable the 128VS0_I signal as an interrupt source 1: Enable the 128VS0_I signal as an interrupt source	
STOPI_EN	4	R/W	0	0: Disable the STOP_I signal as an interrupt source 1: Enable the STOP_I signal as an interrupt source	
DOI_EN	3	R/W	0	0: Disable the D_OUT_I signal as an interrupt source 1: Enable the D_OUT_I signal as an interrupt source	
DII_EN	2	R/W	0	0: Disable the D_IN_I signal as an interrupt source 1: Enable the D_IN_I signal as an interrupt source	
SUBI_EN	1	R/W	0	0: Disable the SUB_I signal as an interrupt source 1: Enable the SUB_I signal as an interrupt source	
SLVI_EN	0	R/W	0	0: Disable the SLV_I signal as an interrupt source 1: Enable the SLV_I signal as an interrupt source	

Register::IIC_status2				0xFF29	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
IIC_FORCE_SCL_L	7	R/W	0	Force SCL = 0 when one of the following tow case happen: 1. IIC_BUF_FULL = 1 in write mode 2. IIC_BUF_EMPTY = 1 in read mode	
FORCE_NACK	6	R/W	0	Force IIC return NACK when one of the following tow case happen: IIC_BUF_FULL = 1 in write mode	
IIC_BUF_OV	5	R/W	0	IIC_DATA_BUFFER Overflow. Write '0' to clear	Rport Wport
IIC_BUF_UN	4	R/W	0	IIC_DATA_BUFFER Underflow. Write '0' to clear	Rport Wport
~~DDC_128VS1_I~~ Not Used	3	R/W	0	~~In DDC2 Transition mode, SCL idle for 128 VSYNC. Write 0 to clear. Write '0' to clear~~ Not Used	Rport Wport
IIC_BUF_FULL	2	R	0	IIC_DATA_BUFFER Full If IIC_DATA buffer is full, this bit is set to "1". (On-line monitor) The IIC_DATA buffer Full status will be on-line-monitor the condition, once it becomes full, it kept high, if it is not-full, then it goes low.	
IIC_BUF_EMPTY	1	R	0	IIC_DATA_BUFFER Empty If IIC_DATA buffer is empty, this bit is set to "1". (On-line monitor) The IIC_DATA buffer Empty status will be on-line-monitor the condition, once it becomes empty, it kept high, if it is not-empty, then it goes low.	
H_WR_I	0	R/W	0	If HDMI DDC detects a STOP condition in write mode, this bit is set to "1" . Write 0 to clear.	rport Wport

Register::IIC_IRQ_control2				0xFF2A	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
AUTO_RST_BUF	7	R/W	0	Auto reset IIC_DATA Buffer 0: disable 1: enable In host (pc) write enable, when IIC write (No START after IIC_SUB), reset IIC_DATA buffer.	
RST_DATA_BUF	6	R/W	0	Reset IIC_DATA buffer 0: Finish 1: Reset	Wport Rport
DATA_BUF_WEN	5	R/W	0	IIC_DATA buffer write enable 0: host (pc) write enable 1: slave (mcu) write enable Both PC and MCU can read IIC_DATA buffer, but only one can write IIC_DATA buffer.	
Dummy_2	4	R/W	0	Reserved	
~~DDC_128VSI1_EN~~ Not Used	3	R/W	0	~~0: Disable the 128VS1_I signal as an interrupt source~~ ~~1: Enable the 128VS_I signal as an interrupt source~~ Not Used	
DDC_BUF_FULL_EN	2	R/W	0	0: Disable the DDC_DATA_BUFFER Full signal as an interrupt source 1: Enable the DDC_DATA_BUFFER Full signal as an interrupt source	
DDC_BUF_EMPTY_EN	1	R/W	0	0: Disable the DDC_DATA_BUFFER Empty signal as an	

				interrupt source 1: Enable the DDC_DATA_BUFFER Empty signal as an interrupt source	
HWI_EN	0	R/W	0	0: Disable the H_WR_I signal as an interrupt source 1: Enable the H_WR_I signal as an interrupt source	

Register::IIC_channel_control				**0xFF2B**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:4	--	0	Reserved	
IIC_BUF_OV_EN	3	R/W	0	0: Disable the IIC_DATA_BUFFER overflow signal as an interrupt source 1: Enable the IIC_DATA_BUFFER overflow signal as an interrupt source	
IIC_BUF_UN_EN	2	R/W	0	0: Disable the IIC_DATA_BUFFER underflow signal as an interrupt source 1: Enable the IIC_DATA_BUFFER underflow signal as an interrupt source	
RLS_SCL_SU	1	R/W	0	Set IIC data Setup Time When holding SCL low 0: Use Delay Chain (~5ns) 1: Use Crystal Clock to increase data setup time relative to SCL clock line	
HCH_SEL	0	R/W	0	Channel Select of DDC-CI 1: from HDMI DDC 0: controlled by CH_SEL	

pin121 / pin122 → DDCSCL3 / DDCSDA3, (HDMI DDC) → **0xFF2B** Bit 0 Enable
pin123/ pin124 → DDCSCL2 / DDCSDA2, (DVI DDC), → **0xFF23** Bit 0 Enable

Register::HDMI_DDC_enable				**0xFF2C**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
H_DDC_ADDR	7:5	R/W	0	HDMI DDC Channel Address Least Significant 3 Bits (The default DDC channel address MSB 4 Bits is "A")	
H_SCL_DBN_SEL	4	R/W	0	SCL Debounce Clock Selection 0: De-bounce reference clock 1: De-bounce clock (after clock divider)	
H_DDC_W_STA	3	R/W	0	HDMI DDC Write Status (for external DDC access only) It is cleared after write.	Rport Wport
H_DDCRAM_W_EN	2	R/W	0	HDMI DDC SRAM Write Enable (for external DDC access only) 0: Disable 1: Enable	
H_DBN_EN	1	R/W	1	HDMI DDC De-bounce Enable 0: Disable 1: Enable (with crystal/4)	
H_DDC_EN	0	R/W	0	HDMI DDC Channel Enable Bit 0: MCU access Enable 1: DDC channel Enable	

Register::HDMI_DDC_control_1				**0xFF2D**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
H_DBN_CLK_SEL	7:6	R/W	0	De-bounce clock divider 00: 1/1 reference clock 01: 1/2 reference clock 1X: 1/4 reference clock	

H_STOP_DBN_SEL	5:4	R/W	3	De-bounce sda stage 00: no latch stage 01: latch one stage 10: latch two stage 11: latch three stage	
H_SYS_CK_SEL	3	R/W	0	De-bounce reference clock 0: crystal clock 1. Serial flash clock (M2PLL / Flash_DIV)	
H_DDC2	2	R/W	0	Force to HDMI DDC to DDC2 mode 0: Normal operation 1: DDC2 is active	
RST_H_DDC	1	R/W	0	Reset HDMI DDC circuit 0: Normal operation 1: reset (auto cleared)	Rport Wport
RVT_H_DDC1_EN	0	R/W	0	HDMI DDC revert to DDC1 enable(SCL idle for 128 VSYNC) 0: Disable 1: Enable	

Register::HDMI_DDC_control_2			**0xFF2E**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
H_SEG_WR_EN	7	R/W	0	Enable interrupt of HDMI segment address write 0: Disable 1: Enable	
Dummy_3	6:5	R/W	0		
HDMI_DDCCI_EN	4	R/W	1	HDMI DDC-CI Channel Enable Switch 0: Disable 1: Enable	
HDMI_DDCISP_EN	3	R/W	1	HDMI DDC ISP Channel Enable Switch 0: Disable 1: Enable	
HDMI_DDCSEG_EN	2	R/W	1	HDMI DDC Segment Channel Enable Switch 0: Disable 1: Enable	
H_SEG_WR	1	R/W	0	HDMI DDC Segment Write Status 0: no external write after clear 1: new external write after clear It is cleared after write	Wport Rport
H_FORCE_SCL_L	0	R/W	0	Force external SCL bus low 1: Driving SCL = 0 after external SCL = 0 0: Release SCL	

The access ports below are used for external host interface only.

Register::ADC_DDC_INDEX			**0XFF2F**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
A_DDC_INDEX	7:0	R/W	0	DDC SRAM Read/Write Index Register [7:0]	Rport Wport

Register::ADC_DDC_ACCESS_PORT			**0XFF30**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
A_DDC_ACCESS_PORT	7:0	R/W	0	DDC SRAM Read/Write Port	Rport Wport

Register::DVI_DDC_INDEX			**0XFF31**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
D_DDC_INDEX	7:0	R/W	0	DDC SRAM Read/Write Index Register [7:0]	Rport Wport

Register::DVI_DDC_ACCESS_PORT				0XFF32	
Name	Bits	R/W	Default	Comments	Config
D_DDC_ACCESS_PORT	7:0	R/W	0	DDC SRAM Read/Write Port	Rport Wport

Register::HDMI_DDC_INDEX				0XFF33	
Name	Bits	R/W	Default	Comments	Config
H_DDC_INDEX	7:0	R/W	0	DDC SRAM Read/Write Index Register [7:0]	Rport Wport

Register::HDMI_DDC_ACCESS_PORT				0XFF34	
Name	Bits	R/W	Default	Comments	Config
H_DDC_ACCESS_PORT	7:0	R/W	0	DDC SRAM Read/Write Port	Rport Wport

Register:: DDCCI_REMAIN_DATA				0XFF35	
Name	Bits	R/W	Default	Comments	Config
Reserved	7:5	--	0	Reserved	
DDCCI_REMAIN_LEN	4:0	R	0	DDCCI Remaining data length (= write_pointer – read_pointer)	

Register:: DVI_SEGMENT_ADDRESS				0XFF36	
Name	Bits	R/W	Default	Comments	Config
DVI_SEG_ADDR	7:1	R/W	0x30	DVI DDC slave address for segment control	
Reserved	0	--	--	Reserved	

Register:: DVI_SEGMENT_DATA				0XFF37	
Name	Bits	R/W	Default	Comments	Config
DVI_SEG_DATA	7:0	R/W	0x00	Data Access for Slave ID, DVI_SEGMENT_ADDRESS, in DVI DDC	Rport Wport

Register:: HDMI_SEGMENT_ADDRESS				0XFF38	
Name	Bits	R/W	Default	Comments	Config
HDMI_SEG_ADDR	7:1	R/W	0x30	HDMI DDC slave address for segment control	
Reserved	0	--	--	Reserved	

Register:: HDMI_SEGMENT_DATA				0XFF39	
Name	Bits	R/W	Default	Comments	Config
HDMI_SEG_DATA	7:0	R/W	0x00	Data Access for Slave ID, HDMI_SEGMENT_ADDRESS, in HDMI DDC	Rport Wport

PWM

RTD2485XD supports 6 channels of PWM DAC. The resolution of each PWM is 12-bit. PWM0, PWM1, PWM2, PWM3, PWM4 and PWM5 are connected to DA0, DA1, DA2, DA3, DA4 and DA5 respectively. The figure below represents the PWM clock generator. Based on the clock, we make up the PWM waveform which frequency is 1/4096 of the PWM clock.
The PWM duty registers have 12-bit resolution. These registers have double buffer mechanism. When

write the MSB bit, the 12-bit data will be loaded.
The PWM frequency is :

$$Fpwm = fclk / 2^{M} / (N+1) / DUT$$

Where each variables are controlled by:

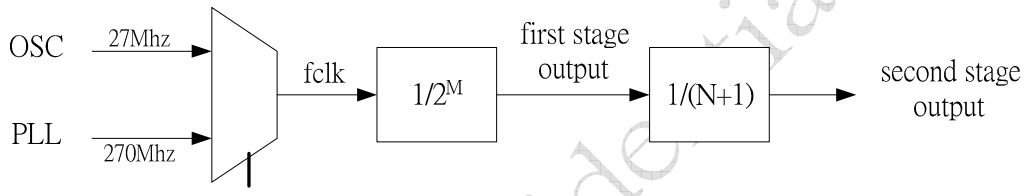
M: PWMx_M

N: PWMxH_N * 256 + PWMxL_N

DUT: (PWMx_DUT_8B == 1)? 256 : 4096

The 12bit duty PWM frequency range is :
fclk=14.3M, fpwm = 3.5KHz ~ 0.1Hz
fclk=27M, fpwm = 6.6KHz ~ 0.2Hz
fclk=243M, fpwm = 60KHz ~ 1.8Hz
The 8bit duty PWM frequency range is :
fclk=14.3M, fpwm = 56KHz ~ 1.6Hz
fclk=27M, fpwm = 105.6KHz ~ 3.2Hz
fclk=243M, fpwm = 960KHz ~ 28.8Hz
fclk=270M, fpwm = 1056KHz ~ 32Hz

PWM clock generator



Register::PWM_CK_SEL				0xFF3A	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PWM_CK_SEL_DUMMY	7:6	R/W	0	dummy	
PWM5_CK_SEL	5	R/W	0	PWMx clock generator input source 0: Crystal 1: PLL output (When PWM5_CK_SEL_HS = 0)	
PWM4_CK_SEL	4	R/W	0	PWMx clock generator input source 0: Crystal 1: PLL output (When PWM5_CK_SEL_HS = 0)	
PWM3_CK_SEL	3	R/W	0	PWMx clock generator input source 0: Crystal 1: PLL output (When PWM5_CK_SEL_HS = 0)	
PWM2_CK_SEL	2	R/W	0	PWMx clock generator input source 0: Crystal 1: PLL output (When PWM5_CK_SEL_HS = 0)	
PWM1_CK_SEL	1	R/W	0	PWMx clock generator input source 0: Crystal 1: PLL output (When PWM5_CK_SEL_HS = 0)	
PWM0_CK_SEL	0	R/W	0	PWMx clock generator input source 0: Crystal	

				1: PLL output (When PWM5_CK_SEL_HS = 0)	

Register::PWM03_M				0xFF3B	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PWM3_M	7:6	R/W	0	PWMx clock first stage divider	
PWM2_M	5:4	R/W	0	PWMx clock first stage divider	
PWM1_M	3:2	R/W	0	PWMx clock first stage divider	
PWM0_M	1:0	R/W	0	PWMx clock first stage divider	

Register::PWM45_M				0xFF3C	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PWM_M_DUMMY	7:4	R/W	0	dummy	
PWM5_M	3:2	R/W	0	PWMx clock first stage divider	
PWM4_M	1:0	R/W	0	PWMx clock first stage divider	

Register::PWM01_N_MSB				0xFF3D	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PWM1H_N	7:4	R/W	0	PWMx clock Second stage divider MSB[11:8]	
PWM0H_N	3:0	R/W	0	PWMx clock Second stage divider MSB[11:8]	

Register::PWM0_N_LSB				0xFF3E	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PWM0L_N	7:0	R/W	0	PWMx clock Second stage divider LSB[7:0]	

Register::PWM1_N_LSB				0xFF3F	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PWM1L_N	7:0	R/W	0	PWMx clock Second stage divider LSB[7:0]	

Register::PWM23_N_MSB				0xFF40	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PWM3H_N	7:4	R/W	0	PWMx clock Second stage divider MSB[11:8]	
PWM2H_N	3:0	R/W	0	PWMx clock Second stage divider MSB[11:8]	

Register::PWM2_N_LSB				0xFF41	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PWM2L_N	7:0	R/W	0	PWMx clock Second stage divider LSB[7:0]	

Register::PWM3_N_LSB				0xFF42	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PWM3L_N	7:0	R/W	0	PWMx clock Second stage divider LSB[7:0]	

Register::PWM45_N_MSB				0xFF43	
Name	Bits	R/W	Default	Comments	Config
PWM4H_N	7:4	R/W	0	PWMx clock Second stage divider MSB[11:8]	
PWM5H_N	3:0	R/W	0	PWMx clock Second stage divider MSB[11:8]	

Register::PWM4_N_LSB				0xFF44	
Name	Bits	R/W	Default	Comments	Config
PWM4L_N	7:0	R/W	0	PWMx clock Second stage divider LSB[7:0]	

Register::PWM5_N_LSB				0xFF45	
Name	Bits	R/W	Default	Comments	Config
PWM5L_N	7:0	R/W	0	PWMx clock Second stage divider LSB[7:0]	

Register::PWML				0xFF46	
Name	Bits	R/W	Default	Comments	Config
PWM_W_DB_WR	7	R/W	0	Write 1 to Set PWM_Width if PWM_W_DB_EN = 1'b1. Auto-Clear after PWM_Width was loaded	Rport Wport
PWM_W_DB_MODE	6	R/W	0	PWM Width Setting Double-Buffer Mode 0: Setting active after PWM_W_DB_WR = 1 1: Setting active after PWM_W_DB_WR = 1 & DVS.	
PWM5L	5	R/W	0	0: enable Active H 1: enable Active L	
PWM4L	4	R/W	0	0: enable Active H 1: enable Active L	
PWM3L	3	R/W	0	0: enable Active H 1: enable Active L	
PWM2L	2	R/W	0	0: enable Active H 1: enable Active L	
PWM1L	1	R/W	0	0: enable Active H 1: enable Active L	
PWM0L	0	R/W	0	0: enable Active H 1: enable Active L	

Register::PWM_VS_CTRL				0xFF47	
Name	Bits	R/W	Default	Comments	Config
PWM_VS_CTRL_DUM	7:6	R/W	0	dummy	
PWM5_VS_RST_EN	5	R/W	0	0: Disable 1: Enable PWM5 reset by DVS	
PWM4_VS_RST_EN	4	R/W	0	0: Disable 1: Enable PWM4 reset by DVS	
PWM3_VS_RST_EN	3	R/W	0	0: Disable 1: Enable PWM3 reset by DVS	
PWM2_VS_RST_EN	2	R/W	0	0: Disable 1: Enable PWM2 reset by DVS	
PWM1_VS_RST_EN	1	R/W	0	0: Disable 1: Enable PWM1 reset by DVS	
PWM0_VS_RST_EN	0	R/W	0	0: Disable 1: Enable PWM0 reset by DVS	

Register::PWM_EN				0xFF48	
Name	Bits	R/W	Default	Comments	Config

PWM_W_DB_EN	7	R/W	0	0: PWM Width set when write MSB 1: PWM Width setting double-buffered enable	
PWM_WIDTH_SEL	6	R/W	0	0: PWMxL_DUT is active 1: PWMxL_DUT is inactive, forced to 4'h0 internally	
PWM5_EN	5	R/W	0	0: PWM output disable 1: PWM output enable	
PWM4_EN	4	R/W	0	0: PWM output disable 1: PWM output enable	
PWM3_EN	3	R/W	0	0: PWM output disable 1: PWM output enable	
PWM2_EN	2	R/W	0	0: PWM output disable 1: PWM output enable	
PWM1_EN	1	R/W	0	0: PWM output disable 1: PWM output enable	
PWM0_EN	0	R/W	0	0: PWM output disable 1: PWM output enable	

Register::PWM_CK				**0xFF49**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PWM_CK_DUMMY	7:6	R/W	0	Dummy	
PWM5_CK	5	R/W	0	0: Select first stage output 1: Select second stage output	
PWM4_CK	4	R/W	0	0: Select first stage output 1: Select second stage output	
PWM3_CK	3	R/W	0	0: Select first stage output 1: Select second stage output	
PWM2_CK	2	R/W	0	0: Select first stage output 1: Select second stage output	
PWM1_CK	1	R/W	0	0: Select first stage output 1: Select second stage output	
PWM0_CK	0	R/W	0	0: Select first stage output 1: Select second stage output	

Register::PWM0H_DUT				**0xFF4A**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PWM0H_DUT	7:0	R/W	0	PWM0[11:4] duty width When write the MSB bit (PWM_W_DB_EN=0) , the 12-bit data will be loaded.	Rport Wport

Register::PWM1H_DUT				**0xFF4B**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PWM1H_DUT	7:0	R/W	0	PWM1[11:4] duty width When write the MSB bit (PWM_W_DB_EN=0) , the 12-bit data will be loaded.	Rport Wport

Register::PWM01L_DUT				**0xFF4C**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PWM1L_DUT	7:4	R/W	0	PWM1[3:0] duty width	Rport Wport
PWM0L_DUT	3:0	R/W	0	PWM0[3:0] duty width	Rport Wport

Register::PWM2H_DUT				**0xFF4D**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PWM2H_DUT	7:0	R/W	0	PWM2[11:4] duty width When write the MSB bit (PWM_W_DB_EN=0) , the 12-bit data	Rport Wport

				will be loaded.	

Register::PWM3H_DUT				0xFF4E	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PWM3H_DUT	7:0	R/W	0	PWM3[11:4] duty width When write the MSB bit (PWM_W_DB_EN=0) , the 12-bit data will be loaded.	Rport Wport

Register::PWM23L_DUT				0xFF4F	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PWM3L_DUT	7:4	R/W	0	PWM3[3:0] duty width	Rport Wport
PWM2L_DUT	3:0	R/W	0	PWM2[3:0] duty width	Rport Wport

Register::PWM4H_DUT				0xFF50	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PWM4H_DUT	7:0	R/W	0	PWM4[11:4] duty width When write the MSB bit (PWM_W_DB_EN=0) , the 12-bit data will be loaded.	Rport Wport

Register::PWM5H_DUT				0xFF51	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PWM5H_DUT	7:0	R/W	0	PWM5[11:4] duty width When write the MSB bit (PWM_W_DB_EN=0) , the 12-bit data will be loaded.	Rport Wport

Register::PWM45L_DUT				0xFF52	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PWM5L_DUT	7:4	R/W	0	PWM5[3:0] duty width	Rport Wport
PWM4L_DUT	3:0	R/W	0	PWM4[3:0] duty width	Rport Wport

Register:: PWM_DUT_TYPE				0xFF53	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PWM_DUT_TYPE_DUMMY	7:6	R/W	0	Dummy	
PWM5_DUT_8B	5	R/W	0	PWM5 duty width type When PWM5_DUT_8B =1, only PWM5H_DUT is used as 8bit-resolution duty, and increase the PWM frequency x 16	
PWM4_DUT_8B	4	R/W	0	PWM4 duty width type When PWM4_DUT_8B =1, only PWM4H_DUT is used as 8bit-resolution duty, and increase the PWM frequency x 16	
PWM3_DUT_8B	3	R/W	0	PWM3 duty width type When PWM3_DUT_8B =1, only PWM3H_DUT is used as 8bit-resolution duty, and increase the PWM frequency x 16	
PWM2_DUT_8B	2	R/W	0	PWM2 duty width type When PWM2_DUT_8B =1, only PWM2H_DUT is used as 8bit-resolution duty, and increase the PWM frequency x 16	
PWM1_DUT_8B	1	R/W	0	PWM1 duty width type When PWM1_DUT_8B =1, only PWM1H_DUT is used	

				as 8bit-resolution duty, and increase the PWM frequency x 16	
PWM0_DUT_8B	0	R/W	0	PWM0 duty width type When PWM0_DUT_8B =1, only PWM0H_DUT is used as 8bit-resolution duty, and increase the PWM frequency x 16	

Register::PWM_CNT_MODE					**0xFF54**
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:6	--	0	Reserved	
PWM5_CNT_MODE	5	R/W	0	0: duty mode 1: high/total counter mode. The total level count is {PWMxH_TOTALCNT, PWMxL_TOTALCNT}, high level count is {PWMxH_DUT, PWMxL_DUT}, and limited H/L transition number by PWMx_ CYCLE_MAX	
PWM4_CNT_MODE	4	R/W	0	0: duty mode 1: high/total counter mode. The total level count is {PWMxH_TOTALCNT, PWMxL_TOTALCNT}, high level count is {PWMxH_DUT, PWMxL_DUT}, and limited H/L transition number by PWMx_ CYCLE_MAX	
PWM3_CNT_MODE	3	R/W	0	0: duty mode 1: high/total counter mode. The total level count is {PWMxH_TOTALCNT, PWMxL_TOTALCNT}, high level count is {PWMxH_DUT, PWMxL_DUT}, and limited H/L transition number by PWMx_ CYCLE_MAX	
PWM2_CNT_MODE	2	R/W	0	0: duty mode 1: high/total counter mode. The total level count is {PWMxH_TOTALCNT, PWMxL_TOTALCNT}, high level count is {PWMxH_DUT, PWMxL_DUT}, and limited H/L transition number by PWMx_ CYCLE_MAX	
PWM1_CNT_MODE	1	R/W	0	0: duty mode 1: high/total counter mode. The total level count is {PWMxH_TOTALCNT, PWMxL_TOTALCNT}, high level count is {PWMxH_DUT, PWMxL_DUT}, and limited H/L transition number by PWMx_ CYCLE_MAX	
PWM0_CNT_MODE	0	R/W	0	0: duty mode 1: high/total counter mode. The total level count is {PWMxH_TOTALCNT, PWMxL_TOTALCNT}, high level count is {PWMxH_DUT, PWMxL_DUT}, and limited H/L transition number by PWMx_ CYCLE_MAX	

I2C Control Module

RTD2485XD provides one I2C master interface only.

Master
In the Random read operation, the slave address and data are clocked in and acknowledged by the slave. The master will generate another start condition. In Current address read operation, the internal data word address counter maintains the data word address accessed during the last read or write operation, incremented by one. The data word address stays valid between operations as long as the power is

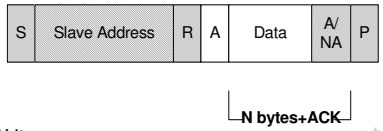
maintained. A Write operation requires data words following the slave address word and acknowledgment. The master terminates the write sequence with a stop condition. In Write with restart operation, master will generate another start condition after transmitting the slave address and data. If slave needs stop condition between data and restart command (Sr) in the Random read operation, software can transmit the Current address read operation following A Write operation. The maximum value of data FIFO (N) is 24. The slave address byte will be written into FIFO register together with data. Software must write the eighth bit of slave address byte to decide the access is read or write.

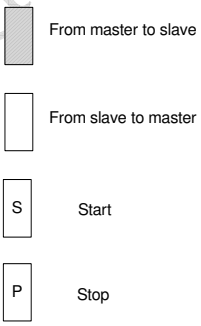
NOTE:
(a) RTD2485XD ~~only~~ supports master and slave function
(b) RTD2485XD doesn't support arbitration mechanism while one system exists over 2 masters
(c) FIFO can't support R/W at the same time. It means that you can't transmit data successively while the byte counts over 24, since MCU have to refresh the FIFO data for the next transmit
~~(d) RTD2485XD only supports HDMI Rx, so SBAIFD & FBAIFD function will not be used for it~~
(e) Master supports the function that slave can hold SCL to zero after ACK when slave can't give master data that master wanted.

Signal Name	Type	Function	Note
SCL	O		
SDA	I/O		

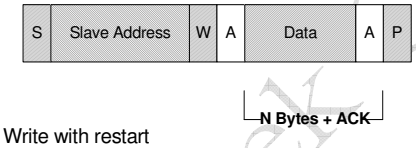
Random Read

Current Address Read

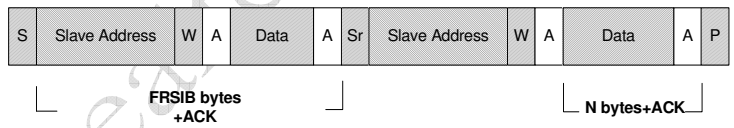




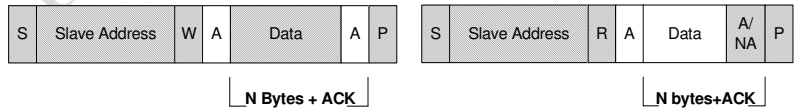
Write



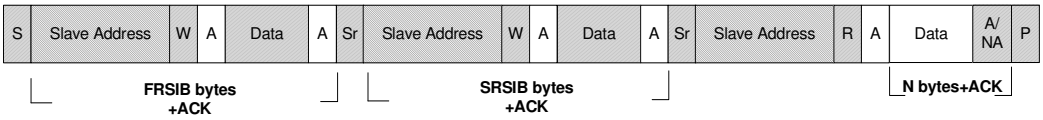
Write with restart



Write + Current Address Read



Read with two restart



HDMI 1.3

HDMI 1.1

So we can identify whether HMDI 1.1 or 1.3 by receiving the first 2 ACK from slaves

Register:: I2CM_CR0				0xFF55	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
IICM_SW_RSTN	7	R/W	0	IIC master software reset 0: Reset, Blocking IICM module 1: Enable IICM module	
CS	6	R/W	0	Command Start 0 = Stop, after completing whole transaction, it returns to zero 1 = Start	wport rport
RWL	5:1	R/W	0	Read/Write data Length for related commands. Not includes slave address byte in FIFO register. When access, controller will parse the byte followed last start (or Sr) byte to know the command type. 0x00 = 1 bytes 0x17 = 24 bytes	
TORE	0	R/W	1	TOR enable If TOR is desired, I2C rate must be constrained form 25kb/s~ 400kb/s. This constraint is due to the time-out register bit.	

Register:: I2CM_CR1				0xFF56	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
TOR	7:0	R/W	0x3A	Time-out register Time-out = TOR x 2 x ((FD10+1)/input clock) (For receive /transmit one bit) If time-out occur, it will trigger Transaction Error Interrupt Flag Note: time-out must > (1 SCL low period + repeat start setup time)	

Register:: I2CM_CR2				0xFF57	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
reserved	7	-	0	Reserved to 0	
BURST	6	R/W	0	Burst mode enabled to write over 24 bytes to slave devices. While burst is enabled, whole I2C will be halted to let MCU write another more bytes to FIFO. After the job done, we have to set I2CM_SR to be high to continue the I2C write job	
SBAIFD	5	R/W	0	Second byte ACK in FRSIB data (for identifying HDMI 1.3 needed) SBAIFD indicates that whether master checks ACK from slave or not after emitting second data in FRSIB data. 0: To check 1: Not check	
FBAIFD	4	R/W	0	First byte ACK in FRSIB data (for identifying HDMI 1.3 needed) FBAIFD indicates that whether master checks ACK from slave or not after emitting first data in FRSIB data. 0: To check	

				1: Not check	
SRSIB	3:2	R/W	0	Second repeat start interval byte (For HDMI need) After transmitting SRSIB bytes that, include slave address & data bytes, follow the first repeat start command, the master will produce second repeat start command. The slave address or device address byte is included in this interval. Default interval is one byte. 0 = 1 bytes; 1=2 byte etc. Note: The eighth bit of slave address or device address byte followed by second repeat start command must be Read.	
FRSIB	1:0	R/W	0	First repeat start interval byte (For HDMI need) After transmitting FRSIB bytes that, include slave address & data bytes, follow the original start command, the master will produce first repeat start command. The original slave address or device address byte is included in this interval. Default interval is one byte. 0 = 1 bytes; 1=2 bytes, etc.	

Register:: I2CM_CR3				**0xFF58**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
reserved	7:5	-	0	Reserved to 0	
RSC	4:3	R/W	0	Repeat start count 00: No repeat start 01: one repeat start 10: two repeat start 11: forbidden	
TEIE	2	R/W	0	Transaction Error Interrupt Enable	
MRCIE	1	R/W	0	Master Receive complete Interrupt Enable	
MTCIE	0	R/W	0	**This Interrupt enable bit serve two conditions, one is "Master Transmit complete Interrupt Enable in single mode" and the other is "Master Transmit partial Done Interrupt Enable in burst mode"**	

Register:: I2CM_STR0				**0xFF59**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
reserved	7	-	0	Reserved to 0	
I2CMD	6:4	R/W	0	I2C master debounce 0: sample rate=(input clk / (FD10+1)) 1: sample rate=(input clk / (FD10+1)) / 2 : 7: sample rate= (input clk / (FD10+1)) / 8	
FTPC	3:0	R/W	0x3	Fall time period count If the value of (Bus clock/FD10) does not approximate 10Mhz, FTPC can make sure that fall time of SCL is more than 300ns.	

Register:: I2CM_STR1				**0xFF5A**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
STA_SUGPIO_C	7:0	R/W	0x09	STA setup time period count In repeat start, the setup time of SCL must match the I2C spec.	

Register:: I2CM_STR2				**0xFF5B**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
SHPC	7:0	R/W	0x09	SCL high period counter (SCL High period=100ns*SHPC) SHPC must include rising time in the I2C spec.	

Register:: I2CM_STR3				**0xFF5C**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**

SLPC	7:0	R/W	0x10	SCL low period counter (SCL low period=100ns*SLPC) SLPC must include falling time in the I2C spec.	

Register:: I2CM_SR				**0xFF5D**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
reserved	7:5	-	0	Reserved to 0	
BSA	4	R/W	0	**Send again Control bit in burst mode** 1: Send 0: Not action Write "1" to clear	wport rport
BMPIF	3	R/W	0	**Burst Mode Pending Interrupt Flag** While setting as burst mode and I2C master transmit data in **TDD**, this flag is asserted. Write "1" to clear	wport rport
TEIF	2	R/W	0	**Transaction Error Interrupt Flag** When master transmit/receive fault or time-out occurrence, I2C controller will lift the flag up and return to bus idle. Write "1" to clear	wport rport
MRCIF	1	R/W	0	**Master Receive complete Interrupt flag** Write "1" to clear	wport rport
MTCIF	0	R/W	0	This Interrupt enable bit serves two conditions, one is "Master Transmit complete Interrupt Enable in single mode" and the other is "Master Transmit partial Done Interrupt Enable in burst mode" Write "1" to clear	wport rport

Register:: I2CM_TD				**0xFF5E**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
TDD	7:0	R/W	0	Target Device Data to receive or transmit	wport rport

Register:: I2CM_CCR				**0xFF5F**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
reserved	7:6	-	0	Reserved to 0	
FD10	5:0	R/W	0x01	Frequency 10M Divisor 0 are forbidden 10M=input clock/(FD10+1) When power on, software must write FD10 to let I2C controller generate ~10 Mhz clock.	

MCU Resister2(Page E) SPI-FLASH

Common Instruction Register

Register::common_inst_en			0xFF60		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
com_inst	7:5	R/W	0x0	000: no operation 001 : write 010 : Read 011 : write after WREN 100 : write after EWSR 101 : Erase	
write_num	4:3	R/W	0x0	Common instruction write number	
rd_num	2:1	R/W	0x0	Common instruction read number	
com_inst_en	0	R/W	0x0	Common instruction enable (auto clear when finish)	Wport Rport

Register::common_op_code			0xFF61		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
com_op	7:0	R/W	0x0	Common instruction op code	

Register::wren_op_code			0xFF62		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
wren_op	7:0	R/W	0x06	Write enable op code	

Register::ewsr_op_code			0xFF63		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ewsr_op	7:0	R/W	0x50	Enable write register op code	

Register::Flash_prog_ISP0			0xFF64		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
prog_h	7:0	R/W	0x00	Flash program/write/dummy/read CRC high byte[23:16]	Wport Rport

Register::Flash_prog_ISP1			0xFF65		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
prog_m	7:0	R/W	0x00	Flash program/write/dummy/read CRC middle byte[15:8]	Wport Rport

Register::Flash_prog_ISP2			0xFF66		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
prog_l	7:0	R/W	0x00	Flash program/write/dummy/read CRC low byte[7:0]	Wport Rport

Register::common_inst_read_port0			0xFF67		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
com_rd_h	7:0	R	0x00	Common instruction read high byte[23:16]	

Register::common_inst_read_port1			0xFF68		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**

com_rd_m	7:0	R	0x00	Common instruction read middle byte[15:8]	

Register::common_inst_read_port2			0xFF69		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
com_rd_l	7:0	R	0x00	Common instruction read low byte[7:0]	

Common Instruction Usage :

1. Set common instruction type.
2. Set common instruction OP code.
3. Set write number (0 ~ 3).
4. Set read number if common instruction type is read.
5. Write data to **Flash_prog_write_dum_readCRC_ISP** if write number > 0 .
6. Execution common instruction enable.
7. Polling common instruction enable in ISP mode → If it is finished and the instruction is read, read the Data in **common_inst_read_port .**
8. It would auto clear in normal mode. Then read the Data in **common_inst_read_port** if the instruction type is read.

Common Instruction Setting Example :

Write function

	com_op	write_num=0			rd_num =0		
WREN	06h						
WRDI	04h						
EWSR	50h						
DP	B9h						
RDP	ABh						

Read function

write_num =0 , rd_num=3

	com_op	rd_num setting					
RDID	9Fh	ID23-ID16	ID15-ID8	ID7-ID0			
JEDEC ID READ*1	9Fh	ID2	ID1	Device ID			

write_num=0, rd_num=1

RDCR	A1h	RD_CR					
RDSR	05h	RD_SR					

write_num=3, rd_num=1 or 2

RES	ABh	DUMMY	DUMMY	DUMMY	Electronic Signature		
REMS	90h	DUMMY	DUMMY	00h	ID	Device ID	

Write after WREN function

	com_op	write_num=1	rd_num=0				
WRSR	01h	WR_SR					

Write after WRSR function

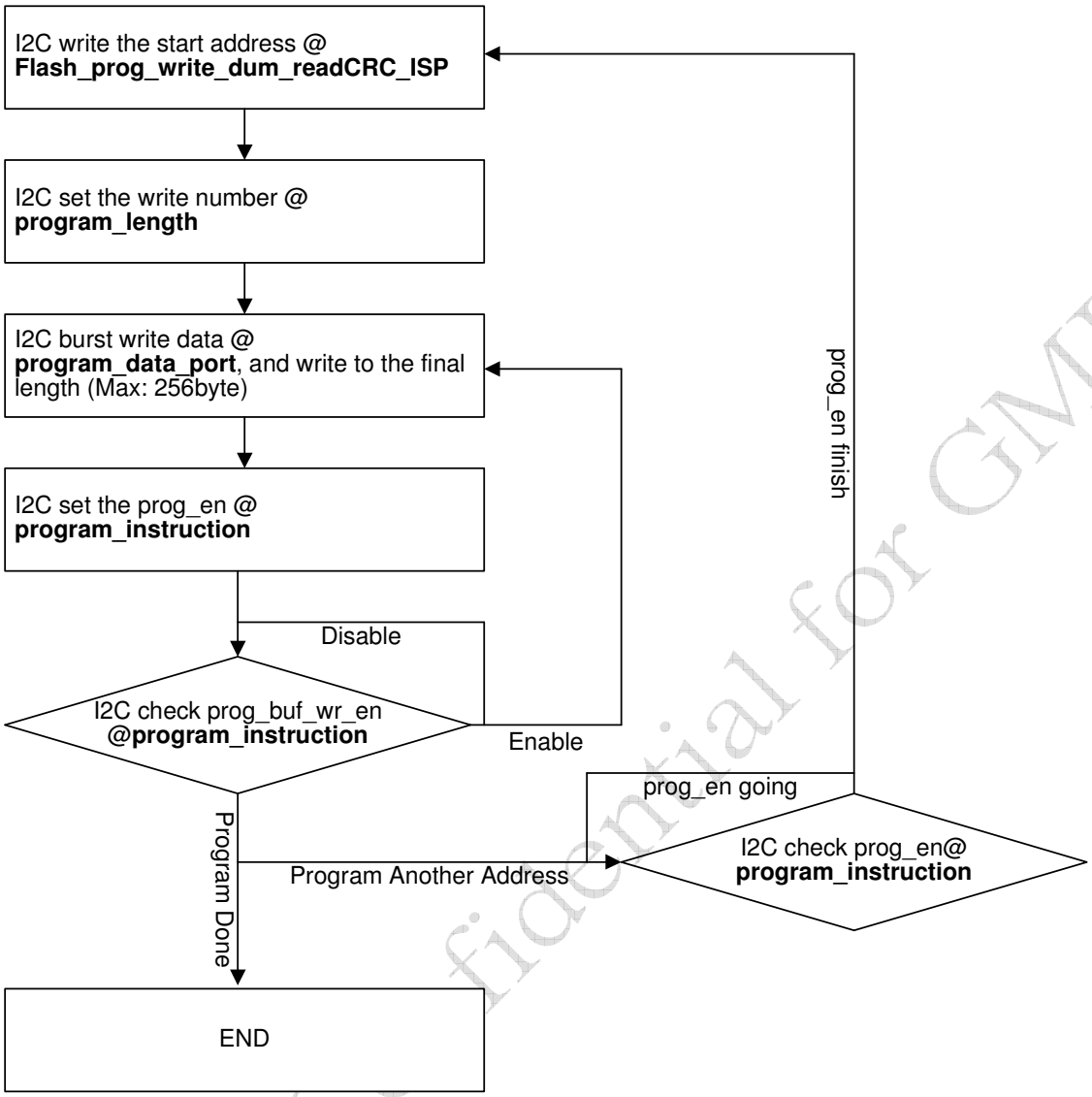
	com_op	write_num=1	rd_num=0				
WRCR	F1h	WR_CR					

Erase function

	com_op	write_num = 0 or 3			rd_num=0		
SECTOR_ER (4k byte)	20h	AD1	AD2	AD3			
BLOCK_ER (64k byte)	D8h	AD1	AD2	AD3			
Page Erase	DBh	AD1	AD2	AD3			
CHIP_ER	C7h						

Program/Read/ISP/CRC Register

ISP Protocol of Program Data :



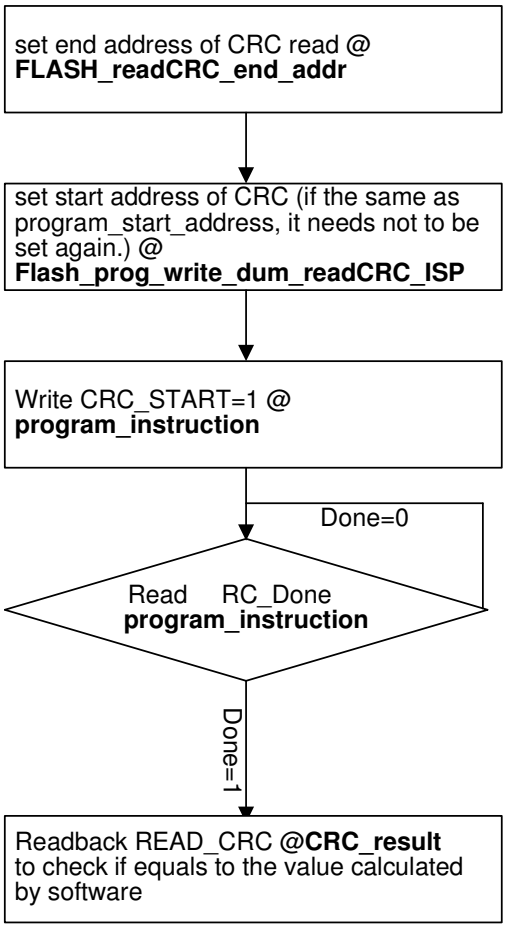
In SPI Mode

1. **We use two 256 bytes buffer for Program.**
2. **We write buffer by DDC Channel.**
3. **We auto switch two buffer during program.**
   → If prog_buf_wr_en = 1, you can keep write data to buffer.

In Normal mode

1. **We use one 256 bytes buffer for Program.**
2. **We write buffer by uP.**
3. **We only use one buffer during program.**
   → We can program Max. 256 bytes one time.

Protocol of CRC of read Data :



1. Only can be used in ISP Mode.
2. When CRC is done, the start address is the same as end address

Register::read_op_code			0xFF6A		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
read_op	7:0	R/W	0x03	Read command op code	Wen_out

Note : 1 It would force flash controller to idle state when write this byte.

Register::fast_read_op_code			0xFF6B		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Fast_read_op	7:0	R/W	0x0B	Fast read command op code Include Fast dual read op code & fast read dual input output op code	Wen_out

Note : 1 It would force flash controller to idle state when write this byte.

Register::read_instruction			0xFF6C		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
read_mode	7:6	R/W	0x0	00: normal read 01: fast read 10: fast dual data read 11: fast dual input and dual output	Wen_out
latch_so_rise	5	R/W	0x0	0: latch Flash SO Data in rising edge 1: latch Flash SO Data in falling edge	Wen_out
drive_si_fall	4	R/W	0x0	0: Output Flash SI Data in falling edge 1: Output Flash SI Data in rising edge	Wen_out
si_dly_sel	3:2	R/W	0x0	00: 0ns 01: 2ns 10: 4ns 11: 6ns	Wen_out

Reserved	1:0	R/W	0x0	00: 0ns 01: Reserved 10: Reserved 11: Reserved	Wen_out

Note: 1. Normally, SPI Flash drive data in falling edge and sample data in rising edge !!!!
2. It would force flash controller to idle state when write this byte.

Register::program_op_code				**0xFF6D**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
prog_op	7:0	R/W	0xff	Program command op code	

Register::read_status_register_op_code				**0xFF6E**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
rdsr_op	7:0	R/W	0x05	Read status register register command op code	

Register::program_instruction				**0xFF6F**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
isp_en	7	R/W	0x0	Enable ISP program : all registers except this register can't write/read when ISP_ENABLE=0 0: disable 1: enable (gating 8051 clock)	Rport Wport
prog_mode	6	R/W	0x0	0: normal mode (other select reference 0xFF6F[3]) 1: AAI mode	
prog_en	5	R/W	0x0	0: finish 1: program on-going (write 1 to start, auto clear when finish)	Rport Wport
prog_buf_wr_en	4	R	0x1	0: can't write data to sram 1: can write data to sram	
Prog_normal_mode	3	R/W	0x0	Program normal mode select 0: page program (1byte OP code + 3byte address + N byte data) 1: single byte program , hardware implement single byte write (1Byte OP + 3Byte Addr + Data0, 1Byte OP + 3Byte Addr + Data1, …)	
crc_start	2	R/W	0x0	When write one, read data from PROG_ST_ADDR to PROG_END_ADDR. And at the same time the CRC is calculated by IC automatically. This bit will be auto cleared when crc_done is 1.(Can be trigger in ISP Mode only)	Rport Wport
crc_done	1	R	0x1	It will show 1 when CRC is done, and will return to 0 when CRC_START is set.	
rst_flash_ctrl	0	R/W	0x0	0: disable 1: software reset flash controller	Rport Wport

No Use Parser::program_data_port				**0xFF70**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
prog_port	7:0	W		Program write data port to SRAM	

Register::program_length				**0xFF71**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**

prog_length	7:0	R/W	0xFF	Program write number	

Register::CRC_end_addr0			0xFF72		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
rdcrc_end_addr_h	7:0	R/W	0x00	Read CRC end address high byte [23:16]	

Register::CRC_end_addr1			0xFF73		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
rdcrc_end_addr_m	7:0	R/W	0x00	Read CRC end address middle byte [15:8]	

Register::CRC_end_addr2			0xFF74		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
rdcrc_end_addr_l	7:0	R/W	0x00	Read CRC end address low byte [7:0]	

Register::CRC_result			0xFF75		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
crc_result	7:0	R	-	CRC value of data between PROG_ST_ADDR and RDCRC_END_ADDR	

Flash timing Register

Register::cen_ctrl			0xFF76		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
cen_high_num	7:4	R/W	0x9	Chip enable high number[3:0] (based on flash clock) Cycle : 1 ~ 16 (0x0 ~ 0xF)	Wen_out
cen_setup_num	3:2	R/W	0x0	Chip enable setup number (based on flash clock) Cycle : 0.5 ~ 3.5 Cycle : 1 ~ 4 (if drive_si_fall)	Wen_out
cen_hold_num	1:0	R/W	0x0	Chip enable hold number (based on flash clock) Cycle : 1 ~ 4 Cycle : 0.5 ~ 3.5 (if drive_si_fall)	Wen_out

Note : 1 It would force flash controller to idle state when write this byte.

Register::AAI_Mode_Byte_Num			0xFF80		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
AAI_Mode_Byte_Num	7:0	R/W	0x1	SST flash, AAI mode, transfer number 0x00: transfer 1 byte 0x01: transfer 1byte 0x02: transfer 2bytes and so on	

Register::ISP_CMD_INSERT			0xFF81		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7	R/W	0	Reserved	
Force_CE_low	6	R/W	0	0: Normal 1: Force CE low between Common instructions (Only effective in burst write from XRAM, between common write/read instructions)	
ISP_CMD_WAIT_RESO_M0	5:4	R/W	0	cmd wait time resolution, m0 divider divide value = 2^{m0} 00: divide 1	

				01: divide 2 10: divide 4 11: divide 8	
ISP_CMD_WAIT_RESO_M1	3:1	R/W	0	cmd wait time resolution, m1 divider divide value = 2^{m1}	
ISP_CMD_INSERT_EN	0	R/W	0	Enable insert CMD before isp program 0: disable 1: enable	

Register::ISP_CMD_LENGTH			**0xFF82**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ISP_CMD_LENGTH	7:0	R/W	0xFF	Totoal max length of ISP CMD: 0x00: length=1 0x01: length=2 … 0xFF: length=256	

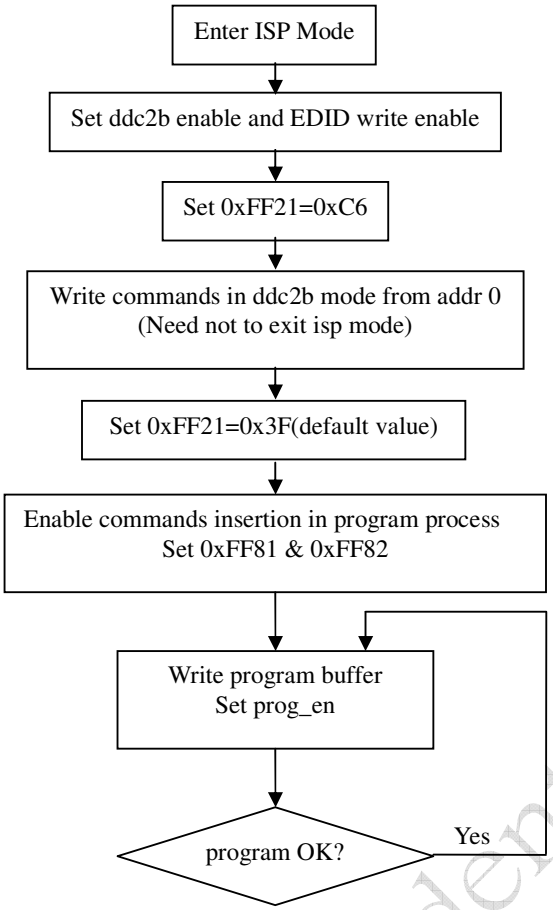
ISP Command Format:

cmd_length(=1+n1) + **cmd_wait_length** + **cmd_opcode** + **n1 byte data**
cmd_length(=1+n2) + **cmd_wait_length** + **cmd_opcode** + **n2 byte data**
…
cmd_length=0 (end)

wait time = cmd_wait_length * 2^{m1} * 2^{m0} * (flash_clk period)
wait time is inserted after current command execution

Example:

01 00 06 → opcode=0x06, no data, no wait time
02 80 01 7C → opcode=0x01, data=0x7C, wait time=128* 2^{m1} * 2^{m0} * (flash_clk period)
01 00 C3 → opcode=0xC3, no data, no wait time
01 00 A5 → opcode=0xA5, no data, no wait time
01 00 C3 → opcode=0xC3, no data, no wait time
01 00 A5 → opcode=0xA5, no data, no wait time
01 00 06 → opcode=0x06, no data, no wait time
02 FF 01 00 → opcode=0x01, data=0x00, wait time=255* 2^{m1} * 2^{m0} * (flash_clk period)
00 →end
To execute the above 8 commands, you need to write 27 bytes to RAM at one time.



ISP command insertion program flow (e.g. for MXIC new flash)

100Mhz SPI Flash Calibration

Register::SPI_CAL_Ctrl			***0xFF9D***		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DLY_CAL_START	7	R/W	0	Write 1 to start SPI calibration. Auto clear when SPI calibration has been completed.	Rport Wport
DLY_CAL_CHECK	6	R/W	1	0: Not check delay out-of-range 1: Check delay out-of-range (> T/2)	
CAL_CLK_DIV	5:4	R/W	0x2	SPI-FLASH clock divider, its clock source is selected by MCU_CLK_SEL, active when DLY_CAL_START 0: divide 1 1: divide 2 2: divide 3 3: divide 4	
DLY_SEL	3:0	R/W	0	Delay selection: (DLY_HI_THR + DLY_LO_THR)/2 reference cali result 0xFF9E 0000: 0.0ns 0001: 0.5ns 0010: 1.0ns 0011: 1.5ns 0100: 2.0ns … 1111: 7.5ns	

Register::SPI_CAL_Result			***0xFF9E***		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
DLY_HI_THR	7:4	R	0	Calibration pass high threshold	

DLY_LO_THR	3:0	R	0	Calibration pass low threshold	

Register::SPI_CAL_Data_00				***0xFF9F***	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
CAL_DATA0	7:0	R/W	0	Calibration golden data0	

Register::SPI_CAL_Data_01				***0xFFA5***	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
CAL_DATA1	7:0	R/W	0	Calibration golden data1	

Register::SPI_CAL_Data_02				***0xFFA6***	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
CAL_DATA2	7:0	R/W	0	Calibration golden data2	

Register::SPI_CAL_ADR_H				***0xFFA7***	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
CAL_ADR_H	7:0	R/W	0	Calibration flash start address[23:16]	

Register::SPI_CAL_ADR_M				***0xFFA8***	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
CAL_ADR_M	7:0	R/W	0	Calibration flash start address[15:8]	

Register::SPI_CAL_ADR_L				***0xFFA9***	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
CAL_ADR_L	7:0	R/W	0	Calibration flash start address[7:0]	

MCU Register2 (Page E) PWM

Register::PWM01_TOTALCNT_MSB				0xFF90	
Name	Bits	R/W	Default	Comments	Config
PWM1H_TOTALCNT	7:4	R/W	0	PWMx , total level count MSB[11:8]	wport rport
PWM0H_TOTALCNT	3:0	R/W	0	PWMx , total level count MSB[11:8]	wport rport

Register::PWM0_TOTALCNT_LSB				0xFF91	
Name	Bits	R/W	Default	Comments	Config
PWM0L_TOTALCNT	7:0	R/W	0	PWMx , total level count LSB[7:0]	wport rport

Register::PWM1_TOTALCNT_LSB				0xFF92	
Name	Bits	R/W	Default	Comments	Config
PWM1L_TOTALCNT	7:0	R/W	0	PWMx , total level count LSB[7:0]	wport rport

Register::PWM23_TOTALCNT_MSB				0xFF93	
Name	Bits	R/W	Default	Comments	Config
PWM3H_TOTALCNT	7:4	R/W	0	PWMx , total level count MSB[11:8]	wport rport
PWM2H_TOTALCNT	3:0	R/W	0	PWMx , total level count MSB[11:8]	wport rport

Register::PWM2_TOTALCNT_LSB				0xFF94	
Name	Bits	R/W	Default	Comments	Config
PWM2L_TOTALCNT	7:0	R/W	0	PWMx , total level count LSB[7:0]	wport rport

Register::PWM3_TOTALCNT_LSB				0xFF95	
Name	Bits	R/W	Default	Comments	Config
PWM3L_TOTALCNT	7:0	R/W	0	PWMx , total level count LSB[7:0]	wport rport

Register::PWM45_TOTALCNT_MSB				0xFF96	
Name	Bits	R/W	Default	Comments	Config
PWM5H_TOTALCNT	7:4	R/W	0	PWMx , total level count MSB[11:8]	wport rport
PWM4H_TOTALCNT	3:0	R/W	0	PWMx , total level count MSB[11:8]	wport rport

Register::PWM4_TOTALCNT_LSB				0xFF97	
Name	Bits	R/W	Default	Comments	Config
PWM4L_TOTALCNT	7:0	R/W	0	PWMx , total level count LSB[7:0]	wport rport

Register::PWM5_TOTALCNT_LSB				0xFF98	
Name	Bits	R/W	Default	Comments	Config
PWM5L_TOTALCNT	7:0	R/W	0	PWMx , total level count LSB[7:0]	wport rport

Register::PWM_CK_SEL_HS				0xFF99	
Name	Bits	R/W	Default	Comments	Config
PWM_CK_SEL_HS_DUMMY	7:6	R/W	0	Dummy	
PWM5_CK_SEL_HS	5	R/W	0	PWMx clock generator input source 0: reference PWM5_CK_SEL 1: using DHS as reference clock The total level count is {PWM5H_TOTALCNT, PWM5L_TOTALCNT}, high level count is {PWM5H_DUT, PWM5L_DUT}, and limited H/L transition number by PWM5_ CYCLE_MAX	

PWM4_CK_SEL_HS	4	R/W	0	PWMx clock generator input source 0: reference PWM4_CK_SEL 1: using DHS as reference clock The total level count is {PWM4H_ TOTALCNT, PWM4L_ TOTALCNT }, high level count is {PWM4H_DUT, PWM4L_DUT}, and limited H/L transition number by PWM4_ CYCLE_MAX	
PWM3_CK_SEL_HS	3	R/W	0	PWMx clock generator input source 0: reference PWM3_CK_SEL 1: using DHS as reference clock The total level count is {PWM3H_ TOTALCNT, PWM3L_ TOTALCNT }, high level count is {PWM3H_DUT, PWM3L_DUT}, and H/L transition number is limited by PWM3_ CYCLE_MAX	
PWM2_CK_SEL_HS	2	R/W	0	PWMx clock generator input source 0: reference PWM2_CK_SEL 1: using DHS as reference clock The total level count is {PWM2H_ TOTALCNT, PWM2L_ TOTALCNT }, high level count is {PWM2H_DUT, PWM2L_DUT}, and H/L transition number is limited by PWM2_ CYCLE_MAX	
PWM1_CK_SEL_HS	1	R/W	0	PWMx clock generator input source 0: reference PWM1_CK_SEL 1: using DHS as reference clock The total level count is {PWM1H_ TOTALCNT , PWM1L_ TOTALCNT }, high level count is {PWM1H_DUT, PWM1L_DUT}, and H/L transition number is limited by PWM1_ CYCLE_MAX	
PWM0_CK_SEL_HS	0	R/W	0	PWMx clock generator input source 0: reference PWM0_CK_SEL 1: using DHS as reference clock The total level count is {PWM0H_ TOTALCNT, PWM0L_ TOTALCNT }, high level count is {PWM0H_DUT, PWM0L_DUT}, and H/L transition number is limited by PWM0_CYCLE_MAX	

Register::PWM01_CYCLE_MAX				**0xFF9A**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PWM0_ CYCLE_MAX	7:4	R/W	0	PWM0 maximum H/L cycle count Output PWM0 clock cycle number between two DVS. When set to 0, there is no limitation. ~~This setting can only make effect by using DHS as reference clock~~	
PWM1_CYCLE_MAX	3:0	R/W	0	PWM1 maximum H/L cycle count Output PWM1 clock cycle number between two DVS. When set to 0, there is no limitation. This setting can only make effect by using DHS as reference clock	

Register::PWM23_CYCLE_MAX				**0xFF9B**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PWM2_ CYCLE_MAX	7:4	R/W	0	PWM2 maximum H/L cycle count Output PWM2 clock cycle number between two DVS. When set to 0, there is no limitation. ~~This setting can only make effect by using DHS as reference clock~~	
PWM3_CYCLE_MAX	3:0	R/W	0	PWM3 maximum H/L cycle count Output PWM3 clock cycle number between two DVS. When set to 0, there is no limitation. ~~This setting can only make effect by using DHS as reference clock~~	

Register::PWM45_CYCLE_MAX				**0xFF9C**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**

PWM4_ CYCLE_MAX	7:4	R/W	0	PWM4 maximum H/L cycle count Output PWM4 clock cycle number between two DVS. When set to 0, there is no limitation. ~~This setting can only make effect by using DHS as reference clock~~	
PWM5_CYCLE_MAX	3:0	R/W	0	PWM5 maximum H/L cycle count Output PWM5 clock cycle number between two DVS. When set to 0, there is no limitation. ~~This setting can only make effect by using DHS as reference clock~~	

MCU register 3 (page F)
GPIO Control

Register::PortD7_pin_reg					0xFF77
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
PD7	0	R/W	1	Input/output value of PD.7	Rport wport

Register::PortD6_pin_reg					0xFF78
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
PD6	0	R/W	1	Input/output value of PD.6	Rport wport

Register::PortD5_pin_reg					0xFF79
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
PD5	0	R/W	1	Input/output value of PD.5	Rport wport

Register::PortD4_pin_reg					0xFF7A
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
PD4	0	R/W	1	Input/output value of PD.4	Rport wport

Register::PortD3_pin_reg					0xFF7B
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
PD3	0	R/W	1	Input/output value of PD.3	Rport wport

Register::PortD2_pin_reg					0xFF7C
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	

PD2	0	R/W	1	Input/output value of PD.2	Rport wport

Register::PortD1_pin_reg **0xFF7D**

Name	Bits	R/W	Default	Comments	Config
Reserved	7:1	--	0	Reserved	
PD1	0	R/W	1	Input/output value of PD.1	Rport wport

Register::PortD0_pin_reg **0xFF7E**

Name	Bits	R/W	Default	Comments	Config
Reserved	7:1	--	0	Reserved	
PD0	0	R/W	1	Input/output value of PD.0	Rport wport

Register::PortB7_pin_reg **0xFF7F**

Name	Bits	R/W	Default	Comments	Config
Reserved	7:1	--	0	Reserved	
PB7	0	R/W	1	Input/output value of PB.7	Rport wport

Register::PortB6_pin_reg **0xFF89**

Name	Bits	R/W	Default	Comments	Config
Reserved	7:1	--	0	Reserved	
PB6	0	R/W	1	Input/output value of PB.6	Rport wport

Register::PortB5_pin_reg **0xFF8A**

Name	Bits	R/W	Default	Comments	Config
Reserved	7:1	--	0	Reserved	
PB5	0	R/W	1	Input/output value of PB.5	Rport wport

Register::PortB4_pin_reg **0xFF8B**

Name	Bits	R/W	Default	Comments	Config
Reserved	7:1	--	0	Reserved	

PB4	0	R/W	1	Input/output value of PB.4	Rport wport

Register::PortB3_pin_reg				**0xFF8C**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
PB3	0	R/W	1	Input/output value of PB.3	Rport wport

Register::PortB2_pin_reg				**0xFF8D**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
PB2	0	R/W	1	Input/output value of PB.2	Rport wport

Register::PortB1_pin_reg				**0xFF8E**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
PB1	0	R/W	1	Input/output value of PB.1	Rport wport

Register::PortB0_pin_reg				**0xFF8F**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
PB0	0	R/W	1	Input/output value of PB.0	Rport wport

Register::PortC3_pin_reg				**0xFFA0**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
PC3	0	R/W	1	Input/output value of PC.3	Rport wport

Register::PortC2_pin_reg				**0xFFA1**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	

PC2	0	R/W	1	Input/output value of PC.2	Rport wport

Register::PortC1_pin_reg				**0xFFA2**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
PC1	0	R/W	1	Input/output value of PC.1	Rport wport

Register::PortC0_pin_reg				**0xFFA3**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
PC0	0	R/W	1	Input/output value of PC.0	Rport wport

Register::PortC4_pin_reg				**0xFFA4**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
PC4	0	R/W	1	Input/output value of PC.4	Rport wport

Register::Port_read_control				**0xFFC0**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PA_pin_reg_n	7	R/W	0	Source selection for PA read back 0: register 1: bus value	
P9_pin_reg_n	6	R/W	0	Source selection for P9 read back 0: register 1: bus value	
P8_pin_reg_n	5	R/W	0	Source selection for P8 read back 0: register 1: bus value	
P7_pin_reg_n	4	R/W	0	Source selection for P7 read back 0: register 1: bus value	
P6_pin_reg_n	3	R/W	0	Source selection for P6 read back 0: register 1: bus value	
P5_pin_reg_n	2	R/W	0	Source selection for P5 read back 0: register 1: bus value	
P3_pin_reg_n	1	R/W	0	Source selection for P3 read back* 0: register 1: bus value	
P1_pin_reg_n	0	R/W	0	Source selection for P1 read back* 0: register 1: bus value	

*: only effect in external MCU control, embedded MCU will control the source by assembly code.

Register::Port52_pin_reg			0xFFC1		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P52	0	R/W	1	Input/output value of P5.2	Rport wport

Register::Port53_pin_reg			0xFFC2		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P53	0	R/W	1	Input/output value of P5.3	Rport wport

Register::Port54_pin_reg			0xFFC3		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P54	0	R/W	1	Input/output value of P5.4	Rport wport

Register::Port55_pin_reg			0xFFC4		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P55	0	R/W	1	Input/output value of P5.5	Rport wport

Register::Port56_pin_reg			0xFFC5		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P56	0	R/W	1	Input/output value of P5.6	Rport wport

Register::Port57_pin_reg			0xFFC6		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P57	0	R/W	1	Input/output value of P5.7	Rport wport

Register::Port60_pin_reg			0xFFC7		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P60	0	R/W	1	Input/output value of P6.0	Rport wport

Register::Port61_pin_reg			0xFFC8		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P61	0	R/W	1	Input/output value of P6.1	Rport wport

Register::Port62_pin_reg			0xFFC9		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P62	0	R/W	1	Input/output value of P6.2	Rport wport

Register::Port63_pin_reg			0xFFCA		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P63	0	R/W	1	Input/output value of P6.3	Rport wport

Register::Port64_pin_reg			0xFFCB		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P64	0	R/W	1	Input/output value of P6.4	Rport wport

Register::Port65_pin_reg			0xFFCC		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P65	0	R/W	1	Input/output value of P6.5	Rport wport

Register::Port66_pin_reg			0xFFCD		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P66	0	R/W	1	Input/output value of P6.6	Rport wport

Register::Port67_pin_reg			0xFFCE		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P67	0	R/W	1	Input/output value of P6.7	Rport wport

Register::Port70_pin_reg			0xFFCF		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P70	0	R/W	1	Input/output value of P7.0	Rport wport

Register::Port71_pin_reg			0XFFD0		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P71	0	R/W	1	Input/output value of P7.1	Rport wport

Register::Port72_pin_reg			0xFFD1		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P72	0	R/W	1	Input/output value of P7.2	Rport wport

Register::Port73_pin_reg			0xFFD2		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P73	0	R/W	1	Input/output value of P7.3	Rport wport

Register::Port74_pin_reg			0xFFD3		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P74	0	R/W	1	Input/output value of P7.4	Rport wport

Register::Port75_pin_reg			0xFFD4		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P75	0	R/W	1	Input/output value of P7.5	Rport wport

Register::Port76_pin_reg			0xFFD5		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P76	0	R/W	1	Input/output value of P7.6	Rport wport

Register::Port80_pin_reg			0xFFD6		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P80	0	R/W	1	Input/output value of P8.0	Rport wport

Register::Port81_pin_reg			0xFFD7		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P81	0	R/W	1	Input/output value of P8.1	Rport wport

Register::Port90_pin_reg			0xFFD8		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P90	0	R/W	1	Input/output value of P9.0	Rport wport

Register::Port91_pin_reg			0xFFD9		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**

Reserved	7:1	--	0	Reserved	
P91	0	R/W	1	Input/output value of P9.1	Rport wport

Register::Port92_pin_reg			**0xFFDA**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P92	0	R/W	1	Input/output value of P9.2	Rport wport

Register::Port93_pin_reg			**0xFFDB**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P93	0	R/W	1	Input/output value of P9.3	Rport wport

Register::Port94_pin_reg			**0xFFDC**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
P94	0	R/W	1	Input/output value of P9.4	Rport wport

Register::Porta0_pin_reg			**0xFFDD**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
PA0	0	R/W	1	Input/output value of PA.0	Rport wport

Register::Porta1_pin_reg			**0xFFDE**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
PA1	0	R/W	1	Input/output value of PA.1	Rport wport

Register::Porta2_pin_reg			**0xFFDF**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	

PA2	0	R/W	1	Input/output value of PA.2	Rport wport

Register::Porta3_pin_reg				0xFFE0	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
PA3	0	R/W	1	Input/output value of PA.3	Rport wport

Register::Porta4_pin_reg				0xFFE1	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:1	--	0	Reserved	
PA4	0	R/W	1	Input/output value of PA.4	Rport wport

SFR Access

When embedded MCU is selected, the P1, P3 GPIO is controlled by SFR. Both of the port groups must be access by below registers when using external MCU. Below registers are useless when embedded MCU is adopted.

Register::Port1_pin_reg				0xFFE2	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
P1	7:0	R/W	0xFF	Input/output value of P1 In 8051 mode, register is READ only In ISP mode, register can be READ and WRITE.	Rport Wport

Register::Port3_pin_reg				0xFFE3	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
P3	7:0	R/W	0xFF	Input/output value of P3 In 8051 mode, register is READ only In ISP mode, register can be READ and WRITE.	Rport Wport

Register:: DVI_EDID_IRQ				0xFFE4	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
~~REV_DUMMY_FFE4~~ SOD_ACCESS_DISABLE	~~7:6~~ 7	R/W	0	~~Dummy~~ 0：Enable SOD Access Xram 1：Disable SOD Access Xram	
REV_DUMMY_FFE4	6	R/W	0	Dummy	
DVI_FORCE_NACK	5	R/W	0	In DDC2B Mode, force DVI SDA no output 0:disable 1:enable	
DVI_FORCE_ACK_ZERO	4	R/W	0	In DDC2B Mode, force DVI SDA Output 0 (ACK) 0:disable 1:enable	

DVI_SCL_IRQ_EN	3	R/W	0	DVI SCL Toggle IRQ enable 0:disable 1:enable	
DVI_SCL_IRQ_STATUS	2	R/W	0	DVI SCL Toggle status，wite 1 to clear 0: no scl toggle after clear 1: new scl toggle after clear	Rport Wport
DVI_EDIDRD_IRQ_EN	1	R/W	0	DVI EDID READ IRQ enable 0:disable 1:enable	
DVI_EDIDRD_STATUS	0	R/W	0	DVI EDID READ STATUS，wite 1 to clear 0: no edid read after clear 1: new edid read after clear	Rport Wport

Register:: HDMI_EDID_IRQ				**0xFFE5**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
REV_DUMMY_FFE5	7:6	R/W	0	Dummy	
HDMI_FORCE_NACK	5	R/W	0	In DDC2B Mode, force HDMI SDA output 1(NACK) 0:disable 1:enable	
HDMI_FORCE_ACK_ZERO	4	R/W	0	In DDC2B Mode force HDMI SDA Output 0 (ACK) 0:disable 1:enable	
HDMI_SCL_IRQ_EN	3	R/W	0	HDMI SCL Toggle IRQ enable 0:disable 1:enable	
HDMI_SCL_IRQ_STATUS	2	R/W	0	HDMI SCL Toggle status，wite 1 to clear 0: no scl toggle after clear 1: new scl toggle after clear	Rport Wport
HDMI_EDIDRD_IRQ_EN	1	R/W	0	HDMI EDID READ IRQ enable 0:disable 1:enable	
HDMI_EDIDRD_STATUS	0	R/W	0	HDMI EDID READ STATUS，wite 1 to clear 0: no edid read after clear 1: new edid read after clear	Rport Wport

Register::Port_read_control_2			**0xFFE6**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PC_pin_reg_n	7	R/W	0	Source selection for PC read back 0: register 1: bus value	
PB_pin_reg_n	6	R/W	0	Source selection for PB read back 0: register 1: bus value	
PD_pin_reg_n	5	R/W	0	Source selection for PD read back 0: register 1: bus value	
Reserved	4:0	--	0	Reserved	

Register:: VGA_EDID_IRQ				**0xFFE7**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
REV_DUMMY_FFE7	7:6	R/W	0	Dummy	
VGA_FORCE_NACK	5	R/W	0	In DDC2B Mode, force VGA SDA output 1(NACK) 0:disable 1:enable	

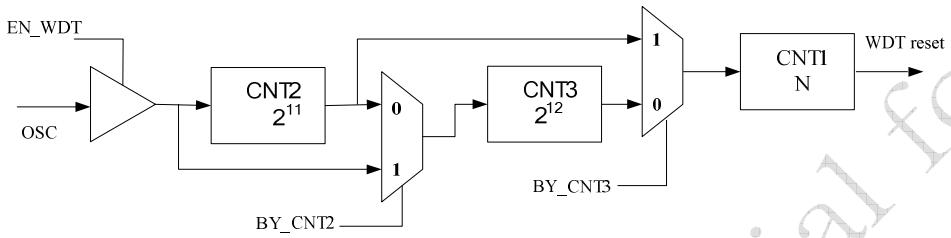
VGA _FORCE_ACK_ZERO	4	R/W	0	In DDC2B Mode force VGA SDA Output 0 (ACK) 0:disable 1:enable	
VGA _SCL_IRQ_EN	3	R/W	0	VGA SCL Toggle IRQ enable 0:disable 1:enable	
VGA _SCL_IRQ_STATUS	2	R/W	0	VGA SCL Toggle status，wite 1 to clear 0: no scl toggle after clear 1: new scl toggle after clear	Rport Wport
VGA _EDIDRD_IRQ_EN	1	R/W	0	VGA EDID READ IRQ enable 0:disable 1:enable	
VGA _EDIDRD_STATUS	0	R/W	0	VGA EDID READ STATUS，wite 1 to clear 0: no edid read after clear 1: new edid read after clear	Rport Wport

Register:: REV_DUMMY4				**0xFFE9**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
REV_DUMMY4	7:2	R/W	00	Dummy4	REV_DUMMY4
Disable RDSR CE	1	R/W	0	0: Enable RDSR CE 1: Disable RDSR CE	
Disable WREN CE	0	R/W	0	0: Enable WREN CE 1: Disable WREN CE	

Watchdog Timer

The Watchdog Timer automatically generates a device reset when it is overflowed. The interval of overflow is about 0.44 sec to 3.5 sec(assume crystal is 14.3MHz) and can be programmed via register CNT1.

Watchdog timer can be configured to generate an IRQ periodically, when watchdog timer count value increases to the threshold value, and if user do not clear the watchdog timer counters, watchdog timer will overflow to generate reset signal.



Register::WATCHDOG_timer			0xFFEA		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
WDT_EN	7	R/W	1	0: Disable watchdog timer 1: Enable watchdog timer	
CLR_WDT	6	W	0	0: No effect 1: Clear all counters of watchdog and latch the value in counter to 0xFFAA~0xFFAD	Rport Wport
SF_HLT_WDT_EN	5	R/W	0	1: Stop watchdog counter by SPI-FLASH access	
BW_HLT_WDT_EN	4	R/W	0	1: Stop watchdog counter by scalar burst write action	
Reserved	3	--	--	Reserved	
CNT1	2:0	R/W	0	The number N of counter1 000~111: 1~8	

- When ISP mode is enabled, watchdog will be disabled by hardware.

Register::WDT_test			0xFFEB		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
WDT_Overflow_Flag	7	R/W	0	when watchdog timer counter counts to threshold value, this bit will be set to 1. Write 1 to clear	Rport Wport
Reserved	6:5	-	0	Reserved	
WDT_TIMER_MODE	4	R/W	0	0: Normal mode 1: Timer mode	
Latch_WDT_CNT_Start	3	W	0	0: Finished 1: Start to latch watchdog timer counter	Rport Wport
Enable_WDT_IRQ	2	R/W	0	0: Disable watchdog timer counter generate IRQ 1: Enable watchdog timer counter generate IRQ this bit decides to generate an IRQ ,when watchdog timer count value increases to the threshold value.	

BY_CNT2	1	R/W	0	Signal bypass counter2* 0: signal pass through counter2 1: bypass	
BY_CNT3	0	R/W	0	Signal bypass counter3* 0: signal pass through counter3 1: bypass	

*When BY_CNT2 and BY_CNT3 are all assigned one (bypass), watchdog will be counted by CNT3

Register::WATCHDOG_CNT2_Value				**0xFFAA**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
WDT_CNT2_L_value	7:0	R	0	value of watchdog timer CNT2[7:0]	

Register:: WATCHDOG_CNT23_Value				**0xFFAB**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
WDT_CNT3_L_value	7:3	R	0	Value of watchdog timer CNT3[4:0]	
WDT_CNT2_H_value	2:0	R	0	Value of watchdog timer CNT2[10:8]	

Register:: WATCHDOG_CNT3_Value				**0xFFAC**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	**7**	**-**	**0**	**Reserved**	
WDT_CNT3_H_value	6:0	R	0	Value of watchdog timer CNT3[11:5]	

Register:: WATCHDOG _CNT1_Value				**0xFFAD**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	**7:3**	**-**	**0**	**Reserved**	
WDT_CNT1_ value	2:0	R	0	Value of watchdog timer CNT1[2:0]	

Write FW code to calculate counter value for an example
Eg.

DWORD dCounterValue;
dCounterValue = (DWORD) (**0xFFAD** <<23) | (**0xFFAC** << 16) | (**0xFFAB << 8**) | (**0xFFAA**);

Register::WATCHDOG_timer_threshold				**0xFFAE**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
WDT_CNT_threshold_L_value	7:0	R/W	0	threshold value TRH[7:0]. if watchdog timer counter matches threshold value, WDT_Overflow_Flag will be set to 1, and can generate an IRQ if Enable_WDT_IRQ set to 1. If BY_CNT3 = 1 & BY_CNT2 = 0 TRH [7:0] = CNT2[7:0] Else TRH [7:0] = CNT3[7:0]	

Register::WATCHDOG_timer_threshold_mask				**0xFFAF**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
WDT_CNT_threshold_L_mask	7:0	R/W	0	Mask value to threshold value, some bits of threshold value will be ignored. 1: mask. 0: no mask.	

Register::WATCHDOG_timer_H_threshold_mask			0xFFB0		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
WDT_CNT_threshold_H_value	7:4	R/W	0	threshold value TRH[11:8]. if watchdog timer counter matches threshold value, WDT_Overflow_Flag will be set to 1, and can generate an IRQ if Enable_WDT_IRQ set to 1. If BY_CNT3 = 1 & BY_CNT2 = 0 TRH [11:8] = CNT2[10:8] Else TRH [11:8] = CNT3[11:8]	
WDT_CNT_threshold_H_mask	3:0	R/W	0	Mask value to threshold value, some bits of threshold value will be ignored. 1: mask. 0: no mask.	

In System Programming

User can program the external serial FLASH by internal hardware without removing serial FLASH from the system. RTD2485XD utilizes DDC channel (ADC/DVI/HDMI DDC) to communicate with IIC host for ISP function. The ISP protocol is mainly compatible with DDC protocol. However, one significant difference is that the LSB of 7-bit ISP address is the address auto increase bit. Thus, we can improve the flash program speed.

Register::ISP_slave_address			0xFFEC		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ISP_ADDR	7:2	R/W	25	ISP slave address	
ISP_ADDR_INC_A	1	R	0	Received LSB of ISP slave address of ADC DDC channel 0: address is nonincrease 1: address is auto-increase	
ISP_ADDR_INC_D	0	R	0	Received LSB of ISP slave address of DVI DDC channel 0: address is nonincrease 1: address is auto-increase	

Register::MCU_control			0xFFED		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
PORT_PIN_REG	7	R/W	1	port_pin_reg_n enable 0: port_pin_reg_n signal is disabled 1: port_pin_reg_n signal is enabled	
ISP_ADDR_INC_H	6	R	0	Received LSB of ISP slave address of HDMI DDC channel 0: address is nonincrease 1: address is auto-increase	
FLASH_CLK_DIV	5:2	R/W	2	SPI-FLASH clock divider, its clock source is selected by MCU_CLK_SEL, default is MCU_CLK_SEL/2	
MCU_CLK_SEL	1	R/W	0	CPU clock source select 0: CPU clock is from Crystal divided by DIV	

				1: CPU clock is from PLL divided by DIV	
CKOUT_SEL	0	R/W	0	CLKO select 0: Select Crystal output 1: Select Mcu clk output	

Register::MCU_clock_control			**0xFFEE**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
ISP_MODE	7	R/W	0	ISP Mode Select 0: Normal ISP mode, I2C protocol (Device_ID, address, Data...) 1: AIO Block Write mode I2C protocol (Device_ID, addr, Length, Data...)	Rport Wport
MCU_PERI_NON_STOP	6	R/W	0	1: keep mcu peripheral running whiel mcu stopped by spi flash access 0: peripheral will be stopped with mcu*.	
MCU_CLK_DIV	5:2	R/W	1	MCU clock is FLASH clock/MCU_CLK_DIV.	
SOF_RST	1	R/W	0	Software reset mcu 0: No effect 1: reset ~~RTD2485XD~~ Scaler IC chip	Rport Wport
SCA_HRST	0	R/W	0	Hardware reset for Scalar 0: No effect 1: reset SCALAR module	

*note: this register bit[6] only has effect on peripheral built in SFR, which are timer 0, 1, 2 and serial port 1.

Register::RAM_test			**0xFFEF**		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
MCU_TSTCLK_SEL	7	R/W	0	1: MCU clock is from external pin 0: MCU clock is from crystal/PLL divided by DIV (0xFFED[1])	
SOD_DATA_LOCATION	6:5	R/W	3	Decide sod data wr/rd start location 0: Not used 1: F900 2: FA00 3: F900 SOD Xram space will map to this addr	
Test_mode	4	R/W	0	When test_mode of scalar is enabled for embedded mcu output, the bus is decided by following settins. 0: 10'h0, flash_mcu_clk_gated, bist_clk2pad, xram_clk2pad, 1'b0, shift_in_clk2pad, eclk_ng2pad, eclk_n2pad, eclk2pad, ad8_ckout, ad8_ckini, ad8_dout_s[7:0], ad8_div_clk2pad, fclk2pad 1: 5'h0, fc_addr[23:0], fc_rd_n	
EXT_RAM_BIST	3	R/W	0	Start BIST function for MCU external RAM (512 bytes) 0: finished and clear 1: start	Rport Wport
EXT_RAM_STA	2	R	0	Test result about MCU external RAM 0: fail 1: ok	

INT_RAM_BIST	1	R/W	0	Start BIST function for MCU internal RAM (256 bytes) 0: finished and clear 1: start	Rport Wport
INT_RAM_STA	0	R	0	Test result about MCU internal RAM 0: fail 1: ok	

mode = 0	mode = 1
0	0
0	0
0	0
0	0
0	0
0	fc_addr[23]
0	fc_addr[22]
0	fc_addr[21]
0	fc_addr[20]
0	fc_addr[19]
flash_mcu_clk_gated	fc_addr[18]
bist_clk2pad	fc_addr[17]
xram_clk2pad	fc_addr[16]
0	fc_addr[15]
shift_in_clk2pad	fc_addr[14]
eclk_ng2pad	fc_addr[13]
eclk_n2pad	fc_addr[12]
eclk2pad	fc_addr[11]
ad8_ckout	fc_addr[10]
ad8_ckini	fc_addr[9]
ad8_dout_s [7]	fc_addr[8]
ad8_dout_s [6]	fc_addr[7]
ad8_dout_s [5]	fc_addr[6]
ad8_dout_s [4]	fc_addr[5]
ad8_dout_s [3]	fc_addr[4]
ad8_dout_s [2]	fc_addr[3]
ad8_dout_s [1]	fc_addr[2]
ad8_dout_s [0]	fc_addr[1]
ad8_div_clk2pad	fc_addr[0]
fclk2pad	fc_rd_n

Xdata-SPI-FLASH Write Protect

Register:: Xdata_SPI_FLASH_Write_Protect			0xFFF0		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:5	R/W	0	Reserved	

Serial_stop_en	4	R/W	0	Stop 8051 serial 0: run 1: stop	
Timer2_stop_en	3	R/W	0	Stop 8051 timer2 0: run 1: stop	
IdleMode_Stop8051_en	2	R/W	0	Stop 8051 clock in idle mode 0: disable 1: enable	
Timer01_stop_en	1	R/W	0	Stop 8051 timer0 and timer1 0: run 1: stop	
XDATA_Flash_En	0	R/W	0	0: Enable xData write to Flash Function **(Default)** 1: Disable xData write to Flash Function	

Register:: Reserved				**0xFFF1**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
Reserved	7:0	R/W	0	Reserved	

Register:: PWM_I2C_CLOCK_STOP				**0xFFF2**	
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
I2C_CKXTAL_STOP	7	R/W	0	I2c crystal clock stop control 0: run 1: stop	
PWM_CKXTAL_STOP	6	R/W	0	PWM crystal clock stop control 0: run 1: stop	
PWM5_OUTCLK_STOP	5	R/W	0	PWM5 output clock stop control 0: run 1: stop	
PWM4_OUTCLK_STOP	4	R/W	0	PWM4 output clock stop control 0: run 1: stop	
PWM3_OUTCLK_STOP	3	R/W	0	PWM3 output clock stop control 0: run 1: stop	
PWM2_OUTCLK_STOP	2	R/W	0	PWM2 output clock stop control 0: run 1: stop	
PWM1_OUTCLK_STOP	1	R/W	0	PWM1 output clock stop control 0: run 1: stop	
PWM0_OUTCLK_STOP	0	R/W	0	PWM0 output clock stop control 0: run 1: stop	

Scalar Interface

Scalar Interface Related Register

External host interface is selected by power-on latch. The internal XFR access will be auto-switched to external host interface by power-on latch, too.

Scalar's register map must reserve addresses for mapping necessary XFR when using external host interface.

Register:: SCA_INF_CONTROL			0xFFF3		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
REG_READ_EN	7	R	0	Enable Read Action of Scalar Interface	
REG_WRITE_EN	6	R	0	Enable Write Action of Scalar Interface	
ADDR_NON_INC	5	R/W	0	1: turn-off address auto inc	
REG_BURCMD_WR	4	R/W	0	Enable burst write function, mcu will halt till action done or an interrupt triggered. *	Rport Wport
REG_BURDAT_WR	3	R/W	0	Enable burst write data to HOST_ADDR, mcu will halt till action done or an interrupt triggered *	Rport Wport
BURST_CMD_ERR	2	R	0	Burst write command error, value of SCA_INF_BWR_COUNT mismatch the length in content	
DIS_INT_RLS	1	R/W	0	1: disable the function of releasing mcu by interrupt	
PAGE_ADDR_MAP	0	R/W	0	0: Using normal XFR address to access all XFR 1: Using external page address for XFR. XFR can only be accessed by SCA_INF_ADDR, SCA_INF_DATA	

*: MCU will be released when interrupt is triggered. The bit value will maintain 1 before the operation is done. Rewriting this bit to 1 to continue the burst cmd/data write mode.

Register:: SCA_INF_ADDR			0xFFF4		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
HOST_ADDR	7:0	R/W	0x00	Host Interface Access Address	Rport Wport

Register:: SCA_INF_DATA			0xFFF5		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
HOST_DATA	7:0	R/W	0x00	Host Interface Access Data In/Out Continuously R/W with/without address auto-increment depends on ADDR_NON_INC	Rport Wport

Register:: SCA_INF_BWR_ADRH			0xFFF6		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BWR_ADRH	7:0	R/W	0x00	Burst Write Command Start Address [23:16]	Rport Wport

Register:: SCA_INF_BWR_ADRM			0xFFF7		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BWR_ADRM	7:0	R/W	0x00	Burst Write Command Start Address [15:8]	Rport Wport

Register:: SCA_INF_BWR_ADRL			0xFFF8		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BWR_ADRL	7:0	R/W	0x00	Burst Write Command Start Address [7:0]	Rport Wport

Register:: SCA_INF_BWR_COUNT_H			0xFFF9		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BWR_BYTE_COUNT_H	7:0	R/W	0x00	Burst Write Command Data Length. Left byte count when interrupt is asserted	Rport Wport

Register:: SCA_INF_BWR_COUNT_L			0xFFFA		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
BWR_BYTE_COUNT_L	7:0	R/W	0x00	Burst Write Command Data Length, left byte count when interrupt is asserted	Rport Wport

Register:: SCA_INF_PERIOD			0xFFFB		
Name	**Bits**	**R/W**	**Default**	**Comments**	**Config**
SCA_INF_PERIOD	7:0	R/W	0x02	Interval Between Two Command TI = SYS_PERIOD * 2 * BWR_PERIOD, it will halt mcu extra time TI in normal access and maintain the interval TI between two access in burst write	

				mode.	

*For 50MHz system clock, BWR_PERIOD can be delayed to 10.2us > 10us for special requirement of OSD register

Table Format for register burst write command format

BYTE code tFONT_GLOBAL[] = {
Length_A, AUTOINC_A, ADDR_A, DATA_A0, DATA_A1……,
Length_B, AUTOINC_B, ADDR_B, DATA_B0, DATA_B1……,
……..
_END
0x4 , 0x0, 0xA0, 01
0x6, 0x0, 0x90, 0x01, 0x02, 0x03
…….
0x0
}
note: AUTOINC = 0 means enable address auto-increment

Example for firmware reference:

1. OSD font table
   // set initial address of data = 0x102030
   BWR_ADRH = 0x10
   BWR_ADRM = 0x20
   BWR_ADRL = 0x30
   // set data byte count = 0x0890
   BWR_BYTE_COUNT_H = 0x08
   BWR_BYTE_COUNT_L = 0x90
   // set scalar's target register address = 0x58
   SCA_INF_ADDR = 0x58
   // enable burst data write
   SCA_INF_CONTROL[3] = 1'b1
2. Command table
   // set initial address of data = 0x102030
   BWR_ADRH = 0x10
   BWR_ADRM = 0x20
   BWR_ADRL = 0x30
   // set data byte count = 0x0890
   BWR_BYTE_COUNT_H = 0x08
   BWR_BYTE_COUNT_L = 0x90
   // enable burst command write
   SCA_INF_CONTROL[4] = 1'b1

3. Burst Write from XRAM
   BWR_ADRH = 0x03
   BWR_ADRM = 0xF9
   BWR_ADRL = 0x00
   Burst_Data_source_sel = Burst_Data_source_sel | 0x02 // Burst write from XRAM
   // set data byte count = 0x0890
   BWR_BYTE_COUNT_H = 0x08
   BWR_BYTE_COUNT_L = 0x90
   // enable burst command write
   SCA_INF_CONTROL[1] = 1'b1 // Disable releasing mcu by interrupt
   SCA_INF_CONTROL[0] = 1'b1 // Access XFR by SCA_INF_ADDR, SCA_INF_DATA
   SCA_INF_CONTROL[4] = 1'b1
   SCA_INF_ADDR = 0x 9F
   SCA_INF_DATA = 0x 0F
   SCA_INF_ADDR = 0x D3
   SCA_INF_DATA = 0x 00 // Clear 0x FFF3[0], Access XFR by XFR address

Table Format for A25L020A close software WP command format

```
BYTE code t A25L020A _CloseWP[] = {
Length_A, AUTOINC_A, ADDR_A, DATA_A0, DATA_A1……,
Length_B, AUTOINC_B, ADDR_B, DATA_B0, DATA_B1……,
……..
_END
0x4, 0x01,0x9F, 0x0E      // Enter Page E
0x4, 0x01, 0xA0, 0x20    // 0x FF60 = 0x20
0x4, 0x01, 0xA1, 0x06    // 0x FF61 = 0x06
0x4, 0x01, 0xA0, 0x21    // 0x FF60 = 0x21

0x4, 0x01, 0xA0, 0x38    // 0x FF60 = 0x38
0x4, 0x01, 0xA1, 0x90    // 0x FF61 = 0x90
0x4, 0x00, 0xA4, 0x00,0x28, 0x00    // 0x FF64 = 0x00, 0x FF65 = 0x28, 0x FF66 = 0x00
0x4, 0x01, 0xA0, 0x39    // 0x FF60 = 0x39

0x4, 0x01, 0xA0, 0x38    // 0x FF60 = 0x38
0x4, 0x01, 0xA1, 0x90    // 0x FF61 = 0x3B
0x4, 0x00, 0xA4, 0x00,0x28, 0x40    // 0x FF64 = 0x00, 0x FF65 = 0x28, 0x FF66 = 0x40
0x4, 0x01, 0xA0, 0x39    // 0x FF60 = 0x39

0x0
}
```
note: AUTOINC = 0 means enable address auto-increment

Bank Switch

Due to only one external SPI-FLASH is used for embedded MCU, it is necessary to allocate the non-volatile memory for both program code and external data (XDATA). In default, the 0x0 ~ 0xFFFF is used for program code, and the 0x10000 ~ 0x1FFFF is used for external data. When 64K-byte FLASH is used, there is no space in FLASH reserved for XDATA. XRAM and XFR are still accessible. More than 64K-byte FLASH can be used by designer's plan. GPIO mode for bank switch is used for 128K-byte FLASH and up to 256x64K bytes FLASH is supported by XFR mode. Both of them are supported by KeilC. The start bank of XDATA is defined by XDATA_BSTART. XRAM is allocated to 0xFB00 – 0xFEFF of XDATA's Bank0. Besides, except XFR address 0xFFFE and 0xFFFF, XFR and XRAM can be chosen if occupy only one bank of XDATA. Designer can set program address to include the whole FLASH address space without using the range reserved for XDATA to avoid partitioning the boundary in the power of 2. The following table is an example. Program is designed to use only 0x00000 – 0x4FFF, but declare a 0x00000 – 0x7FFFF address space. Bank 5 ~ Bank 7 of program code will be empty and not programmed into FLASH. XDATA is allocated in 0x50000 – 0x7FFFF via setting of XFR.

SPI FLASH Addres	Program Address	Xdata	XRAM	XFR
00000-0FFFF	Bank0, 0000-FFFF			
10000-1FFFF	Bank1, 0000-FFFF			
20000-2FFFF	Bank2, 0000-FFFF			
30000-3FFFF	Bank3, 0000-FFFF			
40000-4FFFF	Bank4, 0000-FFFF			
50000-5FFFF	Bank5, 0000-FFFF	Bank0, 0000-FFFF	F900 - FEFF	FF00-FFFF
60000-6FFFF	Bank6, 0000-FFFF	Bank1, 0000-FFFF		FF00-FFFF
70000-7FFFF	Bank7, 0000-FFFF	Bank2, 0001-FFFF		FF00-FFFF

Register::Bank_swich_control **0xFFFC**

Name	Bits	R/W	Default	Comments	Config
Reserved	7:6	--	0	Reserved	
Burst_Data_source_sel	5	R/W	0	0: Burst data from Ext flash 1: Burst data from Int Xram	
Scalar_Addr_Remapping	4	R/W	0	Scalar Addresss remapping to Xdata, (0x00_00 ~ 0x10_FF) 0: Disable 1: Enable	
GLOBAL_XFR	3	R/W	1	1: XFR will occupy the address space of all XDATA banks. 0: only 0xFFFF will occupy the address of all XDATA banks.	
GLOBAL_XRAM	2	R/W	0	1: XRAM will occupy the address space of all XDATA banks. 0: XRAM only occupy XDATA bank 0	
SW_MODE	1	R/W	0	0: using P3.5 as A16 1: using Pbank_switch (0xFFFF)	
BANK_EN	0	R/W	0	1: Enable Bank Switching Function for program address space 0: Disable Bank Switching, program memory space is 64K byte and GLOBAL_XFR, GLOBAL_XRAM, XDATA_BSTART, XDATA_BSEL still can be used to control XDATA memory space.	

Register::XDATA_bank_start **0xFFFD**

Name	Bits	R/W	Default	Comments	Config
XDATA_BSTART	7:0	R/W	0xff	The start bank number for XDATA access.	

Register::XDATA_bank_sel **0xFFFE**

Name	Bits	R/W	Default	Comments	Config
XDATA_BSEL	7:0	R/W	0	First bank number for XDATA access.	

Register::Pbank_switch **0xFFFF**

Name	Bits	R/W	Default	Comments	Config

PBANK_SEL	7:0	R/W	0	Bank number for program code access.	

Pin Share Register (page 10)

GPIO Pin List

GPIO Pin List	Name	GPI	GPO <push-pull>	GPO <Open-Drain>	3.3V Tolerance	5V Tolerance (When Power On)	5V Tolerance (When Power Off)	Pin Share Register	MCU Control
31	**PD.7**	Y	Y	Y	Y			Page10 , 0xB7	0xFF77
32	**PD.6**	Y	Y	Y	Y			Page10 , 0xB7	0xFF78
33	**PD.5**	Y	Y	Y	Y			Page10 , 0xB7	0xFF79
34	**PD.4**	Y	Y	Y	Y			Page10 , 0xB7	0xFF7A
35	**PD.3**	Y			Y			Page10 , 0xB8	0xFF7B
36	**PD.2**	Y	Y	Y	Y			Page10 , 0xB8	0xFF7C
37	**PD.1**	Y	Y	Y	Y			Page10 , 0xB8	0xFF7D
39	**PD.0**	Y	Y	Y			Y	Page10 , 0xB8	0xFF7E
40	**PC.4**	Y	Y	Y			Y	Page10 , 0xB9	0xFFA4
41	PB.7	Y	Y	Y			Y	Page10 , 0xA0	0xFF7F
42	PB.6	Y	Y	Y	Y			Page10 , 0xA0	0xFF89
43	PB.5	Y	Y	Y	Y			Page10 , 0xA0	0xFF8A
44	PB4	Y	Y	Y	Y			Page10 , 0xA8	0xFF8B
45	PB.3	Y	Y	Y	Y			Page10 , 0xA1	0xFF8C
46	PB.2	Y	Y	Y	Y			Page10 , 0xA3	0xFF8D
47	PB.1	Y	Y	Y	Y			Page10 , 0xA6	0xFF8E
48	PB.0	Y	Y	Y	Y			Page10 , 0xA7	0xFF8F
50	P6.0	Y	Y	Y		Y		Page10 , 0xAB	0xFFC7
51	P6.1	Y	Y	Y		Y		Page10 , 0xA3	0xFFC8
52	P6.2	Y	Y	Y		Y		Page10 , 0xA4	0xFFC9
53	P6.3	Y	Y	Y		Y		Page10 , 0xA5	0xFFCA
54	P6.4	Y	Y	Y			Y	Page10 , 0xA0	0xFFCB
55	P6.5	Y	Y	Y			Y	Page10 , 0xA6	0xFFCC
56	P6.6	Y	Y	Y			Y	Page10 , 0xAF	0xFFCD
57	P6.7	Y	Y	Y			Y	Page10 , 0xAF	0xFFCE
58	P3.0	Y	Y	Y			Y	Page10 , 0xA2	SFR Access
59	P3.1	Y	Y	Y			Y	Page10 , 0xA2	SFR Access
63	PC.3	Y	Y	Y			Y	Page10 , 0xA6	0xFFA0
64	P1.0	Y	Y	Y			Y	Page10 , 0xA3	SFR Access
65	P1.1	Y	Y				Y	Page10 , 0xB1	SFR Access
66	P1.2	Y	Y	Y			Y	Page10 , 0xA4	SFR Access
67	P1.3	Y	Y	Y			Y	Page10 , 0xA4	SFR Access
68	P1.4	Y	Y	Y			Y	Page10 , 0xA7	SFR Access
69	P1.5	Y	Y	Y			Y	Page10 , 0xA5	SFR Access
70	P1.6	Y	Y	Y			Y	Page10 , 0xA5	SFR Access
71	P1.7	Y	Y				Y	Page10 , 0xA7	SFR Access
72	PC.2	Y	Y				Y	Page10 , 0xB2	0xFFA1
74	P9.0	Y	Y	Y	Y			Page10 , 0xA9	0xFFD8
75	P9.1	Y	Y	Y	Y			Page10 , 0xA9	0xFFD9
76	P9.2	Y	Y	Y	Y			Page10 , 0xA9	0xFFDA
77	P9.3	Y	Y	Y	Y			Page10 , 0xA9	0xFFDB
78	P9.4	Y	Y	Y	Y			Page10 , 0xA9	0xFFDC
79	PA.0	Y	Y	Y	Y			Page10 , 0xA9	0xFFDD
80	PA.1	Y	Y	Y	Y			Page10 , 0xA9	0xFFDE
81	PA.2	Y	Y	Y	Y			Page10 , 0xA9	0xFFDF
82	PA.3	Y	Y	Y	Y			Page10 , 0xA9	0xFFE0
83	PA.4	Y	Y	Y	Y			Page10 , 0xA9	0xFFE1

Note: Pin 65~Pin 102 (GPI,GPO,GPIO) can not work when power saving & power Down. Please don't use Pin 65~Pin 102 for power status detect function.
(Power Saving CR[01] Bit1 Enable, Power Down CR[01] Bit2 Enable)

GPIO Pin List	Name	GPI	GPO <push-pull>	GPO <Open-Drain>	3.3V Tolerance	5V Tolerance (When Power ON)	5V Tolerance (When Power Off)	Pin Share Register	MCU Control
96	P5.2	Y	Y		Y			Page10 , 0xA8	0xFFC1
97	P5.3	Y	Y	Y	Y			Page10 , 0xA8	0xFFC2
98	P5.4	Y	Y	Y	Y			Page10 , 0xAA	0xFFC3
99	P5.5	Y	Y	Y	Y			Page10 , 0xA9	0xFFC4
100	P5.6	Y	Y	Y			Y	Page10 , 0xA9	0xFFC5
101	P5.7	Y	Y				Y	Page10 , 0xAB	0xFFC6
102	P7.6	Y	Y	Y			Y	Page10 , 0xAA	0xFFD5
103	P7.5	Y	Y	Y			Y	Page10 , 0xAC	0xFFD4
104	P7.4	Y	Y	Y			Y	Page10 , 0xAC	0xFFD3
105	P8.0	Y	Y	Y			Y	Page10 , 0xAA	0xFFD6
108	P8.1	Y	Y	Y			Y	Page10 , 0xAB	0xFFD7
109	P3.2	Y	Y	Y			Y	Page10 , 0xAD	SFR Access
110	P3.3	Y	Y	Y			Y	Page10 , 0xAD	SFR Access
111	P3.4	Y	Y	Y			Y	Page10 , 0xAE	SFR Access
112	P3.5	Y	Y	Y			Y	Page10 , 0xAE	SFR Access
113	P3.6	Y	Y	Y			Y	Page10 , 0xB0	SFR Access
114	P3.7	Y	Y	Y			Y	Page10 , 0xB2	SFR Access
119	PC.1	N	Y	Y			Y	Page10 , 0xA1	0xFFA2
121	P7.3	Y	Y	Y			Y	Page10 , 0xB0	0xFFD2
122	P7.2	Y	Y	Y			Y	Page10 , 0xB0	0xFFD1
123	P7.1	Y	Y	Y			Y	Page10 , 0xB1	0xFFD0
124	P7.0	Y	Y	Y			Y	Page10 , 0xB1	0xFFCF
126	PC.0	Y	Y	Y		Y		Page10 , 0xA1	0xFFA3

PWM Pin List

GPIO Pin List	Name	(Push-pull)	PWM (Open-drain)	3.3V Tolerance	5V Tolerance	5V Tolerance (When Power Off)	Pin Share Register
44	PWM5	Y	Y	Y			Page10 , 0xA8
48	PWM0	Y		Y			Page10 , 0xA7
54	PWM4		Y			Y	Page10 , 0xA0
55	PWM1	Y	Y			Y	Page10 , 0xA6
55	PWM5	Y				Y	Page10 , 0xA6
63	PWM2	Y	Y			Y	Page10 , 0xA6
64	PWM0	Y	Y			Y	Page10 , 0xA3
64	PWM3	Y	Y			Y	Page10 , 0xA3
65	PWM1	Y	Y			Y	Page10 , 0xB1
71	PWM1	Y				Y	Page10 , 0xA7
71	PWM5	Y				Y	Page10 , 0xA7
72	PWM3	Y	Y			Y	Page10 , 0xB2
96	PWM0	Y				Y	Page10 , 0xA8
97	PWM1	Y	Y			Y	Page10 , 0xA8
98	PWM2	Y				Y	Page10 , 0xAA
99	PWM3	Y	Y			Y	Page10 , 0xA9
100	PWM4	Y	Y			Y	Page10 , 0xA9
101	PWM5	Y				Y	Page10 , 0xAB
102	PWM0	Y	Y			Y	Page10 , 0xAA
103	PWM1	Y				Y	Page10 , 0xAC
109	PWM2		Y			Y	Page10 , 0xAD
114	PWM4	Y				Y	Page10 , 0xB2
119	PWM1	Y	Y			Y	Page10 , 0xA1
119	PWM5	Y	Y			Y	Page10 , 0xA1
126	PWM0		Y			Y	Page10 , 0xA1
126	PWM1	Y	Y			Y	Page10 , 0xA1

Register:: PIN_SHARE_CTRL00			0xA0		
Name	Bits	Read/ Write	Reset State	Comments	Config

PBD7	7:6	R/W	0x0	**Pin41** 00: PBD7i <I> <default> 01: PBD7o <O> <open-drain> 10: PBD7o <O> <push-pull> 11: Reserved Effectively only if-Page10_CR_B9[5:4]=2'b10	
PBD6	5:4	R/W	0x0	**Pin 42** 00: PBD6i <I> <default> 01: PBD6o <O> <open-drain> 10: PBD6o <O> <push-pull> 11: IICSCL <IO> <open-drain> Effectively only if Page10_CR_B9[5:4]=2'b10	
PBD5	3:2	R/W	0x0	**Pin43** 00: PBD5i <I> <default> 01: PBD5o <O> <open-drain> 10: PBD5o <O> <push-pull> 11: IICSDA <IO> <open-drain> Effectively only if Page10_CR_B9[5:4]=2'b10	
P6D4	1:0	R/W	0x0	**Pin54 (PAD_ADCA4)** 00: P6D4i <I> <default> 01: P6D4o <O> <open-drain> 10: P6D4o <O> <push-pull> **11: PWM4 <O> <open-drain>**	

Register:: PIN_SHARE_CTRL01			**0xA1**		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
PBD3	7:6	R/W	0x0	**Pin45** 00: PBD3i <I> <default> 01: PBD3o <O> <open-drain> 10: PBD3o <O> <push-pull> 11: SCK <O> Effectively only if Page10_CR_B9[5:4]=2'b10	
PCD1	5:3	R/W	0x0	**Pin119 (PAD_PWM5)** 000: PCD1i <I> <default> 001: PCD1o <O> <open-drain> 010: PCD1o <O> <push-pull> 011: PWM5 <O> <push-pull> 100: SPIF1 <O> 101: PWM5 <O> <open-drain> **110: PWM1 <O> < push-pull>** **111: PWM1 <O> <open-drain>** (Pin 119 : Power on latch Pin) (when AC Power On , Power on latch pin must be "High") Please don't let Pin119 be "Input Pin".	
PCD0	2:0	R/W	0x0	**Pin126 (PAD_CEC)** 000: PCD0i <I> 001: PCD0o <O> <open-drain> 010: PCD0o <O> <push-pull> 011: PWM1 <O> <push-pull> 100: **PWM0 <O> <open-drain>** 101: SPDIF2 <O> 110: PWM1 <O> <open-drain> 111: Reserved	

Register:: PIN_SHARE_CTRL02			**0xA2**		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
DDC1	7	R/W	0x0	**Pin58 (PAD_DDCSCL1)** 0: DDCSCL1 <IO> <open-drain> <default>	

				1: reserved //normal output(refer bit6~4) **Pin59 (PAD_DDCSDA1)** 0: DDCSDA1 <IO> <open-drain> <default> 1: reserved //normal output(refer bit3~1)	
DDCSCL1	6:4	R/W	0x0	**Pin58** 000: P3D0i <I> 001: P3D0o <O> <open-drain> 010: P3D0o <O> <push-pull> 011: RXD <IO> <open-drain> 100: Reserved 101: Reserved	
DDCSDA1	3:1	R/W	0x0	**Pin59** 000: P3D1i <I> 001: P3D1o <O> <open-drain> 010: P3D1o <O> <push-pull> 011: TXD <O> <open-drain> 100: Reserved 101: Reserved	
USB_DDC1_EN	0	R/W	0	when (Page 10 , 0xA2[0] = 1) && (pin55 = 1), disable ddc function of pin58, 59 and swap to pin52, 53.	

Register:: PIN_SHARE_CTRL03 0xA3

Name	Bits	Read/ Write	Reset State	Comments	Config
PBD2	7:6	R/W	0x0	**Pin46** 00: PBD2i <I> <default> 01: PBD2o <O> <open-drain> 10: PBD2o <O> <push-pull> 11: MCK <O> Effectively only if Page10_CR_B9[5:4]=2'b10	
P6D1	5:4	R/W	0x0	**Pin51 (PAD_ADCA1)** 00: P6D1i <I> <default> 01: P6D1o <O> <open-drain> 10: P6D1o <O> <push-pull> 11: ADCA1(8Bit) <I>	
P1D0	3:0	R/W	0x00	**Pin64** 0000: P1D0i <I> <default> 0001: P1D0o <O> <open-drain> 0010: P1D0o <O> <push-pull> 0011: T2 <I> 0100: SD0 <O> 0101: SPDIF0 <O> 0110: Reserved 0111:PWM0 <O><push-pull> 1000: Reserved 1001: PWM0 <O>< open-drain > 1010: INT1<I> 1011: Reserved **1100: PWM3 <O> <push-pull>** **1101: PWM3 <O> <open-drain>**	

Register:: PIN_SHARE_CTRL04 0xA4

Name	Bits	Read/ Write	Reset State	Comments	Config
P6D2	7:6	R/W	0x0	**Pin52 (PAD_ADCA2)** 00: P6D2i <I> <default> 01: P6D2o <O> <open-drain> 10: P6D2o <O> <push-pull> 11: ADCA2 (8Bit) <I>	
P1D2	5:3	R/W	0x0	**Pin66** 000: P1D2i <I> 001: P1D2o <O> <push-pull>	

				010: P1D2o <O> <open-drain> 011: CLKO <O> 100: SCK <O> 101: Reserved 110: Reserved	
P1D3	2:0	R/W	0x0	**Pin67** 000: P1D3i <I> 001: P1D3o <O> <push-pull> 010: P1D3o <O> <open-drain> 011: MCK <O> 100: Reserved 101: Reserved 110: Reserved	

Register:: PIN_SHARE_CTRL05			**0xA5**		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
P6D3	7:6	R/W	0x0	**Pin53 (PAD_ADCA3)** 00: P6D3i <I> <default> 01: P6D3o <O> <open-drain> 10: P6D3o <O> <push-pull> 11: ADCA3 (8Bit) <I>	
P1D5	5:3	R/W	0x0	**Pin69** 000: P1D5i <I> 001: P1D5o <O> <push-pull> 010: P1D5o <O> <open-drain> 011: SD1 <O> 100: Reserved 101: Reserved 110: SPDIF1 <O> 111: IICSCL <IO> <open-drain>	
P1D6	2:0	R/W	0x0	**Pin70** 000: P1D6i <I> 001: P1D6o <O> <push-pull> 010: P1D6o <O> <open-drain>011: SD2 <O> 100: Reserved 101: Reserved 110: SPDIF2 <O> 111: IICSDA <IO> <open-drain>	

Register:: PIN_SHARE_CTRL06			**0xA6**		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
PBD1	7:6	R/W	0x0	**Pin47** 00: PBD1i <I> <default> 01: PBD1o <O> <open-drain> 10: PBD1o <O> <push-pull> 11: SD0 <O> Effectively only if Page10_CR_B9[5:4]=2'b10	
P6D5	5:3	R/W	0x0	**Pin55 (PAD_ADCB0)** 000: P6D5i <I> <default> 001: P6D5o <O> <open-drain> 010: P6D5o <O> <push-pull> 011: PWM1 <O><open-drain> 100: PWM1 <O><push-pull> 101: PWM5 <O><push-pull> 110: Reserved 111: Reserved	
PCD3	2:0	R/W	0x0	**Pin63 (PAD_GPIO63)** 000: PCD3i <I> <default>	

				001: PCD3o <O> <open-drain> 010: PCD3o <O> <push-pull> 011: PWM2 <O><push-pull> 100: Reserved 101: Reserved 110: PWM2 <O><open-drain> 111 :INT0<I> Effective only if Page10, CR0xBA[5] = 1'b0	

Register:: PIN_SHARE_CTRL07			**0xA7**		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
PBD0	7:6	R/W	0x0	**Pin48** 00: PBD0i <I> <default> 01: PBD0o <O> <open-drain> 10: PBD0o <O> <push-pull> 11: PWM0 <O> <push-pull> Effectively only if Page10_CR_B9[5:4]=2'b10	
P1D4	5:3	R/W	0x0	**Pin68** 000: P1D4i <I> 001: P1D4o <O> <push-pull> 010: P1D4o <O> <open-drain> 011: SD0 <O> 100: Reserved 101: Reserved 110: SPDIF0 <O> 111: Reserved	
P1D7	2:0	R/W	0x0	**Pin71** 000: P1D7i <I> 001: P1D7o <O> <push-pull> 010: PWM5 <O><push-pull> 011: SD3 <O> 100: Reserved 101: Reserved 110: SPDIF3 <O> 111: PWM1 <O><push-pull>	

Register:: PIN_SHARE_CTRL08			**0xA8**		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
PBD4	7:5	R/W	0x0	**Pin44** 000: PBD4i <I> <default> 001: PBD4o <O> <open-drain> 010: PBD4o <O> <push-pull> 011: WS <O> 100: SPDIF3 <O> **101: PWM5 <O> <push-pull>** **110: PWM5 <O> <open-drain>** Effectively only if Page10_CR_B9[5:4]=2'b10	
P5D2	4:3	R/W	0x0	**Pin96 (PAD_PWM0)** 00: P5D2i <I> 01: P5D2o <O> <push-pull> 10: DCLK <O> 11: PWM0 <O> <push-pull>	
P5D3	2:0	R/W	0x0	**Pin97 (PAD_PWM1)** 000: P5D3i <I> 001: P5D3o <O> <push-pull> 010: P5D3o <O> <open-drain> 011: PWM1 <O> <push-pull> 100: DVS <O> 101: PWM1 <O> <open-drain> **110:** Reserved	

Register:: PIN_SHARE_CTRL09			0xA9		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
P9PA	7:6	R/W	0x0	**Pin74-83** **(Pin74-Pin78 P9.0~P9.4)** **(Pin79-Pin83 PA.0~PA.4)** 00: Hi-Z <default> 01: P9PAi <I> 10: P9PAo <O> (push-pull)* 11: P9PAo <O> (Open-drain) ~~ie.~~active if single-port LVDS without E/O swap *(for Pin 74,75,76, 77, 78,79,80,81,82,83 , 0xB6 is effectively only if 0xA9(Bit7:6)=2'b10) Pin 74,75,76,77,78,79,80,81,82,83 : Pin share for P9PAo(push-pull) , Audio IIS and Audio SPDIF .	
P5D5	5:3	R/W	0x0	**Pin99 (PAD_PWM3)** 000: P5D5i <I> 001: P5D5o <O> <push-pull> 010: P5D5o <O> <open-drain> 011: PWM3 <O><push-pull> 100: Reserved 101: Reserved 110: PWM3 <O><open-drain> 111: Reserved	
P5D6	2:0	R/W	0x0	**Pin100 (PAD_PWM4)** 000: P5D6i <I> 001: P5D6o <O> <push-pull> 010: P5D6o <O> <open-drain> 011: PWM4 <O><push-pull> 100: Reserved 101: Reserved 110: PWM4 <O><open-drain> **111:** Reserved	

Register:: PIN_SHARE_CTRL0A			0xAA		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
P5D4	7:6	R/W	0x0	**Pin98 (PAD_PWM2)** 00: P5D4i <I> 01: P5D4o <O> <push-pull> 10: P5D4o <O> <open-drain> 11: PWM2 <O> <push-pull>	
P7D6	5:3	R/W	0x0	**Pin102 (PAD_SPDIF3)** 000: P7D6i <I> 001: P7D6o <O> <push-pull> 010: P7D6o <O> <open-drain> 011: PWM0 <O><push-pull> 100: SD3 <O> 101: SPDIF3 <O> 110: Reserved 111: PWM0 <O><open-drain>	
P8D0	2:0	R/W	0x0	**Pin105 (PAD_SPDIF0)** 000: P8D0i <I> 001: P8D0o <O> <push-pull> 010: P8D0o <O> <open-drain> 011: Reserved 100: SD0 <O> 101: SPDIF0 <O>	

				110: Reserved 111: Reserved	

Register:: PIN_SHARE_CTRL0B			0xAB		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
P6D0	7:5	R/W	0x0	**Pin50 (PAD_ADCA0)** 000: P6D0i <I> <default> 001: P6D0o <O> <open-drain> 010: P6D0o <O> <push-pull> 011: ADCA0(8Bit) <I> 100: Reserved 101: Reserved	
P5D7	4:3	R/W	0x0	**Pin101 (PAD_PWM5)** 00: P5D7i <I> 01: P5D7o <O> <push-pull> 10: Reserved 11: PWM5 <O> <push-pull>	
P8D1	2:0	R/W	0x0	**Pin108 (PAD_MCK)** 000: P8D1i <I> 001: P8D1o <O> <push-pull> 010: P8D1o<O><open-drain> 011: CLKO <O> 100: MCK <O> 101: Reserved 110: Reserved	

Register:: PIN_SHARE_CTRL0C			0xAC		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
Reserved	7	R/W	0	Reserved to 0	
P7D5	6:4	R/W	0x0	**Pin103 (PAD_SPDIF2)** 000: P7D5i <I> 001: P7D5o <O> <push-pull> 010: P7D5o <O> <open-drain> 011: PWM1 <O><push-pull> 100: SD2 <O> 101: SPDIF2 <O> 110: Reserved 111: IICSCL <IO> <open-drain> Effective only if Page10, CR0xBA[4] = 1'b0	
P7D4	3:1	R/W	0x0	**Pin104 (PAD_SPDIF1)** 000: P7D4i <I> 001: P7D4o <O> <push-pull> 010: P7D4o <O> <open-drain> 011: SD1 <O> 100: IRQB <O> 101: Reserved 110: SPDIF1 <O> 111: IICSDA <IO> <open-drain>	
Reserved	0	R/W	0	reserved to 0	

Register:: PIN_SHARE_CTRL0D			0xAD		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
IIS_MIRROR	7	R/W	0x0	**Pin110~Pin114 Mirror** 0:Pin110 WS, Pin111 SD0, Pin112 SD1, Pin113 SD2, Pin114 SD3 1:Pin110 SD1, Pin111 SD0, Pin112 WS, Pin113 SCK, Pin114 MCK	
P3D2	6:4	R/W	0x0	**Pin109 (PAD_SCK)**	

				000: P3D2i <I> 001: P3D2o <O> <push-pull> 010: P3D2o <O> <open-drain> 011: INT0 <I> 100: Reserved 101: SCK <O> 110: Reserved **111: PWM2 <O> <open-drain>**	
Reserved	3	R/W	0	reserved to 0	
P3D3	2:0	R/W	0x0	**Pin110 (PAD_WS)** 000: P3D3i <I> <default> 001: P3D3o <O> <open-drain> 010: P3D3o <O> <push-pull> 011: INT1 <I> 100: Reserved 101: WS <O> 110: Reserved 111: Reserved If user need "IIS_SD1" for Pin 110 , PIN_SHARE_CTRL0D :0xAD:Bit 7→"b1" PIN_SHARE_CTRL0D:0xAD:Bit 2:0→"b101"	

Register:: PIN_SHARE_CTRL0E			**0xAE**		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
Reserved	7	R/W	0	reserved to 0	
P3D4	6:4	R/W	0x0	**Pin111 (PAD_SD0)** 000: P3D4i <I> <default> 001: P3D4o <O> <open-drain> 010: P3D4o <O> <push-pull> 011: T0 <I> 100: Reserved 101: SD0 <O> 110: SPDIF0 <O> 111: Reserved	
Reserved	3	R/W	0	reserved to 0	
P3D5	2:0	R/W	0x0	**Pin112 (PAD_SD1)** 000: P3D5i <I> <default> 001: P3D5o <O> <open-drain> 010: P3D5o <O> <push-pull> 011: T1 <I> 100: Reserved 101: SD1 <O> 110: SPDIF1 <O> 111: Reserved If user need "IIS_WS" for Pin 112 , PIN_SHARE_CTRL0D :0xAD:Bit 7→"b1" PIN_SHARE_CTRL0E :0xAE:Bit 2:0→"b101"	

Register:: PIN_SHARE_CTRL0F			**0xAF**		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
P6D6_7	7	R/W	0	**Pin56 (PAD_ADCB1)** 0: reserved //normal output(refer bit6~4) <default> 1: IICSCL <IO> <open-drain> **Pin57 (PAD_ADCB2)** 0: reserved //normal output(refer bit3~1) <default> 1: IICSDA <IO> <open-drain>	
P6D6	6:4	R/W	0x0	**Pin56** 000: P6D6i <I> <default> 001: P6D6o <O> <open-drain> 010: P6D6o <O> <push-pull>	

				011: Reserved 100: Reserved 101: Reserved	
P6D7	3:1	R/W	0x0	**Pin57** 000: P6D7i <I> <default> 001: P6D7o <O> <open-drain> 010: P6D7o <O> <push-pull> 011: Reserved 100: Reserved 101: Reserved	
PIN58_59	0	R/W	0	PIN58_59_Low_Leak 0: normal mode (47K pull up with 4.3~4.4v pin level) 1: low leak mode with duty penalty. (47K pull up with over 4.5v pin level)	

Register:: PIN_SHARE_CTRL10			**0xB0**		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
P3D6	7:5	R/W	0x0	**Pin113 (PAD_SD2)** 000: P3D6i <I> <default> 001: P3D6o <O> <open-drain> 010: P3D6o <O> <push-pull> 011: Reserved 100: SD2 <O> 101: SPDIF2 <O> 110: Reserved 111: Reserved If user need "IIS_SCK" for Pin 113 , PIN_SHARE_CTRL0D :0xAD:Bit 7→"b1" PIN_SHARE_CTRL10 :0xB0:Bit 7:5→"b100"	
DDC3	4	R/W	0x0	**Pin121 (PAD_DDCSCL3)** 0: DDCSCL3 <IO> <open-drain> <default> 1: reserved //normal output(refer bit3~2) **Pin122 (PAD_DDCSDA3)** 0: DDCSDA3 <IO> <open-drain> <default> 1: reserved //normal output(refer bit1~0)	
DDCSCL3	3:2	R/W	0x0	**Pin121 (PAD_DDCSCL3)** 00: P7D3i <I> 01: P7D3o <O> <open-drain> 10: P7D3o <O> <push-pull> 11: Reserved	
DDCSDA3	1:0	R/W	0x0	**Pin122 (PAD_DDCSDA3)** 00: P7D2i <I> 01: P7D2o <O> <open-drain> 10: P7D2o <O> <push-pull> 11: Reserved	

Register:: PIN_SHARE_CTRL11			**0xB1**		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
P1D1	7:5	R/W	0x00	**Pin65** 000: P1D1i <I> 001: P1D1o <O> <push-pull> 010: PWM1 <O><push-pull> 011: T2EX <I> 100: Reserved 101: Reserved 110: WS <O> 111: PWM1 <O><open-drain>	
DDC2	4	R/W	0	**Pin123 (PAD_DDCSDA2)** 0: DDCSDA2 <IO> <open-drain> <default> 1: reserved //normal output(refer bit3~2)	

				Pin124 (PAD_DDCSCL2) 0: DDCSCL2 <IO> <open-drain> <default> 1: reserved //normal output(refer bit1~0)	
DDCSDA2	3:2	R/W	0x0	**Pin123 (PAD_DDCSDA2)** 00: P7D1i <I> 01: P7D1o <O> <open-drain> 10: P7D1o <O> <push-pull> 11: Reserved	
DDCSCL2	1:0	R/W	0x0	**Pin124 (PAD_DDCSCL2)** 00: P7D0i <I> 01: P7D0o <O> <open-drain> 10: P7D0o <O> <push-pull> 11: Reserved	

Register:: PIN_SHARE_CTRL12 0xB2					
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
PCD2	7:5	R/W	0x0	**Pin72** 000: PCD2i <I> <default> 001: PCD2o <O> <push-pull> 010: Reserved 011: Reserved 100: PWM3 <O> <push-pull> 101: PWM3 <O> <open-drain>	
P3D7	4:2	R/W	0x0	**Pin114 (PAD_SD3)** 000: P3D7i <I> <default> 001: P3D7o <O> <open-drain> 010: P3D7o <O> <push-pull> 011: Reserved 100: SD3 <O> 101: SPDIF3 <O> 110: PWM4 <O><push-pull> 111: Reserved If user need "IIS_MCK" for Pin 114 , PIN_SHARE_CTRL0D :0xAD:Bit 7→"b1" PIN_SHARE_CTRL12 :0xB2:Bit 4:2→"b100"	
PIN121_122	1	R/W	0	PIN121_122_Low_Leak 0: normal mode (47K pull up with 4.3~4.4v pin level) 1: low leak mode with duty penalty. (47K pull up with over 4.5v pin level)	
PIN123_124	0	R/W	0	PIN123_124_Low_Leak 0: normal mode (47K pull up with 4.3~4.4v pin level) 1: low leak mode with duty penalty. (47K pull up with over 4.5v pin level)	

Register:: PIN_DRIVING_CTRL1 0xB3					
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
PIN50_53	7	R/W	0	**Driving Current Control00 – Pin50~53** 0: Low 1: High	
PIN54_57	6	R/W	0	**Driving Current Control01– Pin54~57** 0: Low 1: High	
PIN58_59	5	R/W	0	**Driving Current Control02 – Pin58~59** 0: Low (8mA) 1: High (12mA)	
PIN63_64	4	R/W	0	**Driving Current Control03 – Pin~~64~~~~–67~~63~64** 0: Low 1: High	
PIN74_95	3:2	R/W	0x3	**Reserved**	
LVDS_CK_DRV	1:0	R/W	0x03	**Reserved**	

Register:: PIN_DRIVING_CTRL2			0xB4		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
PIN39_40	7	R/W	0	**Driving Current Control – Pin39~40** 0: Low 1: High	
PIN41_48	6	R/W	0	**Driving Current Control05 – Pin41~48** 0: Low 1: High	
PIN110_114	5	R/W	0	**Driving Current Control06– Pin110~114** 0: Low 1: High	
PIN115	4	R/W	0	**SPI_SCLK, Driving Current Control – Pin115** 0: High (4mA) 1: L ow (2mA)	
PIN121_124	3	R/W	0	**Driving Current Control08– Pin121~124-119-126** 0: Low (8mA at pin121-124) 1: High (12mA at pin121-124)	
Pin96_109	2	R/W	0	**Driving Current Control – Pin96~109** 0: Low (4mA) 1: High (8mA)	
Pin74_83	1	R/W	0	**Driving Current Control – Pin74~83、Pin86~95** Active when CR_0x8C-00[1:0] select to TTL 0: Low (~~4mA~~ x1) 1: High (~~8mA~~ x2)	
Pin65_72	0	R/W	0	**Driving Current Control – Pin65~72** 0: Low (4mA) 1: High (8mA)	

Register:: PIN_PULLUP_CTRL3			0xB5		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
PIN50_53	7	R/W	0	**Pull Up Control00 – Pin50~53** 0: Disable 1: Enable (active only in GPI or open-drain case)	
PIN54_57	6	R/W	0	**Pull Up Control01– Pin54~57** 0: Disable 1: Enable (active only in GPI or open-drain case)	
PIN58_59	5	R/W	0	**Pull Up Control02 – Pin58~59** 0: Disable 1: Enable (active only in GPI or open-drain case)	
PIN64_126	4	R/W	0	**Pull Up Control03– Pin63~64-119-126** 0: Disable 1: Enable (active only in GPI or open-drain case)	
PIN110_114	3	R/W	0	**Pull Up Control04– Pin110~114** 0: Disable 1: Enable (active only in GPI or open-drain case)	
PIN121_124	2	R/W	0	**Pull Up Control05 – Pin121~124** 0: Disable 1: Enable (active only in GPI or open-drain case)	
PIN41_48	1	R/W	0	~~**Pull Up Control06 – Pin41~48**~~ ~~0: Disable~~ ~~1: Enable (active only in GPI or open-drain case)~~ Useless,	
PIN39_40	0	R/W	0	**Pull Up Control – Pin39~40** 0: Disable 1: Enable (active only in GPI or open-drain case)	

(for Pin 74,75,78,79 0xB6 is effectively only if 0xA9(Bit7:6)=2'b10)

Register:: PIN_SHARE_CTRL13			0xB6		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
P9D0	7:6	R/W	0x00	**Pin74** 00: P9D0o (push-pull) <O> 01: IIS_SD0 <O> 10: SPDIF0 <O> 11: Reserved TXO3+_8b when DISP_TYPE(8C-00) set to LVDS	
P9D1	5:4	R/W	0x00	**Pin75** 00: P9D1o (push-pull) <O> 01: IIS_MCK <O> 10: SPDIF1 <O> 11: Reserved TXO3-_8b when DISP_TYPE(8C-00) set to LVDS	
P9D4	3:2	R/W	0x00	**Pin78** 00: P9D4o (push-pull) <O> 01: IIS_SCK <O> 10: SPDIF2 <O> 11: Reserved TXO2+_8b when DISP_TYPE(8C-00) set to LVDS	
PAD0	1:0	R/W	0x00	**Pin79** 00: PAD0o (push-pull) <O> 01: IIS_WS <O> 10: SPDIF3 <O> 11: Reserved TXO2-_8b when DISP_TYP(8C-00)E set to LVDS	

Register:: PIN_SHARE_CTRL14			0xB7		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
PDD7	7:6	R/W	0x00	**Pin31** 00: PDD7i <I><default> 01: PDD7o <O><Push-Pull> 10: PDD7o <O><Open-drain> 11: Reserved	
PDD6	5:4	R/W	0x00	**Pin32** 00: PDD6i <I><default> 01: PDD6o <O><Push-Pull> 10: PDD6o <O><Open-drain> 11: Reserved	
PDD5	3:2	R/W	0x00	**Pin33** 00: PDD5i <I><default> 01: PDD5o <O><Push-Pull> 10: PDD5o <O><Open-drain> 11: Reserved	
PDD4	1:0	R/W	0x00	**Pin34** 00: PDD4i <I><default> 01: PDD4o <O><Push-Pull> 10: PDD4o <O><Open-drain> 11: Reserved	

Register:: PIN_SHARE_CTRL15			0xB8		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
PDD3	7:6	R/W	0x00	**Pin35** 00: PDD3i <I><default> 01: Reserved 10: Reserved	

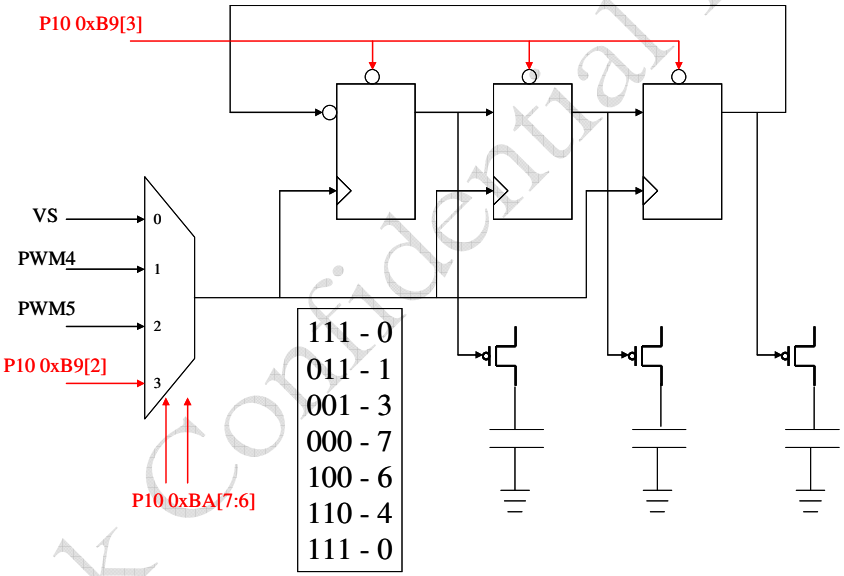
				11: Reserved	
PDD2	5:4	R/W	0x00	**Pin36** 00: PDD2i <I><default> 01: PDD2o <O><Push-Pull> 10: PDD2o <O><Open-drain> 11: Reserved	
PDD1	3:2	R/W	0x00	**Pin37** 00: PDD1i <I><default> 01: PDD1o <O><Push-Pull> 10: PDD1o <O><Open-drain> 11: Reserved	
PDD0	1:0	R/W	0x00	**Pin39** 00: PDD0i <I><default> 01: PDD0o <O><Push-Pull> 10: PDD0o <O><Open-drain> 11: Reserved	

Register:: PIN_SHARE_CTRL16			**0xB9**		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
PIN116_118	7	R/W	0	**Driving Current Control07 – Pin116~118** 0: High (4mA) 1: Low (2mA)	
V8_SEL	6	R/W	0x00	~~V8 source select~~ ~~0: From Pin 31~39~~ ~~1: From Pin 41~48~~	
P41_48_SEL	5:4	R/W	0x00	Pin41~48 function selection ~~00: Video 8~~ 01: AUDIO 10: Normal function, refer to Pin Share Part ~~If Video 8 from Pin31~Pin37, you must set Bit6=0 and Bit[5:4]can set 2'b01 or 2'b10~~	
Xtal_SPREAD_EN	3	R/W	0	Xtal Spread Spectrum Enable 0: Disable, reset capacitor select 1: Enable	
Xtal_SPREAD_MANUAL_SWITCH	2	R/W	0	Xtal Spread Spectrum Source Manual Switch 0: No Change 1: Switch to different frequency User must set "0: No Change" before "1: Switch to different frequency" (Only take effect when Page 10 CRCA[7:6] = 2'b11)	
PCD4	1:0	R/W	0x00	**Pin40** 00: PCD4i <I> <default> ~~01: PCD4o <O> <open-drain>~~ ~~10: PCD4o <O> <push-pull>~~ 01: PCD4o <O> <push-pull> 10: PCD4o <O> <open-drain> 11: Reserved	

Register:: DVI_CTRL_OUT_SEL			**0xBA**		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
Xtal_SPREAD_SOURCE_SEL	7:6	R/W	0x00	Xtal Spread Spectrum Source Select 00: Reference to VS 01: Reference to PWM4 10: Reference to PWM5	

				11: Adjust by FW (reference to Page 10 CRB9[2])	
PCD3	5	R/W	0	DVI CTRL OUT1 (Pin63) output enable 0: Disable (ref. original Pin63 pin share) 1: Enable	
P7D5	4	R/W	0x00	DVI CTRL OUT2 (Pin103) output enable 0: Disable (ref. original Pin103 pin share) 1: Enable	
DVI_CTRL_OUT1	3:2	R/W	0x00	DVI CTRL OUT1 select 00:CTRL0, 01:CTRL1, 10:CTRL2, 11:CTRL3	
DVI_CTRL_OUT2	1:0	R/W	0x00	DVI CTRL OUT2 select 00:CTRL0, 01:CTRL1, 10:CTRL2, 11:CTRL3	

CRBA[7:6] is used to choose switch reference source, the default capacitor setting is 111(minimum offset). Once VS / PWM signal latched, capacitor will change (111→011→001….,000 is maximum offset).
In manual switch mode, user set CRB9[2] will change the capacitor only one time. If user want to change capacitor again, must set "0: No Change" and then set "1: Switch to different frequency".



Register:: PIN_PULLUP_CTRL4			0xBB		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
PIN102_103	7	R/W	0	**Pull Down Control – Pin102~103** 0: Disable 1: Enable	
PIN100_101	6	R/W	0	**Pull Up Control– Pin100~101** 0: Disable 1: Enable	
PIN98_99	5	R/W	0	**Pull Down Control – Pin98~99** 0: Disable 1: Enable	
PIN96_97	4	R/W	0	**Pull Up Control– Pin96~97** 0: Disable 1: Enable	
PIN71_72	3	R/W	0	**Pull Down Control– Pin71~72**	

				0: Disable 1: Enable	
PIN69_70	2	R/W	0	**Pull Up Control – Pin69~70** 0: Disable 1: Enable	
PIN67_68	1	R/W	0	**Pull Down Control – Pin67~68** 0: Disable 1: Enable	
PIN65_66	0	R/W	0	**Pull Up Control – Pin65~66** 0: Disable 1: Enable	

Register:: PIN_PULLUP_CTRL5			**0xBC**		
Name	**Bits**	**Read/ Write**	**Reset State**	**Comments**	**Config**
DDC1_SCHMITT_EN	7	R/W	1	**DDC1** (pin58,pin59) schmitt trigger enable 0: Disable 1: Enable	
DDC2_SCHMITT_EN	6	R/W	1	**DDC2** (pin123,pin124) schmitt trigger enable 0: Disable 1: Enable	
DDC3_SCHMITT_EN	5	R/W	1	**DDC3** (pin121,pin122) schmitt trigger enable 0: Disable 1: Enable	
Reserved	4 ~~4:2~~	R/W	0	**Reserved**	
PIN115	3	R/W	0	**Slew Rate Control – SPI_SCLK** **0: Fast** **1; Slow**	
Pin116_117_PU	2	R/W	1	**Pull Up for Pin 116 & 117** **0: Disable** **1: Enable**	
PIN108_109	1	R/W	0	**Pull Down Control – Pin108~109** 0: Disable 1: Enable	
PIN104_105	0	R/W	0	**Pull Up Control – Pin104~105** 0: Disable 1: Enable	

0xBD ~ 0xBF reserved

Addressing and Accessing Register

ADDRESS	BIT							
	7	6	5	4	3	2	1	0
High Byte	-	-	-	-	-	-	-	A12
Med Byte	A15	A14	A13	A12	A11	A10	A9	A8
Low Byte	A7	A6	A5	A4	A3	A2	A1	A0

Figure 4. Addressing and Accessing Registers

Date	BIT							
Byte 0	D7	D6	D5	D4	D3	D2	D1	D0
Byte 1	D7	D6	D5	D4	D3	D2	D1	D0
Byte 2	D7	D6	D5	D4	D3	D2	D1	D0

Figure 2. Data Registers

All kind of registers can be controlled and accessed by these 2 bytes, and each address contains 3-byte data, details are described as follows:

Write mode: **[A15:A14] select which byte to write**

-00: Byte 0 -01:Byte 1 -10: Byte 2 –11: All

****All data are sorted by these three Bytes (Byte0~Byte2)***

[A13] Auto Load (Double Buffer)

[A12] Address indicator

-0: Window and frame control registers.

-1: Font Select and font map SRAM

[A12:A0] Address mapping

- Font Select and font map SRAM address: 000~~1A9F~~1ABF **~~6.67k~~6.848k*3byte**

-Frame control register address: 000~0xx **(Latch)**

-Window control register address: 100~1xx **(Latch)**

**** Selection of SRAM address or Latch address selection is determined by A12!***

Example:

Bit [15:14]=00

-All data followed are written to byte0 and address increases.

Byte0→Byte0→Byte0… (Address will auto increase)

Bit [15:14] =01

-All data followed are written to byte1 and address increases.

Byte1→Byte1→Byte1… (Address will auto increase)

Bit [15:14] =11

- Address will be increased after each 3-byte data written.

Byte0→Byte1→Byte2→Byte0→Byte1→Byte2… (Address will auto increase)

Window control registers

- Windows all support shadow/border/3D button
- Window0, 5, 6, 7, 8, 9 support gradient functions.
- Window 0, 1, 2, 3, 4, 4-1~4-8, 5, 6, 7, 8, 9 start/end resolution are 1line(pixel)
- All window start and end position include the ***special effect (border/shadow/3D button)*** been assigned
- Font comes after windows by 10 pixels, so you should compensate 10 pixels on windows to meet font position

Window 0 Shadow/Border/Gradient

Address: 100h

Byte 0

Bit	Mode	Function
7	W	Window 0 shadow color index in 64-color LUT [4]
6	W	Window 0 border color index in 64-color LUT [4]
5:3	W	Window 0 shadow/border width or 3D button thickness in pixel unit 000~111: 1 ~ 8 pixel
2:0	W	Window 0 shadow/border height in line unit 000~111: 1 ~ 8 line It must be the same as bit [5:3] for 3D button thickness

Byte 1

Bit	Mode	Function
7:4	W	Window 0 shadow color index in 64-color LUT [3:0] For 3D window, it is the left-top/bottom border color
3:0	W	Window 0 border color index in 64-color LUT [3:0] For 3D window, it is the right-bottom/top border color

Byte 2

Bit	Mode	Function
7	W	R Gradient Polarity 0: Decrease 1: Increase
6	W	G Gradient Polarity 0: Decrease 1: Increase
5	W	B Gradient Polarity 0: Decrease 1: Increase
4:3	W	Gradient level 00: 1 step per level 01: Repeat 2 step per level 10: Repeat 3 step per level 11: Repeat 4 step per level
2	W	Enable Red Color Gradient
1	W	Enable Green Color Gradient
0	W	Enable Blue Color Gradient

Window 0 start position
Address: 101h
Byte 0

Bit	Mode	Function
7:4	W	Window 0 horizontal start[11:8]
3:0	W	Window 0 vertical start [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 0 horizontal start[7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 0 vertical start [7:0]

~~Start position must be increments of four.~~
Window 0 end position
Address: 102h
Byte 0

Bit	Mode	Function
7:4	W	Window 0 horizontal end [11:8]
3:0	W	Window 0 vertical end [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 0 horizontal end [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 0 vertical end [7:0]

- ~~End position must be increments of four.~~

Window 0 control
Address: 103h
Byte 0

Bit	Mode	Function
7	W	Window 0 shadow color index in 64-color LUT [5]
6	W	Window 0 border color index in 64-color LUT [5]
5	W	Window 0 color index in 64-color LUT [5]
4	W	Saturated color mode for gradient (refer to ADDR 150 for saturated color setting) 0: disable 1: enable
3	W	Reversed color mode for gradient (valid while [4] is 1) 0: disable 1: enable
2:0	--	Reserved

Byte 1

Bit	Mode	Function
7:5	W	111: 7 level per gradient 110: 6 level per gradient 101: 5 level per gradient 100: 4 level per gradient 011: 3 level per gradient 010: 2 level per gradient 001: 1 level per gradient

		000: 8 level per gradient
4:0	W	Window 0 color index in 64-color LUT [4:0]

Byte 2 default: 00h

Bit	Mode	Function
7	W	Blend function 0: Disable 1: Enable
6	W	Gradient function 0: Disable 1: Enable
5	W	Gradient direction 0: Horizontal 1: Vertical
4	W	Shadow/Border/3D button 0: Disable 1: Enable
3:1	W	Window 0 Type 000: Shadow Type 1 001: Shadow Type 2 010: Shadow Type3 011: Shadow Type 4 100: 3D Button Type 1 101: 3D Button Type 2 110: Reserved 111: Border
0	W	Window 0 Enable 0: Disable 1: Enable

Window 1 Shadow/Border/Gradient
Address: 104h

Byte 0

Bit	Mode	Function
7	W	Window 1 shadow color index in 64-color LUT [4]
6	W	Window 1 border color index in 64-color LUT [4]
5:3	W	Window 1 shadow/border width or 3D button thickness in pixel unit 000~111: 1 ~ 8 pixel
2:0	W	Window 1 shadow/border height in line unit 000~111: 1 ~ 8 line It must be the same as bit [5:3] for 3D button thickness

Byte 1

Bit	Mode	Function
7:4	W	Window 1 shadow color index in 64-color LUT [3:0] For 3D window, it is the left-top/bottom border color
3:0	W	Window 1 border color index in 64-color LUT [3:0] For 3D window, it is the right-bottom/top border color

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Window 1 start position
Address: 105h

Byte 0

Bit	Mode	Function
7:4	W	Window 1 horizontal start [11:8]
3:0	W	Window 1 vertical start [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 1 horizontal start [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 1 vertical start [7:0]

Window 1 end position
Address: 106h

Byte 0

Bit	Mode	Function
7:4	W	Window 1 horizontal end [11:8]
3:0	W	Window 1 vertical end [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 1 horizontal end [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 1 vertical end [7:0]

Window 1 control
Address: 107h
Byte 0

Bit	Mode	Function
7	W	Window 1 shadow color index in 64-color LUT [5]
6	W	Window 1 border color index in 64-color LUT [5]
5	W	Window 1 color index in 64-color LUT [5]
4:0	--	Reserved

Byte 1

Bit	Mode	Function
7:5	--	Reserved
4:0	W	Window 1 color index in 64-color LUT [4:0]

Byte 2 default: 00h

Bit	Mode	Function
7	W	Blend function 0: Disable 1: Enable
6:5	W	Reserved
4	W	Shadow/Border/3D button 0: Disable 1: Enable
3:1	W	Window 1 Type 000: Shadow Type 1 001: Shadow Type 2 010: Shadow Type3 011: Shadow Type 4 100: 3D Button Type 1 101: 3D Button Type 2 110: Reserved 111: Border
0	W	Window 1 Enable 0: Disable 1: Enable

Window 2 Shadow/Border/Gradient
Address: 108h
Byte 0

Bit	Mode	Function
7	W	Window 2 shadow color index in 64-color LUT [4]
6	W	Window 2 border color index in 64-color LUT [4]
5:3	W	Window 2 shadow/border width or 3D button thickness in pixel unit 000~111: 1 ~ 8 pixel
2:0	W	Window 2 shadow/border height in line unit 000~111: 1 ~ 8 line It must be the same as bit [5:3] for 3D button thickness

Byte 1

Bit	Mode	Function
7:4	W	Window 2 shadow color index in 64-color LUT [3:0] For 3D window, it is the left-top/bottom border color
3:0	W	Window 2 border color index in 64-color LUT [3:0] For 3D window, it is the right-bottom/top border color

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Window 2 start position
Address: 109h
Byte 0

Bit	Mode	Function
7:4	W	Window 2 horizontal start [11:8]
3:0	W	Window 2 vertical start [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 2 horizontal start [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 2 vertical start [7:0]

Window 2 end position
Address: 10Ah
Byte 0

Bit	Mode	Function
7:4	W	Window 2 horizontal end [11:8]
3:0	W	Window 2 vertical end [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 2 horizontal end [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 2 vertical end [7:0]

Window 2 control
Address: 10Bh
Byte 0

Bit	Mode	Function
7	W	Window 2 shadow color index in 64-color LUT [5]
6	W	Window 2 border color index in 64-color LUT [5]
5	W	Window 2 color index in 64-color LUT [5]
4:0	--	Reserved

Byte 1

Bit	Mode	Function
7:5	--	Reserved
4:0	W	Window 2 color index in 64-color LUT [4:0]

Byte 2 default: 00h

Bit	Mode	Function
7	W	Blend function 0: Disable 1: Enable
6:5	W	Reserved
4	W	Shadow/Border/3D button 0: Disable 1: Enable
3:1	W	Window 2 Type

		000: Shadow Type 1 001: Shadow Type 2 010: Shadow Type3 011: Shadow Type 4 100: 3D Button Type 1 101: 3D Button Type 2 110: Reserved 111: Border
0	W	Window 2 Enable 0: Disable 1: Enable

Window 3 Shadow/Border/Gradient
Address: 10Ch
Byte 0

Bit	Mode	Function
7	W	Window 3 shadow color index in 64-color LUT [4]
6	W	Window 3 border color index in 64-color LUT [4]
5:3	W	Window 3 shadow/border width or 3D button thickness in pixel unit 000~111: 1 ~ 8 pixel
2:0	W	Window 3 shadow/border height in line unit 000~111: 1 ~ 8 line It must be the same as bit [5:3] for 3D button thickness

Byte 1

Bit	Mode	Function
7:4	W	Window 3 shadow color index in 64-color LUT [3:0] For 3D window, it is the left-top/bottom border color
3:0	W	Window 3 border color index in 64-color LUT [3:0] For 3D window, it is the right-bottom/top border color

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Window 3 start position
Address: 10Dh
Byte 0

Bit	Mode	Function
7:4	W	Window 3 horizontal start [11:8]
3:0	W	Window 3 vertical start [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 3 horizontal start [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 3 vertical start [7:0]

Window 3 end position
Address: 10Eh
Byte 0

Bit	Mode	Function
7:4	W	Window 3 horizontal end [11:8]
3:0	W	Window 3 vertical end [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 3 horizontal end [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 3 vertical end [7:0]

Window 3 control
Address: 10Fh
Byte 0

Bit	Mode	Function
7	W	Window 3 shadow color index in 64-color LUT [5]
6	W	Window 3 border color index in 64-color LUT [5]
5	W	Window 3 color index in 64-color LUT [5]
4:0	--	Reserved

Byte 1

Bit	Mode	Function
7:5	--	Reserved
4:0	W	Window 3 color index in 64-color LUT [4:0]

Byte 2 default: 00h

Bit	Mode	Function
7	W	Blend function 0: Disable 1: Enable
6:5	W	Reserved
4	W	Shadow/Border/3D button 0: Disable 1: Enable
3:1	W	Window 3 Type 000: Shadow Type 1 001: Shadow Type 2 010: Shadow Type3 011: Shadow Type 4 100: 3D Button Type 1 101: 3D Button Type 2 110: Reserved 111: Border
0	W	Window 3 Enable 0: Disable 1: Enable

Window 4 Shadow/Border/Gradient
Address: 110h
Byte 0

Bit	Mode	Function
7	W	Window 4 shadow color index in 64-color LUT [4]
6	W	Window 4 border color index in 64-color LUT [4]
5:3	W	Window 4 shadow/border width or 3D button thickness in pixel unit 000~111: 1 ~ 8 pixel

2:0	W	Window 4 shadow/border height in line unit 000~111: 1 ~ 8 line It must be the same as bit [5:3] for 3D button thickness

Byte 1

Bit	Mode	Function
7:4	W	Window 4 shadow color index in 64-color LUT [3:0] For 3D window, it is the left-top/ bottom border color
3:0	W	Window 4 border color index in 64-color LUT [3:0] For 3D window, it is the right-bottom/top border color

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Window 4 start position
Address: 111h

Byte 0

Bit	Mode	Function
7:4	W	Window 4 horizontal start [11:8]
3:0	W	Window 4 vertical start [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 4 horizontal start [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 4 vertical start [7:0]

Window 4 end position
Address: 112h

Byte 0

Bit	Mode	Function
7:4	W	Window 4 horizontal end [11:8]
3:0	W	Window 4 vertical end [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 4 horizontal end [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 4 vertical end [7:0]

Window 4 control
Address: 113h

Byte 0

Bit	Mode	Function
7:0	--	Reserved

Byte 1

Bit	Mode	Function
7:4	--	Reserved
3:0	W	Window 4 color index in 64-color LUT [3:0]

Byte 2 default: 00h

Bit	Mode	Function
7	W	Blend function 0: Disable 1: Enable

6:5	W	Reserved
4	W	Shadow/Border/3D button 0: Disable 1: Enable
3:1	W	Window 4 Type 000: Shadow Type 1 001: Shadow Type 2 010: Shadow Type3 011: Shadow Type 4 100: 3D Button Type 1 101: 3D Button Type 2 110: Reserved 111: Border
0	W	Window 4 Enable 0: Disable 1: Enable

Window 5 Shadow/Border/Gradient
Address: 114h
Byte 0

Bit	Mode	Function
7	W	Window 5 shadow color index in 64-color LUT [4]
6	W	Window 5 border color index in 64-color LUT [4]
5:3	W	Window 5 shadow/border width or 3D button thickness in pixel unit 000~111: 1 ~ 8 pixel
2:0	W	Window 5 shadow/border height in line unit 000~111: 1 ~ 8 line It must be the same as bit [5:3] for 3D button thickness

Byte 1

Bit	Mode	Function
7:4	W	Window 5 shadow color index in 64-color LUT [3:0] For 3D window, it is the left-top/bottom border color
3:0	W	Window 5 border color index in 64-color LUT [3:0] For 3D window, it is the right-bottom/top border color

Byte 2

Bit	Mode	Function
7	W	R Gradient Polarity 0: Decrease 1: Increase
6	W	G Gradient Polarity 0: Decrease 1: Increase
5	W	B Gradient Polarity 0: Decrease 1: Increase
4:3	W	Gradient level 00: 1 step per level 01: Repeat 2 step per level 10: Repeat 3 step per level 11: Repeat 4 step per level
2	W	Enable Red Color Gradient
1	W	Enable Green Color Gradient
0	W	Enable Blue Color Gradient

Window 5 start position
Address: 115h
Byte 0

Bit	Mode	Function
7:4	W	Window 5 horizontal start [11:8]
3:0	W	Window 5 vertical start [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 5 horizontal start [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 5 vertical start [7:0]

Window 5 end position
Address: 116h
Byte 0

Bit	Mode	Function
7:4	W	Window 5 horizontal end [11:8]
3:0	W	Window 5 vertical end [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 5 horizontal end [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 5 vertical end [7:0]

Window 5 control
Address: 117h
Byte 0

Bit	Mode	Function
7	W	Window 5 shadow color index in 64-color LUT [5]
6	W	Window 5 border color index in 64-color LUT [5]
5	W	Window 5 color index in 64-color LUT [5]
4	W	Saturated color mode for gradient (refer to ADDR 15A for saturated color setting) 0: disable 1: enable
3	W	Reversed color mode for gradient (valid while [4] is 1) 0: disable 1: enable
2:0	--	Reserved

Byte 1

Bit	Mode	Function
7:5	W	111: 7 level per gradient 110: 6 level per gradient 101: 5 level per gradient 100: 4 level per gradient 011: 3 level per gradient 010: 2 level per gradient 001: 1 level per gradient 000: 8 level per gradient
4:0	W	Window 5 color index in 64-color LUT [4:0]

Byte 2 default: 00h

Bit	Mode	Function
7	W	Blend function 0: Disable 1: Enable
6	W	Gradient function 0: Disable 1: Enable
5	W	Gradient direction 0: Horizontal 1: Vertical
4	W	Shadow/Border/3D button 0: Disable 1: Enable
3:1	W	Window 5 Type 000: Shadow Type 1 001: Shadow Type 2 010: Shadow Type3 011: Shadow Type 4 100: 3D Button Type 1 101: 3D Button Type 2 110: Reserved 111: Border
0	W	Window 5 Enable 0: Disable 1: Enable

Window 6 Shadow/Border/Gradient
Address: 118h
Byte 0

Bit	Mode	Function
7	W	Window 6 shadow color index in 64-color LUT [4]
6	W	Window 6 border color index in 64-color LUT [4]
5:3	W	Window 6 shadow/border width or 3D button thickness in pixel unit 000~111: 1 ~ 8 pixel
2:0	W	Window 6 shadow/border height in line unit 000~111: 1 ~ 8 line It must be the same as bit [5:3] for 3D button thickness

PS: This is for non-rotary, rotate 270, rotate 90 and 180.
Byte 1

Bit	Mode	Function
7:4	W	Window 6 shadow color index in 64-color LUT [3:0] For 3D window, it is the left-top/ bottom border color
3:0	W	Window 6 border color index in 64-color LUT [3:0] For 3D window, it is the right-bottom/top border color

Byte 2

Bit	Mode	Function
7	W	R Gradient Polarity 0: Decrease 1: Increase
6	W	G Gradient Polarity

		0: Decrease 1: Increase
5	W	B Gradient Polarity 0: Decrease 1: Increase
4:3	W	Gradient level 00: 1 step per level 01: Repeat 2 step per level 10: Repeat 3 step per level 11: Repeat 4 step per level
2	W	Enable Red Color Gradient
1	W	Enable Green Color Gradient
0	W	Enable Blue Color Gradient

Window 6 start position
Address: 119h
Byte 0

Bit	Mode	Function
7:4	W	Window 6 horizontal start [11:8]
3:0	W	Window 6 vertical start [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 6 horizontal start [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 6 vertical start [7:0]

Window 6 end position
Address: 11Ah
Byte 0

Bit	Mode	Function
7:4	W	Window 6 horizontal end [11:8]
3:0	W	Window 6 vertical end [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 6 horizontal end [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 6 vertical end [7:0]

Window 6 control
Address: 11Bh
Byte 0

Bit	Mode	Function
7	W	Window 6 shadow color index in 64-color LUT [5]
6	W	Window 6 border color index in 64-color LUT [5]
5	W	Window 6 color index in 64-color LUT [5]
4	W	Saturated color mode for gradient (refer to ADDR 15C for saturated color setting) 0: disable 1: enable
3	W	Reversed color mode for gradient (valid while [4] is 1) 0: disable

		1: enable
2:0	--	Reserved

Byte 1

Bit	Mode	Function
7:5	W	111: 7 level per gradient 110: 6 level per gradient 101: 5 level per gradient 100: 4 level per gradient 011: 3 level per gradient 010: 2 level per gradient 001: 1 level per gradient 000: 8 level per gradient
4:0	W	Window 6 color index in 64-color LUT [4:0]

Byte 2 default: 00h

Bit	Mode	Function
7	W	Blend function 0: Disable 1: Enable
6	W	Gradient function 0: Disable 1: Enable
5	W	Gradient direction 0: Horizontal 1: Vertical
4	W	Shadow/Border/3D button 0: Disable 1: Enable
3:1	W	Window 6 Type 000: Shadow Type 1 001: Shadow Type 2 010: Shadow Type3 011: Shadow Type 4 100: 3D Button Type 1 101: 3D Button Type 2 110: Reserved 111: Border
0	W	Window 6 Enable 0: Disable 1: Enable

Window 7 Shadow/Border/Gradient
Address: 11Ch

Byte 0

Bit	Mode	Function
7	W	Window 7 shadow color index in 64-color LUT [4]
6	W	Window 7 border color index in 64-color LUT [4]
5:3	W	Window 7 shadow/border width or 3D button thickness in pixel unit 000~111: 1 ~ 8 pixel
2:0	W	Window 7 shadow/border height in line unit 000~111: 1 ~ 8 line It must be the same as bit [5:3] for 3D button thickness

PS: This is for non-rotary, rotate 270, rotate 90 and 180.

Byte 1

Bit	Mode	Function
7:4	W	Window 7 shadow color index in 64-color LUT [3:0] For 3D window, it is the left-top/bottom border color
3:0	W	Window 7 border color index in 64-color LUT [3:0] For 3D window, it is the right-bottom/top border color

Byte 2

Bit	Mode	Function
7	W	R Gradient Polarity 0: Decrease 1: Increase
6	W	G Gradient Polarity 0: Decrease 1: Increase
5	W	B Gradient Polarity 0: Decrease 1: Increase
4:3	W	Gradient level 00: 1 step per level 01: Repeat 2 step per level 10: Repeat 3 step per level 11: Repeat 4 step per level
2	W	Enable Red Color Gradient
1	W	Enable Green Color Gradient
0	W	Enable Blue Color Gradient

Window 7 start position

Address: 11Dh

Byte 0

Bit	Mode	Function
7:4	W	Window 7 horizontal start [11:8]
3:0	W	Window 7 vertical start [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 7 horizontal start [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 7 vertical start [7:0]

Window 7 end position

Address: 11Eh

Byte 0

Bit	Mode	Function
7:4	W	Window 7 horizontal end [11:8]
3:0	W	Window 7 vertical end [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 7 horizontal end [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 7 vertical end [7:0]

Window 7 control
Address: 11Fh
Byte 0

Bit	Mode	Function
7	W	Window 7 shadow color index in 64-color LUT [5]
6	W	Window 7 border color index in 64-color LUT [5]
5	W	Window 7 color index in 64-color LUT [5]
4	W	Saturated color mode for gradient (refer to ADDR 15E for saturated color setting) 0: disable 1: enable
3	W	Reversed color mode for gradient (valid while [4] is 1) 0: disable 1: enable
2:0	--	Reserved

Byte 1

Bit	Mode	Function
7:5	W	111: 7 level per gradient 110: 6 level per gradient 101: 5 level per gradient 100: 4 level per gradient 011: 3 level per gradient 010: 2 level per gradient 001: 1 level per gradient 000: 8 level per gradient
4:0	W	Window 7 color index in 64-color LUT [4:0]

Byte 2 default: 00h

Bit	Mode	Function
7	W	Blend function 0: Disable 1: Enable
6	W	Gradient function 0: Disable 1: Enable
5	W	Gradient direction 0: Horizontal 1: Vertical
4	W	Shadow/Border/3D button 0: Disable 1: Enable
3:1	W	Window 7 Type 000: Shadow Type 1 001: Shadow Type 2 010: Shadow Type3 011: Shadow Type 4 100: 3D Button Type 1 101: 3D Button Type 2 110: Reserved 111: Border
0	W	Window 7 Enable 0: Disable

		1: Enable

Window 8 Shadow/Border/Gradient
Address: 120h
Byte 0

Bit	Mode	Function
7	W	Window 8 shadow color index in 64-color LUT [4]
6	W	Window 8 border color index in 64-color LUT [4]
5:3	W	Window 8 shadow/border width or 3D button thickness in pixel unit 000~111: 1 ~ 8 pixel
2:0	W	Window 8 shadow/border height in line unit 000~111: 1 ~ 8 line It must be the same as bit [5:3] for 3D button thickness

PS: This is for non-rotary, rotate 270, rotate 90 and 180.
Byte 1

Bit	Mode	Function
7:4	W	Window 8 shadow color index in 64-color LUT [3:0] For 3D window, it is the left-top/bottom border color
3:0	W	Window 8 border color index in 64-color LUT [3:0] For 3D window, it is the right-bottom/top border color

Byte 2

Bit	Mode	Function
7	W	R Gradient Polarity 0: Decrease 1: Increase
6	W	G Gradient Polarity 0: Decrease 1: Increase
5	W	B Gradient Polarity 0: Decrease 1: Increase
4:3	W	Gradient level 00: 1 step per level 01: Repeat 2 step per level 10: Repeat 3 step per level 11: Repeat 4 step per level
2	W	Enable Red Color Gradient
1	W	Enable Green Color Gradient
0	W	Enable Blue Color Gradient

Window 8 start position
Address: 121h
Byte 0

Bit	Mode	Function
7:4	W	Window 8 horizontal start [11:8]
3:0	W	Window 8 vertical start [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 8 horizontal start [7:0]

Byte 2

Bit	Mode	Function

7:0	W	Window 8 vertical start [7:0]

Window 8 end position
Address: 122h
Byte 0

Bit	Mode	Function
7:4	W	Window 8 horizontal end [11:8]
3:0	W	Window 8 vertical end [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 8 horizontal end [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 8 vertical end [7:0]

Window 8 control
Address: 123h
Byte 0

Bit	Mode	Function
7	W	Window 8 shadow color index in 64-color LUT [5]
6	W	Window 8 border color index in 64-color LUT [5]
5	W	Window 8 color index in 64-color LUT [5]
4	W	Saturated color mode for gradient (refer to ADDR 160 for saturated color setting) 0: disable 1: enable
3	W	Reversed color mode for gradient (valid while [4] is 1) 0: disable 1: enable
2:0	--	Reserved

Byte 1

Bit	Mode	Function
7:5	W	111: 7 level per gradient 110: 6 level per gradient 101: 5 level per gradient 100: 4 level per gradient 011: 3 level per gradient 010: 2 level per gradient 001: 1 level per gradient 000: 8 level per gradient
4:0	W	Window 8 color index in 64-color LUT [4:0]

Byte 2 default: 00h

Bit	Mode	Function
7	W	Blend function 0: Disable 1: Enable
6	W	Gradient function 0: Disable 1: Enable
5	W	Gradient direction 0: Horizontal 1: Vertical

4	W	Shadow/Border/3D button 0: Disable 1: Enable
3:1	W	Window 8 Type 000: Shadow Type 1 001: Shadow Type 2 010: Shadow Type3 011: Shadow Type 4 100: 3D Button Type 1 101: 3D Button Type 2 110: Reserved 111: Border
0	W	Window 8 Enable 0: Disable 1: Enable

Note: Chessboard function and gradient function cannot enable simultaneously

Window 9 Shadow/Border/Gradient
Address: 124h

Byte 0

Bit	Mode	Function
7	W	Window 9 shadow color index in 64-color LUT [4]
6	W	Window 9 border color index in 64-color LUT [4]
5:3	W	Window 9 shadow/border width or 3D button thickness in pixel unit 000~111: 1 ~ 8 pixel
2:0	W	Window 9 shadow/border height in line unit 000~111: 1 ~ 8 line It must be the same as bit [5:3] for 3D button thickness

PS: This is for non-rotary, rotate 270, rotate 90 and 180.

Byte 1

Bit	Mode	Function
7:4	W	Window 9 shadow color index in 64-color LUT [3:0] For 3D window, it is the left-top/bottom border color
3:0	W	Window 9 border color index in 64-color LUT [3:0] For 3D window, it is the right-bottom/top border color

Byte 2

Bit	Mode	Function
7	W	R Gradient Polarity 0: Decrease 1: Increase
6	W	G Gradient Polarity 0: Decrease 1: Increase
5	W	B Gradient Polarity 0: Decrease 1: Increase
4:3	W	Gradient level 00: 1 step per level 01: Repeat 2 step per level 10: Repeat 3 step per level 11: Repeat 4 step per level

2	W	Enable Red Color Gradient
1	W	Enable Green Color Gradient
0	W	Enable Blue Color Gradient

Window 9 start position
Address: 125h
Byte 0

Bit	Mode	Function
7:4	W	Window 9 horizontal start [11:8]
3:0	W	Window 9 vertical start [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 9 horizontal start [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 9 vertical start [7:0]

Window 9 end position
Address: 126h
Byte 0

Bit	Mode	Function
7:4	W	Window 9 horizontal end [11:8]
3:0	W	Window 9vertical end [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 9 horizontal end [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 9 vertical end [7:0]

Window 9 control
Address: 127h
Byte 0

Bit	Mode	Function
7	W	Window 9 shadow color index in 64-color LUT [5]
6	W	Window 9 border color index in 64-color LUT [5]
5	W	Window 9 color index in 64-color LUT [5]
4	W	Saturated color mode for gradient (refer to ADDR 162 for saturated color setting) 0: disable 1: enable
3	W	Reversed color mode for gradient (valid while [4] is 1) 0: disable 1: enable
2:0	--	Reserved

Byte 1

Bit	Mode	Function
7:5	W	111: 7 level per gradient 110: 6 level per gradient 101: 5 level per gradient 100: 4 level per gradient 011: 3 level per gradient 010: 2 level per gradient

		001: 1 level per gradient 000: 8 level per gradient
4:0	W	Window 9 color index in 64-color LUT [4:0]

Byte 2 default: 00h

Bit	Mode	Function
7	W	Blend function 0: Disable 1: Enable
6	W	Gradient function 0: Disable 1: Enable
5	W	Gradient direction 0: Horizontal 1: Vertical
4	W	Shadow/Border/3D button 0: Disable 1: Enable
3:1	W	Window 9 Type 000: Shadow Type 1 001: Shadow Type 2 010: Shadow Type3 011: Shadow Type 4 100: 3D Button Type 1 101: 3D Button Type 2 110: Reserved 111: Border
0	W	Window 9 Enable 0: Disable 1: Enable

Note: Chessboard function and gradient function cannot enable simultaneously

Window 4-1 Shadow/Border/Chessboard
Address: 200h

Byte 0

Bit	Mode	Function
7	W	Window 4-1 shadow color index in 64-color LUT [4]
6	W	Window 4-1 border color index in 64-color LUT [4]
5:3	W	Window 4-1 shadow/border width or 3D button thickness in pixel unit 000~111: 1 ~ 8 pixel
2:0	W	Window 4-1 shadow/border height in line unit 000~111: 1 ~ 8 line It must be the same as bit [5:3] for 3D button thickness

Byte 1

Bit	Mode	Function
7:4	W	Window 4-1 shadow color index in 64-color LUT [3:0] For 3D window, it is the left-top/bottom border color
3:0	W	Window 4-1 border color index in 64-color LUT [3:0] For 3D window, it is the right-bottom/top border color

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Window 4-1 start position

Address: 201h

Byte 0

Bit	Mode	Function
7:4	W	Window 4-1 horizontal start [11:8]
3:0	W	Window 4-1 vertical start [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 4-1 horizontal start [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 4-1 vertical start [7:0]

Window 4-1 end position

Address: 202h

Byte 0

Bit	Mode	Function
7:4	W	Window 4-1 horizontal end [11:8]
3:0	W	Window 4-1 vertical end [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 4-1 horizontal end [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 4-1 vertical end [7:0]

Window 4-1 control
Address: 203h
Byte 0

Bit	Mode	Function
7	W	Window 4-1 shadow color index in 64-color LUT [5]
6	W	Window 4-1 border color index in 64-color LUT [5]
5	W	Window 4-1 color index in 64-color LUT [5]
4:0	--	Reserved

Byte 1

Bit	Mode	Function
7:5	--	Reserved
4:0	W	Window 4-1 color index in 64-color LUT [4:0]

Byte 2 default: 00h

Bit	Mode	Function
7	W	Blend function 0: Disable 1: Enable
6:5	W	Reserved
4	W	Shadow/Border/3D button 0: Disable 1: Enable
3:1	W	Window 4-1 Type 000: Shadow Type 1 001: Shadow Type 2 010: Shadow Type3 011: Shadow Type 4 100: 3D Button Type 1 101: 3D Button Type 2 110: Reserved 111: Border
0	W	Window 4-1 Enable 0: Disable 1: Enable

Window 4-2 Shadow/Border/Chessboard
Address: 204h
Byte 0

Bit	Mode	Function
7	W	Window 4-2 shadow color index in 64-color LUT [4]
6	W	Window 4-2 border color index in 64-color LUT [4]
5:3	W	Window 4-2 shadow/border width or 3D button thickness in pixel unit 000~111: 1 ~ 8 pixel
2:0	W	Window 4-2 shadow/border height in line unit 000~111: 1 ~ 8 line It must be the same as bit [5:3] for 3D button thickness

Byte 1

Bit	Mode	Function
7:4	W	Window 4-2 shadow color index in 64-color LUT [3:0] For 3D window, it is the left-top/bottom border color

3:0	W	Window 4-2 border color index in 64-color LUT [3:0] For 3D window, it is the right-bottom/top border color

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Window 4-2 start position
Address: 205h

Byte 0

Bit	Mode	Function
7:4	W	Window 4-2 horizontal start [11:8]
3:0	W	Window 4-2 vertical start [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 4-2 horizontal start [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 4-2 vertical start [7:0]

Window 4-2 end position
Address: 206h

Byte 0

Bit	Mode	Function
7:4	W	Window 4-2 horizontal end [11:8]
3:0	W	Window 4-2 vertical end [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 4-2 horizontal end [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 4-2 vertical end [7:0]

Window 4-2 control
Address: 207h
Byte 0

Bit	Mode	Function
7	W	Window 4-2 shadow color index in 64-color LUT [5]
6	W	Window 4-2 border color index in 64-color LUT [5]
5	W	Window 4-2 color index in 64-color LUT [5]
4:0	--	Reserved

Byte 1

Bit	Mode	Function
7:5	--	Reserved
4:0	W	Window 4-2 color index in 64-color LUT [4:0]

Byte 2 default: 00h

Bit	Mode	Function
7	W	Blend function 0: Disable 1: Enable
6:5	W	Reserved
4	W	Shadow/Border/3D button 0: Disable 1: Enable
3:1	W	Window 4-2 Type 000: Shadow Type 1 001: Shadow Type 2 010: Shadow Type3 011: Shadow Type 4 100: 3D Button Type 1 101: 3D Button Type 2 110: Reserved 111: Border
0	W	Window 4-2 Enable 0: Disable 1: Enable

Window 4-3 Shadow/Border/Chessboard
Address: 208h
Byte 0

Bit	Mode	Function
7	W	Window 4-3 shadow color index in 64-color LUT [4]
6	W	Window 4-3 border color index in 64-color LUT [4]
5:3	W	Window 4-3 shadow/border width or 3D button thickness in pixel unit 000~111: 1 ~ 8 pixel
2:0	W	Window 4-3 shadow/border height in line unit 000~111: 1 ~ 8 line It must be the same as bit [5:3] for 3D button thickness

Byte 1

Bit	Mode	Function
7:4	W	Window 4-3 shadow color index in 64-color LUT [3:0] For 3D window, it is the left-top/bottom border color

3:0	W	Window 4-3 border color index in 64-color LUT [3:0] For 3D window, it is the right-bottom/top border color

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Window 4-3 start position
Address: 209h

Byte 0

Bit	Mode	Function
7:4	W	Window 4-3 horizontal start [11:8]
3:0	W	Window 4-3 vertical start [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 4-3 horizontal start [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 4-3 vertical start [7:0]

Window 4-3 end position
Address: 20Ah

Byte 0

Bit	Mode	Function
7:4	W	Window 4-3 horizontal end [11:8]
3:0	W	Window 4-3 vertical end [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 4-3 horizontal end [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 4-3 vertical end [7:0]

Window 4-3 control
Address: 20Bh
Byte 0

Bit	Mode	Function
7	W	Window 4-3 shadow color index in 64-color LUT [5]
6	W	Window 4-3 border color index in 64-color LUT [5]
5	W	Window 4-3 color index in 64-color LUT [5]
4:0	--	Reserved

Byte 1

Bit	Mode	Function
7:5	--	Reserved
4:0	W	Window 4-3 color index in 64-color LUT [4:0]

Byte 2 default: 00h

Bit	Mode	Function
7	W	Blend function 0: Disable 1: Enable
6:5	W	Reserved
4	W	Shadow/Border/3D button 0: Disable 1: Enable
3:1	W	Window 4-3 Type 000: Shadow Type 1 001: Shadow Type 2 010: Shadow Type3 011: Shadow Type 4 100: 3D Button Type 1 101: 3D Button Type 2 110: Reserved 111: Border
0	W	Window 4-3 Enable 0: Disable 1: Enable

Window 4-4 Shadow/Border/Chessboard
Address: 20Ch
Byte 0

Bit	Mode	Function
7	W	Window 4-4 shadow color index in 64-color LUT [4]
6	W	Window 4-4 border color index in 64-color LUT [4]
5:3	W	Window 4-4 shadow/border width or 3D button thickness in pixel unit 000~111: 1 ~ 8 pixel
2:0	W	Window 4-4 shadow/border height in line unit 000~111: 1 ~ 8 line It must be the same as bit [5:3] for 3D button thickness

Byte 1

Bit	Mode	Function
7:4	W	Window 4-4 shadow color index in 64-color LUT [3:0] For 3D window, it is the left-top/bottom border color

3:0	W	Window 4-4 border color index in 64-color LUT [3:0] For 3D window, it is the right-bottom/top border color

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Window 4-4 start position
Address: 20Dh

Byte 0

Bit	Mode	Function
7:4	W	Window 4-4 horizontal start [11:8]
3:0	W	Window 4-4 vertical start [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 4-4 horizontal start [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 4-4 vertical start [7:0]

Window 4-4 end position
Address: 20Eh

Byte 0

Bit	Mode	Function
7:4	W	Window 4-4 horizontal end [11:8]
3:0	W	Window 4-4 vertical end [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 4-4 horizontal end [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 4-4 vertical end [7:0]

Window 4-4 control
Address: 20Fh
Byte 0

Bit	Mode	Function
7	W	Window 4-4 shadow color index in 64-color LUT [5]
6	W	Window 4-4 border color index in 64-color LUT [5]
5	W	Window 4-4 color index in 64-color LUT [5]
4:0	--	Reserved

Byte 1

Bit	Mode	Function
7:5	--	Reserved
4:0	W	Window 4-4 color index in 64-color LUT [4:0]

Byte 2 default: 00h

Bit	Mode	Function
7	W	Blend function 0: Disable 1: Enable
6:5	W	Reserved
4	W	Shadow/Border/3D button 0: Disable 1: Enable
3:1	W	Window 4-4 Type 000: Shadow Type 1 001: Shadow Type 2 010: Shadow Type3 011: Shadow Type 4 100: 3D Button Type 1 101: 3D Button Type 2 110: Reserved 111: Border
0	W	Window 4-4 Enable 0: Disable 1: Enable

Window 4-5 Shadow/Border/Chessboard
Address: 210h
Byte 0

Bit	Mode	Function
7	W	Window 4-5 shadow color index in 64-color LUT [4]
6	W	Window 4-5 border color index in 64-color LUT [4]
5:3	W	Window 4-5 shadow/border width or 3D button thickness in pixel unit 000~111: 1 ~ 8 pixel
2:0	W	Window 4-5 shadow/border height in line unit 000~111: 1 ~ 8 line It must be the same as bit [5:3] for 3D button thickness

Byte 1

Bit	Mode	Function
7:4	W	Window 4-5 shadow color index in 64-color LUT [3:0] For 3D window, it is the left-top/bottom border color

3:0	W	Window 4-5 border color index in 64-color LUT [3:0] For 3D window, it is the right-bottom/top border color

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Window 4-5 start position
Address: 211h

Byte 0

Bit	Mode	Function
7:4	W	Window 4-5 horizontal start [11:8]
3:0	W	Window 4-5 vertical start [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 4-5 horizontal start [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 4-5 vertical start [7:0]

Window 4-5 end position
Address: 212h

Byte 0

Bit	Mode	Function
7:4	W	Window 4-5 horizontal end [11:8]
3:0	W	Window 4-5 vertical end [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 4-5 horizontal end [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 4-5 vertical end [7:0]

Window 4-5 control
Address: 213h
Byte 0

Bit	Mode	Function
7	W	Window 4-5 shadow color index in 64-color LUT [5]
6	W	Window 4-5 border color index in 64-color LUT [5]
5	W	Window 4-5 color index in 64-color LUT [5]
4:0	--	Reserved

Byte 1

Bit	Mode	Function
7:5	--	Reserved
4:0	W	Window 4-5 color index in 64-color LUT [4:0]

Byte 2 default: 00h

Bit	Mode	Function
7	W	Blend function 0: Disable 1: Enable
6:5	W	Reserved
4	W	Shadow/Border/3D button 0: Disable 1: Enable
3:1	W	Window 4-5 Type 000: Shadow Type 1 001: Shadow Type 2 010: Shadow Type3 011: Shadow Type 4 100: 3D Button Type 1 101: 3D Button Type 2 110: Reserved 111: Border
0	W	Window 4-5 Enable 0: Disable 1: Enable

Window 4-6 Shadow/Border/Chessboard
Address: 214h
Byte 0

Bit	Mode	Function
7	W	Window 4-6 shadow color index in 64-color LUT [4]
6	W	Window 4-6 border color index in 64-color LUT [4]
5:3	W	Window 4-6 shadow/border width or 3D button thickness in pixel unit 000~111: 1 ~ 8 pixel
2:0	W	Window 1 shadow/border height in line unit 000~111: 1 ~ 8 line It must be the same as bit [5:3] for 3D button thickness

Byte 1

Bit	Mode	Function
7:4	W	Window 4-6 shadow color index in 64-color LUT [3:0] For 3D window, it is the left-top/bottom border color

3:0	W	Window 4-6 border color index in 64-color LUT [3:0] For 3D window, it is the right-bottom/top border color

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Window 4-6 start position
Address: 215h

Byte 0

Bit	Mode	Function
7:4	W	Window 4-6 horizontal start [11:8]
3:0	W	Window 4-6 vertical start [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 4-6 horizontal start [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 4-6 vertical start [7:0]

Window 4-6 end position
Address: 216h

Byte 0

Bit	Mode	Function
7:4	W	Window 4-6 horizontal end [11:8]
3:0	W	Window 4-6 vertical end [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 4-6 horizontal end [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 4-6 vertical end [7:0]

Window 4-6 control
Address: 217h
Byte 0

Bit	Mode	Function
7	W	Window 4-6 shadow color index in 64-color LUT [5]
6	W	Window 4-6 border color index in 64-color LUT [5]
5	W	Window 4-6 color index in 64-color LUT [5]
4:0	--	Reserved

Byte 1

Bit	Mode	Function
7:5	--	Reserved
4:0	W	Window 4-6 color index in 64-color LUT [4:0]

Byte 2 default: 00h

Bit	Mode	Function
7	W	Blend function 0: Disable 1: Enable
6:5	W	Reserved
4	W	Shadow/Border/3D button 0: Disable 1: Enable
3:1	W	Window 4-6 Type 000: Shadow Type 1 001: Shadow Type 2 010: Shadow Type3 011: Shadow Type 4 100: 3D Button Type 1 101: 3D Button Type 2 110: Reserved 111: Border
0	W	Window 4-6 Enable 0: Disable 1: Enable

Window 4-7 Shadow/Border/Chessboard
Address: 218h
Byte 0

Bit	Mode	Function
7	W	Window 4-7 shadow color index in 64-color LUT [4]
6	W	Window 4-7 border color index in 64-color LUT [4]
5:3	W	Window 4-7 shadow/border width or 3D button thickness in pixel unit 000~111: 1 ~ 8 pixel
2:0	W	Window 4-7 shadow/border height in line unit 000~111: 1 ~ 8 line It must be the same as bit [5:3] for 3D button thickness

Byte 1

Bit	Mode	Function
7:4	W	Window 4-7 shadow color index in 64-color LUT [3:0] For 3D window, it is the left-top/bottom border color

3:0	W	Window 4-7 border color index in 64-color LUT [3:0] For 3D window, it is the right-bottom/top border color

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Window 4-7 start position
Address: 219h

Byte 0

Bit	Mode	Function
7:4	W	Window 4-7 horizontal start [11:8]
3:0	W	Window 4-7 vertical start [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 4-7 horizontal start [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 4-7 vertical start [7:0]

Window 4-7 end position
Address: 21Ah

Byte 0

Bit	Mode	Function
7:4	W	Window 4-7 horizontal end [11:8]
3:0	W	Window 4-7 vertical end [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 4-7 horizontal end [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 4-7 vertical end [7:0]

Window 4-7 control
Address: 21Bh
Byte 0

Bit	Mode	Function
7	W	Window 4-7 shadow color index in 64-color LUT [5]
6	W	Window 4-7 border color index in 64-color LUT [5]
5	W	Window 4-7 color index in 64-color LUT [5]
4:0	--	Reserved

Byte 1

Bit	Mode	Function
7:5	--	Reserved
4:0	W	Window 4-7 color index in 64-color LUT [4:0]

Byte 2 default: 00h

Bit	Mode	Function
7	W	Blend function 0: Disable 1: Enable
6:5	W	Reserved
4	W	Shadow/Border/3D button 0: Disable 1: Enable
3:1	W	Window 4-7 Type 000: Shadow Type 1 001: Shadow Type 2 010: Shadow Type3 011: Shadow Type 4 100: 3D Button Type 1 101: 3D Button Type 2 110: Reserved 111: Border
0	W	Window 4-7 Enable 0: Disable 1: Enable

Window 4-8 Shadow/Border/Chessboard
Address: 21Ch
Byte 0

Bit	Mode	Function
7	W	Window 4-8 shadow color index in 64-color LUT [4]
6	W	Window 4-8 border color index in 64-color LUT [4]
5:3	W	Window 4-8 shadow/border width or 3D button thickness in pixel unit 000~111: 1 ~ 8 pixel
2:0	W	Window 4-8 shadow/border height in line unit 000~111: 1 ~ 8 line It must be the same as bit [5:3] for 3D button thickness

Byte 1

Bit	Mode	Function
7:4	W	Window 4-8 shadow color index in 64-color LUT [3:0] For 3D window, it is the left-top/bottom border color

3:0	W	Window 4-8 border color index in 64-color LUT [3:0] For 3D window, it is the right-bottom/top border color

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Window 4-8 start position
Address: 21Dh

Byte 0

Bit	Mode	Function
7:4	W	Window 4-8 horizontal start [11:8]
3:0	W	Window 4-8 vertical start [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 4-8 horizontal start [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 4-8 vertical start [7:0]

Window 4-8 end position
Address: 21Eh

Byte 0

Bit	Mode	Function
7:4	W	Window 4-8 horizontal end [11:8]
3:0	W	Window 4-8 vertical end [11:8]

Byte 1

Bit	Mode	Function
7:0	W	Window 4-8 horizontal end [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 4-8 vertical end [7:0]

Window 4-8 control
Address: 21Fh
Byte 0

Bit	Mode	Function
7	W	Window 4-8 shadow color index in 64-color LUT [5]
6	W	Window 4-8 border color index in 64-color LUT [5]
5	W	Window 4-8 color index in 64-color LUT [5]
4:0	--	Reserved

Byte 1

Bit	Mode	Function
7:5	--	Reserved
4:0	W	Window 4-8 color index in 64-color LUT [4:0]

Byte 2 default: 00h

Bit	Mode	Function
7	W	Blend function 0: Disable 1: Enable
6:5	W	Reserved
4	W	Shadow/Border/3D button 0: Disable 1: Enable
3:1	W	Window 4-8 Type 000: Shadow Type 1 001: Shadow Type 2 010: Shadow Type3 011: Shadow Type 4 100: 3D Button Type 1 101: 3D Button Type 2 110: Reserved 111: Border
0	W	Window 4-8 Enable 0: Disable 1: Enable

Window Special Control

Window 0

Address: 150h

Byte 0

Bit	Mode	Function
7:0	W	Window 0 RED saturated color [7:0]

Byte 1

Bit	Mode	Function
7:0	W	Window 0 GRN saturated color [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 0 BLU saturated color [7:0]

Address: 151h

Byte 0

Bit	Mode	Function
7:0	W	Reserved

Byte 1

Bit	Mode	Function
7:0	W	Reserved

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Window 1

Address: 152h

Byte 0

Bit	Mode	Function
7:0	W	Reserved

Byte 1

Bit	Mode	Function
7:0	W	Reserved

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Address: 153h

Byte 0

Bit	Mode	Function
7:0	W	Reserved

Byte 1

Bit	Mode	Function
7:0	W	Reserved

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Window 2

Address: 154h

Byte 0

Bit	Mode	Function
7:0	W	Reserved

Byte 1

Bit	Mode	Function
7:0	W	Reserved

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Address: 155h

Byte 0

Bit	Mode	Function
7:0	W	Reserved

Byte 1

Bit	Mode	Function
7:0	W	Reserved

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Window 3

Address: 156h

Byte 0

Bit	Mode	Function
7:0	W	Reserved

Byte 1

Bit	Mode	Function
7:0	W	Reserved

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Address: 157h

Byte 0

Bit	Mode	Function
7:0	W	Reserved

Byte 1

Bit	Mode	Function
7:0	W	Reserved

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Window 4

Address: 158h

Byte 0

Bit	Mode	Function
7:0	W	Reserved

Byte 1

Bit	Mode	Function

7:0	W	Reserved

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Address: 159h

Byte 0

Bit	Mode	Function
7:0	W	Reserved

Byte 1

Bit	Mode	Function
7:0	W	Reserved

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Window 5

Address: 15Ah

Byte 0

Bit	Mode	Function
7:0	W	Window 5 RED saturated color [7:0]

Byte 1

Bit	Mode	Function
7:0	W	Window 5 GRN saturated color [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 5 BLU saturated color [7:0]

Address: 15Bh

Byte 0

Bit	Mode	Function
7:0	W	Reserved

Byte 1

Bit	Mode	Function
7:0	W	Reserved

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Window 6

Address: 15Ch

Byte 0

Bit	Mode	Function
7:0	W	Window 6 RED saturated color [7:0]

Byte 1

Bit	Mode	Function
7:0	W	Window 6 GRN saturated color [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 6 BLU saturated color [7:0]

Address: 15Dh

Byte 0

Bit	Mode	Function
7:0	W	Reserved

Byte 1

Bit	Mode	Function
7:0	W	Reserved

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Window 7

Address: 15Eh

Byte 0

Bit	Mode	Function
7:0	W	Window 7 RED saturated color [7:0]

Byte 1

Bit	Mode	Function
7:0	W	Window 7 GRN saturated color [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 7 BLU saturated color [7:0]

Address: 15Fh

Byte 0

Bit	Mode	Function
7:0	W	Reserved

Byte 1

Bit	Mode	Function
7:0	W	Reserved

Byte 2

Bit	Mode	Function
7:0	W	Reserved

Window 8

Address: 160h

Byte 0

Bit	Mode	Function
7:0	W	Window 8 RED saturated color [7:0]

Byte 1

Bit	Mode	Function
7:0	W	Window8 GRN saturated color [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 8 BLU saturated color [7:0]

Address: 161h

Byte 0 default: 0xxx_xxxx

Bit	Mode	Function

7	R/W	Window8 Chessboard color function: 0:disable (default) 1:enble
6:0	--	Reserved

Byte 1

Bit	Mode	Function
7:6	--	Reserved
5:0	R/W	Window8 Chessboard color1

Byte 2

Bit	Mode	Function
7:6	--	Reserved
5:0	R/W	Window8 Chessboard color2

Window 9
Address: 162h

Byte 0

Bit	Mode	Function
7:0	W	Window 9 RED saturated color [7:0]

Byte 1

Bit	Mode	Function
7:0	W	Window 9 GRN saturated color [7:0]

Byte 2

Bit	Mode	Function
7:0	W	Window 9 BLU saturated color [7:0]

Address: 163h

Byte 0

Bit	Mode	Function
7	W	Chessboard color function: 0:disable 1:enble
6:0	--	Reserved

Byte 1

Bit	Mode	Function
7:6	--	Reserved
5:0	W	Chessboard color1

Byte 2

Bit	Mode	Function
7:6	--	Reserved
5:0	W	Chessboard color2

Window Special Control
Window 4-1
Address: 164h

Byte 0

Bit	Mode	Function
7	W	Window 4-1 Chessboard color function:

		0:disable 1:enble **(Reserved)**
6:0	--	Reserved

Byte 1

Bit	Mode	Function
7:6	--	Reserved
5:0	W	Chessboard color1 **(Reserved)**

Byte 2

Bit	Mode	Function
7:6	--	Reserved
5:0	W	Chessboard color2 **(Reserved)**

Window 4-2
Address: 166h

Byte 0

Bit	Mode	Function
7	W	Window 4-2 Chessboard color function: 0:disable 1:enble **(Reserved)**
6:0	--	Reserved

Byte 1

Bit	Mode	Function
7:6	--	Reserved
5:0	W	Chessboard color1 **(Reserved)**

Byte 2

Bit	Mode	Function
7:6	--	Reserved
5:0	W	Chessboard color2 **(Reserved)**

Window 4-3
Address: 168h

Byte 0

Bit	Mode	Function
7	W	Window 4-3 Chessboard color function: 0:disable 1:enble
6:0	--	Reserved

Byte 1

Bit	Mode	Function
7:6	--	Reserved
5:0	W	Chessboard color1

Byte 2

Bit	Mode	Function
7:6	--	Reserved
5:0	W	Chessboard color2

Window 4-4
Address: 16Ah
Byte 0

Bit	Mode	Function
7	W	Window 4-4 Chessboard color function: 0:disable 1:enble
6:0	--	Reserved

Byte 1

Bit	Mode	Function
7:6	--	Reserved
5:0	W	Chessboard color1

Byte 2

Bit	Mode	Function
7:6	--	Reserved
5:0	W	Chessboard color2

Window 4-5
Address: 16Ch
Byte 0

Bit	Mode	Function
7	W	Window 4-5 Chessboard color function: 0:disable 1:enble
6:0	--	Reserved

Byte 1

Bit	Mode	Function
7:6	--	Reserved
5:0	W	Chessboard color1

Byte 2

Bit	Mode	Function
7:6	--	Reserved
5:0	W	Chessboard color2

Window 4-6
Address: 16Eh
Byte 0

Bit	Mode	Function
7	W	Window 4-6 Chessboard color function: 0:disable 1:enble
6:0	--	Reserved

Byte 1

Bit	Mode	Function
7:6	--	Reserved
5:0	W	Chessboard color1

Byte 2

Bit	Mode	Function
7:6	--	Reserved
5:0	W	Chessboard color2

Window 4-7
Address: 170h

Byte 0

Bit	Mode	Function
7	W	Window 4-7 Chessboard color function: 0:disable 1:enble
6:0	--	Reserved

Byte 1

Bit	Mode	Function
7:6	--	Reserved
5:0	W	Chessboard color1

Byte 2

Bit	Mode	Function
7:6	--	Reserved
5:0	W	Chessboard color2

Window 4-8
Address: 172h

Byte 0

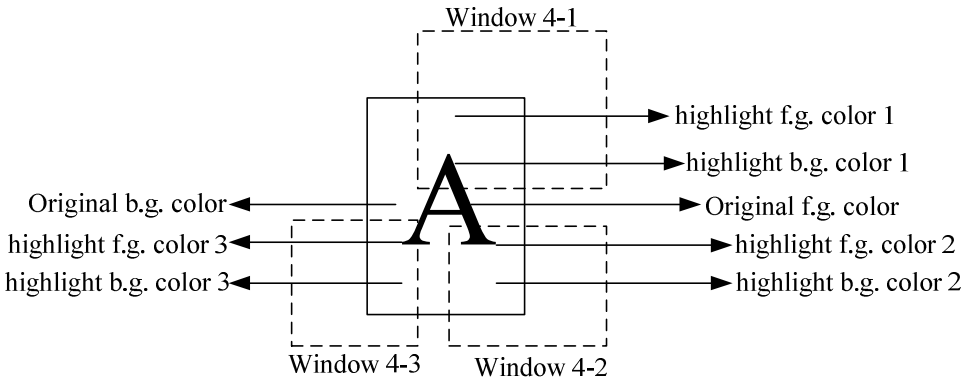
Bit	Mode	Function
7	W	Window 4-8 Chessboard color function: 0:disable 1:enble
6:0	--	Reserved

Byte 1

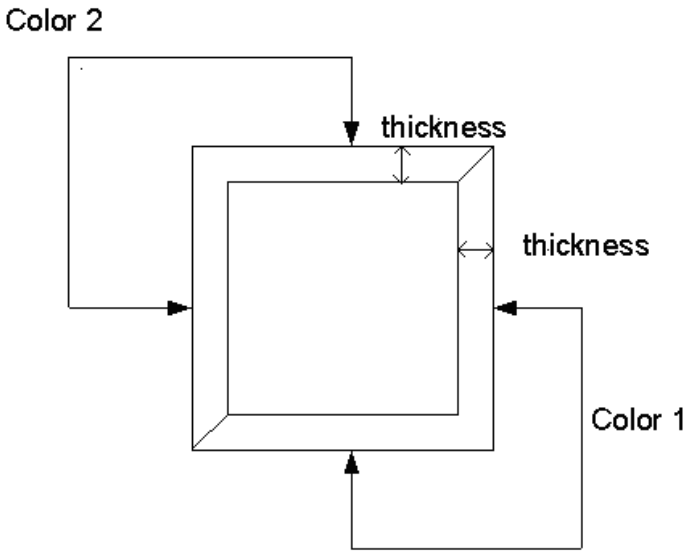
Bit	Mode	Function
7:6	--	Reserved
5:0	W	Chessboard color1

Byte 2

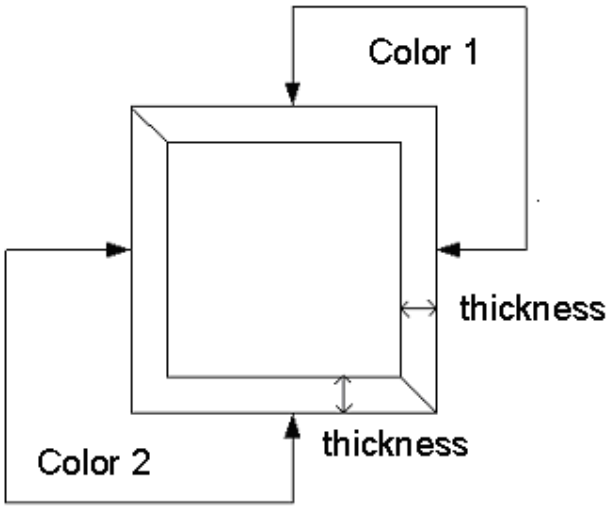
Bit	Mode	Function
7:6	--	Reserved
5:0	W	Chessboard color2



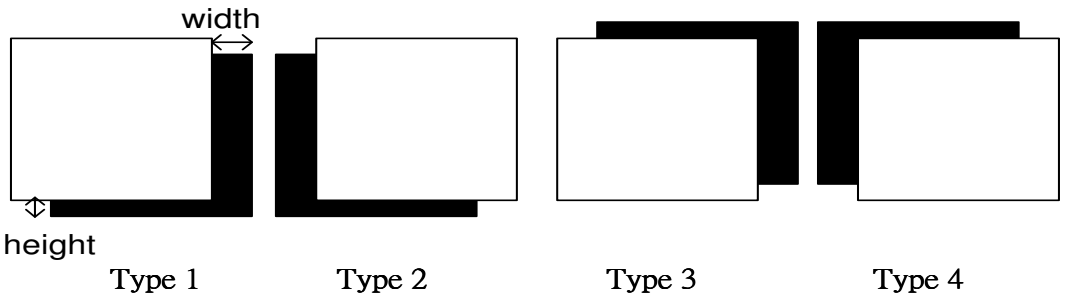
Highlight function of window4-1, window4-2 and window4-3



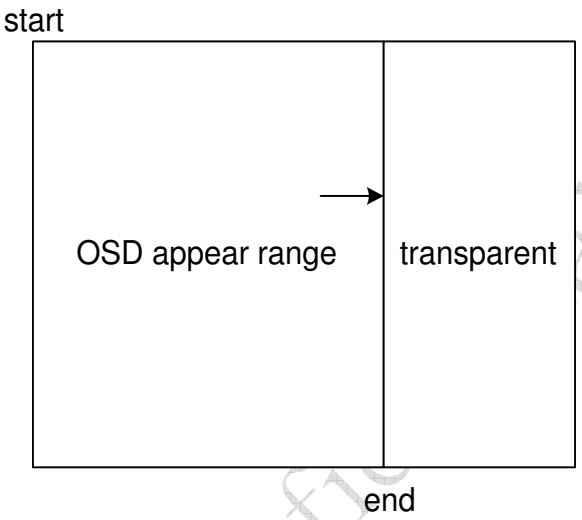
3D Button Type 1



3D Button Type 2



Shadow in all direction



Window mask fade/in out function

Frame control registers

Address: 000h

Byte 0

Bit	Mode	Function
7:0	R/W	Vertical Delay [10:3] The bits define the vertical starting address. Total 2048 step unit: 1 line

Vertical delay minimum should set 1

Byte 1

Bit	Mode	Function
7:0	R/W	Horizontal Delay [9:2] The bits define the horizontal starting address. Total 1024 step unit:4 pixels

Horizontal delay minimum should set 2

Byte 2 default: xxxx_xxx0b

Bit	Mode	Function
7:6	R/W	Horizontal Delay bit [1:0]
5:3	R/W	Vertical Delay [2:0]
2:1	R/W	Display zone, for smaller character width 00: middle 01: left 10: right 11: reserved
0	R/W	OSD enable 0: OSD circuit is inactivated 1: OSD circuit is activated

- When OSD is disabled, Double Width (address 0x003 Byte1[1]) must be disabled to save power.
- These three bytes have their own double-buffer.

Address 001h ~ Address002h are reserved

Address: 002h

Byte 2 Default: 00h

Bit	Mode	Function
7	R/W	Reserved to 0
6	R/W	Window 9 priority 0: lower than font 1: higher than font
5	R/W	Window 8 priority 0: lower than font 1: higher than font
4	R/W	blending type 2 match color 0: match blending type 2 color bit[3:0] 1: match blending type 2 color bit[5:0]
3:2	R/W	Blending color from 64-color LUT [5:4] (blending type 2)
1:0	R/W	Char shadow/border color [5:4]

Address: 003h

Byte 0 Default: 00h

Bit	Mode	Function
7	R/W	Specific color blending (blending type 2) 0: Disable 1: Enable
6:5	R/W	Window 7special function 00: disable 01: blending (blending type 3) 10: window 7 mask region appear 11: window 7 mask region transparent
4	R/W	OSD vertical start input signal source select 0: Select DVS as OSD VSYNC input 1: Select ENA as OSD VSYNC input
3:0	R/W	Blending color from 64-color LUT [3:0] (blending type 2)

Byte 1 Default: 00h

Bit	Mode	Function
7:4	R/W	Char shadow/border color [3:0]
3: 2	R/W	Alpha blending type (blending type 1) 00: Disable alpha blending 01: Only window blending 10: All blending 11: Window and Character background blending
1	R/W	Double width enable (For all OSD including windows and characters) 0: Normal 1: Double
0	R/W	Double Height enable (For all OSD including windows and characters) 0: Normal 1: Double

Total blending area = blending type1 area + blending type 2 area + blending type 3 area

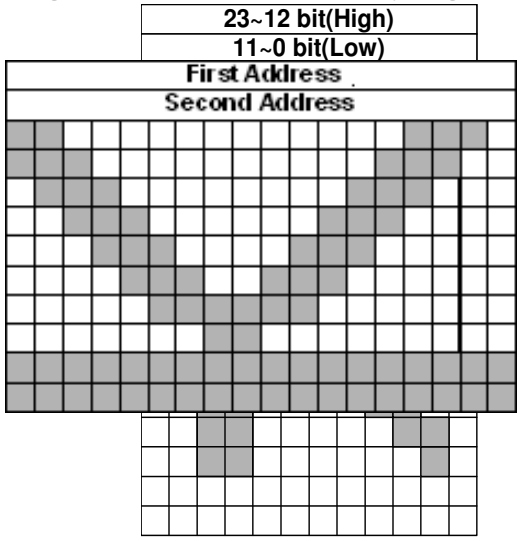
Byte 2 Default: 00h

Bit	Mode	Function
7:6	R/W	**Font downloaded swap control** 0x: No swap

		10: CCW 11: CW
5	R	**Buffer Empty** 0: Empty 1: Not Empty
4	R	**Buffer Valid** 0: Done 1: Buffer is writing to SDRAM
3	R/W	**Reset Buffer** Write 1 to reset and auto-clear after finished.
2	R/W	**Hardware Rotation Enable** 0: Disable 1: Enable (Default) OSD compression function must be enabled simultaneously.
1	R/W	**Global Blinking Enable** 0: Disable 1: Enable
0	R/W	**Rotation** 0: Normal (data latch 24 bit per 24 bit) 1: Rotation (data latch 18 bit per 24 bit)

Bit	7	6	5	4	3	2	1	0
Firmware	A	B	C	D	E	F	G	H
CW	A	E	B	F	C	G	D	H
CCW	E	A	F	B	G	C	H	D

Figure 3 Non-rotated memory alignments



23 **6**

Figure 4 Rotated memory alignments

Base address offset

Address: 004h

Byte 0

Bit	Mode	Function
7:0	R/W	Font Select Base Address[7:0]

Byte 1

Bit	Mode	Function
7:4	R/W	Font Select Base Address[11:8]
3:0	R/W	Font Base Address[3:0]

Byte 2

Bit	Mode	Function
7:0	R/W	Font Base Address[11:4]

When OSD Special Function for POP-ON is enabled (OSD[008]), Font Select Base Address here will not be effective.

OSD Compression

Address: 005h

Byte 0

Bit	Mode	Function
7:4	R/W	4-bit value for VLC code 0
3:0	R/W	4-bit value for VLC code 100

Byte 1

Bit	Mode	Function
7:4	R/W	4-bit value for VLC code 1010
3:0	R/W	4-bit value for VLC code 1011

Byte 2

Bit	Mode	Function
7:4	R/W	4-bit value for VLC code 1100
3:0	R/W	4-bit value for VLC code 1101 0

Address: 006h

Byte 0

Bit	Mode	Function
7:4	R/W	4-bit value for VLC code 1101 1

3:0	R/W	4-bit value for VLC code 1110 0

Byte 1

Bit	Mode	Function
7:4	R/W	4-bit value for VLC code 1110 10
3:0	R/W	4-bit value for VLC code 1110 11

Byte 2

Bit	Mode	Function
7:4	R/W	4-bit value for VLC code 1111 00
3:0	R/W	4-bit value for VLC code 1111 01

Address: 007h

Byte 0

Bit	Mode	Function
7:4	R/W	4-bit value for VLC code 1111 100
3:0	R/W	4-bit value for VLC code 1111 101

Byte 1

Bit	Mode	Function
7:4	R/W	4-bit value for VLC code 1111 110
3:0	R/W	4-bit value for VLC code 1111 1110

Byte 2 default: xxxx_x0000b

Bit	Mode	Function
7:3	--	reserved
2	R/W	Write 3-byte decoded-data into SRAM each time for acceleration 0: disable 1: enable active when compresssion and hardware rotation function are both enabled.
1	R/W	Decide to decode which kind of data 0: decode compressed data 1: decode non-compressed data
0	R/W	OSD compression (4bit/symbol, VLC code 1111_1111 represents the end of data) (only for SRAM) 0: disable 1: enable

Note:

1. If enable OSD compression or auto load (double buffer), only one byte can be read after writing address at 0x90, 0x91.
2. For OSD compression, MSB 4 bits of original byte is first transferred to corresponding VLC code, and then LSB 4 bits is transferred. VLC code is placed from LSB to MSB of compression font. For example, 4-bit value for VLC code 1100 is 4'b0101, and 4-bit value for VLC code 100 is 4'b0001. Original data 0x15 is transferred to compression x0011001.
3. OSD double buffer and compression can't be enabled simultaneous.
4. When power-down mode or lack of crystal clock, OSD compression font can't be write.
5. After OSD enable, it is better to delay 1 DVS to start writing OSD compression data.
6. While decoding non-compressed data, it is necessary to reset compression function while decode is done. Because this is the only way to reset compression circuit and ready for next decode job
7. ADDR007 Byte2 bit[2] is designed to accelerate the speed of writing compression/rotation data into SRAM. When the bit is enabled, 3-byte data will be written to SRAM for each cycle and it is supposed to be used while **compression** and **hardware rotation function** are both enabled. Besides, the result of the acceleration will have the best performance while the burst write mode is enabled at the same time. However the burst write setting should be constrained by the two conditions as following:
   (1)Time of MCU write cycle > 18 x ($T_{m2pll/2}$)
   (2)Time of MCU write cycle > ((4 x ($T_{m2pll/2}$) + 9 x T_{DCLK}) x 9) / 7

OSD Special Function

Address: 008h

Byte 0 Default: 0x00

Bit	Mode	Function

7	R/W	OSD Special Function Enable 0: Disable 1: Enable
6	R/W	OSD Special Function Select (Effective only when Bit[7]=1) 0: ROLL-UP 1: POP-ON
5	R/W	OSD Vertical Boundary Function Enable 0: Disable 1: Enable
4	R/W	Reserved to 0
3	R/W	OSD 2 font function 0: disable (row & font select command refer to the FRAM ADDR 004 1: enable (row command of font A refer to the ADDR 008 byte 1 row command of font B refer to the ADDR 008 byte 2 font select command of font A refer to the ADDR 009 FS0 font select command of font B refer to the ADDR 009 FS1)
2:1	R/W	Reserved to 0
0	R/W	Display Base Select (Effective only when Bit[7:6]=11`b) 0: Base 0 1: Base 1

Byte 1 Default: 0x00

Bit	Mode	Function
7:0	R/W	Row Command Base 0 [7:0]

Byte 2 Default: 0x00

Bit	Mode	Function
7:0	R/W	Row Command Base 1 [7:0]

Address: 009h

Byte 0 Default: 0x00

Bit	Mode	Function
7:4	R/W	Font Select Base 0 [11:8]
3:0	R/W	Font Select Base 1 [11:8]

Byte 1 Default: 0x00

Bit	Mode	Function
7:0	R/W	Font Select Base 0 [7:0]

Byte 2 Default: 0x00

Bit	Mode	Function
7:0	R/W	Font Select Base 1 [7:0] (Not effective when ROLL-UP)

Address: 00Ah

Byte 0 Default: 0x00

Bit	Mode	Function
7	R/W	Reserved
6:4	R/W	OSD Vertical Upper Boundary [10:8]
3	R/W	Reserved
2:0	R/W	OSD Vertical Lower Boundary [10:8]

Byte 1 Default: 0x00

Bit	Mode	Function
7:0	R/W	OSD Vertical Upper Boundary [7:0]

Byte 2 Default: 0x00

Bit	Mode	Function
7:0	R/W	OSD Vertical Lower Boundary [7:0]

Address: 00Bh

Byte 0 Default: 0x00

Bit	Mode	Function
7	R/W	Font Base Address[12]
6	R/W	Window 6 Special Blending Function 0: OFF 1: ON
5:4	R/W	Blending Type of Window 7 00: NO Blending for both F/B 01: NO Blending for Foreground 10: NO Blending for Background 11: Both Blending for F/B
3:2	R/W	Blending Type of Window 6 00: NO Blending for both F/B 01: NO Blending for Foreground 10: NO Blending for Background 11: Both Blending for F/B
1:0	--	Reserved

Byte 1 Default: 0x00

Bit	Mode	Function
7	R/W	2-bit font char select offset [7]
6:0	R/W	2-bit font char select offset [6:0]

Byte 2 Default: 0x00

Bit	Mode	Function
7:0	--	Reserved

Address: 00Ch

Byte 0 Default: 0x00

Bit	Mode	Function
7:4	R/W	FONT A horizontal delay [11:8]
3:0	R/W	FONT A vertical delay [11:8]

Byte 1 Default: 0x00

Bit	Mode	Function
7:0	R/W	FONT A horizontal delay [7:0]

Byte 2 Default: 0x00

Bit	Mode	Function
7:0	R/W	FONT A vertical delay [7:0]

Address: 00Dh

Byte 0 Default: 0x00

Bit	Mode	Function
7:4	R/W	FONT B horizontal delay [11:8] (valid only while 2 font function enabled)
3:0	R/W	FONT B vertical delay [11:8] (valid only while 2 font function enabled)

Byte 1 Default: 0x00

Bit	Mode	Function
7:0	R/W	FONT B horizontal delay [7:0] (valid only while 2 font function enabled)

Byte 2 Default: 0x00

Bit	Mode	Function
7:0	R/W	FONT B vertical delay [7:0] (valid only while 2 font function enabled)

Address: 00Eh

Byte 0 default: x0xx_xxxx

Bit	Mode	Function
7	--	Reserved

6	R/W	Highlight function for window4-1 0: Disable (default) 1: Enable
5:0	R/W	Foreground color1 for highlight

Byte 1

Bit	Mode	Function
7	--	Reserved
6	--	Reserved
5:0	R/W	Background color1 for highlight

Note: Background color1[3:0]=4'b0000 is special for transparent

Byte 2

Bit	Mode	Function
7	--	Reserved
6	--	Reserved
5:0	R/W	Character border/shadow color1 for highlight

Address: 00Fh

Byte 0 default: x0xx_xxxx

Bit	Mode	Function
7	--	Reserved
6	R/W	Highlight function for window4-2 0: Disable (default) 1: Enable
5:0	R/W	Foreground color2 for highlight

Byte 1

Bit	Mode	Function
7	--	Reserved
6	--	Reserved
5:0	R/W	Background color2 for highlight

Note: Background color2[3:0]=4'b0000 is special for transparent

Byte 2

Bit	Mode	Function
7	--	Reserved
6	--	Reserved
5:0	R/W	Character border/shadow color2 for highlight

Address: 010h

Byte 0 default: x0xx_xxxx

Bit	Mode	Function
7	--	Reserved
6	R/W	Highlight function for window4-3 0: Disable (default) 1: Enable
5:0	R/W	Foreground color3 for highlight

Byte 1

Bit	Mode	Function
7	--	Reserved
6	--	Reserved
5:0	R/W	Background color3for highlight

Note: Background color3[3:0]=4'b0000 is special for transparent

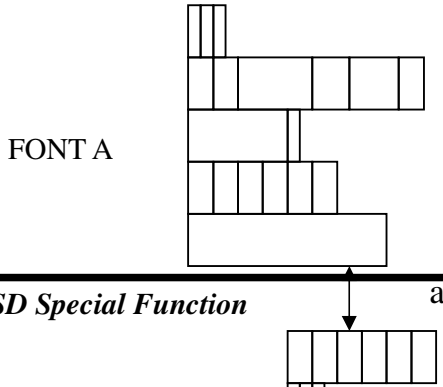
Byte 2

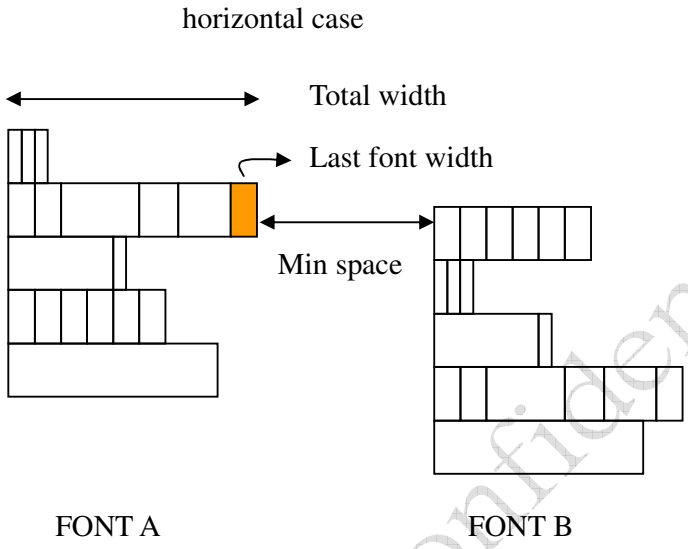
Bit	Mode	Function
7	--	Reserved
6	--	Reserved
5:0	R/W	Character border/shadow color3 for highlight

Note: 1. This highlight function only supports 1-bit font & blank.
2. If highlight function on, the color of blank will become foreground color.
3. The priority of these three windows: window4-3 > window4-2 >window4-1. If these three windows are overlapped, the highlight color of foreground, background and character border/shadow would be only displayed the setting of window4-3.

Note:
1. When OSD Special Function for POP-ON is enabled, Font Select Base Address in OSD[004] will not be effective anymore.
2. When OSD Vertical Boundary Function is enabled, OSD image above upper boundary and below lower boundary will be invisible.
3. When ROLL-UP function is enabled, OSD will always start from the row-command pointed by Base0, and after the row-command pointed by Base1 has been dealt with, the next row-command will be the first one in OSD SRAM. Row-command processing will terminate in the row-command before the one pointed by Base0. (For example, R1 is pointed by Base0, and R5 is pointed by Base1. OSD will show R1 as the first row, followed by R2, R3, R4, R5, and R0 as last row.)
4. When POP-ON function is enabled, OSD will start from the row command pointed by the base selected as display base(selected by OSD[008][0.0]), and terminate when end-command is encountered. That is, all row-command will be separated into two non-overlay subset which is enclosed by the row-command pointed by base and end-command.
5. While 2 font function is enabled (selected by OSD [008][0.3]), FONT A & FONT B can't be overlapped
6. While 2 font function is enabled (selected by OSD [008][0.3]), for vertical case, lines between of FONT A & FONTB must be larger than 1
7. While 2 font function is enabled (selected by OSD [008][0.3]), for horizontal case, pixels between of FONT A & FONT B must follow the rules as following:

Vertical case





Min space = 13

Wirte-Protection Function & Bit Swap Function
Address: 011h
Byte 0

Bit	Mode	Function
7:6	--	Reserved
5	**R/W**	**Blank Command Byte2-Bit[5:4] Enable/Disable** **0: Enable** **1: Disable**
4	R/W	SRAM read function when osd off:

		0: Disable (default) 1: Enable When function is on, need to send 2 times' read command，the 2nd data to be read out is the right value.
3	R/W	2-bit font character command byte0-bit[4]/byte1-bit[7] swap function: 0: Disable (default) 1: Enable
2:1	R/W	Byte selection control for write-protection function: 00: byte 0 01: byte 1 10: byte 2 11: reserved
0	R/W	Write-protection function for font SRAM when osd on: 0: Disable (default) 1: Enable

Byte 1

Bit	Mode	Function
7	R/W	Bit[7] write-protection: 0: Disable (default) 1: Enable
6	R/W	Bit[6] write-protection: 0: Disable (default) 1: Enable
5	R/W	Bit[5] write-protection: 0: Disable (default) 1: Enable
4	R/W	Bit[4] write-protection: 0: Disable (default) 1: Enable
3	R/W	Bit[3] write-protection: 0: Disable (default) 1: Enable
2	R/W	Bit[2] write-protection: 0: Disable (default) 1: Enable
1	R/W	Bit[1] write-protection: 0: Disable (default) 1: Enable
0	R/W	Bit[0] write-protection: 0: Disable (default) 1: Enable

Byte 2

Bit	Mode	Function
7:0	--	Reserved

OSD SRAM (Map and font registers)

R0	R1	R2	….		Rn	End		
C01	C02	B03	C04	…	C11	C12	C13	…
…								
…								

…	Cn1	Cn2	…	1-bit font start	…
…					
…	2-bit font start		…		
…					
4-bit font start	…				
…					
…					

20k bytes SRAM

1. Row Command

R0	R1	R2	R3	R….	Rn	End

Row Command R0~Rn represent the start of new row. Each command contains 3 bytes data which define the length of a row and other attributes. OSD End Command represent the end of OSD. R0 is set in address 0 of SRAM.

2. Character/Blank Command (Font Select)

Character Command is used to select which character font is show. Each command contains three bytes which specify its attribute and 1,2 or 4bit per pixel. Blank Command represents blank pixel to separate the preceding character and following character. Use two or more Blank Command if the character distance exceeds 255 pixel.

The Font Select Base Address in Frame Control Register represents the address of the first character in Row 0, that is, C01 in the above figure. The following character/blank is write in the next address. C11 represents the first character in Row1, C12 represents the second character in Row1, and so on.

The address of the first character Cn1 in Row n = Font Select Base Address + Row 0 font base length + Row 1 font base length + …+Row n-1 font base length.

3. Font

User fonts are stored as bit map data. For normal font, one font has 12x18 pixel, and for rotation font, one has 18x12 pixel. One pixel use 1, 2 or 4 bits.

For 12x18 font,
One 1-bit font requires 9 * 24bit SRAM
One 2-bit font requires 18 * 24bit SRAM
One 4-bit font requires 36 * 24bit SRAM

For 18x12 font,
One 1-bit font requires 9 * 24bit SRAM
One 2-bit font requires 18 * 24bit SRAM
One 4-bit font requires 36 * 24bit SRAM

Font Base Address in Frame Control Register point to the start of 1-bit font.

For normal (12x18) font:
1-bit Font, if CS = 128, Real Address of Font = Font Base Address + 9 * 128
2-bit Font, if CS = 128, Real Address of Font = Font Base Address + 18 * 128
4-bit Font, if CS = 128, Real Address of Font = Font Base Address + 36 * 128

For rotational (18x12) font:
1-bit Font, if CS = 128, Real Address of Font = Font Base Address + 9 * 128
2-bit Font, if CS = 128, Real Address of Font = Font Base Address + 18 * 128
4-bit Font, if CS = 128, Real Address of Font = Font Base Address + 36 * 128
where CS is Character Selector in Character Command.

Note that Row Command, Font Select and Font share the same OSD SRAM.

When we download the font, we have to set the Frame control 003h byte2 [7:6] to set the method of hardware bit swap. If the OSD is Counter-Clock-Wise rotated, we have to set to 0x01 (the 8 bits of every byte of font SRAM downloaded by firmware will be in a sequence of "7 5 3 1 6 4 2 0" (from MSB to LSB) and should be rearranged to "7 6 5 4 3 2 1 0" by hardware). If it is Clock-Wise rotated, we have to set to 0x10 (the 8 bits of every byte of font SRAM downloaded by firmware will be in a sequence of "6 4 2 0 7 5 3 1" (from MSB to LSB) and should be rearranged to "7 6 5 4 3 2 1 0" by hardware). After we finish the downloading or if we don't have to rotate the OSD, we have to set it to 0x00.

Row Command

Byte 0

Bit	Mode	Function
7	W	1: Row Start Command 0: OSD End Command Each row must start with row-command, last word of OSD map must be end-command
6	R/W	VBI OSD function enable (not valid while rotation enabled) 0: normal OSD function as usual 1: support VBI OSD functions like underline, B/F separated blink and 512 fonts select
5	W	1-bit font selection 0: font select from 0-511 1: font select from 512-1023
4:2	W	Character border/shadow 000: None 001: Border 100: Shadow (left-top) 101: Shadow (left-bottom) 110: Shadow (right-top) 111: Shadow (right-bottom)
1	W	Double character width 0: x1 1: x2
0	W	Double character height 0: x1 1: x2

Byte 1

Bit	Mode	Function
7:3	W	Row height (1~32)
2:0	W	Column space 0~7 pixel column space When Char is doubled, so is column space.

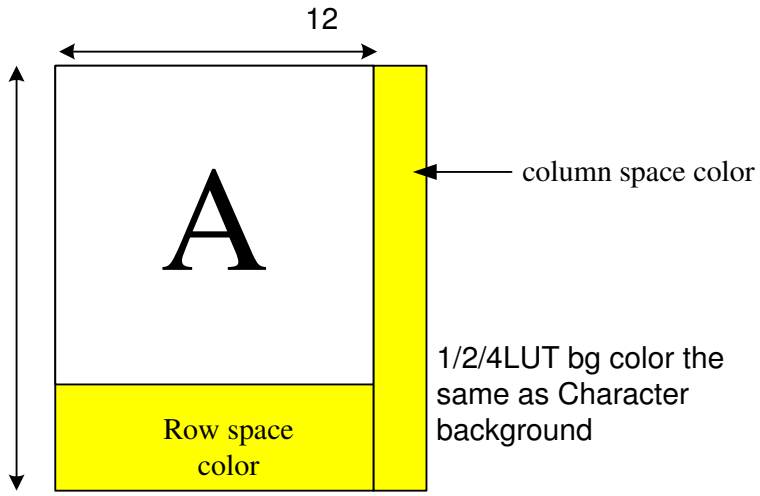
Notice:

When character height/width is doubled, the row height/column space definition also twice. If the row height is larger than character height, the effect is just like space between rows. If it is smaller than character height, it will drop last several bottom line of character.

When using 1/2/4LUT font, column space and font smaller than row height, the color of column space and row space is the same as font background color~~, only 4-bit true-color font mode, the color is transparent~~

Byte 2

Bit	Mode	Function
7:0	W	Row length unit: font base



Character Command (For blank)

Byte 0

Bit	Mode	Function
7	W	0
6	W	Blinking effect 0: Disable 1: Enable
5:4	W	01
3:0	W	Reserved

Byte 1

Bit	Mode	Function
7:0	W	Blank pixel length

At least 3 pixels, and can't exceed 255 pixels.

Byte 2

Bit	Mode	Function
7:6	W	Reserved
5:4	W	Blank color – select one of 64-color LUT [5:4]
3:0	W	Blank color – select one of 64-color LUT [3:0] (0 is special for transparent)

Character Command (For 1-bit RAM font)

Byte 0

Bit	Mode	Function
7	W	Use combined with [5:4]
6	W	Character Blinking effect 0: Disable 1: Enable
5:4	W	{[7], [5:4]} (Font type 000: 1-bit RAM Font When Row command Byte 0[5] is 0 Font select from 0-255 When Row command Byte 0[5] is 1

		Font select from 512-767 Color palette from 16-31 001: Blank 010: 1-bit RAM Font When Row command Byte 0[5] is 0 Font select from 256-511 When Row command Byte 0[5] is 1 Font select from 768-1023 Color palette from 0-15 011: 1-bit RAM Font When Row command Byte 0[5] is 0 Font select from 256-511 When Row command Byte 0[5] is 1 Font select from 768-1023 Color palette from 16-31 100: 1-bit RAM Font When Row command Byte 0[5] is 0 Font select from 0-255 When Row command Byte 0[5] is 1 Font select from 512-767 Color palette from 0-15 101: 4-bit RAM Font Font select from 0-255 Color palette from 0-63 110: 2-bit RAM Font Font select from 0-254 (with Frame Ctrl register 00B byte1 [6:0]) Color palette from 0-15 111: 2-bit RAM Font Font select from 0-254 (with Frame Ctrl register 00B byte1 [6:0]) Color palette from 16-31)
	W	VBI OSD disable: Character width (only for 1-pixel font, doubled when specifying double-width in Row/Blank command register) For 12x18 font: 0100: 4-pixel 0101: 5-pixel 0110: 6-pixel 0111: 7-pixel 1000: 8-pixel 1001: 9-pixel 1010: 10-pixel 1011:11-pixel 1100: 12-pixel For 18x12 Font (rotated) 0000: 4-pixel 0001: 5-pixel 0010: 6-pixel 0011: 7-pixel 0100: 8-pixel 0101: 9-pixel 0110: 10-pixel 0111: 11-pixel 1000: 12-pixel 1001:13-pixel 1010:14-pixel 1011:15-pixel 1100: 16-pixel 1101:17-pixel 1110:18-pixel 1111:36-pixel VBI OSD enable: While VBI OSD enable, 1 bit font will be NO rotated and 12-pixel fonts always. Then the [3:0] setting will be as following: [3]: character select[8] support 512 font while VBI OSD enable [2]: additional blinking effect {[6], [2]} 00: NO blink for both F/B 01: Only blink for Foreground

		10: Only blink for Background 11: Both blink for F/B [1]: Underline enable underline will be at 17th & 18th line and got the same color with foreground [0]: Reserved

When using border/shadow/ effect, the width of the 1-bit font should at least 6 pixel.

Byte 1

Bit	Mode	Function
7:0	W	Character Select [7:0]

Byte 2

Bit	Mode	Function
7:4	W	Foreground color Select one of 16-color from color LUT
3:0	W	Background color Select one of 16-color from color LUT (0 is special for transparent)

Character command (For 2-bit RAM Font)

Byte 0

Bit	Mode	Function
7	W	1
6	W	MSB of Foreground color 11, Background 00
5	W	1
4	W	0: color palette 0-15 1: color palette 16-31
3:1	W	Foreground color 11 Select one of 8 color from color LUT Add Byte0 [6] as MSB for 16-color LUT.
0	W	Background color 00 Bit[2] Select one of 8 color from color LUT

Byte 1

Bit	Mode	Function
7	W	MSB of Foreground color 10, Foreground 01
6:0	W	Character Select [6:0]

When swap function is enable(frame control 0x011h byte0-bit[3]), the definition will be swapped between byte0-bit[4] & byte1-bit[7] in 2-bit font character command

Byte 2

Bit	Mode	Function
7:6	W	Background color 00 Bit[1:0] Select one of 8 color from color LUT While 0 is special for transparent Add Byte0 [6] as MSB for 16-color LUT. Once we fill 0000 or 1000(MSB follow Byte0[6]), BG appears transparent.
5:3	W	Foreground color 10 Select one of 8 color from color LUT Add Byte0 [4] as MSB for 16-color LUT.
2:0	W	Foreground color 01 Select one of 8 color from color LUT Add Byte0 [4] as MSB for 16-color LUT.

Character command (For 4-bit RAM font)

Byte 0

Bit	Mode	Function
7	W	1
6	W	Character Blinking effect 0: Disable 1: Enable
5:4	W	01 (Font type 00: 1-bit RAM Font 01: 4-bit RAM Font 1x: 2-bit RAM Font)
3:0	W	~~(for Byte1[7] = 0)~~ select one color from 16-color LUT as background

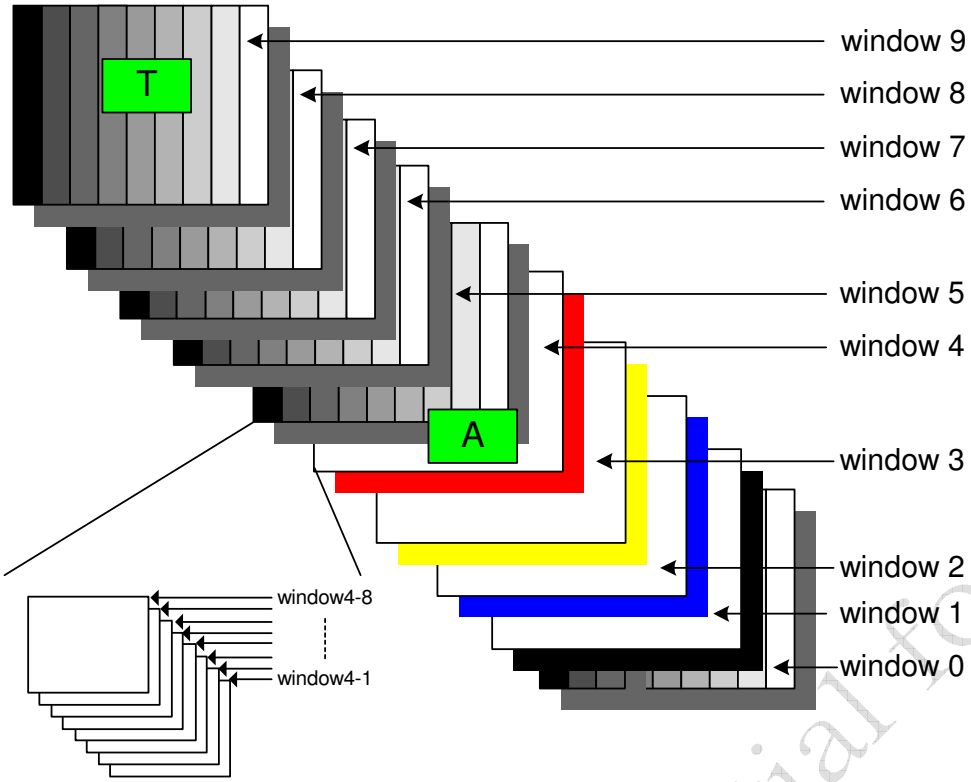
Byte 1

Bit	Mode	Function
7:0	W	Character Select [7:0]

- When 4-bit look-up table mode，color of column space is the same as background.
- When 4-bit look-up table mode and pixel value is 0000, and byte0[3:0]=0000 means transparent.
- ~~When true color mode and pixel value is 0000，it is transparent。~~

Byte 2

Bit	Mode	Function
7:6	W	Color select 12 13 14 15 00: color select 12 13 14 15 01: color select 28 29 30 31 10: color select 44 45 46 47 11: color select 60 61 62 63
5:4	W	Color select 8 9 10 11 00: color select 8 9 10 11 01: color select 24 25 26 27 10: color select 40 41 42 43 11: color select 56 57 58 59
3:2	W	Color select 4 5 6 7 00: color select 4 5 6 7 01: color select 20 21 22 23 10: color select 36 37 38 39 11: color select 52 53 54 55
1:0	W	Color select 0 1 2 3 00: color select 0 1 2 3 01: color select 16 17 18 19 10: color select 32 33 34 35 11: color select 48 49 50 51



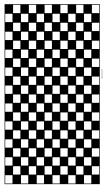
Display Priority

Display Priority

We have six windows with gradient and twelve windows without gradient, the window priority is as above, character should be always on the top layer of the window.

Pattern gen.
Use OSD to replace display pattern generator.

Chess Board: make a font as below



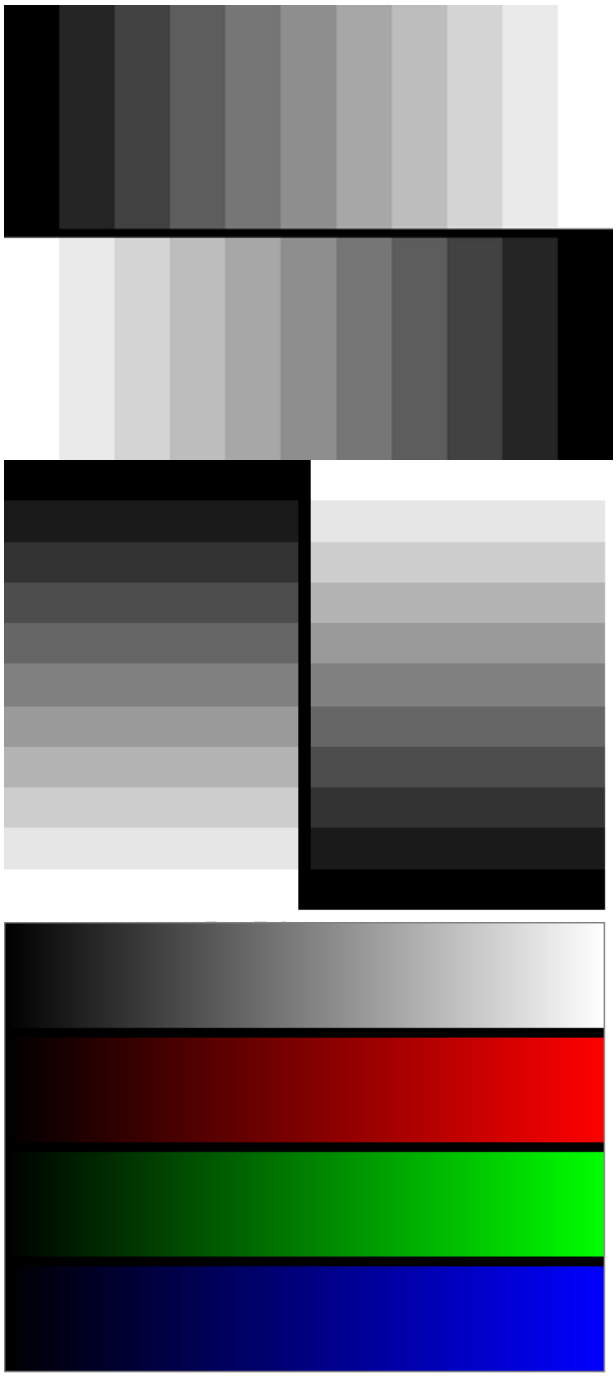
If we want to fill to the full 1280x1024 screen with character, we need 1280*1024 pixels. Required character is:
Using 12*18 font
1280/12 = 106.7 -> 107
1024/18 = 56.9 -> 57
107*57 = 6099 character

The required number of character map is larger than RAM size. We must turn on double width or double height function to reduce the half of character map.

So the basic unit to chessboard is 2x2 pixel. You can use larger chessboard instead of 2x2 pixels unit, such as 4x4 and so on.

Gray level

We can display 256 gray level by gradient window, 8 and 16 gray level by character map. 32 and 64 gray level is not supported.



6. Electric Specification

DC Characteristics

Table 3 Absolute Maximum Ratings

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS
Voltage on Input (5V tolerant)	V_{IN}	-1		5	V
Supply Voltage	PVCC	3.14	3.3	3.47	V
	VCCK	1.14	1.2	1.26	V
Electrostatic Discharge	V_{ESD}			±2.5	kV
Latch-Up	I_{LA}			±100	mA
Ambient Operating Temperature	T_A	0		70	ºC
Storage temperature (plastic)	T_{STG}	-55		125	ºC
Thermal Resistance (Junction to Air)	θ_{JA}			55.238	ºC/W
Junction Acceptable Temperature	T_j			125	ºC

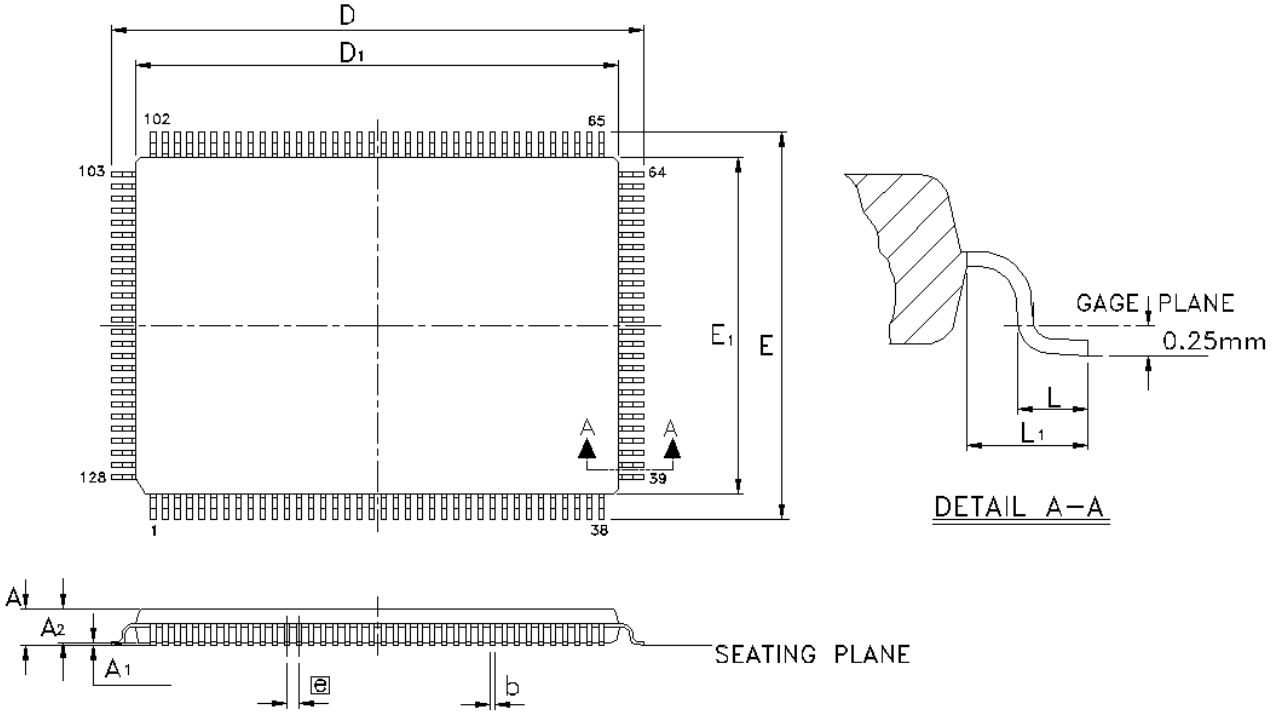
PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS
Reset pulse period	Trst-en[1]	1120			ns
Power on reset period	Tpor-rst[2]	293			ms

1. 16 * x'tal_cycle(1/14.3Mhz)
2. 65536*64*Xtal_cycle(1/14.3Mhz)

7. Mechanical Specification

128 Pin Package (LQFP)

Low Profile Plastic Quad Flat Package 128 Leads 14x20mm^2 Outline



Symbol	Dimension in mm			Dimension in inch		
	Min	Nom	Max	Min	Nom	Max
A	—	—	1.60	—	—	0.063
A_1	0.05	—	0.15	0.002	—	0.006
A_2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.17	0.22	0.27	0.007	0.009	0.011
D	22.00 BSC			0.866 BSC		
D_1	20.00 BSC			0.787 BSC		
E	16.00 BSC			0.630 BSC		
E_1	14.00 BSC			0.551 BSC		
e	0.50 BSC			0.020 BSC		
L	0.45	0.60	0.75	0.018	0.024	0.030
L1	1.00 REF			0.039 REF		

Notes：

1. CONTROLLING DIMENSION：MILLIMETER(mm).
2. REFERENCE DOCUMENTL：JEDEC MS-026.