

***Rockchip  
RK3228  
Technical Reference Manual***

**Revision 0.1  
Oct. 2015**

Rockchip Confidential

## **Revision History**

<b>Date</b>	<b>Revision</b>	<b>Description</b>
2015-10-16	0.1	Initial Release

Rockchip Confidential

## Table of Content

Table of Content .....	3
Figure Index .....	4
Table Index.....	5
Warranty Disclaimer.....	6
Chapter 1 GPU (Graphics Process Unit).....	7
1.1 Overview .....	7
1.2 Block Diagram .....	7
1.3 Register Description.....	8
1.4 Timing Diagram .....	8
1.5 Interface Description .....	9
Chapter 2 Raster Graphic Acceleration (RGA) .....	10
2.1 Overview .....	10
2.2 Block Diagram .....	11
2.3 Function Description .....	12
2.4 Register Description .....	16
2.5 Application Notes .....	41
Chapter 3 Video Output Processor (VOP) .....	42
3.1 Overview .....	42
3.2 Block Diagram .....	43
3.3 Function Description .....	44
3.4 Register Description.....	54
3.5 Timing Diagram .....	125
3.6 Application Notes .....	125
Chapter 4 Multi-format Video Decoder.....	131
4.1 Overview .....	131
4.2 Block Diagram .....	132
4.3 Video frame format .....	132
4.4 Function Description .....	135
4.5 Register Description .....	155
4.6 Interface Description .....	382
4.7 Application Notes .....	382
Chapter 5 Image Enhancement Processor (IEP) .....	384
5.1 Overview .....	384
5.2 Block Diagram .....	385
5.3 Function Description .....	386
5.4 Register Description.....	387
5.5 Application Notes .....	419
Chapter 6 Video Digital Analog Converter (VDAC) .....	421
6.1 Overview .....	421
6.2 Block Diagram .....	421
6.3 Function Description .....	421
6.4 Register Description.....	422
6.5 Application Notes .....	423
Chapter 7 Crypto .....	426
7.1 Overview .....	426
7.2 Block Diagram .....	426
7.3 Register description.....	427
7.4 Application Note.....	476

## Figure Index

Fig. 7-1 GPU block diagram .....	7
Fig. 7-2 GPU interrupt connection .....	9
Fig. 7-3 RGA Block Diagram .....	11
Fig. 7-4 RGA2 in SOC .....	11
Fig. 7-5 RGA Input Data Format .....	12
Fig. 7-6 RGA Dither effect .....	13
Fig. 7-7 layer0 alpha blending calculate flow .....	14
Fig. 7-8 layer1 alpha blending calculate flow .....	15
Fig. 7-9 RGA Gradient Fill .....	15
Fig. 7-10 RGA software main register-region .....	41
Fig. 7-11 VOP Block Diagram .....	44
Fig. 7-12 RGB data format .....	44
Fig. 7-13 YCbCr data format .....	45
Fig. 7-14 BPP little/big endian data format .....	45
Fig. 7-15 VOP Internal DMA .....	45
Fig. 7-16 Virtual display .....	47
Fig. 7-17 X-Mirror and Y-Mirror .....	47
Fig. 7-18 overlay .....	48
Fig. 7-19 overlay timing .....	48
Fig. 7-20 post scaling timing .....	49
Fig. 7-21 Transparency Color Key .....	50
Fig. 7-22 Dither Up .....	50
Fig. 7-23 alpha configuration flow .....	51
Fig. 7-24 Pre-Dither Down .....	54
Fig. 7-25 dsp_out_mode description .....	54
Fig. 7-26 VOP RGB interface timing (SDR) .....	125
Fig. 7-27 VOP RGB mode Programming flow .....	127
Fig. 7-28 RGA command line and command counter .....	129
Fig. 7-29 RGA command sync generation .....	129
Fig. 7-30 the size constraint among A B C .....	130
Fig. 7-31 Decoder in SOC .....	132
Fig. 7-32 VPU Block Diagram .....	132
Fig. 7-33 VCODEC YCbCr 4:2:0 planar format .....	133
Fig. 7-34 VCODEC YCbCr 4:2:0 Semi-planar format .....	133
Fig. 7-35 VCODEC Tile scan mode .....	134
Fig. 7-36 VCODEC YCbCr4:2:2 Interleaved format .....	134
Fig. 7-37 VCODEC AYCbCr 4:4:4 Interleaved format .....	135
Fig. 7-38 VCODEC RGB 16bpp format .....	135
Fig. 7-39 structure of two-level page table .....	136
Fig. 7-40 h264 table .....	140
Fig. 7-41 h264 rps data format .....	142
Fig. 7-42 vp9 prob data format .....	143
Fig. 7-43 vp9 segid data format .....	144
Fig. 7-44 Data format in the DDR of Dec_out_base vp9 .....	144
Fig. 7-45 H264 colmv output format .....	145
Fig. 7-46 VP9 colmv output format .....	145
Fig. 7-47 HEVC colmv output format .....	145
Fig. 7-48 The dataflow of JPEG decoder .....	150
Fig. 7-49 Post-process standalone dataflow .....	152
Fig. 7-50 Post-process Pipe-line Mode Dataflow .....	153
Fig. 7-51 Video Encoder Dataflow .....	155
Fig. 7-52 IEP block diagram .....	385
Fig. 7-53 VDAC Block Diagram .....	421
Fig. 7-54 VDAC Block Diagram .....	424
Fig. 7-1 Crypto Architecture .....	426

## Table Index

Table 7-1 RGA ROP Boolean operations .....	16
Table 7-2 alpha blending mode settings .....	50
Table 7-3 Gather configuration for all format .....	125
Table 7-4 effective immediately register table .....	128
Table 7-5 sps format .....	137
Table 7-6 pps format.....	138
Table 7-7 pps format.....	140
Table 7-8 software rps format.....	142
Table 7-9 hardware rps format .....	142
Table 7-10 error info format .....	145
Table 7-11 normal error table.....	146
Table 7-12 logic error table .....	147
Table 7-13 MPEG-4/H.263 feature .....	148
Table 7-14 MPEG-2/MPEG-1 features .....	148
Table 7-15 VC-1 features .....	149
Table 7-16 JPEG features .....	149
Table 7-17 Post-processor features.....	150
Table 7-18 Requirements for post-processor .....	153
Table 7-19 Post-processor features.....	153
Table 7-20 Video encoder H.264 feature .....	154

## **Warranty Disclaimer**

Rockchip Electronics Co., Ltd makes no warranty, representation or guarantee (expressed, implied, statutory, or otherwise) by or with respect to anything in this document, and shall not be liable for any implied warranties of non-infringement, merchantability or fitness for a particular purpose or for any indirect, special or consequential damages.

Information furnished is believed to be accurate and reliable. However, Rockchip Electronics Co., Ltd assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use.

Rockchip Electronics Co., Ltd's products are not designed, intended, or authorized for using as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Rockchip Electronics Co., Ltd's product could create a situation where personal injury or death may occur, should buyer purchase or use Rockchip Electronics Co., Ltd's products for any such unintended or unauthorized application, buyers shall indemnify and hold Rockchip Electronics Co., Ltd and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, expenses, and reasonable attorney fees arising out of, either directly or indirectly, any claim of personal injury or death that may be associated with such unintended or unauthorized use, even if such claim alleges that Rockchip Electronics Co., Ltd was negligent regarding the design or manufacture of the part.

### **Copyright and Patent Right**

Information in this document is provided solely to enable system and software implementers to use Rockchip Electronics Co., Ltd's products. There are no expressed or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

**Rockchip Electronics Co., Ltd does not convey any license under its patent rights nor the rights of others.**

**All copyright and patent rights referenced in this document belong to their respective owners and shall be subject to corresponding copyright and patent licensing requirements.**

### **Trademarks**

Rockchip and Rockchip™ logo and the name of Rockchip Electronics Co., Ltd's products are trademarks of Rockchip Electronics Co., Ltd. and are exclusively owned by Rockchip Electronics Co., Ltd. References to other companies and their products use trademarks owned by the respective companies and are for reference purpose only.

### **Confidentiality**

The information contained herein (including any attachments) is confidential. The recipient hereby acknowledges the confidentiality of this document, and except for the specific purpose, this document shall not be disclosed to any third party.

### **Reverse engineering or disassembly is prohibited.**

**ROCKCHIP ELECTRONICS CO.,LTD. RESERVES THE RIGHT TO MAKE CHANGES IN ITS PRODUCTS OR PRODUCT SPECIFICATIONS WITH THE INTENT TO IMPROVE FUNCTION OR DESIGN AT ANY TIME AND WITHOUT NOTICE AND IS NOT REQUIRED TO UNDATE THIS DOCUMENTATION TO REFLECT SUCH CHANGES.**

### **Copyright © 2015 Rockchip Electronics Co., Ltd.**

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electric or mechanical, by photocopying, recording, or otherwise, without the prior written consent of Rockchip Electronics Co., Ltd.

## Chapter 1 GPU (Graphics Process Unit)

### 1.1 Overview

The GPU is a hardware accelerator for 2D and 3D graphics systems. Its triangle rate can be 30 Mtris/s, pixel rate can be 300Mpix/s@300MHz.

The GPU supports the following graphics standards:

- OpenGL ES 2.0
- OpenGL ES 1.1
- OpenVG 1.1

The GPU consists of:

- 2 Pixel Processors (PPs)
- 1 geometry Processor (GP)
- 1 Level2 Cache controller (L2)
- 1 Memory Management Unit (MMU) for each GP and PP included in the GPU

The GPU contains a 64-bit APB bus and a 64-bit AXI bus. CPU configures GPU through APB bus, GPU read and write data through AXI bus.

### 1.2 Block Diagram

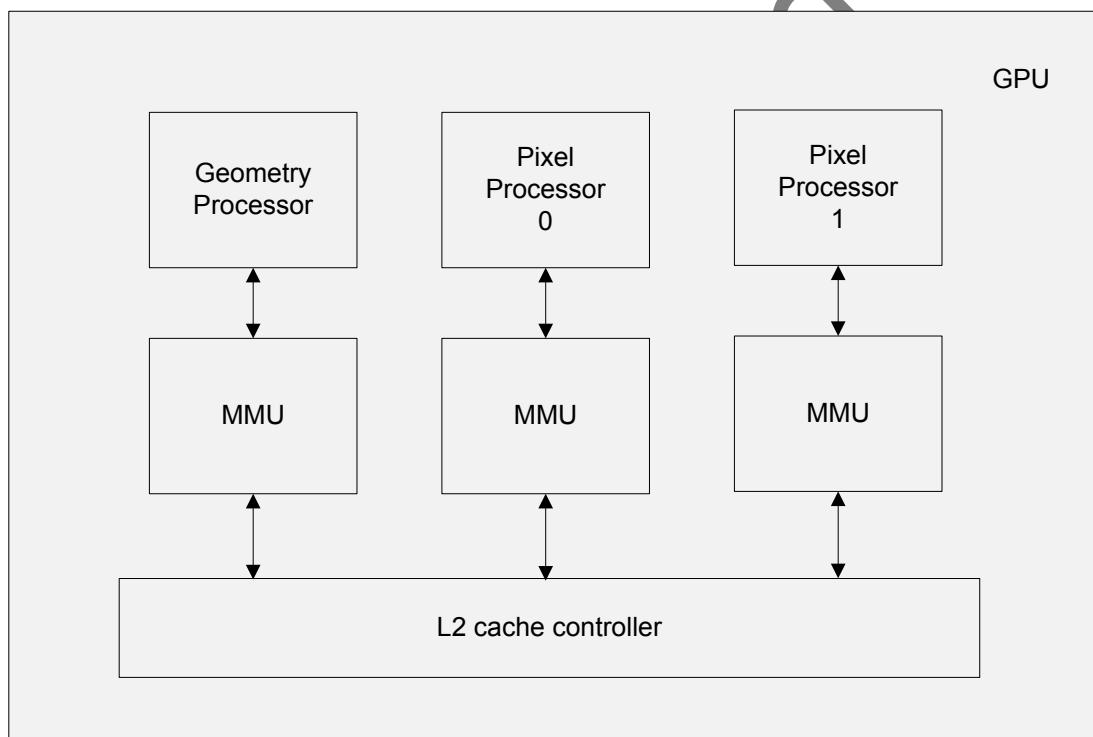


Fig. 7-1 GPU block diagram

The GPU contains 1 geometry processor, 2 pixel processors, 3 MMU and a L2 cache controller. The pixel processor features are:

- Each pixel processor used processes a different tile, enabling a faster turnaround.
- Programmable fragment shader
- Alpha blending
- Complete non-power-of-2 texture support
- Cube mapping
- Fast dynamic branching
- Fast trigonometric functions, including arctangent
- Full floating-point arithmetic
- Framebuffer blend with destination alpha
- Indexable texture samplers
- Line, quad, triangle and point sprites
- No limit on program length

- Perspective correct texturing
- Point sampling, bilinear and trilinear filtering
- Programmable mipmap level-of-detail biasing and replacement
- Stencil buffering, 8-bit
- Two-sided stencil
- Unlimited dependent texture reads
- 4-level hierarchical Z and stencil operations
- Up to 512 times Full scene Anti-Aliasing (FSAA). 4x multisampling times 128x supersampling
- 4-bit per texel compressed texture format

The geometry processor features are:

- Programmable vertex shader
- Flexible input and output formats
- Autonomous operation tile list generation
- Indexed and no-indexed geometry input
- Primitive constructions with points, lines, triangles and quads

The L2 cache controller features are:

- 64KB size
- 4-way set-associative
- Supports up to 32 outstanding AXI transactions
- Implements a standard pseudo-LRU algorithm
- Cache line and line fill burst size is 64 bytes
- Support eight to 64 bytes uncached read bursts and write bursts
- 64-bit interface to memory sub-system
- Support for hit-under-miss and miss-under-miss with the only limitation of AXI ordering rules.

The MMU features are:

- Accesses control registers through the bus infrastructure to configure the memory system
- Each processor has its own MMU to control and translate memory access that the GPU initiates

### 1.3 Register Description

The GPU base address is 0x2000\_0000.

### 1.4 Timing Diagram

The GPU only has a clock input, which is called gpu\_aclk. Gpu\_aclk is generated from the CRU module as shown below

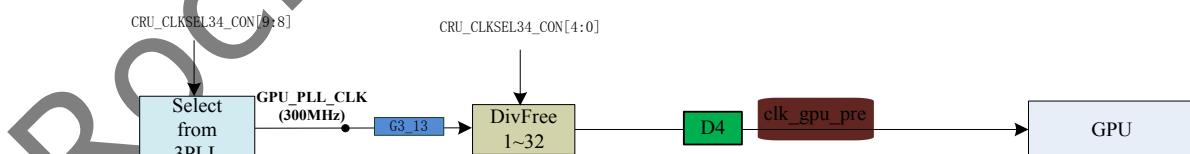


Fig. 7-2 GPU\_ACLK generate block diagram

We can configure CPLL, GPLL and CRU register CRU\_CLKSEL34\_CON to control the gpu\_aclk frequency.

## 1.5 Interface Description

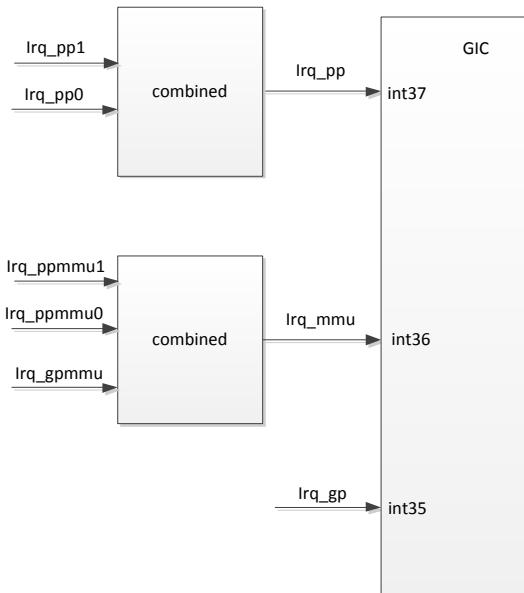


Fig. 7-2GPU interrupt connection

The GPU now has three interrupt output. `irq_ppmmu` and `irq_gpmmu` is combined to `irq_mmu`.

Rockchip Confidential

## Chapter 2 Raster Graphic Acceleration (RGA)

### 2.1 Overview

RGA is a separate 2D raster graphic acceleration unit. It accelerates 2D graphics operations, such as point/line drawing, image scaling, rotation, BitBLT, alpha blending and image blur/sharpness.

RGA supports the following features:

- **Data format**
  - Input data:  
ARGB/RGB888/RGB565/RGB4444/RGB5551/YUV420/YUV422  
Support YUV422SP10bit/YUV420SP10bit
  - Output data:  
ARGB/RGB888/RGB565/RGB4444/RGB5551/YUV420/YUV422
  - Pixel Format conversion, BT.601/BT.709
  - Dither operation
  - Max resolution: 8192x8192 source, 4096x4096 destination
- **Scaling**
  - Down-scaling: Average filter
  - Up-scaling: Bi-cubic filter(source > 2048 would use Bi-linear)
  - Arbitrary non-integer scaling ratio, from 1/8 to 8
- **Rotation**
  - 0, 90, 180, 270 degree rotation
  - x-mirror, y-mirror & rotation operation
- **BitBLT**
  - Block transfer
  - Color palette/Color fill, support with alpha
  - Transparency mode (color keying/stencil test, specified value/value range)
  - Two source BitBLT:
  - A+B=B only BitBLT, A support rotate&scale when B fixed
  - A+B=C second source (B) has same attribute with (C) plus rotation function
- **Alpha Blending**
  - New comprehensive per-pixel alpha(color/alpha channel separately)
  - Fading
- **MMU**
  - 4k/64k page size
  - Four channel: SRC/SRC1/DST/CMD, individual base address and enable control bit
  - TLB pre-fetch

## 2.2 Block Diagram

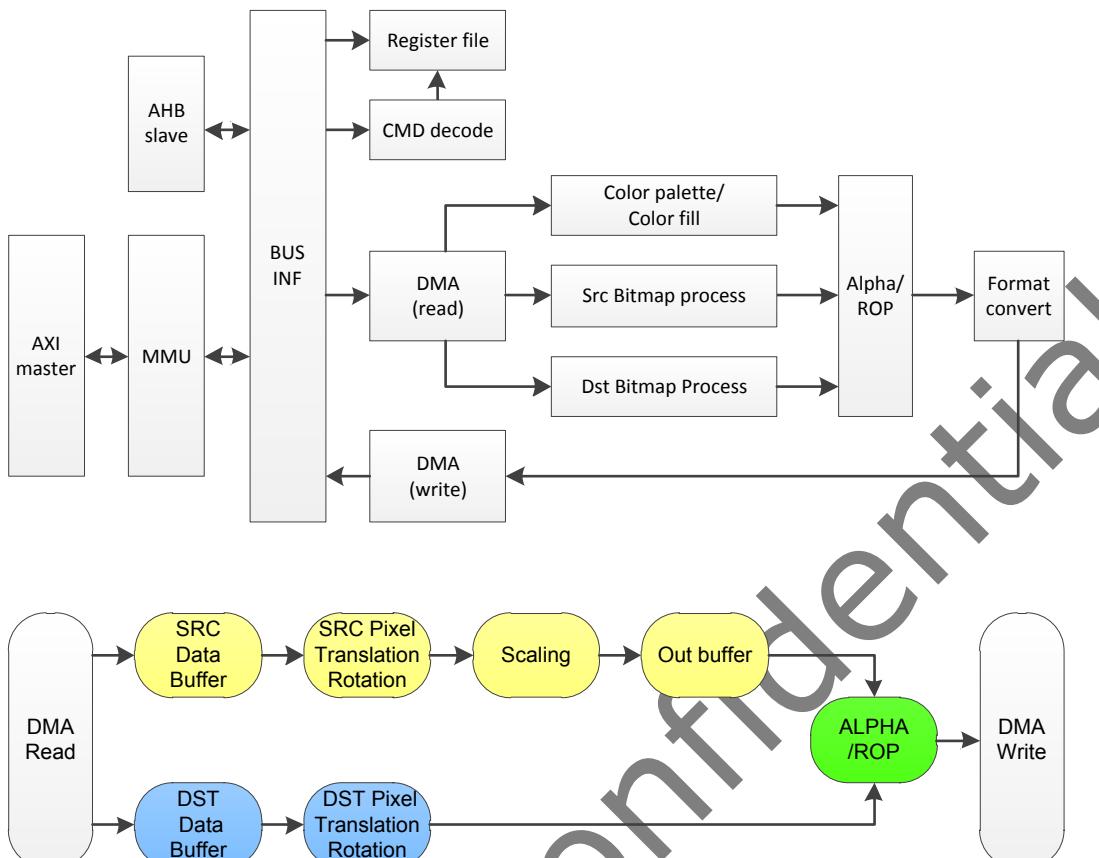


Fig. 7-3 RGA Block Diagram

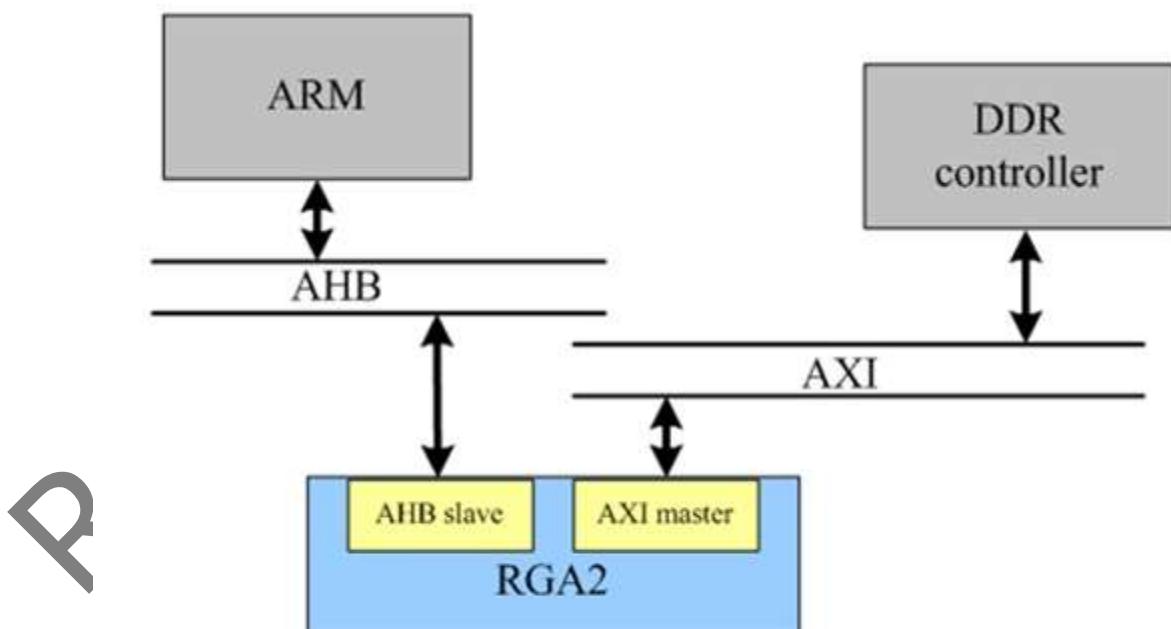


Fig. 7-4 RGA2 in SOC

## 2.3 Function Description

### 2.3.1 Data Format

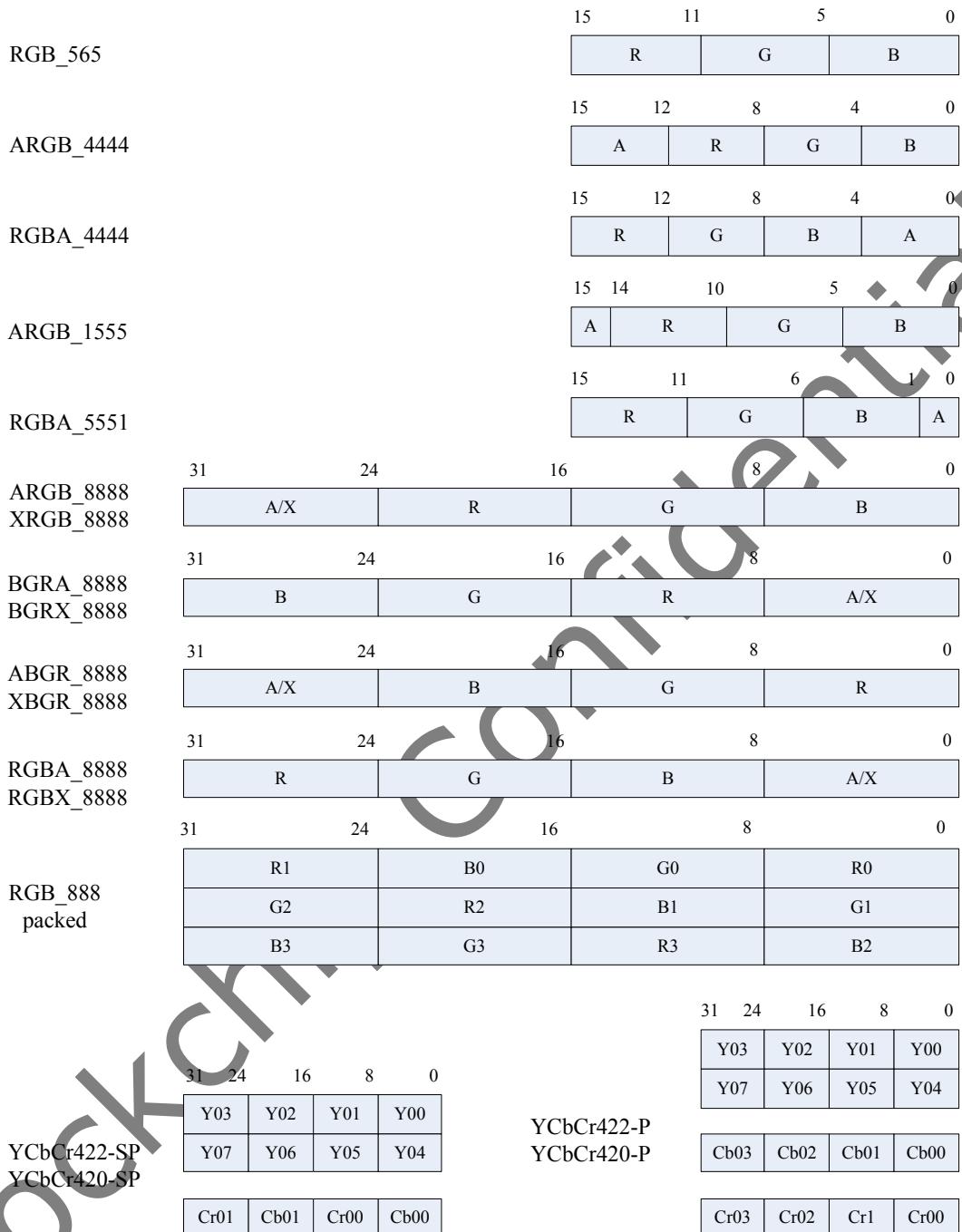


Fig. 7-5 RGA Input Data Format

All input datas (defined by SRC\_IN\_FMT/DST\_IN\_FMT) are converted to ABGR8888. The results are converted to the output data format (defined by DST\_OUT\_FMT).

### 2.3.2 Dithering

There could have dithering operation for source image when the source image format is not RGB565 and the destination format is RGB565.

The down-dithering is done using Dither Allegro.

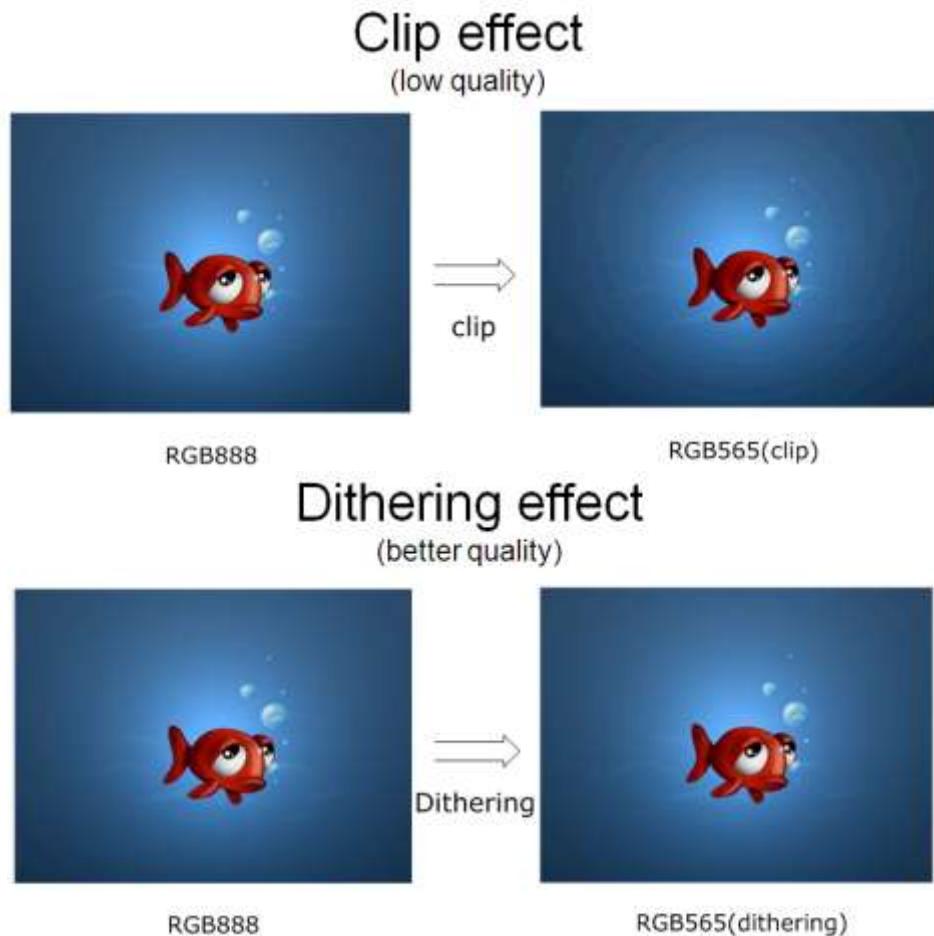


Fig. 7-6 RGA Dither effect

### 2.3.3 Alpha mode

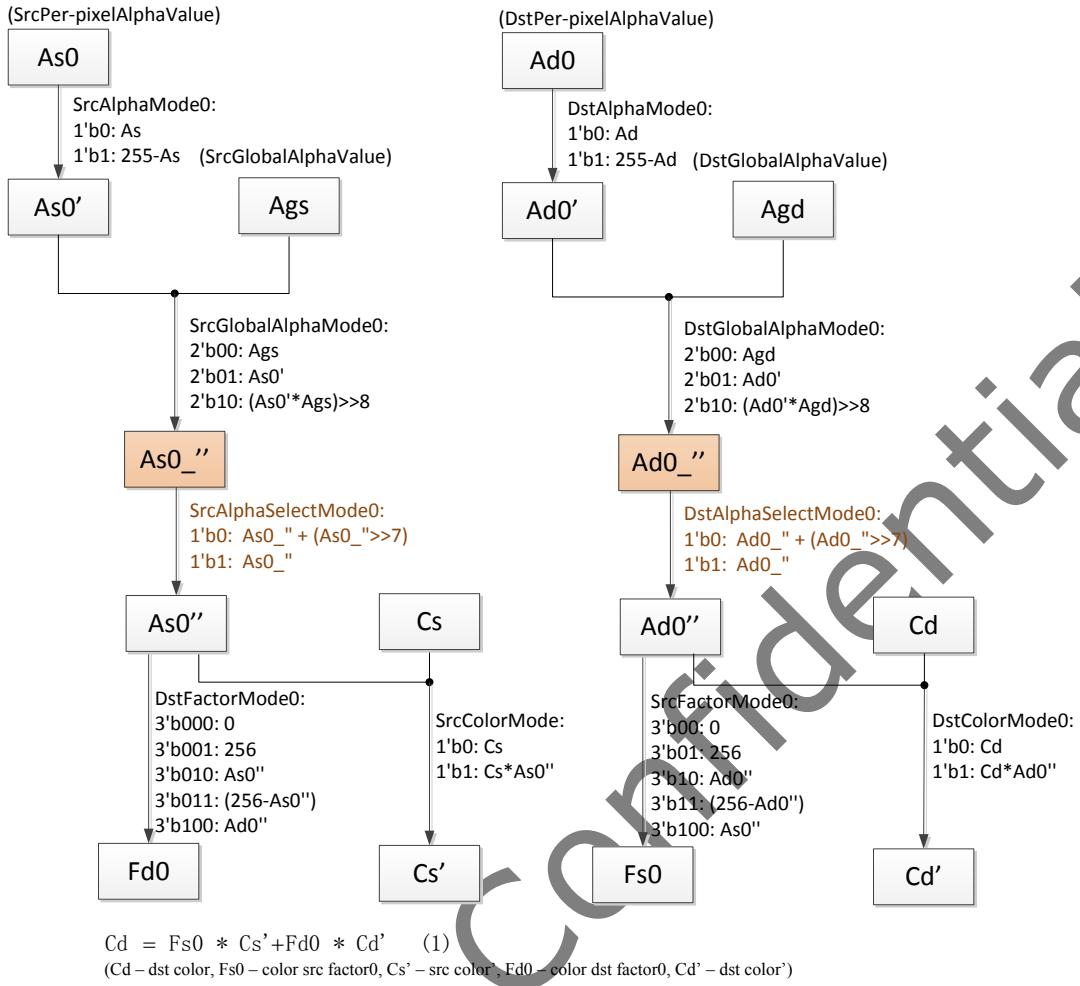


Fig. 7-7 layer0 alpha blending calculate flow

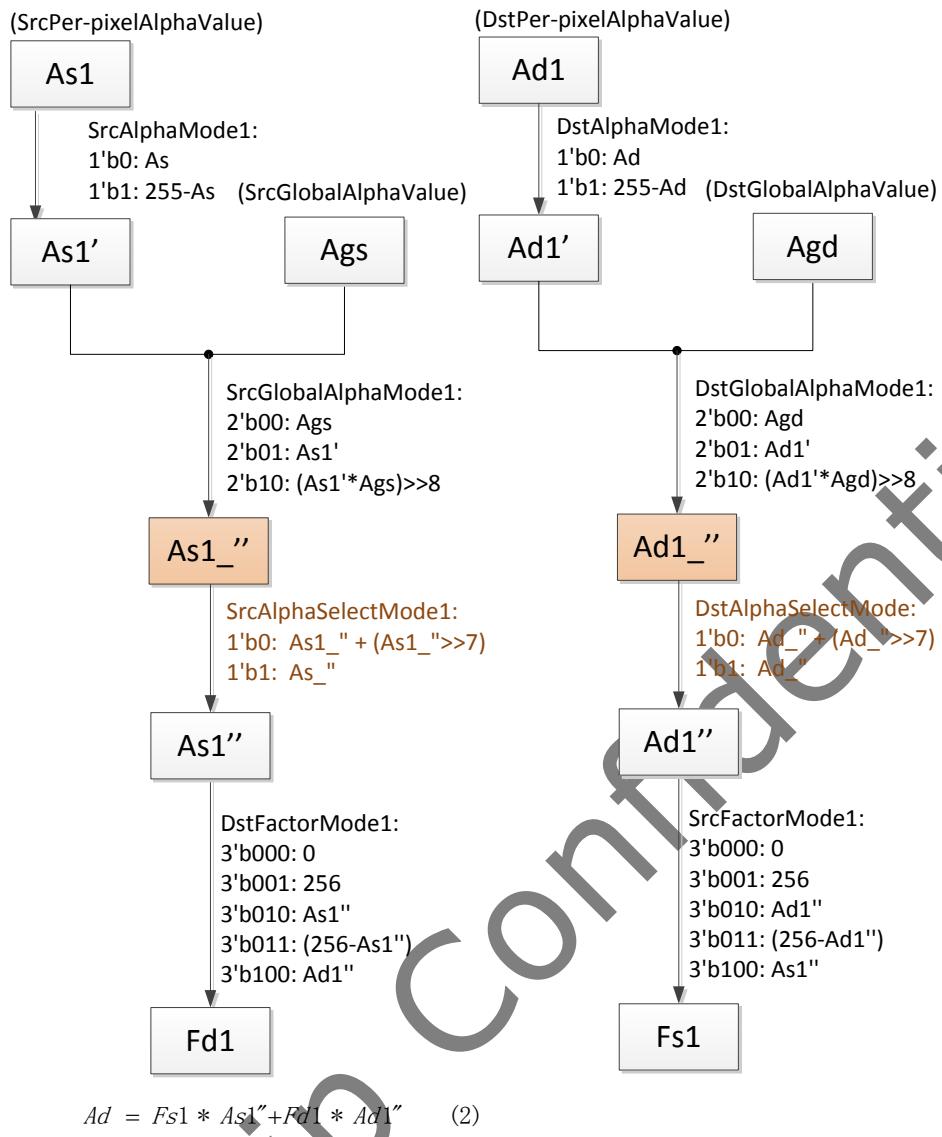


Fig. 7-8 layer1 alpha blending calculate flow

### 2.3.4 Color fill

Two modes of color fill can be done by RGA: solid fill and gradient fill.

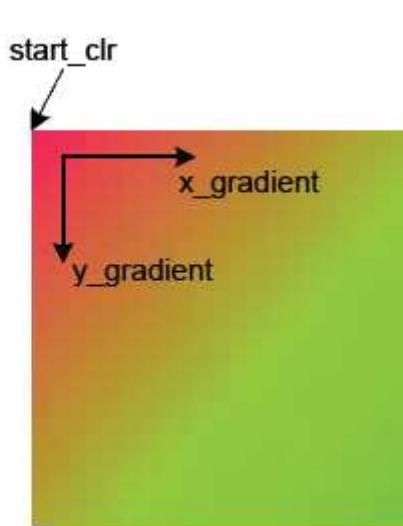


Fig. 7-9 RGA Gradient Fill

Gradient fill using following equations for ARGB calculation of every pixel in different

coordinary.

```
A_cur = (A_start + x*x_A_gradient) +y*y_A_gradient;
R_cur = (R_start + x*x_R_gradient) +y*y_R_gradient;
G_cur = (G_start + x*x_G_gradient) +y*y_G_gradient;
B_cur = (B_start + x*x_B_gradient) +y*y_B_gradient;
```

A\_start, R\_start, G\_start, B\_start is the ARGB value of start point. There are four pairs of values for horizontal and vertical gradient. Saturation operation could be enabled or disabled if the color overflows 255 or underflows 0.

### 2.3.5 Raster Operation (ROP)

Raster operation (ROP) is a Boolean operation between operands, which involve AND, OR, XOR, and NOT operations. For ROP2, operands are P (select pan) and D (Destination bitmap). For ROP3, operands are P (pattern), S (source bitmap) and D (Destination bitmap). For ROP4, operands are P (pattern), S (source bitmap), D (Destination bitmap) and MASK.

Table 7-1 RGA ROP Boolean operations

Operator	Meaning
a	Bitwise AND
n	Bitwise NOT (inverse)
o	Bitwise OR
x	Bitwise exclusive OR (XOR)

### 2.3.6 Scaling

The scaling operation is the image resizing processing of source image. Scaling is done base on ARGB8888 format.

There are three scale modes: scale down (bilinear, Average); scale up(bilinear, Bi-cubic);

## 2.4 Register Description

### 2.4.1 Internal Address Mapping

Slave address can be divided into different length for different usage, which is shown as follows.

### 2.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
RGA2_SYS_CTRL	0x0000	W	0x00000044	RGA system control register
RGA2_CMD_CTRL	0x0004	W	0x00000000	RGA command control register
RGA2_CMD_BASE	0x0008	W	0x12345678	RGA command codes base address register
RGA2_STATUS1	0x000c	W	0x00000000	RGA status register
RGA2_INT	0x0010	W	0x00000000	RGA interrupt register
RGA2_MMU_CTRL0	0x0014	W	0x00000000	RGA MMU control 0 register
RGA2_MMU_CMD_BASE	0x0018	W	0x00000000	Register0000 Abstract
RGA2_STATUS2	0x001c	W	0x00000000	RGA status register
RGA2_WORK_CNT	0x0020	W	0x00000000	work counter
RGA2_VERSION_INFO	0x0028	W	0x00000000	Version number for rga
RGA2_PERF_LATENCY_CTRL0	0x0040	W	0x00000028	Axi performance latency module contrl register0
RGA2_PERF_LATENCY_CTRL1	0x0044	W	0x00000021	PERF_LATENCY_CTRL1
RGA2_PERF_RD_MAX_LATENCY_NUM0	0x0048	W	0x00000000	Read max latency number

Name	Offset	Size	Reset Value	Description
RGA2_PERF_RD_LATENCY_SAMP_NUM	0x004c	W	0x00000000	The number of bigger than configed threshold value
RGA2_PERF_RD_LATENCY_ACC_SUM	0x0050	W	0x00000000	Total sample number
RGA2_PERF_RD_AXI_TOT_AL_BYTE	0x0054	W	0x00000000	perf_rd_axi_total_byte
RGA2_PERF_WR_AXI_TOT_AL_BYTE	0x0058	W	0x00000000	perf_wr_axi_total_byte
RGA2_PERF_WORKING_CNT	0x005c	W	0x00000000	perf_working_cnt
RGA2_MODE_CTRL	0x0100	W	0x00000000	RGA mode control register
RGA2_SRC_INFO	0x0104	W	0x00000000	RGA source information register
RGA2_SRC_BASE0	0x0108	W	0x00000000	source image Y/RGB base address
RGA2_SRC_BASE1	0x010c	W	0x00000000	RGA source image Cb/Cbr base address register
RGA2_SRC_BASE2	0x0110	W	0x00000000	RGA source image Cr base address register
RGA2_SRC_BASE3	0x0114	W	0x00000000	RGA source image 1 base address register
RGA2_SRC_VIR_INFO	0x0118	W	0x00000000	RGA source image virtual stride / RGA source image tile number
RGA2_SRC_ACT_INFO	0x011c	W	0x00000000	RGA source image active width/height register
RGA2_SRC_X_FACTOR	0x0120	W	0x00000000	RGA source image horizontal scaling factor
RGA2_SRC_Y_FACTOR	0x0124	W	0x00000000	RGA source image vertical scaling factor
RGA2_SRC_BG_COLOR	0x0128	W	0x00000000	RGA source image background color
RGA2_SRC_FG_COLOR	0x012c	W	0x00000000	RGA source image foreground color
RGA2_SRC_TR_COLOR0	0x0130	W	0x00000000	RGA source image transparency color min value
RGA2_CP_GR_A	0x0130	W	0x00000000	RGA source image transparency color min value
RGA2_SRC_TR_COLOR1	0x0134	W	0x00000000	Register0000 Abstract
RGA2_CP_GR_B	0x0134	W	0x00000000	RGA source image transparency color max value
RGA2_DST_INFO	0x0138	W	0x00000000	RGA destination format register
RGA2_DST_BASE0	0x013c	W	0x00000000	RGA destination image base address 0 register
RGA2_DST_BASE1	0x0140	W	0x00000000	RGA destination image base address 1 register
RGA2_DST_BASE2	0x0144	W	0x00000000	RGA destination image base address 2 register

Name	Offset	Size	Reset Value	Description
RGA2_DST_VIR_INFO	0x0148	W	0x00000000	RGA destination image virtual width/height register
RGA2_DST_ACT_INFO	0x014c	W	0x00000000	RGA destination image active width/height register
RGA2_ALPHA_CTRL0	0x0150	W	0x00000000	Alpha control register 0
RGA2_ALPHA_CTRL1	0x0154	W	0x00000000	Register0000 Abstract
RGA2_FADING_CTRL	0x0158	W	0x00000000	Fading control register
RGA2_PAT_CON	0x015c	W	0x00000000	Pattern size/offset register
RGA2_ROP_CON0	0x0160	W	0x00000000	ROP code 0 control register
RGA2_CP_GR_G	0x0160	W	0x00000000	RGA color gradient fill step register (color fill mode)
RGA2_ROP_CON1	0x0164	W	0x00000000	ROP code 1 control register
RGA2_CP_GR_R	0x0164	W	0x00000000	RGA color gradient fill step register (color fill mode)
RGA2_MASK_BASE	0x0168	W	0x00000000	RGA mask base address register
RGA2_MMU_CTRL1	0x016c	W	0x00000000	RGA MMU control register 1
RGA2_MMU_SRC_BASE	0x0170	W	0x00000000	RGA source MMU TLB base address
RGA2_MMU_SRC1_BASE	0x0174	W	0x00000000	RGA source1 MMU TLB base address
RGA2_MMU_DST_BASE	0x0178	W	0x00000000	RGA destination MMU TLB base address
RGA2_MMU_ELS_BASE	0x017c	W	0x00000000	RGA ELSE MMU TLB base address

Notes:Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

### 2.4.3 Detail Register Description

#### RGA2\_SYS\_CTRL

Address: Operational Base + offset (0x0000)

RGA system control register

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x0	sw_RST_HANDSAVE_P it would save protect-rstn into initial status if long time dead in protect-rstn status. (auto clear into '0')
6	RW	0x1	sw_RST_PROTECT_E protect-rstn mode enable. it would be ensure all axi write/read operation into completion status when sw_CCLK_SRESET_P or sw_ACLK_SRESET_P valid.
5	RW	0x0	sw_AUTO_RST it would auto-resetn after one frame finish. 0: disable 1: enable
4	RW	0x0	sw_CCLK_SRESET_P RGA core clk domain Soft reset, write '1' to this would reset the RGA engine except config registers.

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
3	WO	0x0	sw_aclk_sreset_p RGA aclk domain Soft reset, write '1' to this would reset the RGA engine except config registers.
2	WO	0x1	sw_auto_ckg RGA auto clock gating enable bit 0: disable 1: enable
1	WO	0x0	sw_cmd_mode RGA command mode 0: slave mode 1: master mode
0	W1C	0x0	sw_cmd_op_st_p RGA operation start bit Only used in passive (slave) control mode

**RGA2\_CMD\_CTRL**

Address: Operational Base + offset (0x0004)

RGA command control register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:13	RO	0x0	reserved
12:3	RW	0x000	sw_cmd_incr_num RGA command increment number
2	WO	0x0	sw_cmd_stop RGA command stop mode Command execution would stop after the current graphic operation finish if set this bit to 1
1	WO	0x0	sw_cmd_incr_valid_p RGA command increment valid (Auto cleared) When setting this bit, 1. The total cmd number would increase by the RGA_INCR_CMD_NUM. 2. RGA would continue running if idle.
0	RW	0x0	sw_cmd_line_st_p RGA command line fetch start (command line reset) (Auto cleared) When fetch start, the total cmd number would reset to RGA_INCR_CMD_NUM.

**RGA2\_CMD\_BASE**

Address: Operational Base + offset (0x0008)

RGA command codes base address register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x12345678	sw_cmd_base RGA command codes base address

**RGA2\_STATUS1**

Address: Operational Base + offset (0x000c)

RGA status register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:20	RO	0x000	sw_cmd_total_num RGA command total number
19:8	RO	0x000	sw_cmd_cur_num RGA command current number
7:1	RO	0x00	Reserved Reserved
0	RO	0x0	sw_rga_sto RGA engine status 0: idle 1: working

**RGA2\_INT**

Address: Operational Base + offset (0x0010)

RGA interrupt register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:11	RO	0x0	reserved
10	RW	0x0	sw_intr_af_e All command finished interrupt enable
9	RW	0x0	sw_intr_mmu_e MMU interrupt enable
8	RW	0x0	sw_intr_err_e Error interrupt enable
7	WO	0x0	sw_intr_cf_clr Current command finished interrupt clear
6	WO	0x0	sw_intr_af_clr All command finished interrupt clear
5	WO	0x0	sw_intr_mmu_clr MMU interrupt clear
4	WO	0x0	sw_intr_err_clr Error interrupt clear
3	RO	0x0	sw_intr_cf Current command finished interrupt flag
2	RO	0x0	sw_intr_af All command finished interrupt flag
1	RO	0x0	sw_intr_mmu MMU interrupt
0	RO	0x0	sw_intr_err Error interrupt flag

**RGA2\_MMU\_CTRL0**

Address: Operational Base + offset (0x0014)

RGA MMU control 0 register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:11	RW	0x0000000	Reserved Reserved
10:9	RW	0x0	sw_els_ch_priority sw_els_ch_priority
8:7	RW	0x0	sw_dst_ch_priority sw_dst_ch_priority
6:5	RW	0x0	sw_src1_ch_priority sw_src1_ch_priority
4:3	RW	0x0	sw_src_ch_priority sw_src_ch_priority
2	RW	0x0	sw_cmd_mmu_flush RGA CMD channel MMU TLB flush: Set 1 to this bit to flush MMU TLB, auto clear
1	RW	0x0	sw_cmd_mmu_en RGA CMD channel MMU enable 0: disable 1: enable
0	RW	0x0	sw_mmu_page_size RGA MMU Page table size 0: 4KB page 1: 64KB page

**RGA2\_MMU\_CMD\_BASE**

Address: Operational Base + offset (0x0018)

Register0000 Abstract

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:28	RO	0x0	reserved
27:0	RW	0x0000000	sw_mmu_cmd_base RGA command MMU TLB base address (word)

**RGA2\_STATUS2**

Address: Operational Base + offset (0x001c)

RGA status register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:13	RO	0x00000	Reserved
12:11	RO	0x0	rpp_mkram_rready rpp_mkram_rready
10:6	RO	0x00	dstrpp_outbuf_rready dstrpp_outbuf_rready
5:2	RO	0x0	srcrpp_outbuf_rready dstrpp_outbuf_rready
1	RO	0x0	bus_error

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
0	RO	0x0	rpp_error

**RGA2\_WORK\_CNT**

Address: Operational Base + offset (0x0020)

work counter

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:27	RW	0x00	Reserved
26:0	RO	0x0000000	sw_work_cnt working counter register RGA total working counter

**RGA2\_VERSION\_INFO**

Address: Operational Base + offset (0x0028)

Version number for rga

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RW	0x00	major IP major version used for IP structure version information
23:20	RW	0x0	minor minor version big feature change under same structure
19:0	RW	0x00000	svnbuid rtl current svn number

**RGA2\_PERF\_LATENCY\_CTRL0**

Address: Operational Base + offset (0x0040)

Axi performance latency module control register0

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:20	RO	0x0	reserved
19:8	RW	0x000	sw_rd_latency_thr sw_rd_latency_thr
7:4	RW	0x2	sw_rd_latency_id sw_rd_latency_id
3	RW	0x0	sw_axi_cnt_type sw_axi_cnt_type sw_axi_cnt_type
2	RW	0x1	sw_axi_perf_frm_type latency mode 1'b0: clear by software configuration 1'b1: clear by frame end

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
1	RW	0x0	sw_axi_perf_clr_e sw_axi_perf_clr_e 1'b0: software clear disable 1'b1: software clear enable
0	RW	0x0	sw_axi_perf_work_e sw_axi_perf_work_e 1'b0: disable 1'b1: enable

**RGA2\_PERF\_LATENCY\_CTRL1**

Address: Operational Base + offset (0x0044)

PERF\_LATENCY\_CTRL1

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:12	RO	0x0	reserved
11:8	RW	0x0	sw_aw_count_id sw_aw_count_id sw_aw_count_id
7:4	RW	0x2	sw_ar_count_id sw_ar_count_id sw_ar_count_id
3	RW	0x0	sw_aw_cnt_id_type sw_aw_cnt_id_type sw_aw_cnt_id_type
2	RW	0x0	sw_ar_cnt_id_type sw_ar_cnt_id_type sw_ar_cnt_id_type
1:0	RW	0x1	sw_addr_align_type sw_addr_align_type sw_addr_align_type

**RGA2\_PERF\_RD\_MAX\_LATENCY\_NUM0**

Address: Operational Base + offset (0x0048)

Read max latency number

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RO	0x0	reserved
15:0	RO	0x0000	rd_max_latency_num_ch0 read max latency value of channel 0

**RGA2\_PERF\_RD\_LATENCY\_SAMP\_NUM**

Address: Operational Base + offset (0x004c)

The number of bigger than configed threshold value

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
------------	-------------	--------------------	--------------------

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x00000000	rd_latency_thr_num_ch0 read latency thr number channel 0

**RGA2\_PERF\_RD\_LATENCY\_ACC\_SUM**

Address: Operational Base + offset (0x0050)

Total sample number

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x00000000	rd_latency_acc_sum

**RGA2\_PERF\_RD\_AXI\_TOTAL\_BYTE**

Address: Operational Base + offset (0x0054)

perf\_rd\_axi\_total\_byte

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	perf_rd_axi_total_byte perf_rd_axi_total_byte perf_rd_axi_total_byte

**RGA2\_PERF\_WR\_AXI\_TOTAL\_BYTE**

Address: Operational Base + offset (0x0058)

perf\_wr\_axi\_total\_byte

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	perf_wr_axi_total_byte perf_wr_axi_total_byte perf_wr_axi_total_byte

**RGA2\_PERF\_WORKING\_CNT**

Address: Operational Base + offset (0x005c)

perf\_working\_cnt

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	perf_working_cnt perf_working_cnt perf_working_cnt

**RGA2\_MODE\_CTRL**

Address: Operational Base + offset (0x0100)

RGA mode control register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:8	RW	0x0000000	Reserved Reserved
7	RW	0x0	sw_intr_cf_e Current command finished interrupt enable

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
6	RW	0x0	sw_gradient_sat Gradient saturation calculation mode 0:clip 1:not-clip
5	RW	0x0	sw_alpha_zero_key ARGB888 alpha zero key mode 0x000000 would be changed to 0x000100(RGB888)/0x0020(RGB565)for ARGB888 to RGBX/RGB565 color key 0: disable 1: enable
4	RW	0x0	sw_cf_rop4_pat Color fill/ROP4 pattern 0: solid color 1: pattern color
3	RW	0x0	sw_bb_mode Bitblt mode 0: SRC + DST => DST 1: SRC + SRC1 => DST
2:0	RW	0x0	sw_render_mode RGA 2D render mode 000: Bitblt 001: Color palette 010: Rectangle fill 011: Update palette LUT/pattern ram

**RGA2\_SRC\_INFO**

Address: Operational Base + offset (0x0104)

RGA source information register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RO	0x0	reserved
28	RW	0x0	sw_src_yuv10_round_e this bit valid when RGA support yuv 10bit picture input 0: yuv 10bit to 8bit round disable 1: yuv 10bit to 8bit round enable
27	RW	0x0	sw_src_yuv10_e this bit valid when RGA support yuv 10bit picture input 0: yuv 10bit disable 1: yuv 10bit enable
26	RW	0x0	sw_vsp_mode 0:by-cubic 1:bi-linear

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
25:24	RW	0x0	sw_bic_coe_sel SRC bicubic scaling coefficient select 00: CATROM 01: MITCHELL 10: HERMITE 11: B-SPLINE
23	RW	0x0	sw_src_dither_up SRC dither up enable 0:disable 1:enable
22:19	RW	0x0	sw_src_trans_e Source transparency enable bits [3]: A value stencil test enable bit [2]: B value stencil test enable bit [1]: G value stencil test enable bit [0]: R value stencil test enable bit
18	RW	0x0	sw_src_trans_mode Source transparency mode 0: normal stencil test (color key) 1: inverted stencil test
17:16	RW	0x0	sw_src_vscl_mode SRC vertical scaling mode 00: no scaling 01: down-scaling 10: up-scaling
15:14	RW	0x0	sw_src_hscl_mode SRC horizontal scaling mode 00: no scaling 01: down-scaling 10: up-scaling
13:12	RW	0x0	sw_src_mir_mode SRC mirror mode 00: no mirror 01: x mirror 10: y mirror 11: x mirror + y mirror
11:10	RW	0x0	sw_src_rot_mode SRC rotation mode 00: 0 degree 01: 90 degree 10: 180 degree 11: 270 degree

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
9:8	RW	0x0	sw_src_csc_mode Source bitmap YUV2RGB conversion mode 00: bypass 01: BT.601-range0 10: BT.601-range1 11: BT.709-range0
7	RW	0x0	sw_cp_endian Source Color palette endian swap 0: big endian 1: little endian
6	RW	0x0	sw_src_uvsleep Source Cb-Cr swap 0: CrCb 1: CbCr
5	RW	0x0	sw_src_alpha_swap Source bitmap data alpha swap 0: ABGR 1: BGRA
4	RW	0x0	sw_src_rbswap Source bitmap data RB swap 0: BGR 1: RGB
3:0	RW	0x0	sw_src_fmt Source bitmap data format 0000: ABGR888 0001: XBGR888 0010: BGR packed 0100: RGB565 0101: ARGB1555 0110: ARGB4444 1000: YUV422SP 1001: YUV422P 1010: YUV420SP 1011: YUV420P 1100: 1BPP (color palette) 1101: 2BPP (color palette) 1110: 4BPP (color palette) 1111: 8BPP (color palette)

**RGA2\_SRC\_BASE0**

Address: Operational Base + offset (0x0108)  
source image Y/RGB base address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_src_base0 source image Y/RGB base address

**RGA2\_SRC\_BASE1**

Address: Operational Base + offset (0x010c)  
 RGA source image Cb/Cbr base address register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_src_base1 source image Cb base address (YUV422/420-P) source image Cb/Cr base address (YUV422/420-SP)

**RGA2\_SRC\_BASE2**

Address: Operational Base + offset (0x0110)  
 RGA source image Cr base address register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:1	RO	0x0	reserved
0	RW	0x0	sw_src_base2 source image Cr base address (YUV422/420-P)

**RGA2\_SRC\_BASE3**

Address: Operational Base + offset (0x0114)  
 RGA source image 1 base address register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:1	RO	0x0	reserved
0	RW	0x0	sw_src_base3 source image 1 RGB base address (source bitblt mode1)

**RGA2\_SRC\_VIR\_INFO**

Address: Operational Base + offset (0x0118)  
 RGA source image virtual stride / RGA source image tile number

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:26	RW	0x00	Reserved
25:16	RW	0x000	sw_mask_vir_stride mask image virtual stride (words)
15	RW	0x0	Reserved Reserved
14:0	RW	0x0000	sw_src_vir_stride src image virtual stride (words)

**RGA2\_SRC\_ACT\_INFO**

Address: Operational Base + offset (0x011c)  
 RGA source image active width/height register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RW	0x0	Reserved2 Reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
28:16	RW	0x0000	sw_src_act_height source image active height
15:13	RW	0x0	Reserved1 Reserved
12:0	RW	0x0000	sw_src_act_width source image active width

**RGA2\_SRC\_X\_FACTOR**

Address: Operational Base + offset (0x0120)

RGA source image horizontal scaling factor

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_src_hsp_factor Source image horizontal up-scaling factor $=(DST\_ACT\_WIDTH/SRC\_ACT\_WIDTH) * 65536$
15:0	RW	0x0000	sw_src_hsd_factor Source image horizontal down-scaling factor $=(SRC\_ACT\_WIDTH/DST\_ACT\_WIDTH) * 65536$

**RGA2\_SRC\_Y\_FACTOR**

Address: Operational Base + offset (0x0124)

RGA source image vertical scaling factor

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_src_vsp_factor Source image vertical up-scaling factor $(DST\_ACT\_HEIGHT/SRC\_ACT\_HEIGHT) * 65536$
15:0	RW	0x0000	sw_src_vsd_factor Source image vertical down-scaling factor $(SRC\_ACT\_HEIGHT/DST\_ACT\_HEIGHT) * 65536$

**RGA2\_SRC\_BG\_COLOR**

Address: Operational Base + offset (0x0128)

RGA source image background color

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_src_bg_color Source image background color ("0" bit color for mono expansion.)

**RGA2\_SRC\_FG\_COLOR**

Address: Operational Base + offset (0x012c)

RGA source image foreground color

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_src_fg_color Source image foreground color Source image foreground color ("1" bit color for mono expansion.) Color fill color, Pan color

**RGA2\_SRC\_TR\_COLOR0**

Address: Operational Base + offset (0x0130)

RGA source image transparency color min value

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RW	0x00	sw_src_trans_amin source image transparency color A min value
23:16	RW	0x00	sw_src_trans_bmin source image transparency color B min value
15:8	RW	0x00	sw_src_trans_gmin source image transparency color G min value
7:0	RW	0x00	sw_src_trans_rmin source image transparency color R min value

**RGA2\_CP\_GR\_A**

Address: Operational Base + offset (0x0130)

RGA source image transparency color min value

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_gradient_y_a Y gradient value of Alpha (signed 8.8)
15:0	RW	0x0000	sw_gradient_x_a X gradient value of Alpha (signed 8.8)

**RGA2\_SRC\_TR\_COLOR1**

Address: Operational Base + offset (0x0134)

Register0000 Abstract

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RW	0x00	sw_src_trans_amax source image transparency color A max value
23:16	RW	0x00	sw_src_trans_bmax source image transparency color B max value
15:8	RW	0x00	sw_src_trans_gmax source image transparency color G max value
7:0	RW	0x00	sw_src_trans_rmax source image transparency color R max value

**RGA2\_CP\_GR\_B**

Address: Operational Base + offset (0x0134)

RGA source image transparency color max value

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
------------	-------------	--------------------	--------------------

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_gradient_y_b Y gradient value of Blue (signed 8.8)
15:0	RW	0x0000	sw_gradient_x_b X gradient value of Blue (signed 8.8)

**RGA2\_DST\_INFO**

Address: Operational Base + offset (0x0138)

RGA destination format register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:19	RW	0x0000	Reserved Reserved
18	RW	0x0	sw_dst_csc_clip BGR2YUV Clip mode(from 0~255 clip to 36~235) 1: clip enable; 0: unclip
17:16	RW	0x0	sw_dst_csc_mode sw_dst_csc_mode sw_dst_csc_mode DST bitmap RGB2YUV conversion mode 00: Bypass 01: BT.601-range0 10: BT.601-range1 11: BT.709-range0
15:14	RW	0x0	sw_dither_mode sw_dither_mode sw_dither_mode DST dither down bit mode 00: 888 to 666 01: 888 to 565 10: 888 to 555 11: 888 to 444
13	RW	0x0	sw_dither_down sw_dither_down sw_dither_down DST dither down enable 0:disable 1:enable
12	RW	0x0	sw_src1_dither_up sw_src1_dither_up sw_src1_dither_up DST/SRC1 dither up enable 0:disable 1:enable

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
11	RW	0x0	<p>sw_src1_alpha_swap sw_src1_alpha_swap sw_src1_alpha_swap Source 1 bitmap data alpha swap 0: ABGR 1: BGRA</p>
10	RW	0x0	<p>sw_src1_rbswap sw_src1_rbswap sw_src1_rbswap Source 1 bitmap data RB swap 0: BGR 1: RGB</p>
9:7	RW	0x0	<p>sw_src1_fmt Source 1 bitmap data format sw_src1_fmt Source 1 bitmap data format 000: ABGR888 001: XBGR888 010: BGR packed 100: RGB565 101: ARGB1555 110: ARGB4444</p>
6	RW	0x0	<p>sw_dst_uvswap Destination Cb-Cr swap Destination Cb-Cr swap 0: CrCb 1: CbCr</p>
5	RW	0x0	<p>sw_dst_alpha_swap Destination bitmap data alpha swap Destination bitmap data alpha swap 0: ABGR 1: BGRA</p>
4	RW	0x0	<p>sw_dst_rbswap Destination bitmap data RB swap Destination bitmap data RB swap 0: BGR 1: RGB</p>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
3:0	RW	0x0	sw_dst_fmt Destination bitmap data format Destination bitmap data format 0000: ABGR888 0001: XBGR888 0010: BGR packed 0100: RGB565 0101: ARGB1555 0110: ARGB4444 1000: YUV422SP 1001: YUV422P 1010: YUV420SP 1011: YUV420P If RGA has yuyv output format feature: 1100: YVYU422(U, LSB) 1101: YVYU420(U, LSB) 1110: VYUY422(Y, LSB) 1111: VYUY420(Y, LSB)

**RGA2\_DST\_BASE0**

Address: Operational Base + offset (0x013c)  
 RGA destination image base address 0 register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_dst_base0 destination image Y/RGB base address

**RGA2\_DST\_BASE1**

Address: Operational Base + offset (0x0140)  
 RGA destination image base address 1 register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_dst_base1 destination image Cb/CbCr base address

**RGA2\_DST\_BASE2**

Address: Operational Base + offset (0x0144)  
 RGA destination image base address 2 register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_dst_base2 destination image Cr base address

**RGA2\_DST\_VIR\_INFO**

Address: Operational Base + offset (0x0148)  
 RGA destination image virtual width/height register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:28	RW	0x0	Reserved2 Reserved
27:16	RW	0x000	sw_src1_vir_stride source image 1 virtual stride (words)
15:12	RW	0x0	Reserved1 Reserved
11:0	RW	0x000	sw_dst_vir_stride destination image virtual stride(words)

**RGA2\_DST\_ACT\_INFO**

Address: Operational Base + offset (0x014c)

RGA destination image active width/height register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:28	RW	0x0	Reserved2 Reserved
27:16	RW	0x000	sw_dst_act_height Destination image active height
15:12	RW	0x0	Reserved1 Reserved
11:0	RW	0x000	sw_dst_act_width Destination image active width

**RGA2\_ALPHA\_CTRL0**

Address: Operational Base + offset (0x0150)

Alpha control register 0

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:21	RW	0x000	Reserved Reserved
20	RW	0x0	sw_mask_endian sw_mask_endian ROP4 mask endian swap 0: big endian 1: little endian
19:12	RW	0x00	sw_dst_global_alpha global alpha value of DST(Agd)
11:4	RW	0x00	sw_src_global_alpha global alpha value of SRC(Ags) fading value in fading mod
3:2	RW	0x0	sw_rop_mode ROP mode select 00: ROP 2 01: ROP 3 10: ROP 4

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
1	RW	0x0	sw_alpha_rop_sel Alpha or ROP select 0: alpha 1: ROP
0	RW	0x0	sw_alpha_rop_e Alpha or ROP enable 0: disable 1: enable

**RGA2\_ALPHA\_CTRL1**

Address: Operational Base + offset (0x0154)

Register0000 Abstract

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RW	0x0	Reserved Reserved
29	RW	0x0	sw_src_alpha_m1 Src Transparent/opaque of alpha channel (As1') 0: As 1: 255-As
28	RW	0x0	sw_dst_alpha_m1 Dst Transparent/opaque of alpha channel (Ad1') 0: Ad 1: 255-Ad
27:26	RW	0x0	sw_src_blend_m1 Alpha src blend mode select of alpha channel (As1_") 00: Ags 01: As1' 10: (As1'*Ags)>>8 11: reserved
25:24	RW	0x0	sw_dst_blend_m1 Alpha dst blend mode select of alpha channel(Ad1_") 00: Agd 01: Ad1' 10: (Ad1'*Agd)>>8 11: reserved
23	RW	0x0	sw_src_alpha_cal_m1 Alpha src calculate mode of alpha channel(As1_") 0: As1_''= As1_"" + (As1_"">>7) 1: As1_''= As1_""
22	RW	0x0	sw_dst_alpha_cal_m1 Alpha dst calculate mode of alpha channel(Ad1_") 0: Ad1_''= Ad1_"" + (Ad1_"">>7) 1: Ad1_''= Ad1_""

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
21:19	RW	0x0	w_src_factor_m1 Src factor mode of alpha channel(Fs1) 000: 0 001: 256 010: Ad1'' 011: 256-Ad1'' 100: As1''
18:16	RW	0x0	sw_dst_factor_m1 Dst factor mode of alpha channel(Fd1) 000: 0 001: 256 010: As1'' 011: 256-As1'' 100: Ad1''
15	RW	0x0	sw_src_alpha_m0 Src Transparent/opaque of color channel (As0') 0: As 1: 255-As
14	RW	0x0	sw_dst_alpha_m0 Dst Transparent/opaque of color channel (Ad0') 0: Ad 1: 255-Ad
13:12	RW	0x0	sw_src_blend_m0 Alpha src blend mode select of color channel (As0_") 00: Ags 01: As0' 10: (As0'*Agd)>>8 11: reserved
11:10	RW	0x0	sw_dst_blend_m0 Alpha dst blend mode select of color channel(Ad0_") 00: Agd 01: Ad0' 10: (Ad0'*Agd)>>8 11: reserved
9	RW	0x0	sw_src_alpha_cal_m0 Alpha src calculate mode of color channel(As0'') 0: As0''= As0_'' + (As0_>>7) 1: As0''= As0_''
8	RW	0x0	sw_dst_alpha_cal_m0 Alpha dst calculate mode of color channel(Ad0'') 0: Ad0''= Ad0_'' + (Ad0_>>7) 1: Ad0''= Ad0_''

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
7:5	RW	0x0	sw_src_factor_m0 Src factor mode of color channel(Fs0) 000: 0 001: 256 010: Ad0'' 011: 256-Ad0'' 100: As0''
4:2	RW	0x0	sw_dst_factor_m0 Dst factor mode of color channel(Fd0) 000: 0 001: 256 010: As0'' 011: 256-As0'' 100: Ad0''
1	RW	0x0	sw_src_color_m0 SRC color select(Cs') 0: Cs 1: Cs * As0''
0	RW	0x0	sw_dst_color_m0 SRC color select(Cd') 0: Cd 1: Cd * Ad0''

**RGA2\_FADING\_CTRL**

Address: Operational Base + offset (0x0158)

Fading control register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:25	RW	0x00	Reserved Reserved
24	RW	0x0	sw_fading_en Fading enable
23:16	RW	0x00	sw_fading_offset_b Fading offset B value
15:8	RW	0x00	sw_fading_offset_g Fading offset G value (Pattern total number when pattern loading)
7:0	RW	0x00	sw_fading_offset_r Fading offset R value (Start point of pattern ram in pattern mode)

**RGA2\_PAT\_CON**

Address: Operational Base + offset (0x015c)

Pattern size/offset register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RW	0x00	sw_pat_offset_y Pattern y offset
23:16	RW	0x00	sw_pat_offset_x Pattern x offset
15:8	RW	0x00	sw_pat_height Pattern height
7:0	RW	0x00	sw_pat_width Pattern width

**RGA2\_ROP\_CON0**

Address: Operational Base + offset (0x0160)

ROP code 0 control register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:25	RW	0x00	Reserved Reserved
24:0	RW	0x00000000	sw_rop3_code0 Rop3 code 0 control bits

**RGA2\_CP\_GR\_G**

Address: Operational Base + offset (0x0160)

RGA color gradient fill step register (color fill mode)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_gradient_y_g Y gradient value of Green (signed 8.8)
15:0	RW	0x0000	sw_gradient_x_g X gradient value of Green (signed 8.8)

**RGA2\_ROP\_CON1**

Address: Operational Base + offset (0x0164)

ROP code 1 control register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:25	RW	0x00	Reserved Reserved
24:0	RW	0x00000000	sw_rop3_code1 Rop3 code 1 control bits

**RGA2\_CP\_GR\_R**

Address: Operational Base + offset (0x0164)

RGA color gradient fill step register (color fill mode)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_gradient_y_r Y gradient value of Red(signed 8.8)
15:0	RW	0x0000	sw_gradient_x_r X gradient value of Red(signed 8.8)

**RGA2\_MASK\_BASE**

Address: Operational Base + offset (0x0168)

RGA mask base address register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_mask_base mask base address in ROP4 mode LUT/ pattern load base address

**RGA2\_MMU\_CTRL1**

Address: Operational Base + offset (0x016c)

RGA MMU control register 1

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:14	RW	0x00000	Reserved Reserved Reserved
13	RW	0x0	sw_els_mmu_flush RGA ELSE channel MMU TLB flush: Set 1 to this bit to flush MMU TLB, auto clear
12	RW	0x0	sw_els_mmu_en RGA ELSE channel MMU enable 0: disable 1: enable
11	RW	0x0	sw_dst_mmu_prefetch_dir sw_dst_mmu_prefetch_dir 0:forward 1:backward
10	RW	0x0	sw_dst_mmu_prefetch_en sw_dst_mmu_prefetch_en 0:disable 1:enable
9	RW	0x0	sw_dst_mmu_flush sw_dst_mmu_flush RGA DST channel MMU TLB flush: Set 1 to this bit to flush MMU TLB, auto clear
8	RW	0x0	sw_dst_mmu_en sw_dst_mmu_en RGA DST channel MMU enable 0: disable 1: enable
7	RW	0x0	sw_src1_mmu_prefetch_dir sw_src1_mmu_prefetch_dir 0:forward 1:backward

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
6	RW	0x0	sw_src1_mmu_prefetch_en sw_src1_mmu_prefetch_en 0:disable 1:enable
5	RW	0x0	sw_src1_mmu_flush sw_src1_mmu_flush RGA SRC1 channel MMU TLB flush: Set 1 to this bit to flush MMU TLB, auto clear
4	RW	0x0	sw_src1_mmu_en sw_src1_mmu_en RGA SRC1 channel MMU enable 0: disable 1: enable
3	RW	0x0	sw_src_mmu_prefetch_dir sw_src_mmu_prefetch_dir 0:forward 1:backward
2	RW	0x0	sw_src_mmu_prefetch_en sw_src_mmu_prefetch_en 0:disable 1:enable
1	RW	0x0	sw_src_mmu_flush RGA SRC channel MMU TLB flush: Set 1 to this bit to flush MMU TLB, auto clear
0	RW	0x0	sw_src_mmu_en RGA SRC channel MMU enable 0: disable 1: enable

**RGA2\_MMU\_SRC\_BASE**

Address: Operational Base + offset (0x0170)

RGA source MMU TLB base address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:28	RO	0x0	reserved
27:0	RW	0x00000000	sw_mmu_src_base RGA source MMU TLB base address (128-bit)

**RGA2\_MMU\_SRC1\_BASE**

Address: Operational Base + offset (0x0174)

RGA source1 MMU TLB base address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:28	RO	0x0	reserved
27:0	RW	0x00000000	sw_mmu_src1_base RGA source1 MMU TLB base address (128-bit)

**RGA2\_MMU\_DST\_BASE**

Address: Operational Base + offset (0x0178)

RGA destination MMU TLB base address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:28	RO	0x0	reserved
27:0	RW	0x00000000	sw_mmu_dst_base RGA destination MMU TLB base address (128-bit)

**RGA2\_MMU\_ELS\_BASE**

Address: Operational Base + offset (0x017C)

RGA ELSE MMU TLB base address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:28	RO	0x0	reserved
27:0	RW	0x00000000	sw_mmu_els_base RGA destination MMU TLB base address (128-bit)

## 2.5 Application Notes

### 2.5.1 Register Partition

There are two types of register in RGA. The first 8 registers (0x0 - 0x1C) are general registers for system configuration including command mode, command parameter, RGA status, general interrupts. The other registers (from 0x100) are command registers for command codes.

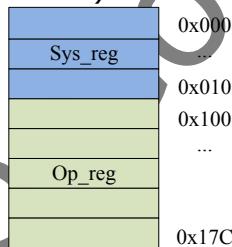


Fig. 7-10 RGA software main register-region

### 2.5.2 Command Modes

RGA has two command modes: slave mode and master mode. In slave mode (RGA\_SYS\_CTRL[1] = 1'b0), 2D graphic command only could be run one by one. CPU set all the command registers in RGA and then start RGA running by setting RGA\_SYS\_CTRL[0] to '1'. In master mode (RGA\_SYS\_CTRL[1] = 1'b1), 2D graphic commands could be run sequentially. After setting command's number to RGA\_CMD\_CTRL[12:3], writing '1' to RGA\_CMD\_CTRL[0] will start the command fetch, then Internal command DMA fetch commands from external command line.

Command line is a collection of several command codes with continuous address. At the first start, the command start address (RGA\_CMD\_ADDR) and command number (RGA\_CMD\_CTRL[12:3]) should be set, then write '1' to cmd\_line\_st (RGA\_CMD\_CTRL[0]) to start the command line fetch. Incremental command is supported by setting cmd\_incr\_num (RGA\_CMD\_CTRL[12:3]) and cmd\_incr\_valid (RGA\_CMD\_CTRL[1]=1'b1)

## Chapter 3 Video Output Processor (VOP)

### 3.1 Overview

VOP is the display interface from memory frame buffer to display device (HDMI and TV set). VOP is connected to an AHB bus through an AHB slave and AXI bus through an AXI master. The register setting is configured through the AHB slave interface and the display frame data is read through the AXI master interface.

#### 3.1.1 Features

- Display interface
- CVBS Interface
  - ◆ 10bit TVE output for VDAC
  - ◆ support PAL/NTSC
- HDMI interface
  - ◆ Support HDMI 2.0 output up to 4K
  - ◆ Support RGB/YCbCr420/YCbCr444 8/10bit.
- Support max output resolution 4K for HDMI, 480/576i for CVBS
- Support configurable polarity of DCLK/HSYNC/VSYNC/DEN
- Display process
- BCSH,10bit
- ◆ Brightness,Contrast,Saturation,Hue adjustment
  - Support display data swap
- ◆ BG swap, RB swap, RG swap, dummy swap
  - Support three YUV2RGB transition modes:
- ◆ YCbCr2RGB (rec601-mpeg/rec601-jpeg/rec709-hd/bt2020)
  - Support two RGB2YUV transition modes:
- ◆ RGB2YCbCr(BT601/BT709/BT2020)
  - Blank/black display
  - standby mode
  - X-MIRROR,Y-MIRROR for win0/win1/hwc
  - scale down for TV overscan
- ◆ horizontal scale down using bilinear, 0.5~1.0
- ◆ vertical scale down using bilinear, 0.5~1.0
  - Layer process
  - Background layer
  - ◆ programmable 30-bit color
  - Win0/Win1 layer
  - ◆ Support data format
    - ◇ RGB888, ARGB888, RGB565,
    - ◇ YCbCr420SP, YCbCr422SP, YCbCr444SP both 8bit and 10bits
  - ◆ Support virtual display
    - ◆ Support 1/8 to 8 scaling-down and scaling-up engine
    - ◇ scale up using bicubic and bilinear
    - ◇ scale down using bilinear and average
    - ◇ per-pix alpha + scale
  - ◆ Support data swap
    - ◇ RGB/BPP: rb\_swap
    - ◇ YUV: mid\_swap,uv\_swap
  - ◆ transparency color key,prior to alpha blending and fading
  - ◆ Support fading/alpha blending
  - ◆ Support interlace output
  - Hardware Cursor layer
  - ◆ Support data format
    - ◇ RGB888, ARGB888, RGB565

- ❖ 8BPP
- ❖ little endian and big endian for BPP
- ❖ BYPASS and LUT mode(32bit LUT, 8bit AA+8bit-RGB)for BPP
- ◆ Support four hwc size: 32x32,64x64,96x96,128x128
- ◆ Support fading/alpha blending
- ◆ Support displaying out of panel,right or bottom
- ◆ Support interlace read and interlace output
- Overlay
  - ◆ support RGB and YUV domain overlay
  - ◆ Support 4 layers,background/win0/win1/hwc
  - ◆ Win0/Win1 overlay position exchangeable
  - ◆ Alpha blending
  - ❖ Support 12 alpha blending modes
  - ❖ Support pre-multiplied alpha
  - ❖ Support global alpha and per\_pix alpha
  - ❖ Support 256 level alpha
- Bus interface
  - Support AMBA 2.0 AHB slave interface for accessing internal registers and LUT memories, 32bit data bus width
  - Support AMBA 3.0 AXI master read interface for loading frame data
  - ◆ 128bit data bus width
  - Support MMU
  - DMA line mode for YUV
  - Support NOC hurry for higher bus priority for win0/win1
  - Support DMA stop mode
  - Win AXI read ID configurable
  - Max read outstanding number
    - ◆ 32 when MMU disable
    - ◆ 31 when MMU enable
- Interrupt
  - One combined interrupt
  - ◆ high active
  - ◆ raw status readable
  - ◆ combinational with interrupt sources

### 3.2 Block Diagram

The architecture is shown in the following figure.

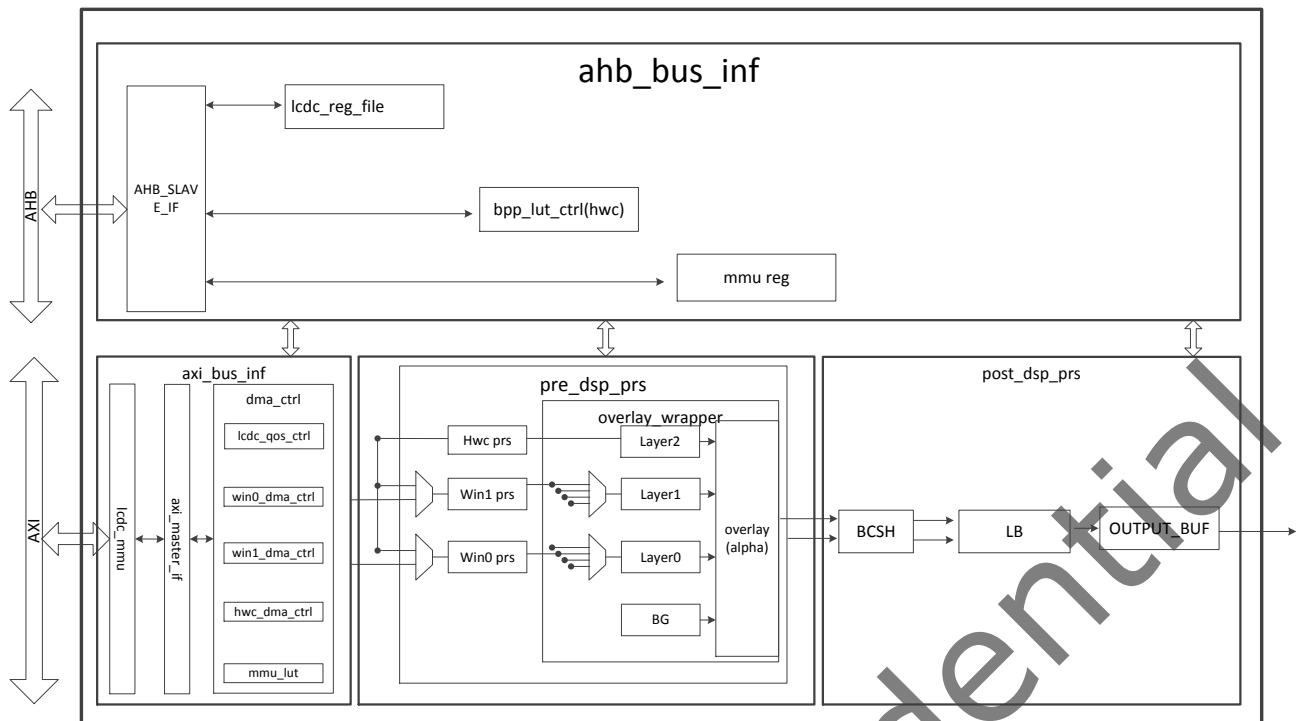


Fig. 7-11 VOP Block Diagram

### 3.3 Function Description

#### 3.3.1 Pixel format

##### 1.RGB

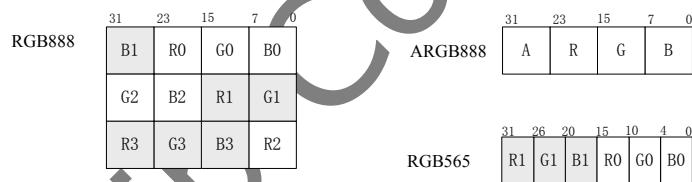


Fig. 7-12 RGB data format

##### 2.YCbCr(8bit)

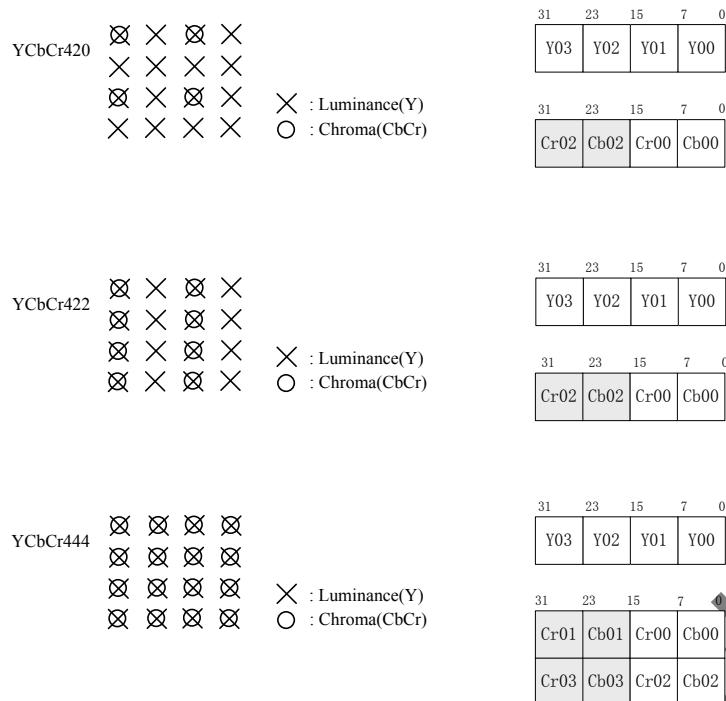


Fig. 7-13 YCbCr data format

YCbCr just support SP

YCbCr-8bit 32bit align

## 2.YCbCr(10bit)

YCbCr-10bit is the same as YCbCr-8bit except it is 640bit align(64 pixels alignn).

### 3.BPP

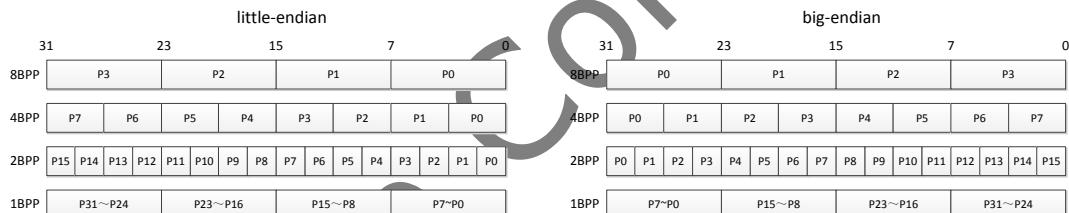


Fig. 7-14 BPP little/big endian data format

### 3.3.2 Pixel Data Path

There are only one data input path for VOP to get display layers' pixel data: internal DMA.

#### 1.Internal DMA

Internal DMA can fetch the pixel data through AXI bus from system memory (DDR) for all the display layers. Data fetching is driven by display output requirement.

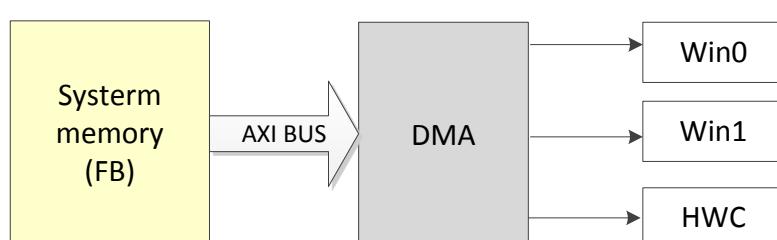


Fig. 7-15 VOP Internal DMA

### 3.3.3 Win Scaling

The scaling operation is the image resizing process by scaling-up or scaling-down the source image from active window size to display window size for displaying on LCD panel or TV set. Horizontal scaling and vertical scaling are realized independently.

#### 1. Scaling factor

(1) scale down or scale up using bilinear

Factor =  $((src*2-3)<<11)/(dst-1)$ ;

(2) scale up using bicubic

Factor =  $((src*2-3)<<15)/(dst-1)$ ;

(3) scale down using average

Factor =  $(dst<<17)/(src*2-1)$ ;

#### 2. win scale line buffer mode

For YUV422/YUV420,

(1) LB\_YUV\_4096X5

If horizontal scale down and dsp\_width is greater than 2560 or if horizontal scale up and act\_width is greater than 2560, please config win\_lb\_mode as LB\_YUV\_4096X5;

(2) LB\_YUV\_2560X8

If horizontal scale down and dsp\_width is less or equal than 2560 or if horizontal scale up and act\_width is less or equal than 2560, please config win\_lb\_mode as LB\_YUV\_4096X5;

For YUV444/RGB,

(1) LB\_RGB\_4096X2

If horizontal scale down and dsp\_width is greater than 2560 or if horizontal scale up and act\_width is greater than 2560, please config win\_lb\_mode as LB\_YUV\_4096X2;

This mode does not support vertical scale.

(2) LB\_RGB\_2560X4

horizontal scale down and dsp\_width is greater than 1920 or if horizontal scale up and act\_width is greater than 1920, please config win\_lb\_mode as LB\_RGB\_2560X4;

This mode only support bilinear for vertical scale up.

(3) LB\_RGB\_1920X5

horizontal scale down and dsp\_width is greater than 1280 or if horizontal scale up and act\_width is greater than 1280, please config win\_lb\_mode as LB\_RGB\_1920X5;

(4) LB\_RGB\_1280X8

horizontal scale down and dsp\_width is less or equal than 2560 or if horizontal scale up and act\_width is less or equal than 2560, please config win\_lb\_mode as LB\_RGB\_1280X8;

### 3.3.4 P2I

It is necessary to display a non-interlaced video signal on an interlaced display panel (such as TV set). Thus "non-interlaced-to-interlaced conversion" is required, we call it P2I.

When interlaced is required, no matter even field or odd field, we get the whole image from bus. After overlay, we discard the odd lines when even field (even lines when odd field). This method can be better.

### 3.3.5 Virtual display

When in virtual display, the active image is part of the virtual (original) image in frame buffer memory.

The virtual width is indicated by setting VIR\_STRIDE for different data format. Note that RGB/BPP has one stride(yrgb\_vir\_stride); YCbCr has two virtual stride(yrgb\_vir\_stride and cbcr\_vir\_stride).

For RGB-8bit and YUV-8bit, the stride should be multiples of word (32-bit), with dummy bytes in the end of virtual line if the original width is not 32-bit aligned.

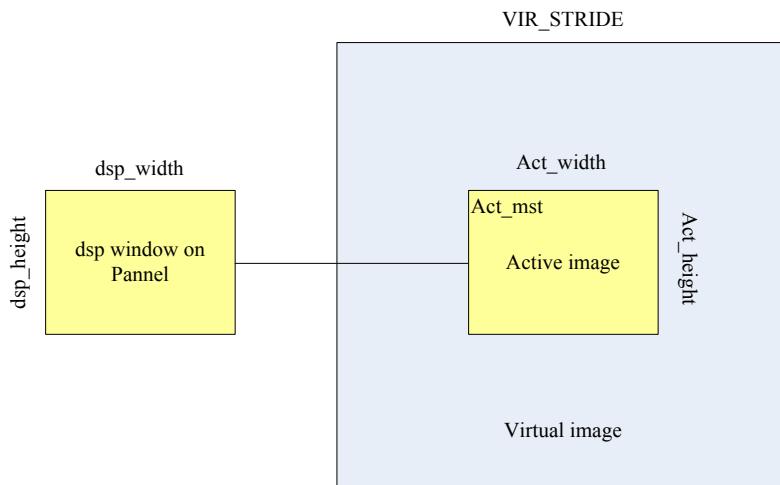


Fig. 7-16 Virtual display

### 3.3.6 MIRROR display

Mirror display is necessary for the panel with mirror timing interface. There are two types of mirror mode: horizontal mirror(X-mirror) and vertical mirror(Y-mirror).

Win0/1 support X-mirror and Y-mirror;

VOP support X-mirror and Y-mirror after overlay.

The default display order is from left to right(L2R) in horizontal direction and from top to bottom(T2B) in vertical direction. However, when X-Mirror is enable, the horizontal display order is from right to left(R2L); when Y-MIRROR is enable, the vertical display order is from bottom to top(B2T).

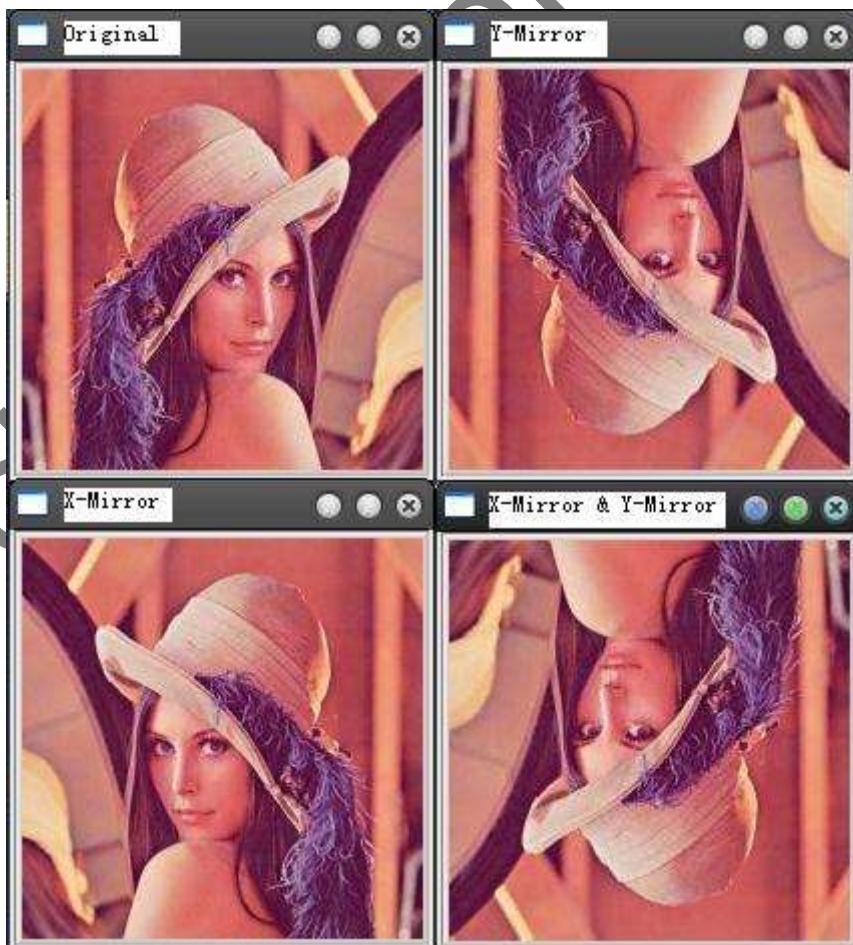


Fig. 7-17 X-Mirror and Y-Mirror

### 3.3.7 Display process

## 1. Overlay display

There are totally 4 layers for overlay display: Background, layer0, layer1, and hardware cursor layer(HWC).

Background is a programmable solid color layer, which is always in the bottom of the display screen.

HWC is always on the top of the display screen.

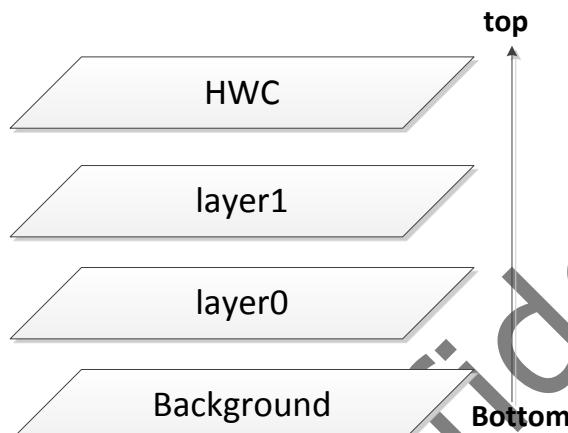


Fig. 7-18 overlay

Following figure is an example of overlay display for win0,win1 and hwc.

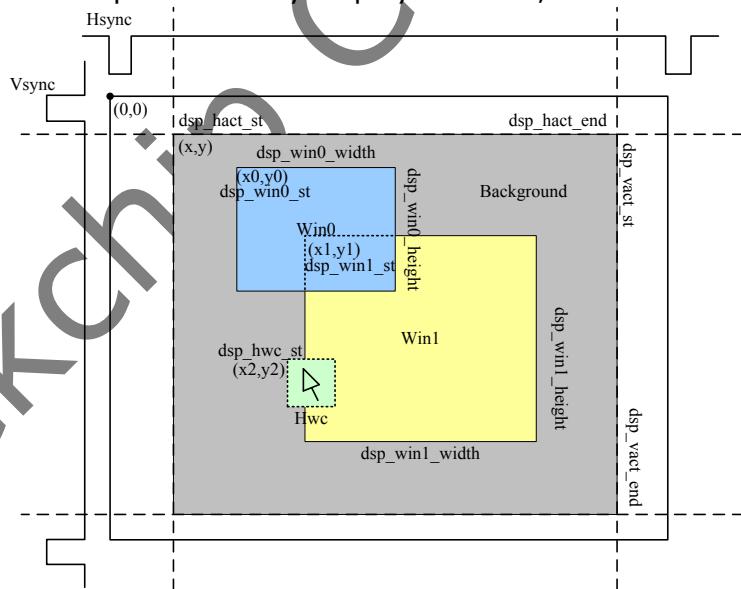


Fig. 7-19 overlay timing

## 2. Post scale down

Post scale down after overlay is supported to fix overscan, that draws the borders of the image beyond the normally visible area on the screen.

The scale ratio of post scale down is 0.5~1.

### Post timing setting

The post scale parameter ,such as,`post_dsp_hact_st`,`post_dsp_hact_end`,  
`post_dsp_vact_st`,`post_dsp_vact_end` can be configured.

When post scaling equal "1", the post scaler parameter are the same as dsp timing parameter.  
eg:

```
post_dsp_hact_st = dsp_hact_st
post_dsp_hact_end = dap_hact_end
post_dsp_vact_st = dsp_vact_st
post_dsp_vact_end = dsp_vact_end
```

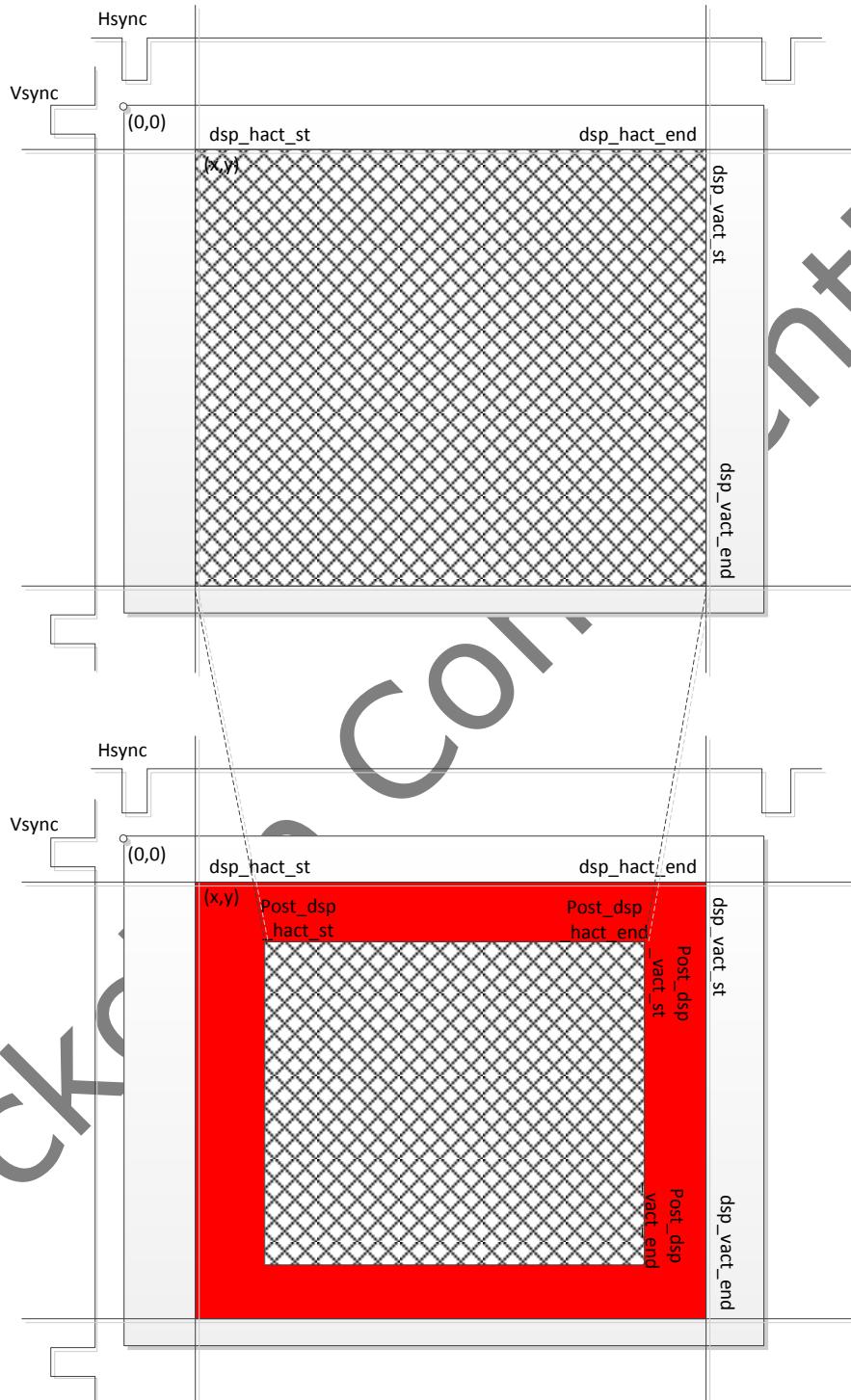


Fig. 7-20 post scaling timing

#### Post scale down factor

For horizontal scale down,factor =  $((src\_width*2-3)<<11)/(dst\_width-1)$ .

For vertical scale down,factor =  $((src\_width*2-3)<<11)/(dst\_width-1)$ .

### 3. Transparency color key

The transparency color key value defines the pixel treated as transparent pixel. The pixel whose value is equal to the color key value could not be visible on the screen, instead of the pixel in the under layer or solid background color.

There are two transparency color key for win0 layer and win1 layer respectively. When color key is enable, the transparency process is done after scaling but before YUV2RGB color space converter.

Moreover, transparency color key is just available for non-scaling mode.

Following figure is an example of transparency color key for win0 and win1.

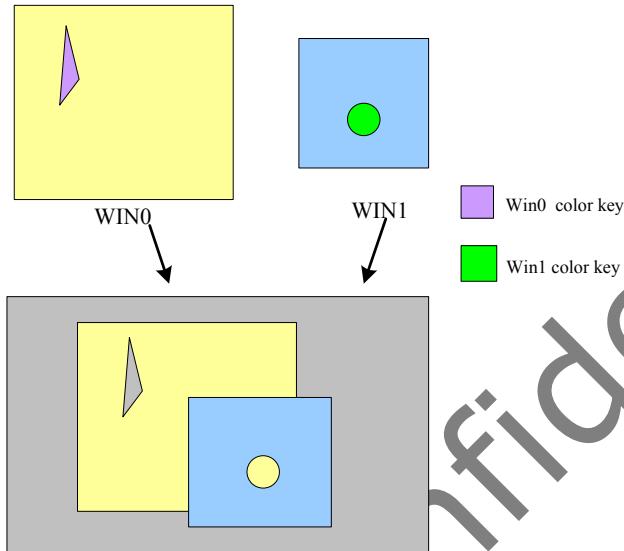


Fig. 7-21 Transparency Color Key

### 4. Replication(dither up)

If the size of panel data bus is larger than the size of source pixel data, i.e., the source input format is RGB565 and display output format is RGB888, you could do bit replication by replicating MSBs to LSBs if replication is enable (VOP\_DSP\_CTRL0[9]=1) or filling with "0" to LSBs if replication is disable (VOP\_DSP\_CTRL0[9]=0).

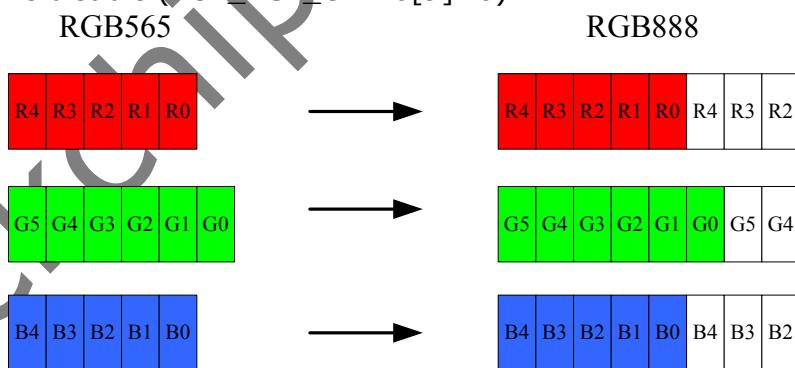


Fig. 7-22 Dither Up

### 5. Alpha blending

There are 12 alpha blending mode between two overlay layers for layer1/layer2/hwc. Layer0 does not support alpha blending with background. VOP supports overlay on RGB or YUV domain.

When in per-pixel mode, the alpha value for every pixel is following with the pixel data. i.e., ARGB, and can be scaled like RGB data. Therefore it is just suitable for win0/win1/hwc layer with ARGB data format.

The alpha blending architecture is shown as follows.

Table 7-2 alpha blending mode settings

Blending Mode	$C_s'$	$F_s$	$C_d'$	$F_d$
---------------	--------	-------	--------	-------

AA_USER_DEFINED	X	User defined	Cd	User defined
AA_CLEAR	X	0	Cd	0
AA_SRC	X	0	Cd	1
AA_DST	X	1	Cd	1
AA_SRC_OVER	Cs	1	Cd	1-As''
AA_DST_OVER	Cs	1-As''	Cd	1
AA_SRC_IN	Cs	As''	Cd	0
AA_DST_IN	X	0	Cd	As''
AA_SRC_OUT	Cs	1-As''	Cd	0
AA_DST_OUT	X	0	Cd	1-As''
AA_SRC_ATOP	Cs	As''	Cd	1-As''
AA_DST_ATOP	Cs	1-As''	Cd	As''
AA_XOR	Cs	1-As''	Cd	1-As''
AA_SRC_OVER_GLOBAL	Cs*As''	Ags''	Cd	1-As''

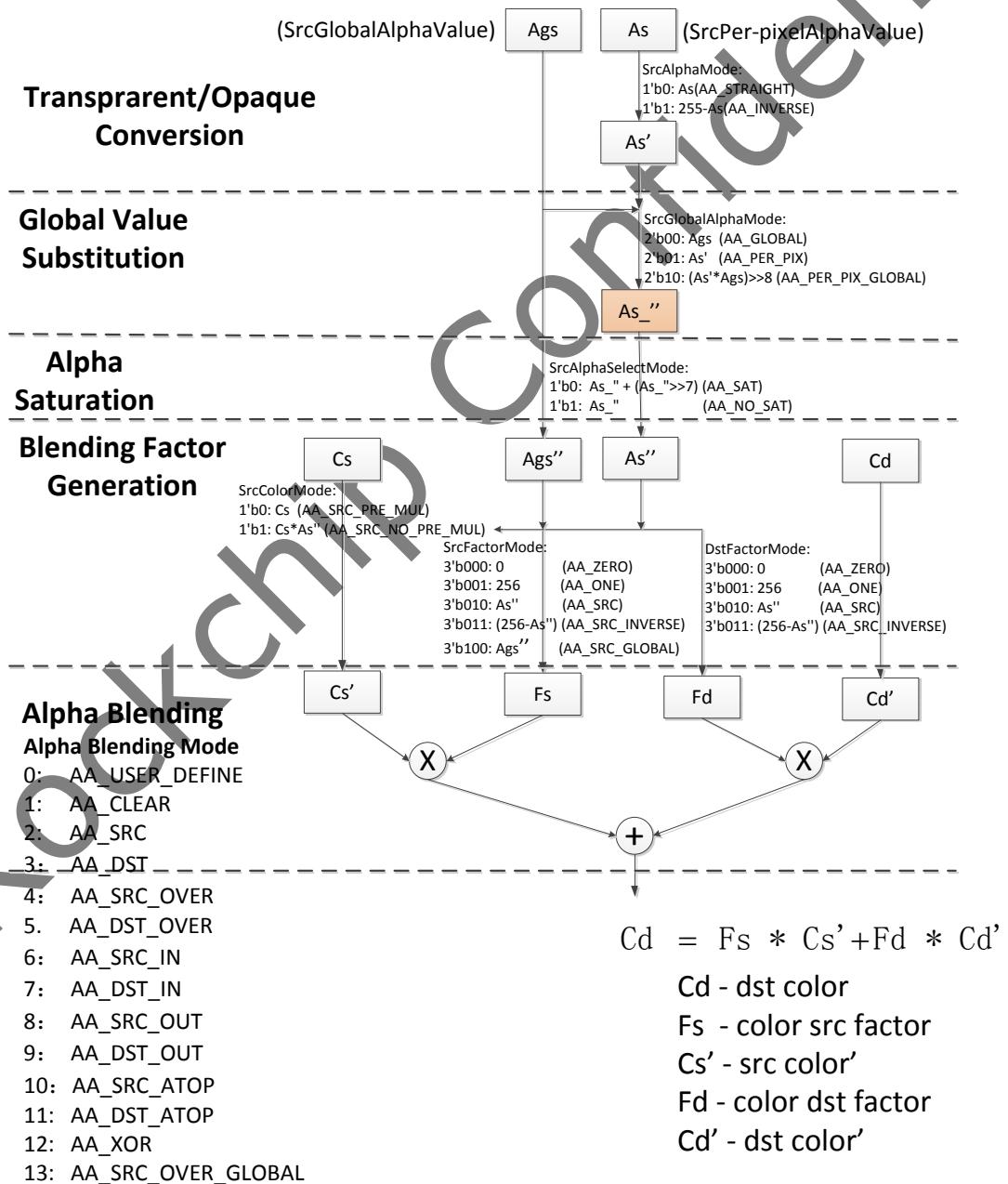


Fig. 7-23 alpha configuration flow

**Pseudo Code:**

```
switch(alpha_config->alpha_blending_mode)
{
    case AA_USER_DEFINE:
        break;
    case AA_CLEAR:
        alpha_config->src_factor_mode=AA_ZERO;
        alpha_config->dst_factor_mode=AA_ZERO;
        break;
    case AA_SRC:
        alpha_config->src_factor_mode=AA_ONE;
        alpha_config->dst_factor_mode=AA_ZERO;
        break;
    case AA_DST:
        alpha_config->src_factor_mode=AA_ZERO;
        alpha_config->dst_factor_mode=AA_ONE;
        break;
    case AA_SRC_OVER:
        alpha_config->src_color_mode=AA_SRC_PRE_MUL;
        alpha_config->src_factor_mode=AA_ONE;
        alpha_config->dst_factor_mode=AA_SRC_INVERSE;
        break;
    case AA_DST_OVER:
        alpha_config->src_color_mode=AA_SRC_PRE_MUL;
        alpha_config->src_factor_mode=AA_SRC_INVERSE;
        alpha_config->dst_factor_mode=AA_ONE;
        break;
    case AA_SRC_IN:
        alpha_config->src_color_mode=AA_SRC_PRE_MUL;
        alpha_config->src_factor_mode=AA_SRC;
        alpha_config->dst_factor_mode=AA_ZERO;
        break;
    case AA_DST_IN:
        alpha_config->src_factor_mode=AA_ZERO;
        alpha_config->dst_factor_mode=AA_SRC;
        break;
    case AA_SRC_OUT:
        alpha_config->src_color_mode=AA_SRC_PRE_MUL;
        alpha_config->src_factor_mode=AA_SRC_INVERSE;
        alpha_config->dst_factor_mode=AA_ZERO;
        break;
    case AA_DST_OUT:
        alpha_config->src_factor_mode=AA_ZERO;
        alpha_config->dst_factor_mode=AA_SRC_INVERSE;
        break;
    case AA_SRC_ATOP:
        alpha_config->src_color_mode=AA_SRC_PRE_MUL;
        alpha_config->src_factor_mode=AA_SRC;
        alpha_config->dst_factor_mode=AA_SRC_INVERSE;
        break;
    case AA_DST_ATOP:
        alpha_config->src_color_mode=AA_SRC_PRE_MUL;
        alpha_config->src_factor_mode=AA_SRC_INVERSE;
        alpha_config->dst_factor_mode=AA_SRC;
        break;
    case AA_XOR:
        alpha_config->src_color_mode=AA_SRC_PRE_MUL;
        alpha_config->src_factor_mode=AA_SRC_INVERSE;
```

```

alpha_config->dst_factor_mode=AA_SRC_INVERSE;
break;
case AA_SRC_OVER_GLOBAL:
alpha_config->src_global_alpha_mode=AA_PER_PIX_GLOBAL;
alpha_config->src_color_mode=AA_SRC_NO_PRE_MUL;
alpha_config->src_factor_mode=AA_SRC_GLOBAL;
alpha_config->dst_factor_mode=AA_SRC_INVERSE;
break;
default:
printf("alpha mode error\n");
break;
}

```

## 6. Brightness,Contrast,Saturation,Hue(BCSH)

BCSH is used to adjust "Brightness,Contrast,Saturation,Hue",like IEP BCSH-8bit. For details,please refer to IEP chapter. The brightness adjust support (-32,31).The yuv data of color bar are 8bits.

## 7. Color space conversion

There are 4 standards for YUV2RGB and RGB2YUV. For yuv2yuv ,we can use change among BT601L, BT601F, BT709L, BT2020 using coefficient registers.

YUV2RGB:

1. yuv to rgb (BT601L)

$$R = 1.164(Y-16) + 1.596(V-128)$$

$$G = 1.164(Y-16) - 0.391(U-128) - 0.813(V-128)$$

$$B = 1.164(Y-16) + 2.018(U-128)$$

2. yuv to rgb (BT601F)

$$R = (Y-16) + 1.402(V-128)$$

$$G = (Y-16) - 0.344(U-128) - 0.714(V-128)$$

$$B = (Y-16) + 1.772(U-128)$$

3. yuv to rgb (BT709L)

$$R = 1.164(Y-16) + 1.793(V-128)$$

$$G = 1.164(Y-16) - 0.213(U-128) - 0.534(V-128)$$

$$B = 1.164(Y-16) + 2.115(U-128)$$

4. yuv to rgb(BT2020)

$$R = 1.1636(Y-64) + 1.6778(V-512)$$

$$G = 1.1636(Y-64) - 0.1872(U-512) - 0.6501(V-512)$$

$$B = 1.1636(Y-64) + 2.1406(U-512)$$

RGB2YUV:

RGB2YUV

1. rgb to yuv(BT601L)

$$Y = 0.257R + 0.504G + 0.098B + 16$$

$$Cb = -0.148R - 0.291G + 0.439B + 128$$

$$Cr = 0.439R - 0.368G - 0.071B + 128$$

2. rgb to yuv(BT601F)

$$Y = 0.299R + 0.5870G + 0.1140B + 0$$

$$Cb = -0.1687R - 0.3313G + 0.5000B + 512$$

$$Cr = 0.500R - 0.4187G - 0.0813B + 512$$

3. rgb to yuv(BT709L)

$$Y = 0.183R + 0.614G + 0.062B + 16$$

$$Cb = -0.101R - 0.338G + 0.439B + 128$$

$$Cr = 0.439R - 0.399G - 0.040B + 128$$

## 3. rgb to yuv(BT2020)

$$\begin{aligned} Y &= 0.2250R + 0.5807G + 0.0508B + 64 \\ Cb &= -0.1223R - 0.3157G + 0.4380B + 512 \\ Cr &= 0.4380R - 0.4028G - 0.0352B + 512 \end{aligned}$$

## 8. Pre-Dither Down

Dithering is an intentional applied form of [noise](#), using to randomize [quantization error](#), and thereby preventing large-scaling patterns such as "banding".

The pixel value is used by dithering process to display the data in a lower color depth on the LCD panel, i.e, the source input format is RGB101010 and display output format is RGB888. When dithering is enable(VOP\_DSP\_CTRL0[2]=1), the output data is generated by dithering algorithm based on the pixel position and the value of removed bits. Otherwise, the MSBs of the pixel color components are output as display data.

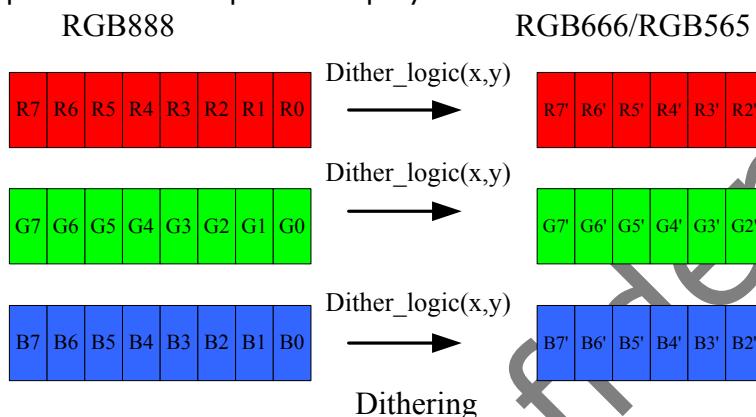
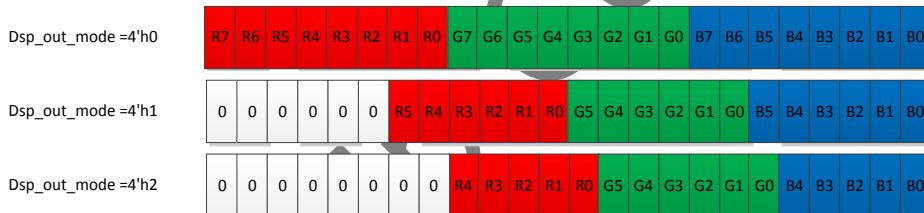


Fig. 7-24 Pre-Dither Down

## 9. Output format

Config `dsp_out_mode` register to adapt a variety of panel interface. As follow:

Fig. 7-25 `dsp_out_mode` description

## 3.4 Register Description

## 3.4.1 Internal Address Mapping

Slave address can be divided into different length for different usage, which is shown as follows.

## 3.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
VOP_REG_CFG_DONE	0x0000	W	0x00000000	Register config done flag
VOP_VERSION_INFO	0x0004	W	0x00000000	Version for vop
VOP_SYS_CTRL	0x0008	W	0x00821800	System control register0
VOP_SYS_CTRL1	0x000c	W	0x0003a000	System control register1
VOP_DSP_CTRL0	0x0010	W	0x00000000	Display control register0
VOP_DSP_CTRL1	0x0014	W	0x0000e400	Display control register1
VOP_DSP_BG	0x0018	W	0x00000000	Background color
VOP_WIN0_CTRL0	0x0030	W	0x3a000040	Win0 ctrl register0

<b>Name</b>	<b>Offset</b>	<b>Size</b>	<b>Reset Value</b>	<b>Description</b>
VOP_WIN0_CTRL1	0x0034	W	0x00000000	Win0 ctrl register1
VOP_WIN0_COLOR_KEY	0x0038	W	0x00000000	Win0 color key register
VOP_WIN0_VIR	0x003c	W	0x01400140	Win0 virtual stride
VOP_WIN0_YRGB_MST	0x0040	W	0x00000000	Win0 YRGB memory start address
VOP_WIN0_CBR_MST	0x0044	W	0x00000000	Win0 Cbr memory start address
VOP_WIN0_ACT_INFO	0x0048	W	0x00ef013f	Win0 active window width/height
VOP_WIN0_DSP_INFO	0x004c	W	0x00ef013f	Win0 display width/height on panel
VOP_WIN0_DSP_ST	0x0050	W	0x000a000a	Win0 display start point on panel
VOP_WIN0_SCL_FACTOR_YRGB	0x0054	W	0x10001000	Win0 YRGB scaling factor
VOP_WIN0_SCL_FACTOR_CBR	0x0058	W	0x10001000	Win0 Cbr scaling factor
VOP_WIN0_SCL_OFFSET	0x005c	W	0x00000000	Win0 scaling start point offset
VOP_WIN0_SRC_ALPHA_CTRL	0x0060	W	0x00000000	Win0 alpha source control register
VOP_WIN0_DST_ALPHA_CTRL	0x0064	W	0x00000000	Win0 alpha destination control register
VOP_WIN0_FADING_CTRL	0x0068	W	0x00000000	Win0 fading contrl register
VOP_WIN0_CTRL2	0x006c	W	0x00000021	Win0 ctrl register2
VOP_WIN1_CTRL0	0x0070	W	0x3a000040	Win1 ctrl register0
VOP_WIN1_CTRL1	0x0074	W	0x00000000	Win1 ctrl register1
VOP_WIN1_COLOR_KEY	0x0078	W	0x00000000	Win1 color key register
VOP_WIN1_VIR	0x007c	W	0x01400140	win1 virtual stride
VOP_WIN1_YRGB_MST	0x0080	W	0x00000000	Win1 YRGB memory start address
VOP_WIN1_CBR_MST	0x0084	W	0x00000000	Win1 Cbr memory start address
VOP_WIN1_ACT_INFO	0x0088	W	0x00ef013f	Win1 active window width/height
VOP_WIN1_DSP_INFO	0x008c	W	0x00ef013f	Win1 display width/height on panel
VOP_WIN1_DSP_ST	0x0090	W	0x000a000a	Win1 display start point on panel
VOP_WIN1_SCL_FACTOR_YRGB	0x0094	W	0x10001000	Win1 YRGB scaling factor
VOP_WIN1_SCL_FACTOR_CBR	0x0098	W	0x10001000	Win1 Cbr scaling factor
VOP_WIN1_SCL_OFFSET	0x009c	W	0x00000000	Win1 scaling start point offset
VOP_WIN1_SRC_ALPHA_CTRL	0x00a0	W	0x00000000	Win1 alpha source control register
VOP_WIN1_DST_ALPHA_CTRL	0x00a4	W	0x00000000	Win1 alpha destination control register
VOP_WIN1_FADING_CTRL	0x00a8	W	0x00000000	Win1 fading contrl register
VOP_WIN1_CTRL2	0x00ac	W	0x00000043	Win1 ctrl register2
VOP_HWC_CTRL0	0x0150	W	0x00000000	Hwc ctrl register0
VOP_HWC_CTRL1	0x0154	W	0x00701d00	Hwc ctrl register1
VOP_HWC_MST	0x0158	W	0x00000000	Hwc memory start address
VOP_HWC_DSP_ST	0x015c	W	0x000a000a	Hwc display start point on panel

Name	Offset	Size	Reset Value	Description
VOP_HWC_SRC_ALPHA_CTRL	0x0160	W	0x00000000	Hwc alpha source control register
VOP_HWC_DST_ALPHA_CTRL	0x0164	W	0x00000000	Hwc alpha destination control register
VOP_HWC_FADE_CTRL	0x0168	W	0x00000000	Hwc fading control register
VOP_HWC_RESERVED1	0x016c	W	0x00000000	Hwc reserved
VOP_POST_DSP_HACT_INFO	0x0170	W	0x000a014a	Post scaler down horizontal start and end
VOP_POST_DSP_VACT_INFO	0x0174	W	0x000a00fa	Panel active horizontal scanning start point and end point
VOP_POST_SCL_FACTOR_YRGB	0x0178	W	0x10001000	Post yrgb scaling factor
VOP_POST_RESERVED	0x017c	W	0x00000000	Post reserved
VOP_POST_SCL_CTRL	0x0180	W	0x00000000	Post scaling start point offset
VOP_POST_DSP_VACT_INFO_F1	0x0184	W	0x000a00fa	Panel active horizontal scanning start point and end point F1
VOP_DSP_HTOTAL_HS_END	0x0188	W	0x014a000a	Panel scanning horizontal width and hsync pulse end point
VOP_DSP_HACT_ST_END	0x018c	W	0x000a014a	Panel active horizontal scanning start point and end point
VOP_DSP_VTOTAL_VS_END	0x0190	W	0x00fa000a	Panel scanning vertical height and vsync pulse end point
VOP_DSP_VACT_ST_END	0x0194	W	0x000a00fa	Panel active vertical scanning start point and end point
VOP_DSP_VS_ST_END_F1	0x0198	W	0x00000000	Vertical scanning start point and vsync pulse end point of even field in interlace mode
VOP_DSP_VACT_ST_END_F1	0x019c	W	0x00000000	Vertical scanning active start point and end point of even field in interlace mode
VOP_BCSH_COLOR_BAR	0x01b0	W	0x00000000	Color bar config register
VOP_BCSH_BCS	0x01b4	W	0xd0010000	Brightness contrast saturation*contrast config register
VOP_BCSH_H	0x01b8	W	0x01000000	Sin hue and cos hue config register
VOP_BCSH_CTRL	0x01bc	W	0x00000000	BCSH control register
VOP_FRC_LOWER01_0	0x01e8	W	0x12844821	FRC lookup table config register010
VOP_FRC_LOWER01_1	0x01ec	W	0x21488412	FRC lookup table config register011
VOP_FRC_LOWER10_0	0x01f0	W	0xa55a9696	FRC lookup table config register100
VOP_FRC_LOWER10_1	0x01f4	W	0x5aa56969	FRC lookup table config register101
VOP_FRC_LOWER11_0	0x01f8	W	0xdeb77bed	FRC lookup table config register110

Name	Offset	Size	Reset Value	Description
VOP_FRC_LOWER11_1	0x01fc	W	0xed7bb7de	FRC lookup table config register111
VOP_INTR_EN0	0x0280	W	0x00000000	Interrupt enable register
VOP_INTR_CLEAR0	0x0284	W	0x00000000	Interrupt clear register
VOP_INTR_STATUS0	0x0288	W	0x00000000	interrupt status
VOP_INTR_RAW_STATUS0	0x028c	W	0x00000000	raw interrupt status
VOP_INTR_EN1	0x0290	W	0x00000000	Interrupt enable register
VOP_INTR_CLEAR1	0x0294	W	0x00000000	Interrupt clear register
VOP_INTR_STATUS1	0x0298	W	0x00000000	interrupt status
VOP_INTR_RAW_STATUS1	0x029c	W	0x00000000	raw interrupt status
VOP_LINE_FLAG	0x02a0	W	0x00000000	Line flag config register
VOP_VOP_STATUS	0x02a4	W	0x00000000	vop status register
VOP_BLANKING_VALUE	0x02a8	W	0x00000000	Register0000 Abstract
VOP_WIN0_DSP_BG	0x02b0	W	0x00000000	Win0 layer background color
VOP_WIN1_DSP_BG	0x02b4	W	0x00000000	Win1 layer background color
VOP_DBG_PERF_LATENCY_CTRL0	0x0300	W	0x00000000	Axi performance latency module contrl register0
VOP_DBG_PERF_RD_MAX_LATENCY_NUM0	0x0304	W	0x00000000	Read max latency number
VOP_DBG_PERF_RD_LATE_NCYTHR_NUM0	0x0308	W	0x00000000	The number of bigger than configed threshold value
VOP_DBG_PERF_RD_LATE_NCY_SAMP_NUM0	0x030c	W	0x00000000	Total sample number
VOP_DBG_WIN0_REG0	0x0320	W	0x00000000	Vop debug win0 register0
VOP_DBG_WIN0_REG1	0x0324	W	0x00000000	Vop debug win0 register1
VOP_DBG_WIN0_REG2	0x0328	W	0x00000000	Vop debug win0 register2
VOP_DBG_WIN0_RESERVED	0x032c	W	0x00000000	Vop debug win0 register3 reserved
VOP_DBG_WIN1_REG0	0x0330	W	0x00000000	Vop debug win1 register0
VOP_DBG_WIN1_REG1	0x0334	W	0x00000000	Vop debug win1 register1
VOP_DBG_WIN1_REG2	0x0338	W	0x00000000	Vop debug win1 register2
VOP_DBG_WIN1_RESERVED	0x033c	W	0x00000000	Vop debug win1 register3 reserved
VOP_DBG_PRE_REG0	0x0360	W	0x00000000	Vop debug pre register0
VOP_DBG_PRE_RESERVED	0x0364	W	0x00000000	Vop debug pre register1 reserved
VOP_DBG_POST_REG0	0x0368	W	0x00000000	Vop debug post register0
VOP_DBG_POST_RESERVED	0x036c	W	0x00000000	Vop debug post register1 reserved
VOP_DBG_DATAO	0x0370	W	0x00000000	debug data output path
VOP_DBG_DATAO_2	0x0374	W	0x00000000	debug data output path 2

Name	Offset	Size	Reset Value	Description
VOP_POST_YUV2YUV_Y2R_COE	0x0480	W	0x00000000	POST_YUV2YUV_Y2R_COE
VOP_POST_YUV2YUV_3x3_COE	0x04a0	W	0x00000000	POST_YUV2YUV_3x3_COE
VOP_POST_YUV2YUV_R2Y_COE	0x04c0	W	0x00000000	POST_YUV2YUV_R2Y_COE
VOP_HWC_LUT_ADDR	0x1800	W	0x00000000	Hwc lut base address
TVE_TVE_MODE_CTRL	0x3e00	W	0x00000000	Mode Control
TVE_TVE_HOR_TIMING1	0x3e04	W	0x00000000	Horizontal Timing 1
TVE_TVE_HOR_TIMING2	0x3e08	W	0x00000000	Horizontal Timing 2
TVE_TVE_HOR_TIMING3	0x3e0c	W	0x00000000	Horizontal Timing 3
TVE_TVE_SUB_CAR_FRQ	0x3e10	W	0x00000000	Sub-carrier Frequency
TVE_TVE_LUMA_FILTER1	0x3e14	W	0x00000000	Luma Filter 1
TVE_TVE_LUMA_FILTER2	0x3e18	W	0x00000000	Luma Filter 2
TVE_TVE_LUMA_FILTER3	0x3e1c	W	0x00000000	Luma Filter 3
TVE_TVE_MACROVISION_RESERVED0	0x3e20	W	0x00000000	tve macrovision reserved0 register
TVE_TVE_MACROVISION_RESERVED1	0x3e24	W	0x00000000	tve macrovision reserved1 register
TVE_TVE_MACROVISION_RESERVED2	0x3e28	W	0x00000000	tve macrovision reserved2 register
TVE_TVE_MACROVISION_RESERVED3	0x3e2c	W	0x00000000	tve macrovision reserved3 register
TVE_TVE_MACROVISION_RESERVED4	0x3e30	W	0x00000000	tve macrovision reserved3 register
TVE_TVE_IMAGE_POSITION	0x3e34	W	0x00000000	tve image position
TVE_TVE_ROUTING	0x3e38	W	0x00000000	Tve Routing
TVE_TVE_VBID_ST_TIMING	0x3e3c	W	0x00000000	tve vbid start timing
TVE_TVE_VBID_LINES	0x3e40	W	0x00000000	tve vbid lines
TVE_TVE_VBID_CC_DATA	0x3e44	W	0x00000000	tve vbid cc data
TVE_TVE_VBID_XDS_DATA	0x3e48	W	0x00000000	tve vbid xds data
TVE_TVE_VBID_CGMS_DATA	0x3e4c	W	0x00000000	tve vbid cgms data
TVE_TVE_SYNC_ADJUST	0x3e50	W	0x00000000	Sync Adjust
TVE_TVE_STATUS	0x3e54	W	0x00000000	TVE Status (Read Only)
TVE_TVE_SURFACE_SIZE_RESERVED	0x3e58	W	0x00000000	tve surface size reserved
TVE_TVE_STREAM_BASE_ADDR_CTRL_RESERVED	0x3e5c	W	0x00000000	tve stream base addr ctrl reserved
TVE_TVE_STRIDE_RESERVED	0x3e60	W	0x00000000	tve stride reserved

Name	Offset	Size	Reset Value	Description
TVE_TVE_MEM_INF_CTRL_RESERVED	0x3e64	W	0x00000000	tve mem inf ctrl reserved
TVE_TVE_CTRL	0x3e68	W	0x00000000	TVE Control
TVE_TVE_INTR_STATUS	0x3e6c	W	0x00000000	Interrupt Status
TVE_TVE_INTR_EN	0x3e70	W	0x00000000	Interrupt Enable
TVE_TVE_INTR_CLR	0x3e74	W	0x00000000	Interrupt Clear
TVE_TVE_COLOR_BUSRT_SAT	0x3e78	W	0x00000000	Colour Burst and Saturation
TVE_TVE_TEXT_HOR_ACT_RESERVED	0x3e7c	W	0x00000000	Teletext Horizontal Active
TVE_TVE_TEXT_VER_ACT_RESERVED	0x3e80	W	0x00000000	tve text ver act reserved
TVE_TVE_TEXT_MODE_RATE_RESERVED	0x3e84	W	0x00000000	tve text mode rate reserved
TVE_TVE_TEXT_FETCH_RESERVED	0x3e88	W	0x00000000	tve text fetch reserved
TVE_TVE_CHROMA_BANDWIDTH	0x3e8c	W	0x00000000	Chroma bandwidth
TVE_TVE_BRIGHTNESS_CONTRAST	0x3e90	W	0x00000000	Brightness and Contrast
TVE_TVE_TEXT_DATA_ID_RESERVED	0x3e94	W	0x00000000	tve text data id reserved
TVE_TVE_ID	0x3e98	W	0xa010000	TVE ID (Read Only)
TVE_TVE_REVISION	0x3e9c	W	0x0010008	TVE Revision (Read Only)
TVE_TVE_CLAMP	0x3ea0	W	0x00000000	TVE Clamp
VOP_MMU_DTE_ADDR	0x3f00	W	0x00000000	MMU current page Table address
VOP_MMU_STATUS	0x3f04	W	0x00000000	MMU status register
VOP_MMU_COMMAND	0x3f08	W	0x00000000	MMU command register
VOP_MMU_PAGE_FAULT_ADDR	0x3f0c	W	0x00000000	MMU logical address of last page fault
VOP_MMU_ZAP_ONE_LINE	0x3f10	W	0x00000000	MMU Zap cache line register
VOP_MMU_INT_RAWSTAT	0x3f14	W	0x00000000	MMU raw interrupt status register
VOP_MMU_INT_CLEAR	0x3f18	W	0x00000000	MMU raw interrupt status register
VOP_MMU_INT_MASK	0x3f1c	W	0x00000000	MMU raw interrupt status register
VOP_MMU_INT_STATUS	0x3f20	W	0x00000000	MMU raw interrupt status register
VOP_MMU_AUTO_GATING	0x3f24	W	0x00000000	MMU auto gating

Notes: Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

### 1.4.3 Detail Register Description

#### VOP\_REG\_CFG\_DONE

Address: Operational Base + offset (0x0000)

Register config done flag

Bit	Attr	Reset Value	Description
-----	------	-------------	-------------

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15:9	RO	0x0	reserved
8	RW	0x0	reg_load_sys_en vop system register config done flag In the first setting of the register, the new value was saved into the mirror register. When all the system register config finish, writing this register to enable the copyright of the mirror register to real register. Then register would be updated at the start of every frame.
7	RW	0x0	reg_load_fbdc_en vop fbdc register config done flag In the first setting of the register, the new value was saved into the mirror register. When all the fbdc register config finish, writing this register to enable the copyright of the mirror register to real register. Then register would be updated at the start of every frame.
6	RW	0x0	reg_load_iep_en vop iep register config done flag In the first setting of the register, the new value was saved into the mirror register. When all the iep register config finish(only 2 signals direct_path_en,direct_path_layer_sel ), writing this register to enable the copyright of the mirror register to real register. Then register would be updated at the start of every frame.
5	RW	0x0	reg_load_hwc_en vop hwc register config done flag In the first setting of the register, the new value was saved into the mirror register. When all the hwc register config finish, writing this register to enable the copyright of the mirror register to real register. Then register would be updated at the start of every frame.
4	RW	0x0	reg_load_win3_en vop win3 register config done flag In the first setting of the register, the new value was saved into the mirror register. When all the win3 register config finish, writing this register to enable the copyright of the mirror register to real register. Then register would be updated at the start of every frame.

Bit	Attr	Reset Value	Description
3	RW	0x0	reg_load_win2_en vop win2 register config done flag In the first setting of the register, the new value was saved into the mirror register. When all the win2 register config finish, writing this register to enable the copyright of the mirror register to real register. Then register would be updated at the start of every frame.
2	RW	0x0	reg_load_win1_en vop win1 register config done flag In the first setting of the register, the new value was saved into the mirror register. When all the win1 register config finish, writing this register to enable the copyright of the mirror register to real register. Then register would be updated at the start of every frame.
1	RW	0x0	reg_load_win0_en vop win0 register config done flag In the first setting of the register, the new value was saved into the mirror register. When all the win0 register config finish, writing this register to enable the copyright of the mirror register to real register. Then register would be updated at the start of every frame.
0	WO	0x0	reg_load_en vop register config done flag In the first setting of the register, the new value was saved into the mirror register. When all the register config finish, writing this register to enable the copyright of the mirror register to real register. Then register would be updated at the start of every frame.

**VOP\_VERSION\_INFO**

Address: Operational Base + offset (0x0004)

Version for vop

Bit	Attr	Reset Value	Description
31:24	RO	0x00	major IP major version used for IP structure
23:16	RO	0x00	minor minor version big feature change under same structure
15:0	RO	0x0000	svnbuid rtl current svn number

**VOP\_SYS\_CTRL**

Address: Operational Base + offset (0x0008)

System control register0

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RW	0x0	io_pad_clk_sel io_pad_clk_sel io_pad_clk_sel
30	RW	0x0	vop_field_tve_pol vop_field_tve_pol vop_field_tve_pol
29	RW	0x0	sw_dac_sel dac output sel for tve in fpga when fpga test: dac output sel for tve 1'b0:dac 3 1'b1:dac 1 others: tve enable 1'b0:disable tve 1'b1:enable tve
28	RW	0x0	sw_genlock genlock for tve genlock for tve in fpga 1'b0:master mode 1'b1:slave mode
27	RW	0x0	sw_uv_offset_en uv offset enable uv offset enable
26	RW	0x0	sw_tve_mode tve mode 1'b0:NTSC 1'b1:PAL
25	RW	0x0	sw_imd_tve_dclk_pol tve dclk pol tve dclk pol
24	RW	0x0	sw_imd_tve_dclk_en tve dclk enable tve dclk enable
23	RW	0x1	auto_gating_en LCDC layer axi-clk auto gating enable 1'b0 : disable auto gating 1'b1 : enable auto gating default auto gating enable

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
22	RW	0x0	<p>vop_standby_en LCDC standby mode Writing "1" to turn LCDC into standby mode, All the layer would disable and the data transfer from frame buffer memory would stop at the end of current frame. The output would be blank.</p> <p>When writing "0" to this bit, standby mode would disable and the LCDC go back to work immediately.</p> <p>1'b0 : disable 1'b1 : enable * Black display is recommended before setting standby mode enable.</p>
21	RW	0x0	<p>vop_dma_stop VOP DMA stop mode 1'b0 : disable 1'b1 : enable * If DMA is working, the stop mode would not be active until current bus transfer is finished.</p>
20	RW	0x0	<p>vop_mmu_en vop mmu enable signal 1'b0 : bypass mmu 1'b1 : enable mmu</p>
19	RW	0x0	<p>win23_pri_opt_mode 1'b0: win2 win3 dma priority enable 1'b1: win2 win3 dma priority disable</p>
18	RW	0x0	<p>post_lb_mode 1'b0 : 4x4096 1'b1 : 8x2048</p>
17	RW	0x1	<p>fs_same_addr_mask_en 1'b0 : disable 1'b1 : enable</p>
16	RW	0x0	<p>overlay_mode 1'b0: RGB overlay 1'b1: YUV overlay</p>
15	RW	0x0	<p>mipi_out_en 1'b0 : gating output clk ,data and control signal 1'b1 : mipi interface enable</p>
14	RW	0x0	<p>edp_out_en 1'b0 : gating output clk ,data and control signal 1'b1 : edp interface enable</p>
13	RW	0x0	<p>hdmi_out_en 1'b0 : gating output clk ,data and control signal 1'b1 : hdmi interface enable</p>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
12	RW	0x1	rgb_out_en 1'b0 : gating output clk ,data and control signal 1'b1 : rgb/lvds interface enable
11	RW	0x1	global_regdone_en global config done enalbe 1'b0: DISABLE 1'b1: ENALBE
10	RW	0x0	edpi_wms_fs edpi wms mode , frame st signal write "1": edpi_wms_mode frame start (when other register is config done) read : wms mode hold status
9	RW	0x0	edpi_wms_mode 1'b1: mipi command mode
8	RW	0x0	edpi_halt_en mipi flow ctrl enable
7:3	RO	0x0	reserved
2:1	RW	0x0	direct_path_layer_sel direct path layer select 2'b00 : select win0 2'b01 : select win1 2'b10 : select win2 2'b11 : select win3
0	RW	0x0	direct_path_en iep direct path enable signal 1'b0 : disable iep direct path 1'b1 : enable iep direct path

**VOP\_SYS\_CTRL1**

Address: Operational Base + offset (0x000c)

System control register1

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RW	0x0	dsp_fp_standby dsp_fp_standby dsp_fp_standby
30:25	RO	0x0	reserved
24	RW	0x0	reg_done_frm reg done frame valid reg done framd valid 1'b0:reg done every field for interlace 1'b1:reg done every frame for interlace
23:22	RW	0x0	noc_hurry_w_value 2'b00: low priority 2'b11: high priority

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
21:20	RW	0x0	noc_hurry_w_mode 2'b00: noc_hurry_w disable 2'b01: left 1/4 fifo empty 2'b10: left 1/2 fifo empty 2'b11: left 3/4 fifo empty
19:18	RO	0x0	reserved
17:13	RW	0x1d	axi_outstanding_max_num axi bus max outstanding number
12	RW	0x0	axi_max_outstanding_en axi bus max outstanding enable
11:10	RW	0x0	noc_win_qos Noc win qos
9	RW	0x0	noc_qos_en Noc qos enable
8:3	RW	0x00	noc_hurry_threshold Noc hurry threshold value
2:1	RW	0x0	noc_hurry_value Noc hurry value
0	RW	0x0	noc_hurry_en Noc hurry enable

**VOP\_DSP\_CTRL0**

Address: Operational Base + offset (0x0010)

Display control register0

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x0	reserved
23	RW	0x0	dsp_y_mir_en 1'b0 : no y_mirror 1'b1 : y_mirror
22	RW	0x0	dsp_x_mir_en 1'b0 : no x_mirror 1'b1 : x_mirror
21	RW	0x0	dsp_yuv_clip YCrCb clip 1'b0 : disable, YCbCr no clip 1'b1 : enable, YCbCr clip before YCbCr2RGB *Y clip: 16~235, CbCr clip: 16~239
20	RW	0x0	dsp_ccir656_avg Cb-Cr filter in CCIR656 mode 1'b0 : drop mode 1'b1 : average mode
19	RW	0x0	dsp_black_en Black display mode When this bit enable, the pixel data output is all black (0x000000)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
18	RW	0x0	dsp_blank_en Blank display mode When this bit enable, the Hsync/Vsync/Den output is blank
17	RW	0x0	dsp_out_zero Hsync/Vsync/Den output software ctrl 1'b0 : normal output 1'b1 : all output '0'
16	RW	0x0	dsp_dummy_swap Display dummy swap enable 1'b0 : B+G+R+dummy 1'b1 : dummy+B+G+R
15	RW	0x0	dsp_delta_swap Display delta swap enable 1'b0 : disable 1'b1 : enable *See detail description in Delta display chapter.
14	RW	0x0	dsp_rg_swap Display output red and green swap enable 1'b0 : RGB 1'b1 : GRB
13	RW	0x0	dsp_rb_swap Display output red and blue swap enable 1'b0 : RGB 1'b1 : BGR
12	RW	0x0	dsp_bg_swap Display output blue and green swap enable 1'b0 : RGB 1'b1 : RBG
11	RW	0x0	dsp_field_pol field polarity when interlace dsp 1'b0 : normal 1'b1 : invert
10	RW	0x0	dsp_interlace Interlace display enable 1'b0 : disable 1'b1 : enable *This mode is related to the ITU-R656 output, the display timing of odd field must be set correctly. (lc当地dsp_vs_st_end_f1/lcdc_dsp_vact_end_f1)
9	RW	0x0	dsp_ddr_phase dclk phase lock 1'b0 : no lock 1'b1 : lock every line

Bit	Attr	Reset Value	Description
8	RW	0x0	dsp_dclk_ddr dclk output mode 1'b0 : SDR 1'b1 : DDR
7:6	RO	0x0	reserved
5	RW	0x0	sw_p2i_en p2i enable 1'b0:disable p2i 1'b1:enable p2i
4	RW	0x0	sw_core_dclk_sel 1'b0: dclk_core sel dclk 1'b1: dclk_core sel dclk div2
3:0	RW	0x0	dsp_out_mode Display output format 4'b0000: Parallel 24-bit RGB888 output R[7:0],G[7:0],B[7:0] 4'b0001: Parallel 18-bit RGB666 output 6'b0,R[5:0],G[5:0],B[5:0] 4'b0010: Parallel 16-bit RGB565 output 8'b0,R[4:0],G[5:0],B[4:0] 4'b0011: Parallel 24-bit RGB888 double pixel mix out phase0:G1[3:0],B1[7:0],G0[3:0],B0[7:0] phase1:R1[7:0],G1[7:4],R0[7:0],G0[7:4] 4'b0100: Serial 2x12-bit 12'b0,G[3:0],B[7:0] + 12'b0,R[7:0],G[7:4] 4'b0101: ITU-656 output mode0 16'b0,pixel_data[7:0] 4'b0110: ITU-656 output mode1 8'b0,pixel_data[7:0],8'b0 4'b0111: ITU-656 output mode2 9'b0,pixel_data[7:0],7'b0 4'b1000: Serial 3x8-bit RGB888 16'b0, B[7:0]+16'b0,G[7:0]+16'b0,R[7:0] 4'b1100: Serial 3x8-bit RGB888 + dummy 16'b0, B[7:0]+16'b0,G[7:0]+16'b0,R[7:0] + dummy 4'b1110: YUV420 output for HDMI 4'b1111: Parallel 30-bit RGBaaa output R[9:0],G[9:0],B[9:0] Others: Reserved.

**VOP\_DSP\_CTRL1**

Address: Operational Base + offset (0x0014)

Display control register1

Bit	Attr	Reset Value	Description
-----	------	-------------	-------------

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RW	0x0	mipi_dclk_pol DCLK invert enable 1'b0 : normal 1'b1 : invert default dclk invert
30	RW	0x0	mipi_den_pol DEN polarity 1'b0 : positive 1'b1 : negative
29	RW	0x0	mipi_vsync_pol VSYNC polarity 1'b0 : negative 1'b1 : positive
28	RW	0x0	mipi_hsync_pol HSYNC polarity 1'b0 : negative 1'b1 : positive
27	RW	0x0	edp_dclk_pol DCLK invert enable 1'b0 : normal 1'b1 : invert default dclk invert
26	RW	0x0	edp_den_pol DEN polarity 1'b0 : positive 1'b1 : negative
25	RW	0x0	edp_vsync_pol VSYNC polarity 1'b0 : negative 1'b1 : positive
24	RW	0x0	edp_hsync_pol HSYNC polarity 1'b0 : negative 1'b1 : positive
23	RW	0x0	hdmi_dclk_pol DCLK invert enable 1'b0 : normal 1'b1 : invert default dclk invert
22	RW	0x0	hdmi_den_pol DEN polarity 1'b0 : positive 1'b1 : negative

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
21	RW	0x0	hdmi_vsync_pol VSYNC polarity 1'b0 : negative 1'b1 : positive
20	RW	0x0	hdmi_hsync_pol HSYNC polarity 1'b0 : negative 1'b1 : positive
19	RW	0x0	rgb_lvds_dclk_pol DCLK invert enable 1'b0 : normal 1'b1 : invert default dclk invert
18	RW	0x0	rgb_lvds_den_pol DEN polarity 1'b0 : positive 1'b1 : negative
17	RW	0x0	rgb_lvds_vsync_pol VSYNC polarity 1'b0 : negative 1'b1 : positive
16	RW	0x0	rgb_lvds_hsync_pol HSYNC polarity 1'b0 : negative 1'b1 : positive
15:14	RW	0x3	dsp_layer3_sel layer3 selection
13:12	RW	0x2	dsp_layer2_sel layer2 selection
11:10	RW	0x1	dsp_layer1_sel layer1 selection
9:8	RW	0x0	dsp_layer0_sel layer0 selection
7	RO	0x0	reserved
6	RW	0x0	dither_up_en 1'b0 : no dither up 1'b1 : rgb565 dither up to rgb888
5	RO	0x0	reserved
4	RW	0x0	dither_down_sel dither down mode select 2'b0 : allegro 2'b1 : FRC

Bit	Attr	Reset Value	Description
3	RW	0x0	dither_down_mode Dither-down mode 1'b0 : RGB888 to RGB565 1'b1 : RGB888 to RGB666
2	RW	0x0	dither_down_en Dither-down enable 1'b0 : disable 1'b1 : enable
1	RW	0x0	pre_dither_down_en 10bit -> 8bit (allegro)
0	RW	0x0	dsp_lut_en Display LUT ram enable 1'b0 : disable 1'b1 : enable *This bit should be "0" when CPU updates the LUT, and should be "1" when Display LUT mode enable.

**VOP\_DSP\_BG**

Address: Operational Base + offset (0x0018)

Background color

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:16	RW	0x00	dsp_bg_red Background Red color 8bit red color
15:8	RW	0x00	dsp_bg_green Background Green color 8bit green color
7:0	RW	0x00	dsp_bg_blue Background Blue color 8bit blue color

**VOP\_WINO\_CTRL0**

Address: Operational Base + offset (0x0030)

Win0 ctrl register0

Bit	Attr	Reset Value	Description
31:30	RW	0x0	win0_dma_burst_length WIN0 DMA read Burst length 2'b00 : burst16 (burst 15 in rgb888 pack mode) 2'b01 : burst8 (burst 12 in rgb888 pack mode) 2'b10 : burst4 (burst 6 in rgb888 pack mode) 2'b11 : reserved
29:25	RW	0x1d	win0_axi_outstanding_max_num win0 out standing max number

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
24	RW	0x0	win0_axi_max_outstanding_en 1'b0 : disable 1'b1 : enable
23	RO	0x0	reserved
22	RW	0x0	win0_y_mir_en 1'b0 : no y_mirror 1'b1 : y_mirror
21	RW	0x0	win0_x_mir_en 1'b0 : no x_mirror 1'b1 : x_mirror
20	RW	0x0	win0_yuv_clip YCrCb clip 1'b0 : disable, YCbCr no clip 1'b1 : enable, YCbCr clip before YCbCr2RGB *Y clip: 16~235, CbCr clip: 16~239
19	RW	0x0	win0_cbr_deflick Win0 Cbr deflick mode 1'b0 : disable 1'b1 : enable
18	RW	0x0	win0_yrgb_deflick win0 YRGB deflick mode 1'b0 : disable 1'b1 : enable
17	RO	0x0	reserved
16	RW	0x0	win0_hw_pre_mul_en 1'b0: no hardware pre multiply mode 1'b1: hardware pre multiply mode
15	RW	0x0	win0_uv_swap Win0 CbCr swap 1'b0 : CrCb 1'b1 : CbCr
14	RW	0x0	win0_mid_swap Win0 Y middle swap 1'b0 : Y3Y2Y1Y0 1'b1 : Y3Y1Y2Y0
13	RW	0x0	win0_alpha_swap win0 alpha swap 1'b0 : ARGB 1'b1 : RGBA
12	RW	0x0	win0_rb_swap win0 RGB RED and BLUE swap 1'b0 : RGB 1'b1 : BGR

Bit	Attr	Reset Value	Description
11:10	RW	0x0	win0_csc_mode Win0 YUV2RGB or RGB2YUV Color space conversion(YUV2RGB): 2'b00 : mpeg 2'b01 : jpeg 2'b10 : hd 2'b11 : mpeg Color space conversion(RGB2YUV): 2'bx0: BT601 2'bx1: BT709
9	RW	0x0	win0_no_outstanding win0 AXI master read outstanding 1'b0 : enable 1'b1 : disable
8	RW	0x0	win0_interlace_read Win0 interlace read mode 1'b0 : disable 1'b1 : enable
7:5	RW	0x2	win0_lb_mode win0 line buffer mode,calc by driver.
4	RW	0x0	win0_fmt_10 0: yuv 8bit fmt mode 1: yuv 10bit fmt mode
3:1	RW	0x0	win0_data_fmt 3'b000 : ARGB888 3'b001 : RGB888 3'b010 : RGB565 3'b100 : YcbCr420 3'b101 : YcbCr422 3'b110 : YcbCr444
0	RW	0x0	win0_en 1'b0 : disable 1'b1 : enable

**VOP\_WIN0\_CTRL1**

Address: Operational Base + offset (0x0034)

Win0 ctrl register1

Bit	Attr	Reset Value	Description
31	RW	0x0	win0_cbr_vsd_mode win0 vertical scaler down mode select 1'b0 : bilinear 1'b1 : average

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
30	RW	0x0	win0_cbr_vsu_mode win0 vertical scaler down mode select 1'b0 : bilinear 1'b1 : bicubic
29:28	RW	0x0	win0_cbr_hsd_mode win0 horizontal scaler down mode select 2'b00 : bilinear 2'b01 : bicubic 2'b10 : average
27:26	RW	0x0	win0_cbr_ver_scl_mode 2'b00 : no scale 2'b01 : scale up 2'b10 : scale down 2'b11 : no scale
25:24	RW	0x0	win0_cbr_hor_scl_mode 2'b00 : no scale 2'b01 : scale up 2'b10 : scale down 2'b11 : no scale
23	RW	0x0	win0_yrgb_vsd_mode win0 vertical scaler down mode select 1'b0 : bilinear 1'b1 : average
22	RW	0x0	win0_yrgb_vsu_mode win0 vertical scaler down mode select 1'b0 : bilinear 1'b1 : bicubic
21:20	RW	0x0	win0_yrgb_hsd_mode win0 horizontal scaler down mode select 2'b00 : bilinear 2'b01 : average
19:18	RW	0x0	win0_yrgb_ver_scl_mode 2'b00 : no scale 2'b01 : scale up 2'b10 : scale down 2'b11 : no scale
17:16	RW	0x0	win0_yrgb_hor_scl_mode 2'b00 : no scale 2'b01 : scale up 2'b10 : scale down 2'b11 : no scale
15	RW	0x0	win0_line_load_mode when yuv fmt, 1'b0: load data by axi trans 1'b1: load data by lines

Bit	Attr	Reset Value	Description
14:12	RW	0x0	win0_cbr_axi_gather_num win0 axi cbr data transfer gather number
11:8	RW	0x0	win0_yrgb_axi_gather_num win0 axi yrgb data transfer gather number
7	RW	0x0	win0_vsd_cbr_gt2 cbr_src/cbr_dst >= 2
6	RW	0x0	win0_vsd_cbr_gt4 cbr_src/cbr_dst >= 4
5	RW	0x0	win0_vsd_yrgb_gt2 yrgb_src/yrgb_dst >= 2
4	RW	0x0	win0_vsd_yrgb_gt4 yrgb_src/yrgb_dst >= 4
3:2	RW	0x0	win0_bic_coe_sel 2'b00 : PRECISE 2'b01 : SPLINE 2'b10 : CATROM 2'b11 : MITCHELL
1	RW	0x0	win0_cbr_axi_gather_en win0 axi bus cbr data gather transfer enable
0	RW	0x0	win0_yrgb_axi_gather_en win0 axi bus yrgb data gather transfer enable

**VOP\_WIN0\_COLOR\_KEY**

Address: Operational Base + offset (0x0038)

Win0 color key register

Bit	Attr	Reset Value	Description
31	RW	0x0	win0_key_en Win0 transparency color key enable 1'b0 : disable; 1'b1 : enable;
30:24	RO	0x0	reserved
23:0	RW	0x000000	win0_key_color Win0 key color 24 bit RGB888

**VOP\_WIN0\_VIR**

Address: Operational Base + offset (0x003c)

Win0 virtual stride

Bit	Attr	Reset Value	Description
31:16	RW	0x0140	win0_vir_stride_uv Number of words of Win0 uv Virtual width

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
15:0	RW	0x0140	win0_vir_stride Win0 Virtual stride Number of words of Win0 yrgb Virtual width ARGB888 : win0_vir_width RGB888 : (win0_vir_width*3/4) + (win0_vir_width%3) RGB565 : ceil(win0_vir_width/2) YUV : ceil(win0_vir_width/4)

**VOP\_WIN0\_YRGB\_MST**

Address: Operational Base + offset (0x0040)

Win0 YRGB memory start address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	win0_yrgb_mst win0 YRGB frame buffer memory start address

**VOP\_WIN0\_CBR\_MST**

Address: Operational Base + offset (0x0044)

Win0 Cbr memory start address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	win0_cbr_mst win0 CBR frame buffer memory start address

**VOP\_WIN0\_ACT\_INFO**

Address: Operational Base + offset (0x0048)

Win0 active window width/height

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RO	0x0	reserved
28:16	RW	0x00ef	win0_act_height Win0 active(original) window height win_act_height = (win0 vertical size -1)
15:13	RO	0x0	reserved
12:0	RW	0x013f	win0_act_width Win0 active(original) window width win_act_width = (win0 horizontal size -1)

**VOP\_WIN0\_DSP\_INFO**

Address: Operational Base + offset (0x004c)

Win0 display width/height on panel

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:28	RO	0x0	reserved
27:16	RW	0x0ef	win0_dsp_height Win0 display window height win0_dsp_height = (win0 vertical size -1)
15:12	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
11:0	RW	0x13f	win0_dsp_width Win0 display window width win0_dsp_width = (win0 horizontal size -1)

**VOP\_WIN0\_DSP\_ST**

Address: Operational Base + offset (0x0050)

Win0 display start point on panel

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RO	0x0	reserved
28:16	RW	0x000a	win0_dsp_yst Win0 vertical start point(y) of the Panel scanning
15:13	RO	0x0	reserved
12:0	RW	0x000a	win0_dsp_xst Win0 horizontal start point(x) of the Panel scanning

**VOP\_WIN0\_SCL\_FACTOR\_YRGB**

Address: Operational Base + offset (0x0054)

Win0 YRGB scaling factor

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x1000	win0_vs_factor_yrgb Win0 YRGB vertical scaling factor: factor=((LCD_C_WIN0_ACT_INFO[31:16]) /(LCD_C_WIN0_DSP_INFO[31:16]))*2^12
15:0	RW	0x1000	win0_hs_factor_yrgb Win0 YRGB horizontal scaling factor: factor=((LCD_C_WIN0_ACT_INFO[15:0]) /(LCD_C_WIN0_DSP_INFO[15:0]))*2^12

**VOP\_WIN0\_SCL\_FACTOR\_CBR**

Address: Operational Base + offset (0x0058)

Win0 Cbr scaling factor

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x1000	win0_vs_factor_cbr Win0 CBR vertical scaling factor: YCbCr420: factor=((LCD_C_WIN0_ACT_INFO[31:16]/ 2) /(LCD_C_WIN0_DSP_INFO[31:16] ))*2^12 YCbCr422,YCbCr444: factor=((LCD_C_WIN0_ACT_INFO[31:16]) /(LCD_C_WIN0_DSP_INFO[31:16] ))*2^12

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
15:0	RW	0x1000	win0_hs_factor_cbr Win0 CBR horizontal scaling factor: YCbCr422,YCbCr420: factor=((LCD_C_WIN0_ACT_INFO[15:0]/2) /(LCD_C_WIN0_DSP_INFO[15:0]))*2^12 YCbCr444: factor=((LCD_C_WIN0_ACT_INFO[15:0]) /(LCD_C_WIN0_DSP_INFO[15:0]))*2^12

**VOP\_WIN0\_SCL\_OFFSET**

Address: Operational Base + offset (0x005c)

Win0 scaling start point offset

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RW	0x00	win0_vs_offset_cbr Cbr Vertical scaling start point offset (0x00~0xff)/0x100 = 0~0.99
23:16	RW	0x00	win0_vs_offset_yrgb Y Vertical scaling start point offset (0x00~0xff)/0x100 = 0~0.99
15:8	RW	0x00	win0_hs_offset_cbr Cbr Horizontal scaling start point offset (0x00~0xff)/0x100 = 0~0.99
7:0	RW	0x00	win0_hs_offset_yrgb Y Horizontal scaling start point offset (0x00~0xff)/0x100 = 0~0.99

**VOP\_WIN0\_SRC\_ALPHA\_CTRL**

Address: Operational Base + offset (0x0060)

Win0 alpha source control register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RW	0x00	win0_fading_value win0 fading value ,8bits
23:16	RW	0x00	win0_src_global_alpha src global alpha
15:9	RO	0x0	reserved
8:6	RW	0x0	win0_src_factor_mode src factor mode
5	RW	0x0	win0_src_alpha_cal_mode src alpha calc mode
4:3	RW	0x0	win0_src_blend_mode src blend mode
2	RW	0x0	win0_src_alpha_mode src alpha mode
1	RW	0x0	win0_src_color_mode src color mode

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
0	RW	0x0	win0_src_alpha_en src alpha en

**VOP\_WIN0\_DST\_ALPHA\_CTRL**

Address: Operational Base + offset (0x0064)

Win0 alpha destination control register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:9	RO	0x0	reserved
8:6	RW	0x0	win0_dst_factor_mode dst factor mode
5:0	RW	0x00	win0_dst_m0_reserved reserved

**VOP\_WIN0\_FADING\_CTRL**

Address: Operational Base + offset (0x0068)

Win0 fading control register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:25	RO	0x0	reserved
24	RW	0x0	layer0_fading_en fading enable
23:16	RW	0x00	layer0_fading_offset_b fading offset blue value
15:8	RW	0x00	layer0_fading_offset_g fading offset green value
7:0	RW	0x00	layer0_fading_offset_r fading offset red value

**VOP\_WIN0\_CTRL2**

Address: Operational Base + offset (0x006c)

Win0 ctrl register2

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:8	RO	0x0	reserved
7:4	RW	0x2	win_rid_win0_cbr axi read id of win0 cbr channel
3:0	RW	0x1	win_rid_win0_yrgb axi read id of win0 yrgb channel

**VOP\_WIN1\_CTRL0**

Address: Operational Base + offset (0x0070)

Win1 ctrl register0

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RW	0x0	win1_dma_burst_length WIN1 DMA read Burst length 2'b00 : burst16 (burst 15 in rgb888 pack mode) 2'b01 : burst8 (burst 12 in rgb888 pack mode) 2'b10 : burst4 (burst 6 in rgb888 pack mode) 2'b11 : reserved
29:25	RW	0x1d	win1_axi_max_outstanding_num win1 out standing max number
24	RW	0x0	win1_axi_max_outstanding_en 1'b0 : disable 1'b1 : enable
23	RO	0x0	reserved
22	RW	0x0	win1_y_mir_en 1'b0 : no y_mirror 1'b1 : y_mirror
21	RW	0x0	win1_x_mir_en 1'b0 : no x_mirror 1'b1 : x_mirror
20	RW	0x0	win1_yuv_clip YCrCb clip 1'b0 : disable, YCbCr no clip 1'b1 : enable, YCbCr clip before YCbCr2RGB *Y clip: 16~235, CbCr clip: 16~239
19	RW	0x0	win1_cbr_deflick Win1 Cbr deflick mode 1'b0 : disable 1'b1 : enable
18	RW	0x0	win1_yrgb_deflick win1 YRGB deflick mode 1'b0 : disable 1'b1 : enable
17	RO	0x0	reserved
16	RW	0x0	win1_hw_pre_mul_en 1'b0: no hardware pre multiply mode 1'b1: hardware pre multiply mode
15	RW	0x0	win1_uv_swap Win1 CbCr swap 1'b0 : CrCb 1'b1 : CbCr
14	RW	0x0	win1_mid_swap Win1 Y middle 8-bit swap 1'b0 : Y3Y2Y1Y0 1'b1 : Y3Y1Y2Y0

Bit	Attr	Reset Value	Description
13	RW	0x0	win1_alpha_swap win1 alpha swap 1'b0 : ARGB 1'b1 : RGBA
12	RW	0x0	win1_rb_swap win1 RGB RED and BLUE swap 1'b0 : RGB 1'b1 : BGR
11:10	RW	0x0	win1_csc_mode Win1 YUV2RGB or RGB2YUV Color space conversion(YUV2RGB): 2'b00 : mpeg 2'b01 : jpeg 2'b10 : hd 2'b11 : mpeg Color space conversion(RGB2YUV): 2'bx0: BT601 2'bx1: BT709
9	RW	0x0	win1_no_outstanding win1 AXI master read outstanding 1'b0 : enable 1'b1 : disable
8	RW	0x0	win1_interlace_read Win1 interlace read mode 1'b0 : disable 1'b1 : enable
7:5	RW	0x2	win1_lb_mode win1 line buffer mode,calc by driver.
4	RW	0x0	win1_fmt_10 1'b0: yuv 8bit fmt mode 1'b1: yuv 10bit fmt mode
3:1	RW	0x0	win1_data_fmt 3'b000 : ARGB888 3'b001 : RGB888 3'b010 : RGB565 3'b100 : YcbCr420 3'b101 : YcbCr422 3'b110 : YcbCr444
0	RW	0x0	win1_en 1'b0 : disable 1'b1 : enable

**VOP\_WIN1\_CTRL1**

Address: Operational Base + offset (0x0074)  
Win1 ctrl register1

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RW	0x0	win1_cbr_vsd_mode win1 vertical scaler down mode select 1'b0 : bilinear 1'b1 : average
30	RW	0x0	win1_cbr_vsu_mode win1 vertical scaler up mode select 1'b0 : bilinear 1'b1 : bicubic
29:28	RW	0x0	win1_cbr_hsd_mode win1 horizontal scaler down mode select 2'b00 : bilinear 2'b01 : bicubic 2'b10 : average
27:26	RW	0x0	win1_cbr_ver_scl_mode 2'b00 : no scale 2'b01 : scale up 2'b10 : scale down 2'b11 : no scale
25:24	RW	0x0	win1_cbr_hor_scl_mode 2'b00 : no scale 2'b01 : scale up 2'b10 : scale down 2'b11 : no scale
23	RW	0x0	win1_yrgb_vsd_mode win1 vertical scaler down mode select 1'b0 : bilinear 1'b1 : average
22	RW	0x0	win1_yrgb_vsu_mode win1 vertical scaler up mode select 1'b0 : bilinear 1'b1 : bicubic
21:20	RW	0x0	win1_yrgb_hsd_mode win1 horizontal scaler down mode select 2'b00 : bilinear 2'b01 : average
19:18	RW	0x0	win1_yrgb_ver_scl_mode 2'b00 : no scale 2'b01 : scale up 2'b10 : scale down 2'b11 : no scale
17:16	RW	0x0	win1_yrgb_hor_scl_mode 2'b00 : no scale 2'b01 : scale up 2'b10 : scale down 2'b11 : no scale

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
15	RW	0x0	win1_line_load_mode when yuv fmt, 1'b0: load data by pixels 1'b1: load data by lines
14:12	RW	0x0	win1_cbr_axi_gather_num win1 axi cbr data transfer gather number
11:8	RW	0x0	win1_yrgb_axi_gather_num win1 axi yrgb data transfer gather number
7	RW	0x0	win1_vsd_cbr_gt2 cbr_src/cbr_dst >= 2
6	RW	0x0	win1_vsd_cbr_gt4 cbr_src/cbr_dst >= 4
5	RW	0x0	win1_vsd_yrgb_gt2 yrgb_src/yrgb_dst >= 2
4	RW	0x0	win1_vsd_yrgb_gt4 yrgb_src/yrgb_dst >= 4
3:2	RW	0x0	win1_bic_coe_sel 2'b00 : PRECISE 2'b01 : SPLINE 2'b10 : CATROM 2'b11 : MITCHELL
1	RW	0x0	win1_cbr_axi_gather_en win1 cbr axi bus gather enable
0	RW	0x0	win1_yrgb_axi_gather_en win1 yrgb axi bus gather enable

**VOP\_WIN1\_COLOR\_KEY**

Address: Operational Base + offset (0x0078)

Win1 color key register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RW	0x0	win1_key_en Win1 transparency color key enable 1'b0 : disable; 1'b1 : enable;
30:24	RO	0x0	reserved
23:0	RW	0x000000	win1_key_color Win1 key color 24 bit RGB888

**VOP\_WIN1\_VIR**

Address: Operational Base + offset (0x007c)

win1 virtual stride

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0140	win1_vir_stride_uv Number of words of Win1 uv Virtual width

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
15:0	RW	0x0140	win1_vir_stride Win1 Virtual stride Number of words of Win1 yrgb Virtual width ARGB888 : win1_vir_width RGB888 : (win1_vir_width*3/4) + (win1_vir_width%3) RGB565 : ceil(win1_vir_width/2) YUV : ceil(win1_vir_width/4)

**VOP\_WIN1\_YRGB\_MST**

Address: Operational Base + offset (0x0080)

Win1 YRGB memory start address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	win1_yrgb_mst win1 YRGB frame buffer memory start address

**VOP\_WIN1\_CBR\_MST**

Address: Operational Base + offset (0x0084)

Win1 Cbr memory start address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	win1_cbr_mst win1 CBR frame buffer memory start address

**VOP\_WIN1\_ACT\_INFO**

Address: Operational Base + offset (0x0088)

Win1 active window width/height

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RO	0x0	reserved
28:16	RW	0x00ef	win1_act_height Win1 active(original) window height win_act_height = (win1 vertical size -1)
15:13	RO	0x0	reserved
12:0	RW	0x013f	win1_act_width Win1 active(original) window width win_act_width = (win1 horizontal size -1)

**VOP\_WIN1\_DSP\_INFO**

Address: Operational Base + offset (0x008c)

Win1 display width/height on panel

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:28	RO	0x0	reserved
27:16	RW	0x0ef	win1_dsp_height Win1 display window height win1_dsp_height = (win1 vertical size -1)
15:12	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
11:0	RW	0x13f	win1_dsp_width Win1 display window width win1_dsp_width = (win1 horizontal size - 1)

**VOP\_WIN1\_DSP\_ST**

Address: Operational Base + offset (0x0090)

Win1 display start point on panel

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RO	0x0	reserved
28:16	RW	0x000a	win1_dsp_yst Win1 vertical start point(y) of the Panel scanning
15:13	RO	0x0	reserved
12:0	RW	0x000a	win1_dsp_xst Win1 horizontal start point(x) of the Panel scanning

**VOP\_WIN1\_SCL\_FACTOR\_YRGB**

Address: Operational Base + offset (0x0094)

Win1 YRGB scaling factor

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x1000	win1_vs_factor_yrgb Win1 YRGB vertical scaling factor: factor=((LCD_C_WIN1_ACT_INFO[31:16]) /(LCD_C_WIN1_DSP_INFO[31:16]))*2^12
15:0	RW	0x1000	win1_hs_factor_yrgb Win1 YRGB horizontal scaling factor: factor=((LCD_C_WIN1_ACT_INFO[15:0]) /(LCD_C_WIN1_DSP_INFO[15:0]))*2^12

**VOP\_WIN1\_SCL\_FACTOR\_CBR**

Address: Operational Base + offset (0x0098)

Win1 Cbr scaling factor

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x1000	win1_vs_factor_cbr Win1 CBR vertical scaling factor: YCbCr420: factor=((LCD_C_WIN1_ACT_INFO[31:16]/ 2) /(LCD_C_WIN1_DSP_INFO[31:16] ))*2^12 YCbCr422,YCbCr444: factor=((LCD_C_WIN1_ACT_INFO[31:16]) /(LCD_C_WIN1_DSP_INFO[31:16] ))*2^12

Bit	Attr	Reset Value	Description
15:0	RW	0x1000	win1_hs_factor_cbr Win1 Cbr horizontal scaling factor: YCbCr422,YCbCr420: $\text{factor} = ((\text{LCDC\_WIN1\_ACT\_INFO}[15:0]/2) / (\text{LCDC\_WIN1\_DSP\_INFO}[15:0])) * 2^{12}$ YCbCr444: $\text{factor} = ((\text{LCDC\_WIN1\_ACT\_INFO}[15:0]) / (\text{LCDC\_WIN1\_DSP\_INFO}[15:0])) * 2^{12}$

**VOP\_WIN1\_SCL\_OFFSET**

Address: Operational Base + offset (0x009c)

Win1 scaling start point offset

Bit	Attr	Reset Value	Description
31:24	RW	0x00	win1_vs_offset_cbr Cbr Vertical scaling start point offset $(0x00 \sim 0xff)/0x100 = 0 \sim 0.99$
23:16	RW	0x00	win1_vs_offset_yrgb Y Vertical scaling start point offset $(0x00 \sim 0xff)/0x100 = 0 \sim 0.99$
15:8	RW	0x00	win1_hs_offset_cbr Cbr Horizontal scaling start point offset $(0x00 \sim 0xff)/0x100 = 0 \sim 0.99$
7:0	RW	0x00	win1_hs_offset_yrgb Y Horizontal scaling start point offset $(0x00 \sim 0xff)/0x100 = 0 \sim 0.99$

**VOP\_WIN1\_SRC\_ALPHA\_CTRL**

Address: Operational Base + offset (0x00a0)

Win1 alpha source control register

Bit	Attr	Reset Value	Description
31:24	RW	0x00	win1_fading_value fading value,8bit
23:16	RW	0x00	win1_src_global_alpha src global alpha
15:9	RO	0x0	reserved
8:6	RW	0x0	win1_src_factor_mode src factor mode
5	RW	0x0	win1_src_alpha_cal_mode src alpha calc mode
4:3	RW	0x0	win1_src_blend_mode src blend mode
2	RW	0x0	win1_src_alpha_mode src alpha mode

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
1	RW	0x0	win1_src_color_mode src color mode
0	RW	0x0	win1_src_alpha_en src alpha en

**VOP\_WIN1\_DST\_ALPHA\_CTRL**

Address: Operational Base + offset (0x00a4)

Win1 alpha destination control register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:9	RO	0x0	reserved
8:6	RW	0x0	win1_dst_factor_m0 dst factor mode
5:0	RW	0x00	win1_dsp_m0_reserved reserved

**VOP\_WIN1\_FADING\_CTRL**

Address: Operational Base + offset (0x00a8)

Win1 fading contrl register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:25	RO	0x0	reserved
24	RW	0x0	win1_fading_en fading enable
23:16	RW	0x00	win1_fading_offset_b fading offset blue value
15:8	RW	0x00	win1_fading_offset_g fading offset green value
7:0	RW	0x00	win1_fading_offset_r fading offset red value

**VOP\_WIN1\_CTRL2**

Address: Operational Base + offset (0x00ac)

Win1 ctrl register2

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:8	RO	0x0	reserved
7:4	RW	0x4	win_rid_win1_cbr axi read id of win1 cbr channel
3:0	RW	0x3	win_rid_win1_yrgb axi read id of win1 yrgb channel

**VOP\_HWC\_CTRL0**

Address: Operational Base + offset (0x0150)

Hwc ctrl register0

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:15	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
14	RW	0x0	hwc_endian_swap hwc 8pp palette data Big-endian/ Little-endian select 1'b0 : Big-endian 1'b1 : Little-endian
13	RW	0x0	hwc_alpha_swap hwc RGB alpha swap 1'b0 : ARGB 1'b1 : RGBA
12	RW	0x0	hwc_rb_swap hwc RGB Red and Blue swap 1'b0 : RGB 1'b1 : BGR
11	RO	0x0	reserved
10	RW	0x0	hwc_csc_mode hwc RGB2YUV Color space conversion: 2'bx0: BT601 2'bx1: BT709
9	RO	0x0	reserved
8	RW	0x0	hwc_interlace_read hwc interlace read mode 1'b0 : disable 1'b1 : enable
7	RO	0x0	reserved
6:5	RW	0x0	hwc_size 2'b00 : 32x32 2'b01 : 64x64 2'b10 : 96x96 2'b11 : 128x128
4	RW	0x0	hwc_mode hwc color mode 1'b0 : normal color mode 1'b1 : reversed color mode
3:1	RW	0x0	hwc_data_fmt 3'b000 : ARGB888 3'b001 : RGB888 3'b010 : RGB565 3'b100: 8bpp 3'b101: 4bpp 3'b110: 2bpp 3'b111: 1bpp
0	RW	0x0	hwc_en 1'b0 : disable 1'b1 : enable

**VOP\_HWC\_CTRL1**

Address: Operational Base + offset (0x0154)

Hwc ctrl register1

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x0	reserved
23:20	RW	0x7	win_rid_hwci axi read id of hwc channel
19:17	RO	0x0	reserved
16	RW	0x0	hwc_lut_en 1'b0 : disable 1'b1 : enable
15	RW	0x0	hwc_y_mir_en 1'b0 : no y_mirror 1'b1 : y_mirror
14	RW	0x0	hwc_no_outstanding hwc AXI master read outstanding 1'b0 : enable 1'b1 : disable
13	RW	0x0	hwc_rgb2yuv_en 1'b0 : enable 1'b1 : disable
12:8	RW	0x1d	hwc_axi_max_outstanding_num hwc axi bus max outstanding number
7	RO	0x0	reserved
6:4	RW	0x0	hwc_axi_gather_num hwc axi gather transfer number
3:2	RW	0x0	hwc_dma_burst_length HWC DMA read Burst length 2'b00 : burst16 (burst 15 in rgb888 pack mode) 2'b01 : burst8 (burst 12 in rgb888 pack mode) 2'b10 : burst4 (burst 6 in rgb888 pack mode) 2'b11 : reserved
1	RW	0x0	hwc_axi_max_outstanding_en 1'b0 : disable 1'b1 : enable
0	RW	0x0	hwc_axi_gather_en 1'b0 : disable 1'b1 : enable

**VOP\_HWC\_MST**

Address: Operational Base + offset (0x0158)

Hwc memory start address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	hwc_mst HWC data memory start address

**VOP\_HWC\_DSP\_ST**

Address: Operational Base + offset (0x015c)

Hwc display start point on panel

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RO	0x0	reserved
28:16	RW	0x000a	hwc_dsp_yst HWC vertical start point(y) of the Panel scanning
15:13	RO	0x0	reserved
12:0	RW	0x000a	hwc_dsp_xst HWC horizontal start point(x) of the Panel scanning

**VOP\_HWC\_SRC\_ALPHA\_CTRL**

Address: Operational Base + offset (0x0160)

Hwc alpha source control register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RW	0x00	hwc_fading_value fading value
23:16	RW	0x00	hwc_src_global_alpha src global alpha
15:9	RO	0x0	reserved
8:6	RW	0x0	hwc_src_factor_mode src factor mode
5	RW	0x0	hwc_src_alpha_cal_mode src alpha calc mode
4:3	RW	0x0	hwc_src_blend_mode src blend mode
2	RW	0x0	hwc_src_alpha_mode src alpha mode
1	RW	0x0	hwc_src_color_mode src color mode
0	RW	0x0	hwc_src_alpha_en src alpha enable

**VOP\_HWC\_DST\_ALPHA\_CTRL**

Address: Operational Base + offset (0x0164)

Hwc alpha destination control register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:9	RO	0x0	reserved
8:6	RW	0x0	hwc_dst_factor_mode dst factor mode
5:0	RW	0x00	hwc_dst_m0_reserved reserved

**VOP\_HWC\_FADING\_CTRL**

Address: Operational Base + offset (0x0168)

Hwc fading contrl register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:25	RO	0x0	reserved
24	RW	0x0	hwc_fading_en 1'b0 : disable 1'b1 : enable
23:16	RW	0x00	hwc_fading_offset_b fading offset blue
15:8	RW	0x00	hwc_fading_offset_g fading offset green
7:0	RW	0x00	hwc_fading_offset_r fading offset red

**VOP\_HWC\_RESERVED1**

Address: Operational Base + offset (0x016c)

Hwc reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**VOP\_POST\_DSP\_HACT\_INFO**

Address: Operational Base + offset (0x0170)

Post scaler down horizontal start and end

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RO	0x0	reserved
28:16	RW	0x000a	dsp_hact_st_post Panel display scanning horizontal active start point
15:13	RO	0x0	reserved
12:0	RW	0x014a	dsp_hact_end_post Panel display scanning horizontal active end point

**VOP\_POST\_DSP\_VACT\_INFO**

Address: Operational Base + offset (0x0174)

Panel active horizontal scanning start point and end point

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RO	0x0	reserved
28:16	RW	0x000a	dsp_vact_st_post Panel display scanning horizontal active start point
15:13	RO	0x0	reserved
12:0	RW	0x00fa	dsp_vact_end_post Panel display scanning horizontal active end point

**VOP\_POST\_SCL\_FACTOR\_YRGB**

Address: Operational Base + offset (0x0178)

Post yrgb scaling factor

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x1000	post_vs_factor_yrgb post YRGB vertical scaling factor: factor=((src_height[31:16]) /(dst_height[31:16]))*2^12
15:0	RW	0x1000	post_hs_factor_yrgb Post YRGB horizontal scaling factor: factor=((src_width[15:0]) /(dst_width[15:0]))*2^12

**VOP\_POST\_RESERVED**

Address: Operational Base + offset (0x017c)

Post reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**VOP\_POST\_SCL\_CTRL**

Address: Operational Base + offset (0x0180)

Post scaling start point offset

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RW	0x0	post_ver_sd_en 1'b0 : post ver scl down disable 1'b1 : post ver scl down enable
0	RW	0x0	post_hor_sd_en 1'b0 : post hor scl down disable 1'b1 : post hor scl down enable

**VOP\_POST\_DSP\_VACT\_INFO\_F1**

Address: Operational Base + offset (0x0184)

Panel active horizontal scanning start point and end point F1

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RO	0x0	reserved
28:16	RW	0x000a	dsp_vact_st_post Panel display scanning horizontal active start point
15:13	RO	0x0	reserved
12:0	RW	0x00fa	dsp_vact_end_post Panel display scanning horizontal active end point

**VOP\_DSP\_HTOTAL\_HS\_END**

Address: Operational Base + offset (0x0188)

Panel scanning horizontal width and hsync pulse end point

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RO	0x0	reserved
28:16	RW	0x014a	dsp_htotal Panel display scanning horizontal period

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
15:13	RO	0x0	reserved
12:0	RW	0x000a	dsp_hs_end Panel display scanning hsync pulse width

**VOP\_DSP\_HACT\_ST\_END**

Address: Operational Base + offset (0x018c)

Panel active horizontal scanning start point and end point

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RO	0x0	reserved
28:16	RW	0x000a	dsp_hact_st Panel display scanning horizontal active start point
15:13	RO	0x0	reserved
12:0	RW	0x014a	dsp_hact_end Panel display scanning horizontal active end point

**VOP\_DSP\_VTOTAL\_VS\_END**

Address: Operational Base + offset (0x0190)

Panel scanning vertical height and vsync pulse end point

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RO	0x0	reserved
28:16	RW	0x00fa	dsp_vtotal Panel display scanning vertical period.
15	RW	0x0	sw_dsp_vtotal_imd dsp vtotal number valid immediately enable. 1'b0 : valid after frame start 1'b1 : valid immediately
14:13	RO	0x0	reserved
12:0	RW	0x000a	dsp_vs_end Panel display scanning vsync pulse width

**VOP\_DSP\_VACT\_ST\_END**

Address: Operational Base + offset (0x0194)

Panel active vertical scanning start point and end point

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RO	0x0	reserved
28:16	RW	0x000a	dsp_vact_st Panel display scanning vertical active start point
15:13	RO	0x0	reserved
12:0	RW	0x00fa	dsp_vact_end Panel display scanning vertical active end point

**VOP\_DSP\_VS\_ST\_END\_F1**

Address: Operational Base + offset (0x0198)

Vertical scanning start point and vsync pulse end point of even field in interlace mode

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RO	0x0	reserved
28:16	RW	0x0000	dsp_vs_st_f1 Panel display scanning vertical vsync start point of 2nd field (interlace display mode)
15:13	RO	0x0	reserved
12:0	RW	0x0000	dsp_vs_end_f1 Panel display scanning vertical vsync end point of 2nd field(interlace display mode)

**VOP\_DSP\_VACT\_ST\_END\_F1**

Address: Operational Base + offset (0x019c)

Vertical scanning active start point and end point of even filed in interlace mode

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RO	0x0	reserved
28:16	RW	0x0000	dsp_vact_st_f1 Panel display scanning vertical active start point of 2nd field (interlace display mode)
15:13	RO	0x0	reserved
12:0	RW	0x0000	dsp_vact_end_f1 Panel display scanning vertical active end point of 2nd field (interlace display mode)

**VOP\_BCSH\_COLOR\_BAR**

Address: Operational Base + offset (0x01b0)

Color bar config register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RW	0x00	color_bar_v v color value
23:16	RW	0x00	color_bar_u u color value
15:8	RW	0x00	color_bar_y y color value
7:1	RO	0x0	reserved
0	RW	0x0	bcs_sh_en 1'b0 : bcs bypass 1'b1 : bcs enable

**VOP\_BCSH\_BCS**

Address: Operational Base + offset (0x01b4)

Brightness contrast saturation\*contrast config register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RW	0x3	out_mode video out mode config register 2'b00 : black 2'b01 : blue 2'b10 : color bar 2'b11 : normal video
29:20	RW	0x100	sat_con Saturation*Contrast*256 : 0,1.992*1.992
19:17	RO	0x0	reserved
16:8	RW	0x100	contrast Contrast*256 : 0,1.992
7:0	RW	0x00	brightness Brightness : -32,31

**VOP\_BCSH\_H**

Address: Operational Base + offset (0x01b8)

Sin hue and cos hue config register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:25	RO	0x0	reserved
24:16	RW	0x100	cos_hue cos hue value
15:9	RO	0x0	reserved
8:0	RW	0x000	sin_hue sin hue value

**VOP\_BCSH\_CTRL**

Address: Operational Base + offset (0x01bc)

BCSH contrl register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:7	RO	0x0	reserved
6	RW	0x0	bcth_r2y_csc_mode Color space conversion: 1'b0: BT601 1'b1: BT709
5	RO	0x0	reserved
4	RW	0x0	bcth_r2y_en 1'b0:bypass 1'b1:enable
3:2	RW	0x0	bcth_y2r_csc_mode Color space conversion(YUV2RGB): 2'b00/01 : mpeg 2'b10 : jpeg 2'b11 : hd
1	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
0	RW	0x0	bcsy_y2r_en 1'b0:bypass 1'b1:enable

**VOP\_FRC\_LOWER01\_0**

Address: Operational Base + offset (0x01e8)

FRC lookup table config register010

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x1284	lower01_frm1 frc parameter lowerbit = 2'b01,frm1
15:0	RW	0x4821	lower01_frm0 frc parameter lowerbit = 2'b01,frm0

**VOP\_FRC\_LOWER01\_1**

Address: Operational Base + offset (0x01ec)

FRC lookup table config register011

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x2148	lower01_frm3 frc parameter lowerbit = 2'b01,frm3
15:0	RW	0x8412	lower01_frm2 frc parameter lowerbit = 2'b01,frm2

**VOP\_FRC\_LOWER10\_0**

Address: Operational Base + offset (0x01f0)

FRC lookup table config register100

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0xa55a	lower10_frm1 frc parameter lowerbit = 2'b10,frm1
15:0	RW	0x9696	lower10_frm0 frc parameter lowerbit = 2'b10,frm0

**VOP\_FRC\_LOWER10\_1**

Address: Operational Base + offset (0x01f4)

FRC lookup table config register101

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x5aa5	lower10_frm3 frc parameter lowerbit = 2'b10,frm3
15:0	RW	0x6969	lower10_frm2 frc parameter lowerbit = 2'b10,frm2

**VOP\_FRC\_LOWER11\_0**

Address: Operational Base + offset (0x01f8)

FRC lookup table config register110

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0xdeb7	lower11_frm1 frc parameter lowerbit = 2'b11,frm1
15:0	RW	0x7bed	lower11_frm0 frc parameter lowerbit = 2'b11,frm0

**VOP\_FRC\_LOWER11\_1**

Address: Operational Base + offset (0x01fc)

FRC lookup table config register111

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0xed7b	lower11_frm3 frc parameter lowerbit = 2'b11,frm3
15:0	RW	0xb7de	lower11_frm2 frc parameter lowerbit = 2'b11,frm2

**VOP\_INTR\_EN0**

Address: Operational Base + offset (0x0280)

Interrupt enable register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	intr_en_dma_finish 1'b0: disable 1'b1: enable
14	RW	0x0	intr_en_mmu 1'b0: disable 1'b1: enable
13	RW	0x0	intr_en_dsp_hold_valid display hold valid interrupt enable 1'b0: disable 1'b1: enable
12	RW	0x0	intr_en_pwm_gen pwm generated interrupt enable 1'b0: Channel 0 Interrupt not generated 1'b1: Channel 0 Interrupt generated
11	RW	0x0	intr_en_post_buf_empty post buffer empty interrupt enable 1'b0: disable 1'b1: enable
10	RW	0x0	intr_en_hwc_empty hwc data empty interrupt enable 1'b0: disable 1'b1: enable

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
9	RW	0x0	intr_en_win3_empty win3 data empty interrupt enable 1'b0: disable 1'b1: enable
8	RW	0x0	intr_en_win2_empty win2 data empty interrupt enable 1'b0: disable 1'b1: enable
7	RW	0x0	intr_en_win1_empty win1 data empty interrupt enable 1'b0: disable 1'b1: enable
6	RW	0x0	intr_en_win0_empty win0 data empty interrupt enable 1'b0: disable 1'b1: enable
5	RW	0x0	intr_en_bus_error Bus error Interrupt enable 1'b0: disable 1'b1: enable
4	RW	0x0	intr_en_line_flag1 Line flag 1 Interrupt enable 1'b0: disable 1'b1: enable
3	RW	0x0	intr_en_line_flag0 Line flag 0 Interrupt enable 1'b0: disable 1'b1: enable
2	RW	0x0	intr_en_addr_same memory start addr same interruption enable 1'b0: disable 1'b1: enable
1	RW	0x0	intr_en_fs_new Frame new start interrupt enable 1'b0: disable 1'b1: enable
0	RW	0x0	intr_en_fs Frame start interrupt enable 1'b0: disable 1'b1: enable

**VOP\_INTR\_CLEAR0**

Address: Operational Base + offset (0x0284)

Interrupt clear register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
------------	-------------	--------------------	--------------------

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	write_mask When every bit HIGH, enable the writing corresponding bit When every bit LOW, don't care the writing corresponding bit
15	RW	0x0	int_clr_dma_finish dma finish interrupt clear(Auto clear)
14	RW	0x0	int_clr_mmu mmu interrupt clear(Auto clear)
13	W1C	0x0	int_clr_dsp_hold_valid display hold valid interrupt clear(Auto clear)
12	W1C	0x0	int_clr_pwm_gen pwm generated interrupt clear(Auto clear) 1'b0: Channel 0 Interrupt not generated 1'b1: Channel 0 Interrupt generated
11	W1C	0x0	int_clr_post_buf_empty post buffer empty interrupt clear(Auto clear)
10	W1C	0x0	int_clr_hwc_empty hwc data empty interrupt clear(Auto clear)
9	W1C	0x0	int_clr_win3_empty win3 data empty interrupt clear(Auto clear)
8	W1C	0x0	int_clr_win2_empty win2 data empty interrupt clear(Auto clear)
7	W1C	0x0	int_clr_win1_empty win1 data empty interrupt clear(Auto clear)
6	W1C	0x0	int_clr_win0_empty win0 data empty interrupt clear(Auto clear)
5	W1C	0x0	int_clr_bus_error Bus error Interrupt clear(Auto clear)
4	W1C	0x0	int_clr_line_flag1 Line flag 1 Interrupt clear(Auto clear)
3	W1C	0x0	int_clr_line_flag0 Line flag 0 Interrupt clear(Auto clear)
2	W1C	0x0	int_clr_addr_same memory start addr same interruption clear(Auto clear)
1	W1C	0x0	int_clr_fs_new Frame new start interrupt clear (Auto clear)
0	W1C	0x0	int_clr_fs Frame start interrupt clear (Auto clear)

**VOP\_INTR\_STATUS0**

Address: Operational Base + offset (0x0288)  
interrupt status

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RO	0x0	reserved
15	RW	0x0	int_status_dma_finish dma finish interrupt status

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
14	RW	0x0	int_status_mmu mmu interrupt status
13	RO	0x0	int_status_dsp_hold_valid display hold valid interrupt status
12	RO	0x0	int_status_pwm_gen pwm generated interrupt status 1'b0: Channel 0 Interrupt not generated 1'b1: Channel 0 Interrupt generated
11	RO	0x0	int_status_post_buf_empty post buffer empty interrupt status
10	RO	0x0	int_status_hwc_empty hwc data empty interrupt status
9	RO	0x0	int_status_win3_empty win3 data empty interrupt status
8	RO	0x0	int_status_win2_empty win2 data empty interrupt status
7	RO	0x0	int_status_win1_empty win1 data empty interrupt status
6	RO	0x0	int_status_win0_empty win0 data empty interrupt status
5	RO	0x0	int_status_bus_error Bus error Interrupt status
4	RO	0x0	int_status_line_flag1 Line flag 1 Interrupt status
3	RO	0x0	int_status_line_flag0 Line flag 0 Interrupt status
2	RW	0x0	int_status_addr_same memory start addr same interruption status
1	RO	0x0	int_status_fs_new Frame start interrupt status(when memory start addr are same,no interruption)
0	RO	0x0	int_status_fs Frame start interrupt status

**VOP\_INTR\_RAW\_STATUS0**

Address: Operational Base + offset (0x028c)

raw interrupt status

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RO	0x0	reserved
15	RO	0x0	int_raw_status_dma_finish dma finish interrupt raw status
14	RO	0x0	int_raw_status_mmu mmu interrupt raw status
13	RO	0x0	int_raw_status_dsp_hold_valid display hold valid interrupt raw status

Bit	Attr	Reset Value	Description
12	RO	0x0	int_raw_status_pwm_gen pwm generated interrupt raw status 1'b0: Channel 0 Interrupt not generated 1'b1: Channel 0 Interrupt generated
11	RO	0x0	int_raw_status_post_buf_empty post buffer empty interrupt raw status
10	RO	0x0	int_raw_status_hwc_empty hwc data empty interrupt raw status
9	RO	0x0	int_raw_status_win3_empty win3 data empty interrupt raw status
8	RO	0x0	int_raw_status_win2_empty win2 data empty interrupt raw status
7	RO	0x0	int_raw_status_win1_empty win1 data empty interrupt raw status
6	RO	0x0	int_raw_status_win0_empty win0 data empty interrupt raw status
5	RO	0x0	int_raw_status_bus_error Bus error Interrupt raw status
4	RO	0x0	int_raw_status_line_frag1 Line flag 1 Interrupt raw status
3	RO	0x0	int_raw_status_line_frag0 Line flag 0 Interrupt raw status
2	RO	0x0	int_raw_status_addr_same memory start addr same interruption raw status
1	RO	0x0	int_raw_status_fs_new Frame start interrupt raw status(when memory start addr are same)
0	RO	0x0	int_raw_status_fs Frame start raw interrupt status Frame start raw interrupt status

**VOP\_INTR\_EN1**

Address: Operational Base + offset (0x0290)

Interrupt enable register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	write_mask
15	RO	0x0	reserved
14	RW	0x0	int_en_wb_finish 1'b0: disable 1'b1: enable
13	RW	0x0	int_en_wb_uv_fifo_full 1'b0: disable 1'b1: enable

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
12	RW	0x0	int_en_wb_yrgb_fifo_full 1'b0: disable 1'b1: enable
11	RW	0x0	int_en_afbcd3_hreg_axi_rresp 1'b0: disable 1'b1: enable
10	RW	0x0	int_en_afbcd3_hreg_dec_resp 1'b0: disable 1'b1: enable
9	RW	0x0	int_en_afbcd2_hreg_axi_rresp 1'b0: disable 1'b1: enable
8	RW	0x0	int_en_afbcd2_hreg_dec_resp 1'b0: disable 1'b1: enable
7	RW	0x0	int_en_afbcd1_hreg_axi_rresp 1'b0: disable 1'b1: enable
6	RW	0x0	int_en_afbcd1_hreg_dec_resp 1'b0: disable 1'b1: enable
5	RW	0x0	int_en_afbcd0_hreg_axi_rresp 1'b0: disable 1'b1: enable
4	RW	0x0	int_en_afbcd0_hreg_dec_resp 1'b0: disable 1'b1: enable
3	RW	0x0	int_en_fbcd3 1'b0: disable 1'b1: enable
2	RW	0x0	int_en_fbcd2 1'b0: disable 1'b1: enable
1	RW	0x0	int_en_fbcd1 1'b0: disable 1'b1: enable
0	RW	0x0	int_en_fbcd0 1'b0: disable 1'b1: enable

**VOP\_INTR\_CLEAR1**

Address: Operational Base + offset (0x0294)

Interrupt clear register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:12	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
11	RW	0x0	int_clr_afbcd3_hreg_axi_rresp interrupt clear (Auto clear)
10	RW	0x0	int_clr_afbcd3_hreg_dec_resp interrupt clear (Auto clear)
9	RW	0x0	int_clr_afbcd2_hreg_axi_rresp interrupt clear (Auto clear)
8	RW	0x0	int_clr_afbcd2_hreg_dec_resp interrupt clear (Auto clear)
7	RW	0x0	int_clr_afbcd1_hreg_axi_rresp interrupt clear (Auto clear)
6	RW	0x0	int_clr_afbcd1_hreg_dec_resp interrupt clear (Auto clear)
5	RW	0x0	int_clr_afbcd0_hreg_axi_rresp interrupt clear (Auto clear)
4	RW	0x0	int_clr_afbcd0_hreg_dec_resp interrupt clear (Auto clear)
3	RW	0x0	int_clr_fbcd3 interrupt clear (Auto clear)
2	RW	0x0	int_clr_fbcd2 interrupt clear (Auto clear)
1	W1C	0x0	int_clr_fbcd1 interrupt clear (Auto clear)
0	W1C	0x0	int_clr_fbcd0 interrupt clear (Auto clear)

**VOP\_INTR\_STATUS1**

Address: Operational Base + offset (0x0298)

interrupt status

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:12	RO	0x0	reserved
11	RW	0x0	int_status_afbcd4_hreg_dec_resp interrupt status
10	RW	0x0	int_status_afbcd3_hreg_dec_resp interrupt status
9	RW	0x0	int_status_afbcd2_hreg_axi_rresp interrupt status
8	RW	0x0	int_status_afbcd2_hreg_dec_resp interrupt status
7	RW	0x0	int_status_afbcd1_hreg_axi_rresp interrupt status
6	RW	0x0	int_status_afbcd1_hreg_dec_resp interrupt status
5	RW	0x0	int_status_afbcd0_hreg_axi_rresp interrupt status

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
4	RW	0x0	int_status_afbcd0_hreg_dec_resp interrupt status
3	RW	0x0	int_status_fbcd3 interrupt status
2	RW	0x0	int_status_fbcd2 interrupt status
1	RO	0x0	int_status_fbcd1 interrupt status
0	RO	0x0	int_status_fbcd0 interrupt status

**VOP\_INTR\_RAW\_STATUS1**

Address: Operational Base + offset (0x029c)  
raw interrupt status

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:12	RO	0x0	reserved
11	RW	0x0	int_raw_status_afbcd3_hreg_axi_rresp interrupt raw status
10	RW	0x0	int_raw_status_afbcd3_hreg_dec_resp interrupt raw status
9	RW	0x0	int_raw_status_afbcd2_hreg_axi_rresp interrupt raw status
8	RW	0x0	int_raw_status_afbcd2_hreg_dec_resp interrupt raw status
7	RW	0x0	int_raw_status_afbcd1_hreg_axi_rresp interrupt raw status
6	RW	0x0	int_raw_status_afbcd1_hreg_dec_resp interrupt raw status
5	RW	0x0	int_raw_status_afbcd0_hreg_axi_rresp interrupt raw status
4	RW	0x0	int_raw_status_afbcd0_hreg_dec_resp interrupt raw status
3	RW	0x0	int_raw_status_fbcd3 interrupt raw status
2	RW	0x0	int_raw_status_fbcd2 interrupt raw status
1	RW	0x0	int_raw_status_fbcd1 interrupt raw status
0	RO	0x0	int_raw_status_fbcd0 interrupt raw status

**VOP\_LINE\_FLAG**

Address: Operational Base + offset (0x02a0)  
Line flag config register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RO	0x0	reserved
28:16	RW	0x0000	dsp_line_flag_num_1 Line number of the Line flag interrupt 1 The display line number when the flag interrupt 1 occur, the range is (0~ DSP_VTOTAL-1).
15:13	RO	0x0	reserved
12:0	RW	0x0000	dsp_line_flag_num_0 Line number of the Line flag interrupt 0 The display line number when the flag interrupt occur, the range is (0~ DSP_VTOTAL-1).

**VOP\_VOP\_STATUS**

Address: Operational Base + offset (0x02a4)  
vop status register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:18	RO	0x0	reserved
17	RW	0x0	dma_stop_valid dma stop valid
16	RW	0x0	mmu_idle mmu idle status
15:13	RO	0x0	reserved
12:0	RO	0x0000	dsp_vcnt read the dsp vertical counter

**VOP\_BLANKING\_VALUE**

Address: Operational Base + offset (0x02a8)  
Register0000 Abstract

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:25	RO	0x0	reserved
24	RW	0x0	blanking_value_config_en 1'b0 : disable blanking value when vop blank 1'b1 : enable blanking value when vop blank
23:0	RW	0x0000000	blanking_value vop output data value when blanking,24bits

**VOP\_WIN0\_DSP\_BG**

Address: Operational Base + offset (0x02b0)  
Win0 layer background color

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RW	0x0	win0_bg_en Win0 layer background enable 1'b0 : disable 1'b1 : enable
30:24	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
23:16	RW	0x00	win0_dsp_bg_red Win0 layer Background Red color
15:8	RW	0x00	win0_dsp_bg_green Win0 layer Background Green color
7:0	RW	0x00	win0_dsp_bg_blue Win0 layer Background Blue color

**VOP\_WIN1\_DSP\_BG**

Address: Operational Base + offset (0x02b4)

Win1 layer background color

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RW	0x0	win1_bg_en Win1 layer background enable 1'b0 : disable 1'b1 : enable
30:24	RO	0x0	reserved
23:16	RW	0x00	win1_dsp_bg_red Win1 layer Background Red color
15:8	RW	0x00	win1_dsp_bg_green Win1 layer Background Green color
7:0	RW	0x00	win1_dsp_bg_blue Win1 layer Background Blue color

**VOP\_DBG\_PERF\_LATENCY\_CTRL0**

Address: Operational Base + offset (0x0300)

Axi performance latency module control register0

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:25	RO	0x0	reserved
24:20	RW	0x00	rd_latency_st_num
19:8	RW	0x000	rd_latency_thr
7:4	RW	0x0	rd_latency_id0
3	RO	0x0	reserved
2	RW	0x0	rd_latency_mode read latency mode 1'b0: clear by frame end 1'b1: clear by software configuration
1	RW	0x0	hand_latency_clr read max latency software clear 1'b0: software clear disable 1'b1: software clear enable
0	RW	0x0	rd_latency_en vop_axi_rd_perf enable signal 1'b0: disable 1'b1: enable

**VOP\_DBG\_PERF\_RD\_MAX\_LATENCY\_NUM0**

Address: Operational Base + offset (0x0304)

Read max latency number

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:17	RO	0x0	reserved
16	RO	0x0	rd_latency_overflow_ch0 overflow flag(bigger than configed threshold value)
15:12	RO	0x0	reserved
11:0	RO	0x000	rd_max_latency_num_ch0 read max latency value of channel 0

**VOP\_DBG\_PERF\_RD\_LATENCY\_THR\_NUM0**

Address: Operational Base + offset (0x0308)

The number of bigger than configed threshold value

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x0	reserved
23:0	RO	0x000000	rd_latency_thr_num_ch0 read latency thr number channel 0

**VOP\_DBG\_PERF\_RD\_LATENCY\_SAMP\_NUM0**

Address: Operational Base + offset (0x030c)

Total sample number

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x0	reserved
23:0	RO	0x000000	rd_latency_samp_num_ch0

**VOP\_DBG\_WIN0\_REG0**

Address: Operational Base + offset (0x0320)

Vop debug win0 register0

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**VOP\_DBG\_WIN0\_REG1**

Address: Operational Base + offset (0x0324)

Vop debug win0 register1

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**VOP\_DBG\_WIN0\_REG2**

Address: Operational Base + offset (0x0328)

Vop debug win0 register2

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RO	0x0	reserved
28:16	RW	0x0000	dbg_win0_yrgb_cmd_line_cnt
15:0	RO	0x0	reserved

**VOP\_DBG\_WIN0\_RESERVED**

Address: Operational Base + offset (0x032c)

Vop debug win0 register3 reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:1	RO	0x0	reserved
0	RW	0x0	reserved reserved

**VOP\_DBG\_WIN1\_REG0**

Address: Operational Base + offset (0x0330)

Vop debug win1 register0

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**VOP\_DBG\_WIN1\_REG1**

Address: Operational Base + offset (0x0334)

Vop debug win1 register1

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**VOP\_DBG\_WIN1\_REG2**

Address: Operational Base + offset (0x0338)

Vop debug win1 register2

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**VOP\_DBG\_WIN1\_RESERVED**

Address: Operational Base + offset (0x033c)

Vop debug win1 register3 reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:1	RO	0x0	reserved
0	RW	0x0	reserved reserved

**VOP\_DBG\_PRE\_REG0**

Address: Operational Base + offset (0x0360)

Vop debug pre register0

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**VOP\_DBG\_PRE\_RESERVED**

Address: Operational Base + offset (0x0364)

Vop debug pre register1 reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:1	RO	0x0	reserved
0	RW	0x0	reserved reserved

**VOP\_DBG\_POST\_REG0**

Address: Operational Base + offset (0x0368)

Vop debug post register0

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**VOP\_DBG\_POST\_RESERVED**

Address: Operational Base + offset (0x036c)

Vop debug post register1 reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:1	RO	0x0	reserved
0	RW	0x0	reserved reserved

**VOP\_DBG\_DATAO**

Address: Operational Base + offset (0x0370)

debug data output path

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RW	0x0	sw_datao_sel debug data ouptut sel 2'b00:rgb output 2'bxx:reserved
29:0	RO	0x0	reserved

**VOP\_DBG\_DATAO\_2**

Address: Operational Base + offset (0x0374)

debug data output path 2

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RW	0x0	sw_datao_sel_2 debug data ouptut sel 2 2'b00:hDMI 2'b01:mipi 2'b10:lvds 2'b11:edp
29:0	RW	0x00000000	vop_data_o_2 debug vop data output 2 debug vop data output 2

**VOP\_POST\_YUV2YUV\_Y2R\_COE**

Address: Operational Base + offset (0x0480)

POST\_YUV2YUV\_Y2R\_COE

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:1	RO	0x0	reserved
0	RW	0x0	reserved

**VOP\_POST\_YUV2YUV\_3x3\_COE**

Address: Operational Base + offset (0x04a0)

POST\_YUV2YUV\_3x3\_COE

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:1	RO	0x0	reserved
0	RW	0x0	reserved

**VOP\_POST\_YUV2YUV\_R2Y\_COE**

Address: Operational Base + offset (0x04c0)

POST\_YUV2YUV\_R2Y\_COE

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:1	RO	0x0	reserved
0	RW	0x0	reserved

**VOP\_HWC\_LUT\_ADDR**

Address: Operational Base + offset (0x1800)

Hwc lut base address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	hwc_lut_addr the head of hwc lut address

**TVE\_TVE\_MODE\_CTRL**

Address: Operational Base + offset (0x3e00)

Mode Control

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:25	RO	0x0	reserved
24	RW	0x0	pal 1'b0 Encoding is NTSC 1'b1 Encoding is PAL Note: 525/60 and 625/50 timing is selected separately
23:20	RO	0x0	reserved
19:16	RW	0x0	clock_ratios 4'bXX00: Reserved 4'bXX01: Timing enable is 1 * pix_clk (HDTV) 4'bXX10: Timing enable is 1 * pix_clk (SDTV) 4'bXX11: Reserved 4'b00XX: Reserved 4'b01XX: Upstream enable is 1 * pix_clk (HDTV) 4'b10XX: Upstream enable is 1/2* pix_clk (SDTV) 4'b11XX: Reserved

Bit	Attr	Reset Value	Description
15:14	RW	0x0	<p>function_select            2'b00: AV Off (TV displays broadcast signal, not SCART signal)            2'b01: Interface dependent effect            2'b10: Interface dependent effect            2'b11: Interface dependent effect</p> <p>Note: these bits are used to generate the SCART Function Select (pin 8, Slow Switching) signal via an optional external interface. They have no effect on the internal operation of the TVE, and could be re-assigned to another board level function.</p>
13	RW	0x0	<p>fast_blank            1'b0: TV uses composite SCART signal            1'b1: TV uses RGB SCART signal</p> <p>Note: This bit is used to generate the SCART Fast Blank signal via an optional external interface, which is used for static switching only. This bit has no effect on the internal operation of the TVE.</p>
12:11	RO	0x0	reserved
10:9	RW	0x0	<p>luma_filter_gain            2'b00: Luma Filter gain is 1.0 (default value)            2'b01: Luma Filter gain is 0.5            2'b10: Luma Filter gain is 2.0            2'b11: Reserved</p>
8	RW	0x0	<p>luma_upsample            1'b0: Luma Filter output is sampled at 13.5 MHz            1'b1: Luma Filter output is up-sampled to 27 MHz</p>
7:3	RO	0x0	reserved
2:1	RW	0x0	<p>csc_input_sel            2'b00: RGB input to 4:4:4+RGB output            2'b01: 4:2:2 input to 4:4:4 output            2'b10: Reserved            2'b11: Bypass (4:4:4 input to 4:4:4 output)</p> <p>The PAL/NTSC encoder requires 4:4:4 out from the CSC block, component out requires RGB or CrYCb for RGB or PbYPr respectively.</p>
0	RW	0x0	<p>color_space            1'b0: BT.601 (PAL/NTSC)            1'b1: BT.709 (ATSC/EIA-720.3)</p> <p>RGB input is converted to 4:4:4 in either colourspace. The PAL/NTSC encoder requires BT.601 colour space, component out requires BT.601 for 480i/576i or BT.709 for 720p/1080i.</p>

**TVE\_TVE\_HOR\_TIMING1**

Address: Operational Base + offset (0x3e04)

## Horizontal Timing 1

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x0	reserved
23:16	RW	0x00	h_burst_end Delay to the end of colour burst (from H Sync Start) Unsigned value, LSB = 37 ns when timing enable = 27 MHz.
15:8	RW	0x00	h_breeze_end Delay to the start of the colour burst (from H Sync Start) Unsigned value, LSB = 37 ns when timing enable = 27 MHz.
7:0	RW	0x00	h_sync_end Width of horizontal sync from H Sync Start Unsigned value, LSB = 37 ns when timing enable = 27 MHz.

**TVE\_TVE\_HOR\_TIMING2**

Address: Operational Base + offset (0x3e08)

## Horizontal Timing 2

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:28	RW	0x0	tv_syncs 4'b0000: Sync generator is stopped 4'b0001: 525/60 interlaced 4'b0010: 625/50 interlaced 4'b0011: PALM 4'b0100: PALN z 4'b0101: i480 (EIA-770.2) 4'b0110: p480 (EIA-770.2) 4'b0111: p720 (EIA-770.3) 4'b1001: i1080 (EIA-770.3) 4'b1101: 262 non-interlaced 4'b1110: 312 non-interlaced Note that PAL colour encoding is separately controlled
27:16	RW	0x000	h_active_end Delay to the end of active video (from H Sync Start) Unsigned value, LSB = 37 ns when timing enable = 27 MHz.
15:13	RO	0x0	reserved
12	RW	0x0	sc_reset_en SC Reset Enable Bit 1'b0: Subcarrier phase is not reset regularly 1'b1: Subcarrier phase is reset every four fields (NTSC) or eight fields (PAL)
11:0	RW	0x000	h_active_st Delay to the start of the active video (from H Sync Start) Unsigned value, LSB = 37 ns.

**TVE\_TVE\_HOR\_TIMING3**

Address: Operational Base + offset (0x3e0c)

## Horizontal Timing 3

Bit	Attr	Reset Value	Description
31:28	RW	0x0	luma_delay_1 Programmable delay for Luma relative to Chroma which is applied to CVBS0 and Y/C outputs.
27:16	RW	0x000	h_total Total number of output pixels in a TV line in units of the timing enable. Applies to SD and HD formats. 525/60 = 1716 625/50 = 1728 720P/60 = 1650 1080IP/60 Hz = 2200 1080IP/50 Hz = 2640.
15:8	RO	0x0	reserved
7:0	RW	0x00	sc_horiz Phase offset applied if subcarrier phase is reset. Signed value adjusts subcarrier reset phase in units of 1.4 degrees per lsb

**TVE\_TVE\_SUB\_CAR\_FRQ**

Address: Operational Base + offset (0x3e10)

Sub-carrier Frequency

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dds_incr Phase increment used in the synthesis of the colour subcarrier Calculated as $2^{32} * F_{sc} / F_{ref}$ , where Fref is the timing enable rate (normally 27 MHz). Note that the Fref must be accurate to 20 ppm for encoded formats. Recommended settings (for 27.00 MHz) are: NTSC/M, NTSC-J 0x21F07BD7 PAL (B, D, G, H, I, N) 0x2A098ACB PAL/Nc 0x21F69446 PAL/M 0x21E6EFA4

**TVE\_TVE\_LUMA\_FILTER1**

Address: Operational Base + offset (0x3e14)

Luma Filter 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	luma_coeff3 8-bit signed
23:16	RW	0x00	luma_coeff2 8-bit signed

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
15:8	RW	0x00	luma_coeff1 8-bit signed
7:0	RW	0x00	luma_coeff0 8-bit signed

**TVE\_TVE\_LUMA\_FILTER2**

Address: Operational Base + offset (0x3e18)

Luma Filter 2

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RW	0x00	luma_coeff7 8-bit signed
23:16	RW	0x00	luma_coeff6 8-bit signed
15:8	RW	0x00	luma_coeff5 8-bit signed
7:0	RW	0x00	luma_coeff4 8-bit signed

**TVE\_TVE\_LUMA\_FILTER3**

Address: Operational Base + offset (0x3e1c)

Luma Filter 3

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RW	0x00	luma_coeff11 8-bit Unsigned
23:16	RW	0x00	luma_coeff10 8-bit signed
15:8	RW	0x00	luma_coeff9 8-bit signed
7:0	RW	0x00	luma_coeff8 8-bit signed

**TVE\_TVE\_MACROVISION\_RESERVED0**

Address: Operational Base + offset (0x3e20)

tve macrovision reserved0 register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**TVE\_TVE\_MACROVISION\_RESERVED1**

Address: Operational Base + offset (0x3e24)

tve macrovision reserved1 register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**TVE\_TVE\_MACROVISION\_RESERVED2**

Address: Operational Base + offset (0x3e28)

tve macrovision reserved2 register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**TVE\_TVE\_MACROVISION\_RESERVED3**

Address: Operational Base + offset (0x3e2c)

tve macrovision reserved3 register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**TVE\_TVE\_MACROVISION\_RESERVED4**

Address: Operational Base + offset (0x3e30)

tve macrovision reserved3 register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**TVE\_TVE\_IMAGE\_POSITION**

Address: Operational Base + offset (0x3e34)

tve image position

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:27	RO	0x0	reserved
26:16	RW	0x000	v_offset Vertical picture start offset (independent of picture blanking) The active area of the source image can be offset above or below the active area of the display. The value is the display line number at which the top line of the frame will be placed. The top of the frame cannot be earlier than line 1.
15:12	RO	0x0	reserved
11:0	RW	0x000	h_offset Horizontal picture start offset (independent of picture blanking) The active area of the source image can be offset to the left or right of the active area of the display. The value is the display pixel number at which the first pixel of the line will be placed. The left edge cannot be earlier than pixel 1.

**TVE\_TVE\_ROUTING**

Address: Operational Base + offset (0x3e38)

Tve Routing

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>																																			
31	RW	0x0	RB_swap 1'b0: No swap 1'b1: Swap Blue/Pb and Red/Pr channels on DACs, relative to the routing shown in 30:28																																			
30:28	RW	0x0	dac_fmt <table style="margin-left: 20px;"> <tr><td></td><td>DAC3</td><td>DAC2</td><td>DAC1</td><td>DAC0</td></tr> <tr><td>3'b000:</td><td>CVBS</td><td>Blue</td><td>Green</td><td>Red</td></tr> <tr><td>3'b001:</td><td>CVBS1</td><td>CVBS2</td><td>Y</td><td>C</td></tr> <tr><td>3'b011:</td><td>Y</td><td>CVBS2</td><td>CVBS1</td><td>C</td></tr> <tr><td>3'b100:</td><td>CVBS</td><td>Pb</td><td>Y</td><td>Pr</td></tr> <tr><td>3'b101:</td><td>C</td><td>Pb</td><td>Y</td><td>Pr</td></tr> </table>		DAC3	DAC2	DAC1	DAC0	3'b000:	CVBS	Blue	Green	Red	3'b001:	CVBS1	CVBS2	Y	C	3'b011:	Y	CVBS2	CVBS1	C	3'b100:	CVBS	Pb	Y	Pr	3'b101:	C	Pb	Y	Pr					
	DAC3	DAC2	DAC1	DAC0																																		
3'b000:	CVBS	Blue	Green	Red																																		
3'b001:	CVBS1	CVBS2	Y	C																																		
3'b011:	Y	CVBS2	CVBS1	C																																		
3'b100:	CVBS	Pb	Y	Pr																																		
3'b101:	C	Pb	Y	Pr																																		
27:24	RW	0x0	sense_on When ON, outputs a steady mid level on the selected DAC(s) to allow load sensing via DAC or external comparators (application dependent). May be applied singly or together <table style="margin-left: 20px;"> <tr><td></td><td>DAC3</td><td>DAC2</td><td>DAC1</td><td>DAC0</td></tr> <tr><td>4'b0000:</td><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td></tr> <tr><td>4'b0001:</td><td>OFF</td><td>OFF</td><td>OFF</td><td>ON</td></tr> <tr><td>4'b0010:</td><td>OFF</td><td>OFF</td><td>ON</td><td>OFF</td></tr> <tr><td>4'b0100:</td><td>OFF</td><td>ON</td><td>OFF</td><td>OFF</td></tr> <tr><td>4'b1000:</td><td>ON</td><td>OFF</td><td>OFF</td><td>OFF</td></tr> <tr><td>4'b1111:</td><td>ON</td><td>ON</td><td>ON</td><td>ON</td></tr> </table>		DAC3	DAC2	DAC1	DAC0	4'b0000:	OFF	OFF	OFF	OFF	4'b0001:	OFF	OFF	OFF	ON	4'b0010:	OFF	OFF	ON	OFF	4'b0100:	OFF	ON	OFF	OFF	4'b1000:	ON	OFF	OFF	OFF	4'b1111:	ON	ON	ON	ON
	DAC3	DAC2	DAC1	DAC0																																		
4'b0000:	OFF	OFF	OFF	OFF																																		
4'b0001:	OFF	OFF	OFF	ON																																		
4'b0010:	OFF	OFF	ON	OFF																																		
4'b0100:	OFF	ON	OFF	OFF																																		
4'b1000:	ON	OFF	OFF	OFF																																		
4'b1111:	ON	ON	ON	ON																																		
23:20	RW	0x0	vbid_on When ON, adds any enabled VBI Data to the specified channel <table style="margin-left: 20px;"> <tr><td></td><td>Luma</td><td>Red/Pr</td><td>Green</td><td>Blue/Pb</td></tr> <tr><td>4'b0000:</td><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td></tr> <tr><td>4'bXXXX1:</td><td></td><td></td><td></td><td>ON</td></tr> <tr><td>4'bXX1X:</td><td></td><td></td><td>ON</td><td></td></tr> <tr><td>4'bX1XX:</td><td></td><td>ON</td><td></td><td></td></tr> <tr><td>4'b1XXX:</td><td>ON</td><td></td><td></td><td></td></tr> </table>		Luma	Red/Pr	Green	Blue/Pb	4'b0000:	OFF	OFF	OFF	OFF	4'bXXXX1:				ON	4'bXX1X:			ON		4'bX1XX:		ON			4'b1XXX:	ON								
	Luma	Red/Pr	Green	Blue/Pb																																		
4'b0000:	OFF	OFF	OFF	OFF																																		
4'bXXXX1:				ON																																		
4'bXX1X:			ON																																			
4'bX1XX:		ON																																				
4'b1XXX:	ON																																					
19:16	RW	0x0	setup_on When ON, adds 7.5IRE setup to the specified channel <table style="margin-left: 20px;"> <tr><td></td><td>Luma</td><td>Red/Pr</td><td>Green</td><td>Blue/Pb</td></tr> <tr><td>4'b0000:</td><td>OFF</td><td>OFF</td><td>OFF</td><td>OFF</td></tr> <tr><td>4'bXXXX1:</td><td></td><td></td><td></td><td>ON</td></tr> <tr><td>4'bXX1X:</td><td></td><td></td><td>ON</td><td></td></tr> <tr><td>4'bX1XX:</td><td></td><td>ON</td><td></td><td></td></tr> <tr><td>4'b1XXX:</td><td>ON</td><td></td><td></td><td></td></tr> </table>		Luma	Red/Pr	Green	Blue/Pb	4'b0000:	OFF	OFF	OFF	OFF	4'bXXXX1:				ON	4'bXX1X:			ON		4'bX1XX:		ON			4'b1XXX:	ON								
	Luma	Red/Pr	Green	Blue/Pb																																		
4'b0000:	OFF	OFF	OFF	OFF																																		
4'bXXXX1:				ON																																		
4'bXX1X:			ON																																			
4'bX1XX:		ON																																				
4'b1XXX:	ON																																					

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
15:12	RW	0x0	<p>AGC_pluse_on When ON, adds AGC and EOF pulses to the specified channel. N.B. This field should not be used for disabling macrovision functionality.</p> <p>Y/C Red/Pr Green Blue/Pb 4'b0000: OFF OFF OFF OFF 4'bXXX1: ON 4'bXX1X: ON 4'bX1XX: ON 4'b1XXX: ON</p>
11:8	RW	0x0	<p>video_on When ON, adds appropriate video format to the specified channel</p> <p>Y/C Red/Pr Green Blue/Pb 4'b0000 : OFF OFF OFF OFF 4'bXXX1: ON 4'bXX1X: ON 4'bX1XX: ON 4'b1XXX: ON</p>
7:4	RW	0x0	<p>sync_on When ON, adds composite sync to the specified channel N.B. Macrovision pseudo-sync pulses (if enabled) will be added to each channel that has sync enabled.</p> <p>Luma Red/Pr Green Blue/Pb 4'b0000: OFF OFF OFF OFF 4'bXXX1: ON 4'bXX1X: ON 4'bX1XX: ON 4'b1XXX: ON</p>
3	RW	0x0	<p>YPP 1'b0: Component output is RGB 1'b1: Component output is YPbPr</p>
2	RW	0x0	<p>chroma_off 1'b0: Chroma is switched ON (normal colour display) 1'b1: Chroma is switched OFF (monochrome display)</p>
1:0	RW	0x0	<p>Picture_Sync_amplitudes 2'bX0: Component has picture/sync ratio of 714/286 (RS343A) 2'bX1: Component has picture/sync ratio of 700/300 (ATSC) 2'b0X : Composite has picture/sync ratio of 714/286 (NTSC) 2'b1X : Composite has picture/sync ratio of 700/300 (PAL)</p>

**TVE\_TVE\_VBID\_ST\_TIMING**

Address: Operational Base + offset (0x3e3c)

tve vbid start timing

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
------------	-------------	--------------------	--------------------

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**TVE\_TVE\_VBID\_LINES**

Address: Operational Base + offset (0x3e40)

tve vbid lines

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**TVE\_TVE\_VBID\_CC\_DATA**

Address: Operational Base + offset (0x3e44)

tve vbid cc data

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**TVE\_TVE\_VBID\_XDS\_DATA**

Address: Operational Base + offset (0x3e48)

tve vbid xds data

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**TVE\_TVE\_VBID\_CGMS\_DATA**

Address: Operational Base + offset (0x3e4c)

tve vbid cgms data

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**TVE\_TVE\_SYNC\_ADJUST**

Address: Operational Base + offset (0x3e50)

Sync Adjust

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:12	RO	0x0	reserved
11:8	RW	0x0	line_adj Signed number of lines by which to change frame length. Affects every frame whilst a non-zero value.
7:0	RW	0x00	pix_adj Signed number of (27MHz) pixels by which to adjust frame length. Affects every frame whilst a non-zero value.

**TVE\_TVE\_STATUS**

Address: Operational Base + offset (0x3e54)

TVE Status (Read Only)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:8	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
7	RO	0x0	pixel_manager_vstt Pixel Manager Vertical Start State 1'b0: Vertical Start not active 1'b1: Vertical Start active
6:4	RO	0x0	pixel_manager_sta Pixel Manager Flow Controller State 3'b000: Idle 3'b001: Seeking 3'b010: Ready 3'b011: First pixel 3'b100: Active 3'b111: Other
3:2	RO	0x0	reserved
1	RO	0x0	sync_gen_VBI Sync Generator Vertical Blanking Interval 1'b0: In VBI 1'b1: Not in VBI
0	RO	0x0	sync_gen_FID Sync Generator Field Identity 1'b0: First Field 1'b1: Second Filed

**TVE\_TVE\_SURFACE\_SIZE\_RESERVED**

Address: Operational Base + offset (0x3e58)

tve surface size reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**TVE\_TVE\_STREAM\_BASE\_ADDR\_CTRL\_RESERVED**

Address: Operational Base + offset (0x3e5c)

tve stream base addr ctrl reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**TVE\_TVE\_STRIDE\_RESERVED**

Address: Operational Base + offset (0x3e60)

tve stride reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**TVE\_TVE\_MEM\_INF\_CTRL\_RESERVED**

Address: Operational Base + offset (0x3e64)

tve mem inf ctrl reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**TVE\_TVE\_CTRL**

Address: Operational Base + offset (0x3e68)

TVE Control

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RW	0x0	disp_rst Software reset for all other TVE functions: 1'b0: Normal operation. 1'b1: Software reset (please do NOT reset the TVE every field or frame).
0	RO	0x0	reserved

**TVE\_TVE\_INTR\_STATUS**

Address: Operational Base + offset (0x3e6c)

Interrupt Status

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RW	0x0	ints_str1orun Indicates over-run for stream str1
0	RW	0x0	ints_str1urun Indicates under-run for stream str1

**TVE\_TVE\_INTR\_EN**

Address: Operational Base + offset (0x3e70)

Interrupt Enable

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RW	0x0	inte_str1orun Enables over-run interrupt for stream str1
0	RW	0x0	inte_str1urun Enables under-run interrupt for stream str1

**TVE\_TVE\_INTR\_CLR**

Address: Operational Base + offset (0x3e74)

Interrupt Clear

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RW	0x0	intclr_str1orun Writing '1' clears over-run interrupt for stream str1. Writing '0' has no effect
0	RW	0x0	intclr_str1urun Writing 磚'clears under-run interrupt for stream str1. Writing 磚'has no effect

**TVE\_TVE\_COLOR\_BUSRT\_SAT**

Address: Operational Base + offset (0x3e78)

Colour Burst and Saturation

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x0	reserved
23:16	RW	0x00	burst_size Colour burst amplitude
15:8	RW	0x00	V_weight Conversion factor for Cr to V
7:0	RW	0x00	U_weight Conversion factor for Cb to U

**TVE\_TVE\_TEXT\_HOR\_ACT\_RESERVED**

Address: Operational Base + offset (0x3e7c)

Teletext Horizontal Active

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**TVE\_TVE\_TEXT\_VER\_ACT\_RESERVED**

Address: Operational Base + offset (0x3e80)

tve text ver act reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**TVE\_TVE\_TEXT\_MODE\_RATE\_RESERVED**

Address: Operational Base + offset (0x3e84)

tve text mode rate reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**TVE\_TVE\_TEXT\_FETCH\_RESERVED**

Address: Operational Base + offset (0x3e88)

tve text fetch reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**TVE\_TVE\_CHROMA\_BANDWIDTH**

Address: Operational Base + offset (0x3e8c)

Chroma bandwidth

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:6	RO	0x0	reserved
5:4	RW	0x0	chroma_bandwidth 2'b00: Chroma bandpass filter is bypassed 2'b01: Chroma bandpass filter is centred on 3.58 MHz 2'b10: Chroma bandpass filter is centred on 4.43 MHz

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
3:0	RW	0x0	cdiff_bandwidth 4'b0000: Colour difference filters OFF (no colour) 4'b0001: Colour difference bandwidth is 0.6 MHz 4'b0010: Colour difference bandwidth is 1.3 MHz 4'b0011: Colour difference bandwidth is 2.0 MHz

**TVE\_TVE\_BRIGHTNESS\_CONTRAST**

Address: Operational Base + offset (0x3e90)

Brightness and Contrast

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RO	0x0	reserved
15:8	RW	0x00	Contrast This value determines the gain of the luma channel. It does not affect the chroma gain.
7:0	RW	0x00	Brightness This value determines the black level on the luma channel during active video only. The setup (if applicable) is applied in addition to this value.

**TVE\_TVE\_TEXT\_DATA\_ID\_RESERVED**

Address: Operational Base + offset (0x3e94)

tve text data id reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x0	reserved

**TVE\_TVE\_ID**

Address: Operational Base + offset (0x3e98)

TVE ID (Read Only)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x0a	group_id 0x0a
23:16	RO	0x01	core_id 0x01
15:0	RO	0x0	reserved

**TVE\_TVE\_REVISION**

Address: Operational Base + offset (0x3e9c)

TVE Revision (Read Only)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x00	oem OEM field = "00000000"
23:16	RO	0x01	major Major revision = "00000001"
15:8	RO	0x00	minor Minor revision= "00000001"

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
7:0	RO	0x08	maint Maintenance revision = □"00001000"

**TVE\_TVE\_CLAMP**

Address: Operational Base + offset (0x3ea0)

TVE Clamp

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RO	0x0	reserved
15:8	RW	0x00	clamp_off Horizontal position in pix_clks/4 where clamp pulse goes low
7:0	RW	0x00	clamp_on Horizontal position in pix_clks/4 where clamp pulse goes high

**VOP\_MMU\_DTE\_ADDR**

Address: Operational Base + offset (0x3f00)

MMU current page Table address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	MMU_DTE_ADDR MMU DTE ADDR

**VOP\_MMU\_STATUS**

Address: Operational Base + offset (0x3f04)

MMU status register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:11	RO	0x0	reserved
10:6	RO	0x00	PAGE_FAULT_BUS_ID Index of master responsible for last page fault
5	RO	0x0	PAGE_FAULT_IS_WRITE The direction of access for last page fault: 1'b0 = Read 1'b1 = Write
4	RO	0x0	REPLAY_BUFFER_EMPTY The MMU replay buffer is empty
3	RO	0x0	MMU_IDLE The MMU is idle when accesses are being translated and there are no unfinished translated accesses.
2	RO	0x0	STAIL_ACTIVE MMU stall mode currently enabled. The mode is enabled by command
1	RO	0x0	PAGE_FAULT_ACTIVE MMU page fault mode currently enabled . The mode is enabled by command.
0	RO	0x0	PAGING_ENABLED Paging is enabled

**VOP\_MMU\_COMMAND**

Address: Operational Base + offset (0x3f08)

MMU command register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:3	RO	0x0	reserved
2:0	WO	0x0	MMU_CMD MMU_CMD. This can be: 0: MMU_ENABLE_PAGING 1: MMU_DISABLE_PAGING 2: MMU_ENABLE_STALL 3: MMU_DISABLE_STALL 4: MMU_ZAP_CACHE 5: MMU_PAGE_FAULT_DONE 6: MMU_FORCE_RESET

**VOP\_MMU\_PAGE\_FAULT\_ADDR**

Address: Operational Base + offset (0x3f0c)

MMU logical address of last page fault

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x00000000	PAGE_FAULT_ADDR address of last page fault

**VOP\_MMU\_ZAP\_ONE\_LINE**

Address: Operational Base + offset (0x3f10)

MMU Zap cache line register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	WO	0x00000000	MMU_ZAP_ONE_LINE address to be invalidated from the page table cache

**VOP\_MMU\_INT\_RAWSTAT**

Address: Operational Base + offset (0x3f14)

MMU raw interrupt status register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RW	0x0	READ_BUS_ERROR read bus error
0	RW	0x0	PAGE_FAULT page fault

**VOP\_MMU\_INT\_CLEAR**

Address: Operational Base + offset (0x3f18)

MMU raw interrupt status register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
1	WO	0x0	READ_BUS_ERROR read bus error
0	WO	0x0	PAGE_FAULT page fault

**VOP\_MMU\_INT\_MASK**

Address: Operational Base + offset (0x3f1c)

MMU raw interrupt status register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RW	0x0	READ_BUS_ERROR read bus error
0	RW	0x0	PAGE_FAULT page fault

**VOP\_MMU\_INT\_STATUS**

Address: Operational Base + offset (0x3f20)

MMU raw interrupt status register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RO	0x0	READ_BUS_ERROR read bus error
0	RO	0x0	PAGE_FAULT page fault

**VOP\_MMU\_AUTO\_GATING**

Address: Operational Base + offset (0x3f24)

MMU auto gating

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:1	RO	0x0	reserved
0	RW	0x0	mmu_auto_gating mmu auto gating when it is 1'b1, the mmu will auto gating it self

## 3.5 Timing Diagram

### 3.5.1 RGB LCD interface timing

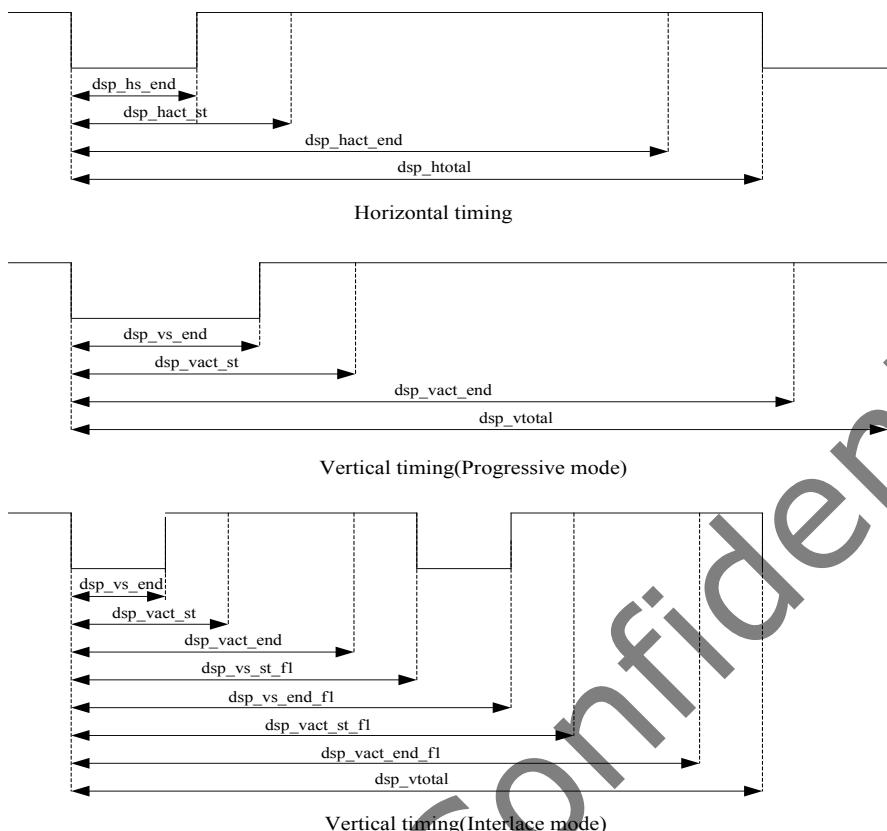


Fig. 7-26 VOP RGB interface timing (SDR)

## 3.6 Application Notes

### 3.6.1 DMA transfer mode

There are three DMA transfer modes for loading win0 or win1 frame data determined by following parameters( $X=0,1,2,3$ ):

dma\_burst\_length  
winX\_no\_outstanding  
winX\_gather\_en  
winX\_gather\_thres

#### 1. auto outstanding transfer mode(random transfer)

When  $\text{win}X\_no\_outstanding$  is 0, multi-bursts transfer command could be sent out to AXI master interface continuously if the internal memory has enough space to store new data. The continuous random burst number is in the range of 1 to 4, mainly depending on the empty level of internal memory,  $\text{dma}_\text{burst}_\text{length}$ , data format and active image width.

#### 2. configured outstanding transfer mode(fixed transfer)

When  $\text{win}X_\text{gather}_\text{en}$  is 1, fixed-number of bursts transfer command should be sent out to AXI master interface continuously if the internal memory has enough space to store new data. The fixed-number is determined by  $\text{win}X_\text{gather}_\text{thres}$ . Since the internal memory size is limited, there is some restriction for the  $\text{win}X_\text{gather}_\text{thres}$  as follows.

Table 7-3 Gather configuration for all format

Gather Threshold	dma_burst_length =2'b00(burst16)	dma_burst_length =2'b01(burst8)	dma_burst_length =2'b10(burst4)
YUV420	0	0,1,2	0,1,2,3
YUV422			
YUV444			
ARGB888			

RGB888 RGB565	0,1,2,3	0,1,2,3	0,1,2,3
8BPP 4BPP 2BPP 1BPP	0,1,2,3	0,1,2,3	0,1,2,3

### 3.6.2 Win0/Win1 dma load mode

If you want to improve the efficiency of accessing external memory for loading winX frame data, you could assert winX\_dma\_load. When winX\_dma\_load is high, winX frame data is loaded in the unit of line composing with one or more burst transfers; otherwise, loaded in the unit of burst transfer. However, it is not suitable for data format YUV420, no-scaling and active width less than 256.

### 3.6.3 WIN BPP LUT

WIN1 LUT/DSP LUT should be configured before displaying if win2/3\_lut\_en/dsp\_lut\_en is high. You could only update these LUTs by software.

When win1\_lut\_load\_en is 0, the WIN LUT data should be refreshed by software, i.e., writing win1 lut data to the internal memory with the start address WIN1\_LUT\_MST. The memory size is 256x25, i.e., lower 25bits valid, and the writing data number is determined by software, . When dsp\_lut\_load\_en is 0, the DSP LUT data should be refreshed by software, i.e., writing dsp lut data to the internal memory with the start address DSP\_LUT\_MST. The memory size is 256x24, i.e., lower 25bits valid, and the writing data number is determined by software. The program model for CABC gamma LUT is similar.

### 3.6.4 DMA QoS request

If you want to get higher priority for VOP to access external memory when the frame data is urgent, a QoS request can be generated and sent out basing on the configured values:

noc\_hurry\_en  
noc\_hurry\_value  
noc\_qos\_en  
noc\_win\_qos

If noc\_qos\_en is enable, a win0/1\_qos\_req is asserted when the empty level of win0/1's linebuffer is greater than the threshold configured in noc\_win\_qos. And it will be disserted when the empty level is smaller than the threshold or noc\_qos\_en is disable.

If noc\_qos\_en is enable, a win0/1\_hurry\_req is asserted when the empty level of win2/3's fifo is greater than the threshold configured in noc\_win\_qos. And it will be disserted when the empty level is smaller than the threshold or noc\_qos\_en is disable.

Either win0/1\_qos\_req or win2/3\_hurry\_req is high, a QoS request will be sent out for VOP.

### 3.6.5 Mirror display

If Y-Mirror display is enable, the frame data is loaded from last line to first line, where the start address of first pixel in last line is defined in  
WIN0/1\_YRGB0\_MST/WIN0/1\_CBR0\_MST/WIN0/1\_YRGB1\_MST/WIN0/1\_CBR\_MST for  
win0/1/2/3 respectively.

Otherwise, the win's frame line data width and virtual stride should be 64bit-aligned for 8bit-RGB/YUV or 128bit-aligned for 10bit YUV if X-Mirror or Y-Mirror display is enable.

### 3.6.6 DDR interface

LCD DDR interface is just suitable for Parallel RGB LCD panel and Serial RGB LCD 2x12 panel.  
If LCD DDR interface is enable, the timing parameters for LCD panel should be even.  
Otherwise, you can synchronize output clock with VSYNC or HSYNC depending on dclk\_ddr\_sync.

### 3.6.7 Interrupt

VOP interrupt is comprised of 10 interrupt sources:  
2 frame start interrupt

same frame address interrupt  
 2 line flag interrupt  
 hold interrupt  
 bus error interrupt  
 win0 empty interrupt  
 win1 empty interrupt  
 hwc empty interrupt  
 post empty interrupt  
 irq\_mmu  
 irq\_tve

Every interrupt has independent interrupt enable (VOP\_INT\_EN, with bit write mask), interrupt clear (VOP\_INT\_CLR, with bit write mask), interrupt status (VOP\_INT\_STATUS).

### 3.6.8 RGB display mode

RGB display mode is used for RGB panel display and CCIR656 output. It is a continuous frames display mode.

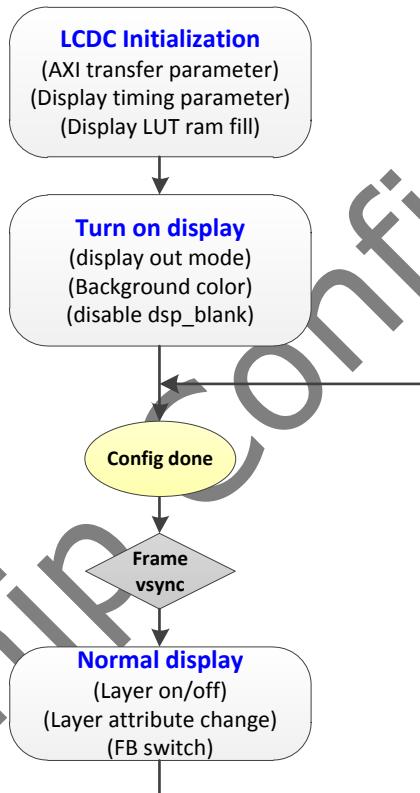


Fig. 7-27 VOP RGB mode Programming flow

#### 1. VOP initialization

VOP initialization should be done before turning display on.

First, AXI bus parameter (VOP\_SYS\_CTRL) should be set for DMA transfer.

Second, display panel/interface timing should be set for display output. The registers are:

VOP\_DSP\_HTOTAL\_HS\_END/ VOP\_DSP\_HACT\_ST\_END/ VOP\_DSP\_VTOTAL\_HS\_END/  
 VOP\_DSP\_VACT\_ST\_END/ VOP\_DSP\_VS\_ST\_END\_F1/ VOP\_DSP\_VACT\_ST\_END\_F1

#### 2. Background display

Before normal display, the background display could be turn on.

First, set display output mode (VOP\_DSP\_CTRL0/1) according to display device.

Second, disable dsp\_blank mode, which would not be enable until frame synchronization.

Finally, writing '1' to "VOP\_REG\_CFG\_DONE" register then all the frame-sync registers will be enable at the beginning of next frame.

#### 3. Normal display

In normal display, all the display layers' attribute could be different according display scenario. So there is a programming loop in this mode.

First, configure all the display layers' attribute registers for the change of image format, location, size, scaling factor, alpha and overlay and so on. Those register would not be enable until frame synchronization.

Finally, write '1' to "VOP\_REG\_CFG\_DONE" register then all the frame-sync registers will be enable at the beginning of next frame.

#### Immediately control register

There are two type register in VOP , one type is effective immediately,the other is effective by frame sync.

Effective immediately registers list as follows,other registers are all effective by frame sync.

Table 7-4 effective immediately register table

<b>register address</b>	<b>Description</b>
0x008[23:21],0x008[15:8]	some dsp ctrl function bit
0x00c	sys ctrl1 register
0x018	background color register
0x038	win0 color key register
0x078	win1 color key register
0x188~0x19c	dsp_timing ctrl registers
0x1ec~0x1f4	frc pattern parameter registers

#### 3.6.9 TV ENCODE

The TV ENCODE core feature :

Single 10-bit output to on-chip VDAC  
PAL/NTSC encoding

Programmable luma filter coefficient

Programmable luma/chroma delay

Programmable brightness/contrast

There is a special register to config In TV mode.

tve\_mode -> dsp\_ctrl0[25]

0 : NTSC mode

1 : PAL mode

Other registers reference to register description.

There list NTSC and PAL typical register config as follow:

```
ntsc_config[][2] ={
    {0x00 , 0x000a0000} ,
    {0x04 , 0x00C07a81} ,
    {0x08 , 0x169800FC} ,
    {0x0C , 0x96B40000} ,
    {0x10 , 0x21F07BD7} ,
    {0x14 , 0x02ff0000} ,
    {0x18 , 0xF40202fd} ,
    {0x1C , 0xF332d919} ,
    {0x34 , 0x001500D6} ,
    {0x38 , 0x0100888C} ,
    {0x3C , 0x00000000} ,
    {0x50 , 0x00000000} ,
    {0x68 , 0x00000000} ,
    {0x78 , 0x0052543C} ,
    {0x8C , 0x00000002} ,
    {0x90 , 0x00008300}};
```

```
pal_config[][2] ={
    {0x00 , 0x010a0006} ,
    {0x04 , 0x00C28381} ,
    {0x08 , 0x2674013D} ,
    {0x0C , 0x00000880} ,
```

```
{0x10 , 0x2A098ACB} ,  
{0x14 , 0x02ff0000} ,  
{0x18 , 0xF40202fd} ,  
{0x1C , 0xF332d919} ,  
{0x34 , 0x001500F6} ,  
{0x38 , 0x4100088A} ,  
{0x3C , 0x00000000} ,  
{0x50 , 0x00000000} ,  
{0x68 , 0x00000000} ,  
{0x78 , 0x0035604c} ,  
{0x8C , 0x00000022} ,  
{0x90 , 0x00009800});
```

When working in TV mode ,the dclk is 27Mhz,the sw\_core\_clk\_sel should enable.

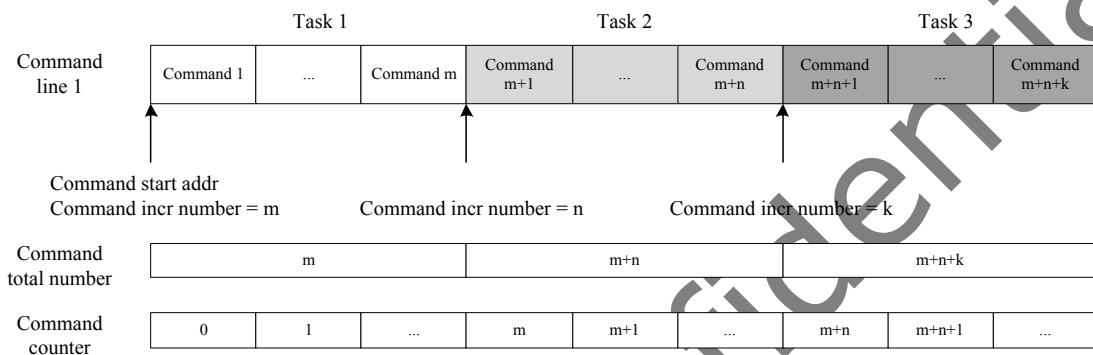


Fig. 7-28 RGA command line and command counter

### **3.6.10 Command Sync**

In slave command mode, command sync is controlled by CPU.

In master command mode, user can enable the current\_cmd\_int (sw\_intr\_cf), command by command to generate a interrupt at the end point of target command operation.

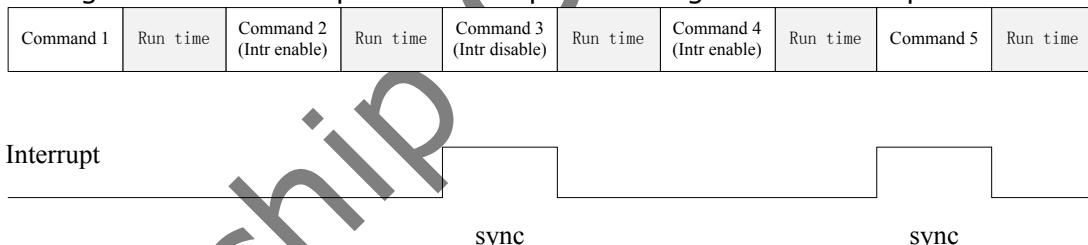


Fig. 7-29 RGA command sync generation

### **3.6.11 ColorPalette Application Notes**

1. Palette/LUT Load into special RAM in ELS\_BUF\_CTRL;
  2. ColorPalette/Pattern interval operations no need to initial LUT/pattern ram if LUT/pattern content no update;

### **3.6.12 Some special application constraint**

1. The algorithm of vertical scale up: must select bicubic algorithm when source picture is smaller or equal to 2k and must select bilinear when bigger than 2k
  2. The effects that The output's definition is near 2k or 4k may not very well when at the scenario that the vertical side is scale up and the horizontal is scale down within range of 2% (such as: 2048x32 → 2008x64)
  3. At the scenario A+B->C, the size among the A B C has some constraint :  
A's size must be equal to C. C's size must equal to B when A+C is no rotation. C's rotation (90degree) size must equal to B1 when A+C is rotation 90 degree .

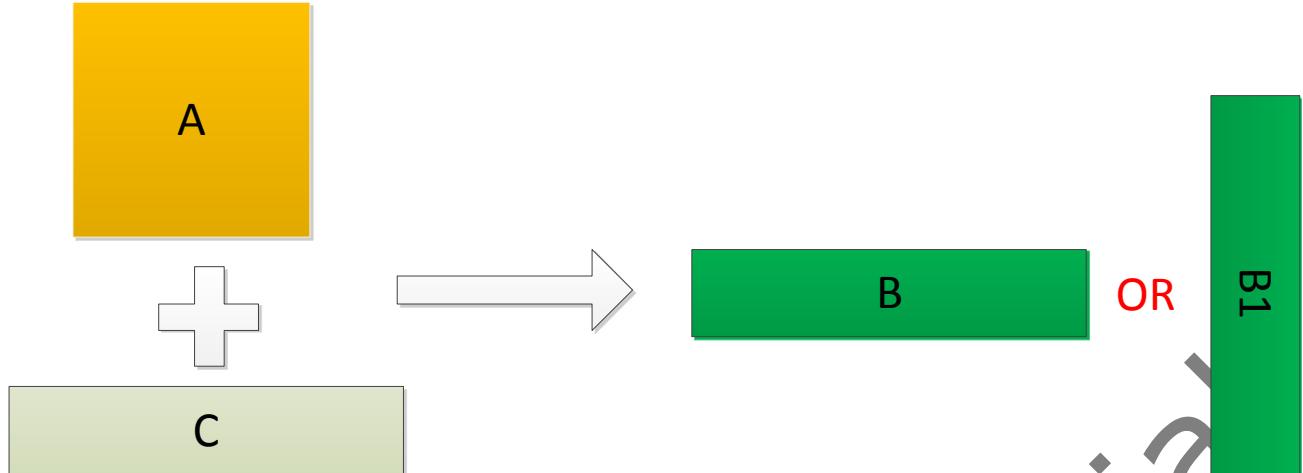


Fig. 7-30 the size constraint among A B C

4. YUV420/422-8bit virtual stride need 8byte align, xoff/yoff need 2byte align;
5. YUV420/422-10bit virtual stride need 16byte align, not support xoff/yoff;
6. Vertical scale down or not && Horizontal bi-cubic scale up src0 width<=2048;  
Vertical scale up && Horizontal bi-cubic scale up src0 width<=1928;
7. Vertical scale down or not && Horizontal bilinear scale up src0 width<=4096;  
Vertical scale up && Horizontal bilinear scale up src0 width<=3856;

Rockchip Confidential

## Chapter 4 Multi-format Video Decoder

### 4.1 Overview

The multi-format video codec (referred as codec in the following paragraph) supports three main world-wide advanced video coding standard (H264/VP9/HEVC) with high decoding performance @4K and h264/jpeg encoding.

The codec is connected to the AHB bus through an AHB slave and the AXI bus through an AXI master. The register configuration is fed into the decoder through the AHB slave interface while the large data such as stream data are transacted between DDR and the decoder through the AXI master interface.

In order to improve large data transaction performance, codec embeds MMU (memory management unit) and supports the cacheable bus operation.

The codec supports the following features:

- Supports HEVC decoding
  - Main and Main10 Profile up to Level 5.1 High Tier: 4096x2304@60 fps
  - Supports frame timeout interrupt , frame finish interrupt, bus error interrupt and bitstream error interrupt
  - Supports RLC write mode, RLC mode and Normal Mode
- Supports H264 decoding
  - The following profiles up to Level 5.1 : 4096x2304 @30fps
    - ◆ Baseline Profile
    - ◆ Main Profile
    - ◆ High Profile
    - ◆ High 10 profile
    - ◆ High 4:2:2 Profile(the MBAFF feature is not supported)
  - Supports frame timeout interrupt, frame finish interrupt and bitstream error interrupt, buffer empty interrupt
  - Supports slice by slice or random size stream decoding
  - Supports error-mode decoding and error info output
  - Supports RLC write mode, RLC mode and Normal Mode
- Supports VP9 decoding
  - Profile 0: 4096x2304@30fps
  - Supports frame timeout interrupt, frame finish interrupt and bitstream error interrupt
  - Supports RLC mode and Normal Mode
- MMU embedded with MMU interrupt support
- Supports MPEG-4 decoding
  - 60fps at 1920x1088
  - Simple Profile, levels 0-6
  - Advanced Simple Profile, levels 0-5
- MPEG-2/ MPEG-1
  - 60fps at 1920x1088
  - Main Profile, low, medium and high levels
- Supports JPEG decoding
  - 48x48 to 8176x8176(66.8 Mpixels), Step size 8 pixels
  - Baseline interleaved, and supports ROI (region of image) decode
- Built-in post processor in H.264(1080p) decoder supports
  - Stand-alone mode: rotation, RGB conversion, scaling, dithering
  - Pipe-line mode:, RGB conversion, scaling, dithering and alpha blending
- Supports encoding of the following standards:
  - H.264: up to HP level 4.1
- Built-in pre-processor in common video encoder supports:
  - rotation, YCbCr conversion

*Notes:AVS format decoder and jpeg encoder is not supported by RK3228 chip.*

## 4.2 Block Diagram

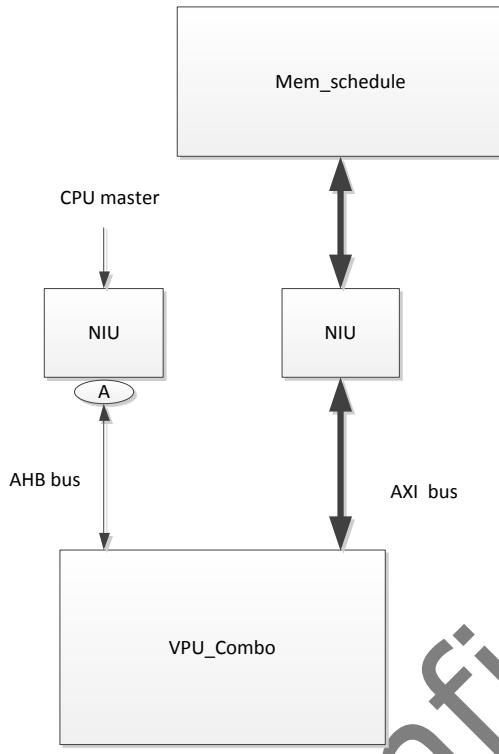


Fig. 7-31 Decoder in SOC

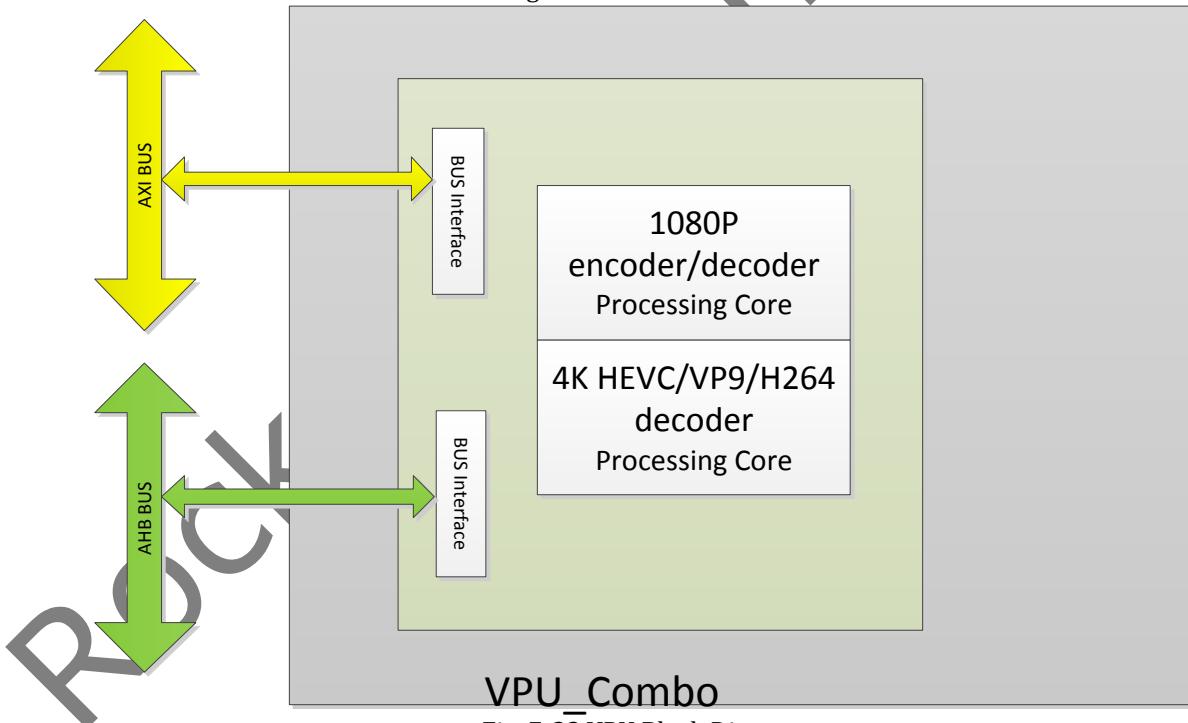


Fig. 7-32 VPU Block Diagram

As shown in the figures above, CPU accesses to the decoder register bank through AHB bus. Bitstream and other necessary data are fed into processing core through AXI read channel, and after several steps of decoding process, decoded pictures and other information data are transferred to designated location in the DDR through AXI write channel.

## 4.3 Video frame format

This chapter describes different input and output video frame formats supported by VCODEC. Each function module has its own supported video frame formats, and this chapter describes all the video frame formats.

### 4.3.1 YCbCr 4:2:0 Planar Format

In the planar format, each video sample component forms one memory plane. Within one plane, the data has to be stored linearly and contiguously in the memory as shown in Fig. 7-33. The luminance samples are stored in raster-scan order ( $Y_0Y_1Y_2Y_3Y_4\dots$ ). The chrominance samples are stored in two planes also in raster scan order ( $Cb_0Cb_1Cb_2Cb_3Cb_4\dots$  and  $Cr_0Cr_1Cr_2Cr_3Cr_4\dots$ ). In this format each pixel takes 12 bits of memory.

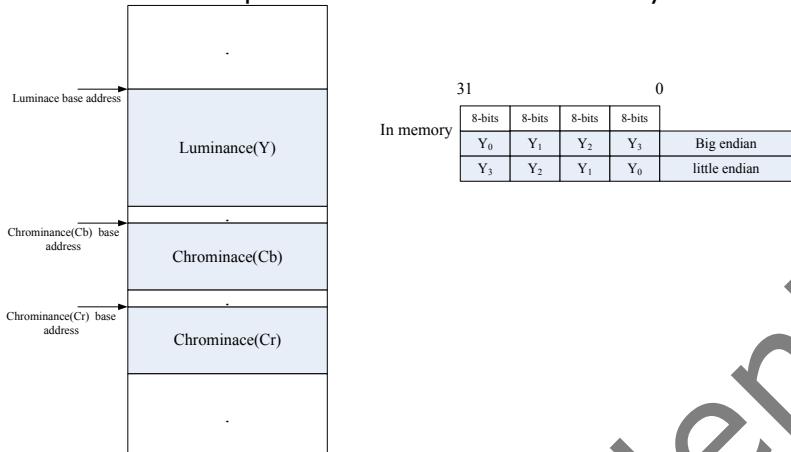


Fig. 7-33 VCODEC YCbCr 4:2:0 planar format

### 4.3.2 YCbCr 4:2:0 Semi-Planar format

In semi-planar YcbCr4:2:0 format the luminance samples from one plane in memory, and chrominance samples from another. Within one plane, the data has to be stored linearly and contiguously in the memory. The luminance pixels are stored in raster-scan order ( $Y_0Y_1Y_2Y_3Y_4\dots$ ). The interleaved chrominance CbCr samples are stored in raster-scan order in memory as  $Cb_0Cr_0Cb_1Cr_1Cb_2Cr_2Cb_3Cr_3Cb_4Cr_4\dots$ .

Semi-Planar format supports both progressive and interlaced format as presented in Fig. 7-34. The interlaced format may be alternative line or each line.

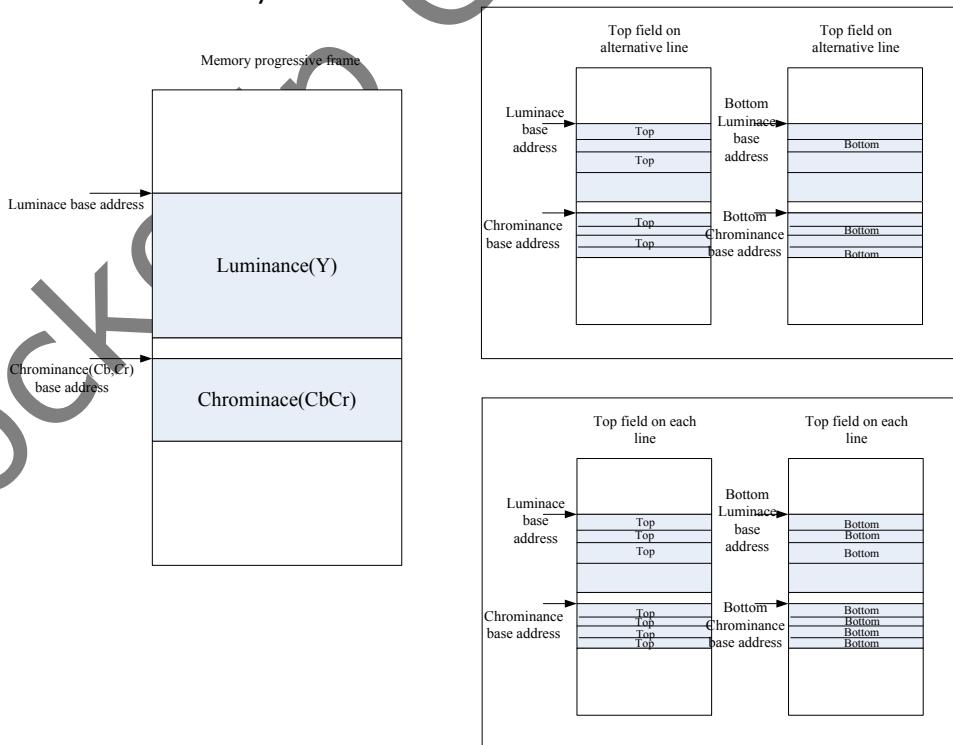


Fig. 7-34 VCODEC YCbCr 4:2:0 Semi-planar format

### 4.3.3 YCbCr 4:2:0 Tiled Semi-Planar Format

Like the YCbCr 4:2:0 semi-planar format, the tiled semi-planar format is also organized in the

memory on two separate planes. The difference between these formats is that in tiled format the pixel samples are not anymore in raster-scan order but are stored macroblock(16x16 pixels) by macroblock. The samples of each macroblock are stored in consecutive addresses and the macroblocks are ordered from left to right and from top to down as Fig. 7-35. When this format used as input data format, it causes the lowest bus load to the system as there is minimal amount of non-sequential memory addressing required when reading the input data to the post-processor.

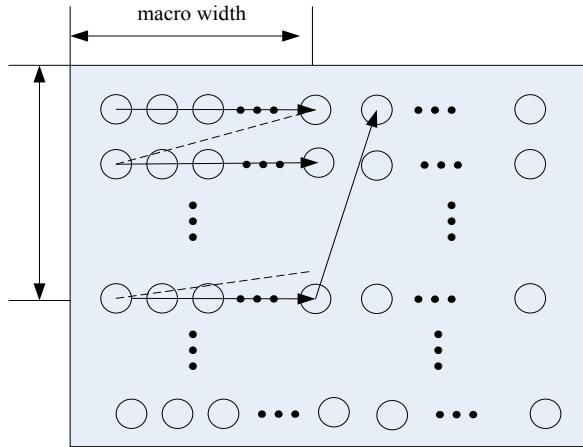


Fig. 7-35 VCODEC Tile scan mode

#### 4.3.4 YCbCr 4:2:2 Interleaved Format

In the interleaved YCbCr 4:2:2 format the pixel samples from a single plane in which the data has to be stored linearly and contiguously as shown in Fig. 7-36. The pixel data is in raster scan order and the chrominance samples are interleaved between the luminance samples as Y0Cb0 Y1Cr0 Y2 Cb1 Y3Cr1 Y4 Cb2.... YCrCb, CbYCrY and CrYCbY component orders are supported also. In this format, each pixel takes 16 bits in the memory.

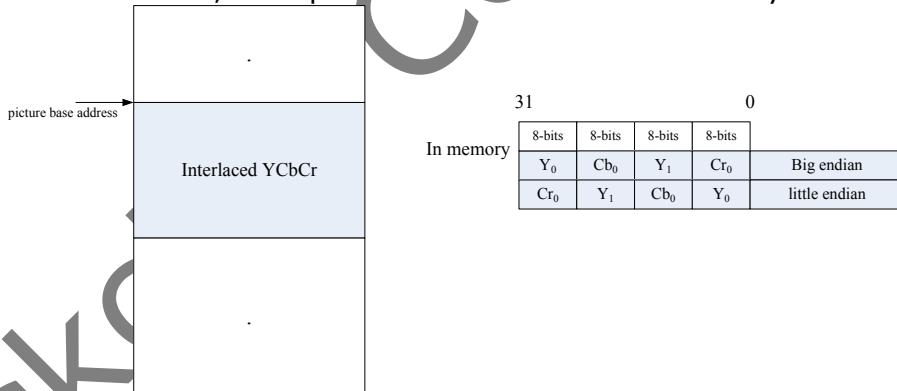


Fig. 7-36 VCODEC YCbCr4:2:2 Interleaved format

#### 4.3.5 AYCbCr 4:4:4 Interleaved Format

In the interleaved YcbCr 4:2:2 format, the pixel samples from a single plane in which the data has to be stored linearly and contiguously as show in Fig. 7-37. The pixel data is in raster scan order and the chrominance and alpha channel samples are interleaved between the luminance samples as A0Y0 Cb0Cr0 A1 Y1 Cb1Cr1....

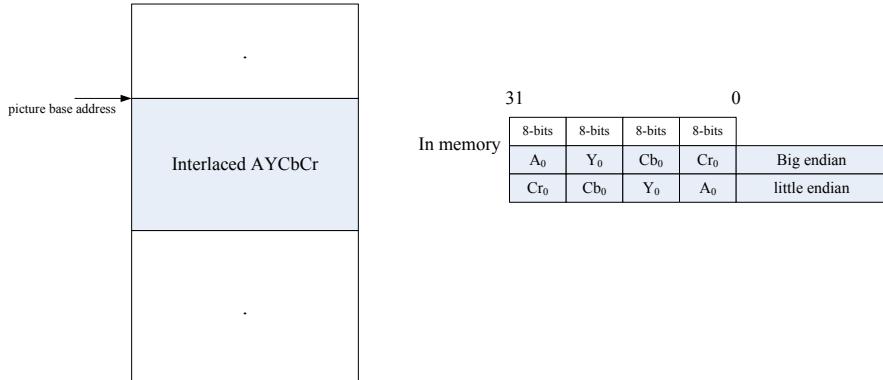


Fig. 7-37 VCODEC AYCbCr 4:4:4 Interleaved format

#### 4.3.6 RGB 16bpp Format

In this format each pixel is represented by 16 or less bits containing the red, blue and green samples. There are several 16bpp formats which use different number of bits for each sample. For example the RGB 5-5-5 format uses 5 bits for each sample and 1 bit is left unused or can represent a transparency flag, where RGB 5-6-5 uses 6 bits for the G sample and 5 bits for R and B samples. Common for all 16bpp types is that two pixels fit into one 32-bit space.

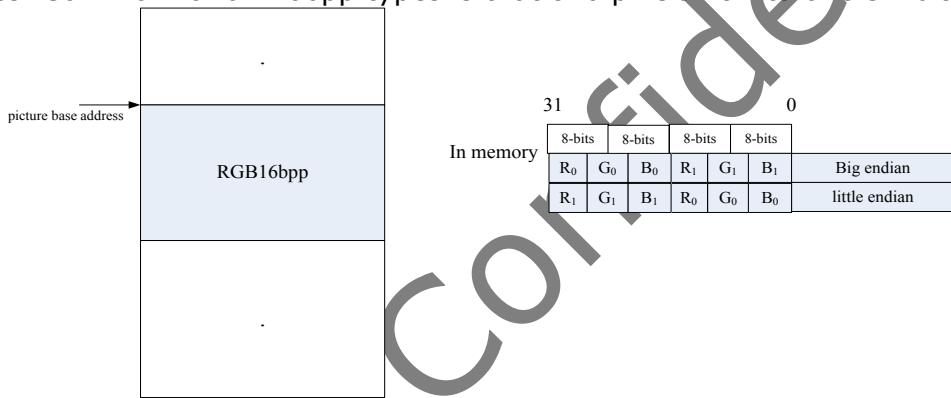


Fig. 7-38 VCODEC RGB 16bpp format

#### 4.3.7 RGB 32bpp Format

Any RGB format that has its pixels represented by more than 16bits each is considered to be of 32bpp type. Typically in this format each pixel is represented by three bytes containing a red, blue and green sample and a 4th byte which can be empty or hold an alpha blending value. Common for all 32bpp types is that only one pixel fit into one 32-bit space. The data has to be stored linearly and contiguously in the memory.

### 4.4 Function Description

#### 4.4.1 HEVC Standard

High Efficiency Video Coding (HEVC) is a video compression standard, a successor to H.264/MPEG-4 AVC (Advanced Video Coding), that was jointly developed by the ISO/IEC Moving Picture Experts Group (MPEG) and ITU-T Video Coding Experts Group (VCEG) as ISO/IEC 23008-2 MPEG-H Part 2 and ITU-T H.265. MPEG and VCEG established a Joint Collaborative Team on Video Coding (JCT-VC) to develop the HEVC standard.

HEVC was designed to substantially improve coding efficiency compared to H.264/MPEG-4 AVC HP, i.e. to reduce bitrate requirements by half with comparable image quality, at the expense of increased computational complexity. HEVC was designed with the goal of allowing video content to have a data compression ratio of up to 1000:1. Depending on the application requirements HEVC encoders can trade off computational complexity, compression rate, robustness to errors, and encoding delay time. Two of the key features where HEVC was improved compared to H.264/MPEG-4 AVC were support for higher resolution video and improved parallel processing methods.

#### 4.4.2 H264 Standard

H.264 or MPEG-4 Part 10, Advanced Video Coding (MPEG-4 AVC) is a video compression format that is currently one of the most commonly used formats for the recording, compression, and distribution of video content. The final drafting work on the first version of the standard was completed in May 2003, and various extensions of its capabilities have been added in subsequent editions.

H.264/MPEG-4 AVC is a block-oriented motion-compensation-based video compression standard developed by the ITU-T Video Coding Experts Group (VCEG) together with the ISO/IEC JTC1 Moving Picture Experts Group (MPEG). The project partnership effort is known as the Joint Video Team (JVT). The ITU-T H.264 standard and the ISO/IEC MPEG-4 AVC standard (formally, ISO/IEC 14496-10 – MPEG-4 Part 10, Advanced Video Coding) are jointly maintained so that they have identical technical content.

#### 4.4.3 VP9 Standard

VP9 is an open and royalty free video coding format being developed by Google. VP9 is a successor to VP8. Chromium, Chrome, Firefox, and Opera support playing VP9 video format in the HTML5 video tag. Development of VP9 started in Q3 2011. One of the goals for VP9 is to reduce the bit rate by 50% compared to VP8 while having the same video quality. Another goal for VP9 is to improve it to the point where it would have better compression efficiency than High Efficiency Video Coding.

#### 4.4.4 MMU

The MMU divides memory into 4KB pages, where each page can be individually configured. For each page the following parameters are specified:

1. Address translation of virtual memory, this enables the processor to work using address that differ from the physical address in the memory system.
2. The permitted types of accesses to that page. Each page can permit read, write, both, or none.

The MMU use 2-level page table structure:

1. The first level, the page directory consists of 1024 directory table entries(DTEs), each pointing to a page table.
2. The second level, the page table consists of 1024 page table entries(PTEs), each pointing to a page in memory.

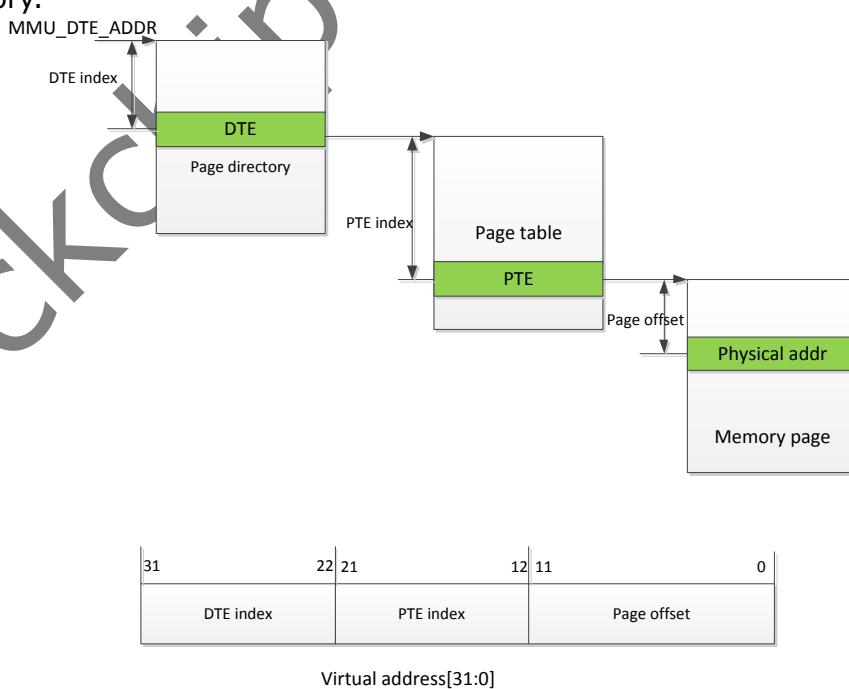


Fig. 7-39 structure of two-level page table

#### 4.4.5 HEVC Working Mode

There are three working modes to be selected for HEVC decoder: RLC Mode, RLC Write Mode, and Normal Mode.

The key differences among three working modes are whether CABAC module and Post-CABAC module are involved into the hardware decoding process.

For RLC mode, CABAC are bypassed and the input bitstream to the Post-CABAC module should be already decoded.

For RLC write mode, the decoded results by CABAC are output to the DDR, and the following decoding processes are stopped.

As for the normal mode, all the modules are involved into the decoding process, and complete decoding results are output. Normally, this mode should be selected.

#### **4.4.6 H264 Working Mode**

There are three working modes which can be chosen when decoding HEVC: RLC Mode, RLC Write Mode, and Normal Mode.

The key differences among three working modes are whether stream-parse module and post-stream-parse modules are involved into the hardware decoding process.

For RLC mode, stream-parse module is bypassed and the input data to the Post-CABAC module should be already parsed which can be recognized by the post-stream-parse modules.

For RLC write mode, the parsed results by stream-parse module are output to the DDR, and the following decoding processes are stopped.

As for the normal mode, all the modules are involved into the decoding process, and complete decoding results are output. Normally, this mode should be selected.

#### **4.4.7 VP9 Working Mode**

There are two working modes to be selected for VP9 decoder: RLC Mode, Normal Mode.

The key differences among these two working modes are whether strmd module is involved into the hardware decoding process.

For RLC mode, strmd is bypassed and the input dec\_ctrl cmd and inter cmd to the Post-strmd module should be already decoded. The dec\_ctrl cmd should be put on the address swreg4\_strm\_rlc\_base while the inter cmd should be put on the address swreg34\_vp9\_intercmd\_base.

As for the normal mode, all the modules are involved into the decoding process, and complete decoding results are output. Normally, this mode should be selected.

#### **4.4.8 Input Data Format for HEVC**

When the decoder works in RLC mode, the parsed stream info (which can be called dec\_ctrl commands) must be needed, which should be 128bit aligned.

When the decoder works in normal mode or RLC write mode, the bitstream, tbl ,pps and rps are needed for decoding.

1. bitstream

The bitstream must be 128bit align.

2. hevc table

hevc table is used for hevc context initialization, and it contains 156 units of 160-byte data.

3. hevc pps

Hevc pps contains 64 units of 80-byte data.

Table 7-5 sps format

video_parameter_set_id	4bit
seq_parameter_set_id	4bit
chroma_format_idc	2bit
pic_width_in_luma_samples	13bit
pic_height_in_luma_samples	13bit
bit_depth_luma	4bit
bit_depth_chroma	4bit
log2_max_pic_order_cnt_lsb	5bit
log2_diff_max_min_luma_coding_block_size	3bit
log2_min_luma_coding_block_size	3bit
log2_min_transform_block_size	3bit

log2_diff_max_min_transform_block_size	2bit
max_transform_hierarchy_depth_inter	3bit
max_transform_hierarchy_depth_intra	3bit
scaling_list_enabled_flag	1bit
amp_enabled_flag	1bit
sample_adaptive_offset_enabled_flag	1bit
pcm_enabled_flag	1bit
pcm_sample_bit_depth_luma	4bit
pcm_sample_bit_depth_chroma	4bit
pcm_loop_filter_disabled_flag	1bit
log2_diff_max_min_pcm_luma_coding_block_size	3bit
log2_min_pcm_luma_coding_block_size	3bit
num_short_term_ref_pic_sets	7bit
long_term_ref_pics_present_flag	1bit
num_long_term_ref_pics_sps	6bit
sps_temporal_mvp_enabled_flag	1bit
strong_intra_smoothing_enabled_flag	1bit
Transform_skip_rotation_enabled_flag	1bit
Transform_skip_context_enabled_flag	1bit
intra_block_copy_enabled_flag	1bit
residual_dpcm_intra_enabled_flag	1bit
residual_dpcm_inter_enabled_flag	1bit
extended_precision_processing_flag	1bit
intra_smoothing_disabled_flag	1bit
Reserved	32bit

Table 7-6 pps format

pps_pic_parameter_set_id	6bit
pps_seq_parameter_set_id	4bit
dependent_slice_segments_enabled_flag	1bit
output_flag_present_flag	1bit
num_extra_slice_header_bits	3bit//?
sign_data_hiding_flag	1bit
cabac_init_present_flag	1bit
num_ref_idx_l0_default_active	4bit
num_ref_idx_l1_default_active	4bit
init_qp_minus26	6bit
constrained_intra_pred_flag	1bit
transform_skip_enabled_flag	1bit
cu_qp_delta_enabled_flag	1bit
Log2MinCuQpDeltaSize	3bit
pps_cb_qp_offset	5bit
pps_cr_qp_offset	5bit
pps_slice_chroma_qp_offsets_present_flag	1bit
weighted_pred_flag	1bit
weighted_bipred_flag	1bit
transquant_bypass_enabled_flag	1bit

tiles_enabled_flag	1bit
entropy_coding_sync_enabled_flag	1bit
pps_loop_filter_across_slices_enabled_flag	1bit
loop_filter_across_tiles_enabled_flag	1bit
deblocking_filter_override_enabled_flag	1bit
pps_deblocking_filter_disabled_flag	1bit
pps_beta_offset_div2	4bit
pps_tc_offset_div2	4bit
lists_modification_present_flag	1bit
log2_parallel_merge_level	3bit
slice_segment_header_extension_present_flag	1bit
log2_transform_skip_max_size_minus2	3bit
If(tiles_enable_flag)	
{	
num_tile_columns	5bit
num_tile_rows	5bit
column_width[20]	8 * 20bits
column_height[22]	8 * 22 bit
}	
Scaling_address	32bit

#### 4. HEVC rps

HEVC rps contains a number of slice data composed by 2 units of 32-byte data.

#### 4.4.9 Input Data for H264

When in RLC mode, dec\_ctrl cmd's are needed, and they all should be 128bit aligned.

When in normal mode or direct write mode, bitstream, tbl ,pps and rps are need, which should be 128bit aligned.

##### 1. bitstream

The bitstream data should be put on the address, which are set as swreg4\_strm\_rlc\_base. The bitstream data should be 128bit aligned, and there are three modes to fetch bitstream: frame by frame, slice by slice, and random.

For frame by frame mode, swreg2\_sysctrl[25] must be configured as 1'b0, and swreg2\_sysctrl[28] must be configured as 1'b0. The bitstream of whole frame must be send to the decoder completely.

For slice by slice mode, swreg2\_sysctrl[25] must be configured as 1'b0, swreg2\_sysctrl[28] must be configured as 1'b1, and swreg1\_int[6] must be configured as 1'b1. The bitstream will be split into one or more slices. When swreg1\_int[16] is 1, we must send the next slice pack to the decoder, and then configure swreg1\_int[10] as 1'b1. Repeat step mention above until the whole slice pack is sent completely.

For random mode, swreg2\_sysctrl[25] must be configured as 1'b1, and swreg1\_int[6] must be configured as 1'b1. The bitstream will be split into random pack .when swreg1\_int[16] is 1, we must send the next pack to the decoder, and then configure swreg1\_int[10] as 1'b1. Repeat the operation until the whole pack is sent completely. When sending a packet at the end of the bitstream, we must configure config[26] as 1'b1.

##### 2. H264 table

H264 table is used for h264 context initialization. It contains four parts, and each part is 128bit align.

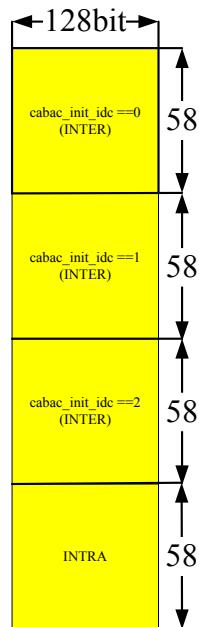


Fig. 7-40 h264 table

## 3. h264 pps

There are 256 units of 256bit data.

Table 7-7 pps format

seq_parameter_set_id	4bit
profile_idc	8bit
constraint_set3_flag	1bit
chroma_format_idc	2bit
bit_depth_luma_minus	3bit
bit_depth_chroma_minus	3bit
qpprime_y_zero_transform_bypass_flag	1bit
log2_max_frame_num_minus4	4bit
max_num_ref_frames	4bit
pic_order_cnt_type	2bit
log2_max_pic_order_cnt_lsb_minus4	4bit
delta_pic_order_always_zero_flag	1bit
pic_width_in_mbs	9bit
pic_height_in_mbs	9bit
frame_mbs_only_flag	1bit
mb_adaptive_frame_field_flag	1bit
direct_8x8_inference_flag	1bit
mvc_extension_enable	1bit
num_views	2bit
view_id[0]	10bit
view_id[1]	10bit
num_anchor_refs_l0	1bit
anchor_ref_l0	10bit
num_anchor_refs_l1	1bit
anchor_ref_l1	10bit
num_non_anchor_refs_l0	1bit
non_anchor_ref_l0	10bit

num_non_anchor_refs_l1	1bit
non_anchor_ref_l1	10bit
Align(128)	3bit
pps_pic_parameter_set_id	8bit
pps_seq_parameter_set_id	5bit
entropy_coding_mode_flag	1bit
bottom_field_pic_order_in_frame_present_flag	1bit
num_ref_idx_l0_default_active_minus1	5bit
num_ref_idx_l1_default_active_minus1	5bit
weighted_pred_flag	1bit
weighted_bipred_idc	2bit
pic_init_qp_minus26	7bit
pic_init_qs_minus26	6bit
chroma_qp_index_offset	5bit
deblocking_filter_control_present_flag	1bit
constrained_intra_pred_flag	1bit
redundant_pic_cnt_present_flag	1bit
transform_8x8_mode_flag	1bit
second_chroma_qp_index_offset	5bit
Scaling_list_enable_flag	1bit
Scaling_list_address	32bit
Is_long_term[16]	16bit
Voidx	16bit
Align(128)	8bit

## 4. h264 rps

The data should be 128bit aligned, and there are two modes to reorder ref pic: software and hardware.

For hardware mode, swreg2\_sysctrl[24] must be configured as 1'b1, and there is 8 data (the unit is 128bit) for rps initialization in one frame. For software mode, swreg2\_sysctrl[24] must be configured as 1'b0, there is 3 data(the unit is 128bit) in one slice, So a frame including the slices number multiplied by 3 the amount of data.

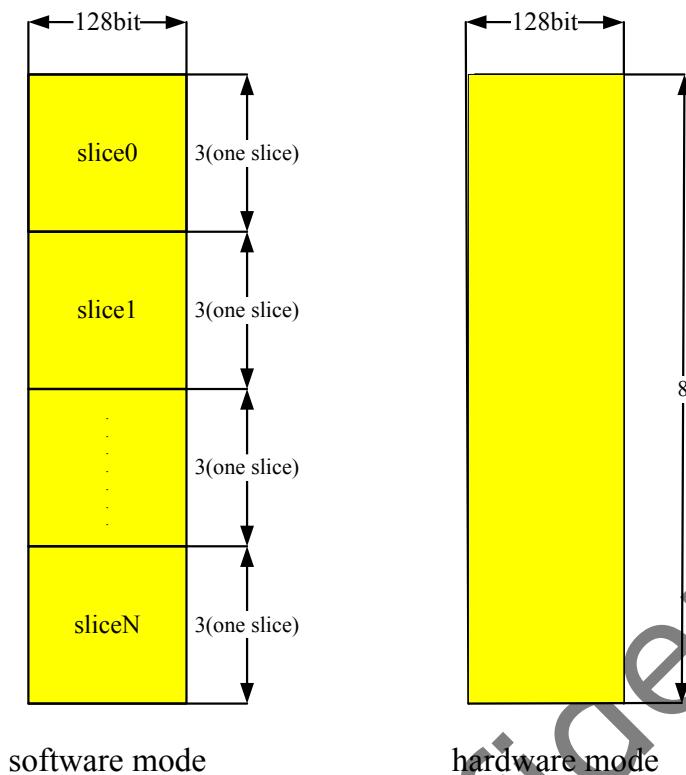


Fig. 7-41 h264 rps data format

For software mode, Each Specific data are listed in the following table:

Table 7-8 software rps format

for(j = 0; j < 2; j++)	
{	
for(i = 0; i < 32; i++)	
{	
dbp_idx[m][j][i]	5bit
bottom_flag[m][j][i]	1bit
View_id	1bit
}	
}	
Align(128)	

For hardware mode, Each Specific data are listed in the following table:

Table 7-9 hardware rps format

Framenum_wrap	256bit
No used	16bit
No used	16bit
for(j = 0; j < 3; j++)	list p,list b0,listb1
{	
for(i = 0; i < 32; i++)	
{	
dbp_idx[j][i]	5bit
bottom_flag[j][i]	1bit
view_id[j][i]	1bit
}	
}	

Align(128)

#### 4.4.10 Input Data for VP9

When in RLC mode, two data inputs are needed, which are dec\_ctrl cmd and inter cmd, and they all should be 128bit aligned.

When in normal mode, three data inputs are needed, which are bitstream, prob and segid from last frame.

##### 1. vp9 prob data format:

For intra\_only frame, there are 149 units of 128bit data; for inter frame, there are 153 units of 128bit data.

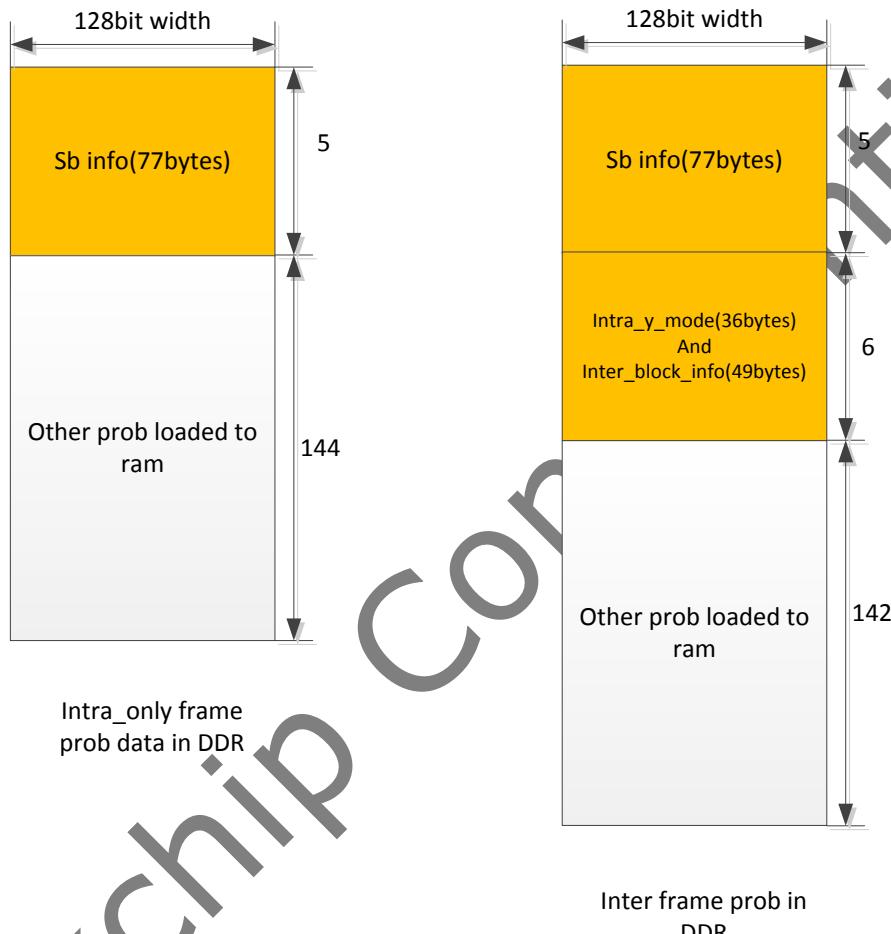


Fig. 7-42 vp9 prob data format

##### 2. the format of segid from last frame data format

The segid data is store with CTU raster scan mode, every CTU contain two 128bits. In the CTU, every 8x8 block has a segid data (Smn) which is 4 bits.

S00	S01	S02	S03				
S10							
				Smn			
							S77

Segid data format in  
a CTU

Fig. 7-43 vp9 segid data format

#### 4.4.11 Output Data Format

##### 1. Decoded frame data format

The decoded frame data are stored in the location with raster scan order where designated in the register configuration. Y component of the video data are stored first, and then UV component of the video data are stored.

S00	S01	S02	S03				
S10							
				Smn			
							S77

Segid data format in  
a CTU

Fig. 7-44 Data format in the DDR of Dec\_out\_base vp9

##### 2. COLMV data

COLMV data, which are needed for inter prediction , are in the DDR space following the YUV

data.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	
pred_flag	field_mode	ref_idx										bottom_flag	n/a										delta_poc									
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	
MV_X																MV_Y																
<b>pred_flag:</b> colpic mv prediction flag [0] <b>field_mode:</b> colpic macroblock is field coded mode [1] <b>ref_idx:</b> colpic reference index [6:2] <b>bottom_flag:</b> colpic macroblock is bottom field flag [7] <b>delta_poc:</b> difference picture order cnt between current pic and reference pic [31:16] <b>MV_X:</b> colpic mv for x direction [47:32] <b>MV_Y:</b> colpic mv for y direction [63:48]																																

Fig. 7-45 H264 colmv output format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
ref_frame_0	ref_frame_1	ref_idx										mv_0_x										mv_0_y													
36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67				
mv_1_x																mv_1_y																			
<b>ref_frame_0</b> : reference for L0, 0:last 1:golden 2:altref [1:0] <b>ref_frame_1</b> : reference for L1, 0:last 1:golden 2:altref [3:2] <b>mv_0_x</b> : reference mv at x direction for L0 [19:4] <b>mv_0_y</b> : reference mv at y direction for L0 [35:20] <b>mv_1_x</b> : reference mv at x direction for L1 [51:36] <b>mv_1_y</b> : reference mv at y direction for L1 [67:52]																																			

Fig. 7-46 VP9 colmv output format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
pred_flag	field_mode	ref_idx										bottom_flag	n/a										delta_poc												
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63				
MV_X																MV_Y																			
<b>pred_flag:</b> colpic mv prediction flag [0] <b>field_mode:</b> colpic macroblock is field coded mode [1] <b>ref_idx:</b> colpic reference index [6:2] <b>bottom_flag:</b> colpic macroblock is bottom field flag [7] <b>delta_poc:</b> difference picture order cnt between current pic and reference pic [31:16] <b>MV_X:</b> colpic mv for x direction [47:32] <b>MV_Y:</b> colpic mv for y direction [63:48]																																			

Fig. 7-47 HEVC colmv output format

### 3. Only for VP9

#### (1) Cur segid output:

The cur segid output data format is the same with segid from last frame.

#### (2) Cur count output:

For intra only frame, there are 98 words for no coeff count and 1440 words for coeff count.

For inter frame, there are 426 words for no coeff count and 2880 words for coeff count.

For every count, its real valid bits are 24, but it used 32bits in the DDR.

Segid data are output only when the decoder is decoding VP9.

## 4.4.12 H264 Error Process

For h264, it has two types of error. One is the normal error, and the other one is the logic error. Whether it is a normal error or a logic error, the swreg76\_h264\_errorinfo\_num[13:0] will record the number of slices in frame, swreg76\_h264\_errorinfo\_num[15] will be set to 1, and swreg76\_h264\_errorinfo\_num[29:16] will record the number of error slices in frame.

#### 1. normal error

For the normal error, it is recorded by the detection of syntax that is beyond the scope.

When swreg1\_int[19] is configured as 1'd1, the hardware will wait the end signal of deblocking and then reset itself. And for normal error, the error info will be put on the address swreg4\_strm\_rlc\_base. Specific data are listed in the following table:

Table 7-10 error info format

start_ctu_address	16bit	When it is inter error,it is error ctu assr,or it is the first ctu of current slice.
end_ctu_address	16bit	The last ctu of the current slice
error_type	16bit	When error_type[8] is equal to 1,it means that the

		slice is wrong. the error_type[7:0] see the 3-2 table
slice_byte_offset	24bit	The first byte of current slice in the bitstream

When The corresponding register shown in the following table is configured as 1'b0,it will not detect The corresponding error.

Table 7-11 normal error table

error_type	Syntax	enable
0	start_mb_nr(syntax)	swreg44_strmd_error_en[0]
1	slice_type	swreg44_strmd_error_en[1]
2	pic_parameter_set_id	swreg44_strmd_error_en[2]
3	Framenum	swreg44_strmd_error_en[3]
4	idr_pic_id	swreg44_strmd_error_en[4]
5	delta_pic_order_cnt_bottom	swreg44_strmd_error_en[5]
6	delta_pic_order_cnt	swreg44_strmd_error_en[6]
7	num_ref_idx_lx_active_minus1	swreg44_strmd_error_en[7]
8	Cabac_init_idc	swreg44_strmd_error_en[8]
9	Slice_qp_delta	swreg44_strmd_error_en[9]
10	disable_deblocking_filter_idc	swreg44_strmd_error_en[10]
11	slice_alpha_c0&beta_offset_div2	swreg44_strmd_error_en[11]
12	modification_of_pic_nums_idc	swreg44_strmd_error_en[12]
13	modification_of_pic_num	swreg44_strmd_error_en[13]
14	abs_diff_pic_num_minus1	swreg44_strmd_error_en[14]
15	abs_diff_pic_num_nonexist	swreg44_strmd_error_en[15]
16	long_term_pic_num	swreg44_strmd_error_en[16]
17	long_term_pic_num_nonexist	swreg44_strmd_error_en[17]
18	abs_diff_view_idx_minus1	swreg44_strmd_error_en[18]
19	memory_management_control_operations	swreg44_strmd_error_en[19]
20	difference_of_pic_nums_minus1	swreg44_strmd_error_en[20]
21	long_term_frame_idx	swreg44_strmd_error_en[21]
22	max_long_term_frame_idx_plus1	swreg44_strmd_error_en[22]
23	log2_weight_denom	swreg44_strmd_error_en[23]

24	pred_weight_offset	swreg44_strmd_error_en[24]
25	View_id	swreg44_strmd_error_en[25]
26	Skip_run	swreg77_h264_error_e[13]
27	Mb_type	swreg77_h264_error_e[14]
28	Coded_block_pattern	swreg77_h264_error_e[15]
29	Mb_qp_delta	swreg77_h264_error_e[16]
30	Ref_idx_IX	swreg77_h264_error_e[17]
31	Mvd_x	swreg77_h264_error_e[18]
32	Mvd_y	swreg77_h264_error_e[19]
33	Sub_mb_type	swreg77_h264_error_e[20]
34	Coeff_token	swreg77_h264_error_e[21]
35	Level_prefix	swreg77_h264_error_e[22]
36	Total_zero	swreg77_h264_error_e[23]
37	Run_before	swreg77_h264_error_e[24]
38	Coeff_abs_level_minus1	swreg77_h264_error_e[25]
39	Reserved	swreg77_h264_error_e[26]
40	Data_runout	swreg77_h264_error_e[28]
255	Inter error	

For normal error except inter error, When swreg1\_int[19] is configured as 1'd0, the hardware will stop the decoder and reset itself. And the error\_type in the table 3-2 will put on the sw\_strmd\_error\_status[27:0], the current ctu\_x will put on the swreg46\_strmd\_error\_ctu[7:0], the current ctu\_x will put on the swreg46\_strmd\_error\_ctu[15:8]. The Inter error be put on the address swreg4\_strm\_rlc\_base any ways.

## 2. logic error

For logic error, it is recorded by the detection of different slice of syntax elements that are the same.

When swreg1\_int[19] is configured as 1'd1, when there is any stream error, the hardware will wait the end signal of deblocking and then reset itself. And the error\_type in the table 3-3 will put on the sw\_strmd\_error\_status[27:0], the current ctu\_x will put on the swreg46\_strmd\_error\_ctu[7:0], the current ctu\_x will put on the swreg46\_strmd\_error\_ctu[15:8].

Table 7-12 logic error table

Error_type	syntax	enable
43	Delta_pic_order_cnt	swreg44_strmd_error_en[27]
44	Delta_pic_order_cnt_bottom	swreg44_strmd_error_en[28]
45	Pic_order_cnt_lsb	swreg44_strmd_error_en[29]
46	idr_pic_id	swreg44_strmd_error_en[30]
47	Bottom_field_flag	swreg44_strmd_error_en[31]
48	Field_pic_flag	swreg77_h264_error_e[0]

49	Frame_num	swreg77_h264_error_e[1]
50	pps_id_logic	swreg77_h264_error_e[2]
51	Slice_type_logic	swreg77_h264_error_e[3]
52	First_mb_in_slice_logic	swreg77_h264_error_e[4]
53	Nal_ref_idc_logic	swreg77_h264_error_e[5]
54	Idr_flag_logic	swreg77_h264_error_e[6]
55	Inter_view_flag_logic	swreg77_h264_error_e[7]
56	Anchor_pic_flag_logic	swreg77_h264_error_e[8]
57	View_id_logic	swreg77_h264_error_e[9]

For logic error, When swreg1\_int[19] is configured as 1'd0, the hardware will stop the decoder and reset itself. And the error\_type shown above will put on the sw\_strmd\_error\_status[27:0], the current ctu\_x will put on the swreg46\_strmd\_error\_ctu[7:0], the current ctu\_x will put on the swreg46\_strmd\_error\_ctu[15:8].

#### 4.4.13 HEVC Error Process

For hevc, it has only one types of error. When an error is detected, the hardware will stop the decoder and reset itself. And the error\_type in the table 3-3 will put on the sw\_strmd\_error\_status[27:0], the current ctu\_x will put on the swreg46\_strmd\_error\_ctu[7:0], the current ctu\_x will put on the swreg46\_strmd\_error\_ctu[15:8].

#### 4.4.14 MPEG-4 decoder

The features that video decoder supports about MPEG-4/H.263 shows as below.

Table 7-13 MPEG-4/H.263 feature

Feature	Decoder support
Input data format	MPEG-4/H.263 elementary video stream
Decoding scheme	Frame by frame(or field by field) Video packet by video packet
Output data format	YCbCr 4:2:0 semi-planar
Supported image size	48x48 to 1920x1088(MPEG-4, Sorenson Spark) 48x48 to 720x576(H.263) Step size 16 pixels
Maximum frame rate	60fps at 1920x1088
Maximum bit rate	As specified by MPEG-4 ASP level5
Error detection and concealment	Supported

The decoder of MPEG-4/H.263 has two operating modes: in the primary mode the HW performs entropy decoding, and in the secondary mode SW performs entropy decoding. Secondary mode is used in MPEG-4 data partitioned stream decoding.

#### 4.4.15 MPEG-2/MPEG-1 decoder

The features of MPEG-2/MPEG-1 supported by decoder are shown as Table 7-14.

Table 7-14 MPEG-2/MPEG-1 features

Feature	Decoder support
Input data format	MPEG-2/MPEG-1 elementary video stream
Decoding scheme	Frame by frame(or field by field) Video packet by video packet
Output data format	YCbCr 4:2:0 semi-planar

Supported image size	48x48 to 1920x1088 Step size 16 pixels
Maximum frame rate	60fps at 1920x1088
Maximum bit rate	As specified by MPEG-2 MP high level
Error detection and concealment	Supported

The dataflow of MPEG-2/MPEG-1 is the same of H.264 HW performs entropy decoding as 错误!未找到引用源。.

#### 4.4.16 VC-1 decoder

The features of VC-1 supported by decoder are shown as Table 7-15.

Table 7-15 VC-1 features

Feature	Decoder support
Input data format	VC-1
Decoding scheme	Frame by frame(or field by field) Slice by slice
Output data format	YCbCr 4:2:0 semi-planar
Supported image size	48x48 to 1920x1088 Step size 16 pixels
Maximum frame rate	30fps at 1920x1088
Maximum bit rate	As specified by VC-1 AP level3
Error detection and concealment	Supported

The VC-1 decoder has only one operating mode in which the HW performs entropy decoding.

#### 4.4.17 JPEG Decoder

JPEG features supported by decoder are as shown in Table 7-16.

Table 7-16 JPEG features

Feature	Decoder support
Input data format	JFIF file format 1.02 YCbCr 4:0:0, 4:2:0, 4:2:2, 4:4:0, 4:1:1 and 4:4:4 semi-planar
Decoding scheme	Input: buffer by buffer, from 5Kb to 8MB at a time① Output: from 1 MB row to 16 Mpixels at a time②
Output data format	YCbCr 4:0:0, 4:2:0, 4:2:2, 4:4:0, 4:1:1 and 4:4:4 semi-planar
Supported image size	48x48 to 8176x8176(66.8 Mpixels) Step size 8 pixels③
Maximum frame rate	Up to 76 million pixels pre second
Maximum bit rate	As specified by the Divx specification
Thumbnail decoding	JPEG compressed thumbnails supported
Error detection	Supported

①Programmable buffer size for optimizing performance and memory consumption. Interrupt will be issued when buffer runs empty, and the control software will load more streams to external memory.

②Programmable output slice for optimizing performance and memory consumption. Interrupt will be issued when the requested area decoded. The control software can be used to switch the decoder output address each time.

③Non-16x16 dividable resolutions will be filled to 16 pixel boundary.

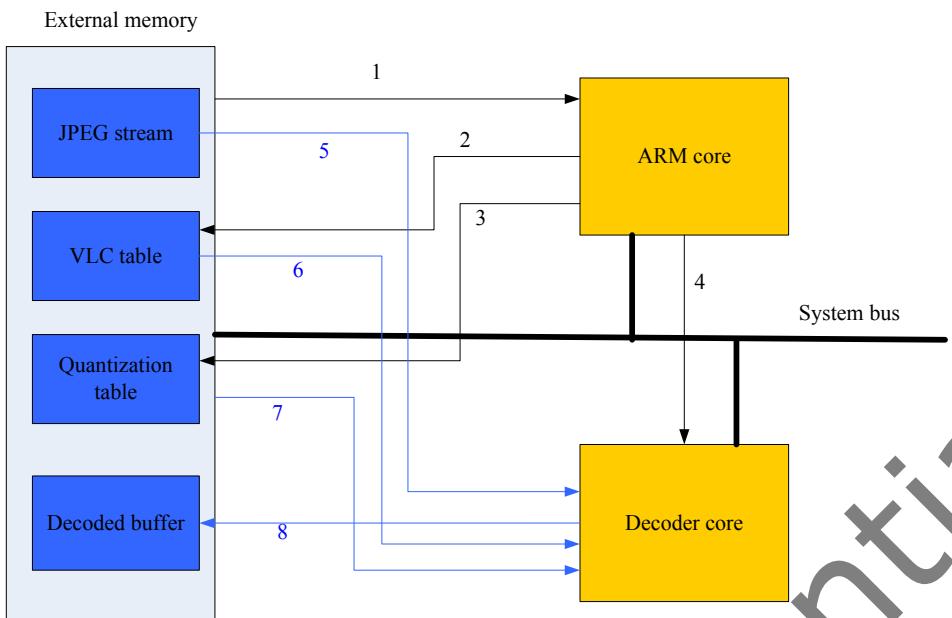


Fig. 7-48 The dataflow of JPEG decoder

The data flow of jpeg decoder is as Fig. 7-48 shown. The decoder software starts decoding the picture by parsing the stream headers (1) and then writes the following items to external memory:

VLC tables (2)

Quantization tables (3)

Last step for the software is to write the hardware control registers and to enable the hardware (4). After starting hardware, SW waits interrupt from HW.

Hardware decodes the picture by reading stream (5), VLC (6) and QP (7) tables. Hardware writes the decoded output picture memory one macroblock at a time (8). When the picture has been fully decoded, or the hardware has run out of stream data, it gives an interrupt with a proper status flag and provides stream end address for software to continue and returns to initial state.

#### 4.4.18 Image Post-processor

The features supported by Post-processor are as show in Table 7-17.

Table 7-17 Post-processor features

Feature	Post-processor support
Input data format	Any format generated by the decoder in combined mode YCbCr 4:2:0 semi-planar YCbCr 4:2:0 planar YCbYCr 4:2:2 YCrYCb 4:2:2 CbYCrY 4:2:2 CrYCbY 4:2:2
Post-processor scheme	Frame by frame. Post-processor handles the image macroblock by macroblock, also in standalone mode.
Input image source	Internal source(combined mode) External source(standalone mode): e.g. a software decoder or camera interface
Output data format	YCbCr 4:2:0 semi-planar YCbCr 4:2:2 YCryCbr 4:2:2 CbYCrY 4:2:2 CrYCbY 4:2:2

	Fully configurable ARGB channel lengths and locations inside 32 bits, e.g. ARGB 32-bit (8-8-8-8), RGB 16-bit(5-6-5), ARGB 16-ibt(4-4-4-4).
Input image size (combined mode)	48x48 to 8176x8176(66.8 Mpixels) Step size 16 pixels
Input image size (stand-alone mode)	Width from 48 to 8176 Height from 48 to 8176 Maximum size limited to 16.7 Mpixels Step size 16 pixels
Output image size	16x16 to 1920x1088 Horizontal step size 8 Vertical step size 2
Image up-scaling①	Bicubic polynomial interpolation with a four-tap horizontal kernel and a two-tap vertical kernel Arbitrary, non-integer scaling ratio, separately for both dimensions. Maximum output width is 3x the input width (within the maximum output image size limit) Maximum output height is 3x the input height - 2 pixels (within the maximum output image size limit) Maximum output height is 2.5x the input height - 2 pixels (within the maximum output image size limit) when running RealVideo, VP8 format in pipeline
Image down-scaling①	Proprietary averaging filter Arbitrary, non-integer scaling ratio separately for both dimensions Unlimited down-scaling ratio
YCbCr to RGB color conversion	BT.601-5 compliant BT.709 compliant User definable conversion coefficient
Dithering	2x2 ordered spatial dithering for 4,5 and 6 bit RGB channel precision
Programmable alpha channel	Constant eight bit value can be set to the alpha channel of the 24-bit RGB output data to control the transparency of the output picture. The resulting 32-bit ARGB data can be used as input data for later alpha blending.
Alpha blending	Output image can be alpha blended with two rectangular areas. YCbCr semi-planar 4:2:0 PP output format is not supported when performing alpha blending. The supported overlay input formats are following. 8 bit alpha value + YCbCr 4:4:4, big endian channel order being A-Y-Cb-Cr, 8 bits each. 8 bit alpha value + 24 bit RGB, big endian channel order being A-R-G-B, 8 bits each
Deinterlacing	Conditional spatial deinterlace filtering. Supports only YCbCr 4:2:0 input format. Usable in stand-alone post-processing mode only.

RGB image contrast adjustment	Segmented linear
RGB image brightness adjustment	Linear
RGB image color saturation adjustment	Linear
De-blocking filter for MPEG-4 simple profile /H.263 /Sorenson	Using a modified H.264 in-loop filter as a post-processing filter. Filtering has to be performed in combined mode
Image cropping / digital zoom	User definable start position, height and width. Can be used with scaling to perform digital zoom. Usable only for JPEG or stand-alone mode.
Picture in picture	Output image can be written to any location inside video memory. Up to 1920x1088 sized displays supported.
Output image masking	Output image writing can be prevented on two rectangular areas in the image. The masking feature is exclusive with alpha blending; however it is possible to have one masking area and one blending area.
Image rotation	Rotation 90,180, or 270 degrees Horizontal flip Vertical flip

① It is not allowed to perform horizontal up-scaling and vertical down-scaling (or vice versa) at the same. If needed, this kind of operation can be performed in two phases.

The PP has two modes: standalone mode and pipe-line mode. In standalone mode, picture processing is performed to any external source. The processing is done independently and asynchronously from the video decoder. The dataflow block gram is as Fig. 7-49 shows.

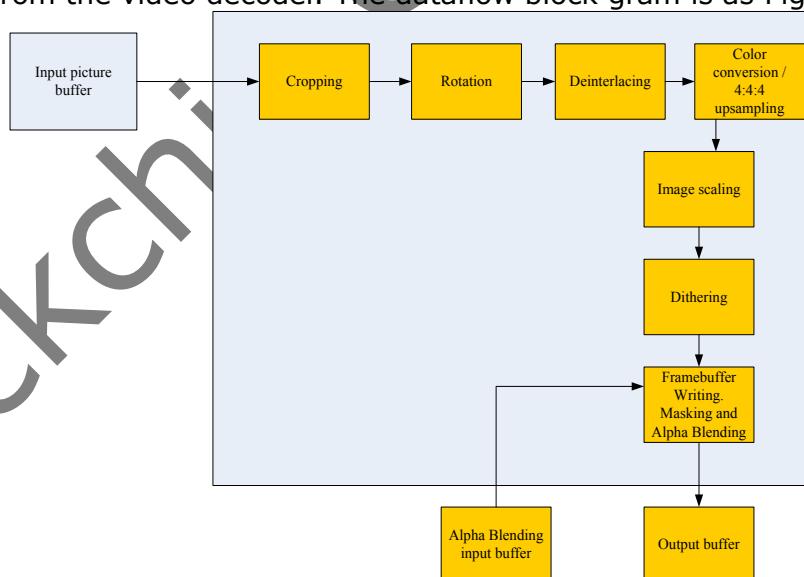


Fig. 7-49 Post-process standalone dataflow

In pipe-line mode, the post-processor works together with the multi-format decoder. The PP will take its input directly from the decoder. The post-processor doesn't have cropping function in pipe-line mode other than combined with jpeg decoder. The dataflow is as Fig. 7-50 show. In the pipe-line mode, most decoder will also put the data to the decoder out buffer other than JPEG decoder. So, JPEG decoder with pipe-line mode will save bus bandwidth when it crops the input picture to a smaller picture.

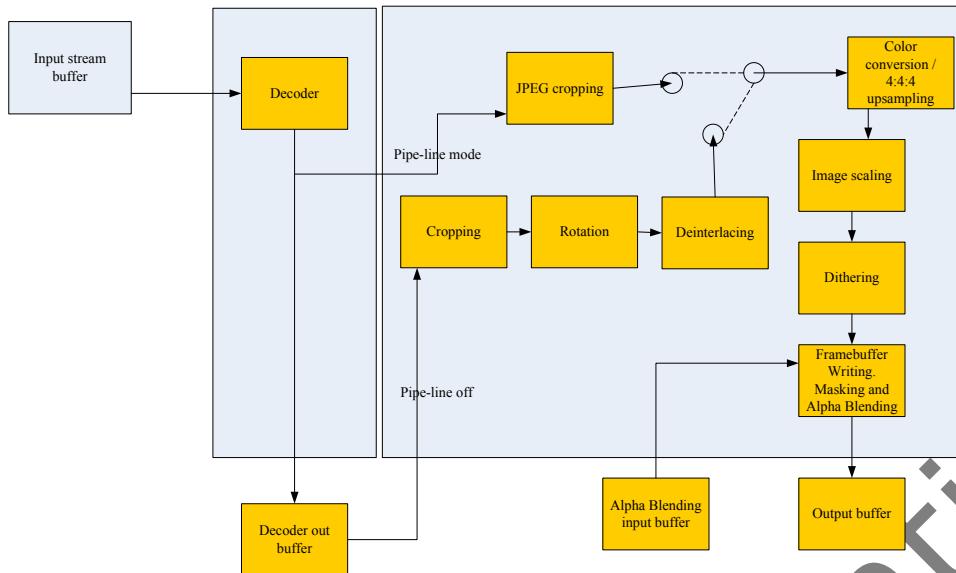


Fig. 7-50 Post-process Pipe-line Mode Dataflow

The post-processor has some restrictions in the input and output picture size. Table 7-18 presents the divisibility requirements for all the post-processor parameters.

Table 7-18 Requirements for post-processor

Output format parameters	YCbCr 4:2:0	YCbCr 4:2:2	RGB16bpp	RGB32bpp
Input picture width and height	16	16	16	16
Cropped picture width and height	8	8	8	8
Cropping start coordinates(x,y)	16	16	16	16
Output picture width	8	8	8	8
Output picture height	2	2	2	2
Masks width and origin X	8	4	4	2
Masks width and origin Y	2	1	1	1
Frame buffer width and origin X	8	4	4	2
Frame buffer height and origin Y	2	1	1	1

#### 4.4.19 Image Pre-processor

Pre-processor is pipelined with the encoder and it can be used only with the encoder. Pre-processor features are presented in Table 7-19.

Table 7-19 Post-processor features

Feature	Encoder support
RGB to YCbCr 4:2:0 color space conversion	BT.601, BT.709 or user defined coefficients conversion for RGB: RGB444 and BGR444 RGB555 and BRG555 RGB565 and BGR565

	RGB888 and BRG888 RGB101010 and BRG101010
YCbCr 4:2:2 to YCbCr 4:2:0 color space conversion	YCbCr formats: YCbCr 4:2:0 planar YCbCr 4:2:0 semi-planar YCbYCr 4:2:2 CbYCrY 4:2:2 interleaved
Cropping	Video – from 8192x8192 to any supported encoding size
Rotation	90 or 270 degrees

#### 4.4.20 H.264 Encoder

The H.264 features supported by encoder are as shown in Table 7-20 .

Table 7-20 Video encoder H.264 feature

Feature	Encoder support
Input data format	YCbCr formats: YCbCr 4:2:0 planar YCbCr 4:2:0 semi-planar YCbYCr 4:2:2① CbYCrY 4:2:2 Interleaved① RGB formats:① RGB444 to BGR444 RGB555 to BGR555 RGB565 to BGR565 RGB888 to BRG888 RGB101010 and BRG 101010
Output data format	H.264: Byte unit stream NAL unit stream
Supported image size	96x96 to 1920x1080(Full HD) Step size 4 pixels
Maximum frame rate	30 fps at 1920 x1080
Bit rate	Maximum 20Mbps Minimum 10kbps

①internally encoder handles image only in 4:2:0 format

Figure Fig. 7-51 illustrates the encoder data flow in H.264 encoding mode. The numbers present the following transactions:

Memory-mapped register writes and reads

Input image read

Reference image write

Reference image read

NAL sizes write from HW

NAL sizes read to SW

Output byte or NAL unit stream write from HW

Output byte or NAL unit stream headers write from SW

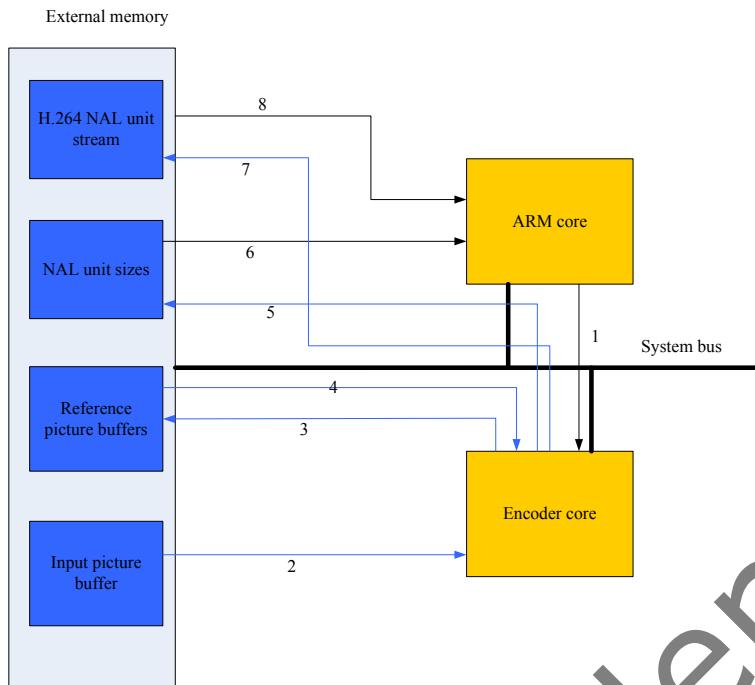


Fig. 7-51 Video Encoder Dataflow

The encoder software starts encoding the first picture by initializing hardware and writing the stream headers. After HW has encoded the image, SW calculates new quantization values for HW, and initializes HW again.

## 4.5 Register Description

### 4.5.1 Internal Address Mapping

This section describes the control/status registers of the codec.

If H264/VP9/HEVC decoder is chosen to work, the register base address is `rkvdec_base`. The reading MMU master register base address is `rkvdec_base+0x480`, the writing MMU register base address is `rkvdec_base + 0x4c0`, and the luma cache control register base address is `rkvdec_base + 0x400`, the chroma cache control register base address is `rkvdec_base + 0x440`.

If vepu is chosen to work, the base address is `vpu_base`. MMU base address is `vpu +0x800`, and cache control base address is `vpu_base + 0xc00`.

If other format is chosen to work, the base address is `vpu_base + 0x400`. MMU base address is `vpu +0x800`, and cache control base address is `vpu_base + 0xc00`.

### 4.5.2 H264/VP9/HEVC Registers Summary

Name	Offset	Size	Reset Value	Description
<code>rkvdec_swreg0_id</code>	0x0000	W	0x68761f00	ID register (read only)
<code>rkvdec_swreg1_int</code>	0x0004	W	0x00000022	interrupt and decoder enable register
<code>rkvdec_swreg2_sysctrl</code>	0x0008	W	0x00000000	Data input and output endian setting and sys ctrl
<code>rkvdec_swreg3_picpar</code>	0x000c	W	0x00000000	picture parameters
<code>rkvdec_swreg4_strm_rlc_base</code>	0x0010	W	0x00000000	the stream or rlc data base address
<code>rkvdec_swreg5_stream_rl_c_len</code>	0x0014	W	0x00000000	amount of stream bytes or rlc data byte in the input buffer or the
<code>rkvdec_swreg6_cabactbl_prob_base</code>	0x0018	W	0x00000000	the base address of cabac table

Name	Offset	Size	Reset Value	Description
rkvdec_swreg7_decout_base	0x001c	W	0x00000000	base address of decoder output picture base address
rkvdec_swreg8_y_virstride	0x0020	W	0x00000000	the ouput picture y virtual stride
rkvdec_swreg9_yuv_virstride	0x0024	W	0x00000000	the ouput picture yuv virtual stride
rkvdec_swreg10_hevc_refer0_base	0x0028	W	0x00000000	base address for reference picture index 0
rkvdec_swreg10_h264_refer0_base	0x0028	W	0x00000000	base address for reference picture index 0
rkvdec_swreg10_vp9_cpr_header_offset	0x0028	W	0x00000000	vp9 compressed header offset
rkvdec_swreg11_hevc_refer1_base	0x002c	W	0x00000000	base address for reference picture index 1
rkvdec_swreg11_h264_refer1_base	0x002c	W	0x00000000	base address for reference picture index 1
rkvdec_swreg11_vp9_refeirlast_base	0x002c	W	0x00000000	base address for last frame
rkvdec_swreg12_hevc_refer2_base	0x0030	W	0x00000000	base address for reference picture index 2
rkvdec_swreg12_h264_refer2_base	0x0030	W	0x00000000	base address for reference picture index 2
rkvdec_swreg12_vp9_refergolden_base	0x0030	W	0x00000000	base address for golden frame
rkvdec_swreg13_hevc_refer3_base	0x0034	W	0x00000000	base address for reference picture index 3
rkvdec_swreg13_h264_refer3_base	0x0034	W	0x00000000	base address for reference picture index 3
rkvdec_swreg13_vp9_referalfter_base	0x0034	W	0x00000000	base address for referalfter frame
rkvdec_swreg14_hevc_refer4_base	0x0038	W	0x00000000	base address for reference picture index 4
rkvdec_swreg14_h264_refer4_base	0x0038	W	0x00000000	base address for reference picture index 4
rkvdec_swreg14_vp9count_base	0x0038	W	0x00000000	vp9 count base addr
rkvdec_swreg15_h264_refer5_base	0x003c	W	0x00000000	base address for reference picture index 5
rkvdec_swreg15_vp9_segidlast_base	0x003c	W	0x00000000	base address for last frame segment id
rkvdec_swreg15_hevc_refer5_base	0x003c	W	0x00000000	base address for reference picture index 5
rkvdec_swreg16_h264_refer6_base	0x0040	W	0x00000000	base address for reference picture index 6

<b>Name</b>	<b>Offset</b>	<b>Size</b>	<b>Reset Value</b>	<b>Description</b>
rkvdec_swreg16_vp9_segi_dcur_base	0x0040	W	0x00000000	base address for cur frame segment id
rkvdec_swreg16_hevc_refer6_base	0x0040	W	0x00000000	base address for reference picture index 6
rkvdec_swreg17_h264_refer7_base	0x0044	W	0x00000000	base address for reference picture index 7
rkvdec_swreg17_vp9_frame_size_last	0x0044	W	0x00000000	vp9 last frame size
rkvdec_swreg17_hevc_refer7_base	0x0044	W	0x00000000	base address for reference picture index 7
rkvdec_swreg18_h264_refer8_base	0x0048	W	0x00000000	base address for reference picture index 8
rkvdec_swreg18_vp9_frame_size_golden	0x0048	W	0x00000000	vp9 golden frame size
rkvdec_swreg18_hevc_refer8_base	0x0048	W	0x00000000	base address for reference picture index 8
rkvdec_swreg19_h264_refer9_base	0x004c	W	0x00000000	base address for reference picture index 9
rkvdec_swreg19_vp9_frame_size_altref	0x004c	W	0x00000000	vp9 alfter frame size
rkvdec_swreg19_hevc_refer9_base	0x004c	W	0x00000000	base address for reference picture index 9
rkvdec_swreg20_h264_refer10_base	0x0050	W	0x00000000	base address for reference picture index 10
rkvdec_swreg20_vp9_segid_grp0	0x0050	W	0x00000000	vp9 segid syntax grp0
rkvdec_swreg20_hevc_refer10_base	0x0050	W	0x00000000	base address for reference picture index 10
rkvdec_swreg21_h264_refer11_base	0x0054	W	0x00000000	base address for reference picture index 11
rkvdec_swreg21_vp9_segid_grp1	0x0054	W	0x00000000	vp9 segid syntax grp1
rkvdec_swreg21_hevc_refer11_base	0x0054	W	0x00000000	base address for reference picture index 11
rkvdec_swreg22_h264_refer12_base	0x0058	W	0x00000000	base address for reference picture index 12
rkvdec_swreg22_vp9_segid_grp2	0x0058	W	0x00000000	vp9 segid syntax grp2
rkvdec_swreg22_hevc_refer12_base	0x0058	W	0x00000000	base address for reference picture index 12
rkvdec_swreg23_h264_refer13_base	0x005c	W	0x00000000	base address for reference picture index 13
rkvdec_swreg23_vp9_segid_grp3	0x005c	W	0x00000000	vp9 segid syntax grp3

<b>Name</b>	<b>Offset</b>	<b>Size</b>	<b>Reset Value</b>	<b>Description</b>
rkvdec_swreg23_hevc_ref er13_base	0x005c	W	0x00000000	base address for reference picture index 13
rkvdec_swreg24_h264_re fer14_base	0x0060	W	0x00000000	base address for reference picture index 14
rkvdec_swreg24_vp9_segi d_grp4	0x0060	W	0x00000000	vp9 segid syntax grp4
rkvdec_swreg24_hevc_ref er14_base	0x0060	W	0x00000000	base address for reference picture index 14
rkvdec_swreg25_vp9_segi d_grp5	0x0064	W	0x00000000	vp9 segid syntax grp5
rkvdec_swreg25_refer0_p oc	0x0064	W	0x00000000	the poc of reference picture index 0
rkvdec_swreg26_vp9_segi d_grp6	0x0068	W	0x00000000	vp9 segid syntax grp6
rkvdec_swreg26_refer1_p oc	0x0068	W	0x00000000	the poc of reference picture index 1
rkvdec_swreg27_vp9_segi d_grp7	0x006c	W	0x00000000	vp9 segid syntax grp7
rkvdec_swreg27_refer2_p oc	0x006c	W	0x00000000	the poc of reference picture index 2
rkvdec_swreg28_vp9_cpr header_config	0x0070	W	0x00000000	vp9 compressed header config info
rkvdec_swreg28_refer3_p oc	0x0070	W	0x00000000	the poc of reference picture index 3
rkvdec_swreg29_vp9_lref _scale	0x0074	W	0x00000000	scaling factor for last reference picture
rkvdec_swreg29_refer4_p oc	0x0074	W	0x00000000	the poc of reference picture index 4
rkvdec_swreg30_vp9_gref _scale	0x0078	W	0x00000000	scaling factor for golden reference picture
rkvdec_swreg30_refer5_p oc	0x0078	W	0x00000000	the poc of reference picture index 5
rkvdec_swreg31_vp9_aref _scale	0x007c	W	0x00000000	scaling factor for alfter reference picture
rkvdec_swreg31_refer6_p oc	0x007c	W	0x00000000	the poc of reference picture index 6
rkvdec_swreg32_vp9_ref _deltas_lastframe	0x0080	W	0x00000000	vp9 ref deltas
rkvdec_swreg32_refer7_p oc	0x0080	W	0x00000000	the poc of reference picture index 7
rkvdec_swreg33_vp9_info _lastframe	0x0084	W	0x00000000	vp9 info for lastframe
rkvdec_swreg33_refer8_p oc	0x0084	W	0x00000000	the poc of reference picture index 8

Name	Offset	Size	Reset Value	Description
rkvdec_swreg34_vp9_intercmd_base	0x0088	W	0x00000000	inter cmd base addr
rkvdec_swreg34_refer9_poc	0x0088	W	0x00000000	the poc of reference picture index 9
rkvdec_swreg35_vp9_intercmd_num	0x008c	W	0x00000000	vp9 intercmd num
rkvdec_swreg35_refer10_poc	0x008c	W	0x00000000	the poc of reference picture index 10
rkvdec_swreg36_vp9_last_tile_size	0x0090	W	0x00000000	vp9 lasttile size
rkvdec_swreg36_refer11_poc	0x0090	W	0x00000000	the poc of reference picture index 11
rkvdec_swreg37_vp9_last_f_hor_virstride	0x0094	W	0x00000000	Register0000 Abstract
rkvdec_swreg37_refer12_poc	0x0094	W	0x00000000	the poc of reference picture index 12
rkvdec_swreg38_vp9_golденf_hor_virstride	0x0098	W	0x00000000	vp9 golden frame horizontal virstride
rkvdec_swreg38_refer13_poc	0x0098	W	0x00000000	the poc of reference picture index 13
rkvdec_swreg39_vp9_altref_eff_hor_virstride	0x009c	W	0x00000000	vp9 altref frame horizontal virstride
rkvdec_swreg39_refer14_poc	0x009c	W	0x00000000	the poc of reference picture index 14
rkvdec_swreg40_cur_poc	0x00a0	W	0x00000000	the poc of cur picture
rkvdec_swreg41_rlcwrite_base	0x00a4	W	0x00000000	the base address or rlcwrite base addr
rkvdec_swreg42_pps_base	0x00a8	W	0x00000000	the base address of pps
rkvdec_swreg43_rps_base	0x00ac	W	0x00000000	the base address of rps
rkvdec_swreg44_strmd_error_en	0x00b0	W	0x00000000	cabac error enable config
rkvdec_swreg45_strmd_error_status	0x00b4	W	0x00000000	cabac error status
rkvdec_swreg45_vp9_error_info0	0x00b4	W	0x00000000	vp9 error info
rkvdec_swreg46_strmd_error_ctu	0x00b8	W	0x00400000	strmd error ctu
rkvdec_swreg47_sao_ctu_position	0x00bc	W	0x00000000	sao ctu position
rkvdec_swreg48_h264_refer15_base	0x00c0	W	0x00000000	base address for reference picture index 15
rkvdec_swreg48_vp9_last_ystride	0x00c0	W	0x00000000	last ref ystride

Name	Offset	Size	Reset Value	Description
rkvdec_swreg49_h264_refer15_poc	0x00c4	W	0x00000000	the poc of reference picture index 15
rkvdec_swreg49_vp9_golden_ystride	0x00c4	W	0x00000000	golden ref ystride
rkvdec_swreg50_h264_refer16_poc	0x00c8	W	0x00000000	the poc of reference picture index 16
rkvdec_swreg50_vp9_altryStride	0x00c8	W	0x00000000	altref ref ystride
rkvdec_swreg51_h264_refer17_poc	0x00cc	W	0x00000000	the poc of reference picture index 17
rkvdec_swreg51_vp9_lastRef_yuvstride	0x00cc	W	0x00000000	vp9 lastref yuv stride
rkvdec_swreg52_vp9_refcolmv_base	0x00d0	W	0x00000000	vp9 colmv ref base
rkvdec_swreg52_h264_refer18_poc	0x00d0	W	0x00000000	the poc of reference picture index 18
rkvdec_swreg53_h264_refer19_poc	0x00d4	W	0x00000000	the poc of reference picture index 19
rkvdec_swreg54_h264_refer20_poc	0x00d8	W	0x00000000	the poc of reference picture index 20
rkvdec_swreg55_h264_refer21_poc	0x00dc	W	0x00000000	the poc of reference picture index 21
rkvdec_swreg56_h264_refer22_poc	0x00e0	W	0x00000000	the poc of reference picture index 22
rkvdec_swreg57_h264_refer23_poc	0x00e4	W	0x00000000	the poc of reference picture index 23
rkvdec_swreg58_h264_refer24_poc	0x00e8	W	0x00000000	the poc of reference picture index 24
rkvdec_swreg59_h264_refer25_poc	0x00ec	W	0x00000000	the poc of reference picture index 25
rkvdec_swreg60_h264_refer26_poc	0x00f0	W	0x00000000	the poc of reference picture index 26
rkvdec_swreg61_h264_refer27_poc	0x00f4	W	0x00000000	the poc of reference picture index 27
rkvdec_swreg62_h264_refer28_poc	0x00f8	W	0x00000000	the poc of reference picture index 28
rkvdec_swreg63_h264_refer29_poc	0x00fc	W	0x00000000	the poc of reference picture index 29
rkvdec_swreg64_performance_cycle	0x0100	W	0x00000000	hevc performance cycle
rkvdec_swreg65_axi_ddr_rdata	0x0104	W	0x00000000	axi ddr read data num
rkvdec_swreg66_axi_ddr_wdata	0x0108	W	0x00000000	axi ddr write data number

Name	Offset	Size	Reset Value	Description
rkvdec_swreg68_perform ance_sel	0x0110	W	0x00000000	
rkvdec_swreg69_perform ance_cnt0	0x0114	W	0x00000000	
rkvdec_swreg70_perform ance_cnt1	0x0118	W	0x00000000	
rkvdec_swreg71_perform ance_cnt2	0x011c	W	0x00000000	
rkvdec_swreg72_h264_re fer30_poc	0x0120	W	0x00000000	the poc of reference picture index 30
rkvdec_swreg73_h264_re fer31_poc	0x0124	W	0x00000000	the poc of reference picture index 31
rkvdec_swreg74_h264_cu r_poc1	0x0128	W	0x00000000	h264 cur poc for bottom filed
rkvdec_swreg75_h264_er rorinfo_base	0x012c	W	0x00000000	h264 error info base addr
rkvdec_swreg76_h264_er rorinfo_num	0x0130	W	0x00000000	h264 error info num
rkvdec_swreg77_h264_er ror_e	0x0134	W	0x00000000	h264 error enable high bits

Notes:**S**ize:**B**- Byte (8 bits) access, **H**W- Half WORD (16 bits) access, **W**-WORD (32 bits) access

Name	Offset	Size	Reset Value	Description
VDPU_SWREG0_NEW_VE RSION	0x0000	W	0x03680000	ID register(read only)
VDPU_SWREG0	0x0000	W	0x00000000	Register0000 Abstract
VDPU_SWREG1	0x0004	W	0x00000000	interrupt register decoder
VDPU_SWREG2	0x0008	W	0x01000400	device configuration register decoder
VDPU_SWREG3	0x000c	W	0x00000001	Device control register 0(deemode, picture type etc)
VDPU_SWREG4_H264	0x0010	W	0x00000000	decoder control register 1(picture parameters)
VDPU_SWREG4	0x0010	W	0x00000000	decoder control register 1(picture parameters)
VDPU_SWREG5	0x0014	W	0x00000000	decoder control register2 (stream decoding table selects)
VDPU_SWREG5_H264	0x0014	W	0x00000000	decoder control register2 (stream decoding table selects)
VDPU_SWREG6	0x0018	W	0x00000000	decoder control register 3( stream buffer information)
VDPU_SWREG7	0x001c	W	0x00000000	decoder control register 4(H264)
VDPU_SWREG8	0x0020	W	0x00000000	decoder control register 5
VDPU_SWREG9	0x0024	W	0x00000000	decoder control register 6
VDPU_SREG10_H264_RLC	0x0028	W	0x00000000	Base address for differential motion vector base address
VDPU_SREG10_H264	0x0028	W	0x00000000	Base address for differential motion vector base address
VDPU_SWREG11_H264_R LC	0x002c	W	0x00000000	decoder control register 7
VDPU_SWREG11_H264	0x002c	W	0x00000000	decoder control register 7

Name	Offset	Size	Reset Value	Description
VDPU_SWREG12	0x0030	W	0x00000000	Base address for RLC data (RLC) / stream start address/decoded
VDPU_SWREG13	0x0034	W	0x00000000	Base address for decoded picture / base address for JPEG deco
VDPU_SWREG14	0x0038	W	0x00000000	Base address for reference picture index 0 / base address for J
VDPU_SWREG15_JPEG_ROI	0x003c	W	0x00000000	JPEG roi control
VDPU_SWREG15	0x003c	W	0x00000000	Base address for reference picture index 1 / JPEG control
VDPU_SWREG16	0x0040	W	0x00000000	base address for reference picture index 2 / List of VLC code len
VDPU_SWREG17	0x0044	W	0x00000000	Base address for reference picture index 3 / List of VLC code le
VDPU_SWREG18	0x0048	W	0x00000000	Base address for reference picture index 4 /MPE
VDPU_SWREG19	0x004c	W	0x00000000	Base address for reference picture index 5
VDPU_SWREG20	0x0050	W	0x00000000	Base address for reference picture index 6
VDPU_SWREG21	0x0054	W	0x00000000	Base address for reference picture index 7
VDPU_SWREG22	0x0058	W	0x00000000	Base address for reference picture index 8
VDPU_SWREG23	0x005c	W	0x00000000	Base address for reference picture index 9
VDPU_SWREG24	0x0060	W	0x00000000	Base address for reference picture index 10
VDPU_SWREG25	0x0064	W	0x00000000	Base address for reference picture index 11
VDPU_SWREG26	0x0068	W	0x00000000	Base address for reference picture index 12
VDPU_SWREG27	0x006c	W	0x00000000	Base address for reference picture index 13
VDPU_SWREG28	0x0070	W	0x00000000	Base address for reference picture index14
VDPU_SWREG29	0x0074	W	0x00000000	Base address for reference picture index15
VDPU_SWREG30	0x0078	W	0x00000000	Reference picture numbers for index 0 and 1 (H264 VLC)
VDPU_SWREG31	0x007c	W	0x00000000	Reference picture numbers for index 2 and 3 (H264 VLC) /
VDPU_SWREG32	0x0080	W	0x00000000	Reference picture numbers for index 4 and 5 (H264 VLC)
VDPU_SWREG33	0x0084	W	0x00000000	Reference picture numbers for index 6 and 7 (H264 VLC)
VDPU_SWREG34	0x0088	W	0x00000000	Reference picture numbers for index 8 and 9 (H264 VLC)
VDPU_SWREG35_JPEG_ROI	0x008c	W	0x00000000	JPEG roi offset/dc base address
VDPU_SWREG35	0x008c	W	0x00000000	Reference picture numbers for index 10 and 11 (H264 VLC)
VDPU_SWREG36	0x0090	W	0x00000000	Reference picture numbers for index 12 and 13 (H264 VLC)

Name	Offset	Size	Reset Value	Description
VDPU_SWREG36_JPEG_ROI	0x0090	W	0x00000000	JPEG roi offset/dc length
VDPU_SWREG37	0x0094	W	0x00000000	Reference picture numbers for index 14 and 15 (H264 VLC)
VDPU_SWREG38	0x0098	W	0x00000000	Reference picture long term flags (H264 VLC) prediction filt
VDPU_SWREG38_H264	0x0098	W	0x00000000	Reference picture numbers for index 12 and 13 (H264 VLC)
VDPU_SWREG39	0x009c	W	0x00000000	Reference picture valid flags (H264 VLC) prediction filter ta
VDPU_SWREG39_H264	0x009c	W	0x00000000	Reference picture numbers for index 12 and 13 (H264 VLC)
VDPU_SWREG40	0x00a0	W	0x00000000	Base address for standard dependent tables
VDPU_SWREG41	0x00a4	W	0x00000000	Base address for direct mode motion vectors
VDPU_SWREG42	0x00a8	W	0x00000000	bi_dir initial ref pic list register (0-2) prediction filter taps
VDPU_SWREG43	0x00ac	W	0x00000000	bi-dir initial ref pic list register (3-5) prediction filter taps
VDPU_SWREG44	0x00b0	W	0x00000000	bi-dir initial ref pic list register (6-8) prediction filter taps
VDPU_SWREG45	0x00b4	W	0x00000000	bi-dir initial ref pic list register (9-11) prediction filter taps
VDPU_SWREG46	0x00b8	W	0x00000000	bi-dir initial ref pic list register (12-14)
VDPU_SWREG47	0x00bc	W	0x00000000	bi-dir and P fwd initial ref pic list register (15 and P 0-3)
VDPU_SWREG48	0x00c0	W	0x00000000	Error concealment register
VDPU_SWREG49	0x00c4	W	0x00000000	Prediction filter tap register for H264, MPEG4
VDPU_SWREG50	0x00c8	W	0xfbb56f80	Synthesis configuration register decoder 0 (read only)
VDPU_SWREG51	0x00cc	W	0x00000000	Reference picture buffer control register
VDPU_SWREG52	0x00d0	W	0x00000000	Reference picture buffer information register 1 (read only)
VDPU_SWREG53	0x00d4	W	0x00000000	Reference picture buffer information register 2 (read only)
VDPU_SWREG54	0x00d8	W	0xe5da0000	Synthesis configuration register decoder 1 (read only)
VDPU_SWREG55	0x00dc	W	0x00000000	Reference picture buffer 2 / Advanced prefetch control register
VDPU_SWREG56	0x00e0	W	0x00000000	Reference buffer information register 3 (read only)
VDPU_SWREG57_INTRA_INTER	0x00e4	W	0x00000000	intra_dll3t,intra_dblspeed,inter_dblspeed,stream_len_hi
VDPU_SWREG57	0x00e4	W	0x00000000	intra_dll3t,intra_dblspeed,inter_dblspeed,stream_len_hi
VDPU_SWREG58	0x00e8	W	0x00000000	Decoder debug register 0 (read only)
VDPU_SWREG59	0x00ec	W	0x00000000	H264 Chrominance 8 pixel interleaved data base
VDPU_SWREG60	0x00f0	W	0x00000000	Interrupt register post-processor

Name	Offset	Size	Reset Value	Description
VDPU_SWREG61	0x00f4	W	0x01010100	Device configuration register post-processor
VDPU_SWREG62	0x00f8	W	0x00000000	Deinterlace control register
VDPU_SWREG63	0x00fc	W	0x00000000	base address for reading post-processing input picture uminan
VDPU_SWREG64	0x0100	W	0x00000000	Base address for reading post-processing input picture Cb/Ch
VDPU_SWREG65	0x0104	W	0x00000000	Base address for reading post-processing input picture Cr
VDPU_SWREG66	0x0108	W	0x00000000	Base address for writing post-processed picture luminance/RGB
VDPU_SWREG67	0x010c	W	0x00000000	Base address for writing post-processed picture Ch
VDPU_SWREG68	0x0110	W	0x00000000	Register for contrast adjusting
VDPU_SWREG69	0x0114	W	0x00000000	Register for colour conversion and contrast adjusting
VDPU_SWREG70	0x0118	W	0x00000000	Register for colour conversion 0
VDPU_SWREG71	0x011c	W	0x00000000	Register for colour conversion 1 + rotation mode
VDPU_SWREG72	0x0120	W	0x00000000	PP input size and -cropping register
VDPU_SWREG73	0x0124	W	0x00000000	PP input picture base address for Y bottom field
VDPU_SWREG74	0x0128	W	0x00000000	PP input picture base for Ch bottom field
VDPU_SWREG79	0x013c	W	0x00000000	Scaling ratio register 1 & padding for B
VDPU_SWREG80	0x0140	W	0x00000000	Scaling register 0 ratio & padding for R and G
VDPU_SWREG81	0x0144	W	0x00000000	Scaling ratio register 2
VDPU_SWREG82	0x0148	W	0x00000000	Rmask register
VDPU_SWREG83	0x014c	W	0x00000000	Gmask register
VDPU_SWREG84	0x0150	W	0x00000000	Bmask register
VDPU_SWREG85	0x0154	W	0x00000000	Post-processor control register
VDPU_SWREG86	0x0158	W	0x00000000	Mask 1 start coordinate register
VDPU_SWREG87	0x015c	W	0x00000000	Mask 2 start coordinate register
VDPU_SWREG88	0x0160	W	0x00000000	Mask 1 size and PP original width register
VDPU_SWREG89	0x0164	W	0x00000000	Mask 2 size register
VDPU_SWREG90	0x0168	W	0x00000000	PiP register 0
VDPU_SWREG91	0x016c	W	0x00000000	PiP register 1 and dithering control
VDPU_SWREG92	0x0170	W	0x00000000	Display width and PP input size extension register
VDPU_SWREG93	0x0174	W	0x00000000	Display width and PP input size extension register
VDPU_SWREG94	0x0178	W	0x00000000	Base address for alpha blend 2 gui component
VDPU_SWREG95	0x017c	W	0x00000000	Base address for alpha blend 2 gui component
VDPU_SWREG98	0x0188	W	0x00000000	PP output width/height extension
VDPU_SWREG99	0x018c	W	0xe000f000	PP fuse register (read only)

Name	Offset	Size	Reset Value	Description
VDPU_SWREG100	0x0190	W	0xff874780	Synthesis configuration register post-processor (read only)
VDPU_SWREG101	0x0194	W	0x00000000	soft reset signals
VDPU_SWREG102	0x0198	W	0x00000000	vpu performance cycle
VDPU_SWREG103	0x019c	W	0x00000000	AXI DDR READ DATA NUM
VDPU_SWREG104	0x01a0	W	0x00000000	Register0000 Abstract
VDPU_SWREG105	0x01a4	W	0x00000000	
VDPU_SWREG106	0x01a8	W	0x00000000	
VDPU_SWREG107	0x01ac	W	0x00000000	

Notes: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

Name	Offset	Size	Reset Value	Description
V_CODEC_MMU_DTE_ADDR	0x0000	W	0x00000000	MMU current page Table address
V_CODEC_MMU_STATUS	0x0004	W	0x00000018	MMU status register
V_CODEC_MMU_COMMAND	0x0008	W	0x00000000	MMU command register
V_CODEC_MMU_PAGE_FAULT_ADDR	0x000c	W	0x00000000	MMU logical address of last page fault
V_CODEC_MMU_ZAP_ONE_LINE	0x0010	W	0x00000000	MMU Zap cache line register
V_CODEC_MMU_INT_RAW_STAT	0x0014	W	0x00000000	MMU raw interrupt status register
V_CODEC_MMU_INT_CLEA_R	0x0018	W	0x00000000	MMU raw interrupt status register
V_CODEC_MMU_INT_MASK	0x001c	W	0x00000000	MMU raw interrupt status register
V_CODEC_MMU_INT_STATUS	0x0020	W	0x00000000	MMU raw interrupt status register
V_CODEC_MMU_AUTO_GATING	0x0024	W	0x00000001	mmu auto gating

Notes: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

Name	Offset	Size	Reset Value	Description
pref_cache_VERSION	0x0000	W	0xcac20101	VERSION register
pref_cache_SIZE	0x0004	W	0x06110206	L2 cache SIZE
pref_cache_STATUS	0x0008	W	0x00000000	Status register
pref_cache_COMMAND	0x0010	W	0x00000000	Command setting register
pref_cache_CLEAR_PAGE	0x0014	W	0x00000000	clear page register
pref_cache_MAX_READS	0x0018	W	0x0000001c	maximum read register
pref_cache_PERFCNT_SRC0	0x0020	W	0x00000000	performance counter 0 source register
pref_cache_PERFCNT_VAL0	0x0024	W	0x00000000	performance counter 0 value register
pref_cache_PERFCNT_SRC1	0x0028	W	0x00000000	performance counter 0 source register
pref_cache_PERFCNT_VAL1	0x002c	W	0x00000000	performance counter 1 value register

Notes: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

#### 4.5.3 H264/VP9/HEVC Detail Register Description

##### rkvdec\_swreg0\_id

Address: Operational Base + offset (0x0000)

ID register (read only)

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:16	RO	0x6876	prod_num product number The ascii code of "hv", which is 0x6876
15	RO	0x0	reserved
14	RO	0x0	codec_flag codec flag 0: only dec 1: dec + enc
13	RO	0x0	reserved
12	RW	0x1	profile hevc profile 0: Main 1: Main10
11:9	RO	0x7	dec_support dec support bits bit0: HEVC support or not, when it is 1'b1, support bit1: H264 support or not bit2: VP9 support or not
8	RO	0x1	level level 0: FHD 1: UHD
7:0	RO	0x00	minor_ver minor version minor version

**rkvdec\_swreg1\_int**

Address: Operational Base + offset (0x0004)

interrupt and decoder enable register

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22	RW	0x0	sw_softreset_rdy when it is 1'b1, it says that softreset has been done
21	RO	0x0	reserved
20	RW	0x0	sw_softrst_en_p softreset enable softreset enable signal write 1 to soft reset, write 0 invalid puls register

Bit	Attr	Reset Value	Description
19	RW	0x0	<p>sw_h264orvp9_error_mode h264 or vp9 error mode for VP9: 1'b0: when there is any stream or colmv error, the hardware will stop the decode and reset itself 1'b1: when there is any stream or colmv error, the hardware will still decode the next slice it is recommend that when vp9 , it is configed to 1'b0 for H264: 1'b0: when there is any stream , the hardware will stop the decoder and reset itself 1'b1: when there is any stream error, the hardware will wait the end signal of deblocking and then reset itself</p>
18	RW	0x0	<p>sw_cabu_end_sta cabac decode end status hevc: cabac decode end status h264&amp; vp9 : streamd decode status</p>
17	RW	0x0	<p>sw_colmv_ref_error_sta colmv ref error status colmv ref error status hevc&amp;vp9: when it is 1'b1, it means that inter module read the invalid dpb frame It will self reset the hardware h264: when it is 1'b1, it means that inter module read the invalid dpb frame. when sw_h264_error_mode is 1'b0, it will self reset the hardware, otherwise it will not</p>
16	RW	0x0	<p>sw_buf_empty_sta buffer empty sta buffer empty status, only when sw_buf_empty_en is 1'b1 , this bit is valid, now is for no valid</p>
15	RW	0x0	<p>sw_dec_timeout_sta decoder timeout interrupt status When high the decoder has been idling for too long. it will self reset the hardware only when sw_dec_timeout_e is 1'b1, this bit is valid</p>
14	RW	0x0	<p>sw_dec_error_sta status bit of input stream error hevc &amp; vp9:when high, an error is found in input data stream decoding. It will self reset the hardware. h264: when high, an error is found in input data stream decoding.when sw_h264_error_mode is 1'b0, it will self reset the hardware, otherwise it will not</p>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
13	RW	0x0	sw_dec_bus_sta bus error status When this bit is high, there is error on the axi bus, it will self reset hardware
12	RW	0x0	sw_dec_rdy_sta decoder ready status when this bit is high, decoder has decoded a picture( the loop filter module send out a frame rdy)
11	RO	0x0	reserved
10	WO	0x0	sw_dec_e_rewrite_valid sw_de_e rewrite valid signal sw_dec_e rewrite valid signal maybe for only when buffer empty, restart the decoder use
9	RW	0x0	sw_dec_irq_raw the raw status of sw_dec_irq the raw status of sw_dec_irq, SW should reset this bit after interrupt is handled
8	RO	0x0	sw_dec_irq decoder IRQ when high, decoder requests an interrupt. $sw\_dec\_irq = sw\_dec\_irq\_raw \&& (sw\_dec\_irq\_dis == 1'b0)$
7	RW	0x0	sw_stmerror_waitdecfifo_empty whether the stream error process wait the decfifo empty when it is 1'b0, the stream error process will no wait the ca2decfifo empty when it is 1'b1, the stream error process will wait the ca2decfifo empty when sw_dec_mode is HEVC and VP9, it always take effect; when sw_dec_mode is H264, it only take effect when sw_h264_error_mode is 1'b0
6	RW	0x0	sw_buf_empty_en buffer empty int enable buffer empty interrupt enable, now is for no use
5	RW	0x1	sw_dec_timeout_e Timeout interrupt enable If enabled HW may return timeout interrupt in case HW gets stucked while decoding picture.
4	RW	0x0	sw_dec_irq_dis decoder IRQ disable When hight, there are no interrupts concerning decoder from HW. Polling must be used to see the interrupt status

Bit	Attr	Reset Value	Description
3	RW	0x0	sw_timeout_mode time out mode timeout mode select 1'b0: TIMEOUT_CYCLES is 241'b1 1'b1: TIMEOUT_CYCLES is 181'b1
2	RO	0x0	reserved
1	RW	0x1	sw_dec_clkgate_e decoder dynamic clock gating enable 0 = clock is running for all structures 1 = clock is gated for decoder structures that are not used
0	RW	0x0	sw_dec_e decoder enable Decoder enable. Setting this bit high will start the decoding operation. HW will reset this when the picture is decoded ready or bus error or time out interrupt is given for all decode format. HW will reset this when picture is processed stream error for vp9 & hevc & (h264 when sw_h264_error_mode is 1'b0)

**rkvdec\_swreg2\_sysctrl**

Address: Operational Base + offset (0x0008)

Data input and output endian setting and sys ctrl

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28	RW	0x0	sw_h264_frame_orslice h264 frame or slice for H264 use 1'b0: frame 1'b1: slice when sw_h264_streamd_mode is 1'b0, this register is valid
27	RW	0x0	sw_h264_firstslice_flag firstslice flag 1'b1: first packet in the frame, for h264 decode to read rps/pps data because the first_mb_in_slice may be wrong, so need this syntax
26	RW	0x0	sw_h264_stream_lastpacket stream last packet flag when sw_h264_stream_mode is 1'b1 sw_h264_stream_lastpacket 1'b0: this packet is not the last packet of frame 1'b1: the packet is the last packet of frame
25	RW	0x0	sw_h264_stream_mode h264 stream mode 1'b0: stream packet is slice by slice or frame by frame, should use sw_h264_frame_orslice 1'b1: stream packet is random, should use sw_h264_stream_last

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
24	RW	0x0	sw_h264_rps_mode h264 rps mode 1'b0: hardware parse rps mode 1'b1: software parse rps mode
23:22	RO	0x0	reserved
21:20	RW	0x0	sw_dec_mode dec mode 2'd0: hevc 2'd1: h264 2'd2: vp9
19	RO	0x0	reserved
18:12	RW	0x00	sw_strm_start_bit exact bit of stream start exact bit of streamd start word where decoding can be started (asosiates with sw_str_rlc_base)
11	RW	0x0	sw_rlc_mode rlc mode enable 0 = HW decodes video from bit stream 1 = HW decodes video from RLC input data
10	RW	0x0	sw_rlc_mode_direct_write cabac decode output direct write cabac decode output direct write enable when this bit is enable , all the module other than cabac and busifd are not work
9	RO	0x0	reserved
8	RW	0x0	sw_out_cbcr_swap output cbcr swap 1'b0: cb(u) is in the lower address, cr(v) is in the higher address 1'b1: cb(u) is in the higher address,cr(v) is in the lower address sw_in_cbcr_swap is the same with sw_out_cbcr_swap
7	RW	0x0	sw_out_swap32_e decoder output data and dpb input data 32bit swap may be used for 64 or 128 bit environment 0 = no swapping of 32 bit words 1 = 32 bit data words are swapped
6	RW	0x0	sw_out_endian dec output data and colmv , dpb data and colmv input endian 0 = little endian 1 = big endian for litter enadian , a data 0x12345678, 0x78 is stored in lower address, 0x12 is stored in higher address

Bit	Attr	Reset Value	Description
5	RW	0x0	sw_str_swap64_e stream 64bit data swap may be used for 128 bit environment 0 = no swapping of 64 bit words 1 = 64 bit data words are swapped
4	RW	0x0	sw_str_swap32_e stream 32bit data swap may be used for 64 or 128 bit environment 0 = no swapping of 32 bit words 1 = 32 bit data words are swapped
3	RW	0x0	sw_str_endian stream data input endian mode 0 = little endian 1 = big endian for litter enadian , a data 0x12345678, 0x78 is stored in lower address, 0x12 is stored in higher address
2	RW	0x0	sw_in_swap64_e input 64bit data swap for other than stream and dpb data may be used for 128 bit environment 0 = no swapping of 64 bit words 1 = 64 bit data words are swapped
1	RW	0x0	sw_in_swap32_e input 32bit data swap for other than stream and dpb data may be used for 64 or 128 bit environment 0 = no swapping of 32 bit words 1 = 32 bit data words are swapped
0	RW	0x0	sw_in_endian decoder input endian mode for other than stream and dpb data 0 = little endian 1 = big endian for litter enadian , a data 0x12345678, 0x78 is stored in lower address, 0x12 is stored in higher address

**rkvdec\_swreg3\_picpar**

Address: Operational Base + offset (0x000c)  
picture parameters

Bit	Attr	Reset Value	Description
-----	------	-------------	-------------

Bit	Attr	Reset Value	Description
31:21	RW	0x000	<p>sw_slice_num_lowbits slice number in a frame hevc:slice number in a frame (0~199, when it is 0, it real means 1 slice in a frame) just only used for rps read. 2013.11.27 change the meaning from count from 1, so it will be in 1~200 2013.11.30 sw_slice_num max value is change to 600, so sw_slice_num expand to 10bit h264:slice number in a frame (0~4095, when it is 1, it real means 1 slice in a frame), for H264, it means sw_slice_num_lowbits vp9: no use</p>
20:12	RW	0x000	<p>sw_uv_hor_virstride uv horizontal virstride picture horizontal virtual stride (the unit is 128bit) the max is <math>(4096 \times 1.5 + 128) / 16 = 0x188</math> suggest this register to config to even for advance ddr performance</p>
11	RW	0x0	<p>sw_slice_num_highbit the highest bit of sw_slice_num the highest bit of sw_slice_num</p>
10:9	RO	0x0	reserved
8:0	RW	0x000	<p>sw_y_hor_virstride picture horizontal virtual stride picture horizontal virtual stride (the unit is 128bit) the max is <math>(4096 \times 1.5 + 128) / 16 = 0x188</math> suggest this register to config to even for advance ddr performance</p>

**rkvdec\_swreg4\_strm\_rlc\_base**

Address: Operational Base + offset (0x0010)

the stream or rlc data base address

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	<p>sw_strm_rlc_base the stream or rlc data base address when swreg2.sw_rlc_mode =1, it is base address for rlc data when swreg2.sw_rlc_mode =0, it is base address for stream , after a frame is decoded ready or error (stream error , time out , bus error) , it is the last address of the stream the address should 128bit align</p>
3:0	RO	0x0	reserved

**rkvdec\_swreg5\_stream\_rlc\_len**

Address: Operational Base + offset (0x0014)

amount of stream bytes or rlc data byte in the input buffer or the

Bit	Attr	Reset Value	Description
31:27	RO	0x0	reserved
26:0	RW	0x00000000	sw_stream_len amount of stream (unit is 8bit) in the input buffer amount of stream 8bits in the input buffer the max of sw_stream_len : $4096 \times 2304 \times 1.5 \times 1.5 = 0x1440000$ 128bits unit: $0x1440000 / 16 = 0x144000$ it is count from 0 2013.10.15 change to 23bit for zty's suggestion 2013.10.28, amount of stream data bytes in input buffer. it is count from 1, change to 27bits

**rkvdec\_swreg6\_cabactbl\_prob\_base**

Address: Operational Base + offset (0x0018)

the base address of cabac table

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_cabactbl_base the base address of cabac table the base address of cabac table the address should 128bit align
3:0	RO	0x0	reserved

**rkvdec\_swreg7\_decout\_base**

Address: Operational Base + offset (0x001c)

base address of decoder output picture base address

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_decout_base base address of decoder output picture addr base address of decoder output picture the address should be 128bit align in H264 decode format, the top field and bottom field are the same addr
3:0	RO	0x0	reserved

**rkvdec\_swreg8\_y\_virstride**

Address: Operational Base + offset (0x0020)

the ouput picture y virtual stride

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19:0	RW	0x00000	sw_y_virstride the output picture y virtual stride the output picture y virtual stride (the unit is 128bit) the max: $(4096 \times 1.5 + 128) \times 2304 = 0xdc8000$ we can know the sw_uvout_base = sw_decout_base + (sw_y_virstride <<4)

**rkvdec\_swreg9\_yuv\_virstride**

Address: Operational Base + offset (0x0024)

the ouput picture yuv virtual stride

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:21	RO	0x0	reserved
20:0	RW	0x0000000	sw_yuv_virstride the ouput picture yuv virtual stride the output picture yuv virtual stride (the unit is 128bit) the max : $(4096 \times 1.5 + 128) \times 2304 \times 1.5 = 0x14ac000$ we can know the sw_mvout_base = sw_decout_base + (sw_yuv_virstride <<4) for yuv422: $4096 \times 2304 \times 2 \times 1.25 = 0x1680000$

**rkvdec\_swreg10\_hevc\_refer0\_base**

Address: Operational Base + offset (0x0028)

base address for reference picture index 0

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:4	RW	0x00000000	sw_refer0_base base address for reference picture index0 base address for reference picture index 0 (the address should be 128bit align)
3:0	RW	0x0	sw_ref_valid_0~3 valid flag for picture index 0 ~3 valid flag for picture index 0 ~3

**rkvdec\_swreg10\_h264\_refer0\_base**

Address: Operational Base + offset (0x0028)

base address for reference picture index 0

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:4	RW	0x00000000	sw_refer0_base base address for reference picture index0 base address for reference picture index0 (the address should be 128bit align)
3	RW	0x0	sw_ref0_colmv_use_flag ref0 colmv use flag ref0 colmv use flag
2	RW	0x0	sw_ref0_botfield_used bottom field is used bottom field is used the same meaning with ref_valid
1	RW	0x0	sw_ref0_topfield_used top field is used top field is used the same meaning with ref_valid

Bit	Attr	Reset Value	Description
0	RW	0x0	sw_ref0_field reference 0 picture field flag 1'b0: frame 1'b1: field

**rkvdec\_swreg10\_vp9\_cprheader\_offset**

Address: Operational Base + offset (0x0028)

vp9 compressed header offset

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:0	RW	0x0000	sw_vp9_cprheader_offset vp9 compressed header offset vp9 compressed header offset, at most 2000 probs, 10bit per prob, 20000 bit at most now is for no use, because it can read from the last syntax of the uncompressed header

**rkvdec\_swreg11\_hevc\_refer1\_base**

Address: Operational Base + offset (0x002c)

base address for reference picture index 1

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer1_base base address for reference picture index 1 base address for reference picture index 1 (the address should be 128bit align)
3:0	RW	0x0	sw_ref_valid_4_7 valid flag for picture index 4 ~7 valid flag for picture index 4 ~7

**rkvdec\_swreg11\_h264\_refer1\_base**

Address: Operational Base + offset (0x002c)

base address for reference picture index 1

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer1_base base address for reference picture index1 base address for reference picture index1 (the address should be 128bit align)
3	RW	0x0	sw_ref1_colmv_use_flag sw_ref1_colmv_use_flag sw_ref1_colmv_use_flag
2	RW	0x0	sw_ref1_botfield_used ref1 bottom field is used ref1 bottom field is used

Bit	Attr	Reset Value	Description
1	RW	0x0	sw_ref1_topfield_used ref1 topfield is used ref1 topfield is used
0	RW	0x0	sw_ref1_field reference 1 picture field flag 1'b0: frame 1'b1: field

**rkvdec\_swreg11\_vp9\_referlast\_base**

Address: Operational Base + offset (0x002c)

base address for last frame

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_vp9last_base base address for last frame base address for last (the address should be 128bit align)
3:0	RO	0x0	reserved

**rkvdec\_swreg12\_hevc\_refer2\_base**

Address: Operational Base + offset (0x0030)

base address for reference picture index 2

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer2_base base address for reference picture index 2 base address for reference picture index 2 (the address should be 128bit align)
3:0	RW	0x0	sw_ref_valid_8_11 valid flag for picture index 8~11 valid flag for picture index 8~11

**rkvdec\_swreg12\_h264\_refer2\_base**

Address: Operational Base + offset (0x0030)

base address for reference picture index 2

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer2_base base address for reference picture index2 base address for reference picture index2 (the address should be 128bit align)
3	RW	0x0	sw_ref2_colmv_use_flag sw_ref2_colmv_use_flag sw_ref2_colmv_use_flag
2	RW	0x0	sw_ref2_botfield_used ref2 bottom field is used ref2 bottom field is used

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
1	RW	0x0	sw_ref2_topfield_used ref2 topfield is used ref2 topfield is used
0	RW	0x0	sw_ref2_field reference 2 picture field flag 1'b0: frame 1'b1: field

**rkvdec\_swreg12\_vp9\_refergolden\_base**

Address: Operational Base + offset (0x0030)

base address for golden frame

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:4	RW	0x00000000	sw_vp9golden_base base address for golden frame base address for golden (the address should be 128bit align)
3:0	RO	0x0	reserved

**rkvdec\_swreg13\_hevc\_refer3\_base**

Address: Operational Base + offset (0x0034)

base address for reference picture index 3

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:4	RW	0x00000000	sw_refer3_base base address for reference picture index 3 base address for reference picture index 3 (the address should be 128bit align)
3	RO	0x0	reserved
2:0	RW	0x0	sw_ref_valid_12_14 valid flag for picture index 12~14 valid flag for picture index 12~14

**rkvdec\_swreg13\_h264\_refer3\_base**

Address: Operational Base + offset (0x0034)

base address for reference picture index 3

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:4	RW	0x00000000	sw_refer3_base base address for reference picture index3 base address for reference picture index3 (the address should be 128bit align)
3	RW	0x0	sw_ref3_colmv_use_flag sw_ref3_colmv_use_flag sw_ref3_colmv_use_flag
2	RW	0x0	sw_ref3_botfield_used ref3 bottom field is used ref3 bottom field is used

Bit	Attr	Reset Value	Description
1	RW	0x0	sw_ref3_topfield_used ref3 topfield is used ref3 topfield is used
0	RW	0x0	sw_ref3_field reference 3 picture field flag 1'b0: frame 1'b1: field

**rkvdec\_swreg13\_vp9\_referalfter\_base**

Address: Operational Base + offset (0x0034)

base address for referalfter frame

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_vp9alfter_base base address for alfter frame base address for alfter (the address should be 128bit align)
3:0	RO	0x0	reserved

**rkvdec\_swreg14\_hevc\_refer4\_base**

Address: Operational Base + offset (0x0038)

base address for reference picture index 4

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer4_base base address for reference picture index 4 base address for reference picture index 4(the address should be 128bit align)
3:0	RO	0x0	reserved

**rkvdec\_swreg14\_h264\_refer4\_base**

Address: Operational Base + offset (0x0038)

base address for reference picture index 4

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer4_base base address for reference picture index4 base address for reference picture index4 (the address should be 128bit align)
3	RW	0x0	sw_ref4_colmv_use_flag sw_ref4_colmv_use_flag sw_ref4_colmv_use_flag
2	RW	0x0	sw_ref4_botfield_used ref4 bottom field is used ref4 bottom field is used
1	RW	0x0	sw_ref4_topfield_used ref4 topfield is used ref4 topfield is used

Bit	Attr	Reset Value	Description
0	RW	0x0	sw_ref4_field reference 4 picture field flag 1'b0: frame 1'b1: field

**rkvdec\_swreg14\_vp9count\_base**

Address: Operational Base + offset (0x0038)

vp9 count base addr

Bit	Attr	Reset Value	Description
31:3	RW	0x00000000	sw_vp9count_base vp9 count base addr vp9 count base addr
2:0	RO	0x0	reserved

**rkvdec\_swreg15\_h264\_refer5\_base**

Address: Operational Base + offset (0x003c)

base address for reference picture index 5

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer5_base base address for reference picture index5 base address for reference picture index5 (the address should be 128bit align)
3	RW	0x0	sw_ref5_colmv_use_flag sw_ref5_colmv_use_flag sw_ref5_colmv_use_flag
2	RW	0x0	sw_ref5_botfield_used ref5 bottom field is used ref5 bottom field is used
1	RW	0x0	sw_ref5_topfield_used ref5 topfield is used ref5 topfield is used
0	RW	0x0	sw_ref5_field reference 5 picture field flag 1'b0: frame 1'b1: field

**rkvdec\_swreg15\_vp9\_segidlast\_base**

Address: Operational Base + offset (0x003c)

base address for last frame segment id

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_vp9segidlast_base base address for vp9 last frame segment id base address for vp9 last frame segment id (the address should be 128bit align)
3:0	RO	0x0	reserved

**rkvdec\_swreg15\_hevc\_refer5\_base**

Address: Operational Base + offset (0x003c)

base address for reference picture index 5

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:4	RW	0x00000000	sw_refer5_base base address for reference picture index 5 base address for reference picture index 5(the address should be 128bit align)
3:0	RO	0x0	reserved

**rkvdec\_swreg16\_h264\_refer6\_base**

Address: Operational Base + offset (0x0040)

base address for reference picture index 6

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:4	RW	0x00000000	sw_refer6_base base address for reference picture index6 base address for reference picture index6 (the address should be 128bit align)
3	RW	0x0	sw_ref6_colmv_use_flag sw_ref6_colmv_use_flag sw_ref6_colmv_use_flag
2	RW	0x0	sw_ref6_botfield_used ref6 botfield is used ref6 botfield is used
1	RW	0x0	sw_ref6_topfield_used ref6 topfield is used ref6 topfield is used
0	RW	0x0	sw_ref6_field reference 6 picture field flag 1'b0: frame 1'b1: field

**rkvdec\_swreg16\_vp9\_segidcur\_base**

Address: Operational Base + offset (0x0040)

base address for cur frame segment id

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:4	RW	0x00000000	sw_vp9segidcur_base base address for vp9 cur frame segment id base address for vp9 cur frame segment id (the address should be 128bit align)
3:0	RO	0x0	reserved

**rkvdec\_swreg16\_hevc\_refer6\_base**

Address: Operational Base + offset (0x0040)

base address for reference picture index 6

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer6_base base address for reference picture index 6 base address for reference picture index 6(the address should be 128bit align)
3:0	RO	0x0	reserved

**rkvdec\_swreg17\_h264\_refer7\_base**

Address: Operational Base + offset (0x0044)

base address for reference picture index 7

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer7_base base address for reference picture index7 base address for reference picture index7 (the address should be 128bit align)
3	RW	0x0	sw_ref7_colmv_use_flag sw_ref7_colmv_use_flag sw_ref7_colmv_use_flag
2	RW	0x0	sw_ref7_botfield_used ref7 bottom field is used ref7 bottom field is used
1	RW	0x0	sw_ref7_topfield_used ref7 topfield is used ref7 topfield is used
0	RW	0x0	sw_ref7_field reference 7 picture field flag 1'b0: frame 1'b1: field

**rkvdec\_swreg17\_vp9\_frame\_size\_last**

Address: Operational Base + offset (0x0044)

vp9 last frame size

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	sw_frameheight_last last frame frame_size_height last frame frame_size_height
15:0	RW	0x0000	sw_frameworkwidth_last last frame frame_size_width last frame frame_size_width

**rkvdec\_swreg17\_hevc\_refer7\_base**

Address: Operational Base + offset (0x0044)

base address for reference picture index 7

Bit	Attr	Reset Value	Description
-----	------	-------------	-------------

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer7_base base address for reference picture index 7 base address for reference picture index 7(the address should be 128bit align)
3:0	RO	0x0	reserved

**rkvdec\_swreg18\_h264\_refer8\_base**

Address: Operational Base + offset (0x0048)

base address for reference picture index 8

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer8_base base address for reference picture index8 base address for reference picture index8 (the address should be 128bit align)
3	RW	0x0	sw_ref8_colmv_use_flag sw_ref8_colmv_use_flag sw_ref8_colmv_use_flag
2	RW	0x0	sw_ref8_botfield_used ref8 bottom field is used ref8 bottom field is used
1	RW	0x0	sw_ref8_topfield_used ref8 topfield is used ref8 topfield is used
0	RW	0x0	sw_ref8_field reference 8 picture field flag 1'b0: frame 1'b1: field

**rkvdec\_swreg18\_vp9\_frame\_size\_golden**

Address: Operational Base + offset (0x0048)

vp9 golden frame size

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	sw_frameheight_golden golden frame_size_height golden frame_size_height
15:0	RW	0x0000	sw_frameworkwidth_golden golden frame_size_width golden frame_size_width

**rkvdec\_swreg18\_hevc\_refer8\_base**

Address: Operational Base + offset (0x0048)

base address for reference picture index 8

Bit	Attr	Reset Value	Description
-----	------	-------------	-------------

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer8_base base address for reference picture index 8 base address for reference picture index 8(the address should be 128bit align)
3:0	RO	0x0	reserved

**rkvdec\_swreg19\_h264\_refer9\_base**

Address: Operational Base + offset (0x004c)

base address for reference picture index 9

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer9_base base address for reference picture index9 base address for reference picture index9 (the address should be 128bit align)
3	RW	0x0	sw_ref9_colmv_use_flag sw_ref9_colmv_use_flag sw_ref9_colmv_use_flag
2	RW	0x0	sw_ref9_botfield_used ref9 bottom field is used ref9 bottom field is used
1	RW	0x0	sw_ref9_topfield_used ref9 topfield is used ref9 topfield is used
0	RW	0x0	sw_ref9_field reference 9 picture field flag 1'b0: frame 1'b1: field

**rkvdec\_swreg19\_vp9\_frame\_size\_altref**

Address: Operational Base + offset (0x004c)

vp9 alfter frame size

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	sw_frameheight_alfter alfter frame_size_height alfter frame_size_height
15:0	RW	0x0000	sw_frameworkwidth_alfter alfter frame_size_width alfter frame_size_width

**rkvdec\_swreg19\_hevc\_refer9\_base**

Address: Operational Base + offset (0x004c)

base address for reference picture index 9

Bit	Attr	Reset Value	Description
-----	------	-------------	-------------

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer9_base base address for reference picture index 9 base address for reference picture index 9(the address should be 128bit align)
3:0	RO	0x0	reserved

**rkvdec\_swreg20\_h264\_refer10\_base**

Address: Operational Base + offset (0x0050)

base address for reference picture index 10

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer10_base base address for reference picture index10 base address for reference picture index10 (the address should be 128bit align)
3	RW	0x0	sw_ref10_colmv_use_flag sw_ref10_colmv_use_flag sw_ref10_colmv_use_flag
2	RW	0x0	sw_ref10_botfield_used ref10 bottom field is used ref10 bottom field is used
1	RW	0x0	sw_ref10_topfield_used ref10 topfield is used ref10 topfield is used
0	RW	0x0	sw_ref10_field reference 10 picture field flag 1'b0: frame 1'b1: field

**rkvdec\_swreg20\_vp9\_segid\_grp0**

Address: Operational Base + offset (0x0050)

vp9 segid syntax grp0

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22	RW	0x0	sw_vp9segid0_frame_skip_en frame skip feature enable frame skip feature enable
21:20	RW	0x0	sw_vp9segid0_referinfo specifies segment i's reference_info which is used to get ref_frame[0] specifies segment i's reference_info which is used to get ref_frame[0]
19	RW	0x0	sw_vp9segid0_referinfo_en frame reference info enable frame reference info enable

Bit	Attr	Reset Value	Description
18:12	RW	0x00	sw_vp9segid0_frame_loopfilter_value frame loopfilter value specifies segment i's loopfilter_delta value which is used to calculate filter level
11	RW	0x0	sw_vp9segid0_frame_loopfilter_value_en frame_loopfilter_value feature enable frame_loopfilter_value feature enable
10:2	RW	0x000	sw_vp9segid0_frame_qp_delta frame qp delta specifies segment i's qp_delta value which is used to calculate y_dequant and uv_dequant
1	RW	0x0	sw_vp9segid0_frame_qp_delta_en frame_qp_delta feature enable frame_qp_delta feature enable
0	RW	0x0	sw_vp9segid_abs_delta abs delta used to decide quant and loopfilter param

**rkvdec\_swreg20\_hevc\_refer10\_base**

Address: Operational Base + offset (0x0050)

base address for reference picture index 10

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer10_base base address for reference picture index 10 base address for reference picture index 10(the address should be 128bit align)
3:0	RO	0x0	reserved

**rkvdec\_swreg21\_h264\_refer11\_base**

Address: Operational Base + offset (0x0054)

base address for reference picture index 11

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer11_base base address for reference picture index11 base address for reference picture index11 (the address should be 128bit align)
3	RW	0x0	sw_ref11_colmv_use_flag sw_ref11_colmv_use_flag sw_ref11_colmv_use_flag
2	RW	0x0	sw_ref11_botfield_used ref11 bottom field is used ref11 bottom field is used
1	RW	0x0	sw_ref11_topfield_used ref11 topfield is used ref11 topfield is used

Bit	Attr	Reset Value	Description
0	RW	0x0	sw_ref11_field reference 11 picture field flag 1'b0: frame 1'b1: field

**rkvdec\_swreg21\_vp9\_segid\_grp1**

Address: Operational Base + offset (0x0054)

vp9 segid syntax grp1

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22	RW	0x0	sw_vp9segid1_frame_skip_en frame skip feature enable frame skip feature enable
21:20	RW	0x0	sw_vp9segid1_referinfo specifies segment i's reference_info which is used to get ref_frame[0] specifies segment i's reference_info which is used to get ref_frame[0]
19	RW	0x0	sw_vp9segid1_referinfo_en frame reference info enable frame reference info enable
18:12	RW	0x00	sw_vp9segid1_frame_loopfilter_value frame loopfilter value specifies segment i's loopfilter_delta value which is used to calculate filter level
11	RW	0x0	sw_vp9segid1_frame_loopfilter_value_en frame_loopfilter_value feature enable frame_loopfilter_value feature enable
10:2	RW	0x000	sw_vp9segid1_frame_qp_delta frame qp delta specifies segment i's qp_delta value which is used to calculate y_dequant and uv_dequant
1	RW	0x0	sw_vp9segid1_frame_qp_delta_en frame_qp_delta feature enable frame_qp_delta feature enable
0	RO	0x0	reserved

**rkvdec\_swreg21\_hevc\_refer11\_base**

Address: Operational Base + offset (0x0054)

base address for reference picture index 11

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer11_base base address for reference picture index 11 base address for reference picture index 11(the address should be 128bit align)

Bit	Attr	Reset Value	Description
3:0	RO	0x0	reserved

**rkvdec\_swreg22\_h264\_refer12\_base**

Address: Operational Base + offset (0x0058)

base address for reference picture index 12

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer12_base base address for reference picture index12 base address for reference picture index12 (the address should be 128bit align)
3	RW	0x0	sw_ref12_colmv_use_flag sw_ref12_colmv_use_flag sw_ref12_colmv_use_flag
2	RW	0x0	sw_ref12_botfield_used ref12 bottom field is used ref12 bottom field is used
1	RW	0x0	sw_ref12_topfield_used ref12 topfield is used ref12 topfield is used
0	RW	0x0	sw_ref12_field reference 12 picture field flag 1'b0: frame 1'b1: field

**rkvdec\_swreg22\_vp9\_segid\_grp2**

Address: Operational Base + offset (0x0058)

vp9 segid syntax grp2

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22	RW	0x0	sw_vp9segid2_frame_skip_en frame skip feature enable frame skip feature enable
21:20	RW	0x0	sw_vp9segid2_referinfo specifies segment i's reference_info which is used to get ref_frame[0] specifies segment i's reference_info which is used to get ref_frame[0]
19	RW	0x0	sw_vp9segid2_referinfo_en frame reference info enable frame reference info enable
18:12	RW	0x00	sw_vp9segid2_frame_loopfilter_value frame loopfilter value specifies segment i's loopfilter_delta value which is used to calculate filter level

Bit	Attr	Reset Value	Description
11	RW	0x0	sw_vp9segid2_frame_looptilter_value_en frame_looptilter_value feature enable frame_looptilter_value feature enable
10:2	RW	0x000	sw_vp9segid2_frame_qp_delta frame qp delta specifies segment i's qp_delta value which is used to calculate y_dequant and uv_dequant
1	RW	0x0	sw_vp9segid2_frame_qp_delta_en frame_qp_delta feature enable frame_qp_delta feature enable
0	RO	0x0	reserved

**rkvdec\_swreg22\_hevc\_refer12\_base**

Address: Operational Base + offset (0x0058)

base address for reference picture index 12

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer12_base base address for reference picture index 12 base address for reference picture index 12(the address should be 128bit align)
3:0	RO	0x0	reserved

**rkvdec\_swreg23\_h264\_refer13\_base**

Address: Operational Base + offset (0x005c)

base address for reference picture index 13

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_refer13_base base address for reference picture index13 base address for reference picture index13 (the address should be 128bit align)
3	RW	0x0	sw_ref13_colmv_use_flag sw_ref13_colmv_use_flag sw_ref13_colmv_use_flag
2	RW	0x0	sw_ref13_botfield_used ref13 bottom field is used ref13 bottom field is used
1	RW	0x0	sw_ref13_topfield_used ref13 topfield is used ref13 topfield is used
0	RW	0x0	sw_ref13_field reference 13 picture field flag 1'b0: frame 1'b1: field

**rkvdec\_swreg23\_vp9\_segid\_grp3**

Address: Operational Base + offset (0x005c)

vp9 segid syntax grp3

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:23	RO	0x0	reserved
22	RW	0x0	sw_vp9segid3_frame_skip_en frame skip feature enable frame skip feature enable
21:20	RW	0x0	sw_vp9segid3_referinfo specifies segment i's reference_info which is used to get ref_frame[0] specifies segment i's reference_info which is used to get ref_frame[0]
19	RW	0x0	sw_vp9segid3_referinfo_en frame reference info enable frame reference info enable
18:12	RW	0x00	sw_vp9segid3_frame_loopfilter_value frame loopfilter value specifies segment i's loopfilter_delta value which is used to calculate filter level
11	RW	0x0	sw_vp9segid3_frame_loopfilter_value_en frame_loopfilter_value feature enable frame_loopfilter_value feature enable
10:2	RW	0x000	sw_vp9segid3_frame_qp_delta frame qp delta specifies segment i's qp_delta value which is used to calculate y_dequant and uv_dequant
1	RW	0x0	sw_vp9segid3_frame_qp_delta_en frame_qp_delta feature enable frame_qp_delta feature enable
0	RO	0x0	reserved

**rkvdec\_swreg23\_hevc\_refer13\_base**

Address: Operational Base + offset (0x005c)

base address for reference picture index 13

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:4	RW	0x0000000	sw_refer13_base base address for reference picture index 13 base address for reference picture index 13(the address should be 128bit align)
3:0	RO	0x0	reserved

**rkvdec\_swreg24\_h264\_refer14\_base**

Address: Operational Base + offset (0x0060)

base address for reference picture index 14

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:4	RW	0x00000000	sw_refer14_base base address for reference picture index14 base address for reference picture index14 (the address should be 128bit align)
3	RW	0x0	sw_ref14_colmv_use_flag sw_ref14_colmv_use_flag sw_ref14_colmv_use_flag
2	RW	0x0	sw_ref14_botfield_used ref14 bottom field is used ref14 bottom field is used
1	RW	0x0	sw_ref14_topfield_used ref14 topfield is used ref14 topfield is used
0	RW	0x0	sw_ref14_field reference 14 picture field flag 1'b0: frame 1'b1: field

**rkvdec\_swreg24\_vp9\_segid\_grp4**

Address: Operational Base + offset (0x0060)

vp9 segid syntax grp4

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:23	RO	0x0	reserved
22	RW	0x0	sw_vp9segid4_frame_skip_en frame skip feature enable frame skip feature enable
21:20	RW	0x0	sw_vp9segid4_referinfo specifies segment i's reference_info which is used to get ref_frame[0] specifies segment i's reference_info which is used to get ref_frame[0]
19	RW	0x0	sw_vp9segid4_referinfo_en frame reference info enable frame reference info enable
18:12	RW	0x00	sw_vp9segid4_frame_loopfilter_value frame loopfilter value specifies segment i's loopfilter_delta value which is used to calculate filter level
11	RW	0x0	sw_vp9segid4_frame_loopfilter_value_en frame_loopfilter_value feature enable frame_loopfilter_value feature enable
10:2	RW	0x000	sw_vp9segid4_frame_qp_delta frame qp delta specifies segment i's qp_delta value which is used to calculate y_dequant and uv_dequant

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
1	RW	0x0	sw_vp9segid4_frame_qp_delta_en frame_qp_delta feature enable frame_qp_delta feature enable
0	RO	0x0	reserved

**rkvdec\_swreg24\_hevc\_refer14\_base**

Address: Operational Base + offset (0x0060)

base address for reference picture index 14

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:4	RW	0x00000000	sw_refer14_base base address for reference picture index 14 base address for reference picture index 14(the address should be 128bit align)
3:0	RO	0x0	reserved

**rkvdec\_swreg25\_vp9\_segid\_grp5**

Address: Operational Base + offset (0x0064)

vp9 segid syntax grp5

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:23	RO	0x0	reserved
22	RW	0x0	sw_vp9segid5_frame_skip_en frame skip feature enable frame skip feature enable
21:20	RW	0x0	sw_vp9segid5_referinfo specifies segment i's reference_info which is used to get ref_frame[0] specifies segment i's reference_info which is used to get ref_frame[0]
19	RW	0x0	sw_vp9segid5_referinfo_en frame reference info enable frame reference info enable
18:12	RW	0x00	sw_vp9segid5_frame_loopfilter_value frame loopfilter value specifies segment i's loopfilter_delta value which is used to calculate filter level
11	RW	0x0	sw_vp9segid5_frame_loopfilter_value_en frame_loopfilter_value feature enable frame_loopfilter_value feature enable
10:2	RW	0x000	sw_vp9segid5_frame_qp_delta frame qp delta specifies segment i's qp_delta value which is used to calculate y_dequant and uv_dequant
1	RW	0x0	sw_vp9segid5_frame_qp_delta_en frame_qp_delta feature enable frame_qp_delta feature enable

Bit	Attr	Reset Value	Description
0	RO	0x0	reserved

**rkvdec\_swreg25\_refer0\_poc**

Address: Operational Base + offset (0x0064)

the poc of reference picture index 0

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer0_poc the poc of reference picture index 0 the poc of reference picture index 0

**rkvdec\_swreg26\_vp9\_segid\_grp6**

Address: Operational Base + offset (0x0068)

vp9 segid syntax grp6

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22	RW	0x0	sw_vp9segid6_frame_skip_en frame skip feature enable frame skip feature enable
21:20	RW	0x0	sw_vp9segid6_referinfo specifies segment i's reference_info which is used to get ref_frame[0] specifies segment i's reference_info which is used to get ref_frame[0]
19	RW	0x0	sw_vp9segid6_referinfo_en frame reference info enable frame reference info enable
18:12	RW	0x00	sw_vp9segid6_frame_loopfilter_value frame loopfilter value specifies segment i's loopfilter_delta value which is used to calculate filter level
11	RW	0x0	sw_vp9segid6_frame_loopfilter_value_en frame_loopfilter_value feature enable frame_loopfilter_value feature enable
10:2	RW	0x000	sw_vp9segid6_frame_qp_delta frame qp delta specifies segment i's qp_delta value which is used to calculate y_dequant and uv_dequant
1	RW	0x0	sw_vp9segid6_frame_qp_delta_en frame_qp_delta feature enable frame_qp_delta feature enable
0	RO	0x0	reserved

**rkvdec\_swreg26\_refer1\_poc**

Address: Operational Base + offset (0x0068)

the poc of reference picture index 1

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer1_poc the poc of reference picture index 1 the poc of reference picture index 1

**rkvdec\_swreg27\_vp9\_segid\_grp7**

Address: Operational Base + offset (0x006c)

vp9 segid syntax grp7

Bit	Attr	Reset Value	Description
31:23	RO	0x0	reserved
22	RW	0x0	sw_vp9segid7_frame_skip_en frame skip feature enable frame skip feature enable
21:20	RW	0x0	sw_vp9segid7_referinfo specifies segment i's reference_info which is used to get ref_frame[0] specifies segment i's reference_info which is used to get ref_frame[0]
19	RW	0x0	sw_vp9segid7_referinfo_en frame reference info enable frame reference info enable
18:12	RW	0x00	sw_vp9segid7_frame_loopfilter_value frame loopfilter value specifies segment i's loopfilter_delta value which is used to calculate filter level
11	RW	0x0	sw_vp9segid7_frame_loopfilter_value_en frame_loopfilter_value feature enable frame_loopfilter_value feature enable
10:2	RW	0x000	sw_vp9segid7_frame_qp_delta frame qp delta specifies segment i's qp_delta value which is used to calculate y_dequant and uv_dequant
1	RW	0x0	sw_vp9segid7_frame_qp_delta_en frame_qp_delta feature enable frame_qp_delta feature enable
0	RO	0x0	reserved

**rkvdec\_swreg27\_refer2\_poc**

Address: Operational Base + offset (0x006c)

the poc of reference picture index 2

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer2_poc the poc of reference picture index 2 the poc of reference picture index 2

**rkvdec\_swreg28\_vp9\_cprheader\_config**

Address: Operational Base + offset (0x0070)

vp9 compressed header config info

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:5	RO	0x0	reserved
4:3	RW	0x0	<p>sw_vp9_frame_reference_mode frame_reference_mode frame_reference_mode specifies frame reference mode. SINGLE_REFERENCE = 0, COMPOUND_REFERENCE = 1, REFERENCE_MODE_SELECT = 2, REFERENCE_MODES = 3,</p> <p>When frame_reference_mode_flag0 is not present ,it equal to 0 by default.</p> <p>When frame_reference_mode_flag1 is not present ,it equal to 0 by default.</p> <p>frame_reference_mode = frame_reference_mode_flag0 == 0 ? frame_reference_mode_flag1 == 0 ? REFERENCE_MODE_SELECT : COMPOUND_REFERENCE) : SINGLE_REFERENCE</p>
2:0	RW	0x0	<p>sw_vp9_tx_mode tx_mode tx_mode specifies frame transform mode. ONLY_4X4 = 0, // only 4x4 transform used ALLOW_8X8 = 1, // allow block transform size up to 8x8 ALLOW_16X16 = 2, // allow block transform size up to 16x16 ALLOW_32X32 = 3, // allow block transform size up to 32x32 TX_MODE_SELECT = 4, // transform specified for each block</p>

**rkvdec\_swreg28\_refer3\_poc**

Address: Operational Base + offset (0x0070)

the poc of reference picture index 3

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer3_poc the poc of reference picture index 3 the poc of reference picture index 3

**rkvdec\_swreg29\_vp9\_lref\_scale**

Address: Operational Base + offset (0x0074)

scaling factor for last reference picture

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	sw_vp9_lref_ver_scale vertical scaling factor for last reference picture vertical scaling factor for last reference picture
15:0	RW	0x0000	sw_vp9_lref_hor_scale horizontal scaling factor for last reference picture horizontal scaling factor for last reference picture $sw_{vp9\_lref\_hor\_scale} = (last\_ref\_width / cur\_width) * 0x4000$

**rkvdec\_swreg29\_refer4\_poc**

Address: Operational Base + offset (0x0074)

the poc of reference picture index 4

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer4_poc the poc of reference picture index 4 the poc of reference picture index 4

**rkvdec\_swreg30\_vp9\_gref\_scale**

Address: Operational Base + offset (0x0078)

scaling factor for golden reference picture

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	sw_vp9_gref_ver_scale vertical scaling factor for golden reference picture vertical scaling factor for golden reference picture
15:0	RW	0x0000	sw_vp9_gref_hor_scale horizontal scaling factor for golden reference picture horizontal scaling factor for golden reference picture $sw_{vp9\_gref\_hor\_scale} = (golden\_ref\_width / cur\_width) * 0x4000$

**rkvdec\_swreg30\_refer5\_poc**

Address: Operational Base + offset (0x0078)

the poc of reference picture index 5

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer5_poc the poc of reference picture index 5 the poc of reference picture index 5

**rkvdec\_swreg31\_vp9\_aref\_scale**

Address: Operational Base + offset (0x007c)

scaling factor for alfter reference picture

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	sw_vp9_aref_ver_scale vertical scaling factor for alfter reference picture vertical scaling factor for alfter reference picture

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
15:0	RW	0x0000	sw_vp9_aref_hor_scale horizontal scaling factor for alfter reference picture horizontal scaling factor for alfter reference picture $sw\_vp9\_gref\_hor\_scale = (alfter\_ref\_width / cur\_width) * 0x4000$

**rkvdec\_swreg31\_refer6\_poc**

Address: Operational Base + offset (0x007c)  
 the poc of reference picture index 6

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer6_poc the poc of reference picture index 6 the poc of reference picture index 6

**rkvdec\_swreg32\_vp9\_ref\_deltas\_lastframe**

Address: Operational Base + offset (0x0080)  
 vp9 ref deltas

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:28	RO	0x0	reserved
27:0	RW	0x00000000	sw_vp9_ref_deltas_lastframe vp9 ref deltas vp9 ref deltas of lastframe, for cal loopfilter filter type use

**rkvdec\_swreg32\_refer7\_poc**

Address: Operational Base + offset (0x0080)  
 the poc of reference picture index 7

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer7_poc the poc of reference picture index 7 the poc of reference picture index 7

**rkvdec\_swreg33\_vp9\_info\_lastframe**

Address: Operational Base + offset (0x0084)  
 vp9 info for lastframe

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:23	RO	0x0	reserved
22:20	RW	0x0	sw_vp9_color_space_lastkeyframe vp9 last keyframe color_space vp9 last keyframe color_space
19	RW	0x0	sw_vp9_last_widhheight_eqcur last width and height equal cur last width and height equal cur frame

Bit	Attr	Reset Value	Description
18	RW	0x0	sw_vp9_last_intra_only last frame intra only flag vp9 last frame intra only flag to give inter command use it is for last_dec_frame
17	RW	0x0	sw_vp9_last_show_frame last show frame flag for cal the flag use_prev_in_find_mv_refs which is to inter cmd it is for last_dec_frame
16	RW	0x0	sw_segmentation_enable_lstdframe segmentation_enable for last frame 1'b1:sw_segmentation_enable for last frame it is for last_dec_frame
15:14	RO	0x0	reserved
13:0	RW	0x0000	sw_vp9_mode_deltas_lastframe vp9 mode deltas vp9 mode deltas it is for last dec frame

**rkvdec\_swreg33\_refer8\_poc**

Address: Operational Base + offset (0x0084)

the poc of reference picture index 8

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer8_poc the poc of reference picture index 8 the poc of reference picture index 8

**rkvdec\_swreg34\_vp9\_intercmd\_base**

Address: Operational Base + offset (0x0088)

inter cmd base addr

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_vp9_intercmd_base vp9 intercmd base addr vp9 inter command base addr, when sw_rlc_mode is 1'b1; when sw_dec_mode is VP9 and sw_rlc_mode is 1'b1, when read this register, after a frame is decoded ready or error (stream error , time out , bus error) , it is the end address of the intercmd
3:0	RO	0x0	reserved

**rkvdec\_swreg34\_refer9\_poc**

Address: Operational Base + offset (0x0088)

the poc of reference picture index 9

Bit	Attr	Reset Value	Description
-----	------	-------------	-------------

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer9_poc the poc of reference picture index 9 the poc of reference picture index 9

**rkvdec\_swreg35\_vp9\_intercmd\_num**

Address: Operational Base + offset (0x008c)

vp9 intercmd num

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x0000000	sw_vp9_intercmd_num sw_vp9_intercmd_num when rlc_mode is 1'b1, for sw_vp9_intercmd_num it's unit is 128bit it count from 1

**rkvdec\_swreg35\_refer10\_poc**

Address: Operational Base + offset (0x008c)

the poc of reference picture index 10

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer10_poc the poc of reference picture index 10 the poc of reference picture index 10

**rkvdec\_swreg36\_vp9\_lasttile\_size**

Address: Operational Base + offset (0x0090)

vp9 lasttile size

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23:0	RW	0x0000000	sw_vp9_lasttile_size vp9 last tile size vp9 last tile size ofr cur frame its unit is byte The real meaning the current frame size

**rkvdec\_swreg36\_refer11\_poc**

Address: Operational Base + offset (0x0090)

the poc of reference picture index 11

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer11_poc the poc of reference picture index 11 the poc of reference picture index 11

**rkvdec\_swreg37\_vp9\_lastf\_hor\_virstride**

Address: Operational Base + offset (0x0094)

Register0000 Abstract

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:25	RO	0x0	reserved
24:16	RW	0x000	sw_vp9_lastfuv_hor_virstride sw_vp9_lastfuv_hor_virstride vp9 last frame uv horizontal virstride
15:9	RO	0x0	reserved
8:0	RW	0x000	sw_vp9_lastfy_hor_virstride vp9 last frame y horizontal virstride vp9 last frame y horizontal virstride

**rkvdec\_swreg37\_refer12\_poc**

Address: Operational Base + offset (0x0094)

the poc of reference picture index 12

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer12_poc the poc of reference picture index 12 the poc of reference picture index 12

**rkvdec\_swreg38\_vp9\_goldenf\_hor\_virstride**

Address: Operational Base + offset (0x0098)

vp9 golden frame horizontal virstride

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:25	RO	0x0	reserved
24:16	RW	0x000	sw_vp9_goldenuv_hor_virstride vp9 goldenuv horizontal virstride vp9 golden_uv horizontal virstirde
15:9	RO	0x0	reserved
8:0	RW	0x000	sw_vp9_goldenfy_hor_virstride sw_vp9_goldenfy_hor_virstirde vp9 golden frame y horizontal virstride

**rkvdec\_swreg38\_refer13\_poc**

Address: Operational Base + offset (0x0098)

the poc of reference picture index 13

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer13_poc the poc of reference picture index 13 the poc of reference picture index 13

**rkvdec\_swreg39\_vp9\_altreff\_hor\_virstride**

Address: Operational Base + offset (0x009c)

vp9 altref frame horizontal virstride

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:25	RO	0x0	reserved

Bit	Attr	Reset Value	Description
24:16	RW	0x000	sw_vp9_altreffuv_hor_virstride vp9 altreffuv horizontal virstride vp9 altreff uv horizontal virstirde
15:9	RO	0x0	reserved
8:0	RW	0x000	sw_vp9_altreffy_hor_virstride sw_vp9_altreffy_hor_virstirde vp9 altref frame y horizontal virstride

**rkvdec\_swreg39\_refer14\_poc**

Address: Operational Base + offset (0x009c)

the poc of reference picture index 14

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer14_poc the poc of reference picture index 14 the poc of reference picture index 14

**rkvdec\_swreg40\_cur\_poc**

Address: Operational Base + offset (0x00a0)

the poc of cur picture

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_cur_poc the poc of the cur picture the poc of the cur picture for H264, it may be cur frame poc or cur top field poc

**rkvdec\_swreg41\_rlcwrite\_base**

Address: Operational Base + offset (0x00a4)

the base address or rlcwrite base addr

Bit	Attr	Reset Value	Description
31:3	RW	0x00000000	sw_rlcwrite_base the base address of rlcwrite the base address of rlcwrite(the address should 64bit align) cabac output write to this rlcwrite base address when sw_rlc_mode_direct_write in swreg2_sysctrl is valid
2:0	RO	0x0	reserved

**rkvdec\_swreg42\_pps\_base**

Address: Operational Base + offset (0x00a8)

the base address of pps

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_pps_base the base address of pps the base address of pps ( the address should 128bit align ) it is for storing sps(sequence parameter set) and pps(picture parameter set)

Bit	Attr	Reset Value	Description
3:0	RO	0x0	reserved

**rkvdec\_swreg43\_rps\_base**

Address: Operational Base + offset (0x00ac)

the base address of rps

Bit	Attr	Reset Value	Description
31:4	RW	0x00000000	sw_rps_base rps base address rps(reference picture set) base address (the address should 128bit align)
3:0	RO	0x0	reserved

**rkvdec\_swreg44\_strmd\_error\_en**

Address: Operational Base + offset (0x00b0)

cabac error enable config

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_strmd_error_e strmd error enable regs strmd error enable regs in HEVC,it is called sw_cabac_error_e for VP9, it use sw_strmd_error_e[3:0] sw_strmd_error_e[0] is for sw_vp9_tilesize_error sw_strmd_error_e[1] is for sw_vp9_segskip_error sw_strmd_error_e[2] is for sw_vp9_error_init_error sw_strmd_error_e[3] is for sw_vp9_uncpr sw_strmd_error_e[4] is for sw_vp9_refscale_error, sw_vp9_refscale_error now is no use

**rkvdec\_swreg45\_strmd\_error\_status**

Address: Operational Base + offset (0x00b4)

cabac error status

Bit	Attr	Reset Value	Description
31:28	RW	0x0	sw_colmv_error_ref_picidx colmv error ref picidx when sw_colmv_ref_error_sta is 1'b1, these bits are used for tell which dpb frame is invalid but is read by inter module it is for H264 and HEVC
27:0	RW	0x00000000	sw_strmd_error_status cabac error status strmd error status in HEVC & H264, it is called cabac error status

**rkvdec\_swreg45\_vp9\_error\_info0**

Address: Operational Base + offset (0x00b4)

vp9 error info

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	vp9_error_info0 vp9_error_info0 for vp9 16 tile cols, every cols contains 4bits vp9_error_info0[3:0] is for col 0 vp9_error_info0[1:0] is to tell tile_rows_cnt[1:0] vp9_error_info0[3:2] is to tell the error type 2'b00: no error 2'b01: tilesize error 2'b10: seg skip error 2'b11: ref scale error

**rkvdec\_swreg46\_strmd\_error\_ctu**

Address: Operational Base + offset (0x00b8)

strmd error ctu

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	sw_vp9_error_ctu0_en to tell whether is any error of vp9 1'b1: there is atleast a error in vp9 strmd now ,is no for use
23	RO	0x0	reserved
22:16	RW	0x40	sw_streamfifo_space2full stream fifo space to full It is for debug use, to tell the stream fifo space to full for HEVC, H264 and VP9
15:8	RW	0x00	sw_strmd_error_ctu_yoffset strmd error ctu yoffset strmd error ctu yoffset for HEVC, H264 and VP9 for VP9, it is the value of stsw_vp9_error_ctu0_y
7:0	RW	0x00	sw_strmd_error_ctu_xoffset strmd error ctu xoffset strmd error ctu xoffset for all HEVC and H264 and VP9 for vp9, it is the value of stsw_vp9_error_ctu0_x

**rkvdec\_swreg47\_sao\_ctu\_position**

Address: Operational Base + offset (0x00bc)

sao ctu position

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25:16	RW	0x000	sw_saowr_yoffset saowr y offset saowr y offset , its unit is 4 pixels

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
15:9	RO	0x0	reserved
8:0	RW	0x000	sw_saowr_xoffset saowr x address offset saowr x address offset, its unit is 128bit

**rkvdec\_swreg48\_h264\_refer15\_base**

Address: Operational Base + offset (0x00c0)

base address for reference picture index 15

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:4	RW	0x00000000	sw_refer15_base base address for reference picture index15 base address for reference picture index15 (the address should be 128bit align)
3	RW	0x0	sw_ref15_colmv_use_flag sw_ref15_colmv_use_flag sw_ref15_colmv_use_flag
2	RW	0x0	sw_ref15_botfield_used ref15 bottom field is used ref15 bottom field is used
1	RW	0x0	sw_ref15_topfield_used ref15 topfield is used ref15 topfield is used
0	RW	0x0	sw_ref15_field reference 15 picture field flag 1'b0: frame 1'b1: field

**rkvdec\_swreg48\_vp9\_last\_ystride**

Address: Operational Base + offset (0x00c0)

last ref ystride

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:20	RO	0x0	reserved
19:0	RW	0x00000	sw_vp9_lastfy_virstride last frame y virstride vp9 last frame y stride

**rkvdec\_swreg49\_h264\_refer15\_poc**

Address: Operational Base + offset (0x00c4)

the poc of reference picture index 15

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer15_poc the poc of reference picture index 15 the poc of reference picture index 15

**rkvdec\_swreg49\_vp9\_golden\_ystride**

Address: Operational Base + offset (0x00c4)

golden ref ystride

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:20	RO	0x0	reserved
19:0	RW	0x00000	sw_vp9_goldeny_virstride golden frame y virstride vp9 golden frame y stride

**rkvdec\_swreg50\_h264\_refer16\_poc**

Address: Operational Base + offset (0x00c8)

the poc of reference picture index 16

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer16_poc the poc of reference picture index 16 the poc of reference picture index 16

**rkvdec\_swreg50\_vp9\_altrefy\_ystride**

Address: Operational Base + offset (0x00c8)

altref ref ystride

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:20	RO	0x0	reserved
19:0	RW	0x00000	sw_vp9_altrefy_virstride altref frame y virstride vp9 altref frame y stride

**rkvdec\_swreg51\_h264\_refer17\_poc**

Address: Operational Base + offset (0x00cc)

the poc of reference picture index 17

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer17_poc the poc of reference picture index 17 the poc of reference picture index 17

**rkvdec\_swreg51\_vp9\_lastref\_yuvstride**

Address: Operational Base + offset (0x00cc)

vp9 lastref yuv stride

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:21	RO	0x0	reserved
20:0	RW	0x000000	sw_vp9_lastref_yuv_virstride lastref frame yuv virstride vp9 lastref frame yuv vir stride

**rkvdec\_swreg52\_vp9\_refcolmv\_base**

Address: Operational Base + offset (0x00d0)

vp9 colmv ref base

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:4	RW	0x00000000	sw_vp9_refcolmv_base vp9 ref colmv base vp9 ref colmv base addr
3:0	RO	0x0	reserved

**rkvdec\_swreg52\_h264\_refer18\_poc**

Address: Operational Base + offset (0x00d0)

the poc of reference picture index 18

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer18_poc the poc of reference picture index 18 the poc of reference picture index 18

**rkvdec\_swreg53\_h264\_refer19\_poc**

Address: Operational Base + offset (0x00d4)

the poc of reference picture index 19

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer19_poc the poc of reference picture index 19 the poc of reference picture index 19

**rkvdec\_swreg54\_h264\_refer20\_poc**

Address: Operational Base + offset (0x00d8)

the poc of reference picture index 20

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer20_poc the poc of reference picture index 20 the poc of reference picture index 20

**rkvdec\_swreg55\_h264\_refer21\_poc**

Address: Operational Base + offset (0x00dc)

the poc of reference picture index 21

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer21_poc the poc of reference picture index 21 the poc of reference picture index 21

**rkvdec\_swreg56\_h264\_refer22\_poc**

Address: Operational Base + offset (0x00e0)

the poc of reference picture index 22

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer22_poc the poc of reference picture index 22 the poc of reference picture index 22

**rkvdec\_swreg57\_h264\_refer23\_poc**

Address: Operational Base + offset (0x00e4)

the poc of reference picture index 23

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer23_poc the poc of reference picture index 23 the poc of reference picture index 23

**rkvdec\_swreg58\_h264\_refer24\_poc**

Address: Operational Base + offset (0x00e8)

the poc of reference picture index 24

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer24_poc the poc of reference picture index 24 the poc of reference picture index 24

**rkvdec\_swreg59\_h264\_refer25\_poc**

Address: Operational Base + offset (0x00ec)

the poc of reference picture index 25

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer25_poc the poc of reference picture index 25 the poc of reference picture index 25

**rkvdec\_swreg60\_h264\_refer26\_poc**

Address: Operational Base + offset (0x00f0)

the poc of reference picture index 26

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer26_poc the poc of reference picture index 26 the poc of reference picture index 26

**rkvdec\_swreg61\_h264\_refer27\_poc**

Address: Operational Base + offset (0x00f4)

the poc of reference picture index 27

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer27_poc the poc of reference picture index 27 the poc of reference picture index 27

**rkvdec\_swreg62\_h264\_refer28\_poc**

Address: Operational Base + offset (0x00f8)

the poc of reference picture index 28

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer28_poc the poc of reference picture index 28 the poc of reference picture index 28

**rkvdec\_swreg63\_h264\_refer29\_poc**

Address: Operational Base + offset (0x00fc)  
the poc of reference picture index 29

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_refer29_poc the poc of reference picture index 29 the poc of reference picture index 29

**rkvdec\_swreg64\_performance\_cycle**

Address: Operational Base + offset (0x0100)  
hevc performance cycle

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_performance_cycle hevc running cycle hevc running cycle if just want to analys a frame performance cycle, should set the register 0 before start a frame

**rkvdec\_swreg65\_axi\_ddr\_rdata**

Address: Operational Base + offset (0x0104)  
axi ddr read data num

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_axi_ddr_rdata axi ddr rdata num axi ddr rdata num, the unit is byte

**rkvdec\_swreg66\_axi\_ddr\_wdata**

Address: Operational Base + offset (0x0108)  
axi ddr write data number

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	sw_axi_ddr_wdata hevc write data byte num hevc write data byte num

**rkvdec\_swreg68\_performance\_sel**

Address: Operational Base + offset (0x0110)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved

Bit	Attr	Reset Value	Description
21:16	RW	0x00	perf_cnt2_sel Field0000 Abstract 0 : don't work; 1 : cycles counter for cabac in buffer empty; 2 : cycles counter for cabac in buffer full; 3 : cycles counter for cabac out buffer empty; 4 : cycles counter for cabac out buffer full; 5 : cycles counter for transd input data ready; 6 : cycles counter for transd write data to recon allow; 7 : cycles counter for dec2transd cmd empty; 8 : cycles counter for dec2transd cmd full; 9 : cycles counter for transd2dblk bs fifo empty; 10: cycles counter for transd2dblk bs fifo full; 11: cycles counter for dec2intra cmd fifo empty; 12: cycles counter for dec2intra cmd fifo full; 13: cycles counter for mc2recon cmd fifo empty; 14: cycles counter for mc2recon cmd fifo full; 15: cycles counter for mc2recon data fifo empty; 16: cycles counter for mc2recon data fifo full; 17: cycles counter for recon2filter data write allow; 18: cycles counter for inter2busifd cmd fifo empty; 19: cycles counter for inter2busifd cmd fifo full; 20: cycles counter for busifd2mc data fifo empty; 21: cycles counter for busifd2mc data fifo full; 22: cycles counter for bus working status; 23: cycles counter for dec2inter cmd fifo empty; 24: cycles counter for dec2inter cmd fifo full; 25: cycles counter for inter2mc cmd fifo empty; 26: cycles counter for inter2mc cmd fifo full; 27: cycles counter for inter2dblk bs fifo empty; 28: cycles counter for inter2dblk bs fifo full; 29: cycles counter for colmv_rbuf_empty; 30: cycles counter for colmv_rbuf_full; 31: cycles counter for colmv_wbuf_empty; 32: cycles counter for colmv_wbuf_da_full; 33: cycles counter for dblk input data valid; 34: cycles counter for dblk can't write data to sao; 35: cycles counter for dec2loopfilter cmd fifo empty; 36: cycles counter for dec2loopfilter cmd fifo full; 37: cycles counter for sao input data valid; 38: cycles counter for busifd hold back sao write data; 39: cycles counter for sao output data valid; 40: counter for dec_ctrl read cmd num
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:8	RW	0x00	perf_cnt1_sel Field0000 Abstract 0 : don't work; 1 : cycles counter for cabac in buffer empty; 2 : cycles counter for cabac in buffer full; 3 : cycles counter for cabac out buffer empty; 4 : cycles counter for cabac out buffer full; 5 : cycles counter for transd input data ready; 6 : cycles counter for transd write data to recon allow; 7 : cycles counter for dec2transd cmd empty; 8 : cycles counter for dec2transd cmd full; 9 : cycles counter for transd2dblk bs fifo empty; 10: cycles counter for transd2dblk bs fifo full; 11: cycles counter for dec2intra cmd fifo empty; 12: cycles counter for dec2intra cmd fifo full; 13: cycles counter for mc2recon cmd fifo empty; 14: cycles counter for mc2recon cmd fifo full; 15: cycles counter for mc2recon data fifo empty; 16: cycles counter for mc2recon data fifo full; 17: cycles counter for recon2filter data write allow; 18: cycles counter for inter2busifd cmd fifo empty; 19: cycles counter for inter2busifd cmd fifo full; 20: cycles counter for busifd2mc data fifo empty; 21: cycles counter for busifd2mc data fifo full; 22: cycles counter for bus working status; 23: cycles counter for dec2inter cmd fifo empty; 24: cycles counter for dec2inter cmd fifo full; 25: cycles counter for inter2mc cmd fifo empty; 26: cycles counter for inter2mc cmd fifo full; 27: cycles counter for inter2dblk bs fifo empty; 28: cycles counter for inter2dblk bs fifo full; 29: cycles counter for colmv_rbuf_empty; 30: cycles counter for colmv_rbuf_full; 31: cycles counter for colmv_wbuf_empty; 32: cycles counter for colmv_wbuf_da_full; 33: cycles counter for dblk input data valid; 34: cycles counter for dblk can't write data to sao; 35: cycles counter for dec2loopfilter cmd fifo empty; 36: cycles counter for dec2loopfilter cmd fifo full; 37: cycles counter for sao input data valid; 38: cycles counter for busifd hold back sao write data; 39: cycles counter for sao output data valid; 40: counter for dec_ctrl read cmd num
7:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:0	RW	0x00	<p>perf_cnt0_sel sel counter0 to cal which signal</p> <p>0 : don't work; 1 : cycles counter for cabac in buffer empty; 2 : cycles counter for cabac in buffer full; 3 : cycles counter for cabac out buffer empty; 4 : cycles counter for cabac out buffer full; 5 : cycles counter for transd input data ready; 6 : cycles counter for transd write data to recon allow; 7 : cycles counter for dec2transd cmd empty; 8 : cycles counter for dec2transd cmd full; 9 : cycles counter for transd2dblk bs fifo empty; 10: cycles counter for transd2dblk bs fifo full; 11: cycles counter for dec2intra cmd fifo empty; 12: cycles counter for dec2intra cmd fifo full; 13: cycles counter for mc2recon cmd fifo empty; 14: cycles counter for mc2recon cmd fifo full; 15: cycles counter for mc2recon data fifo empty; 16: cycles counter for mc2recon data fifo full; 17: cycles counter for recon2filter data write allow; 18: cycles counter for inter2busifd cmd fifo empty; 19: cycles counter for inter2busifd cmd fifo full; 20: cycles counter for busifd2mc data fifo empty; 21: cycles counter for busifd2mc data fifo full; 22: cycles counter for bus working status; 23: cycles counter for dec2inter cmd fifo empty; 24: cycles counter for dec2inter cmd fifo full; 25: cycles counter for inter2mc cmd fifo empty; 26: cycles counter for inter2mc cmd fifo full; 27: cycles counter for inter2dblk bs fifo empty; 28: cycles counter for inter2dblk bs fifo full; 29: cycles counter for colmv_rbuf_empty; 30: cycles counter for colmv_rbuf_full; 31: cycles counter for colmv_wbuf_empty; 32: cycles counter for colmv_wbuf_da_full; 33: cycles counter for dblk input data valid; 34: cycles counter for dblk can't write data to sao; 35: cycles counter for dec2loopfilter cmd fifo empty; 36: cycles counter for dec2loopfilter cmd fifo full; 37: cycles counter for sao input data valid; 38: cycles counter for busifd hold back sao write data; 39: cycles counter for sao output data valid; 40: counter for dec_ctrl read cmd num</p>

**rkvdec\_swreg69\_performance\_cnt0**

Address: Operational Base + offset (0x0114)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	perf_cnt0 Field0000 Abstract Field0000 Description

**rkvdec\_swreg70\_performance\_cnt1**

Address: Operational Base + offset (0x0118)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	perf_cnt1

**rkvdec\_swreg71\_performance\_cnt2**

Address: Operational Base + offset (0x011c)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	perf_cnt2 Field0000 Abstract Field0000 Description

**rkvdec\_swreg72\_h264\_refer30\_poc**

Address: Operational Base + offset (0x0120)

the poc of reference picture index 30

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer30_poc the poc of reference picture index 30 the poc of reference picture index 30

**rkvdec\_swreg73\_h264\_refer31\_poc**

Address: Operational Base + offset (0x0124)

the poc of reference picture index 31

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer31_poc the poc of reference picture index 31 the poc of reference picture index 31

**rkvdec\_swreg74\_h264\_cur\_poc1**

Address: Operational Base + offset (0x0128)

h264 cur poc for bottom filed

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_h264_cur_poc1 h264 cur poc for bottom field h264 cur poc for bottom field

**rkvdec\_swreg75\_h264\_errorinfo\_base**

Address: Operational Base + offset (0x012c)

h264 error info base addr

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:3	RW	0x00000000	sw_errorinfo_base error info base addr error info base addr every slice contains 256 bits error info
2:0	RO	0x0	reserved

**rkvdec\_swreg76\_h264\_errorinfo\_num**

Address: Operational Base + offset (0x0130)

h264 error info num

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:16	RW	0x0000	sw_error_packet_num error packet number error packet number
15	RW	0x0	sw_strmd_detect_error_flag strmd error detect flag streamd detect error flag
14	RO	0x0	reserved
13:0	RW	0x0000	sw_slicedec_num slice dec num h264 decoded num, the max slice num for H264 is 4096

**rkvdec\_swreg77\_h264\_error\_e**

Address: Operational Base + offset (0x0134)

h264 error enable high bits

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:0	RW	0x00000000	sw_h264_error_en_highbits h264 error enable high bits h264 error enable bits

**VDPU\_SWREG0\_NEW\_VERSION**

Address: Operational Base + offset (0x0000)

ID register(read only)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:28	RO	0x0	reserved
27:24	RO	0x3	major_version 0:1080p support 1:2160p support

Bit	Attr	Reset Value	Description
23:16	RO	0x68	minor_version 0: audis 1: audi 2: maybach 3:audib ff:share memory with hevc,so should read verision from hevc register
15:0	RW	0x0000	build the rtl's svn num in ic server

**VDPU\_SWREG0**

Address: Operational Base + offset (0x0000)

Register0000 Abstract

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	pro_num product number
15:12	RW	0x0	major_version major_version major_version
11:4	RW	0x00	minor_version minor_version minor_version
3	RW	0x0	ID_ASCII_EN ASCII type product ID enable ASCII type product ID enable
2:0	RW	0x0	build_version build_version build_version

**VDPU\_SWREG1**

Address: Operational Base + offset (0x0004)

interrupt register decoder

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved
24	RW	0x0	sw_dec_pic_inf B slice detected. This signal is driven high during picture ready interrupt if B-type slice is found. This bit does not launch interrupt but is used to inform SW about h264 tools. [note]:the h264 decoder will use these bits.
23:19	RO	0x0	reserved
18	RW	0x0	sw_dec_timeout Interrupt status bit decoder timeout. When high the decoder 0 has been idling for too long. HW will self reset. Possible only if timeout interrupt is enabled [note]:the h264 decoder will use these bits.

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
17	RW	0x0	<p>sw_dec_slice_int Interrupt status bit dec_slice_decoded. When high SW must set new base addresses for sw_dec_out_base and sw_jpg_ch_out_base before resetting this status bit. Used for JPEG snapshot modes [note]:the JPEG decoder will use these bits.</p>
16	RW	0x0	<p>sw_dec_error_int Interrupt status bit input stream error. When high, an error is found in input data stream decoding. HW will self reset. [note]:the h264 decoder will use these bits.</p>
15	RW	0x0	<p>sw_dec_aso_int Interrupt status bit ASO (Arbitrary Slice Ordering) detected. When high, ASO detected in input data stream decoding. HW will self reset. [note]:the h264 decoder will use these bits.</p>
14	RW	0x0	<p>sw_dec_buffer_int Interrupt status bit input buffer empty. When high, input stream buffer is empty but picture is not ready. HW will not self reset. [note]:the h264 decoder will use these bits.</p>
13	RW	0x0	<p>sw_dec_bus_int Interrupt status bit bus. Error response from bus. HW will self reset [note]:the h264 decoder will use these bits.</p>
12	RW	0x0	<p>sw_dec_rdy_int Interrupt status bit decoder. When this bit is high decoder has decoded a picture. HW will self reset. [note]:the h264 decoder will use these bits.</p>
11:9	RO	0x0	reserved
8	RW	0x0	<p>sw_dec_irq Decoder IRQ. When high, decoder requests an interrupt. SW will reset this after interrupt is handled. [note]:the h264 decoder will use these bits.</p>
7:5	RO	0x0	reserved
4	RW	0x0	<p>sw_dec_irq_dis Decoder IRQ disable. When high, there are no interrupts concerning decoder from HW. Polling must be used to see the interrupt statuses. [note]:the h264 decoder will use these bits.</p>
3:1	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
0	RW	0x0	<p>sw_dec_en decoder enable.</p> <p>Decoder enable. Setting this bit high will start the decoding operation. HW will reset this when picture is processed or ASO or stream error is detected or bus error or timeout interrupt is given.</p> <p>[note]:the h264 decoder will use these bits.</p>

**VDPU\_SWREG2**

Address: Operational Base + offset (0x0008)

device configuration register decoder

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RW	0x01	<p>sw_dec_axi_rd_id</p> <p>Read ID used for decoder reading services in AXI bus (if connected to AXI)</p> <p>[note]:the h264 decoder will use these bits.</p>
23	RW	0x0	<p>sw_dec_timeout_e</p> <p>Timeout interrupt enable. If enabled HW may return timeout interrupt in case HW gets stucked while decoding picture.</p> <p>[note]:the h264 decoder will use these bits.</p>
22	RW	0x0	<p>sw_dec_strswap32_e</p> <p>Decoder input 32bit data swap for stream data (may be used for 64 bit environment):</p> <p>0 = no swapping of 32 bit words</p> <p>1 = 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled))</p> <p>[note]:the h264 decoder will use these bits.</p>
21	RW	0x0	<p>sw_dec_strendian_e</p> <p>Decoder input endian mode for stream data:</p> <p>0 = Big endian (0-1-2-3 order)</p> <p>1 = Little endian (3-2-1-0 order)</p> <p>[note]:the h264 decoder will use these bits.</p>
20	RW	0x0	<p>sw_dec_inswap32_e</p> <p>Decoder input 32bit data swap for other than stream data (may be used for 64 bit environment):</p> <p>0 = no swapping of 32 bit words</p> <p>1 = 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled))</p> <p>[note]:the h264 decoder will use these bits.</p>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
19	RW	0x0	<p>sw_dec_outswap32_e Decoder output 32bit data swap (may be used for 64 bit environment): 0 = no swapping of 32 bit words 1 = 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled)) [note]:the h264 decoder will use these bits.</p>
18	RW	0x0	<p>sw_dec_data_disc_e Data discard enable. Precise burst lengths are used with reading services. Extra data is discarded internally. [note]:the h264 decoder will use these bits.</p>
17	RW	0x0	<p>sw_tiled_mode_msb Tiled mode msb. Concatenated to Tiled mode lsb which form 2 bit tiled mode. Definition of tiledmode: 0 = Tiled mode not enabled 1 = Tiled mode enabled for 8x4 tile size 2,3 Reserved [note]:the h264 decoder will use these bits.</p>
16:11	RW	0x00	<p>sw_dec_latency Decoder master interface additional latency. Can be used to slow down decoder HW between services in steps of 8 clock cycles: 0 = no latency 1 = minimum 8 cycles of IDLE between services 2 = minimum 16 cycles of IDLE between services ... 63 = minimum latency of 504 cycles of IDLE between services [note]:the h264 decoder will use these bits.</p>
10	RW	0x1	<p>sw_dec_clk_gate_e Decoder dynamic clock gating enable: 0 = Clock is running for all structures 1 = Clock is gated for decoder structures that are not used Note: Clock gating value can be changed only when decoder is disabled</p>
9	RW	0x0	<p>sw_dec_in_endian Decoder input endian mode for other than stream data: 0 = Big endian (0-1-2-3 order) 1 = Little endian (3-2-1-0 order) [note]:the h264 decoder will use these bits.</p>
8	RW	0x0	<p>sw_dec_out_endian Decoder output endian mode: 0 = Big endian (0-1-2-3 order) 1 = Little endian (3-2-1-0 order) [note]:the h264 decoder will use these bits.</p>

Bit	Attr	Reset Value	Description
7	RW	0x0	sw_tiled_mode_lsb Tiled mode lsb. Concatenated to Tiled mode msb which form 2 bit tiled mode. Defined in tiled_mode_msbs [note]:the h264 decoder will use these bits.
6	RW	0x0	sw_dec_adv_pre_dis Advanced PREFETCH mode disable (advanced reference picture reading mode for video) [note]:the h264 decoder will use these bits.
5	RW	0x0	sw_dec_scmd_dis AXI Single Command Multiple Data 0 disable. (where only the first addresses of the burst are given from address generator). This bit is used to disable the feature (possible SW workaround if something is not working correctly) [note]:the h264 decoder will use these bits.
4:0	RW	0x00	sw_dec_max_burst Maximum burst length for decoder bus transactions. Valid values: AXI: 1-16 [note]:the h264 decoder will use these bits.

**VDPU\_SWREG3**

Address: Operational Base + offset (0x000c)

Device control register 0(decmode, picture type etc)

Bit	Attr	Reset Value	Description
31:28	RW	0x0	sw_dec_mode Decoding mode: 0 = H.264, 1 = MPEG-4, 2 = H.263, 3 = JPEG, 5 = MPEG-2, 6 = MPEG-1, others = reserved [note]:all the decoder mode will use these bits.
27	RW	0x0	sw_rlc_mode_e RLC mode enable: 1 = HW decodes video from RLC input data + side information (Differential MV's, separate DC coeffs, Intra 4x4 modes, MB control). Valid only for H.264 Baseline and MPEG-4 SP. 0 = HW decodes video from bit stream (VLC mode) + side information [note]:the h264 and MPEG4 decoder will use these bits.
26	RW	0x0	sw_skip_mode
25	RW	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
24	RW	0x0	sw_pjpeg_e Progressive JPEG enable: 0 = baseline JPEG 1 = progressive JPEG
23	RW	0x0	sw_pic_interlace_e Coding mode of the current picture: 0 = progressive 1 = interlaced [note]:the h264 decoder will use these bits.
22	RW	0x0	sw_pic_fieldmode_e Structure of the current picture (residual structure) 0 = frame structure, this means MBAFF structured picture for interlaced sequence 1 = field structure [note]:the h264 decoder will use these bits.
21	RW	0x0	sw_pic_b_e B picture enable for current picture: 0=picture type is I or P depending on sw_pic_inter_e 1=picture type is BI (vc1)/D (mpeg1) or B depending on sw_pic_inter_e (not valid for H264 since its slice based information)
20	RW	0x0	sw_pic_inter_e Picture type. 1= Inter type (P) 0= Intra type (I) See also sw_pic_b_e
19	RW	0x0	sw_pic_topfield_e If field structure is enabled this bit informs which one of the fields is being decoded: 0 = bottom field 1 = top field [note]:the h264 decoder will use these bits.
18	RW	0x0	sw_fwd_interlace_e Coding mode of forward reference picture: 0 = progressive 1 = interlaced Note: for backward reference picture the coding mode is always same as for current picture.
17	RW	0x0	reserved
16	RW	0x0	sw_ref_topfield_e Indicates which field should be used as reference if sw_ref_frames = 0 0 = bottom field 1 = top field used only in VC-1 mode

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
15	RW	0x0	<p>sw_dec_out_dis Disable decoder output picture writing: 0 = Decoder output picture is written to external memory 1 = Decoder output picture is not written to external memory [note]:the h264 decoder will use these bits.</p>
14	RW	0x0	<p>sw_filtering_dis De-block filtering disable 1 = filtering is disabled for current picture 0 = filtering is enabled for current picture [note]:the h264 decoder will use these bits.</p>
13	RW	0x0	<p>sw_pic_fixed_quant sw_pic_fixed_quant (DEC mode is VC-1 and AVS) 0 = Quantization parameter can vary inside picture 1 = Quantization parameter is fixed (pquant) sw_mvc_e(DEC mode is H264) multi view coding enable. Possible for H264 only [note]:the h264 decoder will use these bits.</p>
12	RW	0x0	<p>sw_write_mvs_e Direct mode motion vector write enable for current picture / MPEG2 motion vector write enable for error concealment purposes: 0 = writing disabled for current picture 1 = the direct mode motion vectors are written to extrenal memory. H264 direct mode motion vectors are written to DPB aside with the corresponding reference picture. Other decoding mode dir mode mvs are written to external memory starting from sw_dir_mv_base [note]:the h264 decoder will use these bits.</p>
11	RW	0x0	<p>sw_reftopfirst_e Indicates which FWD reference field has been decoded first. 0 = FWD reference bottom field 1 = FWD reference top field [note]:the h264 decoder will use these bits.</p>
10	RW	0x0	<p>sw_seq_mbaff_e Sequence includes MBAFF coded pictures [note]:the h264 decoder will use these bits.</p>
9	RW	0x0	<p>sw_picord_count_e h264_high config: Picture order count table read enable. If enabled HW will read picture order counts from memory in the beginning of picture [note]:the h264 decoder will use these bits.</p>
8	RW	0x0	<p>sw_dec_timeout_mode dec timeout mode selset when 1'b0,timeout cycle is 181'b1 when 1'b1,timeout cycle is 221'b1 [note]:the h264 decoder will use these bits.</p>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
7:0	RW	0x01	sw_dec_axi_wr_id Write ID used for decoder writing services in AXI bus (if connected to AXI) [note]:the h264 decoder will use these bits.

**VDPU\_SWREG4\_H264**

Address: Operational Base + offset (0x0010)  
decoder control register 1(picture parameters)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:23	RW	0x000	sw_pic_mb_width Picture width in macroblocks = ((width in pixels + 15) /16) [note]:the h264 decoder will use these bits.
22:19	RO	0x0	reserved
18:11	RW	0x00	sw_pic_mb_height_p Picture height in macroblocks =((height in pixels+15)/16). Picture height is informed as size of the (progressive) frame also for single field (of interlaced content) is being decoded [note]:the h264 decoder will use these bits.
10:5	RO	0x0	reserved
4:0	RW	0x00	sw_ref_frames 264: num_ref_frames, maximum number of short and long term reference frames in decoded picture buffer.

**VDPU\_SWREG4**

Address: Operational Base + offset (0x0010)  
decoder control register 1(picture parameters)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:23	RW	0x000	sw_pic_mb_width Picture width in macroblocks = ((width in pixels + 15) /16)
22:19	RW	0x0	sw_mb_width_off The amount of meaningfull horizontal pixels in last MB (width offset) 0 if exactly 16 pixels multiple picture and all the horizontal pixels in last MB are meaningfull
18:11	RW	0x00	sw_pic_mb_height_p Picture height in macroblocks =((height in pixels+15)/16). Picture height is informed as size of the (progressive) frame also for single field (of interlaced content) is being decoded
10:7	RW	0x0	sw_mb_height_off The amount of menaingfull vertical pixels in last MB (height offset 0 if exactly 16 pixels multiple picture and all the vertical pixels in last MB are meaningfull
6	RW	0x0	sw_alt_scan_e indicates alternative vertical scan method used for interlaceedd frames

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
5:3	RW	0x0	sw_pic_mb_w_ext Picture mb width extension. If sw_pic_mb_width does not fit to 9 bits then these bits are used to increase the range upto 11 bits (used as 3 msb)
2:0	RW	0x0	sw_pic_mb_h_ext Picture mb height extension. If sw_pic_mb_height_p does not fit to 9 bits then these bits are used to increase the range upto 11 bits (used as 3 msb)

**VDPU\_SWREG5**

Address: Operational Base + offset (0x0014)  
decoder control register2 (stream decoding table selects)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:26	RW	0x00	sw_strm_start_bit Exact bit of stream start word where decoding can be started (associates with sw_rlc_vlc_base)
25	RW	0x0	sw_sync_marker_e Sync markers enable: '0' = sync markers are not used, '1' = sync markers are used. For progressive JPEG this indicates that there are restart markers in the stream after restart interval steps
24	RW	0x0	sw_type1_quant_e MPEG4: Type 1 quantization enable '0' = type 2 inverse Q method '1' = type 1 inverse Q method (Q-tables used) H264 (h264_high config): scaling matrix enable: '0' = normal transform '1' = use scaling matrix for transform (read from external)
23:19	RW	0x00	sw_ch_qp_offset Chroma Qp filter offset. (For H.264 this offset concerns Cb only)
18:14	RW	0x00	sw_ch_qp_offset2 Chroma Qp filter offset for cr type
13:1	RO	0x0	reserved
0	RW	0x0	sw_fieldpic_flag_e Flag for streamd that field_pic_flag exists in stream

**VDPU\_SWREG5\_H264**

Address: Operational Base + offset (0x0014)  
decoder control register2 (stream decoding table selects)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:26	RW	0x00	sw_strm_start_bit Exact bit of stream start word where decoding can be started (associates with sw_rlc_vlc_base)
25	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
24	RW	0x0	sw_type1_quant_e MPEG4: Type 1 quantization enable '0' = type 2 inverse Q method '1' = type 1 inverse Q method (Q-tables used) H264 (h264_high config): scaling matrix enable: '0' = normal transform '1' = use scaling matrix for transform (read from external)
23:19	RW	0x00	sw_ch_qp_offset Chroma Qp filter offset. (For H.264 this offset concerns Cb only)
18:14	RW	0x00	sw_ch_qp_offset2 Chroma Qp filter offset for cr type
13:1	RO	0x0	reserved
0	RW	0x0	sw_fieldpic_flag_e Flag for streamd that field_pic_flag exists in stream

**VDPU\_SWREG6**

Address: Operational Base + offset (0x0018)  
decoder control register 3( stream buffer information)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RW	0x0	sw_start_code_e Bit for indicating stream start code existence: '0' = stream doesn't contain start codes '1' = stream contains start codes [note]:the h264 decoder will use these bits.
30:25	RW	0x00	sw_init_qp Initial value for quantization parameter (picture quantizer). [note]:the h264 decoder will use these bits.
24	RW	0x0	sw_ch_8pix_ileav_e Enable for additional chrominance data format writing where decoder writes chrominance in group of 8 pixels of Cb and then corresponding 8 pixels of Cr. Data is written to sw_dec_ch8pix_base. Cannot be used if tiled mode is enabled [note]:the h264 decoder will use these bits.

Bit	Attr	Reset Value	Description
23:0	RW	0x000000	<p>sw_stream_len Amount of stream data bytes in input buffer. If the given buffer size is not enough for finishing the picture the corresponding interrupt is given and new stream buffer base address and stream buffer size information should be given (associates with sw_rlc_vlc_base). For VC-1 the buffer must include data for one picture/slice of the picture For H264/MPEG4/H263/MPEG2/MPEG1 the buffer must include at least data for one slice of the picture For JPEG the buffer size must be a multiple of 256 bytes or the amount of data for one picture. [note]:the h264 decoder will use these bits.</p>

**VDPU\_SWREG7**

Address: Operational Base + offset (0x001c)  
decoder control register 4(H264, VC-1 control )

Bit	Attr	Reset Value	Description
31	RW	0x0	<p>sw_cabac_e CABAC enable [note]:the h264 decoder will use these bits.</p>
30	RW	0x0	<p>sw_blackwhite_e '0' = 4:2:0 sampling format '1' = 4:0:0 sampling format (H264 monochroma) [note]:the h264 decoder will use these bits.</p>
29	RW	0x0	<p>sw_dir_8x8_infer_e Specifies the method to use to derive luma motion vectors in B_skip, B_Direct_16x16 and B_direct_8x8_inference_flag (see direct_8x8_inference flag) [note]:the h264 decoder will use these bits.</p>
28	RW	0x0	<p>sw_weight_pred_e Weighted prediction enable for P slices [note]:the h264 decoder will use these bits.</p>
27:26	RW	0x0	<p>sw_weight_bipr_idc weighted prediction specification for B slices: "00" = default weighted prediction is applied to B slices "01" = explicit weighted prediction shall be applied to B slices "10" = implicit weighted prediction shall be applied to B slices [note]:the h264 decoder will use these bits.</p>
25:21	RO	0x0	reserved
20:16	RW	0x00	<p>sw_framenum_len H.264: Bit length of frame_num in data stream [note]:the h264 decoder will use these bits.</p>

Bit	Attr	Reset Value	Description
15:0	RW	0x0000	sw_framenum current frame_num, used to identify short-term reference frames. Used in reference picture reordering [note]:the h264 decoder will use these bits.

**VDPU\_SWREG8**

Address: Operational Base + offset (0x0020)

decoder control register 5(H264 control)

Bit	Attr	Reset Value	Description
31	RW	0x0	sw_const_intra_e constrained_intra_pred_flag equal to 1 specifies that intra prediction uses only neighbouring intra macroblocks in prediction. When equal to 0 also neighbouring inter macroblocks are used in intra prediction process. [note]:the h264 decoder will use these bits.
30	RW	0x0	sw_filt_ctrl_pres deblocking_filter_control_present_flag indicates whether extra variables controlling characteristics of the deblocking filter are present in the slice header. [note]:the h264 decoder will use these bits.
29	RW	0x0	sw_rdpic_cnt_pres redundant_pic_cnt_present_flag specifies whether redundant_pic_cnt syntax elements [note]:the h264 decoder will use these bits.
28	RW	0x0	sw_8x8trans_flag_e 8x8 transform flag enable for stream decoding [note]:the h264 decoder will use these bits.
27:17	RW	0x000	sw_refpic_mk_len Length of decoded reference picture marking bits [note]:the h264 decoder will use these bits.
16	RW	0x0	sw_idr_pic_e IDR (instantaneous decoding refresh) picture flag. [note]:the h264 decoder will use these bits.
15:0	RW	0x0000	sw_idr_pic_id idr_pic_id, identifies IDR (instantaneous decoding refresh) picture [note]:the h264 decoder will use these bits.

**VDPU\_SWREG9**

Address: Operational Base + offset (0x0024)

decoder control register 6

Bit	Attr	Reset Value	Description
-----	------	-------------	-------------

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RW	0x00	sw_pps_id pic_parameter_set_id, identifies the picture parameter set that is referred to in the slice header. [note]:the h264 decoder will use these bits.
23:19	RW	0x00	sw_refidx1_active Specifies the maximum reference index that can be used while decoding inter predicted macro blocks. [note]:the h264 decoder will use these bits.
18:14	RW	0x00	sw_refidx0_active Specifies the maximum reference index that can be used while decoding inter predicted macro blocks. This is same as in previous decoders (width increased with q bit) [note]:the h264 decoder will use these bits.
13:8	RO	0x0	reserved
7:0	RW	0x00	sw_poc_length Length of picture order count field in stream [note]:the h264 decoder will use these bits.

**VDPU\_SREG10\_H264\_RLC**

Address: Operational Base + offset (0x0028)

Base address for differential motion vector base address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_diff_mv_base for H264 and MPEG4, RLC mode: Differential motion vector base address.
1:0	RO	0x0	reserved

**VDPU\_SREG10\_H264**

Address: Operational Base + offset (0x0028)

Base address for differential motion vector base address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:25	RW	0x00	sw_pinit_rlist_f9 initial reference picture list for P forward picid 9
24:20	RW	0x00	sw_pinit_rlist_f8 initial reference picture list for P forward picid 8
19:15	RW	0x00	sw_pinit_rlist_f7 initial reference picture list for P forward picid 7
14:10	RW	0x00	sw_pinit_rlist_f6 initial reference picture list for P forward picid 6
9:5	RW	0x00	sw_pinit_rlist_f5 initial reference picture list for P forward picid 5
4:0	RW	0x00	sw_pinit_rlist_f4 initial reference picture list for P forward picid 4

**VDPU\_SWREG11\_H264\_RLC**

Address: Operational Base + offset (0x002c)  
decoder control register 7

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_i4x4_or_dc_base RLC mode: H.264: Intra prediction 4x4 mode base address. RLC mode: MPEG-4: DC component base address.
1:0	RO	0x0	reserved

**VDPU\_SWREG11\_H264**

Address: Operational Base + offset (0x002c)  
decoder control register 7

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:25	RW	0x00	sw_pinit_rlist_f15 Initial reference picture list for P forward picid 15
24:20	RW	0x00	sw_pinit_rlist_f14 Initial reference picture list for P forward picid 14
19:15	RW	0x00	sw_pinit_rlist_f13 Initial reference picture list for P forward picid 13
14:10	RW	0x00	sw_pint_rlist_f12 Initial reference picture list for P forward picid 12
9:5	RW	0x00	sw_pint_rlist_f11 Initial reference picture list for P forward picid 11
4:0	RW	0x00	sw_pint_rlist_f10 Initial reference picture list for P forward picid 10

**VDPU\_SWREG12**

Address: Operational Base + offset (0x0030)

Base address for RLC data (RLC) / stream start address/decoded

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_rlc_vlc_base RLC mode: Base address for RLC data (swreg3.sw_rlc_mode_e = 1). VLC mode: Stream start address / end addr+I288ess with byte precision (swreg4.rlc_mode_en = 0), start bit number in swreg5.stream_start_bit. When sw_dec_buffer_int is high or sw_dec_e is low this register contains HW return value of last_byte_address (not valid for jpeg) where stream has been read (and used) in accuracy of byte. For debug purposes the last_byte_address is also written when stream error/ASO is detected even though it may not be accurate. [note]:the h264 decoder will use these bits.
1:0	RO	0x0	reserved

**VDPU\_SWREG13**

Address: Operational Base + offset (0x0034)

Base address for decoded picture / base address for JPEG deco

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_dec_out_base Video: Base address for decoder output picture. Points directly to start of decoder output picture or field. JPEG snapshot: Base address for decoder output luminance picture [note]:the h264 decoder will use these bits.
1:0	RO	0x0	reserved

**VDPU\_SWREG14**

Address: Operational Base + offset (0x0038)

Base address for reference picture index 0 / base address for J

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_refer0_base Base address for reference picture index 0. See picture index definition from toplevel_sp [note]:the h264 decoder will use these bits.
1	RW	0x0	sw_refer0_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields [note]:the h264 decoder will use these bits.
0	RW	0x0	sw_refer0_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture [note]:the h264 decoder will use these bits.

**VDPU\_SWREG15\_JPEG\_ROI**

Address: Operational Base + offset (0x003c)

JPEG roi control

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:20	RO	0x0	reserved
19	RW	0x0	sw_jpegroi_in_endian jpeg offset input endian sw_jpegroi_in_endian 0 = big endian (0-1-2-3 order) 1 = little endian (3-2-1-0 order)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
18	RW	0x0	sw_jpegroi_in_swap32 jpeg offset input 32-bit swap sw_jpegroi_in_swap32 0: no swapping of 32 bit words 1: 32bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1 byte order (also little endian should be enabled))
17:16	RW	0x0	sw_roi_sample_size ROI MB num sample each time ROI MB num sample each time 00:1 01:8 10:16 11:8
15:12	RW	0x0	sw_roi_distance roi distance The distance between the sample MB and ROI start MB
11:10	RW	0x0	sw_roi_out_sel roi output selection ROI output selection 00: output offset/dc 01: output picture 10: output offset/dc and picture 11: output offset/dc
9	RW	0x0	sw_roi_decode roi decode JPEG ROI decode 0: build offset/dc table 1: ROI decode
8	RW	0x0	sw_roi_en roi enable JPEG roi mode enable 0: normal jpeg decode mode 1: JPEG roi mode
7:0	RO	0x0	reserved

**VDPU\_SWREG15**

Address: Operational Base + offset (0x003c)

Base address for reference picture index 1 / JPEG control

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_refer1_base Base address for reference picture index 1. See picture index definition from toplevel_sp [note]:the h264 decoder will use these bits.

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
1	RW	0x0	sw_refer1_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer1_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

**VDPU\_SWREG16**

Address: Operational Base + offset (0x0040)

base address for reference picture index 2 / List of VLC code len

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_refer2_base Base address for reference picture index 2. See picture index definition from toplevel_sp [note]:the h264 decoder will use these bits.
1	RW	0x0	sw_refer2_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields [note]:the h264 decoder will use these bits.
0	RW	0x0	sw_refer2_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture [note]:the h264 decoder will use these bits.

**VDPU\_SWREG17**

Address: Operational Base + offset (0x0044)

Base address for reference picture index 3 / List of VLC code len

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_refer3_base Base address for reference picture index 3. See picture index definition from toplevel_sp [note]:the h264 decoder will use these bits.
1	RW	0x0	sw_refer3_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields [note]:the h264 decoder will use these bits.

Bit	Attr	Reset Value	Description
0	RW	0x0	<p>sw_refer3_topc_e</p> <p>Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture</p> <p>[note]:the h264 decoder will use these bits.</p>

**VDPU\_SWREG18**

Address: Operational Base + offset (0x0048)

Base address for reference picture index 4 / VC1 control / MPE

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	<p>sw_refer4_base</p> <p>Base address for reference picture index 4. See picture index definition from toplevel_sp</p> <p>[note]:the h264 decoder will use these bits.</p>
1	RW	0x0	<p>sw_refer4_field_e</p> <p>Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields</p> <p>[note]:the h264 decoder will use these bits.</p>
0	RW	0x0	<p>sw_refer4_topc_e</p> <p>Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture</p> <p>[note]:the h264 decoder will use these bits.</p>

**VDPU\_SWREG19**

Address: Operational Base + offset (0x004c)

Base address for reference picture index 5

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	<p>sw_refer5_base</p> <p>Base address for reference picture index 5. See picture index definition from toplevel_sp</p> <p>[note]:the h264 decoder will use these bits.</p>
1	RW	0x0	<p>sw_refer5_field_e</p> <p>Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields</p> <p>[note]:the h264 decoder will use these bits.</p>
0	RW	0x0	<p>sw_refer5_topc_e</p> <p>Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture</p> <p>[note]:the h264 decoder will use these bits.</p>

**VDPU\_SWREG20**

Address: Operational Base + offset (0x0050)

Base address for reference picture index 6

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_refer6_base Base address for reference picture index 6. See picture index definition from toplevel_sp [note]:the h264 decoder will use these bits.
1	RW	0x0	sw_refer6_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields [note]:the h264 decoder will use these bits.
0	RW	0x0	sw_refer6_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture [note]:the h264 decoder will use these bits.

**VDPU\_SWREG21**

Address: Operational Base + offset (0x0054)

Base address for reference picture index 7

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_refer7_base Base address for reference picture index 7. See picture index definition from toplevel_sp [note]:the h264 decoder will use these bits.
1	RW	0x0	sw_refer7_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields [note]:the h264 decoder will use these bits.
0	RW	0x0	sw_refer7_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture [note]:the h264 decoder will use these bits.

**VDPU\_SWREG22**

Address: Operational Base + offset (0x0058)

Base address for reference picture index 8

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_refer8_base Base address for reference picture index 8. See picture index definition from toplevel_sp

Bit	Attr	Reset Value	Description
1	RW	0x0	sw_refer8_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer8_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

**VDPU\_SWREG23**

Address: Operational Base + offset (0x005c)

Base address for reference picture index 9

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_refer9_base Base address for reference picture index 9. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer9_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer9_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

**VDPU\_SWREG24**

Address: Operational Base + offset (0x0060)

Base address for reference picture index 10

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_refer10_base Base address for reference picture index 10. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer10_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer10_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

**VDPU\_SWREG25**

Address: Operational Base + offset (0x0064)

Base address for reference picture index 11

Bit	Attr	Reset Value	Description
-----	------	-------------	-------------

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:3	RO	0x0	reserved
2	RW	0x0	sw_refer11_base Base address for reference picture index 11. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer11_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer11_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

**VDPU\_SWREG26**

Address: Operational Base + offset (0x0068)

Base address for reference picture index 12

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_refer12_base Base address for reference picture index 12. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer12_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer12_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

**VDPU\_SWREG27**

Address: Operational Base + offset (0x006c)

Base address for reference picture index 13

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_refer13_base Base address for reference picture index 13. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer13_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer13_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

**VDPU\_SWREG28**

Address: Operational Base + offset (0x0070)

Base address for reference picture index14

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_refer14_base Base address for reference picture index 14. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer14_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer14_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

**VDPU\_SWREG29**

Address: Operational Base + offset (0x0074)

Base address for reference picture index15

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_refer15_base Base address for reference picture index 15. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer15_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer15_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

**VDPU\_SWREG30**

Address: Operational Base + offset (0x0078)

Reference picture numbers for index 0 and 1 (H264 VLC)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_refer1_nbr Number for reference picture index 1
15:0	RW	0x0000	sw_refer0_nbr Number for reference picture index 0

**VDPU\_SWREG31**

Address: Operational Base + offset (0x007c)

Reference picture numbers for index 2 and 3 (H264 VLC) /

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_refer3_nbr Number for reference picture index 3
15:0	RW	0x0000	sw_refer2_nbr Number for reference picture index 2

**VDPU\_SWREG32**

Address: Operational Base + offset (0x0080)

Reference picture numbers for index 4 and 5 (H264 VLC)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_refer5_nbr Number for reference picture index 5
15:0	RW	0x0000	sw_refer4_nbr Number for reference picture index 4

**VDPU\_SWREG33**

Address: Operational Base + offset (0x0084)

Reference picture numbers for index 6 and 7 (H264 VLC)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_refer7_nbr Number for reference picture index 7
15:0	RW	0x0000	sw_refer6_nbr Number for reference picture index 6

**VDPU\_SWREG34**

Address: Operational Base + offset (0x0088)

Reference picture numbers for index 8 and 9 (H264 VLC)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_refer9_nbr Number for reference picture index 9
15:0	RW	0x0000	sw_refer8_nbr Number for reference picture index 8

**VDPU\_SWREG35\_JPEG\_ROI**

Address: Operational Base + offset (0x008c)

JPEG roi offset/dc base address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_jpegdcoff_base JPEG roi offset/dc base address JPEG roi offset/dc base address
1:0	RO	0x0	reserved

**VDPU\_SWREG35**

Address: Operational Base + offset (0x008c)

Reference picture numbers for index 10 and 11 (H264 VLC)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_refer11_nbr Number for reference picture index 11
15:0	RW	0x0000	sw_refer10_nbr Number for reference picture index 10

**VDPU\_SWREG36**

Address: Operational Base + offset (0x0090)

Reference picture numbers for index 12 and 13 (H264 VLC)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_refer13_nbr Number for reference picture index 13
15:0	RW	0x0000	sw_refer12_nbr Number for reference picture index 12

**VDPU\_SWREG36\_JPEG\_ROI**

Address: Operational Base + offset (0x0090)

JPEG roi offset/dc length

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:17	RO	0x0	reserved
16:0	RW	0x00000	sw_jpegdcoff_len sw_jpegdcoff_len The number of 64bit jpegdcoff, it can be used both when sw_roi_decode is 1'b0 or 1'b1

**VDPU\_SWREG37**

Address: Operational Base + offset (0x0094)

Reference picture numbers for index 14 and 15 (H264 VLC)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_refer15_nbr Number for reference picture index 15
15:0	RW	0x0000	sw_refer14_nbr Number for reference picture index 14

**VDPU\_SWREG38**

Address: Operational Base + offset (0x0098)

Reference picture long term flags (H264 VLC) prediction filt

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:22	RW	0x000	sw_pred_bc_tap_3_3 Prediction filter set 3, tap 3
21:12	RW	0x000	sw_pred_bc_tap_4_0 Prediction filter set 4, tap 0
11:2	RW	0x000	sw_perd_bc_tap_4_1 Prediction filter set 4, tap 1
1:0	RO	0x0	reserved

**VDPU\_SWREG38\_H264**

Address: Operational Base + offset (0x0098)

Reference picture numbers for index 12 and 13 (H264 VLC)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer_lterm_e long term flag for reference picture index [31:0]

**VDPU\_SWREG39**

Address: Operational Base + offset (0x009c)

Reference picture valid flags (H264 VLC) prediction filter ta

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:22	RW	0x000	sw_pred_bc_tap_4_2 Prediction filter set 4, tap 2
21:12	RW	0x000	sw_pred_bc_tap_4_3 Prediction filter set 4, tap 3
11:2	RW	0x000	sw_pred_bc_tap_5_0 Prediction filter set 5, tap 0
1:0	RO	0x0	reserved

**VDPU\_SWREG39\_H264**

Address: Operational Base + offset (0x009c)

Reference picture numbers for index 12 and 13 (H264 VLC)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer_valid_e valid flag for reference picture index [31:0]

**VDPU\_SWREG40**

Address: Operational Base + offset (0x00a0)

Base address for standard dependent tables

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_qtable_base Base address for standard dependent tables: JPEG= AC,DC, QP tables MPEG4=QP table base address if type 1 quantization is used MPEG2=QP table base address H.264=base address for various tables [note]:the h264 decoder will use these bits.
1:0	RO	0x0	reserved

**VDPU\_SWREG41**

Address: Operational Base + offset (0x00a4)

Base address for direct mode motion vectors

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_dir_mv_base Direct mode motion vector write/read base address. For H264 this is used only for direct mode motion vector write base. Progressive JPEG: ACDC coefficient read/write base address. If current round is for DC components this base address is pointing to luminance (separate base adderesses for chrominances), for AC component rounds this base is used for current type
1:0	RO	0x0	reserved

**VDPU\_SWREG42**

Address: Operational Base + offset (0x00a8)

bi\_dir initial ref pic list register (0-2)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:25	RW	0x00	sw_binit_rlist_b2 Initial reference picture list for bi- direct backward picid 2
24:20	RW	0x00	sw_binit_rlist_f2 Initial reference picture list for bi- direct forward picid 2
19:15	RW	0x00	sw_binit_rlist_b1 Initial reference picture list for bi- direct backward picid 1
14:10	RW	0x00	sw_binit_rlist_f1 Initial reference picture list for bi- direct forward picid 1
9:5	RW	0x00	sw_binit_rlist_b0 Initial reference picture list for bi- direct backward picid 0
4:0	RW	0x00	sw_binit_rlist_f0 Initial reference picture list for bi- direct forward picid 0

**VDPU\_SWREG43**

Address: Operational Base + offset (0x00ac)

bi-dir initial ref pic list register (3-5)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:25	RW	0x00	sw_binit_rlist_b5 Initial reference picture list for bi- direct backward picid 5
24:20	RW	0x00	sw_binit_rlist_f5 Initial reference picture list for bi- direct forward picid 5
19:15	RW	0x00	sw_binit_rlist_b4 Initial reference picture list for bi- direct backward picid 4
14:10	RW	0x00	sw_binit_rlist_f4 Initial reference picture list for bi- direct forward picid 4
9:5	RW	0x00	sw_binit_rlist_b3 Initial reference picture list for bi- direct backward picid 3
4:0	RW	0x00	sw_binit_rlist_f3 Initial reference picture list for bi- direct forward picid 3

**VDPU\_SWREG44**

Address: Operational Base + offset (0x00b0)

bi-dir initial ref pic list register (6-8)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:25	RW	0x00	sw_binit_rlist_b8 Initial reference picture list for bi- direct backward picid 8
24:20	RW	0x00	sw_binit_rlist_f8 Initial reference picture list for bi- direct forward picid 8
19:15	RW	0x00	sw_binit_rlist_b7 Initial reference picture list for bi- direct backward picid 7
14:10	RW	0x00	sw_binit_rlist_f7 Initial reference picture list for bi- direct forward picid 7
9:5	RW	0x00	sw_binit_rlist_b6 Initial reference picture list for bi- direct backward picid 6
4:0	RW	0x00	sw_binit_rlist_f6 Initial reference picture list for bi- direct forward picid 6

**VDPU\_SWREG45**

Address: Operational Base + offset (0x00b4)

bi-dir initial ref pic list register (9- 11)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:25	RW	0x00	sw_binit_rlist_b11 Initial reference picture list for bi- direct backward picid 11
24:20	RW	0x00	sw_binit_rlist_f11 Initial reference picture list for bi- direct forward picid 11
19:15	RW	0x00	sw_binit_rlist_b10 Initial reference picture list for bi- direct backward picid 10
14:10	RW	0x00	sw_binit_rlist_f10 Initial reference picture list for bi- direct forward picid 10
9:5	RW	0x00	sw_binit_rlist_b9 Initial reference picture list for bi- direct backward picid 9
4:0	RW	0x00	sw_binit_rlist_f9 Initial reference picture list for bi- direct forward picid 9

**VDPU\_SWREG46**

Address: Operational Base + offset (0x00b8)

bi-dir initial ref pic list register (12- 14)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:25	RW	0x00	sw_binit_rlist_b14 Initial reference picture list for bi- direct backward picid 14
24:20	RW	0x00	sw_binit_rlist_f14 Initial reference picture list for bi- direct forward picid 14

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
19:15	RW	0x00	sw_binit_rlist_b13 Initial reference picture list for bi- direct backward picid 13
14:10	RW	0x00	sw_binit_rlist_f13 Initial reference picture list for bi- direct forward picid 13
9:5	RW	0x00	sw_binit_rlist_b12 Initial reference picture list for bi- direct backward picid 12
4:0	RW	0x00	sw_binit_rlist_f12 Initial reference picture list for bi- direct forward picid 12

**VDPU\_SWREG47**

Address: Operational Base + offset (0x00bc)  
bi-dir and P fwd initial ref pic list register (15 and P 0-3)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:25	RW	0x00	sw_pinit_rlist_f3 Initial reference picture list for P forward picid 3
24:20	RW	0x00	sw_pinit_rlist_f2 Initial reference picture list for P forward picid 2
19:15	RW	0x00	sw_pinit_rlist_f1 Initial reference picture list for P forward picid 1
14:10	RW	0x00	sw_pinit_rlist_f0 Initial reference picture list for P forward picid 0
9:5	RW	0x00	sw_binit_rlist_b15 Initial reference picture list for bi- direct backward picid 15
4:0	RW	0x00	sw_binit_rlist_f15 Initial reference picture list for bi- direct forward picid 15

**VDPU\_SWREG48**

Address: Operational Base + offset (0x00c0)  
Error concealment register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:23	RW	0x000	sw_startmb_x Start MB from SW for X dimension. Used in error concealment case [note]:the h264 decoder will use these bits.
22:15	RW	0x00	sw_startmb_y Start MB from SW for Y dimension. Used in error concealment case [note]:the h264 decoder will use these bits.
14:0	RO	0x0	reserved

**VDPU\_SWREG49**

Address: Operational Base + offset (0x00c4)  
Prediction filter tap register for H264, MPEG4, VC1

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:22	RW	0x000	sw_pred_bc_tap_0_0 Prediction filter set 0, tap 0 [note]:the h264 decoder will use these bits.
21:12	RW	0x000	sw_pred_bc_tap_0_1 Prediction filter set 0, tap 1 [note]:the h264 decoder will use these bits.
11:2	RW	0x000	sw_pred_bc_tap_0_2 Prediction filter set 0, tap 2 [note]:the h264 decoder will use these bits.
1:0	RO	0x0	reserved

**VDPU\_SWREG50**

Address: Operational Base + offset (0x00c8)

Synthesis configuration register decoder 0 (read only)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RO	0x1	SW_DEC_MPEG2_PROF Decoding format support, MPEG-2 / MPEG-1 '0' = not supported '1' = supported
30:29	RO	0x3	SW_DEC_VC1_PROF Decoding format support, VC-1 0 = not supported 1 = supported up to simple profile 2 = supported up to main profile 3 = supported up to advanced profile
28	RO	0x1	SW_DEC_JPEG_PROF Decoding format support, JPEG 0 = not supported 1 = supported
27:26	RO	0x2	SW_DEC_MPEG4_PROF Decoding format support, MPEG-4 / H.263 0 = not supported 1 = supported up to simple profile 2 = supported up to advanced simple profile
25:24	RO	0x3	SW_DEC_H264_PROF Decoding format support, H.264 0 = not supported 1 = supported up to baseline profile 2 = supported up to high profile labeled stream with restricted high profile tools [note]:the h264 decoder will use these bits.
23	RO	0x1	reserved

Bit	Attr	Reset Value	Description
22	RO	0x0	SW_DEC_PJEPAGE_EXIT Progressive JPEG support: '0' = Not supported '1' = supported
21	RO	0x1	SW_DEC_OBUFF_LEVEL Decoder output buffer level: '0' = 1 MB buffering is used '1' = 4 MB buffering is used [note]:the h264 decoder will use these bits.
20	RO	0x1	SW_REF_BUFF_EXIST [note]:the h264 decoder will use these bits.
19:16	RO	0x5	SW_DEC_BUS_STRD [note]:the h264 decoder will use these bits.
15:14	RO	0x1	SW_DEC_SYNTH_LAN [note]:the h264 decoder will use these bits.
13:12	RO	0x2	SW_DEC_BUS_WIDTH 0 = error 1 = 32 bit bus 2 = 64 bit bus 3 = 128 bit bus [note]:the h264 decoder will use these bits.
11	RO	0x1	reserved
10:0	RO	0x780	SW_DEC_MAX_OWIDTH Max configured decoder video resolution that can be decoded. Informed as width of the picture in pixels [note]:the h264 decoder will use these bits.

**VDPU\_SWREG51**

Address: Operational Base + offset (0x00cc)

Reference picture buffer control register

Bit	Attr	Reset Value	Description
31	RW	0x0	sw_refbu_e Refer picture buffer enable: '0' = refer picture buffer disabled '1' = refer picture buffer enabled. Valid if picture size is QVGA or more [note]:the h264 decoder will use these bits.
30:19	RW	0x000	sw_refbu_thr Reference buffer disable threshold value (cache miss amount). Used to buffer shut down (if more misses than allowed) [note]:the h264 decoder will use these bits.
18:14	RW	0x00	sw_refbu_picid The used reference picture ID for reference buffer usage [note]:the h264 decoder will use these bits.

Bit	Attr	Reset Value	Description
13	RW	0x0	sw_refbu_eval_e Enable for HW internal reference ID calculation. If given threshold level is reached by any picture_id after first MB row, that picture_id is used for reference buffer fill for rest of the picture [note]:the h264 decoder will use these bits.
12	RW	0x0	sw_refbu_fparmod_e Field parity mode enable. Used in rebufferd evaluation mode '0' = use the result field of the evaluation '1' = use the parity mode field [note]:the h264 decoder will use these bits.
11:9	RO	0x0	reserved
8:0	RW	0x000	sw_refbu_y_offset Y offset for rebufferd. This coordinate is used to compensate the global motion of the video for better buffer hit rate [note]:the h264 decoder will use these bits.

**VDPU\_SWREG52**

Address: Operational Base + offset (0x00d0)

Reference picture buffer information register 1 (read only)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	sw_refbu_hit_sum The sum of the rebufferd hits of the picture. Determined for each 8x8 luminance partition of the picture. The proceeding of the HW calculation can be read during HW decoding [note]:the h264 decoder will use these bits.
15:0	RW	0x0000	sw_refbu_intra_sum The sum of the luminance 8x8 intra partitions of the picture. The proceeding of the HW calculation can be read during HW decoding [note]:the h264 decoder will use these bits.

**VDPU\_SWREG53**

Address: Operational Base + offset (0x00d4)

Reference picture buffer information register 2 (read only)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21:0	RW	0x000000	sw_refbu_y_mv_sum The sum of the decoded motion vector y-components of the picture. The first luminance motion vector of each MB is used in calculation. Other motion vectors of the MB are discarded. Each motion vector is saturated between -256 - 255 before calculation. The proceeding of the HW calculation can be read during HW decoding [note]:the h264 decoder will use these bits.

**VDPU\_SWREG54**

Address: Operational Base + offset (0x00d8)

Synthesis configuration register decoder 1 (read only)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RO	0x1	SW_DEC_JPEG_EXTENS JPEG sampling support extension for 411 and 444 samplings and support for bigger max resolution than 16 Mpix (up to 67Mpixels): '0' = not supported '1' = supported
30	RO	0x1	SW_DEC_REFBU_ILACE Refbufferd support for interlaced content: '0' = not supported '1' = supported [note]:the h264 decoder will use these bits.
29	RO	0x1	reserved
28	RO	0x0	SW_REF_BUFF2_EXIST Reference picture buffer 2 usage: '0' = not supported '1' = reference buffer 2 is used [note]:the h264 decoder will use these bits.
27:26	RO	0x1	reserved
25	RO	0x0	SW_DEC_RTL_ROM ROM implementation type (If design includes ROMs) '0': ROMs are implemented from actual ROM units '1': ROMs are impelemted from RTL
24	RO	0x1	reserved
23	RO	0x1	reserved
22	RO	0x1	SW_DEC_AVG_PROF Decoding format support, AVS 0 = not supported 1 = supported
21:20	RO	0x1	SW_DEC_MVC_PROF Decoding format support, MVC 0 = not supported 1 = supported
19	RO	0x1	reserved
18:17	RO	0x1	SW_DEC_TILED_L Tiled mode support level 0 = not supported 1 = supported with 8x4 tile size 2,3 = reserved [note]:the h264 decoder will use these bits.
16:0	RO	0x0	reserved

**VDPU\_SWREG55**

Address: Operational Base + offset (0x00dc)

Reference picture buffer 2 / Advanced prefetch control register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RW	0x0	sw_refbu2_buf_e Refer picture buffer 2 enable: '0' = refer picture buffer disabled '1' = refer picture buffer enabled. Valid if picture size is QVGA or more (can be turned off by HW if threshold value reached) [note]:the h264 decoder will use these bits.
30:19	RW	0x000	sw_refbu2_thr Reference buffer disable threshold value (buffer miss amount). Used to buffer shut down (if more misses than allowed) [note]:the h264 decoder will use these bits.
18:14	RW	0x00	sw_refbu2_picid The used reference picture ID for reference buffer usage [note]:the h264 decoder will use these bits.
13:0	RW	0x0000	sw_apf_threshold Advanced prefetch threshold value. If current MB exceeds the threshold the advanced mode is not used. Value 0 disables threshold usage and advanced prefetch usage is restricted by internal memory limitation only [note]:the h264 decoder will use these bits.

**VDPU\_SWREG56**

Address: Operational Base + offset (0x00e0)

Reference buffer information register 3 (read only)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_refbu_top_sum The sum of the top partitions of the picture [note]:the h264 decoder will use these bits.
15:0	RW	0x0000	sw_refbu_bot_sum The sum of the bottom partitions of the picture [note]:the h264 decoder will use these bits.

**VDPU\_SWREG57\_INTRA\_INTER**

Address: Operational Base + offset (0x00e4)

intra\_dll3t,intra\_dblspeed,inter\_dblspeed,stream\_len\_hi

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:15	RO	0x0	reserved
14:8	RO	0x00	debug_service debug_service signals service_wr[2:0], service_rd[3:0]

Bit	Attr	Reset Value	Description
7	RW	0x0	sw_cache_en cache enable 1'b1: cache enable 1'b0: cache disable when sw_cache_en is 1'b1, sw_pref_sigchan should also be 1'b1
6	RW	0x0	sw_pref_sigchan prefetch single channel enable 1'b1: prefetch single channel enable
5	RW	0x0	sw_axiwr_sel axi write master select 1'b0: auto sel encoder axi signals and decoder axi signals 1'b1: sel decoder axi signals (it only use to set bu_dec_e to 1'b0 in the middle of a frame)
4	RW	0x0	sw_parallel_bus parallel_bus enable when it is set to 1'b1, the axi support read and write service parallel; when it is set to 1'b0, the axi only support read and write serial
3	RW	0x0	sw_intra_dbl3t sw_intra_dbl3t In chroma dc intra prediction, when this bit is enable, there will 3 cycle enhance for every block
2	RW	0x0	sw_intra_dblspeed intra double speed enable Intra double speed enable
1	RW	0x0	sw_inter_dblspeed inter double speed enable Inter double speed enable
0	RW	0x0	sw_stream_len_hi stream length high bit The extension bit of sw_stream_len

**VDPU\_SWREG57**

Address: Operational Base + offset (0x00e4)

intra\_dbl3t,intra\_dblspeed,inter\_dblspeed,stream\_len\_hi

Bit	Attr	Reset Value	Description
31	RW	0x0	fuse_dec_h264 1 = H.264 enabled
30	RW	0x0	fuse_dec_mpeg4 1 = MPEG-4/H.263 enabled
29	RW	0x0	fuse_dec_mpeg2 1 = MPEG-2/MPEG-1 enabled N
28	RW	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
27	RW	0x0	fuse_dec_jpeg Field0000 Abstract 1 = JPEG enabled
26	RW	0x0	reserved
25	RW	0x0	fuse_dec_vc1 1 = VC1 enabled
24	RW	0x0	fuse_dec_pjpeg 1 = Progressive JPEG enabled (Requires also JPEG to be enabled)
23	RW	0x0	reserved
22	RW	0x0	reserved
21	RW	0x0	reserved
20	RW	0x0	reserved
19	RW	0x0	reserved
18	RW	0x0	fuse_dec_mvc enabled (requires also H264 to be enabled)
17:16	RO	0x0	reserved
15	RW	0x0	fuse_dec_maxw_1920 1 = Max video width up to 1920 pixels enabled. Priority coded with priority 1.
14	RW	0x0	fuse_dec_maxw_1280 1 = Max video width up to 1280 pixels enabled. Priority coded with priority 2.
13	RW	0x0	fuse_dec_maxw_720 1 = Max video width up to 720 pixels enabled. Priority coded with priority 3.
12	RW	0x0	fuse_dec_maxw_352 1 = Max video width up to 352 pixels enabled. Priority coded with priority 4
11:8	RO	0x0	reserved
7	RW	0x0	fuse_dec_refbuffer 1 = reference buffer used
6:0	RO	0x0	reserved

**VDPU\_SWREG58**

Address: Operational Base + offset (0x00e8)

Decoder debug register 0 (read only)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RO	0x0	reserved
30	RO	0x0	debug_mv_req mvst_mv_req signal value
29	RO	0x0	debug_rlc_req prtr_res_y_req signal value
28	RO	0x0	debug_res_y_req prtr_res_y_req signal value

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
27	RO	0x0	debug_res_c_req prtr_res_c_req signal value
26	RO	0x0	debug_strm_da_e strm_da_e signal value
25	RO	0x0	debug_framerdy dfbu_framerdy signal value
24	RO	0x0	debug_filter_req dfbu_req_e signal value
23	RO	0x0	debug_referreq0 prbu_referreq0 signal value
22	RO	0x0	debug_referreq1 prbu_referreq1 signal value
21	RO	0x0	reserved
20:0	RO	0x000000	debug_dec_mb_count HW internal MB counter value

**VDPU\_SWREG59**

Address: Operational Base + offset (0x00ec)

H264 Chrominance 8 pixel interleaved data base

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_dec_ch8pix_base Base address for additional chrominance data format where chrominance is interleaved in group of 8 pixels. The usage is enabled by sw_ch_8pix_ilieav_e [note]:the h264 decoder will use these bits.
1:0	RO	0x0	reserved

**VDPU\_SWREG60**

Address: Operational Base + offset (0x00f0)

Interrupt register post-processor

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:14	RO	0x0	reserved
13	RW	0x0	sw_pp_bus_int Interrupt status bit bus. Error response from bus. In pipeline mode this bit is not used
12	RW	0x0	sw_pp_rdy_int Interrupt status bit pp. When this bit is high post processor has processed a picture in external mode. In pipeline mode this bit is not used.
11:9	RO	0x0	reserved
8	RW	0x0	sw_pp_irq Post-processor IRQ. SW will reset this after interrupt is handled. HINTpp is not used for pp if IRQ disable pp is high (sw_pp_irq_n_e = 1). In pipeline mode this bit is not used
7:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4	RW	0x0	sw_pp_irq_dis Post-processor IRQ disable. When high, there are no interrupts from HW concerning post processing. Polling must be used to see the interrupt
3:2	RO	0x0	reserved
1	RW	0x0	sw_pp_pipeline_e Decoder –post-processing pipeline enable: 0 = Post-processing is processing different picture than decoder or is disabled 1 = Post-processing is performed in pipeline with decoder
0	RW	0x0	sw_pp_e External mode post-processing enable. This bit will start the post-processing operation. Not to be used if PP is in pipeline with decoder (sw_pp_pipeline_e = 1). HW will reset this when picture is post-processed.

**VDPU\_SWREG61**

Address: Operational Base + offset (0x00f4)

Device configuration register post-processor

Bit	Attr	Reset Value	Description
31:24	RW	0x01	sw_pp_axi_rd_id Read ID used for AXI PP read services (if connected to AXI)
23:16	RW	0x01	sw_pp_axi_wr_id Write ID used for AXI PP write services (if connected to AXI)
15	RO	0x0	reserved
14	RW	0x0	sw_pp_scmd_dis AXI Single Command Multiple Data disable.
13	RW	0x0	sw_pp_in_a2_endsel Endian/swap select for Alpha blend input source 2: '0' = Use PP in endian/swap definitions (sw_pp_in_endian, sw_pp_in_swap) '1' = Use Ablend source 1 endian/swap definitions
12	RW	0x0	sw_pp_in_a1_swap32 Alpha blend source 1 input 32bit data swap (may be used for 64 bit environment): 0 = no swapping of 32 bit words 1 = 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled))
11	RW	0x0	sw_pp_in_a1_endian Alpha blend source 1 input data byte endian mode. 0 = Big endian (0-1-2-3 order) 1 = Little endian (3-2-1-0 order)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
10	RW	0x0	sw_pp_in_swap32_e PP input 32bit data swap (may be used for 64 bit environment): 0 = no swapping of 32 bit words 1 = 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled))
9	RW	0x0	sw_pp_data_disc_e PP data discard enable. Precise burst lengths are used with reading services. Extra data is discarded internally.
8	RW	0x1	sw_pp_clkgate_e PP dynamic clock gating enable: 1 = Clock is gated from PP structures that are not used 0 = Clock is running for all PP structures Note: Clock gating value can be changed only when PP is not enabled
7	RW	0x0	sw_pp_in_endian PP input picture byte endian mode. Used only if PP is in standalone mode. If PP is running pipelined with the decoder, this bit has no effect. 0 = Big endian (0-1-2-3 order) 1 = Little endian (3-2-1-0 order)
6	RW	0x0	sw_pp_out_endian PP output picture endian mode for YCbCr data or for any data if config value SW_PP_OEN_VERSION=1 0 = Big endian (0-1-2-3 order) 1 = Little endian (3-2-1-0 order) (NOTE: For SW_PP_OEN_VERSION=0 16 bit RGB data this bit works as pixel swapping bit. For 32 bit RGB this bit has no meaning)
5	RW	0x0	sw_pp_out_swap32_e PP output data word swap (may be used for 64 bit environment): 0 = no swapping of 32 bit words 1 = 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order (also little endian should be enabled))
4:0	RW	0x00	sw_pp_max_burst Maximum burst length for PP bus transactions. 1-16

**VDPU\_SWREG62**

Address: Operational Base + offset (0x00f8)

Deinterlace control register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RW	0x0	sw_deint_e De-interlace enable. Input data is in interlaced format and deinterlacing needs to be performed
30	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
29:16	RW	0x0000	sw_deint_threshold Threshold value used in deinterlacing
15	RW	0x0	sw_deint_blend_e Blend enable for de-interlacing
14:0	RW	0x0000	sw_deint_edge_det Edge detect value used for deinterlacing

**VDPU\_SWREG63**

Address: Operational Base + offset (0x00fc)

base address for reading post-processing input picture uminan

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_pp_in_lu_base Base address for post-processing input luminance picture. If PP input picture is fetched from fields this base address is used to point to topfield of the picture. Used in external mode only.
1:0	RO	0x0	reserved

**VDPU\_SWREG64**

Address: Operational Base + offset (0x0100)

Base address for reading post-processing input picture Ch/Ch

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_pp_in_cb_base Base address for post-processing input Cb picture or for both chrominances interleaved). If PP input picture is fetched from fields this base address is used to point to topfield of the picture. Used in external mode only
1:0	RO	0x0	reserved

**VDPU\_SWREG65**

Address: Operational Base + offset (0x0104)

Base address for reading post-processing input picture Cr

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_pp_in_cr_base Base address for post-processing input cr picture. Used in external mode only
1:0	RO	0x0	reserved

**VDPU\_SWREG66**

Address: Operational Base + offset (0x0108)

Base address for writing post-processed picture luminance/RGB

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_pp_out_lu_base Base address for post-processing output picture (luminance/YUYV/RGB).

**VDPU\_SWREG67**

Address: Operational Base + offset (0x010c)

Base address for writing post-processed picture Ch

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_pp_out_ch_base Base address for post-processing output chrominance picture (interleaved chrominance).

**VDPU\_SWREG68**

Address: Operational Base + offset (0x0110)

Register for contrast adjusting

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RW	0x00	sw_contrast_thr1 Threshold value 1, used with contrast adjusting
23:20	RO	0x0	reserved
19:10	RW	0x000	sw_contrast_off2 Offset value 2, used with contrast adjusting
9:0	RW	0x000	sw_contrast_off1 Offset value 1, used with contrast adjusting

**VDPU\_SWREG69**

Address: Operational Base + offset (0x0114)

Register for colour conversion and contrast adjusting

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RW	0x0	sw_pp_in_start_ch For YUYV 422 input format. Enable for start_with_chrominance. '0' = the order is Y0CbY0Cr or Y0CrY0Cb '1' = the order is CbY0CrY0 or CrY0CbY0
30	RW	0x0	sw_pp_in_cr_first For YUYV 422 input format and YCbCr 420 semiplanar format. Enable for Cr first (before Cb) '0' = the order is Y0CbY0Cr or CbY0CrY0 (if 420 semiplanar chrominance: CbCrCbCr) '1' = the order is Y0CrY0Cb or CrY0CbY0 (if 420 semiplanar chrominance: CrCbCrCb)
29	RW	0x0	sw_pp_out_start_ch For YUYV 422 output format. Enable for start_with_chrominance. '0' = the order is Y0CbY0Cr or Y0CrY0Cb '1' = the order is CbY0CrY0 or CrY0CbY0
28	RW	0x0	sw_pp_out_cr_first For YUYV 422 output format. Enable for Cr first (before Cb) '0' = the order is Y0CbY0Cr or CbY0CrY0 '1' = the order is Y0CrY0Cb or CrY0CbY0
27:18	RW	0x000	sw_color_coeffa2 Coefficient a2, used with Y pixel to calculate all color components

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
17:8	RW	0x000	sw_color_coeffa1 Coefficient a1, used with Y pixel to calculate all color components
7:0	RW	0x00	sw_contrast_thr2 Threshold value 2, used with contrast adjusting

**VDPU\_SWREG70**

Address: Operational Base + offset (0x0118)

Register for colour conversion 0

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:20	RW	0x000	sw_color_coeffd Coefficient d, used with Cb to calculate green component value
19:10	RW	0x000	sw_color_coeffc Coefficient c, used with Cr to calculate green component value
9:0	RW	0x000	sw_color_coeffb Coefficient b, used with Cr to calculate red component value

**VDPU\_SWREG71**

Address: Operational Base + offset (0x011c)

Register for colour conversion 1 + rotation mode

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:21	RW	0x000	sw_crop_startx Start coordinate x for the cropped area in macroblocks.
20:18	RW	0x0	sw_rotation_mode Rotation mode: 000 = rotation disabled 001 = rotate + 90 010 = rotate -90 011 = horizontal flip (mirror) 100 = vertical flip 101 = rotate 180
17:10	RW	0x00	sw_color_coefff Coefficient f, used with Y to adjust brightness
9:0	RW	0x000	sw_color_coeffe Coefficient e, used with Cb to calculate blue component value

**VDPU\_SWREG72**

Address: Operational Base + offset (0x0120)

PP input size and -cropping register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RW	0x00	sw_crop_starty Start coordinate y for the cropped area in macroblocks.
23	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
22:18	RW	0x00	sw_rangemap_coef_y Range map value for Y component (RANGE_MAPY+9 in VC-1 standard)
17	RO	0x0	reserved
16:9	RW	0x00	sw_pp_in_height PP input picture height in MBs. Can be cropped from a bigger input picture in external mode
8:0	RW	0x000	sw_pp_in_width PP input picture width in MBs. Can be cropped from a bigger input picture in external mode

**VDPU\_SWREG73**

Address: Operational Base + offset (0x0124)

PP input picture base address for Y bottom field

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_pp_bot_yin_base PP input Y base for bottom field
1:0	RO	0x0	reserved

**VDPU\_SWREG74**

Address: Operational Base + offset (0x0128)

PP input picture base for Ch bottom field

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_pp_bot_cin_base PP input C base for bottom field (mixed chrominance)
1:0	RO	0x0	reserved

**VDPU\_SWREG79**

Address: Operational Base + offset (0x013c)

Scaling ratio register 1 &amp; padding for B

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RW	0x0	sw_rangemap_y_e Range map enable for Y component (RANGE_MAPY_FLAG in VC-1 standard). For VC1 main profile this bit is used as range expansion enable
30	RW	0x0	sw_rangemap_c_e Range map enable for chrominance component RANGE_MAPUV_FLAG in VC-1 standard)
29	RW	0x0	sw_ycbcr_range Defines the YCbCr range in RGB conversion: 0 = 16 --> 235 for Y, 16 --> 240 for Chrominance 1 = 0 --> 255 for all components

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
28	RW	0x0	sw_rgb_pix_in32 RGB pixel amount/ 32 bit word 0 = 1 RGB pixel/32 bit 1 = 2 RGB pixels/32 bit
27:23	RW	0x00	sw_rgb_r_padd Amount of ones that will be padded in front of the R-component
22:18	RW	0x00	sw_rgb_g_padd Amount of ones that will be padded in front of the G-component
17:0	RW	0x00000	sw_scale_wratio Scaling ratio for width (outputw-1/inputw-1)

**VDPU\_SWREG80**

Address: Operational Base + offset (0x0140)

Scaling register 0 ratio &amp; padding for R and G

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RO	0x0	reserved
30	RW	0x0	sw_pp_fast_scale_e 0 = fast downscaling is not enabled 1 = fast downscaling is enabled. The quality of the picture is decreased but performance is improved
29:27	RW	0x0	sw_pp_in_struct PP input data picture structure: 0 = Top field / progressive frame structure: Read input data from top field base address /frame base address and read every line 1 = Bottom field structure: Read input data from bottom field base address and read every line. 2 = Interlaced field structure: Read input data from both top and bottom field base address and take every line from each field. 3 = Interlaced frame structure: Read input data from both top and bottom field base address and take every second line from each field. 4 = Ripped top field structure: Read input data from top field base address and read every second line. 5 = Ripped bottom field structure: Read input data from bottom field base address and read every second line
26:25	RW	0x0	sw_hor_scale_mode Horizontal scaling mode: 00 = Off 01 = Upscale 10 = Downscale

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
24:23	RW	0x0	sw_ver_scale_mode Vertical scaling mode: 00 = Off 01 = Upscale 10 = Downscale
22:18	RW	0x00	sw_rgb_b_padd Amount of ones that will be padded in front of the B-component
17:0	RW	0x00000	sw_scale_hratio Scaling ratio for height (outpuh-1/inpuh-1)

**VDPU\_SWREG81**

Address: Operational Base + offset (0x0144)

Scaling ratio register 2

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_wscale_invra Inverse scaling ratio for width, or ch (inputw-1 / outputw-1)
15:0	RW	0x0000	sw_hscale_invra Inverse scaling ratio for height or cv (inpuh-1 / outpuh-1)

**VDPU\_SWREG82**

Address: Operational Base + offset (0x0148)

Rmask register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_r_mask Bit mask for R component (and alpha channel)

**VDPU\_SWREG83**

Address: Operational Base + offset (0x014c)

Gmask register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_g_mask Bit mask for G component (and alpha channel)

**VDPU\_SWREG84**

Address: Operational Base + offset (0x0150)

Bmask register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_b_mask Bit mask for B component (and alpha channel)

**VDPU\_SWREG85**

Address: Operational Base + offset (0x0154)

Post-processor control register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RW	0x0	<p>sw_pp_in_format  PP input picture data format  0 = YUYV 4:2:2 interleaved (supported only in external mode)  1 = YCbCr 4:2:0 Semi-planar in linear raster-scan format  2 = YCbCr 4:2:0 planar (supported only in external mode)  3 = YCbCr 4:0:0 (supported only in pipelined mode)  4 = YCbCr 4:2:2 Semi-planar (supported only in pipelined mode)  5 = YCbCr 4:2:0 Semi-planar in tiled format (supported only in external mode (8170 decoder only))  6 = YCbCr 4:4:0 Semi-planar (supported only in pipelined mode, possible for jpeg only)  7 = Escape pp input data format. Defined in swreg86</p>
28:26	RW	0x0	<p>sw_pp_out_format  PP output picture data format:  0 = RGB  1 = YCbCr 4:2:0 planar (Not supported)  2 = YCbCr 4:2:2 planar (Not supported)  3 = YUYV 4:2:2 interleaved  4 = YCbCr 4:4:4 planar (Not supported)  5 = YCh 4:2:0 chrominance interleaved  6 = YCh 4:2:2 (Not supported)  7 = YCh 4:4:4 (Not supported)</p>
25:15	RW	0x000	<p>sw_pp_out_height  Scaled picture height in pixels (Must be dividable by 2 or by any if Pixel Accurate PP output configuration is enabled) Max scaled picture height is 1920 pixels or maximum three times the input source height minus 8 pixels</p>
14:4	RW	0x000	<p>sw_pp_out_width  Scaled picture width in pixels. Must be dividable by 8 or by any if Pixel Accurate PP output configuration is enabled. Max scaled picture width is 1920 pixels or maximum three times the input source width minus 8 pixels</p>
3	RW	0x0	<p>sw_pp_out_tiled_e  Tiled mode enable for PP output. Can be used only for YCbYCr 422 output format. Can be used only if correpongind configuration supports this feature. Tile size is 4x4 pixels.</p>
2	RW	0x0	<p>sw_pp_out_swap16_e  PP output swap 16 swaps 16 bit halves inside of 32 bit word. Can be used for 16 bit RGB to change pixel orders but is valid also for any output format  NOTE: requires that configuration of SW_PPD_OEN_VERSION=1</p>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
1	RW	0x0	sw_pp_crop8_r_e PP input picture width is not 16 pixels multiple. Only 8 pixels of the most right MB of the unrotated input picture is used for PP input.
0	RW	0x0	sw_pp_crop8_d_e PP input picture height is not 16 pixels multiple. Only 8 pixel rows of the most down MB of the unrotated input picture is used for PP input.

**VDPU\_SWREG86**

Address: Operational Base + offset (0x0158)

Mask 1 start coordinate register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RW	0x0	sw_pp_in_format_es Escape PP in format. Used if sw_pp_in_format is defined to 7: 0 0 = YCbCr 4:4:4 1 = YCbCr 4:1:1
28	RO	0x0	reserved
27:23	RW	0x00	sw_rangemap_coef_c Range map value for chrominance component (RANGE_MAPUV+9 in VC-1 standard)
22	RW	0x0	sw_mask1_ablend_e Mask 1 alpha blending enable. Instead of masking the output picture the alpha blending is performed. Alpha blending source can be found from alpha blend 1 base address. Alpha blending can be enabled only for RGB/ YUYV 422 data.
21:11	RW	0x000	sw_mask1_starty Vertical start pixel for mask area 1. Defines the y coordinate. Coordinate 0,0 means the up-left corner in PP output luminance picture. See Table 47 for restrictions
10:0	RW	0x000	sw_mask1_startix Horizontal start pixel for mask area 1. Defines the x coordinate. Coordinate 0,0 means the up-left corner in PP output luminance picture. See Table 47 for restrictions

**VDPU\_SWREG87**

Address: Operational Base + offset (0x015c)

Mask 2 start coordinate register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:23	RO	0x0	reserved
22	RW	0x0	sw_mask2_ablend_e Mask 2 alpha blending enable. Instead of masking the output picture the alpha blending is performed. Alpha blending source can be found from alpha blend 2 base address. Alpha blending can be enabled only for RGB/YUYV 422 data.

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
21:11	RW	0x000	sw_mask_starty Vertical start pixel for mask area 2. Defines the y coordinate. Coordinate 0,0 means the up-left corner in PP output Y picture. See Table 47 for restrictions
10:0	RW	0x000	sw_mask2_startx Horizontal start pixel for mask area 2. Defines the x coordinate. Coordinate 0,0 means the up-left corner in PP output Y picture. See Table 47 for restrictions

**VDPU\_SWREG88**

Address: Operational Base + offset (0x0160)

Mask 1 size and PP original width register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:23	RW	0x000	sw_ext_orig_width PP input picture original width in macro blocks.
22	RW	0x0	sw_mask1_e Mask 1 enable. If mask 1 is used this bit is high
21:11	RW	0x000	sw_mask1_endy Mask 1 end coordinate y in pixels (inside of PPD output picture). Range must be between [Mask1StartCoordinateY, ScaledHeight].
10:0	RW	0x000	sw_mask1_endx Mask 1 end coordinate x in pixels (inside of PPD output picture). Range must be between [Mask1StartCoordinateX, ScaledWidth]

**VDPU\_SWREG89**

Address: Operational Base + offset (0x0164)

Mask 2 size register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:23	RO	0x0	reserved
22	RW	0x0	sw_mask2_e Mask 2 enable. If mask 1 is used this bit is high
21:11	RW	0x000	sw_mask2_endy Mask 2 end coordinate y in pixels (inside of PP output picture). Range must be between [Mask2StartCoordinateY, ScaledHeight].
10:0	RW	0x000	sw_mask2_endx Mask 2 end coordinate x in pixels (inside of PP output picture). Range must be between [Mask2StartCoordinateX, ScaledWidth].

**VDPU\_SWREG90**

Address: Operational Base + offset (0x0168)

PiP register 0

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RW	0x0	sw_right_cross_e Right side overcross enable. 0 = No right side overcross, 1 = right side overcross
28	RW	0x0	sw_left_cross_e Left side overcross enable. 0 = No left side overcross, 1 = left side overcross
27	RW	0x0	sw_up_cross_e Upward overcross enable. 0 = No upward overcross, 1 = upward overcross
26	RW	0x0	sw_down_cross_e Downward overcross enable. 0 = No downward overcross, 1 = downward overcross
25:15	RW	0x000	sw_up_cross Amount of upward overcross (vertical pixels outside of display from the upper side). Range must be between [0, ScaledHeight].
14:11	RO	0x0	reserved
10:0	RW	0x000	sw_down_cross Amount of downward overcross (vertical pixels outside of display from the down side). Range must be between [0, ScaledHeight].

**VDPU\_SWREG91**

Address: Operational Base + offset (0x016c)

PiP register 1 and dithering control

Bit	Attr	Reset Value	Description
31:30	RW	0x0	sw_dither_select_r Dithering control for R channel: 00 = dithering disabled 01 = use four-bit dither matrix 10 = use five-bit dither matrix 11 = use six-bit dither matrix
29:28	RW	0x0	sw_dither_select_g Dithering control for G channel: 00 = dithering disabled 01 = use four-bit dither matrix 10 = use five-bit dither matrix 11 = use six-bit dither matrix
27:26	RW	0x0	sw_dither_select_b Dithering control for B channel: 00 = dithering disabled 01 = use four-bit dither matrix 10 = use five-bit dither matrix 11 = use six-bit dither matrix

Bit	Attr	Reset Value	Description
25:24	RO	0x0	reserved
23:22	RW	0x0	<p>sw_pp_tiled_mode Input data is in tiled mode (at the moment valid only for YCbCr 420 data, pipeline or external mode): 0 = Tiled mode not used 1 = Tiled mode enabled for 8x4 sized tiles 2,3 = reserved</p>
21:11	RW	0x000	<p>sw_right_cross Amount of right side overcross (Horizontal pixels outside of display from the right side). Range must be between [0, ScaledWidth].</p>
10:0	RW	0x000	<p>sw_left_cross Amount of left side overcross (Horizontal pixels outside of display from the left side). Range must be between [0, ScaledWidth].</p>

**VDPU\_SWREG92**

Address: Operational Base + offset (0x0170)

Display width and PP input size extension register

Bit	Attr	Reset Value	Description
31:29	RW	0x0	<p>sw_pp_in_h_ext Extended PP input height. Used with JPEG</p>
28:26	RW	0x0	<p>sw_pp_in_w_ext Extended PP input width. Used with JPEG</p>
25:23	RW	0x0	<p>sw_crop_start_x_ext Extended PP input crop start coordinate x. Used with JPEG</p>
22:20	RW	0x0	<p>sw_crop_start_y_ext Extended PP input crop start coordinate y. Used with JPEG</p>
19:12	RO	0x0	reserved
11:0	RW	0x000	<p>sw_display_width Width of the display in pixels. Max HDTV (1920)</p>

**VDPU\_SWREG93**

Address: Operational Base + offset (0x0174)

Display width and PP input size extension register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>sw_abledn1_base Base address for alpha blending input 1 (if mask1 is used in alpha blending mode). Format of data is 24 bit RGB/ YCbCr and endian/swap -mode is as in PP input. Amount of data is informed with mask 1 size or with ablend1_scanline if ablend cropping is supported in configuration.</p>

**VDPU\_SWREG94**

Address: Operational Base + offset (0x0178)  
 Base address for alpha blend 2 gui component

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_ablend2_base Base address for alpha blending input 2 (if mask2 is used in alpha blending mode). Format of data is 24 bit RGB/ YCbCr and endian/swap -mode is as in PP input. Amount of data is informed with mask 2 size or with ablend2_scanline if ablend cropping is supported in configuration.

**VDPU\_SWREG95**

Address: Operational Base + offset (0x017c)  
 Base address for alpha blend 2 gui component

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:26	RO	0x0	reserved
25:13	RW	0x0000	sw_ablend2_scan Scanline width in pixels for Ablend 2. Usage enabled if corresponding configuration bit is enabled
12:0	RW	0x0000	sw_ablend1_scan Scanline width in pixels for Ablend 1. Usage enabled if corresponding configuration bit is enabled

**VDPU\_SWREG98**

Address: Operational Base + offset (0x0188)  
 PP output width/height extension

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RW	0x0	sw_pp_out_h_ext sw_pp_out_h_ext PP output heightextension
0	RW	0x0	sw_pp_out_w_ext sw_pp_out_w_ext PP output widthextension

**VDPU\_SWREG99**

Address: Operational Base + offset (0x018c)  
 PP fuse register (read only)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RO	0x1	fuse_pp_pp 1 = PP enabled
30	RO	0x1	fuse_pp_deint 1 = Deinterlacing enabled
29	RO	0x1	fuse_pp_ablend 1 = Alpha Blending enabled
28:16	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
15	RO	0x1	fuse_pp_maxw_1920 1 = Max PP output width up to 1920 pixels enabled. Priority coded with priority 1
14	RO	0x1	fuse_pp_maxw_1280 1 = Max PP output width up to 1280 pixels enabled. Priority coded with priority 2
13	RO	0x1	fuse_pp_maxw_720 1 = Max PP output width up to 720 pixels enabled. Priority coded with priority 3
12	RO	0x1	fuse_pp_maxw_352 1 = Max PP output width up to 352 pixels enabled. Priority coded with priority 4
11:0	RO	0x0	reserved

**VDPU\_SWREG100**

Address: Operational Base + offset (0x0190)

Synthesis configuration register post-processor (read only)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RO	0x1	SW_ABLEND_CROP_E Alpha blending support for input cropping: '0' : Not supported. External memory must include the exact image of the area being alpha blended '1' : Supported. External memory can include a picture from blended area can be cropped. Requires usage of swreg95
30	RO	0x1	SW_PPD_PIXAC_E Pixel Accurate PP output mode exists: '0' = PIP, Scaling and masks can be adjusted by steps of 8 pixels (width) or 2 pixels (height) '1' = PIP, Scaling and masks can be adjusted by steps of 1 pixel for RGB and 2 pixels for subsampled chroma formats (by using bus specific write strobe functionality)
29	RO	0x1	SW_PPD_TILED_EXIST PP output YCbYCr 422 tiled support (4x4 pixel tiles) '0' = Not supported '1' = Supported
28	RO	0x1	SW_PPD_DITH_EXIST Dithering exists: '0' = no '1' = yes
27:26	RO	0x3	SW_PPD_SCALE_LEVEL Scaling support: 00 = No scaling 01 = Scaling with lo perfomance architecture 10 = Scaling with high performance architecture 11 = Scaling with high performance architecture + fast

Bit	Attr	Reset Value	Description
25	RO	0x1	SW_PPD_DEINT_EXIST De-interlacing exists: '0' = no '1' = yes
24	RO	0x1	SW_PPD_BLEND_EXIST Alpha blending exists: '0' = no '1' = yes
23	RO	0x1	SW_PPD_IBUFF_LEVEL PP input buffering level: '0' = 1 MB input buffering is used '1' = 4 MB input buffering is used
22:19	RO	0x0	reserved
18	RO	0x1	SW_PPD_OEN_VERSION PP output endian version: '0' = Endian mode supported for other than RGB '1' = Endian mode supported for any output format
17	RO	0x1	SW_PPD_OBUFF_LEVEL PP output buffering level: '0' = 1 unit output buffering is used '1' = 4 unit output buffering is used
16	RO	0x1	SW_PPD_PP_EXIST PPD exists: '0' = no '1' = yes
15:14	RO	0x1	SW_PPD_IN_TILED_L PPD input tiled mode support level 0 = not supported 1 = 8x4 tile size supported
13:11	RO	0x0	reserved
10:0	RO	0x780	SW_PPD_MAX_OWIDTH Max supported PP output width in pixels

**VDPU\_SWREG101**

Address: Operational Base + offset (0x0194)  
soft reset signals

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	sw_soft_reset softreset pulse signal softreset pulse signal write to 1'b1, valid; write to 1'b0, invalid;

**VDPU\_SWREG102**

Address: Operational Base + offset (0x0198)

vpu performance cycle

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	vpu_work_cycle vpu working cycle number write initial/reset value in the begin of frame start,then will auto count base this value.

**VDPU\_SWREG103**

Address: Operational Base + offset (0x019c)

AXI DDR READ DATA NUM

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_axi_ddr_rdata axi ddr rdata num axi ddr rdata num, the unit is byte

**VDPU\_SWREG104**

Address: Operational Base + offset (0x01a0)

Register0000 Abstract

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:1	RO	0x0	reserved
0	RW	0x0	sw_axi_ddr_wdata vdpu write data byte num vdpu write data byte num

**VDPU\_SWREG105**

Address: Operational Base + offset (0x01a4)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:20	RO	0x0	reserved
19:16	RW	0x0	mon_sig_sel1 monitor signal selected for cnt1 select the counter be used for which to calculate cycle num 4'b0000: don't work 4'b0001:mv buffer hold back stream decode working cycles 4'b0010:the output fifo of cabac keep full cycles 4'b0011:the Code stream parsing block working cycles 4'b0100:scd block can't write data to scd buffer cycles 4'b0101:The speed of reconsititution and interpolation fast than reference frames feach cycles 4'b0110:The speed of reconsititution and interpolation slow than reference frames feach cycles 4'b0111:the cycles filter block hold back pred block 4'b1000:the cycles of pred block waiting for Residual data 4'b1001:the cycles of bus Related modules working
15:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x0	<p>mon_sig_sel0 monitor signal selected for cnt0 select the counter be used for which to calculate cycle num 4'b0000: don't work 4'b0001:mv buffer hold back stream decode working cycles 4'b0010:the output fifo of cabac keep full cycles 4'b0011:the Code stream parsing block working cycles 4'b0100:scd block can't write data to scd buffer cycles 4'b0101:The speed of reconsititution and interpolation fast than reference frames feach cycles 4'b0110:The speed of reconsititution and interpolation slow than reference frames feach cycles 4'b0111:the cycles filter block hold back pred block 4'b1000:the cycles of pred block waiting for Residual data 4'b1001:the cycles of bus Related modules working</p>

**VDPU\_SWREG106**

Address: Operational Base + offset (0x01a8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>performance_mon_cnt0 the counter for the selected signal valid cycles whic describe in swreg105[3:0] write initial/reset value</p>

**VDPU\_SWREG107**

Address: Operational Base + offset (0x01ac)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>performance_mon_cnt1 Field0000 Abstract the counter for the selected signal valid cycles whic describe in swreg105[19:16] write initial/reset value</p>

**VCODEC\_MMU\_DTE\_ADDR**

Address: Operational Base + offset (0x0000)

MMU current page Table address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>MMU_DTE_ADDR MMU_DTE_ADDR MMU current page Table address</p>

**VCODEC\_MMU\_STATUS**

Address: Operational Base + offset (0x0004)

MMU status register

Bit	Attr	Reset Value	Description

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:11	RO	0x0	reserved
10:6	RO	0x00	PAGEFAULT_BUS_ID page fault bus id Index of master responsible for last page fault
5	RO	0x0	PAGE_FAULT_IS_WRITE page fault access The direction of access for last page fault: 0 = Read 1 = Write
4	RO	0x1	REPLAY_BUFFER_EMPTY replay buffer empty status 1'b1: The MMU replay buffer is empty
3	RO	0x1	MMU_IDLE mmu idle status The MMU is idle when accesses are being translated and there are no unfinished translated accesses. 1'b1: MMU is idle
2	RO	0x0	STAIL_ACTIVE stall active status MMU stall mode currently enabled. The mode is enabled by command 1'b1: MMU is in stall active status
1	RO	0x0	PAGE_FAULT_ACTIVE page fault active status MMU page fault mode currently enabled . The mode is enabled by command. 1'b1: page fault is active
0	RO	0x0	PAGING_ENABLED Paging enabled status 1'b0: paging is disabled 1'b1: Paging is enabled

**VCODEC\_MMU\_COMMAND**

Address: Operational Base + offset (0x0008)

MMU command register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:3	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
2:0	WO	0x0	MMU_CMD mmu cmd MMU_CMD. This can be: 0: MMU_ENABLE_PAGING 1: MMU_DISABLE_PAGING 2: MMU_ENABLE_STALL 3: MMU_DISABLE_STALL 4: MMU_ZAP_CACHE 5: MMU_PAGE_FAULT_DONE 6: MMU_FORCE_RESET

**VCODEC\_MMU\_PAGE\_FAULT\_ADDR**

Address: Operational Base + offset (0x000c)

MMU logical address of last page fault

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x00000000	PAGE_FAULT_ADDR page fault addr address of last page fault

**VCODEC\_MMU\_ZAP\_ONE\_LINE**

Address: Operational Base + offset (0x0010)

MMU Zap cache line register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	WO	0x00000000	MMU_ZAP_ONE_LINE zap one line address to be invalidated from the page table cache

**VCODEC\_MMU\_INT\_RAWSTAT**

Address: Operational Base + offset (0x0014)

MMU raw interrupt status register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RW	0x0	READ_BUS_ERROR read bus error read bus error status
0	RW	0x0	PAGEFAULT page fault page fault status

**VCODEC\_MMU\_INT\_CLEAR**

Address: Operational Base + offset (0x0018)

MMU raw interrupt status register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
1	WO	0x0	READ_BUS_ERROR read bus error write 1 to clear read bus error
0	WO	0x0	PAGE_FAULT page fault clear write 1 to page fault clear

**VCODEC\_MMU\_INT\_MASK**

Address: Operational Base + offset (0x001c)

MMU raw interrupt status register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RW	0x0	READ_BUS_ERROR read bus error enable the read bus interrupt source when this bit is set to 1'b1
0	RW	0x0	PAGE_FAULT page fault mask enable the page fault interrupt source when this bit is set to 1'b1

**VCODEC\_MMU\_INT\_STATUS**

Address: Operational Base + offset (0x0020)

MMU raw interrupt status register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RO	0x0	READ_BUS_ERROR read bus error status 1'b1:read bus error status
0	RO	0x0	PAGE_FAULT page fault status 1'b1:page fault

**VCODEC\_MMU\_AUTO\_GATING**

Address: Operational Base + offset (0x0024)

mmu auto gating

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:1	RO	0x0	reserved
0	RW	0x1	mmu_auto_clkgating mmu_auto_clkgating when it is 1'b1, the mmu will auto gating it self

**pref\_cache\_VERSION**

Address: Operational Base + offset (0x0000)

VERSION register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RO	0xcac2	PRODUCT_ID Field0000 Abstract Field0000 Description
15:8	RO	0x01	VERSION_MAJOR Field0000 Abstract Field0000 Description
7:0	RO	0x01	VERSION_MINOR Field0000 Abstract Field0000 Description

**pref\_cache\_SIZE**

Address: Operational Base + offset (0x0004)

L2 cache SIZE

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x06	External_bus_width Field0000 Abstract Log2 external bus width in bits
23:16	RO	0x11	CACHE_SIZE Field0000 Abstract Log2 cache size in bytes
15:8	RO	0x02	ASSOCIATIVITY Field0000 Abstract Log2 associativity
7:0	RO	0x06	LINE_SIZE Field0000 Abstract Log2 line size in bytes

**pref\_cache\_STATUS**

Address: Operational Base + offset (0x0008)

Status register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RO	0x0	DATA_BUSY Field0000 Abstract set when the cache is busy handling data
0	RO	0x0	CMD_BUSY Field0000 Abstract set when the cache is busy handling commands

**pref\_cache\_COMMAND**

Address: Operational Base + offset (0x0010)

Command setting register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:4	RW	0x0	sw_addrb_sel Field0000 Abstract 2'b00:to sel b[14:6] 2'b01:to sel b[15:9], b[7:6] 2'b10:to sel b[16:10], b[7:6] 2'b11:to sel b[17:11], b[7:6]
3	RO	0x0	reserved
2:0	WO	0x0	COMMAND Field0000 Abstract The possible command is 1 = Clear entire cache

**pref\_cache\_CLEAR\_PAGE**

Address: Operational Base + offset (0x0014)

clear page register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	CLEAR_PAGE Field0000 Abstract writing an address, invalidates all lines in that page from the cache

**pref\_cache\_MAX\_READS**

Address: Operational Base + offset (0x0018)

maximum read register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x1c	MAX_READS Field0000 Abstract Limit the number of outstanding read transactions to this amount

**pref\_cache\_PERFCNT\_SRC0**

Address: Operational Base + offset (0x0020)

performance counter 0 source register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x0	<p>PERFCNT_SRC0 Field0000 Abstract This register holds all the possible source values for Performance Counter 0</p> <p>0: total clock cycles 1: active clock cycles 2: read transactions, master 3: word reads, master 4: read transactions, slave 5: word reads, slave 6: read hit, slave 7: read misses, slave 8: read invalidates, slave 9: cacheable read transactions, slave 10: bad hit nmber, slave</p>

**pref\_cache\_PERFCNT\_VAL0**

Address: Operational Base + offset (0x0024)

performance counter 0 value register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>PERFCNT_VAL0 Field0000 Abstract Performance counter 0 value</p>

**pref\_cache\_PERFCNT\_SRC1**

Address: Operational Base + offset (0x0028)

performance counter 0 source register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	<p>PERFCNT_SRC1 Field0000 Abstract This register holds all the possible source values for Performance Counter 1</p> <p>0: total clock cycles 1: active clock cycles 2: read transactions, master 3: word reads, master 4: read transactions, slave 5: word reads, slave 6: read hit, slave 7: read misses, slave 8: read invalidates, slave 9: cacheable read transactions, slave 10: bad hit nmber, slave</p>

**pref\_cache\_PERFCNT\_VAL1**

Address: Operational Base + offset (0x002c)

performance counter 1 value register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	PERFCNT_VAL1 Field0000 Abstract Performance counter 1 value

**4.5.4 VDPU Registers Summary**

<b>Name</b>	<b>Offset</b>	<b>Size</b>	<b>Reset Value</b>	<b>Description</b>
VDPU_SWREG0_NEW_VERSION	0x0000	W	0x03680000	ID register(read only)
VDPU_SWREG0	0x0000	W	0x00000000	Register0000 Abstract
VDPU_SWREG1	0x0004	W	0x00000000	interrupt register decoder
VDPU_SWREG2	0x0008	W	0x01000400	device configuration register decoder
VDPU_SWREG3	0x000c	W	0x00000001	Device control register 0(deemode, picture type etc)
VDPU_SWREG4_H264	0x0010	W	0x00000000	decoder control register 1(picture parameters)
VDPU_SWREG4	0x0010	W	0x00000000	decoder control register 1(picture parameters)
VDPU_SWREG5	0x0014	W	0x00000000	decoder control register2 (stream decoding table selects)
VDPU_SWREG5_H264	0x0014	W	0x00000000	decoder control register2 (stream decoding table selects)
VDPU_SWREG6	0x0018	W	0x00000000	decoder control register 3( stream buffer information)
VDPU_SWREG7	0x001c	W	0x00000000	decoder control register 4(H264)
VDPU_SWREG8	0x0020	W	0x00000000	decoder control register 5
VDPU_SWREG9	0x0024	W	0x00000000	decoder control register 6
VDPU_SREG10_H264_RLC	0x0028	W	0x00000000	Base address for differential motion vector base address
VDPU_SREG10_H264	0x0028	W	0x00000000	Base address for differential motion vector base address
VDPU_SWREG11_H264_RLC	0x002c	W	0x00000000	decoder control register 7
VDPU_SWREG11_H264	0x002c	W	0x00000000	decoder control register 7
VDPU_SWREG12	0x0030	W	0x00000000	Base address for RLC data (RLC) / stream start address/decoded
VDPU_SWREG13	0x0034	W	0x00000000	Base address for decoded picture / base address for JPEG deco
VDPU_SWREG14	0x0038	W	0x00000000	Base address for reference picture index 0 / base address for J
VDPU_SWREG15_JPEG_ROI	0x003c	W	0x00000000	JPEG roi control
VDPU_SWREG15	0x003c	W	0x00000000	Base address for reference picture index 1 / JPEG control
VDPU_SWREG16	0x0040	W	0x00000000	base address for reference picture index 2 / List of VLC code len
VDPU_SWREG17	0x0044	W	0x00000000	Base address for reference picture index 3 / List of VLC code le
VDPU_SWREG18	0x0048	W	0x00000000	Base address for reference picture index 4 /MPE

Name	Offset	Size	Reset Value	Description
VDPU_SWREG19	0x004c	W	0x00000000	Base address for reference picture index 5
VDPU_SWREG20	0x0050	W	0x00000000	Base address for reference picture index 6
VDPU_SWREG21	0x0054	W	0x00000000	Base address for reference picture index 7
VDPU_SWREG22	0x0058	W	0x00000000	Base address for reference picture index 8
VDPU_SWREG23	0x005c	W	0x00000000	Base address for reference picture index 9
VDPU_SWREG24	0x0060	W	0x00000000	Base address for reference picture index 10
VDPU_SWREG25	0x0064	W	0x00000000	Base address for reference picture index 11
VDPU_SWREG26	0x0068	W	0x00000000	Base address for reference picture index 12
VDPU_SWREG27	0x006c	W	0x00000000	Base address for reference picture index 13
VDPU_SWREG28	0x0070	W	0x00000000	Base address for reference picture index 14
VDPU_SWREG29	0x0074	W	0x00000000	Base address for reference picture index 15
VDPU_SWREG30	0x0078	W	0x00000000	Reference picture numbers for index 0 and 1 (H264 VLC)
VDPU_SWREG31	0x007c	W	0x00000000	Reference picture numbers for index 2 and 3 (H264 VLC) /
VDPU_SWREG32	0x0080	W	0x00000000	Reference picture numbers for index 4 and 5 (H264 VLC)
VDPU_SWREG33	0x0084	W	0x00000000	Reference picture numbers for index 6 and 7 (H264 VLC)
VDPU_SWREG34	0x0088	W	0x00000000	Reference picture numbers for index 8 and 9 (H264 VLC)
VDPU_SWREG35_JPEG_ROI	0x008c	W	0x00000000	JPEG roi offset/dc base address
VDPU_SWREG35	0x008c	W	0x00000000	Reference picture numbers for index 10 and 11 (H264 VLC)
VDPU_SWREG36	0x0090	W	0x00000000	Reference picture numbers for index 12 and 13 (H264 VLC)
VDPU_SWREG36_JPEG_ROI	0x0090	W	0x00000000	JPEG roi offset/dc length
VDPU_SWREG37	0x0094	W	0x00000000	Reference picture numbers for index 14 and 15 (H264 VLC)
VDPU_SWREG38	0x0098	W	0x00000000	Reference picture long term flags (H264 VLC) prediction filt
VDPU_SWREG38_H264	0x0098	W	0x00000000	Reference picture numbers for index 12 and 13 (H264 VLC)
VDPU_SWREG39	0x009c	W	0x00000000	Reference picture valid flags (H264 VLC) prediction filter ta
VDPU_SWREG39_H264	0x009c	W	0x00000000	Reference picture numbers for index 12 and 13 (H264 VLC)
VDPU_SWREG40	0x00a0	W	0x00000000	Base address for standard dependent tables
VDPU_SWREG41	0x00a4	W	0x00000000	Base address for direct mode motion vectors

Name	Offset	Size	Reset Value	Description
VDPU_SWREG42	0x00a8	W	0x00000000	bi_dir initial ref pic list register (0-2) prediction filter taps
VDPU_SWREG43	0x00ac	W	0x00000000	bi-dir initial ref pic list register (3-5) prediction filter taps
VDPU_SWREG44	0x00b0	W	0x00000000	bi-dir initial ref pic list register (6-8) prediction filter taps
VDPU_SWREG45	0x00b4	W	0x00000000	bi-dir initial ref pic list register (9-11) prediction filter taps
VDPU_SWREG46	0x00b8	W	0x00000000	bi-dir initial ref pic list register (12-14)
VDPU_SWREG47	0x00bc	W	0x00000000	bi-dir and P fwd initial ref pic list register (15 and P 0-3)
VDPU_SWREG48	0x00c0	W	0x00000000	Error concealment register
VDPU_SWREG49	0x00c4	W	0x00000000	Prediction filter tap register for H264, MPEG4
VDPU_SWREG50	0x00c8	W	0xfbb56f80	Synthesis configuration register decoder 0 (read only)
VDPU_SWREG51	0x00cc	W	0x00000000	Reference picture buffer control register
VDPU_SWREG52	0x00d0	W	0x00000000	Reference picture buffer information register 1 (read only)
VDPU_SWREG53	0x00d4	W	0x00000000	Reference picture buffer information register 2 (read only)
VDPU_SWREG54	0x00d8	W	0xe5da0000	Synthesis configuration register decoder 1 (read only)
VDPU_SWREG55	0x00dc	W	0x00000000	Reference picture buffer 2 / Advanced prefetch control register
VDPU_SWREG56	0x00e0	W	0x00000000	Reference buffer information register 3 (read only)
VDPU_SWREG57_INTRA_I NTER	0x00e4	W	0x00000000	intra_dll3t,intra_dblspeed,inter_d blspeed,stream_len_hi
VDPU_SWREG57	0x00e4	W	0x00000000	intra_dll3t,intra_dblspeed,inter_d blspeed,stream_len_hi
VDPU_SWREG58	0x00e8	W	0x00000000	Decoder debug register 0 (read only)
VDPU_SWREG59	0x00ec	W	0x00000000	H264 Chrominance 8 pixel interleaved data base
VDPU_SWREG60	0x00f0	W	0x00000000	Interrupt register post-processor
VDPU_SWREG61	0x00f4	W	0x01010100	Device configuration register post-processor
VDPU_SWREG62	0x00f8	W	0x00000000	Deinterlace control register
VDPU_SWREG63	0x00fc	W	0x00000000	base address for reading post-processing input picture uminan
VDPU_SWREG64	0x0100	W	0x00000000	Base address for reading post-processing input picture Cb/Ch
VDPU_SWREG65	0x0104	W	0x00000000	Base address for reading post-processing input picture Cr
VDPU_SWREG66	0x0108	W	0x00000000	Base address for writing post-processed picture luminance/RGB
VDPU_SWREG67	0x010c	W	0x00000000	Base address for writing post-processed picture Ch
VDPU_SWREG68	0x0110	W	0x00000000	Register for contrast adjusting

Name	Offset	Size	Reset Value	Description
VDPU_SWREG69	0x0114	W	0x00000000	Register for colour conversion and contrast adjusting
VDPU_SWREG70	0x0118	W	0x00000000	Register for colour conversion 0
VDPU_SWREG71	0x011c	W	0x00000000	Register for colour conversion 1 + rotation mode
VDPU_SWREG72	0x0120	W	0x00000000	PP input size and -cropping register
VDPU_SWREG73	0x0124	W	0x00000000	PP input picture base address for Y bottom field
VDPU_SWREG74	0x0128	W	0x00000000	PP input picture base for Ch bottom field
VDPU_SWREG79	0x013c	W	0x00000000	Scaling ratio register 1 & padding for B
VDPU_SWREG80	0x0140	W	0x00000000	Scaling register 0 ratio & padding for R and G
VDPU_SWREG81	0x0144	W	0x00000000	Scaling ratio register 2
VDPU_SWREG82	0x0148	W	0x00000000	Rmask register
VDPU_SWREG83	0x014c	W	0x00000000	Gmask register
VDPU_SWREG84	0x0150	W	0x00000000	Bmask register
VDPU_SWREG85	0x0154	W	0x00000000	Post-processor control register
VDPU_SWREG86	0x0158	W	0x00000000	Mask 1 start coordinate register
VDPU_SWREG87	0x015c	W	0x00000000	Mask 2 start coordinate register
VDPU_SWREG88	0x0160	W	0x00000000	Mask 1 size and PP original width register
VDPU_SWREG89	0x0164	W	0x00000000	Mask 2 size register
VDPU_SWREG90	0x0168	W	0x00000000	PiP register 0
VDPU_SWREG91	0x016c	W	0x00000000	PiP register 1 and dithering control
VDPU_SWREG92	0x0170	W	0x00000000	Display width and PP input size extension register
VDPU_SWREG93	0x0174	W	0x00000000	Display width and PP input size extension register
VDPU_SWREG94	0x0178	W	0x00000000	Base address for alpha blend 2 gui component
VDPU_SWREG95	0x017c	W	0x00000000	Base address for alpha blend 2 gui component
VDPU_SWREG98	0x0188	W	0x00000000	PP output width/height extension
VDPU_SWREG99	0x018c	W	0xe000f000	PP fuse register (read only)
VDPU_SWREG100	0x0190	W	0xff874780	Synthesis configuration register post-processor (read only)
VDPU_SWREG101	0x0194	W	0x00000000	soft reset signals
VDPU_SWREG102	0x0198	W	0x00000000	vpu performance cycle
VDPU_SWREG103	0x019c	W	0x00000000	AXI DDR READ DATA NUM
VDPU_SWREG104	0x01a0	W	0x00000000	Register0000 Abstract
VDPU_SWREG105	0x01a4	W	0x00000000	
VDPU_SWREG106	0x01a8	W	0x00000000	
VDPU_SWREG107	0x01ac	W	0x00000000	

Notes: **S**-Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

Name	Offset	Size	Reset Value	Description
V_CODEC_MMU_DTE_ADDR	0x0000	W	0x00000000	MMU current page Table address
V_CODEC_MMU_STATUS	0x0004	W	0x00000018	MMU status register
V_CODEC_MMU_COMMAND	0x0008	W	0x00000000	MMU command register
V_CODEC_MMU_PAGE_FAULT_ADDR	0x000c	W	0x00000000	MMU logical address of last page fault

Name	Offset	Size	Reset Value	Description
V_CODEC_MMU_ZAP_ONE_LINE	0x0010	W	0x00000000	MMU Zap cache line register
V_CODEC_MMU_INT_RAW_STAT	0x0014	W	0x00000000	MMU raw interrupt status register
V_CODEC_MMU_INT_CLEA_R	0x0018	W	0x00000000	MMU raw interrupt status register
V_CODEC_MMU_INT_MASK	0x001c	W	0x00000000	MMU raw interrupt status register
V_CODEC_MMU_INT_STAT_US	0x0020	W	0x00000000	MMU raw interrupt status register
V_CODEC_MMU_AUTO_GATING	0x0024	W	0x00000001	mmu auto gating

Notes: Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

Name	Offset	Size	Reset Value	Description
pref_cache_VERSION	0x0000	W	0xcac20101	VERSION register
pref_cache_SIZE	0x0004	W	0x06110206	L2 cache SIZE
pref_cache_STATUS	0x0008	W	0x00000000	Status register
pref_cache_COMMAND	0x0010	W	0x00000000	Command setting register
pref_cache_CLEAR_PAGE	0x0014	W	0x00000000	clear page register
pref_cache_MAX_READS	0x0018	W	0x0000001c	maximum read register
pref_cache_PERFCNT_SR_C0	0x0020	W	0x00000000	performance counter 0 source register
pref_cache_PERFCNT_VAL_0	0x0024	W	0x00000000	performance counter 0 value register
pref_cache_PERFCNT_SR_C1	0x0028	W	0x00000000	performance counter 0 source register
pref_cache_PERFCNT_VAL_1	0x002c	W	0x00000000	performance counter 1 value register

Notes: Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

#### 4.5.5 VDPU Detail Register Description

##### VDPU\_SWREG0\_NEW\_VERSION

Address: Operational Base + offset (0x0000)

ID register(read only)

Bit	Attr	Reset Value	Description
31:28	RO	0x0	reserved
27:24	RO	0x3	major_version 0:1080p support 1:2160p support
23:16	RO	0x68	minor_version 0: audis 1: audi 2: maybach 3:audib ff:share memory with hevc,so should read verision from hevc register
15:0	RW	0x0000	build the rtl's svn num in ic server

##### VDPU\_SWREG0

Address: Operational Base + offset (0x0000)

Register0000 Abstract

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	pro_num product number
15:12	RW	0x0	major_version major_version major_version
11:4	RW	0x00	minor_version minor_version minor_version
3	RW	0x0	ID_ASCII_EN ASCII type product ID enable ASCII type product ID enable
2:0	RW	0x0	build_version build_version build_version

**VDPU\_SWREG1**

Address: Operational Base + offset (0x0004)  
interrupt register decoder

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:25	RO	0x0	reserved
24	RW	0x0	sw_dec_pic_inf B slice detected. This signal is driven high during picture ready interrupt if B-type slice is found. This bit does not launch interrupt but is used to inform SW about h264 tools. [note]:the h264 decoder will use these bits.
23:19	RO	0x0	reserved
18	RW	0x0	sw_dec_timeout Interrupt status bit decoder timeout. When high the decoder 0 has been idling for too long. HW will self reset. Possible only if timeout interrupt is enabled [note]:the h264 decoder will use these bits.
17	RW	0x0	sw_dec_slice_int Interrupt status bit dec_slice_decoded. When high SW must set new base addresses for sw_dec_out_base and sw_jpg_ch_out_base before resetting this status bit. Used for JPEG snapshot modes [note]:the JPEG decoder will use these bits.
16	RW	0x0	sw_dec_error_int Interrupt status bit input stream error. When high, an error is found in input data stream decoding. HW will self reset. [note]:the h264 decoder will use these bits.

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
15	RW	0x0	sw_dec_aso_int Interrupt status bit ASO (Arbitrary Slice Ordering) detected. When high, ASO detected in input data stream decoding. HW will self reset. [note]:the h264 decoder will use these bits.
14	RW	0x0	sw_dec_buffer_int Interrupt status bit input buffer empty. When high, input stream buffer is empty but picture is not ready. HW will not self reset. [note]:the h264 decoder will use these bits.
13	RW	0x0	sw_dec_bus_int Interrupt status bit bus. Error response from bus. HW will self reset [note]:the h264 decoder will use these bits.
12	RW	0x0	sw_dec_rdy_int Interrupt status bit decoder. When this bit is high decoder has decoded a picture. HW will self reset. [note]:the h264 decoder will use these bits.
11:9	RO	0x0	reserved
8	RW	0x0	sw_dec_irq Decoder IRQ. When high, decoder requests an interrupt. SW will reset this after interrupt is handled. [note]:the h264 decoder will use these bits.
7:5	RO	0x0	reserved
4	RW	0x0	sw_dec_irq_dis Decoder IRQ disable. When high, there are no interrupts concerning decoder from HW. Polling must be used to see the interrupt statuses. [note]:the h264 decoder will use these bits.
3:1	RO	0x0	reserved
0	RW	0x0	sw_dec_en decoder enable. Decoder enable. Setting this bit high will start the decoding operation. HW will reset this when picture is processed or ASO or stream error is detected or bus error or timeout interrupt is given. [note]:the h264 decoder will use these bits.

**VDPU\_SWREG2**

Address: Operational Base + offset (0x0008)

device configuration register decoder

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RW	0x01	sw_dec_axi_rd_id Read ID used for decoder reading services in AXI bus (if connected to AXI) [note]:the h264 decoder will use these bits.

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
23	RW	0x0	<p>sw_dec_timeout_e Timeout interrupt enable. If enabled HW may return timeout interrupt in case HW gets stucked while decoding picture. [note]:the h264 decoder will use these bits.</p>
22	RW	0x0	<p>sw_dec_strswap32_e Decoder input 32bit data swap for stream data (may be used for 64 bit environment): 0 = no swapping of 32 bit words 1 = 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled)) [note]:the h264 decoder will use these bits.</p>
21	RW	0x0	<p>sw_dec_strendian_e Decoder input endian mode for stream data: 0 = Big endian (0-1-2-3 order) 1 = Little endian (3-2-1-0 order) [note]:the h264 decoder will use these bits.</p>
20	RW	0x0	<p>sw_dec_inswap32_e Decoder input 32bit data swap for other than stream data (may be used for 64 bit environment): 0 = no swapping of 32 bit words 1 = 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled)) [note]:the h264 decoder will use these bits.</p>
19	RW	0x0	<p>sw_dec_outswap32_e Decoder output 32bit data swap (may be used for 64 bit environment): 0 = no swapping of 32 bit words 1 = 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled)) [note]:the h264 decoder will use these bits.</p>
18	RW	0x0	<p>sw_dec_data_disc_e Data discard enable. Precise burst lengths are used with reading services. Extra data is discarded internally. [note]:the h264 decoder will use these bits.</p>
17	RW	0x0	<p>sw_tiled_mode_msb Tiled mode msb. Concanated to Tiled mode lsb which form 2 bit tiled mode. Definition of tiledmode: 0 = Tiled mode not enabled 1 = Tiled mode enabled for 8x4 tile size 2,3 Reserved [note]:the h264 decoder will use these bits.</p>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
16:11	RW	0x00	<p>sw_dec_latency</p> <p>Decoder master interface additional latency. Can be used to slow down decoder HW between services in steps of 8 clock cycles:</p> <p>0 = no latency      1 = minimum 8 cycles of IDLE between services      2 = minimum 16 cycles of IDLE between services      ...      63 = minimum latency of 504 cycles of IDLE between services</p> <p>[note]:the h264 decoder will use these bits.</p>
10	RW	0x1	<p>sw_dec_clk_gate_e</p> <p>Decoder dynamic clock gating enable:</p> <p>0 = Clock is running for all structures      1 = Clock is gated for decoder structures that are not used</p> <p>Note: Clock gating value can be changed only when decoder is disabled</p>
9	RW	0x0	<p>sw_dec_in_endian</p> <p>Decoder input endian mode for other than stream data:</p> <p>0 = Big endian (0-1-2-3 order)      1 = Little endian (3-2-1-0 order)</p> <p>[note]:the h264 decoder will use these bits.</p>
8	RW	0x0	<p>sw_dec_out_endian</p> <p>Decoder output endian mode:</p> <p>0 = Big endian (0-1-2-3 order)      1 = Little endian (3-2-1-0 order)</p> <p>[note]:the h264 decoder will use these bits.</p>
7	RW	0x0	<p>sw_tiled_mode_lsb</p> <p>Tiled mode lsb. Concatenated to Tiled mode msb which form 2 bit tiled mode. Defined in tiled_mode_msbs</p> <p>[note]:the h264 decoder will use these bits.</p>
6	RW	0x0	<p>sw_dec_adv_pre_dis</p> <p>Advanced PREFETCH mode disable (advanced reference picture reading mode for video)</p> <p>[note]:the h264 decoder will use these bits.</p>
5	RW	0x0	<p>sw_dec_scmd_dis</p> <p>AXI Single Command Multiple Data 0 disable. (where only the first addresses of the burst are given from address generator). This bit is used to disable the feature (possible SW workaround if something is not working correctly)</p> <p>[note]:the h264 decoder will use these bits.</p>
4:0	RW	0x00	<p>sw_dec_max_burst</p> <p>Maximum burst length for decoder bus transactions. Valid values: AXI: 1-16</p> <p>[note]:the h264 decoder will use these bits.</p>

**VDPU\_SWREG3**

Address: Operational Base + offset (0x000c)  
 Device control register 0(decmode, picture type etc)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:28	RW	0x0	<p>sw_dec_mode            Decoding mode:            0 = H.264,            1 = MPEG-4,            2 = H.263,            3 = JPEG,            5 = MPEG-2,            6 = MPEG-1,            others = reserved</p> <p>[note]:all the decoder mode will use these bits.</p>
27	RW	0x0	<p>sw_rlc_mode_e            RLC mode enable:            1 = HW decodes video from RLC input data + side information            (Differential MV's, separate DC coeffs, Intra 4x4 modes, MB control). Valid only for H.264 Baseline and MPEG-4 SP.            0 = HW decodes video from bit stream (VLC mode) + side information</p> <p>[note]:the h264 and MPEG4 decoder will use these bits.</p>
26	RW	0x0	sw_skip_mode
25	RW	0x0	reserved
24	RW	0x0	<p>sw_pjpeg_e            Progressive JPEG enable:            0 = baseline JPEG            1 = progressive JPEG</p>
23	RW	0x0	<p>sw_pic_interlace_e            Coding mode of the current picture:            0 = progressive            1 = interlaced</p> <p>[note]:the h264 decoder will use these bits.</p>
22	RW	0x0	<p>sw_pic_fieldmode_e            Structure of the current picture (residual structure)            0 = frame structure, this means MBAFF structured picture for interlaced sequence            1 = field structure</p> <p>[note]:the h264 decoder will use these bits.</p>
21	RW	0x0	<p>sw_pic_b_e            B picture enable for current picture:            0=picture type is I or P depending on sw_pic_inter_e            1=picture type is BI (vc1)/D (mpeg1) or B depending on sw_pic_inter_e (not valid for H264 since its slice based information)</p>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
20	RW	0x0	sw_pic_inter_e Picture type. 1= Inter type (P) 0= Intra type (I) See also sw_pic_b_e
19	RW	0x0	sw_pic_topfield_e If field structure is enabled this bit informs which one of the fields is being decoded: 0 = bottom field 1 = top field [note]:the h264 decoder will use these bits.
18	RW	0x0	sw_fwd_interlace_e Coding mode of forward reference picture: 0 = progressive 1 = interlaced Note: for backward reference picture the coding mode is always same as for current picture.
17	RW	0x0	reserved
16	RW	0x0	sw_ref_topfield_e Indicates which field should be used as reference if sw_ref_frames = 0 0 = bottom field 1 = top field used only in VC-1 mode
15	RW	0x0	sw_dec_out_dis Disable decoder output picture writing: 0 = Decoder output picture is written to external memory 1 = Decoder output picture is not written to external memory [note]:the h264 decoder will use these bits.
14	RW	0x0	sw_filtering_dis De-block filtering disable 1 = filtering is disabled for current picture 0 = filtering is enabled for current picture [note]:the h264 decoder will use these bits.
13	RW	0x0	sw_pic_fixed_quant sw_pic_fixed_quant (DEC mode is VC-1 and AVS) 0 = Quantization parameter can vary inside picture 1 = Quantization parameter is fixed (pquant) sw_mvc_e(DEC mode is H264) multi view coding enable. Possible for H264 only [note]:the h264 decoder will use these bits.

Bit	Attr	Reset Value	Description
12	RW	0x0	sw_write_mvs_e Direct mode motion vector write enable for current picture / MPEG2 motion vector write enable for error concealment purposes: 0 = writing disabled for current picture 1 = the direct mode motion vectors are written to external memory. H264 direct mode motion vectors are written to DPB aside with the corresponding reference picture. Other decoding mode dir mode mvs are written to external memory starting from sw_dir_mv_base [note]:the h264 decoder will use these bits.
11	RW	0x0	sw_reftopfirst_e Indicates which FWD reference field has been decoded first. 0 = FWD reference bottom field 1 = FWD reference top field [note]:the h264 decoder will use these bits.
10	RW	0x0	sw_seq_mbaff_e Sequence includes MBAFF coded pictures [note]:the h264 decoder will use these bits.
9	RW	0x0	sw_picord_count_e h264_high config: Picture order count table read enable. If enabled HW will read picture order counts from memory in the beginning of picture [note]:the h264 decoder will use these bits.
8	RW	0x0	sw_dec_timeout_mode dec timeout mode selset when 1'b0,timeout cycle is 181'b1 when 1'b1,timeout cycle is 221'b1 [note]:the h264 decoder will use these bits.
7:0	RW	0x01	sw_dec_axi_wr_id Write ID used for decoder writing services in AXI bus (if connected to AXI) [note]:the h264 decoder will use these bits.

**VDPU\_SWREG4\_H264**

Address: Operational Base + offset (0x0010)  
decoder control register 1(picture parameters)

Bit	Attr	Reset Value	Description
31:23	RW	0x000	sw_pic_mb_width Picture width in macroblocks = ((width in pixels + 15) /16) [note]:the h264 decoder will use these bits.
22:19	RO	0x0	reserved
18:11	RW	0x00	sw_pic_mb_height_p Picture height in macroblocks =((height in pixels+15)/16). Picture height is informed as size of the (progressive) frame also for single field (of interlaced content) is being decoded [note]:the h264 decoder will use these bits.

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
10:5	RO	0x0	reserved
4:0	RW	0x00	sw_ref_frames 264: num_ref_frames, maximum number of short and long term reference frames in decoded picture buffer.

**VDPU\_SWREG4**

Address: Operational Base + offset (0x0010)  
decoder control register 1(picture parameters)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:23	RW	0x000	sw_pic_mb_width Picture width in macroblocks = ((width in pixels + 15) / 16)
22:19	RW	0x0	sw_mb_width_off The amount of meaningful horizontal pixels in last MB (width offset) 0 if exactly 16 pixels multiple picture and all the horizontal pixels in last MB are meaningful
18:11	RW	0x00	sw_pic_mb_height_p Picture height in macroblocks = ((height in pixels+15)/16). Picture height is informed as size of the (progressive) frame also for single field (of interlaced content) is being decoded
10:7	RW	0x0	sw_mb_height_off The amount of meaningful vertical pixels in last MB (height offset 0 if exactly 16 pixels multiple picture and all the vertical pixels in last MB are meaningful)
6	RW	0x0	sw_alt_scan_e indicates alternative vertical scan method used for interlaced frames
5:3	RW	0x0	sw_pic_mb_w_ext Picture mb width extension. If sw_pic_mb_width does not fit to 9 bits then these bits are used to increase the range upto 11 bits (used as 3 msb)
2:0	RW	0x0	sw_pic_mb_h_ext Picture mb height extension. If sw_pic_mb_height_p does not fit to 9 bits then these bits are used to increase the range upto 11 bits (used as 3 msb)

**VDPU\_SWREG5**

Address: Operational Base + offset (0x0014)  
decoder control register2 (stream decoding table selects)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:26	RW	0x00	sw_strm_start_bit Exact bit of stream start word where decoding can be started (associates with sw_rlc_vlc_base)

Bit	Attr	Reset Value	Description
25	RW	0x0	sw_sync_marker_e Sync markers enable: '0' = synch markers are not used, '1' = synch markers are used. For progressive JPEG this indicates that there are restart markers in the stream after restart interval steps
24	RW	0x0	sw_type1_quant_e MPEG4: Type 1 quantization enable '0' = type 2 inverse Q method '1' = type 1 inverse Q method (Q-tables used) H264 (h264_high config): scaling matrix enable: '0' = normal transform '1' = use scaling matrix for transform (read from external
23:19	RW	0x00	sw_ch_qp_offset Chroma Qp filter offset. (For H.264 this offset concerns Cb only)
18:14	RW	0x00	sw_ch_qp_offset2 Chroma Qp filter offset for cr type
13:1	RO	0x0	reserved
0	RW	0x0	sw_fieldpic_flag_e Flag for streamd that field_pic_flag exists in stream

**VDPU\_SWREG5\_H264**

Address: Operational Base + offset (0x0014)

decoder control register2 (stream decoding table selects)

Bit	Attr	Reset Value	Description
31:26	RW	0x00	sw_strm_start_bit Exact bit of stream start word where decoding can be started (associates with sw_rlc_vlc_base)
25	RO	0x0	reserved
24	RW	0x0	sw_type1_quant_e MPEG4: Type 1 quantization enable '0' = type 2 inverse Q method '1' = type 1 inverse Q method (Q-tables used) H264 (h264_high config): scaling matrix enable: '0' = normal transform '1' = use scaling matrix for transform (read from external
23:19	RW	0x00	sw_ch_qp_offset Chroma Qp filter offset. (For H.264 this offset concerns Cb only)
18:14	RW	0x00	sw_ch_qp_offset2 Chroma Qp filter offset for cr type
13:1	RO	0x0	reserved
0	RW	0x0	sw_fieldpic_flag_e Flag for streamd that field_pic_flag exists in stream

**VDPU\_SWREG6**

Address: Operational Base + offset (0x0018)

decoder control register 3( stream buffer information)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RW	0x0	sw_start_code_e Bit for indicating stream start code existence: '0' = stream doesn't contain start codes '1' = stream contains start codes [note]:the h264 decoder will use these bits.
30:25	RW	0x00	sw_init_qp Initial value for quantization parameter (picture quantizer). [note]:the h264 decoder will use these bits.
24	RW	0x0	sw_ch_8pix_ileav_e Enable for additional chrominance data format writing where decoder writes chrominance in group of 8 pixels of Cb and then corresponding 8 pixels of Cr. Data is written to sw_dec_ch8pix_base. Cannot be used if tiled mode is enabled [note]:the h264 decoder will use these bits.
23:0	RW	0x0000000	sw_stream_len Amount of stream data bytes in input buffer. If the given buffer size is not enough for finishing the picture the corresponding interrupt is given and new stream buffer base address and stream buffer size information should be given (associates with sw_rlc_vlc_base). For VC-1 the buffer must include data for one picture/slice of the picture For H264/MPEG4/H263/MPEG2/MPEG1 the buffer must include at least data for one slice of the picture For JPEG the buffer size must be a multiple of 256 bytes or the amount of data for one picture. [note]:the h264 decoder will use these bits.

**VDPU\_SWREG7**

Address: Operational Base + offset (0x001c)

decoder control register 4(H264, VC-1 control )

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RW	0x0	sw_cabac_e CABAC enable [note]:the h264 decoder will use these bits.
30	RW	0x0	sw_blackwhite_e '0' = 4:2:0 sampling format '1' = 4:0:0 sampling format (H264 monochroma) [note]:the h264 decoder will use these bits.
29	RW	0x0	sw_dir_8x8_infer_e Specifies the method to use to derive luma motion vectors in B_skip, B_Direct_16x16 and B_direct_8x8_inference_flag (see direct_8x8_inference flag) [note]:the h264 decoder will use these bits.

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
28	RW	0x0	sw_weight_pred_e Weighted prediction enable for P slices [note]:the h264 decoder will use these bits.
27:26	RW	0x0	sw_weight_bipr_idc weighted prediction specification for B slices: "00" = default weighted prediction is applied to B slices "01" = explicit weighted prediction shall be applied to B slices "10" = implicit weighted prediction shall be applied to B slices [note]:the h264 decoder will use these bits.
25:21	RO	0x0	reserved
20:16	RW	0x00	sw_framenum_len H.264: Bit length of frame_num in data stream [note]:the h264 decoder will use these bits.
15:0	RW	0x0000	sw_framenum current frame_num, used to identify short-term reference frames. Used in reference picture reordering [note]:the h264 decoder will use these bits.

**VDPU\_SWREG8**

Address: Operational Base + offset (0x0020)  
decoder control register 5(H264 control)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RW	0x0	sw_const_intra_e constrained_intra_pred_flag equal to 1 specifies that intra prediction uses only neighbouring intra macroblocks in prediction. When equal to 0 also neighbouring inter macroblocks are used in intra prediction process. [note]:the h264 decoder will use these bits.
30	RW	0x0	sw_filt_ctrl_pres deblocking_filter_control_present_flag indicates whether extra variables controlling characteristics of the deblocking filter are present in the slice header. [note]:the h264 decoder will use these bits.
29	RW	0x0	sw_rdpic_cnt_pres redundant_pic_cnt_present_flag specifies whether redundant_pic_cnt syntax elements [note]:the h264 decoder will use these bits.
28	RW	0x0	sw_8x8trans_flag_e 8x8 transform flag enable for stream decoding [note]:the h264 decoder will use these bits.
27:17	RW	0x000	sw_refpic_mk_len Length of decoded reference picture marking bits [note]:the h264 decoder will use these bits.

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
16	RW	0x0	sw_idr_pic_e IDR (instantaneous decoding refresh) picture flag. [note]:the h264 decoder will use these bits.
15:0	RW	0x0000	sw_idr_pic_id idr_pic_id, identifies IDR (instantaneous decoding refresh) picture [note]:the h264 decoder will use these bits.

**VDPU\_SWREG9**

Address: Operational Base + offset (0x0024)

decoder control register 6

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RW	0x00	sw_pps_id pic_parameter_set_id, identifies the picture parameter set that is referred to in the slice header. [note]:the h264 decoder will use these bits.
23:19	RW	0x00	sw_refidx1_active Specifies the maximum reference index that can be used while decoding inter predicted macro blocks. [note]:the h264 decoder will use these bits.
18:14	RW	0x00	sw_refidx0_active Specifies the maximum reference index that can be used while decoding inter predicted macro blocks. This is same as in previous decoders (width increased with q bit) [note]:the h264 decoder will use these bits.
13:8	RO	0x0	reserved
7:0	RW	0x00	sw_poc_length Length of picture order count field in stream [note]:the h264 decoder will use these bits.

**VDPU\_SREG10\_H264\_RLC**

Address: Operational Base + offset (0x0028)

Base address for differential motion vector base address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_diff_mv_base for H264 and MPEG4, RLC mode: Differential motion vector base address.
1:0	RO	0x0	reserved

**VDPU\_SREG10\_H264**

Address: Operational Base + offset (0x0028)

Base address for differential motion vector base address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
29:25	RW	0x00	sw_pinit_rlist_f9 initial reference picture list for P forward picid 9
24:20	RW	0x00	sw_pinit_rlist_f8 initial reference picture list for P forward picid 8
19:15	RW	0x00	sw_pinit_rlist_f7 initial reference picture list for P forward picid 7
14:10	RW	0x00	sw_pinit_rlist_f6 initial reference picture list for P forward picid 6
9:5	RW	0x00	sw_pinit_rlist_f5 initial reference picture list for P forward picid 5
4:0	RW	0x00	sw_pinit_rlist_f4 initial reference picture list for P forward picid 4

**VDPU\_SWREG11\_H264\_RLC**

Address: Operational Base + offset (0x002c)

decoder control register 7

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_i4x4_or_dc_base RLC mode: H.264: Intra prediction 4x4 mode base address. RLC mode: MPEG-4: DC component base address.
1:0	RO	0x0	reserved

**VDPU\_SWREG11\_H264**

Address: Operational Base + offset (0x002c)

decoder control register 7

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:25	RW	0x00	sw_pinit_rlist_f15 Initial reference picture list for P forward picid 15
24:20	RW	0x00	sw_pinit_rlist_f14 Initial reference picture list for P forward picid 14
19:15	RW	0x00	sw_pinit_rlist_f13 Initial reference picture list for P forward picid 13
14:10	RW	0x00	sw_pint_rlist_f12 Initial reference picture list for P forward picid 12
9:5	RW	0x00	sw_pint_rlist_f11 Initial reference picture list for P forward picid 11
4:0	RW	0x00	sw_pint_rlist_f10 Initial reference picture list for P forward picid 10

**VDPU\_SWREG12**

Address: Operational Base + offset (0x0030)

Base address for RLC data (RLC) / stream start address/decoded

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	<p>sw_rlc_vlc_base RLC mode: Base address for RLC data (swreg3.sw_rlc_mode_e = 1). VLC mode: Stream start address / end addr+I288ess with byte precision (swreg4.rlc_mode_en = 0), start bit number in swreg5.stream_start_bit. When sw_dec_buffer_int is high or sw_dec_e is low this register contains HW return value of last_byte_address (not valid for jpeg) where stream has been read (and used) in accuracy of byte. For debug purposes the last_byte_address is also written when stream error/ASO is detected even though it may not be accurate. [note]:the h264 decoder will use these bits.</p>
1:0	RO	0x0	reserved

**VDPU\_SWREG13**

Address: Operational Base + offset (0x0034)

Base address for decoded picture / base address for JPEG deco

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	<p>sw_dec_out_base Video: Base address for decoder output picture. Points directly to start of decoder output picture or field. JPEG snapshot: Base address for decoder output luminance picture [note]:the h264 decoder will use these bits.</p>
1:0	RO	0x0	reserved

**VDPU\_SWREG14**

Address: Operational Base + offset (0x0038)

Base address for reference picture index 0 / base address for J

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	<p>sw_refer0_base Base address for reference picture index 0. See picture index definition from toplevel_sp [note]:the h264 decoder will use these bits.</p>
1	RW	0x0	<p>sw_refer0_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields [note]:the h264 decoder will use these bits.</p>
0	RW	0x0	<p>sw_refer0_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture [note]:the h264 decoder will use these bits.</p>

**VDPU\_SWREG15\_JPEG\_ROI**

Address: Operational Base + offset (0x003c)

JPEG roi control

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:20	RO	0x0	reserved
19	RW	0x0	sw_jpegroi_in_endian jpeg offset input endian sw_jpegroi_in_endian 0 = big endian (0-1-2-3 order) 1 = little endian (3-2-1-0 order)
18	RW	0x0	sw_jpegroi_in_swap32 jpeg offset input 32-bit swap sw_jpegroi_in_swap32 0: no swapping of 32 bit words 1: 32bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1 byte order (also little endian should be enabled))
17:16	RW	0x0	sw_roi_sample_size ROI MB num sample each time ROI MB num sample each time 00:1 01:8 10:16 11:8
15:12	RW	0x0	sw_roi_distance roi distance The distance between the sample MB and ROI start MB
11:10	RW	0x0	sw_roi_out_sel roi output selection ROI output selection 00: output offset/dc 01: output picture 10: output offset/dc and picture 11: output offset/dc
9	RW	0x0	sw_roi_decode roi decode JPEG ROI decode 0: build offset/dc table 1: ROI decode
8	RW	0x0	sw_roi_en roi enable JPEG roi mode enable 0: normal jpeg decode mode 1: JPEG roi mode
7:0	RO	0x0	reserved

**VDPU\_SWREG15**

Address: Operational Base + offset (0x003c)

Base address for reference picture index 1 / JPEG control

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_refer1_base Base address for reference picture index 1. See picture index definition from toplevel_sp [note]:the h264 decoder will use these bits.
1	RW	0x0	sw_refer1_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer1_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

**VDPU\_SWREG16**

Address: Operational Base + offset (0x0040)

base address for reference picture index 2 / List of VLC code len

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_refer2_base Base address for reference picture index 2. See picture index definition from toplevel_sp [note]:the h264 decoder will use these bits.
1	RW	0x0	sw_refer2_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields [note]:the h264 decoder will use these bits.
0	RW	0x0	sw_refer2_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture [note]:the h264 decoder will use these bits.

**VDPU\_SWREG17**

Address: Operational Base + offset (0x0044)

Base address for reference picture index 3 / List of VLC code len

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_refer3_base Base address for reference picture index 3. See picture index definition from toplevel_sp [note]:the h264 decoder will use these bits.

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
1	RW	0x0	sw_refer3_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields [note]:the h264 decoder will use these bits.
0	RW	0x0	sw_refer3_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture [note]:the h264 decoder will use these bits.

**VDPU\_SWREG18**

Address: Operational Base + offset (0x0048)

Base address for reference picture index 4 / VC1 control / MPE

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_refer4_base Base address for reference picture index 4. See picture index definition from toplevel_sp [note]:the h264 decoder will use these bits.
1	RW	0x0	sw_refer4_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields [note]:the h264 decoder will use these bits.
0	RW	0x0	sw_refer4_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture [note]:the h264 decoder will use these bits.

**VDPU\_SWREG19**

Address: Operational Base + offset (0x004c)

Base address for reference picture index 5

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_refer5_base Base address for reference picture index 5. See picture index definition from toplevel_sp [note]:the h264 decoder will use these bits.
1	RW	0x0	sw_refer5_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields [note]:the h264 decoder will use these bits.

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
0	RW	0x0	<p>sw_refer5_topc_e</p> <p>Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture</p> <p>[note]:the h264 decoder will use these bits.</p>

**VDPU\_SWREG20**

Address: Operational Base + offset (0x0050)

Base address for reference picture index 6

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	<p>sw_refer6_base</p> <p>Base address for reference picture index 6. See picture index definition from toplevel_sp</p> <p>[note]:the h264 decoder will use these bits.</p>
1	RW	0x0	<p>sw_refer6_field_e</p> <p>Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields</p> <p>[note]:the h264 decoder will use these bits.</p>
0	RW	0x0	<p>sw_refer6_topc_e</p> <p>Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture</p> <p>[note]:the h264 decoder will use these bits.</p>

**VDPU\_SWREG21**

Address: Operational Base + offset (0x0054)

Base address for reference picture index 7

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	<p>sw_refer7_base</p> <p>Base address for reference picture index 7. See picture index definition from toplevel_sp</p> <p>[note]:the h264 decoder will use these bits.</p>
1	RW	0x0	<p>sw_refer7_field_e</p> <p>Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields</p> <p>[note]:the h264 decoder will use these bits.</p>
0	RW	0x0	<p>sw_refer7_topc_e</p> <p>Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture</p> <p>[note]:the h264 decoder will use these bits.</p>

**VDPU\_SWREG22**

**RK3228 TRM**

Address: Operational Base + offset (0x0058)

Base address for reference picture index 8

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_refer8_base Base address for reference picture index 8. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer8_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer8_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

**VDPU\_SWREG23**

Address: Operational Base + offset (0x005c)

Base address for reference picture index 9

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_refer9_base Base address for reference picture index 9. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer9_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer9_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

**VDPU\_SWREG24**

Address: Operational Base + offset (0x0060)

Base address for reference picture index 10

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_refer10_base Base address for reference picture index 10. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer10_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer10_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

**VDPU\_SWREG25**

Address: Operational Base + offset (0x0064)

Base address for reference picture index 11

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:3	RO	0x0	reserved
2	RW	0x0	sw_refer11_base Base address for reference picture index 11. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer11_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer11_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

**VDPU\_SWREG26**

Address: Operational Base + offset (0x0068)

Base address for reference picture index 12

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_refer12_base Base address for reference picture index 12. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer12_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer12_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

**VDPU\_SWREG27**

Address: Operational Base + offset (0x006c)

Base address for reference picture index 13

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_refer13_base Base address for reference picture index 13. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer13_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields

Bit	Attr	Reset Value	Description
0	RW	0x0	sw_refer13_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

**VDPU\_SWREG28**

Address: Operational Base + offset (0x0070)

Base address for reference picture index14

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_refer14_base Base address for reference picture index 14. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer14_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer14_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

**VDPU\_SWREG29**

Address: Operational Base + offset (0x0074)

Base address for reference picture index15

Bit	Attr	Reset Value	Description
31:2	RW	0x00000000	sw_refer15_base Base address for reference picture index 15. See picture index definition from toplevel_sp
1	RW	0x0	sw_refer15_field_e Refer picture consist of single fields or frame: '0' = reference picture consists of frame '1' = reference picture consists of fields
0	RW	0x0	sw_refer15_topc_e Which field of reference picture is closer to current picture: '0' = bottom field is closer to current picture '1' = top field is closer to current picture

**VDPU\_SWREG30**

Address: Operational Base + offset (0x0078)

Reference picture numbers for index 0 and 1 (H264 VLC)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	sw_refer1_nbr Number for reference picture index 1
15:0	RW	0x0000	sw_refer0_nbr Number for reference picture index 0

**VDPU\_SWREG31**

Address: Operational Base + offset (0x007c)

Reference picture numbers for index 2 and 3 (H264 VLC) /

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_refer3_nbr Number for reference picture index 3
15:0	RW	0x0000	sw_refer2_nbr Number for reference picture index 2

**VDPU\_SWREG32**

Address: Operational Base + offset (0x0080)

Reference picture numbers for index 4 and 5 (H264 VLC)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_refer5_nbr Number for reference picture index 5
15:0	RW	0x0000	sw_refer4_nbr Number for reference picture index 4

**VDPU\_SWREG33**

Address: Operational Base + offset (0x0084)

Reference picture numbers for index 6 and 7 (H264 VLC)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_refer7_nbr Number for reference picture index 7
15:0	RW	0x0000	sw_refer6_nbr Number for reference picture index 6

**VDPU\_SWREG34**

Address: Operational Base + offset (0x0088)

Reference picture numbers for index 8 and 9 (H264 VLC)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_refer9_nbr Number for reference picture index 9
15:0	RW	0x0000	sw_refer8_nbr Number for reference picture index 8

**VDPU\_SWREG35\_JPEG\_ROI**

Address: Operational Base + offset (0x008c)

JPEG roi offset/dc base address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_jpegdcoff_base JPEG roi offset/dc base address JPEG roi offset/dc base address
1:0	RO	0x0	reserved

**VDPU\_SWREG35**

Address: Operational Base + offset (0x008c)

Reference picture numbers for index 10 and 11 (H264 VLC)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_refer11_nbr Number for reference picture index 11
15:0	RW	0x0000	sw_refer10_nbr Number for reference picture index 10

**VDPU\_SWREG36**

Address: Operational Base + offset (0x0090)

Reference picture numbers for index 12 and 13 (H264 VLC)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_refer13_nbr Number for reference picture index 13
15:0	RW	0x0000	sw_refer12_nbr Number for reference picture index 12

**VDPU\_SWREG36\_JPEG\_ROI**

Address: Operational Base + offset (0x0090)

JPEG roi offset/dc length

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:17	RO	0x0	reserved
16:0	RW	0x00000	sw_jpegdcoff_len sw_jpegdcoff_len The number of 64bit jpegdcoff, it can be used both when sw_roi_decode is 1'b0 or 1'b1

**VDPU\_SWREG37**

Address: Operational Base + offset (0x0094)

Reference picture numbers for index 14 and 15 (H264 VLC)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_refer15_nbr Number for reference picture index 15
15:0	RW	0x0000	sw_refer14_nbr Number for reference picture index 14

**VDPU\_SWREG38**

Address: Operational Base + offset (0x0098)

Reference picture long term flags (H264 VLC) prediction filt

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:22	RW	0x000	sw_pred_bc_tap_3_3 Prediction filter set 3, tap 3
21:12	RW	0x000	sw_pred_bc_tap_4_0 Prediction filter set 4, tap 0

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
11:2	RW	0x000	sw_perd_bc_tap_4_1 Prediction filter set 4, tap 1
1:0	RO	0x0	reserved

**VDPU\_SWREG38\_H264**

Address: Operational Base + offset (0x0098)

Reference picture numbers for index 12 and 13 (H264 VLC)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer_lterm_e long term flag for reference picture index [31:0]

**VDPU\_SWREG39**

Address: Operational Base + offset (0x009c)

Reference picture valid flags (H264 VLC) prediction filter ta

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:22	RW	0x000	sw_pred_bc_tap_4_2 Prediction filter set 4, tap 2
21:12	RW	0x000	sw_pred_bc_tap_4_3 Prediction filter set 4, tap 3
11:2	RW	0x000	sw_pred_bc_tap_5_0 Prediction filter set 5, tap 0
1:0	RO	0x0	reserved

**VDPU\_SWREG39\_H264**

Address: Operational Base + offset (0x009c)

Reference picture numbers for index 12 and 13 (H264 VLC)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_refer_valid_e valid flag for reference picture index [31:0]

**VDPU\_SWREG40**

Address: Operational Base + offset (0x00a0)

Base address for standard dependent tables

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_qtable_base Base address for standard dependent tables: JPEG= AC,DC, QP tables MPEG4=QP table base address if type 1 quantization is used MPEG2=QP table base address H.264=base address for various tables [note]:the h264 decoder will use these bits.
1:0	RO	0x0	reserved

**VDPU\_SWREG41**

Address: Operational Base + offset (0x00a4)

Base address for direct mode motion vectors

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_dir_mv_base Direct mode motion vector write/read base address. For H264 this is used only for direct mode motion vector write base. Progressive JPEG: ACDC coefficient read/write base address. If current round is for DC components this base address is pointing to luminance (separate base addresses for chrominances), for AC component rounds this base is used for current type
1:0	RO	0x0	reserved

**VDPU\_SWREG42**

Address: Operational Base + offset (0x00a8)

bi\_dir initial ref pic list register (0-2)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:25	RW	0x00	sw_binit_rlist_b2 Initial reference picture list for bi- direct backward picid 2
24:20	RW	0x00	sw_binit_rlist_f2 Initial reference picture list for bi- direct forward picid 2
19:15	RW	0x00	sw_binit_rlist_b1 Initial reference picture list for bi- direct backward picid 1
14:10	RW	0x00	sw_binit_rlist_f1 Initial reference picture list for bi- direct forward picid 1
9:5	RW	0x00	sw_binit_rlist_b0 Initial reference picture list for bi- direct backward picid 0
4:0	RW	0x00	sw_binit_rlist_f0 Initial reference picture list for bi- direct forward picid 0

**VDPU\_SWREG43**

Address: Operational Base + offset (0x00ac)

bi-dir initial ref pic list register (3-5)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:25	RW	0x00	sw_binit_rlist_b5 Initial reference picture list for bi- direct backward picid 5
24:20	RW	0x00	sw_binit_rlist_f5 Initial reference picture list for bi- direct forward picid 5
19:15	RW	0x00	sw_binit_rlist_b4 Initial reference picture list for bi- direct backward picid 4
14:10	RW	0x00	sw_binit_rlist_f4 Initial reference picture list for bi- direct forward picid 4
9:5	RW	0x00	sw_binit_rlist_b3 Initial reference picture list for bi- direct backward picid 3

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
4:0	RW	0x00	sw_binit_rlist_f3 Initial reference picture list for bi- direct forward picid 3

**VDPU\_SWREG44**

Address: Operational Base + offset (0x00b0)

bi-dir initial ref pic list register (6-8)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:25	RW	0x00	sw_binit_rlist_b8 Initial reference picture list for bi- direct backward picid 8
24:20	RW	0x00	sw_binit_rlist_f8 Initial reference picture list for bi- direct forward picid 8
19:15	RW	0x00	sw_binit_rlist_b7 Initial reference picture list for bi- direct backward picid 7
14:10	RW	0x00	sw_binit_rlist_f7 Initial reference picture list for bi- direct forward picid 7
9:5	RW	0x00	sw_binit_rlist_b6 Initial reference picture list for bi- direct backward picid 6
4:0	RW	0x00	sw_binit_rlist_f6 Initial reference picture list for bi- direct forward picid 6

**VDPU\_SWREG45**

Address: Operational Base + offset (0x00b4)

bi-dir initial ref pic list register (9- 11)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:25	RW	0x00	sw_binit_rlist_b11 Initial reference picture list for bi- direct backward picid 11
24:20	RW	0x00	sw_binit_rlist_f11 Initial reference picture list for bi- direct forward picid 11
19:15	RW	0x00	sw_binit_rlist_b10 Initial reference picture list for bi- direct backward picid 10
14:10	RW	0x00	sw_binit_rlist_f10 Initial reference picture list for bi- direct forward picid 10
9:5	RW	0x00	sw_binit_rlist_b9 Initial reference picture list for bi- direct backward picid 9
4:0	RW	0x00	sw_binit_rlist_f9 Initial reference picture list for bi- direct forward picid 9

**VDPU\_SWREG46**

Address: Operational Base + offset (0x00b8)

bi-dir initial ref pic list register (12- 14)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
29:25	RW	0x00	sw_binit_rlist_b14 Initial reference picture list for bi- direct backward picid 14
24:20	RW	0x00	sw_binit_rlist_f14 Initial reference picture list for bi- direct forward picid 14
19:15	RW	0x00	sw_binit_rlist_b13 Initial reference picture list for bi- direct backward picid 13
14:10	RW	0x00	sw_binit_rlist_f13 Initial reference picture list for bi- direct forward picid 13
9:5	RW	0x00	sw_binit_rlist_b12 Initial reference picture list for bi- direct backward picid 12
4:0	RW	0x00	sw_binit_rlist_f12 Initial reference picture list for bi- direct forward picid 12

**VDPU\_SWREG47**

Address: Operational Base + offset (0x00bc)

bi-dir and P fwd initial ref pic list register (15 and P 0-3)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:25	RW	0x00	sw_pinit_rlist_f3 Initial reference picture list for P forward picid 3
24:20	RW	0x00	sw_pinit_rlist_f2 Initial reference picture list for P forward picid 2
19:15	RW	0x00	sw_pinit_rlist_f1 Initial reference picture list for P forward picid 1
14:10	RW	0x00	sw_pinit_rlist_f0 Initial reference picture list for P forward picid 0
9:5	RW	0x00	sw_binit_rlist_b15 Initial reference picture list for bi- direct backward picid 15
4:0	RW	0x00	sw_binit_rlist_f15 Initial reference picture list for bi- direct forward picid 15

**VDPU\_SWREG48**

Address: Operational Base + offset (0x00c0)

Error concealment register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:23	RW	0x000	sw_startmb_x Start MB from SW for X dimension. Used in error concealment case [note]:the h264 decoder will use these bits.
22:15	RW	0x00	sw_startmb_y Start MB from SW for Y dimension. Used in error concealment case [note]:the h264 decoder will use these bits.
14:0	RO	0x0	reserved

**VDPU\_SWREG49**

Address: Operational Base + offset (0x00c4)

Prediction filter tap register for H264, MPEG4, VC1

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:22	RW	0x000	sw_pred_bc_tap_0_0 Prediction filter set 0, tap 0 [note]:the h264 decoder will use these bits.
21:12	RW	0x000	sw_pred_bc_tap_0_1 Prediction filter set 0, tap 1 [note]:the h264 decoder will use these bits.
11:2	RW	0x000	sw_pred_bc_tap_0_2 Prediction filter set 0, tap 2 [note]:the h264 decoder will use these bits.
1:0	RO	0x0	reserved

**VDPU\_SWREG50**

Address: Operational Base + offset (0x00c8)

Synthesis configuration register decoder 0 (read only)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RO	0x1	SW_DEC_MPEG2_PROF Decoding format support, MPEG-2 / MPEG-1 '0' = not supported '1' = supported
30:29	RO	0x3	SW_DEC_VC1_PROF Decoding format support, VC-1 0 = not supported 1 = supported up to simple profile 2 = supported up to main profile 3 = supported up to advanced profile
28	RO	0x1	SW_DEC_JPEG_PROF Decoding format support, JPEG 0 = not supported 1 = supported
27:26	RO	0x2	SW_DEC_MPEG4_PROF Decoding format support, MPEG-4 / H.263 0 = not supported 1 = supported up to simple profile 2 = supported up to advanced simple profile
25:24	RO	0x3	SW_DEC_H264_PROF Decoding format support, H.264 0 = not supported 1 = supported up to baseline profile 2 = supported up to high profile labeled stream with restricted high profile tools [note]:the h264 decoder will use these bits.
23	RO	0x1	reserved

Bit	Attr	Reset Value	Description
22	RO	0x0	SW_DEC_PJEPAGE_EXIT Progressive JPEG support: '0' = Not supported '1' = supported
21	RO	0x1	SW_DEC_OBUFF_LEVEL Decoder output buffer level: '0' = 1 MB buffering is used '1' = 4 MB buffering is used [note]:the h264 decoder will use these bits.
20	RO	0x1	SW_REF_BUFF_EXIST [note]:the h264 decoder will use these bits.
19:16	RO	0x5	SW_DEC_BUS_STRD [note]:the h264 decoder will use these bits.
15:14	RO	0x1	SW_DEC_SYNTH_LAN [note]:the h264 decoder will use these bits.
13:12	RO	0x2	SW_DEC_BUS_WIDTH 0 = error 1 = 32 bit bus 2 = 64 bit bus 3 = 128 bit bus [note]:the h264 decoder will use these bits.
11	RO	0x1	reserved
10:0	RO	0x780	SW_DEC_MAX_OWIDTH Max configured decoder video resolution that can be decoded. Informed as width of the picture in pixels [note]:the h264 decoder will use these bits.

**VDPU\_SWREG51**

Address: Operational Base + offset (0x00cc)

Reference picture buffer control register

Bit	Attr	Reset Value	Description
31	RW	0x0	sw_refbu_e Refer picture buffer enable: '0' = refer picture buffer disabled '1' = refer picture buffer enabled. Valid if picture size is QVGA or more [note]:the h264 decoder will use these bits.
30:19	RW	0x000	sw_refbu_thr Reference buffer disable threshold value (cache miss amount). Used to buffer shut down (if more misses than allowed) [note]:the h264 decoder will use these bits.
18:14	RW	0x00	sw_refbu_picid The used reference picture ID for reference buffer usage [note]:the h264 decoder will use these bits.

Bit	Attr	Reset Value	Description
13	RW	0x0	sw_refbu_eval_e Enable for HW internal reference ID calculation. If given threshold level is reached by any picture_id after first MB row, that picture_id is used for reference buffer fill for rest of the picture [note]:the h264 decoder will use these bits.
12	RW	0x0	sw_refbu_fparmod_e Field parity mode enable. Used in rebufferd evaluation mode '0' = use the result field of the evaluation '1' = use the parity mode field [note]:the h264 decoder will use these bits.
11:9	RO	0x0	reserved
8:0	RW	0x000	sw_refbu_y_offset Y offset for rebufferd. This coordinate is used to compensate the global motion of the video for better buffer hit rate [note]:the h264 decoder will use these bits.

**VDPU\_SWREG52**

Address: Operational Base + offset (0x00d0)

Reference picture buffer information register 1 (read only)

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	sw_refbu_hit_sum The sum of the rebufferd hits of the picture. Determined for each 8x8 luminance partition of the picture. The proceeding of the HW calculation can be read during HW decoding [note]:the h264 decoder will use these bits.
15:0	RW	0x0000	sw_refbu_intra_sum The sum of the luminance 8x8 intra partitions of the picture. The proceeding of the HW calculation can be read during HW decoding [note]:the h264 decoder will use these bits.

**VDPU\_SWREG53**

Address: Operational Base + offset (0x00d4)

Reference picture buffer information register 2 (read only)

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21:0	RW	0x0000000	sw_refbu_y_mv_sum The sum of the decoded motion vector y-components of the picture. The first luminance motion vector of each MB is used in calculation. Other motion vectors of the MB are discarded. Each motion vector is saturated between -256 - 255 before calculation. The proceeding of the HW calculation can be read during HW decoding [note]:the h264 decoder will use these bits.

**VDPU\_SWREG54**

Address: Operational Base + offset (0x00d8)

Synthesis configuration register decoder 1 (read only)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RO	0x1	SW_DEC_JPEG_EXTENS JPEG sampling support extension for 411 and 444 samplings and support for bigger max resolution than 16 Mpix (up to 67Mpixels): '0' = not supported '1' = supported
30	RO	0x1	SW_DEC_REFBU_ILACE Refbufferd support for interlaced content: '0' = not supported '1' = supported [note]:the h264 decoder will use these bits.
29	RO	0x1	reserved
28	RO	0x0	SW_REF_BUFF2_EXIST Reference picture buffer 2 usage: '0' = not supported '1' = reference buffer 2 is used [note]:the h264 decoder will use these bits.
27:26	RO	0x1	reserved
25	RO	0x0	SW_DEC_RTL_ROM ROM implementation type (If design includes ROMs) '0': ROMs are implemented from actual ROM units '1': ROMs are impelemted from RTL
24	RO	0x1	reserved
23	RO	0x1	reserved
22	RO	0x1	SW_DEC_AVG_PROF Decoding format support, AVS 0 = not supported 1 = supported
21:20	RO	0x1	SW_DEC_MVC_PROF Decoding format support, MVC 0 = not supported 1 = supported
19	RO	0x1	reserved
18:17	RO	0x1	SW_DEC_TILED_L Tiled mode support level 0 = not supported 1 = supported with 8x4 tile size 2,3 = reserved [note]:the h264 decoder will use these bits.
16:0	RO	0x0	reserved

**VDPU\_SWREG55**

Address: Operational Base + offset (0x00dc)

Reference picture buffer 2 / Advanced prefetch control register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RW	0x0	sw_refbu2_buf_e Refer picture buffer 2 enable: '0' = refer picture buffer disabled '1' = refer picture buffer enabled. Valid if picture size is QVGA or more (can be turned off by HW if threshold value reached) [note]:the h264 decoder will use these bits.
30:19	RW	0x000	sw_refbu2_thr Reference buffer disable threshold value (buffer miss amount). Used to buffer shut down (if more misses than allowed) [note]:the h264 decoder will use these bits.
18:14	RW	0x00	sw_refbu2_picid The used reference picture ID for reference buffer usage [note]:the h264 decoder will use these bits.
13:0	RW	0x0000	sw_apf_threshold Advanced prefetch threshold value. If current MB exceeds the threshold the advanced mode is not used. Value 0 disables threshold usage and advanced prefetch usage is restricted by internal memory limitation only [note]:the h264 decoder will use these bits.

**VDPU\_SWREG56**

Address: Operational Base + offset (0x00e0)

Reference buffer information register 3 (read only)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_refbu_top_sum The sum of the top partitions of the picture [note]:the h264 decoder will use these bits.
15:0	RW	0x0000	sw_refbu_bot_sum The sum of the bottom partitions of the picture [note]:the h264 decoder will use these bits.

**VDPU\_SWREG57\_INTRA\_INTER**

Address: Operational Base + offset (0x00e4)

intra\_dll3t,intra\_dblspeed,inter\_dblspeed,stream\_len\_hi

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:15	RO	0x0	reserved
14:8	RO	0x00	debug_service debug_service signals service_wr[2:0], service_rd[3:0]

Bit	Attr	Reset Value	Description
7	RW	0x0	sw_cache_en cache enable 1'b1: cache enable 1'b0: cache disable when sw_cache_en is 1'b1, sw_pref_sigchan should also be 1'b1
6	RW	0x0	sw_pref_sigchan prefetch single channel enable 1'b1: prefetch single channel enable
5	RW	0x0	sw_axiwr_sel axi write master select 1'b0: auto sel encoder axi signals and decoder axi signals 1'b1: sel decoder axi signals (it only use to set bu_dec_e to 1'b0 in the middle of a frame)
4	RW	0x0	sw_parallel_bus parallel_bus enable when it is set to 1'b1, the axi support read and write service parallel; when it is set to 1'b0, the axi only support read and write serial
3	RW	0x0	sw_intra_dbl3t sw_intra_dbl3t In chroma dc intra prediction, when this bit is enable, there will 3 cycle enhance for every block
2	RW	0x0	sw_intra_dblspeed intra double speed enable Intra double speed enable
1	RW	0x0	sw_inter_dblspeed inter double speed enable Inter double speed enable
0	RW	0x0	sw_stream_len_hi stream length high bit The extension bit of sw_stream_len

**VDPU\_SWREG57**

Address: Operational Base + offset (0x00e4)

intra\_dbl3t,intra\_dblspeed,inter\_dblspeed,stream\_len\_hi

Bit	Attr	Reset Value	Description
31	RW	0x0	fuse_dec_h264 1 = H.264 enabled
30	RW	0x0	fuse_dec_mpeg4 1 = MPEG-4/H.263 enabled
29	RW	0x0	fuse_dec_mpeg2 1 = MPEG-2/MPEG-1 enabled N
28	RW	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
27	RW	0x0	fuse_dec_jpeg Field0000 Abstract 1 = JPEG enabled
26	RW	0x0	reserved
25	RW	0x0	fuse_dec_vc1 1 = VC1 enabled
24	RW	0x0	fuse_dec_pjpeg 1 = Progressive JPEG enabled (Requires also JPEG to be enabled)
23	RW	0x0	reserved
22	RW	0x0	reserved
21	RW	0x0	reserved
20	RW	0x0	reserved
19	RW	0x0	reserved
18	RW	0x0	fuse_dec_mvc enabled (requires also H264 to be enabled)
17:16	RO	0x0	reserved
15	RW	0x0	fuse_dec_maxw_1920 1 = Max video width up to 1920 pixels enabled. Priority coded with priority 1.
14	RW	0x0	fuse_dec_maxw_1280 1 = Max video width up to 1280 pixels enabled. Priority coded with priority 2.
13	RW	0x0	fuse_dec_maxw_720 1 = Max video width up to 720 pixels enabled. Priority coded with priority 3.
12	RW	0x0	fuse_dec_maxw_352 1 = Max video width up to 352 pixels enabled. Priority coded with priority 4
11:8	RO	0x0	reserved
7	RW	0x0	fuse_dec_refbuffer 1 = reference buffer used
6:0	RO	0x0	reserved

**VDPU\_SWREG58**

Address: Operational Base + offset (0x00e8)

Decoder debug register 0 (read only)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RO	0x0	reserved
30	RO	0x0	debug_mv_req mvst_mv_req signal value
29	RO	0x0	debug_rlc_req prtr_res_y_req signal value
28	RO	0x0	debug_res_y_req prtr_res_y_req signal value

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
27	RO	0x0	debug_res_c_req prtr_res_c_req signal value
26	RO	0x0	debug_strm_da_e strm_da_e signal value
25	RO	0x0	debug_framerdy dfbu_framerdy signal value
24	RO	0x0	debug_filter_req dfbu_req_e signal value
23	RO	0x0	debug_referreq0 prbu_referreq0 signal value
22	RO	0x0	debug_referreq1 prbu_referreq1 signal value
21	RO	0x0	reserved
20:0	RO	0x0000000	debug_dec_mb_count HW internal MB counter value

**VDPU\_SWREG59**

Address: Operational Base + offset (0x00ec)  
H264 Chrominance 8 pixel interleaved data base

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_dec_ch8pix_base Base address for additional chrominance data format where chrominance is interleaved in group of 8 pixels. The usage is enabled by sw_ch_8pix_ilieav_e [note]:the h264 decoder will use these bits.
1:0	RO	0x0	reserved

**VDPU\_SWREG60**

Address: Operational Base + offset (0x00f0)

Interrupt register post-processor

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:14	RO	0x0	reserved
13	RW	0x0	sw_pp_bus_int Interrupt status bit bus. Error response from bus. In pipeline mode this bit is not used
12	RW	0x0	sw_pp_rdy_int Interrupt status bit pp. When this bit is high post processor has processed a picture in external mode. In pipeline mode this bit is not used.
11:9	RO	0x0	reserved
8	RW	0x0	sw_pp_irq Post-processor IRQ. SW will reset this after interrupt is handled. HINTpp is not used for pp if IRQ disable pp is high (sw_pp_irq_n_e = 1). In pipeline mode this bit is not used
7:5	RO	0x0	reserved

Bit	Attr	Reset Value	Description
4	RW	0x0	sw_pp_irq_dis Post-processor IRQ disable. When high, there are no interrupts from HW concerning post processing. Polling must be used to see the interrupt
3:2	RO	0x0	reserved
1	RW	0x0	sw_pp_pipeline_e Decoder –post-processing pipeline enable: 0 = Post-processing is processing different picture than decoder or is disabled 1 = Post-processing is performed in pipeline with decoder
0	RW	0x0	sw_pp_e External mode post-processing enable. This bit will start the post-processing operation. Not to be used if PP is in pipeline with decoder (sw_pp_pipeline_e = 1). HW will reset this when picture is post-processed.

**VDPU\_SWREG61**

Address: Operational Base + offset (0x00f4)

Device configuration register post-processor

Bit	Attr	Reset Value	Description
31:24	RW	0x01	sw_pp_axi_rd_id Read ID used for AXI PP read services (if connected to AXI)
23:16	RW	0x01	sw_pp_axi_wr_id Write ID used for AXI PP write services (if connected to AXI)
15	RO	0x0	reserved
14	RW	0x0	sw_pp_scmd_dis AXI Single Command Multiple Data disable.
13	RW	0x0	sw_pp_in_a2_endsel Endian/swap select for Alpha blend input source 2: '0' = Use PP in endian/swap definitions (sw_pp_in_endian, sw_pp_in_swap) '1' = Use Ablend source 1 endian/swap definitions
12	RW	0x0	sw_pp_in_a1_swap32 Alpha blend source 1 input 32bit data swap (may be used for 64 bit environment): 0 = no swapping of 32 bit words 1 = 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled))
11	RW	0x0	sw_pp_in_a1_endian Alpha blend source 1 input data byte endian mode. 0 = Big endian (0-1-2-3 order) 1 = Little endian (3-2-1-0 order)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
10	RW	0x0	sw_pp_in_swap32_e PP input 32bit data swap (may be used for 64 bit environment): 0 = no swapping of 32 bit words 1 = 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled))
9	RW	0x0	sw_pp_data_disc_e PP data discard enable. Precise burst lengths are used with reading services. Extra data is discarded internally.
8	RW	0x1	sw_pp_clkgate_e PP dynamic clock gating enable: 1 = Clock is gated from PP structures that are not used 0 = Clock is running for all PP structures Note: Clock gating value can be changed only when PP is not enabled
7	RW	0x0	sw_pp_in_endian PP input picture byte endian mode. Used only if PP is in standalone mode. If PP is running pipelined with the decoder, this bit has no effect. 0 = Big endian (0-1-2-3 order) 1 = Little endian (3-2-1-0 order)
6	RW	0x0	sw_pp_out_endian PP output picture endian mode for YCbCr data or for any data if config value SW_PP_OEN_VERSION=1 0 = Big endian (0-1-2-3 order) 1 = Little endian (3-2-1-0 order) (NOTE: For SW_PP_OEN_VERSION=0 16 bit RGB data this bit works as pixel swapping bit. For 32 bit RGB this bit has no meaning)
5	RW	0x0	sw_pp_out_swap32_e PP output data word swap (may be used for 64 bit environment): 0 = no swapping of 32 bit words 1 = 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order (also little endian should be enabled))
4:0	RW	0x00	sw_pp_max_burst Maximum burst length for PP bus transactions. 1-16

**VDPU\_SWREG62**

Address: Operational Base + offset (0x00f8)

Deinterlace control register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RW	0x0	sw_deint_e De-interlace enable. Input data is in interlaced format and deinterlacing needs to be performed
30	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
29:16	RW	0x0000	sw_deint_threshold Threshold value used in deinterlacing
15	RW	0x0	sw_deint_blend_e Blend enable for de-interlacing
14:0	RW	0x0000	sw_deint_edge_det Edge detect value used for deinterlacing

**VDPU\_SWREG63**

Address: Operational Base + offset (0x00fc)

base address for reading post-processing input picture uminan

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_pp_in_lu_base Base address for post-processing input luminance picture. If PP input picture is fetched from fields this base address is used to point to topfield of the picture. Used in external mode only.
1:0	RO	0x0	reserved

**VDPU\_SWREG64**

Address: Operational Base + offset (0x0100)

Base address for reading post-processing input picture Ch/Ch

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_pp_in_cb_base Base address for post-processing input Cb picture or for both chrominances interleaved). If PP input picture is fetched from fields this base address is used to point to topfield of the picture. Used in external mode only
1:0	RO	0x0	reserved

**VDPU\_SWREG65**

Address: Operational Base + offset (0x0104)

Base address for reading post-processing input picture Cr

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_pp_in_cr_base Base address for post-processing input cr picture. Used in external mode only
1:0	RO	0x0	reserved

**VDPU\_SWREG66**

Address: Operational Base + offset (0x0108)

Base address for writing post-processed picture luminance/RGB

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_pp_out_lu_base Base address for post-processing output picture (luminance/YUYV/RGB).

**VDPU\_SWREG67**

Address: Operational Base + offset (0x010c)

Base address for writing post-processed picture Ch

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_pp_out_ch_base Base address for post-processing output chrominance picture (interleaved chrominance).

**VDPU\_SWREG68**

Address: Operational Base + offset (0x0110)

Register for contrast adjusting

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RW	0x00	sw_contrast_thr1 Threshold value 1, used with contrast adjusting
23:20	RO	0x0	reserved
19:10	RW	0x000	sw_contrast_off2 Offset value 2, used with contrast adjusting
9:0	RW	0x000	sw_contrast_off1 Offset value 1, used with contrast adjusting

**VDPU\_SWREG69**

Address: Operational Base + offset (0x0114)

Register for colour conversion and contrast adjusting

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RW	0x0	sw_pp_in_start_ch For YUYV 422 input format. Enable for start_with_chrominance. '0' = the order is Y0CbY0Cr or Y0CrY0Cb '1' = the order is CbY0CrY0 or CrY0CbY0
30	RW	0x0	sw_pp_in_cr_first For YUYV 422 input format and YCbCr 420 semiplanar format. Enable for Cr first (before Cb) '0' = the order is Y0CbY0Cr or CbY0CrY0 (if 420 semiplanar chrominance: CbCrCbCr) '1' = the order is Y0CrY0Cb or CrY0CbY0 (if 420 semiplanar chrominance: CrCbCrCb)
29	RW	0x0	sw_pp_out_start_ch For YUYV 422 output format. Enable for start_with_chrominance. '0' = the order is Y0CbY0Cr or Y0CrY0Cb '1' = the order is CbY0CrY0 or CrY0CbY0
28	RW	0x0	sw_pp_out_cr_first For YUYV 422 output format. Enable for Cr first (before Cb) '0' = the order is Y0CbY0Cr or CbY0CrY0 '1' = the order is Y0CrY0Cb or CrY0CbY0
27:18	RW	0x000	sw_color_coeffa2 Coefficient a2, used with Y pixel to calculate all color components

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
17:8	RW	0x000	sw_color_coeffa1 Coefficient a1, used with Y pixel to calculate all color components
7:0	RW	0x00	sw_contrast_thr2 Threshold value 2, used with contrast adjusting

**VDPU\_SWREG70**

Address: Operational Base + offset (0x0118)

Register for colour conversion 0

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:20	RW	0x000	sw_color_coeffd Coefficient d, used with Cb to calculate green component value
19:10	RW	0x000	sw_color_coeffc Coefficient c, used with Cr to calculate green component value
9:0	RW	0x000	sw_color_coeffb Coefficient b, used with Cr to calculate red component value

**VDPU\_SWREG71**

Address: Operational Base + offset (0x011c)

Register for colour conversion 1 + rotation mode

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:21	RW	0x000	sw_crop_startx Start coordinate x for the cropped area in macroblocks.
20:18	RW	0x0	sw_rotation_mode Rotation mode: 000 = rotation disabled 001 = rotate + 90 010 = rotate -90 011 = horizontal flip (mirror) 100 = vertical flip 101 = rotate 180
17:10	RW	0x00	sw_color_coefff Coefficient f, used with Y to adjust brightness
9:0	RW	0x000	sw_color_coeffe Coefficient e, used with Cb to calculate blue component value

**VDPU\_SWREG72**

Address: Operational Base + offset (0x0120)

PP input size and -cropping register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RW	0x00	sw_crop_starty Start coordinate y for the cropped area in macroblocks.
23	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
22:18	RW	0x00	sw_rangemap_coef_y Range map value for Y component (RANGE_MAPY+9 in VC-1 standard)
17	RO	0x0	reserved
16:9	RW	0x00	sw_pp_in_height PP input picture height in MBs. Can be cropped from a bigger input picture in external mode
8:0	RW	0x000	sw_pp_in_width PP input picture width in MBs. Can be cropped from a bigger input picture in external mode

**VDPU\_SWREG73**

Address: Operational Base + offset (0x0124)

PP input picture base address for Y bottom field

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_pp_bot_yin_base PP input Y base for bottom field
1:0	RO	0x0	reserved

**VDPU\_SWREG74**

Address: Operational Base + offset (0x0128)

PP input picture base for Ch bottom field

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RW	0x00000000	sw_pp_bot_cin_base PP input C base for bottom field (mixed chrominance)
1:0	RO	0x0	reserved

**VDPU\_SWREG79**

Address: Operational Base + offset (0x013c)

Scaling ratio register 1 &amp; padding for B

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RW	0x0	sw_rangemap_y_e Range map enable for Y component (RANGE_MAPY_FLAG in VC-1 standard). For VC1 main profile this bit is used as range expansion enable
30	RW	0x0	sw_rangemap_c_e Range map enable for chrominance component RANGE_MAPUV_FLAG in VC-1 standard)
29	RW	0x0	sw_ycbcr_range Defines the YCbCr range in RGB conversion: 0 = 16 --> 235 for Y, 16 --> 240 for Chrominance 1 = 0 --> 255 for all components

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
28	RW	0x0	sw_rgb_pix_in32 RGB pixel amount/ 32 bit word 0 = 1 RGB pixel/32 bit 1 = 2 RGB pixels/32 bit
27:23	RW	0x00	sw_rgb_r_padd Amount of ones that will be padded in front of the R-component
22:18	RW	0x00	sw_rgb_g_padd Amount of ones that will be padded in front of the G-component
17:0	RW	0x00000	sw_scale_wratio Scaling ratio for width (outputw-1/inputw-1)

**VDPU\_SWREG80**

Address: Operational Base + offset (0x0140)

Scaling register 0 ratio &amp; padding for R and G

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RO	0x0	reserved
30	RW	0x0	sw_pp_fast_scale_e 0 = fast downscaling is not enabled 1 = fast downscaling is enabled. The quality of the picture is decreased but performance is improved
29:27	RW	0x0	sw_pp_in_struct PP input data picture structure: 0 = Top field / progressive frame structure: Read input data from top field base address /frame base address and read every line 1 = Bottom field structure: Read input data from bottom field base address and read every line. 2 = Interlaced field structure: Read input data from both top and bottom field base address and take every line from each field. 3 = Interlaced frame structure: Read input data from both top and bottom field base address and take every second line from each field. 4 = Ripped top field structure: Read input data from top field base address and read every second line. 5 = Ripped bottom field structure: Read input data from bottom field base address and read every second line
26:25	RW	0x0	sw_hor_scale_mode Horizontal scaling mode: 00 = Off 01 = Upscale 10 = Downscale

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
24:23	RW	0x0	sw_ver_scale_mode Vertical scaling mode: 00 = Off 01 = Upscale 10 = Downscale
22:18	RW	0x00	sw_rgb_b_padd Amount of ones that will be padded in front of the B-component
17:0	RW	0x00000	sw_scale_hratio Scaling ratio for height (outpuh-1/inpuh-1)

**VDPU\_SWREG81**

Address: Operational Base + offset (0x0144)

Scaling ratio register 2

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	sw_wscale_invra Inverse scaling ratio for width, or ch (inputw-1 / outputw-1)
15:0	RW	0x0000	sw_hscale_invra Inverse scaling ratio for height or cv (inpuh-1 / outpuh-1)

**VDPU\_SWREG82**

Address: Operational Base + offset (0x0148)

Rmask register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_r_mask Bit mask for R component (and alpha channel)

**VDPU\_SWREG83**

Address: Operational Base + offset (0x014c)

Gmask register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_g_mask Bit mask for G component (and alpha channel)

**VDPU\_SWREG84**

Address: Operational Base + offset (0x0150)

Bmask register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_b_mask Bit mask for B component (and alpha channel)

**VDPU\_SWREG85**

Address: Operational Base + offset (0x0154)

Post-processor control register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

Bit	Attr	Reset Value	Description
31:29	RW	0x0	<p>sw_pp_in_format  PP input picture data format  0 = YUYV 4:2:2 interleaved (supported only in external mode)  1 = YCbCr 4:2:0 Semi-planar in linear raster-scan format  2 = YCbCr 4:2:0 planar (supported only in external mode)  3 = YCbCr 4:0:0 (supported only in pipelined mode)  4 = YCbCr 4:2:2 Semi-planar (supported only in pipelined mode)  5 = YCbCr 4:2:0 Semi-planar in tiled format (supported only in external mode (8170 decoder only))  6 = YCbCr 4:4:0 Semi-planar (supported only in pipelined mode, possible for jpeg only)  7 = Escape pp input data format. Defined in swreg86</p>
28:26	RW	0x0	<p>sw_pp_out_format  PP output picture data format:  0 = RGB  1 = YCbCr 4:2:0 planar (Not supported)  2 = YCbCr 4:2:2 planar (Not supported)  3 = YUYV 4:2:2 interleaved  4 = YCbCr 4:4:4 planar (Not supported)  5 = YCh 4:2:0 chrominance interleaved  6 = YCh 4:2:2 (Not supported)  7 = YCh 4:4:4 (Not supported)</p>
25:15	RW	0x000	<p>sw_pp_out_height  Scaled picture height in pixels (Must be dividable by 2 or by any if Pixel Accurate PP output configuration is enabled) Max scaled picture height is 1920 pixels or maximum three times the input source height minus 8 pixels</p>
14:4	RW	0x000	<p>sw_pp_out_width  Scaled picture width in pixels. Must be dividable by 8 or by any if Pixel Accurate PP output configuration is enabled. Max scaled picture width is 1920 pixels or maximum three times the input source width minus 8 pixels</p>
3	RW	0x0	<p>sw_pp_out_tiled_e  Tiled mode enable for PP output. Can be used only for YCbYCr 422 output format. Can be used only if correpongind configuration supports this feature. Tile size is 4x4 pixels.</p>
2	RW	0x0	<p>sw_pp_out_swap16_e  PP output swap 16 swaps 16 bit halves inside of 32 bit word. Can be used for 16 bit RGB to change pixel orders but is valid also for any output format  NOTE: requires that configuration of SW_PPD_OEN_VERSION=1</p>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
1	RW	0x0	sw_pp_crop8_r_e PP input picture width is not 16 pixels multiple. Only 8 pixels of the most right MB of the unrotated input picture is used for PP input.
0	RW	0x0	sw_pp_crop8_d_e PP input picture height is not 16 pixels multiple. Only 8 pixel rows of the most down MB of the unrotated input picture is used for PP input.

**VDPU\_SWREG86**

Address: Operational Base + offset (0x0158)

Mask 1 start coordinate register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RW	0x0	sw_pp_in_format_es Escape PP in format. Used if sw_pp_in_format is defined to 7: 0 0 = YCbCr 4:4:4 1 = YCbCr 4:1:1
28	RO	0x0	reserved
27:23	RW	0x00	sw_rangemap_coef_c Range map value for chrominance component (RANGE_MAPUV+9 in VC-1 standard)
22	RW	0x0	sw_mask1_ablend_e Mask 1 alpha blending enable. Instead of masking the output picture the alpha blending is performed. Alpha blending source can be found from alpha blend 1 base address. Alpha blending can be enabled only for RGB/ YUYV 422 data.
21:11	RW	0x000	sw_mask1_starty Vertical start pixel for mask area 1. Defines the y coordinate. Coordinate 0,0 means the up-left corner in PP output luminance picture. See Table 47 for restrictions
10:0	RW	0x000	sw_mask1_startix Horizontal start pixel for mask area 1. Defines the x coordinate. Coordinate 0,0 means the up-left corner in PP output luminance picture. See Table 47 for restrictions

**VDPU\_SWREG87**

Address: Operational Base + offset (0x015c)

Mask 2 start coordinate register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:23	RO	0x0	reserved
22	RW	0x0	sw_mask2_ablend_e Mask 2 alpha blending enable. Instead of masking the output picture the alpha blending is performed. Alpha blending source can be found from alpha blend 2 base address. Alpha blending can be enabled only for RGB/YUYV 422 data.

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
21:11	RW	0x000	sw_mask_starty Vertical start pixel for mask area 2. Defines the y coordinate. Coordinate 0,0 means the up-left corner in PP output Y picture. See Table 47 for restrictions
10:0	RW	0x000	sw_mask2_startx Horizontal start pixel for mask area 2. Defines the x coordinate. Coordinate 0,0 means the up-left corner in PP output Y picture. See Table 47 for restrictions

**VDPU\_SWREG88**

Address: Operational Base + offset (0x0160)

Mask 1 size and PP original width register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:23	RW	0x000	sw_ext_orig_width PP input picture original width in macro blocks.
22	RW	0x0	sw_mask1_e Mask 1 enable. If mask 1 is used this bit is high
21:11	RW	0x000	sw_mask1_endy Mask 1 end coordinate y in pixels (inside of PPD output picture). Range must be between [Mask1StartCoordinateY, ScaledHeight].
10:0	RW	0x000	sw_mask1_endx Mask 1 end coordinate x in pixels (inside of PPD output picture). Range must be between [Mask1StartCoordinateX, ScaledWidth]

**VDPU\_SWREG89**

Address: Operational Base + offset (0x0164)

Mask 2 size register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:23	RO	0x0	reserved
22	RW	0x0	sw_mask2_e Mask 2 enable. If mask 1 is used this bit is high
21:11	RW	0x000	sw_mask2_endy Mask 2 end coordinate y in pixels (inside of PP output picture). Range must be between [Mask2StartCoordinateY, ScaledHeight].
10:0	RW	0x000	sw_mask2_endx Mask 2 end coordinate x in pixels (inside of PP output picture). Range must be between [Mask2StartCoordinateX, ScaledWidth].

**VDPU\_SWREG90**

Address: Operational Base + offset (0x0168)

PiP register 0

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29	RW	0x0	sw_right_cross_e Right side overcross enable. 0 = No right side overcross, 1 = right side overcross
28	RW	0x0	sw_left_cross_e Left side overcross enable. 0 = No left side overcross, 1 = left side overcross
27	RW	0x0	sw_up_cross_e Upward overcross enable. 0 = No upward overcross, 1 = upward overcross
26	RW	0x0	sw_down_cross_e Downward overcross enable. 0 = No downward overcross, 1 = downward overcross
25:15	RW	0x000	sw_up_cross Amount of upward overcross (vertical pixels outside of display from the upper side). Range must be between [0, ScaledHeight].
14:11	RO	0x0	reserved
10:0	RW	0x000	sw_down_cross Amount of downward overcross (vertical pixels outside of display from the down side). Range must be between [0, ScaledHeight].

**VDPU\_SWREG91**

Address: Operational Base + offset (0x016c)

PiP register 1 and dithering control

Bit	Attr	Reset Value	Description
31:30	RW	0x0	sw_dither_select_r Dithering control for R channel: 00 = dithering disabled 01 = use four-bit dither matrix 10 = use five-bit dither matrix 11 = use six-bit dither matrix
29:28	RW	0x0	sw_dither_select_g Dithering control for G channel: 00 = dithering disabled 01 = use four-bit dither matrix 10 = use five-bit dither matrix 11 = use six-bit dither matrix
27:26	RW	0x0	sw_dither_select_b Dithering control for B channel: 00 = dithering disabled 01 = use four-bit dither matrix 10 = use five-bit dither matrix 11 = use six-bit dither matrix

Bit	Attr	Reset Value	Description
25:24	RO	0x0	reserved
23:22	RW	0x0	<p>sw_pp_tiled_mode Input data is in tiled mode (at the moment valid only for YCbCr 420 data, pipeline or external mode): 0 = Tiled mode not used 1 = Tiled mode enabled for 8x4 sized tiles 2,3 = reserved</p>
21:11	RW	0x000	<p>sw_right_cross Amount of right side overcross (Horizontal pixels outside of display from the right side). Range must be between [0, ScaledWidth].</p>
10:0	RW	0x000	<p>sw_left_cross Amount of left side overcross (Horizontal pixels outside of display from the left side). Range must be between [0, ScaledWidth].</p>

**VDPU\_SWREG92**

Address: Operational Base + offset (0x0170)

Display width and PP input size extension register

Bit	Attr	Reset Value	Description
31:29	RW	0x0	<p>sw_pp_in_h_ext Extended PP input height. Used with JPEG</p>
28:26	RW	0x0	<p>sw_pp_in_w_ext Extended PP input width. Used with JPEG</p>
25:23	RW	0x0	<p>sw_crop_start_x_ext Extended PP input crop start coordinate x. Used with JPEG</p>
22:20	RW	0x0	<p>sw_crop_start_y_ext Extended PP input crop start coordinate y. Used with JPEG</p>
19:12	RO	0x0	reserved
11:0	RW	0x000	<p>sw_display_width Width of the display in pixels. Max HDTV (1920)</p>

**VDPU\_SWREG93**

Address: Operational Base + offset (0x0174)

Display width and PP input size extension register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>sw_abledn1_base Base address for alpha blending input 1 (if mask1 is used in alpha blending mode). Format of data is 24 bit RGB/ YCbCr and endian/swap -mode is as in PP input. Amount of data is informed with mask 1 size or with ablend1_scanline if ablend cropping is supported in configuration.</p>

**VDPU\_SWREG94**

Address: Operational Base + offset (0x0178)  
 Base address for alpha blend 2 gui component

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_ablend2_base Base address for alpha blending input 2 (if mask2 is used in alpha blending mode). Format of data is 24 bit RGB/ YCbCr and endian/swap -mode is as in PP input. Amount of data is informed with mask 2 size or with ablend2_scanline if ablend cropping is supported in configuration.

**VDPU\_SWREG95**

Address: Operational Base + offset (0x017c)  
 Base address for alpha blend 2 gui component

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:26	RO	0x0	reserved
25:13	RW	0x0000	sw_ablend2_scan Scanline width in pixels for Ablend 2. Usage enabled if corresponding configuration bit is enabled
12:0	RW	0x0000	sw_ablend1_scan Scanline width in pixels for Ablend 1. Usage enabled if corresponding configuration bit is enabled

**VDPU\_SWREG98**

Address: Operational Base + offset (0x0188)  
 PP output width/height extension

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RW	0x0	sw_pp_out_h_ext sw_pp_out_h_ext PP output heightextension
0	RW	0x0	sw_pp_out_w_ext sw_pp_out_w_ext PP output widthextension

**VDPU\_SWREG99**

Address: Operational Base + offset (0x018c)  
 PP fuse register (read only)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RO	0x1	fuse_pp_pp 1 = PP enabled
30	RO	0x1	fuse_pp_deint 1 = Deinterlacing enabled
29	RO	0x1	fuse_pp_ablend 1 = Alpha Blending enabled
28:16	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
15	RO	0x1	fuse_pp_maxw_1920 1 = Max PP output width up to 1920 pixels enabled. Priority coded with priority 1
14	RO	0x1	fuse_pp_maxw_1280 1 = Max PP output width up to 1280 pixels enabled. Priority coded with priority 2
13	RO	0x1	fuse_pp_maxw_720 1 = Max PP output width up to 720 pixels enabled. Priority coded with priority 3
12	RO	0x1	fuse_pp_maxw_352 1 = Max PP output width up to 352 pixels enabled. Priority coded with priority 4
11:0	RO	0x0	reserved

**VDPU\_SWREG100**

Address: Operational Base + offset (0x0190)

Synthesis configuration register post-processor (read only)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RO	0x1	SW_ABLEND_CROP_E Alpha blending support for input cropping: '0' : Not supported. External memory must include the exact image of the area being alpha blended '1' : Supported. External memory can include a picture from blended area can be cropped. Requires usage of swreg95
30	RO	0x1	SW_PPD_PIXAC_E Pixel Accurate PP output mode exists: '0' = PIP, Scaling and masks can be adjusted by steps of 8 pixels (width) or 2 pixels (height) '1' = PIP, Scaling and masks can be adjusted by steps of 1 pixel for RGB and 2 pixels for subsampled chroma formats (by using bus specific write strobe functionality)
29	RO	0x1	SW_PPD_TILED_EXIST PP output YCbYCr 422 tiled support (4x4 pixel tiles) '0' = Not supported '1' = Supported
28	RO	0x1	SW_PPD_DITH_EXIST Dithering exists: '0' = no '1' = yes
27:26	RO	0x3	SW_PPD_SCALE_LEVEL Scaling support: 00 = No scaling 01 = Scaling with lo perfomance architecture 10 = Scaling with high performance architecture 11 = Scaling with high performance architecture + fast

Bit	Attr	Reset Value	Description
25	RO	0x1	SW_PPD_DEINT_EXIST De-interlacing exists: '0' = no '1' = yes
24	RO	0x1	SW_PPD_BLEND_EXIST Alpha blending exists: '0' = no '1' = yes
23	RO	0x1	SW_PPD_IBUFF_LEVEL PP input buffering level: '0' = 1 MB input buffering is used '1' = 4 MB input buffering is used
22:19	RO	0x0	reserved
18	RO	0x1	SW_PPD_OEN_VERSION PP output endian version: '0' = Endian mode supported for other than RGB '1' = Endian mode supported for any output format
17	RO	0x1	SW_PPD_OBUFF_LEVEL PP output buffering level: '0' = 1 unit output buffering is used '1' = 4 unit output buffering is used
16	RO	0x1	SW_PPD_PP_EXIST PPD exists: '0' = no '1' = yes
15:14	RO	0x1	SW_PPD_IN_TILED_L PPD input tiled mode support level 0 = not supported 1 = 8x4 tile size supported
13:11	RO	0x0	reserved
10:0	RO	0x780	SW_PPD_MAX_OWIDTH Max supported PP output width in pixels

**VDPU\_SWREG101**

Address: Operational Base + offset (0x0194)  
soft reset signals

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x0	sw_soft_reset softreset pulse signal softreset pulse signal write to 1'b1, valid; write to 1'b0, invalid;

**VDPU\_SWREG102**

Address: Operational Base + offset (0x0198)

vpu performance cycle

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	vpu_work_cycle vpu working cycle number write initial/reset value in the begin of frame start,then will auto count base this value.

**VDPU\_SWREG103**

Address: Operational Base + offset (0x019c)

AXI DDR READ DATA NUM

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	sw_axi_ddr_rdata axi ddr rdata num axi ddr rdata num, the unit is byte

**VDPU\_SWREG104**

Address: Operational Base + offset (0x01a0)

Register0000 Abstract

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:1	RO	0x0	reserved
0	RW	0x0	sw_axi_ddr_wdata vdpu write data byte num vdpu write data byte num

**VDPU\_SWREG105**

Address: Operational Base + offset (0x01a4)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:20	RO	0x0	reserved
19:16	RW	0x0	mon_sig_sel1 monitor signal selected for cnt1 select the counter be used for which to calculate cycle num 4'b0000: don't work 4'b0001:mv buffer hold back stream decode working cycles 4'b0010:the output fifo of cabac keep full cycles 4'b0011:the Code stream parsing block working cycles 4'b0100:scd block can't write data to scd buffer cycles 4'b0101:The speed of reconsititution and interpolation fast than reference frames feach cycles 4'b0110:The speed of reconsititution and interpolation slow than reference frames feach cycles 4'b0111:the cycles filter block hold back pred block 4'b1000:the cycles of pred block waiting for Residual data 4'b1001:the cycles of bus Related modules working
15:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x0	<p>mon_sig_sel0 monitor signal selected for cnt0 select the counter be used for which to calculate cycle num 4'b0000: don't work 4'b0001:mv buffer hold back stream decode working cycles 4'b0010:the output fifo of cabac keep full cycles 4'b0011:the Code stream parsing block working cycles 4'b0100:scd block can't write data to scd buffer cycles 4'b0101:The speed of reconsititution and interpolation fast than reference frames feach cycles 4'b0110:The speed of reconsititution and interpolation slow than reference frames feach cycles 4'b0111:the cycles filter block hold back pred block 4'b1000:the cycles of pred block waiting for Residual data 4'b1001:the cycles of bus Related modules working</p>

**VDPU\_SWREG106**

Address: Operational Base + offset (0x01a8)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>performance_mon_cnt0 the counter for the selected signal valid cycles whic describe in swreg105[3:0] write initial/reset value</p>

**VDPU\_SWREG107**

Address: Operational Base + offset (0x01ac)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>performance_mon_cnt1 Field0000 Abstract the counter for the selected signal valid cycles whic describe in swreg105[19:16] write initial/reset value</p>

**VCODEC\_MMU\_DTE\_ADDR**

Address: Operational Base + offset (0x0000)

MMU current page Table address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>MMU_DTE_ADDR MMU_DTE_ADDR MMU current page Table address</p>

**VCODEC\_MMU\_STATUS**

Address: Operational Base + offset (0x0004)

MMU status register

Bit	Attr	Reset Value	Description

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:11	RO	0x0	reserved
10:6	RO	0x00	PAGEFAULT_BUS_ID page fault bus id Index of master responsible for last page fault
5	RO	0x0	PAGE_FAULT_IS_WRITE page fault access The direction of access for last page fault: 0 = Read 1 = Write
4	RO	0x1	REPLAY_BUFFER_EMPTY replay buffer empty status 1'b1: The MMU replay buffer is empty
3	RO	0x1	MMU_IDLE mmu idle status The MMU is idle when accesses are being translated and there are no unfinished translated accesses. 1'b1: MMU is idle
2	RO	0x0	STAIL_ACTIVE stall active status MMU stall mode currently enabled. The mode is enabled by command 1'b1: MMU is in stall active status
1	RO	0x0	PAGE_FAULT_ACTIVE page fault active status MMU page fault mode currently enabled . The mode is enabled by command. 1'b1: page fault is active
0	RO	0x0	PAGING_ENABLED Paging enabled status 1'b0: paging is disabled 1'b1: Paging is enabled

**VCODEC\_MMU\_COMMAND**

Address: Operational Base + offset (0x0008)

MMU command register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:3	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
2:0	WO	0x0	MMU_CMD mmu cmd MMU_CMD. This can be: 0: MMU_ENABLE_PAGING 1: MMU_DISABLE_PAGING 2: MMU_ENABLE_STALL 3: MMU_DISABLE_STALL 4: MMU_ZAP_CACHE 5: MMU_PAGE_FAULT_DONE 6: MMU_FORCE_RESET

**VCODEC\_MMU\_PAGE\_FAULT\_ADDR**

Address: Operational Base + offset (0x000c)

MMU logical address of last page fault

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RO	0x00000000	PAGE_FAULT_ADDR page fault addr address of last page fault

**VCODEC\_MMU\_ZAP\_ONE\_LINE**

Address: Operational Base + offset (0x0010)

MMU Zap cache line register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	WO	0x00000000	MMU_ZAP_ONE_LINE zap one line address to be invalidated from the page table cache

**VCODEC\_MMU\_INT\_RAWSTAT**

Address: Operational Base + offset (0x0014)

MMU raw interrupt status register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RW	0x0	READ_BUS_ERROR read bus error read bus error status
0	RW	0x0	PAGEFAULT page fault page fault status

**VCODEC\_MMU\_INT\_CLEAR**

Address: Operational Base + offset (0x0018)

MMU raw interrupt status register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
1	WO	0x0	READ_BUS_ERROR read bus error write 1 to clear read bus error
0	WO	0x0	PAGE_FAULT page fault clear write 1 to page fault clear

**VCODEC\_MMU\_INT\_MASK**

Address: Operational Base + offset (0x001c)

MMU raw interrupt status register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RW	0x0	READ_BUS_ERROR read bus error enable the read bus interrupt source when this bit is set to 1'b1
0	RW	0x0	PAGE_FAULT page fault mask enable the page fault interrupt source when this bit is set to 1'b1

**VCODEC\_MMU\_INT\_STATUS**

Address: Operational Base + offset (0x0020)

MMU raw interrupt status register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RO	0x0	READ_BUS_ERROR read bus error status 1'b1:read bus error status
0	RO	0x0	PAGE_FAULT page fault status 1'b1:page fault

**VCODEC\_MMU\_AUTO\_GATING**

Address: Operational Base + offset (0x0024)

mmu auto gating

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:1	RO	0x0	reserved
0	RW	0x1	mmu_auto_clkgating mmu_auto_clkgating when it is 1'b1, the mmu will auto gating it self

**pref\_cache\_VERSION**

Address: Operational Base + offset (0x0000)

VERSION register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RO	0xcac2	PRODUCT_ID Field0000 Abstract Field0000 Description
15:8	RO	0x01	VERSION_MAJOR Field0000 Abstract Field0000 Description
7:0	RO	0x01	VERSION_MINOR Field0000 Abstract Field0000 Description

**pref\_cache\_SIZE**

Address: Operational Base + offset (0x0004)

L2 cache SIZE

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x06	External_bus_width Field0000 Abstract Log2 external bus width in bits
23:16	RO	0x11	CACHE_SIZE Field0000 Abstract Log2 cache size in bytes
15:8	RO	0x02	ASSOCIATIVITY Field0000 Abstract Log2 associativity
7:0	RO	0x06	LINE_SIZE Field0000 Abstract Log2 line size in bytes

**pref\_cache\_STATUS**

Address: Operational Base + offset (0x0008)

Status register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RO	0x0	DATA_BUSY Field0000 Abstract set when the cache is busy handling data
0	RO	0x0	CMD_BUSY Field0000 Abstract set when the cache is busy handling commands

**pref\_cache\_COMMAND**

Address: Operational Base + offset (0x0010)

Command setting register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:6	RO	0x0	reserved

Bit	Attr	Reset Value	Description
5:4	RW	0x0	sw_addrb_sel Field0000 Abstract 2'b00:to sel b[14:6] 2'b01:to sel b[15:9], b[7:6] 2'b10:to sel b[16:10], b[7:6] 2'b11:to sel b[17:11], b[7:6]
3	RO	0x0	reserved
2:0	WO	0x0	COMMAND Field0000 Abstract The possible command is 1 = Clear entire cache

**pref\_cache\_CLEAR\_PAGE**

Address: Operational Base + offset (0x0014)

clear page register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	CLEAR_PAGE Field0000 Abstract writing an address, invalidates all lines in that page from the cache

**pref\_cache\_MAX\_READS**

Address: Operational Base + offset (0x0018)

maximum read register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x1c	MAX_READS Field0000 Abstract Limit the number of outstanding read transactions to this amount

**pref\_cache\_PERFCNT\_SRC0**

Address: Operational Base + offset (0x0020)

performance counter 0 source register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x0	<p>PERFCNT_SRC0 Field0000 Abstract This register holds all the possible source values for Performance Counter 0</p> <p>0: total clock cycles 1: active clock cycles 2: read transactions, master 3: word reads, master 4: read transactions, slave 5: word reads, slave 6: read hit, slave 7: read misses, slave 8: read invalidates, slave 9: cacheable read transactions, slave 10: bad hit nmber, slave</p>

**pref\_cache\_PERFCNT\_VAL0**

Address: Operational Base + offset (0x0024)

performance counter 0 value register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>PERFCNT_VAL0 Field0000 Abstract Performance counter 0 value</p>

**pref\_cache\_PERFCNT\_SRC1**

Address: Operational Base + offset (0x0028)

performance counter 0 source register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	<p>PERFCNT_SRC1 Field0000 Abstract This register holds all the possible source values for Performance Counter 1</p> <p>0: total clock cycles 1: active clock cycles 2: read transactions, master 3: word reads, master 4: read transactions, slave 5: word reads, slave 6: read hit, slave 7: read misses, slave 8: read invalidates, slave 9: cacheable read transactions, slave 10: bad hit nmber, slave</p>

**pref\_cache\_PERFCNT\_VAL1**

Address: Operational Base + offset (0x002c)

performance counter 1 value register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	PERFCNT_VAL1 Field0000 Abstract Performance counter 1 value

**4.5.6 VEPU Registers Summary**

<b>Name</b>	<b>Offset</b>	<b>Size</b>	<b>Reset Value</b>	<b>Description</b>
VEPU_swreg_0	0x0000	W	0x00000000	1st quantization for jpeg lumin table/part 1 for qp quant
VEPU_swreg_1	0x0004	W	0x00000000	2st quantization for jpeg lumin table/part 1 for qp quant
VEPU_swreg_2	0x0008	W	0x00000000	3st quantization for jpeg lumin table/part 1 for qp quant
VEPU_swreg_3	0x000c	W	0x00000000	4st quantization for jpeg lumin table/part 1 for qp zbin
VEPU_swreg_4	0x0010	W	0x00000000	5st quantization for jpeg lumin table/part 1 for qp zbin
VEPU_swreg_5	0x0014	W	0x00000000	6st quantization for jpeg lumin table/part 1 for qp round
VEPU_swreg_6	0x0018	W	0x00000000	7st quantization for jpeg lumin table/part 1 for qp round
VEPU_swreg_7	0x001c	W	0x00000000	8st quantization for jpeg lumin table/part 1 for qp round
VEPU_swreg_8	0x0020	W	0x00000000	9st quantization for jpeg lumin table/part 1 for qp dequant
VEPU_swreg_9	0x0024	W	0x00000000	10st quantization for jpeg lumin table/part 2 for qp quant
VEPU_swreg_10	0x0028	W	0x00000000	11st quantization for jpeg lumin table/part 2 for qp quant
VEPU_swreg_11	0x002c	W	0x00000000	12st quantization for jpeg lumin table/part2 for qp quant
VEPU_swreg_12	0x0030	W	0x00000000	13st quantization for jpeg lumin table/part 2 for qp zbin
VEPU_swreg_13	0x0034	W	0x00000000	14st quantization for jpeg lumin table/part 2 for qp zbin
VEPU_swreg_14	0x0038	W	0x00000000	15st quantization for jpeg lumin table/part 2 for qp round
VEPU_swreg_15	0x003c	W	0x00000000	16st quantization for jpeg lumin table/part 2 for qp round
VEPU_swreg_16	0x0040	W	0x00000000	1st quantization for jpeg chroma table/part 2 for qp round
VEPU_swreg_17	0x0044	W	0x00000000	2st quantization for jpeg chroma table/part 2 for qp dequant

<b>Name</b>	<b>Offset</b>	<b>Size</b>	<b>Reset Value</b>	<b>Description</b>
VEPU_swreg_18	0x0048	W	0x00000000	3st quantization for jpeg chroma table/part 3 for qp quant
VEPU_swreg_19	0x004c	W	0x00000000	4st quantization for jpeg chroma table/part 3 for qp quant
VEPU_swreg_20	0x0050	W	0x00000000	5st quantization for jpeg chroma table/part 3 for qp quant
VEPU_swreg_21	0x0054	W	0x00000000	6st quantization for jpeg chroma table/part 3 for qp zbin
VEPU_swreg_22	0x0058	W	0x00000000	7st quantization for jpeg chroma table/part 3 for qp zbin
VEPU_swreg_23	0x005c	W	0x00000000	8st quantization for jpeg chroma table/part 3 for qp round
VEPU_swreg_24	0x0060	W	0x00000000	9st quantization for jpeg chroma table/part 3 for qp round
VEPU_swreg_25	0x0064	W	0x00000000	10st quantization for jpeg chroma table/part 3 for qp round
VEPU_swreg_26	0x0068	W	0x00000000	11st quantization for jpeg chroma table/part 3 for qp dequant
VEPU_swreg_27	0x006c	W	0x00000000	12st quantization for jpeg chroma/ VP8 segment start address
VEPU_swreg_28	0x0070	W	0x00000000	13st quantization for jpeg chroma/vp8 penalty
VEPU_swreg_29	0x0074	W	0x00000000	14st quantization for jpeg chroma/vp8 penalty
VEPU_swreg_30	0x0078	W	0x00000000	15st quantization for jpeg chroma/vp8 penalty
VEPU_swreg_31	0x007c	W	0x00000000	16st quantization for jpeg chroma/vp8 penalty
VEPU_swreg_32	0x0080	W	0x00000000	vp8 penalty
VEPU_swreg_33	0x0084	W	0x00000000	vp8 penalty
VEPU_swreg_34	0x0088	W	0x00000000	vp8 penalty
VEPU_swreg_40	0x00a0	W	0x00000000	vp8 control
VEPU_swreg_41	0x00a4	W	0x00000000	VP8 ref frame
VEPU_swreg_42	0x00a8	W	0x00000000	vp8 loop filter registers
VEPU_swreg_43	0x00ac	W	0x00000000	vp8 loop filterregister
VEPU_swreg_44	0x00b0	W	0x00000000	Intra slice bitmap
VEPU_swreg_45	0x00b4	W	0x00000000	Intra slice bitmap
VEPU_swreg_46	0x00b8	W	0x00000000	intra macro block sellect register
VEPU_swreg_47	0x00bc	W	0x00000000	CIR intra control register
VEPU_swreg_48	0x00c0	W	0x00000000	input luma start address
VEPU_swreg_49	0x00c4	W	0x00000000	input cb start address
VEPU_swreg_50	0x00c8	W	0x00000000	input cr start address
VEPU_swreg_51	0x00cc	W	0x00000000	stream header bits left register

<b>Name</b>	<b>Offset</b>	<b>Size</b>	<b>Reset Value</b>	<b>Description</b>
VEPU_swreg_52	0x00d0	W	0x00000000	stream header bits left register
VEPU_swreg_53	0x00d4	W	0x00000000	stream buffer register
VEPU_swreg_54	0x00d8	W	0x01010000	axi control register
VEPU_swreg_55	0x00dc	W	0x00000000	qp related
VEPU_swreg_56	0x00e0	W	0x00000000	the luma reference frame start address
VEPU_swreg_57	0x00e4	W	0x00000000	the chroma reference frame start address
VEPU_swreg_58	0x00e8	W	0x00000000	the result of qp sum div2
VEPU_swreg_59	0x00ec	W	0x00000000	Register0000 Abstract
VEPU_swreg_60	0x00f0	W	0x00000000	Register0001 Abstract
VEPU_swreg_61	0x00f4	W	0x00000000	input luminance information
VEPU_swreg_62	0x00f8	W	0x00000000	rlc_sum
VEPU_swreg_63	0x00fc	W	0x00000000	the reconstructed luma start address
VEPU_swreg_64	0x0100	W	0x00000000	the reconstructed chroma start address
VEPU_swreg_65_reuse	0x0104	W	0x00000000	checkpoint 1 and 2/VP8 QP
VEPU_swreg_66_reuse	0x0108	W	0x00000000	checkpoint 3 and 4/VP8 QP
VEPU_swreg_67_reuse	0x010c	W	0x00000000	checkpoint 5 and 6/VP8 QP
VEPU_swreg_68_reuse	0x0110	W	0x00000000	checkpoint 7 and 8/VP8 QP
VEPU_swreg_69_reuse	0x0114	W	0x00000000	checkpoint 9 and 10/VP8 QP
VEPU_swreg_70_reuse	0x0118	W	0x00000000	checkpoint word error 1 and 2/VP8 QP
VEPU_swreg_71_reuse	0x011c	W	0x00000000	checkpoint word error 1 and 2/VP8 QP
VEPU_swreg_72_reuse	0x0120	W	0x00000000	checkpoint word error 1 and 2/VP8 QP
VEPU_swreg_73_reuse	0x0124	W	0x00000000	checkpoint delta QP register/VP8 QP
VEPU_swreg_74	0x0128	W	0x00000000	input image format
VEPU_swreg_75	0x012c	W	0x00000000	intra/inter mode
VEPU_swreg_76_reuse	0x0130	W	0x00000000	encoder control register 0
VEPU_swreg_77	0x0134	W	0x00000000	output stream start address
VEPU_swreg_78	0x0138	W	0x00000000	output control start address
VEPU_swreg_79	0x013c	W	0x00000000	next picture luminance start address
VEPU_swreg_80	0x0140	W	0x00000000	Base address for MV output
VEPU_swreg_81	0x0144	W	0x00000000	the cabac table start address
VEPU_swreg_82	0x0148	W	0x00000000	the first of ROI area register
VEPU_swreg_83	0x014c	W	0x00000000	the second of ROI area register
VEPU_swreg_84	0x0150	W	0x00000000	Stabilization matrix
VEPU_swreg_85	0x0154	W	0x00000000	Stabilization matrix
VEPU_swreg_86	0x0158	W	0x00000000	Stabilization matrix

Name	Offset	Size	Reset Value	Description
VEPU_swreg_87	0x015c	W	0x00000000	Stabilization matrix
VEPU_swreg_88	0x0160	W	0x00000000	Stabilization matrix
VEPU_swreg_89	0x0164	W	0x00000000	Stabilization matrix
VEPU_swreg_90	0x0168	W	0x00000000	Stabilization matrix
VEPU_swreg_91	0x016c	W	0x00000000	Stabilization matrix
VEPU_swreg_92	0x0170	W	0x00000000	Stabilization matrix
VEPU_swreg_93	0x0174	W	0x00000000	the output of Stabilization motion sum
VEPU_swreg_94	0x0178	W	0x00000000	output of Stabilization
VEPU_swreg_95	0x017c	W	0x00000000	RGB to YUV conversion coefficient register
VEPU_swreg_96	0x0180	W	0x00000000	RGB to YUV conversion coefficient register
VEPU_swreg_97	0x0184	W	0x00000000	RGB to YUV conversion coefficient register
VEPU_swreg_98	0x0188	W	0x00000000	RGA MASK
VEPU_swreg_99	0x018c	W	0x00000000	mv related
VEPU_swreg_100_reuse	0x0190	W	0x00000000	QP register/VP8 qp
VEPU_swreg_101_read	0x0194	W	0x1f522780	Register0003 Abstract
VEPU_swreg_102	0x0198	W	0x00000000	mvc related
VEPU_swreg_103	0x019c	W	0x00000000	encoder start
VEPU_swreg_104	0x01a0	W	0x00000000	mb control register
VEPU_swreg_105	0x01a4	W	0x00000000	SWAP
VEPU_swreg_106_reuse	0x01a8	W	0x00000000	encoder control register 1
VEPU_swreg_107_reuse	0x01ac	W	0x00000000	JPEG control register /Penalty
VEPU_swreg_108_reuse	0x01b0	W	0x00000000	intra_slice_bmp2/vp8 counters or probability address
VEPU_swreg_109	0x01b4	W	0x00001000	Register0004 Abstract
VEPU_swreg_110_read	0x01b8	W	0x48311220	product ID
VEPU_swreg_120_183	0x01e0	W	0x00000000	DMV_4p_1p_penalty

Notes:*Size:* **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

#### 4.5.7 VEPU Detail Register Description

##### VEPU\_swreg\_0

Address: Operational Base + offset (0x0000)

1st quantization for jpeg lumin table/part 1 for qp quant

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:16	RW	0x0000	vp8_pt1_qut_dc_y2 part 1 for qp quant dc y2 part 1 for qp quant dc y2
15:14	RO	0x0	reserved
13:0	RW	0x0000	vp8_pt1_qut_dc_y1 part 1 for qp quant dc y1 part1 for qp quant dc y1

**VEPU\_swreg\_1**

Address: Operational Base + offset (0x0004)

2st quantization for jpeg lumin table/part 1 for qp quant

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:16	RW	0x0000	vp8_pt1_qut_ac_y1 part 1 for qp quant ac y1 part 1 for qp quant ac y1
15:14	RO	0x0	reserved
13:0	RW	0x0000	vp8_pt1_qut_dc_ch part 1 for qp quant dc chroma part 1 for qp quant dc chroma

**VEPU\_swreg\_2**

Address: Operational Base + offset (0x0008)

3st quantization for jpeg lumin table/part 1 for qp quant

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:16	RW	0x0000	vp8_pt1_qut_ac_ch part 1 for qp quant ac chroma part 1 for qp quant ac chroma
15:14	RO	0x0	reserved
13:0	RW	0x0000	vp8_pt1_qut_ac_y2 part 1 for qp quant ac y2 part 1 for qp quant ac y2

**VEPU\_swreg\_3**

Address: Operational Base + offset (0x000c)

4st quantization for jpeg lumin table/part 1 for qp zbin

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:27	RO	0x0	reserved
26:18	RW	0x000	vp8_pt1_zb_dc_ch part 1 for qp zbin dc chroma part 1 for qp zbin dc chroma
17:9	RW	0x000	vp8_pt1_zb_dc_y2 part 1 for qp zbin dc y2 part 1 for qp zbin dc y2
8:0	RW	0x000	vp8_pt1_zb_dc_y1 part 1 for qp zbin dc y1 part 1 for qp zbin dc y1

**VEPU\_swreg\_4**

Address: Operational Base + offset (0x0010)

5st quantization for jpeg lumin table/part 1 for qp zbin

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:27	RO	0x0	reserved
26:18	RW	0x000	vp8_pt1_zb_ac_ch part 1 for qp zbin ac chroma part 1 for qp zbin ac chroma
17:9	RW	0x000	vp8_pt1_zb_ac_y2 part 1 for qp zbin ac y2 part 1 for qp zbin ac y2
8:0	RW	0x000	vp8_pt1_zb_ac_y1 part 1 for qp zbin ac y1 part 1 for qp zbin ac y1

**VEPU\_swreg\_5**

Address: Operational Base + offset (0x0014)

6st quantization for jpeg lumin table/part 1 for qp round

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x0	reserved
23:16	RW	0x00	vp8_pt1_rnd_dc_ch part 1 for qp round dc chroma part 1 for qp round dc chroma
15:8	RW	0x00	vp8_pt1_rnd_dc_y2 part 1 for qp round dc y2 part 1 for qp round dc y2
7:0	RW	0x00	vp8_pt1_rnd_dc_y1 part 1 for qp round dc y1 part 1 for qp round dc y1

**VEPU\_swreg\_6**

Address: Operational Base + offset (0x0018)

7st quantization for jpeg lumin table/part 1 for qp round

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x0	reserved
23:16	RW	0x00	vp8_pt1_rnd_ac_ch part 1 for qp round ac chroma part 1 for qp round ac chroma
15:8	RW	0x00	vp8_pt1_rnd_ac_y2 part 1 for qp round ac y2 part 1 for qp round ac y2
7:0	RW	0x00	vp8_pt1_rnd_ac_y1 part 1 for qp round ac y1 part 1 for qp round ac y1

**VEPU\_swreg\_7**

Address: Operational Base + offset (0x001c)

8st quantization for jpeg lumin table/part 1 for qp round

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RO	0x0	reserved
30:25	RW	0x00	vp8_pt1_filter_sel VP8 part1 filter select VP8 part1 filter select
24:17	RW	0x00	vp8_pt1_dequt_dc_ch part 1 for qp dequnt dc chroma part 1 for qp dequnt dc chroma
16:8	RW	0x000	vp8_pt1_deqnt_dc_y2 part 1 for qp dequnt dc y2 part 1 for qp dequnt dc y2
7:0	RW	0x00	vp8_pt1_dequt_dc_y1 part 1 for qp dequnt dc y1 part 1 for qp dequnt dc y1

**VEPU\_swreg\_8**

Address: Operational Base + offset (0x0020)

9st quantization for jpeg lumin table/part 1 for qp dequant

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:27	RO	0x0	reserved
26:18	RW	0x000	vp8_pt1_dequt_ac_ch part 1 for qp dequnt ac chroma part 1 for qp dequnt ac chroma
17:9	RW	0x000	vp8_pt1_dequt_ac_y2 part 1 for qp dequnt ac y2 part 1 for qp dequnt ac y2
8:0	RW	0x000	vp8_pt1_dequt_ac_y1 part 1 for qp dequnt ac y1 part 1 for qp dequnt ac y1

**VEPU\_swreg\_9**

Address: Operational Base + offset (0x0024)

10st quantization for jpeg lumin table/part 2 for qp quant

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:16	RW	0x0000	vp8_pt2_qut_dc_y2 part 2 for qp quant dc y2 part 2 for qp quant dc y2
15:14	RO	0x0	reserved
13:0	RW	0x0000	vp8_pt2_qut_dc_y1 part 2 for qp quant dc y1 part2 for qp quant dc y1

**VEPU\_swreg\_10**

Address: Operational Base + offset (0x0028)

11st quantization for jpeg lumin table/part 2 for qp quant

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:16	RW	0x0000	vp8_pt2_qut_ac_y1 part 2 for qp quant ac y1 part 2 for qp quant ac y1
15:14	RO	0x0	reserved
13:0	RW	0x0000	vp8_pt2_qut_dc_ch part 2 for qp quant dc chroma part 2 for qp quant dc chroma

**VEPU\_swreg\_11**

Address: Operational Base + offset (0x002c)

12st quantization for jpeg lumin table/part2 for qp quant

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:16	RW	0x0000	vp8_pt2_qut_ac_ch part 2 for qp quant ac chroma part 2 for qp quant ac chroma
15:14	RO	0x0	reserved
13:0	RW	0x0000	vp8_pt2_qut_ac_y2 part 2 for qp quant ac y2 part 2 for qp quant ac y2

**VEPU\_swreg\_12**

Address: Operational Base + offset (0x0030)

13st quantization for jpeg lumin table/part 2 for qp zbin

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:27	RO	0x0	reserved
26:18	RW	0x000	vp8_pt2_zb_dc_ch part 2 for qp zbin dc chroma part 2 for qp zbin dc chroma
17:9	RW	0x000	vp8_pt2_zb_dc_y2 part 2 for qp zbin dc y2 part 2 for qp zbin dc y2
8:0	RW	0x000	vp8_pt2_zb_dc_y1 part 2 for qp zbin dc y1 part 2 for qp zbin dc y1

**VEPU\_swreg\_13**

Address: Operational Base + offset (0x0034)

14st quantization for jpeg lumin table/part 2 for qp zbin

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:27	RO	0x0	reserved
26:18	RW	0x000	vp8_pt2_zb_ac_ch part 2 for qp zbin ac chroma part 2 for qp zbin ac chroma

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
17:9	RW	0x000	vp8_pt2_zb_ac_y2 part 2 for qp zbin ac y2 part 2 for qp zbin ac y2
8:0	RW	0x000	vp8_pt2_zb_ac_y1 part 2 for qp zbin ac y1 part 2 for qp zbin ac y1

**VEPU\_swreg\_14**

Address: Operational Base + offset (0x0038)

15st quantization for jpeg lumin table/part 2 for qp round

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x0	reserved
23:16	RW	0x00	vp8_pt2_rnd_dc_ch part 2 for qp round dc chroma part 2 for qp round dc chroma
15:8	RW	0x00	vp8_pt2_rnd_dc_y2 part 2 for qp round dc y2 part 2 for qp round dc y2
7:0	RW	0x00	vp8_pt2_rnd_dc_y1 part 2 for qp round dc y1 part 2 for qp round dc y1

**VEPU\_swreg\_15**

Address: Operational Base + offset (0x003c)

16st quantization for jpeg lumin table/part 2 for qp round

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x0	reserved
23:16	RW	0x00	vp8_pt2_rnd_ac_ch part 2 for qp round ac chroma part 2 for qp round ac chroma
15:8	RW	0x00	vp8_pt2_rnd_ac_y2 part 2 for qp round ac y2 part 2 for qp round ac y2
7:0	RW	0x00	vp8_pt2_rnd_ac_y1 part 2 for qp round ac y1 part 2 for qp round ac y1

**VEPU\_swreg\_16**

Address: Operational Base + offset (0x0040)

1st quantization for jpeg chroma table/part 2 for qp round

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RO	0x0	reserved
30:25	RW	0x00	vp8_pt2_filter_sel VP8 part2 filter select VP8 part2 filter select

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
24:17	RW	0x00	vp8_pt2_dequt_dc_ch part 2 for qp dequant dc chroma part2 for qp dequant dc chroma
16:8	RW	0x000	vp8_pt2_deqnt_dc_y2 part 2 for qp dequant dc y2 part 2 for qp dequant dc y2
7:0	RW	0x00	vp8_pt2_dequt_dc_y1 part 2 for qp dequant dc y1 part 2 for qp dequant dc y1

**VEPU\_swreg\_17**

Address: Operational Base + offset (0x0044)

2st quantization for jpeg chroma table/part 2 for qp dequant

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:27	RO	0x0	reserved
26:18	RW	0x000	vp8_pt2dequt_ac_ch part 2 for qp dequant ac chroma part 2 for qp dequant ac chroma
17:9	RW	0x000	vp8_pt2_dequt_ac_y2 part 2 for qp dequant ac y2 part 2 for qp dequant ac y2
8:0	RW	0x000	vp8_pt2_dequt_ac_y1 part2 for qp dequant ac y1 part2 for qp dequant ac y1

**VEPU\_swreg\_18**

Address: Operational Base + offset (0x0048)

3st quantization for jpeg chroma table/part 3 for qp quant

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:16	RW	0x0000	vp8_pt3_qut_dc_y2 part 3 for qp quant dc y2 part 3 for qp quant dc y2
15:14	RO	0x0	reserved
13:0	RW	0x0000	vp8_pt3_qut_dc_y1 part 3 for qp quant dc y1 part3 for qp quant dc y1

**VEPU\_swreg\_19**

Address: Operational Base + offset (0x004c)

4st quantization for jpeg chroma table/part 3 for qp quant

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
29:16	RW	0x0000	vp8_pt3_qut_ac_y1 part 3 for qp quant ac y1 part 3 for qp quant ac y1
15:14	RO	0x0	reserved
13:0	RW	0x0000	vp8_pt3_qut_dc_ch part 3 for qp quant dc chroma part 3 for qp quant dc chroma

**VEPU\_swreg\_20**

Address: Operational Base + offset (0x0050)

5st quantization for jpeg chroma table/part 3 for qp quant

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:16	RW	0x0000	vp8_pt3_qut_ac_ch part 3 for qp quant ac chroma part 3 for qp quant ac chroma
15:14	RO	0x0	reserved
13:0	RW	0x0000	vp8_pt3_qut_ac_y2 part 3 for qp quant ac y2 part 3 for qp quant ac y2

**VEPU\_swreg\_21**

Address: Operational Base + offset (0x0054)

6st quantization for jpeg chroma table/part 3 for qp zbin

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:27	RO	0x0	reserved
26:18	RW	0x000	vp8_pt3_zb_dc_ch part 3 for qp zbin dc chroma part 3 for qp zbin dc chroma
17:9	RW	0x000	vp8_pt3_zb_dc_y2 part 3 for qp zbin dc y2 part 3 for qp zbin dc y2
8:0	RW	0x000	vp8_pt3_zb_dc_y1 part 3 for qp zbin dc y1 part 3 for qp zbin dc y1

**VEPU\_swreg\_22**

Address: Operational Base + offset (0x0058)

7st quantization for jpeg chroma table/part 3 for qp zbin

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:27	RO	0x0	reserved
26:18	RW	0x000	vp8_pt3_zb_ac_ch part 3 for qp zbin ac chroma part 3 for qp zbin ac chroma

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
17:9	RW	0x000	vp8_pt3_zb_ac_y2 part 3 for qp zbin ac y2 part 3 for qp zbin ac y2
8:0	RW	0x000	vp8_pt3_zb_ac_y1 part 3 for qp zbin ac y1 part 3 for qp zbin ac y1

**VEPU\_swreg\_23**

Address: Operational Base + offset (0x005c)

8st quantization for jpeg chroma table/part 3 for qp round

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x0	reserved
23:16	RW	0x00	vp8_pt3_rnd_dc_ch part 3 for qp round dc chroma part 3 for qp round dc chroma
15:8	RW	0x00	vp8_pt3_rnd_dc_y2 part 3 for qp round dc y2 part 3 for qp round dc y2
7:0	RW	0x00	vp8_pt3_rnd_dc_y1 part 3 for qp round dc y1 part 3 for qp round dc y1

**VEPU\_swreg\_24**

Address: Operational Base + offset (0x0060)

9st quantization for jpeg chroma table/part 3 for qp round

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x0	reserved
23:16	RW	0x00	vp8_pt3_rnd_ac_ch part 3 for qp round ac chroma part 3 for qp round ac chroma
15:8	RW	0x00	vp8_pt3_rnd_ac_y2 part 3 for qp round ac y2 part 3 for qp round ac y2
7:0	RW	0x00	vp8_pt3_rnd_ac_y1 part 3 for qp round ac y1 part 3 for qp round ac y1

**VEPU\_swreg\_25**

Address: Operational Base + offset (0x0064)

10st quantization for jpeg chroma table/part 3 for qp round

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RO	0x0	reserved
30:25	RW	0x00	vp8_pt3_filter_sel VP8 part3 filter select VP8 part3 filter select

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
24:17	RW	0x00	vp8_pt3_dequt_dc_ch part 3 for qp dequant dc chroma part 3 for qp dequant dc chroma
16:8	RW	0x000	vp8_pt3_deqnt_dc_y2 part 3 for qp dequant dc y2 part 3 for qp dequant dc y2
7:0	RW	0x00	vp8_pt3_dequt_dc_y1 part 3 for qp dequant dc y1 part 3 for qp dequant dc y1

**VEPU\_swreg\_26**

Address: Operational Base + offset (0x0068)

11st quantization for jpeg chroma table/part 3 for qp dequant

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:27	RO	0x0	reserved
26:18	RW	0x000	vp8_pt3_dequt_ac_ch part 3 for qp dequant ac chroma part 3 for qp dequant ac chroma
17:9	RW	0x000	vp8_pt3_dequt_ac_y2 part 3 for qp dequant ac y2 part 3 for qp dequant ac y2
8:0	RW	0x000	vp8_pt3_dequt_ac_y1 part 3 for qp dequant ac y1 part 3 for qp dequant ac y1

**VEPU\_swreg\_27**

Address: Operational Base + offset (0x006c)

12st quantization for jpeg chroma/ VP8 segment start address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	vp8_seg_st_adres VP8 segment start address 11st quantization for jpeg chr/ VP8 segment start address

**VEPU\_swreg\_28**

Address: Operational Base + offset (0x0070)

13st quantization for jpeg chroma/vp8 penalty

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:28	RO	0x0	reserved
27:16	RW	0x000	vp8_imp_4x4_1 VP8 intra penalty for 4x4 1st sel VP8 intra penalty for 4x4 1st sel
15:12	RO	0x0	reserved
11:0	RW	0x000	vp8_imp_4x4_0 VP8 intra penalty for 4x4 0st sel VP8 intra penalty for 4x4 0st sel

**VEPU\_swreg\_29**

Address: Operational Base + offset (0x0074)

14st quantization for jpeg chroma/vp8 penalty

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:28	RO	0x0	reserved
27:16	RW	0x000	vp8_imp_4x4_3 VP8 intra penalty for 4x4 3st sel VP8 intra penalty for 4x4 3st sel
15:12	RO	0x0	reserved
11:0	RW	0x000	vp8_imp_4x4_2 VP8 intra penalty for 4x4 2st sel VP8 intra penalty for 4x4 2st sel

**VEPU\_swreg\_30**

Address: Operational Base + offset (0x0078)

15st quantization for jpeg chroma/vp8 penalty

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:28	RO	0x0	reserved
27:16	RW	0x000	vp8_imp_4x4_5 VP8 intra penalty for 4x4 5st sel VP8 intra penalty for 4x4 5st sel
15:12	RO	0x0	reserved
11:0	RW	0x000	vp8_imp_4x4_4 VP8 intra penalty for 4x4 4st sel VP8 intra penalty for 4x4 4st sel

**VEPU\_swreg\_31**

Address: Operational Base + offset (0x007c)

16st quantization for jpeg chroma/vp8 penalty

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:28	RO	0x0	reserved
27:16	RW	0x000	vp8_imp_4x4_7 VP8 intra penalty for 4x4 7st sel VP8 intra penalty for 4x4 7st sel
15:12	RO	0x0	reserved
11:0	RW	0x000	vp8_imp_4x4_6 VP8 intra penalty for 4x4 3st sel VP8 intra penalty for 4x4 6st sel

**VEPU\_swreg\_32**

Address: Operational Base + offset (0x0080)

vp8 penalty

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:28	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
27:16	RW	0x000	vp8_imp_4x4_9 VP8 intra penalty for 4x4 9st sel VP8 intra penalty for 4x4 9st sel
15:12	RO	0x0	reserved
11:0	RW	0x000	vp8_imp_4x4_8 VP8 intra penalty for 4x4 8st sel VP8 intra penalty for 4x4 8st sel

**VEPU\_swreg\_33**

Address: Operational Base + offset (0x0084)

vp8 penalty

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:28	RO	0x0	reserved
27:16	RW	0x000	vp8_imp_16x16_1 VP8 intra penalty for 16x16 1st sel VP8 intra penalty for 16x16 1st sel
15:12	RO	0x0	reserved
11:0	RW	0x000	vp8_imp_16x16_0 VP8 intra penalty for 16x16 0st sel VP8 intra penalty for 16x16 0st sel

**VEPU\_swreg\_34**

Address: Operational Base + offset (0x0088)

vp8 penalty

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:28	RO	0x0	reserved
27:16	RW	0x000	vp8_imp_16x16_3 VP8 intra penalty for 16x16 3st sel VP8 intra penalty for 16x16 3st sel
15:12	RO	0x0	reserved
11:0	RW	0x000	vp8_imp_16x16_2 VP8 intra penalty for 16x16 2st sel VP8 intra penalty for 16x16 2st sel

**VEPU\_swreg\_40**

Address: Operational Base + offset (0x00a0)

vp8 control

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RO	0x0	reserved
30:24	RW	0x00	vp8_lpf_bpred The bpred of loop filter for vp8 The bpred of loop filter for vp8
23	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
22:16	RW	0x00	vp8_lpf_intra The intra macro block of loop filter for vp8 The intra macro block of loop filter for vp8
15:12	RO	0x0	reserved
11:0	RW	0x000	vp8_inter_type inter type for vp8 inter type for vp8

**VEPU\_swreg\_41**

Address: Operational Base + offset (0x00a4)

VP8 ref frame

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:28	RO	0x0	reserved
27:16	RW	0x000	vp8_coef_dmv_ply The coefficient of dmv penalty for VP8 The coefficient of dmv penalty for VP8
15:12	RO	0x0	reserved
11:0	RW	0x000	vp8_ref_frame The reference frame for vp8 The reference frame for vp8

**VEPU\_swreg\_42**

Address: Operational Base + offset (0x00a8)

vp8 loop filter registers

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:23	RO	0x0	reserved
22:16	RW	0x00	vp8_lpf_altnref loop filter for alt reference frame used in vp8 loop filter for alt reference frame
15	RO	0x0	reserved
14:8	RW	0x00	vp8_lpf_lastref loop filter for last reference frame used in vp8 loop filter for last reference frame
7	RO	0x0	reserved
6:0	RW	0x00	vp8_lpf_orgref loop filter for original reference frame used in vp8 loop filter for reference frame

**VEPU\_swreg\_43**

Address: Operational Base + offset (0x00ac)

vp8 loop filterregister

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:23	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
22:16	RW	0x00	vp8_lpf_divmv loop filter for divide mv used in vp8 loop filter for divide mv used in vp8
15	RO	0x0	reserved
14:8	RW	0x00	vp8_lpf_zeromv loop filter for zero mv used in vp8 loop filter for zero mv used in vp8
7	RO	0x0	reserved
6:0	RW	0x00	vp8_lpf_newmv loop filter for new mv used in vp8 loop filter for new mv used in vp8

**VEPU\_swreg\_44**

Address: Operational Base + offset (0x00b0)

Intra slice bitmap

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	intra_slice_bmp0 Intra slice bitmap for slices0 to slices31 bit0 : slices0 bit1 : slices1 bit2 : slices2 ..... bit31 : slices31

**VEPU\_swreg\_45**

Address: Operational Base + offset (0x00b4)

Intra slice bitmap

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	intra_slice_bmp1 Intra slice bitmap for slices32 to slices63 bit0 : slices32 bit1 : slices33 bit2 : slices34 ..... bit31 : slices63

**VEPU\_swreg\_46**

Address: Operational Base + offset (0x00b8)

intra macro block select register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RW	0x00	intra_up_mb_area The up intra macro block's area used in row The top intra macro block's area used in row

Bit	Attr	Reset Value	Description
23:16	RW	0x00	intra_down_mb_area The down intra macro block's area used in row The bottom intra macro block's area used in row
15:8	RW	0x00	intra_left_mb_area The left intra macro block's area used in column The left intra macro block's area used in column
7:0	RW	0x00	intra_right_mb_area The right intra macro block's area used in column The right intra macro block's area used in column

**VEPU\_swreg\_47**

Address: Operational Base + offset (0x00bc)

CIR intra control register

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	cir_first_intra the first macro block selected for cir 0:disable other:enable and be set
15:0	RW	0x0000	cir_intra_mb_itvl the interval for cir intra macro block 0: disable other: enable and be set

**VEPU\_swreg\_48**

Address: Operational Base + offset (0x00c0)

input luma start address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	luma_in_st_adr input luma start address input luma start address

**VEPU\_swreg\_49**

Address: Operational Base + offset (0x00c4)

input cb start address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	cb_in_st_adr input cb start address input cb start address

**VEPU\_swreg\_50**

Address: Operational Base + offset (0x00c8)

input cr start address

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	cr_in_st_addr input cr start address input cr start address

**VEPU\_swreg\_51**

Address: Operational Base + offset (0x00cc)

stream header bits left register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	strm_header_left_hbits the high 32 bit of stram header be left the high 32 bit of stram header be left

**VEPU\_swreg\_52**

Address: Operational Base + offset (0x00d0)

stream header bits left register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	strm_header_left_lbits the low 32 bit of stram header be left the low 32 bit of stram header be left

**VEPU\_swreg\_53**

Address: Operational Base + offset (0x00d4)

stream buffer register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	strm_bufsize_lmt the limit size of steam buffer the limit size of steam buffer

**VEPU\_swreg\_54**

Address: Operational Base + offset (0x00d8)

axi control register

Bit	Attr	Reset Value	Description
31:24	RW	0x01	axi_rd_id axi read id if config 0,it will be modify as 1 by HW auto
23:16	RW	0x01	axi_wr_id axi write id if config 0,it will be modify as 1 by HW auto
15:14	RO	0x0	reserved
13:8	RW	0x00	burst_len burst length burst length
7:3	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
2	RW	0x0	burst_incr_mod_sel burst increment mode select 0: single burst selected 1: incr burst selected
1	RW	0x0	burst_discard on-off burst data dicard 0:disable ,off 1:enable,on
0	RW	0x0	burst_disable disable burst mode for AXI 0: enable 1: disable

**VEPU\_swreg\_55**

Address: Operational Base + offset (0x00dc)

qp related

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RO	0x0	reserved
15:12	RW	0x0	roi_dlt_qp1 1st for delta qp for roi 1st for delta qp for roi
11:8	RW	0x0	roi_dlt_qp2 2st for delta qp for roi 2st for delta qp for roi
7:4	RO	0x0	reserved
3:0	RW	0x0	qp_adjst QP adjustment for mad signed register; range from -8 to 7

**VEPU\_swreg\_56**

Address: Operational Base + offset (0x00e0)

the luma reference frame start address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	luma_ref_st_adr the luma reference frame start address the luma reference frame start address

**VEPU\_swreg\_57**

Address: Operational Base + offset (0x00e4)

the chroma reference frame start address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	chroma_ref_st_adr the chroma reference frame start address the chroma reference frame start address

**VEPU\_swreg\_58**

Address: Operational Base + offset (0x00e8)

the result of qp sum div2

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:11	RW	0x000000	qp_sum_div2 the result of (qp sum)/2 the result of (qp sum)/2
10:0	RO	0x0	reserved

**VEPU\_swreg\_59**

Address: Operational Base + offset (0x00ec)

Register0000 Abstract

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RO	0x0	reserved
28	RW	0x0	h264_qurt_pixmv_dis disable the function of quarter pixel MVs used in h264 1:disable 0:default,enable
27:26	RO	0x0	reserved
25:24	RW	0x0	dblking_flt_mode deblocking filter mode select 0 : enabled 1 : disabled 2 : disabled on slice ( vp8=simple )
23	RO	0x0	reserved
22:21	RW	0x0	h264_cabac_idc the initial idc for cabac used in h264 0,1,2: used 3: no use
20	RW	0x0	entry_code_fmt the format of stream entropy coding h.264: 0: cavlc 1: cabac VP8:boolenc enable
19:18	RO	0x0	reserved
17	RW	0x0	h264_trfmod_8x8 on-off for 8x8 transform used in h264 on-off for 8x8 transform used in h264
16	RW	0x0	h264_res_intermod_4x4 the restriction inter mode selected in 4x4 block the restriction inter mode selected in 4x4 block

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
15	RW	0x0	h264_strm_mod_sel used to select stream mode 0 : NAL unit ; 1 : BYTE
14:8	RW	0x00	h264_slice_num the h264 slice number in one picture 0=one slice in current picture 1=two slice in current picture .....
7:0	RO	0x0	reserved

**VEPU\_swreg\_60**

Address: Operational Base + offset (0x00f0)

Register0001 Abstract

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:22	RO	0x0	reserved
21:16	RW	0x00	strm_st_offset the start offset for stream
15:8	RW	0x00	skip_mb_mode H.264:SKIP macroblock mode VP8 :zero/nearest/near mode penalty
7:6	RO	0x0	reserved
5:4	RW	0x0	right_spill the right edge of image for spill pixels div4 value range:0~3
3:0	RW	0x0	bot_spill the bottom edge of image for spill pixels the bottom edge of image for spill pixels

**VEPU\_swreg\_61**

Address: Operational Base + offset (0x00f4)

input luminance information

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:23	RO	0x0	reserved
22:20	RW	0x0	offset_in_chroma then offset of input chroma byte unit
19	RO	0x0	reserved
18:16	RW	0x0	offset_in_luma then offset of input luminance byte unit
15:14	RO	0x0	reserved

Bit	Attr	Reset Value	Description
13:0	RW	0x0000	row_len_in_luma the row length of input luminance

**VEPU\_swreg\_62**

Address: Operational Base + offset (0x00f8)

rlc\_sum

Bit	Attr	Reset Value	Description
31:22	RO	0x0	reserved
21:0	RW	0x0000000	rlc_sum rlc_sum rlc_sum

**VEPU\_swreg\_63**

Address: Operational Base + offset (0x00fc)

the reconstructed luma start address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	recon_luma_st_adr the reconstructed luma start address the reconstructed luma start address

**VEPU\_swreg\_64**

Address: Operational Base + offset (0x0100)

the reconstructed chroma start address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	recon_chroma_st_adr the reconstructed chroma start address the reconstructed chroma start address

**VEPU\_swreg\_65\_reuse**

Address: Operational Base + offset (0x0104)

checkpoint 1 and 2/VP8 QP

Bit	Attr	Reset Value	Description
31:16	RW	0x0000	h264_chkpt_1 1st word used for check point used in h.264 1st word used for check point used in h.264
15:0	RW	0x0000	h264_chkpt_2 2st word used for check point used in h.264 2st word used for check point used in h.264

**VEPU\_swreg\_66\_reuse**

Address: Operational Base + offset (0x0108)

checkpoint 3 and 4/VP8 QP

Bit	Attr	Reset Value	Description
-----	------	-------------	-------------

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	h264_chkpt_3 3st word used for check point used in h.264 3st word used for check point used in h.264
15:0	RW	0x0000	h264_chkpt_4 4st word used for check point used in h.264 4st word used for check point used in h.264

**VEPU\_swreg\_67\_reuse**

Address: Operational Base + offset (0x010c)  
checkpoint 5 and 6/VP8 QP

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	h264_chkpt_5 5st word used for check point used in h.264 5st word used for check point used in h.264
15:0	RW	0x0000	h264_chkpt_6 6st word used for check point used in h.264 6st word used for check point used in h.264

**VEPU\_swreg\_68\_reuse**

Address: Operational Base + offset (0x0110)  
checkpoint 7 and 8/VP8 QP

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	h264_chkpt_7 7st word used for check point used in h.264 7st word used for check point used in h.264
15:0	RW	0x0000	h264_chkpt_8 8st word used for check point used in h.264 8st word used for check point used in h.264

**VEPU\_swreg\_69\_reuse**

Address: Operational Base + offset (0x0114)  
checkpoint 9 and 10/VP8 QP

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	h264_chkpt_9 9st word used for check point used in h.264 9st word used for check point used in h.264
15:0	RW	0x0000	h264_chkpt_10 10st word used for check point used in h.264 10st word used for check point used in h.264

**VEPU\_swreg\_70\_reuse**

Address: Operational Base + offset (0x0118)  
checkpoint word error 1 and 2/VP8 QP

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
------------	-------------	--------------------	--------------------

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	h264_errchkpt_1 1st word error check point used in h.264 1st word error check point used in h.264
15:0	RW	0x0000	h264_errchkpt_2 2st word error check point used in h.264 2st word error check point used in h.264

**VEPU\_swreg\_71\_reuse**

Address: Operational Base + offset (0x011c)  
checkpoint word error 1 and 2/VP8 QP

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	h264_errchkpt_3 3st word error check point used in h.264 3st word error check point used in h.264
15:0	RW	0x0000	h264_errchkpt_4 4st word error check point used in h.264 4st word error check point used in h.264

**VEPU\_swreg\_72\_reuse**

Address: Operational Base + offset (0x0120)  
checkpoint word error 1 and 2/VP8 QP

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	h264_errchkpt_5 5st word error check point used in h.264 5st word error check point used in h.264
15:0	RW	0x0000	h264_errchkpt_6 6st word error check point used in h.264 6st word error check point used in h.264

**VEPU\_swreg\_73\_reuse**

Address: Operational Base + offset (0x0124)  
checkpoint delta QP register/VP8 QP

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:28	RO	0x0	reserved
27:24	RW	0x0	chkqp_1 1st for delta qp check point 1st for delta qp check point
23:20	RW	0x0	chkqp_2 2st for delta qp check point 2st for delta qp check point
19:16	RW	0x0	chkqp_3 3st for delta qp check point 3st for delta qp check point

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
15:12	RW	0x0	chkqp_4 4st for delta qp check point 4st for delta qp check point
11:8	RW	0x0	chkqp_5 5st for delta qp check point 5st for delta qp check point
7:4	RW	0x0	chkqp_6 6st for delta qp check point 6st for delta qp check point
3:0	RW	0x0	chkqp_7 7st for delta qp check point 7st for delta qp check point

**VEPU\_swreg\_74**

Address: Operational Base + offset (0x0128)

input image format

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:24	RW	0x00	mad_thsld MAD threshold value = (MAD threshold)/256
23:16	RW	0x00	encodered_slices the number of encoder slices which used in h.264 the number of encoder slices which used in h.264
15:8	RO	0x0	reserved
7:4	RW	0x0	img_fmt_in input image format. YUV420P YUV420SP YUV422 UYVY422 RGB565 RGB444 RGB888 RGB101010
3:2	RW	0x0	img_in_rot the input image rotation 0 : no rotation 1 : rotate right 90 degress 2 : rotate left 90 degress
1	RO	0x0	reserved
0	RW	0x0	nal_mode the output of NAL size to base control the output of NAL size to base control

**VEPU\_swreg\_75**

Address: Operational Base + offset (0x012c)

intra/inter mode

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	intramod_16x16 the Intra prediction for 16x16 mode favor
15:0	RW	0x0000	intermod the intra/inter selection for inter macro block mode favor the intra/inter selection for inter macro block mode favor

**VEPU\_swreg\_76\_reuse**

Address: Operational Base + offset (0x0130)

encoder control register 0

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:26	RW	0x00	pps_init_qp pps init qp in picture used in h264 pps init qp in picture used in h264 range : 0~51
25:22	RW	0x0	slice_flt_alpha the offset of slice filter alpha c0 used in h264 offset div2 range : -6~6
21:18	RW	0x0	slice_flt_beta the filter beta offset for h264 slice config value = (real value)/2 signed register range : -6 ~6
17:13	RW	0x00	qp_offset_ch the qp index offset for chroma qp used in h264 signed register range : -12~12
12:9	RO	0x0	reserved
8	RW	0x0	sw_qpass jpeg enc quant bypass
7:5	RO	0x0	reserved
4:1	RW	0x0	idr_picid IDR pic ID IDR pic ID
0	RW	0x0	constr_intra_pred constrained intra prediction constrained intra prediction

**VEPU\_swreg\_77**

Address: Operational Base + offset (0x0134)

output stream start address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	output_strm_st_adr output stream start address output stream start address

**VEPU\_swreg\_78**

Address: Operational Base + offset (0x0138)

output control start address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	output_ctrl_st_adr output control start address output control start address

**VEPU\_swreg\_79**

Address: Operational Base + offset (0x013c)

next picture luminance start address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	next_luma_st_adr next picture luminance start address next picture luminance start address

**VEPU\_swreg\_80**

Address: Operational Base + offset (0x0140)

Base address for MV output

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	mv_out_st_adr MV wr start address

**VEPU\_swreg\_81**

Address: Operational Base + offset (0x0144)

the cabac table start address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	cabac_table_st_adr the cabac table start address H264: cabac table VP8 : probability tables

**VEPU\_swreg\_82**

Address: Operational Base + offset (0x0148)

the first of ROI area register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RW	0x00	first_roi_tmb the top mb column for first roi area (inside area)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
23:16	RW	0x00	first_roi_bmb the bottom mb column for first roi area (outside area)
15:8	RW	0x00	first_roi_lmb the left mb column for first roi area $qp=qp + roi1\_Delta\_Qp$ (inside area)
7:0	RW	0x00	first_roi_rmb the right mb column for first roi area $qp=qp - roi1\_Delta\_Qp$ (outside area)

**VEPU\_swreg\_83**

Address: Operational Base + offset (0x014c)

the second of ROI area register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RW	0x00	second_roi_rmb (inside area)
23:16	RW	0x00	second_roi_bmb (outside area)
15:8	RW	0x00	second_roi_lmb $qp=qp + roi1\_Delta\_Qp$ (inside area)
7:0	RW	0x00	second_roi_tmb $qp=qp - roi1\_Delta\_Qp$ (outside area)

**VEPU\_swreg\_84**

Address: Operational Base + offset (0x0150)

Stabilization matrix

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x0	reserved
23:0	RW	0x000000	stab_matrix1 the 1st output of Stabilization matrix (position@ up-left)

**VEPU\_swreg\_85**

Address: Operational Base + offset (0x0154)

Stabilization matrix

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x0	reserved
23:0	RW	0x000000	stab_matrix2 the 2st output of Stabilization matrix (position @ up)

**VEPU\_swreg\_86**

Address: Operational Base + offset (0x0158)

Stabilization matrix

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x0	reserved
23:0	RW	0x000000	stab_matrix3 the 3st output of Stabilization matrix (position @up-right)

**VEPU\_swreg\_87**

Address: Operational Base + offset (0x015c)

Stabilization matrix

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x0	reserved
23:0	RW	0x000000	stab_matrix4 the 4st output of Stabilization matrix (position @ left)

**VEPU\_swreg\_88**

Address: Operational Base + offset (0x0160)

Stabilization matrix

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x0	reserved
23:0	RW	0x000000	stab_matrix5 the 5st output of Stabilization matrix (position @GMV )

**VEPU\_swreg\_89**

Address: Operational Base + offset (0x0164)

Stabilization matrix

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x0	reserved
23:0	RW	0x000000	stab_matrix6 the 6st output of Stabilization matrix (position@right )

**VEPU\_swreg\_90**

Address: Operational Base + offset (0x0168)

Stabilization matrix

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x0	reserved
23:0	RW	0x000000	stab_matrix7 the 7st output of Stabilization matrix (position@down-left)

**VEPU\_swreg\_91**

Address: Operational Base + offset (0x016c)

Stabilization matrix

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x0	reserved
23:0	RW	0x000000	stab_matrix8 the 8st output of Stabilization matrix (position@down )

**VEPU\_swreg\_92**

Address: Operational Base + offset (0x0170)

Stabilization matrix

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:26	RW	0x00	stab_gmv_vrtl the output of Stabilization GMV vertical signed register range : -16~16
25:24	RO	0x0	reserved
23:0	RW	0x000000	stab_matrix9 the 9st output of Stabilization matrix (position@down-right )

**VEPU\_swreg\_93**

Address: Operational Base + offset (0x0174)

the output of Stabilization motion sum

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	stab_motion_sum the output of Stabilization motion sum read value = (real value)/8 range : 0~1089*253*255*53/8

**VEPU\_swreg\_94**

Address: Operational Base + offset (0x0178)

output of Stabilization

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:8	RW	0x000000	stab_min_value the minimum value output of Stabilization range : 0~255*253*253
7:6	RW	0x0	stab_mod_sel the mode select of Stabilization 0 : disabled 1 : stab only 2 : stab+encode

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
5:0	RW	0x00	stab_hor_gmv the horizontal output of Stabilization GMV signed register range : -16~16

**VEPU\_swreg\_95**

Address: Operational Base + offset (0x017c)

RGB to YUV conversion coefficient register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	rgb2yuv_coe2 the 2st conversion coefficien for RGB to YUV the 2st conversion coefficien for RGB to YUV
15:0	RW	0x0000	rgb2yuv_coe1 the 1st conversion coefficien for RGB to YUV the 1st conversion coefficien for RGB to YUV

**VEPU\_swreg\_96**

Address: Operational Base + offset (0x0180)

RGB to YUV conversion coefficient register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	rgb2yuv_coe5 the 5st conversion coefficien for RGB to YUV the 5st conversion coefficien for RGB to YUV
15:0	RW	0x0000	rgb2yuv_coe3 the 3st conversion coefficien for RGB to YUV the 3st conversion coefficien for RGB to YUV

**VEPU\_swreg\_97**

Address: Operational Base + offset (0x0184)

RGB to YUV conversion coefficient register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RO	0x0	reserved
15:0	RW	0x0000	rgb2yuv_coe6 the 6st conversion coefficien for RGB to YUV the 6st conversion coefficien for RGB to YUV

**VEPU\_swreg\_98**

Address: Operational Base + offset (0x0188)

RGA MASK

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:21	RO	0x0	reserved
20:16	RW	0x00	bcmpt_mask_position the mask msb bit position of rgb B-component range : 0~31
15:13	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
12:8	RW	0x00	gcmpt_mask_position the mask msb bit position of rgb G-component range : 0~31
7:5	RO	0x0	reserved
4:0	RW	0x00	rcmpt_mask_position the mask msb bit position of rgb R-component range : 0~31

**VEPU\_swreg\_99**

Address: Operational Base + offset (0x018c)

mv related

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RO	0x0	reserved
30:21	RW	0x000	mv_1p_ply 1p of differential MV penalty differential MV penalty for 1p
20:11	RW	0x000	mv_1p_4p_ply 1p or 4p of differential MV penalty ME. DMVPenaltyQp
10:1	RW	0x000	mv_4p_ply 4p of differential MV penalty
0	RW	0x0	mutimv_en on-off flag for using exceed one mv every mb on-off flag for using exceed one mv every mb

**VEPU\_swreg\_100\_reuse**

Address: Operational Base + offset (0x0190)

QP register/VP8 qp

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:26	RW	0x00	h264_init_luma_qp Initial luma qp used in h264 range: 0~51
25:20	RW	0x00	h264_max_qp H.264 Minimum QP range : 0~51
19:14	RW	0x00	h264_min_qp Minimum QP range:0~51
13	RO	0x0	reserved
12:0	RW	0x0000	h264_chkpt_distance checkpoint distance for macro block checkpoint distance for macro block

**VEPU\_swreg\_101\_read**

Address: Operational Base + offset (0x0194)

Register0003 Abstract

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:12	RO	0x1f522	HW_CONFIG Field0000 Description
11:0	RO	0x780	MAX_VID_WIDTH Field0000 Description

**VEPU\_swreg\_102**

Address: Operational Base + offset (0x0198)

mvc related

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RO	0x0	reserved
23:20	RW	0x0	mv_favor_16x16 Zero 16x16 MV favor value = (real value)/2.
19:11	RW	0x000	mv_ply_4x4 4x4 Mv Penalty
10:8	RW	0x0	mvc_view_id MVC view_id
7	RW	0x0	mvc_anchor_pic_flag to specifie picture is one part of anchor access unit
6:4	RW	0x0	mvc_priority_id MVC priority_id
3:1	RW	0x0	mvc_temporal_id MVC temporal_id
0	RW	0x0	mvc_inter_view_flag the inter-view prediction of picture MVC inter_view_flag.

**VEPU\_swreg\_103**

Address: Operational Base + offset (0x019c)

encoder start

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RO	0x0	reserved
28:20	RW	0x000	enc_height encoderd height lum height (macroblock unit) H264: [6..255] JPEG: [6..511]

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
19:17	RO	0x0	reserved
16:8	RW	0x000	enc_width the encoder width lum width (macroblock unit) H264: range : 9~255 JPEG: range : 6~511
7:6	RW	0x0	enc_frame_type frame type selected for current frame 0: INTER 1: INTRA(IDR) 2: MVC-INTER
5:4	RW	0x0	enc_fmt encoding format selected 1 : VP8, 2 : JPEG 3 : H264
3:1	RO	0x0	reserved
0	RW	0x0	enc_en encoder enable flag encoder enable

**VEPU\_swreg\_104**

Address: Operational Base + offset (0x01a0)  
mb control register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0000	mb_count_out mb_count_out mb_count_out
15:0	RW	0x0000	mb_cnt macroblock_count macroblock_count

**VEPU\_swreg\_105**

Address: Operational Base + offset (0x01a4)  
SWAP

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RW	0x0	swap8_in input swap 8-bits or not flag 0: no swap 1: swap 8bit

Bit	Attr	Reset Value	Description
30	RW	0x0	swap16_in input swap 16-bits or not flag 0: no swap 1: swap 16bit
29	RW	0x0	swap32_in input swap 32-bits or not flag 0: no swap 1: swap 32bit
28	RW	0x0	swap8_out output swap 8-bits or not flag 0: no swap 1: swap 8bit
27	RW	0x0	swap16_out output swap 16-bits or not flag 0: no swap 1: swap 16bit
26	RW	0x0	swap32_out output swap 32-bits or not flag 0: no swap 1: swap 32bit
25	RO	0x0	reserved
24	RW	0x0	test_irq test irq
23:20	RW	0x0	test_counter test counter test counter
19	RW	0x0	coher_test_reg test register coherency test register coherency
18	RW	0x0	coher_test_mem test memory coherency test memory coherency
17:0	RW	0x000000	test_len test data length

**VEPU\_swreg\_106\_reuse**

Address: Operational Base + offset (0x01a8)  
encoder control register 1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	pic_para_id H.264 picture parameter id set
23:16	RW	0x00	intra_pred_mode intra prediction previous fpr 4x4 mode favor used in h264 H.264 intra prediction previous 4x4 mode favor

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
15:0	RW	0x0000	frame_num H.264 frame number

**VEPU\_swreg\_107\_reuse**

Address: Operational Base + offset (0x01ac)

JPEG control register /Penalty

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved
29:20	RW	0x000	mv_ply_16x8_8x16 Penalty for using 16x8 or 8x16 MV. Penalty for using 16x8 or 8x16 MV
19:10	RW	0x000	mv_ply_8x8 Penalty for using 8x8 MV Penalty for using 8x8 MV
9:0	RW	0x000	mv_ply_8x4_4x8 Penalty for using 8x4 or 4x8 MV. Penalty for using 8x4 or 4x8 MV.

**VEPU\_swreg\_108\_reuse**

Address: Operational Base + offset (0x01b0)

intra\_slice\_bmp2/vp8 counters or probability address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	intra_slice_bmp2 Field0000 Abstract bit0 : slices64 bit1 : slices65 bit2 : slices66 ..... bit31 : slices95 VP8: VP8 counters or probability updates start address

**VEPU\_swreg\_109**

Address: Operational Base + offset (0x01b4)

Register0004 Abstract

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RO	0x0	reserved
28	RW	0x0	int_non Field0000 Abstract Field0000 Description
27:25	RO	0x0	reserved
24	RW	0x0	mv_sad_wren the each MB MV and SAD be writed to mv_wr_st_adr enable
23:21	RO	0x0	reserved

Bit	Attr	Reset Value	Description
20	RW	0x0	rocon_write_dis write reconstructed image disable flag
19:17	RO	0x0	reserved
16	RW	0x0	slice_rdyint_en enable slice ready interrupt enable slice ready interrupt
15:13	RO	0x0	reserved
12	RW	0x1	clk_gating_en clock gating enable flag default clk_gating_en =1'b1
11	RO	0x0	reserved
10	RW	0x0	int_timeout_en enable interrupt for timeout
9	RW	0x0	irq_clr irq clear
8	RW	0x0	irq_dis irq disable
7	RO	0x0	reserved
6	RW	0x0	irq_timeout HW wait timeout flag
5	RW	0x0	irq_buffer_full buffer full flag
4	RW	0x0	irq_bus_error bus error irq
3	RW	0x0	fuse_int Field0000 Abstract Field0000 Description
2	RW	0x0	irq_slice_ready slice ready flag
1	RW	0x0	irq_frame_rdy one frame encoder success flag
0	RW	0x0	enc_irq enc interrupt

**VEPU\_swreg\_110\_read**

Address: Operational Base + offset (0x01b8)

product ID

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RO	0x4831	prod_id Product ID
15:12	RO	0x1	major_num Major number
11:4	RO	0x22	minor_num Minor number
3:0	RO	0x0	synthesis

**VEPU\_swreg\_120\_183**

Address: Operational Base + offset (0x01e0)

DMV\_4p\_1p\_penalty

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	WO	0x00000000	dmv_ply_table DMV 4p/1p penalty table values addr range : 0x01e0~0x02dc swreg120: DMV 4p/1p penalty table values swreg121: DMV 4p/1p penalty table values swreg122: DMV 4p/1p penalty table values swreg123: DMV 4p/1p penalty table values ..... swreg183: DMV 4p/1p penalty table values

**4.5.8 MMU Registers Summary**

<b>Name</b>	<b>Offset</b>	<b>Size</b>	<b>Reset Value</b>	<b>Description</b>
mmu_DTE_ADDR	0x0000	W	0x00000000	MMU current page Table address
mmu_STATUS	0x0004	W	0x00000018	MMU status register
mmu_COMMAND	0x0008	W	0x00000000	MMU command register
mmu_PAGE_FAULT_ADDR	0x000c	W	0x00000000	MMU logical address of last page fault
mmu_ZAP_ONE_LINE	0x0010	W	0x00000000	MMU Zap cache line register
mmu_INT_RAWSTAT	0x0014	W	0x00000000	MMU raw interrupt status register
mmu_INT_CLEAR	0x0018	W	0x00000000	MMU raw interrupt status register
mmu_INT_MASK	0x001c	W	0x00000000	MMU raw interrupt status register
mmu_INT_STATUS	0x0020	W	0x00000000	MMU raw interrupt status register
mmu_AUTO_GATING	0x0024	W	0x00000001	mmu auto gating

Notes: Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access**4.5.9 MMU Detail Register Description****mmu\_DTE\_ADDR**

Address: Operational Base + offset (0x0000)

MMU current page Table address

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
------------	-------------	--------------------	--------------------

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	MMU_DTE_ADDR mmu dte base addr mmu dte base addr , the address must be 4kb aligned

**mmu\_STATUS**

Address: Operational Base + offset (0x0004)

MMU status register

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10:6	RO	0x00	PAGEFAULT_BUS_ID page fault bus id Index of master responsible for last page fault
5	RO	0x0	PAGE_FAULT_IS_WRITE page fault access The direction of access for last page fault: 0 = Read 1 = Write
4	RO	0x1	REPLAY_BUFFER_EMPTY replay buffer empty status 1'b1: The MMU replay buffer is empty
3	RO	0x1	MMU_IDLE mmu idle status The MMU is idle when accesses are being translated and there are no unfinished translated accesses. 1'b1: MMU is idle
2	RO	0x0	STAIL_ACTIVE stall active status MMU stall mode currently enabled. The mode is enabled by command 1'b1: MMU is in stall active status
1	RO	0x0	PAGE_FAULT_ACTIVE page fault active status MMU page fault mode currently enabled . The mode is enabled by command. 1'b1: page fault is active
0	RO	0x0	PAGING_ENABLED Paging enabled status 1'b0: paging is disabled 1'b1: Paging is enabled

**mmu\_COMMAND**

Address: Operational Base + offset (0x0008)

MMU command register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved

Bit	Attr	Reset Value	Description
2:0	WO	0x0	MMU_CMD Field0000 Abstract MMU_CMD. This can be: 0: MMU_ENABLE_PAGING 1: MMU_DISABLE_PAGING 2: MMU_ENABLE_STALL 3: MMU_DISABLE_STALL 4: MMU_ZAP_CACHE 5: MMU_PAGE_FAULT_DONE 6: MMU_FORCE_RESET

**mmu\_PAGE\_FAULT\_ADDR**

Address: Operational Base + offset (0x000c)

MMU logical address of last page fault

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	PAGE_FAULT_ADDR Field0000 Abstract address of last page fault

**mmu\_ZAP\_ONE\_LINE**

Address: Operational Base + offset (0x0010)

MMU Zap cache line register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	MMU_ZAP_ONE_LINE Field0000 Abstract address to be invalidated from the page table cache

**mmu\_INT\_RAWSTAT**

Address: Operational Base + offset (0x0014)

MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	READ_BUS_ERROR Field0000 Abstract read bus error
0	RW	0x0	PAGE_FAULT Field0000 Abstract page fault

**mmu\_INT\_CLEAR**

Address: Operational Base + offset (0x0018)

MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
1	WO	0x0	READ_BUS_ERROR Field0000 Abstract read bus error
0	WO	0x0	PAGE_FAULT Field0000 Abstract page fault

**mmu\_INT\_MASK**

Address: Operational Base + offset (0x001c)

MMU raw interrupt status register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RW	0x0	READ_BUS_ERROR Field0000 Abstract read bus error enable an interrupt source if the corresponding mask bit is set to 1
0	RW	0x0	PAGE_FAULT Field0000 Abstract page fault enable an interrupt source if the corresponding mask bit is set to 1

**mmu\_INT\_STATUS**

Address: Operational Base + offset (0x0020)

MMU raw interrupt status register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RO	0x0	READ_BUS_ERROR Field0000 Abstract read bus error
0	RO	0x0	PAGE_FAULT Field0000 Abstract page fault

**mmu\_AUTO\_GATING**

Address: Operational Base + offset (0x0024)

mmu auto gating

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:1	RO	0x0	reserved
0	RW	0x1	mmu_auto_gating mmu auto gating when it is 1'b1, the mmu will auto gating it self

#### 4.5.10 PREF CACHE Registers Summary

Name	Offset	Size	Reset Value	Description
pref_cache_VERSION	0x0000	W	0xcac20101	VERSION register
pref_cache_SIZE	0x0004	W	0x07110206	L2 cache SIZE
pref_cache_STATUS	0x0008	W	0x00000000	Status register
pref_cache_COMMAND	0x0010	W	0x00000000	Command setting register
pref_cache_CLEAR_PAGE	0x0014	W	0x00000000	clear page register
pref_cache_MAX_READS	0x0018	W	0x0000001c	maximum read register
pref_cache_ENABLE	0x001c	W	0x00000003	enables cacheable accesses and cache read allocation
pref_cache_PERFCNT_SR_C0	0x0020	W	0x00000000	performance counter 0 source register
pref_cache_PERFCNT_VAL_0	0x0024	W	0x00000000	performance counter 0 value register
pref_cache_PERFCNT_SR_C1	0x0028	W	0x00000000	performance counter 0 source register
pref_cache_PERFCNT_VAL_1	0x002c	W	0x00000000	performance counter 1 value register

Notes: Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

#### 4.5.11 PREF CACHE Detail Register Description

##### pref\_cache\_VERSION

Address: Operational Base + offset (0x0000)

VERSION register

Bit	Attr	Reset Value	Description
31:16	RO	0xcac2	PRODUCT_ID
15:8	RO	0x01	VERSION_MAJOR
7:0	RO	0x01	VERSION_MINOR

##### pref\_cache\_SIZE

Address: Operational Base + offset (0x0004)

L2 cache SIZE

Bit	Attr	Reset Value	Description
31:24	RO	0x07	External_bus_width Log2 external bus width in bits
23:16	RO	0x11	CACHE_SIZE Log2 cache size in bytes
15:8	RO	0x02	ASSOCIATIVITY Log2 associativity
7:0	RO	0x06	LINE_SIZE Log2 line size in bytes

##### pref\_cache\_STATUS

Address: Operational Base + offset (0x0008)

Status register

Bit	Attr	Reset Value	Description

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RW	0x0	DATA_BUSY set when the cache is busy handling data
0	RW	0x0	CMD_BUSY set when the cache is busy handling commands

**pref\_cache\_COMMAND**

Address: Operational Base + offset (0x0010)

Command setting register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:4	RO	0x0	reserved
3:0	RW	0x0	COMMAND The possible command is 1 = Clear entire cache

**pref\_cache\_CLEAR\_PAGE**

Address: Operational Base + offset (0x0014)

clear page register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	WO	0x00000000	CLEAR_PAGE writing an address, invalidates all lines in that page from the cache

**pref\_cache\_MAX\_READS**

Address: Operational Base + offset (0x0018)

maximum read register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:5	RO	0x0	reserved
4:0	RW	0x1c	MAX_READS Limit the number of outstanding read transactions to this amount

**pref\_cache\_ENABLE**

Address: Operational Base + offset (0x001c)

enables cacheable accesses and cache read allocation

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:4	RO	0x0	reserved
3	RW	0x0	sw_cache_clk_disgate cache clk disgate cache clk disgate when it is 1'b0, enable cache clk auto clkgating when it is 1'b1, disable cache clk auto clkgating
2	RW	0x0	sw_readbuffer_counter_reject_en counter reject enable default is 1'b0, for enhance cacheable read performance in readbuffer. 1'b1: normal origin counter reject

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
1	RW	0x1	permit_cache_read_allocate cache read allocate 1'b1: permit cache read allocate
0	RW	0x1	permit_cacheable_access cacheable access 1'b1: permit cacheable access

**pref\_cache\_PERFCNT\_SRC0**

Address: Operational Base + offset (0x0020)

performance counter 0 source register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:7	RO	0x0	reserved
6:0	RW	0x00	PERFCNT_SRC0 This register holds all the possible source values for Performance Counter 0 0: disabled 1: total clock cycles 2: active clock cycles 3: read transactions, master 4: word reads, master 5: read transactions, slave 6: word reads, slave 7: read hit, slave 8: read misses, slave 9: read invalidates, slave 10: cacheable read transactions, slave 11: bad hit number, slave

**pref\_cache\_PERFCNT\_VAL0**

Address: Operational Base + offset (0x0024)

performance counter 0 value register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	PERFCNT_VAL0 Performance counter 0 value

**pref\_cache\_PERFCNT\_SRC1**

Address: Operational Base + offset (0x0028)

performance counter 0 source register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:7	RO	0x0	reserved

Bit	Attr	Reset Value	Description
6:0	RW	0x00	<p>PERFCNT_SRC1</p> <p>This register holds all the possible source values for Performance Counter 1</p> <p>0: disabled 1: total clock cycles 2: active clock cycles 3: read transactions, master 4: word reads, master 5: read transactions, slave 6: word reads, slave 7: read hit, slave 8: read misses, slave 9: read invalidates, slave 10: cacheable read transactions, slave 11: bad hit number, slave</p>

**pref\_cache\_PERFCNT\_VAL1**

Address: Operational Base + offset (0x002c)  
 performance counter 1 value register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	<p>PERFCNT_VAL1</p> <p>Performance counter 1 value</p>

## 4.6 Interface Description

## 4.7 Application Notes

### 4.7.1 HEVC Configuration Flow

1. Prepare the data in the DDR.
2. Set the HEVC general system configuration in RKVDEC.swreg2, such as working mode, in/out endian.
3. Set the picture parameters with RKVDEC.swreg3.
4. Set the input and output data base address and HEVC reference configuration with RKVDEC.swreg4~RKVDEC.swreg43.
5. If CABAC error detection is desired, set the RKVDEC.swreg44 to enable the corresponding error detection.
6. Set the interrupt configuration and start the HEVC with HEVC.swreg1.
7. Wait for the frame interrupt, and then get the processed results in the target DDR
8. Clear all the interrupts, and repeat Process2~Process8 to start a new frame decoding if the decoding is not finished yet.

### 4.7.2 H264 Configuration Flow

1. Prepare the data in the DDR, for normal mode, we should prepare bitstream, tbl, pps and rps.
2. Set the h264 general system configuration in RKVDEC.swreg2, such as working mode, in/out endian.
3. Set the picture parameters with RKVDEC.swreg3.
4. Set the input and output data base address and H264 reference configuration with RKVDEC.swreg4~RKVDEC.swreg43.

5. If stream error detection is desired, set the swreg77\_h264\_error\_e and swreg44\_strmd\_error\_en to enable the corresponding error detection.
6. If prefetch function is desired, set the prefetch common registers and clear its TLB. Pay attention, there contains two caches, which are for Y channel and UV channel.
7. If MMU function is desired, set the MMU common registers and clear its TLB. Pay attention, there contains two MMUs, which are for read channel and write channel.
8. Set the interrupt configuration and start the decoder with RKVDEC.swreg1.
9. Wait for the frame interrupt, and then get the processed results in the target DDR. There may be decoded frame, error\_info, cur frame colmv output.  
When the stream mode is not frame by frame mode, we also wait buf empty, and then send the next pack, repeat it until sw\_dec\_rdy\_sta.
10. Clear all the interrupts, and repeat Process2~Process9 to start a new frame decoding if the decoding is not finished yet.

#### **4.7.3 VP9 Configuration Flow**

1. Prepare the data in the DDR, for normal mode, we should prepare bitstream, prob, and segid from last decoded frame.
2. Set the VP9 general system configuration in RKVDEC.swreg2, such as working mode, in/out endian.
3. Set the picture parameters with RKVDEC.swreg3.
4. Set the input and output data base address and HEVC reference configuration with RKVDEC.swreg4~RKVDEC.swreg43.
5. If stream error detection is desired, set the RKVDEC.swreg44 to enable the corresponding error detection.
6. If prefetch function is desired, set the prefetch common registers and clear its TLB. Pay attention, there contains two caches, which are for Y channel and UV channel.
7. If MMU function is desired, set the MMU common registers and clear its TLB. Pay attention, there contains two MMUs, which are for read channel and write channel.
8. Set the interrupt configuration and start the decoder with RKVDEC.swreg1.
9. Wait for the frame interrupt, and then get the processed results in the target DDR. There may be decoded frame, cur frame segid, cur frame count, cur frame colmv output.
10. Clear all the interrupts, and repeat Process2~Process9 to start a new frame decoding if the decoding is not finished yet.

#### **4.7.4 Other formats and encoder Configuration flow**

1. Prepare the decoder data in the DDR memory, and in decoder other than JPEG decoder, the input stream buffer should at least contain a slice or a frame data, otherwise the decoder will produce an interrupt and show error and then reset itself.

2. Config all the registers will be used. and please notice that which be list as follows:

In encoder---- We can configure the registers to control the input picture data format (such as endian and swap), but some input data format are fixed, such as cabac\_table data. And the register VEPU\_SWREG0~31 are JPEG quantization registers. They are write only registers. When you want to write these registers, you should first set VEPU\_SWREG103[0] to 1'b0 and VEPU\_SWREG103[5:4] to 2'b10( select JPEG mode).

In decoder---- The decoder can support ref buffer mode or cacheable mode, but they can't be both enabled. We can config the swreg57[28],swreg57[29] to enable cache and config the swreg65 to control the ref buffer.

4. You should config VDPU\_SWREG57[0] as 1'b1 to enable video decoder. And config VDPU\_SWREG41[0] as 1'b1 to enable pp. If pp performed in pipeline with decoder, you should config VDPU\_SWREG41[4] as 1'b1 and then config VDPU\_SWREG57[0] as 1'b1 to enable decoder and pp. VEPU\_SWREG103[0] set to 1'b1 to enable encoder.

5. Wait for the frame interrupt, and then check if the frame decoder ready interrupt is right or not, after that, you can get the processed results in the target DDR

6. Clear all the interrupts, repeat step 2~5 to start a new frame decoder or encoder.

## Chapter 5 Image Enhancement Processor (IEP)

### 5.1 Overview

The Image Enhancement Processor (IEP) receives data from and transmits data to system main memory by AXI bus.

The features of IEP are as follow:

- Image formatw
  - Input data: XRGB/RGB565/YUV420/YUV422
  - Output data: ARGB/RGB565/YUV420/YUV422
  - ARGB/XRGB/RGB565/YUV swap
  - YUV semi-planar/planar
  - BT601\_I/BT601\_f/BT709\_I/BT709\_f color space conversion
  - RGB dither up/down conversion
  - YUV up/down sampling conversion
  - Max resolution for static image up to 8192x8192
  - Max resolution for dynamic image up to 1920x1080
- Enhancement
  - Hue/Saturation/Brightness/Contrast enhancement
- De-interlace
  - Input 4 fields, output 2 frames mode
  - Input 4 fields, output 1 frames mode
  - Input 2 fields, output 1 frames mode
  - Programmable motion detection coefficient
  - Programmable high frequency factor
  - Programmable edge interpolation parameter
- Interface
  - 32bit AHB bus slave
  - 64bit AXI bus master
  - Combined interrupt output

## 5.2 Block Diagram

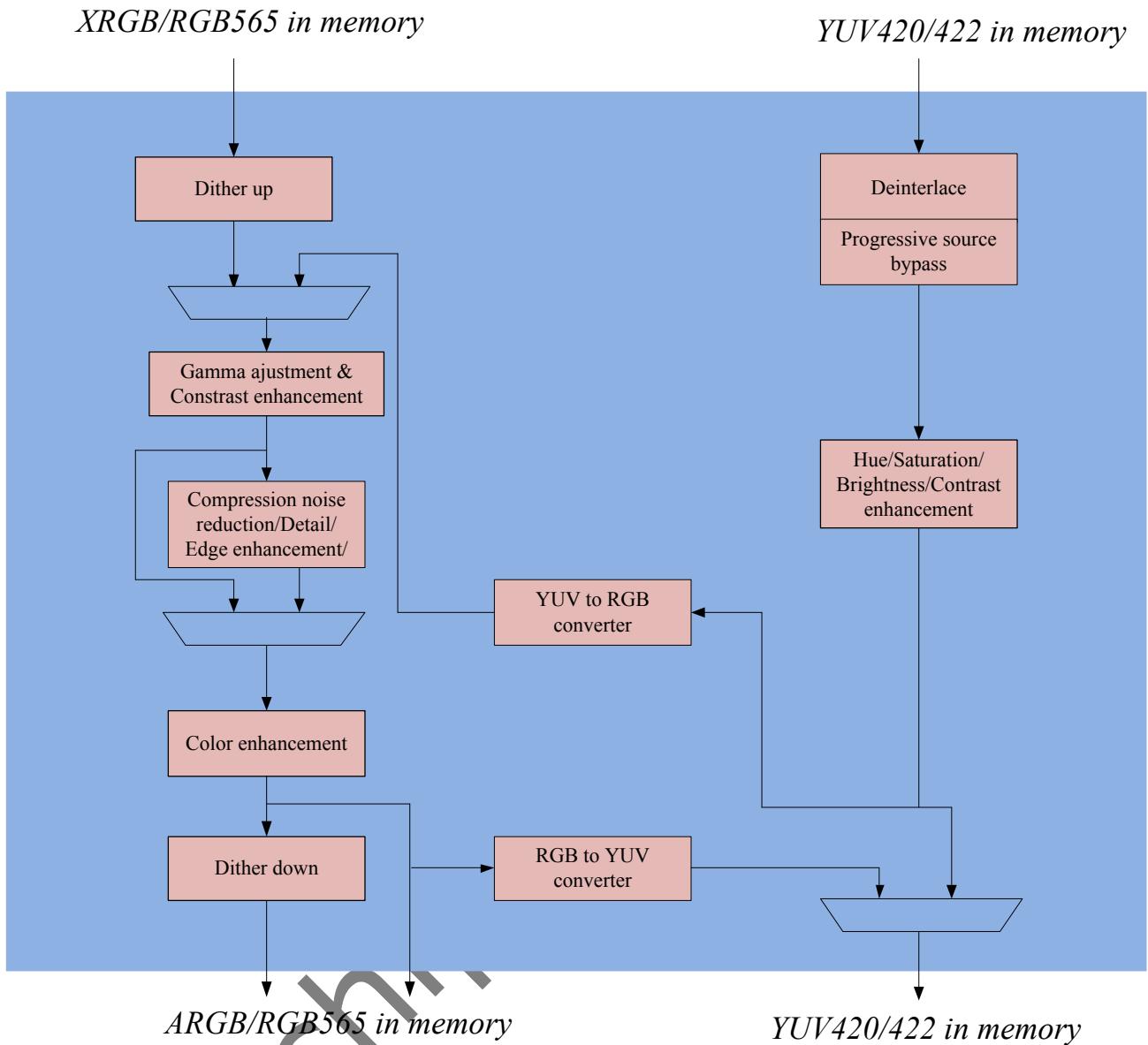


Fig. 7-52 IEP block diagram

The data path in IEP is in the previously diagram. The IEP comprises with:

- **Deinterlace**

There are five deinterlace mode including I4O2 (input 4 fields and output 2 frames once), I4O1B, I4O1T, I2O1B, I2O1T in the deinterlace block. YUV bypass is also supported. Pay attention if compression noise reduction, detail or edge enhancement work together with deinterlace is not allowed.

- **Enhancement**

Not only hue, saturation, brightness, contrast enhancement, but also blue screen, black screen and color bar are supported in YUV domain enhancement block. Gamma adjustment, edge enhancement, detail enhancement and color enhancement are supported in RGB domain enhancement block.

- **Noise Reduction**

Spatial and temporal sampling noise can be reduced in YUV domain noise reduction block. Compression noise can be reduced in RGB domain noise reduction block.

## 5.3 Function Description

### 5.3.1 Deinterlace

There are five deinterlace mode including I4O2, I4O1B, I4O1T, I2O1B and I2O1T in the deinterlace block. The I4O2 mode represents for 4 fields of input images and 2 frames of output images, so all of the two groups of source address registers and two groups of destination address registers need to be configured. For example, if source and destination format are both YUV420, the source address register IEP\_SRC\_ADDR\_YRGB, IEP\_SRC\_ADDR\_CBCR are used for source field0 and field 1, the source address register IEP\_SRC\_ADDR\_Y1, IEP\_SRC\_ADDR\_CBCR1 are used for source field2 and field3. The I4O1B and I4O1T mode have the same input images as the I4O2 mode, but only one frame output is generated once. The I2O1B and I2O1T mode have the same output as I4O1B and I4O1T mode, but only two fields input are needed. If bypass mode is selected, there are not any deinterlace operations. The parameter dil\_ei\_sel, dil\_ei\_radius, dil\_ei\_smooth, dil\_ei\_mode, dil\_hf\_en and dil\_hf\_fct in register IEP\_CONFIG0 and registers IEP\_DIL\_MTN\_TAB0~7 may have different influence in deinterlace effect depend on the type of the image source.

### 5.3.2 Noise reduction

Compression noise reduction is used for reducing the noise after the decompression of picture or video. Before the compression noise reduction is enabled, the IEP\_ENH\_DDE\_COE0/1 from address 0x400 to 0x5FC for difference and distance coefficients must be written firstly. The filter matrix can be selected from 3x3/5x5 and the filter weight can be programmed by configuring IEP\_ENH\_RGB\_CNFG.

### 5.3.3 Enhancement

Not only hue, saturation, brightness, contrast enhancement, but also blue screen, black screen and color bar are supported in this block. IEP\_ENH\_YUV\_CNFG\_0/1/2 registers can be configured to modify the YUV enhance parameters to satisfied with the requirement. Before the gamma adjustment or contrast enhancement is enabled in RGB domain, the IEP\_ENH(CG)\_TAB from address 0x100 to 0x3FC for B, G, R mapping must be written firstly. If the color enhancement is enabled, the IEP\_ENH\_C\_COE must be written the required value. Before the edge or detail enhancement is enabled, the IEP\_ENH\_DDE\_COE0/1 from address 0x400 to 0x5FC for difference and distance coefficients must be written firstly. The filter matrix can be selected from 3x3/5x5 and the filter weight can be programmed by configuring IEP\_ENH\_RGB\_CNFG.

### 5.3.4 Format conversion

The color space conversion either from RGB to YUV or from YUV to RGB has the selections including BT601/709\_L/F mode, and the input can be clipped or not.

If the source format is RGB565, dither up must be enabled. In contrary to the destination format is RGB565, dither down must be enabled.

### 5.3.5 Shadow registers

The configuration registers can be configured at any time, but they cannot have any effect immediately unless config\_done is available and a new frame\_start is enabled. The registers IEP\_RAW\_CONFIG0/1, IEP\_RAW\_VIR\_IMG\_WIDTH, IEP\_RAW\_IMG\_SCL\_FCT, IEP\_RAW\_SRC\_IMG\_SIZE, IEP\_RAW\_ENH\_YUV\_CNFG\_0/1/2 corresponding to the registers have the similar names but without letters \_RAW. They are used for raw register value reading before the configurations really have effect on the new frame.

### 5.3.6 VOP direct path

The IEP\_DST\_ADDR for DMA writing is useless if vop\_path\_en bit is set, because all RGB or YUV data is supplied for VOP directly from local bus via VOP and IEP.

## 5.4 Register Description

### 5.4.1 Registers Summary

Name	Offset	Size	Reset Value	Description
IEP_CONFIG0	0x0000	W	0x00000000	configuration register0
IEP_CONFIG1	0x0004	W	0x00000000	configuration register1
IEP_STATUS	0x0008	W	0x00000000	status register
IEP_INT	0x000c	W	0x00000000	interrupt register
IEP_FRM_START	0x0010	W	0x00000000	frame start
IEP_CONFIG_DONE	0x0018	W	0x00000000	configuration done
IEP_FRM_CNT	0x001c	W	0x00000000	frame counter
IEP_VIR_IMG_WIDTH	0x0020	W	0x01400140	Image virtual width
IEP_IMG_SCL_FCT	0x0024	W	0x20002000	scaling factor
IEP_SRC_IMG_SIZE	0x0028	W	0x00f00140	Source image width/height
IEP_DST_IMG_SIZE	0x002c	W	0x00f00140	Destination image width/height
IEP_DST_IMG_WIDTH_TILE0	0x0030	W	0x00000000	Destination image tile0 width
IEP_DST_IMG_WIDTH_TILE1	0x0034	W	0x00000000	Destination image tile1 width
IEP_DST_IMG_WIDTH_TILE2	0x0038	W	0x00000000	Destination image tile2 width
IEP_DST_IMG_WIDTH_TILE3	0x003c	W	0x00000000	Destination image tile3 width
IEP_ENH_YUV_CNFG_0	0x0040	W	0x00000000	brightness,contrast,saturation adjustment
IEP_ENH_YUV_CNFG_1	0x0044	W	0x00000000	Hue configuration
IEP_ENH_YUV_CNFG_2	0x0048	W	0x00000000	color bar configuration
IEP_ENH_RGB_CNFG	0x004c	W	0x00000000	enhancement RGB configuration
IEP_ENH_C_COE	0x0050	W	0x00000000	rgb color enhancement coefficient
IEP_RAW_CONFIG0	0x0058	W	0x00000000	configuration register0
IEP_RAW_CONFIG1	0x005c	W	0x00000000	configuration register1
IEP_RAW_VIR_IMG_WIDTH	0x0060	W	0x01400140	Image virtual width
IEP_RAW_IMG_SCL_FCT	0x0064	W	0x20002000	scaling factor
IEP_RAW_SRC_IMG_SIZE	0x0068	W	0x00f00140	Source image width/height
IEP_RAW_DST_IMG_SIZE	0x006c	W	0x00f00140	Destination image width/height
IEP_RAW_ENH_YUV_CNF_G_0	0x0070	W	0x00000000	brightness,contrast,saturation adjustment
IEP_RAW_ENH_YUV_CNF_G_1	0x0074	W	0x00000000	Hue configuration
IEP_RAW_ENH_YUV_CNF_G_2	0x0078	W	0x00000000	color bar configuration
IEP_RAW_ENH_RGB_CNF_G	0x007c	W	0x00000000	enhancement RGB configuration

IEP_SRC_ADDR_YRGB	0x0080	W	0x00000000	Start address of source image(Y/RGB)
IEP_SRC_ADDR_CBCR	0x0084	W	0x00000000	Start address of source image(Cb/Cr)
IEP_SRC_ADDR_CR	0x0088	W	0x00000000	Start address of source image(Cr)
IEP_SRC_ADDR_Y1	0x008c	W	0x00000000	Start address of source image(Y)
IEP_SRC_ADDR_CBCR1	0x0090	W	0x00000000	Start address of source image(Cb/Cr)
IEP_SRC_ADDR_CR1	0x0094	W	0x00000000	Start address of source image(Cr)
IEP_SRC_ADDR_Y_IITEMP	0x0098	W	0x00000000	Start address of source image(Y integer part)
IEP_SRC_ADDR_CBCR_ITEMP	0x009c	W	0x00000000	Start address of source image(CBCR integer part)
IEP_SRC_ADDR_CR_IITEMP	0x00a0	W	0x00000000	Start address of source image(CR integer part)
IEP_SRC_ADDR_Y_FTEMP	0x00a4	W	0x00000000	Start address of source image(Y fraction part)
IEP_SRC_ADDR_CBCR_FTEMP	0x00a8	W	0x00000000	Start address of source image(CBCR fraction part)
IEP_SRC_ADDR_CR_FTEMP	0x00ac	W	0x00000000	Start address of source image(CR fraction part)
IEP_DST_ADDR_YRGB	0x00b0	W	0x00000000	Start address of destination image(Y/RGB)
IEP_DST_ADDR_CBCR	0x00b4	W	0x00000000	Start address of destination image(Cb/Cr)
IEP_DST_ADDR_CR	0x00b8	W	0x00000000	Start address of destination image(Cr)
IEP_DST_ADDR_Y1	0x00bc	W	0x00000000	Start address of destination image(Y)
IEP_DST_ADDR_CBCR1	0x00c0	W	0x00000000	Start address of destination image(Cb/Cr)
IEP_DST_ADDR_CR1	0x00c4	W	0x00000000	Start address of destination image(Cr)
IEP_DST_ADDR_Y_IITEMP	0x00c8	W	0x00000000	Start address of destination image(Y integer part)
IEP_DST_ADDR_CBCR_ITEMP	0x00cc	W	0x00000000	Start address of destination image(CBCR integer part)
IEP_DST_ADDR_CR_IITEMP	0x00d0	W	0x00000000	Start address of destination image(CR integer part)
IEP_DST_ADDR_Y_FTEMP	0x00d4	W	0x00000000	Start address of destination image(Y fraction part)
IEP_DST_ADDR_CBCR_FTEMP	0x00d8	W	0x00000000	Start address of destination image(CBCR fraction part)
IEP_DST_ADDR_CR_FTEMP	0x00dc	W	0x00000000	Start address of destination image(CR fraction part)
IEP_DIL_MTN_TAB0	0x00e0	W	0x00000000	Deinterlace motion table0

IEP_DIL_MTN_TAB1	0x00e4	W	0x00000000	Deinterlace motion table1
IEP_DIL_MTN_TAB2	0x00e8	W	0x00000000	Deinterlace motion table2
IEP_DIL_MTN_TAB3	0x00ec	W	0x00000000	Deinterlace motion table3
IEP_DIL_MTN_TAB4	0x00f0	W	0x00000000	Deinterlace motion table4
IEP_DIL_MTN_TAB5	0x00f4	W	0x00000000	Deinterlace motion table5
IEP_DIL_MTN_TAB6	0x00f8	W	0x00000000	Deinterlace motion table6
IEP_DIL_MTN_TAB7	0x00fc	W	0x00000000	Deinterlace motion table7
IEP_ENH(CG)_TAB	0x0100	W	0x00000000	contrast and gamma enhancement table
IEP_ENH_DDE_COE0	0x0400	W	0x00000000	denoise,detail and edge enhancement coefficient
IEP_ENH_DDE_COE1	0x0500	W	0x00000000	denoise,detail and edge enhancement coefficient
IEP_MMU_DTE_ADDR	0x0800	W	0x00000000	MMU current page table address
IEP_MMU_STATUS	0x0804	W	0x00000018	MMU status register
IEP_MMU_CMD	0x0808	W	0x00000000	MMU command register
IEP_MMU_PAGE_FAULT_A DDR	0x080c	W	0x00000000	MMU logic address of last page fault
IEP_MMU_ZAP_ONE_LINE	0x0810	W	0x00000000	MMU zap cache line register
IEP_MMU_INT_RAWSTAT	0x0814	W	0x00000000	MMU raw interrupt status register
IEP_MMU_INT_CLEAR	0x0818	W	0x00000000	MMU interrupt clear register
IEP_MMU_INT_MASK	0x081c	W	0x00000000	MMU interrupt mask register
IEP_MMU_INT_STATUS	0x0820	W	0x00000000	MMU interrupt status register
IEP_MMU_AUTO_GATING	0x0824	W	0x00000001	MMU clock auto gating register

Notes: Size: **B**- Byte (8 bits) access, **HW**- Half WORD (16 bits) access, **W**-WORD (32 bits) access

#### 5.4.2 Detail Register Description

##### IEP\_CONFIG0

Address: Operational Base + offset (0x0000)  
configuration register0

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23	RW	0x0	dil_ei_sel deinterlace edge interpolation select
22:21	RW	0x0	dil_ei_radius deinterlace edge interpolation radius
20	RW	0x0	rgb_con_gam_order RGB contrast enhancement and gamma adjustment operation order select. 0:CG prior to DDE 1:DDE prior to CG (CG represent for contrast & gamma operation, and DDE represent for denoise, detail or edge enhancement operation)

19:18	RW	0x0	rgb_enh_sel RGB enhancement select 00: no operation 01: denoise 10: detail enhancement 11: edge enhancement
17	RW	0x0	rgb_con_gam_en RGB contrast enhancement and gamma adjustment enable 0:disable 1:enable
16	RW	0x0	rgb_color_enh_en RGB color enhancement enable 0:disable 1:enable
15	RW	0x0	dil_ei_smooth deinterlace edge interpolation for smooth effect 0: disable 1: enable
14	RW	0x0	yuv_enh_en yuv enhancement enable 0:disable 1:enable
13	RW	0x0	yuv_dns_en YUV 3D denoise enable 0:disable 1:enable
12	RW	0x0	dil_ei_mode deinterlace edge interpolation 0: disable 1: enable
11	RW	0x0	dil_hf_en deinterlace high frequency calculation enable 0: disable 1: enable
10:8	RW	0x0	dil_mode Deinterlace mode select: 000: YUV deinterlace and bypass path disable; 001: I4O2 mode 010: I4O1B mode 011: I4O1T mode 100: I2O1B mode 101: I2O1T mode 110: bypass mode
7:1	RW	0x00	dil_hf_fct deinterlace high frequency factor

0	RW	0x0	vop_path_en VOP direct path enable 0:disable 1:enable
---	----	-----	--

**IEP\_CONFIG1**

Address: Operational Base + offset (0x0004)

configuration register1

Bit	Attr	Reset Value	Description
31:24	RW	0x00	glb_alpha global alpha value only valid when destination format is ARGB
23	RW	0x0	rgb2yuv_input_clip RGB to YUV input range 0:R/G/B=[0,255] 1:R/G/B=[16,235]
22	RW	0x0	yuv2rgb_input_clip YUV to RGB input range 0:Y/U/V=[0,255] 1:Y=[16,235],U/V=[16,240]
21	RW	0x0	rgb_to_yuv_en RGB to YUV conversion enable 0: disable 1: enable
20	RW	0x0	yuv_to_rgb_en YUV to RGB conversion enable 0: disable 1: enable
19:18	RW	0x0	rgb2yuv_coe_sel rgb2yuv coefficient select 00:bt601_1 01:bt601_f 10:bt709_1 11:bt709_f
17:16	RW	0x0	yuv2rgb_coe_sel yuv2rgb coefficient select 00:bt601_1 01:bt601_f 10:bt709_1 11:bt709_f
15	RW	0x0	dthr_down_en dither down enable 0: disable 1: enable

14	RW	0x0	dthr_up_en dither up enable 0: disable 1: enable
13:12	RW	0x0	dst_yuv_swap destination YUV swap 00:SP UV 01:SP VU 10, 11:P
11:10	RW	0x0	dst_rgb_swap destination RGB swap ARGB destination 00:ARGB 01:ABGR 10:RGBA 11:BGRA RGB565 destination 00,10:RGB 01,11:BGR
9:8	RW	0x0	dst_fmt Output image Format 00 : ARGB 01 : RGB565 10 : YUV422 11 : YUV420
7:6	RO	0x0	reserved
5:4	RW	0x0	src_yuv_swap source YUV swap 00:SP UV 01:SP VU 10, 11:P
3:2	RW	0x0	src_rgb_swap source RGB swap XRGB source 00:XRGB 01:XBGR 10:RGBX 11:BGRX RGB565 source 00,10:RGB 01,11:BGR

1:0	RW	0x0	src_fmt Input image Format 00 : XRGB 01 : RGB565 10 : YUV422 11 : YUV420
-----	----	-----	---

**IEP\_STATUS**

Address: Operational Base + offset (0x0008)

status register

Bit	Attr	Reset Value	Description
31:20	RO	0x0	reserved
19	RO	0x0	rrgb_idle_ack RGB read DMA idle acknowlege
18	RO	0x0	wrgb_idle_ack RGB write DMA idle acknowlege
17	RO	0x0	ryuv_idle_ack YUV read DMA idle acknowlege
16	RO	0x0	wyuv_idle_ack YUV write DMA idle acknowlege
15:9	RO	0x0	reserved
8	RO	0x0	voi_sts vop direct path status 00:idle 01:working
7	RO	0x0	rrgb_sts RGB DMA read status 00:idle 01:working
6	RO	0x0	wrgb_sts RGB DMA write status 00:idle 01:working
5	RO	0x0	ryuv_sts YUV DMA read status 00:idle 01:working
4	RO	0x0	wyuv_sts YUV DMA write status 00:idle 01:working

3	RO	0x0	dde_sts RGB denoise/enhancement status 00:idle 01:working
2	RO	0x0	dil_sts de-interlace or yuv bypass status 00:idle 01:working
1	RO	0x0	scl_sts scaling status 00:idle 01:working
0	RO	0x0	dns_sts YUV 3D denoise status 00:idle 01:working

**IEP\_INT**

Address: Operational Base + offset (0x000c)  
interrupt register

Bit	Attr	Reset Value	Description
31:17	RO	0x0	reserved
16	W1 C	0x0	frm_done_int_clr Frame process done interrupt clear After be set to 1, this bit will be clear automatically.
15:9	RO	0x0	reserved
8	RW	0x0	frm_done_int_en Frame process done interrupt enable: 0: disable; 1: enable;
7:1	RO	0x0	reserved
0	RO	0x0	frm_done_int Frame process done interrupt 0: inactive; 1: active;

**IEP\_FRM\_START**

Address: Operational Base + offset (0x0010)  
frame start

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	W1 C	0x0	frm_start frame start Write 1, self clear.

**IEP\_CONFIG\_DONE**

Address: Operational Base + offset (0x0018)

configuration done

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:1	RO	0x0	reserved
0	RW	0x0	config_done configuration done Wait for frame start to update raw register configuration to really used registers.

**IEP\_FRM\_CNT**

Address: Operational Base + offset (0x001c)

frame counter

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	frm_cnt frame counter Self increase one after a frame operation is finished. Write arbitrary value to clear to zero.

**IEP\_VIR\_IMG\_WIDTH**

Address: Operational Base + offset (0x0020)

Image virtual width

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x0140	dst_vir_image_width Destination virtual image width
15:0	RW	0x0140	src_vir_image_width Source virtual image width

**IEP\_IMG\_SCL\_FCT**

Address: Operational Base + offset (0x0024)

scaling factor

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:16	RW	0x2000	vrt_scl_fct Vertical scale factor up scaling: $vrt\_scl\_fct=floor(src\_image\_height/dst\_image\_height);$ down scaling: $vrt\_scl\_fct=ceiling((dst\_image\_height+1)/(src\_image\_height+1));$
15:0	RW	0x2000	hrz_scl_fct Horizontal scale factor up scaling: $hrz\_scl\_fct=floor(src\_image\_width/dst\_image\_width);$ down scaling: $hrz\_scl\_fct=ceiling((dst\_image\_width+1)/(src\_image\_width+1));$

**IEP\_SRC\_IMG\_SIZE**

Address: Operational Base + offset (0x0028)

Source image width/height

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RO	0x0	reserved
28:16	RW	0x00f0	src_image_height source image height
15:13	RO	0x0	reserved
12:0	RW	0x0140	src_image_width source image width

**IEP\_DST\_IMG\_SIZE**

Address: Operational Base + offset (0x002c)

Destination image width/height

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:29	RO	0x0	reserved
28:16	RW	0x00f0	dst_image_height Destination image height
15:13	RO	0x0	reserved
12:0	RW	0x0140	dst_image_width Destination image width

**IEP\_DST\_IMG\_WIDTH\_TILE0**

Address: Operational Base + offset (0x0030)

Destination image tile0 width

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:12	RO	0x0	reserved
11:0	RW	0x000	dst_image_width_tile0 Destination image tile0 width

**IEP\_DST\_IMG\_WIDTH\_TILE1**

Address: Operational Base + offset (0x0034)

Destination image tile1 width

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:11	RO	0x0	reserved
10:0	RW	0x000	dst_image_width_tile1 Destination image tile1 width

**IEP\_DST\_IMG\_WIDTH\_TILE2**

Address: Operational Base + offset (0x0038)

Destination image tile2 width

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:11	RO	0x0	reserved
10:0	RW	0x000	dst_image_width_tile2 Destination image tile2 width

**IEP\_DST\_IMG\_WIDTH\_TILE3**

Address: Operational Base + offset (0x003c)

Destination image tile3 width

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:11	RO	0x0	reserved
10:0	RW	0x000	dst_image_width_tile3 Destination image tile3 width

**IEP\_ENH\_YUV\_CNFG\_0**

Address: Operational Base + offset (0x0040)

brightness,contrast,saturation adjustment

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:25	RO	0x0	reserved
24:16	RW	0x000	sat_con YUV saturation and contrast adjustment saturation * contrast range from 0 to 1.992*1.992, and this value is saturation* contrast * 128
15:8	RW	0x00	contrast YUV contrast adjustment contrast value range from 0 to 1.992, and this value is contrast*128.
7:6	RO	0x0	reserved
5:0	RW	0x00	brightness YUV brightness adjustment range from -32 to 31 000000:0; 000001:1; ..... 011111:31; 100000:-32; 100001:-31; ..... 111110:-2; 111111:-1;

**IEP\_ENH\_YUV\_CNFG\_1**

Address: Operational Base + offset (0x0044)

Hue configuration

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
------------	-------------	--------------------	--------------------

31:16	RO	0x0	reserved
15:8	RW	0x00	cos_hue the cos function value for hue adjustment sin function value range from 0.866 to 1 ,and this value is cos * 128 ,no sign bit
7:0	RW	0x00	sin_hue the sin function value for hue adjustment sin function value range from -0.5 to 0.5 ,and this value is sin * 128 ,and the high bit is sign bit

**IEP\_ENH\_YUV\_CNFG\_2**

Address: Operational Base + offset (0x0048)

color bar configuration

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved
25:24	RW	0x0	video_mode video mode 00:black screen 01:blue screen 10:color bars 11:normal video
23:16	RW	0x00	color_bar_v color bar v value
15:8	RW	0x00	color_bar_u color bar u value
7:0	RW	0x00	color_bar_y color bar y value

**IEP\_ENH\_RGB\_CNFG**

Address: Operational Base + offset (0x004c)

enhancement RGB configuration

Bit	Attr	Reset Value	Description
31:30	RW	0x0	luma_spat_sel 3D denoise luma spatial coefficient select
29:28	RW	0x0	luma_temp_sel 3D denoise luma temporal coefficient select
27:26	RW	0x0	chroma_spat_sel 3D denoise chroma spatial coefficient select
25:24	RW	0x0	chroma_temp_sel 3D denoise chroma temporal coefficient select

23:16	RW	0x00	enh_threshold enhancement threshold In denoise and detail enhancement operation, more than the threshold, considering as detail; but if less than the threshold, considering as noise, need to be filtered.
15	RO	0x0	reserved
14:8	RW	0x00	enh_alpha enhancement alpha value 0000000:0 0000001:1/16 0000010:2/16 ..... 0001111:15/16 0010000:1 0010001:1+1/16; 0010010:1+2/16; 0010011:1+3/16; ..... 0100000:2; ..... 0110000:3; ..... 1000000:4; ..... 1010000:5; ..... 1100000:6; other : reserved
7:2	RO	0x0	reserved
1:0	RW	0x0	enh_radius enhancement radius 00:R=1 01:R=2 10:R=3 11:R=4

**IEP\_ENH\_C\_COE**

Address: Operational Base + offset (0x0050)

rgb color enhancement coefficient

Bit	Attr	Reset Value	Description
31:7	RO	0x0	reserved
6:5	RW	0x0	c_int_coe color enhancement integer coefficient

4:0	RW	0x00	c_frac_coe color enhancement fraction coefficient
-----	----	------	--

**IEP\_RAW\_CONFIG0**

Address: Operational Base + offset (0x0058)

configuration register0

Bit	Attr	Reset Value	Description
31:24	RO	0x0	reserved
23	RO	0x0	dil_ei_sel deinterlace edge interpolation select
22:21	RO	0x0	dil_ei_radius deinterlace edge interpolation radius
20	RO	0x0	rgb_con_gam_order RGB contrast enhancement and gamma adjustment operation order select. 0:CG prior to DDE 1:DDE prior to CG (CG represent for contrast & gamma operation, and DDE represent for de-noise, detail or edge enhancement operation)
19:18	RO	0x0	rgb_enh_sel RGB enhancement select 00: no operation 01: denoise 10: detail enhancement 11: edge enhancement
17	RO	0x0	rgb_con_gam_en RGB contrast enhancement and gamma adjustment enable 0:disable 1:enable
16	RO	0x0	rgb_color_enh_en RGB color enhancement enable 0:disable 1:enable
15	RO	0x0	dil_ei_smooth deinterlace edge interpolation for smooth effect 0: disable 1: enable
14	RO	0x0	yuv_enh_en yuv enhancement enable 0:disable 1:enable

13	RO	0x0	yuv_dns_en YUV 3D denoise enable 0:disable 1:enable
12	RO	0x0	dil_ei_mode deinterlace edge interpolation 0: disable 1: enable
11	RO	0x0	dil_hf_en deinterlace high frequency calculation enable 0: disable 1: enable
10:8	RO	0x0	dil_mode Deinterlace mode select: 000: YUV deinterlace and bypass path disable; 001: I4O2 mode 010: I4O1B mode 011: I4O1T mode 100: I2O1B mode 101: I2O1T mode 110: bypass mode
7:1	RO	0x00	dil_hf_fct deinterlace high frequency factor
0	WO	0x0	vop_path_en VOP direct path enable 0:disable 1:enable

**IEP\_RAW\_CONFIG1**

Address: Operational Base + offset (0x005c)

configuration register1

Bit	Attr	Reset Value	Description
31:24	RO	0x00	glb_alpha global alpha value only valid when destination format is ARGB
23	RO	0x0	rgb2yuv_input_clip RGB to YUV input range 0:R/G/B=[0,255] 1:R/G/B=[16,235]
22	RO	0x0	yuv2rgb_input_clip YUV to RGB input range 0:Y/U/V=[0,255] 1:Y=[16,235],U/V=[16,240]

21	RO	0x0	rgb_to_yuv_en RGB to YUV conversion enable 0: disable 1: enable
20	RO	0x0	yuv_to_rgb_en YUV to RGB conversion enable 0: disable 1: enable
19:18	RO	0x0	rgb2yuv_coe_sel rgb2yuv coefficient select 00:bt601_1 01:bt601_f 10:bt709_1 11:bt709_f
17:16	RO	0x0	yuv2rgb_coe_sel yuv2rgb coefficient select 00:bt601_1 01:bt601_f 10:bt709_1 11:bt709_f
15	RO	0x0	dthr_down_en dither down enable 0: disable 1: enable
14	RO	0x0	dthr_up_en dither up enable 0: disable 1: enable
13:12	RO	0x0	dst_yuv_swap destination YUV swap 00:SP UV 01:SP VU 10, 11:P
11:10	RO	0x0	dst_rgb_swap destination RGB swap ARGB destination 00:ARGB 01:ABGR 10:RGBA 11:BGRA RGB565 destination 00,10:RGB 01,11:BGR

9:8	RO	0x0	dst_fmt Output image Format 00 : ARGB 01 : RGB565 10 : YUV422 11 : YUV420
7:6	RO	0x0	reserved
5:4	RO	0x0	src_yuv_swap source YUV swap 00:SP UV 01:SP VU 10, 11:P
3:2	RO	0x0	src_rgb_swap source RGB swap XRGB source 00:XRGB 01:XBGR 10:RGBX 11:BGRX RGB565 source 00,10:RGB 01,11:BGR
1:0	RO	0x0	src_fmt Input image Format 00 : XRGB 01 : RGB565 10 : YUV422 11 : YUV420

**IEP\_RAW\_VIR\_IMG\_WIDTH**

Address: Operational Base + offset (0x0060)

Image virtual width

Bit	Attr	Reset Value	Description
31:16	RO	0x0140	dst_vir_image_width Destination virtual image width
15:0	RO	0x0140	src_vir_image_width Source virtual image width

**IEP\_RAW\_IMG\_SCL\_FCT**

Address: Operational Base + offset (0x0064)

scaling factor

Bit	Attr	Reset Value	Description
-----	------	-------------	-------------

31:16	RO	0x2000	vrt_scl_fct Vertical scale factor up scaling: $vrt\_scl\_fct=\text{floor}(\text{src\_image\_height}/\text{dst\_image\_height})$ ; down scaling: $vrt\_scl\_fct=\text{ceiling}((\text{dst\_image\_height}+1)/(\text{src\_image\_height}+1))$ ;
15:0	RO	0x2000	hrz_scl_fct Horizontal scale factor up scaling: $hrz\_scl\_fct=\text{floor}(\text{src\_image\_width}/\text{dst\_image\_width})$ ; down scaling: $hrz\_scl\_fct=\text{ceiling}((\text{dst\_image\_width}+1)/(\text{src\_image\_width}+1))$ ;

**IEP\_RAW\_SRC\_IMG\_SIZE**

Address: Operational Base + offset (0x0068)

Source image width/height

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:16	RO	0x00f0	src_image_height source image height
15:13	RO	0x0	reserved
12:0	RO	0x0140	src_image_width source image width

**IEP\_RAW\_DST\_IMG\_SIZE**

Address: Operational Base + offset (0x006c)

Destination image width/height

Bit	Attr	Reset Value	Description
31:29	RO	0x0	reserved
28:16	RO	0x00f0	dst_image_height Destination image height
15:13	RO	0x0	reserved
12:0	RO	0x0140	dst_image_width Destination image width

**IEP\_RAW\_ENH\_YUV\_CNFG\_0**

Address: Operational Base + offset (0x0070)

brightness,contrast,saturation adjustment

Bit	Attr	Reset Value	Description
31:25	RO	0x0	reserved

24:16	RO	0x000	sat_con YUV saturation and contrast adjustment saturation * contrast range from 0 to 1.992*1.992, and this value is saturation* contrast * 128
15:8	RO	0x00	contrast YUV contrast adjustment contrast value range from 0 to 1.992, and this value is contrast*128.
7:6	RO	0x0	reserved
5:0	RO	0x00	brightness YUV brightness adjustment range from -32 to 31 000000:0; 000001:1; ..... 011111:31; 100000:-32; 100001:-31; ..... 111110:-2; 111111:-1;

**IEP\_RAW\_ENH\_YUV\_CNFG\_1**

Address: Operational Base + offset (0x0074)

Hue configuration

Bit	Attr	Reset Value	Description
31:16	RO	0x0	reserved
15:8	RO	0x00	cos_hue the cos function value for hue adjustment sin function value range from 0.866 to 1 ,and this value is cos * 128 ,no sign bit
7:0	RO	0x00	sin_hue the sin function value for hue adjustment sin function value range from -0.5 to 0.5 ,and this value is sin * 128 ,and the high bit is sign bit

**IEP\_RAW\_ENH\_YUV\_CNFG\_2**

Address: Operational Base + offset (0x0078)

color bar configuration

Bit	Attr	Reset Value	Description
31:26	RO	0x0	reserved

25:24	RO	0x0	video_mode video mode 00:black screen 01:blue screen 10:color bars 11:normal video
23:16	RO	0x00	color_bar_v color bar v value
15:8	RO	0x00	color_bar_u color bar u value
7:0	RO	0x00	color_bar_y color bar y value

**IEP\_RAW\_ENH\_RGB\_CNFG**

Address: Operational Base + offset (0x007c)  
enhancement RGB configuration

Bit	Attr	Reset Value	Description
31:30	RO	0x0	luma_spat_sel 3D denoise luma spatial coefficient select
29:28	RO	0x0	luma_temp_sel 3D denoise luma temporal coefficient select
27:26	RO	0x0	chroma_spat_sel 3D denoise chroma spatial coefficient select
25:24	RW	0x0	chroma_temp_sel 3D denoise chroma temporal coefficient select
23:16	RO	0x00	enh_threshold enhancement threshold In denoise and detail enhancement operation, more than the threshold, considering as detail; but if less than the threshold, considering as noise, need to be filtered.
15	RO	0x0	reserved

			enh_alpha enhancement alpha value 0000000:0 0000001:1/16 0000010:2/16 ..... 0001111:15/16 0010000:1 0010001:1+1/16; 0010010:1+2/16; 0010011:1+3/16; ..... 0100000:2; ..... 0110000:3; ..... 1000000:4; ..... 1010000:5; ..... 1100000:6; other : reserved
14:8	RO	0x00	
7:2	RO	0x0	reserved
1:0	RO	0x0	enh_radius enhancement radius 00:R=1 01:R=2 10:R=3 11:R=4

**IEP\_SRC\_ADDR\_YRGB**

Address: Operational Base + offset (0x0080)

Start address of source image(Y/RGB)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	src_image_yrgb_mst Source image data YRGB start address in Memory

**IEP\_SRC\_ADDR\_CBCR**

Address: Operational Base + offset (0x0084)

Start address of source image(Cb/Cr)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	src_image_cbcrc_mst Source image data CbCr start address in Memory

**IEP\_SRC\_ADDR\_CR**

Address: Operational Base + offset (0x0088)

Start address of source image(Cr)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	src_image_cr_mst Source image data Cr start address in Memory

**IEP\_SRC\_ADDR\_Y1**

Address: Operational Base + offset (0x008c)

Start address of source image(Y)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	src_image_y_mst Source image data Y start address in Memory

**IEP\_SRC\_ADDR\_CBCR1**

Address: Operational Base + offset (0x0090)

Start address of source image(Cb/Cr)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	src_image_cbcr_mst Source image data CbCr start address in Memory

**IEP\_SRC\_ADDR\_CR1**

Address: Operational Base + offset (0x0094)

Start address of source image(Cr)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	src_image_cr_mst Source image data Cr start address in Memory

**IEP\_SRC\_ADDR\_Y\_ITEMP**

Address: Operational Base + offset (0x0098)

Start address of source image(Y integer part)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	src_image_y_mst_itemp Interger part source image data Y start address in Memory

**IEP\_SRC\_ADDR\_CBCR\_ITEMP**

Address: Operational Base + offset (0x009c)

Start address of source image(CBCR integer part)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	src_image_cbcr_mst_cbcr_itemp Interger part source image data CBCR start address in Memory

**IEP\_SRC\_ADDR\_CR\_ITEMP**

Address: Operational Base + offset (0x00a0)

Start address of source image(CR integer part)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	src_image_cr_mst_cr_itemp Integer part source image data CR start address in Memory

**IEP\_SRC\_ADDR\_Y\_FTEMP**

Address: Operational Base + offset (0x00a4)

Start address of source image(Y fraction part)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	src_image_y_mst_ftemp Fraction part source image data Y start address in Memory

**IEP\_SRC\_ADDR\_CBCR\_FTEMP**

Address: Operational Base + offset (0x00a8)

Start address of source image(CBCR fraction part)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	src_image_cbcr_mst_ftemp Fraction part source image data CBCR start address in Memory

**IEP\_SRC\_ADDR\_CR\_FTEMP**

Address: Operational Base + offset (0x00ac)

Start address of source image(CR fraction part)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	src_image_cr_mst_ftemp Fraction part source image data CR start address in Memory

**IEP\_DST\_ADDR\_YRGB**

Address: Operational Base + offset (0x00b0)

Start address of destination image(Y/RGB)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	dst_image_yrgb_mst Destination image data YRGB start address in Memory

**IEP\_DST\_ADDR\_CBCR**

Address: Operational Base + offset (0x00b4)

Start address of destination image(Cb/Cr)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>

31:0	RW	0x00000000	dst_image_cbc_r_mst Destination image data CBCR start address in Memory
------	----	------------	--

**IEP\_DST\_ADDR\_CR**

Address: Operational Base + offset (0x00b8)

Start address of destination image(Cr)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dst_image_cr_mst Destination image data CR start address in Memory

**IEP\_DST\_ADDR\_Y1**

Address: Operational Base + offset (0x00bc)

Start address of destination image(Y)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dst_image_y_mst Destination image data Y start address in Memory

**IEP\_DST\_ADDR\_CBCR1**

Address: Operational Base + offset (0x00c0)

Start address of destination image(Cb/Cr)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dst_image_cbc_r_mst Destination image data CbCr start address in Memory

**IEP\_DST\_ADDR\_CR1**

Address: Operational Base + offset (0x00c4)

Start address of destination image(Cr)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dst_image_cr_mst Destination image data Cr start address in Memory

**IEP\_DST\_ADDR\_Y\_ITEMP**

Address: Operational Base + offset (0x00c8)

Start address of destination image(Y integer part)

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	dst_image_y_mst_itemp Interger part destination image data Y start address in Memory

**IEP\_DST\_ADDR\_CBCR\_ITEMP**

Address: Operational Base + offset (0x00cc)

Start address of destination image(CBCR integer part)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	dst_image_cbcr_mst_itemp Int part destination image data CBCR start address in Memory

**IEP\_DST\_ADDR\_CR\_ITEMP**

Address: Operational Base + offset (0x00d0)

Start address of destination image(CR integer part)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	dst_image_cr_mst_itemp Interger part destination image data CR start address in Memory

**IEP\_DST\_ADDR\_Y\_FTEMP**

Address: Operational Base + offset (0x00d4)

Start address of destination image(Y fraction part)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	dst_image_y_mst_ftemp Fraction part destination image data Y start address in Memory

**IEP\_DST\_ADDR\_CBCR\_FTEMP**

Address: Operational Base + offset (0x00d8)

Start address of destination image(CBCR fraction part)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	dst_image_cbcr_mst_ftemp Fraction part destination image data CBCR start address in Mem

**IEP\_DST\_ADDR\_CR\_FTEMP**

Address: Operational Base + offset (0x00dc)

Start address of destination image(CR fraction part)

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	dst_image_cr_mst_ftemp Fraction part destination image data CR start address

**IEP\_DIL\_MTN\_TAB0**

Address: Operational Base + offset (0x00e0)

Deinterlace motion table0

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RO	0x0	reserved
30:24	RW	0x00	mtn_sub_tab3 motion sub table3
23	RO	0x0	reserved

22:16	RW	0x00	mtn_sub_tab2 motion sub table2
15	RO	0x0	reserved
14:8	RW	0x00	mtn_sub_tab1 motion sub table1
7	RO	0x0	reserved
6:0	RW	0x00	mtn_sub_tab0 motion sub table0

**IEP\_DIL\_MTN\_TAB1**

Address: Operational Base + offset (0x00e4)

Deinterlace motion table1

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:24	RW	0x00	mtn_sub_tab3 motion sub table3
23	RO	0x0	reserved
22:16	RW	0x00	mtn_sub_tab2 motion sub table2
15	RO	0x0	reserved
14:8	RW	0x00	mtn_sub_tab1 motion sub table1
7	RO	0x0	reserved
6:0	RW	0x00	mtn_sub_tab0 motion sub table0

**IEP\_DIL\_MTN\_TAB2**

Address: Operational Base + offset (0x00e8)

Deinterlace motion table2

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:24	RW	0x00	mtn_sub_tab3 motion sub table3
23	RO	0x0	reserved
22:16	RW	0x00	mtn_sub_tab2 motion sub table2
15	RO	0x0	reserved

14:8	RW	0x00	mtn_sub_tab1 motion sub table1
7	RO	0x0	reserved
6:0	RW	0x00	mtn_sub_tab0 motion sub table0

**IEP\_DIL\_MTN\_TAB3**

Address: Operational Base + offset (0x00ec)

Deinterlace motion table3

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RO	0x0	reserved
30:24	RW	0x00	mtn_sub_tab3 motion sub table3
23	RO	0x0	reserved
22:16	RW	0x00	mtn_sub_tab2 motion sub table2
15	RO	0x0	reserved
14:8	RW	0x00	mtn_sub_tab1 motion sub table1
7	RO	0x0	reserved
6:0	RW	0x00	mtn_sub_tab0 motion sub table0

**IEP\_DIL\_MTN\_TAB4**

Address: Operational Base + offset (0x00f0)

Deinterlace motion table4

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RO	0x0	reserved
30:24	RW	0x00	mtn_sub_tab3 motion sub table3
23	RO	0x0	reserved
22:16	RW	0x00	mtn_sub_tab2 motion sub table2
15	RO	0x0	reserved
14:8	RW	0x00	mtn_sub_tab1 motion sub table1
7	RO	0x0	reserved

6:0	RW	0x00	mtn_sub_tab0 motion sub table0
-----	----	------	-----------------------------------

**IEP\_DIL\_MTN\_TAB5**

Address: Operational Base + offset (0x00f4)

Deinterlace motion table5

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:24	RW	0x00	mtn_sub_tab3 motion sub table3
23	RO	0x0	reserved
22:16	RW	0x00	mtn_sub_tab2 motion sub table2
15	RO	0x0	reserved
14:8	RW	0x00	mtn_sub_tab1 motion sub table1
7	RO	0x0	reserved
6:0	RW	0x00	mtn_sub_tab0 motion sub table0

**IEP\_DIL\_MTN\_TAB6**

Address: Operational Base + offset (0x00f8)

Deinterlace motion table6

Bit	Attr	Reset Value	Description
31	RO	0x0	reserved
30:24	RW	0x00	mtn_sub_tab3 motion sub table3
23	RO	0x0	reserved
22:16	RW	0x00	mtn_sub_tab2 motion sub table2
15	RO	0x0	reserved
14:8	RW	0x00	mtn_sub_tab1 motion sub table1
7	RO	0x0	reserved
6:0	RW	0x00	mtn_sub_tab0 motion sub table0

**IEP\_DIL\_MTN\_TAB7**

Address: Operational Base + offset (0x00fc)

Deinterlace motion table7

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31	RO	0x0	reserved
30:24	RW	0x00	mtn_sub_tab3 motion sub table3
23	RO	0x0	reserved
22:16	RW	0x00	mtn_sub_tab2 motion sub table2
15	RO	0x0	reserved
14:8	RW	0x00	mtn_sub_tab1 motion sub table1
7	RO	0x0	reserved
6:0	RW	0x00	mtn_sub_tab0 motion sub table0

**IEP\_ENH(CG)\_TAB**

Address: Operational Base + offset (0x0100)

contrast and gamma enhancement table

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:24	RW	0x00	cg_tab_3 cg table 3 pixel value 3,7,11,15,.....mapping
23:16	RW	0x00	cg_tab_2 cg table 2 pixel value 2,6,10,14,.....mapping
15:8	RW	0x00	cg_tab_1 cg table 1 pixel value 1,5,9,13,.....mapping
7:0	RW	0x00	cg_tab_0 cg table 0 256x8bit contrast & gamma mapping table pixel value 0,4,8,12,.....mapping

**IEP\_ENH(DDE)\_COE0**

Address: Operational Base + offset (0x0400)

denoise,detail and edge enhancement coefficient

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:30	RO	0x0	reserved

29:24	RW	0x00	dde_coe_3 dde coefficient 3 coefficient number 3,7,11,15,.....
23:22	RO	0x0	reserved
21:16	RW	0x00	dde_coe_2 dde coefficient 2 coefficient number 2,6,10,14,.....
15:14	RO	0x0	reserved
13:8	RW	0x00	dde_coe_1 dde coefficient 1 coefficient number 1,5,9,13,.....
7:6	RO	0x0	reserved
5:0	RW	0x00	dde_coe_0 dde coefficient 0 256x6bit coefficient for denoise and detail enhancement coefficient number 0,4,8,12,.....

**IEP\_ENH\_DDE\_COE1**

Address: Operational Base + offset (0x0500)  
denoise,detail and edge enhancement coefficient

Bit	Attr	Reset Value	Description
31:30	RO	0x0	reserved
29:24	RW	0x00	dde_coe_3 dde coefficient 3 coefficient number 3,7,11,15,.....
23:22	RO	0x0	reserved
21:16	RW	0x00	dde_coe_2 dde coefficient 2 coefficient number 2,6,10,14,.....
15:14	RO	0x0	reserved
13:8	RW	0x00	dde_coe_1 dde coefficient 1 coefficient number 1,5,9,13,.....
7:6	RO	0x0	reserved
5:0	RW	0x00	dde_coe_0 dde coefficient 0 81x6bit coefficient for denoise and detail enhancement coefficient number 0,4,8,12,.....

**IEP\_MMU\_DTE\_ADDR**

Address: Operational Base + offset (0x0800)  
MMU current page table address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	mmu_dte_addr page table address

**IEP\_MMU\_STATUS**

Address: Operational Base + offset (0x0804)

MMU status register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:11	RO	0x0	reserved
10:6	RW	0x00	mmu_page_fault_bus_id Index of master responsible for the last page fault
5	RW	0x0	mmu_page_fault_is_write The direction of access for last page fault: 0: read 1: write
4	RW	0x1	mmu_replay_buffer_empty The MMU replay buffer is empty.
3	RW	0x1	mmu_idle the MMU is idle when accesses are being translated and there are no unfinished translated access. The MMU_IDLE signal only reports idle when the MMU processor is idle and accesses are active on the external bus. Note: the MMU can be idle in page fault mode.
2	RW	0x0	mmu_stall_active MMU stall mode currently enabled. The mode is enabled by command.
1	RW	0x0	mmu_page_fault_active MMU page fault mode currently enabled. The mode is enabled by command
0	RW	0x0	mmu_paging_enabled mmu paging is enabled

**IEP\_MMU\_CMD**

Address: Operational Base + offset (0x0808)

MMU command register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:3	RO	0x0	reserved
2:0	RW	0x0	mmu_cmd 0: MMU_ENABLE_PAGING. enable paging. 1: MMU_DISABLE_PAGING. disable paging. 2: MMU_ENABLE_STALL. turn on stall mode. 3: MMU_DISABLE_STALL. turn off stall mode. 4: MMU_ZAP_CACHE. zap the entire page table cache. 5: MMU_PAGE_FAULT_DONE. leave page fault mode. 6: MMU_FORCE_RESET. reset the mmu. The MMU_ENABLE_STALL command can always be issued. Other commands are ignored unless the MMU is idle or stalled.

**IEP\_MMU\_PAGE\_FAULT\_ADDR**

Address: Operational Base + offset (0x080c)

MMU logic address of last page fault

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:0	RW	0x00000000	mmu_page_fault_addr address of last page fault

**IEP\_MMU\_ZAP\_ONE\_LINE**

Address: Operational Base + offset (0x0810)

MMU zap cache line register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:1	RO	0x0	reserved
0	RW	0x0	mmu_zap_one_line address to be invalidated from the page table cache.

**IEP\_MMU\_INT\_RAWSTAT**

Address: Operational Base + offset (0x0814)

MMU raw interrupt status register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RW	0x0	read_bus_error read bus error
0	RW	0x0	page_fault page fault

**IEP\_MMU\_INT\_CLEAR**

Address: Operational Base + offset (0x0818)

MMU interrupt clear register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved
1	RW	0x0	read_bus_error_clear read bus error interrupt clear. write 1 to this register can clear read bus error interrupt.
0	RW	0x0	page_fault_clear page fault interrupt clear, write 1 to this register can clear page fault interrupt.

**IEP\_MMU\_INT\_MASK**

Address: Operational Base + offset (0x081c)

MMU interrupt mask register

<b>Bit</b>	<b>Attr</b>	<b>Reset Value</b>	<b>Description</b>
31:2	RO	0x0	reserved

1	RW	0x0	read_bus_error_int_en read bus error interrupt enable
0	RW	0x0	page_fault_int_en page fault interrupt enable

**IEP\_MMU\_INT\_STATUS**

Address: Operational Base + offset (0x0820)

MMU interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	read_bus_error read bus error interrupt
0	RW	0x0	page_fault page fault interrupt

**IEP\_MMU\_AUTO\_GATING**

Address: Operational Base + offset (0x0824)

MMU clock auto gating register

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x1	mmu_auto_gating mmu clock auto gating when it is 1, the mmu will auto gating itself

## 5.5 Application Notes

### 5.5.1 VOP path disabled configure flow

1. Keep IEP direct path disabled.
2. Configure all registers which are needed at any time.
3. Configure IEP\_CONFIG\_DONE.
4. Configure IEP\_FRM\_START.

### 5.5.2 VOP path enabled configure flow

1. Keep IEP direct path enabled.
2. Configure all IEP registers which are needed.
3. Configure VOP related registers which are needed.
4. Configure CONFIG\_DONE register in VOP only.
5. Wait for frame start from VOP and IEP direct path.

### 5.5.3 VOP path turn on flow

1. Configure all IEP registers which are needed.
2. Configure VOP related registers which are needed.
3. Enable IEP direct path.
4. Enable VOP direct path.
5. Configure CONFIG\_DONE register in VOP only.
6. Wait for frame start from VOP and IEP direct path.

### **5.5.4 VOP path turn off flow**

1. Disable VOP direct path.
2. Disable IEP direct path, so IEP do not receive any other CONFIG\_DONE and frame start from VOP immediately.
3. Configure CONFIG\_DONE register in VOP.
4. Wait for frame start from VOP and IEP direct path, so VOP quit direct path mode completely.
5. Configure IEP registers which are needed at any time.
6. Configure IEP\_CONFIG\_DONE.
7. Configure IEP\_FRM\_START, IEP is working at write back mode now.

Rockchip Confidential

## Chapter 6 Video Digital Analog Converter (VDAC)

### 6.1 Overview

INNOSILICON Video DAC PHY is a small-sized, 27~300MHz, 1-channel, 10bit, high-speed D/A converter optimized for video or graphic applications. This IP designed to support Component(Pr,Y,Pb),Composite(CVBS), and S-Video(Y,C) signal standards for “consumer quality”.

#### 6.1.1 Features

- 10-bit resolution
- Single channel
- Up to 300Msps throughput rate
- Programmable current output: 14.7mA~ 34.8mA with 64 adjustable steps
- Current consumption: 1mA @ $I_{out}$  = 14.7mA, 39mA @ $I_{out}$  = 34.8mA
- 57dBc SFDR @ $I_{out}$  = 14.7, fclk = 300MHz and fout = 5MHz; 45dBc SFDR @ $I_{out}$  = 34.8, fclk = 300MHz and fout = 5MHz;
- Clock frequency : 27MHz to 300MHz
- Cable connection detection
- Build-in bandgap reference
- 1.8V supply for analog and 1.0V supply for digital

### 6.2 Block Diagram

The architecture is shown in the following figure.

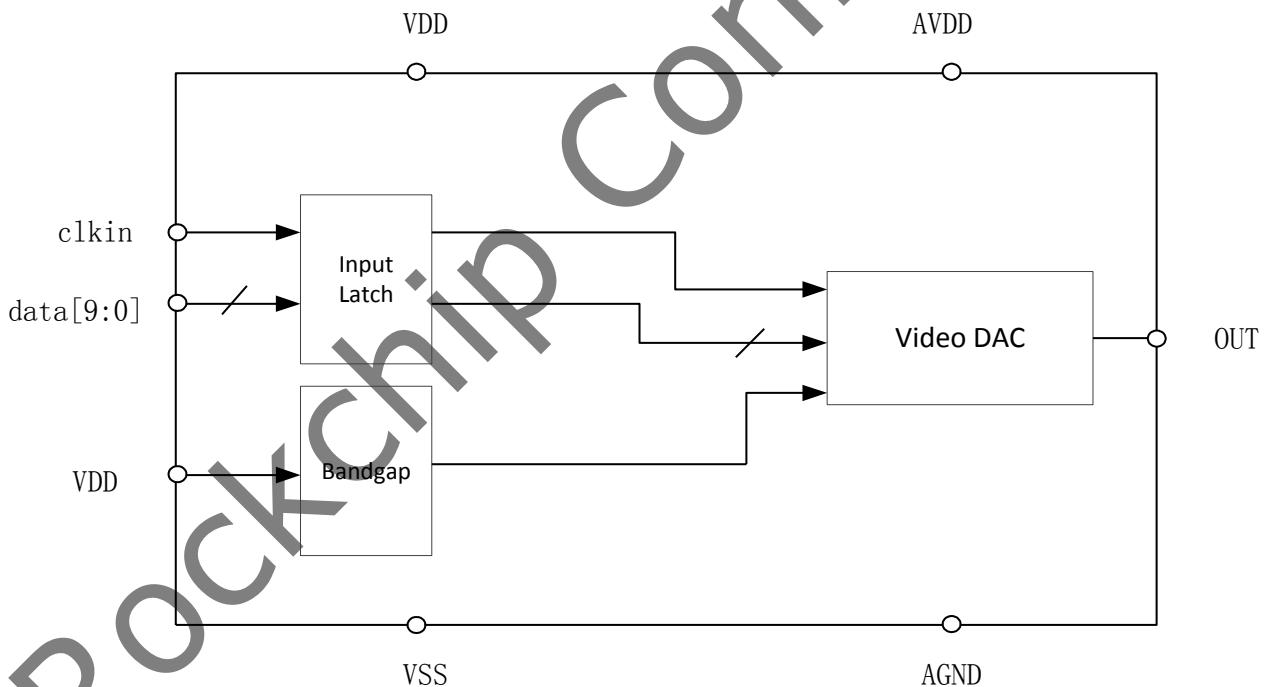


Fig. 7-53 VDAC Block Diagram

### 6.3 Function Description

#### 6.3.1 System configure write timing for apb bus

The Write transfer starts with the address, write data, write signl all changing after the rising edge of the clock. The first clock cycle of the transfer is called the SETUP cycle. After the following clock edge the enable signal PENABLE is asserted and this indicates that ENABLE cycle is taking place. The address, data and control signals all remain valid throughout the ENABLE cycle. The transfer completes at the end of this cycle.

The enable signal, PENABLE, will be de-asserted at the end of the transfer. The select signal

will also go LOW, unless the transfer is to be immediately followed by another transfer to the sample peripheral.

In order to reduce power consumption the address signal and the write signal will not change after a transfer until the next access occurs.

### 6.3.2 System configure read timing for apb bus

The timing of the address, write, select and strobe signals are all the same as for the write transfer. In the case of a read, the slave must provide the data during then ENABLE cycle. The data is sampled on the rising edge of clock at the end of the ENABLE cycle.

## 6.4 Register Description

### 6.4.1 Internal Address Mapping

Slave address can be divided into different length for different usage, which is shown as follows.

### 6.4.2 Registers Summary

Name	Offset	Size	Reset Value	Description
VDAC_VDAC0	0x0000	W	0x000000c0	VDAC0
VDAC_VDAC1	0x0280	W	0x00000070	VDAC1
VDAC_VDAC2	0x0284	W	0x00000020	VDAC2
VDAC_VDAC3	0x0288	W	0x00000030	VDAC3

Notes: **S**-ize: **B**- Byte (8 bits) access, **H**W- Half WORD (16 bits) access, **W**-WORD (32 bits) access

### 1.4.3 Detail Register Description

#### VDAC\_VDAC0

Address: Operational Base + offset (0x0000)

VDAC0

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7	RW	0x1	RST_ANA soft analog reset_n, low reset soft analog reset_n, low reset
6	RW	0x1	RST_DIG soft digital reset_n, low reset soft digital reset_n, low reset
5:0	RO	0x0	reserved

#### VDAC\_VDAC1

Address: Operational Base + offset (0x0280)

VDAC1

Bit	Attr	Reset Value	Description
31:8	RO	0x0	reserved
7:4	RW	0x7	CUR_REF select typical current reference select typical current reference
3:2	RO	0x0	reserved

Bit	Attr	Reset Value	Description
1	RW	0x0	DR_PWR_DOWN vdac driver power down vdac driver power down 1: power down 0: power on
0	RW	0x0	BG_PWR_DOWN vdac band gap power down vdac band gap power down 1: power down 0: power on

**VDAC\_VDAC2**

Address: Operational Base + offset (0x0284)

VDAC2

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:0	RW	0x20	CUR_CTR output current control for DAC output current control for DAC tvdac_sw[5:0]

**VDAC\_VDAC3**

Address: Operational Base + offset (0x0288)

VDAC3

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5	RW	0x1	CAB_EN Enable cable connection detection for DAC Enable cable connection detection for DAC 1: enable 0: disable
4	RW	0x1	CAB_REF reference voltage for cable disconnection detection of DAC reference voltage for cable disconnection detection of DAC 0: select 500mV 1: select 800mV
3:1	RO	0x0	reserved
0	RW	0x0	CAB_FLAG Field0000 Abstract status output for DAC cable connection detection (1 means cable disconnection)

**6.5 Application Notes****6.5.1 CABLE DETECTION**

The DAC channel contains a cable detection circuit to detect the cable plug condition. For

typical application, cable with  $75\Omega$  characteristic impedance is used and DAC output is terminated by  $75\Omega$  double termination. In such case, a  $75\Omega$  source termination resistance is connected to ground at DAC output end. The  $75\Omega$  source termination resistance combined with  $75\Omega$  load termination resistance results in an equivalent load resistance of  $37.5\Omega$ . Therefore, the equivalent load resistance for DAC output is  $37.5\Omega$  when cable is connected. It becomes  $75\Omega$  when cable is not connected. Compared to the case cable is connected, DAC output level will be twice in the case that cable is not connected with identical output current. To start cable detection, controller should enable this function (controlled by register `tvdac_dispdet_en`) and set the 10-bit input data for a DAC channel to be middle level. Then controller should select a proper reference voltage (controlled by register `tvdac_sw`), which will be compared with DAC output level to judge whether cable is connected or not. The reference voltage selection is shown in following table.

Tvdac_sw	Tvdac_dispdet_sel	Reference voltage
6'b000000~6'b011111	1'b0	500mV
6'b100000~6'b111111	1'b1	800mV

If DAC output level is larger than the reference voltage, the cable detection flag signal (`tvdac_dispdet`) will be high and it means cable is disconnected. Otherwise, the cable detection flag signal will be low and it means cable is connected.

Tvdac_dispdet	1	Cable is connected
	0	Cable is disconnected

### 6.5.2 TYPICAL CONFIGURATION

The typical configuration is shown in following figure. DAC output is connected through  $75\Omega$  cable with  $75\Omega$  double termination.

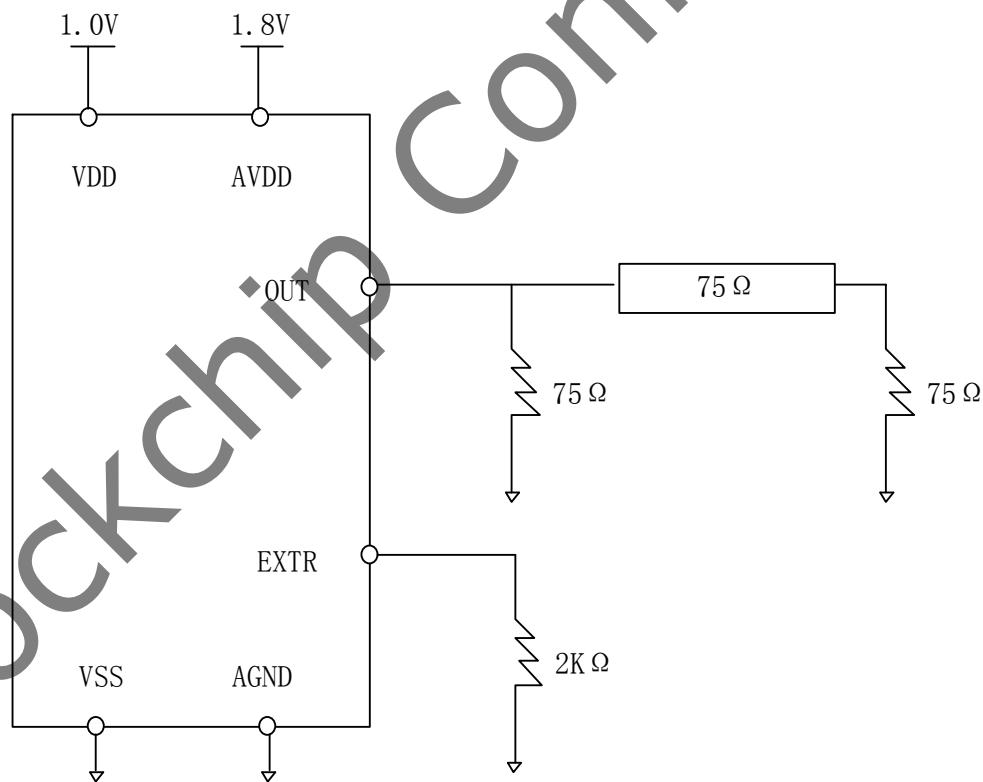


Fig. 7-54 VDAC Block Diagram

Analog supply AVDD should be connected to 1.8V power with decoupling. The digital supply VDD should be connected to digital core.

If external resistor is selected to produce reference current, EXTR should connect a  $2K\Omega$  resistor to ground.

Video DAC is suggested to placed close to the connector, in order to reduce signal noise and reflection due to impedance mismatch.

The DAC outputs are suggested to connect a  $75\Omega$  source termination resistance to ground.

The termination resistors should be placed close to video DAC outputs to minimize reflection.

Rockchip Confidential

## Chapter 7 Crypto

### 7.1 Overview

Crypto is a hardware accelerator of encrypting or decrypting. It supports the most commonly used algorithm: DES/3DES, AES, SHA1, SHA256, MD5 and RSA.

The Crypto supports following features:

- Support AES 128/192/256 bits key mode, ECB/CBC/CTR chain mode, Slave/FIFO mode
- Support DES/3DES (ECB and CBC chain mode), 3DES (EDE/ EEE key mode), Slave/FIFO mode
- Support SHA1/SHA256/MD5 (with hardware padding) HASH function, FIFO mode only
- Support 160 bit Pseudo Random Number Generator (PRNG)
- Support PKA 512/1024/2048 bit Exp Modulator
- Support up to 150M clock frequency

### 7.2 Block Diagram

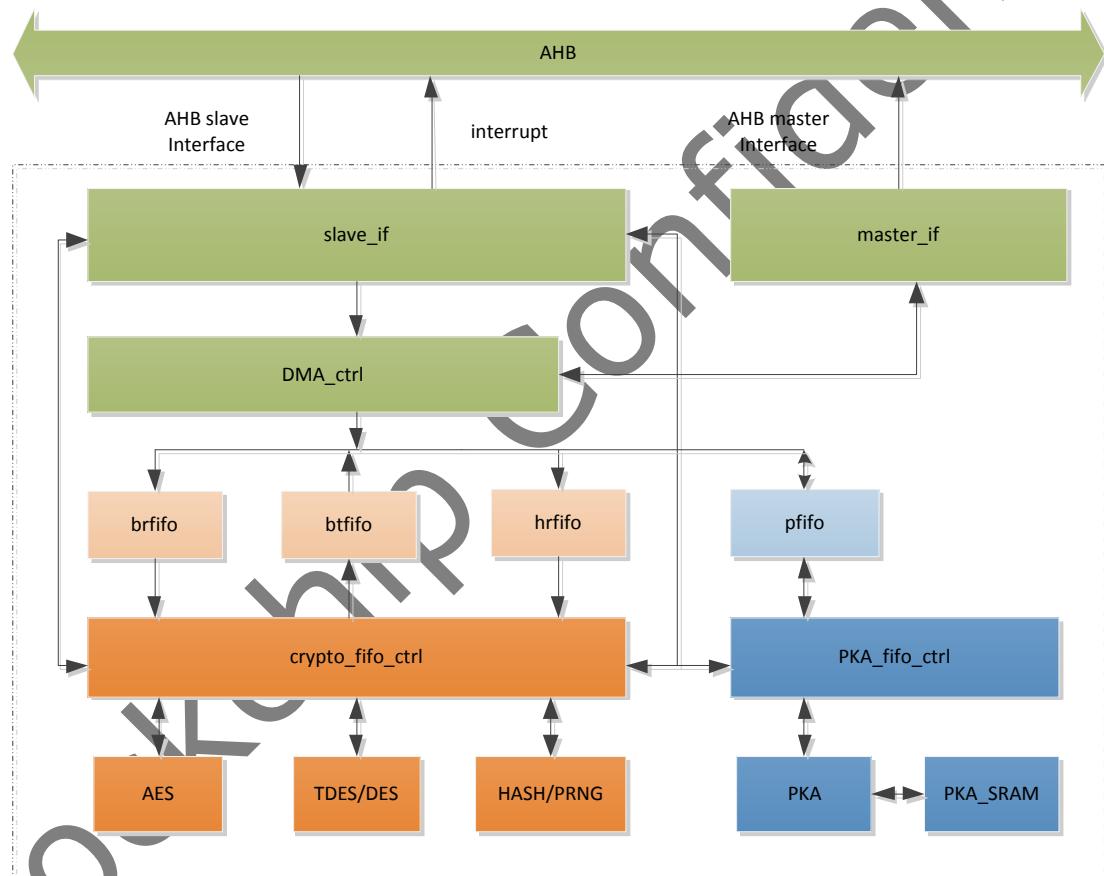


Fig. 7-1 Crypto Architecture

Figure above shows the architecture of Crypto.

## 7.3 Register description

### 7.3.1 Register Summary

CRYPTO_INTSTS	0x0000	W	0x00000000	Interrupt Status Register
CRYPTO_INTENA	0x0004	W	0x00000000	Interrupt Set Register
CRYPTO_CTRL	0x0008	W	0x00000000	Control Register
CRYPTO_CONF	0x000c	W	0x00000000	
CRYPTO_BRDMAS	0x0010	W	0x00000000	Block Receiving DMA Start Address Register
CRYPTO_BTDMAS	0x0014	W	0x00000000	Block Transmiting DMA Start Address Register
CRYPTO_BRDMAL	0x0018	W	0x00000000	Block Receiving DMA Length Register
CRYPTO_HRDMAS	0x001c	W	0x00000000	Hash Receiving DMA Start Address Register
CRYPTO_HRDMAL	0x0020	W	0x00000000	Hash Receiving DMA Length Register
CRYPTO_AES_CTRL	0x0080	W	0x00000000	AES Control Register
CRYPTO_AES_STS	0x0084	W	0x00000000	Status Register
CRYPTO_AES_DIN_0	0x0088	W	0x00000000	AES Input Data 0 Register
CRYPTO_AES_DIN_1	0x008c	W	0x00000000	AES Input Data 1 Register
CRYPTO_AES_DIN_2	0x0090	W	0x00000000	AES Input Data 2 Register
CRYPTO_AES_DIN_3	0x0094	W	0x00000000	AES Input Data 3 Register
CRYPTO_AES_DOUT_0	0x0098	W	0x00000000	AES Output Data 0 Register
CRYPTO_AES_DOUT_1	0x009c	W	0x00000000	AES Output Data 1 Register
CRYPTO_AES_DOUT_2	0x00a0	W	0x00000000	AES Output Data 2 Register
CRYPTO_AES_DOUT_3	0x00a4	W	0x00000000	AES Output Data 3 Register
CRYPTO_AES_IV_0	0x00a8	W	0x00000000	AES IV data 0 Register
CRYPTO_AES_IV_1	0x00ac	W	0x00000000	AES IV data 1 Register

CRYPTO_AES_IV_2	0x00b0	W	0x00000000	AES IV data 2 Register
CRYPTO_AES_IV_3	0x00b4	W	0x00000000	AES IV data 3 Register
CRYPTO_AES_KEY_0	0x00b8	W	0x00000000	AES Key data 0 Register
CRYPTO_AES_KEY_1	0x00bc	W	0x00000000	AES Key data 1 Register
CRYPTO_AES_KEY_2	0x00c0	W	0x00000000	AES Key data 2 Register
CRYPTO_AES_KEY_3	0x00c4	W	0x00000000	AES Key data 3 Register
CRYPTO_AES_KEY_4	0x00c8	W	0x00000000	AES Key data 4 Register
CRYPTO_AES_KEY_5	0x00cc	W	0x00000000	AES Key data 5 Register
CRYPTO_AES_KEY_6	0x00d0	W	0x00000000	AES Key data 6 Register
CRYPTO_AES_KEY_7	0x00d4	W	0x00000000	AES Key data 7 Register
CRYPTO_AES_CNT_0	0x00d8	W	0x00000000	AES Input Counter 0 Register
CRYPTO_AES_CNT_1	0x00dc	W	0x00000000	AES Input Counter 1 Register
CRYPTO_AES_CNT_2	0x00e0	W	0x00000000	AES Input Counter 2 Register
CRYPTO_AES_CNT_3	0x00e4	W	0x00000000	AES Input Counter 3 Register
CRYPTO_TDES_CTRL	0x0100	W	0x00000000	TDES Control Register
CRYPTO_TDES_STS	0x0104	W	0x00000000	Status Register
CRYPTO_TDES_DIN_0	0x0108	W	0x00000000	TDES Input Data 0 Register
CRYPTO_TDES_DIN_1	0x010c	W	0x00000000	TDES Input Data 1 Register
CRYPTO_TDES_DOU_T_0	0x0110	W	0x00000000	TDES Output Data 0 Register
CRYPTO_TDES_DOU_T_1	0x0114	W	0x00000000	TDES Output Data 1 Register
CRYPTO_TDES_IV_0	0x0118	W	0x00000000	TDES IV data 0 Register
CRYPTO_TDES_IV_1	0x011c	W	0x00000000	TDES IV data 1 Register
CRYPTO_TDES_KEY1_0	0x0120	W	0x00000000	TDES Key1 data 1 Register
CRYPTO_TDES_KEY1_1	0x0124	W	0x00000000	TDES Key1 data 1 Register
CRYPTO_TDES_KEY2_0	0x0128	W	0x00000000	TDES Key2 data 0 Register
CRYPTO_TDES_KEY2_1	0x012c	W	0x00000000	TDES Key2 data 1 Register

CRYPTO_TDES_KEY3_0	0x0130	W	0x00000000	TDES Key3 data 0 Register
CRYPTO_TDES_KEY3_1	0x0134	W	0x00000000	TDES Key3 data 1 Register
CRYPTO_HASH_CTRL	0x0180	W	0x00000000	Hash Control Register
CRYPTO_HASH_STS	0x0184	W	0x00000000	Hash Status Register
CRYPTO_HASH_MSG_LEN	0x0188	W	0x00000000	Hash Message Len
CRYPTO_HASH_DOUT_0	0x018c	W	0x00000000	Hash Result Register 0
CRYPTO_HASH_DOUT_1	0x0190	W	0x00000000	Hash Result Register 1
CRYPTO_HASH_DOUT_2	0x0194	W	0x00000000	Hash Result Register 2
CRYPTO_HASH_DOUT_3	0x0198	W	0x00000000	Hash Result Register 3
CRYPTO_HASH_DOUT_4	0x019c	W	0x00000000	Hash Result Register 4
CRYPTO_HASH_DOUT_5	0x01a0	W	0x00000000	Hash Result Register 4
CRYPTO_HASH_DOUT_6	0x01a4	W	0x00000000	Hash Result Register 6
CRYPTO_HASH_DOUT_7	0x01a8	W	0x00000000	Hash Result Register 7
CRYPTO_HASH_SEE_D_0	0x01ac	W	0x00000000	PRNG Seed/HMAC Key Register 0
CRYPTO_HASH_SEE_D_1	0x01b0	W	0x00000000	PRNG Seed/HMAC Key Register 1
CRYPTO_HASH_SEE_D_2	0x01b4	W	0x00000000	PRNG Seed/HMAC Key Register 2
CRYPTO_HASH_SEE_D_3	0x01b8	W	0x00000000	PRNG Seed/HMAC Key Register 3
CRYPTO_HASH_SEE_D_4	0x01bc	W	0x00000000	PRNG Seed/HMAC Key Register 4

CRYPTO_TRNG_CTRL	0x0200	W	0x00000000	TRNG Control
CRYPTO_TRNG_DOUT_0	0x0204	W	0x00000000	TRNG Output Data 0
CRYPTO_TRNG_DOUT_1	0x0208	W	0x00000000	TRNG Output Data 1
CRYPTO_TRNG_DOUT_2	0x020c	W	0x00000000	TRNG Output Data 2
CRYPTO_TRNG_DOUT_3	0x0210	W	0x00000000	TRNG Output Data 3
CRYPTO_TRNG_DOUT_4	0x0214	W	0x00000000	TRNG Output Data 4
CRYPTO_TRNG_DOUT_5	0x0218	W	0x00000000	TRNG Output Data 5
CRYPTO_TRNG_DOUT_6	0x021c	W	0x00000000	TRNG Output Data 6
CRYPTO_TRNG_DOUT_7	0x0220	W	0x00000000	TRNG Output Data 7
CRYPTO_PKA_CTRL	0x0280	W	0x00000000	PKA Control Register
CRYPTO_PKA_M	0x0400	W	0x00000000	
CRYPTO_PKA_C	0x0500	W	0x00000000	
CRYPTO_PKA_N	0x0600	W	0x00000000	
CRYPTO_PKA_E	0x0700	W	0x00000000	

Notes: Size : **B** - Byte (8 bits) access, **HW** - Half WORD (16 bits) access, **W** - WORD (32 bits) access

### 7.3.2 Detail Register Description

#### CRYPTO\_INTSTS

Address: Operational Base + offset (0x0000)  
Interrupt Status Register

			Descript ion
31:6	RO	0x0	reserved
5	RW	0x0	PKA_DONE_INT PKA Done Interrupt
4	W1C	0x0	HASH_DONE_INT Hash Done Interrupt
3	W1C	0x0	HRDMA_ERR_INT Specifies the interrupt of hash receiving DMA Error
2	W1C	0x0	HRDMA_DONE_INT Specifies the interrupt of hash receiving DMA DONE
1	W1C	0x0	BCDMA_ERR_INT Specifies the interrupt of block cipher Error
0	W1C	0x0	BCDMA_DONE_INT Specifies the interrupt of block cipher DONE

**CRYPTO\_INTENA**

Address: Operational Base + offset (0x0004)

Interrupt Set Register

			Descript ion
31:6	RO	0x0	reserved

			Descript ion
5	RW	0x0	PKA_DONE_ENA Set the interrupt Enable of PKA done 1'b1: enable 1'b0: disable
4	RW	0x0	HASH_DONE_ENA Set the interrupt Enable of hash done 1'b1: enable 1'b0: disable
3	RW	0x0	HRDMA_ERR_ENA Set the interrupt Enable of hash receiving DMA Error 1'b1: enable 1'b0: disable
2	RW	0x0	HRDMA_DONE_ENA Set the interrupt Enable of hash receiving DMA DONE 1'b1: enable 1'b0: disable
1	RW	0x0	BCDMA_ERR_ENA Set the interrupt Enable of block cipher DMA Error 1'b1: enable 1'b0: disable
0	RW	0x0	BCDMA_DONE_ENA Set the interrupt Enable of block cipher DMA DONE 1'b1: enable 1'b0: disable

**CRYPTO\_CTRL**

Address: Operational Base + offset (0x0008)  
 Control Register

			Descript ion
31:16	RW	0x0000	Write_Mask
15:10	RO	0x0	reserved
9	RW	0x0	TRNG_FLUSH FLUSH TRNG Software write 1 to start. When finishes, the core will clear it.
8	RWSC	0x0	TRNG_START Start TRNG Software write 1 to start. When finishes, the core will clear it.
7	RWSC	0x0	PKA_FLUSH Software write 1 to start Flush Process. The process will clear BRFIFO, BTIFIFO, and state machine. Then Software should write 0 to end FLUSH Process
6	RW	0x0	HASH_FLUSH Software write 1 to start Flush Process. The process will clear BRFIFO, BTIFIFO, and state machine. Then Software should write 0 to end FLUSH Process
5	RW	0x0	BLOCK_FLUSH Software write 1 to start Flush Process. The process will clear BRFIFO, BTIFIFO, and state machine. Then Software should write 0 to end FLUSH Process. It must last for at least 20 cycles to clean registers and FSM
4	RWSC	0x0	PKA_START Starts/initializes PKA Software write 1 to start. When finishes, the core will clear it.
3	RWSC	0x0	HASH_START Starts/initializes HASH/PRNG/HMAC Software write 1 to start. When finishes, the core will clear it.

			<b>Description</b>
2	RWSC	0x0	BLOCK_START Starts/initializes Block Cipher Software write 1 to start. When finishes, the core will clear it.
1	RWSC	0x0	TDES_START Starts/initializes TDES Software write 1 to start. When finishes, the core will clear it.
0	RWSC	0x0	AES_START Starts/initializes AES Software write 1 to start. When finishes, the core will clear it. Software can also write 0 to clear it.

**CRYPTO\_CONF**

Address: Operational Base + offset (0x000c)

			<b>Description</b>
31:9	RO	0x0	reserved
8	RW	0x0	HR_ADDR_MODE Hash Receive DMA Address Mode 1'b1: fix 1'b0: increment

			Descript ion
7	RW	0x0	BT_ADDR_MODE Block Transmit DMA Address Mode 1'b1: fix 1'b0: increment
6	RW	0x0	BR_ADDR_MODE Block Receive DMA Address Mode 1'b1: fix 1'b0: increment
5	RW	0x0	Byteswap_HRFIFO If this bit is high, then the data read from the bus is byte-swapped in a word boundary. If this bit is low (default), then the data is handed over to the FIFO without byte-swap. For little endian bus, this bit should be 1'b1.
4	RW	0x0	Byteswap_BTFIFO If this bit is high, then the data read from the bus is byte-swapped in a word boundary. If this bit is low (default), then the data is handed over to the FIFO without byte-swap. For little endian bus, this bit should be 1'b1.
3	RW	0x0	Byteswap_BRFIFO If this bit is high, then the data read from the bus is byte-swapped in a word boundary. If this bit is low (default), then the data is handed over to the FIFO without byte-swap. For little endian bus, this bit should be 1'b1.
2	RW	0x0	DESSEL Specifies the Destination block cipher of FIFO. AES(=0)/DES(=1)

			Descript ion
1:0	RW	0x0	HASHINSEL Specifies the following Data from independent source (0) Data from block cipher input (1) Data from block cipher output (2) Reserved (3)

**CRYPTO\_BRDMAS**

Address: Operational Base + offset (0x0010)  
 Block Receiving DMA Start Address Register

			Descript ion
31:0	RW	0x00000000	STARTADDR Specifies the Start Address of DMA The address should be aligned by 32-bit.

**CRYPTO\_BTDMAS**

Address: Operational Base + offset (0x0014)  
 Block Transmitting DMA Start Address Register

			Description
31:0	RW	0x00000000	<b>STARTADDR</b> Specifies the Start Address of DMA The address needs to be aligned by 32-bit.

**CRYPTO\_BRDMAL**

Address: Operational Base + offset (0x0018)

Block Receiving DMA Length Register

			Description
31:0	RW	0x00000000	<b>LENGTH</b> Specifies the Block length of DMA. The length unit is WORD.

**CRYPTO\_HRDMAS**

Address: Operational Base + offset (0x001c)

Hash Receiving DMA Start Address Register

			Descript ion
31:0	RW	0x00000000	<p>STARTADDR</p> <p>Specifies the Start Address of DMA</p> <p>The address needs to be aligned by 32-bit.</p>

**CRYPTO\_HRDML**

Address: Operational Base + offset (0x0020)

Hash Receiving DMA Length Register

			Descript ion
31:0	RW	0x00000000	<p>LENGTH</p> <p>Specifies the Block length of DMA.</p> <p>The length unit is BYTE.</p>

**CRYPTO\_AES\_CTRL**

Address: Operational Base + offset (0x0080)

AES Control Register

			Descript ion
31:12	RO	0x0	reserved
11	RW	0x0	AES_BitSwap_CNT Change the Big-endian and Little-endian by swapping the byte order. 0 = Disables Counter data byte swap 1 = Enables Counter data byte swap
10	RW	0x0	AES_BitSwap_Key Change the Big-endian and Little-endian by swapping the byte order. 0 = Disables Key byte swap 1 = Enables Key byte swap
9	RW	0x0	AES_BitSwap_IV Change the Big-endian and Little-endian by swapping the byte order. 0 = Disables Initial value byte swap 1 = Enables Initial value byte swap
8	RW	0x0	AES_BitSwap_DO Change the Big-endian and Little-endian by swapping the byte order. 0 = Disables Output data byte swap 1 = Enables Output data byte swap
7	RW	0x0	AES_BitSwap_DI Change the Big-endian and Little-endian by swapping the byte order. 0 = Disables Input data byte swap 1 = Enables Input data byte swap

			Descript ion
6	RW	0x0	<p>AES_KeyChange Specifies the AES key change mode selection signal. When the bit is asserted, it will not do key-expansion function to calculate new sub-key. So it is a faster way, when several times of calculation use the same key. But if the keys are different, asserting this bit will have the wrong result. 0 = Key is not changed 1 = Key is changed</p>
5:4	RW	0x0	<p>AES_ChainMode Specifies AES chain mode selection 00 = ECB mode 01 = CBC mode 10 = CTR mode</p>
3:2	RW	0x0	<p>AES_KeySize Specifies the AES key size selection signal 00 : 128-bit key 01 : 192-bit key 10 : 256-bit key</p>
1	RW	0x0	<p>AES_FifoMode Specify AES Fifo Mode 1'b0: Slave mode 1'b1: fifo mode</p>
0	RW	0x0	<p>AES_Enc Specifies the Encryption/ Decryption mode selection signal 0 : Encryption 1 : Decryption</p>

**CRYPTO\_AES\_STS**

Address: Operational Base + offset (0x0084)  
Status Register

			Descript ion
31:1	RO	0x0	reserved
0	RW	0x0	AES_DONE When AES finish, it will be HIGH, And it will not be LOW until it restart . 1: done 0: not done

**CRYPTO\_AES\_DIN\_0**

Address: Operational Base + offset (0x0088)

AES Input Data 0 Register

			Descript ion
31:0	RW	0x00000000	AES_DIN_0 Specifies AES Input data [127:96].

**CRYPTO\_AES\_DIN\_1**

Address: Operational Base + offset (0x008c)

AES Input Data 1 Register

			Description
31:0	RW	0x00000000	AES_DIN_1 Specifies AES Input data [95:64].

**CRYPTO\_AES\_DIN\_2**

Address: Operational Base + offset (0x0090)

AES Input Data 2 Register

			Description
31:0	RW	0x00000000	AES_DIN_2 Specifies AES Input data [63:32]

**CRYPTO\_AES\_DIN\_3**

Address: Operational Base + offset (0x0094)

AES Input Data 3 Register

			Descript ion
31:0	RW	0x00000000	AES_DIN_3 Specifies AES Input data [31:0]

**CRYPTO\_AES\_DOUT\_0**

Address: Operational Base + offset (0x0098)

AES Output Data 0 Register

			Descript ion
31:0	RO	0x00000000	AES_DOUT_0 Specifies AES Output data [127:96].

**CRYPTO\_AES\_DOUT\_1**

Address: Operational Base + offset (0x009c)

AES Output Data 1 Register

			Descript ion
31:0	RO	0x00000000	AES_DOUT_1 Specifies the Output data [95:64].

**CRYPTO\_AES\_DOUT\_2**

Address: Operational Base + offset (0x00a0)

AES Output Data 2 Register

			Descript ion
31:0	RO	0x00000000	AES_DOUT_2 Specifies AES Output data [63:32].

**CRYPTO\_AES\_DOUT\_3**

Address: Operational Base + offset (0x00a4)

AES Output Data 3 Register

			Description
31:0	RO	0x00000000	AES_DOUT_3 Specifies AES Output data [31:0].

**CRYPTO\_AES\_IV\_0**

Address: Operational Base + offset (0x00a8)

AES IV data 0 Register

			Description
31:0	RW	0x00000000	AES_IV_0 Specifies AES Initialization vector [127:96]

**CRYPTO\_AES\_IV\_1**

Address: Operational Base + offset (0x00ac)

AES IV data 1 Register

			Description
31:0	RW	0x00000000	AES_IV_1 Specifies AES Initialization vector [95:64]

**CRYPTO\_AES\_IV\_2**

Address: Operational Base + offset (0x00b0)

AES IV data 2 Register

			Description
31:0	RW	0x00000000	AES_IV_2 Specifies AES Initialization vector [63:32]

**CRYPTO\_AES\_IV\_3**

Address: Operational Base + offset (0x00b4)

AES IV data 3 Register

			Descript ion
31:0	RW	0x00000000	AES_IV_3 Specifies AES Initialization vector [31:0]

**CRYPTO\_AES\_KEY\_0**

Address: Operational Base + offset (0x00b8)

AES Key data 0 Register

			Descript ion
31:0	RW	0x00000000	AES_KEY_0 Specifies AES key data [255:224]

**CRYPTO\_AES\_KEY\_1**

Address: Operational Base + offset (0x00bc)

AES Key data 1 Register

			Description
31:0	RW	0x00000000	AES_KEY_1 Specifies AES key data [223:192]

**CRYPTO\_AES\_KEY\_2**

Address: Operational Base + offset (0x00c0)

AES Key data 2 Register

			Description
31:0	RW	0x00000000	AES_KEY_2 Specifies AES key data [191:160]

**CRYPTO\_AES\_KEY\_3**

Address: Operational Base + offset (0x00c4)

AES Key data 3 Register

			Description
31:0	RW	0x00000000	AES_KEY_3 Specifies AES key data [159:128]

**CRYPTO\_AES\_KEY\_4**

Address: Operational Base + offset (0x00c8)

AES Key data 4 Register

			Description
31:0	RW	0x00000000	AES_KEY_4 Specifies AES key data [127:96]

**CRYPTO\_AES\_KEY\_5**

Address: Operational Base + offset (0x00cc)

AES Key data 5 Register

			Descript ion
31:0	RW	0x00000000	AES_KEY_5 Specifies the key data [95:64]

**CRYPTO\_AES\_KEY\_6**

Address: Operational Base + offset (0x00d0)

AES Key data 6 Register

			Descript ion
31:0	RW	0x00000000	AES_KEY_6 Specifies AES key data [63:32]

**CRYPTO\_AES\_KEY\_7**

Address: Operational Base + offset (0x00d4)

AES Key data 7 Register

			Description
31:0	RW	0x00000000	AES_KEY_7 Specifies the key data [31:0]

**CRYPTO\_AES\_CNT\_0**

Address: Operational Base + offset (0x00d8)

AES Input Counter 0 Register

			Description
31:0	RW	0x00000000	AES_CNT_0 Specifies AES Input Counter [127:96].

**CRYPTO\_AES\_CNT\_1**

Address: Operational Base + offset (0x00dc)

AES Input Counter 1 Register

			Descript ion
31:0	RW	0x00000000	AES_CNT_1 Specifies AES Input Counter [95:64].

**CRYPTO\_AES\_CNT\_2**

Address: Operational Base + offset (0x00e0)

AES Input Counter 2 Register

			Descript ion
31:0	RW	0x00000000	AES_CNT_2 Specifies AES Input Counter[63:32]

**CRYPTO\_AES\_CNT\_3**

Address: Operational Base + offset (0x00e4)

AES Input Counter 3 Register

			Descript ion
31:0	RW	0x00000000	AES_CNT_3 Specifies AES Input Counter [31:0]

**CRYPTO\_TDES\_CTRL**

Address: Operational Base + offset (0x0100)

TDES Control Register

			Descript ion
31:9	RO	0x0	reserved
8	RW	0x0	TDES_BitSwap_Key 0 = Disables Key byte swap 1 = Enables Key byte swap
7	RW	0x0	TDES_BitSwap_IV 0 = Disables Initial value byte swap 1 = Enables Initial value byte swap
6	RW	0x0	TDES_BitSwap_DO 0 = Disables Output data byte swap 1 = Enables Output data byte swap
5	RW	0x0	TDES_BitSwap_DI 0 = Disables Input data byte swap 1 = Enables Input data byte swap

			Descript ion
4	RW	0x0	TDES_ChainMode Specifies TDES chain mode selection 0 : ECB mode 1 : CBC mode
3	RW	0x0	TDES_EEE Specifies the TDES key mode selection 1'b0 : EDE 1'b1 : EEE
2	RW	0x0	TDES_Select Specify DES or TDES cipher 1'b0 : DES 1'b1 : TDES
1	RW	0x0	TDES_FifoMode Specify TDES Fifo Mode 1'b0: Slave mode 1'b1: Fifo mode
0	RW	0x0	TDES_Enc Specifies the Encryption/ Decryption mode selection signal 0 : Encryption 1 : Decryption

**CRYPTO\_TDES\_STS**

Address: Operational Base + offset (0x0104)  
 Status Register

			Descript ion
31:1	RO	0x0	reserved
0	RW	0x0	TDES_DONE When DES/TDES finishes, it will be HIGH, And it will not be LOW until it restart . 1: done 0: not done

**CRYPTO\_TDES\_DIN\_0**

Address: Operational Base + offset (0x0108)

TDES Input Data 0 Register

			Descript ion
31:0	RW	0x00000000	TDES_DIN_0 Specifies TDES Input data [63:32].

**CRYPTO\_TDES\_DIN\_1**

Address: Operational Base + offset (0x010c)

TDES Input Data 1 Register

			Description
31:0	RW	0x00000000	TDES_DIN_1 Specifies TDES Input data [31:0].

**CRYPTO\_TDES\_DOUT\_0**

Address: Operational Base + offset (0x0110)

TDES Output Data 0 Register

			Description
31:0	RO	0x00000000	TDES_DOUT_0 Specifies TDES Output data [63:32].

**CRYPTO\_TDES\_DOUT\_1**

Address: Operational Base + offset (0x0114)

TDES Output Data 1 Register

			Descript ion
31:0	RO	0x00000000	TDES_DOUT_1 Specifies TDES Output data [31:0].

**CRYPTO\_TDES\_IV\_0**

Address: Operational Base + offset (0x0118)

TDES IV data 0 Register

			Descript ion
31:0	RW	0x00000000	TDES_IV_0 Specifies TDES Initialization vector [63:32]

**CRYPTO\_TDES\_IV\_1**

Address: Operational Base + offset (0x011c)

TDES IV data 1 Register

			Descript ion
31:0	RW	0x00000000	TDES_IV_1 Specifies TDES Initialization vector [31:0]

**CRYPTO\_TDES\_KEY1\_0**

Address: Operational Base + offset (0x0120)

TDES Key1 data 1 Register

			Descript ion
31:0	RW	0x00000000	TDES_KEY1_0 Specifies TDES key1 data [63:32]

**CRYPTO\_TDES\_KEY1\_1**

Address: Operational Base + offset (0x0124)

TDES Key1 data 1 Register

			Descript ion
31:0	RW	0x00000000	TDES_KEY1_1 Specifies TDES key1 data [31:0]

**CRYPTO\_TDES\_KEY2\_0**

Address: Operational Base + offset (0x0128)

TDES Key2 data 0 Register

			Descript ion
31:0	RW	0x00000000	TDES_KEY2_0 Specifies TDES key2 data [63:32]

**CRYPTO\_TDES\_KEY2\_1**

Address: Operational Base + offset (0x012c)

TDES Key2 data 1 Register

			Descript ion
31:0	RW	0x00000000	TDES_KEY_1 Specifies TDES key data [31:0]

**CRYPTO\_TDES\_KEY3\_0**

Address: Operational Base + offset (0x0130)

TDES Key3 data 0 Register

			Descript ion
31:0	RW	0x00000000	TDES_KEY3_0 Specifies TDES key3 data [63:32]

**CRYPTO\_TDES\_KEY3\_1**

Address: Operational Base + offset (0x0134)

TDES Key3 data 1 Register

			Descript ion
31:0	RW	0x00000000	AES_KEY3_1 Specifies TDES key3 data [31:0]

**CRYPTO\_HASH\_CTRL**

Address: Operational Base + offset (0x0180)

Hash Control Register

			Descript ion
31:4	RO	0x0	reserved
3	RW	0x0	HASH_SWAP_DO Specifies the Byte swap of data output (hash result) 0 = Does not swap (default) 1 = Swap
2	RW	0x0	HASH_SWAP_DI Specifies the Byte swap of data input. 0 = Does not swap (default) 1 = Swap
1:0	RW	0x0	Engine_Selection 2'b00: SHA1_HASH 2'b01: MD5_HASH 2'b10: SHA256_HASH 2'b11: PRNG

**CRYPTO\_HASH\_STS**

Address: Operational Base + offset (0x0184)

Hash Status Register

				Description
31:1	RO	0x0	reserved	
0	RW	0x0	HASH_DONE Hash Done Signal When HASH finishes, it will be HIGH, And it will not be LOW until it restart 1'b1 : done 1'b0 : not done	

**CRYPTO\_HASH\_MSG\_LEN**

Address: Operational Base + offset (0x0188)

Hash Message Len

				Description
31:0	RW	0x00000000	Msg_size Hash total byte.	

**CRYPTO\_HASH\_DOUT\_0**

Address: Operational Base + offset (0x018c)

Hash Result Register 0

			Descript ion
31:0	RO	0x00000000	HASH_RESULT_0 Specifies the HASH Result [159:128]

**CRYPTO\_HASH\_DOUT\_1**

Address: Operational Base + offset (0x0190)

Hash Result Register 1

			Descript ion
31:0	RO	0x00000000	HASH_RESULT_1 Specifies the HASH Result [127:96]

**CRYPTO\_HASH\_DOUT\_2**

Address: Operational Base + offset (0x0194)

Hash Result Register 2

			Descript ion
31:0	RO	0x00000000	HASH_RESULT_2 Specifies the HASH Result [95:64]

**CRYPTO\_HASH\_DOUT\_3**

Address: Operational Base + offset (0x0198)

Hash Result Register 3

			Descript ion
31:0	RO	0x00000000	HASH_RESULT_3 Specifies the HASH Result [63:32]

**CRYPTO\_HASH\_DOUT\_4**

Address: Operational Base + offset (0x019c)

Hash Result Register 4

			Description
31:0	RO	0x00000000	HASH_RESULT_4 Specifies the HASH Result [31:0]

**CRYPTO\_HASH\_DOUT\_5**

Address: Operational Base + offset (0x01a0)

Hash Result Register 4

			Description
31:0	RO	0x00000000	HASH_RESULT_5 Specifies the HASH Result [31:0]

**CRYPTO\_HASH\_DOUT\_6**

Address: Operational Base + offset (0x01a4)

Hash Result Register 6

			Descript ion
31:0	RO	0x00000000	HASH_RESULT_6 Specifies the HASH Result [31:0]

**CRYPTO\_HASH\_DOUT\_7**

Address: Operational Base + offset (0x01a8)

Hash Result Register 7

			Descript ion
31:0	RO	0x00000000	HASH_RESULT_7 Specifies the HASH Result [31:0]

**CRYPTO\_HASH\_SEED\_0**

Address: Operational Base + offset (0x01ac)

PRNG Seed/HMAC Key Register 0

			Descript ion
31:0	RW	0x00000000	HASH_SEED_0 Specifies PRNG Seed/HMAC Key buffer [159:128]

**CRYPTO\_HASH\_SEED\_1**

Address: Operational Base + offset (0x01b0)  
 PRNG Seed/HMAC Key Register 1

			Descript ion
31:0	RW	0x00000000	HASH_SEED_1 Specifies PRNG Seed/HMAC Key buffer [127:96]

**CRYPTO\_HASH\_SEED\_2**

Address: Operational Base + offset (0x01b4)  
 PRNG Seed/HMAC Key Register 2

			Descript ion
31:0	RW	0x00000000	HASH_SEED_2 Specifies PRNG Seed/HMAC Key buffer [95:64]

**CRYPTO\_HASH\_SEED\_3**

Address: Operational Base + offset (0x01b8)

PRNG Seed/HMAC Key Register 3

			Descript ion
31:0	RW	0x00000000	HASH_SEED_3 Specifies PRNG Seed/HMAC Key buffer [63:32]

**CRYPTO\_HASH\_SEED\_4**

Address: Operational Base + offset (0x01bc)

PRNG Seed/HMAC Key Register 4

			Descript ion
31:0	RW	0x00000000	HASH_SEED_4 Specifies PRNG Seed/HMAC Key buffer [31:0]

**CRYPTO\_TRNG\_CTRL**

Address: Operational Base + offset (0x0200)

TRNG Control

			Descript ion
31:17	RO	0x0	reserved
16	RW	0x0	osc_enable osc_ring enable It control the running of osc_ring. And it is indepent of clock and flush signal. This mean that it can run even when clock is gating or flush is asserted as long as osc_enable is asserted. Before it is used to get TRNG result , please run osc_ring first to get enough entropy. 1'b1: Enable ; 1'b0: Disable ;

			Description
15:0	RW	0x0000	period sample period TRNG use clock_crypto to sample ring osc output, this parameter is specify how many cycles to generate 1 bit random data.

**CRYPTO\_TRNG\_DOUT\_0**

Address: Operational Base + offset (0x0204)  
 TRNG Output Data 0

			Description
31:0	RO	0x00000000	TRNG_DOUT_0

**CRYPTO\_TRNG\_DOUT\_1**

Address: Operational Base + offset (0x0208)  
 TRNG Output Data 1

			Description
31:0	RO	0x00000000	TRNG_DOUT_1

**CRYPTO\_TRNG\_DOUT\_2**

Address: Operational Base + offset (0x020c)

TRNG Output Data 2

			Description
31:0	RO	0x00000000	TRNG_DOUT_2

**CRYPTO\_TRNG\_DOUT\_3**

Address: Operational Base + offset (0x0210)

TRNG Output Data 3

			Description

			Description
31:0	RO	0x00000000	TRNG_DOUT_3

**CRYPTO\_TRNG\_DOUT\_4**

Address: Operational Base + offset (0x0214)

TRNG Output Data 4

			Description
31:0	RO	0x00000000	TRNG_DOUT_4

**CRYPTO\_TRNG\_DOUT\_5**

Address: Operational Base + offset (0x0218)

TRNG Output Data 5

			Description

			Description
31:0	RO	0x00000000	TRNG_DOUT_5

**CRYPTO\_TRNG\_DOUT\_6**

Address: Operational Base + offset (0x021c)

TRNG Output Data 6

			Description
31:0	RO	0x00000000	TRNG_DOUT_6

**CRYPTO\_TRNG\_DOUT\_7**

Address: Operational Base + offset (0x0220)

TRNG Output Data 7

			Description

				Description
31:0	RO	0x00000000	TRNG_DOUT_7	

**CRYPTO\_PKA\_CTRL**

Address: Operational Base + offset (0x0280)

PKA Control Register

				Description
31:2	RO	0x0	reserved	
1:0	RW	0x0	block_size PKA Size  It specifies the bits of N in PKA calculation. 2'b00: 512 bit 2'b01: 1024 bit 2'b10: 2048 bit	

**CRYPTO\_PKA\_M**

Address: Operational Base + offset (0x0400)

			Descript ion
31:0	RW	0x00000000	m PKA input or output data. PKA result = $(M \wedge E) \bmod N$ . When it finishes, the result data is in M position. Start from PKA_M base address, and may contain 512/1024/2048 bits data.

**CRYPTO\_PKA\_C**

Address: Operational Base + offset (0x0500)

			Descript ion
31:1	RO	0x0	reserved
0	RW	0x0	c PKA pre-calculate data, $C = 2^{(2n+2)} \bmod N$

**CRYPTO\_PKA\_N**

Address: Operational Base + offset (0x0600)

				Description
31:0	RW	0x00000000	n	PKA modular

**CRYPTO\_PKA\_E**

Address: Operational Base + offset (0x0700)

				Description
31:0	RW	0x00000000	e	PKA exponent.

## 7.4 Application Note

### 7.4.1 Reset a port

CRU.crypto\_srstn\_req is used to do a soft reset to crypto . Please refer to "Chapter CRU" for more details.

### 7.4.2 Overall Performance

Use CRU.crypto\_div\_con to select crypto frequency:  $F_{crypto} = F_{clock} / (div + 1)$ .

Make sure  $F_{crypto}$  do not exceed 150M.

The performance of crypto FIFO mode is list below.

algorithm	cycle	block size	frequency	throughput rate
DES	17	64 bit	100M	<=376 M bps
TDES	51	64 bit	100M	<=125 M bps
AES	11/13/15	128 bit	100M	<=1160/984/853Mbps

SHA-1	81	512 bit	100M	<=632 Mbps
MD5	65	512 bit	100M	<=787 Mbps

### 7.4.3 Usage

#### 1. Symmetric algorithm

DES/3DES, AES are symmetric algorithms. There are two ways of using these algorithms: Slave mode and FIFO mode.

In Slave mode, you can calculate 1 block size of data by starting the engine. Take AES-128 for example, you should

- Program Input 128 bit Data to AES\_DIN\_0~AES\_DIN\_3
- Program Input 128 bit Key to AES\_KEY\_0~AES\_KEY\_3
- Program control mode to AES\_CTRL to run in different mode
- Program CTRL.AES\_START to run
- wait AES\_STS.DONE High
- Read AES\_DOUT\_0 ~ AES\_DOUT\_3 to get result.

In FIFO mode,

- Program the source address to BRDMAS, the destination address to BTDMAS, program the length in word unit to BRDMAL;
- Program Input 128 bit Key to AES\_KEY\_0~AES\_KEY\_3;
- Program control mode to AES\_CTRL to run in different mode;
- Program INTENA to enable interrupt;
- Program CTRL.BLOCK\_START to start;
- wait interrupt asserted;
- Program INTSTS to clear interrupt status;
- Read the destination address which BTDMA points to.

FIFO mode get much higher throughput rate.

#### 2. HASH

HASH is used to get digest of data. Only support FIFO mode.

There are three source: (1) hr\_fifo; (2) br\_fifo; (3) bt\_fifo.

Take hr\_fifo for example

Program CTRL.HASH\_FLUSH 1'b1 to clear, wait several cycle ( $\geq 10$  cycles), and Program CTRL.HASH\_FLUSH 1'b0

Program data source address to HRDMAS, program 1 time data length in word unit to HRDMAL, program total length in byte unit to HASH\_MSG\_LEN

Program HASH\_CTRL to choose algorithm, for example SHA-256

Program INTENA to enable interrupt;

Program CTRL.HASH\_START 1'b1 to start;

Wait interrupt asserted; Only if HRDMAL length meets can this interrupt be asserted

If you have another section of data to hash, then go to (2), HASH\_MSG\_LEN need not to be programmed;

else go to (8)

wait HASH\_STS.done asserted. Only if Hash\_MSG\_LEN meet can this bit status register asserted.

Read HASH\_DOUT\_0 ~ HASH\_DOUT\_7 to get result.

#### 3. Asymmetric Algorithm

Support 512/1024/2048 bit RSA calculation. It provide the big number calculation. Result = ME mod N

Program CTRL.PKA\_FLUSH 1'b1 to flush RSA module;

Wait CTRL.PKA\_FLUSH to be LOW. It is self-cleared;

Program input\_data(M) to PKA\_M; Program pre\_caculated C to PKA\_C; Program Key(N) to PKA\_N; Program Key(E) to PKA\_E. C =  $2(2n+2) \bmod N$ . n is the required bit of N. For example 2048 bit N, n = 2048;

Program PKA\_CTRL to select RSA size: 512/1024/2048

Program INTENA to enable interrupt;

Program CTRL.PKA\_START to start;

Wait interrupt asserted.

Read PKA\_M to get results.

Rockchip Confidential