# Allwinner DE2.0 Specification

**Revision 1.0**

**Jan.23, 2018**

# Revision History

| Revision | Date | Description |
|---|---|---|
| 1.0 | Jan. 23,2018 | Initial Release Version |
| | | |

# Glossary

The glossary is intended to cover the acronyms used in the document.

| Term | Definition |
|---|---|
| R | Read only/non-Write |
| R/W | Read/Write |
| RC | Read-Clear/non-Write |
| RS | Read-Set/non-Write |
| RC/W | Read-Clear/Write |
| RS/W | Read-Set/Write |
| R/WAC | Read/Write-Automatic-Clear，Clear the bit automatically when the operation of complete.Writing 0 has no effect. |
| R/WC | Read/Write-Clear |
| R/WS | Read/Write-Set |
| RC/WS | Read-Clear/Write-Set |
| RS/WC | Read-Set/Write-Clear |
| R/W1C | Read/Write 1 to Clear, Write 0 has non-effect |
| R/W1S | Read/Write 1 to Set, Write 0 has non-effect |
| R/W1T | Read/Write 1 to Flip, Write 0 has non-effect |
| R/W0C | Read/Write 0 to Clear, Write 1 has non-effect |
| R/W0S | Read/Write 0 to Set, Write 1 has non-effect |
| R/W0T | Read/Write 0 to Flip, Write 1 has non-effect |
| RC/W1S | Read-Clear/Write 1 to Set, Write 0 has non-effect |
| RS/W1C | Read-Set/Write 1 to Clear, Write 0 has non-effect |
| RC/W0S | Read-Clear/Write 0 to Set, Write 1 has non-effect |
| RS/W0C | Read-Set/Write 0 to Clear, Write 1 has non-effect |
| W | Write only/non-Read |
| WC | Write-Clear/non-Read |
| WS | Write-Set/non-Read |
| W1 | After reset, Write at the first time, non-Write after the first time/Read |
| WO1 | After reset, Write at the first time, non-Write after the first time/non-Read |

# Contents



# Figures



# Tables

# 1 .H8 Display_Engine_Top

## 1.1 Overview

The Display Engine(DE) is a hardware composer to transfer image layers from a local bus or a video buffer to the *LCD* interface, The DE support four overlay windows to blending, and support image post-processing in the video channel.

**Feature:**

- Support output size up to 4096x4096
- Support four alpha blending channels for main display, two channels for aux display.
- Support four overlay layers in each channel, and has a independent scaler.
- Support potter-duff compatible blending operation.
- Support input format YUV422/YUV420/YUV411/ARGB8888/XRGB8888/RGB888/ARGB4444/ ARGB1555 and RGB565.
- Support Frame Packing/Top-and-Bottom/Side-by-Side Full/Side-by-Side Half 3D format data.
- Support SmartColor2.0 for excellent display experience.
  - Adaptive edge sharping
  - Adaptive color enhancement
  - Adaptive contrast enhancement and fresh tone rectify.
- Support write back and rotation for high efficient dual display and miracast.

## 1.2 Block Diagram

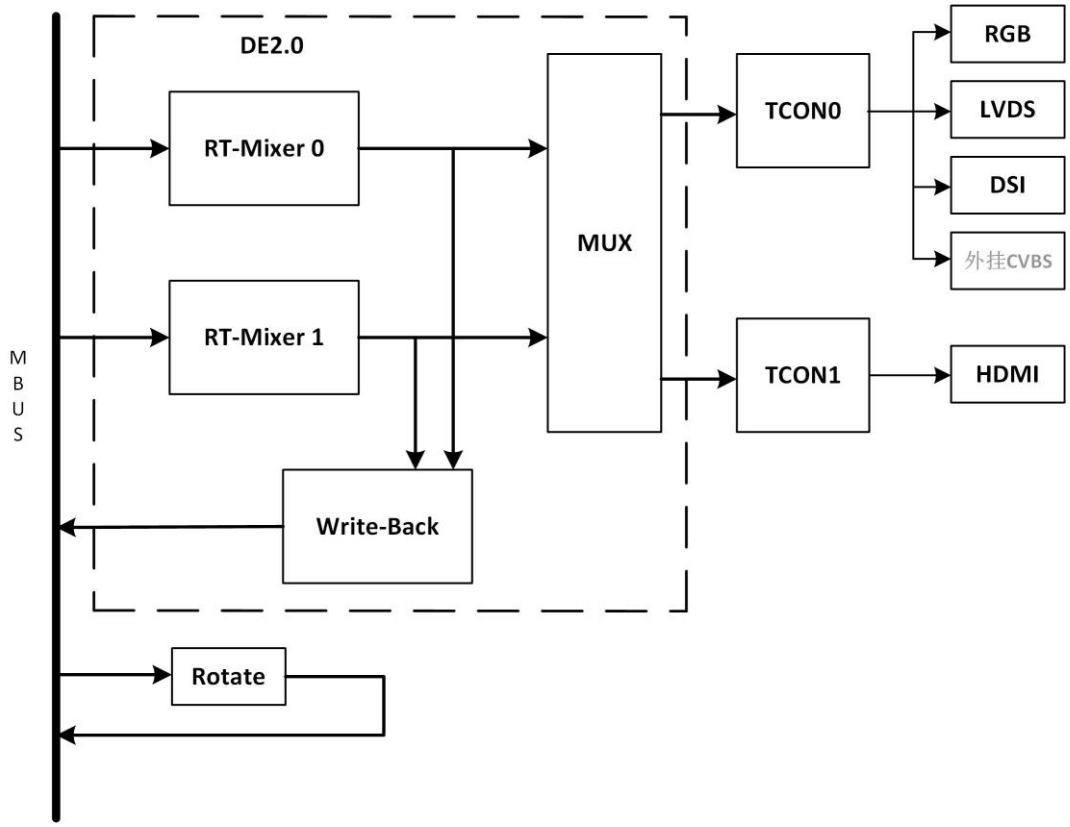



**Figure 1-1. Display Top Level Block Diagram**

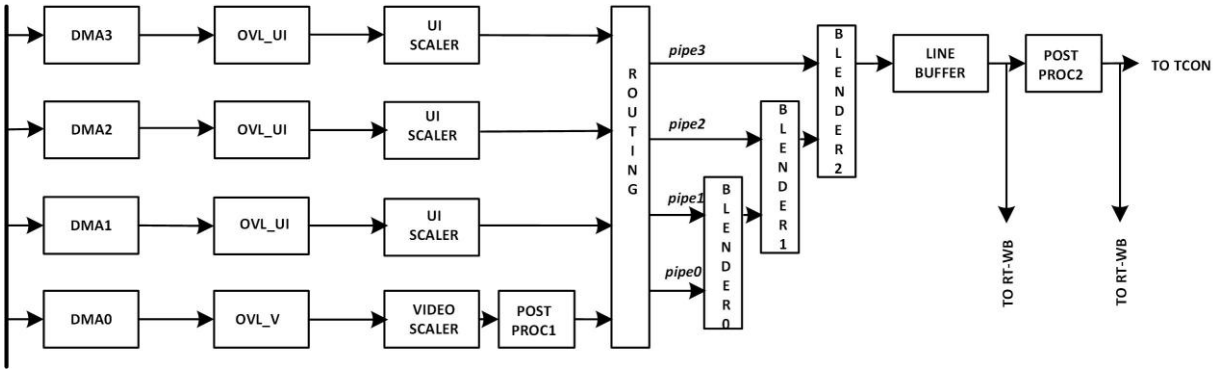



**Figure 1-2. RT-MIXER0 Block Diagram**

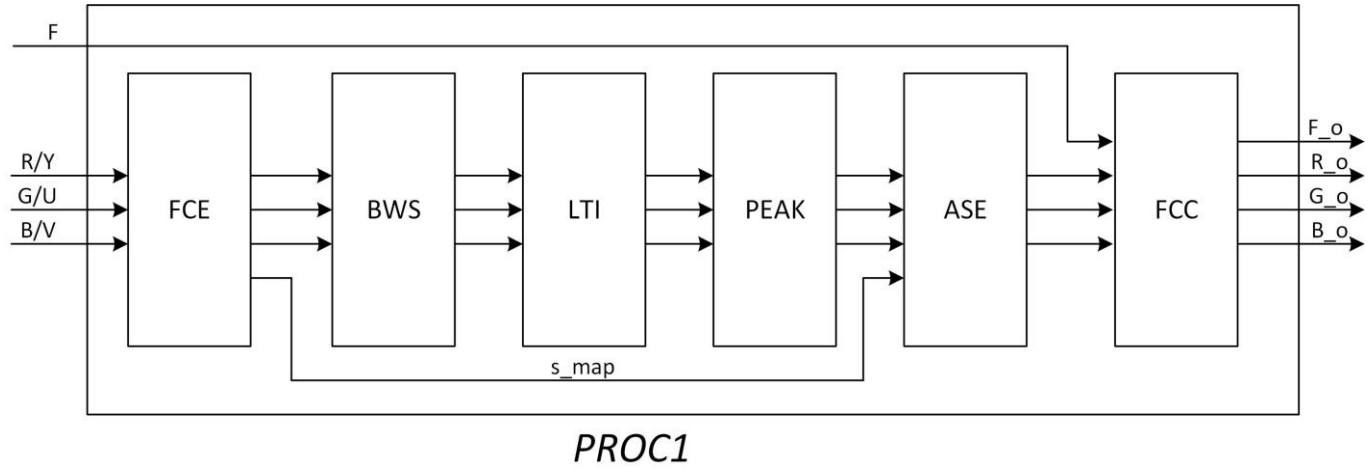



**Figure 1-3. POST_PROC1 Block Diagram**

**DE2.0 post processing, include:**

| | |
|---|---|
| FCE | Fresh and Contrast enhancement |
| BWS | Black and white stretch |
| LTI | Luminance transient improvement |
| PEAK | Luma Peaking |
| ASE | Adaptive Saturation Enhancement |
| FCC | Fancy color curvature change |

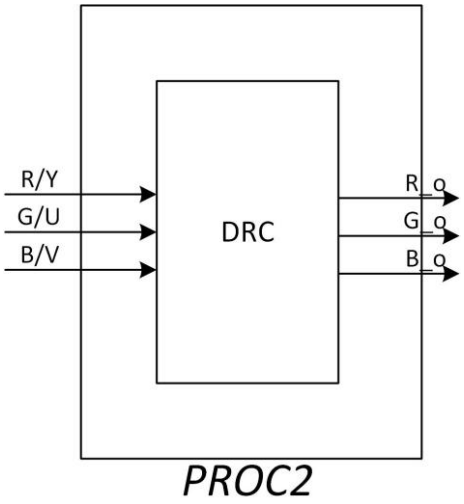



**Figure 1-4. POST_PROC2 Block Diagram**

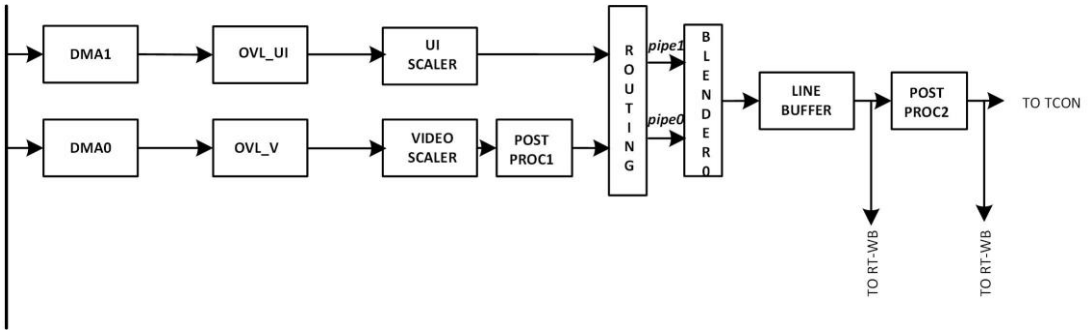



**Figure 1-5. RT-MIXER1 Block Diagram**

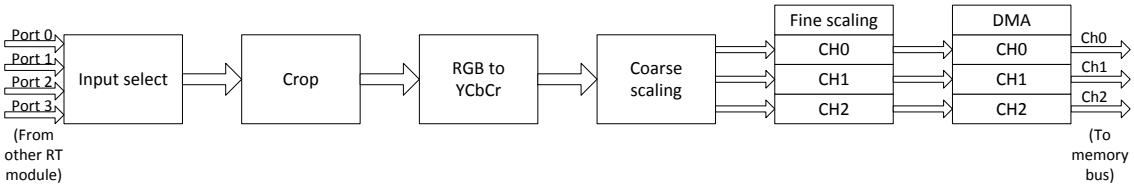



**Figure 1-6. RT-WB Block Diagram**

- Support RGB888 and YUV444 input data format.
- Support input size from 8x4 to 2048x2048.
- Support output size from 8x4 to 2048x2048.
- Support fine down scaling ratio from 1x to 1/2x, and the anti-aliasing filter is 16-phase 4-tap in horizontal, 16-phase 2-tap/4-tap filter in vertical.
- Support coarse down scaling.

## 1.3 Display system configurations and requirements

### 1.3.1 Memory Space Arrangement:

| Module name | Base address | |
|---|---|---|
| DE | 0x01000000 | 0x013FFFFF |

| Module name | Offset Address | Memory Range |
|---|---|---|
| Display system | 0x000000 | 64K |
| RT-WB | 0x010000 | 64K |
| Rotate | 0x020000 | 64K |
| DI | 0x030000 | 128K |
| RT-Mixer0 | 0x100000 | 1M |
| RT-Mixer1 | 0x200000 | 1M |

## 1.4 Register list

**Display system top register list:**

| Register name | Offset | Description |
|---|---|---|

| | | |
|---|---|---|
| SCLK_GATE | 0x000 | **DE SCLK Gating Register** |
| HCLK_GATE | 0x004 | **DE HCLK Gating Register** |
| AHB_RESET | 0x008 | **DE AHB Reset register** |
| SCLK_DIV | 0x00C | **DE SCLK Division register** |
| DE2TCON_MUX | 0x010 | **DE2TCON MUX register** |
| CMD_CTL | 0x014 | **CMD Control register** |
| DI_CTL | 0x01C | **DI Control register** |

## 1.5 Register Description

### 1.5.1 SCLK_GATE

| **Offset: 0x000** | | | **Register Name: SCLK_GATE** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:4 | / | / | Reserved |
| 3 | R/W | 0x0 | ROT_SCLK_GATE<br>0: clock gate<br>1: clock pass |
| 2 | R/W | 0x0 | RT_WB_SCLK_GATE<br>0: clock gate<br>1: clock pass |
| 1 | R/W | 0x0 | CORE1_SCLK_GATE<br>0: clock gate<br>1: clock pass |
| 0 | R/W | 0x0 | CORE0_SCLK_GATE<br>0: clock gate<br>1: clock pass |

### 1.5.2 HCLK_GATE

| **Offset: 0x004** | | | **Register Name: HCLK_GATE** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 29:4 | / | / | Reserved |
| 3 | R/W | 0x0 | ROT_HCLK_GATE<br>0: clock gate<br>1: clock pass |
| 2 | R/W | 0x0 | RT_WB_HCLK_GATE<br>0: clock gate<br>1: clock pass |
| 1 | R/W | 0x0 | CORE1_HCLK_GATE<br>0: clock gate<br>1: clock pass |
| 0 | R/W | 0x0 | CORE0_HCLK_GATE |

| | | | |
|---|---|---|---|
| | | | 0: clock gate<br>1: clock pass |

### 1.5.3 AHB_RESET

| Offset: 0x008 | | | Register Name: AHB_RESET |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 29:4 | / | / | Reserved |
| 3 | R/W | 0x0 | ROT_SCLK_RESET<br>0: reset on<br>1: reset off |
| 2 | R/W | 0x0 | RT_WB_SCLK_RESET<br>0: reset on<br>1: reset off |
| 1 | R/W | 0x0 | CORE1_SCLK_RESET<br>0: reset on<br>1: reset off |
| 0 | R/W | 0x0 | CORE0_HCLK_RESET<br>0: reset on<br>1: reset off |

### 1.5.4 SCLK_DIV

| Offset: 0x00C | | | Register Name: SCLK_DIV |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:16 | / | / | Reserved |
| 15:12 | R/W | 0x0 | ROT_SCLK_DIV |
| 11:8 | R/W | 0x0 | RT_WB_SCLK_DIV |
| 7:4 | R/W | 0x0 | CORE1_SCLK_DIV |
| 3:0 | R/W | 0x0 | CORE0_SCLK_DIV |

### 1.5.5 DE2TCON_MUX

| Offset: 0x010 | | | Register Name: DE2TCON_MUX |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:1 | / | / | Reserved |
| 0 | R/W | 0x0 | DE2TCON_MUX<br>0: MIXER0-〉TCON0；MIXER1-〉TCON1<br>1: MIXER0-〉TCON1；MIXER1-〉TCON0 |

### 1.5.6 CMD_CTL

| Offset: 0x014 | | | Register Name: CMD_CTL |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:20 | / | / | Reserved |
| 19:16 | R/W | 0x0 | RT_WB_CMD_CTL |
| 15:8 | / | / | / |
| 7:4 | R/W | 0x0 | CORE1_CMD_CTL |
| 3:0 | R/W | 0x0 | CORE0_CMD_CTL |

**Note: The number of sending command to DRAM are about N+2.**

### 1.5.7 DI_CTL

| Offset: 0x01C | | | Register Name: DI_CTL |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:5 | / | / | / |
| 4 | R/W | 0x0 | CORE1_DI_CTL<br>0: DI disable<br>1: DI enable<br>Note: if DI enable, VSU use DI data, otherwise use overlay data. |
| 3:0 | / | / | / |
| 0 | R/W | 0x0 | CORE0_DI_CTL<br>0: DI disable<br>1: DI enable<br>Note: if DI enable, VSU use DI data, otherwise use overlay data. |

# 2 .H3 Display_Engine_Top

## 2.1 Overview

- Support input layer size up to 2048x2048, and output size up to 2048x2048.
- Support four alpha blending channel for main display, two channel for aux display.
- Support 4 overlay layer in each channel, and has a independent scaler.
- Support potter-duff compatible blending operation.
- Support display enhancement 2.0 for excellent display experience.
- Support write back for high efficient dual display.

## 2.2 Block Diagram

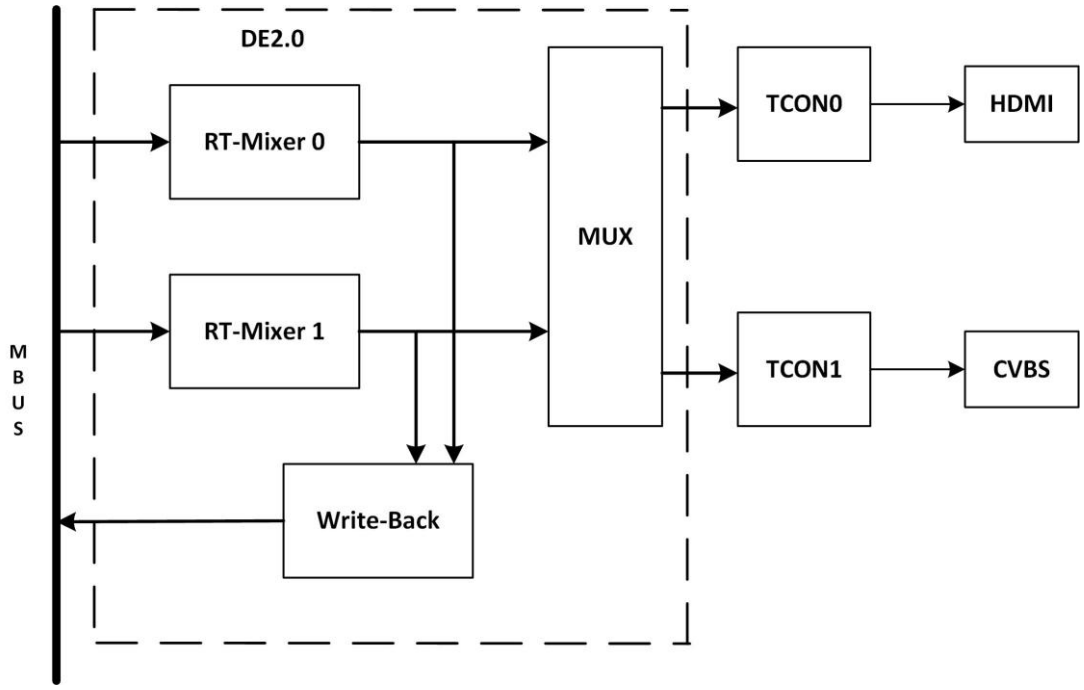



**Figure 2-1. Display Top Level Diagram**

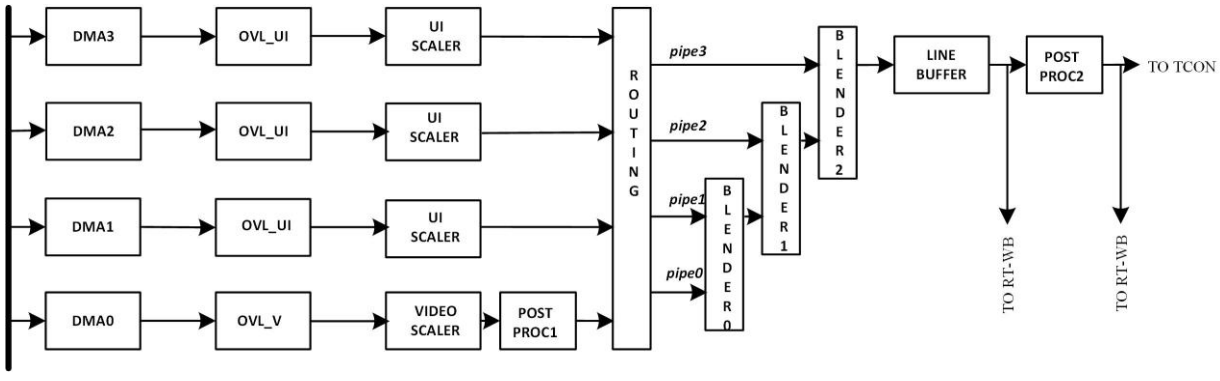



**Figure 2-2. RT-MIXER0 Block Diagram**

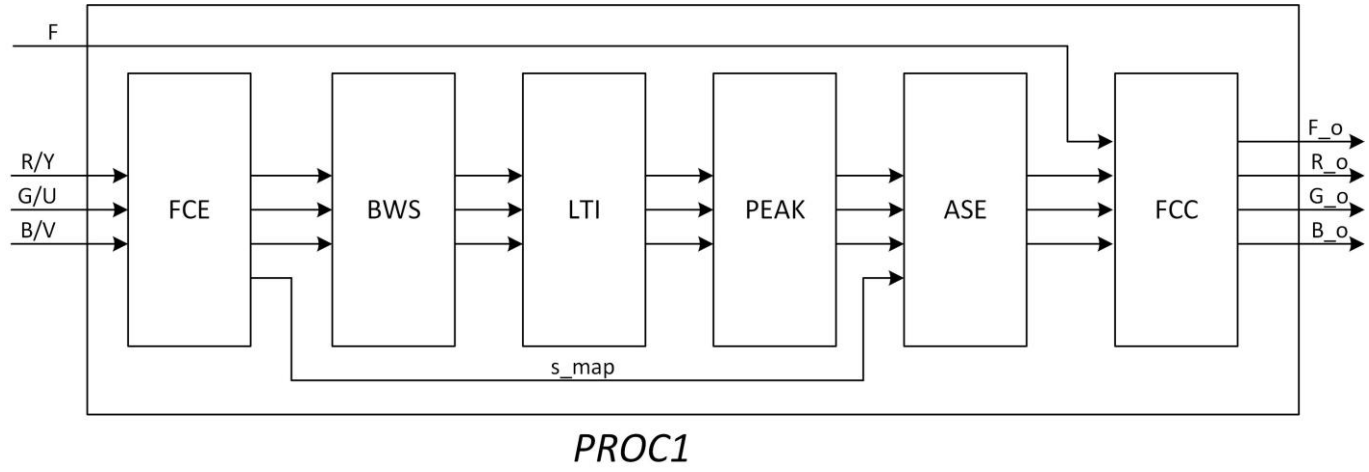



**Figure 2-3. POST_PROC1 Block Diagram**

DE2.0 post processing, include:

FCE : Fresh and Contrast enhancement

BWS : Black and white stretch

LTI : Luminance transient improvement

PEAK: Luma Peaking

ASE : Adaptive Saturation Enhancement

FCC : Fancy color curvature change

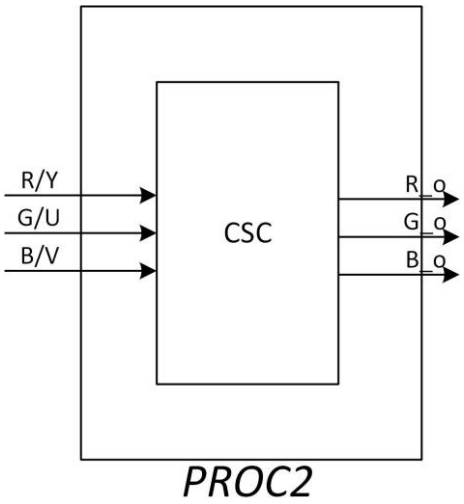



**Figure 2-4. POST_PROC2 Block Diagram**

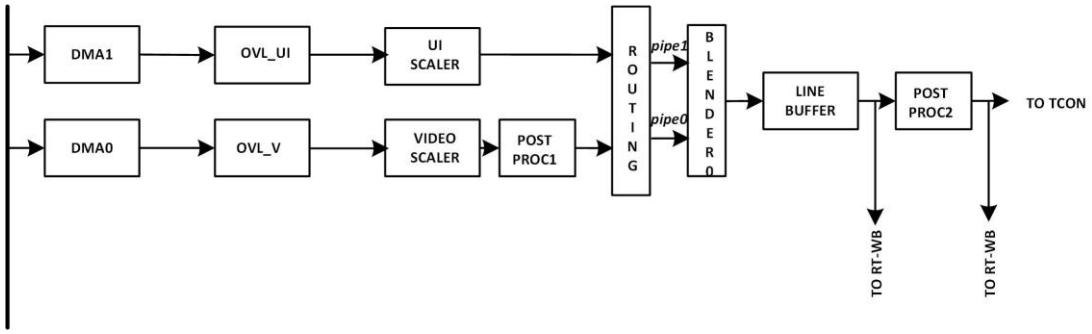



**Figure 2-5. RT-MIXER1 Block Diagram**

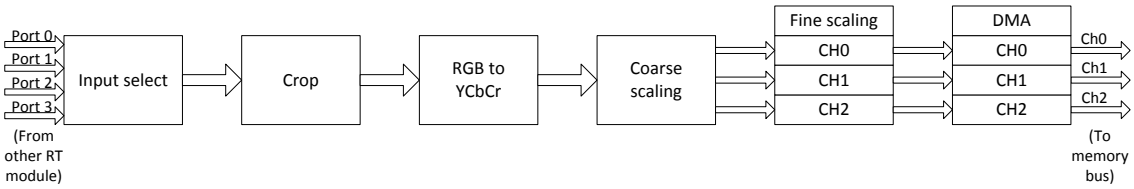



**Figure 2-6. RT-WB Block Diagram**

- Support RGB888 and YUV444 input data format.
- Support input size from 8x4 to 2048x2048.
- Support output size from 8x4 to 2048x2048.
- Support fine down scaling ratio from 1x to 1/2x, and the anti-aliasing filter is 16-phase 4-tap in horizontal, 16-phase 2-tap/4-tap filter in vertical.
- Support coarse down scaling.

## 2.3 Display System Configurations and requirements

### 2.3.1 Speed requirements:

| Module Name | Speed(Mhz) | Description |
|---|---|---|
| RT-MIXER0 | 250 | |
| RT-MIXER1 | 250 | |
| RT-WB | 250 | |

### 2.3.2 Memory Space Arrangement:

| Module name | Base address | |
|---|---|---|
| DE | 0x01000000 | 0x012FFFFF |

| Module name | Memory Range | Offset Address |
|---|---|---|
| Display system | 64K | 0x000000 |
| RT-WB | 64K | 0x010000 |
| RT-Mixer0 | 1M | 0x100000 |
| RT-Mixer1 | 1M | 0x200000 |

## 2.4 Register list

Display system top register list:

| Register name | Offset | Description |
|---|---|---|
| SCLK_GATE | 0x000 | |
| HCLK_GATE | 0x004 | |
| AHB_RESET | 0x008 | |

| | | |
|---|---|---|
| SCLK_DIV | 0x00C | |
| DE2TCON_MUX | 0X010 | |

## 2.5 Register Description

### 2.5.1 SCLK_GATE

| Offset: 0x000 | | | Register Name: SCLK_GATE |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:3 | / | / | Reserved |
| 2 | R/W | 0x0 | RT_WB_SCLK_GATE<br>0: clock gate<br>1: clock pass |
| 1 | R/W | 0x0 | CORE1_SCLK_GATE<br>0: clock gate<br>1: clock pass |
| 0 | R/W | 0x0 | CORE0_SCLK_GATE<br>0: clock gate<br>1: clock pass |

### 2.5.2 HCLK_GATE

| Offset: 0x004 | | | Register Name: HCLK_GATE |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 29:3 | / | / | Reserved |
| 2 | R/W | 0x0 | RT_WB_HCLK_GATE<br>0: clock gate<br>1: clock pass |
| 1 | R/W | 0x0 | CORE1_HCLK_GATE<br>0: clock gate<br>1: clock pass |
| 0 | R/W | 0x0 | CORE0_HCLK_GATE<br>0: clock gate<br>1: clock pass |

### 2.5.3 AHB_RESET

| Offset: 0x008 | | | Register Name: AHB_RESET |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 29:4 | / | / | Reserved |
| 2 | R/W | 0x0 | RT_WB_SCLK_RESET |

| | | | |
|---|---|---|---|
| | | | 0: reset on<br>1: reset off |
| 1 | R/W | 0x0 | CORE1_SCLK_RESET<br>0: reset on<br>1: reset off |
| 0 | R/W | 0x0 | CORE0_HCLK_RESET<br>0: reset on<br>1: reset off |

### 2.5.4 SCLK_DIV

| Offset: 0x00C | | | Register Name: SCLK_DIV |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:12 | / | / | Reserved |
| 11:8 | R/W | 0x0 | RT_WB_SCLK_DIV |
| 7:4 | R/W | 0x0 | CORE1_SCLK_DIV |
| 3:0 | R/W | 0x0 | CORE0_SCLK_DIV |

### 2.5.5 DE2TCON_MUX

| Offset: 0x010 | | | Register Name: DE2TCON_MUX |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:1 | / | / | Reserved |
| 0 | R/W | 0x0 | DE2TCON_MUX<br>0: MIXER0-〉TCON0； MIXER1-〉TCON1<br>1: MIXER0-〉TCON1； MIXER1-〉TCON0 |

### 2.5.6 CMD_CTL

| Offset: 0x014 | | | Register Name: CMD_CTL |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:20 | / | / | Reserved |
| 19:16 | R/W | 0x0 | RT_WB_CMD_CTL |
| 15:8 | / | / | / |
| 7:4 | R/W | 0x0 | CORE1_CMD_CTL |
| 3:0 | R/W | 0x0 | CORE0_CMD_CTL |

**Note: The number of sending command to dram are about N+2**

# 3 .H5 Display_Engine_Top

## 3.1 Overview

The Display Engine(DE) is a hardware composer to transfer image layers from a local bus or a video buffer to the *LCD* interface, The DE support four overlay windows to blending, and support image post-processing in the video channel. The display system block diagram show as the Figure1-1.

**Feature:**

- Support output size up to 4096x4096
- Support four alpha blending channels for main display, two channels for aux display.
- Support four overlay layers in each channel, and has a independent scaler.
- Support potter-duff compatible blending operation.
- Support input format YUV422/YUV420/YUV411/ARGB8888/XRGB8888/RGB888/ARGB4444/ ARGB1555 and RGB565.
- Support Frame Packing/Top-and-Bottom/Side-by-Side Full/Side-by-Side Half 3D format data.
- Support SmartColor2.0 for excellent display experience.
  - Adaptive edge sharping
  - Adaptive color enhancement
  - Adaptive contrast enhancement and fresh tone rectify.
- Support write back for high efficient dual display and miracast.

## 3.2 Block Diagram

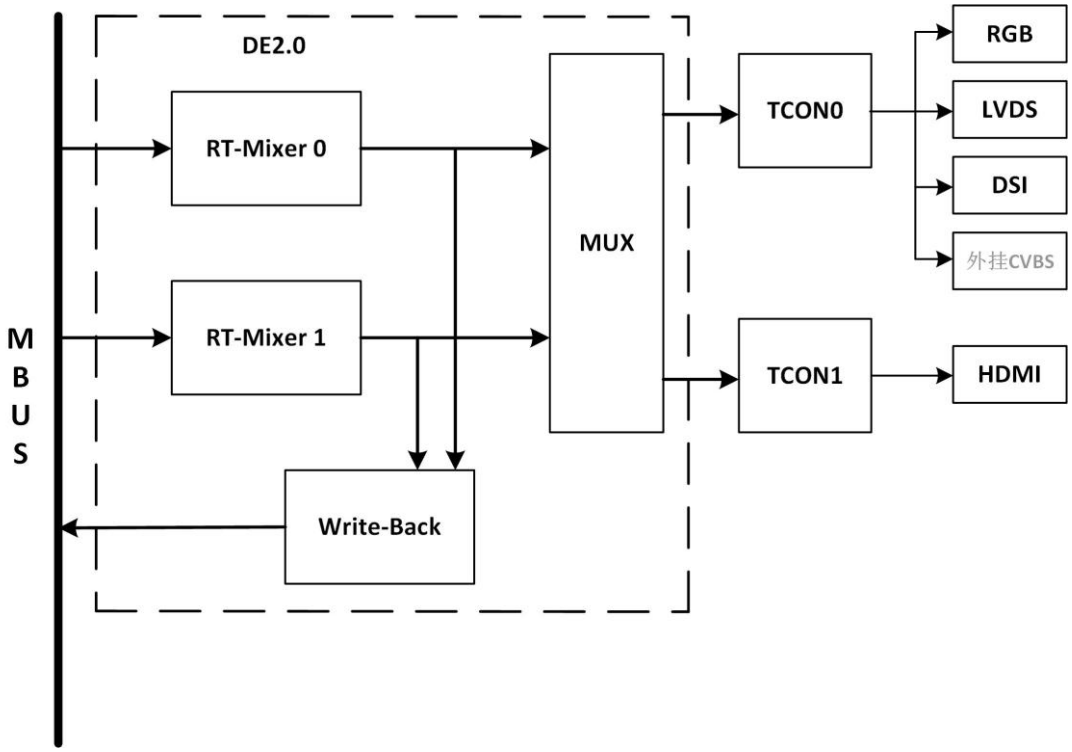



Figure 3-1. DE System Block Diagram

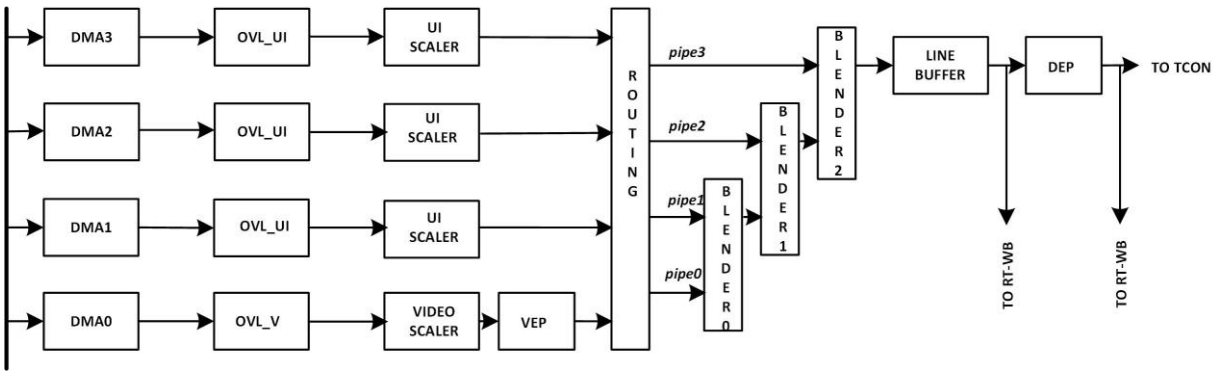



Figure 3-2. RT-Mixer0 Blcok Diagram

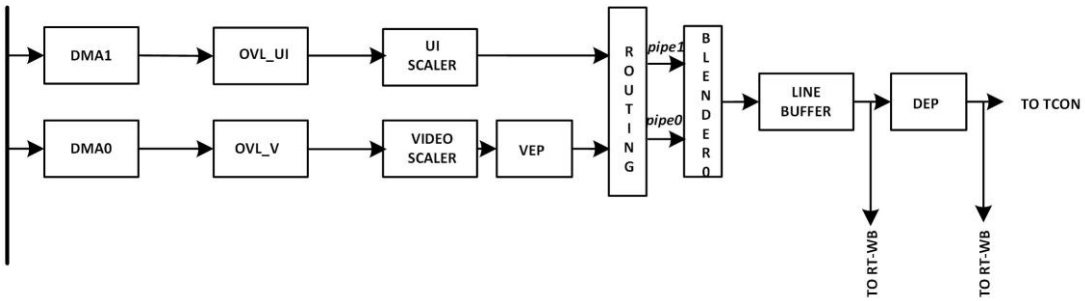



Figure 3-3. RT-Mixer1 Block Diagram

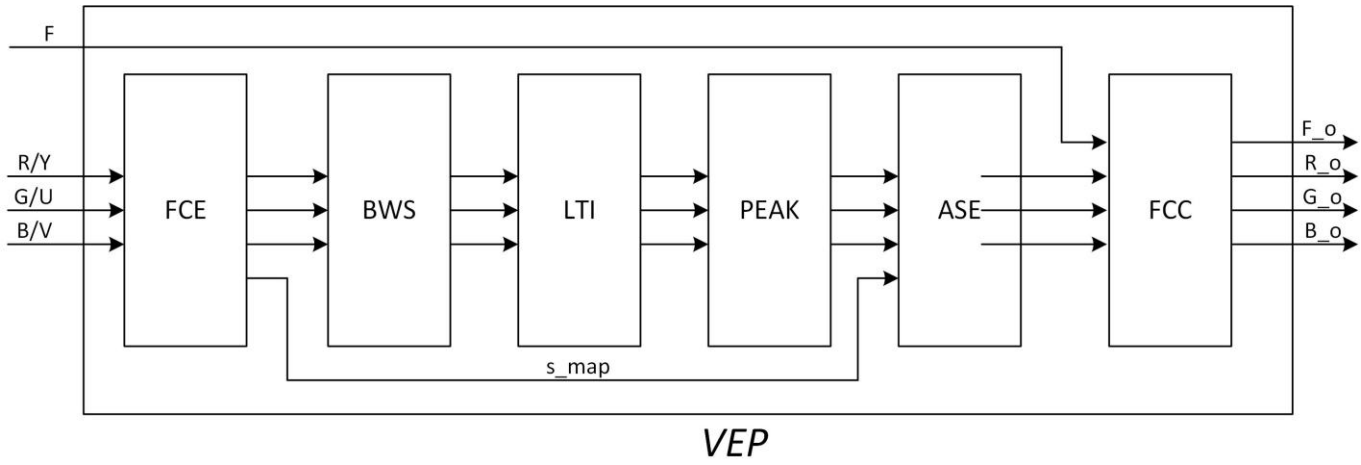



**Figure 3-4. VEP Block Diagram**

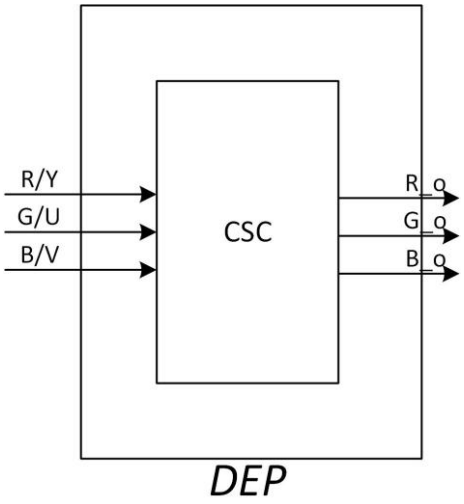



**Figure 3-5. DEP Block Diagram**

# 3.3 Operations and Functional Descriptions

## 3.3.1 Configuration and Requirements

The following table describes the configuration and speed of ***DE System***.

**Table 3-1. RT-Mixer Configuration**

| Module | Pipe | Overlay | FBD | DI | DNR | Scaler Type/ Line Buffer | VEP | Blend | DEP |
|---|---|---|---|---|---|---|---|---|---|
| **RT-Mixer0** | CH0 | 4 layers | N | N | N | VScaler/4096 | Y | Y | CSC |
| | CH1 | 4 layers | N | N | N | GScaler/2048 | N | | |
| | CH2 | 4 layers | N | N | N | GScaler/2048 | N | | |
| | CH3 | 4 layers | N | N | N | GScaler/2048 | N | | |
| **RT-Mixer1** | CH0 | 4 layers | N | N | N | VScaler/2048 | CSC | Y | CSC |
| | CH1 | 4 layers | N | N | N | GScaler/2048 | N | | |

Table 3-2. RT-WB Configuration

| Module | Input | Scaler Type/ Line Buffer |
|---|---|---|
| **RT-WB** | RT-Mixer0 Blending output<br>RT-Mixer0 Post-Proc2 output<br>RT-Mixer1 Blending output<br>RT-Mixer1 Post-Proc2 output | GScaler/2048 |

Table 3-3. DE Memory and Speed requirements

| Module Name | Base Address | Memory Range | Speed(MHz) |
|---|---|---|---|
| **Display System** | 0x01000000 | 32K | 432M |
| **RT-WB** | 0x01010000 | 32K | 432M |
| **RT-Mixer0** | 0x01100000 | 1M | 432M |
| **RT-Mixer1** | 0x01200000 | 1M | 432M |

### 3.3.2 Clock Sources

Display Engine controller get two different clocks,Users can select one of them to make DE Clock Source.The following table describes the clock sources for DE.Users can see ***Clock Controller Unit(CCU)*** for clock setting, configuration and gating information.

Table 3-4. DE Clock Sources

| Clock Sources | Description |
|---|---|
| PLL_DE | PLL_DE,default value is 250MHz for DE. |
| PLL_PERIPH0(2x) | Peripheral Clock0 source, divide to about 250MHz for DE. |

## 3.4 DE Register List

| Module Name | Base Address |
|---|---|
| DE | 0x01000000 |

| Register Name | Offset | Description |
|---|---|---|
| DE_SCLK_GATE | 0x00 | DE SCLK Gating Register |
| DE_HCLK_GATE | 0x04 | DE HCLK Gating Register |
| DE_AHB_RESET | 0x08 | DE AHB Reset register |
| DE_SCLK_DIV | 0x0c | DE SCLK Division register |
| DE2TCON_MUX | 0x10 | DE2TCON MUX register |

## 3.5 DE Register Description

### 3.5.1 DE SCLK Gating Register(Default Value: 0x0000_0000)

| Offset: 0x00 | | | Register Name: DE_SCLK_GATE |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:3 | / | / | Reserved |
| 2 | R/W | 0x0 | RT_WB_SCLK_GATE<br>0: clock gate<br>1: clock pass |
| 1 | R/W | 0x0 | CORE1_SCLK_GATE<br>0: clock gate<br>1: clock pass |
| 0 | R/W | 0x0 | CORE0_SCLK_GATE<br>0: clock gate<br>1: clock pass |

### 3.5.2 DE HCLK Gating Register(Default Value: 0x0000_0000)

| Offset: 0x04 | | | Register Name: DE_HCLK_GATE |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:3 | / | / | / |
| 2 | R/W | 0x0 | WB_HCLK_GATE<br>0: clock gate<br>1: clock pass |
| 1 | R/W | 0x0 | CORE1_HCLK_GATE<br>0: clock gate<br>1: clock pass |
| 0 | R/W | 0x0 | CORE0_HCLK_GATE<br>0: clock gate<br>1: clock pass |

### 3.5.3 DE AHB Reset Register(Default Value: 0x0000_0000)

| Offset: 0x08 | | | Register Name: DE_AHB_RESET |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:3 | / | / | / |
| 2 | R/W | 0x0 | RT_WB_SCLK_RESET<br>0: reset on<br>1: reset off |

| | | | |
|---|---|---|---|
| 1 | R/W | 0x0 | CORE1_SCLK_RESET<br>0: reset on<br>1: reset off |
| 0 | R/W | 0x0 | CORE0_HCLK_RESET<br>0: reset on<br>1: reset off |

### 3.5.4 DE SCLK Division Register(Default Value: 0x0000_0000)

| Offset: 0x0C | | | Register Name: DE_SCLK_DIV |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:12 | / | / | / |
| 11:8 | R/W | 0x0 | WB_SCLK_DIV |
| 7:4 | R/W | 0x0 | CORE1_SCLK_DIV |
| 3:0 | R/W | 0x0 | CORE0_SCLK_DIV |

### 3.5.5 DE2TCON MUX Register(Default Value: 0x0000_0000)

| Offset: 0x10 | | | Register Name: DE2TCON_MUX |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | DE2TCON_MUX<br>0: MIXER0-〉TCON0； MIXER1-〉TCON1<br>1: MIXER0-〉TCON1； MIXER1-〉TCON0 |

# 4 .A83 Display_Engine_Top

## 4.1 Overview

- Support output size up to 2048x2048
- Support four alpha blending channels for main display, two channels for aux display.
- Support four overlay layers in each channel, and has a independent scaler.
- Support potter-duff compatible blending operation.
- Support input format YUV422/YUV420/YUV411/ARGB8888/XRGB8888/RGB888/ARGB4444/ ARGB1555 and RGB565.
- Support Frame Packing/Top-and-Bottom/Side-by-Side Full/Side-by-Side Half 3D format data.
- Support SmartColor2.0 for excellent display experience.
  - Adaptive edge sharping
  - Adaptive color enhancement
  - Adaptive contrast enhancement and fresh tone rectify.
- Support write back and rotation for high efficient dual display and miracast.

## 4.2 Block Diagram

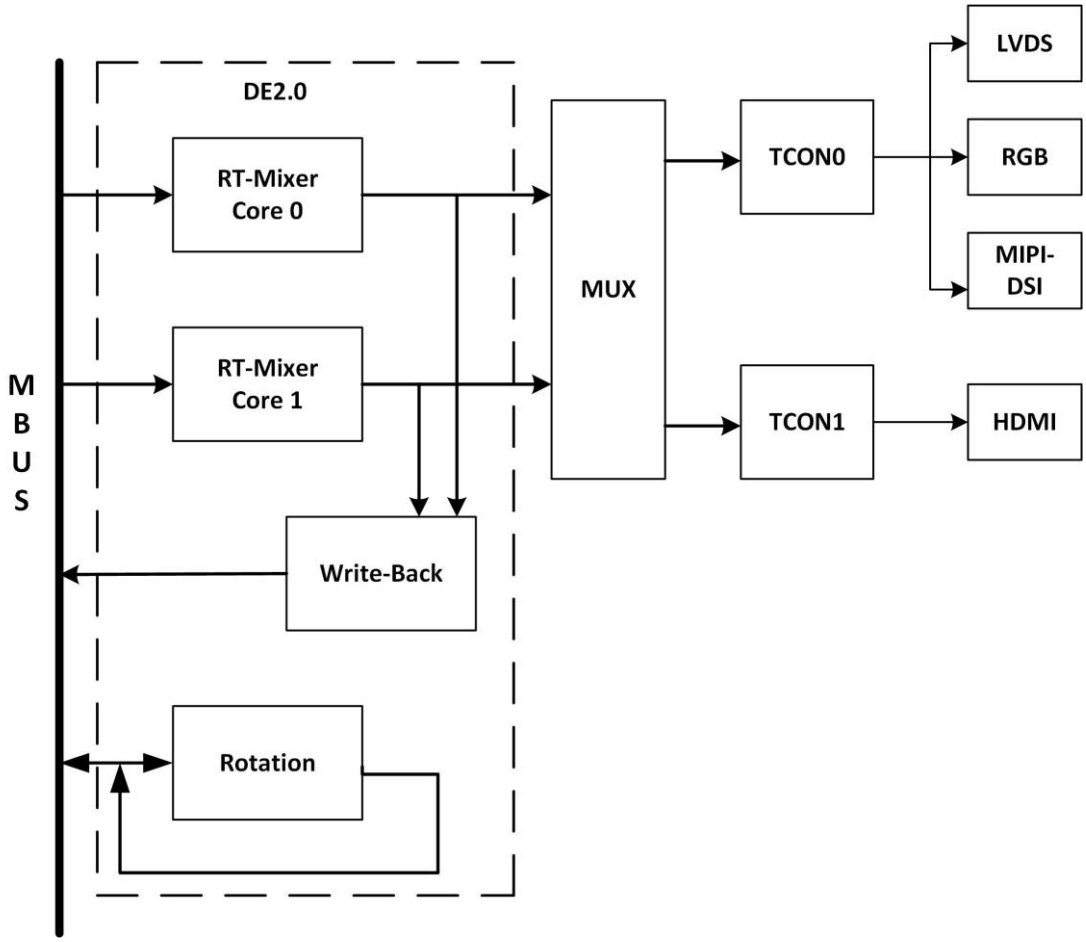



Figure 4-1. DE System Block Diagram

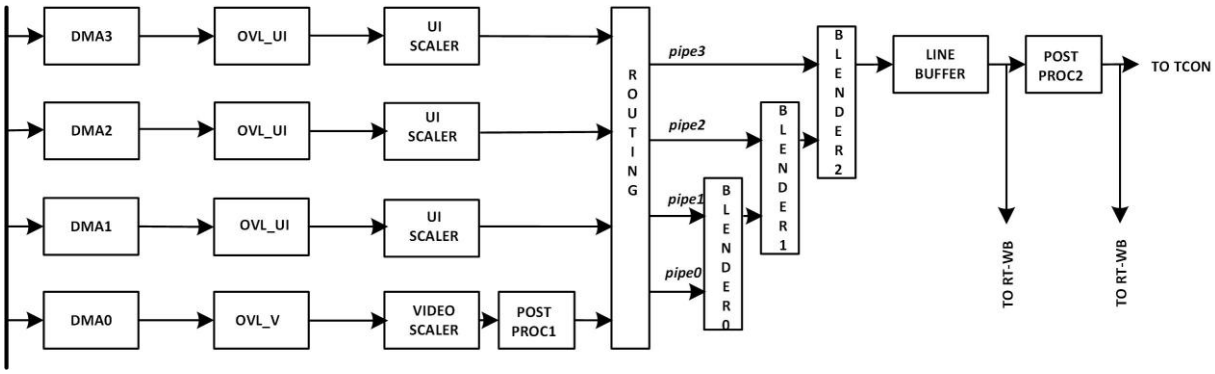



Figure 4-2. RT-MIXER Core0 Block Diagram

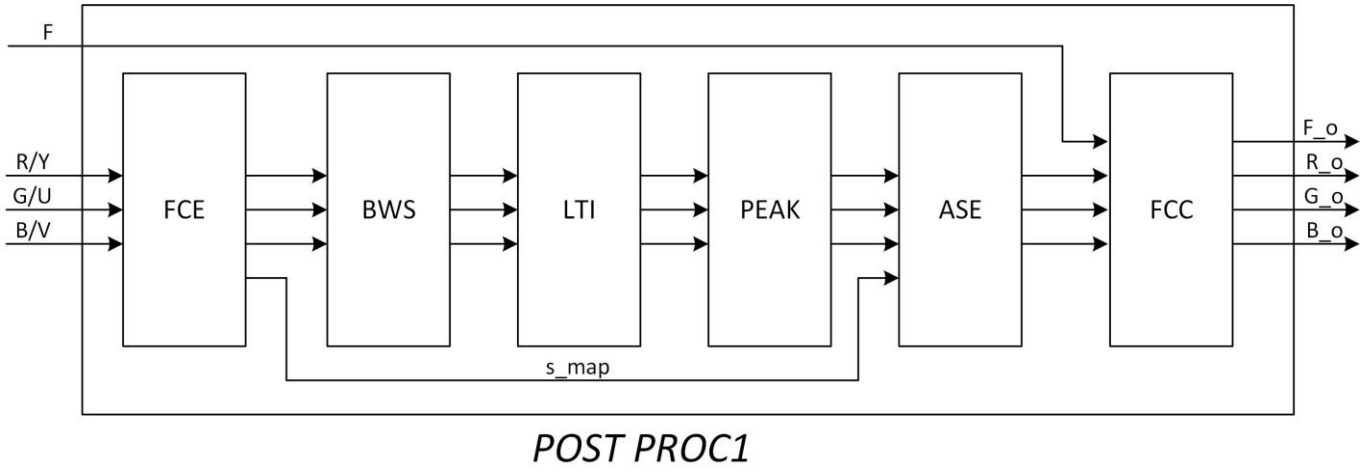



**Figure 4-3. POST_PROC1 Block Diagram**

**DE2.0 post processing, include:**

- FCE : Fresh and Contrast enhancement
- BWS : Black and white stretch
- LTI : Luminance transient improvement
- PEAK: Luma Peaking
- ASE : Adaptive Saturation Enhancement
- FCC : Fancy color curvature change

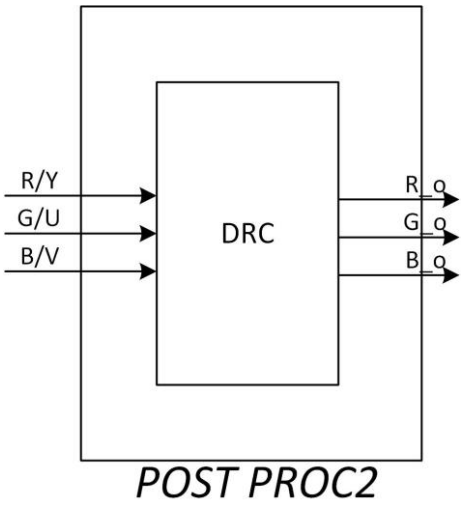



**Figure 4-4. POST_PROC2 Block Diagram**

**NOTE: DRC: Dynamic Range Controller**

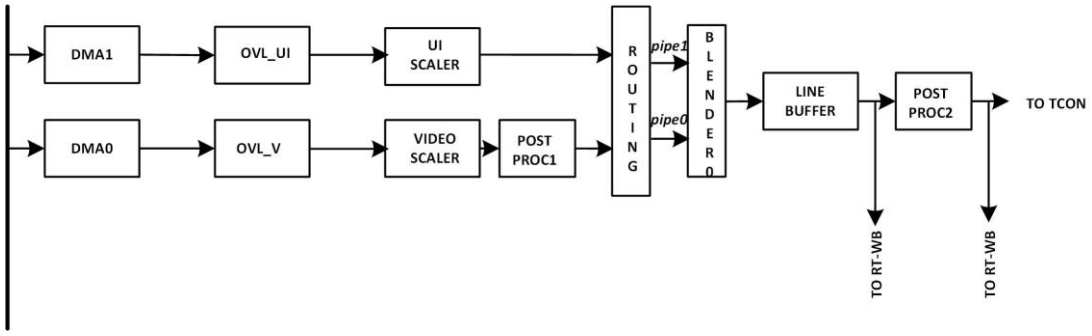



**Figure 4-5. RT-MIXER Core1 Block Diagram**

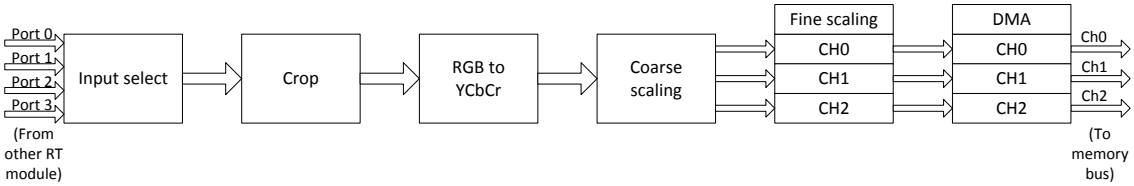



**Figure 4-6. RT-WB Block Diagram**

- Support RGB888 and YUV444 input data format
- Support input size from 8×4 to 4096×4096
- Support output size from 8×4 to 1920*1200
- Support fine down scaling ratio from 1× to 1/2×, and the anti-aliasing filter is 16-phase 4-tap in horizontal , 16-phase 2-tap/4-tap filter in vertical
- Support coarse down scaling.

**Memory to Memory rotation**

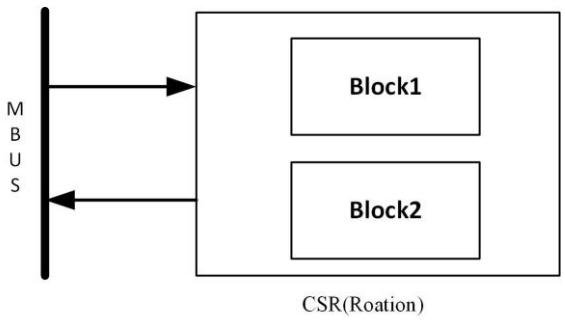



**Figure 4-7. CSR(Rotation) Block Diagram**

**Input Format：**

- 1.16/32bit of RGB and ARGB type，
- YUV422 interleave

- YUV420 combine and planar
- 4.24bit RGB

**Output Format：**

- If the input is RGB type, the output format is same as the input format
- If the input format is YUV type, the output format is YUV planar(YUV is stored at 3 address, respectively)

## 4.3 Display system configurations and requirements

### 4.3.1 Memory Space Arrangement:

| Module name | Base address | |
|---|---|---|
| DE | 0x01000000 | 0x013F0000 |

| Module name | Memory Range | Offset Address |
|---|---|---|
| Display system | 64K | 0x000000 |
| RT-WB | 64K | 0x010000 |
| CSR | 64K | 0x020000 |
| RT-Mixer0 | 1M | 0x100000 |
| RT-Mixer1 | 1M | 0x200000 |

## 4.4 Register list

Display system top register list:

| Register name | Offset | Description |
|---|---|---|
| SCLK_GATE | 0x000 | |
| HCLK_GATE | 0x004 | |
| AHB_RESET | 0x008 | |
| SCLK_DIV | 0x00C | |
| DE2TCON_MUX | 0X010 | |

## 4.5 Register Description

### 4.5.1 SCLK_GATE

| Offset: 0x000 | Register Name: SCLK_GATE |
|---|---|

| Bit | Read/Write | Default/Hex | Description |
|---|---|---|---|
| 31:4 | / | / | Reserved |
| 3 | R/W | 0x0 | CSR_SCLK_GATE<br>0: clock gate<br>1: clock pass |
| 2 | R/W | 0x0 | RT_WB_SCLK_GATE<br>0: clock gate<br>1: clock pass |
| 1 | R/W | 0x0 | CORE1_SCLK_GATE<br>0: clock gate<br>1: clock pass |
| 0 | R/W | 0x0 | CORE0_SCLK_GATE<br>0: clock gate<br>1: clock pass |

### 4.5.2 HCLK_GATE

| Offset: 0x004 | | | Register Name: HCLK_GATE |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 29:4 | / | / | Reserved |
| 3 | R/W | 0x0 | CSR_HCLK_GATE<br>0: clock gate<br>1: clock pass |
| 2 | R/W | 0x0 | RT_WB_HCLK_GATE<br>0: clock gate<br>1: clock pass |
| 1 | R/W | 0x0 | CORE1_HCLK_GATE<br>0: clock gate<br>1: clock pass |
| 0 | R/W | 0x0 | CORE0_HCLK_GATE<br>0: clock gate<br>1: clock pass |

### 4.5.3 AHB_RESET

| Offset: 0x008 | | | Register Name: AHB_RESET |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 29:4 | / | / | Reserved |
| 3 | R/W | 0x0 | CSR_SCLK_RESET<br>0: reset on<br>1: reset off |
| 2 | R/W | 0x0 | RT_WB_SCLK_RESET<br>0: reset on<br>1: reset off |

| | | | |
|---|---|---|---|
| 1 | R/W | 0x0 | CORE1_SCLK_RESET<br>0: reset on<br>1: reset off |
| 0 | R/W | 0x0 | CORE0_HCLK_RESET<br>0: reset on<br>1: reset off |

### 4.5.4 SCLK_DIV

| **Offset: 0x00C** | | | **Register Name: SCLK_DIV** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:16 | / | / | Reserved |
| 15:8 | R/W | 0x0 | CSR_SCLK_DIV |
| 11:8 | R/W | 0x0 | RT_WB_SCLK_DIV |
| 7:4 | R/W | 0x0 | CORE1_SCLK_DIV |
| 3:0 | R/W | 0x0 | CORE0_SCLK_DIV |

### 4.5.5 DE2TCON_MUX

| **Offset: 0x010** | | | **Register Name:** DE2TCON_MUX |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:1 | / | / | Reserved |
| 0 | R/W | 0x0 | DE2TCON_MUX<br>0: MIXER0-〉TCON0； MIXER1-〉TCON1<br>1: MIXER0-〉TCON1； MIXER1-〉TCON0 |

# 5 .Sub_Module_Specification

## 5.1 DE ASE Specification

### 5.1.1 Register List

| Module name | Memory Range | Offset Address |
|---|---|---|
| ASE | 8K | 0xA8000 |

| Register name | Offset | Description |
|---|---|---|
| ASE_CTRL_REG | 0x000 | Global control register |
| ASE_SIZE_REG | 0x004 | ASE size register |
| ASE_WIN0_REG | 0x008 | Windows top position |
| ASE_WIN1_REG | 0x00c | Windows bottom position |
| ASE_WEIGHT | 0x010 | ASE gain register |

### 5.1.2 Register Description

#### 5.1.2.1 Global Control Register

| Offset: 0x000 | | | Register Name: ASE_CTL_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:2 | / | / | Reserved |
| 1 | R/W | 0x0 | **WINDOW_EN**<br>LCE window function enable<br>0:Disable<br>1:Enable<br>Note: When window function enable, only the area inside the window will be processed. |
| 0 | R/W | 0x0 | **ASE_EN**<br>0: disable<br>1: enable |

#### 5.1.2.2 ASE size register

| Offset: 0x004 | | | Register Name: ASE_SIZE_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:28 | / | / | Reserved |
| 27:16 | R/W | 0x0 | **HEIGHT**<br>The real height = The value of these bits add 1 |

| | | | |
|---|---|---|---|
| 15:12 | / | / | / |
| 11:0 | R/W | 0x0 | **WIDTH**<br>The real width = The value of these bits add 1 |

### 5.1.2.3 ASE windows0 register

| **Offset: 0x008** | | | **Register Name: ASE**_WIN0_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:28 | / | / | / |
| 27:16 | R/W | 0 | **WIN_TOP**<br>Window Top position<br>Top position is the left-top y coordinate of display window in pixels |
| 15:12 | / | / | / |
| 11:0 | R/W | 0 | **WIN_LEFT**<br>Window Left position<br>Left position is left-top x coordinate of display window in pixels |

### 5.1.2.4 ASE windows1 register

| **Offset: 0x00C** | | | **Register Name: ASE**_WIN1_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:28 | / | / | / |
| 27:16 | R/W | 0 | **WIN_BOT**<br>Window Bottom position<br>Bottom position is right-bottom y coordinate of display window in pixels |
| 15:12 | / | / | / |
| 11:0 | R/W | 0 | **WIN_RIGHT**<br>Window Right position<br>Right position is right-bottom x coordinate of display window in pixels |

### 5.1.2.5 ASE gain register

| **Offset: 0x010** | | | **Register Name: ASE_WEIGHT** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:6 | / | / | Reserved |
| 5:0 | R/W | 0x0 | ASE_WEIGHT |

## 5.2 DE BWS Specification

### 5.2.1 Overview

The BWS(Black and White Stetch) features:

- Support data format with 8-bit pre channel.
- Support Maximum input frame size 4096×2160
- Interface: Stream-to-stream pixel interface

### 5.2.2 Register List

| Module name | Memory Range | Offset Address |
|---|---|---|
| **BWS** | 8K | 0xA2000 |

| Register name | Offset | Description |
|---|---|---|
| BWS_GCTRL_REG | 0x000 | Control register |
| BWS_SIZE_REG | 0x004 | Size setting register |
| BWS_WIN0_REG | 0x008 | Window setting register0 |
| BWS_WIN1_REG | 0x00C | Window setting register1 |
| BWS_THR0_REG | 0x020 | BWS threshold setting register0 |
| BWS_THR1_REG | 0x024 | BWS threshold setting register1 |
| BWS_SLP0_REG | 0x028 | BWS slope setting register0 |
| BWS_SLP1_REG | 0x02C | BWS slope setting register1 |

### 5.2.3 Register Description

#### 5.2.3.1 BWS_GCTRL_REG

| **Offset: 0x000** | | | **Register Name:** GCTRL_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31 | R/W | 0x0 | **WINDOW_EN**<br>BWS window function enable<br>0:Disable<br>1:Enable<br>Note: When window function enable, only the area inside the window will be processed. |
| 30:1 | / | / | / |
| 0 | R/W | 0x0 | **EN**<br>BWS module enable<br>0: Disable<br>1: Enable<br>Note: When module disable, the clock of the calculation circuit will be gated |

| | | | |
|---|---|---|---|
| | | | automatically. |

**Note:** All bits in **BWS_CTRL_REG** is double-buffered.

#### 5.2.3.2 BWS_SIZE_REG

| **Offset: 0x004** | | | **Register Name:** BWS_SIZE_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:28 | / | / | / |
| 27:16 | R/W | 0x0 | **HEIGHT**<br>Input height<br>The actual height is register value + 1 |
| 15:12 | / | / | / |
| 11:0 | R/W | 0x0 | **WIDTH**<br>Input width<br>The actual width is register value + 1 |

#### 5.2.3.3 BWS_WIN0_REG

| **Offset: 0x008** | | | **Register Name:** BWS_WIN0_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:28 | / | / | / |
| 27:16 | R/W | 0 | **WIN_TOP**<br>Window Top position<br>Top position is the left-top y coordinate of display window in pixels |
| 15:12 | / | / | / |
| 11:0 | R/W | 0 | **WIN_LEFT**<br>Window Left position<br>Left position is left-top x coordinate of display window in pixels |

#### 5.2.3.4 BWS_WIN1_REG

| **Offset: 0x00C** | | | **Register Name:** BWS_WIN1_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:28 | / | / | / |
| 27:16 | R/W | 0 | **WIN_BOT**<br>Window Bottom position<br>Bottom position is right-bottom y coordinate of display window in pixels |
| 15:12 | / | / | / |
| 11:0 | R/W | 0 | **WIN_RIGHT**<br>Window Right position<br>Right position is right-bottom x coordinate of display window in pixels |

#### 5.2.3.5 BWS_THR0_REG

| **Offset: 0x020** | | | **Register Name:** BWS_THR0_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |

| | | | |
|---|---|---|---|
| 23:16 | R/W | 0x0 | **BLACK**<br>Black level stretch threshold 1 |
| 15:8 | / | / | / |
| 7:0 | R/W | 0x0 | **MIN**<br>Black level stretch threshold 0 |

#### 5.2.3.6 BWS_THR1_REG

| **Offset: 0x024** | | | **Register Name:** BWS_THR1_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23:16 | R/W | 0x0 | **MAX**<br>White level stretch threshold 1 |
| 15:8 | / | / | / |
| 7:0 | R/W | 0x0 | **WHITE**<br>White level stretch threshold 0 |

#### 5.2.3.7 BWS_SLP0_REG

| **Offset: 0x028** | | | **Register Name:** BWS_SLP0_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:26 | / | / | / |
| 25:16 | R/W | 0x0 | **SLOPE1**<br>Black level stretch slope 1 |
| 15:10 | / | / | / |
| 9:0 | R/W | 0x0 | **SLOPE0**<br>Black level stretch slope 0 |

#### 5.2.3.8 BWS_SLP1_REG

| **Offset: 0x02C** | | | **Register Name: BWS_SLP1_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:26 | / | / | / |
| 25:16 | R/W | 0x0 | **SLOPE3**<br>White level stretch slope 1 |
| 15:10 | / | / | / |
| 9:0 | R/W | 0x0 | **SLOPE2**<br>White level stretch slope 0 |

# 5.3 DE CSC Specification

## 5.3.1 Overview

The CSC features:

- Support 33-bit data format with 8-bit pre channel.
- Interface: Stream-to-stream pixel interface

## 5.3.2 Register List

| Module | Offset Address | Memory Range |
|---|---|---|
| CSC | 0x011F0000 | 8K |

| Register name | Offset | Description |
|---|---|---|
| CSC_BYPASS_REG | 0x000 | CSC bypass setting register |
| CSC_COEFF0_REG | 0x010, 0x014, 0x018 | CSC CH0 coefficients registers |
| CSC_CONST0_REG | 0x01C | CSC CH0 constant register |
| CSC_COEFF1_REG | 0x020, 0x024, 0x028 | CSC CH1 coefficients registers |
| CSC_CONST1_REG | 0x02C | CSC CH1 constant register |
| CSC_COEFF2_REG | 0x030, 0x034, 0x038 | CSC CH2 coefficients registers |
| CSC_CONST2_REG | 0x03C | CSC CH2 constant register |
| GLB_ALPHA_REG | 0x040 | Direct output alpha value register |

## 5.3.3 Register Description

### 5.3.3.1 CSC_BYPASS_REG

| **Offset:** 0x00 | | | **Register Name: CSC_BYPASS_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | **CSC_BYPASS**<br>CSC bypass<br>0: Bypass<br>1: Not bypass |

### 5.3.3.2 CSC_COEFF0_REG

| **Offset:**<br>C00 component: 0x10<br>C01 component: 0x14 | **Register Name: CSC_COEFF0_REG** |
|---|---|

| C02 component: 0x18 | | | |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x0 | **COEFF**<br>The value equals to coefficient*$2^{10}$ |

#### 5.3.3.3 CSC_CONST0_REG

| **Offset:** 0x1C | | | **Register Name: CSC_CONST0_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:20 | / | / | / |
| 19:0 | R/W | 0x0 | **CONST**<br>The value equals to coefficient*$2^{10}$ |

#### 5.3.3.4 CSC_COEFF1_REG

| **Offset:**<br>C10 component: 0x20<br>C11 component: 0x24<br>C12 component: 0x28 | | | **Register Name: CSC_COEFF1_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x0 | **COEFF**<br>The value equals to coefficient*$2^{10}$ |

#### 5.3.3.5 CSC_CONST1_REG

| **Offset:** 0x2C | | | **Register Name: CSC_CONST1_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:20 | / | / | / |
| 19:0 | R/W | 0x0 | **CONST**<br>The value equals to coefficient*$2^{10}$ |

#### 5.3.3.6 CSC_COEFF2_REG

| **Offset:**<br>C20 component: 0x30<br>C21 component: 0x34<br>C22 component: 0x38 | | | **Register Name: CSC_COEFF2_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x0 | **COEFF**<br>The value equals to coefficient*$2^{10}$ |

#### 5.3.3.7 CSC_CONST2_REG

| Offset: 0x3C | | | Register Name: CSC_CONST2_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:20 | / | / | / |
| 19:0 | R/W | 0x0 | **CONST**<br>The value equals to coefficient*$2^{10}$ |

#### 5.3.3.8 GLB_ALPHA_REG

| Offset: 0x40 | | | Register Name: GLB_ALPHA_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | R/W | 0xFF | **GLOBAL ALPHA**<br>The direct output global alpha value |
| 23:0 | / | / | / |

# 5.4 DE DRC Specification

## 5.4.1 Overview

The dynamic range controller (DRC) is a post-processing module which adjusts the image mapping curve according to the histogram frame by frame. The control function can be defined by the software driver according to the application. A typical application is Content-based backlight control.

**Feature:**

- Support data format with 8-bit per channel
- Support 2048*2048 input/output
- Support HISTOGRAM and DRC in HSV color space

## 5.4.2 Register List

| Module | Offset Address | Memory Range |
|---|---|---|
| DRC | 0x011b0000 | 64K |

| Register Name | Offset | Description |
|---|---|---|
| GNECTL_REG | 0x0000 | Module general control register |
| DRC_SIZE_REG | 0x0004 | DRC size setting register |
| DRC_CTL_REG | 0x0010 | DRC control register |
| DRC_SET_REG | 0x0018 | DRC setting register |
| DRC_WP0_REG | 0x001C | DRC window position register0 |
| DRC_WP1_REG | 0x0020 | DRC window position register1 |
| LH_CTL_REG | 0x0030 | Luminance histogram control register |
| LH_THR0_REG | 0x0034 | Luminance histogram threshold setting register0 |
| LH_THR1_REG | 0x0038 | Luminance histogram threshold setting register1 |
| LH_SLUMN_REG | 0x0040 + N*4<br>(N=0 ~ 7) | Luminance histogram statistics luminance recording register |
| LH_SCNTN_REG | 0x0060 + N*4<br>(N=0 ~ 7) | Luminance histogram statistics counter recording register |
| CSC_C00_REG | 0x00C0 | CSC coefficient 00 register |
| CSC_C01_REG | 0x00C4 | CSC coefficient 01 register |
| CSC_C02_REG | 0x00C8 | CSC coefficient 02 register |
| CSC_C03_REG | 0x00CC | CSC constant 03 register |
| CSC_C10_REG | 0x00D0 | CSC coefficient 10 register |
| CSC_C11_REG | 0x00D4 | CSC coefficient 11 register |
| CSC_C12_REG | 0x00D8 | CSC coefficient 12 register |
| CSC_C13_REG | 0x00DC | CSC constant 13 register |
| CSC_C20_REG | 0x00E0 | CSC coefficient 20 register |

| | | |
|---|---|---|
| CSC_C21_REG | 0x00E4 | CSC coefficient 21 register |
| CSC_C22_REG | 0x00E8 | CSC coefficient 22 register |
| CSC_C23_REG | 0x00EC | CSC constant 23 register |
| DRC_SPACOFF_REGN | 0x0F0 + N*4<br>(N=0,1,2) | DRC spatial coefficient registers |
| DRC_INTCOFF_REGN | 0x100 + N*4<br>(N=0 ~ 63) | DRC intensity coefficient registers |
| DRC_LGCOFF_REGN | 0x200 + N*4<br>(N=0 ~ 127) | DRC Luminance gain coefficient registers |

## 5.4.3 Register Description

### 5.4.3.1 General control register (Default Value: 0x0000_0000)

| Offset: 0x0000 | | | Register Name: GNECTL_REG |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31 | R/W | 0x0 | BIST_EN<br>BIST enable<br>0x0: disable<br>0x1: enable |
| 30:10 | / | / | / |
| 09:08 | R/W | 0x0 | MOD<br>Work mode selection. If bit 0 of the register is set ZERO, the following setting will be ignored.<br>0x0: reserved<br>0x1: reserved<br>0x2: DRC mode<br>0x3: reserved |
| 07:05 | / | / | / |
| 04 | R/W | 0x0 | COEF_SWITCH_EN<br>0x0: DONOT switch<br>0x1: Switch RAM use external RDY_EN<br><br>**Note: When LCD SYNC negative edge comes and COEF_SWITCH_EN is 1, DRC_LGCOFF_REGN will switch to the latest updated RAM if external RDY_EN equal to 1, and then the bit will also be self-cleared if switch action successes.** |
| 03:02 | / | / | / |
| 01 | R/W | 0x0 | Reserved<br>Note: Must set to 0. |
| 00 | R/W | 0x0 | DRC_EN<br>DRC module enable<br>0x0: disabled<br>0x1: enable |

| | | | |
|---|---|---|---|
| | | | **Note: When module disable, the input data will be bypassed to output, and the clock of calculation circuit will be gated automatically.** |

### 5.4.3.2 DRC size setting register (Default Value: 0x0000_0000)

| **Offset: 0x0004** | | | **Register Name: DRC_SIZE_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0x0 | DRC_HEIGHT<br>Display height<br>The real display height = The value of these bits + 1.<br>Note: Must set to 7~2047 when module enable. |
| 15:13 | / | / | / |
| 12:00 | R/W | 0x0 | DRC_WIDTH<br>Display width<br>The real display width = The value of these bits + 1. |

### 5.4.3.3 DRC control register (Default Value: 0x0000_0000)

| **Offset: 0x0010** | | | **Register Name: DRC_CTL_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:10 | / | / | / |
| 09 | R/W | 0x0 | HSV_MODE_EN<br>Enable Using component V calculated from three source components to process Luma Histogram and DRC.<br>0x0: disable<br>0x1: enable<br><br>Note: Must set to 1. |
| 08 | R/W | 0x0 | DRC_WIN_EN<br>Output window function enable<br>0x0: disable<br>0x1: enable |
| 07:01 | / | / | / |
| 00 | R/W | 0x0 | DRC_DB_EN<br>DRC double buffer function enable control<br>0x0: disable<br>0x1: enable |

### 5.4.3.4 DRC setting register (Default Value: 0x0000_0000)

| **Offset: 0x0018** | | | **Register Name: DRC_SET_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:16 | / | / | / |
| 15:08 | R/W | 0x0 | DRC_LGC_ABSLUMPERVAL<br>Abs luminance percent value |

| | | | |
|---|---|---|---|
| 07:02 | / | / | / |
| 01 | R/W | 0x0 | DRC_ADJUST_EN<br>DRC adjust enable<br>0x0: disable<br>0x1: enable |
| 00 | R/W | 0x0 | DRC_LGC_ABSLUMSHF<br>Abs luminance shift bits<br>0x0: shift 8bits<br>0x1: shift 9bits |

#### 5.4.3.5 DRC window position 0 register (Default Value: 0x0000_0000)

| Offset: 0x001C | | | Register Name: DRC_WP0_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:28 | / | / | / |
| 27:16 | R/W | 0x0 | DRC_WIN_TOP<br>Window Top position<br>Top position is the left-top y coordinate of display window in pixels |
| 15:12 | / | / | / |
| 11:00 | R/W | 0x0 | DRC_WIN_LEFT<br>Window Left position<br>Left position is left-top x coordinate of display window in pixels |

#### 5.4.3.6 DRC window position 1 register (Default Value: 0x0000_0000)

| Offset: 0x0020 | | | Register Name: DRC_WP1_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:28 | / | / | / |
| 27:16 | R/W | 0x0 | DRC_WIN_BOT<br>Window Bottom position<br>Bottom position is the right-bottom y coordinate of display window in pixels |
| 15:12 | / | / | / |
| 11:0 | R/W | 0x0 | DRC_WIN_RIGHT<br>Window Right position<br>Right position is the right-bottom x coordinate of display window in pixels |

#### 5.4.3.7 Luminance histogram control register (Default Value: 0x0000_0000)

| Offset: 0x0030 | | | Register Name: LH_CTL_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:02 | / | / | / |
| 1 | R/W | 0x0 | LH_MOD<br>0: Current frame case<br>1: Average case |
| 0 | R/W | 0x0 | LH_REC_CLR |

| | | | |
|---|---|---|---|
| | | | If the bit is set, the all of the luminance statistics recording registers will be clear, and the bit will self-clear when the recording registers is clear done. |

#### 5.4.3.8 Luminance histogram threshold setting 0 register (Default Value: 0x8060_4020)

| Offset: 0x0034 | | | Register Name: LH_THR0_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | R/W | 0x80 | LH_THRES_VAL4<br>Step4 threshold value |
| 23:16 | R/W | 0x60 | LH_THRES_VAL3<br>Step3 threshold value |
| 15:8 | R/W | 0x40 | LH_THRES_VAL2<br>Step2 threshold value |
| 7:0 | R/W | 0x20 | LH_THRES_VAL1<br>Step1 threshold value |

#### 5.4.3.9 Luminance histogram threshold setting 1 register (Default Value: 0x00E0_C0A0)

| Offset: 0x0038 | | | Register Name: LH_THR1_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23:16 | R/W | 0xE0 | LH_THRES_VAL7<br>Step7 threshold value |
| 15:08 | R/W | 0xC0 | LH_THRES_VAL6<br>Step6 threshold value |
| 07:00 | R/W | 0xA0 | LH_THRES_VAL5<br>Step5 threshold value |

Note: When setting LHT_REG0 and LHT_REG1, make sure that THRES_VAL1<THRES_VAL2<…<THRES_VAL7.

#### 5.4.3.10 Luminance histogram statistics lum recording register N (N = 0~7) (Default Value: 0x0000_0000)

| Offset: 0x0040 + N*4 | | | Register Name: LH_SLUM_REGN |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:00 | R/W | 0x0 | LH_LUM_DATA<br>Luminance statistics data |

#### 5.4.3.11 Luminance histogram statistics counter recording register N (N = 0~7) (Default Value: 0x0000_0000)

| Offset: 0x0060 + N*4 | | | Register Name: LH_SCNT_REGN |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23:00 | R/W | 0x0 | LH_CNT_DATA<br>Luminance statistics data |

### 5.4.3.12 CSC coefficient 00 register (Default Value: 0x0000_04A7)

| Offset: 0x00C0 | | | Register Name: CSC_C00_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x04A7 | C00<br>**Note: must set to 0x0400.** |

### 5.4.3.13 CSC coefficient 01 register (Default Value: 0x0000_1E6F)

| Offset: 0x00C4 | | | Register Name: CSC_C01_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x1E6F | C01<br>**Note: must set to 0x0000.** |

### 5.4.3.14 CSC coefficient 02 register (Default Value: 0x0000_1CBF)

| Offset: 0x00C8 | | | Register Name: CSC_C02_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x1CBF | C02<br>Note: must set to 0x0000. |

### 5.4.3.15 CSC constant 03 register (Default Value: 0x0000_0877)

| Offset: 0x00CC | | | Register Name: CSC_C03_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:14 | / | / | / |
| 13:0 | R/W | 0x0877 | C03<br>Note: must set to 0x0000. |

### 5.4.3.16 CSC coefficient 10 register (Default Value: 0x0000_04A7)

| Offset: 0x00D0 | | | Register Name: CSC_C10_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x04A7 | C10<br>Note: must set to 0x0000. |

### 5.4.3.17 CSC coefficient 11 register (Default Value: 0x0000_0000)

| Offset: 0x00D4 | | | Register Name: CSC_C11_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x0000 | C11 |

| | | | |
|---|---|---|---|
| | | | Note: must set to 0x0400. |

### 5.4.3.18 CSC coefficient 12 register (Default Value: 0x0000_0662)

| Offset: 0x00D8 | | | Register Name: CSC_C12_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x0662 | C12<br>Note: must set to 0x0000. |

### 5.4.3.19 CSC constant 13 register (Default Value: 0x0000_3211)

| Offset: 0x00DC | | | Register Name: CSC_C13_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:14 | / | / | / |
| 13:0 | R/W | 0x3211 | C13<br>Note: must set to 0x0000. |

### 5.4.3.20 CSC coefficient 20 register (Default Value: 0x0000_04A7)

| Offset: 0x00E0 | | | Register Name: CSC_C20_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x04A7 | C20<br>Note: must set to 0x0000. |

### 5.4.3.21 CSC coefficient 21 register (Default Value: 0x0000_0812)

| Offset: 0x00E4 | | | Register Name: CSC_C21_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x0812 | C21<br>Note: must set to 0x0000. |

### 5.4.3.22 CSC coefficient 22 register (Default Value: 0x0000_0000)

| Offset: 0x00E8 | | | Register Name: CSC_C22_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x0000 | C22<br>Note: must set to 0x0400. |

### 5.4.3.23 CSC constant 23 register (Default Value: 0x0000_2EB1)

| Offset: 0x00EC | Register Name: CSC_C23_REG |
|---|---|

| Bit | Read/Write | Default/Hex | Description |
|---|---|---|---|
| 31:14 | / | / | / |
| 13:0 | R/W | 0x2EB1 | C23<br>Note: must set to 0x0000. |

### 5.4.3.24 DRC spatial coefficient register N (N=0~2) (Default Value: 0x0000_0000)

| Offset: 0x00F0 + N*4 | | | Register Name: DRC_SPACOFF_REGN |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23:16 | R/W | 0x0 | SPA_COEF2<br>8 bits unsigned spatial coefficient data |
| 15:08 | R/W | 0x0 | SPA_COEF1<br>8 bits unsigned spatial coefficient data |
| 07:00 | R/W | 0x0 | SPA_COEF0<br>8 bits unsigned spatial coefficient data |

### 5.4.3.25 DRC intensity coefficient register N (N=0~63) (Default Value: 0x0000_0000)

| Offset: 0x0100 + N*4 | | | Register Name: DRC_INTCOFF_REGN |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | R/W | 0x0 | INT_COEF3<br>8 bits unsigned intensity coefficient data |
| 23:16 | R/W | 0x0 | INT_COEF2<br>8 bits unsigned intensity coefficient data |
| 15:08 | R/W | 0x0 | INT_COEF1<br>8 bits unsigned intensity coefficient data |
| 07:00 | R/W | 0x0 | INT_COEF0<br>8 bits unsigned intensity coefficient data |

### 5.4.3.26 DRC luminance gain coefficient register N (N=0~127)

| Offset: 0x0200 + N*4 | | | Register Name: DRC_LGCOFF_REGN |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:16 | R/W | UDF | LGC_COEF1<br>16bits luminance gain coefficient, unsigned data<br>The high 5 bits is the integer part<br>The low 11 bits is the decimal part |
| 15:00 | R/W | UDF | LGC_COEF0<br>16bits luminance gain coefficient, unsigned data<br>The high 5 bits is the integer part<br>The low 11 bits is the decimal part |

**Note: Double-buffered registers. When SYNC negative edge comes and COEF_SWITCH_EN is 1, coefficient RAM will switch to the latest updated RAM if external RDY_EN equal to 1.**

# 5.5 DE FCC Specification

## 5.5.1 Overview

Fancy color curvature change (FCC) is to adjust fancy colors so that a better vivid vision effect can be achieved. The FCC has the following features:

- Support RGB888 input and output format
- Support window clipping up to 8192x8192 pixels
- Support local adjustment for hue/ saturation in HSV space
- Support red/green/blue/cyan/magenta/yellow areas modifying in local adjustment mode

## 5.5.2 Register List

| Module name | Memory Range | Offset Address |
|---|---|---|
| FCC | 8K | 0xAA000 |

| Register name | Offset | Description |
|---|---|---|
| FCC_CTL_REG | 0x000 | FCC control register |
| FCC_INPUT_SIZE_REG | 0x004 | FCC input size register |
| FCC_OUTPUT_WIN0_REG | 0x008 | FCC output window setting register0 |
| FCC_OUTPUT_WIN1_REG | 0x00c | FCC output window setting register1 |
| FCC_HUE_RANGE_REG | 0x010 - 0x024 | FCC hue range register |
| FCC_LOCAL_GAIN_REG | 0x030 - 0x044 | FCC local gain register |
| INCSC_BYPASS_REG | 0x050 | |
| INCSC_COEFF0_REG | 0x060 - 0x068 | |
| INCSC_CONST0_REG | 0x06c | |
| INCSC_COEFF1_REG | 0x070 - 0x078 | |
| INCSC_CONST1_REG | 0x07c | |
| INCSC_COEFF2_REG | 0x080 - 0x088 | |
| INCSC_CONST2_REG | 0x08c | |
| GLB_ALPHA_REG | 0x90 | |

## 5.5.3 Register Description

### 5.5.3.1 FCC_CTRL_REG(Default Value: 0x0000_0000)

| Offset: 0x000 | | | Register Name: FCC_CTL_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:9 | / | / | / |
| 8 | R/W | 0x0 | **WIN_EN**<br>Output window function enable<br>0: disable |

| | | | |
|---|---|---|---|
| | | | 1: enable |
| 7:1 | / | / | / |
| 0 | R/W | 0x0 | **Enable**<br>Enable control<br>0:disable<br>1:enable<br>If the bit is disabled, the input data will by-pass to next module. |

#### 5.5.3.2 FCC_INPUT_SIZE_REG(Default Value: 0x0000_0000)

| Offset: 0x004 | | | Register Name: FCC_INPUT_SIZE_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0 | **HEIGHT**<br>Processing height<br>The real display height = The value of these bits + 1. |
| 15:13 | / | / | / |
| 12:00 | R/W | 0 | **WIDTH**<br>Processing width<br>The real display width = The value of these bits + 1. |

#### 5.5.3.3 FCC_OUTPUT_WIN0_REG(Default Value: 0x0000_0000)

| Offset: 0x008 | | | Register Name: FCC_OUTPUT_WIN0_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0 | **WIN_TOP**<br>Window Top position<br>Top position is the left-top y coordinate of display window in pixels |
| 15:13 | / | / | / |
| 12:00 | R/W | 0 | **WIN_LEFT**<br>Window Left position<br>Left position is left-top x coordinate of display window in pixels |

#### 5.5.3.4 FCC_OUTPUT_WIN1_REG(Default Value: 0x0000_0000)

| Offset: 0x00C | | | Register Name: FCC_OUTPUT_WIN1_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0 | **WIN_BOT**<br>Window Bottom position<br>Bottom position is right-bottom y coordinate of display window in pixels |
| 15:12 | / | / | / |
| 12:00 | R/W | 0 | **WIN_RIGHT**<br>Window Right position<br>Right position is right-bottom x coordinate of display window in pixels |

#### 5.5.3.5 FCC_HUE_RANGE_REG(Default Value: 0x0000_0000)

| Offset:<br>H/R component: 0x010<br>H/G component: 0x014<br>H/B component: 0x018<br>H/C component: 0x01c<br>H/M component: 0x020<br>H/Y component: 0x024 | | | Register Name: FCC_HUE_RANGE_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:28 | / | / | / |
| 27:16 | R/W | 0 | HMAX<br>the max value of hue |
| 15:12 | / | / | / |
| 11:0 | R/W | 0 | HMIN<br>the min value of hue |

#### 5.5.3.6 FCC_LOCAL_GAIN_REG

| Offset:<br>H/R component: 0x030<br>H/G component: 0x034<br>H/B component: 0x038<br>H/C component: 0x03c<br>H/M component: 0x040<br>H/Y component: 0x044 | | | Register Name: FCC_LOCAL_HGAIN_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:25 | / | / | / |
| 24:16 | R/W | 0 | HGAIN<br>the local gain of hue(Hgain),this is a sign number. |
| 15:9 | / | / | / |
| 8:0 | R/W | 0 | SGAIN<br>the local gain of Saturation (Sgain),this is a sign number. |

#### 5.5.3.7 INCSC_BYPASS_REG

| Offset: 0x050 | | | Register Name: INCSC_BYPASS_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | **CSC_BYPASS**<br>CSC bypass<br>0: Bypass<br>1: Not bypass |

#### 5.5.3.8 INCSC_COEFF0_REG(Default Value: 0x0000_0000)

| Offset:<br>C00 component: 0x060<br>C01 component: 0x064<br>C02 component: 0x068 | | | Register Name: INCSC_COEFF0_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x0 | **COEFF**<br>The value equals to coefficient*$2^{10}$ |

#### 5.5.3.9 INCSC_CONST0_REG(Default Value: 0x0000_0000)

| Offset:<br>C03 component: 0x06C | | | Register Name: INCSC_CONST0_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:20 | / | / | / |
| 19:0 | R/W | 0x0 | **CONST**<br>The value equals to coefficient*$2^{10}$ |

#### 5.5.3.10 INCSC_COEFF1_REG(Default Value: 0x0000_0000)

| Offset:<br>C10 component: 0x070<br>C11 component: 0x074<br>C12 component: 0x078 | | | Register Name: INCSC_COEFF1_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x0 | **COEFF**<br>The value equals to coefficient*$2^{10}$ |

#### 5.5.3.11 INCSC_CONST1_REG(Default Value: 0x0000_0000)

| Offset:<br>C13 component: 0x07C | | | Register Name: INCSC_CONST1_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:20 | / | / | / |
| 19:0 | R/W | 0x0 | **CONST**<br>The value equals to coefficient*$2^{10}$ |

#### 5.5.3.12 INCSC_COEFF2_REG(Default Value: 0x0000_0000)

| Offset:<br>C20 component: 0x080<br>C21 component: 0x084<br>C22 component: 0x088 | Register Name: INCSC_COEFF2_REG |
|---|---|

| Bit | Read/Write | Default/Hex | Description |
|---|---|---|---|
| 31:13 | / | / | / |
| 12:0 | R/W | 0x0 | **COEFF**<br>The value equals to coefficient*$2^{10}$ |

### 5.5.3.13 INCSC_CONST2_REG(Default Value: 0x0000_0000)

| Offset:<br>C23 component: 0x08C | | | Register Name: INCSC_CONST2_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:20 | / | / | / |
| 19:0 | R/W | 0x0 | **CONST**<br>The value equals to coefficient*$2^{10}$ |

### 5.5.3.14 GLB_ALPHA_REG

| Offset: 0x090 | | | Register Name: GLB_ALPHA_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | R/W | 0xFF | **GLOBAL ALPHA**<br>The global alpha value for video channel |
| 23:0 | / | / | / |

# 5.6 DE FCE Specification

## 5.6.1 Overview

The FCE(Fresh and Contrast Enhancement) features:

- Support data format with 8-bit pre channel.
- Support Maximum input frame size 4096×2160
- Interface: Stream-to-stream pixel interface

## 5.6.2 Register List

| Module name | Memory Range | Offset Address |
|---|---|---|
| **FCE** | 8K | 0xA0000 |

| Register name | Offset | Description |
|---|---|---|
| GCTRL_REG | 0x000 | Control register |
| FCE_SIZE_REG | 0x004 | Size setting register |
| FCE_WIN0_REG | 0x008 | Window setting register0 |
| FCE_WIN1_REG | 0x00C | Window setting register1 |
| LCE_GAIN_REG | 0x010 | LCE gain setting register |
| HIST_SUM_REG | 0x020 | Histogram sum register |
| HIST_STATUS_REG | 0x024 | Histogram status register |
| CE_STATUS_REG | 0x028 | CE LUT status register |
| FTC_GAIN_REG | 0x030 | FTC gain setting register |
| FCE_INCSC_BYPASS_REG | 0x040 | Input CSC bypass setting register |
| CE_LUT_REGN | 0x100+N*4 | CE LUT register N (N=0:63) |
| HIST_CNT_REGN | 0x200+N*4 | Histogram count register N (N=0:255) |

## 5.6.3 Register Description

### 5.6.3.1 GCTRL_REG

| Offset: 0x000 | | | Register Name: GCTRL_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31 | R/W | 0x0 | **WINDOW_EN**<br>LCE/CE/~~FTD~~/FTC window function enable<br>0:Disable<br>1:Enable<br>Note: When window function enable, only the area inside the window will be processed. **Histogram** function will be not effected by window function.<br>Note: double-buffered register. |

| | | | |
|---|---|---|---|
| 30:21 | / | / | / |
| 20 | R/W | 0x0 | **FTC_EN**<br>FTC function enable<br>0: Disable<br>1: Enable<br>Note: double-buffered register. |
| ~~19~~ | ~~R/W~~ | ~~0x0~~ | ~~**FTD_EN**~~<br>~~FTD function enable~~<br>~~0: Disable~~<br>~~1: Enable~~<br>~~Note: double-buffered register.~~ |
| 18 | R/W | 0x0 | **LCE_EN**<br>LCE function enable<br>0: Disable<br>1: Enable<br>Note: double-buffered register. |
| 17 | R/W | 0x0 | **CE_EN**<br>CE function enable<br>0: Disable<br>1: Enable<br>Note: double-buffered register. |
| 16 | R/W | 0x0 | **HIST_EN**<br>Histogram function enable<br>0: Disable<br>1: Enable<br>Note: double-buffered register. |
| 15:1 | / | / | / |
| 0 | R/W | 0x0 | **EN**<br>FCE module enable<br>0: Disable<br>1: Enable<br>Note: When module disable, the clock of the calculation circuit will be gated automatically. |

### 5.6.3.2 FCE_SIZE_REG

| **Offset: 0x004** | | | **Register Name: FCE_SIZE_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:28 | / | / | / |
| 27:16 | R/W | 0x0 | **HEIGHT**<br>Input height<br>The actual height is register value + 1 |
| 15:12 | / | / | / |
| 11:0 | R/W | 0x0 | **WIDTH**<br>Input width<br>The actual width is register value + 1 |

### 5.6.3.3 FCE_WIN0_REG

| Offset: 0x008 | | | Register Name: FCE_WIN0_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:28 | / | / | / |
| 27:16 | R/W | 0 | **WIN_TOP**<br>Window Top position<br>Top position is the left-top y coordinate of display window in pixels |
| 15:12 | / | / | / |
| 11:0 | R/W | 0 | **WIN_LEFT**<br>Window Left position<br>Left position is left-top x coordinate of display window in pixels |

### 5.6.3.4 FCE_WIN1_REG

| Offset: 0x00C | | | Register Name: FCE_WIN1_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:28 | / | / | / |
| 27:16 | R/W | 0 | **WIN_BOT**<br>Window Bottom position<br>Bottom position is right-bottom y coordinate of display window in pixels |
| 15:12 | / | / | / |
| 11:0 | R/W | 0 | **WIN_RIGHT**<br>Window Right position<br>Right position is right-bottom x coordinate of display window in pixels |

### 5.6.3.5 LCE_GAIN_REG

| Offset: 0x010 | | | Register Name: LCE_GAIN_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:16 | / | / | / |
| 15:8 | R/W | 0x0 | **LCE_BLEND**<br>Blending ratio of C2 and SOURCE<br>Note: U0.8 format. Range from 0 to (255/256). |
| 7:6 | / | / | / |
| 5:0 | R/W | 0x0 | **LCE_GAIN**<br>LCE gain<br>Note: U2.4 format. Range from 0 to (63/16). |

### 5.6.3.6 HIST_SUM_REG

| Offset: 0x020 | | | Register Name: HIST_SUM_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R | 0x0 | **SUM**<br>Pixel level sum |

| | | | |
|---|---|---|---|
| | | | **Note: Register will clear when reset module. And double-buffered register refresh when pixel counter equal to WIDTH × HEIGHT.** |

### 5.6.3.7 HIST_STATUS_REG

| **Offset: 0x024** | | | **Register Name: HIST_STATUS_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:8 | R | 0x0 | **HIST_NUM**<br>Histogram pixel number counter |
| 7:2 | / | / | / |
| 1 | R/W | 0x0 | **BIST_EN**<br>BIST enable<br>0: Disable<br>1: Enable |
| 0 | R | 0x0 | **HIST_CNT_VALID**<br>0: AHB access not valid<br>1: AHB access valid<br>Note: This bit switch to 0 when HIST_CNT_REG are updated. |

### 5.6.3.8 CE_STATUS_REG

| **Offset: 0x028** | | | **Register Name:** CE_STATUS_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | **CELUT_ACCESS_SWITCH**<br>0: Module access<br>1: AHB access<br>Note: When module access, CELUT registers can't access through AHB bus. When AHB access, CELUT will return the input address for data output. |

### 5.6.3.9 FTC_GAIN_REG

| **Offset: 0x030** | | | **Register Name:** FTC_GAIN_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23:16 | R/W | 0 | **FTC_H_GAIN_2**<br>FTC gain value2 |
| 15:8 | / | / | / |
| 7:0 | R/W | 0 | **FTC_H_GAIN_1**<br>FTC gain value1 |

### 5.6.3.10 FCE_INCSC_BYPASS_REG

| **Offset: 0x040** | | | **Register Name: FCE_INCSC_BYPASS_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:1 | / | / | / |

| | | | |
|---|---|---|---|
| 0 | R/W | 0x0 | **CSC_BYPASS**<br>CSC bypass<br>0: Bypass<br>1: Not bypass |

### 5.6.3.11 CE_LUT_REGN (N = 0:63)

| **Offset: 0x100 + N*4** | | | **Register Name: CE_LUT_REGN** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | R/W | UDF | CELUT3<br>(N*4+3) level LUT |
| 23:16 | R/W | UDF | CELUT2<br>(N*4+2) level LUT |
| 15:8 | R/W | UDF | CELUT1<br>(N*4+1) level LUT |
| 7:0 | R/W | UDF | CELUT0<br>(N*4) level LUT |

**Note: Registers can access in burst mode.**

### 5.6.3.12 HIST_CNT_REGN (N = 0:255)

| **Offset: 0x200 + N*4** | | | **Register Name: HIST_CNT_REGN** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:22 | / | / | **/** |
| 21:0 | R/W | UDF | **HIST**<br>Level N counter.<br>**Note:**<br>**1.Registers switch to AHB bus access mode when HIST_CNT_VALID = 1. When HIST_CNT_VALID = 0, accessing registers will return 0x00000000.**<br>**2.Registers will be clear to zero when LCD SYNC positive edge comes.**<br>**3.Registers can access in burst mode.** |

# 5.7 DE UIS Specification

## 5.7.1 Overview

UI Scaler(UIS) provides RGB format image resizing function for display engine. It receives data from overlay module, performs the image resizing function, and outputs to routing modules.

The UIS can receive ARGB8888 data format, and then converts to a required size ARGB8888 image for display. Horizontal and vertical direction scaling are implemented independently.

The US features:

- Support ARGB8888 data format with 8-bit pre channel.
- Support input and output size from 4×2 to 4096×4096
- Support 1⁄16× to 32× resize ratio.
- Support 16-phase 4-tap horizontal anti-alias filter, 16-phase linear filter in vertical.
- Point-to-point display size up to **W** pixels/line. (**W** is RTL programmable, default value 2048)

## 5.7.2 Register List

| Register name | Offset | Description |
|---|---|---|
| UIS_CTRL_REG | 0x000 | Control register |
| UIS_STATUS_REG | 0x008 | Status register |
| UIS FIELD_CTRL_REG | 0x00C | Field control register |
| UIS_BIST_REG | 0x010 | BIST control register |
| UIS_OUTSIZE_REG | 0x040 | Output size register |
| UIS_INSIZE_REG | 0x080 | Input size register |
| UIS _HSTEP_REG | 0x088 | Horizontal step register |
| UIS_VSTEP_REG | 0x08C | Vertical step register |
| UIS_HPHASE_REG | 0x090 | Horizontal initial phase register |
| UIS_VPHASE0_REG | 0x098 | Vertical initial phase 0 register |
| UIS_VPHASE1_REG | 0x09C | Vertical initial phase 1 register |
| UIS_HCOEF_REGN | 0x200+N*4 | Horizontal filter coefficient register N (N=0:15) |

**Note:** All registers except some bits in **UIS_CTRL_REG**, **UIS_FIELD_CTRL_REG**, **UIS_STATUS_REG** are double-buffered refreshed by **REG_RDY**.

## 5.7.3 Register Description

### 5.7.3.1 UIS_CTRL_REG

| **Offset: 0x000** | | | **Register Name:** UIS_CTRL_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31 | R/W | 0x0 | **BIST_EN** |

| | | | |
|---|---|---|---|
| | | | BIST enable<br>0: Disable<br>1: Enable |
| 30 | R/W | 0x0 | **CORE_RST**<br>Core circuit reset<br>0: Do noting<br>1: Reset core circuit |
| 29:5 | / | / | / |
| 4 | R/W | 0x0 | **COEF_SWITCH_RDY**<br>Coefficients RAM switch<br>0: DONOT switch<br>1: Switch RAM use **REG_RDY**<br>Note: When **LCD SYNC** go low and **COEF_SWITCH_RDY** is 1, coefficient RAM will switch to the latest updated RAM if **REG_RDY** is 1, and then the bit will also be self-cleared if switch action successes. |
| 3:1 | / | / | / |
| 0 | R/W | 0x0 | **EN**<br>UI Scaler enable<br>0: Disable<br>1: Enable<br>Note: When module disabled, the core clock to the core circuit will be gated, and the input data will be bypassed to down-stream module. |

**Note:** Only bit **EN** is double-buffered.

### 5.7.3.2 UIS_STATUS_REG

| **Offset: 0x008** | | | **Register Name: UIS_STATUS_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:28 | / | / | / |
| 27:16 | R | 0x0 | **LINE_CNT**<br>Output line number |
| 15:5 | / | / | / |
| 4 | R | 0x0 | **BUSY**<br>Core circuit status<br>0: idle (finish, module disable, waiting for **LCD SYNC** negative edge)<br>1: busy (core circuit calculating) |
| 3:0 | / | / | / |

**Note:** Whole WORD is non-double-buffered.

### 5.7.3.3 UIS_FIELD_CTRL_REG

| **Offset: 0x00C** | | | **Register Name: UIS_FIELD_CTRL_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:6 | / | / | / |
| 5 | R/W | 0x0 | **LCD_SYNC_REVERSE**<br>Reverse **LCD SYNC**<br>0: DONOT reverse<br>1: Reverse |

| | | | |
|---|---|---|---|
| 4 | R/W | 0x0 | **LCD_FILED_REVERSE**<br>Reverse **LCD FILED**<br>0: DONOT reverse<br>1: Reverse |
| 3:1 | / | / | / |
| 0 | R/W | 0x0 | **FIELD_SEL_VPHASE_EN**<br>Vertical initial phase switch control<br>0: Vertical initial phase fix to phase0<br>1: Switch Vertical initial phase by **LCD FIELD** (Switch to phase0 when **LCD FILED** is 1, and switch to phase1 when **LCD FIELD** is 0) |

**Note:** Only bit **FIELD_SEL_VPHASE_EN** is double-buffered.

#### 5.7.3.4 UIS_BIST_REG

| **Offset: 0x010** | | | **Register Name: UIS_BIST_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:21 | / | / | / |
| 20:16 | W | 0x0 | **BIST_RAM_SEL**<br>Select the index of RAM for BIST |
| 15:1 | / | / | / |
| 0 | W | 0x0 | **BIST_EN**<br>BIST enable<br>0: Disable<br>1: Enable |

**Note:** Whole WORD is non-double-buffered.

#### 5.7.3.5 UIS_OUTSIZE_REG

| **Offset: 0x040** | | | **Register Name: UIS_OUTSIZE_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0x0 | **OUT_HEIGHT**<br>Output height<br>The actual height is register value + 1 |
| 15:13 | / | / | / |
| 12:0 | R/W | 0x0 | **OUT_WIDTH**<br>Output width<br>The actual width is register value + 1 |

#### 5.7.3.6 UIS_INSIZE_REG

| **Offset: 0x080** | | | **Register Name: UIS_INSIZE_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0x0 | **IN_HEIGHT**<br>Input height<br>The actual height is register value + 1 |
| 15:13 | / | / | / |

| | | | |
|---|---|---|---|
| 12:0 | R/W | 0x0 | **IN _WIDTH**<br>Input width<br>The actual width is register value + 1 |

#### 5.7.3.7 UIS_HSTEP_REG

| Offset: 0x088 | | | Register Name: UIS_HSTEP_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:25 | / | / | / |
| 24:20 | R/W | 0x0 | **HSTEP_INT**<br>The integer part of horizontal scale ratio |
| 19:2 | R/W | 0x0 | **HSTEP_FRAC**<br>The fraction part of horizontal scale ratio |
| 1:0 | / | / | / |

#### 5.7.3.8 UIS_VSTEP_REG

| Offset: 0x08C | | | Register Name: UIS_VSTEP_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:25 | / | / | / |
| 24:20 | R/W | 0x0 | **VSTEP_INT**<br>The integer part of vertical scale ratio |
| 19:2 | R/W | 0x0 | **VSTEP_FRAC**<br>The fraction part of vertical scale ratio |
| 1:0 | / | / | / |

#### 5.7.3.9 UIS_HPHASE_REG

| Offset: 0x090 | | | Register Name: UIS_HPHASE_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23:20 | R/W | 0x0 | **HPHASE_INT**<br>The integer part of horizontal initial phase |
| 19:2 | R/W | 0x0 | **HPHASE_FRAC**<br>The fraction part of horizontal initial phase |
| 1:0 | / | / | / |

**Note:** *HPHASE* is a SIGNED parameter.

#### 5.7.3.10 UIS_VPHASE0_REG

| Offset: 0x098 | | | Register Name: UIS_VPHASE0_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23:20 | R/W | 0x0 | **VPHASE0_INT**<br>The integer part of vertical initial phase0 |
| 19:2 | R/W | 0x0 | **VPHASE0_FRAC**<br>The fraction part of vertical initial phase0 |

| | | | |
|---|---|---|---|
| 1:0 | / | / | / |

**Note:** *VPHASE0* is a SIGNED parameter.

### 5.7.3.11 UIS_VPHASE1_REG

| Offset: 0x09C | | | Register Name: UIS_VPHASE1_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23:20 | R/W | 0x0 | **VPHASE1_INT**<br>The integer part of vertical initial phase1 |
| 19:2 | R/W | 0x0 | **VPHASE1_FRAC**<br>The fraction part of vertical initial phase1 |
| 1:0 | / | / | / |

**Note:** *VPHASE1* is a SIGNED parameter.

### 5.7.3.12 UIS_HCOEF_REGN (N = 0:15)

| Offset: 0x200 + N*4 | | | Register Name: UIS_HCOEF_REGN |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | R/W | 0x0 | **COEF3**<br>The most right hand-side pixel coefficient |
| 23:16 | R/W | 0x0 | **COEF2**<br>The right hand-side pixel coefficient |
| 15:8 | R/W | 0x0 | **COEF1**<br>The left hand-side pixel coefficient |
| 7:0 | R/W | 0x0 | **COEF0**<br>The most left hand-side pixel coefficient |

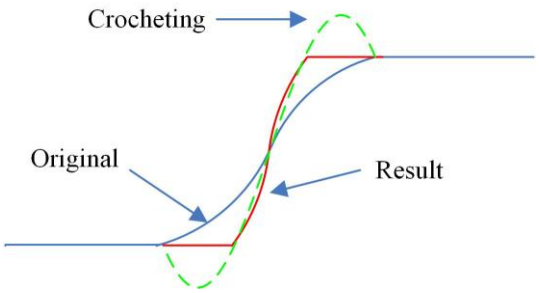
**Note:** HCOEFF is a two's complement. The register value equals to coefficient*$2^6$. **N** represents the phase.

# 5.8 DE LTI Specification

## 5.8.1 Overview

Luminance transient improvement (DLTI) is a method to enhance the image quality . The method of the luminance signal transition bands were detected, when the luminance signal transition, so that the signal transition zone is narrow, steep edge, so as to improve the clarity of the image, the image is more clear and bright.

This algorithm uses the one-dimensional stacking hook signal method commonly used, through a hook signal superimposed on the input signal, the signal edge becomes steeper, and then calculate the signal of a window within the scope of the maximum and minimum values, will limit the signal after superposition in the minimum and maximum range, to get the final results. Sketch of the algorithm is as follows:




## 5.8.2 Register List

| Module name | Memory Range | Offset Address |
|---|---|---|
| **LTI** | 8K | 0xA4000 |

| Register name | Offset | Description |
|---|---|---|
| LTI_EN | 0x000 | Global control register |
| LTI_SIZE | 0x00C | LTI size register |
| LTI_FIR_COFF0 | 0x010 | LTI FIR filter coefficient register0 |
| LTI_FIR_COFF1 | 0x014 | LTI FIR filter coefficient register1 |
| LTI_FIR_COFF2 | 0x018 | LTI FIR filter coefficient register2 |
| LTI_FIR_GAIN | 0x01C | LTI FIR filter gain register |
| LTI_COR_TH | 0x020 | LTI coring threshold register |
| LTI_DIFF_CTL | 0x024 | LTI scaling coefficient/offset of frist derivation |
| LTI_EDGE_GAIN | 0x028 | LTI edge gain |
| LTI_OS_CON | 0x02C | LTI coring/clipping threshold of Y shoot value |
| LTI_WIN_EXPANSION | 0x030 | LTI window range control register |
| LTI_EDGE_LEVEL_TH | 0x034 | LTI edge level threshold in edge-adaptive filtering |
| LTI_WIN0_REG | 0x038 | LTI window setting register0 |

| | | |
|---|---|---|
| LTI_WIN1_REG | 0x03c | LTI window setting register1 |
| LTI_INCSC_BYPASS_REG | 0x050 | Input CSC bypass setting register |
| LTI_INCSC_COEFF0_REG | 0x060, 0x064, 0x068 | Input CSC CH0 coefficients registers |
| LTI_INCSC_CONST0_REG | 0x06C | Input CSC CH0 constant register |
| LTI_INCSC_COEFF1_REG | 0x070, 0x074, 0x078 | Input CSC CH1 coefficients registers |
| LTI_INCSC_CONST1_REG | 0x07C | Input CSC CH1 constant register |
| LTI_INCSC_COEFF2_REG | 0x080, 0x084, 0x088 | Input CSC CH2 coefficients registers |
| LTI_INCSC_CONST2_REG | 0x08C | Input CSC CH2 constant register |
| LTI_OUTCSC_BYPASS_REG | 0x090 | Output CSC bypass setting register |
| LTI_OUTCSC_COEFF0_REG | 0x0A0, 0x0A4, 0x0A8 | Output CSC CH0 coefficients registers |
| LTI_OUTCSC_CONST0_REG | 0x0AC | Output CSC CH0 constant register |
| LTI_OUTCSC_COEFF1_REG | 0x0B0, 0x0B4, 0x0B8 | Output CSC CH1 coefficients registers |
| LTI_OUTCSC_CONST1_REG | 0x0BC | Output CSC CH1 constant register |
| LTI_OUTCSC_COEFF2_REG | 0x0C0, 0x0C4, 0x0C8 | Output CSC CH2 coefficients registers |
| LTI_OUTCSC_CONST2_REG | 0x0CC | Output CSC CH2 constant register |

## 5.8.3 Register Description

### 5.8.3.1 Global control register

| **Offset: 0x000** | | | **Register Name: LTI_CTL** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:25 | / | / | Reserved |
| 24 | R/W | 0x0 | **WIN_EN**<br>0: disable<br>1: enable |
| 23:17 | / | / | / |
| 16 | R/W | 0x0 | **NONLINEAR_LIMIT_EN**<br>0: disable<br>1: enable |
| 15:9 | / | / | / |
| 8 | R/W | 0x0 | **SEL**<br>0: select window range<br>1: select first differ |
| 7:1 | / | / | / |
| 0 | R/W | 0x0 | **LTI_EN**<br>0: LTI close<br>1: LTI open |

### 5.8.3.2 LTI size register

| **Offset: 0x00C** | | | **Register Name: LTI_SIZE** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:28 | / | / | Reserved |
| 27:16 | R/W | 0x0 | **HEIGHT** |

| | | | |
|---|---|---|---|
| | | | The real height = The value of these bits add 1 |
| 15:12 | / | / | / |
| 11:0 | R/W | 0x0 | **WIDTH**<br>The real width = The value of these bits add 1 |

### 5.8.3.3 LTI FIR filter coefficient register0

| **Offset: 0x010** | | | **Register Name: LTI_FIR_COFF0** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23:16 | R/W | 0x0 | **C1**<br>FIR filter coefficient C1<br>Note: in two's complement. |
| 15:8 | / | / | / |
| 7:0 | R/W | 0x0 | **C0**<br>FIR filter coefficient C0<br>Note: in two's complement. |

### 5.8.3.4 LTI FIR filter coefficient register1

| **Offset: 0x014** | | | **Register Name: LTI_FIR_COFF1** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23:16 | R/W | 0x0 | **C3**<br>FIR filter coefficient C3<br>Note: in two's complement. |
| 15:8 | / | / | / |
| 7:0 | R/W | 0x0 | **C2**<br>FIR filter coefficient C2<br>Note: in two's complement. |

### 5.8.3.5 LTI FIR filter coefficient register2

| **Offset: 0x018** | | | **Register Name: LTI_FIR_COFF2** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:8 | / | / | / |
| 7:0 | R/W | 0x0 | **C4**<br>FIR filter coefficient C4<br>Note: in two's complement. |

### 5.8.3.6 LTI FIR filter gain register

| **Offset: 0x01C** | | | **Register Name: LTI_FIR_GAIN** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:4 | / | / | / |
| 3:0 | R/W | 0x0 | **FIR FILTER GAIN** |

### 5.8.3.7 LTI coring threshold register

| Offset: 0x020 | | | Register Name: LTI_COR_TH |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31:10 | / | / | / |
| 9:0 | R/W | 0x0 | **LTI_COR_TH**<br>Coring threshold |

### 5.8.3.8 LTI differ filter control register

| Offset: 0x024 | | | Register Name: LTI_DIFF_CTL |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31:21 | / | / | / |
| 20:16 | R/W | 0x0 | **DIFF_SLOPE**<br>Differ gain slope control |
| 15:8 | / | / | / |
| 7:0 | R/W | 0x0 | **DIFF_OFFSET**<br>Differ gain offset control |

### 5.8.3.9 LTI adjustable gain parameter register

| Offset: 0x028 | | | Register Name: LTI_EDGE_GAIN |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31:5 | / | / | / |
| 4:0 | R/W | 0x0 | **LTI_EDGE_GAIN**<br>Edge adjustable gain parameter |

### 5.8.3.10 LTI The overshoot of control register

| Offset: 0x02C | | | Register Name:LTI_OS_CON |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31 | / | / | / |
| 30:28 | R/W | 0x0 | **PEAK_LIMIT**<br>Shifting bit for nonlinear limit |
| 27:24 | / | / | / |
| 23:16 | R/W | 0x0 | **CLIP_Y**<br>Clipping threshold of Y shoot value |
| 15:8 | / | / | / |
| 7:0 | R/W | 0x0 | **CORE_X**<br>Coring threshold of Y shoot value |

### 5.8.3.11 LTI window range expansion register

| Offset: 0x030 | | | Register Name: LTI_WIN_EXPANSION |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |

| | | | |
|---|---|---|---|
| 31:8 | / | / | / |
| 7:0 | R/W | 0x0 | **LTI_WIN_EXPANSION**<br>Window expansion size |

### 5.8.3.12 LTI edge strength threshold register

| Offset: 0x034 | | | Register Name: LTI_EDGE_ELVEL_TH |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:8 | / | / | / |
| 7:0 | R/W | 0x0 | **LTI_EDGE_ELVEL_TH**<br>Edge level threshold in edge-adaptive filtering |

### 5.8.3.13 LTI_WIN0_REG

| Offset: 0x038 | | | Register Name: LTI_EDGE_ELVEL_TH |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:8 | / | / | / |
| 27:16 | R/W | 0X0 | **WIN_TOP**<br>Window Top position |
| 15:12 | / | / | / |
| 11:0 | R/W | 0x0 | **LTI_LEFT**<br>Window Left position |

### 5.8.3.14 LTI_WIN1_REG

| Offset: 0x03c | | | Register Name: LTI_EDGE_ELVEL_TH |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:8 | / | / | / |
| 27:16 | R/W | 0X0 | **WIN_BOT**<br>Window bottom position |
| 15:12 | / | / | / |
| 11:0 | R/W | 0x0 | **LTI_WIGHT**<br>Window wright position |

### 5.8.3.15 LTI_INCSC_BYPASS_REG

| Offset: 0x50 | | | Register Name: LTI_INCSC_BYPASS_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | **CSC_BYPASS**<br>CSC bypass<br>0: Bypass<br>1: Not bypass |

### 5.8.3.16 LTI_INCSC_COEFF0_REG

| Offset:<br>C00 component: 0x60<br>C01 component: 0x64<br>C02 component: 0x68 | | | Register Name: LTI_INCSC_COEFF0_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x0 | **COEFF**<br>The value equals to coefficient*$2^{10}$ |

### 5.8.3.17 LTI_INCSC_CONST0_REG

| **Offset: 0x6C** | | | **Register Name: LTI_INCSC_CONST0_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:20 | / | / | / |
| 19:0 | R/W | 0x0 | **CONST**<br>The value equals to coefficient*$2^{10}$ |

### 5.8.3.18 LTI_INCSC_COEFF1_REG

| **Offset:**<br>**C10 component: 0x70**<br>**C11 component: 0x74**<br>**C12 component: 0x78** | | | **Register Name: LTI_INCSC_COEFF1_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x0 | **COEFF**<br>The value equals to coefficient*$2^{10}$ |

### 5.8.3.19 LTI_INCSC_CONST1_REG

| **Offset:** 0x7C | | | **Register Name: LTI_INCSC_CONST1_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:20 | / | / | / |
| 19:0 | R/W | 0x0 | **CONST**<br>The value equals to coefficient*$2^{10}$ |

### 5.8.3.20 LTI_INCSC_COEFF2_REG

| **Offset:**<br>**C20 component: 0x80**<br>**C21 component: 0x84**<br>**C22 component: 0x88** | | | **Register Name: LTI_INCSC_COEFF2_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |

| | | | |
|---|---|---|---|
| 12:0 | R/W | 0x0 | **COEFF**<br>The value equals to coefficient*$2^{10}$ |

### 5.8.3.21 LTI_INCSC_CONST2_REG

| **Offset:** 0x8C | | | **Register Name: LTI_INCSC_CONST2_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:20 | / | / | / |
| 19:0 | R/W | 0x0 | **CONST**<br>The value equals to coefficient*$2^{10}$ |

### 5.8.3.22 LTI_OUTCSC_BYPASS_REG

| **Offset:** 0x90 | | | **Register Name: LTI_OUTCSC_BYPASS_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | **CSC_BYPASS**<br>CSC bypass<br>0: Bypass<br>1: Not bypass |

### 5.8.3.23 LTI_OUTCSC_COEFF0_REG

| **Offset:**<br>C00 component: 0xA0<br>C01 component: 0xA4<br>C02 component: 0xA8 | | | **Register Name: LTI_OUTCSC_COEFF0_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x0 | **COEFF**<br>The value equals to coefficient*$2^{10}$ |

### 5.8.3.24 LTI_OUTCSC_CONST0_REG

| **Offset: 0xAC** | | | **Register Name: LTI_OUTCSC_CONST0_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:20 | / | / | / |
| 19:0 | R/W | 0x0 | **CONST**<br>The value equals to coefficient*$2^{10}$ |

### 5.8.3.25 LTI_OUTCSC_COEFF1_REG

| **Offset:**<br>**C10 component: 0xB0**<br>**C11 component: 0xB4**<br>**C12 component: 0xB8** | | | **Register Name: LTI_OUTCSC_COEFF1_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |

| | | | |
|---|---|---|---|
| 31:13 | / | / | / |
| 12:0 | R/W | 0x0 | **COEFF**<br>The value equals to coefficient*$2^{10}$ |

### 5.8.3.26 LTI_OUTCSC_CONST1_REG

| **Offset:** 0xBC | | | **Register Name: LTI_OUTCSC_CONST1_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:20 | / | / | / |
| 19:0 | R/W | 0x0 | **CONST**<br>The value equals to coefficient*$2^{10}$ |

### 5.8.3.27 LTI_OUTCSC_COEFF2_REG

| **Offset:**<br>**C20 component: 0xC0**<br>**C21 component: 0xC4**<br>**C22 component: 0xC8** | | | **Register Name: LTI_OUTCSC_COEFF2_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x0 | **COEFF**<br>The value equals to coefficient*$2^{10}$ |

### 5.8.3.28 LTI_OUTCSC_CONST2_REG

| **Offset:** 0xCC | | | **Register Name: LTI_OUTCSC_CONST2_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:20 | / | / | / |
| 19:0 | R/W | 0x0 | **CONST**<br>The value equals to coefficient*$2^{10}$ |

# 5.9 DE Peaking Specification

## 5.9.1 Overview

The Luma Peaking module enhances mid-high frequence of luma channel in images. The chroma channels will bypass.

The Luma Peaking module including following feature:

- Maximum input frame size: 4096×2160.
- Data width: 8bit /10bit(hardware programmable) per channel.
- Interface: Stream-to-stream pixel interface.

## 5.9.2 Block Diagram

Figure 5-1 shows the block diagram of luma peaking module. It is a stream-to-stream module with uniform input and output interface. It contains some parts and their function list followed:

- Control logic: Status machine control, registers operation.
- Peaking calculation unit: luma peaking, chroma bypassing.
- MUX: select processed data or input data (input data pass-through works without clock)

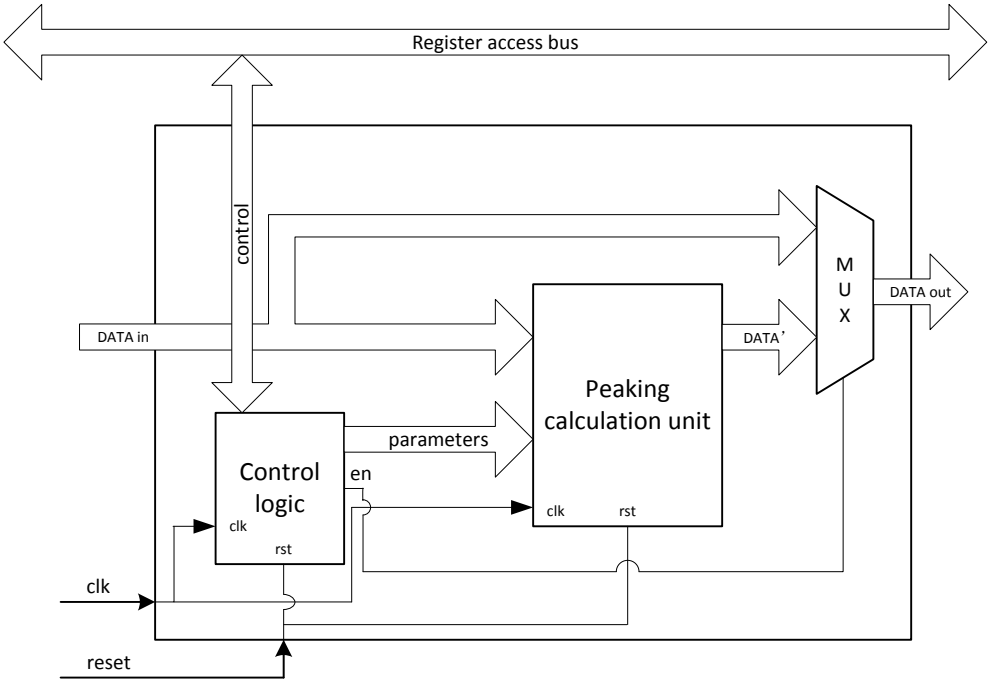



**Figure 5-1. Luma peaking module block diagram**

### 5.9.3 Register list

| Module name | Memory Range | Offset Address |
|---|---|---|
| **PEAKING** | 8K | 0xA6000 |

| Register Name | Offset | Description |
|---|---|---|
| LP_CTRL_REG | 0x000 | LP module control register |
| LP_SIZE_REG | 0x004 | LP size setting register |
| LP_WIN0_REG | 0x008 | LP window setting register0 |
| LP_WIN1_REG | 0x00C | LP window setting register1 |
| LP_FILTER_REG | 0x010 | LP filter setting register |
| LP_CSTM_FILTER0_REG | 0x014 | LP custom filter setting register0 |
| LP_CSTM_FILTER1_REG | 0x018 | LP custom filter setting register1 |
| LP_CSTM_FILTER2_REG | 0x01C | LP custom filter setting register2 |
| LP_GAIN_REG | 0x020 | LP gain setting register |
| LP_GAINCTRL_REG | 0x024 | LP gain control setting register |
| LP_SHOOTCTRL_REG | 0x028 | LP shoot control setting register |
| LP_CORING_REG | 0x02C | LP coring setting register |
| LP_INCSC_BYPASS_REG | 0x050 | Input CSC bypass setting register |
| LP_INCSC_COEFF0_REG | 0x060, 0x064, 0x068 | Input CSC CH0 coefficients registers |
| LP_INCSC_CONST0_REG | 0x06C | Input CSC CH0 constant register |
| LP_INCSC_COEFF1_REG | 0x070, 0x074, 0x078 | Input CSC CH1 coefficients registers |
| LP_INCSC_CONST1_REG | 0x07C | Input CSC CH1 constant register |
| LP_INCSC_COEFF2_REG | 0x080, 0x084, 0x088 | Input CSC CH2 coefficients registers |
| LP_INCSC_CONST2_REG | 0x08C | Input CSC CH2 constant register |
| LP_OUTCSC_BYPASS_REG | 0x090 | Output CSC bypass setting register |
| LP_OUTCSC_COEFF0_REG | 0x0A0, 0x0A4, 0x0A8 | Output CSC CH0 coefficients registers |
| LP_OUTCSC_CONST0_REG | 0x0AC | Output CSC CH0 constant register |
| LP_OUTCSC_COEFF1_REG | 0x0B0, 0x0B4, 0x0B8 | Output CSC CH1 coefficients registers |
| LP_OUTCSC_CONST1_REG | 0x0BC | Output CSC CH1 constant register |
| LP_OUTCSC_COEFF2_REG | 0x0C0, 0x0C4, 0x0C8 | Output CSC CH2 coefficients registers |
| LP_OUTCSC_CONST2_REG | 0x0CC | Output CSC CH2 constant register |

### 5.9.4 Register Description

#### 5.9.4.1 LP_CTRL_REG

| **Offset**: 0x00 | | | **Register name: LP_CTRL_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:9 | / | / | / |
| 8 | R/W | 0x0 | **WIN_EN**<br>Output window function enable<br>0: disable |

| | | | |
|---|---|---|---|
| | | | 1: enable |
| 7:1 | / | / | / |
| 0 | R/W | 0x0 | **EN**<br>LP Module enable<br>0: Disable<br>1: Enable |

#### 5.9.4.2 LP_SIZE_REG

| **Offset**: 0x04 | | | **Register name: LP_SIZE_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:28 | / | / | / |
| 27:16 | R/W | 0 | **HEIGHT**<br>Processing height<br>The real display height = The value of these bits + 1. |
| 15:12 | / | / | / |
| 11:00 | R/W | 0 | **WIDTH**<br>Processing width<br>The real display width = The value of these bits + 1. |

#### 5.9.4.3 LP_WIN0_REG

| **Offset**: 0x08 | | | **Register name: LP_WIN0_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:28 | / | / | / |
| 27:16 | R/W | 0 | **WIN_TOP**<br>Window Top position<br>Top position is the left-top y coordinate of display window in pixels |
| 15:12 | / | / | / |
| 11:00 | R/W | 0 | **WIN_LEFT**<br>Window Left position<br>Left position is left-top x coordinate of display window in pixels |

#### 5.9.4.4 LP_WIN1_REG

| **Offset**: 0x0C | | | **Register name: LP_WIN1_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:28 | / | / | / |
| 27:16 | R/W | 0 | **WIN_BOT**<br>Window Bottom position<br>Bottom position is right-bottom y coordinate of display window in pixels |
| 15:12 | / | / | / |
| 11:00 | R/W | 0 | **WIN_RIGHT**<br>Window Right position<br>Right position is right-bottom x coordinate of display window in pixels |

### 5.9.4.5 LP_FILTER_REG

| Offset: 0x10 | | | Register Name: LP_FILTER_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31 | R/W | 0x0 | **FILTER_SEL**<br>Filter selection:<br>0: Default filter.<br>1: Custom filter. |
| 30:22 | / | / | / |
| 21:16 | R/W | 0x0 | **HP_RATIO**<br>Default high-pass filter ratio<br>Note: in two's complement. |
| 15:14 | / | / | / |
| 13:8 | R/W | 0x0 | **BP0_RATIO**<br>Default band-pass filter0 ratio<br>Note: in two's complement. |
| 7:6 | / | / | / |
| 5:0 | R/W | 0x0 | **BP1_RATIO**<br>Default band-pass filter1 ratio<br>Note: in two's complement. |

### 5.9.4.6 LP_CSTM_FILTER0_REG

| Offset: 0x14 | | | Register Name: LP_CSTM_FILTER0_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:25 | / | / | / |
| 24:16 | R/W | 0x0 | **C1**<br>Custom filter coefficient C1.<br>Note: in two's complement. |
| 15:9 | / | / | / |
| 8:0 | R/W | 0x0 | **C0**<br>Custom filter coefficient C0.<br>Note: in two's complement. |

### 5.9.4.7 LP_CSTM_FILTER1_REG

| Offset: 0x18 | | | Register Name: LP_CSTM_FILTER1_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:25 | / | / | / |
| 24:16 | R/W | 0x0 | **C3**<br>Custom filter coefficient C3.<br>Note: in two's complement. |
| 15:9 | / | / | / |
| 8:0 | R/W | 0x0 | **C2** |

| | | | |
|---|---|---|---|
| | | | Custom filter coefficient C2.<br>Note: in two's complement. |

### 5.9.4.8 LP_CSTM_FILTER2_REG

| Offset: 0x1C | | | Register Name: LP_CSTM_FILTER2_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:9 | / | / | / |
| 8:0 | R/W | 0x0 | **C4**<br>Custom filter coefficient C4.<br>Note: in two's complement. |

### 5.9.4.9 LP_GAIN_REG

| Offset: 0x20 | | | Register Name: LP_CSTM_FILTER0_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:8 | / | / | / |
| 7:0 | R/W | 0x0 | **GAIN**<br>Peaking gain setting. |

### 5.9.4.10 LP_GAINCTRL_REG

| Offset: 0x24 | | | Register Name: LP_GAINCTRL_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23:16 | R/W | 0x0 | **DIF_UP**<br>Gain control: limitation threshold.<br>Note: use low 8bits for 8bit data width. |
| 15:8 | / | / | / |
| 4:0 | R/W | 0x0 | **BETA**<br>Gain control: large gain limitation.<br>Note: in two's complement. |

### 5.9.4.11 LP_SHOOTCTRL_REG

| Offset: 0x28 | | | Register Name: LP_SHOOTCTRL_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:6 | / | / | / |
| 5:0 | R/W | 0x0 | **NEG_GAIN**<br>Undershoot gain control. |

### 5.9.4.12 LP_CORING_REG

| Offset: 0x2C | | | Register Name: LP_CORING_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |

| | | | |
|---|---|---|---|
| 31:8 | / | / | / |
| 7:0 | R/W | 0x0 | **CORTHR**<br>Coring threshold.<br>Note: use low 8bits for 8bit data width. |

### 5.9.4.13 LP_INCSC_BYPASS_REG

| **Offset:** 0x50 | | | **Register Name: LP_INCSC_BYPASS_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | **CSC_BYPASS**<br>CSC bypass<br>0: Bypass<br>1: Not bypass |

### 5.9.4.14 LP_INCSC_COEFF0_REG

| **Offset:**<br>C00 component: 0x60<br>C01 component: 0x64<br>C02 component: 0x68 | | | **Register Name: LP_INCSC_COEFF0_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x0 | **COEFF**<br>The value equals to coefficient*$2^{10}$ |

### 5.9.4.15 LP_INCSC_CONST0_REG

| **Offset:** 0x6C | | | **Register Name: LP_INCSC_CONST0_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:20 | / | / | / |
| 19:0 | R/W | 0x0 | **CONST**<br>The value equals to coefficient*$2^{10}$ |

### 5.9.4.16 LP_INCSC_COEFF1_REG

| **Offset:**<br>C10 component: 0x70<br>C11 component: 0x74<br>C12 component: 0x78 | | | **Register Name: LP_INCSC_COEFF1_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x0 | **COEFF**<br>The value equals to coefficient*$2^{10}$ |

### 5.9.4.17 LP_INCSC_CONST1_REG

| Offset: 0x7C | | | Register Name: LP_INCSC_CONST1_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:20 | / | / | / |
| 19:0 | R/W | 0x0 | **CONST**<br>The value equals to coefficient*$2^{10}$ |

### 5.9.4.18 LP_INCSC_COEFF2_REG

| Offset:<br>C20 component: 0x80<br>C21 component: 0x84<br>C22 component: 0x88 | | | Register Name: LP_INCSC_COEFF2_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x0 | **COEFF**<br>The value equals to coefficient*$2^{10}$ |

### 5.9.4.19 LP_INCSC_CONST2_REG

| Offset: 0x8C | | | Register Name: LP_INCSC_CONST2_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:20 | / | / | / |
| 19:0 | R/W | 0x0 | **CONST**<br>The value equals to coefficient*$2^{10}$ |

### 5.9.4.20 LP_OUTCSC_BYPASS_REG

| Offset: 0x90 | | | Register Name: LP_OUTCSC_BYPASS_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | **CSC_BYPASS**<br>CSC bypass<br>0: Bypass<br>1: Not bypass |

### 5.9.4.21 LP_OUTCSC_COEFF0_REG

| Offset:<br>C00 component: 0xA0<br>C01 component: 0xA4<br>C02 component: 0xA8 | | | Register Name: LP_OUTCSC_COEFF0_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x0 | **COEFF**<br>The value equals to coefficient*$2^{10}$ |

### 5.9.4.22 LP_OUTCSC_CONST0_REG

| Offset: 0xAC | | | Register Name: LP_OUTCSC_CONST0_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:20 | / | / | / |
| 19:0 | R/W | 0x0 | **CONST**<br>The value equals to coefficient*$2^{10}$ |

### 5.9.4.23 LP_OUTCSC_COEFF1_REG

| Offset:<br>C10 component: 0xB0<br>C11 component: 0xB4<br>C12 component: 0xB8 | | | Register Name: LP_OUTCSC_COEFF1_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x0 | **COEFF**<br>The value equals to coefficient*$2^{10}$ |

### 5.9.4.24 LP_OUTCSC_CONST1_REG

| Offset: 0xBC | | | Register Name: LP_OUTCSC_CONST1_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:20 | / | / | / |
| 19:0 | R/W | 0x0 | **CONST**<br>The value equals to coefficient*$2^{10}$ |

### 5.9.4.25 LP_OUTCSC_COEFF2_REG

| Offset:<br>C20 component: 0xC0<br>C21 component: 0xC4<br>C22 component: 0xC8 | | | Register Name: LP_OUTCSC_COEFF2_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R/W | 0x0 | **COEFF**<br>The value equals to coefficient*$2^{10}$ |

### 5.9.4.26 LP_OUTCSC_CONST2_REG

| Offset: 0xCC | | | Register Name: LP_OUTCSC_CONST2_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:20 | / | / | / |
| 19:0 | R/W | 0x0 | **CONST**<br>The value equals to coefficient*$2^{10}$ |

# 5.10 DE RT-MIXER Specification

## 5.10.1 Overview

The RT-mixer Core consist of dma, overlay, scaler and blender block. It supports 4 layers overlay in one pipe, and its result can scaler up or down to blender in the next processing. There's the feature description as follows:

- Support layer size up to 2048x204 pixels
- Support pre-multiply alpha image data
- Support four layers overlay in every pipe and four pipes alpha blending
- Support color key
- Support *Porter-Duff* alpha blending
- Support input format *YUV422/YUV420/YUV411/ARGB8888/XRGB8888/RGB888/ARGB4444/ ARGB1555 and RGB565*
- Support 3D format image data

## 5.10.2 Block Diagram

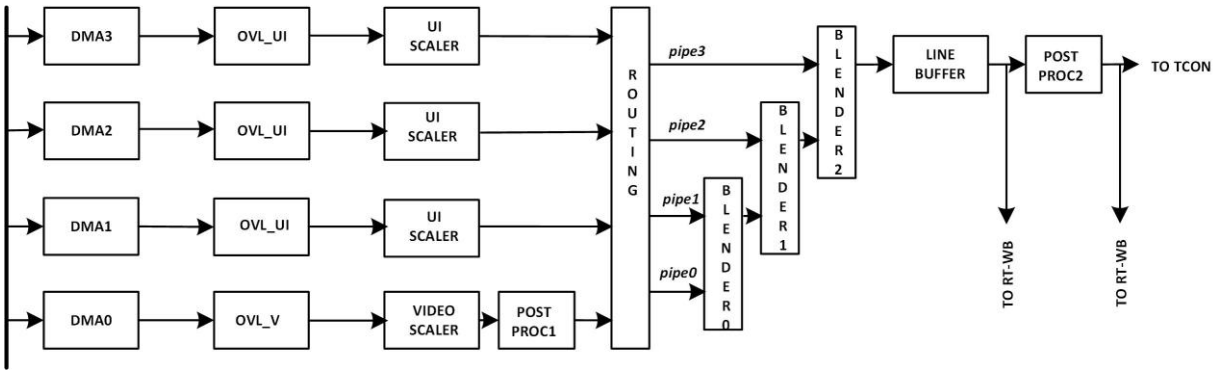



**Figure 5-2. RT-Mixer General Diagram**

## 5.10.3 DE RT-Mixer Description

### 5.10.3.1 Input Data Memory Layout

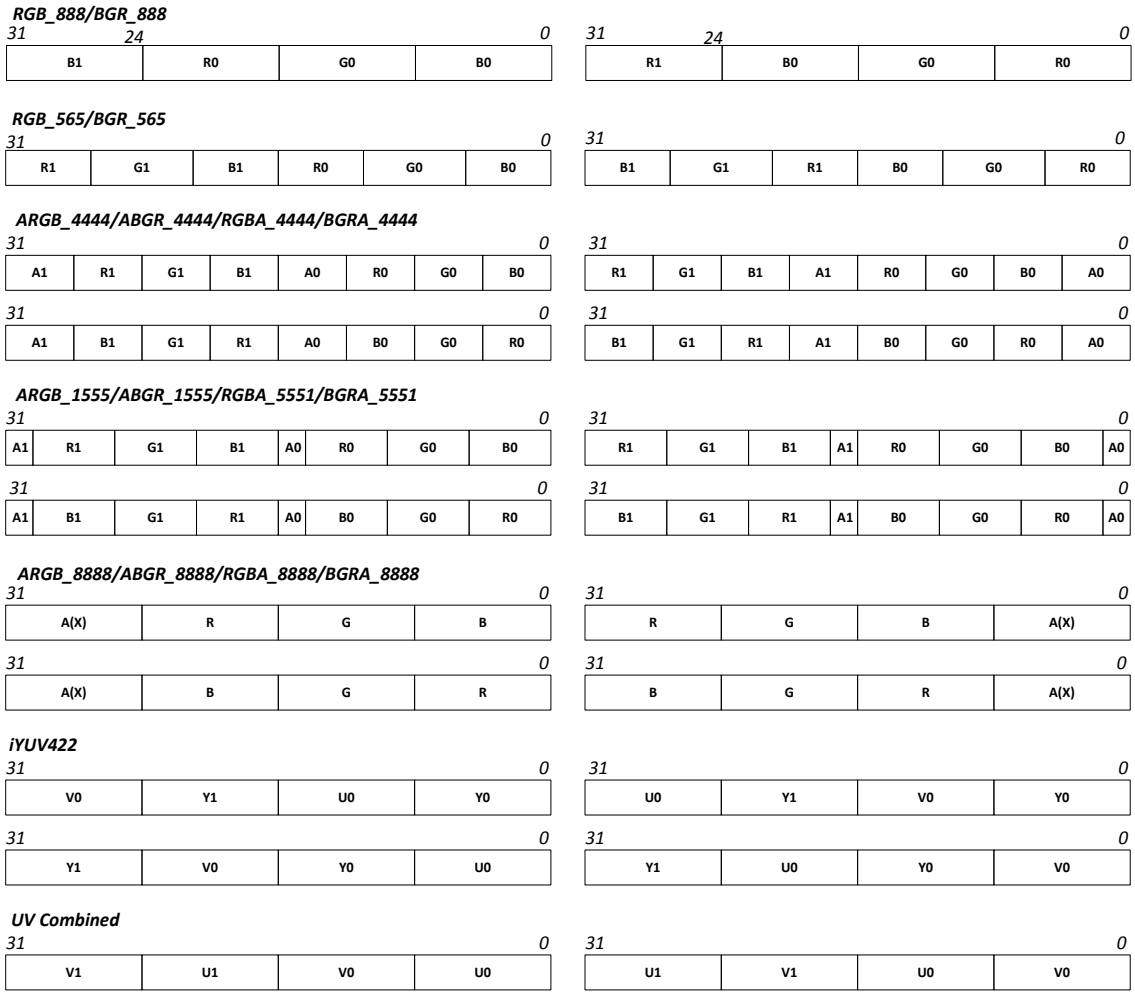



**Figure 5-3. Input data pixel sequence**

### 5.10.3.2 Overlay

Figure 5-4 is the overlay processing include layer memory data access and overlay relationship, the detail as following.

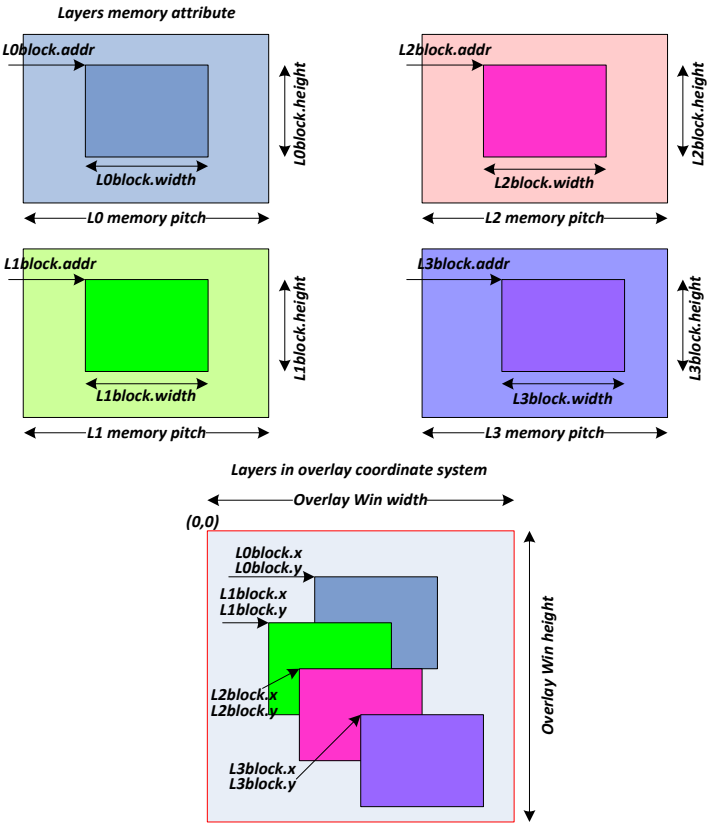



**Figure 5-4. Layers memory access and overlay processing**

**Note: the layer priority is layer3>layer2>layer1>layer0**

### 5.10.3.3 Scaling

**UI scaler :** Please refer to << Display_Engine2.0_UI_Scaler_SPEC>>

**Video scaler:** Please refer to << Display_Engine2.0_Video_Scaler_SPEC>>

**Routing:** The routing connects N channels and N input of blending together. Any channel can connect to any pipe. Programmer should make sure every pipe should connect one different channel only.

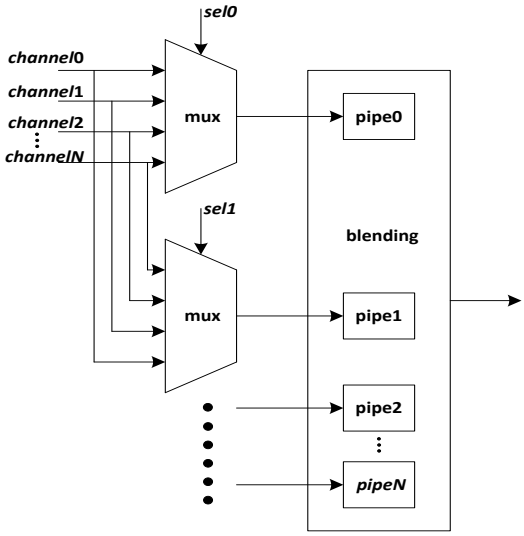



**Figure 5-5. N Channel select pipe route processing**

### 5.10.3.4 Blender

**Blender input data storing:**

The following diagram is about overlay data remapping in the blender pipe.

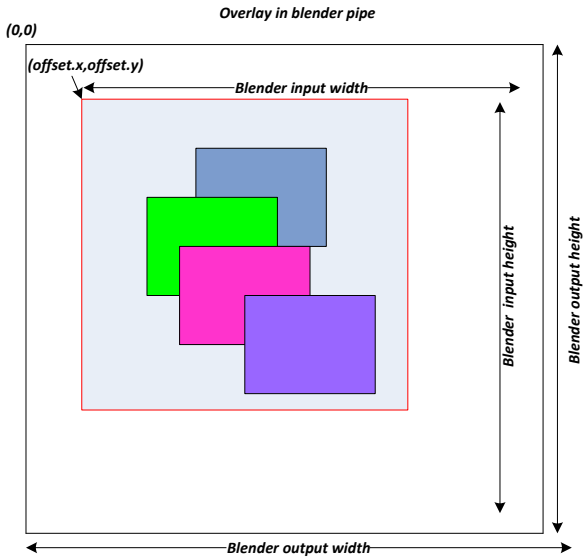



**Figure 5-6. Pipe memory access diagram**

## 5.10.4 Memory Mapping List

| Module name | Memory Range | Offset Address |
|---|---|---|
| GLB | 4K | 0x00000 |
| BLD | 4K | 0x01000 |
| OVL_V(CH0) | 4K | 0x02000 |
| OVL_UI(CH1) | 4K | 0x03000 |
| OVL_UI(CH2) | 4K | 0x04000 |
| OVL_UI(CH3) | 4K | 0x05000 |
| VIDEO_SCALER(CH0) | 128K | 0x20000 |
| UI_SCALER1(CH1) | 64K | 0x40000 |
| UI_SCALER2(CH2) | 64K | 0x50000 |
| UI_SCALER3(CH3) | 64K | 0x60000 |
| POST_PROC1 | 64K | 0xA0000 |
| POST_PROC2 | 64K | 0xB0000 |
| DMA | 128K | 0xC0000 |

## 5.10.5 Register List

| Register name | Offset | Description |
|---|---|---|
| GLB_CTL | 0x000 | Global control register |
| GLB_STS | 0x004 | Global status register |
| GLB_DBUFFER | 0x008 | Global double buffer control register |
| GLB_SIZE | 0x00C | Global size register |
| | | |

| | | |
|---|---|---|
| OVL_V_ATTCTL | 0x000 + N*0x30 | OVL_V attribute control register(N=0,1,2,3) |
| OVL_V_MBSIZE | 0x004 + N*0x30 | OVL_V memory block size register(N=0,1,2,3) |
| OVL_V_COOR | 0x008 + N*0x30 | OVL_V memory block coordinate register(N=0,1,2,3) |
| OVL_V_PITCH0 | 0x00C + N*0x30 | OVL_V memory pitch register0(N=0,1,2,3) |
| OVL_V_PITCH1 | 0x010 + N*0x30 | OVL_V memory pitch register1(N=0,1,2,3) |
| OVL_V_PITCH2 | 0x014 + N*0x30 | OVL_V memory pitch register2(N=0,1,2,3) |
| OVL_V_TOP_LADD0 | 0x018 + N*0x30 | OVL_V top field memory block low address register0(N=0,1,2,3) |
| OVL_V_TOP_LADD1 | 0x01C + N*0x30 | OVL_V top field memory block low address register1(N=0,1,2,3) |
| OVL_V_TOP_LADD2 | 0x020 + N*0x30 | OVL_V top field memory block low address register2(N=0,1,2,3) |
| OVL_V_BOT_LADD0 | 0x024 + N*0x30 | OVL_V bottom field memory block low address register0(N=0,1,2,3) |
| OVL_V_BOT_LADD1 | 0x028 + N*0x30 | OVL_V bottom field memory block low address register1(N=0,1,2,3) |
| OVL_V_BOT_LADD2 | 0x02C + N*0x30 | OVL_V bottom field memory block low address register2(N=0,1,2,3) |
| OVL_V_FILL_COLOR | 0x0C0 + N*0x4 | OVL_V fill color register(N=0,1,2,3) |
| OVL_V_TOP_HADD0 | 0x0D0 | OVL_V top field memory block high address register0 |
| OVL_V_TOP_HADD1 | 0x0D4 | OVL_V top field memory block high address register1 |
| OVL_V_TOP_HADD2 | 0x0D8 | OVL_V top field memory block high address register2 |
| OVL_V_BOT_HADD0 | 0x0DC | OVL_V bottom field memory block high address register0 |
| OVL_V_BOT_HADD1 | 0x0E0 | OVL_V bottom field memory block high address register1 |
| OVL_V_BOT_HADD2 | 0x0E4 | OVL_V bottom field memory block high address register2 |
| OVL_V_SIZE | 0x0E8 | OVL_V overlay window size register |
| OVL_V_HDS_CTL0 | 0x0F0 | OVL_V horizontal down sample control register0 |
| OVL_V_HDS_CTL1 | 0x0F4 | OVL_V horizontal down sample control register1 |
| OVL_V_VDS_CTL0 | 0x0F8 | OVL_V vertical down sample control register0 |
| OVL_V_VDS_CTL1 | 0x0FC | OVL_V vertical down sample control register1 |
| | | |
| OVL_UI_ATTCTL | 0x000 + N*0x20 | OVL_UI attribute control register(N=0,1,2,3) |
| OVL_UI_MBSIZE | 0x004 + N*0x20 | OVL_UI memory block size register(N=0,1,2,3) |
| OVL_UI_COOR | 0x008 + N*0x20 | OVL_UI memory block coordinate register(N=0,1,2,3) |
| OVL_UI_PITCH | 0x00C + N*0x20 | OVL_UI memory pitch register(N=0,1,2,3) |
| OVL_UI_TOP_LADD | 0x010 + N*0x20 | OVL_UI top field memory block low address register(N=0,1,2,3) |
| OVL_UI_BOT_LADD | 0x014 + N*0x20 | OVL_UI bottom field memory block low address register(N=0,1,2,3) |
| OVL_UI_FILL_COLOR | 0x018 + N*0x20 | OVL_UI fill color register(N=0,1,2,3) |
| OVL_UI_TOP_HADD | 0x080 | OVL_UI top field memory block high address register |
| OVL_UI_BOT_HADD | 0x084 | OVL_UI bottom field memory block high address register |
| OVL_UI_SIZE | 0x088 | OVL_UI overlay window size register |
| | | |
| BLD_FILLCOLOR_CTL | 0x000 | BLD fill color control register |
| BLD_FILL_COLOR | 0x004 + N*0x14 | BLD fill color register(N=0,1,2,3,4) |
| BLD_CH_ISIZE | 0x008 + N*0x14 | BLD input memory size register(N=0,1,2,3,4) |
| BLD_CH_OFFSET | 0x00C + N*0x14 | BLD input memory offset register(N=0,1,2,3,4) |
| BLD_CH_RTCTL | 0x080 | BLD routing control register |
| BLD_PREMUL_CTL | 0x084 | BLD pre-multiply control register |
| BLD_BK_COLOR | 0x088 | BLD background color register |
| BLD_SIZE | 0x08C | BLD output size setting register |
| BLD_CTL | 0x090 – 0x09C | BLD control register |

| | | |
|---|---|---|
| BLD_KEY_CTL | 0x0B0 | BLD color key control register |
| BLD_KEY_CON | 0x0B4 | BLD color key configuration register |
| BLD_KEY_MAX | 0x0C0 – 0x0CC | BLD color key max register |
| BLD_KEY_MIN | 0x0E0 – 0x0EC | BLD color key min register |
| BLD_OUT_COLOR | 0x0FC | BLD output color control register |

## 5.10.6 GLB Register Description

**Note:** all registers in GLB are not double buffer.

**Note:** all registers in GLB are not double buffer.

### 5.10.6.1 Global control register

| **Offset: 0x000** | | | **Register Name: GLB_CTL** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:14 | / | / | / |
| 13:12 | R/W | 0x0 | OUT_DATA_WB<br>Output data for RT-WB<br>0:RT-WB fetch data after DEP port<br>1:RT-WB fetch data before DEP port<br>Other: Reserved |
| 11:10 | / | / | / |
| 9 | R/W | 0x0 | lcd flied in reverse |
| 8 | R/W | 0x0 | lcd sync in reverse |
| 7:6 | / | / | / |
| 5 | R/W | 0x0 | ERROR_IRQ_EN<br>Hardware error IRQ enable<br>0:disable<br>1:enable |
| 4 | R/W | 0x0 | FINISH_IRQ_EN<br>Mission finish IRQ enable<br>0:disable<br>1:enable |
| 3:1 | / | / | / |
| 0 | R/W | 0x0 | EN<br>RT enable/disable<br>0: disable<br>1: enable |

### 5.10.6.2 Global status register

| **Offset: 0x004** | | | **Register Name: GLB_STS** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:9 | / | / | / |
| 8 | R | 0x0 | EVEN_ODD<br>0: even field |

| | | | |
|---|---|---|---|
| | | | 1: odd field<br>This bit is the flag for output data in interlace mode |
| 7:6 | / | / | / |
| 5 | R | 0x0 | ERROR<br>Hardware error status |
| 4 | R | 0x0 | BUSY<br>Module working status<br>0:idle<br>1:running busy |
| 3:2 | / | / | / |
| 1 | R/W | 0x0 | ERROR_IRQ<br>Hardware error IRQ<br>It will be set when hardware error occur, and cleared by writing 1. |
| 0 | R/W | 0x0 | FINISH_IRQ<br>Mission finish IRQ<br>It will be set when 1 frame operation accomplished, and cleared by writing 1. |

#### 5.10.6.3 Global double buffer control register

| Offset: 0x008 | | | Register Name: GLB_DBUFFER |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | DOUBLE_BUFFER_RDY<br>0: no change<br>1: register value be ready for update<br>Note:This bit is self-cleared by writing 1 after update. |

#### 5.10.6.4 Global size register

| Offset: 0x00C | | | Register Name: GLB_SIZE |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0x0 | HEIGHT<br>Height<br>The Height = The value of these bits add 1 |
| 15:13 | / | / | / |
| 12:0 | R/W | 0x0 | WIDTH<br>Width<br>The Width = The value of these bits add 1 |

### 5.10.7 OVL_V Register Description

**Note: all registers in OVL_V are double buffer.**

### 5.10.7.1 OVL_V attribute control register

| Offset: 0x000+N*0x30 (N=0,1,2,3) | | | Register Name: OVL_V_ATTCTL |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23 | R/W | 0x0 | LAY0_TOP_BOTTOM_ADDR_EN<br>0: disable<br>1: enable<br>When this bit is disable the layer address use top field address in both top and bottom field, otherwise use the top and bottom field address separate in top and bottom field. |
| 22:16 | / | / | / |
| 15 | R/W | 0x0 | Video_UI_SEL<br>Video Overlay or UI Overlay Select<br>0: Video Overlay(using Video Overlay Layer Input data format)<br>1: UI Overlay(using UI Overlay Layer Input data format) |
| 14:13 | / | / | / |
| 12:8 | R/W | 0x0 | LAY_FBFMT<br>Video Overlay Layer Input data format<br>0x00: Interleaved YUV422(V0Y1U0Y0)<br>0x01: Interleaved YUV422(Y1V0Y0U0)<br>0x02: Interleaved YUV422(U0Y1V0Y0)<br>0x03: Interleaved YUV422(Y1U0Y0V0)<br>0x04: Planar YUV422 UV combined(V1U1V0U0)<br>0x05: Planar YUV422 UV combined(U1V1U0V0)<br>0x06: Planar YUV422<br>0x07: Reserved<br>0x08: Planar YUV420 UV combined(V1U1V0U0)<br>0x09: Planar YUV420 UV combined(U1V1U0V0)<br>0x0A: Planar YUV420<br>0x0B: Reserved<br>0x0C: Planar YUV411 UV combined(V1U1V0U0)<br>0x0D: Planar YUV411 UV combined(U1V1U0V0)<br>0x0E: Planar YUV411<br>Other: Reserved<br>All video layers must be the same format, programmer should confirm it.<br><br>UI Overlay Layer Input data format<br>0x00: ARGB_8888<br>0x01: ABGR_8888<br>0x02: RGBA_8888<br>0x03: BGRA_8888<br>0x04: XRGB_8888<br>0x05: XBGR_8888<br>0x06: RGBX_8888 |

| | | | |
|---|---|---|---|
| | | | 0x07: BGRX_8888<br>0x08: RGB_888<br>0x09: BGR_888<br>0x0A: RGB_565<br>0x0B: BGR_565<br>0x0C: ARGB_4444<br>0x0D: ABGR_4444<br>0x0E: RGBA_4444<br>0x0F: BGRA_4444<br>0x10: ARGB_1555<br>0x11: ABGR_1555<br>0x12: RGBA_5551<br>0x13: BGRA_5551<br>Other: Reserved<br>All ui layers' alpha is useless. |
| 7:5 | / | / | / |
| 4 | R/W | 0x0 | LAY_FILLCOLOR_EN<br>0: disable<br>1:enable<br>When the layer fill-color is enabled, the layer data will use the fill-color. |
| 3:1 | / | / | / |
| 0 | R/W | 0x0 | LAY0_EN<br>Layer0 enable/disable<br>0: disabled<br>1: enabled |

**Note: the layer priority is layer3>layer2>layer1>layer0**

### 5.10.7.2 OVL_V memory block size register

| **Offset: 0x004+N*0x30 (N=0,1,2,3)** | | | **Register Name: OVL_V_MBSIZE** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0x0 | LAY_HEIGHT<br>Layer Height<br>The Layer Height = The value of these bits add 1 |
| 15:13 | / | / | / |
| 12:0 | R/W | 0x0 | LAY_WIDTH<br>Layer Width<br>The Layer Width = The value of these bits add 1 |

### 5.10.7.3 OVL_V memory block coordinate register

| **Offset: 0x008+N*0x30 (N=0,1,2,3)** | **Register Name: OVL_V_COOR** |
|---|---|

| Bit | Read/Write | Default/Hex | Description |
|---|---|---|---|
| 31:16 | R/W | 0x0 | LAY_YCOOR<br>Y coordinate<br>Y is the left-top y coordinate of layer on overlay window in pixels |
| 15:0 | R/W | 0x0 | LAY_XCOOR<br>X coordinate<br>X is left-top x coordinate of the layer on overlay window in pixels |

Setting the layer0-layer3 the coordinate (left-top) on overlay window control information

### 5.10.7.4 OVL_V memory pitch register0

| Offset: 0x00C+N*0x30 (N=0,1,2,3) | | | Register Name: OVL_V_PITCH0 |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0x0 | LAY_PITCH<br>Layer memory pitch in bytes |

**Note: The setting of this register is Y channel.**

### 5.10.7.5 OVL_V memory pitch register1

| Offset: 0x010+N*0x30 (N=0,1,2,3) | | | Register Name: OVL_V_PITCH1 |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0x0 | LAY_PITCH<br>Layer memory pitch in bytes |

**Note: The setting of this register is U/UV channel.**

### 5.10.7.6 OVL_V memory pitch register2

| Offset: 0x014+N*0x30 (N=0,1,2,3) | | | Register Name: OVL_V_PITCH2 |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0x0 | LAY_PITCH<br>Layer memory pitch in bytes |

**Note: The setting of this register is V channel.**

### 5.10.7.7 OVL_V top field memory block low address register0

| Offset: 0x018+N*0x30 (N=0,1,2,3) | | | Register Name: OVL_V_TOP_LADD0 |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0x0 | LAYMB_LADD<br>Memory Block Start Address<br>Layer Memory Block Address in bytes |

**Note: The setting of this register is Y channel address.**

### 5.10.7.8 OVL_V top field memory block low address register1

| Offset: 0x01C+N*0x30 (N=0,1,2,3) | | | Register Name: OVL_V_TOP_LADD1 |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0x0 | LAYMB_LADD<br>Memory Block Start Address<br>Layer Memory Block Address in bytes |

**Note: The setting of this register is U/UV channel address.**

### 5.10.7.9 OVL_V top field memory block low address register2

| Offset: 0x020+N*0x30 (N=0,1,2,3) | | | Register Name: OVL_V_TOP_LADD2 |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0x0 | LAYMB_LADD<br>Memory Block Start Address<br>Layer Memory Block Address in bytes |

**Note: The setting of this register is V channel address.**

### 5.10.7.10 OVL_V bottom field memory block low address register0

| Offset: 0x024+N*0x30 (N=0,1,2,3) | | | Register Name: OVL_V_BOT_LADD0 |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0x0 | LAYMB_LADD<br>Memory Block Start Address<br>Layer Memory Block Address in bytes |

**Note: The setting of this register is Y channel address.**

### 5.10.7.11 OVL_V bottom field memory block low address register1

| Offset: 0x028+N*0x30 (N=0,1,2,3) | | | Register Name: OVL_V_BOT_LADD1 |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0x0 | LAYMB_LADD<br>Memory Block Start Address<br>Layer Memory Block Address in bytes |

**Note: The setting of this register is U/UV channel address.**

### 5.10.7.12 OVL_V bottom field memory block low address register2

| Offset: 0x02C+N*0x30 (N=0,1,2,3) | Register Name: OVL_V_BOT_LADD2 |
|---|---|

| Bit | Read/Write | Default/Hex | Description |
|---|---|---|---|
| 31:0 | R/W | 0x0 | LAYMB_LADD<br>Memory Block Start Address<br>Layer Memory Block Address in bytes |

**Note: The setting of this register is V channel address.**

### 5.10.7.13 OVL_V fill color register

| Offset: 0x0C0+N*0x4 (N=0,1,2,3) | | | Register Name: OVL_V_FILL_COLOR |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23:16 | R/W | 0x0 | Y/R<br>Y/Red fill color value |
| 15:8 | R/W | 0x0 | U/G<br>U/Green fill color value |
| 7:0 | R/W | 0x0 | V/B<br>V/Blue fill color value |

### 5.10.7.14 OVL_V top field memory block high address register0

| Offset: 0x0D0 | | | Register Name: OVL_V_TOP_HADD0 |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | R/W | 0x0 | LAY3MB_HADD<br>Layer3<br>Layer Memory Block Address in bytes |
| 23:16 | R/W | 0x0 | LAY2MB_HADD<br>Layer2<br>Layer Memory Block Address in bytes |
| 15:8 | R/W | 0x0 | LAY1MB_HADD<br>Layer1<br>Layer Memory Block Address in bytes |
| 7:0 | R/W | 0x0 | LAY0MB_HADD<br>Layer0<br>Layer Memory Block Address in bytes |

**Note: The setting of this register is Y channel address.**

### 5.10.7.15 OVL_V top field memory block high address register1

| Offset: 0x0D4 | | | Register Name: OVL_V_TOP_HADD1 |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | R/W | 0x0 | LAY3MB_HADD<br>Layer3<br>Layer Memory Block Address in bytes |
| 23:16 | R/W | 0x0 | LAY2MB_HADD |

| | | | |
|---|---|---|---|
| | | | Layer2<br>Layer Memory Block Address in bytes |
| 15:8 | R/W | 0x0 | LAY1MB_HADD<br>Layer1<br>Layer Memory Block Address in bytes |
| 7:0 | R/W | 0x0 | LAY0MB_HADD<br>Layer0<br>Layer Memory Block Address in bytes |

**Note: The setting of this register is U/UV channel address.**

### 5.10.7.16 OVL_V top field memory block high address register2

| **Offset: 0x0D8** | | | **Register Name: OVL_V_TOP_HADD2** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | R/W | 0x0 | LAY3MB_HADD<br>Layer3<br>Layer Memory Block Address in bytes |
| 23:16 | R/W | 0x0 | LAY2MB_HADD<br>Layer2<br>Layer Memory Block Address in bytes |
| 15:8 | R/W | 0x0 | LAY1MB_HADD<br>Layer1<br>Layer Memory Block Address in bytes |
| 7:0 | R/W | 0x0 | LAY0MB_HADD<br>Layer0<br>Layer Memory Block Address in bytes |

**Note: The setting of this register is V channel address.**

### 5.10.7.17 OVL_V bottom field memory block high address register0

| **Offset: 0x0DC** | | | **Register Name: OVL_V_BOT_HADD0** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | R/W | 0x0 | LAY3MB_HADD<br>Layer3<br>Layer Memory Block Address in bytes |
| 23:16 | R/W | 0x0 | LAY2MB_HADD<br>Layer2<br>Layer Memory Block Address in bytes |
| 15:8 | R/W | 0x0 | LAY1MB_HADD<br>Layer1<br>Layer Memory Block Address in bytes |
| 7:0 | R/W | 0x0 | LAY0MB_HADD<br>Layer0<br>Layer Memory Block Address in bytes |

**Note: The setting of this register is Y channel address.**

### 5.10.7.18 OVL_V bottom field memory block high address register1

| Offset: 0x0E0 | | | Register Name: OVL_V_BOT_HADD1 |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31:24 | R/W | 0x0 | LAY3MB_HADD<br>Layer3<br>Layer Memory Block Address in bytes |
| 23:16 | R/W | 0x0 | LAY2MB_HADD<br>Layer2<br>Layer Memory Block Address in bytes |
| 15:8 | R/W | 0x0 | LAY1MB_HADD<br>Layer1<br>Layer Memory Block Address in bytes |
| 7:0 | R/W | 0x0 | LAY0MB_HADD<br>Layer0<br>Layer Memory Block Address in bytes |

**Note: The setting of this register is U/UV channel address.**

### 5.10.7.19 OVL_V bottom field memory block high address register2

| Offset: 0x0E4 | | | Register Name: OVL_V_BOT_HADD2 |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31:24 | R/W | 0x0 | LAY3MB_HADD<br>Layer3<br>Layer Memory Block Address in bytes |
| 23:16 | R/W | 0x0 | LAY2MB_HADD<br>Layer2<br>Layer Memory Block Address in bytes |
| 15:8 | R/W | 0x0 | LAY1MB_HADD<br>Layer1<br>Layer Memory Block Address in bytes |
| 7:0 | R/W | 0x0 | LAY0MB_HADD<br>Layer0<br>Layer Memory Block Address in bytes |

**Note: The setting of this register is V channel address.**

### 5.10.7.20 OVL_V overlay window size register

| Offset: 0x0E8 | | | Register Name: OVL_V_SIZE |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31:29 | / | / | / |
| 28:16 | R/W | 0x0 | HEIGHT<br>Overlay Height<br>The Overlay Height = The value of these bits add 1 |

| | | | |
|---|---|---|---|
| 15:13 | / | / | / |
| 12:0 | R/W | 0x0 | WIDTH<br>Overlay Width<br>The Overlay Width = The value of these bits add 1 |

**when all the layers are disable the overlay has no output data, and by pass.**

### 5.10.7.21 OVL_V horizontal down sampling control register0

| Offset: 0x0F0 | | | Register Name: OVL_V_HDS_CTL0 |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:30 | / | / | / |
| 29:16 | R/W | 0x0 | N<br>The step size for overlay data fetch.<br>Note: Refer to "Figure 6 Video overlay data fetch rule" |
| 15:14 | / | / | / |
| 13:0 | R/W | 0x0 | M<br>The counter threshold for fetch data<br>Note: Refer to "Figure 6 Video overlay data fetch rule" |

**The setting of this register is Y channel.**

### 5.10.7.22 OVL_V horizontal down sampling control register1

| Offset: 0x0F4 | | | Register Name: OVL_V_HDS_CTL1 |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:30 | / | / | / |
| 29:16 | R/W | 0x0 | N<br>The step size for overlay data fetch.<br>Note: Refer to "Figure 6 Video overlay data fetch rule" |
| 15:14 | / | / | / |
| 13:0 | R/W | 0x0 | M<br>The counter threshold for fetch data<br>Note: Refer to "Figure 6 Video overlay data fetch rule" |

**The setting of this register is UV channel.**

### 5.10.7.23 OVL_V vertical down sampling control register0

| Offset: 0x0F8 | | | Register Name: OVL_V_VDS_CTL0 |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:30 | / | / | / |
| 29:16 | R/W | 0x0 | N<br>The step size for overlay data fetch.<br>Note: Refer to "Figure 6 Video overlay data fetch rule" |
| 15:14 | / | / | / |
| 13:0 | R/W | 0x0 | M<br>The counter threshold for fetch data |

| | | | |
|---|---|---|---|
| | | | Note: Refer to "Figure 6 Video overlay data fetch rule" |

**The setting of this register is Y channel.**

### 5.10.7.24 OVL_V vertical down sampling control register1

| Offset: 0x0FC | | | Register Name: OVL_V_VDS_CTL1 |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:30 | / | / | / |
| 29:16 | R/W | 0x0 | N<br>The step size for overlay data fetch.<br>Note: Refer to "Figure 6 Video overlay data fetch rule" |
| 15:14 | / | / | / |
| 13:0 | R/W | 0x0 | M<br>The counter threshold for fetch data<br>Note: Refer to "Figure 6 Video overlay data fetch rule" |

**The setting of this register is UV channel.**

## 5.10.8 OVL_UI Register Description

**Note: all registers in OVL_UI are double buffer.**

### 5.10.8.1 OVL_UI attribute control register

| Offset: 0x000+N*0x20 (N=0,1,2,3) | | | Register Name: OVL_UI_ATTR_CTL |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | R/W | 0x0 | LAY_GLBALPHA<br>Globe alpha value<br>Alpha value is used for this layer |
| 23 | R/W | 0x0 | TOP_BOTTOM_ADDR_EN<br>0: disable<br>1: enable<br>When this bit is disable the layer address use top field address in both top and bottom field, otherwise use the top and bottom field address separate in top and bottom field. |
| 22:18 | / | / | / |
| 17:16 | R/W | 0x0 | LAY_PREMUL_CTL<br>Layer input pre-multiply alpha control<br>0x0: Input layer data is non-pre-multiply data and it needn't to unify pre-multiply data.<br>0x1: Input layer data is non-pre-multiply data and it need to unify pre-multiply data.<br>0x2: pre-multiply input layer<br>Other: Reserved |
| 15:13 | / | / | / |

| | | | |
|---|---|---|---|
| 12:8 | R/W | 0x0 | LAY_FBFMT<br>Input data format<br>0x00: ARGB_8888<br>0x01: ABGR_8888<br>0x02: RGBA_8888<br>0x03: BGRA_8888<br>0x04: XRGB_8888<br>0x05: XBGR_8888<br>0x06: RGBX_8888<br>0x07: BGRX_8888<br>0x08: RGB_888<br>0x09: BGR_888<br>0x0A: RGB_565<br>0x0B: BGR_565<br>0x0C: ARGB_4444<br>0x0D: ABGR_4444<br>0x0E: RGBA_4444<br>0x0F: BGRA_4444<br>0x10: ARGB_1555<br>0x11: ABGR_1555<br>0x12: RGBA_5551<br>0x13: BGRA_5551<br>Other: Reserved |
| 7:5 | / | / | / |
| 4 | R/W | 0x0 | LAY_FILLCOLOR_EN<br>0: disable<br>1:enable<br>When the layer fill-color is enabled, the layer data will use the fill-color. |
| 3 | / | / | / |
| 2:1 | R/W | 0x0 | LAY_ALPHA_MODE<br>Layer input alpha mode<br>0:Ignore<br>Input alpha value = pixels alpha, if no pixel alpha, the alpha value equal 0xff<br><br>1:Globe alpha enable<br>Ignore pixel alpha value<br>Input alpha value = globe alpha value<br><br>2: Globe alpha mix pixel alpha<br>Input alpha value = globe alpha value * pixels alpha value<br>3:Reserved |
| 0 | R/W | 0x0 | LAY_EN<br>Layer enable/disable<br>0: disabled<br>1: enabled |

**Note: the layer priority is layer3>layer2>layer1>layer0**

### 5.10.8.2 OVL_UI memory block size register

| Offset: 0x004+N*0x20 (N=0,1,2,3) | | | Register Name: OVL_UI_MBSIZE |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0x0 | LAY_HEIGHT<br>Layer Height<br>The Layer Height = The value of these bits add 1 |
| 15:13 | / | / | / |
| 12:0 | R/W | 0x0 | LAY_WIDTH<br>Layer Width<br>The Layer Width = The value of these bits add 1 |

### 5.10.8.3 OVL_UI memory block coordinate register

| Offset: 0x008+N*0x20 (N=0,1,2,3) | | | Register Name: OVL_UI_COOR |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:16 | R/W | 0x0 | LAY_YCOOR<br>Y coordinate<br>Y is the left-top y coordinate of layer on overlay window in pixels |
| 15:0 | R/W | 0x0 | LAY_XCOOR<br>X coordinate<br>X is left-top x coordinate of the layer on overlay window in pixels |

**Setting the layer0-layer3 the coordinate (left-top) on overlay window control information**

### 5.10.8.4 OVL_UI memory pitch register

| Offset: 0x00C+N*0x20 (N=0,1,2,3) | | | Register Name: OVL_UI_PITCH |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0x0 | LAY_PITCH<br>Layer memory pitch in bytes |

### 5.10.8.5 OVL_UI top field memory block low address register

| Offset: 0x010+N*0x20 (N=0,1,2,3) | | | Register Name: OVL_UI_TOP_LADD |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0x0 | LAYMB_LADD<br>Memory Block Start Address<br>Layer Memory Block Address in bytes |

### 5.10.8.6 OVL_UI bottom field memory block low address register

| Offset: 0x014+N*0x20 (N=0,1,2,3) | | | Register Name: OVL_UI_BOT_LADD |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0x0 | LAYMB_LADD<br>Memory Block Start Address<br>Layer Memory Block Address in bytes |

### 5.10.8.7 OVL_UI fill color register

| Offset: 0x018+N*0x20 (N=0,1,2,3) | | | Register Name: OVL_UI_FILL_COLOR |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | R/W | 0x0 | Alpha<br>Alpha fill color value |
| 23:16 | R/W | 0x0 | RED<br>Red fill color value |
| 15:8 | R/W | 0x0 | GREEN<br>Green fill color value |
| 7:0 | R/W | 0x0 | BLUE<br>Blue fill color value |

### 5.10.8.8 OVL_UI top field memory block high address register

| Offset: 0x080 | | | Register Name: OVL_UI_TOP_HADD |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | R/W | 0x0 | LAY3MB_HADD<br>Layer3<br>Layer Memory Block Address in bytes |
| 23:16 | R/W | 0x0 | LAY2MB_HADD<br>Layer2<br>Layer Memory Block Address in bytes |
| 15:8 | R/W | 0x0 | LAY1MB_HADD<br>Layer1<br>Layer Memory Block Address in bytes |
| 7:0 | R/W | 0x0 | LAY0MB_HADD<br>Layer0<br>Layer Memory Block Address in bytes |

### 5.10.8.9 OVL_UI bottom field memory block high address register

| Offset: 0x084 | | | Register Name: OVL_UI_BOT_HADD |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | R/W | 0x0 | LAY3MB_HADD<br>Layer3 |

| | | | |
|---|---|---|---|
| | | | Layer Memory Block Address in bytes |
| 23:16 | R/W | 0x0 | LAY2MB_HADD<br>Layer2<br>Layer Memory Block Address in bytes |
| 15:8 | R/W | 0x0 | LAY1MB_HADD<br>Layer1<br>Layer Memory Block Address in bytes |
| 7:0 | R/W | 0x0 | LAY0MB_HADD<br>Layer0<br>Layer Memory Block Address in bytes |

#### 5.10.8.10 OVL_UI overlay window size register

| Offset: 0x088 | | | Register Name: OVL_UI_SIZE |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0x0 | HEIGHT<br>Overlay Height<br>The Overlay Height = The value of these bits add 1 |
| 15:13 | / | / | / |
| 12:0 | R/W | 0x0 | WIDTH<br>Overlay Width<br>The Overlay Width = The value of these bits add 1 |

**Note: When all the layers are disable the overlay has no output data, and by pass.**

**Channel2~4 register definition is same as OVL_UI's register definition.**

### 5.10.9 BLD Register Description

Note: all registers in BLD are double buffer.

#### 5.10.9.1 BLD fill color control register

| Offset: 0x000 | | | Register Name: BLD_FILL_COLOR_CTL |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12 | R/W | 0x0 | P4_EN<br>Pipe4 enable/disable<br>0: disabled<br>1: enabled |
| 11 | R/W | 0x0 | P3_EN<br>Pipe3 enable/disable<br>0: disabled<br>1: enabled |
| 10 | R/W | 0x0 | P2_EN<br>Pipe2 enable/disable |

| | | | |
|---|---|---|---|
| | | | 0: disabled<br>1: enabled |
| 9 | R/W | 0x0 | P1_EN<br>Pipe1 enable/disable<br>0: disabled<br>1: enabled |
| 8 | R/W | 0x0 | P0_EN<br>Pipe0 enable/disable<br>0: disabled<br>1: enabled |
| 7:5 | / | / | / |
| 4 | R/W | 0x0 | P4_FCEN<br>Pipe4 fill color enable/disable<br>0: disabled<br>1: enabled |
| 3 | R/W | 0x0 | P3_FCEN<br>Pipe3 fill color enable/disable<br>0: disabled<br>1: enabled |
| 2 | R/W | 0x0 | P2_FCEN<br>Pipe2 fill color enable/disable<br>0: disabled<br>1: enabled |
| 1 | R/W | 0x0 | P1_FCEN<br>Pipe1 fill color enable/disable<br>0: disabled<br>1: enabled |
| 0 | R/W | 0x0 | P0_FCEN<br>Pipe0 fill color enable/disable<br>0: disabled<br>1: enabled |

### 5.10.9.2 BLD fill color register

| **Offset: 0x004+N*0x10 (N=0,1,2,3,4)** | | | **Register Name: BLD_FILL_COLOR** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | R/W | 0x0 | Alpha<br>Alpha fill color value |
| 23:16 | R/W | 0x0 | RED<br>Red fill color value |
| 15:8 | R/W | 0x0 | GREEN<br>Green fill color value |
| 7:0 | R/W | 0x0 | BLUE<br>Blue fill color value |

### 5.10.9.3 BLD input memory size register

| Offset: 0x008+N*0x10 (N=0,1,2,3,4) | | | Register Name: BLD_CH_ISIZE |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0x0 | HEIGHT<br>Pipe input memory height<br>The input height = The value of these bits add 1 |
| 15:13 | / | / | / |
| 12:0 | R/W | 0x0 | WIDTH<br>Pipe input memory width<br>The input width = The value of these bits add 1 |

### 5.10.9.4 BLD input memory offset register

| Offset: 0x00C+N*0x10 (N=0,1,2,3,4) | | | Register Name: BLD_CH_OFFSET |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:16 | R/W | 0x0 | YCOOR<br>Y coordinate<br>Y is the left-top y coordinate of the pipe on blender memory window in pixels |
| 15:0 | R/W | 0x0 | XCOOR<br>X coordinate<br>X is the left-top x coordinate of the pipe on blender memory window in pixels |

### 5.10.9.5 BLD routing control register

| Offset: 0x080 | | | Register Name: BLD_CH_RTCTL |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:16 | / | / | / |
| 15:12 | R/W | 0x3 | P3_RTCTL<br>Pipe3 select channel control<br>0:from channel0<br>1:from channel1<br>2:from channel2<br>3:from channel3<br>Other: reserved |
| 11:8 | R/W | 0x2 | P2_RTCTL<br>Pipe2 select channel control<br>0:from channel0<br>1:from channel1<br>2:from channel2<br>3:from channel3<br>Other: reserved |
| 7:4 | R/W | 0x1 | P1_RTCTL |

| | | | |
|---|---|---|---|
| | | | Pipe1 select channel control<br>0:from channel0<br>1:from channel1<br>2:from channel2<br>3:from channel3<br>Other: reserved |
| 3:0 | R/W | 0x0 | P0_RTCTL<br>Pipe0 select channel control<br>0:from channel0<br>1:from channel1<br>2:from channel2<br>3:from channel3<br>Other: reserved |

**Note: Setting 2 or more channels in the same pipe is illegal, programmer should confirm it.**

### 5.10.9.6 BLD pre-multiply control register

| Offset: 0x084 | | | Register Name: BLD_PREMUL_CTL |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:4 | / | / | / |
| 3 | R/W | 0x0 | P3_ALPHA_MODE<br>Pipe3 input alpha mode<br>0:all alpha data is no-pre-multiply alpha<br>1:all alpha data is pre-multiply alpha |
| 2 | R/W | 0x0 | P2_ALPHA_MODE<br>Pipe2 input alpha mode<br>0:all alpha data is no-pre-multiply alpha<br>1:all alpha data is pre-multiply alpha |
| 1 | R/W | 0x0 | P1_ALPHA_MODE<br>Pipe1 input alpha mode<br>0:all alpha data is no-pre-multiply alpha<br>1:all alpha data is pre-multiply alpha |
| 0 | R/W | 0x0 | P0_ALPHA_MODE<br>Pipe0 input alpha mode<br>0:all alpha data is no-pre-multiply alpha<br>1:all alpha data is pre-multiply alpha |

### 5.10.9.7 BLD background color register

| Offset: 0x088 | | | Register Name: BLD_BK_COLOR |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23:16 | R/W | 0x0 | RED<br>Red background color value |
| 15:8 | R/W | 0x0 | GREEN |

| | | | |
|---|---|---|---|
| | | | Green background color value |
| 7:0 | R/W | 0x0 | BLUE<br>Blue background color value |

### 5.10.9.8 BLD output size setting register

| Offset: 0x08C | | | Register Name: BLD_SIZE |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0x0 | BLD_HEIGHT<br>Blender height<br>The real blender height = The value of these bits add 1 |
| 15:13 | / | / | / |
| 12:0 | R/W | 0x0 | BLD_WIDTH<br>Blender width<br>The real blender width = The value of these bits add 1 |

### 5.10.9.9 BLD control register

| Offset:<br>Blender0: 0x090<br>Blender1: 0x094<br>Blender2: 0x098<br>Blender3: 0x09C | | | Register Name: BLD_CTL |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:28 | / | / | / |
| 27:24 | R/W | 0x3 | BLEND_AFD<br>Specifies the coefficient that used in destination alpha data $Q_d$.<br>0x0: 0<br>0x1: 1<br>0x2: $A_s$<br>0x3: 1-$A_s$<br>Other: Reserved |
| 23:20 | / | / | / |
| 19:16 | R/W | 0x1 | BLEND_AFS<br>Specifies the coefficient that used in source alpha data $Q_s$.<br>0x0: 0<br>0x1: 1<br>0x2: $A_d$<br>0x3: 1-$A_d$<br>Other: Reserved |
| 15:12 | / | / | / |
| 11:8 | R/W | 0x3 | BLEND_PFD<br>Specifies the coefficient that used in destination pixel data $F_d$.<br>0x0: 0 |

| | | | |
|---|---|---|---|
| | | | 0x1: 1<br>0x2: $A_s$<br>0x3: 1-$A_s$<br>Other: Reserved |
| 7:4 | / | / | / |
| 3:0 | R/W | 0x1 | BLEND_PFS<br>Specifies the coefficient that used in source pixel data $F_s$.<br>0x0: 0<br>0x1: 1<br>0x2: $A_d$<br>0x3: 1-$A_d$<br>Other: Reserved |

### 5.10.9.10 BLD color key control register

| Offset: 0x0B0 | | | Register Name: BLD_KEY_CTL |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:15 | / | / | |
| 14:13 | R/W | 0x0 | KEY3_MATCH_DIR<br>In Alpha Blender3<br>0: when the pixel value matches destination image, it displays the pixel form source image.<br>1: when the pixel value matches source image, it displays the pixel form destination image.<br>1x: Reserved |
| 12 | R/W | 0x0 | KEY3_EN<br>Enable color key<br>0: disabled color key<br>1: enable color key in Alpha Blender3. |
| 11 | / | / | / |
| 10:9 | R/W | 0x0 | KEY2_MATCH_DIR<br>In Alpha Blender2<br>0: when the pixel value matches destination image, it displays the pixel form source image.<br>1: when the pixel value matches source image, it displays the pixel form destination image.<br>1x: Reserved |
| 8 | R/W | 0x0 | KEY2_EN<br>Enable color key<br>0: disabled color key<br>1: enable color key in Alpha Blender2. |
| 7 | / | / | |
| 6:5 | R/W | 0x0 | KEY1_MATCH_DIR<br>In Alpha Blender1<br>0: when the pixel value matches destination image, it displays the pixel form source image. |

| | | | |
|---|---|---|---|
| | | | 1: when the pixel value matches source image, it displays the pixel form destination image.<br>1x: Reserved |
| 4 | R/W | 0x0 | KEY1_EN<br>Enable color key<br>0: disabled color key<br>1: enable color key in Alpha Blender1. |
| 3 | / | / | |
| 2:1 | R/W | 0x0 | KEY0_MATCH_DIR<br>In Alpha Blender0<br>0: when the pixel value matches destination image, it displays the pixel form source image.<br>1: when the pixel value matches source image, it displays the pixel form destination image.<br>1x: Reserved |
| 0 | R/W | 0x0 | KEY0_EN<br>Enable color key<br>0: disabled color key<br>1: enable color key in Alpha Blender0. |

### 5.10.9.11 BLD color key configuration register

| Offset: 0x0B4 | | | Register Name: BLD_KEY_CON |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:27 | / | / | / |
| 26 | R/W | 0x0 | KEY3R_MATCH<br>Red Match Rule<br>0: match if (Color Min=<Color<=Color Max) the red condition is true, else the condition is false.<br>1: match if (Color>Color Max or Color<Color Min) the red condition is true, else the condition is false. |
| 25 | R/W | 0x0 | KEY3G_MATCH<br>Green Match Rule<br>0: match if (Color Min=<Color<=Color Max) the green condition is true, else the condition is false.<br>1: match if (Color>Color Max or Color<Color Min) the green condition is true, else the condition is false. |
| 24 | R/W | 0x0 | KEY3B_MATCH<br>Blue Match Rule<br>0: match if (Color Min=<Color<=Color Max) the blue condition is true, else the condition is false.<br>1: match if (Color>Color Max or Color<Color Min) the blue condition is true, else the condition is false. |
| 23:19 | / | / | / |
| 18 | R/W | 0x0 | KEY2R_MATCH<br>Red Match Rule |

| | | | |
|---|---|---|---|
| | | | 0: match if (Color Min=<Color<=Color Max) the red condition is true, else the condition is false.<br>1: match if (Color>Color Max or Color<Color Min) the red condition is true, else the condition is false. |
| 17 | R/W | 0x0 | KEY2G_MATCH<br>Green Match Rule<br>0: match if (Color Min=<Color<=Color Max) the green condition is true, else the condition is false.<br>1: match if (Color>Color Max or Color<Color Min) the green condition is true, else the condition is false. |
| 16 | R/W | 0x0 | KEY2B_MATCH<br>Blue Match Rule<br>0: match if (Color Min=<Color<=Color Max) the blue condition is true, else the condition is false.<br>1: match if (Color>Color Max or Color<Color Min) the blue condition is true, else the condition is false. |
| 15:11 | / | / | / |
| 10 | R/W | 0x0 | KEY1R_MATCH<br>Red Match Rule<br>0: match if (Color Min=<Color<=Color Max) the red condition is true, else the condition is false.<br>1: match if (Color>Color Max or Color<Color Min) the red condition is true, else the condition is false. |
| 9 | R/W | 0x0 | KEY1G_MATCH<br>Green Match Rule<br>0: match if (Color Min=<Color<=Color Max) the green condition is true, else the condition is false.<br>1: match if (Color>Color Max or Color<Color Min) the green condition is true, else the condition is false. |
| 8 | R/W | 0x0 | KEY1B_MATCH<br>Blue Match Rule<br>0: match if (Color Min=<Color<=Color Max) the blue condition is true, else the condition is false.<br>1: match if (Color>Color Max or Color<Color Min) the blue condition is true, else the condition is false. |
| 7:3 | / | / | / |
| 2 | R/W | 0x0 | KEY0R_MATCH<br>Red Match Rule<br>0: match if (Color Min=<Color<=Color Max) the red condition is true, else the condition is false.<br>1: match if (Color>Color Max or Color<Color Min) the red condition is true, else the condition is false. |
| 1 | R/W | 0x0 | KEY0G_MATCH<br>Green Match Rule<br>0: match if (Color Min=<Color<=Color Max) the green condition is true, else the condition is false. |

| | | | |
|---|---|---|---|
| | | | 1: match if (Color>Color Max or Color<Color Min) the green condition is true, else the condition is false. |
| 0 | R/W | 0x0 | KEY0B_MATCH<br>Blue Match Rule<br>0: match if (Color Min=<Color<=Color Max) the blue condition is true, else the condition is false.<br>1: match if (Color>Color Max or Color<Color Min) the blue condition is true, else the condition is false. |

**Note: when R/G/B channel condition is true the color pass through otherwise is false.**

### 5.10.9.12 BLD color key max register

| **Offset:**<br>**CK0: 0x0C0**<br>**CK1: 0x0C4**<br>**CK2: 0x0C8**<br>**CK3: 0x0CC** | | | **Register Name: BLD_KEY_MAX** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23:16 | R/W | 0x0 | MAX_R<br>Red<br>Red color key max |
| 15:8 | R/W | 0x0 | MAX_G<br>Green<br>Green color key max |
| 7:0 | R/W | 0x0 | MAX_B<br>Blue<br>Blue color key max |

### 5.10.9.13 BLD color key min register

| **Offset:**<br>**CK0: 0x0E0**<br>**CK1: 0x0E4**<br>**CK2: 0x0E8**<br>**CK3: 0x0EC** | | | **Register Name: BLD_KEY_MIN** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23:16 | R/W | 0x0 | MIN_R<br>Red<br>Red color key min |
| 15:8 | R/W | 0x0 | MIN_G<br>Green<br>Green color key min |
| 7:0 | R/W | 0x0 | MIN_B |

| | | | |
|---|---|---|---|
| | | | Blue<br>Blue color key min |

### 5.10.9.14 BLD output color control register

| Offset: 0x0FC | | | Register Name: BLD_OUT_COLOR |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:2 | / | / | / |
| 1 | R/W | 0x0 | ITLMOD_EN<br>Output interlace mode enable<br>0:disable<br>1:enable<br>When output interlace mode software programmer should confirm the blender output height is even. |
| 0 | R/W | 0x0 | PREMUL_EN<br>Output color control<br>0: output normal data(A ' will be $A_{abcde}$ ')<br>1: output pre-multiply data(A ' = 1) |

# 5.11 DE RT-WB Controller Specification

## 5.11.1 Overview

The Real-time write-back controller (RT-WB) provides data capture function for display engine. It captures data from RT-mixer module, performs the image resizing function, and then write-back to SDRAM.

The RT-WB can receive RGB888 or YUV444 data format, and then converts to YV12/NV12/NV21 or interleaved RGB888/pRGB888/pBGR888 for write-back. Horizontal and vertical direction scaling-down are implemented independently.

The RT-WB features:

- Support RGB888 and YUV444 input data format
- Support input size from 8×4 to 4096×4096
- Support output size from 8×4 to **W**×4096 (**W** is RTL programmable)
- Support fine down scaling ratio from 1× to 1/2×, and the anti-aliasing filter is 16-phase 4-tap in horizontal , 16-phase 2-tap filter in vertical.
- Support coarse down scaling.

**Table 5-1. RTL programmable parameters table**

| Parameter name | Default | Possible value | Description |
|---|---|---|---|
| W | 2048 | 1280/2048/4096 | 3 channels line buffer length |
| FINE_SCALE_EXIST | 1 | 1/0 | Define fine scaler exist or not. |
| PORT_NUM | 4 | 2/4/8 | Input port number |
| M | 16 | 16 | Phase number |
| FRAC | 18 | 18 | Phase adder fraction part bit width |

## 5.11.2 Block Diagram

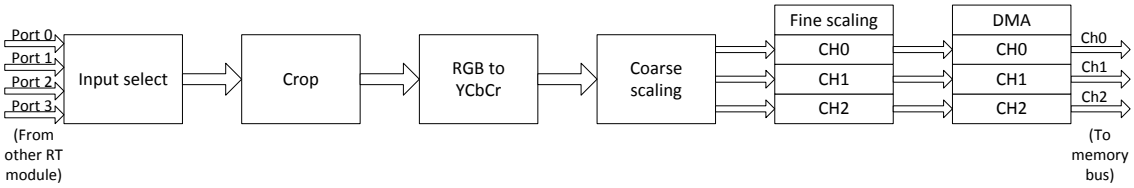



**Figure 5-7. RT-WB Block Diagram**

## 5.11.3 Operations and Function Descriptions

### 5.11.3.1 Write-back flow

The main flow when new frame starts description in Figure 5-8.

The new frame starts with a positive edge of *START*, which is a input from RT-TOP. RT-WB module will check signals like BUSY, OVERFLOW, FINISH to determine the last frame is success to write-back or not. If BUSY is still 1, TIME_OUT will be set, and users will know that the last frame did not write-back successfully. The OVERFLOW is also a error status signal.

It represents the module is too slow to write-back all the data to memory bus in last frame. These two errors will cause a local reset action to clear the whole scaler and some parts in DMA circuit.

A write-back address switch circuit can select the next write-back address automatically. The detail switch function description in Figure 5-9.

After local reset action or address switch, the circuit will check the WB_START is set or not. When WB_START is 1, the write-back function will be activated and this frame will be processed.

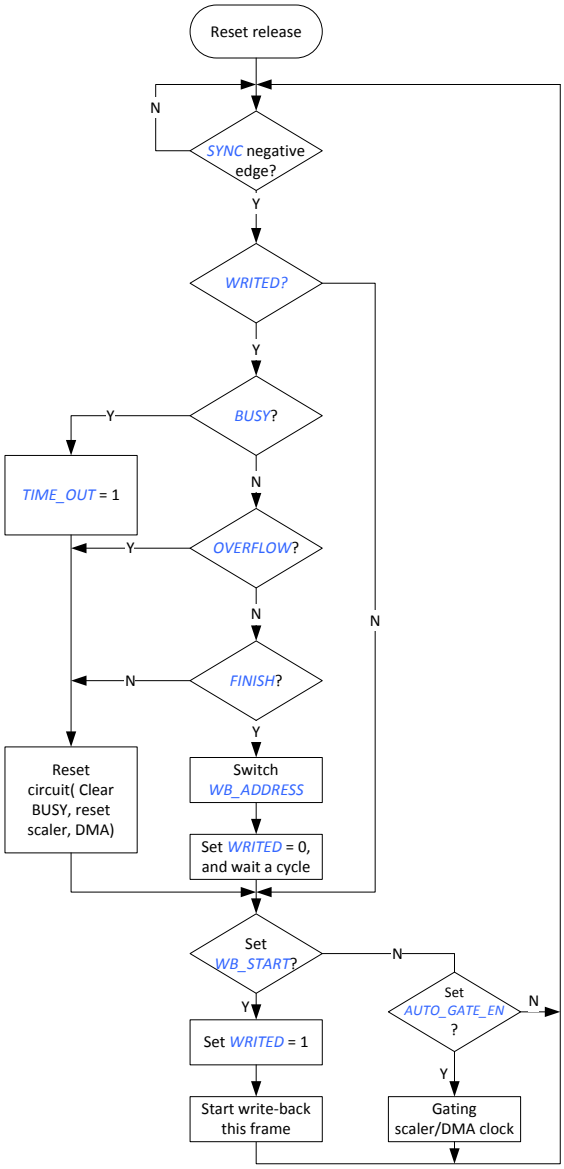



**Figure 5-8. The main control flow of RT-WB**

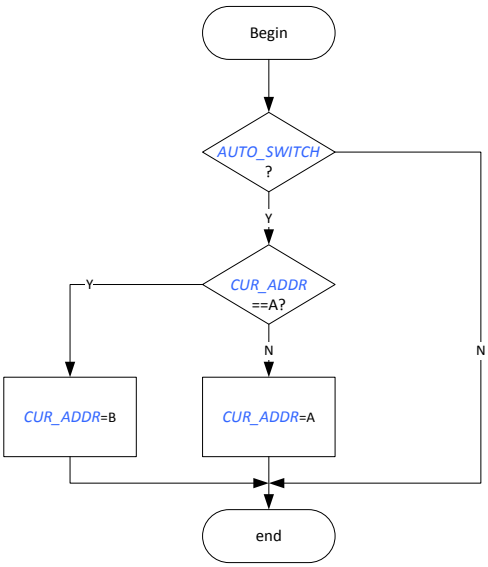



**Figure 5-9. Write-back address switch function**

### 5.11.3.2 Key signal and key register

| Signal | Type | Description | Note |
|---|---|---|---|
| **BUSY** | Read only register | Scaler or DMA busy status | Becomes 1 when WB_START takes effect, becomes 0 when FIFO overflowed,write-back time-out or write-back finished, or local reset, or global reset. |
| **WB_START** | R/W register | Write-back enable register | Set 1 to start a new frame write-back, becomes 0 when write-back started, or global reset. |
| **SYNC** | Input signal | New frame start signal from RT TOP module | Negative edge of SYNC represents a new display frame. |
| **OVERFLOW** | R/W register | FIFO overflowed flag | Becomes 1 when FIFO overflowed ( write-back too slow), becomes 0 when write 1 to it, or global reset. |
| **FINISH** | R/W register | Write-back successful finish flag | Becomes 1 when write-back finished successfully, becomes 0 when write 1 to it, or global reset. |
| **TIME_OUT** | R/W register | Write-back time out flag | When SYNC negative edge comes, write-back has not finished yet , this bit will set 1. Becomes 0 when write 1 to it, or global reset. |
| **WRITED** | Internal signal | One frame has been written-back | Becomes 1 when WB_START takes effect, becomes 0 when next SYNC negative edge comes, or global reset. |
| **IRQ** | R/W register | Write-back end flag | When WRITED negative edge comes, IRQ becomes 1, becomes 0 when write 1 to it, or global reset. |
| **INTR** | Output signal | Interrupt signal to GIC module | INTR = *IRQ* & *INT_EN* |
| **GLOBAL RESET** | Input signal | Global reset from RT TOP module | Clear the whole scaler circuit, some parts of DMA circuit, and status register/WB_START. |

| | | | |
|---|---|---|---|
| **AUTO SWITCH** | R/W register | Write-back address automatic switch enable | Set 1 to enable function, set 0 to disable. |
| **CUR_ADDRx (x=0,1,2)** | Internal register | Channel x current write back start address | |
| **AUTO_GATE_EN** | R/W register | Enable automatic gating clock. | Set 1 to enable function, set 0 to disable. If AUTO_GATE_EN==1, when module idle, the clock to scaler and DMA will be gated. And when a write-back frame starts, the clock gate will be released. |
| **CLK_GATE** | R/W register | Clock gate of scaler and DMA module | If AUTO_GATE_EN==0, set 1 to release clock, set 0 to gating clock. No use when AUTO_GATE_EN==1. |

### 5.11.4 RT-WB Register List

| Module name | Memory Range | Offset Address |
|---|---|---|
| **RT-WB** | 64K | 0x01010000 |

| Register name | Offset | Description |
|---|---|---|
| **WB_GCTRL_REG** | 0x000 | Module general control register |
| **WB_SIZE_REG** | 0x004 | Input size register |
| **WB_CROP_COORD_REG** | 0x008 | Cropping coordinate register |
| **WB_CROP_SIZE_REG** | 0x00c | Cropping size register |
| **WB_A_CH0_ADDR_REG** | 0x010 | Write-back Group A channel 0 address register |
| **WB_A_CH1_ADDR_REG** | 0x014 | Write-back Group A channel 1 address register |
| **WB_A_CH2_ADDR_REG** | 0x018 | Write-back Group A channel 2 address register |
| **WB_A_HIGH_ADDR_REG** | 0x01c | Write-back Group A address high bit register |
| **WB_B_CH0_ADDR_REG** | 0x020 | Write-back Group B channel 0 address register |
| **WB_B_CH1_ADDR_REG** | 0x024 | Write-back Group B channel 1 address register |
| **WB_B_CH2_ADDR_REG** | 0x028 | Write-back Group B channel 2 address register |
| **WB_B_HIGH_ADDR_REG** | 0x02c | Write-back Group B address high bit register |
| **WB_CH0_PITCH_REG** | 0x030 | Write-back channel 0 pitch register |
| **WB_CH12_PITCH_REG** | 0x034 | Write-back channel 1/2 pitch register |
| **WB_ADDR_SWITCH_REG** | 0x040 | Write-back address switch setting register |
| **WB_FORMAT_REG** | 0x044 | Output format register |
| **WB_INT_REG** | 0x048 | Interrupt control register |
| **WB_STATUS_REG** | 0x04c | Module status register |
| | | |
| **WB_BYPASS_REG** | 0x054 | Bypass control register |
| **WB_CS_HORZ_REG** | 0x070 | Coarse scaling horizontal setting register |
| **WB_CS_VERT_REG** | 0x074 | Coarse scaling vertical setting register |
| **WB_FS_INSIZE_REG** | 0x080 | Fine scaling input size register |
| **WB_FS_OUTSIZE_REG** | 0x084 | Fine scaling output size register |
| **WB_FS_HSTEP_REG** | 0x088 | Fine scaling horizontal step register |

| | | |
|---|---|---|
| **WB_FS_VSTEP_REG** | 0x08C | Fine scaling vertical step register |
| **WB_DEBUG_REG** | 0x0FC | Debug register |
| | | |
| **WB_CH0_HCOEF_REGN** | 0x200 + N*4 | Channel 0 horizontal coefficient register N ( N = 0,1,2,...,15) |
| **WB_CH1_HCOEF_REGN** | 0x280 + N*4 | Channel 1/2 horizontal coefficient register N ( N = 0,1,2,...,15) |

**Note: None of these registers is double-buffer.**

## 5.11.5 RT-WB Register Description

### 5.11.5.1 WB_GCTRL_REG

| **Offset: 0X000** | | | **Register name: WB_GCTRL_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31 | R/W | 0x0 | **BIST_EN**<br>Enable BIST mode<br>0: Disable<br>1: Enable |
| 30: | / | / | / |
| 29 | R/W | 0x0 | **CLK_GATE**<br>Clock gate of scaler and DMA module<br>0: Gate<br>1: Release |
| 28 | R/W | 0x0 | **AUTO_GATE_EN**<br>Enable automatic clock gating<br>0: Disable<br>1: Enable |
| 27:18 | / | / | / |
| 17:16 | R/W | 0x0 | **IN_PORT_SEL**<br>Input port selection<br>0: port 0<br>1: port 1<br>2: port 2<br>3: port 3 |
| 15:5 | / | / | / |
| 4 | R/W | 0x0 | SOFT_RESET<br>Reset the whole scaler and DMA.<br>0:Reset release.<br>1:Reset hold. |
| 3:1 | / | / | / |
| 0 | R/W | 0x0 | **WB_START**<br>Start write-back process.<br>0: Do nothing.<br>1: Start. |

### 5.11.5.2 WB_SIZE_REG

| Offset: 0X004 | | | Register name: WB_SIZE_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0x0 | **HEIGHT**<br>Input height<br>The real input height = The value of these bits + 1. |
| 15:13 | / | / | / |
| 12:0 | R/W | 0x0 | **WIDTH**<br>Input width<br>The real input width = The value of these bits + 1. |

### 5.11.5.3 WB_CROP_COORD_REG

| Offset: 0X008 | | | Register name: WB_CROP_COORD_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0x0 | **CROP_TOP**<br>Cropping top position<br>Top position is the left-top y coordinate of input window in pixels |
| 15:13 | / | / | / |
| 12:0 | R/W | 0x0 | **CROP_LEFT**<br>Cropping left position<br>Left position is left-top x coordinate of input window in pixels |

### 5.11.5.4 WB_CROP_SIZE_REG

| Offset: 0X00C | | | Register name: WB_CROP_SIZE_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0x0 | **CROP_HEIGHT**<br>Cropping region height<br>The real cropping region height = the value of these bits + 1. |
| 15:13 | / | / | / |
| 12:0 | R/W | 0x0 | **CROP_WIDTH**<br>Cropping region width<br>The real cropping region width = the value of these bits + 1. |

### 5.11.5.5 WB_A_CH0_ADDR_REG

| Offset: 0X010 | | | Register name: WB_A_CH0_ADDR_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0x0 | **ADDR** |

| | | | |
|---|---|---|---|
| | | | Note: In BYTE.<br>When output format is RGB, **ADDR** must 4 bytes aligning. |

### 5.11.5.6 WB_A_CH1_ADDR_REG

| **Offset**: 0X014 | | | **Register name: WB_A_CH1_ADDR_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0x0 | **ADDR**<br>Note: In BYTE. |

### 5.11.5.7 WB_A_CH2_ADDR_REG

| **Offset**: 0X018 | | | **Register name: WB_A_CH2_ADDR_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0 | **ADDR**<br>Note: In BYTE. |

### 5.11.5.8 WB_A_HIGH_ADDR_REG

| **Offset**: 0X01C | | | **Register name: WB_A_HIGH_ADDR_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | **/** |
| 23:16 | R/W | 0x0 | **CH2_H_ADDR** |
| 15:8 | R/W | 0x0 | **CH1_H_ADDR** |
| 7:0 | R/W | 0x0 | **CH0_H_ADDR** |

### 5.11.5.9 WB_B_CH0_ADDR_REG

| **Offset**: 0X020 | | | **Register name: WB_B_CH0_ADDR_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:00 | R/W | 0x0 | **ADDR**<br>Note: In BYTE.<br>When output format is RGB, **ADDR** must 4 bytes aligning. |

### 5.11.5.10 WB_B_CH1_ADDR_REG

| **Offset**: 0X024 | | | **Register name: WB_B_CH1_ADDR_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0x0 | **ADDR**<br>Note: In BYTE. |

### 5.11.5.11 WB_B_CH2_ADDR_REG

| **Offset**: 0X028 | | | **Register name: WB_B_CH2_ADDR_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0x0 | **ADDR**<br>Note: In BYTE. |

### 5.11.5.12 WB_B_HIGH_ADDR_REG

| Offset: 0X02C | | | Register name: WB_B_HIGH_ADDR_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | **/** |
| 23:16 | R/W | 0x0 | **CH2_H_ADDR** |
| 15:8 | R/W | 0x0 | **CH1_H_ADDR** |
| 7:0 | R/W | 0x0 | **CH0_H_ADDR** |

### 5.11.5.13 WB_CH0_PITCH_REG

| Offset: 0X030 | | | Register name: WB_CH0_PITCH_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0x0 | **PITCH**<br>Write-back channel 0 pitch in BYTE. |

### 5.11.5.14 WB_CH12_PITCH_REG

| Offset: 0X034 | | | Register name: WBC_CH12_PITCH_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0x0 | **PITCH**<br>Write-back channel 1/2 pitch in BYTE. |

### 5.11.5.15 WB_ADDR_SWITCH_REG

| Offset: 0X040 | | | Register name: WB_ADDR_SWITCH_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:21 | / | / | / |
| 20 | R/W | 0x0 | **MANUAL_GROUP**<br>When **AUTO_SWITCH** is 0, set this bit will switch the group.<br>0: Group A<br>1: Group B |
| 19:17 | / | / | / |
| 16 | R/W | 0x0 | **AUTO_SWITCH**<br>Write-back address automatic switch enable<br>0: Disable<br>1: Enable |
| 15:01 | / | / | / |
| 0 | R | 0x0 | **CUR_GROUP**<br>When **AUTO_SWITCH** is 1, this bit will show the index of group using.<br>0: Group A<br>1: Group B |

### 5.11.5.16 WB_FORMAT_REG

| Offset: 0X044 | | | Register name: WB_FORMAT_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |

| | | | |
|---|---|---|---|
| 31:04 | / | / | / |
| 03:0 | R/W | 0x0 | **FORMAT**<br>Output format selection<br>0000: RGB888 (R in high address)<br>0001: BGR888 (B in high address)<br>0010: Reserved<br>0011: Reserved<br>0100: pRGB888(pad equal to 0xff)(p in high address)<br>0101: pBGR888(pad equal to 0xff)(p in high address)<br>0110: BGRp888(pad equal to 0xff)(B in high address)<br>0111: RGBp888(pad equal to 0xff)(R in high address)<br>1000:Planar YUV420<br>1001:Reserved<br>1010:Reserved<br>1011:Reserved<br>1100:Planar YUV420 UV combined ( V1U1V0U0, V1 in high address)<br>1101:Planar YUV420 UV combined ( U1V1U0V0, U1 in high address)<br>1110:Reserved<br>1111:Reserved |

### 5.11.5.17 WB_INT_REG

| **Offset**: 0X048 | | | **Register name: WB_INT_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:1 | / | / | / |
| 0 | R/W | 0x0 | **INT_EN**<br>Write-back interrupt enable<br>0: Disable<br>1: Enable |

### 5.11.5.18 WB_STATUS_REG

| **Offset**: 0X04C | | | **Register name: WB_STATUS_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:09 | / | / | / |
| 8 | R | 0x0 | **BUSY**<br>Write-back process status<br>0: write-back end or write-back disable<br>1: write-back in process |
| 7 | / | / | / |
| 6 | R/W | 0x0 | **TIME_OUT**<br>Write-back TIME_OUT error flag<br>0: No error<br>1: Error |
| 5 | R/W | 0x0 | **OVERFLOW**<br>Write-back FIFO overflow error flag |

| | | | |
|---|---|---|---|
| | | | 0: No error<br>1: Error |
| 4 | R/W | 0x0 | **FINISH**<br>Write-back process finish flag<br>0: write-back not finish or fail<br>1: write-back finished successfully |
| 3:1 | / | / | / |
| 0 | R/W | 0x0 | **IRQ**<br>Write-back process end flag<br>0: write-back not end<br>1: write-back end |

### 5.11.5.19 WB_BYPASS_REG

| **Offset:** 0x054 | | | **Register Name: WB_BYPASS_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:3 | / | / | / |
| 2 | R/W | 0x0 | **FS_EN**<br>Enable Fine Scaling function<br>0: Bypass Fine scaling<br>1: Enable Fine scaling |
| 1 | R/W | 0x0 | **CS_EN**<br>Enable Coarse Scaling function<br>0: Bypass Coarse scaling<br>1: Enable Coarse scaling |
| 0 | R/W | 0x0 | **CSC_EN**<br>Enable RGB to YPbPr color space conversion<br>0: Bypass CSC<br>1: Enable CSC |

### 5.11.5.20 WB_CS_HORZ_REG

| **Offset: 0x070** | | | **Register Name: WB_CS_HORZ_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0x0 | **N** |
| 15:13 | / | / | / |
| 12:0 | R/W | 0x0 | **M** |

### 5.11.5.21 WB_CS_VERT_REG

| **Offset: 0x074** | | | **Register Name: WB_CS_VERT_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0x0 | **N** |

| | | | |
|---|---|---|---|
| 15:13 | / | / | / |
| 12:0 | R/W | 0x0 | **M** |

### 5.11.5.22 WB_FS_INSIZE_REG

| **Offset: 0x080** | | | **Register Name:** WB_FS_INSIZE_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0x0 | **IN_HEIGHT**<br>Channel 0 input height<br>The actual height is register value + 1 |
| 15:13 | / | / | / |
| 12:0 | R/W | 0x0 | **IN_WIDTH**<br>Channel 0 input width<br>The actual width is register value + 1 |

### 5.11.5.23 WB_FS_OUTSIZE_REG

| **Offset: 0x084** | | | **Register Name: WB_FS_OUTSIZE_REG** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0x0 | **OUT_HEIGHT**<br>Channel 0 output height<br>The actual height is register value + 1 |
| 15:13 | / | / | / |
| 12:0 | R/W | 0x0 | **OUT_WIDTH**<br>Channel 0 output width<br>The actual width is register value + 1 |

### 5.11.5.24 WB_FS_HSTEP_REG

| **Offset: 0x088** | | | **Register Name:** WB_FS_HSTEP_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:22 | / | / | / |
| 21:20 | R/W | 0x0 | **HSTEP_INT**<br>The integer part of channel 0 horizontal scale ratio |
| 19:2 | R/W | 0x0 | **HSTEP_FRAC**<br>The fraction part of channel 0 horizontal scale ratio |
| 1:0 | / | / | / |

### 5.11.5.25 WB_FS_VSTEP_REG

| **Offset: 0x08C** | | | **Register Name:** WB_FS_VSTEP_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:22 | / | / | / |

| | | | |
|---|---|---|---|
| 21:20 | R/W | 0x0 | **VSTEP_INT**<br>The integer part of channel 0 vertical scale ratio |
| 19:2 | R/W | 0x0 | **VSTEP_FRAC**<br>The fraction part of channel 0 vertical scale ratio |
| 1:0 | / | / | / |

### 5.11.5.26 WB_DEBUG_REG

| **Offset: 0x0FC** | | | **Register Name:** WB_DEBUG_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:13 | / | / | / |
| 12:0 | R | 0x0 | **INPUT_LINE_CNT**<br>Input line counter |

### 5.11.5.27 WB_CH0_HCOEF_REGN (N = 0:15)

| **Offset: 0x200 + N*4** | | | **Register Name:** WB_CH0_HCOEF_REGN |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | W | UDF | **COEF3**<br>The most right hand-side pixel coefficient |
| 23:16 | W | UDF | **COEF2**<br>The right hand-side pixel coefficient |
| 15:8 | W | UDF | **COEF1**<br>The left hand-side pixel coefficient |
| 7:0 | W | UDF | **COEF0**<br>The most left hand-side pixel coefficient |

**Note: This coefficients are signed. The coefficient value equals to coefficient*$2^6$. N represents the phase.**

### 5.11.5.28 WB_CH1_HCOEF_REGN (N = 0:15)

| **Offset: 0x280 + N*4** | | | **Register Name:** WB_CH1_HCOEF_REGN |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | W | UDF | **COEF3**<br>The most right hand-side pixel coefficient |
| 23:16 | W | UDF | **COEF2**<br>The right hand-side pixel coefficient |
| 15:8 | W | UDF | **COEF1**<br>The left hand-side pixel coefficient |
| 7:0 | W | UDF | **COEF0**<br>The most left hand-side pixel coefficient |

**Note: This coefficients are signed. The coefficient value equals to coefficient*$2^6$. N represents the phase.**

# 5.12 DE VSU Specification

## 5.12.1 Overview

The Video Scaler (VS) provides YUV format image resizing function for display engine. It receives data from overlay module, performs the image resizing function, and outputs to video post-processing modules.
The VS can receive YUV420/YUV422/YUV411 data format, and then converts to YUV444 for display. Horizontal and vertical direction scaling are implemented independently.
The VS features:

- Support YUV420/YUV422/YUV411 data format with 8-bit pre channel
- Support input and output size from 8×4 to 4096×4096
- Support 1/16× to 32× resize ratio
- Support **M**-phase 8-tap horizontal anti-alias filter, **M**-phase 4-tap vertical anti-alias filter (**M** = 16/32 etc)
- Point-to-point display size up to **W** pixels/line. (**W** is RTL programmable)

**Table 5-2. RTL programmable parameters table**

| Parameter name | Default | Possible value | Description |
|---|---|---|---|
| M | 32 | 16/32 | Phase number |
| W | 2048 | 1280/2048/4096 | Y channel line buffer length, U/V channel should divide by 2 |
| FRAC | 19 | 18/19 | Phase adder fraction part bit width |
| VCLIP | 6 | 0~6 integer | Vertical filtering right shift bit width |

## 5.12.2 Block Diagram

Figure 5-10 shows the block diagram of Video Scaler. It is a stream-to-stream module. The input interface which contains independent F/Y/U/V channel, receives data separately from up-stream module. The four channels also have independent resizing path for U/V format conversion. Together with the Y channel, the F channel is a 1-bit data channel that scales using the nearest neighborhood method which data represents the valid data flag generated by overlay module. After resizing to FYUV1444, data outputs in pixel mode to down-stream module. It contains some parts and their function list followed:

- Control logic: status machine control, registers operation, VPHASE selection.
- Resizing: three independent channels line buffer control, horizontal resize, vertical resize.

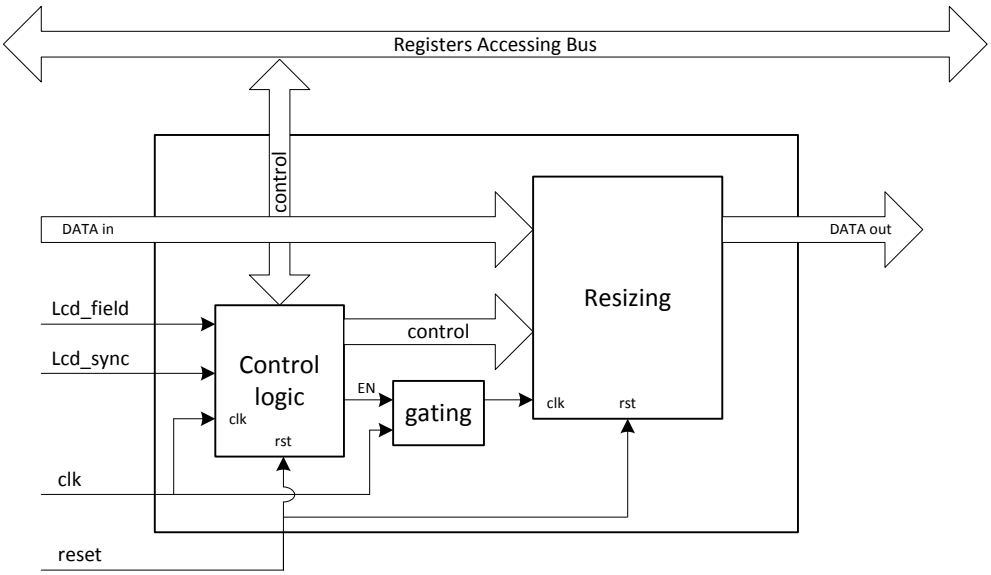



**Figure 5-10. VS block diagram**

## 5.12.3 Video Scaler Register List

| Register name | Offset | Description |
|---|---|---|
| VS_CTRL_REG | 0x000 | Control register |
| VS_STATUS_REG | 0x008 | Status register |
| VS_FIELD_CTRL_REG | 0x00C | Field control register |
| VS_OUT_SIZE_REG | 0x040 | Output size register |
| VS_Y_SIZE_REG | 0x080 | Y channel size register |
| VS_Y_HSTEP_REG | 0x088 | Y channel horizontal step register |
| VS_Y_VSTEP_REG | 0x08C | Y channel vertical step register |
| VS_Y_HPHASE_REG | 0x090 | Y channel horizontal initial phase register |
| VS_Y_VPHASE0_REG | 0x098 | Y channel vertical initial phase 0 register |
| VS_Y_VPHASE1_REG | 0x09C | Y channel vertical initial phase 1 register |
| VS_C_SIZE_REG | 0x0C0 | C channel size register |
| VS_C_HSTEP_REG | 0x0C8 | C channel horizontal step register |
| VS_C_VSTEP_REG | 0x0CC | C channel vertical step register |
| VS_C_HPHASE_REG | 0x0D0 | C channel horizontal initial phase register |
| VS_C_VPHASE0_REG | 0x0D8 | C channel vertical initial phase 0 register |
| VS_C_VPHASE1_REG | 0x0DC | C channel vertical initial phase 1 register |
| VS_Y_HCOEF0_REGN | 0x200+N*4 | Y channel horizontal filter coefficient0 register N (N=0:(M-1)) |
| VS_Y_HCOEF1_REGN | 0x300+N*4 | Y channel horizontal filter coefficient1 register N (N=0:(M-1)) |
| VS_Y_VCOEF_REGN | 0x400+N*4 | Y channel vertical filter coefficient register N (N=0:(M-1)) |
| VS_C_HCOEF0_REGN | 0x600+N*4 | C channel horizontal filter coefficient0 register N (N=0:(M-1)) |
| VS_C_HCOEF1_REGN | 0x700+N*4 | C channel horizontal filter coefficient1 register N (N=0:(M-1)) |
| VS_C_VCOEF_REGN | 0x800+N*4 | C channel vertical filter coefficient register N (N=0:(M-1)) |

**Note:** All registers except some bits in **VS_CTRL_REG**, **VS_FIELD_CTRL_REG**, **VS_STATUS_REG** are double-buffered refreshed by **REG_RDY**.

## 5.12.4 Video Scaler Register Description

### 5.12.4.1 VS_CTRL_REG

**Note: Only bit EN is double-buffered.**

| Offset: 0x000 | | | Register Name: VS_CTRL_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31 | R/W | 0x0 | **BIST_EN**<br>BIST enable<br>0: Disable<br>1: Enable |
| 30 | R/W | 0x0 | **CORE_RST**<br>Core circuit reset<br>0: Do noting<br>1: reset core circuit |
| 29:5 | / | / | / |
| 4 | R/W | 0x0 | **COEF_SWITCH_EN**<br>Coefficients RAM switch<br>0: DONOT switch<br>1: Switch RAM use **REG_RDY**<br>Note: When **LCD SYNC** go low and **COEF_SWITCH_EN** is 1, coefficient RAM will switch to the latest updated RAM if **REG_RDY** is 1, and then the bit will also be self-cleared if switch action successes. |
| 3:1 | / | / | / |
| 0 | R/W | 0x0 | **EN**<br>Video Scaler enable<br>0: Disable<br>1: Enable<br>Note: When module disabled, the core clock to the core circuit will be gated. |

### 5.12.4.2 VS_STATUS_REG

**Note: Whole WORD is non-double-buffered.**

| Offset: 0x008 | | | Register Name: VS_STATUS_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:28 | / | / | / |
| 27:16 | R | 0x0 | **LINE_CNT**<br>Output line number |
| 15:5 | / | / | / |
| 4 | R | 0x0 | **BUSY**<br>Core circuit status<br>0: idle (finish, module disable, waiting for **LCD SYNC** negative edge)<br>1: busy (core circuit calculating) |
| 3:0 | / | / | / |

### 5.12.4.3 VS_FIELD_CTRL_REG

**Note: Only bit FIELD_SEL_VPHASE_EN is double-buffered.**

| Offset: 0x00C | | | Register Name: VS_FIELD_CTRL_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:6 | / | / | / |
| 5 | R/W | 0x0 | **LCD_SYNC_REVERSE**<br>Reverse **LCD SYNC**<br>0: DONOT reverse<br>1: Reverse |
| 4 | R/W | 0x0 | **LCD_FILED_REVERSE**<br>Reverse **LCD FILED**<br>0: DONOT reverse<br>1: Reverse |
| 3:1 | / | / | / |
| 0 | R/W | 0x0 | **FIELD_SEL_VPHASE_EN**<br>Vertical initial phase switch control<br>0: Vertical initial phase fix to phase0<br>1: Switch Vertical initial phase by **LCD FIELD** (Switch to phase0 when **LCD FILED** is 1, and switch to phase1 when **LCD FIELD** is 0) |

### 5.12.4.4 VS_OUT_SIZE_REG

| Offset: 0x040 | | | Register Name: VS_OUT_SIZE_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0x0 | **OUT_HEIGHT**<br>Output height<br>The actual height is register value + 1 |
| 15:13 | / | / | / |
| 12:0 | R/W | 0x0 | **OUT_WIDTH**<br>Output width<br>The actual width is register value + 1 |

### 5.12.4.5 VS_Y_SIZE_REG

| Offset: 0x080 | | | Register Name: VS_Y_SIZE_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0x0 | **Y_HEIGHT**<br>Y channel input height<br>The actual height is register value + 1 |
| 15:13 | / | / | / |
| 12:0 | R/W | 0x0 | **Y_WIDTH**<br>Y channel input width<br>The actual width is register value + 1 |

### 5.12.4.6 VS_Y_HSTEP_REG

| Offset: 0x088 | | | Register Name: VS_Y_HSTEP_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23:20 | R/W | 0x0 | **Y_HSTEP_INT**<br>The integer part of Y channel horizontal scale ratio |
| 19:1 | R/W | 0x0 | **Y_HSTEP_FRAC**<br>The fraction part of Y channel horizontal scale ratio |
| 0 | / | / | / |

### 5.12.4.7 VS_Y_VSTEP_REG

| Offset: 0x08C | | | Register Name: VS_Y_VSTEP_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23:20 | R/W | 0x0 | **Y_VSTEP_INT**<br>The integer part of Y channel vertical scale ratio |
| 19:1 | R/W | 0x0 | **Y_VSTEP_FRAC**<br>The fraction part of Y channel vertical scale ratio |
| 0 | / | / | / |

### 5.12.4.8 VS_Y_HPHASE_REG

| Offset: 0x090 | | | Register Name: VS_Y_HPHASE_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23:20 | R/W | 0x0 | **Y_HPHASE_INT**<br>The integer part of Y channel horizontal initial phase |
| 19:1 | R/W | 0x0 | **Y_HPHASE_FRAC**<br>The fraction part of Y channel horizontal initial phase |
| 0 | / | / | / |

### 5.12.4.9 VS_Y_VPHASE0_REG

| Offset: 0x098 | | | Register Name: VS_Y_VPHASE0_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23:20 | R/W | 0x0 | **Y_VPHASE0_INT**<br>The integer part of Y channel vertical initial phase0 |
| 19:1 | R/W | 0x0 | **Y_VPHASE0_FRAC**<br>The fraction part of Y channel vertical initial phase0 |
| 0 | / | / | / |

### 5.12.4.10 VS_Y_VPHASE1_REG

| Offset: 0x09C | | | Register Name: VS_Y_VPHASE1_REG |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31:24 | / | / | / |
| 23:20 | R/W | 0x0 | **Y_VPHASE1_INT**<br>The integer part of Y channel vertical initial phase1 |
| 19:1 | R/W | 0x0 | **Y_VPHASE1_FRAC**<br>The fraction part of Y channel vertical initial phase1 |
| 0 | / | / | / |

### 5.12.4.11 VS_C_SIZE_REG

| Offset: 0x0C0 | | | Register Name: VS_C_SIZE_REG |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31:29 | / | / | / |
| 28:16 | R/W | 0x0 | **C_HEIGHT**<br>C channel input height<br>The actual height is register value + 1 |
| 15:13 | / | / | / |
| 12:0 | R/W | 0x0 | **C_WIDTH**<br>C channel input width<br>The actual width is register value + 1 |

### 5.12.4.12 VS_C_HSTEP_REG

| Offset: 0x0C8 | | | Register Name: VS_C_HSTEP_REG |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31:24 | / | / | / |
| 23:20 | R/W | 0x0 | **C_HSTEP_INT**<br>The integer part of C channel horizontal scale ratio |
| 19:1 | R/W | 0x0 | **C_HSTEP_FRAC**<br>The fraction part of C channel horizontal scale ratio |
| 0 | / | / | / |

### 5.12.4.13 VS_C_VSTEP_REG

| Offset: 0x0CC | | | Register Name: VS_C_VSTEP_REG |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31:24 | / | / | / |
| 23:20 | R/W | 0x0 | **C_VSTEP_INT**<br>The integer part of C channel vertical scale ratio |
| 19:1 | R/W | 0x0 | **C_VSTEP_FRAC**<br>The fraction part of C channel vertical scale ratio |
| 0 | / | / | / |

### 5.12.4.14 VS_C_HPHASE_REG

| Offset: 0x0D0 | | | Register Name: VS_C_HPHASE_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23:20 | R/W | 0x0 | **C_HPHASE_INT**<br>The integer part of C channel horizontal initial phase |
| 19:1 | R/W | 0x0 | **C_HPHASE_FRAC**<br>The fraction part of C channel horizontal initial phase |
| 0 | / | / | / |

### 5.12.4.15 VS_Y_VPHASE0_REG

| Offset: 0x0D8 | | | Register Name: VS_Y_VPHASE0_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23:20 | R/W | 0x0 | **C_VPHASE0_INT**<br>The integer part of C channel vertical initial phase0 |
| 19:1 | R/W | 0x0 | **C_VPHASE0_FRAC**<br>The fraction part of C channel vertical initial phase0 |
| 0 | / | / | / |

### 5.12.4.16 VS_C_VPHASE1_REG

| Offset: 0x0DC | | | Register Name: VS_C_VPHASE1_REG |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | / | / | / |
| 23:20 | R/W | 0x0 | **C_VPHASE1_INT**<br>The integer part of C channel vertical initial phase1 |
| 19:1 | R/W | 0x0 | **C_VPHASE1_FRAC**<br>The fraction part of C channel vertical initial phase1 |
| 0 | / | / | / |

### 5.12.4.17 VS_Y_HCOEF0_REGN (N = 0:(M-1))

| Offset: 0x200 + N*4 | | | Register Name: VS_Y_HCOEF0_REGN |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | R/W | UDF | **COEF3** |
| 23:16 | R/W | UDF | **COEF2** |
| 15:8 | R/W | UDF | **COEF1** |
| 7:0 | R/W | UDF | **COEF0** |

**Note: HCOEFF0 is a two's complement. The register value equals to coefficient*$2^6$. N represents the phase.**

### 5.12.4.18 VS_Y_HCOEF1_REGN (N = 0:(M-1))

| Offset: 0x300 + N*4 | Register Name: VS_Y_HCOEF1_REGN |
|---|---|

| Bit | Read/Write | Default/Hex | Description |
|---|---|---|---|
| 31:24 | R/W | UDF | **COEF7** |
| 23:16 | R/W | UDF | **COEF6** |
| 15:8 | R/W | UDF | **COEF5** |
| 7:0 | R/W | UDF | **COEF4** |

**Note: HCOEFF1 is a two's complement. The register value equals to coefficient*$2^6$. N represents the phase.**

### 5.12.4.19 VS_Y_VCOEF_REGN (N = 0:(M-1))

| Offset: 0x400 + N*4 | | | Register Name: VS_Y_VCOEF_REGN |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | R/W | UDF | **COEF3** |
| 23:16 | R/W | UDF | **COEF2** |
| 15:8 | R/W | UDF | **COEF1** |
| 7:0 | R/W | UDF | **COEF0** |

**Note: VCOEFF is a two's complement. The register value equals to coefficient*$2^6$. N represents the phase.**

### 5.12.4.20 VS_C_HCOEF0_REGN (N = 0:(M-1))

| Offset: 0x600 + N*4 | | | Register Name: VS_C_HCOEF0_REGN |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | R/W | UDF | **COEF3** |
| 23:16 | R/W | UDF | **COEF2** |
| 15:8 | R/W | UDF | **COEF1** |
| 7:0 | R/W | UDF | **COEF0** |

**Note: HCOEFF0 is a two's complement. The register value equals to coefficient*$2^6$. N represents the phase.**

### 5.12.4.21 VS_C_HCOEF1_REGN (N = 0:(M-1))

| Offset: 0x700 + N*4 | | | Register Name: VS_C_HCOEF1_REGN |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | R/W | UDF | **COEF7** |
| 23:16 | R/W | UDF | **COEF6** |
| 15:8 | R/W | UDF | **COEF5** |
| 7:0 | R/W | UDF | **COEF4** |

**Note: HCOEFF1 is a two's complement. The register value equals to coefficient*$2^6$. N represents the phase.**

### 5.12.4.22 VS_C_VCOEF_REGN (N = 0:(M-1))

| Offset: 0x800 + N*4 | | | Register Name: VS_C_VCOEF_REGN |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:24 | R/W | UDF | **COEF3** |
| 23:16 | R/W | UDF | **COEF2** |
| 15:8 | R/W | UDF | **COEF1** |
| 7:0 | R/W | UDF | **COEF0** |

**Note: VCOEFF is a two's complement. The register value equals to coefficient*$2^6$. N represents the phase.**

# 5.13 DE Rotation Specification

## 5.13.1 Overview

- Support 1/2/3 address data copy
- Support input and output size from 8x8 to 2048x2048
- Support horizontal and vertical flip, clockwise 0/90/180/270 degree rotate
- Support input/output format YUV422/YUV420/ARGB8888/XRGB8888

## 5.13.2 Block Diagram

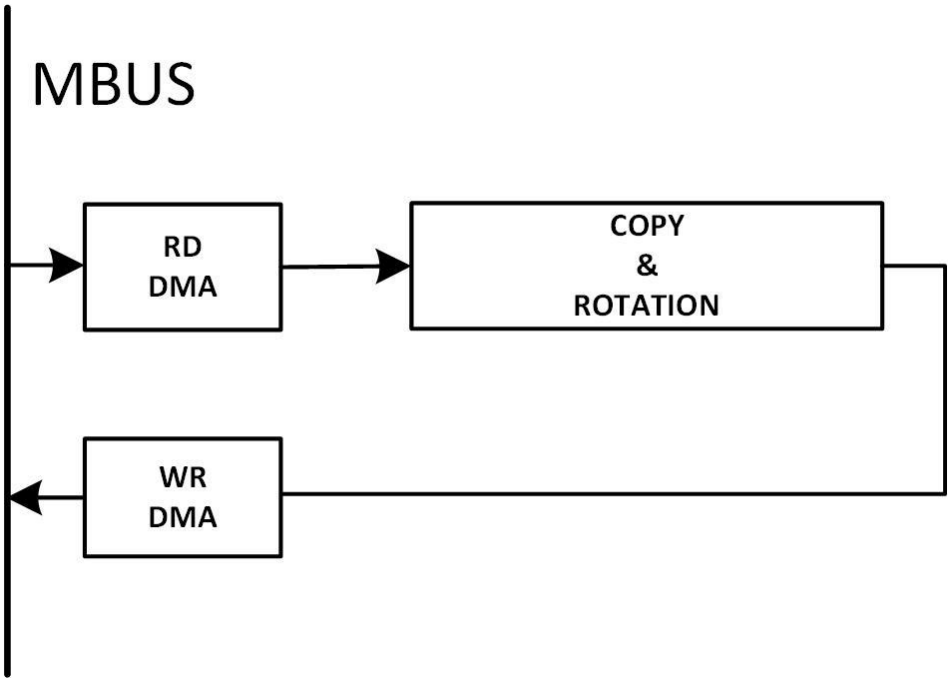



**Figure 5-11. CSR Block Diagram**

## 5.13.3 Description

## 5.13.4 Copy & Rotation

There are several types of rotation: clockwise 0/90/180/270 degree Rotation and H-Flip/V-Flip. Operation of Copy is the same as a 0 degree rotation.

Copy & Rotation will not change RGB color format. But for YUV422/YUV420 input, all output will be in YUV 420 planer.

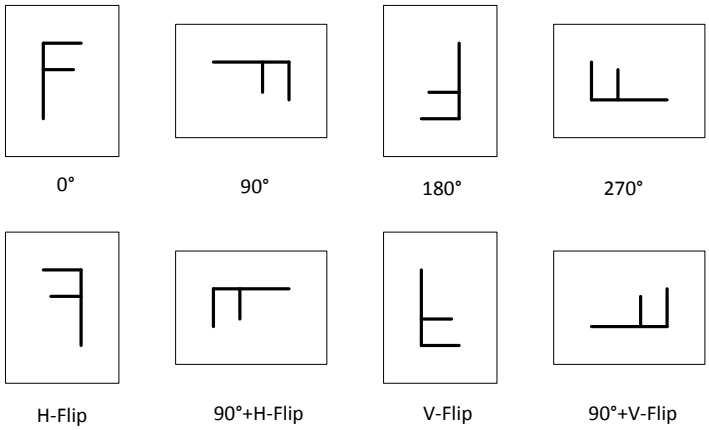



**Figure 5-12. Rotation Diagram**

**Speed requirement**:

For ARGB : 1pixel/1cycle

For RGB565 : 2pixel/1cycle

For YUV420 : 4pixel/1cycle

| Test Format | Rotation Angle | Test Resolution | Boundary Condition | |
|---|---|---|---|---|
| ARGB8888 | 0,90,180,270<br>0+HFlip;<br>90+HFlip;<br>0+VFlip;<br>90+VFlip; | 4x4,<br>4096x4,4x4096<br>2048x1536<br>Typical Resolution | In/Out Pitch: | 16byte Align |
| | | | In/Out Width/Height: | 1pixel Align |
| | | | Input Address: | 16byte Align |
| | | | Output Address: | 4byte Align |
| YUV420 /YUV422 Semi-Planar | | 8x8,<br>4096x8, 8x4096<br>2048x1536<br>Typical Resolution | In/Out Pitch:<br>U Pitch = V Pitch | 16byte Align |
| | | | In/Out Width/Height: | 2pixel Align |
| YUV420 /YUV422 Planar | | | Input Address: | 16byte Align |
| | | | Y Output Address: | 2byte Align |
| | | | UV Output Address: | 1byte Align<br>When the format is Planar, the remainder of U and V base address divided by 32byte must be equal. |
| RGB888 | | 4x4,<br>4096x4,4x4096<br>2048x1536<br>Typical Resolution | In/Out Pitch: | 16byte Align |
| | | | In/Out Width/Height: | 1pixel Align |
| | | | Input Address: | 16byte Align |
| | | | Output Address: | (4*N + 3)byte<br>(4*N + 6)byte<br>(4*N + 9)byte<br>(4*N + 12)byte |

## 5.13.5 Register List

| Register name | Offset | Description |
|---|---|---|
| GLB_CTL | 0x000 | Global control register |

| | | |
|---|---|---|
| INT | 0x004 | Interrupt register |
| IFMT | 0x020 | Input data attribute register |
| IDATA_SIZE | 0x024 | Input data size register |
| IDATA_MEN_PITCH0 | 0x030 | Input Y/RGB/ARGB memory pitch register |
| IDATA_MEN_PITCH1 | 0x034 | Input U/UV memory pitch register |
| IDATA_MEN_PITCH2 | 0x038 | Input V memory pitch register |
| IMEN_LADD0 | 0x040 | Input Y/RGB/ARGB memory address register0 |
| IMEN_HADD0 | 0x044 | Input Y/RGB/ARGB memory address register1 |
| IMEN_LADD1 | 0x048 | Input U/UV memory address register0 |
| IMEN_HADD1 | 0x04C | Input U/UV memory address register1 |
| IMEN_LADD2 | 0x050 | Input V memory address register0 |
| IMEN_HADD2 | 0x054 | Input V memory address register1 |
| ODATA_SIZE | 0x084 | Output data size register |
| ODATA_MEN_PITCH0 | 0x090 | Output Y/RGB/ARGB memory pitch register |
| ODATA_MEN_PITCH1 | 0x094 | Output U/UV memory pitch register |
| ODATA_MEN_PITCH2 | 0x098 | Output V memory pitch register |
| OMEN_LADD0 | 0x0A0 | Output Y/RGB/ARGB memory address register0 |
| OMEN_HADD0 | 0x0A4 | Output Y/RGB/ARGB memory address register1 |
| OMEN_LADD1 | 0x0A8 | Output U/UV memory address register0 |
| OMEN_HADD1 | 0x0AC | Output U/UV memory address register1 |
| OMEN_LADD2 | 0x0B0 | Output V memory address register0 |
| OMEN_HADD2 | 0x0B4 | Output V memory address register1 |

## 5.13.6 Register Description

### 5.13.6.1 Global control register

| **Offset: 0x000** | | | **Register Name: CSR_CTL** |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31 | R/W | 0x0 | **START**<br>0: idle<br>1:start<br>Write 1 to this register bit will start the CSR, when finish, it's self-cleaned by hardware. |
| 30 | R/W | 0x0 | **RESET**<br>0: normal<br>1: assert reset |
| 29:22 | / | / | reserved |
| 21:16 | R/W | 0x3f | Burst length<br>0x07:burst8<br>0x0f:burst16<br>0x1f:burst32<br>0x3f:burst64 |
| 15:8 | / | / | / |
| 7 | R/W | 0x0 | **H-FLIP** |

| | | | |
|---|---|---|---|
| | | | 0: disable<br>1:enable |
| 6 | R/W | 0x0 | **V-FLIP**<br>0: disable<br>1: enable |
| 5:4 | R/W | 0x0 | **Rotation function**<br>00: clockwise 0 degree<br>01: clockwise 90 degree<br>10: clockwise 180 degree<br>11: clockwise 270 degree |
| 3:2 | / | / | reserved |
| 1:0 | R/W | 0x0 | **MODE_SEL**<br>CSR mode select<br>00: disable<br>01: copy& rotation<br>other:reserved |

### 5.13.6.2 Interrupt register

| **Offset: 0x004** | | | **Register Name: INT** |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31:17 | / | / | / |
| 16 | R/W | 0x0 | **FINISH_IRQ_EN**<br>0: disable FINISH IRQ<br>1: enable FINISH IRQ |
| 15:2 | / | / | / |
| 0 | R/W | 0x0 | **FINISH**<br>Mission finish flag<br>It will be set when job is finished, and cleared by writing 1. |

### 5.13.6.3 Input data attribute register

| **Offset: 0x020** | | | **Register Name: IFMT** |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:6 | / | / | / |
| 5:0 | R/W | 0x0 | **IFMT**<br>Input data format<br>0x00: ARGB_8888<br>0x01: ABGR_8888<br>0x02: RGBA_8888<br>0x03: BGRA_8888<br>0x04: XRGB_8888<br>0x05: XBGR_8888<br>0x06: RGBX_8888<br>0x07: BGRX_8888 |

| | | | |
|---|---|---|---|
| | | | 0x08~0x23: Reserved<br>0x24: Planar YUV422 UV combined(V1U1V0U0)<br>0x25: Planar YUV422 UV combined(U1V1U0V0)<br>0x26: Planar YUV422<br>0x27: Reserved<br>0x28: Planar YUV420 UV combined(V1U1V0U0)<br>0x29: Planar YUV420 UV combined(U1V1U0V0)<br>0x2A: Planar YUV420<br>Other: Reserved |

### 5.13.6.4 Input data size register

| Offset: 0x024 | | | Register Name: IDATA_SIZE |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0x0 | **HEIGHT**<br>The Height = The value of these bits add 1 |
| 15:13 | / | / | / |
| 12:0 | R/W | 0x0 | **WIDTH**<br>The Width = The value of these bits add 1 |

### 5.13.6.5 Input Y/RGB/ARGB memory pitch register

| Offset: 0x030 | | | Register Name: IDATA_MEN_PITCH0 |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31:0 | R/W | 0x0 | **PITCH0**<br>Input Y/RGB data memory pitch in bytes<br>Should be 128bit aligned. |

**Note: The setting of this register is Y/RGB channel address.**

### 5.13.6.6 Input U/UV memory pitch register

| Offset: 0x034 | | | Register Name: IDATA_MEN_PITCH 1 |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31:0 | R/W | 0x0 | **PITCH1**<br>Input U/UV data memory pitch in bytes<br>Should be 128bit aligned. |

**Note: The setting of this register is U/UV channel address.**

### 5.13.6.7 Input V memory pitch register

| Offset: 0x038 | | | Register Name: IDATA_MEN_PITCH2 |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31:0 | R/W | 0x0 | **PITCH2**<br>Input V data memory pitch in bytes. |

| | | | |
|---|---|---|---|
| | | | Should be 128bit aligned.<br>Must be equal to input U/UV pitch. |

**Note: The setting of this register is V channel address.**

### 5.13.6.8 Input Y/RGB/ARGB memory address register0

| Offset: 0x040 | | | Register Name: IMEN_LADD0 |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0x0 | **LADD**<br>Memory Block Start Address bit[31:0]<br>Should be 128bit aligned. |

**Note: The setting of this register is Y/RGB channel address.**

### 5.13.6.9 Input Y/RGB/ARGB memory address register1

| Offset: 0x044 | | | Register Name: IMEN_HADD0 |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:8 | / | / | / |
| 7:0 | R/W | 0x0 | **HADD**<br>Memory Block Start Address bit[39:32]<br>Layer Memory Block Address in bytes |

**Note: The setting of this register is Y/RGB channel address.**

### 5.13.6.10 Input U/UV memory address register0

| Offset: 0x048 | | | Register Name: IMEN_LADD1 |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0x0 | **LADD**<br>Memory Block Start Address bit[31:0]<br>Should be 128bit aligned. |

**Note: The setting of this register is U/UV channel address.**

### 5.13.6.11 Input U/UV memory address register1

| Offset: 0x04c | | | Register Name: IMEN_HADD1 |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:8 | / | / | / |
| 7:0 | R/W | 0x0 | **HADD**<br>Memory Block Start Address bit[39:32]<br>Layer Memory Block Address in bytes |

Note: The setting of this register is U/UV channel address.

### 5.13.6.12 Input V memory address register0

| Offset: 0x050 | Register Name: IMEN_LADD2 |
|---|---|

| Bit | Read/Write | Default/Hex | Description |
|---|---|---|---|
| 31:0 | R/W | 0x0 | **LADD**<br>Memory Block Start Address bit[31:0]<br>Should be 128bit aligned. |

**Note: The setting of this register is V channel address.**

### 5.13.6.13 Input V memory address register1

| Offset: 0x054 | | | Register Name: IMEN_HADD2 |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | R/W | 0x0 | **HADD**<br>Memory Block Start Address bit[39:32]<br>Layer Memory Block Address in bytes |

**Note: The setting of this register is V channel address.**

### 5.13.6.14 Output data size register

| Offset: 0x084 | | | Register Name: ODATA_SIZE |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:29 | / | / | / |
| 28:16 | R/W | 0x0 | **HEIGHT**<br>The Height = The value of these bits add 1 |
| 15:13 | / | / | / |
| 12:0 | R/W | 0x0 | **WIDTH**<br>The Width = The value of these bits add 1 |

### 5.13.6.15 Output Y/RGB/ARGB memory pitch register

| Offset: 0x090 | | | Register Name: ODATA_MEN_PITCH0 |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0x0 | **PITCH0**<br>output Y/RGB data memory pitch in bytes<br>Should be 128bit aligned. |

**Note: The setting of this register is Y/RGB channel address.**

### 5.13.6.16 Output U/UV memory pitch register

| Offset: 0x094 | | | Register Name: ODATA_MEN_PITCH 1 |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0x0 | **PITCH1**<br>U/UV data memory pitch in bytes<br>Should be 128bit aligned. |

**Note: The setting of this register is U/UV channel address.**

### 5.13.6.17 Output V memory pitch register

| Offset: 0x098 | | | Register Name: ODATA_MEN_PITCH2 |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31:0 | R/W | 0x0 | **PITCH2**<br>Output V data memory pitch in bytes<br>Should be 128bit aligned.<br>Must be equal to output U/UV pitch. |

**Note: The setting of this register is V channel address.**

### 5.13.6.18 Output Y/RGB/ARGB memory address register0

| Offset: 0x0a0 | | | Register Name: OMEN_LADD0 |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31:0 | R/W | 0x0 | **LADD**<br>Memory Block Start Address bit[31:0] |

**Note: The setting of this register is Y/RGB channel address.**

### 5.13.6.19 Output Y/RGB/ARGB memory address register1

| Offset: 0x0a4 | | | Register Name: OMEN_HADD0 |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | R/W | 0x0 | **HADD**<br>Memory Block Start Address bit[39:32]<br>Layer Memory Block Address in bytes |

**Note: The setting of this register is Y/RGB channel address.**

### 5.13.6.20 Output U/UV memory address register0

| Offset: 0x0a8 | | | Register Name: OMEN_LADD1 |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31:0 | R/W | 0x0 | **LADD**<br>Memory Block Start Address bit[31:0] |

**Note: The setting of this register is U/UV channel address.**

### 5.13.6.21 Output U/UV memory address register1

| Offset: 0x0ac | | | Register Name: OMEN_HADD1 |
|---|---|---|---|
| Bit | Read/Write | Default/Hex | Description |
| 31:8 | / | / | / |
| 7:0 | R/W | 0x0 | **HADD**<br>Memory Block Start Address bit[39:32]<br>Layer Memory Block Address in bytes |

**Note: The setting of this register is U/UV channel address.**

### 5.13.6.22 Output V memory address register0

| Offset: 0x0b0 | | | Register Name: OMEN_LADD2 |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:0 | R/W | 0x0 | **LADD**<br>Memory Block Start Address bit[31:0] |

**Note: The setting of this register is V channel address.**

### 5.13.6.23 Output V memory address register1

| Offset: 0x0b4 | | | Register Name: OMEN_HADD2 |
|---|---|---|---|
| **Bit** | **Read/Write** | **Default/Hex** | **Description** |
| 31:8 | / | / | / |
| 7:0 | R/W | 0x0 | **HADD**<br>Memory Block Start Address bit[39:32]<br>Layer Memory Block Address in bytes |

**Note: The setting of this register is V channel address.**