# RTD 2525/2545/2555 LRH

# Flat Panel Display Controller

**Preliminary**
**Revision**
**Version 1.00**
**Last updated: 2008/7/16**

# 1. Features

## General
- Programmable Scaling up and down
- No external memory required.
- Require only one crystal to generate all timing.
- Programmable 3.3V detection for multi-power domain in a system
- 2 channels 8 bits PWM output, and wide range selectable PWM frequency.
- Support input format up to 1920-pixel width
- DDC/CI(MCCS) supported

## Analog RGB Input Interface
- Integrated 8-bit triple-channel 210 MHz ADC/PLL (optional)
- 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
- Auto Black Level Adjustment

## Digital Video Input Interface
- Support 8-bit video (ITU 656) format input
- Built-in YUV to RGB color space converter & de-interlace

## DVI Compliant Digital Input Interface (Optional)
- Single link on-chip TMDS receiver
- Up to 165Mhz
- Adaptive algorithm for TMDS capability
- Data enable only mode support
- HDCP 1.1 supported (optional)

## 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

## 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
- 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/XVYCC function for video sharpness

## VividColor<sup>TM</sup>
- Independent color management (ICM)
- Dynamic contrast control (DCC)

## LiveShow<sup>TM</sup> Function
- High-performance RTC (response time compensation) with embedded SDRAM.

## Output Interface
- Fully programmable display timing generator
- 1 and 2 pixel/clock panel support and up to 140MHz(RTD2525LRH)/170MHz(RTD2545LRH)/190MHz(RTD2555LRH)
- Support LVDS output interface
- Spread-Spectrum DPLL to reduce EMI
- Fixed Last Line output for perfect panel capability
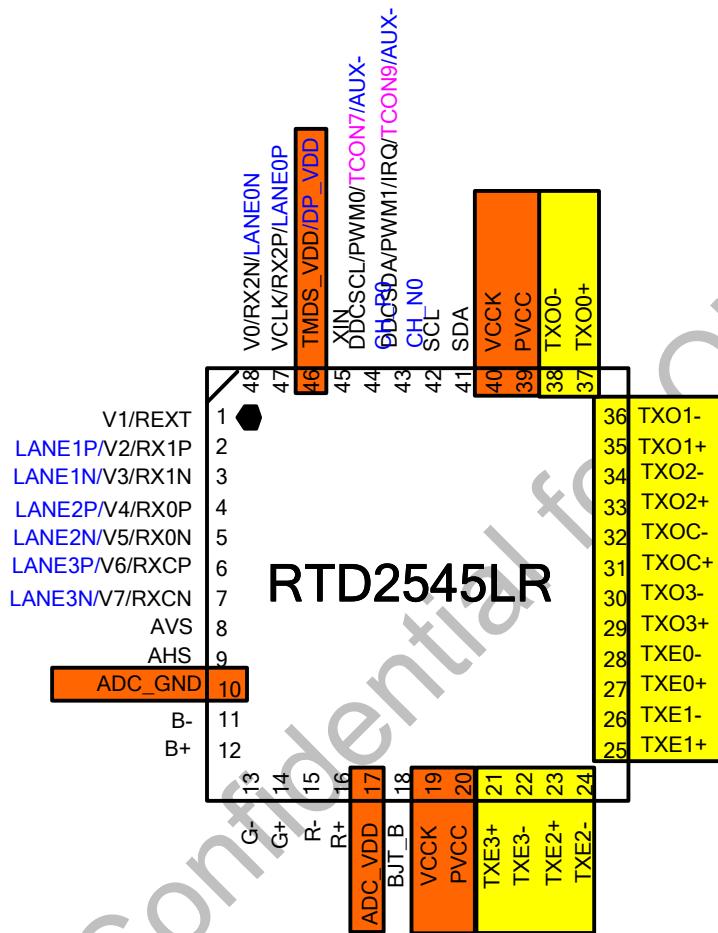
## Host Interface
- Support MCU serial bus interface.
- Support MCU dual edge data latch.

## Embedded OSD
- Embedded 16.5K SRAM dynamically stores OSD command and fonts
- Support multi-color RAM font, 1, 2 and 4-bit per pixel
- 16 color palette with 4096 true color selection
- Maximum 8 window with alpha-blending/ gradient/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 hardware proportional font
- Decompress OSD font

## Power & Technology
- 48 pin QFN package
- Embedded voltage regulator
- 0.11um process 3.3V / 1.2V power supplier

**Pin Out Diagram of RTD 2525/2545/2555 LRH**

■ **48 pin-out List**
■ **INPUT PORT**

| Name | I/O | No | Description | Note |
|------|-----|-----|-------------|------|
| B0+ | AI | 12 | Positive BLUE analog input | |
| B0- | AI | 11 | Negative BLUE analog input | |
| G0+ | AI | 14 | Positive GREEN analog input | |
| G0- | AI | 13 | Negative GREEN analog input | |
| R0+ | AI | 16 | Positive RED analog input | |
| R0- | AI | 15 | Negative RED analog input | |
| AVS | I | 8 | ADC vertical sync input<br>General Schmitt trigger | no power 5V tolerance |
| AHS | I | 9 | ADC horizontal sync input<br>Adjustable Schmitt trigger | no power 5V tolerance |

■ **PLL**

| Name | I/O | Pin No | Description | Note |
|------|-----|--------|-------------|------|
| XI | AI | 45 | Reference clock input from external crystal or from single-ended CMOS/TTL OSC | 3.3V tolerance |

■ **Host interface**

| Name | I/O | Pin No | Description | Note |
|------|-----|--------|-------------|------|
| SDA | I/O | 41 | Serial control I/F data (Open drain w/ ST) | 5V tolerance |
| SCL | O | 42 | Serial control I/F clock (Open drain w/ ST) | 5V tolerance |

■ **TMDS**

| Name | I/O | Pin No | Description | Note |
|------|-----|--------|-------------|------|
| REXT | AI | 1 | Impedance Match Reference. | |
| RX2P | AI | 47 | Differential Data Input | |
| RX2N | AI | 48 | Differential Data Input | |
| RX1P | AI | 2 | Differential Data Input | |
| RX1N | AI | 3 | Differential Data Input | |
| RX0P | AI | 4 | Differential Data Input | |
| RX0N | AI | 5 | Differential Data Input | |
| RXCP | AI | 6 | Differential Data Input | |
| RXCN | AI | 7 | Differential Data Input | |

■ **Video 8 (Optional)**

| Name | I/O | Pin No | Description |
|------|-----|--------|-------------|
| V8_0 ~ V8_7 | I | 48~7 | Video 8 data input (3.3V tolerance) |
| VCLK | I | 47 | Video8 clock input (3.3V tolerance) |

■ **Power and Ground**

| Name | I/O | Pin No | Description |
|------|-----|--------|-------------|
| ADC_GND | G | 10 | ADC 1.2V Ground |
| ADC_VDD | P | 17 | ADC 1.2V Power |
| TMDS_VDD | P | 46 | TMDS 3.3V Power |
| VCCK | P | 19/40 | Digital core Power |
| PVCC | P | 20/39 | Pad 3.3V Power |

■ **LVDS Display Interface**

| Name | I/O | No | Description |
|------|-----|-----|-------------|
| TXE3+ | O | 21 | LVDS Differential Data Output |
| TXE3- | O | 22 | LVDS Differential Data Output |
| TXE2+ | O | 23 | LVDS Differential Data Output |
| TXE2- | O | 24 | LVDS Differential Data Output |

| | | | |
|---|---|---|---|
| TXE1+ | O | 25 | LVDS Differential Data Output |
| TXE1- | O | 26 | LVDS Differential Data Output |
| TXE0+ | O | 27 | LVDS Differential Data Output |
| TXE0- | O | 28 | LVDS Differential Data Output |
| TXO3+ | O | 29 | LVDS Differential Data Output |
| TXO3- | O | 30 | LVDS Differential Data Output |
| TXOC+ | O | 31 | LVDS Differential CLK Output |
| TXOC- | O | 32 | LVDS Differential CLK Output |
| TXO2+ | O | 33 | LVDS Differential Data Output |
| TXO2- | O | 34 | LVDS Differential Data Output |
| TXO1+ | O | 35 | LVDS Differential Data Output |
| TXO1- | O | 36 | LVDS Differential Data Output |
| TXO0+ | O | 37 | LVDS Differential Data Output |
| TXO0- | O | 38 | LVDS Differential Data Output |

■ **DDC/CI Channel**

| Name | I/O | No | Description |
|---|---|---|---|
| DDCSDA | I/O | 43 | Open drain, no power 5V tolerance with **Schmitt trigger pad** |
| DDCSCL | I | 44 | Open drain, no power 5V tolerance with **Schmitt trigger pad** |

■ **PWM**

| Name | I/O | No | Description |
|---|---|---|---|
| PWM0 | O | 44 | Open drain, with 5V tolerance |
| PWM1 | O | 43 | Open drain, with 5V tolerance |

■ **MISC**

| Name | I/O | No | Description |
|---|---|---|---|
| BJT_B | O | 18 | Embedded regulator P type BJT control pin out |

■ **Timing Controller**

| Name | I/O | No | Description |
|---|---|---|---|
| TCON7 | O | 44 | Timing controller output |
| TCON9 | O | 43 | Timing controller output |

## 2. Chip Data Path Block Diagram

```
ADC ┐
     ├─► VGIP ─► Digital ─► CSC ─► SD ─► FIFO/
HDMI ┘              Filter                    FRC
                                                │
Contrast/  ◄─ sRGB ◄─ DCC/ICM ◄─ Y-Peaking ◄─ SU ◄┘
Brightness
    │
    └─► Gamma ─┐
               ├─► Overlay ─► Dithering ─► LiveShow ─► LVDS
     OSD ──────┘
```
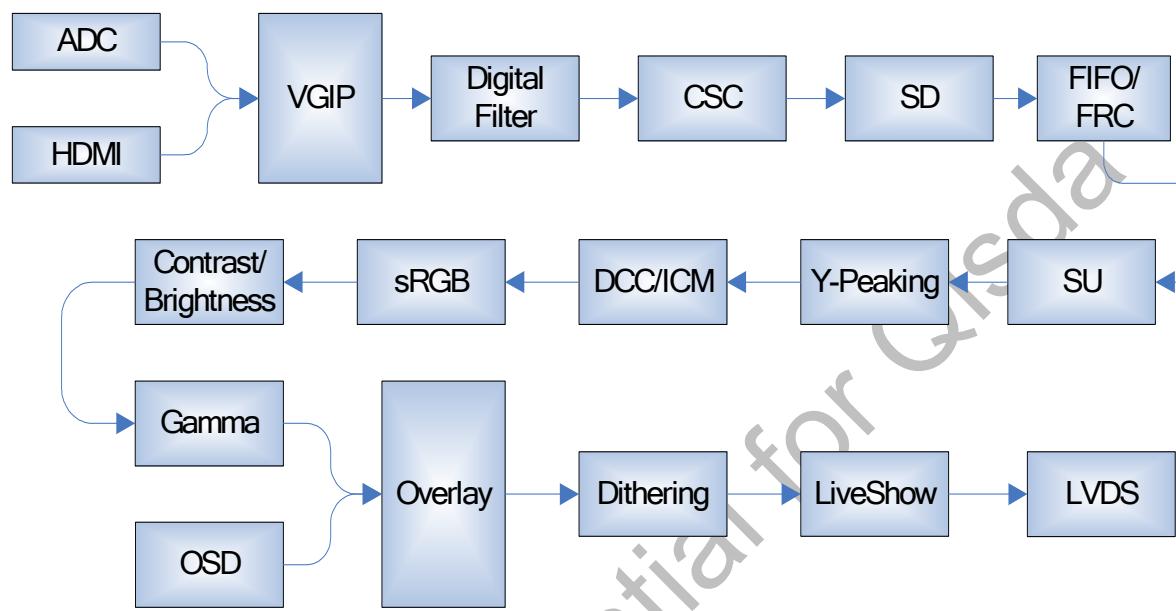
Figure 1

## 3.    Register Description

### Global Event Flag

| Register::ID_Reg | | | | | 0x00 |
|---|---|---|---|---|---|
| **Name** | **Bit** | **R/W** | **Default** | **Description** | **Config** |
| ID | 7:0 | R | 0x51 | **MSB 4 bits:** 0000 product code <br> **LSB 4 bits:** 0001 rev. code | |

| Register:: Host_ctrl | | | | | 0x01 |
|---|---|---|---|---|---|
| **Name** | **Bit** | **R/W** | **Default** | **Description** | **Config.** |
| Rev | 7 | --- | 0 | Reserved | |
| Reset_chk | 6 | R/W | 0 | **Reset Check** <br> Once scalar is reset, this value will be cleared to 0.The purpose of it is to check if LVR has been triggered. <br> 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** <br> 0: Normal <br> 1: Enable power down mode(Default) <br> Turn off ADC RGB Channel/ ADC Band-gap/ SOG/ DPLL/ LVDS/ADC PLL/ SYNC- PROC/ TMDS / HDMI-Audio PLL/m2pll <br> Note: For LVDS Power Control, refer to following table. | |
| PS_EN | 1 | R/W | 0 | **Power Saving Mode Enable** <br> 0: Normal(Default) <br> 1: Enable power saving mode <br> Turn off ADC RGB channel/ DPLL/ LVDS/ ADC PLL/ m2pll <br> When power down or power saving function is enabled, internal mcu clock is forced to crystal clock. <br> 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/RSD Display Output is double.

| DISP_TYPE CR 8E-00[1:0] | DATA_TYPE CR 28[2] | Port | Power Control |
|---|---|---|---|
| **LVDS** [01] | **Double** [1] | **LVDS Mid** | **Power Down/Power Saving** |
| **LVDS** [01] | **Double** [1] | **LVDS Even** | **CR01 [2]/CR 01[1]** |
| **LVDS** [01] | **Double** [1] | **LVDS Odd** | |
| **LVDS** [01] | **Single** [0] | **LVDS Mid** | **Power Up LVDS Mid-Port CR8C-A0 [6]** |
| **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 | |

| Name | Bit | R/W | Default | Description | Config. |
|------|-----|-----|---------|-------------|---------|
| | | | | 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 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.

---

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

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

| | | | | | |
|---|---|---|---|---|---|
| IRQ_IENA | 0 | R/W | 0 | **IRQ (Input ENA Start Event Occurred Status)**<br>0: Disable IENA start as interrupt source<br>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<br>0:disable<br>1:enable | |
| New_m_state_IRQ_en | 3 | R/W | 0 | IRQ enable of New mode status<br>0:disable<br>1:enable | |
| Change_m_happen_IRQ_en | 2 | R/W | 0 | IRQ enable of change mode happen status<br>0:disable<br><br>1:enable | |
| DP_IRQ | 1 | R | --- | Display port IRQ status | |
| Reserved | 0 | --- | --- | Reserved | |

| **Register:: New_added_status1** | | | | | **0x08** |
|---|---|---|---|---|---|
| **Name** | **Bit** | **R/W** | **Default** | **Description** | **Config.** |
| Reserved | 7:0 | --- | --- | Reserved | |

*Address: 09~0B Reserved*

**Watch Dog**

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

| Bit | Mode | Function |
|---|---|---|
| 7 | R/W | **Auto Switch When Input HSYNC/VSYNC Error**<br>0: Disable (Default)<br>1: Enable<br>(See CR02[6] and CR02[5]) |
| 6 | R/W | **Auto Switch When Input HSYNC/VSYNC Timeout or Overflow**<br>0: Disable (Default)<br>1: Enable<br>(See CR52[4] and CR54[5:4]) |
| 5 | R/W | **Auto Switch When Display VSYNC Timeout**<br>0: Disable (Default)<br>1: Enable |
| 4 | R/W | **Auto Switch When ADC-PLL Unlock**<br>0: Disable (Default)<br>1: Enable |
| 3 | R/W | **Auto Switch When Overflow or Underflow (for Frame-Sync Display)**<br>0: Disable (Default)<br>1: Enable |
| 2 | R/W | **Watch-Dog Action if Event Happened (for Display Timing)**<br>0: Disable (Default)<br>1: Free Run |
| 1 | R/W | **Watch-Dog Action if Event Happened (for Display Data)**<br>0: Disable (Default)<br>1: Background<br>(Turn off overlay function and switch to background display simultaneously) |
| 0 | R | **Display VSYNC Timeout Flag (for CR0C[5])**<br>0: DVS is present<br>1: DVS is timeout<br>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**<br>0: Disable (Default)<br>1: Enable<br>(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: Video8 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) |
|   |     | <table><tr><td>Register</td><td>Trigger Event</td></tr><tr><td>PLLPHASE(CRB3,CRB4)</td><td>Falling edge of Ivactive</td></tr></table> |

| | | Add 1-clk Delay to IHS Delay (CR12[4]) HSYNC Synchronize Edge (CR12[3]) | |
|---|---|---|---|
| | | 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 |

| Bit | Mode | Function |
|---|---|---|
| 3:2 | R/W | **Input Pixel Format**<br>00: Embedded ADC (ADC_HS)(Default)<br>01: Embedded TMDS<br>10: Video8<br>11: Reserved |
| 1 | R/W | **Input Graphic/Video Mode**<br>0: From analog input (input captured by 'Input Capture Window') (Default)<br>1: From digital input (captured start by 'enable signal', but sill stored in 'capture window size') |
| 0 | R/W | **Input Sampling Run Enable**<br>0: No data is transferred (Default)<br>1: Sampling input pixels |

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

| Bit | Mode | Function |
|---|---|---|
| 7 | R/W | **Safe Mode**<br>0: Normal (Default)<br>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)<br>0: Enable (Default)<br>1: Disable |
| 5 | R/W | **HS Signal Inverted for Field Detection**<br>0: Negative Edge (Default)<br>1: Positive Edge |
| 4 | R/W | **Input Video ODD Signal Invert Enable**<br>0: Not inverted (ODD = positive polarity) (Default)<br>1: Inverted (ODD = negative polarity) |
| 3 | R/W | **Input VS Signal Polarity Inverted**<br>0: Not inverted (VS = positive polarity) (Default)<br>1: Inverted (VS = negative polarity) |
| 2 | R/W | **Input HS Signal Polarity Inverted**<br>0: Not inverted (HS = positive polarity) (Default)<br>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 |
| 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 | **Video 4:2:2->4:4:4 Enable before Scale-Down (Duplicate)** |
| | | 0: Disable (Default) |
| | | 1: Enable |
| 2 | R/W | **Decode Video8 when ADC or TMDS Active** |
| | | 0: Disable (Default) |
| | | 1: Enable |
| 1 | R/W | **EAV Error Correction Enable in Video-8** |
| | | 0: Disable |
| | | 1: Enable |
| 0 | R/W | **Internal ODD Signal Selection** |
| | | 0: ODD signal from EAV or YPbPr (Default) |
| | | 1: Internal Field Detection ODD signal (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] |
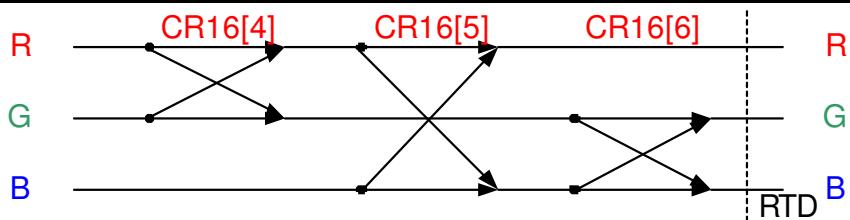
*Address: 15*  **IPH_ACT_STA_L (Input Horizontal Active Start Low)**  Default: 00h

| 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 | **Video8 -C-Port Input Latch Bus MSB to LSB Swap Control:** <br> 0: Normal (Default) <br> 1: Swap Video8 -C-port MSB to LSB sequence into LSB to MSB |
| 6 | R/W | **ADC Input G/B Swap** <br> 0: No Swap <br> 1: Swap |
| 5 | R/W | **ADC Input R/B Swap** <br> 0: No Swap <br> 1: Swap |
| 4 | R/W | **ADC Input R/G Swap** <br> 0: No Swap <br> 1: Swap |
| 3 | R/W | **Double Clock Input** <br> 0: Single Clock <br> 1: Double Clock <br> this bit should be set double clock when using video 8 input |
| 2:0 | --- | **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) <br> 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) <br> 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) <br> 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<br>01: 32 pixels<br>10: 64 pixels<br>11: 96 pixels |
| 5:4 | R/W | **Input VSYNC Delay for Auto Function** (Counted by Input HSYNC)<br>　00: No delay<br>　01: 3 line<br>　10: 7 line<br>　11: 15 line |
| 3 | R/W | **Select DataEnable or HSync to adjust clock phase**<br>　0: use DataEable to adjust clock phase (Default)<br>　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) |

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

| Bit | Mode | Function |
|---|---|---|
| 7 | R/W | **Reorder the data flow**<br>　0: dfilter -> color_conversion -> dithering -> HSD<br>**1: dfilter -> dithering -> color_conversion -> HSD** |
| 6:4 | --- | Reserved |
| 3 | R/W | **Video 4:2:2->4:4:4 Enable before Scale-Down**<br>　0: Disable (Default)<br>　1: Enable<br>　(This bit should be always enable when in Video8/ HDMI YUV422 mode.) |
| 2 | R/W | **Video 4:2:2->4:4:4 Mode Select**<br>　0: Interpolation (Default)<br>　1: Duplicate<br>(This bit would be work only while CR1F[3] is enable) |
| 1 | R/W | **Output 444 Format** (only work in **Interpolation Mode**)<br>　0: $Y_0U_0V_0$, $Y_1(U_0+U_2)/2$ $(V_0+V_2)/2$, $Y_2U_2V_2$, $Y_3(U_2+U_4)/2$ $(V_2+V_4)/2$…<br>　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**)<br>　0: Sequence 444 result: Y, U, V<br>　1: Sequence 444 result: Y, V, U |
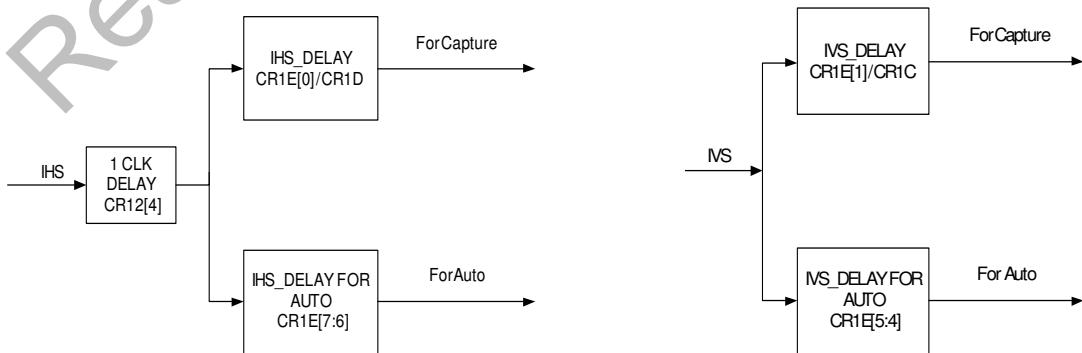


Figure 15:　Input HSYNC/VSYNC Delay Path Diagram

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

| Bit | Mode | Function |
|-----|------|----------|
| 7:6 | --- | Reserved |
| 5:4 | R/W | **V8clk divider:**<br>00: div 2 (Default)<br>01: div 4<br>10: div 8<br>11: reserved |
| 3 | --- | Reserved |
| 2:0 | R/W | **V8clk_phase:**<br>000: phase 0 (Default)<br>001: phase 1<br>010: phase 2 (not work while div2)<br>011: phase 3 (not work while div2)<br>100: phase 4 (not work while div2 & div4)<br>101: phase 5 (not work while div2 & div4)<br>110: phase 6 (not work while div2 & div4)<br>111: phase 7 (not work while div2 & div4) |

## FIFO Frequency

*Address: 22*　　　**FIFO Frequency**　　　　　　　　　　　Default: 00h

| Bit | Mode | Function |
|-----|------|----------|
| 7 | R/W | **Test Mode**<br>0: Disable<br>1: Input data of VGIP Replaced by Background Color in CR6D |
| 6:3 | R/W | **Reserved to 0** |
| 2 | R/W | **Internal Xtal Frequency**<br>0: $F_{xtal}$<br>1: $F_{xtal}$ * M2PLL_M / M2PLL_N / 10 |
| 1:0 | R/W | **FIFO Frequency**<br>00: MPLL<br>01: ICLK<br>10: DCLK<br>11: M2PLL |

## 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:**<br>0: Use line interpolation mode (Default)<br>1: Use drop line mode<br>(Note: This bit is only valid while CR23[0]=1'b1.) |
| 6 | R | **Bist for Line Buffer one & two ok**<br>0: Fail |

| | | 1: Ok |
|---|---|---|
| 5 | --- | **Reserved** |
| 4 | R/W | **Line Buffer Bist Function Start (Auto clear to 0 when finish)**<br>0: Finish<br>1: Start |
| 3 | R/W | **Horizontal non-linear scale down**<br>0: linear<br>1: non-linear |
| 2 | R/W | **Vertical Scale-Down Compensation**<br>0: Disable (Default)<br>1: Enable |
| 1 | R/W | **Horizontal scale down function enable:**<br>0: Disable scale down function (Default)<br>1: Enable scale down function |
| 0 | R/W | **Vertical scale down function enable:**<br>0: Disable scale down function (Default)<br>1: Enable scale down function<br>(Note: There is a bit to select **interpolation** or **dropping** for vertical scale down at CR24[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 |
|---|---|---|

| 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<br>0 : 1 frame<br>1 : 16 frame |
| 6 | R/W | Random generator mode<br>**0 : x^9 + x^3 + 1**<br>**1 : x^29+x^6+x^4+x+1 (Green, Blue, Red )** |
| 5 | R/W | Data update (RED)<br>0 : reference data enable(pixel base)<br>1: reference horizontal data enable end(line base) |
| 4 | R/W | Data update (GREEN)<br>0 : reference data enable<br>1: reference horizontal data enable end |
| 3 | R/W | Data update (BLUE)<br>0 : reference data enable<br>1: reference horizontal data enable end |
| 2 | R/W | Pattern generator mode (RED)<br>0 : random generator (ref. CR25-10[6]<br>1 : pattern generator (reg. CR25-11[2]) |
| 1 | R/W | Pattern generator mode (GREEN)<br>0 : random generator (ref. CR25-10[6]<br>1 : pattern generator (reg. CR25-11[1]) |
| 0 | R/W | Pattern generator mode (BLUE)<br>0 : random generator (ref. CR25-10[6]<br>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)<br>0 : Out(n) = Out(n-1)<br>1: Out(n) = Out(n-1) + 1 |
| 1 | R/W | Pattern generator (GREEN)<br>0 : Out(n) = Out(n-1)<br>1: Out(n) = Out(n-1) + 1 |
| 0 | R/W | Pattern generator (BLUE)<br>0 : Out(n) = Out(n-1)<br>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
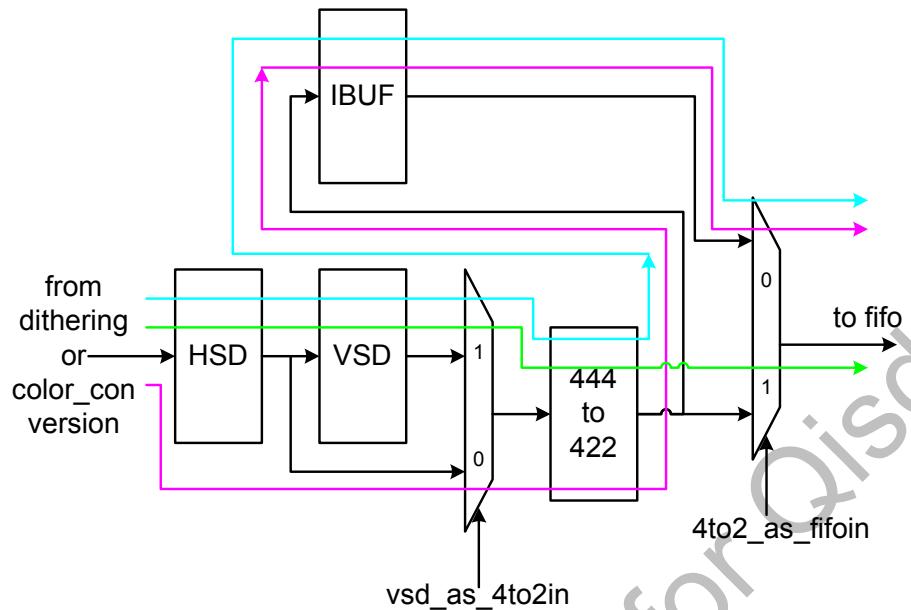
| Bit | Mode | Function |
|---|---|---|
| 7-0 | R/W | BLUE Initial Value [7:0] |

*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] |

| Register::I_YUV444to422 | | | | | 0x26 | |
|-------------------------|------|------------|-------------|----------|------|--------|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Comments** | | **Config** |
| VSD_AS_4TO2IN | 7 | R/W | 0 | This bit decides the data flow in I-domain:<br>0: HSD output as YUV444to422 input data<br>1: VSD output as YUV444to422 input data | | |
| 4TO2_AS_FIFOIN | 6 | R/W | 0 | This bit decides the data flow in I-domain:<br>0: I_BUF output as FIFO input data<br>1: YUV444to422 output as FIFO input data | | |
| LS_RSV_26_54 | 5:4 | R/W | 0 | Reserved | | |
| 444TO422_EN | 3 | R/W | 0 | In I-domain, YUV 444 to 422:<br>0: Disable<br>1: Enable | | |
| LS_RSV_26_2 | 2 | R/W | 0 | Reserved | | |
| INTERPOLATE | 1 | R/W | 0 | In I-domain, YUV 444 to 422:<br>0: Drop C directly<br>a. uv_mode = 0: $Y_0U_0, Y_1V_0, Y_2U_2, Y_3V_2\ldots$<br>b. uv_mode = 1: $Y_0U_0, Y_1V_1, Y_2U_2, Y_3V_3\ldots$<br>1: Interpolation Mode<br>a. uv_mode = 0: $Y_0(U_0+U_1)/2$ , $Y_1(V_0+V_1)/2$, $Y_2(U_2+U_3)/2$ , $Y_3(V_2+V_3)/2\ldots$<br>b. uv_mode = 1: $Y_0(U_0+U_1)/2$ , $Y_1(V_1+V_2)/2$, $Y_2(U_2+U_3)/2$ , $Y_3(V_3+V_4)/2\ldots$ | | |
| UV_MODE | 0 | R/W | 0 | In I-domain, 444to422 U/V type<br>0: U0 V0 U2 V2 U4 V4 …<br>1: U0 V1 U2 V3 U4 V5 … | | |

→ (cyan) dithering/color_conv => HSD => VSD(drop line) => 444to422 => I_BUF => new_fifo

→ (green) dithering/color_conv => HSD => VSD(interpolation) => 444to422 => new_fifo

→ (magenta) dithering/color_conv => HSD => 444to422 => I_BUF => new_fifo　　(defult setting)

*Address: 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)** <br> 0: wait for input IVS trigger <br> 1: force enable |
| 6 | R/W | **Display Data Output Inverse Enable** <br> 0: Disable (Default) <br> 1: Enable　(only when data bus clamp to 0) |
| 5 | R/W | **Display Output Force to Background Color** <br> 0: Display output operates normally <br> 1: Display output is forced to the color as selected by background color (CR6D) (Default) |
| 4 | R/W | **Display 18 bit RGB Mode Enable** <br> 0: All individual output pixels are full 24-bit RGB (Default) <br> 1: All individual output pixels are truncated to 18-bit RGB (LSB 2 bits = 0) |
| 3 | R/W | **Frame Sync Mode Enable** |

| Bit | Mode | Function |
|---|---|---|
| | | 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** <br> **0 : Disable** <br> **1 : Enable** <br><br> <table><tr><td>Register</td><td>Trigger Event</td></tr><tr><td>DH_TOTAL</td><td>DVS Rising</td></tr><tr><td>**ODD_FIXED_LAST** <br> **EVEN_FIXED_LAST**</td><td>DVS Rising</td></tr></table> |
| 6 | R/W | **Display Double Buffer Ready** <br> 0: Not Ready to Apply <br> 1: Ready to Apply <br> 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 |
|---|---|---|
| 7:4 | -- | **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]:** <br> 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] |

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] |

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:**<br>000: DCLK rising edge correspondents with output display data<br>001: 1ns delay<br>010: 2ns delay<br>011: 3ns delay<br>100: 4ns delay<br>101: 5ns delay<br>110: 6ns delay<br>111: 7ns delay |
| 3 | --- | **Reserved** |
| 2 | --- | **Reserved** |
| 1 | R/W | **DCLK Output Enable**<br>0: Disable<br>1: Enable |
| 0 | R/W | **DCLK Polarity Inverted**<br>0: Disable<br>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]** |

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

| Bit | Mode | Function |
|---|---|---|
| 7 | R/W | **New Timing Enable**<br>0: Disable |

| | | 1: Enable |
|---|---|---|
| 6 | R/W | **Line Compensation Enable**<br>0: Disable<br>1: Enable |
| 5 | R/W | **Pixel Compensation Enable**<br>0: Disable<br>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**<br>**0: no inverse**<br>**1: inverse** |
| 6 | R/W | **SSCG New Timing Mode Even/Odd last line setting enable**<br>**0: disable**<br>**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**<br>00 : CRC after scale-down (before SDRAM)<br>01 : CRC after scale-down (before SDRAM)<br>10 : CRC after all processing<br>11 : Reserved |
| 5:1 | -- | **Reserved to 0** |
| 0 | R/W | **Output CRC Control:**<br>0: Stop or finish (Default)<br>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 | 1<sup>st</sup> read=> Output CRC-24 bit 23~16 |
|---|---|---|
| | | 2<sup>nd</sup> read=> Output CRC-24 bit 15~8 |
| | | 3<sup>rd</sup> 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] |

*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** |
| 6:5 | -- | **Reserved to 0** |
| 4:0 | R/W | **Scaling factor port address** |

- 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:

$(600/768) \times 2^{20} = 0.78125 \times 2^{20} = 819200 = C8000h = 0Ch, 80h, 00h.$

*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 Chroma Filter Coefficient Access**<br>0: Disable (Default)<br>1: Enable |
| 6 | R/W | **Select Chroma H/V User Defined Filter Coefficient Table for Access Channel**<br>0: 1st coefficient table (Default)<br>1: 2nd coefficient table |
| 5 | R/W | **Select Chroma Horizontal user defined filter coefficient table**<br>0: 1st Horizontal Coefficient Table (Default)<br>1: 2nd Horizontal Coefficient Table |
| 4 | R/W | **Select Chroma Vertical user defined filter coefficient table**<br>0: 1st Vertical Coefficient Table (Default)<br>1: 2nd Vertical Coefficient Table |
| 3 | R/W | **Enable Luminance Filter Coefficient Access**<br>0: Disable (Default) |

| | | |
|---|---|---|
| | | 1: Enable |
| 2 | R/W | **Select Luminance H/V User Defined Filter Coefficient Table for Access Channel**<br>　0: 1st coefficient table (Default)<br>1: 2nd coefficient table |
| 1 | R/W | **Select Luminance Horizontal user defined filter coefficient table**<br>　0: 1st Horizontal Coefficient Table (Default)<br>1: 2nd Horizontal Coefficient Table |
| 0 | R/W | **Select Luminance Vertical user defined filter coefficient table**<br>　0: 1st Vertical Coefficient Table (Default)<br>1: 2nd Vertical Coefficient Table |

- 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.

- When CR35[7] and CR35[3] are zero, the write counter of FILTER_PORT is reset to zero. You should reset counter before another setting.

- If both CR35[7] and CR35[3] are one, you can set chroma and luminance coefficient at the same time.

*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)<br>　The number of input HS from IVS to DVS.<br>　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** <br> **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** <br> **0 : disable** <br> **1: enable** |
| 6 | R/W | **Reference field in last line measure** <br> **0 : 0dd** <br> **1 : Even** |
| 5:2 | R/W | **Reserved to 0** |
| 1 | R/W | **Interlaced_FS_Delay_Fine_Tuning** <br> 0: Disable (Default) <br> 1: Enable |
| 0 | R/W | **Internal ODD-signal inverse for Interlaced_FS_Delay_Fine_Tuning** <br> 0: No invert (Default) <br> 1: Invert |

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

| Bit | Mode | Function |
|-----|------|----------|
| 7 | R/W | **Last-line-width / DV-Total Selector :** <br> 0: CR44 [3:0] and CR45 indicate last-line width counted by display clock (Default) <br> 1: CR44 [3:0] and CR45 indicate DHS total number between 2 DVS. |
| 6 | R/W | **DV sync with 4X clock** <br> 0: Disable <br> 1: Enable |
| 5 | R/W | **BIST Test Enable** <br> 0: Disable <br> 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)** <br> 0: Normal Run (Default) <br> 1: Power Down |
| 6 | R/W | **HSYNC Type Detection Auto Run** <br> 0: manual (Default) <br> 1: automatic |
| 5 | R/W | **De-composite circuit enable** <br> 0: Disable (Default) <br> 1: Enable |
| 4 | R/W | **Input Sync. Source selection** <br> 0: HS_RAW(SS/CS) (Default) <br> 1: SOG/SOY |
| 3 | R/W | **SOG Source Selection** <br> 0: SOG0/SOY0 (Default) <br> 1: reserved |
| 2 | R/W | **VGA-ADC HS/VS Source** <br> 0: 1$^{ST}$ HS/VS (Default) <br> 1: reserved |
| 1 | R/W | **Measured by Crystal Clock (Result shown in CR59) (in Digital Mode)** <br> 0: Input Active Region (Vertical IDEN start to IDEN stop) (measure at IDEN STOP) (Default) <br> 1: Display Active Region(Vertical DEN start to DEN stop) (measure at DEN STOP) <br> The function should work correctly when IVS or DVS occurs and enable by CR50[4]. |
| 0 | R/W | **HSYNC & VSYNC Measured Mode** <br> 0: HS period counted by crystal clock & VS period counted by HS (Analog mode) (Default) <br> 1: H resolution counted by input clock & V resolution counted by ENA (Digital mode) <br> (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:**<br>0: Not inverted (Default)<br>1: Inverted |
| 6 | R/W | **COAST Signal Output Enable:**<br>0: Disable (Default)<br>1: Enable |
| 5 | R/W | **HS_OUT Signal Invert Enable:**<br>0: Not inverted (Default)<br>1: Inverted |
| 4 | R/W | **HS_OUT Signal Output Enable:**<br>0: Disable (Default)<br>1: Enable |
| 3 | R/W | **CS_RAW Inverted Enable**<br>0: Normal (Default)<br>1: Invert |
| 2 | R/W | **CLAMP Signal Output Enable**<br>0: Disable (Default)<br>1: Enable |
| 1 | R/W | **HS Recovery in Coast**<br>0: Disable (Default) (SS/SOY)<br>1: Enable (CS or SOG ) |
| 0 | R/W | **HSYNC Synchronize source**<br>0: AHS (Default)<br>1: Feedback HS |

*Address: 49*　　　　**SYNC_CTRL (SYNC Control Register)**　　　　　　　**Default: 06h**

| Bit | Mode | Function |
|---|---|---|
| 7 | R/W | **CLK Inversion to latch Feedback HS for Coast Recovery**<br>**( Coast Recovery means HS feedback to replace input HS)**<br>0: Non Inversion (Default)<br>1: Inversion |
| 6 | R/W | **Select HS_OUT Source Signal**<br>0: Bypass (SeHs)(Use in Separate Mode)<br>1: Select De-Composite HS out(DeHs) (In Composite mode) |
| 5 | R/W | **Select ADC_VS Source Signal (Auto switch in Auto Run Mode)**<br>0: VS_RAW<br>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 |
| | | 1: Select SeHS or DeHS by CR49[6] (Default) |
| 1:0 | R/W | **Measure HSYNC/VSYNC Source Select:** |
| | | 00: TMDS |
| | | 01: VIDEO8 |
| | | 10: ADC_HS1/ADC_VS (Default) |
| | | 11: CS_RAW/VS_RAW |

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

| Bit | Mode | Function |
|---|---|---|
| 7 | R | **Even/Odd Field of YPbPr** (By Line-Count Mode) |
| | | 0: Even |
| | | 1: Odd |
| 6 | R | **The Toggling of Polarity of YPbPr Field Happened** (By Line-Count Mode) |
| | | 0: No toggle |
| | | 1: Toggle |
| 5 | R | **Even/Odd Field of YPbPr** (By VS-Position Mode) |
| | | 0: Even |
| | | 1: Odd |
| 4 | R | **The Toggling of Polarity of YPbPr 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**<br>000: No Signal<br>001: Not Support<br>010: YPbPr<br>011: Serration Composite SYNC<br>100: XOR/OR-Type Composite SYNC with Equalizer<br>101: XOR/OR-Type Composite SYNC without Equalizer<br>110: HSYNC with VS_RAW (Separate HSYNC)<br>111: HSYNC without VS_RAW (HSYNC only)<br>Reference when Hsync type detection auto run result ready (CR4C[7]) |
| 3 | R/W | **Reserved to 0** |
| 2:0 | R/W | **VSYNC counter level count [10:8] MSB**<br>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)**<br>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**<br>0: Period or polarity can't get continuous stable status.<br>1: Both polarity and period are stable. |

| | | |
|---|---|---|
| 6 | R | **Stable Polarity**<br>0: Negative<br>1: Positive<br>Compare each line's polarity; if we get continuous Ñ 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:**<br>00: 32 (Default)<br>01: 64<br>10: 96<br>11: 128 |
| 3 | R/W | **Stable Period Tolerance**<br>0: ±2 crystal clks (Default)<br>1: ±4 crystal clks |
| 2 | R/W | **VSYNC measure invert Enable**<br>0: Disable (Default)<br>1: Enable |
| 1 | R/W | **Pop Up Stable Value**<br>0: No Pop Up (Default)<br>1: Pop Up Result, (CR4A[2:0], CR4B[7:0], CR4E[3], CR50[2:0], CR51[7:0]) |
| 0 | R/W | **Stable Measure Start**<br>0 : Stop (Default)<br>1 : Start |

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

| Bit | Mode | Function |
|---|---|---|
| 7 | R | **Measure One Frame Status**<br>0: Finished after 1 frame measuring / Measure finished<br>1: Measuring Now |
| 6 | R | **CS_RAW Inverted by Auto Run Mode**<br>0: Not inverted<br>1: Inverted |
| 5 | R/W | **HS_OUT Bypass PLL into VGIP**<br>0: Disable (Default)<br>1: Enable |
| 4 | R/W | **Active Region Measure Enable**<br>0: Disable (Default)<br>1: Enable |
| 3 | R/W | **ADC_VS Source Select in Test Mode**<br>0: Select ADC_VS Source in Normal Mode or Auto Mode by CR47[6] (Default) |

| Bit | Mode | Function |
|---|---|---|
| | | 1: Select ADC_VS Source in Test Mode (Select VS_RAW or DeVS by CR49[5]) |
| 2:0 | R | **Stable Period[10:8]** <br> 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]** <br> 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** <br> 0: Disable (Default) <br> 1: Enable |
| 6 | R/W | **Pop Up Period Measurement Result** <br> 0: No Pop Up (Default) <br> 1: Pop Up Result |
| 5 | R/W | **Start a HS & VS period / H & V resolution & polarity measurement (on line monitor)** <br> 0: Finished/Disable (Default) <br> 1: Enable to start a measurement, auto cleared after finished |
| 4 | R | **Over-flow bit of Input HSYNC Period Measurement** <br> 0: No Over-flow occurred <br> 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** <br> 0: negative polarity (high period is longer than low one) <br> 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)**<br>0: No Time Out<br>1: Time Out occurred |
| 4 | R | **Over-flow bit of Input VSYNC Period Measurement**<br>0: No Over-flow occurred<br>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)<br>**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)<br>**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**<br>00: Don't care (CR58[3] will never go high)<br>01: 4-clock<br>10: 8-clock<br>11: 16-clock |
| 5:4 | R/W | **VSYNC_MAX_DELTA**<br>00: Don't care (CR58[2] will never go high)<br>01: 2-HSYNC<br>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**<br>0: Disable<br>1: Enable (Default) |
| 0 | R/W | **HSYNC/VSYNC High Period Measurement Result Select**<br><br>0: HSYNC<br><br>1: VSYNC<br><br>(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<br>0: Clamp source from normal HS<br>1: Clamp source from CS_RAW |
| 6 | R/W | **Sync Processor Time-Clock Test Mode**<br><br>0: Normal (Default)<br><br>1: Enable Test Mode;    (switch 70ns-ck to the time-out & polarity counters) |
| 5:3 | R/W | Sync Processor Test Signals Output Selection<br>000: Disable On-line Sync-Processor Test-Signal Output (Default)<br>001: adc_vs, adc_hs, adc_field, sog, vs_raw, cs_raw, hs_out, coast<br>010: cs_hs, hs_yprpb_postiv, input_signal_be_inverted, search_finish,<br>    load_search_stable48_result, load_finish_stable48_result, cap_hit, cap_miss<br>011: cs_hs ,cap_window ,de_hs ,de_vs ,de_coast ,clamp_mask ,cap_hit ,cap_miss<br>100: cs_raw, hs_for_decmp, auto_det_rdy, auto_result_rdy, flg_cnt_is50ms, flg_cnt_is80ms,<br>    hs_for_mv, mv_occur<br>101: mode_det_of, stb_per_chg, stb_pol_chg, vs_raw_vld, cap_32unlock, eq_occur, hs_pol_chg,<br>    cap_unlock<br>110: vs1_meas, hs1_meas, meas_clk, ms_now, reg_ms_1_frame_now, hsper_of , vsper_of, |

| | | |
|---|---|---|
| | | ms_timeout |
| | | 111: adc_vs, clamp_mask, hs_clamp_g, hs_clamp_rb, vga_online_clamp3, vga_online_clamp2, vga_online_clamp1, vga_online_clamp0 |
| 2:0 | R | **The Number of Input HS between 2 Input VSYNC**. LSB bit [2:0] for YPbPr |

*Address:5B Reserved*

*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 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. |

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

| Bit | Mode | Function |
|---|---|---|
| 7:0 | R/W | **End of Output Clamp 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. |

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

| Bit | Mode | Function |
|---|---|---|
| 7:0 | R/W | **Start of Output Clamp 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. |

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

| Bit | Mode | Function |
|---|---|---|
| 7:0 | R/W | **End of Output Clamp 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. |

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

| Bit | Mode | Function |
|---|---|---|
| | | |

| Bit | Mode | Function |
|---|---|---|
| 7 | R/W | **Clamp Trigger Edge Inverse for Y/G Channel**<br><br>0: Trailing edge (Default)<br><br>1: Leading edge |
| 6 | R/W | **Clamp Trigger Edge Inverse for B/Pb and R/Pr Channel**<br><br>0: Trailing edge (Default)<br><br>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**<br><br>0: Disable (Default)<br><br>1: Enable |
| 6 | R/W | **Select Clamp Mask as De VS**<br><br>0: Disable<br><br>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**<br><br>0: ADC_Clock (Default)<br><br>1: Crystal Clock |
| 6 | R/W | **Clamp Counter Unit (0x5D-00 – 0x5D-03)**<br><br>0: Double Pixels (Default)<br><br>1: Single Pixel |
| 5 | R/W | **ADC1_clamp_enable**<br><br>0: Disable (Default)<br><br>1: Enable |
| 4 | R/W | **ADC0_clamp_enable**<br><br>0: Disable (Default)<br><br>1: Enable |
| 3 | R/W | **ADC-3 Clamp Source**<br><br>0: Clamp-G (Default)<br><br>1: Clamp-BR |
| 2 | R/W | **ADC-2 Clamp Source**<br><br>0: Clamp-G (Default)<br><br>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<br>5'h00: ±6 crystal clks for capture window<br>5'h01 ~ 5'b1F : ±1 ~ ±31 crystal clks for capture window |

**Address: 5D-09  DETECTION_TOLERANCE_SETTING**         Default: 00h

| Bit | Mode | Function |
|---|---|---|
| 7 | R/W | **Reserved to 0** |
| 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<br>5'h00: Disable De-bounce Function (Default)<br>5'h01 ~ 5'h1F : De-bounce 1 ~ 31 crystal clks for de-composite |

**Address: 5D-0A  DEVS_CAP_NUM_H**                   Default: 00h

| Bit | Mode | Function |
|---|---|---|
| 7:4 | R/W | **Reserved to 0** |
| 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~0F Reserved*

## Macro Vision

*Address: 5D-10*  **MacroVision Control**                                    **Default: 00h**

| Bit | Mode | Function |
|-----|------|----------|
| 7:4 | R/W | **Skip Line[3:0]**<br>  Skip Lines after Vsync detected |
| 3:2 | R/W | **Reserved to 0** |
| 1 | R | **MacroVision Detected** (On-line monitor)<br>  When detected Macrovision occurred, this bit set to 1, else clear to 0. |
| 0 | R/W | **MacroVision Enable**<br>0: Disable (Default)<br>1: Enable |

*Address: 5D-11*  **MacroVision Start Line in Even Field**

| Bit | Mode | Function |
|-----|------|----------|
| 7 | R/W | **Reserved to 0** |
| 6:0 | R | **MacroVision Start Line in Even Field [6:0]** |

*Address: 5D-12*  **MacroVision End Line in Even Field**

| Bit | Mode | Function |
|-----|------|----------|
| 7 | R | **Indicate the validity of Macro Vision Line in Even Field**<br>  0: not valid<br>  1: valid |
| 6:0 | R | **MacroVision End Line 0 [6:0]** |

*Address: 5D-13*  **MacroVision Start Line in Odd Field**

| Bit | Mode | Function |
|-----|------|----------|
| 7 | R/W | **Reserved to 0** |
| 6:0 | R | **MacroVision Start Line in Odd Field [6:0]** |

*Address: 5D-14*  **MacroVision End Line in Odd Field**

| Bit | Mode | Function |
|-----|------|----------|
| 7 | R | **Indicate the validity of Macro Vision Line in Odd Field**<br>  0: not valid<br>  1: valid |
| 6:0 | R | **MacroVision End Line in Odd Field [6:0]** |

*Address: 5D-15*  **Macro Vision Detect De-bounce**                         **Default: 00h**

| Bit | Mode | Function |
|-----|------|----------|
| 7:5 | R/W | **Reserved to 0** |

| 4:0 | R/W | H-sync for Macro-Vision Detection De-bounce Length<br>5'h00 ~ 5'h07: De-bounce 7 crystal clks for de-composite (Default)<br>5'h08 ~ 5'h1F: De-bounce 8 ~ 31 crystal clks for de-composite |
|-----|-----|------|
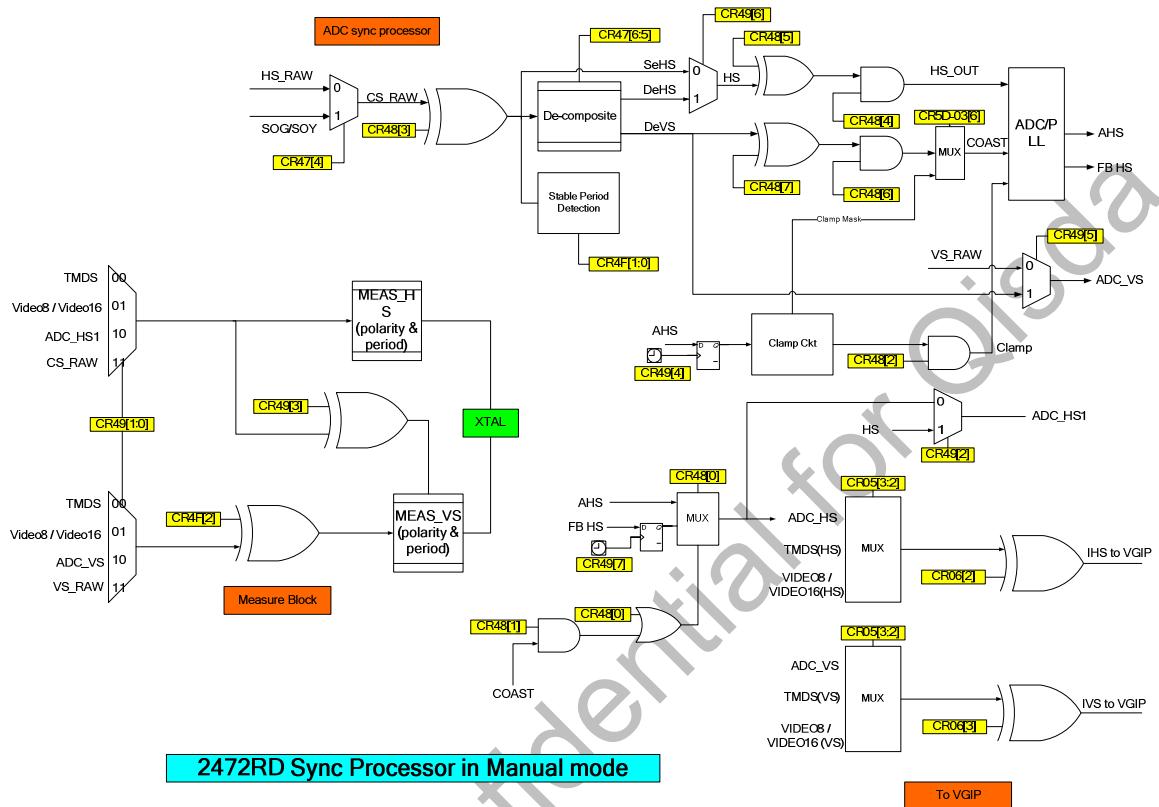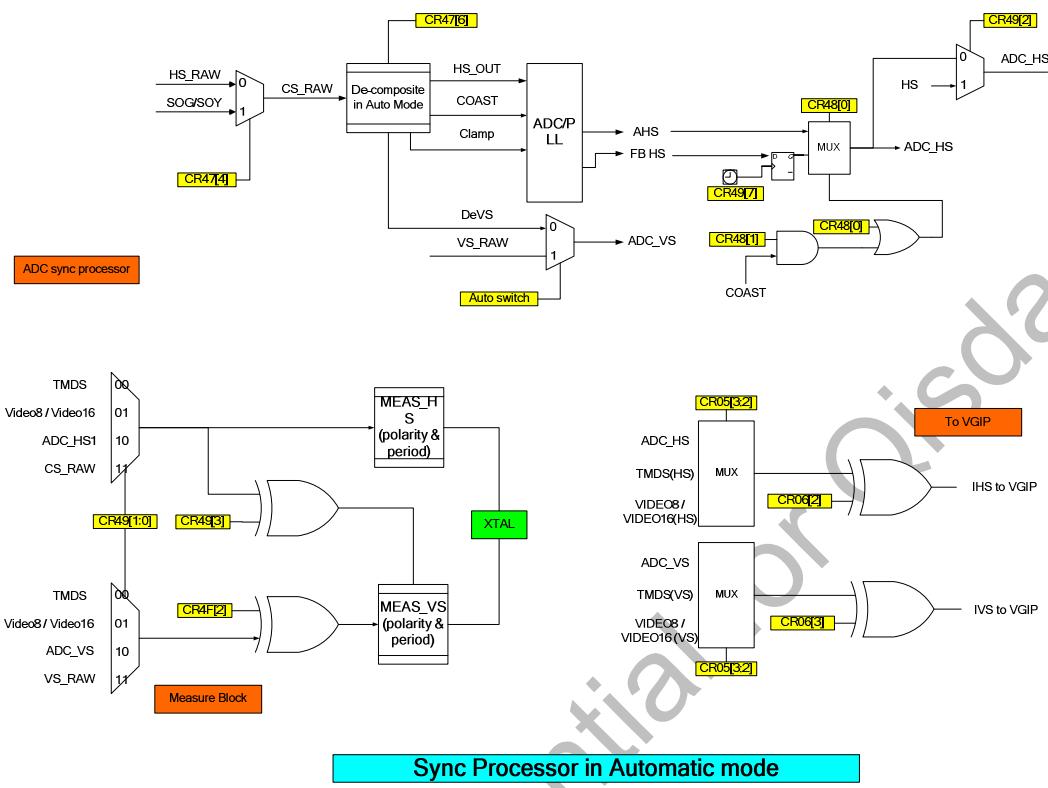


Figure 16:　Sync processor

Sync processor in Automatic mode

*Address 0x5E is reserved*

# 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** |

*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 |
|-----|------|----------|
| **Bit** | **Mode** | **Function** |

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

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

| Bit | Mode | Function |
|---|---|---|
| 7:6 | R/W | **Contrast / brightness application control**<br><br>00: Set A used on full region<br>01: Set B used inside highlight window<br>10: Set A used outside highlight window<br>11: Set A used outside highlight window, and Set B used inside highlight window |
| 5:4 | R/W | **Gamma application control**<br><br>00: gamma used on full region<br>01: gamma used inside window<br>10: gamma used outside window<br>11: reserved |

| **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 |

| **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 |

| Bit | Mode | Function |
|---|---|---|
| 3:2 | R/W | **DCC/ICM application control**<br><br>00: DCC/ICM used on full region<br>01: DCC/ICM used inside window<br>10: DCC/ICM used outside window<br>11: Reserved |

| ICM<br>(CRE0[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 |

| DCC<br>(CRE4[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 |

| Bit | Mode | Function |
|---|---|---|
| 1:0 | R/W | **Peaking/Coring application control**<br><br>00:Full region<br>01: Inside window<br>10: Outside window<br>11: Reserved |

| Peaking<br>(CR9A[6]) | Application control | Inside window | Outside window |
|---|---|---|---|
| 0 | X | bypass | bypass |
| 1 | CR61-0C[5:4]=00 \|\| CR60[6]=0 | Peaking | Peaking |
| 1 | CR61-0C[5:4]=01 && CR60[6]=1 | Peaking | bypass |
| 1 | CR61-0C[5:4]=10 && CR60[6]=1 | bypass | Peaking |

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

| Bit | Mode | Function |
|---|---|---|

| 7:6 | R/W | **sRGB application control** |
|---|---|---|
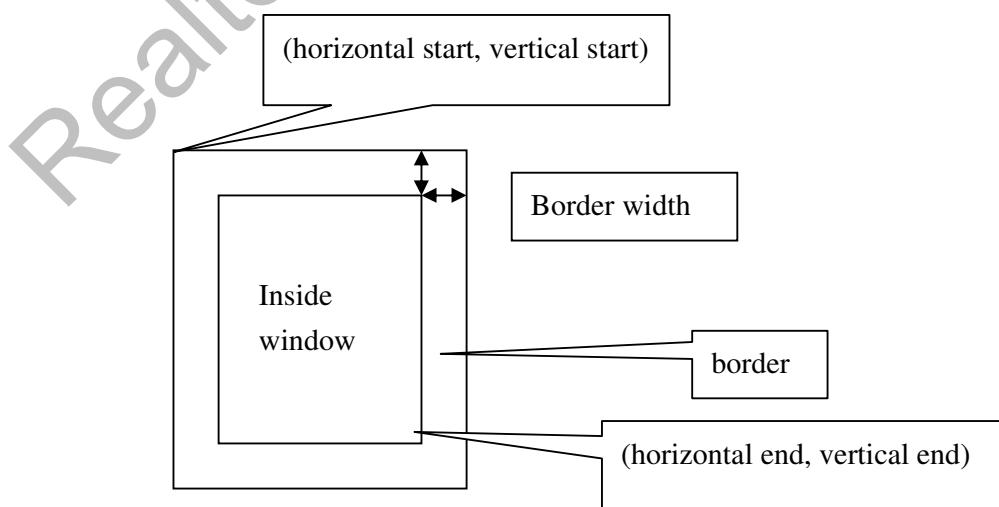
00: sRGB used on full region

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_APP_CTRL** |
|---|---|---|

00: DCR used on full region.

01: DCR used inside highlight window.

10: DCR used outside highlight window.

11: Reserved.

| DCR(Page 7 CRD8[0]) | Application control | Inside window | Outside window |
|---|---|---|---|
| 0 | X | bypass | bypass |
| 1 | CR61-0D[5:4]=00 \|\| CR60[6]=0 | DCR | DCR |
| 1 | CR61-0D[5:4]=01 && CR60[6]=1 | DCR | bypass |
| 1 | CR61-0D[5:4]=10 && CR60[6]=1 | bypass | DCR |

| 3:0 | -- | **Reserved to 0** |
|---|---|---|

(horizontal start, vertical start)

Border width

Inside window

border

(horizontal end, vertical end)

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

- sRGB output saturation to 1023 and Clamp to 0

- sRGB Output is 10 bit

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

## Contrast/Brightness Coefficient

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

| Bit | Mode | Function |
|---|---|---|
| 7 | R/W | **Enable Contrast /Brightness access port** |
| 6:4 | -- | **Reserved** |
| 3:0 | 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:** <br> 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:** <br> 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:** <br> 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:** <br> 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:** <br> 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:** <br><br> 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:** <br><br> 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:** <br><br> 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:** <br><br> 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:** <br><br> 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:** <br><br> 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:** <br><br> Valid range: 0(00h) ~ 1(80h)   ~ 2(FFh) |

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

## 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
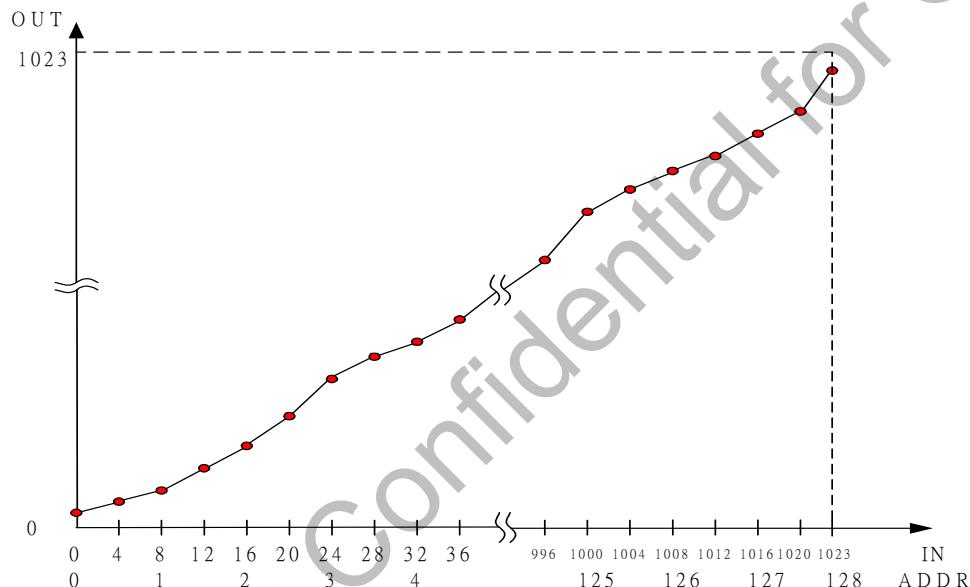
- When CR67[3] is set to 1, we can directly specify the initial address of Gamma Table in this port.

- 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}



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

| Bit | Mode | Function |
|-----|------|----------|
| 7 | R/W | **Enable Access Channels for Gamma Correction Coefficient:**<br>0: disable these channels (Default)<br>1: enable these channels |
| 6 | R/W | **Gamma table enable**<br>0: by pass (Default)<br>1: enable |
| 5:4 | R/W | **Color Channel of Gamma Table**<br>00: Red Channel    (Default)<br>01: Green Channel<br>10: Blue Channel<br>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. |
| 2:0 | -- | **Reserved to 0** |
| | | |

- Access Gamma_Access register will reset GAMMA_PORT index.

*Address: 68*      **GAMMA_BIST (Color Control Register)**          **Default: 00h**

| Bit | Mode | Function |
|---|---|---|
| 7 | R/W | **Test_mode**<br>0: Disable, dither_out = dither_result[9:2];      // truncate to integer number (Default)<br>1: Enable, dither_out = dither_result[7:0];      // propagate decimal part for test |
| 6 | R/W | **sRGB multiplier coefficient precision**<br>**0: 1-bit Shift-left (Default)**<br>**1: 2-bit Shift-left** |
| 5:2 | -- | **Reserved to 0** |
| | | |
| 1 | R/W | **Gamma BIST_Progress**<br>0: BIST is done (Default)<br>1: BIST is running |
| 0 | R | **Gamma BIST Test Result(It will go low first during BIST period)**<br>0: SRAM Fail<br>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 |
|-----|-----|
| D02 | D03 |

| D10 | D11 |
|-----|-----|
| D12 | D13 |

| D20 | D21 |
|-----|-----|
| D22 | D23 |

| D30 | D31 |
|-----|-----|
| 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** <br> 00: disable (Default) <br> 01: enable access dithering sequence table <br> 10: enable access dithering table <br> 11: enable access temporal offset | |
| Dither_en | 5 | R/W | 0 | **Enable Dithering Function** <br> 0: disable (Default) <br> 1: enable | |
| Dither_temp | 4 | R/W | 0 | **Temporal Dithering** <br> 0: Disable (Default) <br> 1: Enable | |
| Dither_table | 3 | R/W | 0 | **Dithering Table Value Sign** <br> 0: unsigned <br> 1: signed (2's complement) | |
| Dither_mode | 2 | R/W | 0 | **Dithering Mode** <br> 0: New (Default) <br> 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** <br> 0: Disable (Default) <br> 1: Enable | |

| Register:: DITHERING_CTRL2 | | | | | 0x6B |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| Reserved | 7:1 | R/W | **0** | **Reserved** | |
| Dither_Table_Ref | 0 | R/W | 1 | **Table reference** <br> 0: By VS/HS <br> 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** <br> 0: Disable(Reset CR6D Write Pointer to R) <br> 1: Enable |
| 4:2 | R/W | **Alpha blending level (Also enable OSD frame control register 0x003 byte 1[3:2]** <br> 000: Disable (Default) <br> 001 ~111: 1/8~ 7/8 |
| 1 | R/W | **Overlay Sampling Mode Select:** <br> 0: single pixel per clock (Default) <br> 1: dual pixels per clock (The OSD will be zoomed 2X in horizontal scan line) |
| 0 | R/W | **Overlay Port Enable:** <br> 0: Disable (Default) <br> 1: Enable <br> **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:** <br> 0: Disable (Default) <br> 1: Enable |
| 6 | R/W | **Reserved to 0** |
| 5:0 | R/W | **Overlay 16x24 Look-Up-Table Write Address [5:0]** |

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

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

| Bit | Mode | Function |
|---|---|---|
| 7:0 | W | **Color Palette 16x24 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, … R15, G15, and B15] and the address counter will be automatic increment and circular from 0 to 47.

## 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** |
| 1:0 | -- | **Reserved to 0** |

*Address: 79* **DIFF_THRESHOLD**

| Bit | Mode | Function |
|-----|------|----------|
| 7:0 | R/W | **Difference Threshold**<br>(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.<br>0: Disable (Default)<br>1: Enable |
| 6 | R/W | **Field_Select:** Select Even or Odd field. Active when **Field_Select_Enable**.<br>0: Active when ODD signal is "0" (Default)<br>1: Active when ODD signal is "1" |
| 5 | **R/W** | **Low Pass Filter (121-LPF)**<br>0: Disable (Default)<br>1: Enable |
| 4 | R/W | **Auto Function Acceleration :**<br>0: Disable (Default)<br>1: Enable<br>For auto-balance (CR7D[1]=0), this function must be disabled. |
| 3:2 | R/W | **Vertical boundary search:**<br>00: 1 pixel over threshold (Default)<br>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] |

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

| Bit | Mode | Function |
|-----|------|----------|
| 7:0 | R | **Auto phase measurement result:** bit[7:0] <br> The measured value of R or G or B color max or min. (Auto-Balance) |

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:: DITHERING_DATA_ACCESS | | | | | 0x88 |
|-----|------|-------|-------|----------|--------|
| Name | Bits | Read/ | Reset | Comments | Config |

| | | Write | State | | |
|---|---|---|---|---|---|
| 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 |
|---|---|
| D02 | D03 |

| D10 | D11 |
|---|---|
| D12 | D13 |

| D20 | D21 |
|---|---|
| D22 | D23 |

| D30 | D31 |
|---|---|
| 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:: DITHERING_CTRL1 | | | | | 0x89 | |
|---|---|---|---|---|---|---|
| **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** <br> 0: New (Default) <br> 1: Old | |
| Dither_V_Fram_M | 1 | R/W | 0 | **Vertical Frame Modulation** <br> 0: Disable (Default) <br> 1: Enable | |
| Dither_VH_Fram_M | 0 | R/W | 0 | **Horizontal Frame Modulation** <br> 0: Disable (Default) <br> 1: Enable | |

*Address 0x8A are reserved*

## Embedded Timing Controller

**Address: 8B**     **TCON_ADDR _PORT**                               **Default: 00h**

| Bit | Mode | Function |
|---|---|---|
| 7:0 | **R/W** | **Address port for embedded TCON access** |

**Address: 8C**     **TCON_DATA _PORT**                               **Default: 00h**

| Bit | Mode | Function |
|---|---|---|
| 7:0 | **R/W** | **Data port for embedded TCON access** |

**Address: 8C-00**   **TC_CTRL0 (Timing Controller control register1)**      **Default: 01h**

| Bit | Mode | Function |
|---|---|---|
| 7 | R/W | **Enable Timing Controller Function (Global)** <br> 0: Disable (Default) <br> 1: Enable <br> All TCON pins will be initialized when enabled and goes low when disabled. |
| 6 | R/W | **TCON [n] Toggle Function Reset** <br> 0: Not reset (Default) <br> 1: reset by DVS |
| 5 | R/W | **Inactive Period Data Controlled by internal TCON [13]** <br> 0: DEN (Default) <br> 1: TCON [13] |
| 4 | R/W | **TCON_HS compensation** <br> 0: Real TCON_HS = TCON_HS-4 <br> 1: Real TCON_HS = TCON_HS-27 |

| | | If setting TCON_HS > DH_Total, then setting TCON_HS must subtract DH_Total. |
|---|---|---|
| 3 | --- | **Reserve to 0** |
| 2 | --- | **Reserve to 0** |
| 1:0 | R/W | **DISP_TYPE**<br>01: LVDS (Default)<br>others are reserved |

**Address: 8C-01** TC_CTRL1 (Timing Controller control register1)  Default: 00h

| Bit | Mode | Function |
|---|---|---|
| 7:0 | R/W | **Reserved to 0** |

**Address: 8C-02** Pixel Threshold MSB  Default: 00h

| Bit | Mode | Function |
|---|---|---|
| 7 | R/W | **2-Line Sum of Difference Threshold 1 Value: bit [8], ie:TH1 (also refer to CR8C-03)** |
| 6 | R/W | **2-Line Sum of Difference Threshold 2 Value: bit [8], ie:TH2 (also refer to CR8C-04)** |
| 5:0 | R/W | **Over Difference Line Threshold Value: bit [9:4]**<br>Notes: Bit[3:0] are zeros |
| | | |

**Address: 8C-03** Pixel Threshold High Value for Smart Polarity (TH1)  Default: 00h

| Bit | Mode | Function |
|---|---|---|
| 7:0 | R/W | **2 line Sum of Difference Threshold 1 Value: bit [7:0], ie:TH1 (Also refer to CR8C-02[7])** |

**Address: 8C-04** Pixel Threshold Low Value for Smart Polarity (TH2)  Default: 00h

| Bit | Mode | Function |
|---|---|---|
| 7:0 | R/W | **2 line Sum of Difference Threshold 2 Value: bit [7:0], ie:TH2 (Also refer to CR8C-02[6])** |

**Address: 8C-05** Line Threshold Value for Smart Polarity  Default: 00h

| Bit | Mode | Function |
|---|---|---|
| 7 | R/W | **Measure Dot Pattern over Threshold**<br>1: Run.<br>   Auto: always measure (Reference to CR8C-05[5])<br>   Manual: start to measure, clear after finish<br>0: Stop |
| 6 | R | **Dot Pattern Sum of Difference Measure Result**<br>1: Over threshold<br>0: Under threshold |
| 5 | R/W | **Anti-Flicker Auto-Measure Control**<br>1: Auto<br>0: Manual |

| Bit | Mode | Function |
|---|---|---|
| 4:1 | R/W | **Reserved** |
| 0 | R/W | **Anti-Flicker Measure Mode**<br>0: Dot-Based (Original)<br>1: Pixel-Based |

**Over Difference Line Threshold Value shall not exceed 0x190.**

*Address: 8C-06~07 Reserved to 0*

## TCON Horizontal/Vertical Timing Setting

**Address: 8C-08** **TCON [0]_VS_LSB (TCON [0] Vertical Start LSB Register)**

| Bit | Mode | Function |
|---|---|---|
| 7:0 | W | **Line number [7:0] at which TCON control generation begins** |

**Address: 8C-09** **TCON [0]_VS_MSB (TCON [0] Vertical Start/End MSB Register)**

| Bit | Mode | Function |
|---|---|---|
| 7:4 | W | **Line number [11:8] at which TCON control generation ends** |
| 3:0 | W | **Line number [11:8] at which TCON control generation begins** |

**Address: 8C-0A** **TCON [0]_VE_LSB (TCON [0] Vertical End LSB Register)**

| Bit | Mode | Function |
|---|---|---|
| 7:0 | W | **Line number [7:0] at which TCON control generation ends** |

**Address: 8C-0B** **TCON [0]_HS_LSB (TCON [0] Horizontal Start LSB Register)**

| Bit | Mode | Function |
|---|---|---|
| 7:0 | W | **Pixel count [7:0] at which TCON goes active** |

**Address: 8C-0C** **TCON [0]_HS_MSB (TCON [0] Horizontal Start/End MSB Register)**

| Bit | Mode | Function |
|---|---|---|
| 7:4 | W | **Pixel count [11:8] at which TCON goes inactive** |
| 3:0 | W | **Pixel count [11:8] at which TCON goes active** |

To be triggered on rising edge of the DCLK

**Address: 8C-0D** **TCON [0]_HE_LSB (TCON [0] Horizontal End LSB Register)**

| Bit | Mode | Function |
|---|---|---|
| 7:0 | W | **Pixel count [7:0] at which TCON goes inactive** |

If the register number is large than display format, the horizontal component is always on.

Real TCON_HS = TCON_HS-4, Real TCON_HS = TCON_HS-4

**Address: 8C-0E** **TCON [0]_CTRL (TCON [0] Control Register)**      **Default: 00h**

| Bit | Mode | Function |
|---|---|---|
| 7 | R/W | **TCON [n] Enable** (Local)<br>0: Disable (TCON [n] output clamp to '0') (Default)<br>1: Enable |

| 6 | R/W | **Polarity Control** |
|---|---|---|
| | | 0: Normal output (Default) |
| | | 1: Inverted output |
| 5:4 | -- | **Reserved to 0** |
| 3 | R/W | **Toggle Circuit Enable/Disable** |
| | | 0: Normal TCON output (Default) |
| | | 1: Toggle Circuit enable |
| | | When using toggle circuit enable mode, the TCON[n] will be 1 clock earlier than TCON[n-1] and then toggling together, finally output will be 1 clock delay comparing to toggling result. |
| 2:0 | R/W | **TCON [13:10] & TCON [7:4] (TCON Combination Select)** |
| | |    **TCON [13] has inactive data controller function.** |
| | |    **TCON [13]~[10] has dot masking function** |
| | |    **TCON [7] has flicking reduce function.** |
| | | 000: Normal TCON output (Default) |
| | | 001: Select TCON [n] "AND" with TCON [n-1] |
| | | 010: Select TCON [n] "OR" with TCON [n-1] |
| | | 011: Select TCON [n] "XOR" with TCON [n-1] |
| | | 100: Select TCON [n-1] rising edge as toggle trigger signal (when toggle enable) |
| | | 101: Select TCON [n-1] rising edge as toggle trigger signal, then "AND" (when toggle enable) |
| | | 110: Select TCON [n-1] rising edge as toggle trigger signal, then "OR" (when toggle enable) |
| | | 111: Select TCON [n] and TCON [n-1] on alternating frames. |
| | | ----------------------------------------------------------------------------------------------------------------- |
| | | **TCON [9:8] (TCON Combination Select)** |
| | | 000: Normal TCON output |
| | | 001: Select TCON [n] "AND" with TCON [n-1] |
| | | 010: Select TCON [n] "OR" with TCON [n-1] |
| | | 011: Select TCON [n] "XOR" with TCON [n-1] |
| | | 100: Select TCON [n-1] rising edge as toggle trigger signal (when toggle enable) |
| | | 101: Select TCON [n-1] rising edge as toggle trigger signal, then "AND" (when toggle enable) |
| | | 110: Select TCON [n-1] rising edge as toggle trigger signal, then "OR" (when toggle enable) |
| | | 111: Select TCON [n] and TCON [n-1] reference ODD signal as alternating frames. |
| | | ----------------------------------------------------------------------------------------------------------------- |
| | | **TCON [3] (TCON Combination Select)** |
| | | 000: Normal TCON output |
| | | 001: Select TCON [3] "AND" with TCON [2] |
| | | 010: Select TCON [3] "OR" with TCON [2] |
| | | 011: Select TCON [3] "XOR" with TCON [2] |
| | | 100: Select TCON [2] rising edge as toggle trigger signal (when toggle enable) |

101: Select TCON [2] rising edge as toggle trigger signal, then "AND" (when toggle enable)

110: Select TCON [2] rising edge as toggle trigger signal, then "OR" (when toggle enable)

111: Select reset(ODD=0) or set(ODD=1) TCON [3] by DVS, when toggle function enable

---------------------------------------------------------------------------------------------------------------------------

**TCON [2] (Clock Toggle Function**)//toggle function is inactive

　00x: Normal TCON output

　010: Select DCLK/2 when TCON [2] is "0"

　011: Select DCLK/2 when TCON [2] is "1"

　100: Select DCLK/4 when TCON [2] is "0"

　101: Select DCLK/4 when TCON [2] is "1"

　110: Select DCLK/8 when TCON [2] is "0"

　111: Select DCLK/8 when TCON [2] is "1"

---------------------------------------------------------------------------------------------------------------------------

**TCON [1]**

　xx0: Normal TCON output

　xx1: Reverse-Control Signal output

---------------------------------------------------------------------------------------------------------------------------

**TCON [0]**

　00x: Normal TCON output

　010: EVEN "REV" 18/24-bit function ("REV0" on TCON [0])

　　　ODD "REV" 18/24-bit function ("REV1" on TCON [1])

　011: ALL "REV" 36/48-bit function ("REV" on TCON [0], can also on TCON [1])

　100: EVEN data Output Inversion Controlled by TCON [0] is "0"

　　　ODD data Output Inversion Controlled by TCON [1] is "0"

　101: EVEN data Output Inversion Controlled by TCON [0] is "1"

　　　ODD data Output Inversion Controlled by TCON [1] is "1"

## Dot Masking

**Address: 8C-5F/67/6F/77　　TC_DOT_MASKING_CTRL**　　　　　　　　　　　**Default: 00h**

| Bit | Mode | Function |
|-----|------|----------|
| 7:3 | R/W | **Reserved to 0** |
| 2 | R/W | **Red Dot Masking Enable**<br>0: Disable (Default)<br>1: Enable |
| 1 | R/W | **Green Dot Masking Enable**<br>0: Disable (Default)<br>1: Enable |

| 0 | R/W | **Blue Dot Masking Enable** |
|---|---|---|
| | | 0: Disable (Default) |
| | | 1: Enable |

When applying dot masking, the timing setting for TCON will be

Real TCON_Mask_STA = TCON_STA+2

Real TCON_Mask_END = TCON_END +2

**TCON [0] ~ TCON [13] Control Registers Address Map**

| Address | Data(# bits) | Default |
|---|---|---|
| 0A,09,08 | TCON [0]_VS_REG (11) | |
| 0D,0C,0B | TCON [0]_HS_REG (11) | |
| 0E | TCON [0]_CTRL_REG | 00 |
| 0F | Reserved | |
| | | |
| 12,11,10 | TCON [1]_VS_REG (11) | |
| 15,14,13 | TCON [1]_HS_REG (11) | |
| 16 | TCON [1]_CTRL_REG | 00 |
| 17 | Reserved | |
| | | |
| 1A,19,18 | TCON [2]_VS_REG (11) | |
| 1D,1C,1B | TCON [2]_HS_REG (11) | |
| 1E | TCON [2]_CTRL_REG | 00 |
| 1F | Reserved | |
| | | |
| 22,21,20 | TCON [3]_VS_REG (11) | |
| 25,24,23 | TCON [3]_HS_REG (11) | |
| 26 | TCON [3]_CTRL_REG | 00 |
| 27 | Reserved | |
| | | |
| 2A,29,28 | TCON [4]_VS_REG (11) | |
| 2D,2C,2B | TCON [4]_HS_REG (11) | |
| 2E | TCON [4]_CTRL_REG | 00 |
| 2F | Reserved | |
| | | |
| 32,31,30 | TCON [5]_VS_REG (11) | |
| 35,34,33 | TCON [5]_HS_REG (11) | |
| 36 | TCON [5]_CTRL_REG | 00 |

| 37 | Reserved | |
| | | |
| 3A,39,38 | TCON [6]_VS_REG (11) | |
| 3D,3C,3B | TCON [6]_HS_REG (11) | |
| 3E | TCON [6]_CTRL_REG | 00 |
| 3F | Reserved | |
| | | |
| 42,41,40 | TCON [7]_VS_REG (11) | |
| 45,44,43 | TCON [7]_HS_REG (11) | |
| 46 | TCON [7]_CTRL_REG | 00 |
| 47 | Reserved | |
| | | |
| 4A,49,48 | TCON [8]_VS_REG (11) | |
| 4D,4C,4B | TCON [8]_HS_REG (11) | |
| 4E | TCON [8]_CTRL_REG | 00 |
| 4F | Reserved | |
| | | |
| 52,51,50 | TCON [9]_VS_REG (11) | |
| 55,54,53 | TCON [9]_HS_REG (11) | |
| 56 | TCON [9]_CTRL_REG | 00 |
| 57 | Reserved | |
| | | |
| 5A,59,58 | TCON [10]_VS_REG (11) | |
| 5D,5C,5B | TCON [10]_HS_REG (11) | |
| 5E | TCON [10]_CTRL_REG | 00 |
| 5F | TCON [10]_CTRL_REG | |
| | | |
| 62,61,60 | TCON [11]_VS_REG (11) | |
| 65,64,63 | TCON [11]_HS_REG (11) | |
| 66 | TCON [11]_CTRL_REG | 00 |
| 67 | TCON [11]_CTRL_REG | 00 |
| | | |
| 6A,69,68 | TCON [12]_VS_REG (11) | |
| 6D,6C,6B | TCON [12]_HS_REG (11) | |
| 6E | TCON [12]_CTRL_REG | 00 |
| 6F | TCON [12]_CTRL_REG | 00 |

| | | |
|---|---|---|
| 72,71,70 | TCON [13]_VS_REG (11) | |
| 75,74,73 | TCON [13]_HS_REG (11) | |
| 76 | TCON [13]_CTRL_REG | 00 |
| 77 | TCON [13]_CTRL_REG | 00 |
| | | |

## Control for LVDS

**Address: 8C-A0**　　　**LVDS_CTRL0**　　　　　　　**Default: 00h**

| Bit | Mode | Function |
|---|---|---|
| 7 | -- | **Reserved to 0** |
| | | |
| 5 | R/W | **Power Up LVDS Even-Port (pin 86~95)**<br>0: Power down (Default)<br>1: Normal |
| 4 | R/W | **Power Up LVDS Odd-Port (pin 74~83)**<br>0: Power down (Default)<br>1: Normal |
| 3:2 | R/W | **Watch Dog Model**<br>00: Enable Watch Dog(Default)<br>01: Keep PLL VCO = 1V<br>1x: Disable Watch Dog |
| 1 | R/W | **Reserved to 0** |
| 0 | R | **Watch Dog Control Flag**<br>0: Watch dog not active (Default)<br>1: Watch dog active, Reset PLL and set VCO = 1V |

**Address: 8C-A1**　　　**LVDS_CTRL1**　　　　　　　**Default: 14h**

| Bit | Mode | Function |
|---|---|---|
| 7 | R/W | **CKLAGL:** Inverse the CK port<br>0: lead (Default)<br>1: lag T/14 |
| | | |
| 5:3 | R/W | **STSTL [2:0]: select test attribute**<br>000: WD<br>001: VCOM<br>010: IB40u (default)<br>011: IBVOCM<br>100: PLLTST-fbak |

| | | 101: PLLTST-fin |
| | | 110: LVTST-CKDIN |
| | | 111: LVTST-LVDSIN[6] |
| 2:0 | R/W | **LVDS Output Common Mode** (Default: 100) |
| | | 000 : 1.07v |
| | | 001 : 1.12v |
| | | 010 : 1.17v |
| | | 011 : 1.22v |
| | | 100 : 1.29v (Default) |
| | | 101 : 1.33v |
| | | 110 : 1.38v |
| | | 111 : 1.43v |

| Address: 8C-A2 | | LVDS_CTRL2 | Default: 43h |
|---|---|---|---|
| **Bit** | **Mode** | **Function** | |
| 7:6 | R/W | **SBGL** 00: 1.164V 01: 1.244V (Default) 10: 1.324V 11: 1.404V | |
| 5 | R/W | **ENIB40UX2L: Double the LVDS output swing** 0: 1X 1: 2X | |
| 4 | R/W | **SIBXL** : select 20uA source If DISP_TYPE=LVDS 0: from Bandgap (Default) 1: from ADC | |
| 3 | R/W | **PLL lock edge** 0: positive 1: negative | |
| 2:0 | R/W | **SIBGENL (LVDS Current Source correction),** **40u: LVDS driving current, 100u: LVDS VOCM(default)** 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 |
|---|---|---|
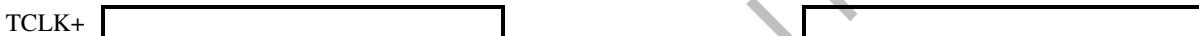
**Address: 8C-A3**　　　　**LVDS_CTRL3**　　　　　　　　**Default: 1Ch**

| Bit | Mode | Function |
|---|---|---|
| 7 | R/W | **ENVBPBL: Enable VCO_D2S Current Up**<br>0: disable (Default)<br>1: enable |
| 6 | R/W | **Reserved to 0** |
| 5:3 | R/W | **SIL [2:0]**: PLL charge pump current (I=5uA+5uA*code) (Default: 011) |
| 2:1 | R/W | **SRL [1:0]**: PLL resistor (R=6K+2K*code) (Default: 10) |
| 0 | R/W | **BMTS**: Bit-Mapping Table Select<br>0: Table 1 (Default)<br>1: Table 2 |

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)

**Address: 8C-A4**　　　　**LVDS_CTRL4**　　　　　　　　**Default: 80h**

| Bit | Mode | Function |
|---|---|---|
| 7:6 | R/W | **E_RSV**: even port reserve signal select<br>11: Always '1' |

| | | 10: Always '0' |
| | | 01: TCON [11] |
| | | 00: PWM_0 |
| 5:4 | R/W | **E_DEN**: even port data enable signal select |
| | | 11: Always '1' |
| | | 10: Always '0' |
| | | 01: TCON [9] |
| | | 00: DENA |
| 3:2 | R/W | **E_VS**: even port VS signal select |
| | | 11: Always '1' |
| | | 10: DENA |
| | | 01: TCON [7] |
| | | 00: DVS |
| 1:0 | R/W | **E_HS**: even port HS signal select |
| | | 11: Always '1' |
| | | 10: DENA |
| | | 01: TCON [5] |
| | | 00: DHS |

| Address: 8C-A5 | | LVDS_CTRL5 | Default: 80h |
|---|---|---|---|
| **Bit** | **Mode** | **Function** | |
| 7:6 | R/W | **O_RSV**: odd port reserve signal select | |
| | | 11: Always '1' | |
| | | 10: Always '0' | |
| | | 01: TCON [13] | |
| | | 00: PWM_1 | |
| 5:4 | R/W | **O_DEN**: odd port data enable signal select | |
| | | 11: Always '1' | |
| | | 10: Always '0' | |
| | | 01: TCON [9] | |
| | | 00: DENA | |
| 3:2 | R/W | **O_VS**: odd port VS signal select | |
| | | 11: Always '1' | |
| | | 10: DENA | |
| | | 01: TCON [7] | |
| | | 00: DVS | |
| 1:0 | R/W | **O_HS**: odd port HS signal select | |
| | | 11: Always '1' | |
| | | 10:DENA | |

| | | 01: TCON [5] |
|---|---|---|
| | | 00: DHS |

**Address: 8C-A6**      **LVDS_CTRL6**           **Default: 02h**

| Bit | Mode | Function |
|---|---|---|
| 7 | R/W | **RSDS Differential pair PN swap (data)** (Also refer to CR29[6:4])<br><br>0: No Swap (Default)<br><br>1: Swap |
| 6:4 | -- | Reserved to 0 |
| 3 | R/W | **DALAGL:** Inverse the Data port<br><br>0: lead (Default)<br><br>1: lag T/14 |
| 2 | R/W | **Reserved** |
| 1 | R/W | **ENDUL:** Odd clock pair current double(odd B port)<br><br>0: 1X<br><br>1: 2X (Default) |
| 0 | -- | **Reserved to 0** |

## 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** | |

| Register::TEST_MODE | | | | | 0x00 |
|---|---|---|---|---|---|
| **Name** | **Bit** | **R/W** | **Default** | **Description** | **Config** |
| Select_data_test_mode | 7 | R/W | 0 | Select Data Test mode MSB(for 48pin)<br>0: select Data test mode [15:0] to pin 36~21<br>1: select Data test mode [29:16] to pin 36~21 | |
| Test mode select | 6:5 | R/W | 00 | 00:Normal<br>01:test_output mode<br>Others are Reserved | |

| Test_output_Mode | 4 | R/W | 0 | 0:Select Data test mode<br>Select Data test output to <u>128pin{124~121,114~108,105~100,72~64, 54~51} depend on bit6~bit4</u><br><u>48pin(36~21)depend on bit7~bit4</u><br>1:PLL test mode<br>{dpll, m2pll,audio_pll, mpll, ckt_pll27x, ck108_pll27x, dpllstatus,test1out, test2out, fav4, xclk} will be outputted to <u>128pin{124,113,51,71,110,101,108, 105,54,67,64}</u><br>48pin{29,37,21,35,none,none,25,23,27,3 3,31}<br>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[9:2], Green[9:2], Blue[9:2] through VGIP<br>0001: 1'b0, Z0TST[3:0], adc_clk, Red[7:0], Green[7:0], Blue[7:0] After Scale Down<br>**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<br>**0011:** Z0TST[3:0], adc_clk, MCUWR, MCURD, MCU_ADR_INC, MIN[7:0], MCUWR, MCURD, MADR[7:0], SDMOUT_TST[3:0]<br>**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<br>**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<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, 1'b0<br>**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]<br>**1000:** AUDIO_DAC enable signal test pin: dac_2ch_otpin[29:0]<br>**1001:** VSDMAIN test mode: pclk_tst, 3'b0, vsd_act, vsd_den, vsd_pr[7:0], vsd_y[7:0], vsd_pb[7:0]<br>**1010:** ~~Auto_soy test mode~~ reserved<br>**1011:** 6'b0, ADC SNR RGB 8-bitx3 output<br>**1100:** Embedded MCU test out mode | |

| | | | | **1101:** HDMI test in mode<br>128 pin HDMI_TST_IN [29:0] assign to {124~121, 114~108, 105~100, 72~64, 54~51}<br>48 pin HDMI_TST_IN [15:0] assign to {36~21}<br>**1110:** HDMI test out mode<br>128 pin HDMI_TST_OUT [29:0] assign to {124~121, 114~108, 105~100, 72~64, 54~51}<br>48 pin HDMI_TST_OUT [15:0] assign to {36~21}<br>**1111:** 6'b0, ADC SNR RGB 8-bitx3 output reserved<br>When set to 0010/0110/0111,test_s1 & test_s2 can be assigned by "Select_Tst_s1s2"<br>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<br>0: output disabled<br>1: output enabled | |
| M2pll_OEN | 6 | R/W | 0 | M2PLL frequency output enable<br>0: output disabled<br>1: output enabled | |
| Audio_pll_OEN | 5 | R/W | 0 | Audio_PLL frequency output enable<br>0: output disabled<br>1: output enabled | |
| MPLL_OEN | 4 | R/W | 0 | MPLL frequency output enable<br>0: output disabled<br>1: output enabled | |
| CLK108_PLL27X_OEN | 3 | R/W | 0 | CLK108_PLL27X frequency output enable<br>0: output disabled<br>1: output enabled | |
| Test1out_OEN | 2 | R/W | 0 | Test1out frequency output enable<br>0: output disabled<br>1: output enabled | |
| Test2out_OEN | 1 | R/W | 0 | Test2out frequency output enable<br>0: output disabled<br>1: output enabled | |
| Fav4_OEN | 0 | R/W | 0 | Fav4 frequency output enable<br>0: output disabled<br>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<br>0: output disabled<br>1: output enabled | |
| CKT_PLL27X_OEN | 6 | R/W | 0 | CKT_PLL27X frequency output enable | |

| | | | | 0: output disabled<br>1: output enabled | |
|---|---|---|---|---|---|
| Rev | 5:0 | --- | --- | Reserved | |

| Register::TST_CLK_CTRL2 | | | | | 0x03 |
|---|---|---|---|---|---|
| **Name** | **Bit** | **R/W** | **Default** | **Description** | **Config** |
| MPLL_DIV_CTRL | 7:6 | R/W | 00 | MPLL frequency is divided by<br>00:1<br>01:2<br>10:4<br>11:8 | |
| DPLL_DIV_CTRL | 5:4 | R/W | 00 | DPLL frequency is divided by<br>00:1<br>01:2<br>10:4<br>11:8 | |
| M2pll_DIV_CTRL | 3:2 | R/W | 11 | M2PLL frequency is divided by<br>00:1<br>01:2<br>10:4<br>11:8 | |
| Audio_pll_DIV_CTRL | 1:0 | R/W | 00 | Audio PLL frequency is divided by<br>00:1<br>01:2<br>10:4<br>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<br>00:1<br>01:2<br>10:4<br>11:8 | |
| Test1out _DIV_CTRL | 5:4 | R/W | 00 | Test1out frequency is divided by<br>00:1<br>01:2<br>10:4 | |

| | | | | 11:8 | |
|---|---|---|---|---|---|
| Test2out _DIV_CTRL | 3:2 | R/W | 00 | Test2out frequency is divided by<br>00:1<br>01:2<br>10:4<br>11:8 | |
| 108_pll27xDIV_CTRL | 1 | R/W | 0 | 0:Divided by 1<br>1:Divided by 4 | |
| Ckt_pll27x<br>DIV_CTRL | 0 | R/W | 0 | 0:divided by 1<br>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**<br>**3'b000: DPLL clock (TIE LOW NOW)**<br>**3'b001: PLLS fbk clock**<br>**3'b010: CKOAD2(High Speed)**<br>**3'b011: PLL status**<br>**3'b100: HSOUT**<br>**3'b101: ADC clock(from PLLS)(High Speed)**<br>**3'b110: Empty Flag**<br>**3'b111: BVS(Video8)** | |
| Reserved | 3 | R/W | 0 | **Reserved** | |
| Select_Tst_s2 | 2:0 | R/W | 010 | **Select test function of test_s1**<br>**3'b000: PLLS phase swallow clock (High speed)**<br>**3'b001: DPLL status(TIE LOW NOW)**<br>**3'b010: PLLS phase0 clock(High speed)**<br>**3'b011: M2PLL clock(Not in APLL)**<br>**3'b100: HSFB**<br>**3'b101: TP2_MX5**<br>**3'b110: Full Flag**<br>**3'b111: BHS(Video8)** | |

| **Register::** Select_Tstinclock | | | | | **0x06** |
|---|---|---|---|---|---|
| **Name** | **Bit** | **R/W** | **Default** | **Description** | **Config** |
| DPLL_TST_IN | 7 | **R/W** | 0 | **0:Normal** <br> **1:DCLK enter from pin 34** | |
| ADCPLL_TST_IN | 6 | **R/W** | 0 | **0:Normal** <br> **1:ADC CLK enter from pin 35** | |
| M2PLL_TST_IN | 5 | **R/W** | 0 | **0:Normal** <br> **1:M2PLL CLK enter from pin 36** | |
| HDMI_CP_ACLK _TST_IN | 4 | **R/W** | 0 | **0:Normal** <br> **1:HDMI_CP_ACLK enter from pin 37** | |
| HDMI_CP_CLK _TST_IN | 3 | **R/W** | 0 | **0:Normal** <br> **1:HDMI_CP_CLK enter from pin 38** | |
| SCAN_CLK_TST_I N | 2 | **R/W** | 1 | **0:Normal** <br> **1:SCAN CLK enter from pin 21** | |
| DPLL_NDIV2_EN | 1 | **R/W** | 0 | **DPLL Test Mode Divider Enable** <br> **0:use pin 34 div2 as dclk** <br> **1:use pin 34 as dclk** | |
| MPLL_TST_IN | 0 | **R/W** | 0 | **0:Normal** <br> **1:MPLL CLK enter from pin 33** | |

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** <br> **1:Enable** | |
| ADC_TEST_MODE _2 | 6 | **R/W** | 0 | **Useless** | |
| FIFO_TEST_MODE | 5 | **R/W** | 0 | 0:Disable <br> 1:Enable test the CRC from FIFO，and open the Pattern Gen to d domain. <br> 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** |
| Rev | 7:2 | --- | --- | Reserved | |
| ADC_TEST_ADDR[9:8] | 1:0 | R/W | 0X00 | Read the FIFO initial Addr. | |

| 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 | |

## 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**<br>0: stop (Default)<br>1: start (auto clear) |
| 6 | R | **BIST Result**<br>0: fail (Default)<br>1: success |
| 5 | R | **MCU writes data when OSD ON status (Queue 1 byte data)**<br>0: MCU writes data to OSD but not to real position (There is one level buffer here)<br>1: MCU doesn't write data, or data has been written to real position |
| 4 | R | **Double_Buffer_Write_Status**<br>0: double buffer write out is finish, or data write to double buffer is not ready, or no double buffer function.<br>1: after data write to dbuf and before dbuf write out, such that double buffer is busy. |
| 3 | R/W | **OSDADRHSB**<br>**0: If initial address lower than or equal to 12K**<br>**1: If initial address higher than 12K**<br>The bit will be designed to control 16.5K 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)**<br>**000~101=>1~6** |

*Address: 94* **OSD_TEST**

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

*Address:95~97 Reserved*

# Digital Filter

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

| Bit | Mode | Function |
|-----|------|----------|
| 7:4 | R/W | **Access Port Write Enable**<br>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] = 000 ~ 110** | | |
| **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 and mismatch or offset value of 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 |

## Color Conversion (Input Domain)

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

| Bit | Mode | Function |
|---|---|---|
| 7 | R/W | Y_OUT Shift<br> 0: Bypass<br> 1: Y_Out+64 |
| 6 | R/W | CbCr_Out_Shift:<br> 0 : Bypass<br> 1: Cb_Out+512, Cr_Out+512 |
| 5 | --- | Reserved |
| 4 | R/W | Color Conversion Type<br>0: YUV->RGB<br> 1: RGB->YUV (U,V are translated to unsigned 10-bit number) |
| 3 | R/W | Enable YUV/RGB coefficient Access:<br> 0: Disable<br> 1: Enable<br>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<br> 0: Bypass |

| | | 1: Cb-512, Cr-512 (MSB Inversion) |
|---|---|---|
| 1 | R/W | Y Gain/Offset:<br> 0 : Bypass<br> 1: (Y-64)*1.164 |
| 0 | R/W | Enable YUV <-> RGB Conversion:<br> 0: Disable YUV<->RGB conversion (Default)<br> 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)

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.

Coefficient Sequence (18-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).

Default value: h00=0105h,h01=0202h,h02=0063h,h10=ff69h,h11=fed8h,h12=01c0h,h20=01c0h,h21=fe89h,h22=ff8h

| | | | | | |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

## 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)<br>Page 0: Embedded ADC/ABL/LVR/Smith trigger<br>Page 1: PLL<br>Page 2: HDMI/DVI/HDCP<br>Page 3: LiveShowTM Control<br>Page 4: SDRAM Control<br>Page 5: SDR_FIFO Control<br>Page 6: Reserved<br>Page 7: Vivid color/DCC/ICM<br>Page 8: Reserved<br>Page 9: Reserved<br>Page A: Reserved<br>Page B: DisplayPort<br>Page C: DisplayPort<br>Page D: MCU<br>Page E: MCU<br>Page F: MCU<br>Page 10: SDRAM Pin Control<br>Others: reserved |

# Embedded ADC (Page 0)

| Register::ADC_POWER_SOG_SOY_CONTROL[7:0] | | | | | 0XBA |
|---|---|---|---|---|---|
| **Name** | **Bit** | **R/W** | **Default** | **Description** | **Config** |
| Reserved | 7:6 | R/W | 0b0 | Reserved | |
| ADC_SOG1_DAC | 5:0 | R/W | 0b100000 | Reserved(SOG0/1 DAC input), 1-A only, SOG1 Reserved. | |
| | | | | | |

| 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** | 0b0 | 2x Clock Polarity (0:Normal 1:Inverted) | |
| ADC_2X_SAMPLE[5] | 5 | **R/W** | 0b0 | 1x Clock Polarity (0:Normal 1:Inverted) | |
| ADC_2X_SAMPLE[4:3] | 4:3 | **R/W** | 0b00000 | Reserved | |
| ADC_2X_SAMPLE[2] | 2 | **R/W** | 0b0 | <span style="color:red">clock input select (0:from CKOAD_V33, 1:from CKOAD_V12)</span> | |
| ADC_2X_SAMPLE[1:0] | 1:0 | **R/W** | 0b0 | Reserved | |

| 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 Stablizer(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 | R,B Clamp Vaule from G (0: No 1: Yes) ADC Gain Calibration | | |
| ADC_TEST[6:4] | 6:4 | **R/W** | 0b000 | Test Ouput Selection (PAD: SOGIN0, SOGIN1) SOGIN0 (000:X 001:gnd 010:gnd 011:gnd 100:gnd 101:vmid 110:voffset 111:vdd) | | |
| ADC_TEST[3] | 3 | **R/W** | 0b0 | Reserved | | |
| ADC_TEST[2] | 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 | --- | 0 | Reserved | |
| ADC_POWER[6] | 6 | R/W | 0 | 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 | --- | 0 | Reserved | |
| ADC_POWER[3] | 3 | R/W | 0b1 | Bandgap 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[7:4] | 7:4 | R/W | 0101 | Reserved | |
| 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 | -- | Reserved | |

| Register:: ADC_IBIAS1[7:0] | | | | | 0XC8 |
|---|---|---|---|---|---|
| Name | Bit | R/W | Default | Description | Config |
| ADC_IBIAS1[7:6] | 7:6 | R/W | 0b01 | Reserved | |
| ADC_IBIAS1[5:4] | 5:4 | R/W | 01 | Reserved | |
| ADC_IBIAS1[3:2] | 3:2 | R/W | 0b01 | Bias Current of LSADC6 (00:16u 01:20u 10:24u 11:28u) | |
| ADC_IBIAS1[1:0] | 1:0 | R/W | 0b01 | Bias Current of LSADC10 (00:16u 01:20u 10:24u 11:28u) | |

| Register:: ADC_IBIAS2[7:0] | | | | | 0XC9 |
|---|---|---|---|---|---|
| Name | Bit | R/W | Default | Description | Config |
| ADC_IBIAS2[7:5] | 7:5 | R/W | 0b01 | Reserved | |
| | | | | | |
| ADC_IBIAS2[4:2] | 4:2 | R/W | 0b001 | Bias Current of DAC (000:22.5u 001:25u 010:27.5u 011:30u) | |
| ADC_IBIAS2[1:0] | 1:0 | R/W | 0b01 | Bias Current of Audio_DAC (00:32u 01:40u 10:48u 11:56u) | |

| 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** | 0b100 | Bias Current of SH,MDAC (000:6u 001:8u 010:10u 011:12u 100:14 101:17 110:20u 111:24u) Bias Current of SUBADC (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 (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[3:2] | 3:2 | **R/W** | 0b01 | Reserved | |
| ADC_IBIAS4[1:0] | 1:0 | **R/W** | 0b01 | Bias Current of M2PLL_20u (00:16u 01:20u 10:24u 11:28u) | |

| Register:: ADC_ VBIAS0[7:0] | | | | | 0XCC |
|---|---|---|---|---|---|
| **Name** | **Bit** | **R/W** | **Default** | **Description** | **Config** |
| ADC_VBIAS0[7:6] | 7:6 | **R/W** | 0b01 | Reserved | |
| ADC_VBIAS0[5:4] | 5:4 | **R/W** | 0b01 | 1.5v regulator    adjuset (00:1.4    01:1.5 10:1.6 11:1.7) | |
| ADC_VBIAS0[3:2] | 3:2 | **R/W** | 0b00 | Reserved | |
| 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 | ADC Gain Calibration (0: Normal 1: Calibration) | |
| 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[1:0] | 1:0 | **R/W** | 0b01 | Reserved | |

| 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 | Bandwidth (00: 75M 01: 150M 10: 300M 11: 500M) | |

| 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_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_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[7:4] | 7:4 | R/W | 0b0000 | SOG0 input MUX select (0000: R0, 0001:G0, 0010:B0, 0011: SOG0) | |
| ADC_SOG_CMP[3:0] | 3:0 | R/W | 0b0000 | Reserved | |

| Register:: ADC_DCR_CTRL[7:0] | | | | | 0XD3 |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Bit** | **R/W** | **Default** | **Description** | **Config** |
| ADC_DCR_CTRL[7] | 7 | R/W | 0 | Red_0 DC Restore Enable (0:Disable 1:Enable) | |

| Name | Bit | R/W | Default | Description | Config |
|------|-----|-----|---------|-------------|--------|
| ADC_DCR_CTRL[6] | 6 | **R/W** | 0 | Green_0 DC Restore Enable (0:Disable 1:Enable) | |
| ADC_DCR_CTRL[5] | 5 | **R/W** | 0 | Blue_0 DC Restore Enable (0:Disable 1:Enable) | |
| ADC_DCR_CTRL[4] | 4 | **R/W** | 0 | SOG0 DC Restore Enable(0:Disable 1:Enable) | |
| ADC_DCR_CTRL[3] | 3 | **R/W** | 0 | Reserved | |
| ADC_DCR_CTRL[2] | 2 | **R/W** | 0 | Reserved | |
| ADC_DCR_CTRL[1] | 1 | **R/W** | 0 | Reserved | |
| ADC_DCR_CTRL[0] | 0 | **R/W** | 0 | Reserved | |

| Register:: ADC_CLAMP_CTRL0[7:0] | | | | | 0XD4 |
|---------------------------------|-----|-----|---------|-------------|--------|
| **Name** | **Bit** | **R/W** | **Default** | **Description** | **Config** |
| ADC_CLAMP_CTRL0[7] | 7 | **R/W** | 0 | Red_0 Clamp Enable (0: Disable 1:Enable) | |
| ADC_CLAMP_CTRL0[6] | 6 | **R/W** | 0 | Green_0 Clamp Enable (0: Disable 1:Enable) | |
| ADC_CLAMP_CTRL0[5] | 5 | **R/W** | 0 | Blue_0 Clamp Enable (0: Disable 1:Enable) | |
| ADC_CLAMP_CTRL0[4] | 4 | **R/W** | 0 | SOG0 Clamp Enable(0:Disable 1:Enable) | |
| ADC_CLAMP_CTRL0[3] | 3 | **R/W** | 0 | Reserved | |
| ADC_CLAMP_CTRL0[2] | 2 | **R/W** | 0 | Reserved | |
| ADC_CLAMP_CTRL0[1] | 1 | **R/W** | 0 | Reserved | |
| ADC_CLAMP_CTRL0[0] | 0 | **R/W** | 0 | Reserved | |

| Register:: ADC_CLAMP_CTRL1[7:0] | | | | | 0XD5 |
|---------------------------------|-----|-----|---------|-------------|--------|
| **Name** | **Bit** | **R/W** | **Default** | **Description** | **Config** |

| ADC_CLAMP_CTR L1[7] | 7 | R/W | 0 | Red channel clamp voltage (0: IR=400mV   1: IR=100mV) | |
|---|---|---|---|---|---|
| ADC_CLAMP_CTR L1[6] | 6 | R/W | 0 | Red channel clamp source select   (0: R0=DAC, R1=IR   1: R0=IR, R1=DAC) | |
| ADC_CLAMP_CTR L1[5] | 5 | R/W | 0 | Green channel clamp voltage (0: IR=400mV   1: IR=100mV) | |
| ADC_CLAMP_CTR L1[4] | 4 | R/W | 0 | Green channel clamp source select   (0: G0=DAC, G1=IR   1: G0=IR, G1=DAC) | |
| ADC_CLAMP_CTR L1[3] | 3 | R/W | 0 | Blue channel clamp mode select (0: IR=400mv   1: IR=100mV) | |
| ADC_CLAMP_CTR L1[2] | 2 | R/W | 0 | Blue channel clamp source select   (0: B0=DAC, B1=IR   1: B0=IR, B1=DAC) | |
| ADC_CLAMP_CTR L1[1] | 1 | R/W | 0 | SOG0 clamp -300mV (0: noraml clamp 1:clamp -300m) //IR | |
| ADC_CLAMP_CTR L1[0] | 0 | R/W | 0 | Reserved | |

| Register:: ADC_CLAMP_CTRL2[7:0] | | | | | 0XD6 |
|---|---|---|---|---|---|
| **Name** | **Bit** | **R/W** | **Default** | **Description** | **Config** |
| ADC_CLAMP_CTR L2[7:6] | 7:6 | R/W | 0b01 | Input0 and SOG0 DC restore resister (00:open   01:500k   10:1M   11:5M) | |
| ADC_CLAMP_CTR L1[5:4] | 5:4 | R/W | 0b01 | *Reserved* | |
| ADC_CLAMP_CTR L1[3] | 3 | R/W | 0 | RGB input range adjust (0: 0.5V-1.0V, 1:0.25V-1.25V) | |
| ADC_CLAMP_CTR L1[2] | 2 | R/W | 0 | Red channel clamp to top (0: noraml 1: top) | |
| ADC_CLAMP_CTR L1[1] | 1 | R/W | 0 | Green channel clamp to top (0: noraml 1: top) | |
| ADC_CLAMP_CTR L1[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 | --- | 0b0 | Reserved | |
| ADC_SOG0_DAC[5 | 5:0 | R/W | 0b100000 | SOG0 DAC input | |

| :0] | | | | | |
|---|---|---|---|---|---|

<br>

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

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

<br>

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

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

<br>

| Register:: EBD_REGLATOR_VOL[7:0] | | | | | 0XDF |
|---|---|---|---|---|---|
| **Name** | **Bit** | **R/W** | **Default** | **Description** | **Config** |
| **EBD_REGLATOR_** <br> **VOL[7:6]** | 7:6 | **R/W** | 0b00 | Reserved | |
| RBG33 | 5:4 | **R/W** | 00 | Select bandgap output voltage @ TT60 <br><br> 00 : Vbg = 1.22 V <br><br> 01 : Vbg = 1.13 V <br><br> 10 : Vbg = 1.32 V <br><br> 11 : Vbg = 1.41 V | |
| VINSEL | 3 | **R/W** | 0 | Select reference voltage to REG_OP <br><br> 0 : from bandgap | |

| | | | | | |
|---|---|---|---|---|---|
| | | | | 1 : from " Vdd of power supply\* resistance ratio " | |
| VSEL | 2:0 | **R/W** | | Select output voltage of regulator<br>000 : 1.203 V<br>001 : 1.143 V<br>010 : 1.085 V<br>011 : 1.246 V<br>100 : 1.298 V<br>101 : 1.355 V<br>110 : 1.424 V | |
| | | | 000 | 111 : 1.508 V | |

## ABL(Page 0)

*Address: E2*    **AUTO_BLACK_LEVEL_CTRL1**                    **Default: 00h**

| Bit | Mode | Function |
|---|---|---|
| 7 | R/W | **ABL Mode**<br><br>0: RBG (Default)<br><br>1: YPbPr |
| 6 | R/W | **On-line/Off-line ABL Mode**<br><br>0: Off-line (Default)<br><br>1: On-line |
| 5:4 | R/W | **Width of ABL region in each line**<br><br>00: 16 pixels (Default)<br><br>01: 32 pixels<br><br>10: 64 pixels<br><br>11: 4 pixels |
| 3 | R | **R/Pr Channel ABL Result (write clear)**<br><br>0: not equal<br><br>1: equal (Black Level = Target Value)<br><br>On-line mode: \|Black Level - Target Value\| <=LOCK_MGN<br><br>Off-line mode: \|Black Level - Target Value\| <= EQ_MGN |
| 2 | R | **G/Y Channel ABL Result (write clear)**<br><br>0: not equal<br><br>1: equal (Black Level = Target Value)<br><br>On-line mode: \|Black Level - Target Value\| <= LOCK _MGN<br><br>Off-line mode: \|Black Level - Target Value\| <= EQ_MGN |
| 1 | R | **B/Pb Channel ABL Result (write clear)**<br><br>0: not equal<br><br>1: equal (Black Level = Target Value)<br><br>On-line mode: \|Black Level - Target Value\| <= LOCK _MGN<br><br>Off-line mode: \|Black Level - Target Value\| <= EQ_MGN |
| 0 | R/W | **Auto Black Level Enable (write 0 force stop)**<br><br>0: Finished/Disable (Default)<br><br>1: Enable to start ABL, auto cleared after finished<br><br>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: 84h**

| Bit | Mode | Function |
|-----|------|----------|
| 7:6 | R/W | **Line averaged for each ABL adjustment**<br><br>00: 8<br><br>01: 16<br><br>10: 32 (Default)<br><br>11: 64 |
| 5 | -- | **Reserved** |
| 4:0 | R/W | **Start Vertical Position of ABL in each line**<br><br>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**<br><br>0000: 1<br><br>0001: 2 (Default)<br><br>0010: 3<br><br>0011: 4<br><br>….<br><br>1111:16<br><br>(Pb/Pr Target level is fixed 128) |
| 3:2 | R/W | **Lock Margin**<br><br>00: 1 (Default)<br><br>01: 2<br><br>10: 4<br><br>11: 6 |
| 1:0 | R/W | **End Vertical Position of ABL measurement region [9:8]**<br><br>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: 82h**

| Bit | Mode | Function |
|---|---|---|
| 7:0 | R/W | **End Vertical Position of ABL measurement region [7:0]**<br>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: 04h**

| Bit | Mode | Function |
|---|---|---|
| 7:0 | R/W | **Start Position of ABL in Each Line**<br>Determine the start position of auto-black-level after the trailing edge of reference signal.<br>(When ABL mode in YPbPr, the reference signal is input Hsync. In RGB mode, the reference signal is clamp signal.) |

- 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: C0h**

| Bit | Mode | Function |
|---|---|---|
| 7:6 | R/W | **Large Error Margin (L_MGN) (For on-line Mode)**<br>00: 2<br>01: 4<br>10: 6<br>11: 8 (Default) |
| 5:4 | R/W | **Max. Frame/Field Count (For off-line mode)**<br>00: 4 (Default)<br>01: 5<br>10: 6<br>11: 7 |
| 3 | -- | **Reserved** |
| 2:0 | R/W | **Lines delayed between each measurement region (For on-line Mode)**<br>000: 16 (Default)<br>001: 32<br>010: 64<br>011: 128<br>100: 192<br>101: 256<br>110: 384<br>111: 640 |

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

| Bit | Mode | Function |
|-----|------|----------|
| 7 | -- | **Reserved** |
| 6 | R/W | **Equal Condition (Off-line mode)**<br>0: To trigger status until measurement achieve Max Frame/Field Count.<br>1: To trigger status once if \|Black Level - Target Value\| <= EQ_MGN. (Default)<br>(If set 0, the ABL Result will not go low even noise comes for the next frames.) |
| 5 | R/W | **Measure Pixels Method**<br>1: Minimum value (Default)<br>0: Average value |
| 4 | R/W | **Measure Error Flag Reset**<br>0: Normal<br>1: Reset |
| 3 | R | **Measure Error Flag**<br>0: Normal<br>1: Error<br>(This flag is occurred when Hsync trailing edge is met during measurement.) |
| 2 | R/W | **Hsync Start Reference Select**<br>0: HS leading edge (Default)<br>1: HS trailing edge |
| 1:0 | R/W | **Equal margin (EQ_MGN)**<br>00: 0 (Default)<br>01: 1<br>10: 2<br>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.)** |

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

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

| Bit | Mode | Function |
|-----|------|----------|
| 7:0 | R | Noise Value of Blue Channel in Test Mode after Equal status is triggered. (only show MSB 8bit.) |
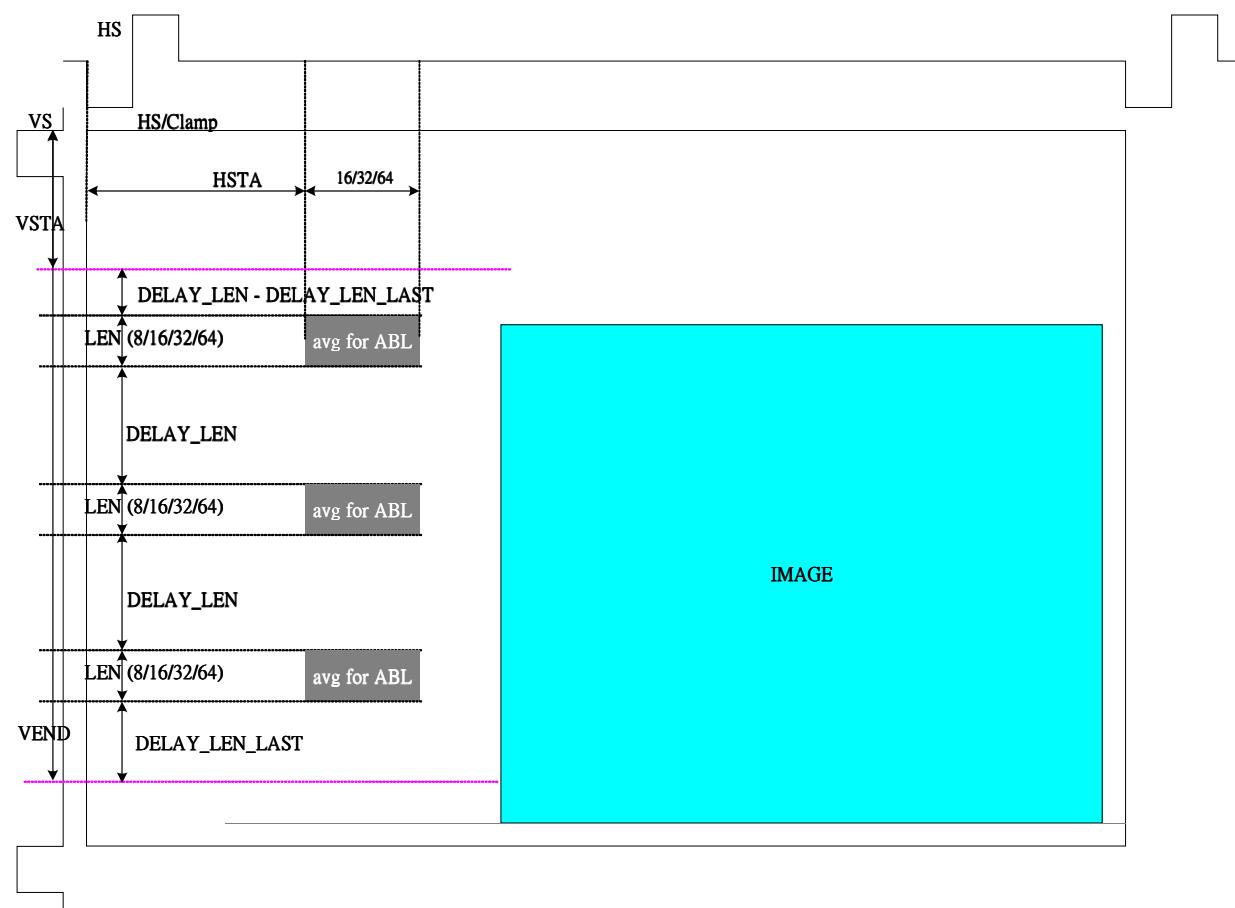
Figure-2: Auto Black Level active region – case 1
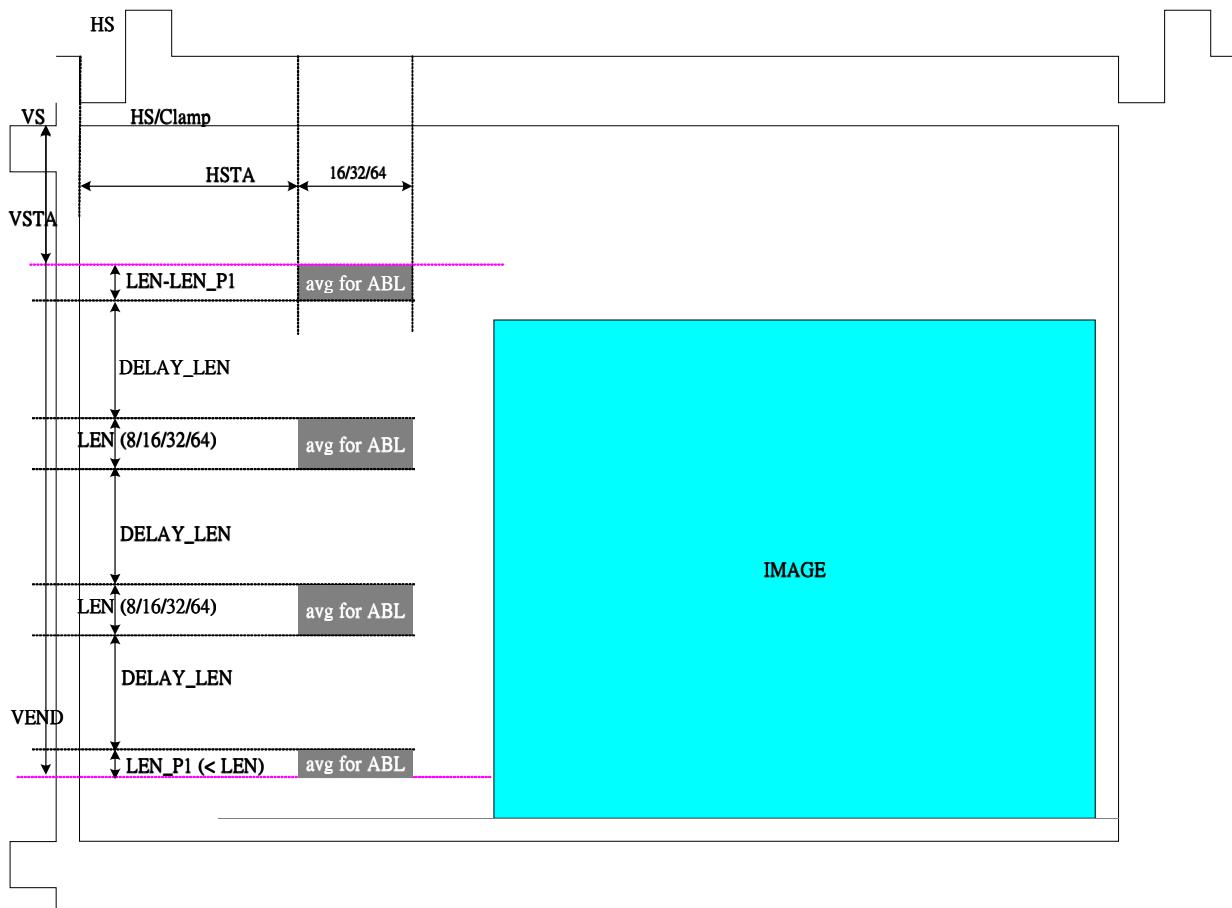


Figure-3: Auto Black Level active region – case 2

## LVR(Page 0)

| Bit | Mode | Function |
|-----|------|----------|
| 7:6 | R/W | **Negative Threshold Value For Power on Reset** <br><br> 00:1.8V <br><br> 01:2.2V <br><br> 10:2.0V (Default) <br><br> 11:2.4V |
| 5:4 | R/W | **PORMCUVSET (LVR Threshold Voltage)** <br><br> 00:1.22V <br><br> 01:1.13V (Default) <br><br> 10:1.32V <br><br> 11:1.41V |
| 3:0 | R/W | **Reserved to 0x04** |

## Smith trigger(Page 0)

*Address:F 4*　　　**HS_SCHMITT_TRIGGE_CTRL**　　　　　　　　Default:E1h

| Bit | Mode | Function |
|---|---|---|
| 7 | R/W | **HSYNC Schmitt Power Down (Only for Schmitt trigger new mode)**<br><br>0: Power down<br><br>1: Normal (Default) |
| 6 | R/W | **Polarity Select**<br><br>0: Negative HSYNC (high level)<br><br>1: Positive HSYNC (low level) (Default) |
| 5 | R/W | **Schmitt Trigger Mode**<br><br>0: Old mode<br><br>1: New mode(Default) |
| 4 | R/W | **Threshold Voltage Fine Tune (only for Schmitt trigger new mode)**<br><br>0: 0V (Default)<br><br>1: -0.1V |
| 3:2 | R/W | **Positive Threshold Voltage** |
| 1:0 | R/W | **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 = $V_t^- - 0.1* bit[4]$

## MEMORY PLL (Page 0)

| Register::MPLL_M | | | | 0xF5 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| MPLL_M[7:0] | 7:0 | R/W | 4E | **MPLL DPM value - 2** | |

| Register::MPLL_N | | | | 0xF6 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| MPLL_RESERVE1 | 7 | R/W | 0 | **Reserved** | |

| MPLL_BPN | 6 | R/W | 0 | **MPLLBPN**<br><br>0: N divider enable.<br><br>1: N divider disable, OUT=ckxtal. | |
|---|---|---|---|---|---|
| MPLL_O[1:0] | 5:4 | R/W | 1 | **MPLL Output Divider**<br><br>00: Div1<br><br>01: Div2 (Default)<br><br>10: Div4<br><br>11: Div8 | |
| MPLL_N[3:0] | 3:0 | R/W | 3 | **MPLL DPN value - 2** | |

- Assume MPLL_M=0x7D, DPM=0x7D+2=127; MPLL_N=0x0A, DPN=0x0A+2=12; Divider=1/4, F_IN = 24.576MHz. F_MPLL = F_IN x DPM / DPN x Divider  = 24.576 x 127 / 12 / 4 = 65.024MHz.

CRF5~CRF6 are double buffer.

| Register::MPLL_CRNT | | | | 0xF7 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| MPLL_RS[2:0] | 7:5 | R/W | 3 | **MPLL Loop Filter Resister Control**<br><br>000: 16K<br><br>001: 18K<br><br>010: 20K<br><br>011: 22K (Default)<br><br>100: 24K<br><br>101: 26K<br><br>110: 28K<br><br>111: 30K | |
| MPLL_CS[1:0] | 4:3 | R/W | 2 | **MPLL Loop Filter Capacitor Control**<br><br>00: 18p<br><br>01: 20p<br><br>10: 24p (Default)<br><br>11: 28p | |
| MPLL_IP[2:0] | 2:0 | R/W | 2 | **MPLL Charger Pump Current    Control**<br><br>Icp=(2.5uA+2.5uA*bit[0]+5uA*bit[1]+10uA*bit[2])<br><br>Keep DPM/Icp constant=10.67 | |

| Register::MPLL_WD | | | | 0xF8 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |

| Name | Bits | R/W | Default | Comments | Config |
|---|---|---|---|---|---|
| MPLL_WDO | 7 | R | 0 | **MPLL WD Status**<br>0: Normal<br>1: Abnormal | |
| MPLL_WDRST | 6 | R/W | 0 | **MPLL WD Reset**<br>0: Normal (Default)<br>1: Reset | |
| MPLL_WDSET | 5 | R/W | 0 | **MPLL WD Set**<br>0: Normal (Default)<br>1: Set | |
| MPLL_FUPDN | 4 | R/W | 1 | **MPLL Freqency Tuning**<br>0: Freq Up<br>1: Freq Dn(Default) | |
| MPLL_STOP | 3 | R/W | 1 | **MPLL Freqency Tuning**<br>0: Disable<br>1: Enable (Default) | |
| MPLL_FREEZE | 2 | R/W | 0 | **MPLL Output Freeze**<br>0: Normal (Default)<br>1: Freeze<br>Active high. | |
| MPLL_VCORSTB | 1 | R/W | 0 | **Reset VCO**<br>0: Normal (Default)<br>1: Reset<br>Active high. | |
| MPLL_PWDN | 0 | R/W | 1 | **Power Down MPLL**<br>0: Power on<br>1: Power down(Default)<br>Active high. | |

| Register::MPLL_CAL | | | | 0xF9 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| MPLL_VCOMD[1:0] | 7:6 | R/W | 3 | **MPLL VCO Default Mode**<br>00: VCO slowest<br>11: VCO fastest (Default) | |
| MPLL_CALBP | 5 | R/W | 0 | **MPLL Bypass Calibration**<br>0: Reference by Calibration Result(Default)<br>1: Reference by CR-F9[7:6]<br>Active high. | |

| | | | | | |
|---|---|---|---|---|---|
| MPLL_CALSW | 4 | R/W | 0 | **Calibration Validated**<br><br>Go high after power on 1200us.<br><br>0: Reference by CR-F9[7:6]<br><br>1: Refernect by Calibration Result | |
| MPLL_CALLCH | 3 | R/W | 0 | **Latch Calibration**<br><br>Go high after power on 1100us.<br><br>0: Disable Latch<br><br>1: Enable Latch | |
| MPLL_CMPEN | 2 | R/W | 0 | **CMP Enable**<br><br>Go high after power on 1000us.<br><br>0: Diable CMPEN<br><br>1: Enable CMPEN | |
| MPLL_CP | 1 | R/W | 0 | **CP Control**<br><br>0: 1.77pF`<br><br>1: 2.1pF | |
| MPLL_RESERVE | 0 | R/W | 1 | **Reserved for MPLL Phase Swallow Circuit**<br><br>0: Path0<br><br>1: Path1 | |

## ADC PLL (Page 1)

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

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

**PE_MEARSURE** Default: 8'b0xxxxxxx

| Bit | Mode | Function |
|-----|------|----------|
| 7 | R/W | **PE Measure Enable** (auto clear when finished)<br><br>0: Disable (Default)<br><br>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**<br><br>0: Disable (Default)<br><br>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**<br><br>0: clear (Default)<br><br>1: write '1' to clear PE Max. Value |
| 6 | R/W | **Enable APLL Setting**<br><br>0: Disable (Default)<br><br>1: Enable (Auto clear when finished)<br><br>When CRAA[5] enabled, enable this bit will write P_CODE, I_CODE, PLL M/N, PLL K, PLLDIV<br><br>and DDS SUM_I at the end of input vertical data enable |
| 5 | R/W | **Enable Fast PLL Mechanism**<br><br>0: Disable (Default)<br><br>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**<br>0: Disable (Default)<br>1: Enable (Auto clear when finished) |
| 3 | R/W | **DDS SUM_I Setting Updated Enable**<br>0: Disable (Default)<br>1: Enable (Auto clear when finished) |
| 2 | R/W | **Measure SUM_I**<br>0: Disable<br>1: Enable (Auto clear after finish) |
| 1 | R/W | **Enable Port AB**<br>0: Disable Port AB Access<br>1: Enable Port AB Access<br>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**<br>1: Select SUM_I_NOW [26:0] for read<br>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)**<br>**1st [00000, SUM_I [26:24]]**<br>**2nd SUM_I [23:16]**<br>**3rd SUM_I [15:8]**<br>**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: 09h** |
|---|---|---|---|
| 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** <br><br> 0: N is followed by the value of REG AD [2:0] <br><br> 1: N is always 1 | |
| 2:0 | R/W | **PLLN[2:0] (PLL DPN value – 2) (default is 000)** <br><br> **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)** <br><br> 000: 7K <br><br> 001: 9.5K <br><br> 010: 12K <br><br> 011: 14.5K(Default) | |

| Bit | Mode | Function |
|---|---|---|
| | | 100: 17K |
| | | 101: 19.5K |
| | | 110: 22K |
| | | 111: 24.5K |
| 4:0 | R/W | **PLLSI [4:0] (PLL Charger Pump Current IchDpll) (Default: 00011b)**<br>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)**<br>0: Normal (Default)<br>1: Abnormal |
| 6 | R/W | **PLLWDRST (PLL WD Reset)**<br>0: Normal (Default)<br>1: Reset |
| 5 | R/W | **PLLWDSET (PLL WD Set)**<br>0: Normal (Default)<br>1: Set |
| 4:3 | R/W | **PLLWDVSET[1:0] (PLL WD Voltage Set)**<br>00: 2.46V<br>01: 1.92V(Default)<br>10: 1.36V<br>11: 1.00V |
| 2 | R/W | **HS_dds2synp latch edge**<br>0: falling edge (Default)<br>1: rising |
| 1 | R/W | **Reset DDS**<br>0: normal (Default)<br>1: reset whole DDS |
| 0 | R/W | **PLLPWDN (PLL Power Down)**<br>0: Normal Run<br>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)**<br><br>00: Single Clock Mode (Default)<br><br>01: Single Inverse-Clock Mode<br><br>10: External Clock Mode<br><br>11: Dual Clock Mode (1x and 2x Clock) |
| 0 | R | **Swallow phase enable (K mask disabled)**<br><br>**The pll can't enable swallow phase function while pll just be power up. Waiting for 64 clock**<br><br>**cycles then start to enable phase swallow function.**<br><br>**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**<br><br>0: Manual<br><br>1: Look-Up-Table (default) |
| 5 | R/W | **PLLPH0PATH**<br><br>0: Short Path (Default)<br><br>1: Long Path (Compensate PLL_ADC path delay) |
| 4 | R/W | **PLLD2**<br><br>0:ADC CLK=1/2 VCO CLK (Default)<br><br>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]**.<br><br>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**<br>00: Original phase selected by X,Y and 16-phase pre-select<br>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:**<br>00: normal mapping P_CODE x G value (default)<br><br>01: nonlinear mapping I     smaller than Q(PE)   2   4   8   16   32   64<br>                 P_CODE x      1   2   4   8   32   128   128<br>10: nonlinear mapping II<br>                 P_CODE x      1   2   2   8   32   256   256<br>11: nonlinear mapping III<br>                 P_CODE x      1   2   8   16   32   128   512 |
| 5:2 | R/W | **G value**<br>0000: 0<br>0001: 1<br>0010: 4<br>0011: 16<br>0100: 64<br>0101: 128<br>0110: 256 (default)<br>0111: 512<br>1000: 1/4<br>1001: 1/16<br>1010: 1/64<br>1011: reserved to 0<br>1100: reserved to 0<br>1101: reserved to 0<br>1110: reserved to 0<br>1111: reserved to 0 |
| 1 | R/W | *Adaptive tracking enable for I_CODE*<br>0: disable to use adaptive I_CODE    (default)<br>1: enable to use adaptive I_CODE |
| 0 | R/W | *Adaptive tracking enable for P_CODE*<br>0: disable to use adaptive P_CODE    (default)<br>*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<br>00: 2 (default)<br>01: 4<br>10: 8<br>11: 15 |
| 5:4 | R/W | Threshold times to decide if starting adaptive tracking while Q(PE) < Threshold value successively<br>00: 3 (default)<br>01: 7<br>10: 11<br>11: 15 |
| 3 | R/W | Mask high speed testing pins (test1out, test2out, fav4)<br>0: normal<br>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<br>0: disable (default)<br>1: enable |
| 1:0 | R/W | Decrease ratio for adaptive tracking<br>Adaptive tracking will be enabled while getting Q (PE) <=2 for over 8 times, and it will be triggered only under delay-chain mode<br>00: 1/2<br>01: 1/4<br>10: 1/8 (default)<br>11: 1/16 |

*Address: B8*　　　**DDS_MIX_1**　　　　　　　　　　　　　　　　　**Default: 06h**

| Bit | Mode | Function |
|---|---|---|
| 7:6 | R | DDS tracking state [1:0]<br>00: not lock<br>01: lock<br>10: unlock but not using new tracking mode yet<br>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<br>000: 2<br>001: 4<br>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 |

| Bit | Mode | Function |
|---|---|---|
| 7:0 | R/W | **P_code_max[16:9]** |
| | | **Set p_code_max value to clamp the GAIN of APLL** |

| Bit | Mode | Function |
|---|---|---|
| 7:0 | R/W | **P_code_max[8:1]** |
| | | **Set p_code_max value to clamp the GAIN of APLL** |

| 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)**<br>**00: cnt=7 => 59.6ns**<br>**01: cnt=15 => 117ns**<br>**10: cnt=23 => 184.4ns (default)**<br>**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)**<br>**00: every 1 cycle sample**<br>**01: every 2 cycle sample**<br>**10: every 3 cycle sample (default)**<br>**11: every 4 cycle sample** |
| 3 | R/W | **Delay chain reset period select**<br>**0: short reset (2ns) (default)**<br>**1: long reset (1 fbck)** |
| 2 | R/W | **Reset delay chain saturation flag**<br>**0: normal (default)**<br>**1: reset flag** |
| 1 | R | **Delay chain saturation flag**<br>**0: not saturate**<br>**1: saturate => it need to enlarge the sample period or set bigger N code** |
| 0 | R/W | **APLL_free_run enable**<br>**0: normal state (default)**<br>**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    (Page 1-CRA6[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**<br>**000: 600 (default)**<br>**001: 850**<br>**010: 1100**<br>**011: 1350**<br>**100: 1600**<br>**101: 1850**<br>**110: 2100**<br>**111: 2350** |
| 0 | R/W | **new_mode_i_code_en**<br>**0: while new mode enable, I code will have no effect on SUM_I. All phase error will be compensated by P code (default)**<br>**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**<br>0: N divider enable.<br>1: N divider disable, OUT=ckxtal. | |
| DPLL_O[1:0] | 5:4 | R/W | 1 | **DPLL Output Divider**<br>00: Div1<br>01: Div2 (Default)<br>10: Div4<br>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**<br>000: 16K<br>001: 18K<br>010: 20K<br>011: 22K (Default)<br>100: 24K<br>101: 26K<br>110: 28K<br>111: 30K | |
| DPLL_CS[1:0] | 4:3 | R/W | 2 | **DPLL Loop Filter Capacitor Control**<br>00: 18p<br>01: 20p<br>10: 24p (Default)<br>11: 28p | |
| DPLL_IP[2:0] | 2:0 | R/W | 2 | **DPLL Charger Pump Current   Control**<br>Icp=(2.5uA+2.5uA*bit[0]+5uA*bit[1]+10uA*bit[2])<br>Keep DPM/Icp constant=10.67 | |

## DCLK Spread Spectrum (Page 1)

| Register::DPLL_WD | | | | 0xC2 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| DPLL_WDO | 7 | R | 0 | **DPLL WD Status**<br>0: Normal<br>1: Abnormal | |
| DPLL_WDRST | 6 | R/W | 0 | **DPLL WD Reset**<br>0: Normal (Default)<br>1: Reset | |
| DPLL_WDSET | 5 | R/W | 0 | **DPLL WD Set**<br>0: Normal (Default)<br>1: Set | |
| DPLL_FUPDN | 4 | R/W | 1 | **DPLL Freqency Tuning**<br>0: Freq Up<br>1: Freq Dn(Default) | |
| DPLL_STOP | 3 | R/W | 1 | **DPLL Freqency Tuning**<br>0: Disable<br>1: Enable (Default) | |
| DPLL_FREEZE | 2 | R/W | 0 | **DPLL Output Freeze**<br>0: Normal (Default)<br>1: Freeze<br>Active high. | |
| DPLL_VCORSTB | 1 | R/W | 0 | **Reset VCO**<br>0: Normal (Default)<br>1: Reset<br>Active high. | |
| DPLL_PWDN | 0 | R/W | 1 | **Power Down DPLL**<br>0: Power on<br>1: Power down(Default)<br>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**<br>00: VCO slowest<br>11: VCO fastest (Default) | |

| | | | | | | |
|---|---|---|---|---|---|---|
| DPLL_CALBP | 5 | R/W | 0 | **DPLL Bypass Calibration**<br>0: Reference by Calibration result(Default)<br>1: Reference by CRC3[7:6]<br>Active high. | |
| DPLL_CALSW | 4 | R/W | 0 | **Calibration Validated**<br>Go high after power on 1200us.<br>0: Reference by CRC3[7:6]<br>1: Refernect by cal result | |
| DPLL_CALLCH | 3 | R/W | 0 | **Latch Calibration**<br>Go high after power on 1100us.<br>0: Disable Latch<br>1: Enable Latch | |
| DPLL_CMPEN | 2 | R/W | 0 | **CMP Enable**<br>Go high after power on 1000us.<br>0: Diable CMPEN<br>1: Enable CMPEN | |
| DPLL_CP | 1 | R/W | 0 | **CP Control**<br>0: 1.77pF`<br>1: 2.1pF | |
| DPLL_RESERVE | 0 | R/W | 1 | **Reserved for DPLL Phase Swallow Circuit**<br>0: Path0<br>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**<br>0 : disable<br>1: enable (auto clear when finish)<br>It should work on DDS Spread Spectrum Output function enable.<br>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**<br>0: Disable (Default) | | |

| Name | Bits | R/W | Default | Comments | Config |
|---|---|---|---|---|---|
| | | | | 1: Enable | |
| DPLL_EVEN_OD_SEL | 5 | R/W | 0 | **Even / Odd Field Select**<br>0: Even (Default)<br>1: Odd | |
| DPLL_FUPDN | 4 | R/W | 1 | **DPLLFUPDN (DPLL Frequency Tuning)**<br>0: Freq Up<br>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%)<br>The bigger setting, the spreading range will bigger, but not uniform | |
| DCLK_FMDIV | 3 | R/W | 0 | **Spread Spectrum FMDIV** (SSP_FMDIV)//(0)<br>0: 33K<br>1: 66K | |
| DCLK_READY | 2 | R/W | 0 | **Spread Spectrum Setting Ready for Writing** (Auto Clear)<br>0: Not ready<br>1: Ready to write | |
| FREQ_SYNTHESIS_SEL | 1:0 | R/W | 0 | **Frequency Synthesis Select** (F & F-N*dF)<br>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** |
| RSV_CA_76 | 7:6 | -- | 0 | **Reserved to 0** | |
| MEASURE_PHASE | 5 | R/W | 0 | **Measure the Phase about Fixed DVTOTAL & Last Line DHTOTAL Function** <br> 0 : Disable <br> 1 : Enable (Auto clear when finish) | |
| MARK_PHASE_TRACKING | 4 | R/W | 0 | **Mark Phase tracking about Fixed DVTOTAL & Last Line DHTOTAL Function** <br> 0 : Disable <br> 1 : Enable | |
| NED_FIXED_LAST_LINE_MODE | 3 | R/W | 0 | **Enable New Design Function in Fixed Last Line Mode** <br> 0: Disable (Default) <br> 1: Enable | |
| DCLK_DDS | 2 | R/W | 0 | **DDS Spread Spectrum Test Enable** <br> 0: Disable (Default) <br> 1: Enable | |
| DCLK_FIXED_LAST_LINE_EN | 1 | R/W | 0 | **Enable the Fixed DVTOTAL & Last Line DHTOTAL Function** <br> 0: Disable (Default) <br> 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 RTD2472RD, 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] | |

# MCLK Spread Spectrum (Page 1)

| Register:: MCLK_FINE_TUNE_OFFSET_MSB | | | | 0xCD | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |

| Name | Bits | R/W | Default | Comments | Config |
|---|---|---|---|---|---|
| RSV_FA_74 | 7:4 | --- | 0 | **Reserved** | |
| MCLK_OFFSET[11:8] | 3:0 | R/W | 0 | **MCLK Offset** [11:8] | |

| Register:: MCLK_FINE_TUNE_OFFSET_LSB | | | | 0xCE | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| MCLK_OFFSET[7:0] | 7:0 | R/W | 0 | **MCLK Offset** [7:0] | |

| Register:: MCLK_SPREAD_SPECTRUM | | | | 0xCF | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| MCLK_SPREAD_RANGE | 7:4 | R/W | 3 | **MCLK Spreading range** (0.0~7.5%) The bigger setting, the spreading range will bigger, but not uniform | |
| MCLK_FMDIV | 3 | R/W | 0 | **Spread Spectrum FMDIV** (SSP_FMDIV)//(0) 0: 33K 1: 66K | |
| MCLK_READY | 2 | R/W | 0 | **Spread Spectrum Setting Ready for Writing** (Auto Clear) 0: Not ready 1: Ready to write | |
| MCLK_DDS | 1 | R/W | 0 | **DDS Spread Spectrum Test Enable** 0: Disable (Default) 1: Enable | |
| MCLK_DDS_EN | 0 | R/W | 0 | **Enable DDS Spread Spectrum Output Function** 0: Disable (Default) 1: Enable | |

● The "Spread Spectrum Setting Ready for Writing" means 3 kinds of registers will be set after this bit is set:
1. MCLK spreading range
2. Spread spectrum FMDIV
3. MCLK offset setting

| 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_75 | 7:5 | --- | 0 | Reserved | |
| TARGET_DPLLUPDB | 4 | R/W | 1 | **Target DPLLUPDN (DPLL Frequency Tuning Up/Down)**<br>0: Freq Up<br>1: Freq Down (Default) | |
| 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 |
| RSV_D4_74 | 7:4 | --- | 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** | |

## 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 | 69 | **M2PLL DPM value – 2 (M)**<br><br>* PLL output=input*(M/P) | |

| Register::M2PLL_N | | | | 0xE1 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| M2PLL_CP | 7 | R/W | 0 | **CP Control**<br>0:CP=1.77pF<br>1:CP=2.1pF | |
| M2PLL_BPN | 6 | R/W | 0 | **M2PLLBPN=0 , N divder enable**<br>**M2PLLBPN=1, N divder disable , OUT=ckxtal** | |
| M2PLL_O[1:0] | 5:4 | R/W | 1 | **M2PLL Output divider**<br>00:Div1, 01:Div2, 10:Div4, 11:Div8 | |
| M2PLL_N[3:0] | 3:0 | R/W | 3 | **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)**<br>000:16K, 001:18K, 010:20K, 011:22K<br>100: 24K, 101: 26K, 110:28K, 111:30K | |
| M2PLL_CS[1:0] | 4:3 | R/W | 2 | **M2PLL Loop Filter Capacitor Control(Cs)**<br>00:18p, 01:20p, 10:24p, 11:28p | |
| M2PLL_IP[1:0] | 2:0 | R/W | 2 | **M2PLL Charge Pump Current Control**<br>Icp=(2.5uA+2.5uA*bit[0]+5uA*bit[1]+10uA*bit[2])<br>Keep DPM/Icp constant=10.67 | |

| Register::M2PLL_WD | | | | 0xE5 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| M2PLL_WDO | 7 | R | 0 | **M2PLL WD Status register**<br>0:Normal<br>1:Abnormal | |
| M2PLL_WDRST | 6 | R/W | 0 | **M2PLL WD Reset**<br>0:Normal | |

| | | | | 1:Reset | |
|---|---|---|---|---|---|
| M2PLL_WDSET | 5 | R/W | 0 | **M2PLL WD Set** <br> 0:Normal <br> 1:Set | |
| M2PLL_VCOMD[1:0] | 4:3 | R/W | 3 | **M2PLL VCO Default mode** <br> 00: VCO slowest <br> 11: VCO fastest | |
| M2PLL_FREEZE | 2 | R/W | 0 | **M2PLL Output Freeze** <br> 0:Normal <br> 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)** | |

## AUDIO DAC (Page 1)

| Register:: BB_POWER0 | | | 0xF0 | | |
|---|---|---|---|---|---|
| **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 | 1 | 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 reference 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 | 1 | Power down control for HPOUT amplifier (0:power down, 1:power on) | |

| Register:: BB_POWER1 | | | 0xF1 | | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |

| Reserved | 7:2 | -- | 0 | Reserved | |
|----------|-----|-----|---|----------|--|
| BB_POW_MBIAS | 1 | R/W | 1 | Power down control for bias generator (0:power down, 1:power on) | |
| BB_POW_VREF | 0 | R/W | 1 | Power down control for analog ground generator (0:power down, 1:power on) | |

| Register:: AIN_CONTROL 0xF2 | | | | | |
|---------------------------|-----|-----|--------|----------|--------|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| Reserved | 7 | -- | 0 | Reserved | |
| BB_EN_AIN | 6 | R/W | 0 | Enable AIN (0:disable, 1:enable) | |
| BB_AINVOL | 5:0 | R/W | 0x27 | Volume control for AIN (00h:-58.5dB~37h:24dB) | |

| Register:: DAC_CONTROL 0xF3 | | | | | |
|----------------------------|-----|-----|--------|----------|--------|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| Reserved | 7:6 | -- | 0 | Reserved | |
| BB_DACVOL | 5:0 | R/W | 0x27 | Volume control for DAC output (00h:-58.5dB~37h:24dB) | |

| Register:: AOUT_CONTROL 0xF4 | | | | | |
|-----------------------------|-----|-----|--------|----------|--------|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| BB_MUTE_AOUT_L | 7 | R/W | 1 | Mute control for AOUT_L (0:unmute, 1:mute) | |
| BB_MUTE_AOUT_R | 6 | R/W | 1 | Mute control for AOUT_R (0:unmute, 1:mute) | |
| 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:disable, 1:enable) | |
| BB_SWAP_AOUT | 3 | R/W | 0 | Swap L/R control for AOUT (0:No swap, 1: Swap L/R) | |
| Reserved | 2:0 | -- | 0 | Reserved | |

| Register:: HPOUT_CONTROL 0xF5 |
|-------------------------------|

| Name | Bits | R/W | Default | Comments | Config |
|---|---|---|---|---|---|
| BB_MUTE_HPOUT_L | 7 | R/W | 1 | Mute control for HPOUT_L (0:unmute, 1:mute) | |
| BB_MUTE_HPOUT_R | 6 | R/W | 1 | Mute control for HPOUT_R (0:unmute, 1:mute) | |
| BB_MUX_HPOUT | 5 | R/W | 0 | Source selection for HPOUT (0:from DAC, 1:from AIN) | |
| BB_OUTEN_HPOUT | 4 | R/W | 0 | Output enable for HPOUT (0:disable, 1:enable) | |
| BB_SWAP_HPOUT | 3 | R/W | 0 | Swap L/R control for HPOUT (0:No swap, 1: Swap L/R) | |
| Reserved | 2:0 | -- | 0 | Reserved | |

| Register:: MBIAS_CONTROL0 0xF6 | | | | | |
|---|---|---|---|---|---|
| Name | Bits | R/W | Default | Comments | Config |
| BB_MBIAS_AMP | 7:6 | R/W | 10 | Bias current selection for output amplifier (00b:10u, 01b:15u, 10b:20u, 11b:30u) | |
| BB_MBIAS_DACVREF | 5:4 | R/W | 10 | Bias current selection for DACVREF (00b:10u, 01b:15u, 10b:20u, 11b:30u) | |
| BB_MBIAS_DAOP | 3:2 | R/W | 10 | Bias current selection for DAOP (00b:10u, 01b:15u, 10b:20u, 11b:30u) | |
| BB_MBIAS_DAREFBUF | 1:0 | R/W | 10 | Bias current selection for DAREFBUF (00b:10u, 01b:15u, 10b:20u, 11b:30u) | |

| Register:: MBIAS_CONTROL1 0xF7 | | | | | |
|---|---|---|---|---|---|
| Name | Bits | R/W | Default | Comments | Config |
| BB_MBIAS_DF2SE | 7:6 | R/W | 10 | Bias current selection for DF2SE (00b:10u, 01b:15u, 10b:20u, 11b:30u) | |
| BB_MBIAS_IN_MC3 | 5:4 | R/W | 10 | Bias current selection for input buffer (00b:10u, 01b:15u, 10b:20u, 11b:30u) | |
| BB_MBIAS_VOL | 3:2 | R/W | 10 | Bias current selection for volume control (00b:10u, 01b:15u, 10b:20u, 11b:30u) | |
| BB_MBIAS_VREF | 1:0 | R/W | 10 | Bias current selection for analog ground generator (00b:10u, 01b:15u, 10b:20u, 11b:30u) | |

| Register:: VREF_CONTROL 0xF8 |
|---|

| Name | Bits | R/W | Default | Comments | Config |
|------|------|-----|---------|----------|--------|
| BB_VREF_VAG | 7:6 | R/W | 01 | 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) | |
| Reserved | 4:0 | -- | 0 | Reserved | |

| Register:: MODULATOR_CONTROL          0xF9 | | | | | |
|------|------|-----|---------|----------|--------|
| **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** | |
| **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** | |

| Register:: BIST_CONTROL                  0xFA | | | | | |
|------|------|-----|---------|----------|--------|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| BB_BIST_MODE | 7 | R/W | 0 | 0: Disable 1: Enable | |
| BB_BIST_RST_N | 6 | R/W | 1 | 0: Reset 1: No Reset | |
| **BB_BIST_DONE** | **5** | **R** | **0** | **0: BIST is running** **1: BIST done** | |
| BB_BIST_FAIL | 4 | R | 0 | 0: BIST ok 1: BIST fail | |

| BB_FT_EN | 3 | R/W | 0 | For FT and test performance。<br>1:Iinput PCM data from test pin in.<br>0:PCM data from digital circuit. | |
| BB_48PIN_MODE | 2 | R/W | 0 | 1:16bits TEST IN (PCM DATA).<br>0:22bits TEST IN. (PCM DATA) | BB_48PIN_MODE |
| Reserved | 1:0 | -- | 0 | Reserved | |
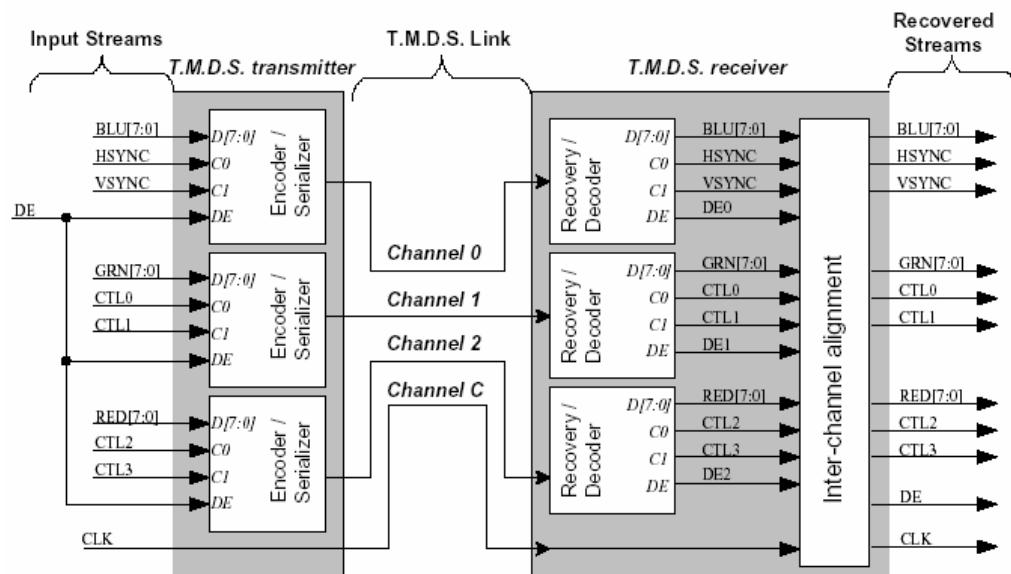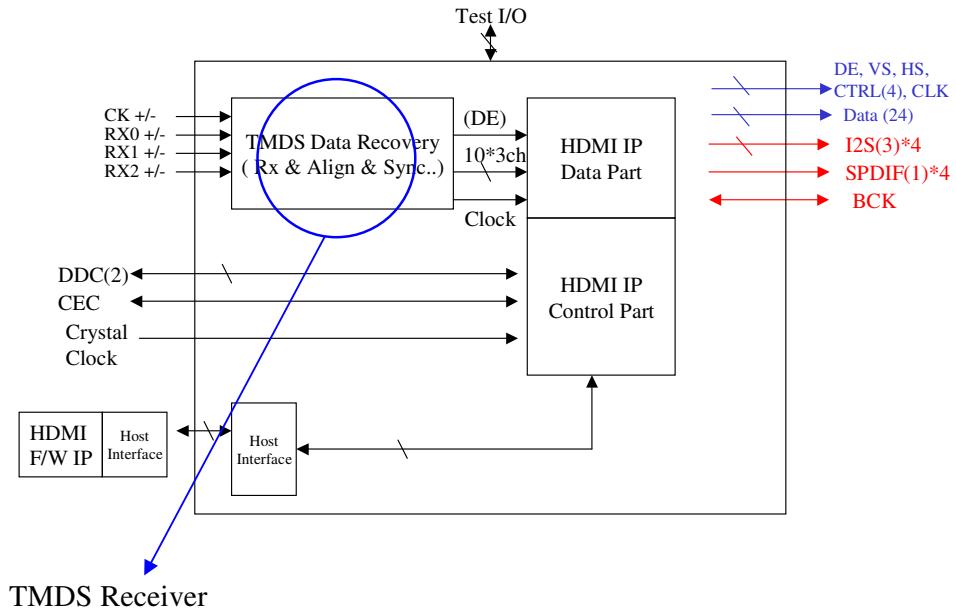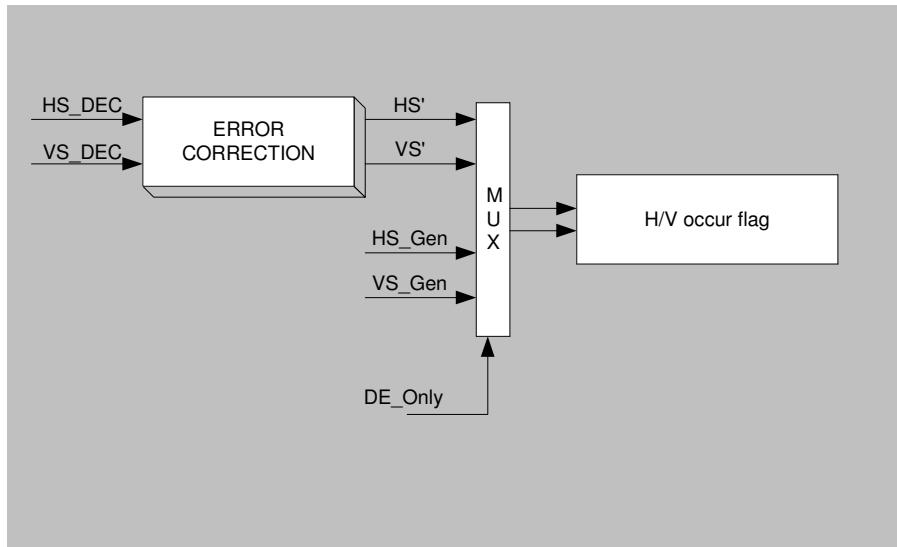
# Overall HDMI System Function Block (Page 2)

Test I/O

CK +/-
RX0 +/-
RX1 +/-
RX2 +/-

TMDS Data Recovery
( Rx & Align & Sync..)

(DE)

10*3ch

Clock

HDMI IP
Data Part

HDMI IP
Control Part

DE, VS, HS,
CTRL(4), CLK
Data (24)
I2S(3)*4
SPDIF(1)*4
BCK

DDC(2)
CEC
Crystal
Clock

HDMI
F/W IP

Host
Interface

Host
Interface

TMDS Receiver

Input Streams

T.M.D.S. transmitter

T.M.D.S. Link

T.M.D.S. receiver

Recovered
Streams

BLU[7:0]
HSYNC
VSYNC

D[7:0]
C0
C1
DE

Encoder /
Serializer

DE

GRN[7:0]
CTL0
CTL1

D[7:0]
C0
C1
DE

Encoder /
Serializer

RED[7:0]
CTL2
CTL3

D[7:0]
C0
C1
DE

Encoder /
Serializer

CLK

Channel 0

Channel 1

Channel 2

Channel C

Recovery /
Decoder

D[7:0]
C0
C1
DE

BLU[7:0]
HSYNC
VSYNC
DE0

Recovery /
Decoder

D[7:0]
C0
C1
DE

GRN[7:0]
CTL0
CTL1
DE1

Recovery /
Decoder

D[7:0]
C0
C1
DE

RED[7:0]
CTL2
CTL3
DE2

Inter-channel alignment

BLU[7:0]
HSYNC
VSYNC

GRN[7:0]
CTL0
CTL1

RED[7:0]
CTL2
CTL3

DE

CLK

| Register: TMDS_MSR | | | | 0XA1 |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| TMM | 7 | R/W | 0 | **Transition measurement method**<br>0: Measure the number of transition for N-clock duration (TMDS_NCP[3:0])<br>1: Measure the number of transition smaller than 16/64 clock period (TMDS_CTC) for 1-frame duration |
| MT | 6:4 | R/W | 0 | **Measure times(exponential of 2)**<br>000: 1<br>001: 2<br>010: 4<br> 011: 8<br>100: 16<br>101: Not available<br>110: Not available<br>111: Not available<br>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)**<br>0000: 16<br>0001: 1<br>0010: 2<br>0011: 3<br>…….<br>1111: 15<br>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**<br> 0: Stop measure, Cleared after finish (Default)<br> 1: Start measure |
| MRS | 6:5 | R/W | 0 | **Measure Result Select**<br> 00: AVE Value (Default) |

| | | | | 01: Max Value |
|---|---|---|---|---|
| | | | | 10: Min Value |
| MS | 4:3 | R/W | 0 | **Measure Select**<br>00: Measure Hsync transition times before error correction.<br>01: Measure Hsync transition times after error correction.<br>10: Measure Data Enable transition times before error correction.<br>11: Measure Data Enable transition times after error correction. |
| Reserved | 2:1 | --- | | **Reserved** |
| CTC | 0 | R/W | 0 | **Criterion of Transition Count**, duration smaller than<br>0: 16 clock<br>1: 64 clock |

| Register: TMDS_MRR1 | | | 0XA3 | |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| Reserved | 7 | --- | 0 | **Reserved** |
| VMR | 6:0 | R | 0 | **Value Of Measure Result**[6:0] (Item refer to MS) |

| Register:: TMDS_CTRL | | | 0XA4 | |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| BCD | 7 | R | x | **B-Channel Detect**　(DE low 128 clock)(write 1 clear)<br>0:no<br>1:yes |
| GCD | 6 | R | x | **G Channel Detect**　(DE low 128 clock)(write 1 clear)<br>0: no<br>1:yes |
| RCD | 5 | R | x | **R Channel Detect**　(DE low 128 clock)(write 1 clear)<br>0: no<br>1: yes |
| HO | 4 | R | x | **Hsync Occur** (write 1 clear)<br>0: no<br>1: yes |
| YO | 3 | R | x | **Vsync Occur** (write 1 clear)<br>0: no<br>1: yes |
| CRCTS | 2:1 | R/W | 0 | **CRC Type Select**<br>00: do CRC only with DE<br>01: do CRC only with DIEN (Data Island Enable)<br>10: do CRC with both DE and DIEN<br>11: reserved |
| CRCC | 0 | R/W | 0 | **CRC Check** |

| Register:: TMDS_CRCOB2 | | | 0XA5 | |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| CRCOB2 | 7:0 | R | -- | 1st read=> Output CRC-48 bit 47~40<br>2nd read=> Output CRC-48 bit 39~32<br>3rd read=> Out put CRC-48 bit 31~24<br>4th read=> Out put CRC-48 bit 23~16<br>5th read=> Out put CRC-48 bit 15~8<br>6th read=> Out put CRC-48 bit 7~0 |

- The read pointer should be reset when 1. CRC Output Byte is written 2. CRC Check starts.
- 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**<br>0: Disable (Default)<br>1: Enable |
| TRCOE | 6 | R/W | 0 | **TMDS R Channel Output Enable**<br>0: Disable (Default)<br>1: Enable |
| TGCOE | 5: | R/W | 0 | **TMDS G Channel Output Enable**<br>0: Disable (Default)<br>1: Enable |
| TBCOE | 4 | R/W | 0 | **TMDS B Channel Output Enable**<br>0: Disable (Default)<br>1: Enable |
| OCKE | 3 | R/W | 0 | **OCLK Enable**<br>0: Disable (Default)<br>1: Enable |
| OCKIE | 2 | R/W | 0 | **OCLK Invert Enable**<br>0: Normal (Default)<br>1: Enable |
| Reserved | 1 | R/W | 0 | **Reserved** |
| CLK25XINV | 0 | R/W | 0 | **Input 1x Clock Invert**<br>0: No Invert (Default)<br>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**<br>0: Disable (Default)<br>1: Enable |
| BRCW | 6 | R/W | 0 | **B/R channel swap**<br>0: No swap (Default)<br>1: Swap |
| PNSW | 5 | R/W | 0 | **P/N Swap**<br>0:No swap(Default)<br>1:swap |
| ICCAF | 4 | R/W | 0 | **Input Channel control by auto function**<br>0: Manual<br>1: Auto (Default) |
| ECC | 3 | R/W | 0 | **Enable Clock channel: turn on clock channel PLL (For manual use)**<br>0: Disable (Default)<br>1: Enable |
| ERIP | 2 | R/W | 0 | **Enable Red input port (For manual use, cut off 50ohm internal resistor)**<br>0: Disable (Default)<br>1: Enable |
| EGIP | 1 | R/W | 0 | **Enable Green input port (For manual use, cut off 50ohm internal resistor)**<br>0: Disable (Default)<br>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** |
| Reserved | 7:0 | -- | -- | **Reserved** |

| Register:: TMDS_ACC1 | | | 0XA9 | |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| Reserved | 7:0 | -- | -- | **Reserved** |

| Register:: TMDS_ABC | | | 0xAA | |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| Reserved | 7:0 | R/W | -- | **Reserved to 0** |

| Register:: TMDS_ACC2 | | | 0XAB | |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| Reserved | 7:0 | R/W | -- | **Reserved to 0** |

| Register:: TMDS_Z0CC2 | | | 0xAC | |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| DDCDBNC | 7 | R/W | 1 | **HDCP DDC DEBOUNCE**<br>**0: Disable**<br>**1: Enable** |
| HDE | 6 | R/W | 0 | **HDMI/DVI function enable** (HDCP enable is moved to HDCP )<br> 0: Disable, gated clock and cut off TMDS pull up resistor for saving power.<br> 1: Enable, |
| Reserved | 5:0 | 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)<br>0: disable<br>1: enable |
| RESERVED | 6:0 | -- | 0 | Reserved. |

| Register:: TMDS_RPS | | | 0XAE | |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| Reserved | 7:0 | -- | 0 | **Reserved to 0** |

| Register:: TMDS_WDC | | | 0xAF | |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |

| Reserved | 7:0 | -- | 0 | **Reserved to 0** |
|---|---|---|---|---|

Register 0xB0~0xB3 are reserved.

| 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<br>0: disable<br>1: enable |
| reserved | 6:3 | -- | 0 | Reserved. |
| CPTEST | 2 | R | 0 | **CPTEST**<br>0: normal mode, in which clock and data from analog are used.<br>1: select TSTCKIN/TSTDIN as input 2X5 clock and data respectively, for TESTING. |
| HMTM | 1:0 | R/W | 0 | **HDCP MP TESTING MODE**<br>Force CTL[3:0] always equal to<br>00:Original<br>01:CTRL=1001<br>10:CTRL=1000<br>11:CTRL=0000 |

| Register:: TMDS_UDC1 | | | | 0xB6 |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| no_clk_in | 7 | R | 0 | No clock input.<br>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 | **ERRC_SEL<1:0>**<br>000: original signal<br>001: 1 cycle debouncing<br>010: 1+8 cycle debouncing<br>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 | **DEBUG_SEL** |

| **Register:: TMDS_DPC1** | | | | **0XB8** |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| reserved | 7:4 | -- | 0 | Reserved. |
| dpc_clk_sou rce | 3 | R/W | 0 | Select the reference clock of deep color pll<br><br>0: recovered tmds clock<br><br>1: original tmds clock |
| dpc_auto | 2 | R/W | 1 | 0: manual mode (CD/PP/default_ phase fields are specified by FW)<br><br>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.<br><br>(If dpc_auto(0xB8[2]) ==0, this bit is R/W; otherwise, it is read-only) |
| dpc_pp_vali d | 0 | R/W | 0 | Phase valid of HDMI 1.3 Deep color mode.<br><br>(If dpc_auto(0xB8[2]) ==0, this bit is R/W; otherwise, it is read-only) |

# HDCP 1.3 (Page 2)

| ■ Register:: HDCP_CR | | | | 0XC0 |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| Reserve | 7 | R | 0 | **Reserved.** |
| Reserve | 6 | R | 0 | **Reserved.** |
| IVSP | 5 | R | 0 | **Indicate VSYNC Polarity**<br>0: Positive, which means VS pulse is high.<br>1: Negative |
| INVVS | 4 | R/W | 0 | **Invert VSYNC for HDCP**<br>High: Inverted<br>Low: Not Inverted |
| IVSPM | 3 | R/W | 0 | **Indicate VSYNC Polarity Mode:**<br>High: manual, decided by INVVS<br>Low: auto, indicate by IVSP |
| MADDF | 2 | R/W | 0 | **MCU Access DDC data first**<br>0: enable DDC channel and MCU access only when DDC is not busy<br>1: disable DDC channel and MCU access only |
| DKAPDE | 1 | R/W | 0 | **Device Key Access Port download enable**<br>High: enable<br>Low: disable, this would reset the address of Device Key Access Port to 0. |
| Enable | 0 | R/W | 0 | **HDCP Enable**<br>High: Auto Enable HDCP function, when Tx I2C write Aksv,<br>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 $7^{th}$, $15^{th}$, 23rd, 31st, $39^{th}$, $47^{th}$, $55^{th}$ bits inserted.<br>The inserted data are '0'.And the write sequence is:<br>{D0-Byte0, D0-Byte1, D0-Byte2, D0-Byte3,D0-Byte4, D0-Byte5, D0-Byte6, D0-Byte7},<br>{D1-Byte0, D1-Byte1, 1-Byte2,D1-Byte3,<br>D1-Byte4, D1-Byte5, D1-Byte6, D1-Byte7},<br>………………………………………<br>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**<br>**0: non active**<br>**1: active** |
| DDCSEL | 2:0 | R/W | 0 | **DDC Channel SEL for Key Access**<br>00: DDCSCL1/DDCSDA1<br>01: DDCSCL2/DDCSDA2<br>1x: Reserved. |
| APAI | 0 | R/W | 0 | **HDCP Accessing Port Auto Increase** (For Host Side) |

| | | | | 0: auto increase<br>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** |

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.

**Following register is assigned by "HDCP-address port", "HDCP-data port"**

| Register:: HDCP_FCR | | | | 0xC0 |
| --- | --- | --- | --- | --- |
| Name | Bits | R/W | Reset State | Comments |
| Reserved | 7 | R | -- | **Reserved** |
| FC | 6:0 | R | 0 | **HDCP_frame counter**[6:0] |

| Register:: HDCP_SIR | | | | 0xC1 |
| --- | --- | --- | --- | --- |
| 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** |
| 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].**<br>Since Ainfo[1] in DDC port is 0 at most of time, we need to know what Tx wrote. |
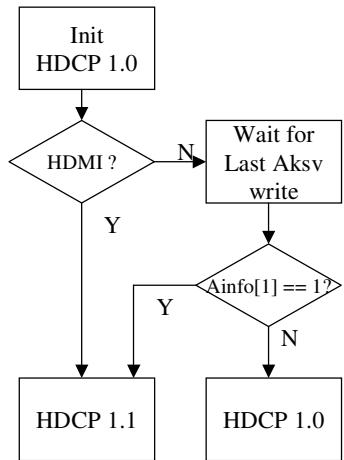
HDCP 1.1/1.0 decide flow.

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.
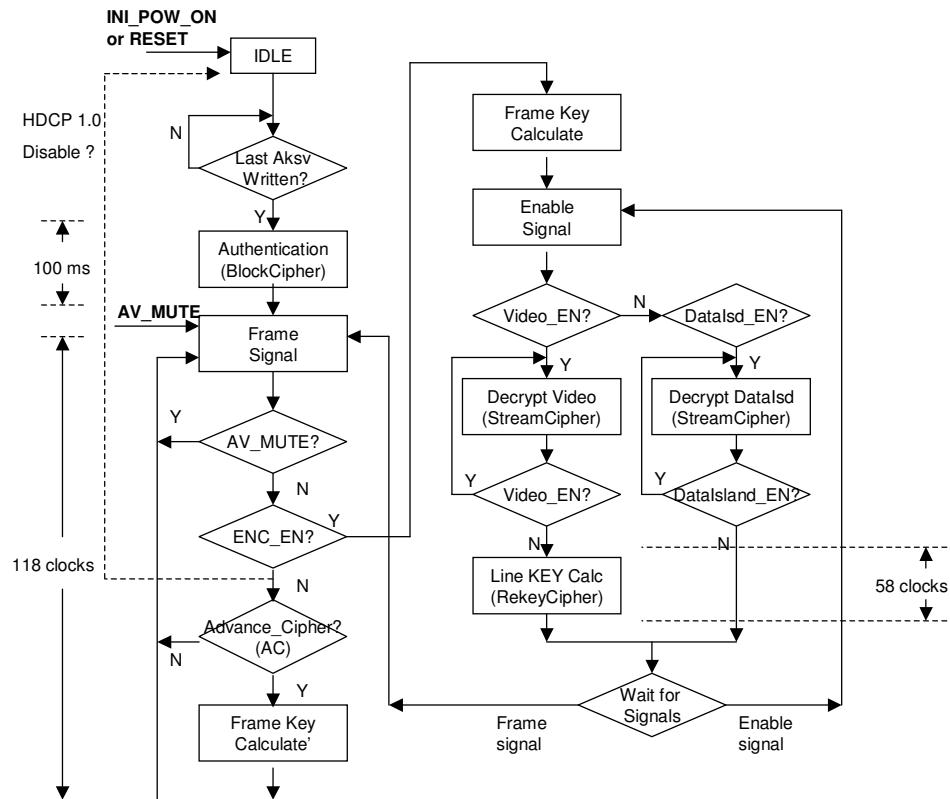
OESS is the same as HDCP 1.0. We could tell it by Ainfo[1] in DDC.

## HDCP 1.0/1.1 decide flow
### (Before Auth)

```
          ┌──────────┐
          │   Init   │
          │ HDCP 1.0 │
          └────┬─────┘
               │
           ◇───┴───◇   N    ┌──────────┐
          ◇  HDMI ?  ◇─────▶│ Wait for │
           ◇───┬───◇        │ Last Aksv│
               │            │  write   │
             Y │            └────┬─────┘
               │                 │
               │            ◇─────┴──────◇
               │       Y   ◇ Ainfo[1] == 1? ◇
               │    ┌──────◇──────┬─────◇
               │    │            N │
               ▼    ▼              ▼
          ┌──────────┐      ┌──────────┐
          │ HDCP 1.1 │      │ HDCP 1.0 │
          └──────────┘      └──────────┘
```

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. |
| 12 | | | |
| 13 | | | |
| 14 | | | |

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.
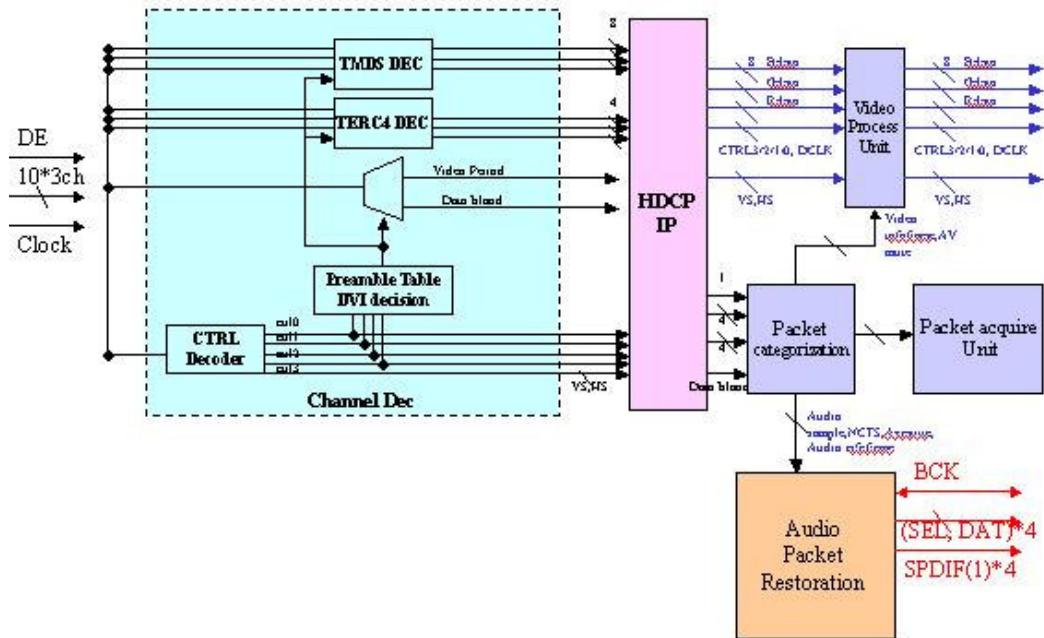
                           & b. AC = 1 or ENC_EN = 1.

NOTE :
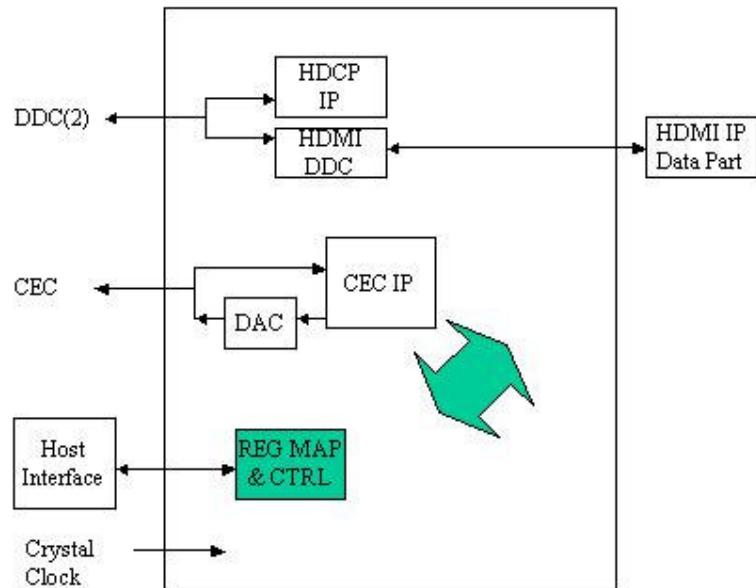
1. HDCP output must be always enable for DVI/HDMI.
2. The sub-descriptions i of Ri & j of Pj are the same.

# HDMI Video and Audio Part (Page 2)
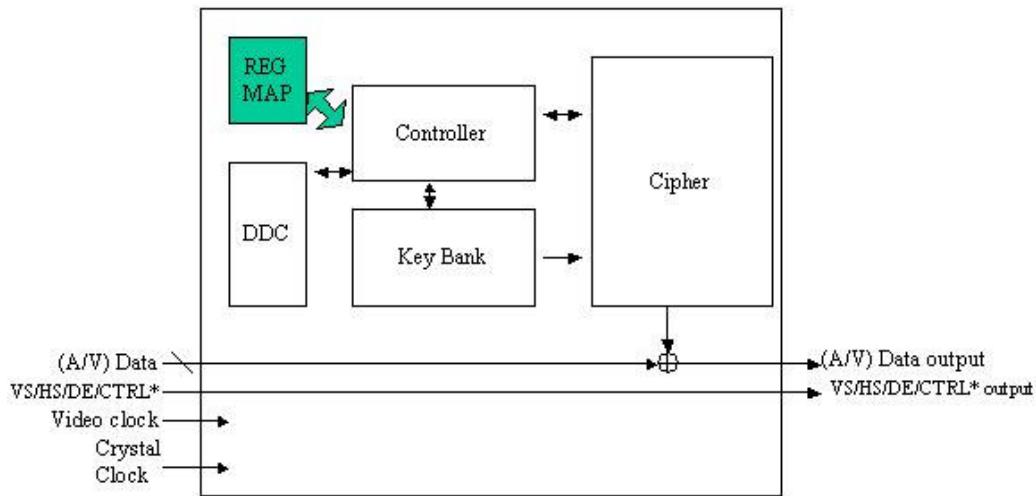
*HDMI IP Data Part*



*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<br>0: If read/write "HDMI data port" continuously without assign "HDMI address port", address would be not added by one automatically.<br>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 |
| 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 |
| 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 |

| Register:: HDMI_SR | | | | 0xCB |
|--------------------|------|-----|---------------|----------|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| Reserved | 7 | --- | 0 | **Reserved** |
| AVMUTE | 6 | R | 0 | **AV_MUTE flag of General Control Packet**<br>0: If HW receive Clear _AVMUTE flag of General Control Packet ,this bit shall assign to 0 until HW receive Set _AVMUTE<br>1: If HW receive Set_AVMUTE flag of General Control Packet ,this bit shall assign to 1 until HW receive Clear_AVMUTE<br>**Note :**<br>If HW never receives "General Control Packet", this bit shall set to 0.<br>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**<br>0: LPCM<br>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.**<br>  (write 1 clear for each bit)<br>1: non-lock<br>0: lock |
| AFIFOOF | 2 | R | 0 | **0: Audio FIFO isn't overflow for X samples**<br>**1: Audio FIFO is overflow for X sample**<br>**(write 1 clear for each bit)**<br>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**<br>**1: Audio FIFO is underflow for Y sample**<br>**(write 1 clear for each bit)**<br>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.**<br>0: DVI<br>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**<br>6: RSV1 received<br>5: RSV0 received |
| PVS | 4:0 | R | 0 | **Packet Variation Status**<br>0: AVI infoframe<br>1: Audio infoframe<br>2: ACP<br>3: ISRC1<br>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 1$^{st}$ 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 | | | 0x00 | |
|------|------|-----|-------------|----------|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| Reserved | 7:4 | -- | 0 | **Reserved to 0** |
| MODE | 3 | R/W | 0 | **HDMI/DVI switch mode**<br>0: Auto detect flow is as fig.1<br>1: Manual |
| MSMODE | 2 | R/W | 0 | **Manual switch HDMI/DVI**<br>0: DVI<br>1: HDMI |
| CABS | 1 | R/W | 0 | **DVI/HDMI condition A, B select**<br>0: condition A: Detect data island preamble + data island guard band (appear count is decided by "N")<br>    condition B: Detect if data island preamble + data island guard band |

| | | | | |
|---|---|---|---|---|
| | | | | appear in continuous 30 or 2 frames(decide by bit 0)<br>1: condition A: Detect data island preamble + data island guard band & video preamble + video guard band(appear count is decided by "N")<br>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)<br>0: 2 frames<br>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
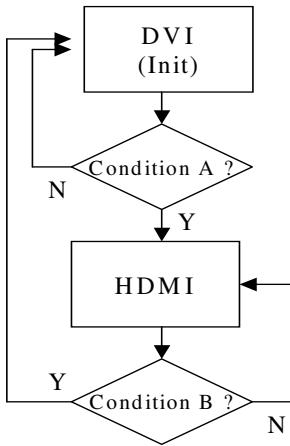


Fig 1

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.
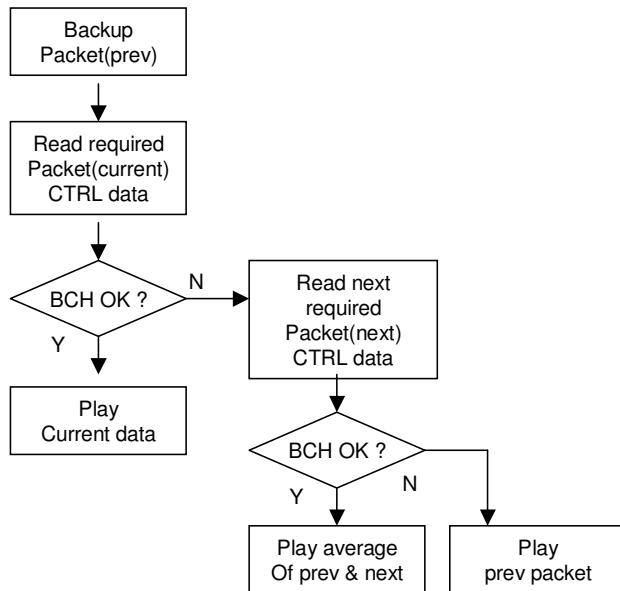
| Register:: HDMI_N_VAL | | | | 0x01 |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| NVAL | 7:0 | R/W | 1 | **N=**<br>00 : X<br>01 : 1<br>FF : 255<br>N = 1 ~ 255 , N can't be assigned to 0x00 |

| Register:: HDMI_BCHCR | | | | 0x02 |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| Reserved | 7:6 | --- | 0 | **Reserved to 0** |
| SPCSS | 5 | R/W | 0 | **SPDIF preamble channel status Source,** When PLL is non-lock<br>0: Input audio sample (normal)<br>1: Internal system |
| ENRWE | 4 | R/W | 0 | **Enable noise reduction when BCH error is greater than one.**<br>1: Enable noise reduction<br>0: Disable noise reduction |
| BCHE | 3 | R/W | 1 | **BCH function enable**<br>1: Enable BCH function<br>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 1.**<br>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 1.** |
| | | | | 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 | | | 0x03 | |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| Reserved | 7 | --- | 0 | **Reserved to 0** |
| 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** |
| | | | | 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 | | | | 0x04 |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| Reserved | 7:6 | --- | 0 | **Reserved** |
| BISTR | 5 | R | 1 | **Audio FIFO BIST Result**<br>0: fail<br>1: success |
| BISTS | 4 | R/W | 0 | **Audio FIFO BIST Start** (embedded test pattern)<br>0: stop<br>1: start(auto clear) |
| AFIFOF | 3 | R | 0 | **Audio FIFO Full (write clear)**<br>0: Indicate FIFO is not full.<br>1: Indicate FIFO is full. |
| AFIFOE | 2 | R | 0 | **Audio FIFO Empty(write clear)**<br>0: Indicate FIFO is not empty.<br>1: Indicate FIFO is empty. |
| Reserved | 1:0 | --- | 0 | **Reserved to 0** |

| Register:: HDMI_MAGCR | | | | 0x05 |
|---|---|---|---|---|
| **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.<br>8'h00 = 0<br>8'hFF = $1 - 2^{-8}$ |
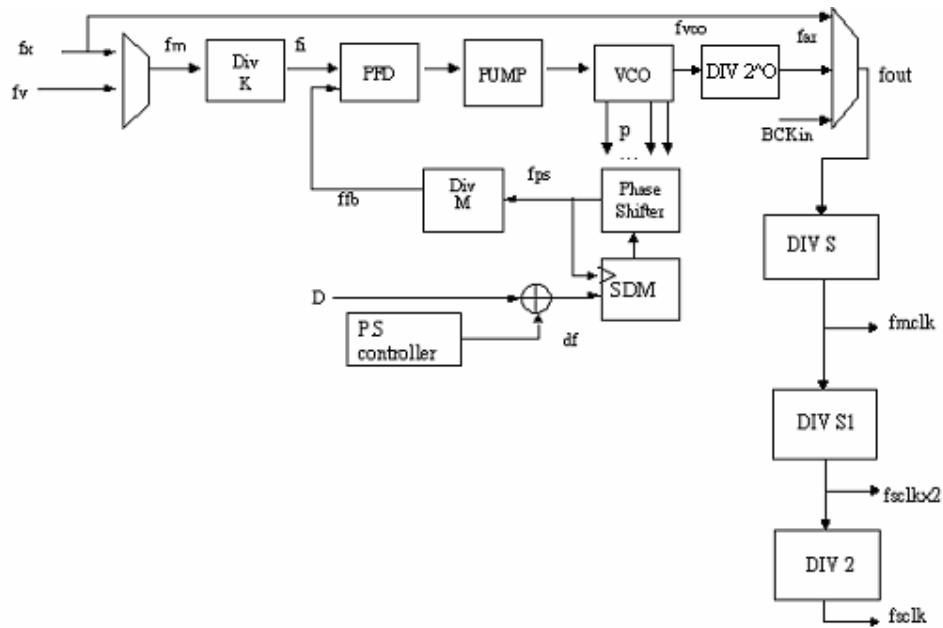
Only valid when "Manual Gain control" is enabled in "manual audio output flow"

| Register:: HDMI_MAGCR | | | | 0x06 |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset** | **Comments** |

| | | | State | |
|---|---|---|---|---|
| Reserved | 7:6 | --- | 0 | **Reserved to 0** |
| AGI | 5:3 | R/W | 4 | **Auto Gain Incremental**<br>000 : $2^{-8}$<br>001 : $2^{-7}$<br>010 : $2^{-6}$<br>…<br>111 : $2^{-1}$ |
| AGD | 2:0 | R/W | 4 | **Auto Gain Delay**<br>000 : $2^0$ sample<br>001 : $2^1$ samples<br>010 : $2^2$ samples<br>…<br>111 : $2^7$ samples<br>The total meanings of this byte are:<br>When function is on, gain increase from 0 to 1 with 'incremental' per 'delay'.<br>When function is off, gain decrease from 1 to 0 with '-inc' per 'delay'.<br>So that the default value means increase $2^{-5}$ per 16 samples. |

Only valid in "auto audio output flow"

## 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 | | | | 0x10 |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| ICMUX | 7 | R/W | 0 | **Input Clock MUX**<br>1: use video clock as input<br>0: use crystal clock as input |
| OCS | 6:5 | R/W | 2 | **Output Clock Select**<br>00: use crystal clock as output clock.<br>01: use BCKin as output clock<br>1X: use generated clock, far, as output clock (must set when power-saving) |
| DBDCB | 4 | R/W | 0 | **Double Buffer Download Control Bit**<br>Enable is also triggered by HW, ref. "Phase error mode".<br>1: write current data to active buffer.<br>0: after write done, this bit would be cleared automatically.<br>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<br>0000: div 2<br>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 | | | | 0x11 |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| MC | 7:0 | R/W | 4E | **M Code**<br>00: div 2<br>FF: div 257 |

| Register:: HDMI_SCAPR | | | | 0x12 |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| SLC | 7 | R/W | 0 | **S1 code**<br>0: div 1<br>1: div 2 |
| SC | 6:0 | R/W | 5 | **S/2 code** |

| Register:: HDMI_DCAPR0 | | | | 0x13 |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| DCAPR | 7:0 | R/W | 0 | **D[15:8]** |

| Register:: HDMI_DCAPR1 | | | | 0x14 |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| DCAPR | 7:0 | R/W | 0 | **D[7:0]** |

| Register:: HDMI_PSCR | | | | 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**<br>000 : xx<br>001 : n=2,don't use this value for normal case<br>010 : n=3<br>011 : n=4<br>100 : n=5<br>101 : n=6<br>110 : n=7<br>111 : n=8 |
| ETCN | 4 | R/W | 0 | **Enable tracking of CTS & N**<br>0: disable.<br>1: enable. |
| ETFD | 3 | R/W | 0 | **Enable tracking of the trend of FIFO depth**<br>0: disable.<br>1: enable. |
| ETFBC | 2 | R/W | 0 | **Enable tracking of FIFO boundary condition**<br>(This bit is suggested to be 1)<br>0: disable.<br>1: enable. |
| PECS | 1:0 | R/W | 1 | **Phase error count source(CTS & N)**<br>00 : phase error counted by video clock<br>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.

| Register:: HDMI_FDDR | | | | 0x16 |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| FDDR | 7:0 | R | 0 | **FIFO depth at DE rising**, this unit is number of samples, |

| Register:: HDMI_FDDF | | | | 0x17 |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| FDDF | 7:0 | R | 0 | **FIFO depth at DE falling** |

| Register:: HDMI_MFDDR | | | | 0x18 |
|---|---|---|---|---|
| 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 | | | | 0x19 |
|---|---|---|---|---|
| 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 | | | | 0x1A |
|---|---|---|---|---|
| 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 | | | | 0x1B |
|---|---|---|---|---|
| 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. |

| Register:: HDMI_ICPSNCR0 | | | | 0x1C |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |

| IC | 7:0 | R/W | 0 | I code of N/CTS [15:8] |
|---|---|---|---|---|

| Register:: HDMI_ICPSNCR1 | | | 0x1D | |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| IC | 7:0 | R/W | 0 | I code of N/CTS [7:0] |

| Register:: HDMI_PCPSNCR0 | | | 0x1E | |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| PC | 7:0 | R/W | 0 | P code of N/CTS [15:8] |

| Register:: HDMI_PCPSNCR1 | | | 0x1F | |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| PC | 7:0 | R/W | 0 | P code of N/CTS [7:0] |

| Register:: HDMI_ICTPSR0 | | | 0x20 | |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| ICT | 7:0 | R/W | 0 | I code of trend [15:8] |

| Register:: HDMI_ICTPSR1 | | | 0x21 | |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| ICT | 7:0 | R/W | 0 | I code of trend [7:0] |

| Register:: HDMI_PCTPSR0 | | | 0x22 | |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| PCT | 7:0 | R/W | 0 | P code of trend [15:8] |

| Register:: HDMI_PCTPSR1 | | | 0x23 | |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| PCT | 7:0 | R/W | 0 | P code of trend [7:0] |

| Register:: HDMI_ICBPSR0 | | | 0x24 | |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| ICB | 7:0 | R/W | 0 | I code of boundary [15:8] |

| Register:: HDMI_ICBPSR1 | | | 0x25 | |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| ICB | 7:0 | R/W | 0 | I code of boundary [7:0] |

| Register:: HDMI_PCBPSR0 | | | 0x26 | |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| PCB | 7:0 | R/W | 0 | P code of boundary [15:8] |

| Register:: HDMI_PCBPSR1 | | | 0x27 | |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| PCB | 7:0 | R/W | 0 | P code of boundary [7:0] |

| Register:: HDMI_NTx1024TR0 | | | | 0x28 |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| Reserved | 7:4 | --- | 0 | **Reserved to 0** |
| RM | 3 | R/W | 0 | **Restart measure**. Measure the length of 1024 Tv by crystal. The result is readable from the following bits.<br>1: enable measure. Writing 1 would clear the answer. This bit would be auto cleared after measure done.<br>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 | | | | 0x29 |
|---|---|---|---|---|
| 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 | | | | 0x2A |
|---|---|---|---|---|
| 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.<br>8'h00: 16 crystal clock.<br>8'h7F: 128 * 16 crystal clock. |

| Register:: HDMI_NCPER | | | | 0x2B |
|---|---|---|---|---|
| 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 | | | | 0x2C |
|---|---|---|---|---|
| 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 | | | | 0x2D |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| CMVTC | 7 | R/W | 0 | **Clear max value ( 18, 19 ) when trend condition is sure.**<br>1: Enable trend to clear max value<br>0: Disable this function |
| CMVBC | 6 | R/W | 0 | **Clear max value (18, 19) when boundary condition is sure.**<br>1: Enable boundary to clear max value<br>0: Disable this function |
| SSDMOU | 5 | R | 0 | **Flag of sum_r of SDM overflow/underflow ( Read only )**<br>1: Overflow or underflow happened<br>0: No overflow, no underflow |
| TEF | 4 | R/W | 0 | **Trend Error Flag**<br>1: Detect up and down at the same time. Clear only when disable SDM ( 2D[1] = 0 )<br>0: Trend is ok. |
| W1C5 | 3 | W | 0 | **Write 1 to clear bit [5]** |

| Name | Bits | R/W | Reset State | Comments |
|------|------|-----|-------------|----------|
| PEM | 2 | R/W | 0 | **Phase Error Mode**, <br> 1: Use delay to calculate, each unit is 0.1 ns. <br> 0: Use clock to calculate, the clock select is at "PE count source". |
| ESDM | 1 | R/W | 0 | **Enable SDM**(phase swallow) <br> 1: Enable <br> 0: Disable, there won't be phase swallow operating in the loop of PLL. |
| Reserved | 0 | --- | 0 | **Reserved** |

| Register:: HDMI_APDMCR | | | 0x2E | |
|------|------|-----|-------------|----------|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| Reserved | 7:6 | --- | 0 | **Reserved** |
| EDM | 5 | R/W | 0 | **Enable Debug Mode** <br> 0: Normal run <br> 1: Enable when test mode |
| PST | 4 | R/W | 0 | **Phase swallow trend** <br> 0: Fast direction <br> 1: Slow direction |
| PSC | 3:0 | R/W | 0 | **Phase Swallow Cycle.** <br> 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 1$^{st}$ 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 | | | 0x30 | |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| Reserved | 7 | -- | 0 | **Reserved** |
| AOC | 6 | R/W | 0 | **Audio output enable/disable control**<br>1: Enable<br>0: Disable<br>If this bit is enabled, audio output signal would be controlled by bit4.<br>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**<br>1: Normal output<br>0: Mute<br>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".<br>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**<br>0: Mute<br>1: Disable |
| VE | 3 | R/W | 0 | **Video Enable**<br>1: Enable video output<br>0: disable video output |
| AMPIC | 2 | R/W | 0 | **Audio Mute Pin Invert Control, execute when mute/disable happens**.<br>0: when event (audio mute or disable) occur, set this pin to low voltage, others maintain high.<br>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**<br>0: when event (video disable) occurred, set this pin to low voltage, others maintain high.<br>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**<br>0: no inverse, which means H : IRQ,    L : no IRQ<br>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 | | | | 0x31 |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| ASMFE | 7 | R/W | 1 | **Auto SET_ AVMUTE function enable**<br>0: If HW receives SET_AVMUTE flag, don't mute/disable audio & disable video by HW.<br>1: If HW receives SET_AVMUTE flag, mute/disable audio & disable video by HW.<br>*Note:*<br>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" |
| Reserved | 6 | --- | 1 | **Reserved to 1** |
| Reserved | 5 | --- | 0 | **Reserved to 0** |
| AWDCT | 4 | R/W | 0 | **Audio watch dog for audio coding type(Decode from SPDIF, code type only include LPCM or Non-LPCM)**<br>0: If coding type is different with previous type, don't mute/disable audio by HW.<br>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**<br>0: If audio PLL is non-lock, don't mute/disable audio by HW.<br>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.**<br>0: If audio FIFO is overflow for X samples , don't mute/disable audio by HW.<br>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.**<br>0: If audio FIFO is underflow for Y samples, don't mute/disable audio by HW.<br>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**<br>0: Channel status bit 1<br>1: Valid bit |

Audio/Video watch dog for "packet acquire mechanism" is listed in Packet acquire mechanism Unit.

| Register:: HDMI_WDCR1 | | | | 0x32 |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| AWDCK | 7 | R/W | 0 | **Audio Watch Dog For TMDS clock**<br>1: If TMDS clock disappears, mute or disable audio.<br>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**<br>1: If layout field is different with previous value, mute or disable audio.<br>0: If layout field is different with previous value, don't mute or disable audio. |
| Rev | 5 | --- | 0 | **Reserved** |
| VWDACT | 4 | R/W | 0 | **Video Watch Dog For Audio Coding Type**<br>1: If coding type is different with previous type, disable video<br>0: If coding type is different with previous type, don't disable video |
| XV | 3:0 | R/W | 0 | **X Value**<br>0000: 1<br>0001: 3<br>~ |

| | | | | 1111: 31 |
|---|---|---|---|---|

**Register:: HDMI_WDCR2** 　　　　　　　　　　0x33

| Name | Bits | R/W | Reset State | Comments |
|---|---|---|---|---|
| VWDAP | 7 | R/W | 0 | **Video Watch dog enable for audio PLL**<br>1: If audio PLL is non-lock, disable video<br>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**<br>1: If layout field is different with previous value, disable Video.<br>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.**<br>1: If audio FIFO is overflow for "X" samples, disable Video.<br>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 .**<br>1: If audio FIFO is underflow for "Y" samples, disable Video.<br>0: If audio FIFO is overflow for "Y" samples, don't disable Video |
| YV | 3:0 | R/W | 0 | **Y value**<br>0000:1<br>0001:3<br>~<br>1111:31 |

**Register:: HDMI_DBCR** 　　　　　　　　　　0x34

| Name | Bits | R/W | Reset State | Comments |
|---|---|---|---|---|
| Reserved | 7:4 | --- | 0 | **Reserved** |
| ALDBFv | 3 | R/W | 0 | **Auto Load Double Buffer** when TMDS clock disappear<br>0: If TMDS clock disappear , don't assign "double buffer download control bit" to 1 by HW<br>1: If TMDS clock disappear, assign "double buffer download control bit" to 1 by HW.<br>*Note:*<br>If this bit is 0, "phase tracking control bits" shall be downloaded to $2^{nd}$ buffer by assigned "double buffer download control bit" to 1.<br>If this bit is 1, "phase tracking control bits" shall be downloaded to $2^{nd}$ 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.**<br>0: If audio FIFO is overflow for X samples, don't assign "double buffer download control bit" to 1 by HW<br>1: If FIFO is overflow for X samples, assign "double buffer download control bit" to 1 by HW.<br>*Note:*<br>If this bit is 0, "phase tracking control bits" shall be downloaded to $2^{nd}$ buffer by assigned "double buffer download control bit" to 1.<br>If this bit is 1, "phase tracking control bits" shall be downloaded to $2^{nd}$ 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.**<br>0: If audio FIFO is underflow for Y samples, don't assign "double buffer download control bit" to 1 by HW<br>1: If FIFO is underflow for Y samples, assign "double buffer download control bit" to 1 by HW.<br>*Note:*<br>If this bit is 0, "phase tracking control bits" shall be downloaded to $2^{nd}$ buffer by assigned "double buffer download control bit" to 1.<br>If this bit is 1, "phase tracking control bits" shall be downloaded to $2^{nd}$ buffer by HW if FIFO is underflow for Y samples. |
| ALDBPN | 0 | R/W | 0 | **Auto Load Double Buffer when PLL non-lock.**<br>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 | | | 0x35 | |
|---|---|---|---|---|
| **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 | | | 0x36 | |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| Reserved | 7 | --- | 0 | **Reserved** |
| PLLTM | 6 | R/W | 0 | **PLL test mode enable** <br> 1: enable <br> 0: disable |
| FPSD | 5 | R/W | 0 | **1st phase shift direction** <br> 0: upwards <br> 1: downwards |
| SPSD | 4 | R/W | 0 | **2nd phase shift direction** <br> 0: upwards <br> 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 | | | 0x38 | |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| DPLLC2 | 7 | R/W | 1 | **DPLLPWDN** <br> 0: power up <br> 1: power down |
| DPLLC1 | 6 | R/W | 0 | **DPLLFREEZE** <br> 0: normal <br> 1: freeze |
| DPLLC0 | 5:4 | R/W | 0 | **DPLLO** <br> div 2^(DPLLO) |
| DPLL_CAL BP | 3 | R/W | 0 | **DPLL bypass calibration(active high)** |
| DPLL_CAL SW | 2 | R/W | 0 | **calibration validated (go high after power on 1200us)** |
| DPLL_CAL LCH | 1 | R/W | 0 | **latch calibration (go high after power on 1100us)** |
| DPLL_CMP EN | 0 | R/W | 0 | **cmp enable (go high after power on 1000us)** |
| | | | | |

| Register:: HDMI_DPCR1 | | | 0x39 | |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset** | **Comments** |

| | | | State | |
|---|---|---|---|---|
| DPLL_RS | 7:5 | R/W | 3 | **DPLL Loop Filter Resister Control**<br>RS: 000:16K   001:18K   010:20K   011:22K<br>     100: 24K    101: 26K 110:28K 111:30K |
| DPLL_CS | 4:3 | R/W | 2 | **DPLL Loop Filter Capacitor Control**<br>CS= 00:18p, 01:20p, 10:24p, 11:28p |
| DPLL_IP | 2:0 | R/W | 2 | **DPLL Charge Pump Current Control**<br>Icp=(2.5uA+2.5uA*bit[0]+5uA*bit[1]+10uA*bit[2])<br>Keep DPM/Icp constant=10.67 |

| Register:: HDMI_DPCR2 | | | | 0x3A |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| DPLLSTATUS | 7 | R | 0 | **DPLLSTATUS(DPLL WD Status)**<br>0:Normal<br>1:Abnormal |
| DPLLWDRST | 6 | R/W | 0 | **DPLLWDRST(DPLL WD Reset)**<br>0:Normal<br>1:Reset |
| DPLLWDSET | 5 | R/W | 0 | **DPLLWDSET(DPLL WD Set)**<br>0:Normal<br>1:Set |
| DPLL_VCOMD | 4:3 | R/W | 3 | **DPLL VCO Default mode**<br>00: VCO slowest, 11: VCO fastest |
| DPLLRESERVE | 2 | R/W | 1 | **DPLLRESERVE, phase swallow circuit clock select**<br>0: fvco, default is 1<br>1: fps |
| DPLLSTOP | 1 | R/W | 1 | **DPLLSTOP(DPLL Frequency Tuning Enable)**<br>0:Disable<br>1:Enable |
| DPLL_CP | 0 | R/W | 0 | **CP Control     0:CP=1.77pF    1:CP=2.1pF** |

| Register:: HDMI_DPCR3 | | | | 0x3B |
|---|---|---|---|---|
| **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**<br> 0: divider K enable<br> 1: divider K disable(K=1) |
| DPLL_RESERVE1 | 1 | R/W | 0 | **DPLL_RESERVE1** |
| DPLLVCORSTB | 0 | R/W | 0 | **RESET VCO    (active high)** |

## Packet Acquire Mechanism

| Register:: HDMI_AWDSR | | | | 0x40 |
|---|---|---|---|---|
| **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 | | | | 0x41 |
|---|---|---|---|---|
| **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 | | | | 0x42 |
|---|---|---|---|---|
| **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 | | | | 0x43 |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| PT | 7:0 | R/W | 0 | **Packet Type of RSV1 packet** |

| Register:: HDMI_PTRSV2 | | | | 0x44 |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| PT | 7:0 | R/W | 0 | **Packet Type of RSV2 packet** |

| Register:: HDMI_PVGCR0 | | | | 0x45 |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| PVSEF | 7:0 | R/W | FF | **Bit7 ~ Bit0 of packet variation status enable flag** |

| Register:: HDMI_PVGCR1 | | | | 0x46 |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| PVSEF | 7:0 | R/W | FF | **Bit15 ~ Bit8 of packet variation status enable flag** |

| Register:: HDMI_PVGCR2 | | | | 0x47 |
|---|---|---|---|---|
| **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 | | | | 0x48 |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |

| PVS | 7:0 | R | 0 | **Bit7 ~ Bit0 of packet variation status** |
|-----|-----|---|---|---|

| **Register:: HDMI_PVSR1** | | | | 0x49 |
|-----|------|-----|-------|----------|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| PVS | 7:0 | R | 0 | **Bit15 ~ Bit8 of packet variation status** |

| **Register:: HDMI_PVSR2** | | | | 0x4A |
|-----|------|-----|-------|----------|
| **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** | | | | 0x50 |
|-----|------|-----|-------|----------|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| EOI | 7 | R/W | 0 | **EVEN/ODD Inverse** <br> 0: Normal |

| | | | | 1: Inverse |
|---|---|---|---|---|
| EOT | 6 | R | 0 | **EVEN/ODD Toggle (write 1 clear)**<br>0: Progressive<br>1: Interlace |
| SE | 5 | R | 0 | **EVEN/ODD signal error (write 1 clear)**<br>0: Normal<br>1: Error |
| RS | 4 | R/W | 0 | **The reference signal for executing Info-frame automatically.**<br>0: DEN<br>1: VSYNC |
| DSC | 3:0 | R/W | 0 | **Down sample control (only valid if Video Down Sampling Auto Mode Disable)**<br>0000: pixel down sample for 1 time( no down sample )<br>0001: pixel down sample for 2 times<br>0010: pixel down sample for 3 times<br>0011: pixel down sample for 4 times<br>0100: pixel down sample for 5 times<br>0101: pixel down sample for 6 times<br>0110: pixel down sample for 7 times<br>0111: pixel down sample for 8 times<br>1000: pixel down sample for 9 times<br>1001: pixel down sample for 10 times<br>others : XXX |

| Register:: HDMI_ACRCR | | | | 0x51 |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| HDIRQ | 7 | R/W | 0 | **HDMI/DVI change interrupt enable**<br>**0:disable**<br>**1:enable** |
| CSAM | 6 | R/W | 1 | **Color Space Translation**<br>0: Manual<br>1: Auto |
| CSC | 5:4 | R/W | 0 | **Color Space Control (if CSAM=1, CSC will be read-only)**<br>00: RGB<br>01: YCrCb-422<br>10: YCrCb-444<br>11: Reserved |
| Reserved | 3 | -- | 0 | **Reserved to 0** |
| PRDSAM | 2 | R/W | 1 | **Pixel Repetition down sampling auto mode**<br>1: auto, the circuit resolve the repeat number, and enable it in next frame. The result could be read in bits for repeat number.<br>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**<br>0: No pop up<br>1: Pop up result (Pop up CTS&N which is acquired completely. If present N&CTS is acquiring, pop up previous complete N&CTS)<br>If the info is updating, HW will refuse this command. |
| PUCSR | 0 | R/W | 0 | **Pop up channel status result**<br>0: No pop up<br>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 | | | | 0x52 |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| CTS | 7:0 | R | 0 | **CTS in usage, CTS[19:12]** |

| Register:: HDMI_ACRSR1 | | | | 0x53 |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| CTS | 7:0 | R | 0 | **CTS in usage, CTS[11:4]** |

| Register:: HDMI_ACRSR2 | | | | 0x54 |
|---|---|---|---|---|
| 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 | | | | 0x55 |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| N | 7:0 | R | 0 | **N in usage, N[15:8]** |

| Register:: HDMI_ACRSR4 | | | | 0x56 |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| N | 7:0 | R | 0 | **N in usage, N[7:0]** |

| Register:: HDMI_ACS0 | | | | 0x57 |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| CS | 7:0 | R | 0 | **Channel status bit7~ bit0** |

| Register:: HDMI_ACS1 | | | | 0x58 |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| CS | 7:0 | R | 0 | **Channel status bit 15~ bit 8** |

| Register:: HDMI_ACS2 | | | | 0x59 |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| CS | 7:0 | R | 0 | **Channel status bit23~ bit 16** |

| Register:: HDMI_ACS3 | | | | 0x5A |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| CS | 7:0 | R | 0 | **Channel status bit 31~ bit 24** |

| Register:: HDMI_ACS4 | | | | 0x5B |
|---|---|---|---|---|
| Name | Bits | R/W | Reset State | Comments |
| CS | 7:0 | R | 0 | **Channel status bit 39~ bit 32** |

| Register:: HDMI_INTCR | | | | 0x60 |
|---|---|---|---|---|
| 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)** |
| AVMUTE | 6 | R/W | 0 | **If get General control packet and the corresponding Set_AVMUTE flag & Clear_AVMUTE flag is different with previous values** <br> 0: IRQ don't occur. <br> 1: IRQ occur. |

| FIFOD | 5 | R/W | 0 | **If FIFO depth reach Target** (Used for manual audio flow)<br>0: IRQ don't occur<br>1: IRQ occur |
|-------|---|-----|---|---|
| ACT | 4 | R/W | 0 | **Audio Coding Type**<br>0: If audio coding type is different with previous value, IRQ doesn't occur.<br>1: If audio coding type is different with previous value, IRQ occurs. |
| APLL | 3 | R/W | 0 | **Audio PLL**<br>0: If audio PLL is non-lock, IRQ doesn't occur<br>1: If audio PLL is non-lock, IRQ occurs |
| AFIFOO | 2 | R/W | 0 | **Audio FIFO Overflow**<br>0: If audio FIFO is overflow for X samples , IRQ doesn't occur.<br>1: If audio FIFO is overflow for X samples , IRQ occurs. |
| AFIFOU | 1 | R/W | 0 | **Audio FIFO Underflow**<br>0: If audio FIFO is underflow for Y samples , IRQ doesn't occur.<br>1: If audio FIFO is underflow for Y samples , IRQ occurs. |
| VC | 0 | R/W | 0 | 1: If video clock is higher than 165Mhz or lower than 25Mhz (refer to NL), IRQ doesn't occur.<br>0: If video clock is higher than 165Mhz or lower than 25Mhz (refer to NL), IRQ occurs. |

| **Register:: HDMI_ALCR** | | | **0x61** | |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| LO1 | 7:6 | R/W | 0 | **Speaker location of I2S #1 & SPDIF OUT#1**<br>00: from SubPacket0 of Audio Sample Packet<br>01: from SubPacket1 of Audio Sample Packet<br>10: from SubPacket2 of Audio Sample Packet<br>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** | | | **0x62** | |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| SPDIFO1 | 7 | R/W | 0 | **SPDIF 1 Output Switch**<br>0: cutoff<br>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**<br>0: cutoff<br>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** | | | **0x70** | |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| Reserved | 7:6 | --- | 0 | **Reserved to 0** |
| NVLGB | 5 | R | 0 | **Video No Leading Guard Band**<br>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<br>Write 1 to clear. |
| NALGB | 4 | R | 0 | **Audio No Leading Guard Band**<br>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<br>Write 1 to clear. |
|---|---|---|---|---|
| NATGB | 3 | R | 0 | **Audio No Trailing Guard Band**<br>If audio packets without trailing GB, this bit would be assigned to 1 until clear this bit.<br>Write 1 to clear. |
| NGB | 2 | R | 0 | **No Guard Band**<br>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.<br>Write 1 to clear. |
| PE | 1 | R | 0 | **Packet Error**<br>If size of Data Island Packet is not times of 32, this bit would be assigned to 1 until clear this bit.<br>Write 1 to clear. |
| GCP | 0 | R | 0 | **General Control Packet error flag**:<br>If HW receive General Control Packet with Clear_AVMUTE=1 & Set_AVMUTE=1 ,assign this bit to 1 until clear this bit<br>Write 1 to clear. |

| Register:: HDMI_ASR0 | | | | 0x71 |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| Reserved | 7:3 | --- | 0 | **Reserved to 0** |
| FsRE | 2 | R | 0 | **Fs Regeneration Error**<br>If CTS & N received 0, this bit would be assigned to 1 until clear this bit<br>Write 1 to clear. |
| FsIF | 1 | R | 0 | **Fs from InfoFrame**<br>If audio frequency from InfoFrame ready, this bit would be assigned to 1 until clear this bit<br>Write 1 to clear. |
| FsCS | 0 | R | 0 | **Fs from Channel Status**<br>If audio frequency from Channel Status ready, this bit would be assigned to 1 until clear this bit<br>Write 1 to clear. |

| Register:: HDMI_ASR1 | | | | 0x72 |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| Reserved | 7 | --- | 0 | **Reserved** |
| FBIF | 6:4 | R | 0 | **Frequency bits from info frame**<br>000: refer to channel status bits<br>001: 32k<br>010: 44.1k<br>011: 48k<br>100: 88.2k<br>101: 96k<br>110: 176.4k<br>111: 192k |
| FBCS | 3:0 | R | 0 | **Frequency bits from channel status**. (pop up with channel status simultaneously)<br>0010: 22.05k<br>0000: 44.1k<br>1001: 88.2k<br>0011: 176.4k<br>0110: 24k<br>0100: 48k<br>0101: 96k<br>0111: 192k<br>1100: 32k |

| | | | | 1000: Sampling frequency not inidicated |
|---|---|---|---|---|

| Register:: TMDS_DPC_SET0 | | | 0x80 | |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| dpc_en | 7 | R/W | 0 | |
| phase_errcnt_in | 6:4 | R/W | 0 | Max. times of phase error to rise error flag<br><br>3'b000 → count 8 times<br>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"<br><br>3'b000 → count 8 times<br>3'b001~3'b111→ count 1~7 times |
| phase_clr_sel | 0 | R/W | 0 | Unit of "phase_clrcnt_in"<br><br>0: Use V sync<br>1: Use H sync |

| Register:: TMDS_DPC_SET1 | | | 0x81 | |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| set_full_noti | 7:4 | R/W | 0 | Set full notifier level (recommend: 3'd7)<br>3'b000~3'b111→ set 0~7 |
| set_empty_noti | 3:0 | R/W | 0 | Set empty notifier level (recommend: 3'd3)<br>3'b000~3'b111→ set 0~7 |

| Register:: TMDS_DPC_SET2 | | | 0x82 | |
|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Reset State** | **Comments** |
| fifo_errcnt_in | 7:5 | R/W | 0 | Max. times of FIFO error to rise error flag<br><br>3'b000 → count 8 times<br>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 | | | 0x83 | |
|---|---|---|---|---|
| **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 |

# LiveShow<sup>TM</sup> Control (Page 3)

| Register::LS_CTRL0 | | | | 0xA1 |
|---|---|---|---|---|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Comments** |
| LS_BYPASS | 7 | R/W | 0 | Display Pixel Resolution<br>0: Bypass LiveShow<sup>TM</sup> Processing<br>1: Enable LiveShow<sup>TM</sup> Procesing |
| LS_BUF_EN | 6 | R/W | 0 | Enable SDRAM Buffer Access<br>0: Disable<br>1: Enable |
| LS_PD_EST | 5 | R/W | 0 | Level Estimation<br>0: Disable<br>1: Enable |
| LS_CPRS_EN | 4 | R/W | 0 | Huffman Data Compression<br>0: Disable<br>1: Enable |
| LS_GAIN_EN | 3 | R/W | 0 | Delta Gain Adjustment<br>0: Disable (Delta Gain=1)<br>1: Enable |
| LS_DISP_RES | 2 | R/W | 0 | Display Pixel Resolution<br>0: 8-bit<br>1: 6-bit |
| SOURCE_RES | 1:0 | R/W | 0 | Source Pixel Resolution<br>00: 6bit<br>01: 5bit<br>1x: 4bit<br>(Pixel Resolution after rounding in previous path.<br>Pixel Resolution for Compression.) |

| Register::LS_CTRL1 | | | | 0xA2 |
|---|---|---|---|---|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Comments** |
| LS_IN_WIN | 7 | R/W | 0 | LiveShow<sup>TM</sup> Inside Highlight Window<br>0: Disable<br>1: Enable |
| LS_OUT_WIN | 6 | R/W | 0 | LiveShow<sup>TM</sup> Outside Highlight Window<br>0: Disable |

| Name | Bits | Read/Write | Reset State | Description |
|---|---|---|---|---|
| | | | | 1: Enable |
| LS_OFST_EN | 5 | R/W | 0 | Offset Compensation<br>0: Disable<br>1: Enable |
| LS_NR_EN | 4 | R/W | 0 | Low-Bit Noise Reduction<br>0: Disable<br>1: Enable |
| LS_NR_MD | 3 | R/W | 0 | Low-Bit Noise Reduction Mode<br>0: RGB Independent Mode (Old mode: RTD2363-like)<br>1: RGB Related Mode (New Mode) |
| LS_NR_THD | 2:0 | R/W | 0 | Low-Bit Noise Reduction Threshold<br>000`b: 4<br>001`b: 6<br>010`b: 8<br>011`b: 10<br>100`b: 12<br>101`b: 14<br>110`b: 16<br>111`b: 18 |

| Register::LS_CPRS_CTRL | | | | 0xA4 |
|---|---|---|---|---|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Description** |
| IM_CPRS_TYPE | 7 | R/W | 0 | Compression Type<br>0: Channel G, R reference to B<br>1: 3 Independent Channels |
| LS_MEM_RES | 6:5 | R/W | 0 | SDRAM Pixel Resolution<br>00: 4-bit<br>01: 5-bit<br>1x: 6-bit<br>(Effective only when LS_CPRS_EN=1, SDRAM pixel resolution must always equal or less than source pixel resolution (SOURCE_RES)) |
| RGB_YC_SEL | 4 | R/W | 0 | New OD algorithm<br>0: RGB rounding/compression<br>1: YC rounding |
| DE_BUF_EMPTY_DLY | 3 | R/W | 1 | Decode buf empty flag delay option<br>0: no delay for decode pre buf empty flag |

| | | | | 1: delay until buf_wr_adr reach 15 to let empty flag be normal |
|---|---|---|---|---|
| LS_RSV_A4_20 | 2:0 | R/W | 0 | Reserved |

| Register::TG_SIZE_H | | | | 0xA5 |
|---|---|---|---|---|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Description** |
| LS_RSV_A5_71 | 7:1 | R/W | 0 | Reserved |
| TG_SIZE_H | 0 | R/W | 0 | Target Size for Compression (Unit: 64 bit) <br><br> Threshold = $\{(num\_break*6)+ \sum (num\_n*length\_n)\}/64+7$ <br><br> Note: <br> 1. Header(2x64)+Dummy rounding Effect(3x64)+Reserved Block(2x64) =7x64-bit <br> 2. num_n = number matched for code n criteria. <br> 3. length_n = length of code n, calculated by Huffman tree generation. <br> 4. num_break = number of pixel matched the break criteria. <br> 5. Max. target-size = 256 (unit: 64-bit) <br><br> Threshold must be set in even number |

| Register::TG_SIZE_L | | | | 0xA6 |
|---|---|---|---|---|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Description** |
| TG_SIZE_L | 7:0 | R/W | 0x00 | Target Size for Compression (Unit: 64 bit) <br><br> Threshold = $\{(num\_break*6)+ \sum (num\_n*length\_n)\}/64+7$ <br><br> Note: <br> 1. Header(2x64)+Dummy rounding Effect(3x64)+Reserved Block(2x64) =7x64-bit <br> 2. num_n = number matched for code n criteria. <br> 3. length_n = length of code n, calculated by Huffman tree generation. <br> 4. num_break = number of pixel matched the break criteria. <br> 5. Max. target-size = 256 (unit: 64-bit) <br><br> Threshold must be set in even number |

| Register::GRP_NUM_H | | | | 0xA7 |
|---|---|---|---|---|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Description** |
| LS_RSV_A7_72 | 7:3 | R/W | 0 | Reserved |
| GRP_NUM_H | 2:0 | R/W | 0 | Number of Pixel per Group to be Analyzed and Compressed. (max. group number = 1280) |

| Register::GRP_NUM_L | | | | 0xA8 |
|---|---|---|---|---|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Description** |
| GRP_NUM_L | 7:0 | R/W | 0 | Number of Pixel per Group to be Analyzed and Compressed. (max. group number = 1280) |

| Register::FAIL_CNT_H | | | | 0xA9 |
|---|---|---|---|---|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Description** |
| LS_RSV_A9_74 | 7:4 | R/W | 0 | Reserved |
| FAIL_CNT_H | 3:0 | R | 0 | The Count of Compression Fail |

| Register::FAIL_CNT_L | | | | 0xAA |
|---|---|---|---|---|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Description** |
| FAIL_CNT_L | 7:0 | R | 0 | The Count of Compression Fail (Updated when DVS occurred) |

Compression Format:

{1}, {symbol0 3-bit code length, 7bit code}, …, {symbol7 3-bit code length, 7bit code}, {B0 code}, {G0 code}, {R0 code}, {B1 code}, {G1 code}, {R1 code}….

Non-Compression Format:

{0}, {B0 4-bit MSB}, {G0 4-bit MSB}, {R0 4-bit MSB}, {B1 4-bit MSB}, {G1 4-bit MSB}, {R1 4-bit MSB}….

| Register::LS_LUT_ROW_ADDR | | | | 0xAE |
|---|---|---|---|---|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Description** |
| LS_LUT_ACS_EN | 7 | R/W | 0 | LUT Access Enable<br>0: Disabled<br>1: Enabled |
| LS_RSV_AE_66 | 6 | R/W | 0 | Reserved |
| LS_LUT_ROW | 5:0 | R/W | 0 | LUT Row Selector(Current Frame as index) |

| Register::LS_LUT_COL_ADDR | | | | 0xAF |
|---|---|---|---|---|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Description** |
| LS_LUT_SEL | 7:6 | R/W | 0 | LUT Channel Selector<br>00: Red Channel<br>01: Green Channel<br>10: Blue Channel |

| | | | | 11: All Channels |
| LS_LUT_COL | 5:0 | R/W | 0 | LUT Column Selector (Previous Frame as index) |

| Register::LS_LUT_DATA | | | | 0xB0 |
|---|---|---|---|---|
| Name | Bits | Read/Write | Reset State | Description |
| LS_LUT_DATA | 7:0 | R/W | 0 | LUT Data Port |

| Register::DELTA_GAIN | | | | 0xB1 |
|---|---|---|---|---|
| Name | Bits | Read/Write | Reset State | Description |
| LS_RSV_B1_77 | 7 | R/W | 0 | Reserved |
| DELTA_GAIN | 6:0 | R/W | 0 | Delta Gain Setting<br>0x00 -> Gain = 0<br>0x40 -> Gain = 1<br>0x7F -> Gain =127/64<br>(Effective only when LS_GAIN_EN=1) |

| Register::UDST_THD | | | | 0xB2 |
|---|---|---|---|---|
| Name | Bits | Read/Write | Reset State | Description |
| LS_RSV_B2_77 | 7 | R/W | 0 | Reserved |
| UDST_THD | 6:0 | R/W | 0 | Undershoot Threshold (2's complement)<br>0x00 -> THD = 0<br>0x7F -> THD = -127<br>(Effective only when LS_OFST_EN =1) |

| Register::OVST_THD | | | | 0xB3 |
|---|---|---|---|---|
| Name | Bits | Read/Write | Reset State | Description |
| LS_RSV_B3_77 | 7 | R/W | 0 | Reserved |
| OVST_THD | 6:0 | R/W | 0 | Overshoot Threshold<br>0x00 -> THD = 0<br>0x7F -> THD = 127<br>(Effective only when LS_OFST_EN =1) |

| Register::UDST_GAIN | | | | 0xB4 |
|---|---|---|---|---|
| Name | Bits | Read/Write | Reset State | Description |
| LS_RSV_B4_76 | 7:6 | R/W | 0 | Reserved |
| UDST_GAIN | 5:0 | R/W | 0 | Undershoot Gain<br>0x00 -> Gain = 0/128 |

| | | | | 0x3F -> Gain = 63/128 |
| --- | --- | --- | --- | --- |
| | | | | (Effective only when LS_OFST_EN =1) |

| **Register::OVST_GAIN** | | | | **0xB5** |
| --- | --- | --- | --- | --- |
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Description** |
| LS_RSV_B5_76 | 7:6 | R/W | 0 | Reserved |
| OVST_GAIN | 5:0 | R/W | 0 | Overshoot Gain |
| | | | | 0x00 -> Gain = 0/128 |
| | | | | 0x3F -> Gain = 63/128 |
| | | | | (Effective only when LS_OFST_EN =1) |

| **Register::LS_STATUS0** | | | | **0xB6** |
| --- | --- | --- | --- | --- |
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Comments** |
| LS_RBUF_FULL | 7 | R | 0 | Set if BUF_R is full (On-line monitor) |
| LS_RBUF_EPTY | 6 | R | 0 | Set if BUF_R is empty (On-line monitor) |
| LS_RBUF_UDFW | 5 | R | 0 | Set if BUF_R is underflow |
| LS_WBUF_FULL | 4 | R | 0 | Set if BUF_W is full (On-line monitor) |
| LS_WBUF_EPTY | 3 | R | 0 | Set if BUF_W is empty (On-line monitor) |
| LS_WBUF_OVFW | 2 | R | 0 | Set if BUF_W is overflow |
| Reserved | 1 | -- | 0 | Reserved |
| LS_STATUS0_RST | 0 | R/W | 0 | Write 1 to reset BUF and FIFO status (Auto clear after done) |

| **Register::LS_STATUS1** | | | | **0xB7** |
| --- | --- | --- | --- | --- |
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Comments** |
| LS_RFIFO_FULL | 7 | R | 0 | Set if FIFO_R is full (On-line monitor) |
| LS_RFIFO_EPTY | 6 | R | 0 | Set if FIFO_R is empty (On-line monitor) |
| LS_RFIFO_OVFW | 5 | R | 0 | Set if FIFO_R is overflow before LS_RFIFO_UDFW is set |
| LS_RFIFO_UDFW | 4 | R | 0 | Set if FIFO_R is underflow before LS_RFIFO_OVFW is set |
| LS_WFIFO_FULL | 3 | R | 0 | Set if FIFO_W is full (On-line monitor) |
| LS_WFIFO_EPTY | 2 | R | 0 | Set if FIFO_W is empty (On-line monitor) |
| LS_WFIFO_OVFW | 1 | R | 0 | Set if FIFO_W is overflow before LS_WFIFO_UDFW is set |

| LS_WFIFO_UDFW | 0 | R | 0 | Set if FIFO_W is underflow before LS_WFIFO_OVFW is set |
|---|---|---|---|---|

| Register::LS_WTLVL_W | | | | 0xC0 |
|---|---|---|---|---|
| Name | Bits | Read/Write | Reset State | Comments |
| Reserved | 7 | -- | 0 | Reserved |
| LS_WTLVL_W | 6:0 | R/W | 0x40 | When FIFO depth is over WTLVL, FIFO write data ((NUM*LEN)+REM) * 64 = one frame/line data<br>The assigned value multiplied by 2 is the real value. |

| Register::LS_WTLVL_R | | | | 0xC1 |
|---|---|---|---|---|
| Name | Bits | Read/Write | Reset State | Comments |
| Reserved | 7 | -- | 0 | Reserved |
| LS_WTLVL_R | 6:0 | R/W | 0x40 | When FIFO depth is over WTLVL, FIFO write data ((NUM*LEN)+REM) * 64 = one frame/line data<br>The assigned value multiplied by 2 is the real value. |

| Register::LS_MEM_FIFO_RW_NUM_H | | | | 0xC2 |
|---|---|---|---|---|
| Name | Bits | Read/Write | Reset State | Comments |
| LS_MFRW_NO_H | 7:0 | R/W | 0x01 | LS_MEN_FIFO_RW_NUM [15:8]<br>The Read/Write times of total memory access. |

| Register::LS_MEM_FIFO_RW_NUM_L | | | | 0xC3 |
|---|---|---|---|---|
| Name | Bits | Read/Write | Reset State | Comments |
| LS_MFRW_NO_L | 7:0 | R/W | 0x00 | LS_MEM_FIFO_RW_NUM [7:0]<br>The Read/Write times of total memory access. |

| Register::LS_MEM_FIFO_RW_LEN | | | | 0xC4 |
|---|---|---|---|---|

| Name | Bits | Read/Write | Reset State | Comments |
|---|---|---|---|---|
| LS_MFRW_LEN | 7:0 | R/W | 0x80 | LS_MEM_FIFO_RW_LEN [7:0] <br> The Read/Write number of words in each memory access. |

| Register::LS_MEM_FIFO_RW_REMAIN | | | | 0xC5 |
|---|---|---|---|---|
| Name | Bits | Read/Write | Reset State | Comments |
| LS_MFRW_RM | 7:0 | R/W | 0x80 | LS_MEM_FIFO_RW_ REMAIN [7:0] <br> The Read/Write number of words at the last access. This register must be 4X. |

| Register::LS_MEM_START_ADDR_H | | | | 0xC6 |
|---|---|---|---|---|
| Name | Bits | Read/Write | Reset State | Comments |
| LS_RSV_C6_77 | 7 | R/W | 0 | Reserved |
| LS_MEM_ADR_H | 6:0 | R/W | 0x00 | LS_MEM_START_ADDR [22:16] <br> Start address of LS memory block (Total 22/23 bits) . |

If the columns per bank are 256, and Bank = 4 , SDRAM address [22:0] is: 1'b0+R[11:0]+B[1:0]+C[7:0]

If the columns per bank are 256, and Bank = 2 , SDRAM address [22:0] is: 2'b0+R[11:0]+B[0]+C[7:0]

If the columns per bank are 512, and Bank = 4 , SDRAM address [22:0] is: R[11:0]+B[1:0]+C[8:0]

If the columns per bank are 512, and Bank = 2 , SDRAM address [22:0] is: 1'b0+R[11:0]+B[0]+C[8:0]

| Register::LS_MEM_START_ADDR_M | | | | 0xC7 |
|---|---|---|---|---|
| Name | Bits | Read/Write | Reset State | Comments |
| LS_MEM_ADR_M | 7:0 | R/W | 0x00 | LS_MEM_START_ADDR [15:8] <br> Start address of LS memory block (Total 22/23 bits) |

| Register::LS_MEM_START_ADDR_L | | | | 0xC8 |
|---|---|---|---|---|
| Name | Bits | Read/Write | Reset State | Comments |
| LS_MEM_ADR_L | 7:0 | R/W | 0x00 | LS_MEM_START_ADDR [7:0] <br> Start address of LS memory block (Total 22/23 |

| | | | | bits) |
|---|---|---|---|---|

| Register::LS_BIST_CTRL | | | | | 0xC9 |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Comments** | |
| LS_TEST | 7:6 | R/W | 0 | Reserved for testing | |
| LS_RSV_C9_5 | 5 | R/W | 0 | Reserved | |
| FREEZE_MODE | 4 | R/W | 0 | Freeze mode enable | |
| LS_TEST_EN | 3 | R/W | 0 | LiveShow$^{TM}$ Test Enable.<br>0: Disable<br>1: Enable | |
| LS_TEST_MODE | 2 | R/W | 0 | LiveShow$^{TM}$ Test Mode.<br>0: Bypass interpolated delta<br>1: Bypass LUT4 value | |
| LS_BIST_START | 1 | R/W | 0 | LiveShow$^{TM}$ Memory BIST Start.<br>Set 1 to start and auto-clear after finished. | |
| LS_BIST_RESULT | 0 | R | 0 | LiveShow$^{TM}$ Memory BIST Result.<br>0: Failed<br>1: Pass | |

| Register::LS_COMP_CHK | | | | | 0xCA |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Comments** | **Config** |
| LS_RSV_C9_77 | 7 | R/W | 0 | Reserved | |
| PREBUF_UDFW | 6 | R | 0 | Decompression previous buffer under flow | |
| PREBUF_STA_CLR | 5 | R | 0 | Decompression previous buffer status clear | wclr_out |
| REBUF_OVFW_VDUM | 4 | R | 0 | Reorder buffer overflow by vsync dummy purge, the previous frame result | |
| REBUF_OVFW_GDUM_MSB | 3:0 | R | 0 | Reorder buffer overflow by group dummy purge, the previous frame result, update by vsync, msb | |

| Register::LS_COMP_REOVFW | | | | | 0xCB |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Comments** | **Config** |
| REBUF_OVFW_GD UM_LSB | 7:0 | R | 0 | Reorder buffer overflow by group dummy purge, the previous frame result, update by vsync, lsb | |

| Register::LS_FRAME0 | | | | | 0xCC |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Comments** | **Config** |
| CUR_BYPASS_EN | 7 | R/W | 0 | In Current frame path, ByPass the conversion path and go through OD_LUT directly:<br>  0: Disable<br>  1: Enable | |
| LS_RSV_CC_65 | 6:5 | R/W | 0 | Reserved | |
| CUR_RGB2YUV_EN | 4 | R/W | 0 | In Current frame path, RGB to YUV:<br>  0: Disable<br>  1: Enable | |
| CUR_444TO422_EN | 3 | R/W | 0 | In Current frame path, YUV 444 to 422:<br>  0: Disable<br>  1: Enable | |
| LS_RSV_CC_21 | 2:1 | R/W | 0 | Reserved | |
| UV_MODE | 0 | R/W | 0 | In Current/Previous frame path, 444to422 U/V type<br>0: U0 V0 U2 V2 U4 V4 …<br>1: U0 V1 U2 V3 U4 V5 … | |

| Register::LS_FRAME1 | | | | | 0xCD |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Comments** | **Config** |
| CUR_422TO444_EN | 7 | R/W | 0 | In Current frame path, YUV 422 to 444:<br>  0: Disable<br>  1: Enable | |
| CUR_DUPLICATE | 6 | R/W | 0 | In Current frame path, YUV 422 to 444:<br>  0: Interpolation Mode<br>  1: Duplication Mode<br><br>Interpolation Mode:<br>Original sequence: Y0U0, Y1V0, Y2U2, Y3V2, Y4U4, Y5V4,....<br>Final sequences: Y0U0V0, Y1((U0+U2)/2)((V0+V2)/2), Y2U2V2, Y3((U2+U4)/2)((V2+V4)/2), Y4U4V4, ...<br><br>Duplication Mode:<br>Original sequence: Y0U0, Y1V0, Y2U2, Y3V2, Y4U4, Y5V4,.... | |

| Name | Bits | Read/Write | Reset State | Comments | Config |
|---|---|---|---|---|---|
| | | | | Final sequences: Y0U0V0, Y1U0V0, Y2U2V2, Y3U2V2, Y4U4V4, Y5U5V5,... | |
| LS_RSV_CD_5 | 5 | R/W | 0 | Reserved | |
| CUR_YUV2RGB_EN | 4 | R/W | 0 | In Current frame path, YUV to RGB:<br>  0: Disable<br>  1: Enable | |
| PRE_422TO444_EN | 3 | R/W | 0 | In Previous frame path, YUV 422 to 444:<br>  0: Disable<br>  1: Enable | |
| PRE_DUPLICATE | 2 | R/W | 0 | In Previous frame path, YUV 422 to 444:<br>  0: Interpolation Mode<br>  1: Duplication Mode | |
| LS_RSV_CD_1 | 1 | R/W | 0 | Reserved | |
| PRE_YUV2RGB_EN | 0 | R/W | 0 | In Previous frame path, YUV to RGB:<br>  0: Disable<br>  1: Enable | |

| Register::LS_FRAME2 | | | | | 0xCE |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Comments** | **Config** |
| CUR_ROUND | 7:6 | R/W | 0 | The Current Pixel after RGB->YUV, 444->422 rounding mode setting<br>00: no rounding (keep original 10bit)<br>01: 6bit rounding<br>10: 4bit rounding<br>11: 5bit rounding | |
| EST_VALUE | 5:0 | R/W | 0 | User Defined Level Estimation Value:<br>0~63<br>  (When Level Estimation Enabled, i.e. CRA1[5]==1, and working under YC rounding mode, CRA4[4] = 1'b1) | |

## SDRAM Control (Page 4)

| Register::SDR_CTRL0 | | | | 0xA1 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| SDR_RSV_A1_76 | 7:6 | R/W | 0 | Reserved | |
| SDR_WR_DELAY | 5:3 | R/W | 'b011 | Delay from Row Active to Write<br>000: Reserved<br>001: Reserved<br>010: 2 MCLK<br>011: 3 MCLK<br>100: 4 MCLK<br>101: 5 MCLK<br>110: Reserved<br>111: Reserved | |
| SDR_RD_DELAY | 2:0 | R/W | 'b011 | Delay from Row Active to Data Valid<br>000: Reserved<br>001: Reserved<br>010: 2 MCLK<br>011: 3 MCLK<br>100: 4 MCLK<br>101: 5 MCLK<br>110: Reserved<br>111: Reserved | |

| Register::SDR_CTRL1 | | | | 0xA2 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| SDR_CL | 7:5 | R/W | 'b011 | CAS Latency of SDRAM<br>000: Reserved<br>001: Reserved<br>010: 2 MCLK<br>011: 3 MCLK<br>100: Reserved<br>101: Reserved<br>110: Reserved<br>111: Reserved | |

| | | | | If MCLK >100MHz, SDR_CL should be 3 MCLK. | |
|---|---|---|---|---|---|
| SDR_RSV_A2_40 | 4:0 | R/W | 4 | Reserved | |

| Register::SDR_AREF_TIME | | | | 0xA3 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| SDR_AREF_TIME | 7:0 | R/W | 0x0D | Auto Refresh Time. (The period of initial refresh time in MCLK cycle) | |

| Register::SDR_PRCG | | | | 0xA4 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| SDR_PRCG_BIT | 7 | R/W | 0 | Precharge All Banks by<br>0: A8<br>1: A10 | |
| SDR_PRCG_DO | 6 | R/W | 0 | Force to Precharge All Banks | |
| SDR_COL_NUM | 5 | R/W | 0 | Columns per Bank<br>0: 256<br>1: 512 | |
| SDR_RESET | 4 | R/W | 0 | SDR Reset<br>0: Normal<br>1: Reset | |
| SDR_PRCG_DLY | 3:0 | R/W | 3 | Precharge Delay Cycle (The interval from precharge to next valid command) | |

| Register::SDR_MEM_TYPE | | | | 0xA5 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| SDR_MEM_SIZE | 7:6 | R/W | 0 | SDRAM Memory Size<br>00: 1Mx16x1pcs<br>01: 1Mx16x2pcs<br>10: 2Mx32x1pcs<br>11: 2Mx32x2pcs | |
| SDR_BANK_SEL | 5 | R/W | 0 | Banks per SDRAM<br>0: 4 bank | |

| Name | Bits | Read/Write | Reset State | Comments | Config |
|------|------|------------|-------------|----------|--------|
| | | | | 1: 2 bank | |
| SDR_ABR_STATUS | 4 | R | 0 | Arbiter Recovery Happen | |
| SDR_ABR_REC_EN | 3 | R/W | 0 | Arbiter Recovery Enable, Reset State Machine | |
| SDR_CAS_LATN | 2:0 | R/W | 1 | CAS Latency for Controller<br>000: Reserved<br>001: 1<br>010: 1<br>011: 2<br>100: 2<br>101: 3<br>110: 3<br>111: 4 | |

| Register::SDR_SLEW_RATE | | | | 0xA6 | |
|------|------|------------|-------------|----------|--------|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Comments** | **Config** |
| SDR_RSV_A6_73 | 7:3 | R/W | 0 | Reserved | |
| SDR_AUTO_GATING | 2 | R/W | 0 | Auto gating CKE<br>0: Disable<br>1: Enable | |
| SDR_CKE_L | 1 | R/W | 0 | Force CKE Low (for power-down mode)<br>0: Disable<br>1: Enable | |
| SDR_CKE_H | 0 | R/W | 0 | Force CKE High (for testing)<br>0: Disable<br>1: Enable | |

| Register::SDR_AREF_CNT | | | | 0xA7 | |
|------|------|------------|-------------|----------|--------|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Comments** | **Config** |
| SDR_AREF_CNT8 | 7:0 | R/W | 0x81 | Number of Auto Refresh ( N*8 ) | |

| Register::RESERVED | | | | 0xA8 | |
|------|------|------------|-------------|----------|--------|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Comments** | **Config** |
| Reserved | 7:0 | R/W | 0 | Reserved | |

| Register::RESERVED | | | | | 0xA9 |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| Reserved | 7:0 | R/W | 0 | Reserved | |

| Register::SDR_RSC_AREF | | | | | 0xAA |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| SDR_RSC_AREF | 7:0 | R | 0xFF | Token Ring Bit[7:0] | |

| Register::SDR_RSC_MCU | | | | | 0xAB |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| SDR_RSC_MCU | 7:0 | R/W/D | 0x20 | Token Ring Bit[7:0] | |

| Register::SDR_RSC_CAP1 | | | | | 0xAC |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| SDR_RSC_CAP1 | 7:0 | R/W/D | 0xAA | Token Ring Bit[7:0] | |

| Register::RESERVED | | | | | 0xAD |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| Reserved | 7:0 | R/W | 0 | Reserved | |

| Register::SDR_RSC_MAIN | | | | | 0xAE |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| SDR_RSC_MAIN | 7:0 | R/W/D | 0x55 | Token Ring Bit[7:0] | |

| Register::RESERVED | | | | | 0xAF |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset** | **Comments** | **Config** |

| | | Write | State | | |
|---|---|---|---|---|---|
| Reserved | 7:0 | R/W | 0 | Reserved | |

| Register::SDR_RSC_RTC_RD | | | | 0xB0 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| SDR_RSC_RTC_RD | 7:0 | R/W/D | 0xAA | Token Ring Bit[7:0] | |

| Register::SDR_RSC_RTC_WR | | | | 0xB1 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| SDR_RSC_RTC_WR | 7:0 | R/W/D | 0x55 | Token Ring Bit[7:0] | |

- *(Double-Buffer) Token-Ring access before AREF*

| Register::RESERVED | | | | 0xB2 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| Reserved | 7:0 | R/W | 0 | Reserved | |

| Register::RESERVED | | | | 0xB3 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| Reserved | 7:0 | R/W | 0 | Reserved | |

| Register::SDR_ABTR_STATUS0 | | | | 0xB4 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| SDR_ABTR_RTCR | 7 | R | 0 | RTC Read Arbiter Status | |
| SDR_ABTR_RTCW | 6 | R | 0 | RTC Write Arbiter Status | |
| SDR_ABTR_MAIN | 5 | R | 0 | MAIN Read Arbiter Status | |
| Reserved | 4 | -- | 0 | Reserved | |
| SDR_ABTR_CAP1 | 3 | R | 0 | CAP1 Write Arbiter Status | |
| Reserved | 2 | -- | 0 | Reserved | |
| SDR_ABTR_MCU | 1 | R | 0 | MCU R/W Arbiter Status | |

| SDR_ABTR_AREF | 0 | R | 0 | AREF Arbiter Status | |
|---|---|---|---|---|---|

- *Write-clear*

| Register::SDR_ABTR_STATUS 1 | | | | 0xB5 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| Reserved | 7:1 | -- | 0 | Reserved | |
| SDR_RESET_RDY | 0 | R | 0 | SDR Reset Ready | |

| Register::RESERVED | | | | 0xB6 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| Reserved | 7:0 | R/W | 0 | Reserved | |

| Register::RESERVED | | | | 0xB7 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| Reserved | 7:0 | R/W | 0 | Reserved | |

| Register::SDR_ADDR_H | | | | 0xB8 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| Reserved | 7 | -- | 0 | Reserved | |
| SDR_ADDR_H | 6:0 | R/W | 0 | **SDR_ADDR [22:16]** If the columns per bank are 256,bit[5:0] is assigned to R[11:6] and bit[6] is reserved. If the columns per bank are 512,bit[6:0] is assigned to R[11:5] | |

If the columns per bank are 256, SDRAM address [22:0] is: R[11:0]+B[1:0]+C[7:0]

If the columns per bank are 512, SDRAM address [22:0] is: R[11:0]+B[1:0]+C[8:0]

| Register::SDR_ADDR_M | | | | 0xB9 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| SDR_ADDR_M | 7:0 | R/W | 0 | **SDR_ADDR [15:8]** If the columns per bank are 256, bit[7:2] is assigned to R[5:0] bit[1:0] is assigned to B[1:0] If the columns per bank are 512,bit bit[7:3] is assigned to R[4:0] bit[2:1] is assigned to B[1:0] bit[0] is assigned to C[8] | |

| Register::SDR_ADDR_L | | | | 0xBA | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| SDR_ADDR_L | 7:0 | R/W | 0 | **SDR_ADDR [7:0]** Bitp7:0] is assigned to C[7:0] regardless of columns per bank are 256 or 512. | |

| Register::SDR_ACCESS_CMD | | | | 0xBB | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| SDR_ACS_CMD | 7:5 | R/W | 0 | **SDR_ACCESS_COMMAND** (clear to 000 after finish) 000: NOP or Finish 001: Pre-charge (all bank or single bank) 010: Auto-Refresh (step by step or auto arbiter) 011: Load Mode Register (step by step or auto initialization) (Load Mode Register will reset DLL of DRAM, we must idle ~200cycles before next "READ") 100: WRITE command (Buf→SDR) 101: READ command(SDR→Buf) 110: Reserved 111: Reserved | |
| SDR_DBUF_IDX | 4:0 | R/W | 0 | **DATA_BUFFER_INDEX** Specifies the next access byte in the buffer. | |

| Register::SDR_DATA_BUF | | | | 0xBC | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| SDR_DATA_BUF | 7:0 | R/W | 0 | **SDR_DATA_BUFFER** Sequential 8-word (32 byte) READ/WRITE from low to high address auto-increase. DATA_BUFFER_INDEX specifies the next access byte in the buffer. | |

- *SDR_ACCESS (Read/Write) can be used for MCU acces*
- *How to modify only one-byte in SDR? Read 32 bytes, only modify one index-select byte, Write 32 bytes.*

| Register::SDR_MCU_RD_LEN | | | | 0xBD | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| Mcurd_tst_en | 7 | R/W | 0 | On-line test mcurd SDRAM enable | |
| Mcurd_len | 6:0 | R/W | 0 | Mcu Read SRAM Length | |

| Register::phase calibration | | | | 0xBE | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |

| Name | Bits | Read/Write | Reset State | Comments | Config |
|---|---|---|---|---|---|
| reserve | 7:5 | R/W | 0 | | |
| Ph_cal_up_sel | 4 | R/W | 0 | Phase calibration wait event select<br>0 : SDRAM write (not include MCU write to SDRAM)<br>1: display vertical front porch | |
| Phcal_wait_en | 3 | R/W | 0 | Calibration wait event (ref. CR_BE[4]) | |
| Phcal_en | 2 | R/W | 0 | Phase calibration enable<br>0: finish<br>1: enable (auto-clear by HW) | |
| Rd_ph_db_en | 1 | R/W | 0 | Double buffer enable(Update SRAM DATAT DQS fine dly) | |
| Rd_ph_db_start | 0 | R/W | 0 | Start double buffer (ref. CR_BE[4])<br>0: finish<br>1: start (auto clear by HW) | |

| Register::calibration_result | | | | 0xBF | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Comments** | **Config** |
| reserve | 7 | R | 0 | | |
| Phcal_cnt | 6:0 | R | 0 | | |

| Register::SDR_CLK_DLY1 | | | | 0xC0 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Comments** | **Config** |
| mclko_inv | 7 | R/W/D | 0 | Mclk ouput invert<br>0 : non invert<br>1: inert | |
| reserve | 6:0 | R/W | 0 | | |

| Register::SDR_CLK_DLY2 | | | | 0xC1 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Comments** | **Config** |
| mclk_fine_tune | 7:0 | R/W/D | 0 | Mclk delay fine tune[7:0] | |

| Register::DQS0_DLY1 | | | | 0xC2 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| SDR_D0_LFT_OFF | 7 | R | 0 | SDRAM Data [15:0] Latch Fine-Tune Status<br>0: Actived<br>1: Inactived | |
| dqs0_coarse_dly | 6:5 | R/W/D | 0 | SDRAM Data [15:0]coarse dly [1:0]<br>00: 0<br>01: 90<br>10: 180<br>11: 270 | |
| reserve | 4:0 | R/W | 0 | | |

| Register::DQS0_DLY2 | | | | 0xC3 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| dqs0_fine_dly | 7:0 | R/W/D | 0 | SDRAM Data[15:0] fine dly [7:0] | |

| Register::DQS1_DLY1 | | | | 0xC4 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| SDR_D1_LFT_OFF | 7 | R | 0 | SDRAM Data [31:16] Latch Fine-Tune Status<br>0: Actived<br>1: Inactived | |
| dqs1_coarse_dly | 6:5 | R/W/D | 0 | SDRAM Data[31:16] coarse dly [1:0]<br>00: 0<br>01: 90<br>10: 180<br>11: 270 | |
| reserve | 4:0 | R/W | 0 | | |

| Register::DQS1_DLY2 | | | | 0xC5 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |

| dqs1_fine_dly | 7:0 | R/W/D | 0 | SDRAM Data[31:16] fine dly [7:0] | |

| Register::DQS2_DLY1 | | | | 0xC6 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| SDR_D2_LFT_OFF | 7 | R | 0 | SDRAM Data [47:32] Latch Fine-Tune Status<br>0: Actived<br>1: Inactived | |
| Dqs2_coarse_dly | 6:5 | R/W/D | 0 | SDRAM Data[47:32] coarse dly [1:0]<br>00: 0<br>01: 90<br>10: 180<br>11: 270 | |
| reserve | 4:0 | R/W | 0 | | |

Note: reserve for 64 bits SDRAM

| Register::DQS2_DLY2 | | | | 0xC7 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| Dqs2_fine_dly | 7:0 | R/W/D | 0 | SDRAM Data[47:32] fine dly [7:0] | |

Note: reserve for 64 bits SDRAM

| Register::DQS3_DLY1 | | | | 0xC8 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| SDR_D3_LFT_OFF | 7 | R | 0 | SDRAM Data [63:48] Latch Fine-Tune Status<br>0: Actived<br>1: Inactived | |
| Dqs3_coarse_dly | 6:5 | R/W/D | 0 | SDRAM Data coarse dly [1:0]<br>00: 0<br>01: 90<br>10: 180<br>11: 270 | |
| reserve | 4:0 | R/W | 0 | | |

Note: reserve for 64 bits SDRAM

| Register::DQS3_DLY2 | | | | 0xC9 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| Dqs3_fine_dly | 7:0 | R/W/D | 0 | SDRAM Data[63:48] fine dly [7:0] | |

Note: reserve for 64 bits SDRAM

| Register::SEC_DQS0_DLY | | | | 0xCA | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| Sec_dqs0_fine_dly | 7:0 | R/W | 0 | SDRAM Data[15:0] fine dly [7:0]<br>Phase switch setting for on-line mcurd to check phase | |

| Register::SEC_DQS1_DLY | | | | 0xCB | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| Sec_dqs1_fine_dly | 7:0 | R/W | 0 | SDRAM Data[31:16] fine dly [7:0]<br>Phase switch setting for on-line mcurd to check phase | |

| Register::SEC_DQS2_DLY | | | | 0xCC | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| Sec_dqs2_fine_dly | 7:0 | R/W | 0 | SDRAM Data[47:32] fine dly [7:0]<br>Phase switch setting for on-line mcurd to check phase | |

Note: reserve for 64 bits SDRAM

| Register::SEC_DQS3_DLY | | | | 0xCD | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| Sec_dqs3_fine_dly | 7:0 | R/W | 0 | SDRAM Data[63:48] fine dly [7:0]<br>Phase switch setting for on-line mcurd to check phase | |

Note: reserve for 64 bits SDRAM

*Address: CE~FB Reserved*

| Register::extended_mode_register | | | | 0xFC | |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| Mcurd_crc_en | 7 | R/W | 0 | CRC Enable for MCU Read from SDRAM only | |
| reserve | 6:5 | R/W | 0 | | |
| Emr_config[11:8] | 3:0 | R/W | 0 | extended_mode_register[11:8] | |

| Register::extended_mode_register | | | | 0xFD | |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| Emr_config[7:0] | 7:0 | R/W | 0 | extended_mode_register[7:0] | |

| Register::random_generator | | | | 0xFE | |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| Rst_random_sel | 7 | R/W | 0 | SDRAM controller random generator reset<br>0: dvs<br>1: ivs | |
| Rst_crc_sel | 6 | R/W | 0 | SDRAM controller CRC reset<br>0: dvs<br>1: ivs | |
| Random_en | 5 | R/W | 0 | SDRAM controller random generator enable | |
| Crc_start | 4 | R/W | 0 | SDRAM controller CRC start<br>0: finish<br>1: start (auto-clear by HW) | |
| Crc_adr_port | 3:0 | R/W | 0 | Address port for CRFF | |

| Register::CRC_DATA_PORT | | | | 0xFF | |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| Crc_data_port | 7:0 | R | 0 | CRC Data Port | |

**Reserved (Page 5)**
**Reserved (Page 6)**

## Vivid color-Video Color Space Conversion(page 7)

| Register:: YUV2RGB_CTRL | | | | | 0xBF |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| Dummy | 7:2 | R/W | 0 | Reserved | |
| Access | 1 | R/W | 0 | Enable YUV/RGB coefficient Access 0: Disable 1: Enable | |
| Enable | 0 | R/W | 0 | Enable YUV to RGB Conversion 0: Disable YUV-to-RGB conversion 1: Enable YUV-to-RGB conversion | |

| Register:: YUV2RGB_ACCESS | | | | | 0xC0 |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| Write_Enabled | 7:3 | R/W | 0 | YUV Coefficient Write Enable: 00000: K11 high byte 00001: K11 low byte 00010: K13 high byte 00011: K13 low byte 00100: K22 high byte 00101: K22 low byte 00110: K23 high byte 00111: K23 low byte 01000: K32 high byte 01001: K32 low byte 01010: Roffset high byte 01011: Roffset low byte 01100: Goffset high byte 01101: Goffset low byte 01110: Boffset high byte 01111: Boffset low byte 10000: Rgain high byte 10001: Rgain low byte 10010: Ggain high byte 10011: Ggain low byte 10100: Bgain high byte 10101: Bgain low byte 10110~11111: reserved | |
| Cb_Cr_Clamp | 2 | R/W | 0 | Cb Cr Clamp 0: Bypass 1: Cb-(128), Cr-(128) | |
| Y_Clamp | 1 | R/W | 0 | Y Clamp 0: Bypass 1: Y-(16) | |
| Y Signed | 0 | R/W | 0 | Y Signed Selection 0: (Y-16)-> Unsigned 1: (Y-16)-> Signed | |

| Register:: YUV_RGB_COEF_DATA | | | | | 0xC1 |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| COEF | 7:0 | W | - | COEF_DATA[7:0] | |

$$\text{YUV/RGB matrix} \quad \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} K_{11} & 0 & K_{13} \\ K_{11} & -K_{22} & -K_{23} \\ K_{11} & K_{32} & 0 \end{bmatrix} \begin{bmatrix} Y \ or \ (Y-16) \\ U \ or \ (U-128) \\ V \ or \ (V-128) \end{bmatrix} + \begin{bmatrix} R_{offset} \\ G_{offset} \\ B_{offset} \end{bmatrix}$$

Then,

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} R_{gain} \times R' \\ G_{gain} \times G' \\ B_{gain} \times B' \end{bmatrix}$$

Where
- Y: S(9,0)./ U(9,0) when CR C0[0]=0
- U, V: S(8,0).
- K11: U(12, 10) 12 bits, 2 bit integer and 10-bit fractional bits. (Default: 0x0400h)
- K13: U(11, 10) 11 bits, 1 bit integer and 10-bit fractional bits (Default: 0x048Fh)
- K22, K23: U(10, 10) 10 bits, all fractional bits (Default: K22: 0x0194h, K23: 0x0252h)
- K32: U(12, 10) 12 bits, 2 bit integer and 10-bit fractional bits (Default: 0x0820h)
- K11': S(15,4)
- Roffset, Goffset, Boffset: S(14,4) 14 bits, 10 bit signed integer and 4-bit fractional bits. (Default: 0x000h)
- K13': S(15,4)
- K22', K23': S(11,2)
- K32': S(13,2)
- Rgain, Ggain, Bgain: U(10, 9) 10bits, 1 bit integer and 9-bit fractional bits. (Default: 0x0200h)

| Operation | Description |
|---|---|
| K11' = K11*Y | U(12,10) * S(9,0) = S(21,10) truncating to S(15,4) |
| K13' = K13*V | U(11,10) * S(8,0) = S(19,10) truncating to S(14,4) |
| R'' = K11'+K13' | S(15,4) + S(14,4) = S(15,4) |
| R' = R'' + Roffset | S(15,4) + S(14,4) = S(15,4) truncating to S(13,2) |
| K22' = K22*U | U(10,10) * S(9,0) = S(19,10) truncating to S(13,4) |
| K23' = K23*V | U(10,10) * S(8,0) = S(18,10) truncating to S(13,4) |
| G'' = K11'-K22'-K23' | S(15,4)+ S(13,4)+ S(13,4) = S(15,4) |
| G' = G'' + Goffset | S(15,4) + S(14,4) = S(15,4) truncating to S(13,2) |
| K32' = K32*U | U(12,10) * S(8,0) = S(20,10) truncating to S(15,4) |
| B'' = K11'+K32' | S(15,4)+ S(15,4) = S(15,4) |
| B' = B'' + Boffset | S(15,4) + S(14,4) = S(15,4) truncating to S(13,2) |
| R=Rgain*R' | U(10,9)*S(13,2)=S(23,11) rounding to U(10,0) (clamp) |
| G=Ggain*G' | U(10,9)*S(13,2)=S(23,11) rounding to U(10,0) (clamp) |
| B=Bgain*B' | U(10,9)*S(13,2)=S(23,11) rounding to U(10,0) (clamp) |

**Address 0xC2~0xC6 are 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**<br>0: Disable<br>1: Enable | |
| Y_FORMULA | 6 | R/W | 0 | **Y_FORMULA**<br>0: Y = (2R+5G+B)/8<br>1: Y = (5R+8G+3B)/16 | |
| SC_EN | 5 | R/W | 0 | **SOFT_CLAMP**<br>0: Disable | |

| Name | Bits | R/W | Default | Comments | Config |
|---|---|---|---|---|---|
| | | | | 1: Enable | |
| DCC_MODE | 4 | R/W | 0 | **DCC_MODE**<br> 0: Auto Mode<br> 1: Manual Mode | |
| SCG_EN | 3 | R/W | 0 | **SCENE_CHANGE**<br> 0: Disable Scene-Change Function<br> 1: Enable Scene-Change Function in Auto Mode | |
| BWL_EXP | 2 | R/W | 0 | **BWL_EXP**<br> 0: Disable Black/White Level Expansion<br> 1: Enable Black/White Level Expansion in Auto Mode | |
| PAGE_SEL | 1:0 | R/W | 0 | **DCC_PAGE_SEL**<br> 00: Page 0 (for Histogram / Ymin-max / Soft-Clamping / Scene-Change)<br> 01: Page 1 (for Y-Curve / WBL Expansion)<br> 10: Page 2 (for Calculation Parameter)<br> 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**<br> 0: Disable<br> 1: Enable<br> 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**<br> 0: Disable<br> 1: Enable | |
| BLD_MODE | 4 | R/W | 0 | **Blending Factor Control Mode**<br> 0: old mode<br> 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;`<br><br>`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;<br>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;<br>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**<br> 0: Disable Black-Background Exception<br> 1: Enable Black-Background Exception | |
| Reserved | 6:4 | -- | -- | Reserved | |
| BBE_THD | 3:0 | R/W | 0x4 | **BBE_THD**<br>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**<br> 0: Disable Histogram Noise Filter<br> 1: Enable Histogram Noise Filter | |
| HNFLT_THD | 6:4 | R/W | 0 | **HNFLT_THD**<br> Threshold for Histogram Noise Filter | |
| YNFLT_EN | 3 | R/W | 0 | **YNFLT_ENA**<br> 0: Disable Ymax / Ymin Noise Filter<br> 1: Enable Ymax / Ymin Noise Filter | |
| YNFLT_THD | 2:0 | R/W | 0 | **YNFLT_THD**<br>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**<br>0: Disable RH0 Limiter<br>1: Enable RH0 Limiter | |
| RH1_LIMITER | 6 | R/W | 0 | **RH1_LIMITER**<br>0: Disable RH1 Limiter<br>1: Enable RH1 Limiter | |
| REAL_MA_LEN | 5:3 | R | -- | **Real MOV_AVG_LEN**<br>may be different with MOV_AVG_LEN, if SCG enable | |
| MOV_AVG_LEN | 2:0 | R/W | 0 | **MOV_AVG_LEN**<br>000: Histogram Moving Average Length = 1<br>001: Histogram Moving Average Length = 2<br>010: Histogram Moving Average Length = 4<br>011: Histogram Moving Average Length = 8 | |

| | | | | 100: Histogram Moving Average Length = 16<br>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**<br> 0: old mode (2553V)<br> 1: new mode (2622) | |
| Reserved | 6:5 | -- | -- | Reserved | |
| SCG_PERIOD | 4:0 | R/W | 0x10 | **Scene-Change Mode Period = 1~32.**<br> Note: SCG_PERIOD >= MOV_AVG_LEN,<br>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=DIFF[10:0]>>3 | |

| 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]**<br>**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**<br>00: Yi = 512 (Z8)<br>01: Yi = 576 (Z9)<br>10: Yi = 640 (Z10)<br>11: Yi = 704 (Z11) | |
| WL_BLD_VAL | 5:4 | R/W | 0x00 | **White-Level Blending Factor**<br>00: 0 (user-defined curve)<br>01: R/2<br>10: R<br>11: 2R | |
| BL_RANGE | 3:2 | R/W | 0x00 | **Black-Level Range**<br>00: Yi = 448 (Z7)<br>01: Yi = 384 (Z6)<br>10: Yi = 320 (Z5)<br>11: Yi = 256 (Z4) | |
| BL_BLD_VAL | 1:0 | R/W | 0x00 | **Black-Level Blending Factor**<br>00: 0 (user-defined curve)<br>01: R/2<br>10: R<br>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**<br>**Addr 01: SRAM_01** | |

| | | | | .... <br><br> **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** <br> 0: disable <br> 1: enable | |
| RAM_Mode | 6 | R/W | 0 | **RAM_Mode** <br> 0: dclk domain mode (normal mode, BIST) <br> 1: MCU domain mode (SCG test) | |
| Reserved | 5:2 | -- | -- | Reserved | |
| BIST_PERIOD | 1 | R | -- | **BIST_Period** <br> 0: BIST is done <br> 1: BIST is running | |
| BIST_OK | 0 | R | -- | **BIST_OK** <br> 0: SRAM fail <br> 1: SRAM ok | |

## ICM (Page 7)

| Address: D0 | ICM Control | Default: 00h |
|---|---|---|

| Bit | Mode | Function |
|---|---|---|
| 7 | R/W | **ICM Enable** <br> 0: Disable <br> 1: Enable |
| 6 | R/W | **Y Correction Mode** <br> 0: dY = (8dU+dV)/8 <br> 1: dY = (6dU+dV)/8 |
| 5 | R/W | **ICM U/V Delta Range:** <br> 0: Original -128~+127 <br> 1: Double -256~254 |
| 4 | R/W | **CM0 Enable** <br> 0: Disable <br> 1: Enable |
| 3 | R/W | **CM1 Enable** <br> 0: Disable <br> 1: Enable |
| 2 | R/W | **CM2 Enable** <br> 0: Disable <br> 1: Enable |
| 1 | R/W | **CM3 Enable** <br> 0: Disable <br> 1: Enable |
| 0 | R/W | **CM4 Enable** <br> 0: Disable <br> 1: Enable |

| Address: D1 | ICM_SEL | Default: 00h |
|---|---|---|

| Bit | Mode | Function |
|---|---|---|
| 7:5 | R/W | **ICM Test Mode** <br> 000: disable <br> 001: bypass U, V result <br> 010: bypass hue/saturation result <br> 011: bypass dU, dV value |

| Bit | Mode | Function |
|---|---|---|
| | | 1xx: R,B as LUT input, and bypass LUT output to R/G/B output |
| 4 | -- | reserved |
| 3 | R/W | CM5 Enable<br>0: Disbale<br>1: Enable |
| 2:0 | R/W | CM Select<br>000: Select Chroma Modifier 0 for Accessing Through Data Port<br>001: Select Chroma Modifier 1 for Accessing Through Data Port<br>010: Select Chroma Modifier 2 for Accessing Through Data Port<br>011: Select Chroma Modifier 3 for Accessing Through Data Port<br>100: Select Chroma Modifier 4 for Accessing Through Data Port<br>101: Select Chroma Modifier 5 for Accessing Through Data Port<br>110~111: reserved |

**Address: D2    ICM_ADDR**    Default: 00h

| Bit | Mode | Function |
|---|---|---|
| 7:0 | 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<br>00: CM[N] left width = 64<br>01: CM[N] left width = 128<br>10: CM[N] left width = 256<br>11: CM[N] left width = 512 |
| 5:4 | W | CM[N]_LBUF<br>00: CM[N] left Buffer = 0<br>01: CM[N] left Buffer = 64<br>10: CM[N] left Buffer = 128<br>11: CM[N] left Buffer = 256 |
| 3:2 | W | CM[N]_RWID<br>00: CM[N] right width = 64<br>01: CM[N] right width = 128<br>10: CM[N] right width = 256<br>11: CM[N] right width = 512 |
| 1:0 | W | CM[N]_RBUF<br>00: CM[N] right Buffer = 0<br>01: CM[N] right Buffer = 64<br>10: CM[N] right Buffer = 128<br>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<br>Addr 04: V Offset 00, -128~127<br>Addr 05: U Offset 01, -128~127<br>Addr 06: V Offset 01, -128~127<br>Addr 07: U Offset 02, -128~127<br>Addr 08: V Offset 02, -128~127<br>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** |

## Y-Peaking filter and coring control (For Display Domain) (Page 7)

**Address: D6**    peaking/coring access port control                                    Default: 00h

| Bit | Mode | Function |
|-----|------|----------|
| 7 | R/W | **Enable peaking / coring access port** |
| 6 | R/W | **Peaking/coring Enable**<br><br>0: Disable<br><br>1: Enable |
| 5 | R/W | **Y peaking Coefficient Resolution**<br><br>0: n/32<br><br>1: n/64 |

| Bit | Mode | Function |
|-----|------|----------|
| 4:3 | -- | **Reserved** |
| 2:0 | R/W | **Peaking/coring port address** |

**Address: D7-00  Peaking_Coef0**

| Bit | Mode | Function |
|-----|------|----------|
| 7:0 | R/W | **Coefficient C0 of Peaking filter:**<br>  Valid Range: -128/32(-128) ~ 127/32 (127) (2's complement) |

**Address: D7-01  Peaking_Coef1**

| Bit | Mode | Function |
|-----|------|----------|
| 7:0 | R/W | **Coefficient C1 of Peaking filter:**<br>  Valid Range: -128/32(-128) ~ 127/32 (127) (2's complement) |

**Address: D7-02  Peaking_Coef2**

| Bit | Mode | Function |
|-----|------|----------|
| 7:0 | R/W | **Coefficient C2 of Peaking filter:**<br>  Valid Range: -128/32(-128) ~ 127/32 (127) (2's complement) |

**Address: D7-03  Coring_Min**

| Bit | Mode | Function |
|-----|------|----------|
| 7:5 | R/W | **Reserved** |
| 4:0 | R/W | **Coring Minimum value** |

**Address: D7-04  Coring_Max_Pos**

| Bit | Mode | Function |
|-----|------|----------|
| 7:0 | R/W | **Coring Maximum Positive value** |

**Address: D7-05  Coring_Max_Neg**

| Bit | Mode | Function |
|-----|------|----------|
| 7:0 | R/W | **Coring Maximum Negitive value**   (2's complement) |

$Y'[n] = C0*Y[n] + C1*(Y[n-1]+Y[n+1]) + C2*(Y[n-2]+Y[n+2])$ , $-256 <= Y' <= 255$

$Ypeak = Y'[n] - Coring\_Min$,   if $Y'[n] >= 0$,

$\quad\quad = Y'[n] + Coring\_Min$,   if $Y'[n] < 0$

if ( $|Y'[n]|$  <= Coring_Min )

$\quad\quad Y''[n] = 0$,

else if Ypeak >= Coring_Max_Pos

$\quad\quad Y''[n] = Coring\_Max\_Pos$

else if Ypeak <= Coring_Max_Neg

$\quad\quad Y''[n] = Coring\_Max\_Neg$

else

$\quad\quad Y''[n] = Ypeak$

$Yo[n] = Y[n] + Y''[n]$, $0 <= Yo[n] <= 255$

# 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.

| 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] | |

<br>

| 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] | |

<br>

| 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] | |

<br>

| 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. | |

<br>

| 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. | |

<br>

| 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. | |
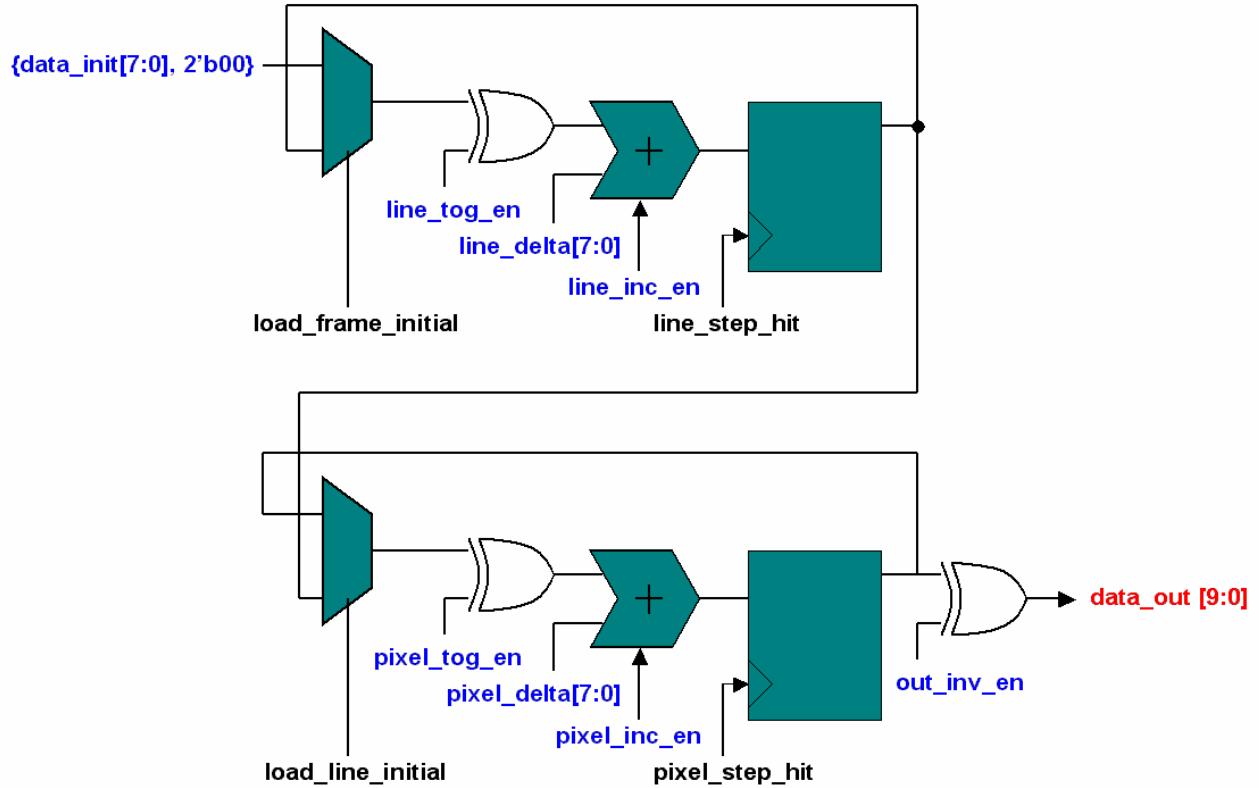
## Pattern Generator in Display Domain (Page 7)

RTD3182 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 pixel-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 | |

| Name | Bits | R/W | Default | Comments | Config |
|---|---|---|---|---|---|
| LINE_G_TOG_EN | 3 | R/W | 0 | Data toggled in each pixel-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 pixel-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 ).**

**Reserved (Page 8)**

**Reserved (Page 9)**

**Reserved (Page A)**

**Reserved (Page B)**

**Reserved (Page C)**

## Register 1(page D)

**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<br>1. Select M2PLL as clock source, but M2PLL power down, power saving or output disable, clear this bit to disable the interrupt | Rport Wport |
| CEC_IRQ_EVENT | 5 | R/W | 0 | **CEC Event Status**<br>1. IF CEC func IRQ event occurred since the last status cleared | Rport Wport |
| SCA_IRQ_EVENT | 4 | R/W | 0 | **Scalar-related Event Status**<br>1. IF Scalar integrated IRQ event occurred since the last status cleared | Rport Wport |
| Reserved | 3:1 | -- | 0 | Reserved to 0 | |
| DDC_IRQ_EVENT | 0 | R/W | 0 | **DDC Event   Status**<br>1: If the DDC IRQ event occurred since the last status cleared | Rport Wport |

| Register:: REV_DUMMY1 | | | | 0xFF02 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| REV_DUMMY1 | 7:0 | R/W | 00 | Dummy1 | |

**DDC**

RTD3580 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.
- 

Besides, the DDC_RAM1, DDC_RAM2, 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, , DVI DDC. The XSRAM for MCU is 512 bytes and ADC DDC/HDMI DDC is used with 256 bytes.

   The DDC of RTD3580 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 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 = 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 = 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 |

| Name | Bits | R/W | Default | Comments | Config |
|---|---|---|---|---|---|
| A_DDCRAM_W_EN | 2 | R/W | 0 | ADC DDC SRAM Write Enable (for external DDC access only)<br>0: Disable<br>1: Enable | |
| A_DBN_EN | 1 | R/W | 1 | ADC DDC De-bounce Enable<br>0: Disable<br>1: Enable (with crystal/4) | |
| A_DDC_EN | 0 | R/W | 0 | ADC DDC Channel Enable Bit<br>0: MCU access Enable<br>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<br>00: 1/1 reference clock<br>01: 1/2 reference clock<br>1X: 1/4 reference clock | |
| A_STOP_DBN_SEL | 5:4 | R/W | 0 | De-bounce sda stage<br>0X: latch one stage<br>10: latch two stage<br>11: latch three stage | |
| A_SYS_CK_SEL | 3 | R/W | 0 | De-bounce reference clock<br>0: crystal clock<br>1. Serial flash clock (M2PLL / Flash_DIV) | |
| A_DDC2 | 2 | R/W | 0 | Force to ADC DDC to DDC2 mode<br>0: Normal operation<br>1: DDC2 is active | |
| RST_A_DDC | 1 | R/W | 0 | Reset ADC DDC circuit<br>0: Normal operation<br>1: reset (auto cleared) | Rport<br><br>wport |
| RVT_A_DDC1_EN | 0 | R/W | 0 | ADC DDC revert to DDC1 enable(SCL idle for 128 VSYNC)<br>0: Disable<br>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<br>0: Disable<br>1: Enable | |
| Reserved | 6:2 | -- | -- | Reserved | |
| A_SEG_WR | 1 | R/W | 0 | ADC DDC Segment Write Status<br>0: no external write after clear<br>1: new external write after clear<br>It is cleared after write | Wport<br>Rport |
| A_FORCE_SCL_L | 0 | R/W | 0 | Force external SCL bus low<br>1: Driving SCL = 0 after external SCL = 0<br>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<br>(The default DDC channel address MSB 4 Bits is "A") | |
| D_SCL_DBN_SEL | 4 | R/W | 0 | SCL Debounce Clock Selection<br>0: De-bounce clock (after clock divider)<br>1: De-bounce reference clock | |
| D_DDC_W_STA | 3 | R/W | 0 | DVI DDC External Write Status (for external DDC access only)<br>It is cleared after write. | Wport<br><br>rport |
| D_DDCRAM_W_EN | 2 | R/W | 0 | DVI DDC External Write Enable (for external DDC access only)<br>0: Disable<br>1: Enable | |
| D_DBN_EN | 1 | R/W | 1 | DVI DDC Debounce Enable<br>0: Disable<br>1: Enable (with crystal/4) | |
| D_DDC_EN | 0 | R/W | 0 | DVI DDC Channel Enable Switch<br>0: MCU access Enable<br>1: External DDC access Enable | |

| Register::DVI_DDC_control_1 | | | | 0xFF1F | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |

| Name | Bits | R/W | Default | Comments | Config |
|---|---|---|---|---|---|
| D_DBN_CLK_SEL | 7:6 | R/W | 0 | De-bounce clock divider<br>00: 1/1 reference clock<br>01: 1/2 reference clock<br>1X: 1/4 reference clock | |
| D_STOP_DBN_SEL | 5:4 | R/W | 0 | De-bounce sda stage<br>0X: latch one stage<br>10: latch two stage<br>11: latch three stage | |
| D_SYS_CK_SEL | 3 | R/W | 0 | De-bounce reference clock<br>0: crystal clock<br>1. Serial flash clock (M2PLL / Flash_DIV) | |
| D_DDC2 | 2 | R/W | 0 | Force to DVI DDC to DDC2 mode<br>0: Normal operation<br>1: DDC2 is active | |
| RST_D_DDC | 1 | R/W | 0 | Reset DVI DDC circuit<br>0: Normal operation<br>1: reset (auto cleared) | Rport<br><br>wport |
| RVT_D_DDC1_EN | 0 | R/W | 0 | DVI DDC revert to DDC1 enable(SCL idle for 128 VSYNC)<br>0: Disable<br>1: Enable | |

| Register::DVI_DDC_control_2 | | | | 0xFF20 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| Reserved | 7:2 | -- | -- | Reserved | |
| D_SEG_WR | 1 | R/W | 0 | DVI DDC Segment Write Status<br>0: no external write after clear<br>1: new external write after clear<br>It is cleared after write | Wport<br>Rport |
| D_FORCE_SCL_L | 0 | R/W | 0 | Force external SCL bus low<br>1: Driving SCL = 0 after external SCL = 0<br>0: Release SCL | |

| Register::DDCRAM_partition | | | | 0xFF21 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| Reserved | 7:6 | - | | | |
| ADDCRAM_ST | 5:4 | R/W | 0x3 | ADDC RAM Start Address is 0x00 +<br>ADDCRAM_ST*0x80, ADDCRAM SIZE = | |

| | | | | DDDCRAM_ST – ADDCRAM_ST | |
|---|---|---|---|---|---|
| DDDCRAM_ST | 3:2 | R/W | 0x3 | DDDC RAM Start Address is 0x80 + DDDCRAM_ST*0x80,   DDDCRAM SIZE = HDDCRAM_ST – DDDCRAM_ST | |
| Reserved | 1:0 | | | Reserved | |

| Register::VSYNC_Sel | | | | | 0xFF22 |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| Reserved | 7:4 | - | | | |
| VS_CON1 | 3:2 | R/W | 0 | 00: VSYNC1 signal is connected to ADC DDC<br>01: VSYNC1 signal is connected to DVI DDC<br>1x: VSYNC1 signal is not connected | |
| Reserved | 1:0 | -- | 0 | Reserved | |

**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<br> 0: from ADC DDC<br> 1: from DVI DDC | |

| 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<br><br>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<br><br>Wport |
| DDC_128VS1_I | 5 | R/W | 0 | In DDC2 Transition mode, SCL idle for 128 VSYNC. Write 0 to clear. | Rport<br><br>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<br><br>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 (FF25) to clear. | |
| D_IN_I | 2 | R | 0 | If IIC_DATA_IN latched, this bit is set to "1" . Read IIC_data_in (FF24) to clear. | |
| SUB_I | 1 | R/W | 0 | If IIC_SUB latched, this bit is set to "1" Write 0 to clear. | Rport<br><br>Wport |
| SLV_I | 0 | R/W | 0 | If IIC_SLAVE latched, this bit is set to "1" Write 0 to clear. | Rport<br><br>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<br>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<br>1: Enable the D_WR_I signal as an interrupt source | |
| DDC_128VSI1_EN | 5 | R/W | 0 | 0: Disable the 128VS1_I signal as an interrupt source<br>1: Enable the 128VS1_I signal as an interrupt source | |

| Name | Bits | R/W | Default | Comments | Config |
|---|---|---|---|---|---|
| STOPI_EN | 4 | R/W | 0 | 0: Disable the STOP_I signal as an interrupt source<br>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<br>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<br>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<br>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<br>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:<br>1. IIC_BUF_FULL = 1 in write mode<br>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:<br>IIC_BUF_FULL = 1 in write mode | |
| IIC_BUF_OV | 5 | R/W | 0 | IIC_DATA_BUFFER Overflow. Write '0' to clear | Rport<br><br>Wport |
| IIC_BUF_UN | 4 | R/W | 0 | IIC_DATA_BUFFER Underflow. Write '0' to clear | Rport<br><br>Wport |
| DDC_128VS2_I | 3 | R/W | 0 | In DDC2 Transition mode, SCL idle for 128 VSYNC. Write 0 to clear. Write '0' to clear | Rport<br><br>Wport |
| IIC_BUF_FULL | 2 | R | 0 | IIC_DATA_BUFFER Full<br>If IIC_DATA buffer is full, this bit is set to "1". (On-line monitor)<br>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. | |
|---|---|---|---|---|---|
| Reserved | 0 | R/W | 0 | Reserved | 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 | 4:3 | R/W | 0 | Reserved | |
| 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 | |
| Reserved | 0 | | | Reserved | |

| Register::IIC_channel_control | | | | 0xFF2B | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |

| Reserved | 7:2 | -- | 0 | Reserved | |
|---|---|---|---|---|---|
| RLS_SCL_SU | 1 | R/W | 0 | Set IIC data Setup Time When holding SCL low<br><br>0: Use Delay Chain (~5ns)<br><br>1: Use Crystal Clock to increase data setup time relative to SCL clock line | |
| Reserved | 0 | -- | 0 | Reserved | |

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<br>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<br>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<br>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<br>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<br><br>(= 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 |

**PWM**

RTD3580 supports 6 channels of PWM DAC. The resolution of each PWM is 12-bit. PWM0, PWM1, PWM2, are connected to DA0, DA1, DA2, 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.
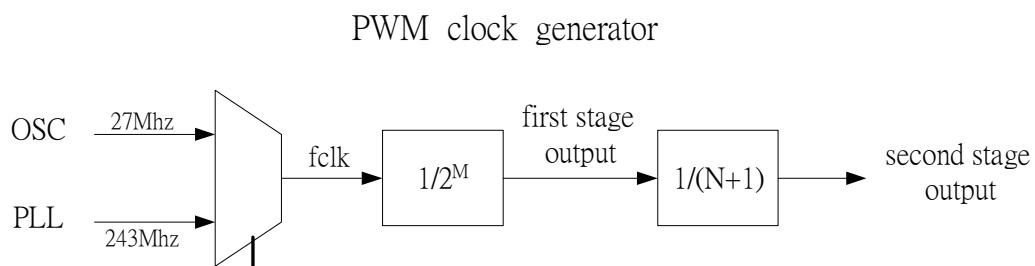
The PWM frequency is :

$$Fpwm = fclk / 2^M / (N+1) / 4096$$

The PWM frequency range is :

fclk=27M, fpwm = 6.6KHz ~ 0.2Hz

fclk=243M, fpwm = 60KHz ~ 1.8Hz

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 | |
| Reserved | 5:3 | | | Reserved | |
| PWM2_CK_SEL | 2 | R/W | 0 | PWMx clock generator input source<br>0: Crystal<br>1: PLL output | |
| PWM1_CK_SEL | 1 | R/W | 0 | PWMx clock generator input source<br>0: Crystal<br>1: PLL output | |
| PWM0_CK_SEL | 0 | R/W | 0 | PWMx clock generator input source<br>0: Crystal<br>1: PLL output | |

| Register::PWM03_M | | | | 0xFF3B | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| Reserved | 7:6 | | | Reserved | |
| 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 | |
| Reserved | 3:0 | | | Reserved | |

| 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** |
| Reserved | 7:4 | | | Reserved | |
| 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::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<br>0: Setting active after PWM_W_DB_WR = 1<br>1: Setting active after PWM_W_DB_WR = 1 & DVS. | |
| Reserved | 5:3 | | | Reserved | |
| PWM2L | 2 | R/W | 0 | 0: enable Active H<br>1: enable Active L | |
| PWM1L | 1 | R/W | 0 | 0: enable Active H<br>1: enable Active L | |
| PWM0L | 0 | R/W | 0 | 0: enable Active H<br>1: enable Active L | |

| Register::PWM_VS_CTRL | | | | | 0xFF47 |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |

| Name | Bits | R/W | Default | Comments | Config |
|---|---|---|---|---|---|
| PWM_VS_CTRL_DUM | 7:6 | R/W | 0 | dummy | |
| Reserved | 5:3 | | | Reserved | |
| PWM2_VS_RST_EN | 2 | R/W | 0 | 0: Disable<br>1: Enable PWM2 reset by DVS | |
| PWM1_VS_RST_EN | 1 | R/W | 0 | 0: Disable<br>1: Enable PWM1 reset by DVS | |
| PWM0_VS_RST_EN | 0 | R/W | 0 | 0: Disable<br>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<br>1: PWM Width setting double-buffered enable | |
| PWM_WIDTH_SEL | 6 | R/W | 0 | 0: PWMxL_DUT is active<br>1: PWMxL_DUT is inactive, forced to 4'h0 internally | |
| Reserved | 5:3 | | | Reserved | |
| PWM2_EN | 2 | R/W | 0 | 0: PWM output disable<br>1: PWM output enable | |
| PWM1_EN | 1 | R/W | 0 | 0: PWM output disable<br>1: PWM output enable | |
| PWM0_EN | 0 | R/W | 0 | 0: PWM output disable<br>1: PWM output enable | |

| Register::PWM_CK | | | | 0xFF49 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| PWM_CK_DUMMY | 7:6 | R/W | 0 | Dummy | |
| Reserved | 5:3 | | | Reserved | |
| PWM2_CK | 2 | R/W | 0 | 0: Select first stage output<br>1: Select second stage output | |
| PWM1_CK | 1 | R/W | 0 | 0: Select first stage output<br>1: Select second stage output | |
| PWM0_CK | 0 | R/W | 0 | 0: Select first stage output<br>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<br>When write the MSB bit (PWM_W_DB_EN=0) , the 12-bit | **RPORT** |

| data will be loaded. | | | | | **WPORT** |
|---|---|---|---|---|---|

| **Register::PWM1H_DUT** | | | | | 0xFF4B |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| PWM1H_DUT | 7:0 | R/W | 0 | PWM1[11:4] duty width<br>When write the MSB bit (PWM_W_DB_EN=0) , the 12-bit data will be loaded. | **RPORT**<br><br>**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**<br><br>**WPORT** |
| PWM0L_DUT | 3:0 | R/W | 0 | PWM0[3:0] duty width | **RPORT**<br><br>**WPORT** |

| **Register::PWM2H_DUT** | | | | | 0xFF4D |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| PWM2H_DUT | 7:0 | R/W | 0 | PWM2[11:4] duty width<br>When write the MSB bit (PWM_W_DB_EN=0) , the 12-bit data will be loaded. | **RPORT**<br><br>**WPORT** |

| **Register::PWM23L_DUT** | | | | | 0xFF4F |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| Reserved | 7:4 | | | Reserved | |
| PWM2L_DUT | 3:0 | R/W | 0 | PWM2[3:0] duty width | **RPORT**<br><br>**WPORT** |

| **Register:: REV_DUMMY3** | | | | 0xFF53 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| REV_DUMMY3 | 7:0 | R/W | 00 | Dummy3 | |

## Register 2(page E)

0xFF76~0xFF93 Reserved

### Pin-Share

| Register:: PIN_SHARE_CTRL00 | | | | 0xFF94 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| VIDEO8_SEL | 7 | R/W | 0x0 | **Video8 Source Select**<br>0: Pin47~48, Pin1~7 (QFN48) or<br>    Pin62~64, Pin1~6 (QFN64) <default><br>1: Pin59, Pin62~63, Pin1~6 (QFN64) | |
| Reserved | 6:3 | R/W | 0x00 | reserved to 0 | |
| SDRAM_en | 2 | R/W | 0x00 | **SDRAM enable**<br>0: no SDRAM <default><br>1: MCM SDRAM | |
| SDRAM_size | 1 | R/W | 0x0 | **SDRAM size**<br>0: 1Mx16 SDRAM <default><br>1: 1Mx32 SDRAM | |
| Package_Type | 0 | R/W | 0x0 | **Package Type**<br>0: 48pin package<default><br>1: 64pin package | |

| Register:: PIN_SHARE_CTRL01 | | | | 0xFF95 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/ Write** | **Reset State** | **Comments** | **Config** |
| DQM3 | 7 | R/W | 0x0 | **SDR DQM3 (UDQM) – Data Input/Output Mask**<br>0: non-active<br>1: active | |
| DQM2 | 6 | R/W | 0x0 | **SDR DQM2 (LDQM) – Data Input/Output Mask**<br>0: non-active<br>1: active | |
| DQM1 | 5 | R/W | 0x0 | **SDR DQM1 – Data Input/Output Mask**<br>0: non-active<br>1: active | |
| DQM0 | 4 | R/W | 0x0 | **SDR DQM0 – Data Input/Output Mask**<br>0: non-active<br>1: active | |
| DDCSCL1 | 3:2 | R/W | 0x0 | **Pin44 (48pin) / Pin58 (64pin) (PAD_DDCSCL1)**<br>00: DDCSCL <IO> <open-drain> <default><br>01: PWM0 <O><br>10: TCON7 <O><br>11: AUX-CH_P0 | |
| DDCSDA1 | 1:0 | R/W | 0x0 | **Pin43 (48pin) / Pin57 (64pin) (PAD_DDCSDA1)**<br>00: DDCSDA <IO> <open-drain> <default><br>01: PWM1 <O><br>10: TCON9 <O><br>11: AUX-CH_N0/IRQ<br>Effectively only if CRFF95[3:2] != 2'b11,when | |

| | | | | CRFF95[3:2]=11,output is AUX-CH_N0 | |
|---|---|---|---|---|---|

| Register:: PIN_SHARE_CTRL01 | | | 0xFF96 | | |
|---|---|---|---|---|---|
| Name | Bits | Read/Write | Reset State | Comments | Config |
| DDCSDA2 | 7:6 | R/W | 0x0 | **Pin56 (64pin) (PAD_DDCSDA2)**<br>00: DDCSDA <IO> <open-drain> <default><br>01: MCK <O><br>10: TCON11 <O><br>11: AUX-CH_N1 | |
| AUD_HOUTL | 5:4 | R/W | 0x0 | **Pin26 (64pin)**<br>00: AUDIO (HOUTL) <default><br>01: TCON0 <O><br>10: PWM0 <O> | |
| AUD_HOUTR | 3 | R/W | 0x0 | Pin27-28 (64pin)<br>0: AUDIO (HOUTR, REF) <default><br>1: TCON7, TCON5 <O> | |
| DDCSDA2 | 2:0 | R/W | 0x0 | **Pin55 (64pin) (PAD_DDCSCL2)**<br>000: DDCSCL <IO> <open-drain> <default><br>001: SCK <O><br>010: TCON8 <O><br>011: AUX-CH_P1<br>100: IRQ <O> | |

| Register:: PIN_SHARE_CTRL03 | | | 0xFF97 | | |
|---|---|---|---|---|---|
| Name | Bits | Read/Write | Reset State | Comments | Config |
| Reserved | 7:6 | R/W | 0x0 | reserved to 0 | |
| SPDIF0 | 5:4 | R/W | 0x0 | **Pin59 (64pin) (PAD_SPDIF0)**<br>00: TCON10 <O> <default><br>01: PWM1 <O><br>10: SD0 <O><br>11: SPDIF0 <O><br>Effectively only if CRFF96[7] = 1'b0 | |
| SCL | 3:2 | R/W | 0x0 | **Pin42 (48pin) / Pin54 (64pin) (PAD_SCL)**<br>00: TCON13 <O> <default><br>01: PWM2 <O><br>10: WS <O><br>11: SPDIF0 <O><br>Effectively only in 1-wire host interface condition | |
| CEC | 1:0 | R/W | 0x0 | **Pin30 (64pin) (PAD_CEC)**<br>00: CEC <O> <default><br>01: PWM1 <O><br>10: TCON3 <O><br>11: Reserved | |

| Register:: PIN_DRIVING_CTRL10 | | | 0xFF98 | | |
|---|---|---|---|---|---|
| Name | Bits | Read/Write | Reset State | Comments | Config |
| E2CTRL10_7 | 7 | R/W | 0 | **Driving Current Control – Pin26~28 (64pin)**<br>0: Low<br>1: High | |
| E2CTRL10_6 | 6 | R/W | 0 | reserved to 0 | |
| E2CTRL10_5_4 | 5:4 | R/W | 0x2 | **Driving Current Control – Pin21~38 (48pin) /<br>Pin33~50 (64pin)** | |

| Name | Bits | Read/Write | Reset State | Comments | Config |
|---|---|---|---|---|---|
| | | | | - TTL<br>  x0: 8mA<br>  x1: 12mA<br>- LVDS<br>  00: 2.5mA<br>  01: 3.0mA<br>  10: 3.5mA<br>  11: 4.0mA | |
| E2CTRL10_3 | 3 | R/W | 0 | ~~Schmitt Trigger Control – Pin30 (64pin)~~<br>~~0: On~~<br>~~1: Off~~<br>reserved to 0 | |
| E2CTRL10_2 | 2 | R/W | 0 | ~~Slew Rate Control – Pin30 (64pin)~~<br>~~0: Fast~~<br>~~1: Slow~~<br>reserved to 0 | |
| E2CTRL10_1 | 1 | R/W | 0 | Driving Current Control – Pin30 (64pin)<br>0: Low<br>1: High | |
| E2CTRL10_0 | 0 | R/W | 0 | reserved to 0 | |

| Register:: PIN_DRIVING_CTRL11 | | | 0xFF99 | | |
|---|---|---|---|---|---|
| Name | Bits | Read/Write | Reset State | Comments | Config |
| E2CTRL11_7 | 7 | R/W | 0 | Schmitt Trigger Control – Pin41~42 (48pin) /<br>Pin53~54 (64pin)<br>0: On<br>1: Off | |
| E2CTRL11_6 | 6 | R/W | 0 | Slew Rate Control – Pin41~42 (48pin) /<br>Pin53~54 (64pin)<br>0: Fast<br>1: Slow | |
| E2CTRL11_5 | 5 | R/W | 0 | Driving Current Control – Pin41~42 (48pin) /<br>Pin53~54 (64pin)<br>0: Low<br>1: High | |
| E2CTRL11_4 | 4 | R/W | 0 | reserved to 0 | |
| E2CTRL11_3 | 3 | R/W | 0 | ~~Schmitt Trigger Control – Pin55~56 (64pin)~~<br>~~0: On~~<br>~~1: Off~~<br>reserved to 0 | |
| E2CTRL11_2 | 2 | R/W | 0 | ~~Slew Rate Control – Pin55~56 (64pin)~~<br>~~0: Fast~~<br>~~1: Slow~~<br>reserved to 0 | |
| E2CTRL11_1 | 1 | R/W | 0 | Driving Current Control – Pin55~56 (64pin)<br>0: Low<br>1: High | |
| E2CTRL11_0 | 0 | R/W | 0 | reserved to 0 | |

| Register:: PIN_DRIVING_CTRL12 | | | 0xFF9A | | |
|---|---|---|---|---|---|
| Name | Bits | Read/Write | Reset State | Comments | Config |
| E2CTRL12_7 | 7 | R/W | 0 | ~~Schmitt Trigger Control – Pin43~44 (48pin) /~~<br>~~Pin57~58 (64pin)~~<br>~~0: On~~<br>~~1: Off~~<br>reserved to 0 | |
| E2CTRL12_6 | 6 | R/W | 0 | ~~Slew Rate Control – Pin43~44 (48pin) /~~ | |

| Name | Bits | Read/Write | Reset State | Comments | Config |
|------|------|-----------|-------------|----------|--------|
| | | | | ~~Pin57~58 (64pin)~~<br>~~0: Fast~~<br>~~1: Slow~~<br>reserved to 0 | |
| E2CTRL12_5 | 5 | R/W | 0 | **Driving Current Control – Pin43~44 (48pin) / Pin57~58 (64pin)**<br>0: Low<br>1: High | |
| E2CTRL12_4 | 4 | R/W | 0 | reserved to 0 | |
| E2CTRL12_3 | 3 | R/W | 0 | **Schmitt Trigger Control – Pin59 (64pin)**<br>0: On<br>1: Off | |
| E2CTRL12_2 | 2 | R/W | 0 | **Slew Rate Control – Pin59 (64pin)**<br>0: Fast<br>1: Slow | |
| E2CTRL12_1 | 1 | R/W | 0 | **Driving Current Control – Pin59 (64pin)**<br>0: Low<br>1: High | |
| E2CTRL12_0 | 0 | R/W | 0 | reserved to 0 | |

| Register:: PIN_DRIVING_CTRL13 | | | 0xFF9B | | |
|------|------|-----------|-------------|----------|--------|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Comments** | **Config** |
| E2CTRL13_7 | 7 | R/W | 0 | **Schmitt Trigger Control – SDR CLK**<br>0: On<br>1: Off | |
| E2CTRL13_6 | 6 | R/W | 0 | **Slew Rate Control – SDR CLK**<br>0: Fast<br>1: Slow | |
| E2CTRL13_5 | 5 | R/W | 0 | **Driving Current Control – SDR CLK**<br>0: Low<br>1: High | |
| E2CTRL13_4 | 4 | R/W | 0 | reserved to 0 | |
| E2CTRL13_3 | 3 | R/W | 0 | **Schmitt Trigger Control – SDR Control**<br>0: On<br>1: Off | |
| E2CTRL13_2 | 2 | R/W | 0 | **Slew Rate Control – SDR Control**<br>0: Fast<br>1: Slow | |
| E2CTRL13_1 | 1 | R/W | 0 | **Driving Current Control – SDR Control**<br>0: Low<br>1: High | |
| E2CTRL13_0 | 0 | R/W | 0 | reserved to 0 | |

| Register:: PIN_DRIVING_CTRL14 | | | 0xFF9C | | |
|------|------|-----------|-------------|----------|--------|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Comments** | **Config** |
| E2CTRL14_7 | 7 | R/W | 0 | **Schmitt Trigger Control – DQ31~24 (DQ8~11)**<br>0: On<br>1: Off | |
| E2CTRL14_6 | 6 | R/W | 0 | **Slew Rate Control – DQ31~24 (DQ8~11)**<br>0: Fast<br>1: Slow | |
| E2CTRL14_5 | 5 | R/W | 0 | **Driving Current Control – DQ31~24 (DQ8~11)**<br>0: Low<br>1: High | |
| E2CTRL14_4 | 4 | R/W | 0 | reserved to 0 | |
| E2CTRL14_3 | 3 | R/W | 0 | **Schmitt Trigger Control – DQ23~16 (DQ4~7)**<br>0: On | |

| Name | Bits | Read/Write | Reset State | Comments | Config |
|---|---|---|---|---|---|
| | | | | 1: Off | |
| E2CTRL14_2 | 2 | R/W | 0 | **Slew Rate Control – DQ23~16 (DQ4~7)**<br>0: Fast<br>1: Slow | |
| E2CTRL14_1 | 1 | R/W | 0 | **Driving Current Control – DQ23~16 (DQ4~7)**<br>0: Low<br>1: High | |
| E2CTRL14_0 | 0 | R/W | 0 | reserved to 0 | |

| Register:: PIN_DRIVING_CTRL15 | | | 0xFF9D | | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Comments** | **Config** |
| E2CTRL15_7 | 7 | R/W | 0 | **Schmitt Trigger Control – DQ15~8 (DQ12~15)**<br>0: On<br>1: Off | |
| E2CTRL15_6 | 6 | R/W | 0 | **Slew Rate Control – DQ15~8 (DQ12~15)**<br>0: Fast<br>1: Slow | |
| E2CTRL15_5 | 5 | R/W | 0 | **Driving Current Control– DQ15~8 (DQ12~15)**<br>0: Low<br>1: High | |
| E2CTRL15_4 | 4 | R/W | 0 | reserved to 0 | |
| E2CTRL15_3 | 3 | R/W | 0 | **Schmitt Trigger Control – DQ7~0 (DQ0~3)**<br>0: On<br>1: Off | |
| E2CTRL15_2 | 2 | R/W | 0 | **Slew Rate Control – DQ7~0 (DQ0~3)**<br>0: Fast<br>1: Slow | |
| E2CTRL15_1 | 1 | R/W | 0 | **Driving Current Control – DQ7~0 (DQ0~3)**<br>0: Low<br>1: High | |
| E2CTRL15_0 | 0 | R/W | 0 | reserved to 0 | |

| Register:: PIN_DRIVING_CTRL16 | | | 0xFF9E | | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **Read/Write** | **Reset State** | **Comments** | **Config** |
| E2CTRL16_7 | 7 | R/W | 0 | **Schmitt Trigger Control – SDR Address**<br>0: On<br>1: Off | |
| E2CTRL16_6 | 6 | R/W | 0 | **Slew Rate Control – SDR Address**<br>0: Fast<br>1: Slow | |
| E2CTRL16_5 | 5 | R/W | 0 | **Driving Current Control – SDR Address**<br>0: Low<br>1: High | |
| E2CTRL16_4 | 4 | R/W | 0 | reserved to 0 | |
| E2CTRL16_3 | 3 | R/W | 0 | **Schmitt Trigger Control – SDR DQM**<br>0: On<br>1: Off | |
| E2CTRL16_2 | 2 | R/W | 0 | **Slew Rate Control – SDR DQM**<br>0: Fast<br>1: Slow | |
| E2CTRL16_1 | 1 | R/W | 0 | **Driving Current Control – SDR DQM**<br>0: Low<br>1: High | |
| E2CTRL16_0 | 0 | R/W | 0 | reserved to 0 | |

**Address: 0xFF9F Reserved to 0**

**CEC function**

## *CEC Control Register*

In CEC function, write_reg pulses should have distances larger than 3 XTAL clk period at least.

| Register:: cec_cr_1 | | | | 0xFFAA | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| Reserved | 7:5 | -- | -- | | |
| ini_adr_sel | 4 | R/W | 0 | 1:initial address change<br>0:use original address | |
| ini_adr | 3:0 | R/W | 0x1 | Initial address when ini_adr_sel = 1 | |

| Register::cec_cr0 | | | | 0xFFAB | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| cec_mode | 7:6 | R/W | 0x0 | 00: Disable CEC module<br>01: Enable CEC Normal Operation<br>10: **PAD** Output Test Mode.<br>11: Digital Loopback, Tx Data will be loopback before **PAD**.<br>Note. 1. As CEC module is disabled, RX will not ACK any transaction which destination address is the same with CECLOCADDR or 0xf. | |
| test_mode_pad_data | 5 | R/W | 0x1 | 0: CEC PAD output low<br>1: CEC PAD output high | |
| test_mode_pad_en | 4 | R/W | 0x0 | 0： output high impedance<br>1：PAD output enable<br>This bit is active with CEC_Mode=10 only. | |
| logical_addr | 3:0 | R/W | 0xF | CEC device logical (local) address | |

| Register::cec_cr1 | | | | 0xFFAC | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| timer_div | 7:0 | R/W | 0x14 | DAC ENP(Enable Pulse) divides into Timer Enable Pulse. And Timer Enable Pulse is equal to Input Sample Enable Pulse. Its default value is 0.8MHz divides into 20 to 40KHz(25us). 。<br>CEC clock frequency is used for the bit timers in the receiver and transmitter modes. | |

| Register::cec_cr2 | | | | 0xFFAD | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| pre_div | 7:0 | R/W | 0x21 | Divisor for CEC DAC Clock<br>BusCLK　　CECDiv　　CK_CEC<br>162MHz　　202　　　0.8019MHz | |

| | | | | 27MHz | 33 | 0.8182MHz | |
|---|---|---|---|---|---|---|---|

| **Register::cec_cr3** | | | | | **0xFFAE** | | |
|---|---|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | | | **Config** |
| unreg_ack_en | 7 | R/W | 0x0 | If rx logical addr = 0xF, when receiving a broadcast signal (destination addr = 0xF)　　　　1 : response ack　　　　　　　　　　2 : non to response ack | | | |
| pad_s_ctrl | 6:5 | R/W | 0x1 | **CEC PAD Current Control of Charge Pump** **00: 0.75 uA** **01: 1 uA** 10: 1.25 uA 11: 1.5 uA | | | |
| pad_delay | 4:0 | R/W | 0x03 | **The delay from CEC PAD going high to being disable. Delay: (1+CECPADDELAY)*25us** Typical Value: 01~03 (50us~100us) For Normal Mode only. | | | |

| **Register::cec_rt0** | | | | | **0xFFAF** | | |
|---|---|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | | | **Config** |
| cec_rt0_rsv | 7:6 | R/W | 0x0 | Reserved Register | | | |
| wt_cnt | 5:0 | R | - | Retry Wait Time | | | |

| **Register::cec_rt1** | | | | | **0xFFB0** | | |
|---|---|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | | | **Config** |
| cec_rt1_rsv | 7:5 | R/W | 0x0 | | | | |
| lattest | 4 | R | - | 1: The last initiator own CEC bus is this device | | | |
| retry_no | 3:0 | R/W | 0x5 | Maximum re-transmission times for a single frame, when device is a initiator and device detect low impedance error. In continue mode, retry is inactive. | | | |

| **Register::cec_rx0** | | | | | **0xFFB1** | | |
|---|---|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | | | **Config** |
| rx_en | 7 | R | - | Write 1 to enable Rx As CEC_enable=1 and CECRxEn=0, RX will ACK the transaction which destination address is the same with CECLOCADDR or 0xf | | | wclr_out |
| rx_rst | 6 | R/W | 0x0 | Write 1 to reset Rx State and its FIFO status After finishing each transaction, software should reset Rx part to clear CECRxEOM, CECRxINT and CECRxFIFOov status bits. | | | |
| rx_continuous | 5 | R/W | 0x0 | 0/1：Normal mode / Continuous mode In continuous mode, RxINT will be set to 1 when Rx receive new 8 bytes or EOM. In normal mode, RxINT will be set to 1 iff Rx receive EOM. | | | |
| rx_int_en | 4 | R/W | 0x0 | 1 : CEC Rx interrupt enable If enabled, hardware will trigger interrupt per 8 bytes received or EOM | | | |
| init_addr | 3:0 | R | - | The latest Initiator Address (when device is a follower) | | | |

| **Register::cec_rx1** | | | | | **0xFFB2** | | |
|---|---|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | | | **Config** |
| rx_eom | 7 | R | - | When EOM is received, RxEn will be reset to 0 and | | | |

| | | | | RxINT will be set to 1. | |
|---|---|---|---|---|---|
| rx_int | 6 | R | - | 1 : CEC Rx interrupt pending (write 1 to clear) | wclr_out |
| rx_fifo_ov | 5 | R | - | 1 : Overflow status for CEC 16-byte FIFO | |
| rx_fifo_cnt | 4:0 | R | - | The number of byte has been received by Rx | |

| Register::cec_tx0 | | | | 0xFFB3 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| tx_en | 7 | R | - | Write 1 to enable Tx transmission<br><br>Tx will detect signal free time, and then transmission and re-try automatically. | wclr_out |
| tx_rst | 6 | R/W | 0x0 | Write 1 to reset Tx State and its FIFO status<br><br>After finishing each transaction, software should reset Tx part to clear CECTxEOM, CECTxINT and CECTxFIFOud status bits. | |
| tx_continuous | 5 | R/W | 0x0 | Tx continuous mode<br>0: Normal mode<br><br>1: Continuous mode, software should clear this bit as the last byte is written into Tx FIFO to indicate the end of transmitting data. | |
| tx_int_en | 4 | R/W | 0x0 | 1 : CEC Tx interrupt enable<br><br>If enabled, hardware will trigger interrupt per 8 bytes transmitted or EOM | |
| dest_addr | 3:0 | R/W | 0x0 | Destination Address (when device is a initiator) | |

| Register::cec_tx1 | | | | 0xFFB4 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| tx_eom | 7 | R | - | The transmission has ended. | |
| tx_int | 6 | R | - | 1 : CEC Tx interrupt pending (write 1 to clear) | wclr_out |
| tx_fifo_ud | 5 | R | - | 1 : Underflow status for CEC 16-byte Tx FIFO | |
| tx_fifo_cnt | 4:0 | R | - | The number of byte will been transmitted by Tx | |

Note : following table illustrates the status with the combination of CECTxEn, CECTxEOM, CECTxINT and CECTxContinue after transmitting.

| | CECTxEn | CECTxEOM | CECTxINT | CECTxContinue |
|---|---|---|---|---|
| Complete transmission incorrectly and not in Continue Mode | 0 | 0 | 1 | 0 |
| Complete transmission correctly and not in Continue Mode | 0 | 1 | 1 | 0 |
| Complete transmission incorrectly and in Continue Mode | 0 | 0 | 1 | 0 |
| Transmitted 8 bytes correctly and still in Continue Mode, software should push data into Tx fifo as necessary | 1 | 0 | 1 | 1 |
| Complete transmission and in Continue Mode | 0 | 1 | 1 | 0 (because software clear to 0 after pushing |

| | | | | remaining datum into TX fifo) |
|---|---|---|---|---|
| TX fifo is underflow (in continue mode only) Note : this is the same with CECTxFIFOud=1 | 0 | 0 | 1 | 1 |

| Register::cec_tx_fifo | | | | 0xFFB5 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| tx_dat | 7:0 | R/W | - | Tx FIFO data output port | rport wport |

| Register::cec_rx_fifo | | | | 0xFFB6 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| rx_dat | 7:0 | R/W | - | Rx FIFO data input port | rport wport |

| Register::cec_rx_start0 | | | | 0xFFB7 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| rx_start_low | 7:0 | R/W | 0x8C | Minimum width (3.5ms) | |

| Register::cec_rx_start1 | | | | 0xFFB8 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| rx_start_period | 7:0 | R/W | 0xBC | Maximum width (4.7ms) | |

| Register::cec_rx_data0 | | | | 0xFFB9 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| rx_data_sample | 7:0 | R/W | 0x2A | Sample Time (1.05ms) | |

| Register::cec_rx_data1 | | | | 0xFFBA | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| rx_data_period | 7:0 | R/W | 0x52, | Minimum data bit width (2.05ms) | |

| Register::cec_tx_start0 | | | | 0xFFBB | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| tx_start_low | 7:0 | R/W | 0x94 | 3.7ms (0.025*148) | |

| Register::cec_tx_start1 | | | | 0xFFBC | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| tx_start_high | 7:0 | R/W | 0x20 | 0.8ms (4.5ms – 3.7ms) | |

| Register::cec_tx_data0 | | | | 0xFFBD | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| tx_data_low | 7:0 | R/W | 0x18 | 0.6ms | |

| Register::cec_tx_data1 | | | | 0xFFBE | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| tx_data_01 | 7:0 | R/W | 0x24 | 0.9ms | |

| Register::cec_tx_data2 | | | | 0xFFBF | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| tx_data_high | 7:0 | R/W | 0x24 | 0.9ms | |

# Register 3(page F)

**CEC Analog Function**

| Register::AUT_OK_CONTROL | | | | 0xFFE7 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| CEC27k_EN | 7 | R/W | 1 | 27K pull up resistor enable<br>0: Disable<br>1: Enable | |
| CEC27K_AUTOK | 6 | R/W | 1 | CEC 27K pull up resistor auto calibration enable<br>0: Disable<br>1: Enable | |
| CEC_ENTST | 5 | R/W | 0 | CEC debug enable<br>0: Disable<br>1: Enable | |
| CEC27K_ADJR | 4:0 | R/W | 0x10 | CEC 27K   pull up resistor hand mode setting: | |

| Register::CEC_ANALOG_R | | | | 0xFFE8 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| Reserved | 7 | -- | -- | Reserved to 0 | |
| CEC_TST | 6:1 | R | - | CEC block debug signal out | |
| CEC_Z0_OK | 0 | R | - | CEC 27K Calibration | |

| Register:: REV_DUMMY4 | | | | 0xFFE9 | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| REV_DUMMY4 | 7:0 | R/W | 00 | Dummy4 | |

| Register::MCU_control | | | | 0xFFED | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| Reserved | 7:6 | -- | -- | Reserved | |
| 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<br><br>0: CPU clock is from Crystal divided by DIV<br>1: CPU clock is from PLL divided by DIV | |
| Reserved | 0 | -- | -- | Reserved | |

| Register::MCU_clock_control | | | | 0xFFEE | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| Reserved | 7:6 | -- | -- | Rerserved | |
| 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<br>0: No effect<br>1: reset RTD3580 | Rport Wport |
| SCA_HRST | 0 | R/W | 0 | Hardware reset for Scalar<br>0: No effect<br>1: reset SCALAR module | |

| Register::RAM_test | | | | 0xFFEF | |
|---|---|---|---|---|---|
| **Name** | **Bits** | **R/W** | **Default** | **Comments** | **Config** |
| reserved | 7:4 | -- | 0 | Reserved | |
| EXT_RAM_BIST | 3 | R/W | 0 | Start BIST function for MCU external RAM (512 bytes)<br> 0: finished and clear<br> 1: start | Rport Wport |

| EXT_RAM_STA | 2 | R | 0 | Test result about MCU external RAM<br><br>0: fail<br><br>1: ok | |
|---|---|---|---|---|---|
| reserved | 1:0 | -- | 0 | Reserved | |

## Embedded OSD

*Addressing and Accessing Register*

| ADDRESS | BIT | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| High 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

      **[A11:A0] Address mapping**

      - Font Select and font map SRAM address: 000~EFF    **3.75k*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 support gradient functions.
- Window 4, 5, 6, 7 start/end resolution are 1line(pixel), Window 0, 1, 2, 3 start/end resolution are 4line(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 |
|-----|------|----------|
| 7:6 | -- | Reserved |
| 5:3 | W | Window 0 shadow/border width or 3D button thickness in pixel unit<br>000~111: 1 ~ 8 pixel |
| 2:0 | W | Window 0 shadow/border height in line unit<br>000~111: 1 ~ 8 line<br>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 16-color LUT<br>For 3D window, it is the left-top/bottom border color |
| 3:0 | W | Window 0 border color index in 16-color LUT<br>For 3D window, it is the right-bottom/top border color |

Byte 2

| Bit | Mode | Function |
|-----|------|----------|

| | | |
|---|---|---|
| 7 | W | R Gradient Polarity<br>0: Decrease<br>1: Increase |
| 6 | W | G Gradient Polarity<br>0: Decrease<br>1: Increase |
| 5 | W | B Gradient Polarity<br>0: Decrease<br>1: Increase |
| 4:3 | W | Gradient level<br>00: 1 step per level<br>01: Repeat 2 step per level<br>10: Repeat 3 step per level<br>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 |
|-----|------|----------|
| 7:2 | W | Window 0 horizontal start [5:0] |
| 1 | W | Window 0 horizontal start [11] pixel |
| 0 | W | Window 0 vertical start [11] line |

Byte 1

| Bit | Mode | Function |
|-----|------|----------|
| 7:5 | W | Window 0 vertical start [2:0] line |
| 4:0 | W | Window 0 horizontal start [10:6] pixel |

Byte 2

| Bit | Mode | Function |
|-----|------|----------|
| 7:0 | W | Window 0 vertical start [10:3] line |

Start position must be increments of four.

**Window 0 end position**

**Address: 102h**

Byte 0

| Bit | Mode | Function |
|-----|------|----------|
| 7:2 | W | Window 0 horizontal end [5:0] |
| 1 | W | Window 0 horizontal end [11] pixel |
| 0 | W | Window 0 vertical end [11] line |

Byte 1

| Bit | Mode | Function |
|-----|------|----------|
| 7:5 | W | Window 0 vertical end [2:0] line |
| 4:0 | W | Window 0 horizontal end [10:6] pixel |

Byte 2

| Bit | Mode | Function |
|-----|------|----------|
| 7:0 | W | Window 0 vertical end [10:3] line |

● End position must be increments of four.

**Window 0 control**

**Address: 103h**

Byte 0

| Bit | Mode | Function |
|-----|------|----------|
| 7:0 | -- | Reserved |

Byte 1

| Bit | Mode | Function |
|-----|------|----------|
| 7 | -- | Reserved |

| Bit | Mode | Function |
|-----|------|----------|
| 6:4 | 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 |
| 3:0 | W | Window 0 color index in 16-color LUT |

Byte 2                                                  default: 00h

| Bit | Mode | Function |
|-----|------|----------|
| 7 | W | Reserved |
| 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 |
|-----|------|----------|
| 7:6 | W | Reserved |
| 5:3 | W | Window 1 shadow/border width or 3D button thickness in pixel unit<br>000~111: 1 ~ 8 pixel |
| 2:0 | W | Window 1 shadow/border height in line unit<br>000~111: 1 ~ 8 line<br>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 16-color LUT<br>For 3D window, it is the left-top/bottom border color |
| 3:0 | W | Window 1 border color index in 16-color LUT<br>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 |
|-----|------|----------|
| 7:2 | W | Window 1 horizontal start [5:0] |
| 1 | W | Window 1 horizontal start [11] pixel |
| 0 | W | Window 1 vertical start [11] line |

Byte 1

| Bit | Mode | Function |
|-----|------|----------|
| 7:5 | W | Window 1 vertical start [2:0] line |
| 4:0 | W | Window 1 horizontal start [10:6] pixel |

Byte 2

| Bit | Mode | Function |
|-----|------|----------|
| 7:0 | W | Window 1 vertical start [10:3] line |

Start position must be increments of four.

**Window 1 end position**

**Address: 106h**

Byte 0

| Bit | Mode | Function |
|-----|------|----------|
| 7:2 | W | Window 1 horizontal end [5:0] |
| 1 | W | Window 1 horizontal end [11] pixel |
| 0 | W | Window 1 vertical end [11] line |

Byte 1

| Bit | Mode | Function |
|-----|------|----------|
| 7:5 | W | Window 1 vertical end [2:0] line |
| 4:0 | W | Window 1 horizontal end [10:6] pixel |

Byte 2

| Bit | Mode | Function |
|-----|------|----------|
| 7:0 | W | Window 1 vertical end [10:3] line |

End position must be increments of four.

**Window 1 control**

**Address: 107h**

Byte 0

| Bit | Mode | Function |
|-----|------|----------|
| 7:0 | -- | Reserved |

Byte 1

| Bit | Mode | Function |
|-----|------|----------|
| 7:4 | -- | Reserved |
| 3:0 | W | Window 1 color index in 16-color LUT |

Byte 2                                                                    default: 00h

| Bit | Mode | Function |
|-----|------|----------|
| 7:5 | W | Reserved |
| 4 | W | Shadow/Border/3D button<br>0: Disable<br>1: Enable |
| 3:1 | W | Window 1 Type<br>000: Shadow Type 1<br>001: Shadow Type 2<br>010: Shadow Type3<br>011: Shadow Type 4<br>100: 3D Button Type 1<br>101: 3D Button Type 2<br>110: Reserved<br>111: Border |
| 0 | W | Window 1 Enable<br>0: Disable<br>1: Enable |

**Window 2 Shadow/Border/Gradient**

**Address: 108h**

Byte 0

| Bit | Mode | Function |
|-----|------|----------|
| 7:6 | W | Reserved |
| 5:3 | W | Window 2 shadow/border width or 3D button thickness in pixel unit<br>000~111: 1 ~ 8 pixel |
| 2:0 | W | Window 2 shadow/border height in line unit<br>000~111: 1 ~ 8 line<br>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 16-color LUT |
| | | For 3D window, it is the left-top/bottom border color |
| 3:0 | W | Window 2 border color index in 16-color LUT |
| | | 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 |
|---|---|---|
| 7:2 | W | Window 2 horizontal start [5:0] |
| 1 | W | Window 2 horizontal start [11] pixel |
| 0 | W | Window 2 vertical start [11] line |

Byte 1

| Bit | Mode | Function |
|---|---|---|
| 7:5 | W | Window 2 vertical start [2:0] line |
| 4:0 | W | Window 2 horizontal start [10:6] pixel |

Byte 2

| Bit | Mode | Function |
|---|---|---|
| 7:0 | W | Window 2 vertical start [10:3] line |

Start position must be increments of four.

**Window 2 end position**

**Address: 10Ah**

Byte 0

| Bit | Mode | Function |
|---|---|---|
| 7:2 | W | Window 2 horizontal end [5:0] |
| 1 | W | Window 2 horizontal end [11] pixel |
| 0 | W | Window 2 vertical end [11] line |

Byte 1

| Bit | Mode | Function |
|---|---|---|
| 7:5 | W | Window 2 vertical end [2:0] line |
| 4:0 | W | Window 2 horizontal end [10:6] pixel |

Byte 2

| Bit | Mode | Function |
|---|---|---|
| 7:0 | W | Window 2 vertical end [10:3] line |

End position must be increments of four.

**Window 2 control**

**Address: 10Bh**

Byte 0

| Bit | Mode | Function |
|-----|------|----------|
| 7:0 | -- | Reserved |

Byte 1

| Bit | Mode | Function |
|-----|------|----------|
| 7:4 | -- | Reserved |
| 3:0 | W | Window 2 color index in 16-color LUT |

Byte 2                                                                          default: 00h

| Bit | Mode | Function |
|-----|------|----------|
| 7:5 | W | Reserved |
| 4 | W | Shadow/Border/3D button<br>0: Disable<br>1: Enable |
| 3:1 | W | Window 2 Type<br>000: Shadow Type 1<br>001: Shadow Type 2<br>010: Shadow Type3<br>011: Shadow Type 4<br>100: 3D Button Type 1<br>101: 3D Button Type 2<br>110: Reserved<br>111: Border |
| 0 | W | Window 2 Enable<br>0: Disable<br>1: Enable |

**Window 3 Shadow/Border/Gradient**

**Address: 10Ch**

Byte 0

| Bit | Mode | Function |
|-----|------|----------|
| 7:6 | W | Reserved |
| 5:3 | W | Window 3 shadow/border width or 3D button thickness in pixel unit<br>000~111: 1 ~ 8 pixel |
| 2:0 | W | Window 3 shadow/border height in line unit<br>000~111: 1 ~ 8 line<br>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 16-color LUT<br>For 3D window, it is the left-top/bottom border color |
| 3:0 | W | Window 3 border color index in 16-color LUT<br>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 |
|-----|------|----------|
| 7:2 | W | Window 3 horizontal start [5:0] |
| 1 | W | Window 3 horizontal start [11] pixel |
| 0 | W | Window 3 vertical start [11] line |

Byte 1

| Bit | Mode | Function |
|-----|------|----------|
| 7:5 | W | Window 3 vertical start [2:0] line |
| 4:0 | W | Window 3 horizontal start [10:6] pixel |

Byte 2

| Bit | Mode | Function |
|-----|------|----------|
| 7:0 | W | Window 3 vertical start [10:3] line |

Start position must be increments of four.

## Window 3 end position

## Address: 10Eh

Byte 0

| Bit | Mode | Function |
|-----|------|----------|
| 7:2 | W | Window 3 horizontal end [5:0] |
| 1 | W | Window 3 horizontal end [11] pixel |
| 0 | W | Window 3 vertical end [11] line |

Byte 1

| Bit | Mode | Function |
|-----|------|----------|
| 7:5 | W | Window 3 vertical end [2:0] line |
| 4:0 | W | Window 3 horizontal end [10:6] pixel |

Byte 2

| Bit | Mode | Function |
|-----|------|----------|

| Bit | Mode | Function |
|---|---|---|
| 7:0 | W | Window 3 vertical end [10:3] line |

End position must be increments of four.

**Window 3 control**

**Address: 10Fh**

Byte 0

| Bit | Mode | Function |
|---|---|---|
| 7:0 | -- | Reserved |

Byte 1

| Bit | Mode | Function |
|---|---|---|
| 7:4 | -- | Reserved |
| 3:0 | W | Window 3 color index in 16-color LUT |

Byte 2                                                              default: 00h

| Bit | Mode | Function |
|---|---|---|
| 7:5 | W | Reserved |
| 4 | W | Shadow/Border/3D button<br>0: Disable<br>1: Enable |
| 3:1 | W | Window 3 Type<br>000: Shadow Type 1<br>001: Shadow Type 2<br>010: Shadow Type3<br>011: Shadow Type 4<br>100: 3D Button Type 1<br>101: 3D Button Type 2<br>110: Reserved<br>111: Border |
| 0 | W | Window 3 Enable<br>0: Disable<br>1: Enable |

**Window 4 Shadow/Border/Gradient**

**Address: 110h**

Byte 0

| Bit | Mode | Function |
|---|---|---|
| 7:6 | W | Reserved |
| 5:3 | W | Window 4 shadow/border width or 3D button thickness in pixel unit<br>000~111: 1 ~ 8 pixel |
| 2:0 | W | Window 4 shadow/border height in line unit<br>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 16-color LUT |
| | | For 3D window, it is the left-top/ bottom border color |
| 3:0 | W | Window 4 border color index in 16-color LUT |
| | | 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 |
| --- | --- | --- |
| 7:2 | W | Window 4 horizontal start [5:0] |
| 1 | W | Window 4 horizontal start [11] pixel |
| 0 | W | Window 4 vertical start [11] line |

Byte 1

| Bit | Mode | Function |
| --- | --- | --- |
| 7:5 | W | Window 4 vertical start [2:0] line |
| 4:0 | W | Window 4 horizontal start [10:6] pixel |

Byte 2

| Bit | Mode | Function |
| --- | --- | --- |
| 7:0 | W | Window 4 vertical start [10:3] line |

**Window 4 end position**

**Address: 112h**

Byte 0

| Bit | Mode | Function |
| --- | --- | --- |
| 7:2 | W | Window 4 horizontal end [5:0] |
| 1 | W | Window 4 horizontal end [11] pixel |
| 0 | W | Window 4 vertical end [11] line |

Byte 1

| Bit | Mode | Function |
| --- | --- | --- |
| 7:5 | W | Window 4 vertical end [2:0] line |
| 4:0 | W | Window 4 horizontal end [10:6] pixel |

Byte 2

| Bit | Mode | Function |
| --- | --- | --- |
| 7:0 | W | Window 4 vertical end [10:3] line |

**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 16-color LUT |

Byte 2                                                                          default: 00h

| Bit | Mode | Function |
|-----|------|----------|
| 7:5 | W | Reserved |
| 4 | W | Shadow/Border/3D button<br>0: Disable<br>1: Enable |
| 3:1 | W | Window 4 Type<br>000: Shadow Type 1<br>001: Shadow Type 2<br>010: Shadow Type3<br>011: Shadow Type 4<br>100: 3D Button Type 1<br>101: 3D Button Type 2<br>110: Reserved<br>111: Border |
| 0 | W | Window 4 Enable<br>0: Disable<br>1: Enable |

**Window 5 Shadow/Border/Gradient**

**Address: 114h**

Byte 0

| Bit | Mode | Function |
|-----|------|----------|
| 7:6 | W | Reserved |
| 5:3 | W | Window 5 shadow/border width or 3D button thickness in pixel unit<br>000~111: 1 ~ 8 pixel |
| 2:0 | W | Window 5 shadow/border height in line unit<br>000~111: 1 ~ 8 line<br>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 16-color LUT |
| | | For 3D window, it is the left-top/bottom border color |
| 3:0 | W | Window 5 border color index in 16-color LUT |
| | | 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 |
|---|---|---|
| 7:2 | W | Window 5 horizontal start [5:0] |
| 1 | W | Window 5 horizontal start [11] pixel |
| 0 | W | Window 5 vertical start [11] line |

Byte 1

| Bit | Mode | Function |
|---|---|---|
| 7:5 | W | Window 5 vertical start [2:0] line |
| 4:0 | W | Window 5 horizontal start [10:6] pixel |

Byte 2

| Bit | Mode | Function |
|---|---|---|
| 7:0 | W | Window 5 vertical start [10:3] line |

**Window 5 end position**

**Address: 116h**

Byte 0

| Bit | Mode | Function |
|-----|------|----------|
| 7:2 | W | Window 5 horizontal end [5:0] |
| 1 | W | Window 5 horizontal end [11] pixel |
| 0 | W | Window 5 vertical end [11] line |

Byte 1

| Bit | Mode | Function |
|-----|------|----------|
| 7:5 | W | Window 5 vertical end [2:0] line |
| 4:0 | W | Window 5 horizontal end [10:6] pixel |

Byte 2

| Bit | Mode | Function |
|-----|------|----------|
| 7:0 | W | Window 5 vertical end [10:3] line |

**Window 5 control**

**Address: 117h**

Byte 0

| Bit | Mode | Function |
|-----|------|----------|
| 7:0 | -- | Reserved |

Byte 1

| Bit | Mode | Function |
|-----|------|----------|
| 7 | -- | Reserved |
| 6:4 | W | 111: 7 level per gradient<br>110: 6 level per gradient<br>101: 5 level per gradient<br>100: 4 level per gradient<br>011: 3 level per gradient<br>010: 2 level per gradient<br>001: 1 level per gradient<br>000: 8 level per gradient |
| 3:0 | W | Window 5 color index in 16-color LUT |

Byte 2                                                                default: 00h

| Bit | Mode | Function |
|-----|------|----------|
| 7 | W | Reserved |
| 6 | W | Gradient function<br>0: Disable |

| Bit | Mode | Function |
|---|---|---|
| | | 1: Enable |
| 5 | W | Gradient direction<br>0: Horizontal<br>1: Vertical |
| 4 | W | Shadow/Border/3D button<br>0: Disable<br>1: Enable |
| 3:1 | W | Window 5 Type<br>000: Shadow Type 1<br>001: Shadow Type 2<br>010: Shadow Type3<br>011: Shadow Type 4<br>100: 3D Button Type 1<br>101: 3D Button Type 2<br>110: Reserved<br>111: Border |
| 0 | W | Window 5 Enable<br>0: Disable<br>1: Enable |

**Window 6 Shadow/Border/Gradient**

**Address: 118h**

Byte 0

| Bit | Mode | Function |
|---|---|---|
| 7:6 | W | Reserved |
| 5:3 | W | Window 6 shadow/border width or 3D button thickness in pixel unit<br>000~111: 1 ~ 8 pixel |
| 2:0 | W | Window 6 shadow/border height in line unit<br>000~111: 1 ~ 8 line<br>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 16-color LUT<br>For 3D window, it is the left-top/ bottom border color |
| 3:0 | W | Window 6 border color index in 16-color LUT<br>For 3D window, it is the right-bottom/top border color |

Byte 2

| Bit | Mode | Function |
|---|---|---|

| Bit | Mode | Function |
|-----|------|----------|
| 7 | W | R Gradient Polarity<br>0: Decrease<br>1: Increase |
| 6 | W | G Gradient Polarity<br>0: Decrease<br>1: Increase |
| 5 | W | B Gradient Polarity<br>0: Decrease<br>1: Increase |
| 4:3 | W | Gradient level<br>00: 1 step per level<br>01: Repeat 2 step per level<br>10: Repeat 3 step per level<br>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 |
|-----|------|----------|
| 7:2 | W | Window 6 horizontal start [5:0] |
| 1 | W | Window 6 horizontal start [11] pixel |
| 0 | W | Window 6 vertical start [11] line |

Byte 1

| Bit | Mode | Function |
|-----|------|----------|
| 7:5 | W | Window 6 vertical start [2:0] line |
| 4:0 | W | Window 6 horizontal start [10:6] pixel |

Byte 2

| Bit | Mode | Function |
|-----|------|----------|
| 7:0 | W | Window 6 vertical start [10:3] line |


**Window 6 end position**

**Address: 11Ah**

Byte 0

| Bit | Mode | Function |
|-----|------|----------|
| 7:2 | W | Window 6 horizontal end [5:0] |
| 1 | W | Window 6 horizontal end [11] pixel |

| 0 | W | Window 6 vertical end [11] line |
|---|---|---|

Byte 1

| Bit | Mode | Function |
|---|---|---|
| 7:5 | W | Window 6 vertical end [2:0] line |
| 4:0 | W | Window 6 horizontal end [10:6] pixel |

Byte 2

| Bit | Mode | Function |
|---|---|---|
| 7:0 | W | Window 6 vertical end [10:3] line |

**Window 6 control**

**Address: 11Bh**

Byte 0

| Bit | Mode | Function |
|---|---|---|
| 7:0 | -- | Reserved |

Byte 1

| Bit | Mode | Function |
|---|---|---|
| 7 | -- | Reserved |
| 6:4 | W | 111: 7 level per gradient<br>110: 6 level per gradient<br>101: 5 level per gradient<br>100: 4 level per gradient<br>011: 3 level per gradient<br>010: 2 level per gradient<br>001: 1 level per gradient<br>000: 8 level per gradient |
| 3:0 | W | Window 6 color index in 16-color LUT |

Byte 2                                                                default: 00h

| Bit | Mode | Function |
|---|---|---|
| 7 | W | Reserved |
| 6 | W | Gradient function<br>0: Disable<br>1: Enable |
| 5 | W | Gradient direction<br>0: Horizontal<br>1: Vertical |
| 4 | W | Shadow/Border/3D button<br>0: Disable<br>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<br>0: Disable<br>1: Enable |

**Window 7 Shadow/Border/Gradient**

**Address: 11Ch**

Byte 0

| Bit | Mode | Function |
|---|---|---|
| 7:6 | W | Reserved |
| 5:3 | W | Window 7 shadow/border width or 3D button thickness in pixel unit<br>000~111: 1 ~ 8 pixel |
| 2:0 | W | Window 7 shadow/border height in line unit<br>000~111: 1 ~ 8 line<br>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 16-color LUT<br>For 3D window, it is the left-top/bottom border color |
| 3:0 | W | Window 7 border color index in 16-color LUT<br>For 3D window, it is the right-bottom/top border color |

Byte 2

| Bit | Mode | Function |
|---|---|---|
| 7 | W | R Gradient Polarity<br>0: Decrease<br>1: Increase |
| 6 | W | G Gradient Polarity<br>0: Decrease<br>1: Increase |
| 5 | W | B Gradient Polarity<br>0: Decrease |

| Bit | | Mode | Function |
|---|---|---|---|
| | | | 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 |
|---|---|---|
| 7:2 | W | Window 7 horizontal start [5:0] |
| 1 | W | Window 7 horizontal stat [11] pixel |
| 0 | W | Window 7 vertical start [11] line |

Byte 1

| Bit | Mode | Function |
|---|---|---|
| 7:5 | W | Window 7 vertical start [2:0] line |
| 4:0 | W | Window 7 horizontal start [10:6] pixel |

Byte 2

| Bit | Mode | Function |
|---|---|---|
| 7:0 | W | Window 7 vertical start [10:3] line |

**Window 7 end position**

**Address: 11Eh**

Byte 0

| Bit | Mode | Function |
|---|---|---|
| 7:2 | W | Window 7 horizontal end [5:0] |
| 1 | W | Window 7 horizontal end [11] pixel |
| 0 | W | Window 7 vertical end [11] line |

Byte 1

| Bit | Mode | Function |
|---|---|---|
| 7:5 | W | Window 7 vertical end [2:0] line |
| 4:0 | W | Window 7 horizontal end [10:6] pixel |

Byte 2

| Bit | Mode | Function |
|---|---|---|
| 7:0 | W | Window 7 vertical end [10:3] line |

**Window 7 control**

**Address: 11Fh**

Byte 0

| Bit | Mode | Function |
|-----|------|----------|
| 7:0 | -- | Reserved |

Byte 1

| Bit | Mode | Function |
|-----|------|----------|
| 7 | -- | Reserved |
| 6:4 | W | 111: 7 level per gradient<br>110: 6 level per gradient<br>101: 5 level per gradient<br>100: 4 level per gradient<br>011: 3 level per gradient<br>010: 2 level per gradient<br>001: 1 level per gradient<br>000: 8 level per gradient |
| 3:0 | W | Window 7 color index in 16-color LUT |
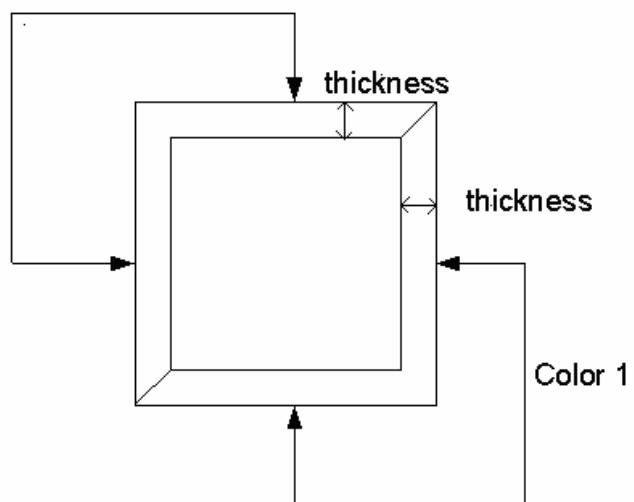
Byte 2                                                                          default: 00h

| Bit | Mode | Function |
|-----|------|----------|
| 7 | W | Reserved |
| 6 | W | Gradient function<br>0: Disable<br>1: Enable |
| 5 | W | Gradient direction<br>0: Horizontal<br>1: Vertical |
| 4 | W | Shadow/Border/3D button<br>0: Disable<br>1: Enable |
| 3:1 | W | Window 7 Type<br>000: Shadow Type 1<br>001: Shadow Type 2<br>010: Shadow Type3<br>011: Shadow Type 4<br>100: 3D Button Type 1<br>101: 3D Button Type 2<br>110: Reserved<br>111: Border |

| 0 | W | Window 7 Enable |
|---|---|---|
|   |   | 0: Disable |
|   |   | 1: Enable |
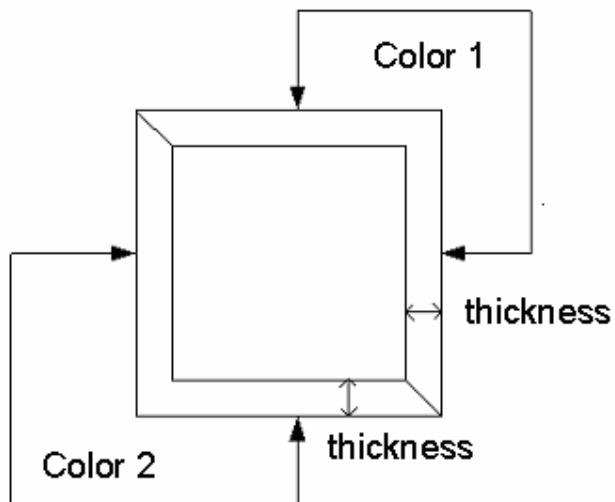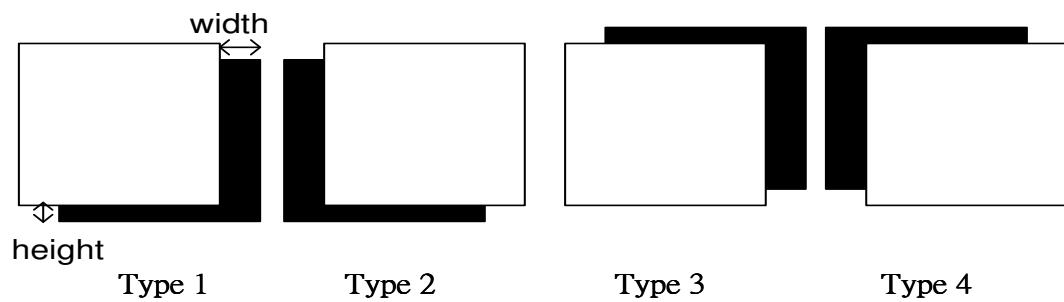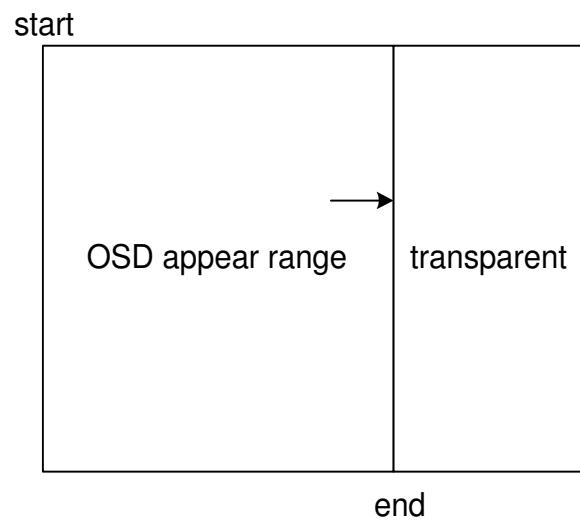
3D Button Type 1



3D Button Type 2

width

height

Type 1          Type 2          Type 3          Type 4

Shadow in all direction

start

OSD appear range      transparent

end

**Window mask fade/in out function**

## Frame control registers

**Address: 000h**

Byte 0

| Bit | Mode | Function |
|-----|------|----------|
| 7:0 | R/W | Vertical Delay [10:3] <br><br> 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] <br><br> The bits define the horizontal starting address. Total 1024 step unit:4 pixels |
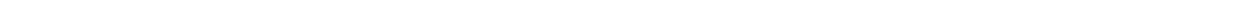
Horizontal delay minimum should set 2

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 <br> 00: middle <br> 01: left <br> 10: right <br> 11: reserved |
| 0 | R/W | OSD enable <br> 0: OSD circuit is inactivated <br> 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: 003h**

Byte 0                                                      Default: 00h

| Bit | Mode | Function |
|-----|------|----------|
| 7 | R/W | Specific color blending (blending type 2)<br>0: Disable<br>1: Enable |
| 6:5 | R/W | Window 7special function<br>00: disable<br>01: blending (blending type 3)<br>10: window 7 mask region appear<br>11: window 7 mask region transparent |
| 4 | R/W | OSD vertical start input signal source select<br>0: Select DVS as OSD VSYNC input<br>1: Select ENA as OSD VSYNC input |
| 3:0 | R/W | Blending color from 16-color LUT (blending type 2) |

Byte 1                                                      Default: 00h

| Bit | Mode | Function |
|-----|------|----------|
| 7:4 | R/W | Char shadow/border color |
| 3: 2 | R/W | Alpha blending type (blending type 1)<br>00: Disable alpha blending<br>01: Only window blending<br>10: All blending<br>11: Window and Character background blending |
| 1 | R/W | Double width enable (For all OSD including windows and characters)<br>0: Normal<br>1: Double |
| 0 | R/W | Double Height enable (For all OSD including windows and characters)<br>0: Normal<br>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**<br>0x: No swap<br>10: CCW<br>11: CW |

| 5 | R | **Buffer Empty**<br>0: Empty<br>1: Not Empty |
|---|---|---|
| 4 | R | **Buffer Valid**<br>0: Done<br>1: Buffer is writing to SDRAM |
| 3 | R/W | **Reset Buffer**<br>Write 1 to reset and auto-clear after finished. |
| 2 | R/W | **Hardware Rotation Enable**<br>0: Disable<br>1: Enable (Default)<br>OSD compression function must be enabled simultaneously. |
| 1 | R/W | **Global Blinking Enable**<br>0: Disable<br>1: Enable |
| 0 | R/W | **Rotation**<br>0: Normal (data latch 24 bit per 24 bit)<br>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 |

| 23~12 bit(High) |
| 11~0 bit(Low) |
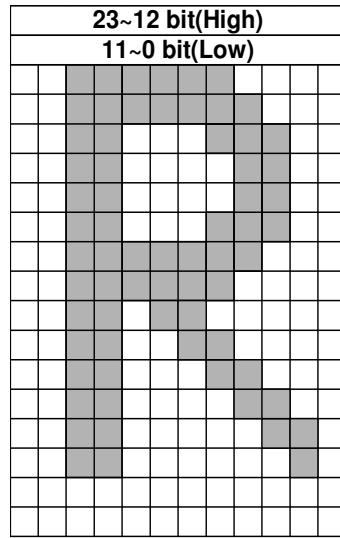
**Figure 3 Non-rotated memory alignments**

**23**                                                  **6**



| First Address |
| Second Address |
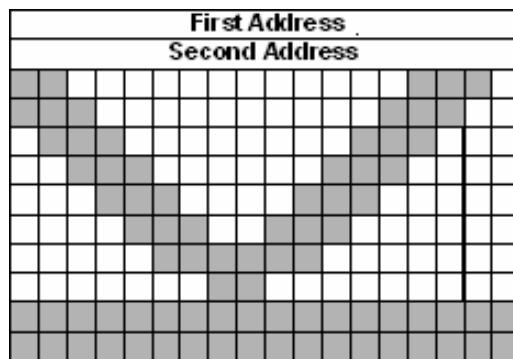
**Figure 4 Rotated memory alignments**

**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_xxx0b

| Bit | Mode | Function |
|-----|------|----------|
| 7:1 | -- | reserved |
| 0 | R/W | OSD compression (4bit/symbol, VLC code 1111_1111 represents the end of data) (only for SRAM)<br>0: disable<br>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.

## OSD Special Function
### Address: 008h

Byte 0                                                                Default: 0x00

| Bit | Mode | Function |
|-----|------|----------|
| 7 | R/W | OSD Special Function Enable<br>0: Disable<br>1: Enable |
| 6 | R/W | OSD Special Function Select (Effective only when Bit[7]=1)<br>0: ROLL-UP<br>1: POP-ON |
| 5 | R/W | OSD Vertical Boundary Function Enable<br>0: Disable<br>1: Enable |
| 4:1 | R/W | Reserved to 0 |
| 0 | R/W | Display Base Select (Effective only when Bit[7:6]=11`b)<br>0: Base 0<br>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<br>0: OFF<br>1: ON |
| 5:4 | R/W | Blending Type of Window 7<br>00: NO Blending for both F/B<br>01: Only Blending for Foreground<br>10: Only Blending for Background<br>11: Both Blending for F/B |

| Bit | Mode | Function |
|---|---|---|
| 3:2 | R/W | Blending Type of Window 6<br>00: NO Blending for both F/B<br>01: Only Blending for Foreground<br>10: Only Blending for Background<br>11: Both Blending for F/B |
| 2:0 | -- | Reserved |

Byte 1                                                                    Default: 0x00

| Bit | Mode | Function |
|---|---|---|
| 7:0 | -- | Reserved |

Byte 2                                                                    Default: 0x00

| Bit | Mode | Function |
|---|---|---|
| 7:0 | -- | Reserved |

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.

## *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 | … |
| | … |
| | … |

**16.5k 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 12 * 24bit SRAM

One 2-bit font requires 24 * 24bit SRAM

One 4-bit font requires 48 * 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 + 12 * 128

2-bit Font, if CS = 128, Real Address of Font = Font Base Address + 24 * 128

4-bit Font, if CS = 128, Real Address of Font = Font Base Address + 48 * 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 002h byte1 [1:0] 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<br>0: OSD End Command<br>Each row must start with row-command, last word of OSD map must be end-command |
| 6 | R/W | VBI OSD function enable<br>0: normal OSD function as usual<br>1: support VBI OSD functions like underline, B/F separated blink and 512 fonts select |
| 5 | W | Reserved |
| 4:2 | W | Character border/shadow<br>000: None<br>001: Border<br>100: Shadow (left-top)<br>101: Shadow (left-bottom)<br>110: Shadow (right-top)<br>111: Shadow (right-bottom) |
| 1 | W | Double character width<br>0: x1<br>1: x2 |
| 0 | W | Double character height<br>0: x1<br>1: x2 |

Byte 1
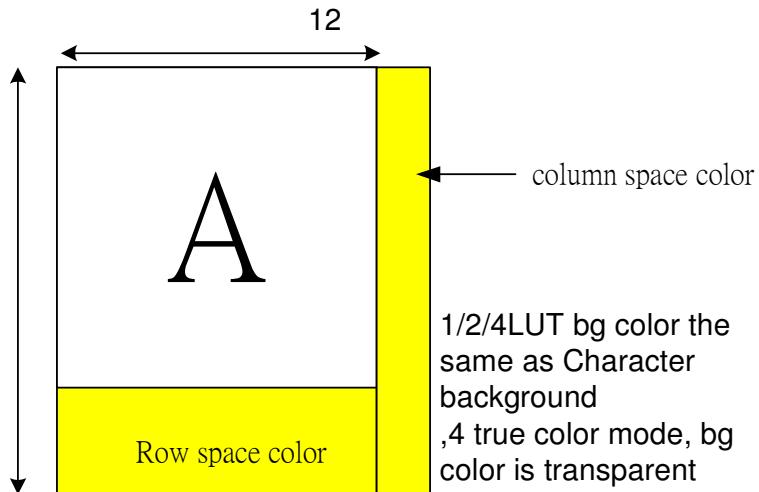
| Bit | Mode | Function |
|---|---|---|
| 7:3 | W | Row height (1~32) |
| 2:0 | W | Column space<br>0~7 pixel column space<br>When Char is doubled, so is column space. |

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<br>0: Disable<br>1: Enable |
| 5:0 | 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 |
|-----|------|----------|
| 7:5 | W | Reserved |
| 4 | W | Reserved |
| 3:0 | W | Blank color – select one of 16-color LUT |

| | | (0 is special for transparent) |

**Character Command (For 1-bit RAM font)**

Byte 0

| Bit | Mode | Function |
|-----|------|----------|
| 7 | W | 1 |
| 6 | W | Character Blinking effect<br>0: Disable<br>1: Enable |
| 5:4 | W | 00<br>(Font type<br>00: 1-bit RAM Font<br>01: 4-bit RAM Font<br>1x: 2-bit RAM Font) |
| 3:0 | W | VBI OSD disable:<br>Character width (only for 1-pixel font, doubled when specifying<br>double-width in Row/Blank command register)<br>For 12x18 font:<br>0100: 4-pixel    0101: 5-pixel    0110: 6-pixel    0111: 7-pixel<br>1000: 8-pixel    1001: 9-pixel    1010: 10-pixel    1011:11-pixel<br>1100: 12-pixel<br>For 18x12 Font (rotated)<br>0000: 4-pixel    0001: 5-pixel    0010: 6-pixel    0011: 7-pixel<br>0100: 8-pixel    0101: 9-pixel    0110: 10-pixel    0111: 11-pixel<br>1000: 12-pixel 1001:13-pixel 1010:14-pixel 1011:15-pixel<br>1100: 16-pixel 1101:17-pixel 1110:18-pixel<br>VBI OSD enable:<br>While VBI OSD enable, 1 bit font will be NO rotated and 12-pixel fonts<br>always. Then the [3:0] setting will be as following:<br>[3]: character select[8]<br>support 512 font while VBI OSD enable<br>[2]: additional blinking effect<br>{[6], [2]}<br>00: NO blink for both F/B<br>01: Only blink for Foreground<br>10: Only blink for Background<br>11: Both blink for F/B<br>[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<br>Select one of 16-color from color LUT |
| 3:0 | W | Background color<br>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 | MSB of Foreground color 10, Foreground 01 |
| 3:1 | W | Foreground color 11<br>Select one of 8 color from color LUT<br>Add Byte0 [6] as MSB for 16-color LUT. |
| 0 | W | Background color 00 Bit[2]<br>Select one of 8 color from color LUT |

Byte 1

| Bit | Mode | Function |
|-----|------|----------|
| 7:0 | W | Character Select [7:0] |

Byte 2

| Bit | Mode | Function |
|-----|------|----------|
| 7:6 | W | Background color 00 Bit[1:0]<br>Select one of 8 color from color LUT<br>While 0 is special for transparent<br>Add Byte0 [6] as MSB for 16-color LUT.<br>Once we fill 0000 or 1000(MSB follow Byte0[6]), BG appears transparent. |
| 5:3 | W | Foreground color 10 |

| Bit | Mode | Function |
|---|---|---|
| | | 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 |
| | | (for Byte1[7] = 1) |
| | | Red color level |
| | | MSB 4 bits for 8 bits color level (LSB 4 bits are 1111) |

Byte 1

| Bit | Mode | Function |
|---|---|---|
| 7 | W | 0: 4bit Look Up Table, 0000'b is transparent. |
| | | 1: 3bit specify R,G,B pattern, color level defined in Byte0[3:0],Byte2. One |
| | | mask bit defines foreground or background. |
| 6:0 | W | Character Select [6:0] |
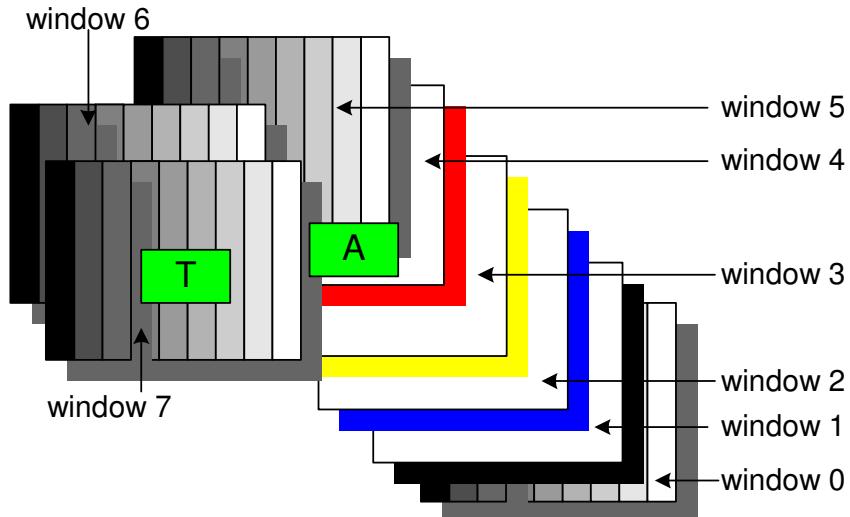
- When 4-bit look-up table mode，color of column space is the same as background.
- When 4-bit look-up table mode and pixel value is 0000, and byte0[3:0]=0000 means transparent.
- When true color mode and pixel value is 0000，it is transparent。

Byte 2

| Bit | Mode | Function |
|---|---|---|
| 7:4 | W | (for Byte1[7] = 1) |
| | | Green color level |
| | | MSB 4 bits for 8 bits color level (LSB 4 bits are 1111) |

| 3:0 | W | (for Byte1[7] = 1) |
| | | Blue color level |
| | | MSB 4 bits for 8 bits color level (LSB 4 bits are 1111) |
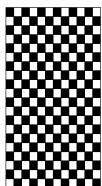
Display Priority

We have four windows with gradient and four 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
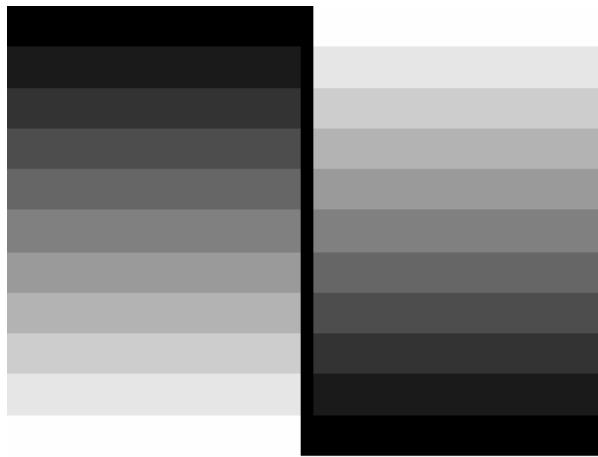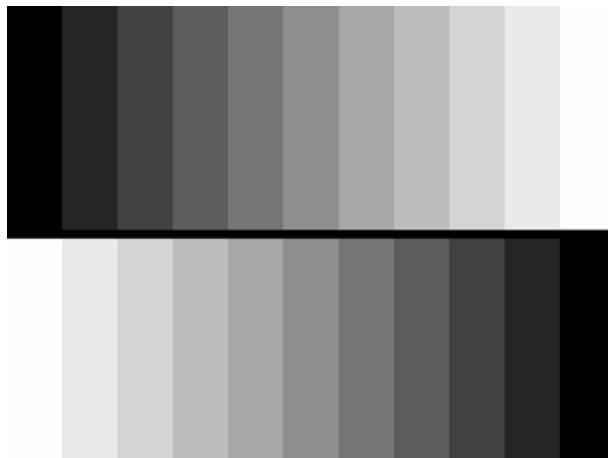
The required number of character map is larger than RAM size. We must turn on double width
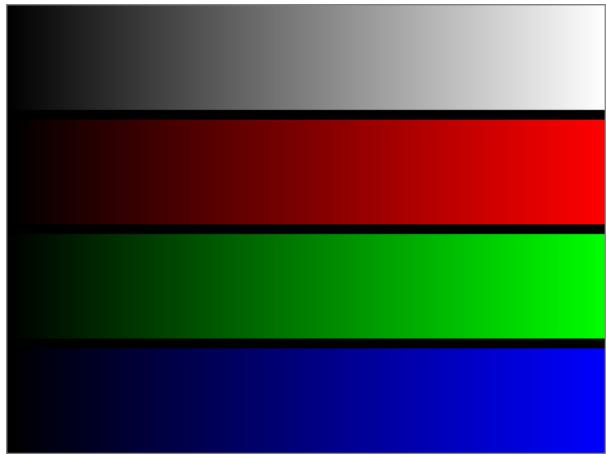
or double height function to reduce the half of character map.

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.

# 4. 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 | 3.0 | 3.3 | 3.6 | V |
| Supply Voltage | VCCK | 1.08 | 1.2 | 1.32 | V |
| Electrostatic Discharge | $V_{ESD}$ | | | ±2.5 | kV |
| Latch-Up | $I_{LA}$ | | | ±100 | mA |
| Ambient Operating Temperature | $T_A$ | 0 | | 70 | ℃ |
| Storage temperature (plastic) | $T_{STG}$ | -55 | | 125 | ℃ |
| Thermal Resistance (Junction to Air) | $\theta_{JA}$ | | | 47.4 ＊ | ℃/W |
| Junction Acceptable Temperature | $T_j$ | | | 125 | ℃ |

＊

Under 2-layer PCB

Dimension 50 x 70 mm,

Thickness: 1.6mm
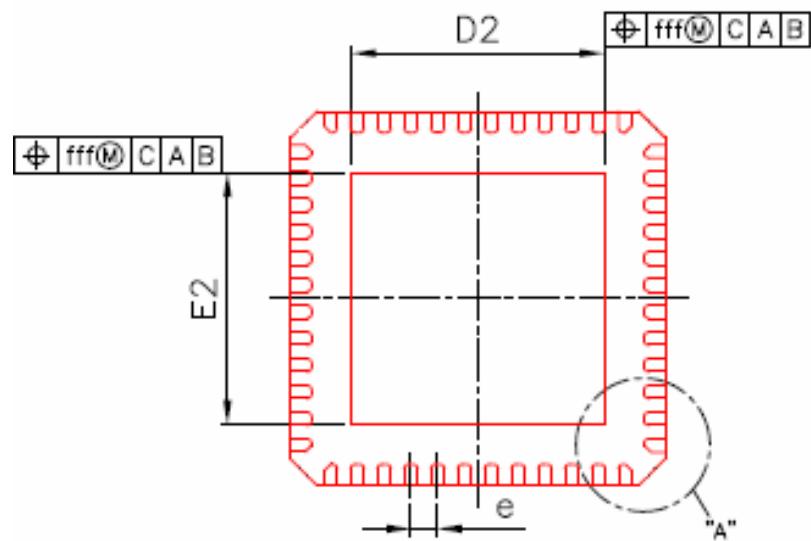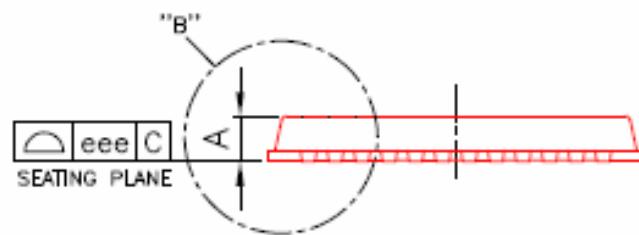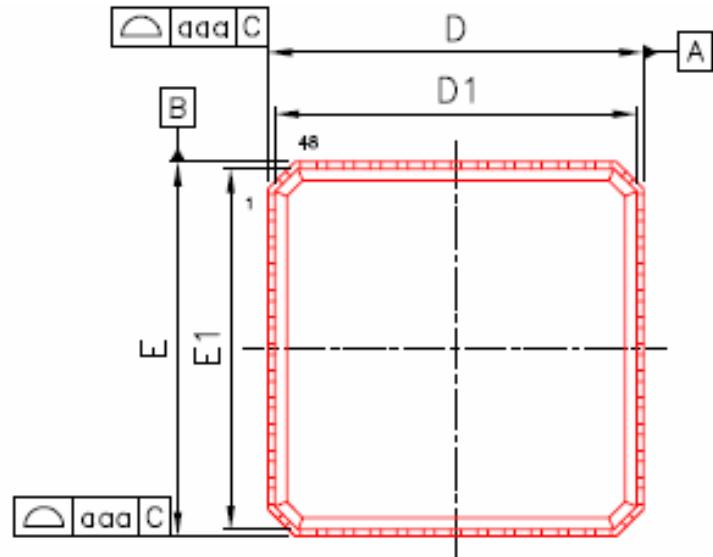
Top layer: 65% coverage of Cu, 0.5oz thickness
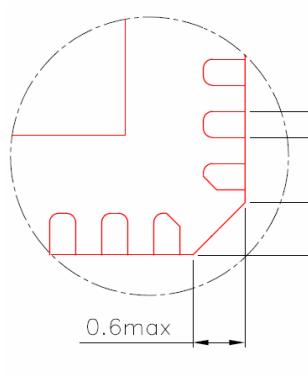
Bottom layer: 95% coverage of Cu, 0.5oz thickness

Via Underneath Package: 12 (Diameter: 12 mil)
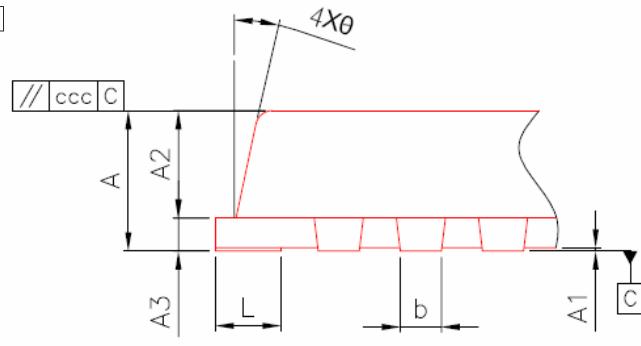
# 5. Mechanical Specification

**Plastic Quad Flat No-Lead Package 48 Leads 7x7mm$^2$ Outline**

DETAIL : "A"

DETAIL : "B"

| Symbol | Dimension in mm | | | Dimension in inch | | |
|---|---|---|---|---|---|---|
| | Min | Nom | Max | Min | Nom | Max |
| A | 0.75 | 0.85 | 1.00 | 0.030 | 0.034 | 0.039 |
| $A_1$ | 0.00 | 0.02 | 0.05 | 0.000 | 0.001 | 0.002 |
| $A_2$ | 0.55 | 0.65 | 0.80 | 0.022 | 0.026 | 0.032 |
| $A_3$ | 0.20 REF | | | 0.008 REF | | |
| b | 0.18 | 0.25 | 0.30 | 0.007 | 0.010 | 0.012 |
| D/E | 7.00BSC | | | 0.276BSC | | |
| $D_1/E_1$ | 6.75BSC | | | 0.266BSC | | |
| $D_2/E_2$ | 4.80 | 5.05 | 5.30 | 0.189 | 0.199 | 0.209 |
| e | 0.50BSC | | | 0.020BSC | | |
| L | 0.30 | 0.40 | 0.50 | 0.012 | 0.016 | 0.020 |
| θ | 0° | — | 14° | 0° | — | 14° |
| aaa | — | — | 0.15 | — | — | 0.006 |
| bbb | — | — | 0.10 | — | — | 0.004 |
| ccc | — | — | 0.10 | — | — | 0.004 |
| ddd | — | — | 0.05 | — | — | 0.002 |
| eee | — | — | 0.08 | — | — | 0.003 |
| fff | — | — | 0.10 | — | — | 0.004 |

Notes：

1. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION.
2. CONTROLLING DIMENSION：MILLIMETER(mm).
3. REFERENCE DOCUMENTL：JEDEC MO-220.

# 6. Ordering Information

| Part | ADC | DVI | OD | Resolution | Output | PKG |
|---|---|---|---|---|---|---|
| **RTD2525LRH** | 210M | Y | Y | 1440x900 | LVDS | QFN48 |
| **RTD2545LRH** | 210M | Y | Y | 1680x1050 | LVDS | QFN48 |
| **RTD2555LRH** | 210M | Y | Y | 1920x1050 | LVDS | QFN48 |